

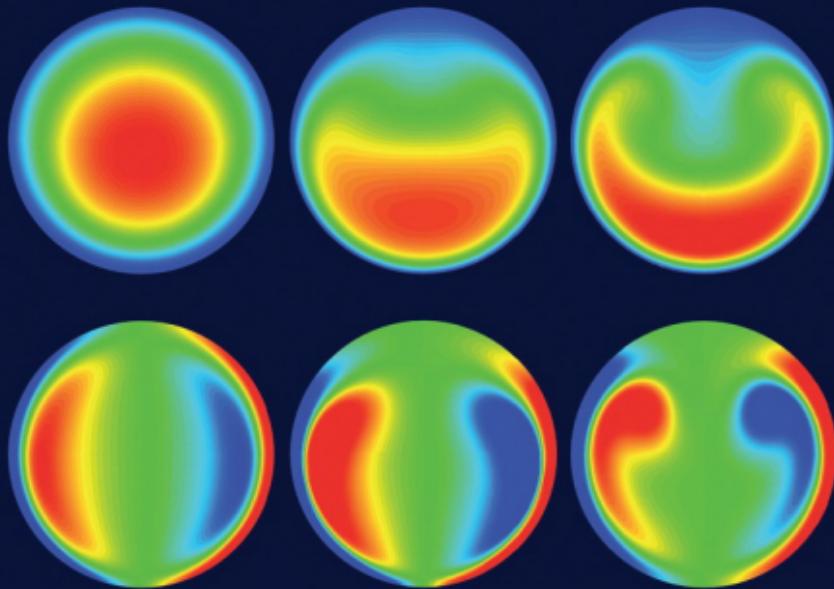
Volume 2

Numerical Models for Differential Problems

Alfio Quarteroni

MS&A

Modeling, Simulation & Applications



Springer

To Fulvia, Silvia and Marzia

MS&A

Series Editors:

Alfio Quarteroni (*Editor-in-Chief*) • Tom Hou • Claude Le Bris • Anthony T. Patera • Enrique Zuazua

Alfio Quarteroni

Numerical Models for Differential Problems



Springer

ALFIO QUARTERONI

MOX, Department of Mathematics “F. Brioschi”

Politecnico di Milano

Milan, Italy and

CMCS-IACS

Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

Translated by Silvia Quarteroni from the original Italian edition:
A. Quarteroni, Modellistica Numerica per Problemi Differenziali. 4^a ed.,
Springer-Verlag Italia, Milano 2008

Library of Congress Control Number: 2009922766

ISBN 978-88-470-1070-3 Springer Milan Berlin Heidelberg New York
e-ISBN 978-88-470-1071-0 Springer Milan Berlin Heidelberg New York

Springer-Verlag is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Italia, Milan 2009

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in other ways, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the Italian Copyright Law in its current version, and permissions for use must always be obtained from Springer. Violations are liable to prosecution under the Italian Copyright Law.

9 8 7 6 5 4 3 2 1

Typesetting with Latex: PTP-Berlin, Protago-TeX-Production GmbH, Germany

Cover-Design: Francesca Tonon, Milan

Printing and Binding: Grafiche Porpora, Cernusco S/N

Printed in Italy

Springer-Verlag Italia Srl – Via Decembrio 28 – 20137 Milano

Preface

Differential equations (DE) provide the baseline of many mathematical models for real life applications. Seldom, these equations can be solved in “closed” form: the exact solution can rarely be characterized through explicit mathematical formulae that are easily computable. Almost invariably, one has to resort to appropriate numerical methods whose scope is the approximation (or discretization) of the exact differential model and, henceforth, of the exact solution.

This book offers a comprehensive and self-contained presentation of some of the most successful numerical methods for DE, their analysis, and their application to some classes of problems that are commonly encountered in applications.

Most often, we will deal with partial differential equations (PDE), for both steady problems (in multiple space dimensions) and time dependent problems (for either one or several space variables). However, some material will be specifically devoted to ordinary differential equations (ODE) for one-dimensional boundary-value problems, in those cases where this discussion is interesting in itself or relevant to the PDE case.

Our presentation will primarily concern the finite element (FE) method, the most popular discretization technique for engineering design and analysis. However, although to a lesser extent, we will also address other techniques, like finite differences (FD), finite volumes (FV), and spectral methods, as well as other ad-hoc methods for specific types of problems. Occasionally, the interplay between different methods and a comparative assessment of their performance will be addressed.

We also introduce and analyze numerical strategies that are aimed at reducing the computational complexity of differential problems, such as operator splitting and fractional step methods for time discretization, preconditioning, techniques for grid adaptivity, domain decomposition (DD) methods for parallel computing, reduced basis (RB) methods for efficiently solving parametrized PDEs.

Besides the classic linear elliptic, parabolic and hyperbolic equations, we will consider some less simple model problems that arise in various fields of application: linear and nonlinear conservation laws, advection-diffusion equations with dominating advection, Navier-Stokes equations, saddle-point problems, and optimal control problems.

Let us summarize the main contents of the various chapters.

In Chapter 1 we give a short survey of PDEs and their classification. Chapter 2 introduces the main concepts and theoretical results of functional analysis that will extensively be used throughout the book.

Chapter 3 illustrates boundary-value problems for elliptic equations (in one and several dimensions), their weak or variational formulation, the treatment of boundary conditions and the analysis of well posedness. Several examples of physical interest are introduced.

Chapter 4 is a central one. The Galerkin method for the numerical discretization of elliptic boundary-value problems is formulated and analyzed in an abstract functional setting. The Galerkin FE method is then introduced, firstly in one dimension for the reader's convenience, and then in several dimensions. FE spaces and FE interpolation operators are constructed, stability and convergence results proven and several kinds of error estimates are derived. Finally, we present some grid adaptive procedures based on either a priori or a posteriori error estimates.

Chapter 5 illustrates the numerical approximation of parabolic problems: we consider first semi-discrete (continuous in time) Galerkin approximations, then fully-discrete approximations based on FD schemes for time discretization. Stability and convergence are derived for both approaches.

Chapters 6, 7 and 8 are devoted to the algorithmic aspects and practical implementation of FE methods. More specifically, Chapter 6 illustrates the main techniques for grid generation, Chapter 7 surveys the basic algorithms for the solution of ill-conditioned linear algebraic systems that arise from FE approximations, and Chapter 8 presents the main operational phases of a FE code together with a complete working example.

Chapter 9 presents some basic principles behind finite volume methods for the approximation of diffusion-transport-reaction equations. FV methods are commonly used in computational fluid dynamics owing to their intrinsic, built-in conservation properties.

Chapter 10 addresses spectral methods under their multiple declinations (Galerkin, collocation, and the spectral element method), and analyzes their superior properties of accuracy.

Chapter 11 focuses on singularly perturbed elliptic boundary-value problems, in particular diffusion-transport equations and diffusion-reaction equations, with small diffusion. The exact solutions to these kinds of problems may feature steep gradients in tiny subregions of the computational domains, the so-called internal or boundary layers. A great deal of attention is paid to the issue of stabilization techniques that are used to prevent the on-rise of oscillatory numerical solutions. Upwinding techniques are discussed for FD approximations, and their analogy with FE with artificial diffusion is analyzed. Other stabilization approaches are introduced and analyzed in the FE context, yielding to sub-grid generalized Galerkin methods, Petrov-Galerkin methods and the Galerkin Least-Squares method.

Chapters 12, 13 and 14 are centered on the approximation of first order hyperbolic equations. The first chapter focuses on classic FD methods. Stability is investigated using both the energy method and the Von-Neumann analysis. By the latter we can

better investigate the properties of dissipation and dispersion featured by a numerical scheme. Chapter 13 is devoted to spatial approximation by FE methods, including the discontinuous Galerkin (DG) method, and spectral methods. Special emphasis is drawn to characteristic compatibility conditions for the boundary treatment of hyperbolic systems. Chapter 14 offers a very quick introduction to the numerical approximation of nonlinear conservation laws. This subject however is so important that the interested reader is advised to consult the referenced specialized monographs.

Chapter 15 is devoted to Navier-Stokes equations for incompressible flows, their analysis, and their numerical approximation by FE, FV and spectral methods. A general stability and convergence theory is developed for spatial approximation of saddle-point problems, which comprises strategies for stabilization. Several time discretization approaches are then proposed and analyzed, among which we mention finite differences, characteristic methods, fractional step methods and algebraic factorization techniques. Special attention is devoted to the numerical treatment of interfaces in the case of multiphase flows.

Chapter 16 illustrates the subject of optimal control for elliptic PDEs. The problem is firstly formulated at the continuous level, where conditions of optimality are obtained following two different approaches. Then we address the interplay between optimization and numerical approximation. We present several examples of optimal control problems, some of them of elementary character, a few others involving physical processes of applicative relevance.

Chapter 17 regards domain decomposition methods. These techniques are specifically devised for parallel computing and for the treatment of multiphysics PDE problems. Both families of Schwarz (with overlapping subdomains) and Schur (with disjointed subdomains) methods are illustrated. Their convergence properties of optimality (grid invariance) and scalability (subdomain-size invariance) are analyzed. Several examples of domain decomposition preconditioners are provided and tested numerically.

Finally, in Chapter 18 we introduce the reduced basis (RB) method for the efficient solution of PDEs. RB methods allow for rapid and reliable evaluation of input-output relationships in which the output is expressed as a functional of a field variable that is the solution of a parametrized PDE. Parametrized PDEs model several processes that are relevant in applications, such as steady and unsteady heat and mass transfer, acoustics, and solid and fluid mechanics, just to mention a few. The input-parameter vector may characterize either the geometric configuration of the domain, some physical properties, or boundary conditions and source terms. The combination with an efficient a posteriori error estimation, and the splitting between offline and online calculations, are key factors for RB methods to be computationally successful.

Limitation of space and our own experience has resulted in the partial (sometimes total) omission of many important topics that we would have liked to address. This list includes, e.g., the approximation of equations for structural analysis and electromagnetic wave propagation. The reader interested in a thorough treatment of these subjects is addressed to the monographs quoted in the references.

This book is intended primarily for graduate students in Mathematics, Engineering, Physics and Computer Science. However we also think that it could be useful to scientists and engineers interested in scientific computation of differential problems. Each chapter is meant to provide a coherent teaching unit on a specific topic. In particular, the first eight chapters can be regarded as a comprehensive and self-contained textbook on finite elements for elliptic and parabolic PDEs, chapters 9 to 15 as an advanced course on numerical methods for PDEs, and the last three chapters contain more refined and sophisticated topics for the numerical solution of complex PDE problems.

This book has been the basis for graduate-level courses at the Politecnico di Milano and the EPFL (École Polytechnique Fédérale de Lausanne). We would like to take this opportunity to thank the many people (students, colleagues and readers) who have contributed at various stages, and in many different ways, to the preparation of this book and to the improvement of the early drafts. A (far from complete) list includes Luca Dedé, Marco Discacciati, Luca Formaggia, Loredana Gaudio, Paola Gervasio, Stefano Micheletti, Nicola Parolini, Anthony T. Patera, Simona Perotto, Gianluigi Rozza, Fausto Saleri, Benjamin Stamm, Alberto Valli, Alessandro Veneziani, and Christoph Winkelmann.

Special thanks to Luca Paglieri for technical assistance, to Francesca Bonadei from Springer-Verlag for supporting this project since its very first Italian edition, and, last but not least, to Silvia Quarteroni for her translation from Italian.

Milan and Lausanne, February 2009

Alfio Quarteroni

Contents

Preface	V
1 A brief survey on partial differential equations	1
1.1 Definitions and examples	1
1.2 Numerical solution	3
1.3 PDE Classification	5
1.3.1 Quadratic form associated to a PDE	8
1.4 Exercises	9
2 Elements of functional analysis	11
2.1 Functionals and bilinear forms	11
2.2 Differentiation in linear spaces	13
2.3 Elements of distributions	15
2.3.1 Square-integrable functions	17
2.3.2 Derivation in the sense of distributions	18
2.4 Sobolev spaces	20
2.4.1 Regularity of the $H^k(\Omega)$ spaces	21
2.4.2 The $H_0^1(\Omega)$ space	22
2.4.3 Trace operators	23
2.5 The spaces $L^\infty(\Omega)$ and $L^p(\Omega)$, with $1 \leq p < \infty$	24
2.6 Adjoint operators of a linear operator	26
2.7 Spaces of time-dependent functions	27
2.8 Exercises	28
3 Elliptic equations	31
3.1 An elliptic problem example: the Poisson equation	31
3.2 The Poisson problem in the one-dimensional case	32
3.2.1 Homogeneous Dirichlet problem	33
3.2.2 Non-homogeneous Dirichlet problem	39
3.2.3 Neumann Problem	39

3.2.4	Mixed homogeneous problem	40
3.2.5	Mixed (or Robin) boundary conditions.....	40
3.3	The Poisson problem in the two-dimensional case	41
3.3.1	The homogeneous Dirichlet problem	41
3.3.2	Equivalence, in the sense of distributions, between weak and strong form of the Dirichlet problem	43
3.3.3	The problem with mixed, non homogeneous conditions ...	44
3.3.4	Equivalence, in the sense of distributions, between weak and strong form of the Neumann problem	46
3.4	More general elliptic problems	48
3.5	Existence and uniqueness theorem	50
3.6	Adjoint operator and adjoint problem	51
3.7	Exercises	56
4	The Galerkin finite element method for elliptic problems	61
4.1	Approximation via the Galerkin method	61
4.2	Analysis of the Galerkin method	63
4.2.1	Existence and uniqueness	63
4.2.2	Stability	64
4.2.3	Convergence	64
4.3	The finite element method in the one-dimensional case	66
4.3.1	The space X_h^1	67
4.3.2	The space X_h^2	68
4.3.3	The approximation with linear finite elements	71
4.3.4	Interpolation operator and interpolation error	73
4.3.5	Estimate of the finite element error in the H^1 norm.....	75
4.4	Finite elements, simplices and barycentric coordinates	76
4.4.1	An abstract definition of finite element in the Lagrangian case	76
4.4.2	Simplices	78
4.4.3	Barycentric coordinates	78
4.5	The finite element method in the multi-dimensional case	80
4.5.1	Finite element solution of the Poisson problem	82
4.5.2	Conditioning of the stiffness matrix	85
4.5.3	Estimate of the approximation error in the energy norm...	88
4.5.4	Estimate of the approximation error in the L^2 norm	96
4.6	Grid adaptivity	99
4.6.1	A priori adaptivity based on derivatives reconstruction....	100
4.6.2	A posteriori adaptivity	103
4.6.3	Numerical examples of adaptivity	107
4.6.4	A posteriori error estimates in the L^2 norm	110
4.6.5	A posteriori estimates of a functional of the error	112
4.7	Exercises	114

5 Parabolic equations	119
5.1 Weak formulation and its approximation	120
5.2 A priori estimates	123
5.3 Convergence analysis of the semi-discrete problem	126
5.4 Stability analysis of the θ -method	130
5.5 Convergence analysis of the θ -method	134
5.6 Exercises	136
6 Generation of 1D and 2D grids	139
6.1 Grid generation in 1D	139
6.2 Grid of a polygonal domain	142
6.3 Generation of structured grids	144
6.4 Generation of non-structured grids	147
6.4.1 Delaunay triangulation	147
6.4.2 Advancing front technique	151
6.5 Regularization techniques	153
6.5.1 Diagonal exchange	154
6.5.2 Node displacement	155
7 Algorithms for the solution of linear systems	159
7.1 Direct methods	159
7.2 Iterative methods	162
7.2.1 Classical iterative methods	162
7.2.2 Gradient and conjugate gradient methods	164
7.2.3 Krylov subspace methods	167
8 Elements of finite element programming	173
8.1 Operational phases of a finite element code	174
8.1.1 Code in a nutshell	176
8.2 Numerical computation of integrals	177
8.2.1 Numerical integration using barycentric coordinates	179
8.3 Storage of sparse matrices	182
8.4 Assembly phase	186
8.4.1 Coding geometrical information	188
8.4.2 Coding of functional information	192
8.4.3 Mapping between reference and physical element	193
8.4.4 Construction of local and global systems	197
8.4.5 Boundary conditions prescription	201
8.5 Integration in time	204
8.6 A complete example	207
9 The finite volume method	217
9.1 Some basic principles	218
9.2 Construction of control volumes for vertex-centered schemes	220
9.3 Discretization of a diffusion-transport-reaction problem	223

9.4	Analysis of the finite volume approximation	225
9.5	Implementation of boundary conditions	226
10	Spectral methods	227
10.1	The spectral Galerkin method for elliptic problems	227
10.2	Orthogonal polynomials and Gaussian numerical integration	231
10.2.1	Orthogonal Legendre polynomials	231
10.2.2	Gaussian integration	234
10.2.3	Gauss-Legendre-Lobatto formulae	235
10.3	G-NI methods in one dimension	237
10.3.1	Algebraic interpretation of the G-NI method	239
10.3.2	Conditioning of the stiffness matrix in the G-NI method	241
10.3.3	Equivalence between G-NI and collocation methods	242
10.3.4	G-NI for parabolic equations	245
10.4	Generalization to the two-dimensional case	247
10.4.1	Convergence of the G-NI method	250
10.5	G-NI and SEM-NI methods for a one-dimensional model problem	257
10.5.1	The G-NI method	257
10.5.2	The SEM-NI method	261
10.6	Spectral methods on triangles and tetrahedra	264
10.7	Exercises	268
11	Diffusion-transport-reaction equations	271
11.1	Weak problem formulation	271
11.2	Analysis of a one-dimensional diffusion-transport problem	274
11.3	Analysis of a one-dimensional diffusion-reaction problem	278
11.4	Finite elements and finite differences (FD)	280
11.5	The mass-lumping technique	281
11.6	Decentered FD schemes and artificial diffusion	284
11.7	Eigenvalues of the diffusion-transport equation	286
11.8	Stabilization methods	289
11.8.1	Artificial diffusion and decentered finite element schemes	289
11.8.2	The Petrov-Galerkin method	292
11.8.3	The artificial diffusion and streamline-diffusion methods in the two-dimensional case	292
11.8.4	Consistence and truncation error for the Galerkin and generalized Galerkin methods	294
11.8.5	Symmetric and skew-symmetric part of an operator	295
11.8.6	Strongly consistent methods (GLS, SUPG)	296
11.8.7	Analysis of the GLS method	298
11.8.8	Stabilization through bubble functions	304
11.9	Some numerical tests	306
11.10	An example of goal-oriented adaptivity	307
11.11	Exercises	309

12 Finite differences for hyperbolic equations	313
12.1 A scalar transport problem	313
12.1.1 An a priori estimate	315
12.2 Systems of linear hyperbolic equations	317
12.2.1 The wave equation	319
12.3 The finite difference method	320
12.3.1 Discretization of the scalar equation	321
12.3.2 Discretization of linear hyperbolic systems	323
12.3.3 Boundary treatment	324
12.4 Analysis of the finite difference methods	324
12.4.1 Consistency and convergence	324
12.4.2 Stability	325
12.4.3 Von Neumann analysis and amplification coefficients	330
12.4.4 Dissipation and dispersion	335
12.5 Equivalent equations	337
12.5.1 The upwind scheme case	337
12.5.2 The Lax-Friedrichs and Lax-Wendroff case	341
12.5.3 On the meaning of coefficients in equivalent equations	342
12.5.4 Equivalent equations and error analysis	342
12.6 Exercises	343
13 Finite elements and spectral methods for hyperbolic equations	345
13.1 Temporal discretization	345
13.1.1 The forward and backward Euler schemes	345
13.1.2 The upwind, Lax-Friedrichs and Lax-Wendroff schemes	347
13.2 Taylor-Galerkin schemes	350
13.3 The multi-dimensional case	356
13.3.1 Semi-discretization: strong and weak treatment of the boundary conditions	356
13.3.2 Temporal discretization	359
13.4 Discontinuous finite elements	362
13.4.1 The one-dimensional case	362
13.4.2 The multi-dimensional case	367
13.5 Approximation using spectral methods	370
13.5.1 The G-NI method in a single interval	370
13.5.2 The DG-SEM-NI method	374
13.6 Numerical treatment of boundary conditions for hyperbolic systems	376
13.6.1 Weak treatment of boundary conditions	379
13.7 Exercises	382
14 Nonlinear hyperbolic problems	383
14.1 Scalar equations	383
14.2 Finite difference approximation	388
14.3 Approximation by discontinuous finite elements	389
14.4 Nonlinear hyperbolic systems	397

15	Navier-Stokes equations	401
15.1	Weak formulation of Navier-Stokes equations	403
15.2	Stokes equations and their approximation	407
15.3	Saddle-point problems	411
15.3.1	Problem formulation	411
15.3.2	Problem analysis	412
15.3.3	Galerkin approximation, stability and convergence analysis	416
15.4	Algebraic formulation of the Stokes problem	420
15.5	An example of stabilized problem	424
15.6	A numerical example	426
15.7	Time discretization of Navier-Stokes equations	427
15.7.1	Finite difference methods	429
15.7.2	Characteristics (or Lagrangian) methods	430
15.7.3	Fractional step methods	431
15.8	Algebraic factorization methods and preconditioners for saddle-point systems	435
15.9	Free surface flow problems	440
15.9.1	Navier-Stokes equations with variable density and viscosity	441
15.9.2	Boundary conditions	443
15.9.3	Application to free surface flows	444
15.10	Interface evolution modeling	445
15.10.1	Explicit interface descriptions	445
15.10.2	Implicit interface descriptions	446
15.11	Finite volume approximation	450
15.12	Exercises	453
16	Optimal control of partial differential equations	457
16.1	Definition of optimal control problems	457
16.2	A control problem for linear systems	459
16.3	Some examples of optimal control problems for the Laplace equation	460
16.4	On the minimization of linear functionals	461
16.5	The theory of optimal control for elliptic problems	464
16.6	Some examples of optimal control problems	468
16.6.1	A Dirichlet problem with distributed control	468
16.6.2	A Neumann problem with distributed control	469
16.6.3	A Neumann problem with boundary control	470
16.7	Numerical tests	470
16.8	Lagrangian formulation of control problems	476
16.8.1	Constrained optimization in \mathbb{R}^n	476
16.8.2	The solution approach based on the Lagrangian	477
16.9	Iterative solution of the optimal control problem	480
16.10	Numerical examples	484
16.10.1	Heat dissipation by a thermal fin	485

16.10.2	Thermal pollution in a river	487
16.11	A few considerations about observability and controllability	489
16.12	Two alternative paradigms for numerical approximation	490
16.13	A numerical approximation of an optimal control problem for advection-diffusion equations	492
16.13.1	The strategies “optimize–then–discretize” and “discretize–then–optimize”	494
16.13.2	A posteriori error estimates	495
16.13.3	A test problem on control of pollutant emission	497
16.14	Exercises	499
17	Domain decomposition methods	501
17.1	Three classical iterative DD methods	502
17.1.1	Schwarz method	502
17.1.2	Dirichlet-Neumann method	504
17.1.3	Neumann-Neumann algorithm	506
17.1.4	Robin-Robin algorithm	506
17.2	Multi-domain formulation of Poisson problem and interface conditions	507
17.2.1	The Steklov-Poincaré operator	507
17.2.2	Equivalence between Dirichlet-Neumann and Richardson methods	509
17.3	Multidomain formulation of the finite element approximation of the Poisson problem	512
17.3.1	The Schur complement	514
17.3.2	The discrete Steklov-Poincaré operator	515
17.3.3	Equivalence between Dirichlet-Neumann and Richardson methods in the discrete case	518
17.4	Generalization to the case of many subdomains	519
17.4.1	Some numerical results	521
17.5	DD preconditioners in case of many subdomains	523
17.5.1	Jacobi preconditioner	524
17.5.2	Bramble-Pasciak-Schatz preconditioner	526
17.5.3	Neumann-Neumann preconditioner	526
17.6	Schwarz iterative methods	530
17.6.1	Algebraic form of Schwarz method for finite element discretizations	531
17.6.2	Schwarz preconditioners	533
17.6.3	Two-level Schwarz preconditioners	536
17.7	An abstract convergence result	539
17.8	Interface conditions for other differential problems	540
17.9	Exercises	544

18 Reduced basis approximation for parametrized partial differential equations	547
18.1 Elliptic coercive parametric PDEs	548
18.1.1 An illustrative example	550
18.2 Geometric parametrization	551
18.2.1 Affine geometry precondition	551
18.2.2 Affine mappings: single subdomain	553
18.2.3 Piecewise affine mappings: multiple subdomains	556
18.2.4 Bilinear forms	557
18.2.5 A second illustrative example	560
18.3 The reduced basis method	562
18.3.1 Reduced basis approximation and spaces	562
18.3.2 Sampling strategies	566
18.4 Convergence of RB approximations	567
18.4.1 A priori convergence theory: single parameter case: $P = 1$.	567
18.4.2 Convergence: $P > 1$	568
18.5 A posteriori error estimation	572
18.5.1 Preliminaries	573
18.5.2 Error bounds	573
18.5.3 Offline-online computational procedure	574
18.6 Historical perspective, background and extensions	576
18.7 Exercises	577
References	581
Index	595

A brief survey on partial differential equations

The purpose of this chapter is to recall the basic concepts related to partial differential equations (PDE, in short). For a wider coverage see [RR04], [Eva98], [LM68], [Sal08].

1.1 Definitions and examples

Partial differential equations are differential equations containing derivatives of the unknown function with respect to several variables (temporal or spatial). In particular, if we denote by u the unknown function in the $d + 1$ independent variables $\mathbf{x} = (x_1, \dots, x_d)^T$ and t , we denote by

$$\mathcal{P}(u, g) = F\left(\mathbf{x}, t, u, \frac{\partial u}{\partial t}, \frac{\partial u}{\partial x_1}, \dots, \frac{\partial u}{\partial x_d}, \dots, \frac{\partial^{p_1+\dots+p_d+p_t} u}{\partial x_1^{p_1} \dots \partial x_d^{p_d} \partial t^{p_t}}, g\right) = 0 \quad (1.1)$$

a generic PDE, g being the set of data on which the PDE depends, while $p_1, \dots, p_d, p_t \in \mathbb{N}$.

We say that (1.1) is of *order* q if q is the maximal order of the partial derivatives appearing in the equation, i.e. the maximum value taken by the integer $p_1 + p_2 + \dots + p_d + p_t$.

If (1.1) depends linearly on the unknown u and on its derivatives, the equation is said to be *linear*. In the particular case where the derivatives having maximal order only appear linearly (with coefficients which may depend on lower-order derivatives), the equation is said to be *quasi-linear*. It is said to be *semi-linear* when it is quasi-linear and the coefficients of the maximal order derivatives only depend on \mathbf{x} and t , and not on the solution u . Finally, if the equation contains no terms which are independent of the unknown function u , the PDE is said to be *homogeneous*.

We list below some examples of frequently encountered PDEs in the applied sciences.

Example 1.1 A first-order linear equation is the *transport* (or *advection*) *equation*

$$\frac{\partial u}{\partial t} + \nabla \cdot (\beta u) = 0, \quad (1.2)$$

having denoted by

$$\nabla \cdot \mathbf{v} = \operatorname{div}(\mathbf{v}) = \sum_{i=1}^d \frac{\partial v_i}{\partial x_i}, \quad \mathbf{v} = (v_1, \dots, v_d)^T,$$

the *divergence operator*. Integrated on a region $\Omega \subset \mathbb{R}^d$, (1.2) expresses the mass conservation of a material system (a continuous media) occupying the region Ω . The u variable is the system's density, while $\beta(\mathbf{x}, t)$ is the velocity of a particle in the system that occupies position \mathbf{x} at time t . ■

Example 1.2 Linear second order equations include:

the *potential equation*

$$-\Delta u = f, \quad (1.3)$$

that describes the diffusion of a fluid in a homogeneous and isotropic region $\Omega \subset \mathbb{R}^d$, but also the vertical displacement of an elastic membrane;

the *heat* (or *diffusion*) *equation*

$$\frac{\partial u}{\partial t} - \Delta u = f; \quad (1.4)$$

the *wave equation*

$$\frac{\partial^2 u}{\partial t^2} - \Delta u = 0. \quad (1.5)$$

We have denoted by

$$\Delta u = \sum_{i=1}^d \frac{\partial^2 u}{\partial x_i^2} \quad (1.6)$$

the *Laplace operator* (*Laplacian*). ■

Example 1.3 An example of a quasi-linear first-order equation is the *Burgers equation*

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x_1} = 0,$$

while its variant obtained by adding a second-order perturbation

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x_1} = \epsilon \frac{\partial^2 u}{\partial x_1^2}, \quad \epsilon > 0,$$

is an example of a semi-linear equation.

Another second-order, non-linear equation, is

$$\left(\frac{\partial^2 u}{\partial x_1^2} \right)^2 + \left(\frac{\partial^2 u}{\partial x_2^2} \right)^2 = f.$$

■

A function $u = u(x_1, \dots, x_d, t)$ is said to be a *solution* (or a *particular integral*) of (1.1) if it makes (1.1) an identity, once it is replaced in (1.1) together with all of its derivatives. The set of all solutions of (1.1) is called the *general integral* of (1.1).

Example 1.4 The transport equation in the one-dimensional case,

$$\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x_1} = 0, \quad (1.7)$$

admits a general integral of the form $u = w(x_1 + t)$, w being a sufficiently regular arbitrary function (see Exercise 2). Similarly, the one-dimensional wave equation

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x_1^2} = 0 \quad (1.8)$$

admits as a general integral

$$u(x_1, t) = w_1(x_1 + t) + w_2(x_1 - t),$$

w_1 and w_2 being two sufficiently regular arbitrary functions (see Exercise 3). ■

Example 1.5 Let us consider the one-dimensional heat equation

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x_1^2} = 0,$$

for $0 < x < 1$ and $t > 0$, with boundary conditions

$$u(0, t) = u(1, t) = 0, \quad t > 0$$

and initial condition $u|_{t=0} = u_0$. Its solution is

$$u(x_1, t) = \sum_{j=1}^{\infty} u_{0,j} e^{-(j\pi)^2 t} \sin(j\pi x_1),$$

where $u_0 = u|_{t=0}$ is the initial datum and

$$u_{0,j} = 2 \int_0^1 u_0(x_1) \sin(j\pi x_1) dx_1, \quad j = 1, 2, \dots$$

■

1.2 Numerical solution

In general, it is not possible to obtain a solution of (1.1) in closed (explicit) form. Indeed, the available analytical integration methods (such as the technique of separation of variables) are of limited applicability. On the other hand, even in the case where a general integral is known, it is not guaranteed that a particular integral may be determined. Indeed, in order to obtain the latter, it will be necessary to assign appropriate conditions on u (and/or its derivatives) at the boundary of the domain Ω .

Besides, from the examples provided it appears as evident that the general integral depends on a number of *arbitrary functions* (and not on arbitrary *constants*, as it happens for ordinary differential equations), so that the imposition of the boundary conditions will result in the solution of mathematical problems that are generally rather involved.

Thus, from a theoretical point of view, the analysis of a given PDE is often bound to investigating *existence*, *uniqueness*, and, possibly, *regularity* of its solutions, but lacks practical tools for their actual determination.

It follows that it is extremely important to have *numerical methods* at one's disposal, that allow to construct an approximation u_N of the exact solution u and to evaluate (in some suitable norm) the error $u_N - u$ committed when substituting to the exact solution u the approximate solution u_N . In general, $N \geq 1$ is a positive integer that denotes the (finite) dimension of the approximate problem. Schematically, we will obtain the following situation:

$$\begin{array}{ccc} \mathcal{P}(u, g) = 0 & & \text{Exact PDE} \\ \downarrow & & [\text{numerical methods}] \\ \mathcal{P}_N(u_N, g_N) = 0 & & \text{Approximate PDE.} \end{array}$$

We have denoted by g_N an approximation of the set of data g on which the PDE depends, and with \mathcal{P}_N the new functional relation characterizing the approximated problem. For simplicity, one can set $u = u(g)$ and $u_N = u_N(g_N)$.

We will present several numerical methods starting from Chap. 4. Here, we only recall their main features. A numerical method is *convergent* if

$$\|u - u_N\| \rightarrow 0 \quad \text{as } N \rightarrow \infty$$

for a given norm. More precisely, we have convergence if and only if

$$\forall \varepsilon > 0, \exists N_0 = N_0(\varepsilon) > 0, \exists \delta = \delta(N_0, \varepsilon) : \forall N > N_0, \forall g_N \text{ s.t. } \|g - g_N\| < \delta,$$

$$\|u(g) - u_N(g_N)\| \leq \varepsilon.$$

(The norm used for the data is not necessarily the same as that used for the solutions.) A direct verification of the convergence of a numerical method may not be easy. A verification of its consistency and stability properties is recommendable instead. A numerical method is said to be *consistent* if

$$\mathcal{P}_N(u, g) \rightarrow 0 \quad \text{as } N \rightarrow \infty, \tag{1.9}$$

and *strongly consistent* (or *fully consistent*) if

$$\mathcal{P}_N(u, g) = 0 \quad \forall N \geq 1. \tag{1.10}$$

Notice that (1.9) can be equivalently formulated as

$$\mathcal{P}_N(u, g) - \mathcal{P}(u, g) \rightarrow 0 \quad \text{as } N \rightarrow \infty.$$

This expresses the property that \mathcal{P}_N (the approximated PDE) “tends” to \mathcal{P} (the exact one) for $N \rightarrow \infty$. Instead, we say that a numerical method is *stable* if to small perturbations to the data correspond small perturbations to the solution. More precisely,

$$\forall \varepsilon > 0, \exists \delta = \delta(\varepsilon) > 0 : \forall \delta g_N : \|\delta g_N\| < \delta \Rightarrow \|\delta u_N\| \leq \varepsilon, \forall N \geq 1;$$

$u_N + \delta u_N$ being the solution of the *perturbed problem*

$$\mathcal{P}_N(u_N + \delta u_N, g_N + \delta g_N) = 0.$$

(See also [QSS07, Chap. 2] for an in-depth coverage.)

The fundamental result, known as the Lax-Richtmyer *equivalence theorem*, finally guarantees that

Theorem 1.1 *If a method is consistent, then it is convergent if and only if it is stable.*

Other important properties will obviously influence the choice of a numerical method, such as its *convergence rate* (i.e. the order with respect to $1/N$ with which the error tends to zero) and its *computational cost*, that is the computation time and memory required to implement such method on the computer.

1.3 PDE Classification

Partial differential equations can be classified into three different families: *elliptic*, *parabolic* and *hyperbolic* equations, for each of which appropriate specific numerical methods will be considered. For the sake of brevity, here we will limit ourselves to the case of a linear second-order PDE, with constant coefficients, of the form $Lu = G$,

$$Lu = A \frac{\partial^2 u}{\partial x_1^2} + B \frac{\partial^2 u}{\partial x_1 \partial x_2} + C \frac{\partial^2 u}{\partial x_2^2} + D \frac{\partial u}{\partial x_1} + E \frac{\partial u}{\partial x_2} + Fu, \quad (1.11)$$

with assigned function G and $A, B, C, D, E, F \in \mathbb{R}$. (Notice that any of the x_i variables could represent the temporal variable.) In that case, the classification is carried on based on the sign of the *discriminant*, $\Delta = B^2 - 4AC$. In particular:

- if $\Delta < 0$ the equation is said to be *elliptic*,
- if $\Delta = 0$ the equation is said to be *parabolic*,
- if $\Delta > 0$ the equation is said to be *hyperbolic*.

Example 1.6 The wave equation (1.8) is hyperbolic, while the potential equation (1.3) is elliptic. An example of a parabolic problem is given by the heat equation (1.4), but also by the following *diffusion-transport equation*

$$\frac{\partial u}{\partial t} - \mu \Delta u + \nabla \cdot (\beta u) = 0$$

where the constant $\mu > 0$ and the vector field β are given. ■

The criterion introduced above makes the classification depend on the sole coefficients of the maximal-order derivatives and is justified via the following argument. As the reader will recall, the quadratic algebraic equation

$$Ax_1^2 + Bx_1x_2 + Cx_2^2 + Dx_1 + Ex_2 + F = G,$$

represents a hyperbola, a parabola or an ellipse in the cartesian plane (x_1, x_2) depending whether Δ is positive, null or negative. This parallel motivates the name assigned to the three classes of partial derivative operators.

Let us investigate the difference between the three classes more attentively. Let us suppose, without this being restrictive, that D, E, F and G be null. We look for a change of variables of the form

$$\xi = \alpha x_2 + \beta x_1, \quad \eta = \gamma x_2 + \delta x_1, \quad (1.12)$$

with α, β, γ and δ to be chosen so that Lu becomes a multiple of $\partial^2 u / \partial \xi \partial \eta$. Since

$$\begin{aligned} Lu &= (A\beta^2 + B\alpha\beta + C\alpha^2) \frac{\partial^2 u}{\partial \xi^2} \\ &\quad + (2A\beta\delta + B(\alpha\delta + \beta\gamma) + 2C\alpha\gamma) \frac{\partial^2 u}{\partial \xi \partial \eta} + (A\delta^2 + B\gamma\delta + C\gamma^2) \frac{\partial^2 u}{\partial \eta^2}, \end{aligned} \quad (1.13)$$

we need to require that

$$A\beta^2 + B\alpha\beta + C\alpha^2 = 0, \quad A\delta^2 + B\gamma\delta + C\gamma^2 = 0. \quad (1.14)$$

If $A = C = 0$, the trivial trasformation $\xi = x_2, \eta = x_1$ (for instance) provides Lu in the desired form.

Let us then suppose that A or C be not null. It is not restrictive to suppose $A \neq 0$. Then, if $\alpha \neq 0$ and $\gamma \neq 0$, we can divide the first equation of (1.14) by α^2 and the second one by γ^2 . We find two identical quadratic equations for the fractions β/α and δ/γ . By solving them, we have

$$\frac{\beta}{\alpha} = \frac{1}{2A} \left[-B \pm \sqrt{\Delta} \right], \quad \frac{\delta}{\gamma} = \frac{1}{2A} \left[-B \pm \sqrt{\Delta} \right].$$

In order for the transformation (1.12) to be non-singular, the quotients β/α and δ/γ must be different. We must therefore take the positive sign in one case, and the negative sign in the other. Moreover, we must assume $\Delta > 0$. If Δ were indeed null, the two fractions would still be coincident, while if Δ were negative none of the two fractions could be real. To conclude, we can take the following values as coefficients of transformation (1.12):

$$\alpha = \gamma = 2A, \quad \beta = -B + \sqrt{\Delta}, \quad \delta = -B - \sqrt{\Delta}.$$

Correspondingly, (1.12) becomes

$$\xi = 2Ax_2 + [-B + \sqrt{\Delta}] x_1, \quad \eta = 2Ax_2 + [-B - \sqrt{\Delta}] x_1,$$

and, after the transformation, the original differential problem $Lu = 0$ becomes

$$Lu = -4A\Delta \frac{\partial^2 u}{\partial \xi \partial \eta} = 0. \quad (1.15)$$

(For ease of notation, we still denote by u the transformed solution and by L the transformed differential operator.) The case $A = 0$ and $C \neq 0$ can be treated in a similar way by taking $\xi = x_1$, $\eta = x_2 - (C/B)x_1$.

To conclude, the original term Lu can become a multiple of $\partial^2 u / \partial \xi \partial \eta$ based on the transformation (1.12) if and only if $\Delta > 0$ and in such case, as we have anticipated, the problem is said to be *hyperbolic*. It is easy to verify that the general solution of problem (1.15) is

$$u = p(\xi) + q(\eta),$$

p and q being arbitrary differentiable functions in one variable. The lines $\xi = \text{constant}$ and $\eta = \text{constant}$ are said to be the *characteristics* of L and are characterized by the fact that on these lines, functions p and q , respectively, remain constant. In particular, possible discontinuities of the solution u propagate along the characteristic lines (this will be shown in more detail in Chap. 12). Indeed, if $A \neq 0$, by identifying x_1 with t and x_2 with x , the transformation

$$x' = x - \frac{B}{2A}t, \quad t' = t,$$

transforms the hyperbolic operator L s.t.

$$Lu = A \frac{\partial^2 u}{\partial t^2} + B \frac{\partial^2 u}{\partial t \partial x} + C \frac{\partial^2 u}{\partial x^2}$$

in a multiple of the wave operator L s.t.

$$Lu = \frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2}, \quad \text{with } c^2 = \Delta/4A^2.$$

The latter is the wave operator in a coordinate system moving with velocity $-B/2A$. The characteristic lines of the wave operator are the lines verifying

$$\left(\frac{dt}{dx} \right)^2 = \frac{1}{c^2},$$

that is

$$\frac{dt}{dx} = \frac{1}{c} \quad \text{and} \quad \frac{dt}{dx} = -\frac{1}{c}.$$

When $\Delta = 0$, as previously stated L is *parabolic*. In this case there exists only one value of β/α in correspondence of which the coefficient of $\partial^2 u / \partial \xi^2$ in (1.13) becomes zero: precisely, $\beta/\alpha = -B/(2A)$. On the other hand, since $B/(2A) = 2C/B$, this

choice also implies that the coefficient of $\partial^2 u / \partial \xi \partial \eta$ becomes zero. Hence, the change of variables

$$\xi = 2Ax_2 - Bx_1, \quad \eta = x_1,$$

transforms the original problem $Lu = 0$ into the following

$$Lu = A \frac{\partial^2 u}{\partial \eta^2} = 0,$$

the general solution of which has the form

$$u = p(\xi) + \eta q(\xi).$$

A parabolic operator therefore has only one family of characteristics, precisely $\xi = \text{constant}$. The discontinuities in the derivatives of u propagate along such characteristic lines.

Finally, if $\Delta < 0$ (*elliptic* operators) there does not exist any choice of β/α or δ/γ that makes the coefficients $\partial^2 u / \partial \xi^2$ and $\partial^2 u / \partial \eta^2$ null. However, the transformation

$$\xi = \frac{2Ax_2 - Bx_1}{\sqrt{-\Delta}}, \quad \eta = x_1,$$

transforms $Lu = 0$ into

$$Lu = A \left(\frac{\partial^2 u}{\partial \xi^2} + \frac{\partial^2 u}{\partial \eta^2} \right) = 0,$$

i.e. a multiple of the potential equation. The latter has therefore no family of characteristic lines.

1.3.1 Quadratic form associated to a PDE

We can associate to equation (1.11) the so-called principal symbol S^p defined by

$$S^p(\mathbf{x}, \mathbf{q}) = -A(\mathbf{x})q_1^2 - B(\mathbf{x})q_1 q_2 - C(\mathbf{x})q_2^2.$$

This quadratic form can be represented in matrix form as follows:

$$S^p(\mathbf{x}, \mathbf{q}) = \mathbf{q}^T \begin{bmatrix} -A(\mathbf{x}) & -\frac{1}{2}B(\mathbf{x}) \\ -\frac{1}{2}B(\mathbf{x}) & -C(\mathbf{x}) \end{bmatrix} \mathbf{q}. \quad (1.16)$$

A quadratic form is said to be *definite* if all of the eigenvalues of its associated matrix have the same sign (either positive or negative); it is *indefinite* if the matrix has eigenvalues of both signs; it is *degenerate* if the matrix is singular.

It can then be said that equation (1.11) is elliptic if its quadratic form (1.16) is definite (positive or negative), hyperbolic if it is indefinite, and parabolic if it is degenerate.

The matrices associated to the potential equation (1.3), the (one-dimensional) heat equation (1.4) and the wave equation (1.5) are given respectively by

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

and are positive definite in the first case, singular in the second case, and indefinite in the third case.

1.4 Exercises

1. Classify the following equations based on their order and linearity:

$$(a) \quad \left[1 + \left(\frac{\partial u}{\partial x_1} \right)^2 \right] \frac{\partial^2 u}{\partial x_2^2} - 2 \frac{\partial u}{\partial x_1} \frac{\partial u}{\partial x_2} \frac{\partial^2 u}{\partial x_1 \partial x_2} + \left[1 + \left(\frac{\partial u}{\partial x_2} \right)^2 \right] \frac{\partial^2 u}{\partial x_1^2} = 0,$$

$$(b) \quad \rho \frac{\partial^2 u}{\partial t^2} + K \frac{\partial^4 u}{\partial x_1^4} = f,$$

$$(c) \quad \left(\frac{\partial u}{\partial x_1} \right)^2 + \left(\frac{\partial u}{\partial x_2} \right)^2 = f.$$

[*Solution:* (a) quasi-linear, second-order; it is Plateau's equation which governs, under appropriate hypotheses, the plane motion of a fluid. The u appearing in the equation is the so-called *kinetic potential*; (b) linear, fourth-order. It is the *vibrating rod* equation, ρ is the rod's density, while K is a positive quantity that depends on the geometrical properties of the rod itself; (c) non-linear, first-order.]

2. Reduce the one-dimensional transport equation (1.7) to an equation of the form $\partial w / \partial y = 0$, having set $y = x_1 - t$ and obtain that $u = w(x_1 + t)$ is a solution of the original equation.

[*Solution:* operate the substitution of variables $z = x_1 + t$, $y = x_1 - t$, $u(x_1, t) = w(y, z)$. In such way $\partial u / \partial x_1 = \partial w / \partial z + \partial w / \partial y$, where $\partial u / \partial t = \partial w / \partial z - \partial w / \partial y$, and thus $-2 \partial w / \partial y = 0$. Note at this point that the equation obtained thereby admits a solution $w(y, z)$ that does not depend on y and, using the original variables, we get $u = w(x_1 + t)$.]

3. Prove that the wave equation

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x_1^2} = 0,$$

with constant c , admits as a solution $u(x_1, t) = w_1(x_1 + ct) + w_2(x_1 - ct)$, w_1 and w_2 being two sufficiently regular arbitrary functions.

[*Solution:* proceed as in Exercise 2, by applying the substitution of variables $y = x_1 + ct$, $z = x_1 - ct$ and setting $u(x_1, t) = w(y, z)$.]

4. Verify that the Kortevég-de-Vries equation

$$\frac{\partial u}{\partial t} + \beta \frac{\partial u}{\partial x_1} + \alpha \frac{\partial^3 u}{\partial x_1^3} = 0,$$

admits a general integral of the form $u = a \cos(kx_1 - \omega t)$ with an appropriate ω to be determined, and a , β and α being assigned constants. This equation describes the position u of a fluid with respect to a reference position, in the presence of long wave propagation.

[*Solution:* the given u satisfies the equation only if $\omega = k\beta - \alpha k^3$.]

5. Consider the equation

$$x_1^2 \frac{\partial^2 u}{\partial x_1^2} - x_2^2 \frac{\partial^2 u}{\partial x_2^2} = 0$$

with $x_1 x_2 \neq 0$. Classify it and determine its characteristic lines.

6. Consider the generic second-order semi-linear differential equation

$$a(x_1, x_2) \frac{\partial^2 u}{\partial x_1^2} + 2b(x_1, x_2) \frac{\partial^2 u}{\partial x_1 \partial x_2} + c(x_1, x_2) \frac{\partial^2 u}{\partial x_2^2} + f(u, \nabla u) = 0,$$

where $\nabla u = \left(\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2} \right)^T$ is the gradient of u . Write the equation of its characteristic lines and deduce from it the classification of the proposed equation, by distinguishing the different cases.

7. Set $r(\mathbf{x}) = |\mathbf{x}| = (x_1^2 + x_2^2)^{1/2}$ and define $u(\mathbf{x}) = \ln(r(\mathbf{x}))$, $\mathbf{x} \in \mathbb{R}^2 \setminus \{\mathbf{0}\}$. Verify that

$$\Delta u(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega,$$

where Ω is any given open set such that $\bar{\Omega} \subset \mathbb{R}^2 \setminus \{\mathbf{0}\}$.

[*Solution:* observe that

$$\frac{\partial^2 u}{\partial x_i^2} = \frac{1}{r^2} \left(1 - \frac{2x_i^2}{r^2} \right), \quad i = 1, 2.$$

2

Elements of functional analysis

In this chapter, we recall a number of concepts used extensively in this textbook: functionals and bilinear forms, distributions, Sobolev spaces, L^p spaces. For a more in-depth reading, the reader can refer to e.g. [Sal08], [Yos74], [Bre86], [LM68], [Ada75].

2.1 Functionals and bilinear forms

Definition 2.1 Given a function space V , we call *functional* on V an operator associating a real number to each element of V

$$F : V \mapsto \mathbb{R}.$$

The functional is often denoted as $F(v) = \langle F, v \rangle$, an expression called *duality* or *crochet*.

A functional is said to be *linear* if it is linear with respect to its argument, that is if

$$F(\lambda v + \mu w) = \lambda F(v) + \mu F(w) \quad \forall \lambda, \mu \in \mathbb{R}, \quad \forall v, w \in V.$$

A linear functional is *bounded* if there is a constant $C > 0$ such that

$$|F(v)| \leq C\|v\|_V \quad \forall v \in V. \tag{2.1}$$

A linear and bounded functional on a Banach space (i.e. a normed and complete space) is also continuous. We then define the space V' , called *dual* of V , as the set of linear and bounded functionals on V , that is

$$V' = \{F : V \mapsto \mathbb{R} \text{ s.t. } F \text{ is linear and bounded}\},$$

and we provide it with the norm $\|\cdot\|_{V'}$ defined as

$$\|F\|_{V'} = \sup_{v \in V \setminus \{0\}} \frac{|F(v)|}{\|v\|_V}. \tag{2.2}$$

The constant C appearing in (2.1) is greater or equal to $\|F\|_{V'}$.

The following theorem, called identification or representation theorem ([Yos74]), holds.

Theorem 2.1 (Riesz representation theorem) *Let H be a Hilbert space, that is a Banach space whose norm is induced by a scalar product $(\cdot, \cdot)_H$. For each linear and bounded functional f on H there exists a unique element $x_f \in H$ such that*

$$f(y) = (y, x_f)_H \quad \forall y \in H, \quad \text{and} \quad \|f\|_{H'} = \|x_f\|_H. \quad (2.3)$$

Conversely, each element $x \in H$ identifies a linear and bounded functional f_x on H such that

$$f_x(y) = (y, x)_H \quad \forall y \in H \quad \text{and} \quad \|f_x\|_{H'} = \|x\|_H. \quad (2.4)$$

If H is a Hilbert space, its dual space H' of linear and bounded functionals on H is a Hilbert space too. Moreover, thanks to Theorem 2.1, there exists a bijective and isometric (i.e. norm-preserving) transformation $f \leftrightarrow x_f$ between H' and H thanks to which H' and H can be identified as sets (but not as vector spaces). We can denote this transformation as follows:

$$\begin{aligned} \Lambda_H : H &\rightarrow H', & x \rightarrow f_x = \Lambda_H x, \\ \Lambda_H^{-1} : H' &\rightarrow H, & f \rightarrow x_f = \Lambda_H^{-1} f. \end{aligned} \quad (2.5)$$

We now introduce the notion of form.

Definition 2.2 *Given a normed functional space V we call form an application which associates to each pair of elements of V a real number*

$$a : V \times V \mapsto \mathbb{R}.$$

A form is called:

bilinear if it is linear with respect to both its arguments, i.e. if

$$a(\lambda u + \mu w, v) = \lambda a(u, v) + \mu a(w, v) \quad \forall \lambda, \mu \in \mathbb{R}, \forall u, v, w \in V,$$

$$a(u, \lambda w + \mu v) = \lambda a(u, w) + \mu a(u, v) \quad \forall \lambda, \mu \in \mathbb{R}, \forall u, v, w \in V;$$

continuous if there exists a constant $M > 0$ such that

$$|a(u, v)| \leq M \|u\|_V \|v\|_V \quad \forall u, v \in V; \quad (2.6)$$

symmetric if

$$a(u, v) = a(v, u) \quad \forall u, v \in V; \quad (2.7)$$

positive (or positive definite) if

$$a(v, v) > 0 \quad \forall v \in V; \quad (2.8)$$

coercive if there exists a constant $\alpha > 0$ such that

$$a(v, v) \geq \alpha \|v\|_V^2 \quad \forall v \in V. \quad (2.9)$$

Definition 2.3 Let X and Y be two Hilbert spaces. We say that X is contained in Y with continuous injection if there exists a constant C such that $\|w\|_Y \leq C\|w\|_X \forall w \in X$. Moreover X is dense in Y if each element belonging to Y can be obtained as the limit, in the $\|\cdot\|_Y$ norm, of a sequence of elements of X .

Given two Hilbert spaces V and H , such that $V \subset H$, the injection of V in H is continuous and moreover V is dense in H , we have that H is a subspace of V' , the dual of V , and we have

$$V \subset H \simeq H' \subset V'. \quad (2.10)$$

For elliptic problems, the spaces V and H will typically be chosen respectively as $H^1(\Omega)$ (or one of its subspaces, $H_0^1(\Omega)$ or $H_{\Gamma_D}^1(\Omega)$) and $L^2(\Omega)$, see Chap. 3.

For control problems, the space H typically represents the functional space of the forcing term f of the state equation (16.1) and, sometimes, the one where we seek the control u , see Chap. 16.

Definition 2.4 A linear and bounded (hence continuous) operator \mathcal{T} between two functional spaces X and Y , is an isomorphism if it relates biunivocally the elements of the spaces X and Y and moreover its inverse \mathcal{T}^{-1} exists. If also $X \subset Y$ holds, such isomorphism is called canonic.

2.2 Differentiation in linear spaces

In this section, we briefly report the notions of differentiability and differentiation for applications on linear functional spaces; for a further analysis of this topic, as well as an extension of such notions to more general cases, see [KF89].

Let us begin by considering the notion of *strong* (or *Fréchet*) *differential*:

Definition 2.5 Let X and Y be two normed linear spaces and F an application of X in Y , defined on an open set $E \subset X$; such application is called differentiable at $x \in E$ if there exists a linear and bounded operator $L_x : X \rightarrow Y$ such that:

$$\forall \varepsilon > 0, \exists \delta > 0 : \|F(x+h) - F(x) - L_x h\|_Y \leq \varepsilon \|h\|_X \quad \forall h \in X \text{ with } \|h\|_X < \delta.$$

We call the expression $L_x h$ (or $L_x[h]$), which generates an element in Y for each $h \in X$, strong differential (or Fréchet differential) of the application F at $x \in E$; the operator L_x is called strong derivative of the application F at x and is generally denoted as $F'(x)$, that is $F'(x) = L_x$.

From the definition, we deduce that a differentiable application in x is also continuous in x . We report hereafter some properties deriving from this definition:

- if $F(x) = \text{constant}$, then $F'(x)$ is the null operator, that is $L_x[h] = 0, \forall h \in X$;
- the strong derivative of a continuous linear application $F(x)$ is the application itself, that is $F'(x) = F(x)$;
- given two continuous applications F and G of X in Y , if these are differentiable at x_0 , so are the applications $F + G$ and αF , for all $\alpha \in \mathbb{R}$, and we have:

$$(F + G)'(x_0) = F'(x_0) + G'(x_0),$$

$$(\alpha F)'(x_0) = \alpha F'(x_0).$$

Consider now the following definition of *weak* (or *Gâteaux*) *differential*:

Definition 2.6 Let F be an application of X in Y ; we call weak (or Gâteaux) differential of the application F at x the limit:

$$DF(x, h) = \lim_{t \rightarrow 0} \frac{F(x + th) - F(x)}{t} \quad \forall h \in X,$$

where $t \in \mathbb{R}$ and the convergence of the limit must be intended with respect to the norm of the space Y . If the weak differential $DF(x, h)$ is linear (in general it is not), it can be expressed as

$$DF(x, h) = F'_G(x)h \quad \forall h \in X.$$

The linear and bounded operator $F'_G(x)$ is called weak derivative (or Gâteaux derivative) of F .

Moreover, we have

$$F(x + th) - F(x) = tF'_G(x)h + o(t) \quad \forall h \in X,$$

which implies

$$\|F(x + th) - F(x) - tF'_G(x)h\| = o(t) \quad \forall h \in X.$$

Note that if an application F has a strong derivative, then it also admits a weak derivative, coinciding with the strong one; the converse instead is not generally true. However, the following theorem holds (see [KF89]):

Theorem 2.2 *If in a neighborhood $U(x_0)$ of x_0 there exists a weak derivative $F'_G(x)$ of the application F and such derivative is a function of x in such neighborhood, continuous in x_0 , then the strong derivative $F'(x_0)$ for x_0 exists too and coincides with the weak one, that is $F'(x_0) = F'_G(x_0)$.*

2.3 Elements of distributions

In this section, we want to recall the main definitions regarding the theory of distributions and Sobolev spaces, useful for a better comprehension of the subjects introduced in the textbook. For a more in-depth treatment, see, e.g., the monographs [Bre86], [Ada75] and [LM68].

Let Ω be an open set of \mathbb{R}^n and $f : \Omega \mapsto \mathbb{R}$.

Definition 2.7 *By support of a function f we mean the closure of the set where the function itself takes values different from zero*

$$\text{supp } f = \overline{\{\mathbf{x} : f(\mathbf{x}) \neq 0\}}.$$

A function $f : \Omega \mapsto \mathbb{R}$ is said to have a *compact support* in Ω if there exists a compact set¹ $K \subset \Omega$ such that $\text{supp } f \subset K$.

At this point, we can provide the following definition:

Definition 2.8 $\mathcal{D}(\Omega)$ *is the space of infinitely differentiable functions with compact support in Ω , that is*

$$\mathcal{D}(\Omega) = \{f \in C^\infty(\Omega) : \exists K \subset \Omega, \text{ compact} : \text{supp } f \subset K\}.$$

¹ With $\Omega \subset \mathbb{R}^n$, a compact set is a closed and bounded set

We introduce the multi-index notation for the derivatives. Let $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ be an n-uple of non-negative integers (called *multi-index*) and let $f : \Omega \mapsto \mathbb{R}$ be a function defined in $\Omega \subset \mathbb{R}^n$. We will use the following notation

$$D^\alpha f(\mathbf{x}) = \frac{\partial^{|\alpha|} f(\mathbf{x})}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \dots \partial x_n^{\alpha_n}},$$

$|\alpha| = \alpha_1 + \alpha_2 + \dots + \alpha_n$ being the length of the multi-index coinciding with the order of differentiation of f .

In the space $\mathcal{D}(\Omega)$ we can introduce the following notion of convergence:

Definition 2.9 Given a sequence $\{\phi_k\}$ of functions of $\mathcal{D}(\Omega)$ we say that these converge in $\mathcal{D}(\Omega)$ to a function ϕ and we will write $\phi_k \xrightarrow{\mathcal{D}(\Omega)} \phi$ if:

1. the supports of the ϕ_k functions are all contained in a fixed compact K of Ω ;
2. we have uniform convergence of the derivatives of all orders, that is

$$D^\alpha \phi_k \longrightarrow D^\alpha \phi \quad \forall \alpha \in \mathbb{N}^n.$$

We are now able to define the space of distributions on Ω :

Definition 2.10 Let T be a linear transformation from $\mathcal{D}(\Omega)$ into \mathbb{R} and let us denote by $\langle T, \varphi \rangle$ the value taken by T on the element $\varphi \in \mathcal{D}(\Omega)$. We say that T is continuous if

$$\lim_{k \rightarrow \infty} \langle T, \varphi_k \rangle = \langle T, \varphi \rangle$$

where $\{\varphi_k\}_{k=1}^\infty$ is an arbitrary sequence of $\mathcal{D}(\Omega)$ that converges toward $\varphi \in \mathcal{D}(\Omega)$. We call distribution on Ω any linear and continuous transformation T from $\mathcal{D}(\Omega)$ into \mathbb{R} . The space of distributions on Ω is therefore given by the dual space $\mathcal{D}'(\Omega)$ of $\mathcal{D}(\Omega)$.

The action of a distribution $T \in \mathcal{D}'(\Omega)$ on a function $\phi \in \mathcal{D}(\Omega)$ will always be denoted via the identity pairing $\langle T, \phi \rangle$.

Example 2.1 Let \mathbf{a} be a point of the Ω set. The *Dirac delta* relative to point \mathbf{a} is the distribution $\delta_{\mathbf{a}}$ defined by the following relation

$$\langle \delta_{\mathbf{a}}, \phi \rangle = \phi(\mathbf{a}) \quad \forall \phi \in \mathcal{D}(\Omega).$$

■

For another example, see Exercise 4. Also in $\mathcal{D}'(\Omega)$ we introduce a notion of convergence:

Definition 2.11 A sequence of distributions $\{T_n\}$ converges to a distribution T in $\mathcal{D}'(\Omega)$ if we have

$$\lim_{n \rightarrow \infty} \langle T_n, \phi \rangle = \langle T, \phi \rangle \quad \forall \phi \in \mathcal{D}(\Omega).$$

2.3.1 Square-integrable functions

We consider the space of square-integrable functions on $\Omega \subset \mathbb{R}^n$,

$$\mathbf{L}^2(\Omega) = \{f : \Omega \mapsto \mathbb{R} \text{ s.t. } \int_{\Omega} (f(\mathbf{x}))^2 d\Omega < +\infty\}.$$

More precisely, $\mathbf{L}^2(\Omega)$ is a space of *equivalence classes* of measurable functions, the equivalence relation to be intended as follows: v is equivalent to w if and only if v and w are equal almost everywhere, i.e. they differ at most on a subset of Ω with zero measure. The notation “almost everywhere in Ω ” (in short, a.e. in Ω) means exactly “for all the $\mathbf{x} \in \Omega$, except at most a set of points with zero measure”.

The space $\mathbf{L}^2(\Omega)$ is a Hilbert space whose scalar product is

$$(f, g)_{\mathbf{L}^2(\Omega)} = \int_{\Omega} f(\mathbf{x})g(\mathbf{x}) d\Omega.$$

The norm in $\mathbf{L}^2(\Omega)$ is the one induced by this scalar product, i.e.

$$\|f\|_{\mathbf{L}^2(\Omega)} = \sqrt{(f, f)_{\mathbf{L}^2(\Omega)}}.$$

To each function $f \in \mathbf{L}^2(\Omega)$ we associate a distribution $T_f \in \mathcal{D}'(\Omega)$ defined in the following way

$$\langle T_f, \phi \rangle = \int_{\Omega} f(\mathbf{x})\phi(\mathbf{x}) d\Omega \quad \forall \phi \in \mathcal{D}(\Omega).$$

The following result holds:

Lemma 2.1 The space $\mathcal{D}(\Omega)$ is dense in $\mathbf{L}^2(\Omega)$.

Thanks to the latter, it is possible to prove that the correspondence $f \rightarrow T_f$ is injective, thus we can identify $\mathbf{L}^2(\Omega)$ with a subset of $\mathcal{D}'(\Omega)$, writing:

$$\mathbf{L}^2(\Omega) \subset \mathcal{D}'(\Omega).$$

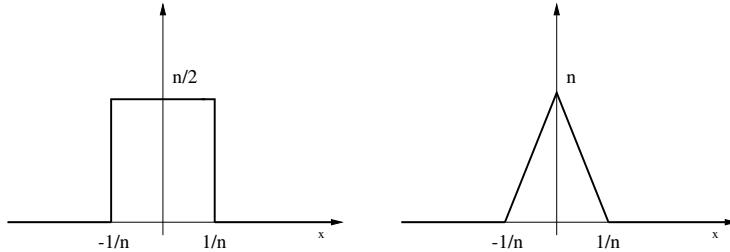


Fig. 2.1. The characteristic function of the interval $[-1/n, 1/n]$ (left) and the triangular function f_n (right)

Example 2.2 Let $\Omega = \mathbb{R}$ and let us denote by $\chi_{[a,b]}(x)$ the *characteristic function* of the interval $[a, b]$, defined as

$$\chi_{[a,b]}(x) = \begin{cases} 1 & \text{if } x \in [a, b], \\ 0 & \text{otherwise.} \end{cases}$$

Let us then consider the sequence of functions $f_n(x) = \frac{n}{2}\chi_{[-1/n, 1/n]}(x)$ (see Fig. 2.1).

We want to verify that the sequence $\{T_{f_n}\}$ of the distributions associated to the former converges to the distribution δ_0 , i.e. the Dirac delta relative to the origin. As a matter of fact, for each function $\phi \in \mathcal{D}(\Omega)$, we have

$$\langle T_{f_n}, \phi \rangle = \int_{\mathbb{R}} f_n(x) \phi(x) dx = \frac{n}{2} \int_{-1/n}^{1/n} \phi(x) dx = \frac{n}{2} [\Phi(1/n) - \Phi(-1/n)],$$

Φ being a primitive of ϕ . If we now set $h = 1/n$, we can write

$$\langle T_{f_n}, \phi \rangle = \frac{\Phi(h) - \Phi(-h)}{2h}.$$

When $n \rightarrow \infty$, $h \rightarrow 0$ and thus, following the definition of derivative, we have

$$\frac{\Phi(h) - \Phi(-h)}{2h} \rightarrow \Phi'(0).$$

By construction, $\Phi' = \phi$ and therefore

$$\langle T_{f_n}, \phi \rangle \rightarrow \phi(0) = \langle \delta_0, \phi \rangle,$$

having used the definition of δ_0 (see Example 2.1).

The same limit can be obtained by taking a sequence of triangular functions (see Fig. 2.1) or gaussian functions, instead of rectangular ones (provided that they still have unit integral).

Finally, we point out that in the usual metrics, such sequences converge to a function which is null almost everywhere. ■

2.3.2 Derivation in the sense of distributions

Let $\Omega \subset \mathbb{R}^n$ and $T \in \mathcal{D}'(\Omega)$. Its derivatives $\frac{\partial T}{\partial x_i}$ in the *sense of distributions* are distributions defined in the following way

$$\langle \frac{\partial T}{\partial x_i}, \phi \rangle = -\langle T, \frac{\partial \phi}{\partial x_i} \rangle \quad \forall \phi \in \mathcal{D}(\Omega), \quad i = 1, \dots, n.$$

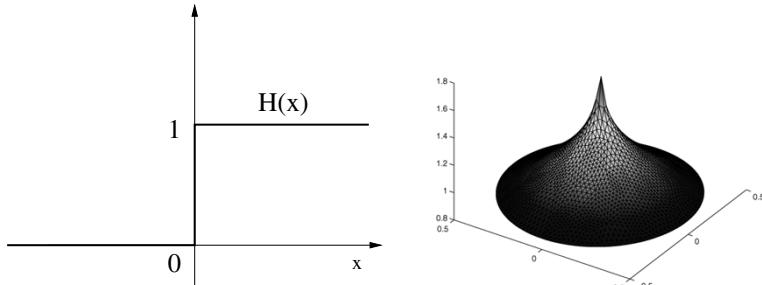


Fig. 2.2. The Heaviside function (left). At the right, the function of Example 2.6 with $k = 1/3$. Note that this function tends to infinity in the origin

In a similar way, we define derivatives of arbitrary order. Precisely, for each multi-index $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$, we have that $D^\alpha T$ is a new distribution defined as

$$\langle D^\alpha T, \phi \rangle = (-1)^{|\alpha|} \langle T, D^\alpha \phi \rangle \quad \forall \phi \in \mathcal{D}(\Omega).$$

Example 2.3 The *Heaviside function* on \mathbb{R} (see Fig. 2.2) is defined as

$$H(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x \leq 0. \end{cases}$$

The derivative of the distribution T_H associated to the latter is the Dirac distribution relative to the origin (see Example 2.1); upon identifying the H function with the associated T_H distribution, we will then write

$$\frac{dH}{dx} = \delta_0.$$

■

Differentiation in the context of distributions enjoys some important properties that do not hold in the more restricted context of differentiation for functions in classical terms.

Property 2.1 *The set $\mathcal{D}'(\Omega)$ is closed with respect to the differentiation operation (in the sense of distributions), that is each distribution is infinitely differentiable and its distributional derivatives are themselves distributions.*

Property 2.2 *Differentiation in $\mathcal{D}'(\Omega)$ is a continuous operation, in the sense that if $T_n \xrightarrow{\mathcal{D}'(\Omega)} T$ for $n \rightarrow \infty$, then it also results that $D^\alpha T_n \xrightarrow{\mathcal{D}'(\Omega)} D^\alpha T$ for $n \rightarrow \infty$, for each multi-index α .*

We finally note that differentiation in the sense of distributions is an extension of the classical differentiation of functions. Indeed, if a function f is differentiable with continuity (in classical sense) on Ω , then the derivative of the distribution T_f

corresponding to f coincides with the distribution $T_{f'}$ corresponding to the classical derivative f' of f (see Exercise 7).

We will invariably identify a function f of $L^2(\Omega)$ with the corresponding distribution T_f of $\mathcal{D}'(\Omega)$, writing f in place of T_f . Similarly, when we talk about derivatives, we will always refer to the latter in the sense of distributions.

2.4 Sobolev spaces

In paragraph 2.3.1, we have noted that the functions of $L^2(\Omega)$ are particular distributions. However, it is not guaranteed that their derivatives (in the sense of distributions) are still functions of $L^2(\Omega)$, as shown in the following example.

Example 2.4 Let $\Omega \subset \mathbb{R}$ and let $[a, b] \subset \Omega$. Then, the *characteristic function* of the interval $[a, b]$ (see Example 2.2) belongs to $L^2(\Omega)$, while its distributional derivative $d\chi_{[a,b]}/dx = \delta_a - \delta_b$ (see Example 2.3) does not. ■

It is therefore reasonable to introduce the following spaces:

Definition 2.12 Let Ω be an open set of \mathbb{R}^n and k be a positive integer. We call Sobolev space of order k on Ω the space formed by the totality of functions of $L^2(\Omega)$ such that all their (distributional) derivatives up to order k belong to $L^2(\Omega)$:

$$H^k(\Omega) = \{f \in L^2(\Omega) : D^\alpha f \in L^2(\Omega) \quad \forall \alpha : |\alpha| \leq k\}.$$

It results, obviously, that $H^{k+1}(\Omega) \subset H^k(\Omega)$ for each $k \geq 0$ and this inclusion is continuous. The space $L^2(\Omega)$ is sometimes denoted by $H^0(\Omega)$.

The Sobolev spaces $H^k(\Omega)$ are Hilbert spaces with respect to the following scalar product

$$(f, g)_k = \sum_{|\alpha| \leq k} \int_{\Omega} (D^\alpha f)(D^\alpha g) d\Omega,$$

from which descend the norms

$$\|f\|_{H^k(\Omega)} = \sqrt{(f, f)_k} = \sqrt{\sum_{|\alpha| \leq k} \int_{\Omega} (D^\alpha f)^2 d\Omega}. \quad (2.11)$$

Finally, we define the seminorms

$$|f|_{H^k(\Omega)} = \sqrt{\sum_{|\alpha|=k} \int_{\Omega} (D^\alpha f)^2 d\Omega},$$

so that (2.11) becomes

$$\|f\|_{H^k(\Omega)} = \sqrt{\sum_{m=0}^k \|f\|_{H^m(\Omega)}^2}.$$

Example 2.5 If $n = 1$ and $k = 1$ we have:

$$\begin{aligned} (f, g)_1 &= (f, g)_{H^1(\Omega)} = \int_{\Omega} fg \, d\Omega + \int_{\Omega} f'g' \, d\Omega; \\ \|f\|_1 &= \|f\|_{H^1(\Omega)} = \sqrt{\int_{\Omega} f^2 \, d\Omega + \int_{\Omega} f'^2 \, d\Omega} = \sqrt{\|f\|_{L^2(\Omega)}^2 + \|f'\|_{L^2(\Omega)}^2}; \\ |f|_1 &= |f|_{H^1(\Omega)} = \sqrt{\int_{\Omega} (f')^2 \, d\Omega} = \|f'\|_{L^2(\Omega)}. \end{aligned}$$

■

2.4.1 Regularity of the $H^k(\Omega)$ spaces

We now want to relate the fact that a function belongs to a space $H^k(\Omega)$ to its continuity properties.

Example 2.6 Let $\Omega = B(0, 1) \subset \mathbb{R}^2$ be the ball centered at the origin and of radius 1. Then, the following function, represented in Fig. 2.2 (right),

$$f(x_1, x_2) = \left| \ln \frac{1}{\sqrt{x_1^2 + x_2^2}} \right|^k \quad (2.12)$$

with $0 < k < 1/2$ belongs to $H^1(\Omega)$, but denotes a singularity at the origin and therefore it is neither continuous nor bounded. A similar conclusion can be drawn for

$$f(x_1, x_2) = \ln(-\ln(x_1^2 + x_2^2)),$$

this time with $\Omega = B(0, 1/2) \subset \mathbb{R}^2$. ■

Not all of the functions of $H^1(\Omega)$ are therefore continuous if Ω is an open set of \mathbb{R}^2 (or \mathbb{R}^3). In general, the following result holds:

Property 2.3 If Ω is an open set of \mathbb{R}^n , $n \geq 1$ provided with a “sufficiently regular” boundary, then

$$H^k(\Omega) \subset C^m(\overline{\Omega}) \quad \text{if } k > m + \frac{n}{2}.$$

In particular, in one spatial dimension ($n = 1$), the functions of $H^1(\Omega)$ are continuous (they are indeed *absolutely continuous*, see [Sal08] and [Bre86]), while in two or three dimensions they are not necessarily so. Instead, the functions of $H^2(\Omega)$ are indeed continuous for $n = 1, 2, 3$.

2.4.2 The $H_0^1(\Omega)$ space

If Ω is bounded, the space $\mathcal{D}(\Omega)$ is not dense in $H^1(\Omega)$. We can then give the following definition:

Definition 2.13 We denote by $H_0^1(\Omega)$ the closure of $\mathcal{D}(\Omega)$ in $H^1(\Omega)$.

The functions of $H_0^1(\Omega)$ enjoy the following properties:

Property 2.4 (Poincaré inequality) Let Ω be a bounded set of \mathbb{R}^n ; then there exists a constant C_Ω such that:

$$\|v\|_{L^2(\Omega)} \leq C_\Omega |v|_{H^1(\Omega)} \quad \forall v \in H_0^1(\Omega). \quad (2.13)$$

Proof. Ω being bounded, we can always find a sphere $S_D = \{\mathbf{x} : |\mathbf{x} - \mathbf{g}| < D\}$ with center \mathbf{g} and radius $D > 0$, containing Ω . Since $\mathcal{D}(\Omega)$ is dense in $H_0^1(\Omega)$ it is sufficient to prove the inequality for a function $u \in \mathcal{D}(\Omega)$. (In the general case where $v \in H_0^1(\Omega)$ it will suffice to build a sequence $u_i \in \mathcal{D}(\Omega)$, $i = 1, 2, \dots$ converging to v in the norm of $H^1(\Omega)$, apply the inequality to the terms of the sequence and pass to the limit.) Integrating by parts and exploiting the fact that $\operatorname{div}(\mathbf{x} - \mathbf{g}) = n$,

$$\begin{aligned} \|u\|_{L^2(\Omega)}^2 &= n^{-1} \int_{\Omega} n \cdot |u(\mathbf{x})|^2 d\Omega = -n^{-1} \int_{\Omega} (\mathbf{x} - \mathbf{g}) \cdot \nabla(|u(\mathbf{x})|^2) d\Omega \\ &= -2n^{-1} \int_{\Omega} (\mathbf{x} - \mathbf{g}) \cdot [u(\mathbf{x}) \nabla u(\mathbf{x})] d\Omega \leq 2n^{-1} \|\mathbf{x} - \mathbf{g}\|_{L^\infty(\Omega)} \|u\|_{L^2(\Omega)} \|u\|_{H^1(\Omega)} \\ &\leq 2n^{-1} D \|u\|_{L^2(\Omega)} \|u\|_{H^1(\Omega)}. \end{aligned}$$

◊

As an immediate consequence, we have that:

Property 2.5 The seminorm $\|v\|_{H^1(\Omega)}$ is on the space $H_0^1(\Omega)$ a norm that results to be equivalent to the norm $\|v\|_{H^1(\Omega)}$.

Proof. We recall that two norms, $\|\cdot\|$ and $\|\cdot\|$, are said to be *equivalent* if there exist two positive constants c_1 and c_2 , such that

$$c_1 \|v\| \leq \|v\| \leq c_2 \|v\| \quad \forall v \in V.$$

As $\|v\|_1 = \sqrt{|v|_1^2 + \|v\|_0^2}$ it is evident that $|v|_1 \leq \|v\|_1$. Viceversa, exploiting the property 2.4,

$$\|v\|_1 = \sqrt{|v|_1^2 + \|v\|_0^2} \leq \sqrt{|v|_1^2 + C_\Omega^2 |v|_1^2} \leq C_\Omega^* |v|_1,$$

from which we deduce the equivalence of the two norms. \diamond

In a similar way, we define the spaces $H_0^k(\Omega)$ as the closure of $\mathcal{D}(\Omega)$ in $H^k(\Omega)$.

2.4.3 Trace operators

Let Ω be a domain of \mathbb{R}^n . By that we mean:

- an open bounded interval if $n = 1$;
- an open bounded connected set, with a sufficiently regular boundary $\partial\Omega$. For instance, a polygon if $n = 2$ (i.e. a domain whose boundary is a finite union of segments), or a polyhedron if $n = 3$ (i.e. a domain whose boundary is a finite union of polygons).

Let v be an element of $H^1(\Omega)$: the remarks formulated in Sec. 2.4.1 show that it is not simple to define the “value” of v on the boundary of Ω , a value that we will call the *trace* of v on $\partial\Omega$. We exploit the following result:

Theorem 2.3 *Let Ω be a domain of \mathbb{R}^n provided with a “sufficiently regular” boundary $\partial\Omega$, and let $k \geq 1$. There exists one and only one linear and continuous application*

$$\gamma_0 : H^k(\Omega) \mapsto L^2(\partial\Omega),$$

such that $\gamma_0 v = v|_{\partial\Omega}$, $\forall v \in H^k \cap C^0(\overline{\Omega})$; $\gamma_0 v$ is called trace of v on $\partial\Omega$. The continuity of γ_0 implies that there exists a constant $C > 0$ such that

$$\|\gamma_0 v\|_{L^2(\Gamma)} \leq C \|v\|_{H^k(\Omega)}.$$

The result still holds if we consider the trace operator $\gamma_\Gamma : H^k(\Omega) \mapsto L^2(\Gamma)$ where Γ is a sufficiently regular portion with positive measure of the boundary of Ω .

Owing to this result, Dirichlet boundary conditions make sense when seeking solutions v in $H^k(\Omega)$, with $k \geq 1$, provided we interpret the boundary value in the sense of the trace.

Remark 2.1 The trace operator γ_Γ is not surjective on $L^2(\Gamma)$. In particular, the set of functions of $L^2(\Gamma)$ which are traces of functions of $H^1(\Omega)$ constitutes a subspace

of $L^2(\Gamma)$ denoted by $H^{1/2}(\Gamma)$ which is characterized by intermediate regularity properties between those of $L^2(\Gamma)$ and those of $H^1(\Gamma)$. More generally, for every $k \geq 1$ there exists a unique linear and continuous application $\gamma_0 : H^k(\Omega) \mapsto H^{k-1/2}(\Gamma)$ such that $\gamma_0 v = v|_{\Gamma}$ for each $v \in H^k(\Omega) \cap C^0(\overline{\Omega})$. •

The trace operators allow for an interesting characterization of the previously defined space $H_0^1(\Omega)$. Indeed, we have the following property:

Property 2.6 *Let Ω be a domain of \mathbb{R}^n provided with a sufficiently regular boundary $\partial\Omega$ and let γ_0 be the trace operator from $H^1(\Omega)$ in $L^2(\partial\Omega)$. We then have*

$$H_0^1(\Omega) = \text{Ker}(\gamma_0) = \{v \in H^1(\Omega) : \gamma_0 v = 0\}$$

In other words, $H_0^1(\Omega)$ is formed by the functions of $H^1(\Omega)$ having null trace on the boundary. Analogously, we define $H_0^2(\Omega)$ as the subspace of functions of $H^2(\Omega)$ whose traces, together with the traces of the normal derivative, vanish at the boundary.

2.5 The spaces $L^\infty(\Omega)$ and $L^p(\Omega)$, with $1 \leq p < \infty$

The $L^2(\Omega)$ space can be generalized in the following way: for each real number p with $1 \leq p < \infty$ we can define the following space of equivalence classes of measurable functions

$$L^p(\Omega) = \{v : \Omega \mapsto \mathbb{R} \text{ s.t. } \int_{\Omega} |v(\mathbf{x})|^p d\Omega < \infty\},$$

whose norm is given by

$$\|v\|_{L^p(\Omega)} = \left(\int_{\Omega} |v(\mathbf{x})|^p d\Omega \right)^{1/p}.$$

Furthermore, we define the space

$$L_{loc}^1(\Omega) = \{f : \Omega \rightarrow \mathbb{R} \text{ s.t. } f|_K \in L^1(K) \text{ for each compact set } K \subset \Omega\}.$$

If $1 \leq p < \infty$, then $\mathcal{D}(\Omega)$ is dense in $L^p(\Omega)$.

In the case where $p = \infty$, we define $L^\infty(\Omega)$ to be the space of functions that are bounded a.e. in Ω . Its norm is defined as follows

$$\begin{aligned} \|v\|_{L^\infty(\Omega)} &= \inf\{C \in \mathbb{R} : |v(x)| \leq C, \text{ a.e. in } \Omega\} \\ &= \sup\{|v(x)|, \text{ a.e. in } \Omega\}. \end{aligned} \tag{2.14}$$

For $1 \leq p \leq \infty$, the spaces $L^p(\Omega)$, provided with the norm $\|\cdot\|_{L^p(\Omega)}$, are Banach spaces.

We recall the *Hölder inequality*: given $v \in L^p(\Omega)$ and $w \in L^{p'}(\Omega)$ with $1 \leq p \leq \infty$ and $\frac{1}{p} + \frac{1}{p'} = 1$, then $vw \in L^1(\Omega)$ and

$$\int_{\Omega} |v(\mathbf{x}) w(\mathbf{x})| d\Omega \leq \|v\|_{L^p(\Omega)} \|w\|_{L^{p'}(\Omega)}. \quad (2.15)$$

The index p' is called conjugate of p .

If $1 < p < \infty$, then $L^p(\Omega)$ is a reflexive space: this means that any linear and continuous form $\varphi : L^p(\Omega) \rightarrow \mathbb{R}$ can be identified to an element of $L^{p'}(\Omega)$, i.e. there exists a unique $g \in L^{p'}(\Omega)$ such that

$$\varphi(f) = \int_{\Omega} f(\mathbf{x}) g(\mathbf{x}) d\Omega \quad \forall f \in L^p(\Omega).$$

If $p = 2$, then $p' = 2$, so the Hölder inequality becomes

$$(v, w)_{L^2(\Omega)} \leq \|v\|_{L^2(\Omega)} \|w\|_{L^2(\Omega)} \quad \forall v, w \in L^2(\Omega). \quad (2.16)$$

As such, it is known as Cauchy-Schwarz inequality. Moreover, the following inequality holds

$$\|vw\|_{L^2(\Omega)} \leq \|v\|_{L^4(\Omega)} \|w\|_{L^4(\Omega)} \quad \forall v, w \in L^4(\Omega). \quad (2.17)$$

If $\Omega \subset \mathbb{R}^n$ is a domain, if $1 \leq p \leq q \leq \infty$

$$L^q(\Omega) \subset L^p(\Omega) \subset L^1(\Omega) \subset L^1_{loc}(\Omega).$$

If Ω is unbounded, we always have

$$L^p(\Omega) \subset L^1_{loc}(\Omega) \quad \forall p \geq 1.$$

Moreover, if $\Omega \subset \mathbb{R}^n$ and, if $n > 1$, the boundary $\partial\Omega$ is polygonal (more generally, it is Lipschitz continuous), we have the following continuous inclusions:

- if $0 < 2s < n$ then $H^s(\Omega) \subset L^q(\Omega) \quad \forall q \text{ s.t. } 1 \leq q \leq q^* \text{ with } q^* = 2n/(n - 2s);$
 - if $2s = n$ then $H^s(\Omega) \subset L^q(\Omega) \quad \forall q \text{ s.t. } 1 \leq q < \infty;$
 - if $2s > n$ then $H^s(\Omega) \subset C^0(\overline{\Omega}).$
- (2.18)

Finally, we introduce the Sobolev space $W^{k,p}(\Omega)$, k a non-negative integer and $1 \leq p \leq \infty$, as the space of functions $v \in L^p(\Omega)$ such that all the distributional derivatives of v of order up to k are a function of $L^p(\Omega)$. In short

$$W^{k,p}(\Omega) = \{v \in L^p(\Omega) : D^\alpha v \in L^p(\Omega) \text{ for each non-negative multi-index } \alpha \text{ s.t. } |\alpha| \leq k\}.$$

For $1 \leq p < \infty$ this is a Banach space with norm

$$\|v\|_{W^{k,p}(\Omega)} = \left(\sum_{|\alpha| \leq k} \|D^\alpha v\|_{L^p(\Omega)}^p \right)^{1/p}.$$

Its seminorm $|v|_{W^{k,p}(\Omega)}$ is defined similarly provided the sum is bounded to those multi-integers α s.t. $|\alpha| = k$.

Note that, for $k = 0$, $W^{k,p}(\Omega) = L^p(\Omega)$, and that, for $p = 2$, $W^{k,2}(\Omega) = H^k(\Omega)$.

2.6 Adjoint operators of a linear operator

Let X and Y be two Banach spaces and $\mathcal{L}(X, Y)$ be the space of linear and bounded operators from X to Y . Given $L \in \mathcal{L}(X, Y)$, the *adjoint* (or *conjugate*) *operator* of L is another operator $L' : Y' \rightarrow X'$ defined by

$${}_{X'}\langle L'f, x \rangle_X = {}_{Y'}\langle f, Lx \rangle_Y \quad \forall f \in Y', x \in X. \quad (2.19)$$

L' is a linear and bounded operator between Y' and X' , that is $L' \in \mathcal{L}(Y', X')$, moreover $\|L'\|_{\mathcal{L}(Y', X')} = \|L\|_{\mathcal{L}(X, Y)}$, where we have set

$$\|L\|_{\mathcal{L}(X, Y)} = \sup_{\substack{x \in X \\ x \neq 0}} \frac{\|Lx\|_Y}{\|x\|_X}. \quad (2.20)$$

In the case where X and Y are two Hilbert spaces, an additional adjoint operator, $L^T : Y \rightarrow X$, called *transpose* of L , can be introduced. It is defined by

$$(L^T y, x)_X = (y, Lx)_Y \quad \forall x \in X, y \in Y. \quad (2.21)$$

Here, $(\cdot, \cdot)_X$ denotes the scalar product of X , while $(\cdot, \cdot)_Y$ denotes the scalar product of Y . The above definition can be explained as follows: for any given element $y \in Y$, the real-valued function $x \rightarrow (y, Lx)_Y$ is linear and continuous, hence defining an element of X' . Then, thanks to the Riesz theorem (Theorem 2.1) there exists an element x of X which we name $L^T y$ that satisfies (2.21). Such operator belongs to $\mathcal{L}(Y, X)$ (that is, it is linear and bounded from Y to X) and moreover

$$\|L^T\|_{\mathcal{L}(Y, X)} = \|L\|_{\mathcal{L}(X, Y)}. \quad (2.22)$$

Thus, in the case where X and Y are two Hilbert spaces, we have two notions of adjoint operator, L' and L^T . The relation between the two operators is

$$\Lambda_X L^T = L' \Lambda_Y, \quad (2.23)$$

Λ_X and Λ_Y being Riesz's canonical isomorphisms from X to X' and from Y to Y' , respectively (see (2.5)). Indeed, $\forall x \in X, y \in Y$,

$$\begin{aligned} {}_{X'}\langle \Lambda_X L^T y, x \rangle_X &= (L^T y, x)_X = (y, Lx)_Y = {}_{Y'}\langle \Lambda_Y y, Lx \rangle_Y \\ &= {}_{X'}\langle L' \Lambda_Y y, x \rangle_X. \end{aligned}$$

Identity (2.23) can be equivalently expressed by stating that the diagram in Fig. 2.3 is commutative.

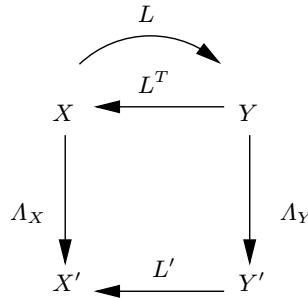


Fig. 2.3. The adjoint operators L^T and L' of the operator L

2.7 Spaces of time-dependent functions

When considering space-time functions $v(\mathbf{x}, t)$, $\mathbf{x} \in \Omega \subset \mathbb{R}^n$, $n \geq 1$, $t \in (0, T)$, $T > 0$, it is natural to introduce the functional space

$$L^q(0, T; W^{k,p}(\Omega)) = \left\{ v : (0, T) \rightarrow W^{k,p}(\Omega) \text{ s.t. } v \text{ is measurable and } \int_0^T \|v(t)\|_{W^{k,p}(\Omega)}^q dt < \infty \right\}, \quad (2.24)$$

where $k \geq 0$ is a non-negative integer, $1 \leq q < \infty$, $1 \leq p \leq \infty$, endowed with the norm

$$\|v\|_{L^q(0, T; W^{k,p}(\Omega))} = \left(\int_0^T \|v(t)\|_{W^{k,p}(\Omega)}^q dt \right)^{1/q}. \quad (2.25)$$

For every $t \in (0, T)$ we have used the shorthand notation $v(t)$ to indicate the function:

$$v(t) : \Omega \rightarrow \mathbb{R}, \quad v(t)(\mathbf{x}) = v(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Omega. \quad (2.26)$$

The spaces $L^\infty(0, T : W^{k,p}(\Omega))$ and $C^0([0, T]; W^{k,p}(\Omega))$ are defined in a similar way.

When dealing with time-dependent initial-boundary value problems, the following result can be useful to derive a-priori estimates and stability inequalities.

Lemma 2.2 (Gronwall) Let $A \in L^1(t_0, T)$ be a non-negative function, g and φ two continuous functions on $[t_0, T]$. If φ is such that

$$\varphi(t) \leq g(t) + \int_{t_0}^t A(\tau)\varphi(\tau)d\tau \quad \forall t \in [t_0, T], \quad (2.27)$$

then, if g is non-decreasing,

$$\varphi(t) \leq g(t) \exp\left(\int_{t_0}^t A(\tau)d\tau\right) \quad \forall t \in [t_0, T]. \quad (2.28)$$

A discrete counterpart of this lemma, that will be useful when dealing with fully discrete (in space and time) approximations of initial-boundary value problems, is the following

Lemma 2.3 (discrete Gronwall lemma) Assume that k_n is a non-negative sequence, and that the sequence φ_n satisfies

$$\varphi_0 \leq g_0, \quad \varphi_n \leq g_0 + \sum_{m=0}^{n-1} p_m + \sum_{m=0}^{n-1} k_m \varphi_m, \quad n \geq 1. \quad (2.29)$$

If $g_0 \geq 0$ and $p_m \geq 0$ for $m \geq 0$, then

$$\varphi_n \leq (g_0 + \sum_{m=0}^{n-1} p_m) \exp(\sum_{m=0}^{n-1} k_m), \quad n \geq 1. \quad (2.30)$$

For the proof of these two lemmas, see, e.g., [QV94, Chap. 1]

2.8 Exercises

1. Let $\Omega = (0, 1)$ and, for $\alpha > 0$, $f(x) = x^{-\alpha}$. For which α do we have $f \in L^p(\Omega)$, $1 \leq p < \infty$? Is there a $\alpha > 0$ for which $f \in L^\infty(\Omega)$?
2. Let $\Omega = (0, \frac{1}{2})$ and $f(x) = \frac{1}{x(\ln x)^2}$. Show that $f \in L^1(\Omega)$.
3. Prove for which $\alpha \in \mathbb{R}$ we have that $f \in L^1_{loc}(0, 1)$, with $f(x) = x^{-\alpha}$.

4. Let $u \in L^1_{loc}(\Omega)$. Define $T_u \in \mathcal{D}'(\Omega)$ as follows

$$\langle T_u, \varphi \rangle = \int_{\Omega} \varphi(\mathbf{x}) u(\mathbf{x}) d\Omega \quad \forall \varphi \in \mathcal{D}(\Omega).$$

Verify that T_u is effectively a distribution, and that the application $u \rightarrow T_u$ is injective. We can therefore identify u with T_u , and conclude by observing that $L^1_{loc}(\Omega) \subset D'(\Omega)$.

5. Show that the function defined as follows:

$$\begin{aligned} f(x) &= e^{1/(x^2-1)} \text{ if } x \in (-1, 1) \\ f(x) &= 0 \text{ if } x \in [-\infty, -1] \cup [1, +\infty[\end{aligned}$$

belongs to $\mathcal{D}(\mathbb{R})$.

6. Prove that for the function f defined in (2.12) we have

$$\|f\|_{H^1(\Omega)}^2 = 2\pi \int_0^r |\log s|^{2k} s ds + 2\pi k^2 \int_0^r \frac{1}{s} |\log s|^{2k-2} ds,$$

hence f belongs to $H^1(\Omega)$ for every $0 < k < \frac{1}{2}$.

7. Let $\varphi \in C^1(-1, 1)$. Show that the derivative $\frac{d\varphi}{dx}$ computed in the classical sense is equal to $\frac{d\varphi}{dx}$ computed in the sense of distributions, after observing that $C^0(-1, 1) \subset L^1_{loc}(-1, 1) \subset \mathcal{D}'(-1, 1)$.
8. Prove that, if $\Omega = (a, b)$, the Poincaré inequality (2.13) holds with $C_\Omega = (b-a)/\sqrt{2}$.

[*Solution:* observe that, thanks to the Cauchy-Schwarz inequality, it is

$$v(x) = \int_a^x v'(t) dt \leq \left(\int_a^x [v'(t)]^2 dt \right)^{1/2} \left(\int_a^x 1 dt \right)^{1/2} \leq \sqrt{x-a} \|v'\|_{L^2(a,b)}$$

$$\text{whence } \|v\|_{L^2(a,b)}^2 \leq \|v'\|_{L^2(a,b)}^2 \int_a^b (x-a) dx.$$

3

Elliptic equations

This chapter is devoted to the introduction of elliptic problems and to their weak formulation. Although our introduction is quite basic, the complete novice to functional analysis is invited to consult Chap. 2 before reading it.

For the sake of simplicity, most of our derivation will be given for one-dimensional and two-dimensional problems. However, the generalization to three-dimensional problems is (almost always) straightforward.

3.1 An elliptic problem example: the Poisson equation

Consider a domain $\Omega \subset \mathbb{R}^2$, i.e. an open bounded and connected set, and let $\partial\Omega$ be its boundary. We denote by \mathbf{x} the spatial variable pair (x_1, x_2) . The problem under examination is

$$-\Delta u = f \quad \text{in } \Omega, \tag{3.1}$$

where $f = f(\mathbf{x})$ is a given function and the symbol Δ denotes the Laplacian operator (1.6) in two dimensions. (3.1) is an elliptic, linear, non-homogeneous (if $f \neq 0$) second-order equation. We call (3.1) the *strong formulation* of the Poisson equation. We also recall that, in the case where $f = 0$, equation (3.1) is known as the Laplace equation.

Physically, u can represent the vertical displacement of an elastic membrane due to the application of a force with intensity equal to f , or the electric potential distribution due to an electric charge with density f .

To obtain a unique solution, suitable boundary conditions must be added to (3.1), that is we need information about the behavior of the solution u at the domain boundary $\partial\Omega$. For instance, the value of the displacement u on the boundary can be assigned

$$u = g \quad \text{on } \partial\Omega, \tag{3.2}$$

where g is a given function, and in such case we will talk about a *Dirichlet problem*. The case where $g = 0$ is said to be *homogeneous*.

Alternatively, the value of the *normal derivative* of u can be imposed

$$\nabla u \cdot \mathbf{n} = \frac{\partial u}{\partial n} = h \quad \text{on } \partial\Omega,$$

\mathbf{n} being the outward unit normal vector on $\partial\Omega$ and h an assigned function. The associated problem is called a *Neumann problem* and corresponds, in the case of the membrane problem, to imposing the traction at the boundary of the membrane itself. Once again, the case $h = 0$ is said to be *homogeneous*.

Finally, different types of conditions can be assigned to different portions of the boundary of the computational domain Ω . For instance, supposing that $\partial\Omega = \Gamma_D \cup \Gamma_N$ with $\overset{\circ}{\Gamma}_D \cap \overset{\circ}{\Gamma}_N = \emptyset$, the following conditions can be imposed:

$$\begin{cases} u = g & \text{on } \Gamma_D, \\ \frac{\partial u}{\partial n} = h & \text{on } \Gamma_N. \end{cases}$$

The notation $\overset{\circ}{\Gamma}$ has been used to indicate the interior of Γ . In such a case, the associated problem is said to be *mixed*.

Also in the case of homogeneous Dirichlet problems where f is a continuous function in $\overline{\Omega}$ (the closure of Ω), it is not guaranteed that problem (3.1), (3.2) admits a regular solution. For instance, if $\Omega = (0, 1) \times (0, 1)$ and $f = 1$, u could not belong to the space $C^2(\overline{\Omega})$. Indeed, if it were so, we would have

$$-\Delta u(0, 0) = -\frac{\partial^2 u}{\partial x_1^2}(0, 0) - \frac{\partial^2 u}{\partial x_2^2}(0, 0) = 0$$

as the boundary conditions would imply that $u(x_1, 0) = u(0, x_2) = 0$ for all x_1, x_2 belonging to $[0, 1]$. Hence u could not satisfy equation (3.1), that is

$$-\Delta u = 1 \quad \text{in } (0, 1) \times (0, 1).$$

What can be learned from this counterexample is that, even if $f \in C^0(\overline{\Omega})$, it makes no sense in general to look for a solution $u \in C^2(\overline{\Omega})$ to problem (3.1), (3.2), while one has greater probabilities to find a solution $u \in C^2(\Omega) \cap C^0(\overline{\Omega})$ (a larger space than $C^2(\overline{\Omega})$!).

We are therefore interested in finding an alternative formulation to the strong one, also because, as we will see in the following section, the latter does not allow the treatment of some physically significant cases. For instance, it is not guaranteed that, in the presence of non-smooth data, the physical solution lies in the space $C^2(\Omega) \cap C^0(\overline{\Omega})$, and not even that it lies in $C^1(\Omega) \cap C^0(\overline{\Omega})$.

3.2 The Poisson problem in the one-dimensional case

Our first step is the introduction of the weak formulation of a simple boundary-value problem in one dimension.

3.2.1 Homogeneous Dirichlet problem

Let us consider the homogeneous Dirichlet problem in the one-dimensional interval $\Omega = (0, 1)$

$$\begin{cases} -u''(x) = f(x), & 0 < x < 1, \\ u(0) = 0, & u(1) = 0. \end{cases} \quad (3.3)$$

This problem governs, for instance, the equilibrium configuration of an elastic string with tension equal to one, fixed at the extrema, in a small displacement configuration and subject to a transversal force with intensity f . The overall force acting on section $(0, x)$ of the string is

$$F(x) = \int_0^x f(t) dt.$$

The function u describes the vertical displacement of the string relative to the resting position $u = 0$.

The strong formulation (3.3) is in general inadequate. If we consider, for instance, the case where the elastic string is subject to a charge concentrated in one or more points (in such case f can be represented via Dirac “deltas”), the physical solution exists and is continuous, but not differentiable. See the graphs of Fig. 3.1, where the case of a unit charge concentrated only in the point $x = 0.5$ is considered (left) and in the two points $x = 0.4$ and $x = 0.6$ (right). These functions cannot be solutions of (3.3), as the latter would require the solution to have a continuous second derivative. Similar considerations hold in the case where f is a piecewise constant function. For instance, in the case represented in Fig. 3.2 of a null load except for the interval $[0.4, 0.6]$ where it is equal to -1 , the analytical solution is only of class $C^1([0, 1])$, since it is given by

$$u(x) = \begin{cases} -\frac{1}{10}x & \text{for } x \in [0, 0.4], \\ \frac{1}{2}x^2 - \frac{1}{2}x + \frac{2}{25} & \text{for } x \in [0.4, 0.6], \\ -\frac{1}{10}(1-x) & \text{for } x \in [0.6, 1]. \end{cases}$$

A formulation of the problem alternative to the strong one is therefore necessary to allow reducing the order of the derivation required for the unknown solution u . We move from a second order differential problem to a first-order one in integral form, which is called the *weak formulation* of the differential problem.

To this end, we operate a sequence of formal transformations of (3.3), without worrying at this stage whether all the operations appearing in it are allowed. We start by multiplying equation (3.3) by a (so far arbitrary) *test function* v and integrating on the interval $(0, 1)$,

$$-u''v = fv \Rightarrow -\int_0^1 u''v dx = \int_0^1 fv dx.$$

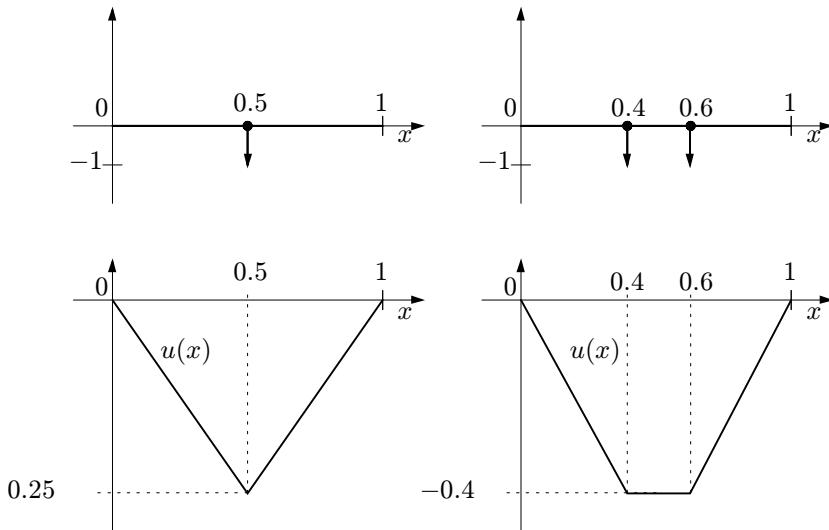


Fig. 3.1. We display on the left the equilibrium configuration of the string corresponding to the unit charge concentrated in $x = 0.5$, represented in the upper part of the figure. On the right we display the one corresponding to two unit charges concentrated in $x = 0.4$ and $x = 0.6$, also represented in the upper part of the figure

We apply the integration by parts formula to the first integral, with the purpose of eliminating the second derivative, in order to impose a lower regularity on the solution.

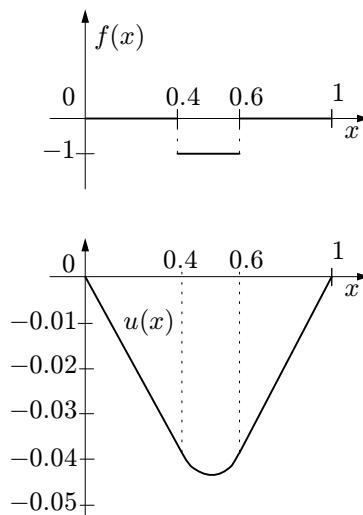


Fig. 3.2. Displacement relative to the discontinuous charge represented in the upper part of the figure

We find

$$-\int_0^1 u''v \, dx = \int_0^1 u'v' \, dx - [u'v]_0^1.$$

Since u is known at the boundary, we can consider only test functions which vanish at the extrema of the interval, hence the contribution of the boundary terms vanishes. In this way, the equation becomes

$$\int_0^1 u'v' \, dx = \int_0^1 fv \, dx. \quad (3.4)$$

The test function space V must therefore be such that if $v \in V$ then $v(0) = v(1) = 0$. Note that the solution u , being null at the boundary and having the same requirements of regularity as the test functions, will also be sought in the same space V .

It is now left to specify the regularity requirements which must be satisfied by the space V , so that all the operations introduced make sense. Evidently, if u and v belonged to $C^1([0, 1])$, we would have $u', v' \in C^0([0, 1])$ and therefore the integral appearing in the left-hand side of (3.4) would make sense. However, the examples in Fig. 3.1 tell us that the physical solutions might not be continuously differentiable: we must therefore require a lower regularity. Moreover, even when $f \in C^0([0, 1])$, there is no certainty that the problem admits solutions in the space

$$V = \{v \in C^1([0, 1]) : v(0) = v(1) = 0\}. \quad (3.5)$$

This may be attributed to the fact that such vector space, when provided with a scalar product

$$[u, v]_1 = \int_0^1 u'v' \, dx, \quad (3.6)$$

is not a complete space, that is, not all of the Cauchy sequences with values in V converge to an element of V . (Verify as an exercise that (3.6) is indeed a scalar product.)

Let us then proceed as follows. We recall the definition of the spaces L^p of the functions which are p -th power Lebesgue integrable. For $1 \leq p < \infty$, these are defined as follows (see Sec. 2.5):

$$L^p(0, 1) = \{v : (0, 1) \mapsto \mathbb{R} \text{ s.t. } \|v\|_{L^p(0, 1)} = \left(\int_0^1 |v(x)|^p \, dx \right)^{1/p} < +\infty\}.$$

Since we want the integral $\int_0^1 u'v' \, dx$ to be well defined, the minimum requirement on u' and v' is that the product $u'v'$ lies in $L^1(0, 1)$. To this purpose, the following property holds:

Property 3.1 Given two functions $\varphi, \psi : (0, 1) \rightarrow \mathbb{R}$, if

φ^2, ψ^2 are integrable then $\varphi\psi$ is integrable,

that is, equivalently,

$$\varphi, \psi \in L^2(0, 1) \implies \varphi\psi \in L^1(0, 1).$$

This result is a direct consequence of the *Cauchy-Schwarz inequality*:

$$\left| \int_0^1 \varphi(x)\psi(x) dx \right| \leq \|\varphi\|_{L^2(0,1)} \|\psi\|_{L^2(0,1)}, \quad (3.7)$$

where

$$\|\varphi\|_{L^2(0,1)} = \sqrt{\int_{\Omega} |\varphi(x)|^2 dx} \quad (3.8)$$

is the norm of φ in $L^2(0, 1)$. Since $\|\varphi\|_{L^2(0,1)}, \|\psi\|_{L^2(0,1)} < \infty$ by hypothesis, this proves that there also exists a (finite) integral of $\varphi\psi$.

In order for the integrals appearing in (3.4) to make sense, functions, as well as their derivatives, must be square integrable. We therefore define the *Sobolev space*

$$H^1(0, 1) = \{v \in L^2(0, 1) : v' \in L^2(0, 1)\}.$$

The derivative must be interpreted in the sense of distributions (see Sec. 2.3). Let us hence choose as V the following subspace of $H^1(0, 1)$,

$$H_0^1(0, 1) = \{v \in H^1(0, 1) : v(0) = v(1) = 0\},$$

constituted by the functions of $H^1(0, 1)$ that are null at the extrema of the interval. If we suppose $f \in L^2(0, 1)$, the integral on the right-hand side of (3.4) also makes sense. Problem (3.3) is then reduced to the following integral problem,

$$\text{find } u \in V : \int_0^1 u'v' dx = \int_0^1 fv dx \quad \forall v \in V, \quad (3.9)$$

with $V = H_0^1(0, 1)$.

Remark 3.1 In fact, the space $H_0^1(0, 1)$ is the closure, with respect to the scalar product (3.6), of the space defined in (3.5).

The functions of $H^1(0, 1)$ are not necessarily differentiable in a traditional sense, that is $H^1(0, 1) \not\subset C^1([0, 1])$. For instance, functions that are piecewise continuous on a partition of the interval $(0, 1)$ with derivatives that do not match at all endpoints of the partition belong to $H^1(0, 1)$ but not to $C^1([0, 1])$. Hence, also continuous but not differentiable solutions of the previous examples are considered. •

The weak problem (3.9) turns out to be equivalent to a *variational problem*, thanks to the following result:

Theorem 3.1 *The problem*

$$\text{find } u \in V : \begin{cases} J(u) = \min_{v \in V} J(v) & \text{with} \\ J(v) = \frac{1}{2} \int_0^1 (v')^2 dx - \int_0^1 fv dx, \end{cases} \quad (3.10)$$

is equivalent to problem (3.9), in the sense that u is a solution of (3.9) if and only if u is a solution of (3.10).

Proof. Suppose that u is a solution of the variational problem (3.10). Then, setting $v = u + \delta w$, with $\delta \in \mathbb{R}$, we have that

$$J(u) \leq J(u + \delta w) \quad \forall w \in V.$$

The function $\psi(\delta) = J(u + \delta w)$ is a quadratic function in δ with minimum reached for $\delta = 0$. Thus,

$$\psi'(\delta) \Big|_{\delta=0} = \frac{\partial J(u + \delta w)}{\partial \delta} \Big|_{\delta=0} = 0.$$

From the definition of derivative we have

$$\frac{\partial J(u + \delta w)}{\partial \delta} = \lim_{\delta \rightarrow 0} \frac{J(u + \delta w) - J(u)}{\delta} \quad \forall w \in V.$$

Let us consider the term $J(u + \delta w)$:

$$\begin{aligned} J(u + \delta w) &= \frac{1}{2} \int_0^1 [(u + \delta w)']^2 dx - \int_0^1 f(u + \delta w) dx \\ &= \frac{1}{2} \int_0^1 [u'^2 + \delta^2 w'^2 + 2\delta u' w'] dx - \int_0^1 f u dx - \int_0^1 f \delta w dx \\ &= J(u) + \frac{1}{2} \int_0^1 [\delta^2 w'^2 + 2\delta u' w'] dx - \int_0^1 f \delta w dx. \end{aligned}$$

Henceforth,

$$\frac{J(u + \delta w) - J(u)}{\delta} = \frac{1}{2} \int_0^1 [\delta w'^2 + 2u' w'] dx - \int_0^1 f w dx.$$

Passing to the limit for $\delta \rightarrow 0$ and imposing that it vanishes, we obtain

$$\int_0^1 u'w' dx - \int_0^1 fw dx = 0 \quad \forall w \in V,$$

that is, u satisfies the weak problem (3.9).

Conversely, if u is a solution of (3.9), by setting $v = \delta w$, we have in particular that

$$\int_0^1 u'\delta w' dx - \int_0^1 f\delta w dx = 0,$$

and therefore

$$\begin{aligned} J(u + \delta w) &= \frac{1}{2} \int_0^1 [(u + \delta w)']^2 dx - \int_0^1 f(u + \delta w) dx \\ &= \frac{1}{2} \int_0^1 u'^2 dx - \int_0^1 fu dx + \int_0^1 u'\delta w' dx - \int_0^1 f\delta w dx + \frac{1}{2} \int_0^1 \delta^2 w'^2 dx \\ &= J(u) + \frac{1}{2} \int_0^1 \delta^2 w'^2 dx. \end{aligned}$$

Since

$$\frac{1}{2} \int_0^1 \delta^2 w'^2 dx \geq 0 \quad \forall w \in V, \forall \delta \in \mathbb{R},$$

we deduce that

$$J(u) \leq J(v) \quad \forall v \in V,$$

that is u also satisfies the variational problem (3.10). \diamond

Remark 3.2 (Principle of virtual work) Let us consider again the problem of studying the configuration assumed by a unit tension string, fixed at the extrema and subject to a forcing term f , described by equation (3.3). We indicate with v an admissible displacement of the string (that is a null displacement at the extrema) from the equilibrium position u . Equation (3.9), expressing the equality between the work performed by the internal forces and by the external forces in correspondence to the displacement v , is nothing but the *principle of virtual work* of mechanics. Moreover, as in our case there exists a potential (indeed, $J(w)$ defined in (3.10) expresses the potential global energy corresponding to the configuration w of the system), the principle of virtual works establishes that any displacement allowed by the equilibrium configuration causes an increment of the system's potential energy. In this sense, Theorem 3.1 states that the weak solution is also the one minimizing the potential energy. •

3.2.2 Non-homogeneous Dirichlet problem

In the non-homogeneous case the boundary conditions in (3.3) are replaced by

$$u(0) = g_0, \quad u(1) = g_1,$$

g_0 and g_1 being two assigned values.

We can reconduct to the homogeneous case by noticing that if u is a solution of the non-homogeneous problem, then the function $\hat{u} = u - [(1-x)g_0 + xg_1]$ is a solution of the corresponding homogeneous problem (3.3). The function $R_g = (1-x)g_0 + xg_1$ is said *lifting* (or *extension*, or *prolongation*) of the boundary data.

3.2.3 Neumann Problem

Let us now consider the following Neumann problem

$$\begin{cases} -u'' + \sigma u = f, & 0 < x < 1, \\ u'(0) = h_0, & u'(1) = h_1, \end{cases}$$

σ being a positive function and h_0, h_1 two real numbers. We observe that in the case where $\sigma = 0$ the solution of this problem would not be unique, being defined up to an additive constant. By applying the same procedure followed in the case of the Dirichlet problem, that is by multiplying the equation by a test function v , integrating on the interval $(0, 1)$ and applying the formula of integration by parts, we get the equation

$$\int_0^1 u'v' \, dx + \int_0^1 \sigma uv \, dx - [u'v]_0^1 = \int_0^1 fv \, dx.$$

Let us suppose $f \in L^2(0, 1)$ and $\sigma \in L^\infty(0, 1)$ that is that σ be a bounded function almost everywhere (a.e.) on $(0, 1)$ (see (2.14)). The boundary term is known thanks to the Neumann conditions. On the other hand, the unknown u is not known at the boundary in this case, hence it must not be required that v be null at the boundary. The weak formulation of the Neumann problem is therefore: *find $u \in H^1(0, 1)$ such that*

$$\int_0^1 u'v' \, dx + \int_0^1 \sigma uv \, dx = \int_0^1 fv \, dx + h_1 v(1) - h_0 v(0) \quad \forall v \in H^1(0, 1). \quad (3.11)$$

In the homogeneous case $h_0 = h_1 = 0$, the weak problem is characterized by the same equation as the Dirichlet case, but the space V of test functions is now $H^1(0, 1)$ instead of $H_0^1(0, 1)$.

3.2.4 Mixed homogeneous problem

Analogous considerations hold for the mixed homogeneous problem, that is when we have a homogeneous Dirichlet condition in one extreme and a homogeneous Neumann condition in the other,

$$\begin{cases} -u'' + \sigma u = f, & 0 < x < 1, \\ u(0) = 0, & u'(1) = 0. \end{cases} \quad (3.12)$$

In such case it must be required that the test functions be null in $x = 0$. Setting $\Gamma_D = \{0\}$ and defining

$$H_{\Gamma_D}^1(0, 1) = \{v \in H^1(0, 1) : v(0) = 0\},$$

the weak formulation of problem (3.12) is: *find $u \in H_{\Gamma_D}^1(0, 1)$ such that*

$$\int_0^1 u'v' dx + \int_0^1 \sigma uv dx = \int_0^1 fv dx \quad \forall v \in H_{\Gamma_D}^1(0, 1),$$

with $f \in L^2(0, 1)$ and $\sigma \in L^\infty(0, 1)$. The formulation is once again the same as in the homogeneous Dirichlet problem, however the space where to find the solution changes.

3.2.5 Mixed (or Robin) boundary conditions

Finally, consider the following problem

$$\begin{cases} -u'' + \sigma u = f, & 0 < x < 1, \\ u(0) = 0, & u'(1) + \gamma u(1) = r, \end{cases}$$

where $\gamma > 0$ and r are two assigned constants.

Also in this case, we will use test functions that are null at $x = 0$, the value of u being known thereby. As opposed to the Neumann case, the boundary term for $x = 1$, deriving from the integration by parts, no longer provides a known quantity, but a term proportional to the unknown u . As a matter of fact, we have

$$-[u']_0^1 = -rv(1) + \gamma u(1)v(1).$$

The weak formulation is therefore: *find $u \in H_{\Gamma_D}^1(0, 1)$ such that*

$$\int_0^1 u'v' dx + \int_0^1 \sigma uv dx + \gamma u(1)v(1) = \int_0^1 fv dx + rv(1) \quad \forall v \in H_{\Gamma_D}^1(0, 1).$$

A boundary condition that is a linear combination between the value of u and the value of its first derivative is called *Robin* (or *Newton*, or *third type*) *condition*.

3.3 The Poisson problem in the two-dimensional case

In this section, we consider the problems at the limits associated to the Poisson equation in the two-dimensional case.

3.3.1 The homogeneous Dirichlet problem

The problem consists in finding u such that

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \quad (3.13)$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain with boundary $\partial\Omega$. We proceed in a similar way as for the one-dimensional case. By multiplying the differential equation in (3.13) by an arbitrary function v and integrating on Ω , we find

$$-\int_{\Omega} \Delta u v \, d\Omega = \int_{\Omega} f v \, d\Omega.$$

At this point, it is necessary to apply the multi-dimensional analogous of the one-dimensional formula of integration by parts. This can be obtained by applying the divergence (Gauss) theorem by which

$$\int_{\Omega} \operatorname{div}(\mathbf{a}) \, d\Omega = \int_{\partial\Omega} \mathbf{a} \cdot \mathbf{n} \, d\gamma, \quad (3.14)$$

$\mathbf{a}(\mathbf{x}) = (a_1(\mathbf{x}), a_2(\mathbf{x}))^T$ being a sufficiently regular vector function and $\mathbf{n}(\mathbf{x}) = (n_1(\mathbf{x}), n_2(\mathbf{x}))^T$ the outward unit normal vector on $\partial\Omega$. If we apply (3.14) first to the function $\mathbf{a} = (\varphi\psi, 0)^T$ and then to $\mathbf{a} = (0, \varphi\psi)^T$, we get the relations

$$\int_{\Omega} \frac{\partial \varphi}{\partial x_i} \psi \, d\Omega = - \int_{\Omega} \varphi \frac{\partial \psi}{\partial x_i} \, d\Omega + \int_{\partial\Omega} \varphi \psi n_i \, d\gamma, \quad i = 1, 2. \quad (3.15)$$

Note also that if we take $\mathbf{a} = \mathbf{b}\varphi$, where \mathbf{b} and φ are respectively a vector and a scalar field, then (3.14) yields

$$\int_{\Omega} \varphi \operatorname{div} \mathbf{b} \, d\Omega = - \int_{\Omega} \mathbf{b} \cdot \nabla \varphi \, d\Omega + \int_{\partial\Omega} \mathbf{b} \cdot \mathbf{n} \varphi \, d\gamma \quad (3.16)$$

which is called *Green formula* for the divergence operator.

We exploit (3.15) by keeping into account the fact that $\Delta u = \operatorname{div} \nabla u = \sum_{i=1}^2 \frac{\partial}{\partial x_i} \left(\frac{\partial u}{\partial x_i} \right)$. Supposing that all the integrals that appear are meaningful, we find

$$\begin{aligned} - \int_{\Omega} \Delta u v \, d\Omega &= - \sum_{i=1}^2 \int_{\Omega} \frac{\partial}{\partial x_i} \left(\frac{\partial u}{\partial x_i} \right) v \, d\Omega \\ &= \sum_{i=1}^2 \int_{\Omega} \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_i} \, d\Omega - \sum_{i=1}^2 \int_{\partial\Omega} \frac{\partial u}{\partial x_i} v n_i \, d\gamma \\ &= \int_{\Omega} \sum_{i=1}^2 \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_i} \, d\Omega - \int_{\partial\Omega} \left(\sum_{i=1}^2 \frac{\partial u}{\partial x_i} n_i \right) v \, d\gamma. \end{aligned}$$

We obtain the following relation, called *Green formula* for the Laplacian

$$- \int_{\Omega} \Delta u v \, d\Omega = \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega - \int_{\partial\Omega} \frac{\partial u}{\partial n} v \, d\gamma. \quad (3.17)$$

Similarly to the one-dimensional case, the homogeneous Dirichlet problem will lead us to choosing test functions that vanish at the boundary, and, consequently, the boundary term that appears in (3.17) will in turn vanish.

Taking this into account, we get the following weak formulation for problem (3.13)

$$\text{find } u \in H_0^1(\Omega) : \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega = \int_{\Omega} f v \, d\Omega \quad \forall v \in H_0^1(\Omega), \quad (3.18)$$

f being a function of $L^2(\Omega)$ and having set

$$\begin{aligned} H^1(\Omega) &= \{v : \Omega \rightarrow \mathbb{R} \text{ s.t. } v \in L^2(\Omega), \frac{\partial v}{\partial x_i} \in L^2(\Omega), i = 1, 2\}, \\ H_0^1(\Omega) &= \{v \in H^1(\Omega) : v = 0 \text{ on } \partial\Omega\}. \end{aligned}$$

The derivatives must be understood in the sense of distributions and the condition $v = 0$ on $\partial\Omega$ in the sense of the traces (see Chap. 2).

In particular, we observe that if $u, v \in H_0^1(\Omega)$, then $\nabla u, \nabla v \in [L^2(\Omega)]^2$ and therefore $\nabla u \cdot \nabla v \in L^1(\Omega)$. The latter property is obtained by applying the following inequality

$$|(\nabla u, \nabla v)| \leq \|\nabla u\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)},$$

a direct consequence of the *Cauchy-Schwarz inequality* (2.16).

Hence, the integral appearing at the left of (3.18) is perfectly meaningful and so is the one appearing at the right.

Similarly to the one-dimensional case, it can be shown also in the two-dimensional case that problem (3.18) is equivalent to the following *variational problem*

$$\text{find } u \in V : \begin{cases} J(u) = \inf_{v \in V} J(v), \text{ with} \\ J(v) = \frac{1}{2} \int_{\Omega} |\nabla v|^2 d\Omega - \int_{\Omega} fv d\Omega, \end{cases}$$

having set $V = H_0^1(\Omega)$.

We can rewrite the weak formulation (3.18) in a more compact way by introducing the following form

$$a : V \times V \rightarrow \mathbb{R}, \quad a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v d\Omega, \quad (3.19)$$

and the following functional

$$F : V \rightarrow \mathbb{R}, \quad F(v) = \int_{\Omega} fv d\Omega$$

(functionals and forms are introduced in Chap. 2).

Problem (3.18) therefore becomes:

$$\text{find } u \in V : \quad a(u, v) = F(v) \quad \forall v \in V.$$

We notice that $a(\cdot, \cdot)$ is a bilinear form (that is, linear with respect to both its arguments), while F is a linear functional. Then

$$|F(v)| \leq \|f\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \leq \|f\|_{L^2(\Omega)} \|v\|_{H^1(\Omega)}.$$

Consequently, F is also bounded. Following the definition (2.2), its norm is bounded by $\|F\|_{V'} \leq \|f\|_{L^2(\Omega)}$. Consequently, F belongs to V' , the dual space of V , that is the set of linear and continuous functionals defined on V (see Sec. 2.1).

3.3.2 Equivalence, in the sense of distributions, between weak and strong form of the Dirichlet problem

We want to prove that the equations of problem (3.13) are actually satisfied by the weak solution, albeit only in the sense of distributions.

To this end, we consider the weak formulation (3.18). Let $\mathcal{D}(\Omega)$ now be the space of functions that are infinitely differentiable and with compact support in Ω (see Chap. 2). We recall that $\mathcal{D}(\Omega) \subset H_0^1(\Omega)$. Hence, by choosing $v = \varphi \in \mathcal{D}(\Omega)$ in (3.18), we have

$$\int_{\Omega} \nabla u \cdot \nabla \varphi d\Omega = \int_{\Omega} f \varphi d\Omega \quad \forall \varphi \in \mathcal{D}(\Omega). \quad (3.20)$$

By applying Green's formula (3.17) to the left-hand side of (3.20), we find

$$-\int_{\Omega} \Delta u \varphi \, d\Omega + \int_{\partial\Omega} \frac{\partial u}{\partial n} \varphi \, d\gamma = \int_{\Omega} f \varphi \, d\Omega \quad \forall \varphi \in \mathcal{D}(\Omega),$$

where the integrals are to be understood in the sense of duality, that is:

$$\begin{aligned} -\int_{\Omega} \Delta u \varphi \, d\Omega &= {}_{\mathcal{D}'(\Omega)} \langle -\Delta u, \varphi \rangle_{\mathcal{D}(\Omega)}, \\ \int_{\partial\Omega} \frac{\partial u}{\partial n} \varphi \, d\gamma &= {}_{\mathcal{D}'(\partial\Omega)} \langle \frac{\partial u}{\partial n}, \varphi \rangle_{\mathcal{D}(\partial\Omega)}. \end{aligned}$$

Since $\varphi \in \mathcal{D}(\Omega)$, the boundary integral is null, so that

$${}_{\mathcal{D}'(\Omega)} \langle -\Delta u - f, \varphi \rangle_{\mathcal{D}(\Omega)} = 0 \quad \forall \varphi \in \mathcal{D}(\Omega),$$

which corresponds to saying that $-\Delta u - f$ is the null distribution, that is

$$-\Delta u = f \quad \text{in } \mathcal{D}'(\Omega).$$

The differential equation (3.13) is therefore verified, as long as we intend the derivatives in the sense of distributions and we interpret the equality between $-\Delta u$ and f not in a pointwise sense, but in the sense of distributions (and thus almost everywhere in Ω). Finally, the fact that u vanishes on the boundary (in the sense of traces) is a direct consequence of u being in $H_0^1(\Omega)$.

3.3.3 The problem with mixed, non homogeneous conditions

The problem we want to solve is now the following

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = g & \text{on } \Gamma_D, \\ \frac{\partial u}{\partial n} = \phi & \text{on } \Gamma_N, \end{cases} \quad (3.21)$$

where Γ_D and Γ_N yield a partition of $\partial\Omega$, that is $\Gamma_D \cup \Gamma_N = \partial\Omega$, $\overset{\circ}{\Gamma}_D \cap \overset{\circ}{\Gamma}_N = \emptyset$ (see Fig. 3.3).

In the case of the Neumann problem, where $\Gamma_D = \emptyset$, the data f and ϕ must verify the following *compatibility condition*

$$-\int_{\partial\Omega} \phi \, d\gamma = \int_{\Omega} f \, d\Omega \quad (3.22)$$

so that the problem can have a solution. Condition (3.22) is deduced by integrating the differential equation in (3.21) and applying the divergence theorem (3.14)

$$-\int_{\Omega} \Delta u \, d\Omega = -\int_{\Omega} \operatorname{div}(\nabla u) \, d\Omega = -\int_{\partial\Omega} \frac{\partial u}{\partial n} \, d\gamma.$$

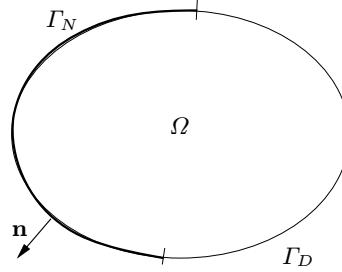


Fig. 3.3. The computational domain Ω

Moreover, we observe that also in the case of the Neumann problem, the solution is defined only up to an additive constant. In order to have uniqueness it would be sufficient, for example, to find a function with null average in Ω .

Let us now suppose that $\Gamma_D \neq \emptyset$ in order to ensure the uniqueness of the solution to the strong problem without conditions of compatibility on the data. Let us also suppose that $f \in L^2(\Omega)$, $g \in H^{1/2}(\Gamma_D)$ and $\phi \in L^2(\Gamma_N)$, having denoted by $H^{1/2}(\Gamma_D)$ the space of functions of $L^2(\Gamma_D)$ that are traces of functions of $H^1(\Omega)$ (see Sec. 2.4.3).

Thanks to Green's formula (3.17), we obtain from (3.21)

$$\int_{\Omega} \nabla u \cdot \nabla v \, d\Omega - \int_{\partial\Omega} \frac{\partial u}{\partial n} v \, d\gamma = \int_{\Omega} f v \, d\Omega. \quad (3.23)$$

We recall that $\partial u / \partial n = \phi$ on Γ_N and by exploiting the additivity of integrals, (3.23) becomes

$$\int_{\Omega} \nabla u \cdot \nabla v \, d\Omega - \int_{\Gamma_D} \frac{\partial u}{\partial n} v \, d\gamma - \int_{\Gamma_N} \phi v \, d\gamma = \int_{\Omega} f v \, d\Omega. \quad (3.24)$$

By imposing that the test function v vanish on Γ_D , the first boundary integral appearing in (3.24) vanishes. The mixed problem therefore admits the following weak formulation

$$\text{find } u \in V_g : \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega = \int_{\Omega} f v \, d\Omega + \int_{\Gamma_N} \phi v \, d\gamma \quad \forall v \in V, \quad (3.25)$$

having denoted by V the space

$$V = H_{\Gamma_D}^1(\Omega) = \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\}, \quad (3.26)$$

and having set

$$V_g = \{v \in H^1(\Omega) : v|_{\Gamma_D} = g\}.$$

The formulation (3.25) is not satisfactory, not only because the choice of spaces is “asymmetrical” ($v \in V$, while $u \in V_g$), but mainly because V_g is an affine manifold, but not a subspace of $H^1(\Omega)$ (indeed, it is not true that linear combinations of elements of V_g are still elements of V_g).

We then proceed in a similar way as seen in Sec. 3.2.2. We suppose to know a function R_g , called *lifting of the boundary data*, such that

$$R_g \in H^1(\Omega), \quad R_g|_{\Gamma_D} = g.$$

Furthermore, we suppose that such lifting be continuous, i.e. that

$$\exists C > 0 : \|R_g\|_{H^1(\Omega)} \leq C \|g\|_{H^{1/2}(\Gamma_D)} \forall g \in H^{1/2}(\Gamma_D).$$

We set $\mathring{u} = u - R_g$ and we begin by observing that $\mathring{u}|_{\Gamma_D} = u|_{\Gamma_D} - R_g|_{\Gamma_D} = 0$, that is $\mathring{u} \in H^1_{\Gamma_D}(\Omega)$. Moreover, since $\nabla u = \nabla \mathring{u} + \nabla R_g$, problem (3.25) becomes

$$\text{find } \mathring{u} \in H^1_{\Gamma_D}(\Omega) : \quad a(\mathring{u}, v) = F(v) \quad \forall v \in H^1_{\Gamma_D}(\Omega), \quad (3.27)$$

having defined the bilinear form $a(\cdot, \cdot)$ as in (3.19), while the linear functional F now takes the form

$$F(v) = \int_{\Omega} fv \, d\Omega + \int_{\Gamma_N} \phi v \, d\gamma - \int_{\Omega} \nabla R_g \cdot \nabla v \, d\Omega.$$

The problem is now symmetric since the space where the (new) unknown solution is sought coincides with the test function space.

The Dirichlet conditions are said to be *essential* as they are imposed explicitly in the functional space in which the problem is set.

The Neumann conditions are instead said to be *natural*, as they are satisfied implicitly by the solution of the problem (to this end, see Sec. 3.3.4). This difference in treatment has important repercussions on the approximate problems.

Remark 3.3 The reduction of the problem to a “symmetric” form allows to obtain a linear system with a symmetric matrix when solving the problem numerically (for instance via the finite elements method). •

Remark 3.4 Building a lifting R_g of a boundary function with an arbitrary form can turn out to be problematic. Such task is simpler in the context of a numerical approximation, where one generally builds a lifting of an approximation of the g function (see Chap. 4). •

3.3.4 Equivalence, in the sense of distributions, between weak and strong form of the Neumann problem

Let us consider the non homogeneous Neumann problem

$$\begin{cases} -\Delta u + \sigma u = f & \text{in } \Omega, \\ \frac{\partial u}{\partial n} = \phi & \text{on } \partial\Omega, \end{cases} \quad (3.28)$$

where σ is a positive constant or, more generally, a function $\sigma \in L^\infty(\Omega)$ such that $\sigma(\mathbf{x}) \geq \alpha_0$ a.e. in Ω , for a well-chosen constant $\alpha_0 > 0$. Let us also suppose that $f \in L^2(\Omega)$ and that $\phi \in L^2(\partial\Omega)$. By proceeding as in Sec. 3.3.3, the following weak formulation can be derived

find $u \in H^1(\Omega)$:

$$\int_{\Omega} \nabla u \cdot \nabla v \, d\Omega + \int_{\Omega} \sigma u v \, d\Omega = \int_{\Omega} f v \, d\Omega + \int_{\partial\Omega} \phi v \, d\gamma \quad \forall v \in H^1(\Omega). \quad (3.29)$$

By taking $v = \varphi \in \mathcal{D}(\Omega)$ and counterintegrating by parts, we obtain

$$_{\mathcal{D}'(\Omega)} \langle -\Delta u + \sigma u - f, \varphi \rangle_{\mathcal{D}(\Omega)} = 0 \quad \forall \varphi \in \mathcal{D}(\Omega).$$

Hence

$$-\Delta u + \sigma u = f \quad \text{in } \mathcal{D}'(\Omega)$$

i.e.

$$-\Delta u + \sigma u - f = 0 \quad \text{a.e. in } \Omega. \quad (3.30)$$

In the case where $u \in C^2(\Omega)$ the application of Green's formula (3.17) in (3.29) leads to

$$\int_{\Omega} (-\Delta u + \sigma u - f)v \, d\Omega + \int_{\partial\Omega} \left(\frac{\partial u}{\partial n} - \phi \right) v \, d\gamma = 0 \quad \forall v \in H^1(\Omega),$$

and therefore, thanks to (3.30),

$$\frac{\partial u}{\partial n} = \phi \quad \text{on } \partial\Omega.$$

In the case where the solution u of (3.29) is only in $H^1(\Omega)$ the generalized Green formula can be used, which states that there exists a unique linear and continuous functional $g \in (H^{1/2}(\partial\Omega))'$ (called generalized normal derivative), which operates on the space $H^{1/2}(\partial\Omega)$ satisfying

$$\int_{\Omega} \nabla u \cdot \nabla v \, d\Omega = \langle -\Delta u, v \rangle + \ll g, v \gg \quad \forall v \in H^1(\Omega).$$

We have denoted by $\langle \cdot, \cdot \rangle$ the duality between $H^1(\Omega)$ and its dual, and by $\ll \cdot, \cdot \gg$ the duality between $H^{1/2}(\partial\Omega)$ and its dual. Clearly g coincides with the classical normal derivative of u if u has sufficient regularity. For the sake of simplicity we use the notation $\partial u / \partial n$ for the generalized normal derivative in the remainder of this chapter. We therefore obtain that for $v \in H^1(\Omega)$

$$\langle -\Delta u + \sigma u - f, v \rangle + \ll \partial u / \partial n - \phi, v \gg = 0;$$

thanks to (3.30), we finally conclude that

$$\ll \partial u / \partial n - \phi, v \gg = 0 \quad \forall v \in H^1(\Omega),$$

and thus that $\partial u / \partial n = \phi$ a.e. on $\partial\Omega$.

3.4 More general elliptic problems

Let us now consider the problem

$$\begin{cases} -\operatorname{div}(\mu \nabla u) + \sigma u = f & \text{in } \Omega, \\ u = g & \text{on } \Gamma_D, \\ \mu \frac{\partial u}{\partial n} = \phi & \text{on } \Gamma_N, \end{cases} \quad (3.31)$$

where $\Gamma_D \cup \Gamma_N = \partial\Omega$ with $\overset{\circ}{\Gamma}_D \cap \overset{\circ}{\Gamma}_N = \emptyset$. We will suppose that $f \in L^2(\Omega)$, $\mu, \sigma \in L^\infty(\Omega)$. Furthermore, we suppose that $\exists \mu_0 > 0$ such that $\mu(\mathbf{x}) \geq \mu_0$ and $\sigma(\mathbf{x}) \geq 0$ a.e. in Ω . Only in the case where $\sigma = 0$ we will require that Γ_D be non-empty in order to prevent the solution from losing uniqueness. Finally, we will suppose that g and ϕ are sufficiently regular functions on $\partial\Omega$, for instance $g \in H^{1/2}(\Gamma_D)$ and $\phi \in L^2(\Gamma_N)$.

Also in this case, we proceed by multiplying the equation by a test function v and by integrating (once again formally) on the domain Ω :

$$\int_{\Omega} [-\operatorname{div}(\mu \nabla u) + \sigma u] v \, d\Omega = \int_{\Omega} fv \, d\Omega.$$

By applying Green's formula we obtain

$$\int_{\Omega} \mu \nabla u \cdot \nabla v \, d\Omega + \int_{\Omega} \sigma u v \, d\Omega - \int_{\partial\Omega} \mu \frac{\partial u}{\partial n} v \, d\gamma = \int_{\Omega} fv \, d\Omega,$$

which can also be rewritten as

$$\int_{\Omega} \mu \nabla u \cdot \nabla v \, d\Omega + \int_{\Omega} \sigma u v \, d\Omega - \int_{\Gamma_D} \mu \frac{\partial u}{\partial n} v \, d\gamma = \int_{\Omega} fv \, d\Omega + \int_{\Gamma_N} \mu \frac{\partial u}{\partial n} v \, d\gamma.$$

The function $\mu \partial u / \partial n$ is called *conormal derivative* of u associated to the operator $-\operatorname{div}(\mu \nabla u)$. On Γ_D we impose that the test function v is null, while on Γ_N we impose that the conormal derivative is equal to ϕ . We obtain

$$\int_{\Omega} \mu \nabla u \cdot \nabla v \, d\Omega + \int_{\Omega} \sigma u v \, d\Omega = \int_{\Omega} fv \, d\Omega + \int_{\Gamma_N} \phi v \, d\gamma.$$

Having denoted by R_g a lifting of g , we set $\overset{\circ}{u} = u - R_g$. The weak formulation of problem (3.31) is therefore

$$\begin{aligned} & \text{find } \overset{\circ}{u} \in H_{\Gamma_D}^1(\Omega) : \\ & \int_{\Omega} \mu \nabla \overset{\circ}{u} \cdot \nabla v \, d\Omega + \int_{\Omega} \sigma \overset{\circ}{u} v \, d\Omega = \int_{\Omega} fv \, d\Omega \\ & - \int_{\Omega} \mu \nabla R_g \cdot \nabla v \, d\Omega - \int_{\Omega} \sigma R_g v \, d\Omega + \int_{\Gamma_N} \phi v \, d\gamma \quad \forall v \in H_{\Gamma_D}^1(\Omega). \end{aligned}$$

We define the bilinear form

$$a : V \times V \rightarrow \mathbb{R}, \quad a(u, v) = \int_{\Omega} \mu \nabla u \cdot \nabla v \, d\Omega + \int_{\Omega} \sigma u v \, d\Omega,$$

and the linear and continuous functional

$$F : V \rightarrow \mathbb{R}, \quad F(v) = -a(R_g, v) + \int_{\Omega} fv \, d\Omega + \int_{\Gamma_N} \phi v \, d\gamma. \quad (3.32)$$

The previous problem can then be rewritten as

$$\text{find } \overset{\circ}{u} \in H_{\Gamma_D}^1(\Omega) : \quad a(\overset{\circ}{u}, v) = F(v) \quad \forall v \in H_{\Gamma_D}^1(\Omega). \quad (3.33)$$

A yet more general problem than (3.31) is the following

$$\begin{cases} Lu = f & \text{in } \Omega, \\ u = g & \text{on } \Gamma_D, \\ \frac{\partial u}{\partial n_L} = \phi & \text{on } \Gamma_N, \end{cases}$$

where, as usual, $\Gamma_D \cup \Gamma_N = \partial\Omega$, $\overset{\circ}{\Gamma}_D \cap \overset{\circ}{\Gamma}_N = \emptyset$, and having defined

$$Lu = - \sum_{i,j=1}^2 \frac{\partial}{\partial x_i} \left(a_{ij} \frac{\partial u}{\partial x_j} \right) + \sigma u.$$

The a_{ij} coefficients are functions defined on Ω . The derivative

$$\frac{\partial u}{\partial n_L} = \sum_{i,j=1}^2 a_{ij} \frac{\partial u}{\partial x_j} n_i \quad (3.34)$$

is called *conormal derivative* of u associated to the operator L (it coincides with the normal derivative when $Lu = -\Delta u$).

Let us suppose that $\sigma(\mathbf{x}) \in L^\infty(\Omega)$ and that $\exists \alpha_0 > 0$ such that $\sigma(\mathbf{x}) \geq \alpha_0$ a.e. in Ω . Furthermore, let us suppose that the coefficients $a_{ij} : \bar{\Omega} \rightarrow \mathbb{R}$ are continuous functions $\forall i, j = 1, 2$ and that there exists a positive constant α such that

$$\forall \xi = (\xi_1, \xi_2)^T \in \mathbb{R}^2 \quad \sum_{i,j=1}^2 a_{ij}(\mathbf{x}) \xi_i \xi_j \geq \alpha \sum_{i=1}^2 \xi_i^2 \quad \text{a.e. in } \Omega. \quad (3.35)$$

In such case, the weak formulation is still the same as (3.33), the functional F is still the one introduced in (3.32), while

$$a(u, v) = \int_{\Omega} \left(\sum_{i,j=1}^2 a_{ij} \frac{\partial u}{\partial x_j} \frac{\partial v}{\partial x_i} + \sigma u v \right) \, d\Omega. \quad (3.36)$$

It can be shown (see Exercise 2) that under the ellipticity hypothesis on the coefficients (3.35), this bilinear form is continuous and coercive, in the sense of definitions (2.6) and (2.9). These properties will be exploited in the analysis of well-posedness of problem (3.33) (see Sec. 3.5).

Elliptic problems for fourth-order operators are proposed in Exercises 4 and 6, while an elliptic problem deriving from the linear elasticity theory is analyzed in Exercise 7.

Remark 3.5 (Robin conditions) The case where Robin boundary conditions are enforced on the whole boundary, say

$$\mu \frac{\partial u}{\partial n} + \gamma u = 0 \quad \text{on } \partial\Omega,$$

requires more care. The weak form of the problem reads

$$\text{find } u \in H^1(\Omega) : a(u, v) = \int_{\Omega} f v d\Omega \quad \forall v \in H^1(\Omega),$$

where the bilinear form $a(u, v) = \int_{\Omega} \mu \nabla u \cdot \nabla v d\Omega + \int_{\Omega} \gamma u v d\Omega$ this time is not coercive if $\gamma < 0$. The analysis of this problem can be carried out by means of the Peetre-Tartar lemma, see [EG04].

•

3.5 Existence and uniqueness theorem

The following fundamental result holds (refer to Sec. 2.1 for definitions):

Lemma 3.1 (Lax-Milgram) *Let V be a Hilbert space, $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$ a continuous and coercive bilinear form, $F(\cdot) : V \rightarrow \mathbb{R}$ a linear and continuous functional. Then, there exists one unique solution to the problem*

$$\text{find } u \in V : \quad a(u, v) = F(v) \quad \forall v \in V. \quad (3.37)$$

Proof. This is based on two classical results of Functional Analysis: the Riesz representation theorem (see Theorem 2.1, Chap. 2), and the Banach closed range theorem. The interested reader can refer to, e.g., [QV94, Chap. 5]. ◇

The Lax-Milgram Lemma thus ensures that the weak formulation of an elliptic problem is well posed, as long as the hypotheses on the form $a(\cdot, \cdot)$ and on the functional $F(\cdot)$ are verified. Several consequences derive from this Lemma. We report one of the most important in the following Corollary.

Corollary 3.1 *The solution of (3.37) is bounded by the data, that is*

$$\|u\|_V \leq \frac{1}{\alpha} \|F\|_{V'},$$

where α is the coercivity constant of the bilinear form $a(\cdot, \cdot)$, while $\|F\|_{V'}$ is the norm of the functional F , see (2.2).

Proof. It is sufficient to choose $v = u$ in (3.37) and then to use the coercivity of the bilinear form $a(\cdot, \cdot)$. Indeed, we have

$$\alpha \|u\|_V^2 \leq a(u, u) = F(u).$$

On the other hand, since F is linear and continuous, it is also bounded and the upper bound

$$|F(u)| \leq \|F\|_{V'} \|u\|_V$$

holds, hence the thesis follows. \diamond

Remark 3.6 If the bilinear form $a(\cdot, \cdot)$ is also *symmetric*, that is

$$a(u, v) = a(v, u) \quad \forall u, v \in V,$$

then (3.37) is equivalent to the following variational problem (see Exercise 1)

$$\begin{cases} \text{find } u \in V : & J(u) = \min_{v \in V} J(v), \\ \text{with } J(v) = \frac{1}{2} a(v, v) - F(v). \end{cases} \quad (3.38)$$

•

3.6 Adjoint operator and adjoint problem

In this section we will introduce the concept of *adjoint* of a given operator in Hilbert spaces, as well as the *adjoint* (or *dual*) problem of a given boundary value problem. Then we will show how to obtain dual problems, with associated boundary conditions. The adjoint problem of a given differential problem plays a fundamental role, for instance, in the context of derivations of error estimates for Galerkin methods, both a priori and a posteriori (see Sections 4.5.4 and 4.6.4-4.6.5, respectively), but also for the solution of optimal control problems, as we will see in Chap. 16.

Let V be a Hilbert space with scalar product $(\cdot, \cdot)_V$ and norm $\|\cdot\|_V$, and let V' be its dual space. Let $a : V \times V \rightarrow \mathbb{R}$ be a continuous and coercive bilinear form and let $A : V \rightarrow V'$ be its associated elliptic operator, that is $A \in \mathcal{L}(V, V')$,

$$_{V'} \langle Av, w \rangle_V = a(v, w) \quad \forall v, w \in V. \quad (3.39)$$

Let $a^* : V \times V \rightarrow \mathbb{R}$ be the bilinear form defined by

$$a^*(w, v) = a(v, w) \quad \forall v, w \in V, \quad (3.40)$$

and consider the operator $A^* : V \rightarrow V'$ associated to the form $a^*(\cdot, \cdot)$, that is

$$V' \langle A^* w, v \rangle_V = a^*(w, v) \quad \forall v, w \in V. \quad (3.41)$$

Thanks to (3.40) we have the following relation, known as the *Lagrange identity*

$$V' \langle A^* w, v \rangle_V = V' \langle Av, w \rangle_V \quad \forall v, w \in V. \quad (3.42)$$

Note that this is precisely the equation that stands at the base of the definition (2.19) of the adjoint of a given operator A acting between a Hilbert space and its dual. For coherence with (2.19), we should have noted this operator A' . However, we prefer to denote it A^* because the latter notation is more customarily used in the context of elliptic boundary value problems.

If $a(\cdot, \cdot)$ is a symmetric form, $a^*(\cdot, \cdot)$ coincides with $a(\cdot, \cdot)$ and A^* with A . In such case A is said to be *self-adjoint*; A is said to be *normal* if $AA^* = A^*A$.

Naturally, the identity operator I is self-adjoint ($I = I^*$), while if an operator is self-adjoint, then it is also normal.

Some properties of the adjoint operators which are a consequence of the previous definition, are listed below:

- A being linear and continuous, then also A^* is, that is $A^* \in \mathcal{L}(V, V')$;
- $\|A^*\|_{\mathcal{L}(V, V')} = \|A\|_{\mathcal{L}(V, V')}$ (these norms are defined in (2.20));
- $(A + B)^* = A^* + B^*$;
- $(AB)^* = B^* A^*$;
- $(A^*)^* = A$;
- $(A^{-1})^* = (A^*)^{-1}$ (if A is invertible);
- $(\alpha A)^* = \alpha A^* \quad \forall \alpha \in \mathbb{R}$.

When we need to find the adjoint (or dual) problem of a given (primal) problem, we will use the Lagrange identity to characterize the differential equation of the dual problem, as well as its boundary conditions.

We provide an example of such a procedure, starting from a simple one-dimensional diffusion transport equation, completed by homogeneous Robin-Dirichlet boundary conditions

$$\begin{cases} Av = -v'' + v' = f, & x \in I = (0, 1), \\ v'(0) + \beta v(0) = 0, & v(1) = 0, \end{cases} \quad (3.43)$$

assuming β constant. Note that the weak form of this problem is

$$\text{find } u \in V \text{ s.t. } a(u, v) = \int_0^1 fv dx \quad \forall v \in V, \quad (3.44)$$

being $V = \{v \in H^1(0, 1) : v(1) = 0\}$ and

$$a : V \times V \rightarrow \mathbb{R}, \quad a(u, v) = \int_0^1 (u' - u)v' dx - (\beta + 1)u(0)v(0).$$

Thanks to (3.40) we obtain, $\forall v, w \in V$,

$$\begin{aligned} a^*(w, v) &= a(v, w) = \int_0^1 (v' - v)w' dx - (\beta + 1)v(0)w(0) \\ &= - \int_0^1 v(w'' + w') dx + [vw']_0^1 - (\beta + 1)v(0)w(0) \\ &= \int_0^1 (-w'' - w')v dx - [w'(0) + (\beta + 1)w(0)]v(0). \end{aligned}$$

Since definition (3.41) must hold, we will have

$$A^*w = -w'' - w' \quad \text{in } \mathcal{D}'(0, 1).$$

Moreover, as $v(0)$ is arbitrary, w will need to satisfy the boundary conditions

$$[w' + (\beta + 1)w](0) = 0, \quad w(1) = 0.$$

We observe that the transport field of the dual problem has an opposite direction with respect to that of the primal problem. Moreover, to homogeneous Robin-Dirichlet boundary conditions for the primal problem (3.43) correspond conditions of exactly the same nature for the dual problem.

The procedure illustrated for problem (3.43) can clearly be extended to the multidimensional case. In Table 3.3 we provide a list of several differential operators with boundary conditions, and their corresponding adjoint operators with associated boundary conditions. (On the functions appearing in the table, assume all the necessary regularity for the considered differential operators to be well-defined). We note, in particular, that to a given type of primal conditions do not necessarily correspond dual conditions of the same type, and that, for an operator that is not self-adjoint, to a conservative (resp. non-conservative) formulation of the primal problem there does correspond a non-conservative (resp. conservative) formulation of the dual one.

The extension of the analysis in the previous section to the non-linear case is not so immediate. For simplicity, we consider the one-dimensional problem

$$\begin{cases} A(v)v = -v'' + vv' = f, & x \in I = (0, 1), \\ v(0) = v(1) = 0, \end{cases} \quad (3.45)$$

having denoted by $A(v)$ the operator

$$A(v)\cdot = -\frac{d^2}{dx^2} + v\frac{d}{dx}. \quad (3.46)$$

Table 3.3. Differential operators and boundary conditions (B.C.) for the primal problem and corresponding dual (adjoint) operators (with associated boundary conditions)

<i>Primal operator</i>	<i>Primal B.C.</i>	<i>Dual (adjoint) operator</i>	<i>Dual B.C.</i>
$-\Delta u$	$u = 0 \text{ on } \Gamma$ $\frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega \setminus \Gamma$	$-\Delta w$	$w = 0 \text{ on } \Gamma,$ $\frac{\partial w}{\partial n} = 0 \text{ on } \partial\Omega \setminus \Gamma$
$-\Delta u + \sigma u$ $\operatorname{div} \mathbf{b} = 0$	$u = 0 \text{ on } \Gamma,$ $\frac{\partial u}{\partial n} + \gamma u = 0 \text{ on } \partial\Omega \setminus \Gamma$	$-\Delta w + \sigma w$ $\operatorname{div} \mathbf{b} = 0$	$w = 0 \text{ on } \Gamma,$ $\frac{\partial w}{\partial n} + \gamma w = 0 \text{ on } \partial\Omega \setminus \Gamma$
$-\Delta u + \mathbf{b} \cdot \nabla u + \sigma u,$ $\operatorname{div} \mathbf{b} = 0$	$u = 0 \text{ on } \Gamma,$ $\frac{\partial u}{\partial n} + \gamma u = 0 \text{ on } \partial\Omega \setminus \Gamma$	$-\Delta w - \mathbf{b} \cdot \nabla w + \sigma w,$ $\operatorname{div} \mathbf{b} = 0$	$w = 0 \text{ on } \Gamma,$ $\frac{\partial w}{\partial n} + (\mathbf{b} \cdot \mathbf{n} + \gamma)w = 0 \text{ on } \partial\Omega \setminus \Gamma$
$-\Delta u + \mathbf{b} \cdot \nabla u + \sigma u,$ $\operatorname{div} \mathbf{b} = 0$	$u = 0 \text{ on } \Gamma,$ $\frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega \setminus \Gamma$	$-\Delta w - \mathbf{b} \cdot \nabla w + \sigma w,$ $\operatorname{div} \mathbf{b} = 0$	$w = 0 \text{ on } \Gamma,$ $\frac{\partial w}{\partial n} + \mathbf{b} \cdot \mathbf{n} w = 0 \text{ on } \partial\Omega \setminus \Gamma$
$-\operatorname{div}(\mu \nabla u) + \mathbf{b} \cdot \nabla u + \sigma u$ $\operatorname{div} \mathbf{b} = 0$	$u = 0 \text{ on } \Gamma,$ $\mu \frac{\partial u}{\partial n} - \mathbf{b} \cdot \mathbf{n} u = 0 \text{ on } \partial\Omega \setminus \Gamma$	$-\operatorname{div}(\mu \nabla w) - \mathbf{b} \cdot \nabla w + \sigma w,$ $\operatorname{div} \mathbf{b} = 0$	$w = 0 \text{ on } \Gamma,$ $\mu \frac{\partial w}{\partial n} = 0 \text{ on } \partial\Omega \setminus \Gamma$
$-\operatorname{div}(\mu \nabla u) + \mathbf{b} \cdot \nabla u + \sigma u,$ $\operatorname{div} \mathbf{b} = 0$	$u = 0 \text{ on } \Gamma,$ $\mu \frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega \setminus \Gamma$	$-\operatorname{div}(\mu \nabla w) - \operatorname{div}(\mathbf{b} w) + \sigma w,$ $\operatorname{div} \mathbf{b} = 0$	$w = 0 \text{ on } \Gamma,$ $\mu \frac{\partial w}{\partial n} + \mathbf{b} \cdot \mathbf{n} w = 0 \text{ on } \partial\Omega \setminus \Gamma$

The Lagrange identity (3.42) is now generalized as

$${}_{V'} \langle A(v)u, w \rangle_V = {}_V \langle u, A^*(v)w \rangle_{V'} \quad (3.47)$$

for each $u \in D(A)$ and $w \in D(A^*)$, $D(A)$ being the set of functions of class C^2 that are null at $x = 0$ and $x = 1$, and $D(A^*)$ the domain of the adjoint (or dual) operator A^* whose properties will be identified by imposing the fulfillment of (3.47). Starting from such identity, let us see which adjoint operator A^* and which dual boundary conditions we get for problem (3.45). By integrating by parts the diffusion term twice and the transport term of order one once, we obtain

$$\begin{aligned} {}_{V'} \langle A(v)u, w \rangle_V &= - \int_0^1 u'' w \, dx + \int_0^1 v u' w \, dx \\ &= \int_0^1 u' w' \, dx - u' w \Big|_0^1 - \int_0^1 (v w)' u \, dx + v u w \Big|_0^1 \\ &= - \int_0^1 u w'' \, dx + u w' \Big|_0^1 - u' w \Big|_0^1 - \int_0^1 (v w)' u \, dx + v u w \Big|_0^1. \end{aligned} \quad (3.48)$$

Let us analyze the boundary terms separately, by expliciting the contributions at both extrema. In order to guarantee (3.47), it must be

$$u(1) w'(1) - u(0) w'(0) - u'(1) w(1) + u'(0) w(0) + v(1) u(1) w(1) - v(0) u(0) w(0) = 0$$

for each u and $v \in D(A)$. We observe that the fact that u belongs to $D(A)$ allows us to immediately vanish the two first and two last terms, so that we end up having

$$-u'(1) w(1) + u'(0) w(0) = 0.$$

Since such relation must hold for each $u \in D(A)$, we must choose homogeneous Dirichlet conditions for the dual operator, i.e.

$$w(0) = w(1) = 0. \quad (3.49)$$

Reverting to (3.48), we then have

$$\begin{aligned} {}_{V'} \langle A(v)u, w \rangle_V &= - \int_0^1 u'' w \, dx + \int_0^1 v u' w \, dx \\ &= - \int_0^1 u w'' \, dx - \int_0^1 (v w)' u \, dx = {}_V \langle u, A^*(v)w \rangle_{V'}. \end{aligned}$$

The adjoint operator A^* of the primal operator A defined in (3.46) therefore results to be

$$A^*(v) \cdot = -\frac{d^2 \cdot}{dx^2} + \frac{d}{dx} v.$$

while the dual boundary conditions are provided by (3.49). To conclude, we note that the dual problem is always linear, even though we started from a non-linear primal problem.

For more details on the differentiation and on the analysis of the adjoint problems, we refer the reader to, e.g., [Mar95].

3.7 Exercises

1. Prove that the weak problem (3.37) is equivalent to the variational problem (3.38) if the bilinear form is coercive and symmetric.

[*Solution:* let $u \in V$ be the solution of the weak problem and let w be a generic element of V . Thanks to the bilinearity and to the symmetry of the form, we find

$$\begin{aligned} J(u + w) &= \frac{1}{2}[a(u, u) + 2a(u, w) + a(w, w)] - [F(u) + F(w)] \\ &= J(u) + [a(u, w) - F(w)] + \frac{1}{2}a(w, w) = J(u) + \frac{1}{2}a(w, w). \end{aligned}$$

Thanks to the coercivity we then obtain that $J(u + w) \geq J(u) + (\alpha/2)\|w\|_V^2$, that is $\forall v \in V$ with $v \neq u$, $J(v) > J(u)$. Conversely, if u is a minimum for J , then by writing the extremality condition $\lim_{\delta \rightarrow 0} (J(u + \delta v) - J(u)) / \delta = 0$ we find (3.37).]

2. Prove that the bilinear form (3.36) is continuous and coercive under the hypotheses listed in the text on the coefficients.

[*Solution:* the bilinear form is obviously continuous. Thanks to the hypothesis (3.35) and to the fact that $\sigma \in L^\infty(\Omega)$ is positive a.e. in Ω , it is also coercive as

$$a(v, v) \geq \alpha|v|_{H^1(\Omega)}^2 + \alpha_0\|v\|_{L^2(\Omega)}^2 \geq \min(\alpha, \alpha_0)\|v\|_V^2 \quad \forall v \in V.$$

We point out that if $V = H^1(\Omega)$ then the condition $\alpha_0 > 0$ is necessary for the bilinear form to be coercive. In the case where $V = H_0^1(\Omega)$, it is sufficient that $\alpha_0 > -\alpha/C_\Omega^2$, C_Ω being the constant intervening in the Poincaré inequality. In this case, the equivalence between $\|\cdot\|_{H^1(\Omega)}$ and $|\cdot|_{H^1(\Omega)}$ can indeed be exploited. We have denoted by $|v|_{H^1(\Omega)} = \|\nabla v\|_{L^2(\Omega)}$ the seminorm of v in $H^1(\Omega)$ (see Example 2.11 in Chap. 2).]

3. Let $V = H_0^1(0, 1)$, $a : V \times V \rightarrow \mathbb{R}$ and $F : V \rightarrow \mathbb{R}$ be defined in the following way:

$$F(v) = \int_0^1 (-1 - 4x)v(x) dx, \quad a(u, v) = \int_0^1 (1 + x)u'(x)v'(x) dx.$$

Prove that the problem: find $u \in V$ s.t. $a(u, v) = F(v) \quad \forall v \in V$, admits a unique solution. Moreover, verify that it coincides with $u(x) = x^2 - x$.

[*Solution:* it can be easily shown that the bilinear form is continuous and coercive in V . Then, since F is a linear and continuous functional, thanks to the Lax-Milgram lemma, we can conclude that there exists a unique solution in V . We verify that the latter is indeed $u(x) = x^2 - x$. The latter function belongs for sure to V (since it is continuous and differentiable and such that $u(0) = u(1) = 0$). Moreover, from the relation

$$\int_0^1 (1+x)u'(x)v'(x) dx = - \int_0^1 ((1+x)u'(x))'v(x) dx = \int_0^1 (-1-4x)v(x) dx$$

that holds $\forall v \in V$, we deduce that in order for u to be a solution we must have $((1+x)u'(x))' = 1+4x$ almost everywhere in $(0, 1)$. Such property holds for the proposed u .]

4. Find the weak formulation of the problem

$$\begin{cases} \Delta^2 u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \\ \frac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega, \end{cases}$$

$\Omega \subset \mathbb{R}^2$ being a bounded open set with regular boundary $\partial\Omega$, $\Delta^2 = \Delta\Delta$ the bilaplacian operator and $f \in L^2(\Omega)$ an assigned function.

[*Solution:* the weak formulation, obtained by applying Green's formula twice to the bilaplacian operator, is

$$\text{find } u \in H_0^2(\Omega) : \int_{\Omega} \Delta u \Delta v d\Omega = \int_{\Omega} fv d\Omega \quad \forall v \in H_0^2(\Omega), \quad (3.50)$$

where $H_0^2(\Omega) = \{v \in H^2(\Omega) : v = 0, \partial v / \partial n = 0 \text{ on } \partial\Omega\}$.

5. For each function v of the Hilbert space $H_0^2(\Omega)$, defined in Exercise 4, it can be shown that the seminorm $|\cdot|_{H^2(\Omega)}$ defined as $|v|_{H^2(\Omega)} = (\int_{\Omega} |\Delta v|^2 d\Omega)^{1/2}$, is in fact equivalent to the norm $\|\cdot\|_{H^2(\Omega)}$. Using such property, prove that problem (3.50) admits a unique solution.

[*Solution:* let us set $V = H_0^2(\Omega)$. Then,

$$a(u, v) = \int_{\Omega} \Delta u \Delta v d\Omega \quad \text{and } F(v) = \int_{\Omega} fv d\Omega$$

are a bilinear form from $V \times V \rightarrow \mathbb{R}$ and a linear and continuous functional, respectively. To prove existence and uniqueness it is sufficient to invoke the Lax-Milgram lemma as the bilinear form is coercive and continuous. Indeed, thanks to the equivalence between norm and seminorm, there exist two positive constants α and M such that

$$a(u, u) = |u|_V^2 \geq \alpha \|u\|_V^2, \quad |a(u, v)| \leq M \|u\|_V \|v\|_V.$$

6. Write the weak formulation of the fourth-order problem

$$\begin{cases} -\operatorname{div}(\mu \nabla u) + \Delta^2 u + \sigma u = 0 & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \\ \frac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega, \end{cases}$$

by introducing appropriate functional spaces, knowing that $\Omega \subset \mathbb{R}^2$ is a bounded open set with regular boundary $\partial\Omega$ and that $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ are known functions defined on Ω .

[*Solution:* proceed as in the two previous exercises by supposing that the coefficients μ and σ lie in $L^\infty(\Omega)$.]

7. Let $\Omega \subset \mathbb{R}^2$ be a domain with a smooth boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$ and $\overset{\circ}{\Gamma}_D \cap \overset{\circ}{\Gamma}_N = \emptyset$. By introducing appropriate functional spaces, find the weak formulation of the following linear elasticity problem

$$\begin{cases} -\sum_{j=1}^2 \frac{\partial}{\partial x_j} \sigma_{ij}(\mathbf{u}) = f_i & \text{in } \Omega, \quad i = 1, 2, \\ u_i = 0 & \text{on } \Gamma_D, \quad i = 1, 2, \\ \sum_{j=1}^2 \sigma_{ij}(\mathbf{u}) n_j = g_i & \text{on } \Gamma_N, \quad i = 1, 2, \end{cases} \quad (3.51)$$

having denoted as usual by $\mathbf{n} = (n_1, n_2)^T$ the outward unit normal vector to $\partial\Omega$, by $\mathbf{u} = (u_1, u_2)^T$ the unknown vector, and by $\mathbf{f} = (f_1, f_2)^T$ and $\mathbf{g} = (g_1, g_2)^T$ two assigned vector functions. Moreover, it has been set for $i, j = 1, 2$,

$$\sigma_{ij}(\mathbf{u}) = \lambda \operatorname{div}(\mathbf{u}) \delta_{ij} + 2\mu \epsilon_{ij}(\mathbf{u}), \quad \epsilon_{ij}(\mathbf{u}) = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right),$$

λ and μ being two positive constants and δ_{ij} the Kronecker symbol. The system (3.51) allows to describe the displacement \mathbf{u} of an elastic body, homogeneous and isotropic, that occupies in its equilibrium position the region Ω , under the action of an external body force whose density is \mathbf{f} and of a surface charge distributed on Γ_N with intensity \mathbf{g} (see Fig. 3.4).

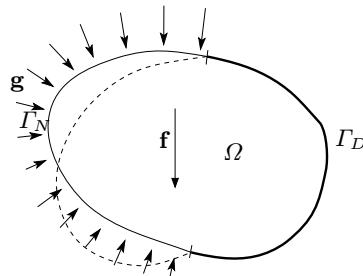


Fig. 3.4. A partially constrained body subject to the action of an external charge

[*Solution:* the weak formulation of (3.51) can be found by observing that $\sigma_{ij} = \sigma_{ji}$ and by using the following Green formula

$$\begin{aligned} \sum_{i,j=1}^2 \int_{\Omega} \sigma_{ij}(\mathbf{u}) \epsilon_{ij}(\mathbf{v}) \, d\Omega &= \sum_{i,j=1}^2 \int_{\partial\Omega} \sigma_{ij}(\mathbf{u}) n_j v_i \, d\gamma \\ &\quad - \sum_{i,j=1}^2 \int_{\Omega} \frac{\partial \sigma_{ij}(\mathbf{u})}{\partial x_j} v_i \, d\Omega. \end{aligned} \quad (3.52)$$

By assuming $\mathbf{v} \in V = (\mathbf{H}_{\Gamma_D}^1(\Omega))^2$ (the space of vectorial functions that have components $v_i \in \mathbf{H}_{\Gamma_D}^1(\Omega)$ for $i = 1, 2$), the weak formulation reads

find $\mathbf{u} \in V$ such that $a(\mathbf{u}, \mathbf{v}) = F(\mathbf{v}) \quad \forall \mathbf{v} \in V,$
with

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} \lambda \operatorname{div}(\mathbf{u}) \operatorname{div}(\mathbf{v}) \, d\Omega + 2\mu \sum_{i,j=1}^2 \int_{\Omega} \epsilon_{ij}(\mathbf{u}) \epsilon_{ij}(\mathbf{v}) \, d\Omega, \\ F(\mathbf{v}) &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\Omega + \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{v} \, d\gamma. \end{aligned}$$

In order for the integrals to make sense, it will be sufficient to require $\mathbf{f} \in (\mathbf{L}^2(\Omega))^2$ and $\mathbf{g} \in (\mathbf{L}^2(\Gamma_N))^2$.]

8. Prove, by applying the Lax-Milgram Lemma, that the solution of the weak formulation (3.52) exists and is unique under appropriate conditions on the regularity of the data and knowing that the following *Korn inequality* holds:

$$\exists C_0 > 0 : \sum_{i,j=1}^2 \int_{\Omega} \epsilon_{ij}(\mathbf{v}) \epsilon_{ij}(\mathbf{v}) \, d\Omega \geq C_0 \|\mathbf{v}\|_V^2 \quad \forall \mathbf{v} \in V.$$

[*Solution:* consider the weak formulation introduced in the solution to the previous exercise. The bilinear form defined in (3.52) is continuous and also coercive thanks to the Korn inequality. F is a linear and continuous functional; hence, by the Lax-Milgram lemma, the solution exists and is unique.]

4

The Galerkin finite element method for elliptic problems

In this chapter, we describe the numerical solution of the elliptic boundary-value problems considered in Chap. 3 by introducing the Galerkin method. We then illustrate the finite element method as a particular case. The latter will be further developed in the following chapters.

4.1 Approximation via the Galerkin method

As seen in Chap. 3.2, the weak formulation of a generic elliptic problem set on a domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, can be written in the following way

$$\text{find } u \in V : \quad a(u, v) = F(v) \quad \forall v \in V, \quad (4.1)$$

V being an appropriate Hilbert space, subspace of $H^1(\Omega)$, $a(\cdot, \cdot)$ being a continuous and coercive bilinear form from $V \times V$ in \mathbb{R} , $F(\cdot)$ being a continuous linear functional from V in \mathbb{R} . Under such hypotheses, the Lax-Milgram Lemma of Sec. 3.5 ensures existence and uniqueness of the solution.

Let V_h be a family of spaces that depends on a positive parameter h , such that

$$V_h \subset V, \quad \dim V_h = N_h < \infty \quad \forall h > 0.$$

The approximate problem takes the form

$$\text{find } u_h \in V_h : \quad a(u_h, v_h) = F(v_h) \quad \forall v_h \in V_h, \quad (4.2)$$

and is called *Galerkin problem*. Denoting with $\{\varphi_j, j = 1, 2, \dots, N_h\}$ a basis of V_h , it suffices that (4.2) be verified for each function of the basis, as all the functions in the space V_h are a linear combination of the φ_j . We will then require that

$$a(u_h, \varphi_i) = F(\varphi_i), \quad i = 1, 2, \dots, N_h. \quad (4.3)$$

Obviously, since $u_h \in V_h$,

$$u_h(\mathbf{x}) = \sum_{j=1}^{N_h} u_j \varphi_j(\mathbf{x}),$$

where the $u_j, j = 1, \dots, N_h$, are unknown coefficients. The equations (4.3) then become

$$\sum_{j=1}^{N_h} u_j a(\varphi_j, \varphi_i) = F(\varphi_i), \quad i = 1, 2, \dots, N_h. \quad (4.4)$$

We denote by \mathbf{A} the matrix (called *stiffness* matrix) with elements

$$a_{ij} = a(\varphi_j, \varphi_i),$$

and by \mathbf{f} the vector with components $f_i = F(\varphi_i)$. If we denote by \mathbf{u} the vector having as components the unknown coefficients u_j , (4.4) is equivalent to the linear system

$$\mathbf{A}\mathbf{u} = \mathbf{f}. \quad (4.5)$$

We point out some characteristics of the stiffness matrix that are independent of the basis chosen for V_h , but exclusively depend on the properties of the weak problem that is being approximated. Others instead, such as the condition number or the sparsity structure, depend on the basis under exam and are therefore reported in the sections dedicated to the specific numerical methods. For instance, bases formed by functions with small support are appealing as all the elements a_{ij} relating to basis functions having supports with null intersections will result to be null. More in general, from a computational viewpoint, the most convenient choices of V_h will be the ones requiring a modest computational effort for the computation of the matrix elements as well as the known term \mathbf{f} .

Theorem 4.1 *The matrix \mathbf{A} associated to the discretization of an elliptic problem with the Galerkin method is positive definite.*

Proof. We recall that a matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ is said to be positive definite if

$$\mathbf{v}^T \mathbf{B} \mathbf{v} \geq 0 \quad \forall \mathbf{v} \in \mathbb{R}^n \quad \text{and also } \mathbf{v}^T \mathbf{B} \mathbf{v} = 0 \Leftrightarrow \mathbf{v} = \mathbf{0}. \quad (4.6)$$

The correspondence

$$\mathbf{v} = (v_i) \in \mathbb{R}^{N_h} \leftrightarrow v_h(x) = \sum_{j=1}^{N_h} v_j \phi_j \in V_h \quad (4.7)$$

defines a bijection between the spaces \mathbb{R}^{N_h} and V_h . Given a generic vector $\mathbf{v} = (v_i)$ di \mathbb{R}^{N_h} , thanks to the bilinearity and coercivity of the form $a(\cdot, \cdot)$, we obtain

$$\begin{aligned} \mathbf{v}^T \mathbf{A} \mathbf{v} &= \sum_{j=1}^{N_h} \sum_{i=1}^{N_h} v_i a_{ij} v_j = \sum_{j=1}^{N_h} \sum_{i=1}^{N_h} v_i a(\varphi_j, \varphi_i) v_j \\ &= \sum_{j=1}^{N_h} \sum_{i=1}^{N_h} a(v_j \varphi_j, v_i \varphi_i) = a \left(\sum_{j=1}^{N_h} v_j \varphi_j, \sum_{i=1}^{N_h} v_i \varphi_i \right) \\ &= a(v_h, v_h) \geq \alpha \|v_h\|_V^2 \geq 0. \end{aligned}$$

Moreover, if $\mathbf{v}^T \mathbf{A} \mathbf{v} = 0$, then, by what we have just obtained, $\|v_h\|_V^2 = 0$ too, i.e. $v_h = 0$ and so $\mathbf{v} = \mathbf{0}$. Consequently, the thesis is proven as the two conditions in (4.6) are fulfilled. \diamond

Furthermore, the following property can be proven (see Exercise 4):

Property 4.1 *The matrix \mathbf{A} is symmetric if and only if the bilinear form $a(\cdot, \cdot)$ is symmetric.*

For instance, in the case of the Poisson problem with either Dirichlet (3.18) or mixed (3.27) boundary conditions, the matrix \mathbf{A} is symmetric and positive definite. The numerical solution of such a system can be efficiently performed both using direct methods such as the Cholesky factorization, and iterative methods such as the conjugate gradient method (see Chap. 7 and, e.g., [QSS07, Chap. 4]).

4.2 Analysis of the Galerkin method

In this section, we aim at studying the Galerkin method, and in particular at verifying three of its fundamental properties:

- *existence* and *uniqueness* of the discrete solution u_h ;
- *stability* of the discrete solution u_h ;
- *convergence* of u_h to the exact solution u of problem (4.1), for $h \rightarrow 0$.

4.2.1 Existence and uniqueness

The Lax-Milgram Lemma, stated in Sec. 3.5, holds for any Hilbert space, hence, in particular, for the space V_h , as the latter is a closed subspace of the Hilbert space V . Furthermore, the bilinear form $a(\cdot, \cdot)$ and the functional $F(\cdot)$ are the same as in the variational problem (4.1). The hypotheses required by the Lemma are therefore fulfilled. The following result then derives:

Corollary 4.1 *The solution of the Galerkin problem (4.2) exists and is unique.*

It is nonetheless instructive to provide a constructive proof of this Corollary without using the Lax-Milgram Lemma. As we have seen, indeed, the Galerkin problem (4.2) is equivalent to the linear system (4.5). Proving the existence and uniqueness for one means to automatically prove the existence and uniqueness of the other. We therefore focus our attention on the linear system (4.5).

The matrix \mathbf{A} is invertible as the unique solution of system $\mathbf{A}\mathbf{u} = \mathbf{0}$ is the identically null solution. This immediately descends from the fact that \mathbf{A} is positive definite. Consequently, the linear system (4.5) admits a unique solution, hence also its corresponding Galerkin problem admits a unique solution.

4.2.2 Stability

Corollary 3.1 allows us to provide the following stability result.

Corollary 4.2 *The Galerkin method is stable, uniformly with respect to h , as the following upper bound holds for the solution*

$$\|u_h\|_V \leq \frac{1}{\alpha} \|F\|_{V'}$$

The stability of the method guarantees that the norm $\|u_h\|_V$ of the discrete solution remains bounded for h tending to zero, uniformly with respect to h . Equivalently, it guarantees that $\|u_h - w_h\|_V \leq \frac{1}{\alpha} \|F - G\|_{V'}$, u_h and w_h being numerical solutions corresponding to two different data F and G .

4.2.3 Convergence

We now want to prove that the weak solution of the Galerkin problem converges to the solution of the weak problem (4.1) when h tends to zero. Consequently, by taking a sufficiently small h , it will be possible to approximate the exact solution u as accurately as desired by the Galerkin solution u_h .

Let us first prove the following consistency property.

Lemma 4.1 (Céa) *The Galerkin method is strongly consistent, that is*

$$a(u - u_h, v_h) = 0 \quad \forall v_h \in V_h. \quad (4.8)$$

Proof. Since $V_h \subset V$, the exact solution u satisfies the weak problem (4.1) for each element $v = v_h \in V_h$, hence we have

$$a(u, v_h) = F(v_h) \quad \forall v_h \in V_h. \quad (4.9)$$

By subtracting side to side (4.2) from (4.9), we obtain

$$a(u, v_h) - a(u_h, v_h) = 0 \quad \forall v_h \in V_h,$$

from which, thanks to the bilinearity of the form $a(\cdot, \cdot)$, the thesis follows. \diamond

Let us point out that (4.9) coincides with the definition of strong consistency given in (1.10).

Property (4.8) is known as Galerkin orthogonality. The reason is that, if $a(\cdot, \cdot)$ is symmetric, it defines a scalar product in V . Then, the consistency property is interpreted as the orthogonality with respect to the scalar product $a(\cdot, \cdot)$, between the

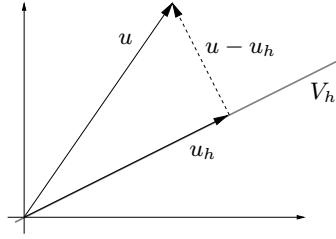


Fig. 4.1. Geometric interpretation of the Céa lemma

approximation error, $u - u_h$, and the subspace V_h . In this sense, analogously to the euclidian case, the solution u_h of the Galerkin method is said to be the *orthogonal projection* on V_h of the exact solution u . Among all elements of V_h , v_h is the one minimizing the distance to the exact solution u in the *energy norm*, i.e. in the following norm induced by the scalar product $a(\cdot, \cdot)$:

$$\|u - u_h\|_a = \sqrt{a(u - u_h, u - u_h)}.$$

Remark 4.1 The geometric interpretation of the Galerkin method makes sense only in the case where the form $a(\cdot, \cdot)$ is symmetric. However, this does not impair the generality of the method or its consistency property in the case where the bilinear form is not symmetric. •

Let us now consider the value taken by the bilinear form when both its arguments are equal to $u - u_h$. If v_h is an arbitrary element of V_h we obtain

$$a(u - u_h, u - u_h) = a(u - u_h, u - v_h) + a(u - u_h, v_h - u_h).$$

The last term is null thanks to (4.8), as $v_h - u_h \in V_h$. Moreover

$$|a(u - u_h, u - v_h)| \leq M \|u - u_h\|_V \|u - v_h\|_V,$$

having exploited the continuity of the bilinear form. On the other hand, by the coercivity of $a(\cdot, \cdot)$ it follows

$$a(u - u_h, u - u_h) \geq \alpha \|u - u_h\|_V^2$$

hence we have

$$\|u - u_h\|_V \leq \frac{M}{\alpha} \|u - v_h\|_V \quad \forall v_h \in V_h.$$

Such inequality holds for all functions $v_h \in V_h$ and therefore we find

$$\|u - u_h\|_V \leq \frac{M}{\alpha} \inf_{w_h \in V_h} \|u - w_h\|_V. \quad (4.10)$$

It is then evident that in order for the method to converge, it will be sufficient to require that, for h tending to zero, the space V_h tends to “fill” the entire space V . Precisely, it must turn out that

$$\lim_{h \rightarrow 0} \inf_{v_h \in V_h} \|v - v_h\|_V = 0 \quad \forall v \in V. \quad (4.11)$$

In that case, the Galerkin method is convergent and it can be written that

$$\lim_{h \rightarrow 0} \|u - u_h\|_V = 0.$$

The space V_h must therefore be carefully chosen in order to guarantee the density property (4.11). Once this requirement is satisfied, convergence will be verified in any case, independently of how u is made; conversely, the speed with which the discrete solution converges to the exact solution, i.e. the order of decay of the error with respect to h , will depend, in general, on both the choice of V_h and the regularity of u (see Theorem 4.3).

Remark 4.2 Obviously, $\inf_{v_h \in V_h} \|u - v_h\|_V \leq \|u - u_h\|_V$. Consequently, by (4.10), if $\frac{M}{\alpha}$ is of the order of unity, the error due to the Galerkin method can be identified with the best approximation error for u in V_h . In any case, both errors have the same infinitesimal order with respect to h . •

Remark 4.3 In the case where $a(\cdot, \cdot)$ is a symmetric bilinear form, and also continuous and coercive, then (4.10) can be improved as follows (see Exercise 5)

$$\|u - u_h\|_V \leq \sqrt{\frac{M}{\alpha}} \inf_{w_h \in V_h} \|u - w_h\|_V. \quad (4.12)$$

4.3 The finite element method in the one-dimensional case

Let us suppose that Ω be an interval (a, b) . The goal of this section is to create approximations of the space $H^1(a, b)$, that depend on a parameter h . To this end, we introduce a partition \mathcal{T}_h of (a, b) in $N + 1$ subintervals $K_j = (x_{j-1}, x_j)$, also called *elements*, having width $h_j = x_j - x_{j-1}$ with

$$a = x_0 < x_1 < \dots < x_N < x_{N+1} = b, \quad (4.13)$$

and set $h = \max_j h_j$.

Since the functions of $H^1(a, b)$ are continuous functions on $[a, b]$, we can construct the following family of spaces

$$X_h^r = \left\{ v_h \in C^0(\overline{\Omega}) : v_h|_{K_j} \in \mathbb{P}_r \ \forall K_j \in \mathcal{T}_h \right\}, \quad r = 1, 2, \dots \quad (4.14)$$

having denoted by \mathbb{P}_r the space of polynomials with degree lower than or equal to r in the variable x . The spaces X_h^r are all subspaces of $H^1(a, b)$ as they are constituted by differentiable functions except for at most a finite number of points (the vertices x_i of the partition \mathcal{T}_h). They represent possible choices for the space V_h , provided that the boundary conditions are properly incorporated. The fact that the functions of X_h^r are locally (elementwise) polynomials will make the stiffness matrix easy to compute.

We must now choose a basis $\{\varphi_i\}$ for the X_h^r space. It is convenient, by what exposed in Sec. 4.1, that the support of the generic basis function φ_i have non-empty intersection only with that of a negligible number of other functions of the basis. In such way, many elements of the stiffness matrix will be null. It is also convenient that the basis be *Lagrangian*: in that case, the coefficients of the expansion of a generic function $v_h \in X_h^r$ on the basis itself will be the values taken by v_h in carefully chosen points, which we call *nodes* and which, as we will see, generally form a superset of the vertices of \mathcal{T}_h . This does not prevent the use of non-lagrangian bases, especially in their hierarchical version (as we will see later). We now provide some examples of bases for the spaces X_h^1 and X_h^2 .

4.3.1 The space X_h^1

It is constituted by the piecewise continuous and linear functions on a partition \mathcal{T}_h of (a, b) of the form (4.13). Since only one straight line can pass by two different points and the functions of X_h^1 are continuous, the *degrees of freedom* of the functions of this space, i.e. the values that must be assigned to univocally define the functions themselves, will be equal to the number $N + 2$ of vertices of the partition itself. In this case, therefore, nodes and vertices coincide. Consequently, having assigned $N + 2$ basis functions φ_i , $i = 0, \dots, N + 1$, the whole space X_h^1 will be completely defined. The characteristic Lagrangian basis functions are characterized by the following property

$$\varphi_i \in X_h^1 \quad \text{such that} \quad \varphi_i(x_j) = \delta_{ij}, \quad i, j = 0, 1, \dots, N + 1,$$

δ_{ij} being the Kronecker delta. The function φ_i is therefore piecewise linear and equal to one at x_i and zero at the remaining nodes of the partition (see Fig. 4.2). Its expression is given by

$$\varphi_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{for } x_{i-1} \leq x \leq x_i, \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & \text{for } x_i \leq x \leq x_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \quad (4.15)$$

Obviously φ_i has the union of the only intervals $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$ as support, if $i \neq 0$ or $i \neq N + 1$ (for $i = 0$ or $i = N + 1$ the support will be limited to the

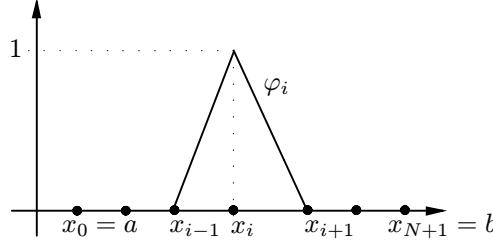


Fig. 4.2. The basis function of X_h^1 associated to node x_i

interval $[x_0, x_1]$ or $[x_N, x_{N+1}]$, respectively). Consequently, only the basis functions φ_{i-1} and φ_{i+1} have a support with non-empty intersection with that of φ_i , henceforth the stiffness matrix is tridiagonal as $a_{ij} = 0$ if $j \notin \{i-1, i, i+1\}$.

As visible in expression (4.15) the two basis functions φ_i and φ_{i+1} defined on each interval $[x_i, x_{i+1}]$, basically repeat themselves with no changes, up to a scaling factor linked to the length of the interval itself. In practice, the two basis functions φ_i and φ_{i+1} can be obtained by transforming two basis functions $\widehat{\varphi}_0$ and $\widehat{\varphi}_1$ built once and for all on a reference interval, typically the $[0, 1]$ interval.

To this end, it is sufficient to exploit the fact that the generic interval (x_i, x_{i+1}) of the partition of (a, b) can be obtained starting from the interval $(0, 1)$ via the linear transformation $\phi : [0, 1] \rightarrow [x_i, x_{i+1}]$ defined as

$$x = \phi(\xi) = x_i + \xi(x_{i+1} - x_i). \quad (4.16)$$

If we define the two basis functions $\widehat{\varphi}_0$ and $\widehat{\varphi}_1$ on $[0, 1]$ as

$$\widehat{\varphi}_0(\xi) = 1 - \xi, \quad \widehat{\varphi}_1(\xi) = \xi,$$

the basis functions φ_i and φ_{i+1} on $[x_i, x_{i+1}]$ will simply be given by

$$\varphi_i(x) = \widehat{\varphi}_0(\xi(x)), \quad \varphi_{i+1}(x) = \widehat{\varphi}_1(\xi(x))$$

since $\xi(x) = (x - x_i)/(x_{i+1} - x_i)$ (see Fig. 4.3 and 4.4).

This way of proceeding (defining the basis on a reference element and then transforming it on a specific element) will be of fundamental importance when considering problems in several dimensions.

4.3.2 The space X_h^2

The functions of X_h^2 are piecewise polynomials of degree 2 on each interval of \mathcal{T}_h and, consequently, are univocally set once the values they take in three distinct points of each interval K_j are assigned. To guarantee the continuity of the functions of X_h^2 two of these points will be the extrema of the generic interval of \mathcal{T}_h , the third will be the midpoint of the latter. The degrees of freedom of the space X_h^2 are therefore the

values of v_h taken at the extrema of the intervals composing the partition \mathcal{T}_h and at their midpoints. We order the nodes starting from $x_0 = a$ to $x_{2N+2} = b$; in such way the midpoints correspond to the nodes with odd indices, and the extrema to the nodes with even indices (refer to Exercise 6 for alternative numberings).

Exactly as in the previous case the Lagrangian basis for X_h^2 is the one formed by the functions

$$\varphi_i \in X_h^2 \quad \text{such that} \quad \varphi_i(x_j) = \delta_{ij}, \quad i, j = 0, 1, \dots, 2N + 2.$$

These are therefore piecewise quadratic functions that are equal to 1 at the node to which they are associated and are null at the remaining nodes. We report the explicit expression of the generic basis function associated to the extrema of the intervals in the partition:

$$(i \text{ even}) \quad \varphi_i(x) = \begin{cases} \frac{(x - x_{i-1})(x - x_{i-2})}{(x_i - x_{i-1})(x_i - x_{i-2})} & \text{if } x_{i-2} \leq x \leq x_i, \\ \frac{(x_{i+1} - x)(x_{i+2} - x)}{(x_{i+1} - x_i)(x_{i+2} - x_i)} & \text{if } x_i \leq x \leq x_{i+2}, \\ 0 & \text{otherwise.} \end{cases}$$

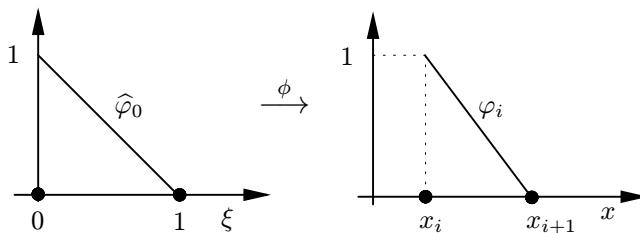


Fig. 4.3. The basis function φ_i in $[x_i, x_{i+1}]$ and the corresponding basis function $\hat{\varphi}_0$ on the reference element

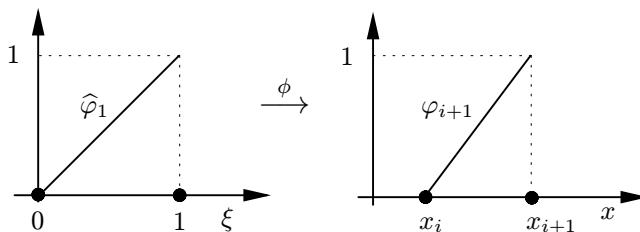


Fig. 4.4. The basis function φ_{i+1} in $[x_i, x_{i+1}]$ and the corresponding basis function $\hat{\varphi}_1$ on the reference element

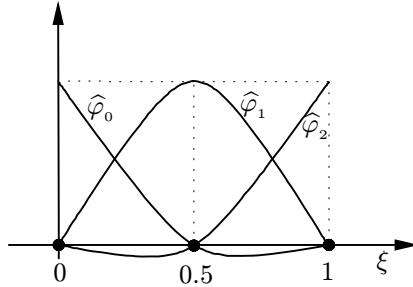


Fig. 4.5. The basis functions X_h^2 on the reference interval

For the midpoints of the intervals, we have

$$(i \text{ odd}) \quad \varphi_i(x) = \begin{cases} \frac{(x_{i+1} - x)(x - x_{i-1})}{(x_{i+1} - x_i)(x_i - x_{i-1})} & \text{if } x_{i-1} \leq x \leq x_{i+1}, \\ 0 & \text{otherwise.} \end{cases}$$

As in the case of linear finite elements, in order to describe the basis it is sufficient to provide the expression of the basis functions on the reference interval $[0, 1]$ and then to transform the latter via (4.16). We have

$$\widehat{\varphi}_0(\xi) = (1 - \xi)(1 - 2\xi), \quad \widehat{\varphi}_1(\xi) = 4(1 - \xi)\xi, \quad \widehat{\varphi}_2(\xi) = \xi(2\xi - 1).$$

We report a representation of these functions in Fig. 4.5. Note that the generic basis function φ_{2i+1} relative to node x_{2i+1} has a support coinciding with the element to which the midpoint belongs. For its peculiar form it is known as *bubble function*.

As previously anticipated, we can also introduce other non-lagrangian bases. A particularly interesting one is the one constructed (locally) by the three functions

$$\widehat{\psi}_0(\xi) = 1 - \xi, \quad \widehat{\psi}_1(\xi) = \xi, \quad \widehat{\psi}_2(\xi) = (1 - \xi)\xi.$$

A basis of this kind is said to be *hierarchical* as, to construct the basis for X_h^2 , it exploits the basis functions of the immediately lower-dimension space, X_h^1 . It is convenient from a computational viewpoint if one decides, during the approximation of a problem, to increase only locally, i.e. only for such elements, the degree of interpolation (that is if one intends to perform the so-called adaptivity in the degree, or *adaptivity of type p*).

The Lagrange polynomials are linearly independent by construction. In general however, such property must be verified to ensure that the set of chosen polynomials is effectively a basis. In the case of functions $\widehat{\psi}_0$, $\widehat{\psi}_1$ and $\widehat{\psi}_2$ we must verify that

$$\text{if } \alpha_0 \widehat{\psi}_0(\xi) + \alpha_1 \widehat{\psi}_1(\xi) + \alpha_2 \widehat{\psi}_2(\xi) = 0 \quad \forall \xi, \quad \text{then} \quad \alpha_0 = \alpha_1 = \alpha_2 = 0.$$

Indeed, the equation

$$\alpha_0 \widehat{\psi}_0(\xi) + \alpha_1 \widehat{\psi}_1(\xi) + \alpha_2 \widehat{\psi}_2(\xi) = \alpha_0 + \xi(\alpha_1 - \alpha_0 + \alpha_2) - \alpha_2 \xi^2 = 0$$

implies $\alpha_0 = 0$, $\alpha_2 = 0$ and therefore $\alpha_1 = 0$. We notice that the stiffness matrix in the case of finite elements of degree 2 will be pentadiagonal.

By proceeding in the same way it will be possible to generate bases for X_h^r with an arbitrary positive integer r : we point out however that as the polynomial degree increases, the number of degrees of freedom increases and so does the computational cost of solving the linear system (4.5). Moreover, a well known fact from the polynomial interpolation theory, the use of high degrees combined with equispaced node distributions, leads to less and less stable approximations, in spite of the theoretical increase in accuracy. A successful remedy is provided by the spectral element approximation that, using well-chosen nodes (the ones from the Gaussian quadrature), allows to generate approximations with arbitrarily high accuracy. To this purpose see Chap. 10.

4.3.3 The approximation with linear finite elements

We now examine how to approximate the following problem

$$\begin{cases} -u'' + \sigma u = f, & a < x < b, \\ u(a) = 0, & u(b) = 0, \end{cases}$$

whose weak formulation, as we have seen in the previous chapter, is

$$\text{find } u \in H_0^1(a, b) : \int_a^b u' v' dx + \int_a^b \sigma u v dx = \int_a^b f v dx \quad \forall v \in H_0^1(a, b).$$

As we did in (4.13), we introduce a decomposition \mathcal{T}_h of $(0, 1)$ in $N + 1$ subintervals K_j and use linear finite elements. We therefore introduce the space

$$V_h = \{v_h \in X_h^1 : v_h(a) = v_h(b) = 0\} \quad (4.17)$$

that is the space of piecewise linear functions that vanish at the boundary (a function of such space has been introduced in Fig. 4.6). This is a subspace of $H_0^1(a, b)$.

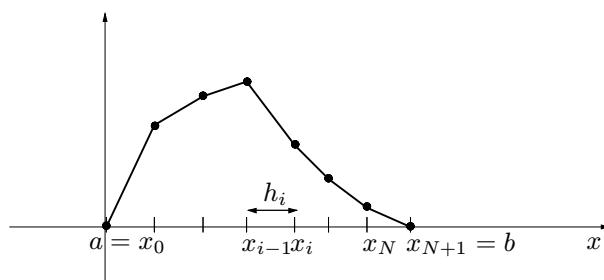


Fig. 4.6. Example of a function of V_h

The corresponding finite element problem is therefore given by

$$\text{find } u_h \in V_h : \int_a^b u'_h v'_h \, dx + \int_a^b \sigma u_h v_h \, dx = \int_a^b f v_h \, dx \quad \forall v_h \in V_h. \quad (4.18)$$

We use as a basis of X_h^1 the set of hat functions defined in (4.15) by caring to only consider the indices $1 \leq i \leq N$. By expressing u_h as a linear combination of such functions $u_h(x) = \sum_{i=1}^N u_i \varphi_i(x)$, and imposing that (4.18) be satisfied for each element of the basis of V_h , we obtain a system of N equations

$$\mathbf{A}\mathbf{u} = \mathbf{f}, \quad (4.19)$$

where

$$\begin{aligned} \mathbf{A} &= [a_{ij}], \quad a_{ij} = \int_a^b \varphi'_j \varphi'_i \, dx + \int_a^b \sigma \varphi_j \varphi_i \, dx; \\ \mathbf{u} &= [u_i]; \quad \mathbf{f} = [f_i], \quad f_i = \int_a^b f \varphi_i \, dx. \end{aligned}$$

Note that $u_i = u_h(x_i)$, $1 \leq i \leq N$, that is the finite element unknowns are the nodal values of the finite element solution u_h .

To find the numerical solution u_h it is now sufficient to solve the linear system (4.19).

In the case of linear finite elements, the stiffness matrix \mathbf{A} is not only sparse, but also results to be tridiagonal. To compute its elements, we proceed as follows. As we have seen it is not necessary to directly operate on the basis functions on the single intervals, but it is sufficient to refer to the ones defined on the reference interval: it will then be enough to appropriately transform the integrals that appear in the definition of the coefficients of \mathbf{A} .

A generic non-null element of the stiffness matrix is given by

$$a_{ij} = \int_a^b (\varphi'_i \varphi'_j + \sigma \varphi_i \varphi_j) \, dx = \int_{x_{i-1}}^{x_i} (\varphi'_i \varphi'_j + \sigma \varphi_i \varphi_j) \, dx + \int_{x_i}^{x_{i+1}} (\varphi'_i \varphi'_j + \sigma \varphi_i \varphi_j) \, dx.$$

Let us consider the first addendum by supposing $j = i - 1$. Evidently, via the coordinate transformation (4.16), we can re-write it as

$$\begin{aligned} &\int_{x_{i-1}}^{x_i} (\varphi'_i \varphi'_{i-1} + \sigma \varphi_i \varphi_{i-1}) \, dx = \\ &\int_0^1 [\varphi'_i(x(\xi)) \varphi'_{i-1}(x(\xi)) + \sigma(x(\xi)) \varphi_i(x(\xi)) \varphi_{i-1}(x(\xi))] h_i \, d\xi, \end{aligned}$$

having noted that $dx = d(x_{i-1} + \xi h_i) = h_i d\xi$. On the other hand $\varphi_i(x(\xi)) = \widehat{\varphi}_1(\xi)$ and $\varphi_{i-1}(x(\xi)) = \widehat{\varphi}_0(\xi)$. We also note that

$$\frac{d}{dx} \varphi_i(x(\xi)) = \frac{d\xi}{dx} \widehat{\varphi}'_1(\xi) = \frac{1}{h_i} \widehat{\varphi}'_1(\xi).$$

Similarly, we find that $\varphi'_{i-1}(x(\xi)) = (1/h_i) \widehat{\varphi}'_0(\xi)$. Hence, the element $a_{i,i-1}$ becomes

$$a_{i,i-1} = \int_0^1 \left(\frac{1}{h_i} \widehat{\varphi}'_1(\xi) \widehat{\varphi}'_0(\xi) + \sigma \widehat{\varphi}_1(\xi) \widehat{\varphi}_0(\xi) h_i \right) d\xi.$$

The advantage of this expression lies in the fact that in the case of constant coefficients, all the integrals appearing within matrix A can be computed once and for all. We will see in the multi-dimensional case that this way of proceeding maintains its importance also in the case of variable coefficients.

4.3.4 Interpolation operator and interpolation error

Let us set $I = (a, b)$. For each $v \in C^0(\overline{I})$, we define *interpolant* of v in the space of X_h^1 , determined by the partition \mathcal{T}_h , the function $\Pi_h^1 v$ such that

$$\Pi_h^1 v(x_i) = v(x_i) \quad \forall x_i, \text{ node of the partition, } i = 0, \dots, N + 1.$$

By using the Lagrangian basis $\{\varphi_i\}$ of the space X_h^1 , the interpolant can be expressed in the following way

$$\Pi_h^1 v(x) = \sum_{i=0}^{N+1} v(x_i) \varphi_i(x).$$

Hence, when v and a basis of X_h^1 are known, the interpolant of v is easy to compute. The operator $\Pi_h^1 : C^0(\overline{I}) \mapsto X_h^1$ mapping a function v to its interpolant $\Pi_h^1 v$ is called *interpolation operator*.

Analogously, we can define the operators $\Pi_h^r : C^0(\overline{I}) \mapsto X_h^r$, for all $r \geq 1$. Having denoted by $\Pi_{K_j}^r$ the local interpolation operator mapping a function v to the polynomial $\Pi_{K_j}^r v \in \mathbb{P}_r(K_j)$, interpolating v at the $r + 1$ nodes of the element $K_j \in \mathcal{T}_h$, we define $\Pi_h^r v$ as

$$\Pi_h^r v \in X_h^r : \quad \Pi_h^r v|_{K_j} = \Pi_{K_j}^r (v|_{K_j}) \quad \forall K_j \in \mathcal{T}_h. \quad (4.20)$$

Theorem 4.2 Let $v \in H^{r+1}(I)$, for $r \geq 1$, and let $\Pi_h^r v \in X_h^r$ be its interpolating function defined in (4.20). The following estimate of the interpolation error holds

$$|v - \Pi_h^r v|_{H^k(I)} \leq C_{k,r} h^{r+1-k} |v|_{H^{r+1}(I)} \quad \text{for } k = 0, 1. \quad (4.21)$$

The constants $C_{k,r}$ are independent of v and h . We recall that $H^0(I) = L^2(I)$ and that $|\cdot|_{H^0(I)} = \|\cdot\|_{L^2(I)}$.

Proof. We prove (4.21) for the case $r = 1$, and refer to [QV94, Chap. 3] or [Cia78] for the more general case. We start by observing that if $v \in H^{r+1}(I)$ then $v \in C^r(I)$. In particular, for $r = 1$, $v \in C^1(I)$. Let us set $e = v - \Pi_h^1 v$. Since $e(x_j) = 0$ for each node x_j , the Rolle theorem allows to conclude that there exist some $\xi_j \in K_j = (x_{j-1}, x_j)$, with $j = 1, \dots, N + 1$, for which we have $e'(\xi_j) = 0$.

$\Pi_h^1 v$ being a linear function in each interval K_j , we obtain that for $x \in K_j$

$$e'(x) = \int_{\xi_j}^x e''(s) ds = \int_{\xi_j}^x v''(s) ds,$$

from which we deduce that

$$|e'(x)| \leq \int_{x_{j-1}}^{x_j} |v''(s)| ds \quad \text{for } x \in K_j.$$

Now, by using the Cauchy-Schwarz inequality we obtain

$$|e'(x)| \leq \left(\int_{x_{j-1}}^{x_j} 1^2 ds \right)^{1/2} \left(\int_{x_{j-1}}^{x_j} |v''(s)|^2 ds \right)^{1/2} \leq h^{1/2} \left(\int_{x_{j-1}}^{x_j} |v''(s)|^2 ds \right)^{1/2}. \quad (4.22)$$

Hence,

$$\int_{x_{j-1}}^{x_j} |e'(x)|^2 dx \leq h^2 \int_{x_{j-1}}^{x_j} |v''(s)|^2 ds. \quad (4.23)$$

An upper bound for $e(x)$ can be obtained by noting that, for each $x \in K_j$, $e(x) = \int_{x_{j-1}}^x e'(s) ds$, and therefore, by applying inequality (4.22),

$$|e(x)| \leq \int_{x_{j-1}}^{x_j} |e'(s)| ds \leq h^{3/2} \left(\int_{x_{j-1}}^{x_j} |v''(s)|^2 ds \right)^{1/2}.$$

Hence,

$$\int_{x_{j-1}}^{x_j} |e(x)|^2 dx \leq h^4 \int_{x_{j-1}}^{x_j} |v''(s)|^2 ds. \quad (4.24)$$

By summing over the indices j from 1 to $N + 1$ in (4.23) and (4.24) we obtain the inequalities

$$\left(\int_a^b |e'(x)|^2 dx \right)^{1/2} \leq h \left(\int_a^b |v''(x)|^2 dx \right)^{1/2}$$

and

$$\left(\int_a^b |e(x)|^2 dx \right)^{1/2} \leq h^2 \left(\int_a^b |v''(x)|^2 dx \right)^{1/2}$$

respectively, that correspond to the desired estimates (4.21) for $r = 1$, with $C_{k,1} = 1$ and $k = 0, 1$. \diamond

4.3.5 Estimate of the finite element error in the H^1 norm

Thanks to the result (4.21) we can obtain an estimate of the approximation error of the finite element method.

Theorem 4.3 *Let $u \in V$ be the exact solution of the variational problem (4.1) (in our case $\Omega = I = (a, b)$) and u_h its approximate solution via the finite element method of degree r , i.e. the solution of problem (4.2) where $V_h = X_h^r \cap V$. Moreover, let $u \in H^{p+1}(I)$, for a suitable p such that $r \leq p$. Then, the following inequality, also called a priori error estimate, holds*

$$\|u - u_h\|_V \leq \frac{M}{\alpha} Ch^r |u|_{H^{r+1}(I)}, \quad (4.25)$$

C being a constant independent of u and h .

Proof. From (4.10), by setting $w_h = \Pi_h^r u$, the interpolant of degree r of u in the space V_h , we obtain

$$\|u - u_h\|_V \leq \frac{M}{\alpha} \|u - \Pi_h^r u\|_V.$$

The right-hand side can now be bounded from above via the interpolation error estimate (4.21) for $k = 1$, from which the thesis follows. \diamond

It follows from the latter theorem that, in order to increase the accuracy, two different strategies can be followed: reducing h , i.e. refining the grid, or increasing r , that is using finite elements of higher degree. However, the latter strategy makes sense only if the solution u is regular enough: as a matter of fact, from (4.25) we immediately infer that, if $u \in V \cap H^{p+1}(I)$, the maximum value of r that it makes sense to take is $r = p$. Values higher than r do not ensure a better rate of convergence: therefore

if the solution is not very regular it is not convenient to use finite elements of high degree, as the greater computational cost is not compensated by an improvement of the convergence. An interesting case is when the solution only has the minimum regularity ($p = 0$). From the relations (4.10) and (4.11) we obtain that there is convergence anyhow, but the estimate (4.25) is no longer valid. It is then impossible to say how the norm V of the error tends to zero when h decreases. We summarize these situations in Table 4.1.

Table 4.1. Order of convergence with respect to h for the finite element method for varying regularity of the solution and degree r of the finite elements. We have highlighted on each column the result corresponding to the “optimal” choice of the polynomial degree

	$r \ u \in H^1(I)$	$u \in H^2(I)$	$u \in H^3(I)$	$u \in H^4(I)$	$u \in H^5(I)$
1 converge	h^1	h^1	h^1	h^1	h^1
2 converge	h^1	h^2	h^2	h^2	h^2
3 converge	h^1	h^2	h^3	h^3	h^3
4 converge	h^1	h^2	h^3	h^4	h^4

In general, we can state that: if $u \in H^{p+1}(I)$, for a given $p > 0$, then there exists a constant C independent of u and h , such that

$$\|u - u_h\|_{H^1(I)} \leq Ch^s |u|_{H^{s+1}(I)}, \quad s = \min\{r, p\}. \quad (4.26)$$

4.4 Finite elements, simplices and barycentric coordinates

Before introducing finite element spaces in 2D and 3D domains we can attempt to provide a formal definition of *finite element*.

4.4.1 An abstract definition of finite element in the Lagrangian case

From the examples we considered we can deduce that there are three ingredients allowing to characterize univocally a finite element in the general case, i.e. independently of the dimension:

- the domain of definition K of the element. In the one-dimensional case it is an interval, in the two-dimensional case it is generally a triangle but it can also be a quadrilateral; in the three-dimensional case it can be a tetrahedron, a prism or a hexahedron;
- a space of polynomials Π_r of dimension N_r defined on K and a basis $\{\varphi_j\}_{j=1}^{N_r}$ of Π_r ;

- a set of functionals on Π_r , $\Sigma = \{\gamma_i : \Pi_r \rightarrow \mathbb{R}\}_{i=1}^{N_r}$ satisfying $\gamma_i(\varphi_j) = \delta_{ij}$, δ_{ij} being the Kronecker delta. These allow to univocally identify the coefficients $\{\alpha_j\}_{j=1}^{N_r}$ of the expansion of a polynomial $p \in \Pi_r$ with respect to the chosen basis, $p(x) = \sum_{j=1}^{N_r} \alpha_j \varphi_j(x)$. As a matter of fact, we have $\alpha_i = \gamma_i(p)$, $i = 1, \dots, N_r$. These coefficients are called *degrees of freedom* of the finite element.

In the case of *Lagrange finite elements* the chosen basis is provided by the Lagrange polynomials and the degree of freedom α_i is equal to the value taken by the polynomial p at a point \mathbf{a}_i of K , called *node*, that is we have $\alpha_i = p(\mathbf{a}_i)$, $i = 1, \dots, N_r$. We can then set, with a slight notation abuse, $\Sigma = \{\mathbf{a}_j\}_{j=1}^{N_r}$, as knowing the position of the nodes allows us to find the degrees of freedom (notice however that this is not true in general, think only of the case of the hierarchical basis introduced previously). In the remainder, we will exclusively refer to the case of Lagrange finite elements.

In the construction of a Lagrange finite element, the choice of nodes is not arbitrary. Indeed, the problem of interpolation on a given set K may be ill-posed. For this reason the following definition proves useful:

Definition 4.1 A set $\Sigma = \{\mathbf{a}_j\}_{j=1}^{N_r}$ of points of K is called *unisolvant* on Π_r if, given N_r arbitrary scalars α_j , $j = 1, \dots, N_r$, there exists a unique function $p \in \Pi_r$ such that

$$p(\mathbf{a}_j) = \alpha_j, \quad j = 1, \dots, N_r.$$

In such case, the triple (K, Σ, Π_r) is called *Lagrangian finite element*. In the case of Lagrangian finite elements, the element is generally recalled by citing the sole polynomial space: hence the linear finite elements introduced previously are called \mathbb{P}_1 , the quadratic ones \mathbb{P}_2 , and so forth.

As we have seen in the 1D case, for the finite elements \mathbb{P}_1 and \mathbb{P}_2 it is convenient to define the finite element starting from a reference element \widehat{K} ; typically this is the interval $(0, 1)$. It will commonly be the right triangle with vertices $(0, 0)$, $(1, 0)$ and $(0, 1)$ in the two-dimensional case (when using triangular elements). (See Sec. 4.4.2 for the case in arbitrary dimensions.) Hence, via a transformation ϕ , we move to the finite element defined on K . The transformation therefore concerns the finite element as a whole. More precisely, we observe that if $(\widehat{K}, \widehat{\Sigma}, \widehat{\Pi}_r)$ is a Lagrangian finite element and $\phi : \widehat{K} \rightarrow \mathbb{R}^d$ a continuous and injective application, and we define

$$K = \phi(\widehat{K}), \quad P_r = \{p : K \rightarrow \mathbb{R} : p \circ \phi \in \widehat{\Pi}_r\}, \quad \Sigma = \phi(\widehat{\Sigma}),$$

then (K, Σ, P_r) is still said to be a Lagrangian finite element. The space of polynomials defined on triangles and tetrahedra can be introduced as follows.

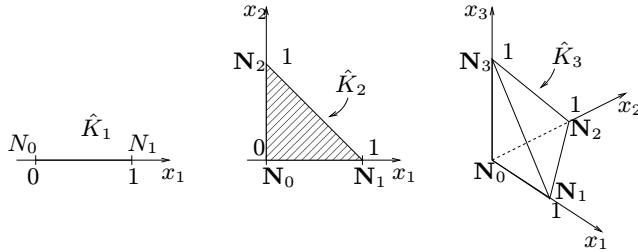


Fig. 4.7. The unitary simplex in \mathbb{R}^d , $d = 1, 2, 3$

4.4.2 Simplices

If $\{\mathbf{N}_0, \dots, \mathbf{N}_d\}$ are $d+1$ points in \mathbb{R}^d , $d \geq 1$, and the vectors $\{\mathbf{N}_1 - \mathbf{N}_0, \dots, \mathbf{N}_d - \mathbf{N}_0\}$ are linearly independent, then the convex hull of $\{\mathbf{N}_0, \dots, \mathbf{N}_d\}$ is called a *simplex*, and $\{\mathbf{N}_0, \dots, \mathbf{N}_d\}$ area called the *vertices* of the simplex. The *unitary simplex* of \mathbb{R}^d is the set

$$\hat{K}_d = \{\mathbf{x} \in \mathbb{R}^d : x_i \geq 0, 1 \leq i \leq d, \sum_{i=1}^d x_i \leq 1\}, \quad (4.27)$$

and it is a unitary interval in \mathbb{R}^1 , a unitary triangle in \mathbb{R}^2 , a unitary tetrahedron in \mathbb{R}^3 (see Fig. 4.7). Its vertices are ordered in such a way that the cartesian coordinates of \mathbf{N}_i are all null unless the i -th one that is equal to 1. On a d -dimensional simplex, the space of polynomials \mathbb{P}_r is defined as follows

$$\mathbb{P}_r = \{p(\mathbf{x}) = \sum_{\substack{0 \leq i_1, \dots, i_d \\ i_1 + \dots + i_d \leq r}} a_{i_1 \dots i_d} x_1^{i_1} \dots x_d^{i_d}, \quad a_{i_1 \dots i_d} \in \mathbb{R}\}. \quad (4.28)$$

Then

$$N_r = \dim \mathbb{P}_r = \binom{r+d}{r} = \frac{1}{d!} \prod_{k=1}^d (r+k). \quad (4.29)$$

4.4.3 Barycentric coordinates

For a given simplex K in \mathbb{R}^d (see Sect. 4.5.1) it is sometimes convenient to consider a coordinate frame alternative to the cartesian one, that of the *barycentric coordinates*. The latter are $d+1$ functions, $\{\lambda_0, \dots, \lambda_d\}$, defined as follows

$$\lambda_i : \mathbb{R}^d \rightarrow \mathbb{R}, \quad \lambda_i(\mathbf{x}) = 1 - \frac{(\mathbf{x} - \mathbf{N}_i) \cdot \mathbf{n}_i}{(\mathbf{N}_j - \mathbf{N}_i) \cdot \mathbf{n}_i}, \quad 0 \leq i \leq d. \quad (4.30)$$

For every $i = 0, \dots, d$ let F_i denote the *face* of K opposite to \mathbf{N}_i ; F_i is in fact a vertex if $d = 1$, an edge if $d = 2$, a triangle if $d = 3$. In (4.30), \mathbf{n}_i denotes the outward

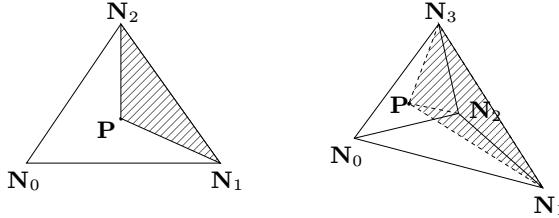


Fig. 4.8. The barycentric coordinate λ_i of the point \mathbf{P} is the ratio $\frac{|K_i|}{|K|}$ between the measure of simplex K_i (whose vertices are \mathbf{P} and $\{\mathbf{P}_j, j \neq i\}$) and that of the given simplex K (a triangle on the left, a tetrahedron on the right). The shadowed simplex is K_0

normal to F_i , while N_j is an arbitrary vertex belonging to F_i . The definition of λ_i is however independent of which vertex of F_i is chosen.

Barycentric coordinates have a geometrical meaning. Indeed, for every point \mathbf{P} belonging to K , its barycentric coordinate λ_i , $0 \leq i \leq d$, represents the ratio between the measure of the simplex K_i whose vertices are \mathbf{P} and the vertices of K sitting on the face F_i opposite to the vertex \mathbf{P}_i , and the measure of K . See Fig. 4.8.

Remark 4.4 Let us consider the unitary simplex \hat{K}_d , whose vertices $\{\hat{N}_0, \dots, \hat{N}_d\}$ are ordered in such a way that all the cartesian coordinates of \mathbf{N}_i are null unless x_i which is equal to one. Then

$$\lambda_i = x_i, \quad 1 \leq i \leq d, \quad \lambda_0 = 1 - \sum_{i=1}^d \lambda_i. \quad (4.31)$$

The barycentric coordinate λ_i is therefore an affine function that is equal to 1 at \mathbf{N}_i and vanishes on the face F_i opposite to \mathbf{N}_i .

On a general simplex K in \mathbb{R}^d , the following *partition of unity* property is satisfied

$$0 \leq \lambda_i(x) \leq 1, \quad \sum_{i=0}^d \lambda_i(\mathbf{x}) = 1 \quad \forall \mathbf{x} \in K. \quad (4.32)$$

•

A point \mathbf{P} belonging to the interior of K has therefore all its barycentric coordinates positive. This property is useful whenever one has to check which triangle in 2D or tetrahedron in 3D a given point belongs to, a situation that occurs when using Lagrangian derivatives (see Sect. 15.7.2) or computing suitable quantities (fluxes, streamlines, etc.) as a post-processing of finite element computations.

A noticeable property is that the center of gravity of K has all its barycentric coordinates equal to $(d+1)^{-1}$. Another remarkable property is that

$$\varphi_i = \lambda_i, \quad 0 \leq i \leq d, \quad (4.33)$$

where $\{\varphi_i, 0 \leq i \leq d\}$ are the characteristic Lagrangian functions on the simplex K of degree $r = 1$, that is

$$\varphi_i \in \mathbb{P}_1(K_d), \quad \varphi_i(\mathbf{N}_j) = \delta_{ij}, \quad 0 \leq j \leq d. \quad (4.34)$$

(See Fig. 4.10, left, for the nodes.)

For $r = 2$ the above identity (4.33) does not hold anymore, however the characteristic Lagrangian functions $\{\varphi_i\}$ can still be expressed in terms of the barycentric coordinates $\{\lambda_i\}$ as follows:

$$\begin{cases} \varphi_i = \lambda_i(2\lambda_i - 1), & 0 \leq i \leq d, \\ \varphi_{d+i+j} = 4\lambda_i\lambda_j, & 0 \leq i < j \leq d. \end{cases} \quad (4.35)$$

For $0 \leq i \leq d$, φ_i is the characteristic Lagrangian function associated to the vertex \mathbf{N}_i , while for $0 \leq i < j \leq d$, φ_{d+i+j} is the characteristic Lagrangian function associated to the midpoint of the edge whose endpoints are the vertices \mathbf{N}_i and \mathbf{N}_j (see Fig. 4.10, center).

The previous identities justify the name of “coordinates” that is used for the λ_i ’s. Indeed, if \mathbf{P} is a generic point of the simplex K , its cartesian coordinates $\{x_j^{(P)}, 1 \leq j \leq d\}$ can be expressed in terms of the barycentric coordinates $\{\lambda_i^{(P)}, 0 \leq i \leq d\}$ as follows

$$x_j^{(P)} = \sum_{i=0}^d \lambda_i^{(P)} x_j^{(i)}, \quad 1 \leq j \leq d, \quad (4.36)$$

where $\{x_j^{(i)}, 1 \leq j \leq d\}$ denote the cartesian coordinates of the i -th vertex \mathbf{N}_i of the simplex K .

4.5 The finite element method in the multi-dimensional case

In this section we extend the finite element method introduced previously for one-dimensional problems to the case of boundary-value problems in multi-dimensional regions. We will also specifically refer to the case of simplices. Many of the presented results are in any case immediately extensible to more general finite elements (see, for instance, [QV94]).

For the sake of simplicity, most often we will consider domains $\Omega \subset \mathbb{R}^2$ with polygonal shape and meshes (or grids) \mathcal{T}_h which represent their coverage with non-overlapping triangles. For this reason, \mathcal{T}_h is also called a triangulation. We refer to Chap. 6 for a more detailed description of the essential features of a generic grid \mathcal{T}_h . This way, the discretized domain

$$\Omega_h = \text{int}\left(\bigcup_{K \in \mathcal{T}_h} K\right)$$

represented by the internal part of the union of the triangles of \mathcal{T}_h perfectly coincides with Ω . We recall that we denote by $\text{int}(A)$ the internal part of the set A , that is the

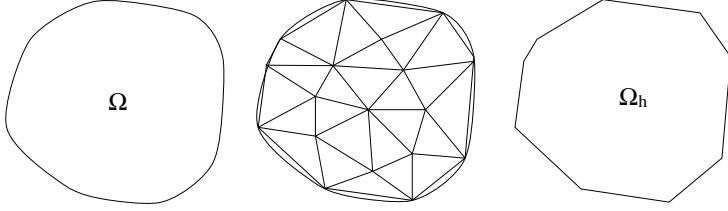


Fig. 4.9. Example of the grid of a non-polygonal domain. The grid induces an approximation Ω_h of the domain Ω such that $\lim_{h \rightarrow 0} \text{meas}(\Omega - \Omega_h) = 0$. This issue is not addressed in the present text. The interested reader may consult, for instance, [Cia78] or [SF73]

region obtained by excluding the boundary from A . In fact, we will not discuss the issue relating to the approximation of a non-polygonal domain with a finite element grid (see Fig. 4.9). Hence, from now on we will adopt the symbol Ω to denote without distinction both the computational domain and its (optional) approximation.

Also in the multidimensional case, the h parameter is related to the spacing of the grid. Having set $h_K = \text{diam}(K)$, for each $K \in \mathcal{T}_h$, where $\text{diam}(K) = \max_{x,y \in K} |x - y|$ is the *diameter* of element K , we define $h = \max_{K \in \mathcal{T}_h} h_K$. Moreover, we will impose that the grid satisfy the following *regularity* condition. Let ρ_K be the diameter of the circle inscribed in the triangle K (also called *sphericity* of K); a family of grids $\{\mathcal{T}_h, h > 0\}$ is said to be *regular* if, for a suitable $\delta > 0$, the condition

$$\frac{h_K}{\rho_K} \leq \delta \quad \forall K \in \mathcal{T}_h \quad (4.37)$$

is verified. We observe that condition (4.37) instantly excludes very deformed (i.e. stretched) triangles, and hence the option of using *anisotropic* computational grids.

On the other hand, anisotropic grids are often used in the context of fluid dynamics problems in the presence of boundary layers. See Remark 4.6, and especially references [AFG⁺00, DV02, FMP04]. Additional details on the generation of grids on two-dimensional domains are provided in Chap. 6.

We denote by \mathbb{P}_r the space of polynomials of global degree less than or equal to r , for $r = 1, 2, \dots$. According to the general formula (4.28) we find

$$\begin{aligned} \mathbb{P}_1 &= \{p(x_1, x_2) = a + bx_1 + cx_2, \text{ with } a, b, c \in \mathbb{R}\}, \\ \mathbb{P}_2 &= \{p(x_1, x_2) = a + bx_1 + cx_2 + dx_1x_2 + ex_1^2 + fx_2^2, \text{ with } a, b, c, d, e, f \in \mathbb{R}\}, \\ &\vdots \\ \mathbb{P}_r &= \{p(x_1, x_2) = \sum_{i,j \geq 0, i+j \leq r} a_{ij}x_1^i x_2^j, \text{ with } a_{ij} \in \mathbb{R}\}. \end{aligned}$$

According to (4.29), the spaces \mathbb{P}_r have dimension $(r+1)(r+2)/2$. For instance, it results that $\dim \mathbb{P}_1 = 3$, $\dim \mathbb{P}_2 = 6$ and $\dim \mathbb{P}_3 = 10$, hence on every element of the grid \mathcal{T}_h the generic function v_h is well defined whenever its value at 3, 6 resp. 10 suitably chosen nodes, is known (see Fig. 4.10). The nodes for linear ($r = 1$),

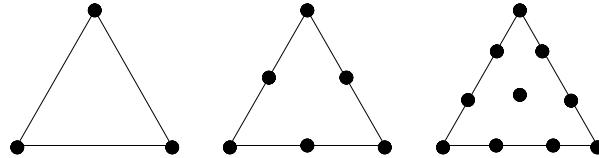


Fig. 4.10. Nodes for linear ($r = 1$, left), quadratic ($r = 2$, center) and cubic ($r = 3$, right) polynomials on a triangle. Such sets of nodes are unisolvant

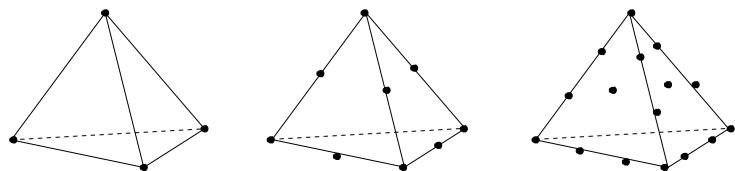


Fig. 4.11. Nodes for linear ($r = 1$, left), quadratic ($r = 2$, center) and cubic ($r = 3$, right) polynomials on a tetrahedron (only those on visible faces are shown)

quadratic ($r = 2$), and cubic ($r = 3$) polynomials on a three dimensional simplex are shown in Fig. 4.11.

4.5.1 Finite element solution of the Poisson problem

We introduce the space of finite elements

$$X_h^r = \{v_h \in C^0(\overline{\Omega}) : v_h|_K \in \mathbb{P}_r \ \forall K \in \mathcal{T}_h\}, \quad r = 1, 2, \dots \quad (4.38)$$

that is the space of globally continuous functions that are polynomials of degree r on the single triangles (elements) of the triangulation \mathcal{T}_h .

Moreover, we define

$$\overset{\circ}{X}_h^r = \{v_h \in X_h^r : v_h|_{\partial\Omega} = 0\}. \quad (4.39)$$

The spaces X_h^r e $\overset{\circ}{X}_h^r$ are suitable for the approximation of $H^1(\Omega)$, resp. $H_0^1(\Omega)$, thanks to the following property (for its proof see, e.g., [QV94]):

Property 4.2 A sufficient condition for a function v to belong to $H^1(\Omega)$ is that $v \in C^0(\overline{\Omega})$ and moreover that v belong to $H^1(K) \ \forall K \in \mathcal{T}_h$.

Having set $V_h = \overset{\circ}{X}_h^r$, we can introduce the following finite element problem for the approximation of the Poisson problem (3.1) with Dirichlet boundary condition (3.2), in the homogeneous case (that is with $g = 0$)

$$\text{find } u_h \in V_h : \int_{\Omega} \nabla u_h \cdot \nabla v_h \, d\Omega = \int_{\Omega} f v_h \, d\Omega \quad \forall v_h \in V_h. \quad (4.40)$$

As in the one-dimensional case, each function $v_h \in V_h$ is characterized, univocally, by the values it takes at the nodes \mathbf{N}_i , with $i = 1, \dots, N_h$, of the grid \mathcal{T}_h (excluding the boundary nodes where $v_h = 0$); consequently, a basis in the space V_h can be the set of the characteristic Lagrangian functions $\varphi_j \in V_h$, $j = 1, \dots, N_h$, such that

$$\varphi_j(\mathbf{N}_i) = \delta_{ij} = \begin{cases} 0 & i \neq j, \\ 1 & i = j, \end{cases} \quad i, j = 1, \dots, N_h. \quad (4.41)$$

In particular, if $r = 1$, the nodes are vertices of the elements, with the exception of those vertices belonging to the boundary of Ω , while the generic function φ_j is linear on each triangle and is equal to 1 at the node \mathbf{N}_j and 0 at all the other nodes of the triangulation (see Fig. 4.12).

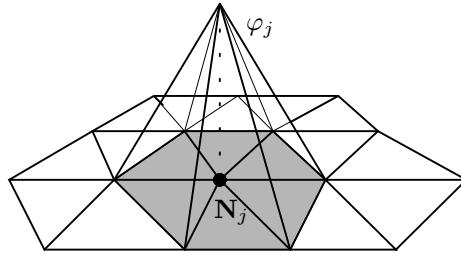


Fig. 4.12. The basis function φ_j of the space X_h^1 and its support

A generic function $v_h \in V_h$ can be expressed through a linear combination of the basis functions of V_h in the following way

$$v_h(\mathbf{x}) = \sum_{i=1}^{N_h} v_i \varphi_i(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega, \text{ with } v_i = v_h(\mathbf{N}_i). \quad (4.42)$$

By expressing the discrete solution u_h in terms of the basis $\{\varphi_j\}$ via (4.42), $u_h(\mathbf{x}) = \sum_{j=1}^{N_h} u_j \varphi_j(\mathbf{x})$, with $u_j = u_h(\mathbf{N}_j)$, and imposing that it verifies (4.40) for each function of the basis itself, we find the following linear system of N_h equations in the N_h unknowns u_j , equivalent to problem (4.40),

$$\sum_{j=1}^{N_h} u_j \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_i \, d\Omega = \int_{\Omega} f \varphi_i \, d\Omega, \quad i = 1, \dots, N_h. \quad (4.43)$$

The stiffness matrix has dimensions $N_h \times N_h$ and is defined as

$$\mathbf{A} = [a_{ij}] \quad \text{with} \quad a_{ij} = \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_i \, d\Omega. \quad (4.44)$$

Moreover, we introduce the vectors

$$\mathbf{u} = [u_j] \quad \text{with} \quad u_j = u_h(\mathbf{N}_j), \quad \mathbf{f} = [f_i] \quad \text{with} \quad f_i = \int_{\Omega} f \varphi_i \, d\Omega. \quad (4.45)$$

The linear system (4.43) can then be written as

$$\mathbf{A}\mathbf{u} = \mathbf{f}. \quad (4.46)$$

As in the one-dimensional case, the unknowns are the nodal values of the finite element solution. It is evident that, since the *support* of the generic function with basis φ_i is only formed by the triangles having node \mathbf{N}_i in common, \mathbf{A} is a sparse matrix. In particular, the number of non-null elements of \mathbf{A} is of the order of N_h as a_{ij} is different from zero only if \mathbf{N}_j and \mathbf{N}_i are nodes of the same triangle. It is not guaranteed instead that \mathbf{A} has a definite structure (e.g. banded), as that will depend on how the nodes are numbered.

Let us consider now the case of a *non-homogeneous* Dirichlet problem represented by equations (3.1)-(3.2). We have seen in the previous chapter that we can in any case resort to the homogeneous case through a lifting (also called extension, or prolongation) of the boundary datum. In the corresponding discrete problem we build a lifting of a well-chosen approximation of the boundary datum, by proceeding in the following way.

We denote by N_h the internal nodes of the grid \mathcal{T}_h and by N_h^t the total number, thus including the boundary nodes, that for the sake of simplicity we will suppose to be numbered last. The set of boundary nodes will then be formed by $\{\mathbf{N}_i, i = N_h + 1, \dots, N_h^t\}$. A possible approximation g_h of the boundary datum g can be obtained by interpolating g on the space formed by the trace functions on $\partial\Omega$ of functions of X_h^r . This can be written as a linear combination of the traces of the basis functions of X_h^r associated to the boundary nodes

$$g_h(\mathbf{x}) = \sum_{i=N_h+1}^{N_h^t} g(\mathbf{N}_i) \varphi_i|_{\partial\Omega}(\mathbf{x}) \quad \forall \mathbf{x} \in \partial\Omega. \quad (4.47)$$

Its lifting $R_{g_h} \in X_h^r$ is constructed as follows

$$R_{g_h}(\mathbf{x}) = \sum_{i=N_h+1}^{N_h^t} g(\mathbf{N}_i) \varphi_i(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega. \quad (4.48)$$

In Fig. 4.13, we provide an example of a possible lifting of a non-homogeneous Dirichlet boundary datum (3.2), in the case where g has a non-constant value. The finite element formulation of the Poisson problem then becomes

find $\overset{\circ}{u}_h \in V_h$:

$$\int_{\Omega} \nabla \overset{\circ}{u}_h \cdot \nabla v_h \, d\Omega = \int_{\Omega} f v_h \, d\Omega - \int_{\Omega} \nabla R_{g_h} \cdot \nabla v_h \, d\Omega \quad \forall v_h \in V_h. \quad (4.49)$$

The approximate solution will then be provided by $u_h = \overset{\circ}{u}_h + R_{g_h}$.

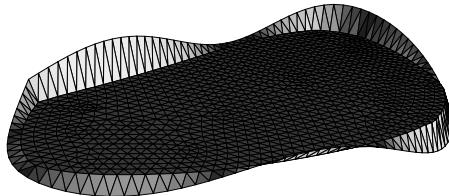


Fig. 4.13. Example of a lifting of a non-homogeneous Dirichlet boundary datum $u = g$, g being variable

Notice that, thanks to the particular lifting we adopted, we can give the following algebraic interpretation to (4.49)

$$\mathbf{A}\mathbf{u} = \mathbf{f} - \mathbf{B}\mathbf{g}$$

where \mathbf{A} and \mathbf{f} are defined as in (4.44) and (4.45), now with $u_j = \overset{\circ}{u}_h(\mathbf{N}_j)$. Having set $N_h^b = N_h^t - N_h$ (this is the number of boundary nodes), the vector $\mathbf{g} \in \mathbb{R}^{N_h^b}$ and the matrix $\mathbf{B} \in \mathbb{R}^{N_h \times N_h^b}$ have respectively the components

$$g_i = g(\mathbf{N}_{i+N_h}), \quad i = 1, \dots, N_h^b,$$

$$b_{ij} = \int_{\Omega} \nabla \varphi_{j+N_h} \cdot \nabla \varphi_i \, d\Omega, \quad i = 1, \dots, N_h, \quad j = 1, \dots, N_h^b.$$

Remark 4.5 Matrices \mathbf{A} and \mathbf{B} are both sparse. An efficient program will store exclusively their non-null elements. (See, e.g., [Saa96] for a description of possible storage formats for sparse matrices, and also Chap. 8). In particular, thanks to the special lifting we have adopted, in the \mathbf{B} matrix, all the lines corresponding to non-adjacent nodes to a boundary node will be null. (Two grid nodes are said to be adjacent if there exists an element $K \in \mathcal{T}_h$ to which they both belong.) •

4.5.2 Conditioning of the stiffness matrix

We have seen that the stiffness matrix $\mathbf{A} = [a(\varphi_j, \varphi_i)]$ associated to the Galerkin problem and therefore, in particular, to the finite element method, is positive definite; moreover \mathbf{A} is symmetric if the bilinear form $a(\cdot, \cdot)$ is symmetric.

For a symmetric and positive definite matrix, its condition number with respect to the norm 2 is given by

$$K_2(\mathbf{A}) = \frac{\lambda_{max}(\mathbf{A})}{\lambda_{min}(\mathbf{A})},$$

$\lambda_{max}(\mathbf{A})$ and $\lambda_{min}(\mathbf{A})$ being the maximum and minimum eigenvalues, respectively, of \mathbf{A} .

It can be proven that, both in the one-dimensional and the multi-dimensional case, the following relation holds for the stiffness matrix

$$K_2(\mathbf{A}) = Ch^{-2}, \quad (4.50)$$

where C is a constant independent of the h parameter, but dependent on the degree of the finite elements being used.

To prove (4.50), we recall that the eigenvalues of the matrix \mathbf{A} verify the relation

$$\mathbf{A}\mathbf{v} = \lambda_h \mathbf{v},$$

\mathbf{v} being the eigenvector corresponding to the eigenvalue λ_h . Let v_h be the function of the space V_h whose nodal values are the components v_i of \mathbf{v} , see (4.7). We suppose $a(\cdot, \cdot)$ to be symmetric, thus \mathbf{A} is symmetric and its eigenvalues are real and positive. We then have

$$\lambda_h = \frac{(\mathbf{A}\mathbf{v}, \mathbf{v})}{|\mathbf{v}|^2} = \frac{a(v_h, v_h)}{|\mathbf{v}|^2} \quad (4.51)$$

where $|\cdot|$ is the Euclidean vector norm. We suppose that the grid family $\{\mathcal{T}_h, h > 0\}$ is regular (i.e. satisfies (4.37)) and moreover is *quasi-uniform*, i.e. such that there exists a constant $\tau > 0$:

$$\min_{K \in \mathcal{T}_h} h_K \geq \tau h \quad \forall h > 0.$$

We now observe that, under the hypotheses made on \mathcal{T}_h , the following *inverse inequality* holds (for the proof, refer to [QV94])

$$\exists C_I > 0 \quad : \quad \forall v_h \in V_h, \quad \|\nabla v_h\|_{L^2(\Omega)} \leq C_I h^{-1} \|v_h\|_{L^2(\Omega)}, \quad (4.52)$$

the constant C_I being independent of h . We can now prove that there exist two constants $C_1, C_2 > 0$ such that, for each $v_h \in V_h$ as in (4.7), we have

$$C_1 h^d |\mathbf{v}|^2 \leq \|v_h\|_{L^2(\Omega)}^2 \leq C_2 h^d |\mathbf{v}|^2 \quad (4.53)$$

d being the spatial dimension, with $d = 1, 2, 3$. For the proof in the general case we refer to [QV94], Proposition 6.3.1. We here limit ourselves to proving the second inequality in the one-dimensional case ($d = 1$) and for linear finite elements. Indeed, on each element $K_i = [x_{i-1}, x_i]$, we have

$$\int_{K_i} v_h^2(x) dx = \int_{K_i} (v_{i-1}\varphi_{i-1}(x) + v_i\varphi_i(x))^2 dx,$$

with φ_{i-1} e φ_i defined according to (4.15). Then, a direct computation shows that

$$\int_{K_i} v_h^2(x) dx \leq 2 \left(v_{i-1}^2 \int_{K_i} \varphi_{i-1}^2(x) dx + v_i^2 \int_{K_i} \varphi_i^2(x) dx \right) = \frac{2}{3} h_i (v_{i-1}^2 + v_i^2)$$

with $h_i = x_i - x_{i-1}$. The inequality

$$\|v_h\|_{L^2(\Omega)}^2 \leq C h |\mathbf{v}|^2$$

with $C = 4/3$, can be found by simply summing the intervals K and observing that each nodal contribution v_i is counted twice.

On the other hand, from (4.51), we obtain, thanks to the continuity and coercivity of the bilinear form $a(\cdot, \cdot)$,

$$\alpha \frac{\|v_h\|_{H^1(\Omega)}^2}{|\mathbf{v}|^2} \leq \lambda_h \leq M \frac{\|v_h\|_{H^1(\Omega)}^2}{|\mathbf{v}|^2},$$

M and α being the continuity and coercivity constant, respectively. Now, $\|v_h\|_{H^1(\Omega)}^2 \geq \|v_h\|_{L^2(\Omega)}^2$ by the definition of the norm in $H^1(\Omega)$, while $\|v_h\|_{H^1(\Omega)} \leq C_3 h^{-1} \|v_h\|_{L^2(\Omega)}$ (for a well-chosen constant $C_3 > 0$) thanks to (4.52). Thus, by using inequalities (4.53), we obtain

$$\alpha C_1 h^d \leq \lambda_h \leq M C_3^2 C_2 h^{-2} h^d.$$

We therefore have

$$\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \leq \frac{M C_3^2 C_2}{\alpha C_1} h^{-2}$$

that is (4.50).

When the grid-size h decreases, the condition number of the stiffness matrix increases, and therefore the associated system becomes more and more ill-conditioned. In particular, if the datum \mathbf{f} of the linear system (4.46) is subject to a perturbation $\delta\mathbf{f}$ (i.e. it is affected by error), the latter in turn affects the solution with a perturbation $\delta\mathbf{u}$; it can then be proven that, if there are no perturbations on the matrix A , then

$$\frac{|\delta\mathbf{u}|}{|\mathbf{u}|} \leq K_2(A) \frac{|\delta\mathbf{f}|}{|\mathbf{f}|}.$$

It is evident that the higher is the conditioning number, the more the solution resents from the perturbation on the data. (On the other hand, notice that the latter is always affected by perturbations on the data caused by the inevitable roundoff errors introduced by the computer.)

As a further example we can study how conditioning affects the solution method. Consider, for instance, solving the linear system (4.46) using the conjugate gradient method (see Chap. 7). Then a sequence $\mathbf{u}^{(k)}$ of approximate solutions is iteratively constructed, converging to the exact solution \mathbf{u} . In particular, we have

$$\|\mathbf{u}^{(k)} - \mathbf{u}\|_A \leq 2 \left(\frac{\sqrt{K_2(A)} - 1}{\sqrt{K_2(A)} + 1} \right)^k \|\mathbf{u}^{(0)} - \mathbf{u}\|_A,$$

having denoted by $\|\mathbf{v}\|_A = \sqrt{\mathbf{v}^T A \mathbf{v}}$ the so-called “norm A ” of a generic vector $\mathbf{v} \in \mathbb{R}^{N_h}$. If we define

$$\rho = \frac{\sqrt{K_2(A)} - 1}{\sqrt{K_2(A)} + 1},$$

such quantity gives an idea of the convergence rate of the method: the closer ρ is to 0, the faster the method converges, the closer ρ is to 1, the slower the convergence

will be. Indeed, following (4.50), the more accurate one wants to be, by decreasing h , the more ill-conditioned the system will be, and therefore the more “problematic” its solution will turn out to be.

This calls for the system to be preconditioned, i.e. it is necessary to find an invertible matrix P , called *preconditioner*, such that

$$K_2(P^{-1}A) \ll K_2(A),$$

and then to apply the iterative method to the system preconditioned with P (see Chap. 7).

4.5.3 Estimate of the approximation error in the energy norm

Analogously to the one-dimensional case, for each $v \in C^0(\overline{\Omega})$ we define *interpolant* of v in the space of X_h^1 , determined by the grid \mathcal{T}_h , the function $\Pi_h^1 v$ such that

$$\Pi_h^1 v(\mathbf{N}_i) = v(\mathbf{N}_i) \quad \text{for each node } \mathbf{N}_i \text{ of } \mathcal{T}_h, \text{ for } i = 1, \dots, N_h.$$

If $\{\varphi_i\}$ is the Lagrangian basis of the space X_h^1 , then

$$\Pi_h^1 v(\mathbf{x}) = \sum_{i=1}^{N_h} v(\mathbf{N}_i) \varphi_i(\mathbf{x}).$$

The operator $\Pi_h^1 : C^0(\overline{\Omega}) \rightarrow X_h^1$, associating to a continuous function v its interpolant $\Pi_h^1 v$ is called *interpolation operator*.

Analogously, we can define an operator $\Pi_h^r : C^0(\overline{\Omega}) \rightarrow X_h^r$, for each integer $r \geq 1$. Having denoted by Π_K^r the local interpolation operator associated to a continuous function v the polynomial $\Pi_K^r v \in \mathbb{P}_r(K)$, interpolating v in the degrees of freedom of the element $K \in \mathcal{T}_h$, we define

$$\Pi_h^r v \in X_h^r : \quad \Pi_h^r v|_K = \Pi_K^r(v|_K) \quad \forall K \in \mathcal{T}_h. \quad (4.54)$$

We will suppose that \mathcal{T}_h belongs to a family of regular grids of Ω .

In order to obtain an estimate for the approximation error $\|u - u_h\|_V$ we follow a similar procedure to the one used in Theorem 4.3 for the one-dimensional case. The first step is to derive a suitable estimate for the interpolation error. To this end, we will obtain useful information starting from the geometric parameters of each triangle K , i.e. its diameter h_K and sphericity ρ_K . Moreover, we will exploit the affine and invertible transformation $F_K : \widehat{K} \rightarrow K$ between the reference triangle \widehat{K} and the generic triangle K (see Fig. 4.14). Such map is defined by $F_K(\hat{\mathbf{x}}) = B_K \hat{\mathbf{x}} + \mathbf{b}_K$, with $B_K \in \mathbb{R}^{2 \times 2}$ and $\mathbf{b}_K \in \mathbb{R}^2$, and satisfies the relation $F_K(\widehat{K}) = K$. We recall that the choice of the reference triangle \widehat{K} is not univocal.

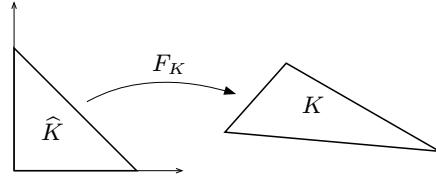


Fig. 4.14. The map F_K between the reference triangle \hat{K} and the generic triangle K

We will need some preliminary results.

Lemma 4.2 (Transformation of the seminorms) *For each integer $m \geq 0$ and each $v \in H^m(K)$, let $\hat{v} : \hat{K} \rightarrow \mathbb{R}$ be the function defined by $\hat{v} = v \circ F_K$. Then $\hat{v} \in H^m(\hat{K})$. Moreover, there exists a constant $C = C(m) > 0$ such that:*

$$|\hat{v}|_{H^m(\hat{K})} \leq C \|B_K\|^m |\det B_K|^{-\frac{1}{2}} |v|_{H^m(K)}, \quad (4.55)$$

$$|v|_{H^m(K)} \leq C \|B_K^{-1}\|^m |\det B_K|^{\frac{1}{2}} |\hat{v}|_{H^m(\hat{K})}, \quad (4.56)$$

$\|\cdot\|$ being the matrix norm associated to the euclidean vector norm $|\cdot|$, i.e.

$$\|B_K\| = \sup_{\xi \in \mathbb{R}^2, \xi \neq 0} \frac{|B_K \xi|}{|\xi|}. \quad (4.57)$$

Proof. Since $C^m(K) \subset H^m(K)$ with dense inclusion, for each $m \geq 0$, we can limit ourselves to proving the previous two inequalities for the functions of $C^m(K)$, then extending by density the result to the functions of $H^m(K)$. The derivatives in the remainders will therefore have to be intended in the classical sense. We recall that

$$|\hat{v}|_{H^m(\hat{K})} = \left(\sum_{|\alpha|=m} \int_{\hat{K}} |D^\alpha \hat{v}|^2 d\hat{x} \right)^{1/2},$$

by referring to Chap. 2.3 for the definition of the derivative D^α . By using the chain rule for the differentiation of composite functions, we obtain

$$\|D^\alpha \hat{v}\|_{L^2(\hat{K})} \leq C \|B_K\|^m \sum_{|\beta|=m} \|(D^\beta v) \circ F_K\|_{L^2(\hat{K})}.$$

Then

$$\|D^\alpha \hat{v}\|_{L^2(\hat{K})} \leq C \|B_K\|^m |\det B_K|^{-\frac{1}{2}} \|D^\alpha v\|_{L^2(K)}.$$

Inequality (4.55) follows after summing on the multi-index α , for $|\alpha| = m$. The result (4.56) can be proven by proceeding in a similar way. \diamond

Lemma 4.3 (Estimates for the norms $\|B_K\|$ and $\|B_K^{-1}\|$) We have the following upper bounds:

$$\|B_K\| \leq \frac{h_K}{\hat{\rho}}, \quad (4.58)$$

$$\|B_K^{-1}\| \leq \frac{\hat{h}}{\rho_K}, \quad (4.59)$$

\hat{h} and $\hat{\rho}$ being the diameter and the sphericity of the reference triangle \hat{K} .

Proof. Thanks to (4.57) we have

$$\|B_K\| = \frac{1}{\hat{\rho}} \sup_{\xi \in \mathbb{R}^2, |\xi|=\hat{\rho}} |B_K \xi|.$$

For each ξ , with $|\xi| = \hat{\rho}$, we can find two points \hat{x} and $\hat{y} \in \hat{K}$ such that $\hat{x} - \hat{y} = \xi$. Since $B_K \xi = F_K(\hat{x}) - F_K(\hat{y})$, we have $|B_K \xi| \leq h_K$, that is (4.58).

An analogous procedure leads to the result (4.59). \diamond

What we now need is an estimate in $H^m(\hat{K})$ of the seminorm of $(v - \Pi_K^r v) \circ F_K$, for each function v of $H^m(K)$. In the remainder, we denote the interpolant $\Pi_K^r v \circ F_K$ with $[\Pi_K^r v]^\wedge$. The nodes of K are $\mathbf{N}_i^K = F_K(\hat{\mathbf{N}}_i)$, $\hat{\mathbf{N}}_i$ being the nodes of \hat{K} , and, analogously, the basis functions $\hat{\varphi}_i$ defined on \hat{K} are identified by the relation $\hat{\varphi}_i = \varphi_i^K \circ F_K$, having denoted by φ_i^K the basis functions associated to the element K . Thus,

$$[\Pi_K^r v]^\wedge = \Pi_K^r v \circ F_K = \sum_{i=1}^{M_K} v(\mathbf{N}_i^K) \varphi_i^K \circ F_K = \sum_{i=1}^{M_K} v(F_K(\hat{\mathbf{N}}_i)) \hat{\varphi}_i = \Pi_{\hat{K}}^r \hat{v},$$

M_K being the number of nodes on K determined by the choice made for the degree r . It then follows that

$$|(v - \Pi_K^r v) \circ F_K|_{H^m(\hat{K})} = |\hat{v} - \Pi_{\hat{K}}^r \hat{v}|_{H^m(\hat{K})}. \quad (4.60)$$

In order to estimate the second member of the previous equality, we start by proving the following result:

Lemma 4.4 (Bramble-Hilbert Lemma) *Let $\widehat{L} : \mathbf{H}^{r+1}(\widehat{K}) \rightarrow \mathbf{H}^m(\widehat{K})$, with $m \geq 0$ and $r \geq 0$, be a linear and continuous transformation such that*

$$\widehat{L}(\hat{p}) = 0 \quad \forall \hat{p} \in \mathbb{P}_r(\widehat{K}). \quad (4.61)$$

Then, for each $\hat{v} \in \mathbf{H}^{r+1}(\widehat{K})$, we have

$$|\widehat{L}(\hat{v})|_{\mathbf{H}^m(\widehat{K})} \leq \|\widehat{L}\|_{\mathcal{L}(\mathbf{H}^{r+1}(\widehat{K}), \mathbf{H}^m(\widehat{K}))} \inf_{\hat{p} \in \mathbb{P}_r(\widehat{K})} \|\hat{v} + \hat{p}\|_{\mathbf{H}^{r+1}(\widehat{K})}, \quad (4.62)$$

where $\mathcal{L}(\mathbf{H}^{r+1}(\widehat{K}), \mathbf{H}^m(\widehat{K}))$ denotes the space of linear and continuous transformations $l : \mathbf{H}^{r+1}(\widehat{K}) \rightarrow \mathbf{H}^m(\widehat{K})$ the norm of which is

$$\|l\|_{\mathcal{L}(\mathbf{H}^{r+1}(\widehat{K}), \mathbf{H}^m(\widehat{K}))} = \sup_{v \in \mathbf{H}^{r+1}(\widehat{K}), v \neq 0} \frac{\|l(v)\|_{\mathbf{H}^m(\widehat{K})}}{\|v\|_{\mathbf{H}^{r+1}(\widehat{K})}}. \quad (4.63)$$

Proof. Let $\hat{v} \in \mathbf{H}^{r+1}(\widehat{K})$. For each $\hat{p} \in \mathbb{P}_r(\widehat{K})$, thanks to (4.61) and to the norm definition (4.63), we obtain

$$|\widehat{L}(\hat{v})|_{\mathbf{H}^m(\widehat{K})} = |\widehat{L}(\hat{v} + \hat{p})|_{\mathbf{H}^m(\widehat{K})} \leq \|\widehat{L}\|_{\mathcal{L}(\mathbf{H}^{r+1}(\widehat{K}), \mathbf{H}^m(\widehat{K}))} \|\hat{v} + \hat{p}\|_{\mathbf{H}^{r+1}(\widehat{K})}.$$

The result (4.62) can be deduced thanks to the fact that \hat{p} is arbitrary. \diamond

The following result (whose proof is given, e.g., in [QV94, Chap. 3]) provides the last necessary tool to obtain the estimate for the interpolation error that we are seeking.

Lemma 4.5 (Deny-Lions Lemma) *For each $r \geq 0$, there exists a constant $C = C(r, \widehat{K})$ such that*

$$\inf_{\hat{p} \in \mathbb{P}_r} \|\hat{v} + \hat{p}\|_{\mathbf{H}^{r+1}(\widehat{K})} \leq C |\hat{v}|_{\mathbf{H}^{r+1}(\widehat{K})} \quad \forall \hat{v} \in \mathbf{H}^{r+1}(\widehat{K}). \quad (4.64)$$

As a consequence of the two previous lemmas, we can provide the following

Corollary 4.3 *Let $\widehat{L} : \mathbf{H}^{r+1}(\widehat{K}) \rightarrow \mathbf{H}^m(\widehat{K})$, with $m \geq 0$ and $r \geq 0$, be a linear and continuous transformation such that $\widehat{L}(\hat{p}) = 0 \forall \hat{p} \in \mathbb{P}_r(\widehat{K})$. Then, there exists a constant $C = C(r, \widehat{K})$ such that, for each $\hat{v} \in \mathbf{H}^{r+1}(\widehat{K})$, we have*

$$|\widehat{L}(\hat{v})|_{\mathbf{H}^m(\widehat{K})} \leq C \|\widehat{L}\|_{\mathcal{L}(\mathbf{H}^{r+1}(\widehat{K}), \mathbf{H}^m(\widehat{K}))} |\hat{v}|_{\mathbf{H}^{r+1}(\widehat{K})}. \quad (4.65)$$

We are now able to prove the sought interpolation error estimate.

Theorem 4.4 (Local estimate of the interpolation error) *Let $r \geq 1$ and $0 \leq m \leq r + 1$. Then, there exists a constant $C = C(r, m, \hat{K}) > 0$ such that*

$$|v - \Pi_K^r v|_{H^m(K)} \leq C \frac{h_K^{r+1}}{\rho_K^m} |v|_{H^{r+1}(K)} \quad \forall v \in H^{r+1}(K). \quad (4.66)$$

Proof. From Property 2.3 we derive first of all that $H^{r+1}(K) \subset C^0(K)$, for $r \geq 1$. The interpolation operator Π_K^r thus results to be well defined in $H^{r+1}(K)$. By using, in order, the results (4.56), (4.60), (4.59) and (4.65), we have

$$\begin{aligned} |v - \Pi_K^r v|_{H^m(K)} &\leq C_1 \|B_K^{-1}\|^m |\det B_K|^{\frac{1}{2}} |\hat{v} - \Pi_{\hat{K}}^r \hat{v}|_{H^m(\hat{K})} \\ &\leq C_1 \frac{\hat{h}^m}{\rho_K^m} |\det B_K|^{\frac{1}{2}} \underbrace{|\hat{v} - \Pi_{\hat{K}}^r \hat{v}|_{H^m(\hat{K})}}_{\hat{L}(\hat{v})} \\ &\leq C_2 \frac{\hat{h}^m}{\rho_K^m} |\det B_K|^{\frac{1}{2}} \|\hat{L}\|_{\mathcal{L}(H^{r+1}(\hat{K}), H^m(\hat{K}))} |\hat{v}|_{H^{r+1}(\hat{K})} \\ &= C_3 \frac{1}{\rho_K^m} |\det B_K|^{\frac{1}{2}} |\hat{v}|_{H^{r+1}(\hat{K})}, \end{aligned}$$

$C_1 = C_1(m)$, $C_2 = C_2(r, m, \hat{K})$ and $C_3 = C_3(r, m, \hat{K})$ being suitably chosen constants. We note that the result (4.65) has been applied by identifying \hat{L} with the operator $I - \Pi_{\hat{K}}^r$, with $(I - \Pi_{\hat{K}}^r)\hat{p} = 0$, for $\hat{p} \in \mathbb{P}_r(\hat{K})$. Moreover the quantity \hat{h}^m and the norm of the operator \hat{L} have been included in the constant C_3 .

At this point, by applying (4.55) and (4.58) we obtain the result (4.66), that is

$$|v - \Pi_K^r v|_{H^m(K)} \leq C_4 \frac{1}{\rho_K^m} \|B_K\|^{r+1} |v|_{H^{r+1}(K)} \leq C_5 \frac{h_K^{r+1}}{\rho_K^m} |v|_{H^{r+1}(K)}, \quad (4.67)$$

$C_4 = C_4(r, m, \hat{K})$ and $C_5 = C_5(r, m, \hat{K})$ being two well-chosen constants. The quantity ρ^{r+1} generated by (4.58) and relating to the sphericity of the reference element has been directly included in the constant C_5 . \diamond

Finally, we can prove the global estimate for the interpolation error:

Theorem 4.5 (Global estimate for the interpolation error) *Let $\{\mathcal{T}_h\}_{h>0}$ be a family of regular grids of the domain Ω and let $m = 0, 1$ and $r \geq 1$. Then, there exists a constant $C = C(r, m, \hat{K}) > 0$ such that*

$$|v - \Pi_h^r v|_{H^m(\Omega)} \leq C \left(\sum_{K \in \mathcal{T}_h} h_K^{2(r+1-m)} |v|_{H^{r+1}(K)}^2 \right)^{1/2} \quad \forall v \in H^{r+1}(\Omega). \quad (4.68)$$

In particular, we obtain

$$|v - \Pi_h^r v|_{H^m(\Omega)} \leq C h^{r+1-m} |v|_{H^{r+1}(\Omega)} \quad \forall v \in H^{r+1}(\Omega). \quad (4.69)$$

Proof. Thanks to (4.66) and to the regularity condition (4.37), we have

$$\begin{aligned} |v - \Pi_h^r v|_{H^m(\Omega)}^2 &= \sum_{K \in \mathcal{T}_h} |v - \Pi_K^r v|_{H^m(K)}^2 \\ &\leq C_1 \sum_{K \in \mathcal{T}_h} \left(\frac{h_K^{r+1}}{\rho_K^m} \right)^2 |v|_{H^{r+1}(K)}^2 \\ &= C_1 \sum_{K \in \mathcal{T}_h} \left(\frac{h_K}{\rho_K} \right)^{2m} h_K^{2(r+1-m)} |v|_{H^{r+1}(K)}^2 \\ &\leq C_1 \delta^{2m} \sum_{K \in \mathcal{T}_h} h_K^{2(r+1-m)} |v|_{H^{r+1}(K)}^2, \end{aligned}$$

i.e. (4.68), with $C_1 = C_1(r, m, \hat{K})$ and $C = C_1 \delta^{2m}$. (4.69) follows thanks to the fact that $h_K \leq h$, for each $K \in \mathcal{T}_h$, and that

$$|v|_{H^p(\Omega)} = \left(\sum_{K \in \mathcal{T}_h} |v|_{H^p(K)}^2 \right)^{1/2},$$

for each integer $p \geq 0$. \diamond

In the $m = 0$ case, regularity of the grid is not necessary to obtain the estimate (4.69). This is no longer true for $m = 1$. As a matter of fact, given a triangle K and a function $v \in H^{r+1}(K)$, with $r \geq 1$, it can be proven that the following inequality holds [QV94],

$$|v - \Pi_h^r v|_{H^m(K)} \leq \tilde{C} \frac{h_K^{r+1}}{\rho_K^m} |v|_{H^{r+1}(K)}, \quad m = 0, 1,$$

with \tilde{C} independent of v and \mathcal{T}_h . Hence, in the $m = 1$ case for a family of regular grids we obtain (4.69) by setting $C = \delta \tilde{C}$, δ being the constant appearing in (4.37).

On the other hand, the need for a regularity condition can be proven by considering the particular case where, for each $C > 0$, a (non-regular) grid can be constructed for which inequality (4.69) is not true, as we are about to prove in the following example which relates to the case $r = 1$.

Example 4.1 Consider the triangle K_l illustrated in Fig. 4.15, with vertices $(0, 0)$, $(1, 0)$, $(0.5, l)$, with $l \leq \frac{\sqrt{3}}{2}$, and the function $v(x_1, x_2) = x_1^2$. Clearly $v \in H^2(K_l)$ and its linear interpolant on K_l is given by $\Pi_h^1 v(x_1, x_2) = x_1 - (4l)^{-1}x_2$. Since in this case $h_{K_l} = 1$, the inequality (4.69), applied to the single triangle K_l , would yield

$$|v - \Pi_h^1 v|_{H^1(K_l)} \leq C |v|_{H^2(K_l)}. \quad (4.70)$$

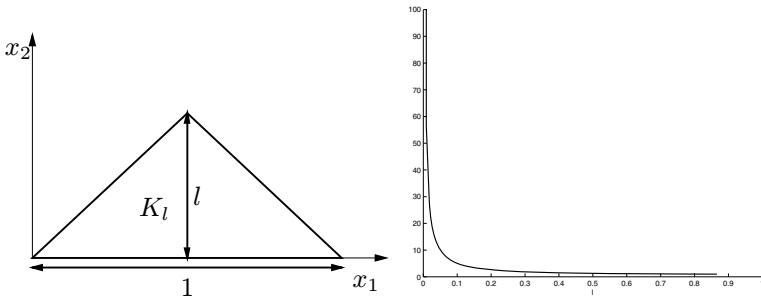


Fig. 4.15. The triangle K_l (left) and the behavior of the relation $|v - \Pi_h^1 v|_{H^1(K_l)}/|v|_{H^2(K_l)}$ as a function of l (right)

Let us now consider the behavior of the relation

$$\eta_l = \frac{|v - \Pi_h^1 v|_{H^1(K_l)}}{|v|_{H^2(K_l)}}$$

when l tends to zero, that is when the triangle is squeezed. We note that allowing l to tend to zero is equivalent to violating the regularity condition (4.37) as, for small enough values of l , $h_{K_l} = 1$, while, denoting by p_{K_l} the perimeter of K_l and by $|K_l|$ we denote the surface of the element K_l , the sphericity of K_l

$$\rho_{K_l} = \frac{4|K_l|}{p_{K_l}} = \frac{2l}{1 + \sqrt{1 + 4l^2}}$$

tends to zero. We have

$$\eta_l \geq \frac{\|\partial_{x_2}(v - \Pi_h^1 v)\|_{L^2(K_l)}}{|v|_{H^2(K_l)}} = \left(\frac{\int_{K_l} \left(\frac{1}{4l}\right)^2 d\mathbf{x}}{2l} \right)^{\frac{1}{2}} = \frac{1}{8l}.$$

Hence $\lim_{l \rightarrow 0} \eta_l = +\infty$ (see Fig. 4.15). Consequently, there cannot exist a constant C , independent of \mathcal{T}_h , for which (4.70) holds. ■

The theorem on interpolation error estimate immediately provides us with an estimate of the approximation error of the Galerkin method. The proof is analogous to that of Theorem 4.3 for the one-dimensional case. Indeed, it is sufficient to apply (4.10) and Theorem 4.5 (for $m = 1$) to obtain the following error estimate:

Theorem 4.6 *Let $u \in V$ be the exact solution of the variational problem (4.1) and u_h its approximate solution using the finite element method of degree r . If $u \in H^{r+1}(\Omega)$, then the following a priori error estimates hold:*

$$\|u - u_h\|_{H^1(\Omega)} \leq \frac{M}{\alpha} C \left(\sum_{K \in \mathcal{T}_h} h_K^{2r} |u|_{H^{r+1}(K)}^2 \right)^{1/2}, \quad (4.71)$$

$$\|u - u_h\|_{H^1(\Omega)} \leq \frac{M}{\alpha} C h^r |u|_{H^{r+1}(\Omega)}, \quad (4.72)$$

C being a constant independent of h and u .

Also in the multi-dimensional case, in order to increase the accuracy two different strategies can therefore be followed:

1. decreasing h , i.e. refine the grid;
2. increasing r , i.e. use finite elements of higher degree.

However, the latter approach can only be pursued if the solution u is regular enough. In general, we can say that, if $u \in H^{p+1}(\Omega)$ for some $p > 0$, then

$$\|u - u_h\|_{H^1(\Omega)} \leq C h^s |u|_{H^{s+1}(\Omega)}, \quad s = \min\{r, p\}, \quad (4.73)$$

as already observed in the one-dimensional case (see (4.26)). Moreover, it is possible to prove an error estimate in the maximum norm. For instance, if $r = 1$, one has

$$\|u - u_h\|_{L^\infty(\Omega)} \leq C h^2 |\log h| |u|_{W^{2,\infty}(\Omega)}$$

where C is a positive constant independent of h and the last term on the right hand side is the seminorm of u in the Sobolev space $W^{2,\infty}(\Omega)$ (see Sect. 2.5). For the proof of this and other error estimates in $W^{k,\infty}(\Omega)$ -norms see, e.g., [Cia78] and [BS94].

Remark 4.6 (Case of anisotropic grids) The interpolation error estimate (4.66) (and the consequent discretization error estimate) can be generalized in the case of *anisotropic grids*. In such case however, the left term of (4.66) takes a more complex expression: these estimates, in fact, because of their *directional* nature, must take into account information coming from characteristic directions associated to the single triangles which replace the “global” information concentrated in the seminorm $|v|_{H^{r+1}(K)}$. The interested reader can consult [Ape99, FP01]. Moreover, we refer to Fig. 4.18 and 11.14 for examples of anisotropic grids. •

4.5.4 Estimate of the approximation error in the L^2 norm

The inequality (4.72) provides an estimate of the approximation error in the energy norm. Analogously, it is possible to obtain an error estimate in the L^2 norm. Since the latter norm is weaker than the previous one, one must expect a higher convergence rate with respect to h .

Lemma 4.6 (Elliptic regularity) *Consider the homogeneous Dirichlet problem*

$$\begin{cases} -\Delta w = g & \text{in } \Omega, \\ w = 0 & \text{on } \partial\Omega, \end{cases}$$

with $g \in L^2(\Omega)$. If $\partial\Omega$ is sufficiently regular (for instance, if $\partial\Omega$ is a curve of class C^2 , or else if Ω is a convex polygon), then $w \in H^2(\Omega)$ and moreover there exists a constant $C > 0$ such that

$$\|w\|_{H^2(\Omega)} \leq C\|g\|_{L^2(\Omega)}. \quad (4.74)$$

For the proof see, e.g., [Bre86, Gri76].

Theorem 4.7 *Let $u \in V$ be the exact solution of the variational problem (4.1) and u_h its approximate solution obtained with the finite element method of degree r . Moreover, let $u \in H^{p+1}(\Omega)$ for a given $p > 0$. Then, the following a priori error estimate in the norm of $L^2(\Omega)$ holds*

$$\|u - u_h\|_{L^2(\Omega)} \leq Ch^{s+1}|u|_{H^{s+1}(\Omega)}, \quad s = \min\{r, p\}, \quad (4.75)$$

C being a constant independent of h and u .

Proof. We will limit ourselves to proving this result for the Poisson problem (3.13), the weak formulation of which is given in (3.18). Let $e_h = u - u_h$ be the approximation error and consider the following auxiliary Poisson problem (called *adjoint problem*, see Sec. 3.6) with known term given by the error function e_h

$$\begin{cases} -\Delta\phi = e_h & \text{in } \Omega, \\ \phi = 0 & \text{on } \partial\Omega, \end{cases} \quad (4.76)$$

whose weak formulation is

$$\text{find } \phi \in V : \quad a(\phi, v) = \int_{\Omega} e_h v \, d\Omega \quad \forall v \in V, \quad (4.77)$$

with $V = H_0^1(\Omega)$. Taking $v = e_h$ ($\in V$), we have

$$\|e_h\|_{L^2(\Omega)}^2 = a(\phi, e_h).$$

Since the bilinear form is symmetric, by the Galerkin orthogonality (4.8) we have

$$a(e_h, \phi_h) = a(\phi_h, e_h) = 0 \quad \forall \phi_h \in V_h.$$

It follows that

$$\|e_h\|_{L^2(\Omega)}^2 = a(\phi, e_h) = a(\phi - \phi_h, e_h). \quad (4.78)$$

Now, taking $\phi_h = \Pi_h^1 \phi$, applying the Cauchy-Schwarz inequality to the bilinear form $a(\cdot, \cdot)$ and using the interpolation error estimate (4.69) we obtain

$$\|e_h\|_{L^2(\Omega)}^2 \leq |e_h|_{H^1(\Omega)} |\phi - \phi_h|_{H^1(\Omega)} \leq |e_h|_{H^1(\Omega)} C h |\phi|_{H^2(\Omega)}. \quad (4.79)$$

Notice that the interpolation operator Π_h^1 can be applied to ϕ as, thanks to Lemma 4.6, $\phi \in H^2(\Omega)$ and thus, in particular, $\phi \in C^0(\overline{\Omega})$, thanks to property 2.3 in Chap. 2.

By applying Lemma 4.6 to the adjoint problem (4.76) we obtain the inequality

$$|\phi|_{H^2(\Omega)} \leq C \|e_h\|_{L^2(\Omega)}, \quad (4.80)$$

which, applied to (4.79), eventually provides

$$\|e_h\|_{L^2(\Omega)} \leq C h |e_h|_{H^1(\Omega)},$$

where C accounts for all the constants which appeared so far. By now exploiting the error estimate in the energy norm (4.72), we obtain (4.75). \diamond

Let us generalize the result we have just proven for the Poisson problem to the case of a generic elliptic boundary-value problem approximated with finite elements and for which an estimate of the approximation error in the energy norm such as (4.72) holds, and so does an elliptic regularity property analogous to the one expressed in Lemma 4.6.

In particular, let us consider the case where the bilinear form $a(\cdot, \cdot)$ is not necessarily symmetric. Let u be the exact solution of the problem

$$\text{find } u \in V : \quad a(u, v) = (f, v) \quad \forall v \in V, \quad (4.81)$$

and u_h the solution of the Galerkin problem

$$\text{find } u_h \in V_h : \quad a(u_h, v_h) = (f, v_h) \quad \forall v_h \in V_h.$$

Finally, suppose that the error estimate (4.72) holds and let us consider the following problem, which we will call *adjoint problem* of (4.81): for each $g \in L^2(\Omega)$,

$$\text{find } \phi = \phi(g) \in V : \quad a^*(\phi, v) = (g, v) \quad \forall v \in V, \quad (4.82)$$

where we have defined (see (3.40))

$$a^* : V \times V \rightarrow R, \quad a^*(w, v) = a(v, w) \quad \forall w, v \in V. \quad (4.83)$$

Obviously, should a be symmetric, the two problems coincide, as seen for instance in the case of problem (4.77).

Notice that the unknown is now the second argument of $a(\cdot, \cdot)$, while in the primal problem (4.81) the unknown is the first argument of $a(\cdot, \cdot)$. Let us suppose that for the solution u of the primal problem (4.81) an elliptic regularity result holds; it can then be verified that the same result is valid for the adjoint problem (4.82), that is

$$\exists C > 0 : \quad \|\phi(g)\|_{H^2(\Omega)} \leq C \|g\|_{L^2(\Omega)} \quad \forall g \in L^2(\Omega).$$

In particular, this is true for a generic elliptic problem with Dirichlet or Neumann (but not mixed) data on a polygonal and convex domain Ω [Gri76]. We now choose $g = e_h$ and denote, for simplicity, $\phi = \phi(e_h)$. Furthermore, having chosen $v = e_h$, we have

$$\|e_h\|_{L^2(\Omega)}^2 = a(e_h, \phi).$$

Since, by the elliptic regularity of the adjoint problem, $\phi \in H^2(\Omega)$ and $\|\phi\|_{H^2(\Omega)} \leq C \|e_h\|_{L^2(\Omega)}$ thanks to the Galerkin orthogonality, we have that

$$\begin{aligned} \|e_h\|_{L^2(\Omega)}^2 &= a(e_h, \phi) = a(e_h, \phi - \Pi_h^1 \phi) \\ &\leq C_1 \|e_h\|_{H^1(\Omega)} \|\phi - \Pi_h^1 \phi\|_{H^1(\Omega)} \\ &\leq C_2 \|e_h\|_{H^1(\Omega)} h \|\phi\|_{H^2(\Omega)} \\ &\leq C_3 \|e_h\|_{H^1(\Omega)} h \|e_h\|_{L^2(\Omega)}, \end{aligned}$$

where we have exploited the continuity of the form $a(\cdot, \cdot)$ and the estimate (4.72). Thus

$$\|e_h\|_{L^2(\Omega)} \leq C_3 h \|e_h\|_{H^1(\Omega)},$$

from which (4.75) follows, using the estimate (4.73) of the error in $H^1(\Omega)$.

Remark 4.7 The technique illustrated above, depending upon the use of the adjoint problem for the estimate of the L^2 -norm of the discretization error, is known in the literature as *Aubin-Nitsche trick* [Aub67, Nit68]. Several examples of the determination of the adjoint of a given problem will be presented in Sec. 3.6. •

Example 4.2 We consider the model problem $-\Delta u + u = f$ in $\Omega = (0, 1)^2$ with $u = g$ on $\partial\Omega$. Suppose to choose the known term f and the function g so that the exact solution of the problem is $u(x, y) = \sin(2\pi x) \cos(2\pi y)$. We solve such a problem with the Galerkin method with finite elements of degree 1 and 2 on a uniform grid with stepsize h . The graph of Fig. 4.16 shows the behavior of the error when the grid-size h decreases, both in the norm $L^2(\Omega)$ and in that of $H^1(\Omega)$. As shown by inspecting the slope of the lines in the figure, the error decrease when using L^2 norm (crossed lines) is quadratic if linear finite elements are used (solid line), and cubic when quadratic finite elements are used (etched line).

With respect to the H^1 norm (lines without crosses) instead, there is a linear reduction of the error with respect to the linear finite elements (solid line), and quadratic when quadratic finite elements are used (etched line). Fig. 4.17 shows the solution on the grid with grid-size 1/8 obtained with linear (left) and quadratic (right) finite elements. ■

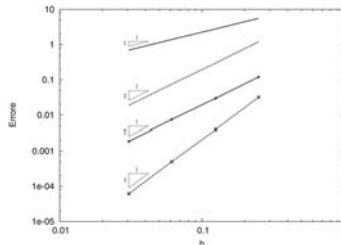


Fig. 4.16. Behavior with respect to h of the error in $H^1(\Omega)$ norm (lines without crosses) and in $L^2(\Omega)$ norm (lines with crosses) for linear (solid lines) and quadratic (etched lines) finite elements for the solution of the problem reported in Example 4.2

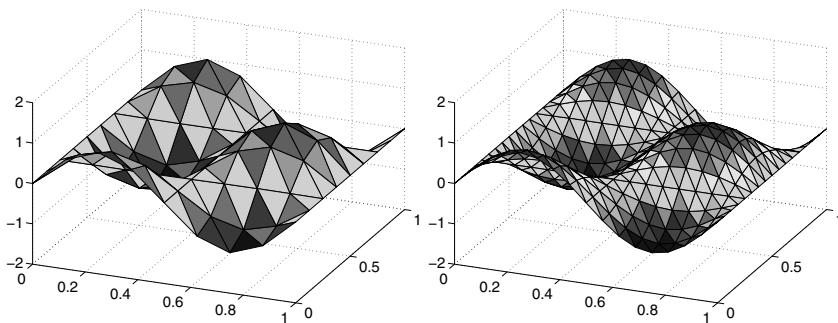


Fig. 4.17. Solutions computed using piecewise linear (left) and piecewise quadratic (right) finite elements on a uniform grid with grid-size 1/8

4.6 Grid adaptivity

In Theorems 4.6 and 4.7 we have derived some a priori estimates for the finite element approximation error.

Since the h parameter is the maximal length of the finite element edges, if we referred to (4.72) we would be tempted to refine the grid everywhere in the hope of reducing the error $\|u - u_h\|_{H^1(\Omega)}$. However, it is more convenient to refer to (4.71) where the upper bound is the sum of elemental contributions involving the solution seminorm $|u|_{H^{r+1}(K)}$ on each element K and the local grid-size h_K .

Indeed, in order to have an efficient grid that minimizes the number of necessary elements to obtain the desired accuracy, we can *equidistribute* the error on each element $K \in \mathcal{T}_h$. In particular, we would like to obtain

$$h_K^r |u|_{H^{r+1}(K)} \simeq \eta \quad \forall K \in \mathcal{T}_h,$$

where η is a well chosen constant that only depends on the desired accuracy and on the number of elements of the grid.

A larger contribution from $|u|_{H^{r+1}(K)}$ (due to a more pronounced variability of $u|_K$) will need to be balanced either by a smaller local grid-size h_K or by a higher polynomial degree r . In the first case, we will talk about *h-adaptivity* of the grid, in the

second case of *p-adaptivity* (where p stands for “polynomial”). In the remainder of this chapter we will only focus on the first technique. However, we refer to Chap. 10 for the analysis of error estimates which are better suited for polynomial adaptivity.

The remarks made up to now, although correct, result in fact to be of little use as the solution u is not known. We can therefore proceed according to different strategies. The first way is to use the a priori error estimate (4.71) by replacing the exact solution u with a well chosen approximation, easily computable on each single element. In such case, we talk about *a priori adaptivity*.

A second approach is instead based on the use of an *a posteriori error estimate* able to link the approximation error to the behavior of the approximate numerical solution u_h , known after numerically solving the problem. In such case, the optimal computational grid will be constructed through an iterative process where *solution*, *error estimate* and *modification of the computational grid* are repeated until reaching the requested accuracy. In this case, we talk about *a posteriori adaptivity*.

The a priori and a posteriori adaptivity strategies are not mutually exclusive, actually they can coexist. For instance, having generated an appropriate starting grid through an a priori adaptivity, the latter can be further refined through a posteriori analysis.

4.6.1 A priori adaptivity based on derivatives reconstruction

An a priori adaptivity technique is based on the estimate (4.71) where the derivatives of u are carefully approximated on each element, in the purpose of estimating the local seminorms of u . To do this, an approximate solution u_{h^*} is used, computed on a tentative grid with stepsize h^* , with h^* large enough so that the computation is cheap, but not too large to generate an excessive error in the approximation of the derivatives, which could affect the effectiveness of the whole procedure.

We exemplify the algorithm for linear finite elements, in which case (4.71) takes the form

$$\|u - u_h\|_{H^1(\Omega)} \leq C \left(\sum_{K \in \mathcal{T}_h} h_K^2 |u|_{H^2(K)}^2 \right)^{\frac{1}{2}} \quad (4.84)$$

(C accounts for the continuity and coercivity constants of the bilinear form). Our aim is to eventually solve our problem on a grid \mathcal{T}_h guaranteeing that the right-hand size of (4.84) stands below a predefined tolerance $\epsilon > 0$. Let us suppose that we have computed a solution, say u_{h^*} , on a preliminary grid \mathcal{T}_{h^*} with N^* triangles. We use u_{h^*} to approximate the second derivatives of u that intervene in the definition of the seminorm $|u|_{H^2(K)}^2$. Since u_{h^*} does not have any continuous second derivatives in Ω , it is necessary to proceed with an adequate *reconstruction technique*, that for each node \mathbf{N}_i of the grid we consider the set (*patch*) $K_{\mathbf{N}_i}$ of the elements sharing \mathbf{N}_i as a node (that is the set of the elements forming the support of φ_i , see Fig. 4.12). We then find the planes $\pi_i^j(\mathbf{x}) = \mathbf{a}_i^j \cdot \mathbf{x} + b_i^j$ by minimizing

$$\int_{K_{\mathbf{N}_i}} \left| \pi_i^j(\mathbf{x}) - \frac{\partial u_{h^*}}{\partial x_j}(\mathbf{x}) \right|^2 d\mathbf{x}, \quad j = 1, 2, \quad (4.85)$$

solving a two-equation system for the coefficients \mathbf{a}_i^j and b_i^j . This can be regarded as the local projection phase. We thus build a piecewise linear approximation $\mathbf{g}_{h^*} \in (X_{h^*}^1)^2$ of the gradient ∇u_{h^*} defined as

$$[\mathbf{g}_{h^*}(\mathbf{x})]^j = \sum_i \pi_i^j(\mathbf{x}_i) \varphi_i(\mathbf{x}), \quad j = 1, 2, \quad (4.86)$$

where the sum spans over all the nodes \mathbf{N}_i of the grid. Once the gradient is reconstructed we can proceed in two different ways, based on the type of reconstruction that we want to obtain for the second derivatives. We recall first of all that the Hessian matrix associated to a function u is defined by $\mathbf{D}^2(u) = \nabla(\nabla u)$, that is

$$[\mathbf{D}^2(u)]_{i,j} = \frac{\partial^2 u}{\partial x_i \partial x_j}, \quad i, j = 1, 2.$$

A *piecewise constant* approximation of the latter is obtained by setting, for each $K^* \in \mathcal{T}_{h^*}$,

$$\mathbf{D}_h^2|_{K^*} = \frac{1}{2} (\nabla \mathbf{g}_{h^*} + (\nabla \mathbf{g}_{h^*})^T)|_{K^*}. \quad (4.87)$$

Notice the use of the symmetric form of the gradient, which is necessary for Hessian symmetry.

Should one be interested in a piecewise linear reconstruction of the Hessian, the same projection technique defined by (4.85) and (4.86) can be directly applied to the reconstructed \mathbf{g}_{h^*} , by then symmetrizing the matrix obtained in this way via (4.87). In any case, we are now able to compute an approximation of $|u|_{H^2(K^*)}$ on a generic triangle K^* of \mathcal{T}_{h^*} , an approximation that will obviously be linked to the reconstructed \mathbf{D}_h^2 .

From (4.84) we deduce that, to obtain the approximate solution u_h with an error smaller than or equal to a predefined tolerance ϵ , we must construct a new grid \mathcal{T}_h^{new} such that

$$\sum_{K \in \mathcal{T}_h^{new}} h_K^2 |u|_{H^2(K)}^2 \simeq \sum_{K \in \mathcal{T}_h^{new}} h_K^2 \sum_{i,j=1}^2 \|[\mathbf{D}_h^2]_{ij}\|_{L^2(K)}^2 \leq \left(\frac{\epsilon}{C}\right)^2.$$

Ideally, one would wish the error to be equidistributed on each element K of the new grid.

A possible adaptation procedure then consists in generating the new grid by appropriately partitioning all of the N^* triangles K^* of \mathcal{T}_{h^*} for which we have

$$\eta_{K^*}^2 = h_{K^*}^2 \sum_{i,j=1}^2 \|[\mathbf{D}_h^2]_{ij}\|_{L^2(K^*)}^2 > \frac{1}{N^*} \left(\frac{\epsilon}{C}\right)^2. \quad (4.88)$$

This method is said to be a *refinement* as it only aims at creating a *finer* grid with respect to the initial one, but it clearly does not allow to fully satisfy the equidistribution condition.

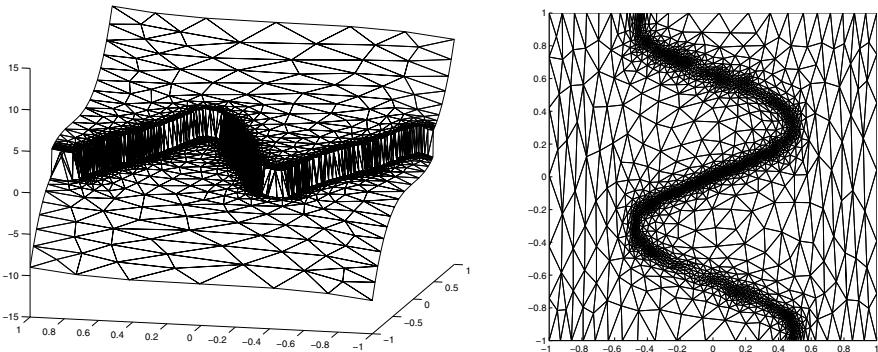


Fig. 4.18. The function u (left) and the third adapted grid (right) for Example 4.3

More sophisticated algorithms also allow to *derefine* the grid in presence of the triangles for which the inequality (4.88) is verified with the sign \ll (i.e. much smaller than) instead of $>$. However, derefinement procedures are of more difficult implementation than refinement ones. Hence, one often prefers to construct the new grid from scratch (a procedure called *remeshing*). For this purpose, on the basis of the error estimate, the following element-wise constant *spacing function* H is introduced

$$H|_{K^*} = \frac{\epsilon}{C\sqrt{N^*} \left(\sum_{i,j=1}^2 \|[\mathbf{D}_h^2]_{ij}\|_{L^2(K)}^2 \right)^{1/2} |u_{h^*}|_{H^2(K^*)}} \quad \forall K^* \in \mathcal{T}_{h^*} \quad (4.89)$$

and is used to construct the adapted grid by applying one of the grid generation algorithms illustrated in Chap. 6. The adaptation algorithm often requires the function H to be continuous and linear on each triangle. In this case we can again resort to a local projection, like that in (4.85).

The adaptation can then be repeated for the solution computed on the new grid, until inequality (4.88) is inverted on all of the elements.

Remark 4.8 The C constant appearing in inequality (4.84) can be estimated by applying the same inequality to known functions (which makes therefore possible to compute the exact error). An alternative that does not require explicitly knowing C consists in realizing the grid that equally distributes the error for a number N^* of fixed a priori elements. In this case the value of H computed by setting ϵ and C to one in (4.89) is rescaled, by multiplying it by a constant, so that the new grid has a number N^* of elements fixed a priori. •

Example 4.3 We consider the function $u(x, y) = 10x^3 + y^3 + \tan^{-1}(10^{-4}/(\sin(5y) - 2x))$ on the domain $\Omega = (-1, 1)^2$, which features a strong gradient across the curve $x = 0.5 \sin(5y)$, as it can be observed from Fig. 4.18 on the left. Starting from an initial structured grid constituted

by 50 triangles and using an adaptive procedure guided by the Hessian of u , we obtain, after 3 iterations, the grid in Fig. 4.18 (right), made of 3843 elements. As it can be observed, most of the triangles are located in the proximity of the function jump: indeed, while few medium-large surface triangles are necessary to describe u in a satisfactory way in the regions located uphill and downhill from the jump, the abrupt variation of u in presence of discontinuities requires the use of small triangles, i.e. a reduced discretization grid-size. Furthermore, we note the anisotropic nature of the grid in Fig. 4.18, visible by the presence of elements whose shape is very stretched with respect to that of an equilateral triangle (typical of an isotropic grid). Such grid has been obtained by generalizing the estimator (4.88) to the anisotropic case. The idea is essentially to *separately* exploit the information provided by the components $[\mathbf{D}_h^2]_{ij}$ instead of “mixing” them through the $L^2(K^*)$ norm. By using the same adaptive procedure in the isotropic case (i.e. the estimator in (4.88)), we would have obtained, after 3 iterations, an adapted grid made of 10535 elements. ■

4.6.2 A posteriori adaptivity

The procedures described in the previous section can be unsatisfactory as the reconstruction of u 's derivatives starting from u_{h^*} is often subject to errors that are not easy to quantify.

A radical alternative consists in adopting *a posteriori estimates* of the error. The latter do not make use of the a priori estimate (4.71) (and consequently of any approximate derivatives of the unknown solution u). Rather, they are obtained as a function of *computable* quantities, normally based on the so-called *residue* of the approximate solution, which provides a measurement of how well the discrete solution satisfies the differential problem on each element of the given grid. Let us consider, as an example, the Poisson problem (3.13). Its weak formulation is given by (3.18), while its approximation using finite elements is described by (4.40), where V_h is the space $\overset{\circ}{X}_h^r$ defined in (4.39). For each $v \in H_0^1(\Omega)$ and for each $v_h \in V_h$, we have, thanks to the Galerkin orthogonality property (4.8) and exploiting (3.18),

$$\begin{aligned} \int_{\Omega} \nabla(u - u_h) \cdot \nabla v \, d\Omega &= \int_{\Omega} \nabla(u - u_h) \cdot \nabla(v - v_h) \, d\Omega \\ &= \int_{\Omega} f(v - v_h) \, d\Omega - \int_{\Omega} \nabla u_h \cdot \nabla(v - v_h) \, d\Omega \\ &= \int_{\Omega} f(v - v_h) \, d\Omega + \sum_{K \in \mathcal{T}_h} \int_K \Delta u_h(v - v_h) \, d\Omega - \sum_{K \in \mathcal{T}_h} \int_{\partial K} \frac{\partial u_h}{\partial n}(v - v_h) \, d\gamma \quad (4.90) \\ &= \sum_{K \in \mathcal{T}_h} \int_K (f + \Delta u_h)(v - v_h) \, d\Omega - \sum_{K \in \mathcal{T}_h} \int_{\partial K} \frac{\partial u_h}{\partial n}(v - v_h) \, d\gamma. \end{aligned}$$

We observe that all the local integrals make sense.

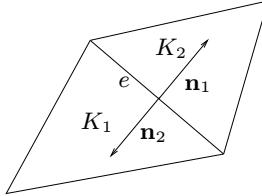


Fig. 4.19. Triangles involved in the definition of the jump of the normal derivative of u_h through an internal side e

Having denoted by e a side of the generic triangle K , we define a *jump* of the normal derivative of u_h through an internal side e the quantity

$$\left[\frac{\partial u_h}{\partial n} \right]_e = \nabla u_h|_{K_1} \cdot \mathbf{n}_1 + \nabla u_h|_{K_2} \cdot \mathbf{n}_2 = (\nabla u_h|_{K_1} - \nabla u_h|_{K_2}) \cdot \mathbf{n}_1, \quad (4.91)$$

where K_1 and K_2 are the two triangles sharing the side e , whose normal outgoing unit vectors are given by \mathbf{n}_1 and \mathbf{n}_2 respectively, with $\mathbf{n}_1 = -\mathbf{n}_2$ (see Fig. 4.19). In order to extend such definition also to the boundary sides, we introduce the so-called *generalized jump*, given by

$$\left[\frac{\partial u_h}{\partial n} \right] = \begin{cases} \left[\frac{\partial u_h}{\partial n} \right]_e & \text{for } e \in \mathcal{E}_h, \\ 0 & \text{for } e \in \partial\Omega, \end{cases} \quad (4.92)$$

where \mathcal{E}_h indicates the set of inner sides in the grid. We note that, in the case of linear finite elements, (4.92) identifies a piecewise constant function defined on all the sides of the grid \mathcal{T}_h . Moreover, the definition (4.92) can be suitably modified in the case where problem (3.13) is completed with boundary conditions that are not necessarily of Dirichlet type.

Thanks to (4.92) we can therefore write that

$$\begin{aligned} - \sum_{K \in \mathcal{T}_h \setminus \partial\Omega} \int \frac{\partial u_h}{\partial n} (v - v_h) d\gamma &= - \sum_{K \in \mathcal{T}_h} \sum_{e \in \partial K} \int_e \frac{\partial u_h}{\partial n} (v - v_h) d\gamma \\ &= - \sum_{K \in \mathcal{T}_h} \sum_{e \in \partial K} \frac{1}{2} \int_e \left[\frac{\partial u_h}{\partial n} \right] (v - v_h) d\gamma = - \frac{1}{2} \sum_{K \in \mathcal{T}_h \setminus \partial\Omega} \int_K \left[\frac{\partial u_h}{\partial n} \right] (v - v_h) d\gamma, \end{aligned} \quad (4.93)$$

where the factor $1/2$ takes into account the fact that each internal side e of the grid is shared by two elements. Moreover, since $v - v_h = 0$ on the boundary, in (4.92) we could assign any value different from zero in presence of $e \in \partial\Omega$ as the terms of (4.93) associated to the boundary sides would be null in any case.

By now inserting (4.93) in (4.90) and applying the Cauchy-Schwarz inequality, we obtain

$$\begin{aligned} \left| \int_{\Omega} \nabla(u - u_h) \cdot \nabla v \, d\Omega \right| &\leq \sum_{K \in \mathcal{T}_h} \left\{ \|f + \Delta u_h\|_{L^2(K)} \|v - v_h\|_{L^2(K)} \right. \\ &\quad \left. + \frac{1}{2} \left\| \left[\frac{\partial u_h}{\partial n} \right] \right\|_{L^2(\partial K)} \|v - v_h\|_{L^2(\partial K)} \right\}. \end{aligned} \quad (4.94)$$

Now we look for $v_h \in V_h$ that allows to express the norms of $v - v_h$ as a function of a well-chosen norm of v . Moreover, we want this norm to be “local”, i.e. computed over a region \tilde{K} containing K but as little as possible. If v were continuous, we could take as v_h the Lagrangian interpolant of v and use the previously cited interpolation error estimates on K . Unfortunately, in our case $v \in H^1(\Omega)$ and therefore it is not necessarily continuous. However, if \mathcal{T}_h is a regular grid, we can introduce the so-called Clément interpolation operator $\mathcal{R}_h : H^1(\Omega) \rightarrow V_h$ defined, in the case of linear finite elements, as

$$\mathcal{R}_h v(\mathbf{x}) = \sum_{\mathbf{N}_j} (P_j v)(\mathbf{N}_j) \varphi_j(\mathbf{x}) \quad \forall v \in H^1(\Omega), \quad (4.95)$$

where $P_j v$ denotes the plane defined on the patch $K_{\mathbf{N}_j}$ of the grid elements that share the node \mathbf{N}_j (see Fig. 4.20), determined by the relations

$$\int_{K_{\mathbf{N}_j}} (P_j v - v) \psi \, d\mathbf{x} = 0 \quad \text{for } \psi = 1, x, y,$$

and where the φ_j are the characteristic Lagrangian basis functions of the finite element space under exam.

For each $v \in H^1(\Omega)$ and each $K \in \mathcal{T}_h$, the following inequalities hold (see, e.g., [BG98, BS94, Clé75]):

$$\|v - \mathcal{R}_h v\|_{L^2(K)} \leq C_1 h_K |v|_{H^1(\tilde{K})},$$

$$\|v - \mathcal{R}_h v\|_{L^2(\partial K)} \leq C_2 h_K^{\frac{1}{2}} \|v\|_{H^1(\tilde{K})},$$

where C_1 and C_2 are two positive constants that depend on the minimal angle of the elements of the triangulation, while $\tilde{K} = \{K_j \in \mathcal{T}_h : K_j \cap K \neq \emptyset\}$ represents the union of K with all the triangles that share a side or a vertex with it (see Fig. 4.20). By choosing in (4.94) $v_h = \mathcal{R}_h v$, setting $C = \max(C_1, C_2)$ and using the discrete Cauchy-Schwarz inequality, we obtain

$$\begin{aligned} \left| \int_{\Omega} \nabla(u - u_h) \cdot \nabla v \, d\Omega \right| &\leq C \sum_{K \in \mathcal{T}_h} \rho_K(u_h) \|v\|_{H^1(\tilde{K})} \\ &\leq C \left(\sum_{K \in \mathcal{T}_h} [\rho_K(u_h)]^2 \right)^{\frac{1}{2}} \left(\sum_{K \in \mathcal{T}_h} \|v\|_{H^1(\tilde{K})}^2 \right)^{\frac{1}{2}}. \end{aligned}$$

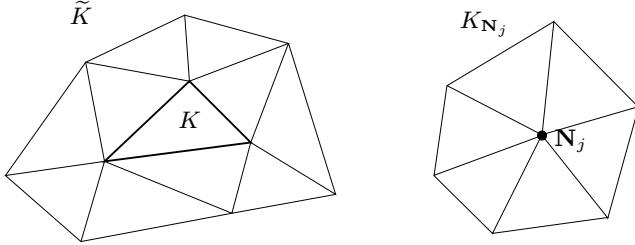


Fig. 4.20. The set \tilde{K} of elements that have in common with K at least a node of the grid (left) and the set K_{N_j} of the elements that share node N_j (right)

We have denoted by

$$\rho_K(u_h) = h_K \|f + \Delta u_h\|_{L^2(K)} + \frac{1}{2} h_K^{\frac{1}{2}} \left\| \left[\frac{\partial u_h}{\partial n} \right] \right\|_{L^2(\partial K)} \quad (4.96)$$

the so-called *local residue*, constituted by the internal residue $\|f + \Delta u_h\|_{L^2(K)}$ and by the boundary residue $\left\| \left[\frac{\partial u_h}{\partial n} \right] \right\|_{L^2(\partial K)}$.

We now observe that, since \mathcal{T}_h is regular, the number of elements in \tilde{K} is necessarily limited by a positive integer independent of h , which we denote by n . Thus,

$$\left(\sum_{K \in \mathcal{T}_h} \|v\|_{H^1(\tilde{K})}^2 \right)^{\frac{1}{2}} \leq \sqrt{n} \|v\|_{H^1(\Omega)}.$$

Finally, having chosen $v = u - u_h$ and applying the Poincaré inequality (2.13), we find

$$\|u - u_h\|_{H^1(\Omega)} \leq C \sqrt{n} \left(\sum_{K \in \mathcal{T}_h} [\rho_K(u_h)]^2 \right)^{\frac{1}{2}}, \quad (4.97)$$

where the constant C now also includes the contribution from the Poincaré constant. This a posteriori error estimate is called *residual-based*.

Notice that $\rho_K(u_h)$ is an effectively computable quantity, being a function of the datum f , of the geometric parameter h_K and of the computed solution u_h . The most delicate point of this analysis is the not always immediate estimate of the constants C and n .

The a posteriori estimate (4.97) can, for instance, be used in order to guarantee that

$$\frac{1}{2} \epsilon \leq \frac{\|u - u_h\|_{H^1(\Omega)}}{\|u_h\|_{H^1(\Omega)}} \leq \frac{3}{2} \epsilon, \quad (4.98)$$

$\epsilon > 0$ being a pre-established tolerance. To this end, via an iterative procedure illustrated in Fig. 4.21, we can locally refine and derefine the grid \mathcal{T}_h until when, for each K , the following *local* inequalities are satisfied

$$\frac{1}{4} \frac{\epsilon^2}{N} \|u_h\|_{H^1(\Omega)}^2 \leq [\rho_K(u_h)]^2 \leq \frac{9}{4} \frac{\epsilon^2}{N} \|u_h\|_{H^1(\Omega)}^2, \quad (4.99)$$

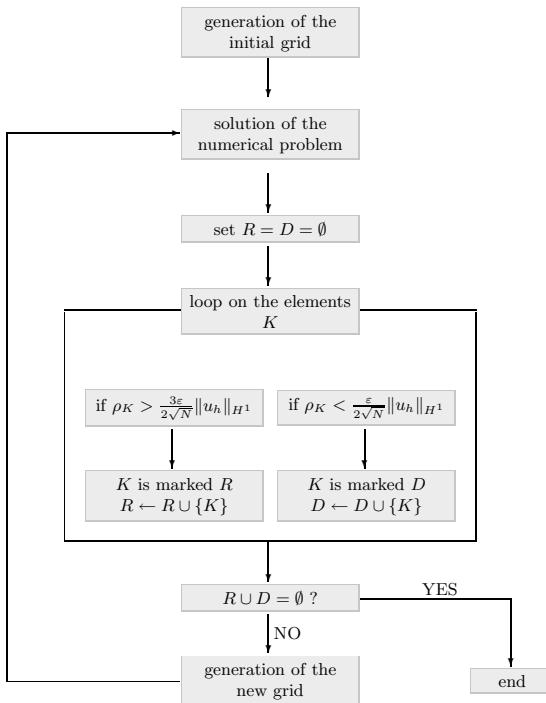


Fig. 4.21. Example of iterative grid adaptation procedure

having denoted by N the number of elements of the grid \mathcal{T}_h . This ensures that the *global* inequalities (4.98) are satisfied, up to the contribution of the constant $C\sqrt{n}$. Alternatively, we can construct a well-chosen grid spacing function H , analogously to what was done in Sec. 4.6.1.

Naturally, the flow diagram reported in Fig. 4.21 can also be used for boundary-value problems differing from (4.40).

4.6.3 Numerical examples of adaptivity

We illustrate the concept of grid adaptivity on two simple differential problems. For this purpose, we adopt the iterative procedure reported in Fig. 4.21, although we will limit ourselves to the sole refinement phase. The derefinement process results indeed to be of more difficult implementation: as a matter of fact, most commonly used software only allow to refine the initial grid, hence it will be necessary to choose the latter to be suitably coarse.

Finally, for both reported examples, the reference estimator for the discretization error is represented by the right term of (4.97).

First example. Let us consider the problem $-\Delta u = f$ in $\Omega = (-1, 1)^2$, with homogeneous Dirichlet conditions on the whole boundary $\partial\Omega$. Moreover, we choose a forcing term f such that the exact solution is $u = \sin(\pi x) \sin(\pi y) \exp(10x)$. We begin the adaptive procedure by starting from a uniform initial grid, made of 324 elements, and with a tolerance $\epsilon = 0.2$. The iterative procedure converges after 7 iterations. We report in Fig. 4.22 the initial grid together with three of the adapted grids obtained in this way, while Table 4.2 summarizes the number \mathcal{N}_h of elements of the grid \mathcal{T}_h , the relative error $\|u - u_h\|_{H^1(\Omega)} / \|u_h\|_{H^1(\Omega)}$ and the normalized estimator $\eta / \|u_h\|_{H^1(\Omega)}$ on the initial grid and on the six first adapted grids.

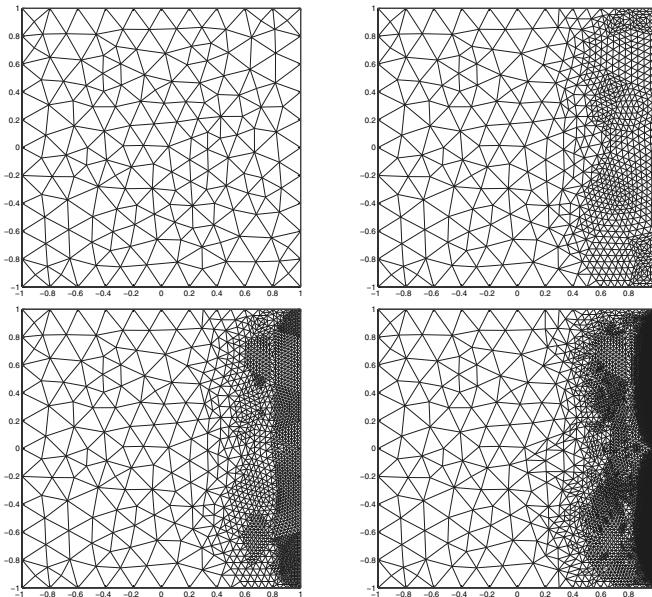


Fig. 4.22. Initial grid (top left) and three grids adapted by choosing the adaptive procedure of Fig. 4.21, at the second (top right), third (bottom left) and fifth (bottom right) iteration

The grids in Fig. 4.22 provide a qualitative feedback for the reliability of the chosen adaptivity procedure: as expected, triangles tend to concentrate in those regions where u attains its extrema. On the other hand, the values in Table 4.2 also allow to perform a quantitative analysis: both the relative error and the normalized estimator progressively decrease, when the iterations increase. However, we can notice an average overestimate of about 10-11 times with respect to the fixed tolerance ϵ . This is not unusual and can basically be explained by the fact that the constant $C\sqrt{n}$ in the inequalities (4.98) and (4.99) has been neglected (i.e. set to 1). It is clear that such choice actually leads to requiring a tolerance $\tilde{\epsilon} = \epsilon/(C\sqrt{n})$, that will therefore coincide with the original ϵ only in the case where we have $C\sqrt{n} \sim 1$. More precise procedures, taking the constant $C\sqrt{n}$ into account, are in any case possible by starting, e.g., from the (theoretical and numerical) analysis provided in [BDR92, EJ88].

Table 4.2. Cardinality, relative error and normalized estimator associated with the initial grid and with the six first adaptive grids

iteration	\mathcal{N}_h	$\ u - u_h\ _{H^1(\Omega)} / \ u_h\ _{H^1(\Omega)}$	$\eta / \ u_h\ _{H^1(\Omega)}$
0	324	0.7395	5.8333
1	645	0.3229	3.2467
2	1540	0.1538	1.8093
3	3228	0.0771	0.9782
4	7711	0.0400	0.5188
5	17753	0.0232	0.2888
6	35850	0.0163	0.1955

Second example. Let us consider the problem $-\Delta u = 0$ in $\Omega = \{\mathbf{x} = r(\cos \theta, \sin \theta)^T, r \in (0, 1), \theta \in (0, \frac{3}{4}\pi)\}$, with u appropriately assigned on the boundary of Ω so that $u(r, \theta) = r^{4/3} \sin(\frac{4}{3}\theta)$ is the exact solution. Such function features low regularity in a neighborhood of the origin. Suppose we approximate such problem via the Galerkin method using linear finite elements on the quasi-uniform grid reported in the left of Fig. 4.23, made of 138 triangles. As noticeable by the distortion in the isolines of u_h in the left of Fig. 4.24, the solution obtained in this way is quite inaccurate near the origin. We now use the estimator (4.97) to generate an adapted grid which better suits the approximation of u . By following an adaptive procedure such as the one illustrated

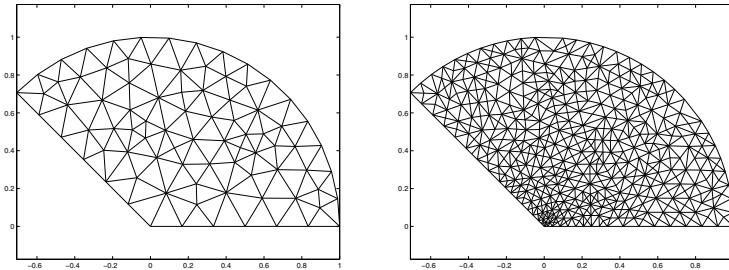


Fig. 4.23. Initial grid (left) and twentieth adapted grid (right)

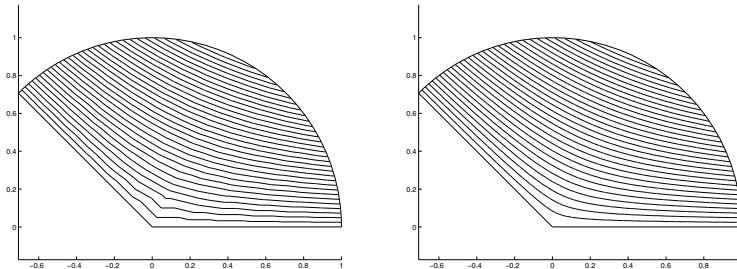


Fig. 4.24. Isolines of the linear finite element solution on the initial grid (left) and on the twentieth adapted grid (right)

in Fig. 4.21 we obtain after 20 steps the grid made of 859 triangles reported in Fig. 4.23 on the right. As it can be observed in Fig. 4.24 on the right, the isolines associated to the corresponding discrete solution denote a higher regularity, an evidence of the quality improvement of the solution. As a comparison, in order to obtain a solution characterized by the same accuracy ϵ with respect to the norm H^1 of the error (required to be equal to 0.01) on a uniform grid, 2208 triangles are necessary.

4.6.4 A posteriori error estimates in the L^2 norm

Besides (4.97) it is possible to derive an a posteriori estimate of the error in L^2 norm. To this end, we will again resort to the duality technique of Aubin-Nitsche used in Sec. 4.5.4, and in particular we will consider the adjoint problem (4.76) associated to the Poisson problem (3.13). Moreover, we will suppose that the domain Ω is sufficiently regular (for instance, a convex polygon) in order to guarantee that the elliptic regularity result (4.74) stated in Lemma 4.6 is true.

Moreover, we will exploit the following local estimates for the interpolation error associated with the operator Π_h^r applied to functions $v \in H^2(\Omega)$

$$\|v - \Pi_h^r v\|_{L^2(\partial K)} \leq \tilde{C}_1 h_K^{\frac{3}{2}} |v|_{H^2(K)} \quad (4.100)$$

(see [BS94] or [Cia78]), and

$$\|v - \Pi_h^r v\|_{L^2(K)} \leq \tilde{C}_2 h_K^2 |v|_{H^2(K)}. \quad (4.101)$$

The latter inequality is obtained from (4.67).

Starting from the adjoint problem (4.76) and exploiting the Galerkin orthogonality (4.8), we have, for each $\phi_h \in V_h$,

$$\|e_h\|_{L^2(\Omega)}^2 = \sum_{K \in \mathcal{T}_h} \int_K f (\phi - \phi_h) \, d\Omega - \sum_{K \in \mathcal{T}_h} \int_K \nabla u_h \cdot \nabla (\phi - \phi_h) \, d\Omega.$$

Counterintegrating by parts, we obtain

$$\|e_h\|_{L^2(\Omega)}^2 = \sum_{K \in \mathcal{T}_h} \int_K (f + \Delta u_h) (\phi - \phi_h) \, d\Omega - \sum_{K \in \mathcal{T}_h} \int_{\partial K} \frac{\partial u_h}{\partial n} (\phi - \phi_h) \, d\gamma.$$

Using the definition (4.92) of generalized jump of the normal derivative of u_h across the triangle sides and setting $\phi_h = \Pi_h^r \phi$, we have

$$\begin{aligned} \|e_h\|_{L^2(\Omega)}^2 &= \sum_{K \in \mathcal{T}_h} \left[\int_K (f + \Delta u_h) (\phi - \Pi_h^r \phi) \, d\Omega \right. \\ &\quad \left. - \frac{1}{2} \int_{\partial K} \left[\frac{\partial u_h}{\partial n} \right] (\phi - \Pi_h^r \phi) \, d\gamma \right]. \end{aligned} \quad (4.102)$$

We estimate the two terms in the right-hand side separately. By using the Cauchy-Schwarz inequality and (4.101), it follows that

$$\begin{aligned} \left| \int_K (f + \Delta u_h) (\phi - \Pi_h^r \phi) d\Omega \right| &\leq \|f + \Delta u_h\|_{L^2(K)} \|\phi - \Pi_h^r \phi\|_{L^2(K)} \\ &\leq \tilde{C}_2 h_K^2 \|f + \Delta u_h\|_{L^2(K)} |\phi|_{H^2(K)}. \end{aligned} \quad (4.103)$$

Moreover, thanks to (4.100) we obtain

$$\begin{aligned} \left| \int_{\partial K} \left[\frac{\partial u_h}{\partial n} \right] (\phi - \Pi_h^r \phi) d\gamma \right| &\leq \left\| \left[\frac{\partial u_h}{\partial n} \right] \right\|_{L^2(\partial K)} \|\phi - \Pi_h^r \phi\|_{L^2(\partial K)} \\ &\leq \tilde{C}_1 h_K^{\frac{3}{2}} \left\| \left[\frac{\partial u_h}{\partial n} \right] \right\|_{L^2(\partial K)} |\phi|_{H^2(K)}. \end{aligned} \quad (4.104)$$

By now inserting (4.103) and (4.104) in (4.102) and applying the discrete Cauchy-Schwarz inequality we have

$$\begin{aligned} \|e_h\|_{L^2(\Omega)}^2 &\leq C \sum_{K \in \mathcal{T}_h} h_K \rho_K(u_h) |\phi|_{H^2(K)} \leq C \sqrt{\sum_{K \in \mathcal{T}_h} [h_K \rho_K(u_h)]^2} |\phi|_{H^2(\Omega)} \\ &\leq C \sqrt{\sum_{K \in \mathcal{T}_h} [h_K \rho_K(u_h)]^2} \|e_h\|_{L^2(\Omega)}, \end{aligned}$$

with $C = \max(\tilde{C}_1, \tilde{C}_2)$, having introduced the notation (4.96) and having exploited the elliptic regularity property (4.80) in the last inequality. We can then conclude that

$$\|u - u_h\|_{L^2(\Omega)} \leq C \left(\sum_{K \in \mathcal{T}_h} h_K^2 [\rho_K(u_h)]^2 \right)^{\frac{1}{2}}, \quad (4.105)$$

$C > 0$ being a constant independent of h .

Remark 4.9 Among the most widespread a posteriori estimates in engineering, we cite for its simplicity and computational effectiveness the estimator proposed by Zienkiewicz and Zhu in the context of a finite element approximation of linear elasticity problems [CZ87]. The basic idea of this estimator is very simple. Suppose we want to control the energy norm $(\int_{\Omega} |\nabla u - \nabla u_h|^2 d\Omega)^{1/2}$ of the discretization error associated to a finite element approximation of the model problem (3.13). This estimator replaces the exact gradient ∇u in the latter norm by a corresponding reconstruction obtained through a suitable post-processing of the discrete solution u_h . During the years, several “recipes” have been proposed in the literature for the construction of the gradient ∇u (see, e.g., [ZZ92, Rod94, PWY90, LW94, NZ04, BMMP06]). The same procedure illustrated in Sec. 4.6.1 that leads to the reconstructed g_{h^*} defined in (4.86) can be used here for this purpose. Thus, having chosen a reconstruction, say

$G_R(u_h)$, of ∇u , the Zienkiewicz and Zhu-type estimator is represented by the quantity $\eta = (\int_{\Omega} |G_R(u_h) - \nabla u_h|^2 d\Omega)^{1/2}$. Clearly, to each new definition of $G_R(u_h)$ corresponds a new error estimator. For this reason, a posteriori error estimators with such structure are commonly called *recovery-based*. •

4.6.5 A posteriori estimates of a functional of the error

In the previous section, the adjoint problem (4.76) has been used in a purely formal way, as the error e_h , that represents its forcing term, is unknown.

There exists another family of a posteriori estimators of the error, again based on the adjoint problem, which, instead, explicitly use the information provided by the latter (see, e.g., [Ran99]). In such case, an estimate is provided for a suitable functional J of the error e_h , instead of for a suitable norm of e_h . This prerogative turns out to be particularly useful when one wants to provide significant estimates of the error for quantities of physical relevance, such as, for instance, resistance or drag in the case of bodies immersed in fluids, average values of concentration, strains, deformations, fluxes, etc. For this purpose, it will be sufficient to operate a suitable choice for the functional J . This type of adaptivity is called *goal-oriented*. To illustrate this new paradigm, let us still refer to the Poisson problem (3.13) and assume that we want to control the error of a given functional $J : H_0^1(\Omega) \rightarrow \mathbb{R}$ of the solution u . Let us consider the following weak formulation of the corresponding adjoint problem

$$\text{find } \phi \in V : \quad \int_{\Omega} \nabla \phi \cdot \nabla w d\Omega = J(w) \quad \forall w \in V, \quad (4.106)$$

with $V = H_0^1(\Omega)$. By using the Galerkin orthogonality and proceeding as done in the previous section, we find

$$\begin{aligned} J(e_h) &= \int_{\Omega} \nabla e_h \cdot \nabla \phi d\Omega = \sum_{K \in \mathcal{T}_h} \left[\int_K (f + \Delta u_h) (\phi - \phi_h) d\Omega \right. \\ &\quad \left. - \frac{1}{2} \int_{\partial K} \left[\frac{\partial u_h}{\partial n} \right] (\phi - \phi_h) d\gamma \right], \end{aligned} \quad (4.107)$$

where $\phi_h \in V_h$ is typically a convenient interpolant of ϕ . By using the Cauchy-Schwarz inequality on each element K , we obtain

$$\begin{aligned} |J(e_h)| &= \left| \int_{\Omega} \nabla e_h \cdot \nabla \phi d\Omega \right| \leq \sum_{K \in \mathcal{T}_h} \left(\|f + \Delta u_h\|_{L^2(K)} \|\phi - \phi_h\|_{L^2(K)} \right. \\ &\quad \left. + \frac{1}{2} \left\| \left[\frac{\partial u_h}{\partial n} \right] \right\|_{L^2(\partial K)} \|\phi - \phi_h\|_{L^2(\partial K)} \right) \\ &\leq \sum_{K \in \mathcal{T}_h} \left[\rho_K(u_h) \max \left(\frac{1}{h_K} \|\phi - \phi_h\|_{L^2(K)}, \frac{1}{h_K^{1/2}} \|\phi - \phi_h\|_{L^2(\partial K)} \right) \right], \end{aligned}$$

$\rho_K(u_h)$ being defined according to (4.96). We now introduce the so-called *local weights*

$$\omega_K(\phi) = \max \left(\frac{1}{h_K} \|\phi - \phi_h\|_{L^2(K)}, \frac{1}{h_K^{1/2}} \|\phi - \phi_h\|_{L^2(\partial K)} \right). \quad (4.108)$$

Thus,

$$|J(e_h)| \leq \sum_{K \in \mathcal{T}_h} \rho_K(u_h) \omega_K(\phi). \quad (4.109)$$

We can observe that, in contrast to residue-type estimates introduced in Sec. 4.6.2 and 4.6.4, the estimate (4.109) depends not only on the discrete solution u_h but also on the solution ϕ of the dual problem. In particular, having considered the local estimator $\rho_K(u_h)\omega_K(\phi)$, we can say that, while the residue $\rho_K(u_h)$ measures how the discrete solution approximates the differential problem under exam, the weight $\omega_K(\phi)$ takes into account how this information is propagated in the domain as an effect of the chosen functional. Hence, the grids obtained for different choices of the functional J , i.e. of the forcing term of the adjoint problem (4.106), will be different even if we start from the same differential problem (for more details, we refer to Example 11.10). Moreover, to make the estimate (4.109) efficient, we proceed by replacing the norms $\|\phi - \phi_h\|_{L^2(K)}$ and $\|\phi - \phi_h\|_{L^2(\partial K)}$ in (4.108) with suitable estimates of the interpolation error, having chosen ϕ_h as a suitable interpolant of the dual solution ϕ .

We point out two particular cases. Choosing $J(w) = \int_{\Omega} w e_h d\Omega$ in (4.106) we would find again the estimate (4.105) for the L^2 -norm of the discretization error, provided of course that we can guarantee that the elliptic regularity result (4.74), stated in Lemma 4.6, is true. Instead, if we are interested in controlling e_h in a point \mathbf{x} of Ω , it will be indeed sufficient to define J as $J(w) = w' \langle \delta_{\mathbf{x}}, w \rangle_W$, with $W = H_0^1(\Omega) \cap C^0(\overline{\Omega})$ and $\delta_{\mathbf{x}}$ being the Dirac delta function relative to the point \mathbf{x} (see Chap. 2).

Remark 4.10 The a-posteriori analysis of this section, as well as that of the previous sections 4.6.2 and 4.6.4, can be extended to the case of more complex and significant differential problems, such as e.g., transport and diffusion problems, and of more general boundary conditions (see Example 11.10). The procedure remains basically the same. What changes is the definition of the local residue (4.96) and of the generalized jump (4.92). Indeed, while $\rho_K(u_h)$ directly depends on the differential formulation of the problem under exam, $[\partial u_h / \partial n]$ will need to take into account the conditions assigned on the boundary. •

For a more in-depth description of the adaptivity techniques provided up to now and for a presentation of other possible adaptive techniques, we refer the reader to [Ver96, Ran99, AO00].

4.7 Exercises

1. *Heat transfer in a thin rod.*

Let us consider a thin rod of length L , having temperature t_0 at the extremum $x = 0$ and insulated at the other extremum $x = L$. Let us suppose that the transversal section of the rod has constant surface equal to A and that the perimeter of A be p . The temperature t of the rod at a generic point $x \in (0, L)$ then satisfies the following mixed boundary-value problem

$$\begin{cases} -kAt'' + \sigma pt = 0, & x \in (0, L), \\ t(0) = t_0, & t'(L) = 0, \end{cases} \quad (4.110)$$

having denoted by k the thermal conductivity coefficient and by σ the convective transfer coefficient.

Verify that the exact solution of this problem is

$$t(x) = t_0 \frac{\cosh[m(L-x)]}{\cosh(mL)},$$

with $m = \sqrt{\sigma p / kA}$. Write the weak formulation of (4.110), then its Galerkin-finite element approximation. Show how the approximation error in the $H_0^1(0, L)$ -norm depends on the parameters k , σ , p and t_0 .

Finally, solve this problem using linear and quadratic finite elements on uniform grids, then evaluate the approximation error.

2. *Temperature of a fluid between two parallel plates.*

We consider a viscous fluid located between two horizontal plates, parallel and at a distance of $2H$. Suppose that the upper plate, having temperature t_{sup} , moves at a relative speed of U with respect to the lower one, having temperature t_{inf} . In such case the temperature $t : (0, 2H) \rightarrow \mathbb{R}$ of the fluid satisfies the following Dirichlet problem

$$\begin{cases} -\frac{d^2t}{dy^2} = \alpha(H-y)^2, & y \in (0, 2H), \\ t(0) = t_{inf}, & t(2H) = t_{sup}, \end{cases}$$

where $\alpha = \frac{4U^2\mu}{H^4k}$,

k being the thermal conductivity coefficient and μ the viscosity of the fluid. Find the exact solution $t(y)$, then write the weak formulation and the Galerkin-finite element formulation.

[*Solution:* the exact solution is

$$t(y) = -\frac{\alpha}{12}(H-y)^4 + \frac{t_{inf} - t_{sup}}{2H}(H-y) + \frac{t_{inf} + t_{sup}}{2} + \frac{\alpha H^4}{12}.$$

3. *Flection of a rope.*

Let us consider a rope with tension T and unit length, fixed at the extrema. The function $u(x)$, measuring the vertical displacement of the rope when subject to a transversal charge of intensity w , satisfies the following Dirichlet problem

$$\begin{cases} -u'' + \frac{k}{T}u = \frac{w}{T} & \text{in } (0, 1), \\ u(0) = 0, & \\ u(1) = 0, & \end{cases}$$

having indicated with k the elasticity coefficient of the rope. Write the weak formulation and the Galerkin-finite element formulation.

4. Prove Property 4.1.

[*Solution:* it suffices to observe that $a_{ij} = a(\varphi_j, \varphi_i) \forall i, j.$]

5. Prove (4.12).

[*Solution:* since the form is symmetric, the procedure contained in Remark 3.2 can be repeated, noting that the solution u_h satisfies problem $a(u_h, v_h) = a(u, v_h)$ for each $v_h \in V_h$. We deduce therefore that u_h minimizes $J(v_h) = a(v_h, v_h) - 2a(u, v_h)$ and therefore also minimizes $J^*(v_h) = J(v_h) + a(u, u) = a(u - v_h, u - v_h)$ (the last equality is made possible thanks to the symmetry of the bilinear form). On the other hand,

$$\sqrt{\alpha}\|u - v_h\|_V \leq \sqrt{a(u - v_h, u - v_h)} \leq \sqrt{M}\|u - v_h\|_V,$$

hence the desired result.]

6. Given a partition of an interval (a, b) into $N + 1$ sub-intervals, suppose to first number the extrema of the single sub-intervals and then their midpoints. Is this numbering more or less convenient than the one introduced in Sec. 4.3 for the discretization of the Poisson problem with finite elements in X_h^2 ? Suppose to solve the linear system by a factorization method.

[*Solution:* the obtained matrix still has only five diagonals different from zero, as the one obtained using the numbering proposed in Sec. 4.3. However, it features a higher bandwidth. Consequently, in case it is factorized, it is subject to a larger fill-in, as shown in Fig. 4.25.]

7. Consider the following one-dimensional boundary-value problem

$$\begin{cases} -(\alpha u')' + \gamma u = f, & 0 < x < 1, \\ u = 0 & \text{at } x = 0, \\ \alpha u' + \delta u = 0 & \text{at } x = 1, \end{cases}$$

where $\alpha = \alpha(x)$, $\gamma = \gamma(x)$, $f = f(x)$ are assigned functions with $0 \leq \gamma(x) \leq \gamma_1$ and $0 < \alpha_0 \leq \alpha(x) \leq \alpha_1 \forall x \in [0, 1]$, while $\delta \in \mathbb{R}$. Moreover, suppose that $f \in L^2(0, 1)$.

Write the problem's weak formulation specifying the appropriate functional spaces and hypotheses on the data to guarantee existence and uniqueness of the solution. Suppose to find an approximate solution u_h using the linear finite element method. What can be said about the existence, stability and accuracy of u_h ?

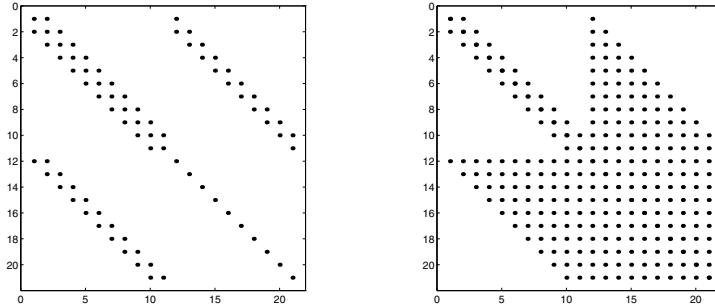


Fig. 4.25. Left: the sparsity pattern of the Galerkin finite element matrix associated to a discretization using 10 elements of the one-dimensional Poisson problem with quadratic finite elements. The numbering of the unknowns is the one reported in Exercise 6. Right: the pattern of the L and U factors of A. Note that, because of the fill-in, the number of non-null finite elements has increased from 81 in the matrix to 141 in the factors

[*Solution:* we seek $u \in V = \{v \in H^1(0, 1) : v(0) = 0\}$ such that $a(u, v) = F(v) \quad \forall v \in V$ where

$$a(u, v) = \int_0^1 \alpha u' v' dx + \int_0^1 \gamma u v dx + \delta u(1)v(1), \quad F(v) = \int_0^1 f v dx.$$

The existence and uniqueness of the solution of the weak problem are guaranteed if the hypotheses of the Lax-Milgram lemma hold. The form $a(\cdot, \cdot)$ is continuous as we have

$$|a(u, v)| \leq 2 \max(\alpha_1, \gamma_1) \|u\|_V \|v\|_V + |\delta| |v(1)| |u(1)|,$$

from which, considering that $u(1) = \int_0^1 u' dx$, we obtain

$$|a(u, v)| \leq M \|u\|_V \|v\|_V \quad \text{with } M = 3 \max(\alpha_1, \gamma_1, |\delta|).$$

We have coercivity if $\delta \geq 0$ as in such case we find

$$a(u, u) \geq \alpha_0 \|u'\|_{L^2(0,1)}^2 + u^2(1)\delta \geq \alpha_0 \|u'\|_{L^2(0,1)}^2.$$

To find the inequality in $\|\cdot\|_V$ invoking the Poincaré inequality (2.13), it suffices to prove that

$$\frac{1}{1 + C_\Omega^2} \|u\|_V^2 \leq \|u'\|_{L^2(0,1)}^2,$$

and then to conclude that

$$a(u, u) \geq \alpha^* \|u\|_V^2 \quad \text{with } \alpha^* = \frac{\alpha_0}{1 + C_\Omega^2}.$$

The fact that F is a linear and continuous functional can be verified immediately. The finite element method is a Galerkin method with $V_h = \{v_h \in X_h^1 : v_h(0) = 0\}$. Consequently, thanks to Corollaries 4.1, 4.2 we deduce that the solution u_h exists and is unique. [From the estimate (4.72) we furthermore deduce that, since $r = 1$, the error measured in the norm of V will tend to zero linearly with respect to h .]

8. Consider the following two-dimensional boundary-value problem

$$\begin{cases} -\operatorname{div}(\alpha \nabla u) + \gamma u = f & \text{in } \Omega \subset \mathbb{R}^2, \\ u = 0 & \text{on } \Gamma_D, \\ \alpha \nabla u \cdot \mathbf{n} = 0 & \text{on } \Gamma_N, \end{cases}$$

Ω being a bounded open domain with regular boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, $\overset{\circ}{\Gamma_D} \cap \overset{\circ}{\Gamma_N} = \emptyset$ with unit normal outgoing vector \mathbf{n} ; $\alpha \in L^\infty(\Omega)$, $\gamma \in L^\infty(\Omega)$, and $f \in L^2(\Omega)$ are three assigned functions with $\gamma(\mathbf{x}) \geq 0$ and $0 < \alpha_0 \leq \alpha(\mathbf{x})$ a.e. in Ω .

Analyze the existence and uniqueness of the weak solution and the stability of the solution obtained using the Galerkin method. Suppose that $u \in H^4(\Omega)$. Up to which polynomial degree would it be convenient to get by using a finite element approximation?

[*Solution:* the weak problem consists in finding $u \in V = H_{\Gamma_D}^1$ such that $a(u, v) = F(v) \forall v \in V$, where

$$a(u, v) = \int_{\Omega} \alpha \nabla u \nabla v \, d\Omega + \int_{\Omega} \gamma u v \, d\Omega, \quad F(v) = \int_{\Omega} f v \, d\Omega.$$

The bilinear form is continuous; indeed

$$\begin{aligned} |a(u, v)| &\leq \int_{\Omega} \alpha |\nabla u| |\nabla v| \, d\Omega + \int_{\Omega} |\gamma| |u| |v| \, d\Omega \\ &\leq \|\alpha\|_{L^\infty(\Omega)} \|\nabla u\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)} + \|\gamma\|_{L^\infty(\Omega)} \|u\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \\ &\leq M \|u\|_V \|v\|_V, \end{aligned}$$

having taken $M = 2 \max\{\|\alpha\|_{L^\infty(\Omega)}, \|\gamma\|_{L^\infty(\Omega)}\}$. Moreover, it is coercive (see the solution to Exercise 7) with coercivity constant given by $\alpha^* = \frac{\alpha_0}{1 + C_{\Omega}^2}$. Since

F is a linear and bounded functional, owing to the Lax-Milgram lemma the weak solution exists and is unique. As far as the Galerkin method is concerned, we introduce a subspace V_h of V with finite dimension. Then, there exists a unique solution u_h of the Galerkin problem: find $u_h \in V_h$ such that $a(u_h, v_h) = F(v_h) \forall v_h \in V_h$. Moreover, by Corollary 4.2 we have stability. As far as the choice of the optimal polynomial degree r is concerned, it is sufficient to note that the exponent s appearing in (4.26) is the minimum value between r and $p = 3$. Hence, it will be convenient to use elements of degree 3.]

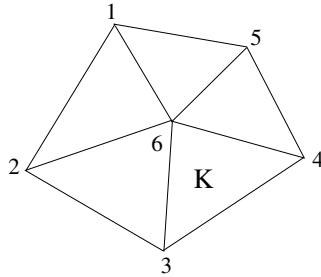


Fig. 4.26. Patch of elements for the assembly of the global matrix A

The fundamental steps of a finite element code can be summarized as follows:

- input the data;
- build the grid $\mathcal{T}_h = \{K\}$;
- build the local matrices A_K and the right-hand side elements f_K ;
- assemble the global matrix A and the one of the known term \mathbf{f} ;
- solve the linear system $A\mathbf{u} = \mathbf{f}$;
- post-process the results.

Suppose we use linear finite elements and consider the patch in Fig. 4.26.

- Referring to steps (c) and (d), explicitly write the matrix T_K allowing to pass from the local matrix A_K to the global matrix A via a transformation of the kind $T_K^T A_K T_K$. What is the dimension of such matrix?
- What sparsity pattern characterizes the A matrix associated to the patch in Fig. 4.26?
- Explicitly write the elements of the matrix A as a function of the elements of the local matrices A_K .
- In the case of a general grid \mathcal{T}_h with N_V vertices and N_T triangle, what dimension does the global matrix A have in the case of linear and quadratic finite elements, respectively?

For a more exhaustive treatment of this subject, we refer to Chap. 11.

- Prove the results summarized in Table 3.3 by using the Lagrange identity (3.42).

5

Parabolic equations

In this chapter, we consider parabolic equations of the form

$$\frac{\partial u}{\partial t} + Lu = f, \quad \mathbf{x} \in \Omega, t > 0, \quad (5.1)$$

where Ω is a domain of \mathbb{R}^d , $d = 1, 2, 3$, $f = f(\mathbf{x}, t)$ is a given function, $L = L(\mathbf{x})$ is a generic elliptic operator acting on the unknown $u = u(\mathbf{x}, t)$. When solved only for a bounded temporal interval, say for $0 < t < T$, the region $Q_T = \Omega \times (0, T)$ is called *cylinder* in the space $\mathbb{R}^d \times \mathbb{R}^+$ (see Fig. 5.1). In the case where $T = +\infty$, $Q = \{(\mathbf{x}, t) : \mathbf{x} \in \Omega, t > 0\}$ will be an infinite cylinder.

Equation (5.1) must be completed by assigning an initial condition

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (5.2)$$

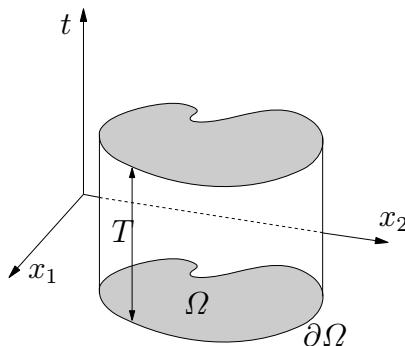


Fig. 5.1. The cylinder $Q_T = \Omega \times (0, T)$, $\Omega \subset \mathbb{R}^2$

together with boundary conditions, which can take the following form

$$\begin{aligned} u(\mathbf{x}, t) &= \varphi(\mathbf{x}, t), & \mathbf{x} \in \Gamma_D \text{ and } t > 0, \\ \frac{\partial u(\mathbf{x}, t)}{\partial n} &= \psi(\mathbf{x}, t), & \mathbf{x} \in \Gamma_N \text{ and } t > 0, \end{aligned} \quad (5.3)$$

where u_0 , φ and ψ are given functions and $\{\Gamma_D, \Gamma_N\}$ provides a boundary partition, that is $\Gamma_D \cup \Gamma_N = \partial\Omega$, $\overset{\circ}{\Gamma}_D \cap \overset{\circ}{\Gamma}_N = \emptyset$. For obvious reasons, Γ_D is called Dirichlet boundary and Γ_N Neumann boundary.

In the one-dimensional case, the problem

$$\begin{aligned} \frac{\partial u}{\partial t} - \nu \frac{\partial^2 u}{\partial x^2} &= f, & 0 < x < d, \quad t > 0, \\ u(x, 0) &= u_0(x), & 0 < x < d, \\ u(0, t) &= u(d, t) = 0, & t > 0, \end{aligned} \quad (5.4)$$

describes the evolution of the temperature $u(x, t)$ at point x and time t of a metallic bar of length d occupying the interval $[0, d]$, whose thermal conductivity is ν and whose extrema are kept at a constant temperature of zero degrees. The function u_0 describes the initial temperature, while f represents the calorific production (per unit length) provided by the bar. For this reason, (5.4) is called *heat equation*. For a particular case, see Example 1.5 of Chap. 1.

5.1 Weak formulation and its approximation

In order to numerically solve problem (5.1)-(5.3), we will introduce a weak formulation, as we did to handle elliptic problems.

We proceed formally, by multiplying for each $t > 0$ the differential equation by a test function $v = v(\mathbf{x})$ and integrating on Ω . We set $V = H_{\Gamma_D}^1(\Omega)$ (see (3.26)) and for each $t > 0$ we seek $u(t) \in V$ s.t.

$$\int_{\Omega} \frac{\partial u(t)}{\partial t} v \, d\Omega + a(u(t), v) = \int_0^t f(t) v \, d\Omega \quad \forall v \in V, \quad (5.5)$$

with $u(0) = u_0$, where $a(\cdot, \cdot)$ is the bilinear form associated to the elliptic operator L , and where we have supposed for simplicity $\varphi = 0$ and $\psi = 0$. The modification of (5.5) in the case where $\varphi \neq 0$ and $\psi \neq 0$ is left to the reader.

A sufficient condition for the existence and uniqueness of the solution to problem (5.5) is that the following hypotheses hold:

the bilinear form $a(\cdot, \cdot)$ is continuous and *weakly coercive*, that is

$$\exists \lambda \geq 0, \exists \alpha > 0 : a(v, v) + \lambda \|v\|_{L^2(\Omega)}^2 \geq \alpha \|v\|_V^2 \quad \forall v \in V,$$

so we find again for $\lambda = 0$ the standard definition of coercivity. Moreover, we require $u_0 \in L^2(\Omega)$ and $f \in L^2(Q)$. Then, problem (5.5) admits a unique solution $u \in L^2(\mathbb{R}^+; V) \cap C^0(\mathbb{R}^+; L^2(\Omega))$, with $V = H_{T_D}^1(\Omega)$.

For the definition of these functional spaces, see Sec. 2.7. For the proof, see [QV94, Sec. 11.1.1].

Some a priori estimates of the solution u will be provided in the following section.

We now consider the Galerkin approximation of problem (5.5):
for each $t > 0$, find $u_h(t) \in V_h$ s.t.

$$\int_{\Omega} \frac{\partial u_h(t)}{\partial t} v_h \, d\Omega + a(u_h(t), v_h) = \int_{\Omega} f(t) v_h \, d\Omega \quad \forall v_h \in V_h \quad (5.6)$$

with $u_h(0) = u_{0h}$, where $V_h \subset V$ is a suitable space of finite dimension and u_{0h} is a convenient approximation of u_0 in the space V_h . Such problem is called *semi-discretization* of (5.5), as the temporal variable has not yet been discretized.

To provide an algebraic interpretation of (5.6) we introduce a basis $\{\varphi_j\}$ for V_h (as we did in the previous chapters), and we observe that it suffices that (5.6) is verified for the basis functions in order to be satisfied by all the functions of the subspace. Moreover, as for each $t > 0$ the solution to the Galerkin problem belongs to the subspace as well, we will have

$$u_h(\mathbf{x}, t) = \sum_{j=1}^{N_h} u_j(t) \varphi_j(\mathbf{x}),$$

where the $\{u_j(t)\}$ coefficients represent the unknowns of problem (5.6).

Denoting by $\dot{u}_j(t)$ the derivatives of the function $u_j(t)$ with respect to time, (5.6) becomes

$$\int_{\Omega} \sum_{j=1}^{N_h} \dot{u}_j(t) \varphi_j \varphi_i \, d\Omega + a \left(\sum_{j=1}^{N_h} u_j(t) \varphi_j, \varphi_i \right) = \int_{\Omega} f(t) \phi_i \, d\Omega, \quad i = 1, 2, \dots, N_h,$$

that is

$$\sum_{j=1}^{N_h} \dot{u}_j(t) \underbrace{\int_{\Omega} \varphi_j \varphi_i \, d\Omega}_{m_{ij}} + \sum_{j=1}^{N_h} u_j(t) \underbrace{a(\varphi_j, \varphi_i)}_{a_{ij}} = \underbrace{\int_{\Omega} f(t) \phi_i \, d\Omega}_{f_i(t)}, \quad i = 1, 2, \dots, N_h. \quad (5.7)$$

If we define the vector of unknowns $\mathbf{u} = (u_1(t), u_2(t), \dots, u_{N_h}(t))^T$, the mass matrix $M = [m_{ij}]$, the stiffness matrix $A = [a_{ij}]$ and the right-hand side vector $\mathbf{f} = (f_1(t), f_2(t), \dots, f_{N_h}(t))^T$, the system (5.7) can be rewritten in matrix form as

$$M \dot{\mathbf{u}}(t) + A \mathbf{u}(t) = \mathbf{f}(t).$$

For the numerical solution of this ODE system, many finite difference methods are available. See, e.g., [QSS07, Chap. 11]. Here we limit ourselves to considering the so-called θ -method. The latter discretizes the temporal derivative by a simple incremental ratio and replaces the other terms via a linear combination of the value at time t^k and of the value at time t^{k+1} , depending on the real parameter θ ($0 \leq \theta \leq 1$),

$$M \frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} + A[\theta \mathbf{u}^{k+1} + (1 - \theta) \mathbf{u}^k] = \theta \mathbf{f}^{k+1} + (1 - \theta) \mathbf{f}^k. \quad (5.8)$$

As usual, the real positive parameter $\Delta t = t^{k+1} - t^k$, $k = 0, 1, \dots$, denotes the discretization step (here assumed to be constant), while the super-index k indicates that the quantity under consideration refers to the time t^k . Let us see some particular cases of (5.8):

- for $\theta = 0$ we obtain the *forward Euler* (or *explicit Euler*) method

$$M \frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} + A \mathbf{u}^k = \mathbf{f}^k$$

which is first-order accurate with respect to Δt ;

- for $\theta = 1$ we have the *backward Euler* (or *implicit Euler*) method

$$M \frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} + A \mathbf{u}^{k+1} = \mathbf{f}^{k+1}$$

which is itself first-order with respect to Δt ;

- for $\theta = 1/2$ we have the *Crank-Nicolson* (or *trapezoidal*) method

$$M \frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} + \frac{1}{2} A (\mathbf{u}^{k+1} + \mathbf{u}^k) = \frac{1}{2} (\mathbf{f}^{k+1} + \mathbf{f}^k)$$

which is second-order accurate with respect to Δt . (More precisely, $\theta = 1/2$ is the only value for which we obtain a second-order method.)

Let us consider the two extreme cases, $\theta = 0$ and $\theta = 1$. For both, we obtain a system of linear equations: if $\theta = 0$, the system to solve has matrix $\frac{M}{\Delta t}$, in the second case it has matrix $\frac{M}{\Delta t} + A$. We observe that the M matrix is invertible, being positive definite (see Exercise 1).

In the $\theta = 0$ case, if we make matrix M diagonal, we actually decouple the system equations. This operation is performed by executing the so-called *lumping* of the mass matrix (see Sec. 11.5). However, this scheme is not unconditionally stable (see Sec. 5.4) and, in the case where V_h is a subspace of finite elements, we have the following stability condition (see Sec. 5.4)

$$\exists c > 0 : \Delta t \leq ch^2 \quad \forall h > 0,$$

that does not allow an arbitrary choice of Δt with respect to h .

In the case $\theta > 0$, the system will have the form $K \mathbf{u}^{k+1} = \mathbf{g}$, where \mathbf{g} is the known term and $K = \frac{M}{\Delta t} + \theta A$. Such matrix is however invariant in time (the operator

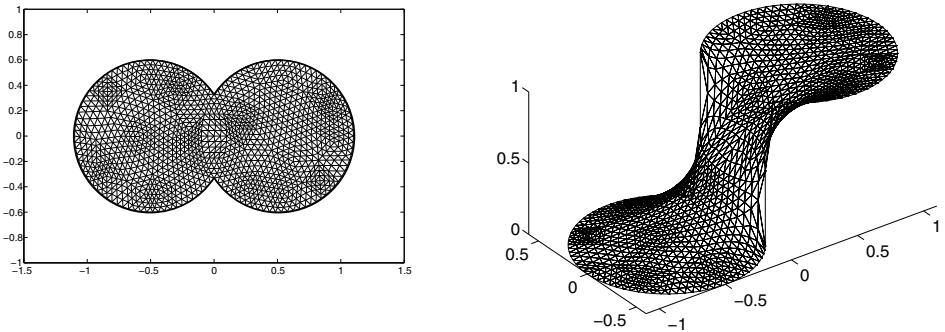


Fig. 5.2. Solution of the heat equation for the problem of Example 5.1

L , and therefore the matrix A , being independent of time); if the space mesh does not change, it can then be factorized once and for all at the beginning of the process. Since M is symmetric, if A is symmetric too, the K matrix associated to the system will also be symmetric. Hence, we can use, for instance, the Cholesky factorization, $K=H H^T$, H being lower triangular. At each time step, we will therefore have to solve two triangular systems in N_h unknowns:

$$\begin{aligned} Hy &= g, \\ H^T u^{k+1} &= y \end{aligned}$$

(see Chap. 7 and also [QSS07, Chap. 3]).

Example 5.1 Let us suppose to solve the heat equation $\frac{\partial u}{\partial t} - 0.1\Delta u = 0$ on the domain $\Omega \subset \mathbb{R}^2$ of Fig. 5.2 (left) which is the union of two circles of radius 0.5 and center $(-0.5, 0)$ resp. $(0.5, 0)$. We assign Dirichlet conditions on the whole boundary taking $u(\mathbf{x}, t) = 1$ for the points on $\partial\Omega$ for which $x_1 \geq 0$ and $u(\mathbf{x}, t) = 0$ if $x_1 < 0$. The initial condition is $u(\mathbf{x}, 0) = 1$ for $x_1 \geq 0$ and null elsewhere. In Fig. 5.2, we report the solution obtained at time $t = 1$. We have used linear finite elements in space and the implicit Euler method in time with $\Delta t = 0.01$. As it can be seen, the initial discontinuity has been regularized, in accordance with the boundary conditions. ■

5.2 A priori estimates

Let us consider problem (5.5); since the corresponding equations must hold for each $v \in V$, it will be legitimate to pose $v = u(t)$ (t being given), solution of the problem itself, yielding

$$\int_{\Omega} \frac{\partial u(t)}{\partial t} u(t) d\Omega + a(u(t), u(t)) = \int_{\Omega} f(t) u(t) d\Omega \quad \forall t > 0. \quad (5.9)$$

Considering the individual terms, we have

$$\int_{\Omega} \frac{\partial u(t)}{\partial t} u(t) d\Omega = \frac{1}{2} \frac{\partial}{\partial t} \int_{\Omega} |u(t)|^2 d\Omega = \frac{1}{2} \frac{\partial}{\partial t} \|u(t)\|_{L^2(\Omega)}^2. \quad (5.10)$$

If we assume for simplicity that the bilinear form is coercive (with coercivity constant equal to α), we obtain

$$a(u(t), u(t)) \geq \alpha \|u(t)\|_V^2,$$

while thanks to the Cauchy-Schwarz inequality, we find

$$(f(t), u(t)) \leq \|f(t)\|_{L^2(\Omega)} \|u(t)\|_{L^2(\Omega)}. \quad (5.11)$$

In the remainder, we will often use the following *Young inequality*

$$\forall a, b \in \mathbb{R}, \quad ab \leq \varepsilon a^2 + \frac{1}{4\varepsilon} b^2 \quad \forall \varepsilon > 0, \quad (5.12)$$

that derives from the elementary inequality

$$\left(\sqrt{\varepsilon} a - \frac{1}{2\sqrt{\varepsilon}} b \right)^2 \geq 0.$$

Using first the Poincaré inequality (2.13) and next the Young inequality, we obtain

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} \|u(t)\|_{L^2(\Omega)}^2 + \alpha \|\nabla u(t)\|_{L^2(\Omega)}^2 &\leq \|f(t)\|_{L^2(\Omega)} \|u(t)\|_{L^2(\Omega)} \\ &\leq \frac{C_\Omega^2}{2\alpha} \|f(t)\|_{L^2(\Omega)}^2 + \frac{\alpha}{2} \|\nabla u(t)\|_{L^2(\Omega)}^2. \end{aligned} \quad (5.13)$$

Then, by integrating in time we obtain, for all $t > 0$,

$$\begin{aligned} \|u(t)\|_{L^2(\Omega)}^2 + \alpha \int_0^t \|\nabla u(s)\|_{L^2(\Omega)}^2 ds \\ \leq \|u_0\|_{L^2(\Omega)}^2 + \frac{C_\Omega^2}{\alpha} \int_0^t \|f(s)\|_{L^2(\Omega)}^2 ds. \end{aligned} \quad (5.14)$$

This is an a priori energy estimate. Different kinds of a priori estimates can be obtained as follows. Note that

$$\frac{1}{2} \frac{d}{dt} \|u(t)\|_{L^2(\Omega)}^2 = \|u(t)\|_{L^2(\Omega)} \frac{d}{dt} \|u(t)\|_{L^2(\Omega)}.$$

Then from (5.9), using (5.10) and (5.11) we obtain (still using the Poincaré inequality)

$$\begin{aligned} \|u(t)\|_{L^2(\Omega)} \frac{d}{dt} \|u(t)\|_{L^2(\Omega)} + \frac{\alpha}{C_\Omega} \|u(t)\|_{L^2(\Omega)} \|\nabla u(t)\|_{L^2(\Omega)} \\ \leq \|f(t)\|_{L^2(\Omega)} \|u(t)\|_{L^2(\Omega)}, \quad t > 0. \end{aligned}$$

If $\|u(t)\|_{L^2(\Omega)} \neq 0$ (otherwise we should proceed differently, however the final result is still true) we can divide by $\|u(t)\|_{L^2(\Omega)}$ and integrate in time to yield

$$\|u(t)\|_{L^2(\Omega)} \leq \|u_0\|_{L^2(\Omega)} + \int_0^t \|f(s)\|_{L^2(\Omega)} ds, \quad t > 0. \quad (5.15)$$

This is a further a priori estimate.

Let us now use the first inequality in (5.13), and integrate in time to yield

$$\begin{aligned} & \|u(t)\|_{L^2(\Omega)}^2 + 2\alpha \int_0^t \|\nabla u(s)\|^2 ds \\ & \leq \|u_0\|_{L^2(\Omega)}^2 + 2 \int_0^t \|f(s)\|_{L^2(\Omega)} \|u(s)\|_{L^2(\Omega)} ds \\ & \leq \|u_0\|_{L^2(\Omega)}^2 + 2 \int_0^t \|f(s)\|_{L^2(\Omega)} \cdot (\|u_0\|_{L^2(\Omega)}^2 + \int_0^s \|f(\tau)\|_{L^2(\Omega)} d\tau) ds \\ & \quad (\text{using (5.15)}) \\ & = \|u_0\|_{L^2(\Omega)}^2 + 2 \int_0^t \|f(s)\|_{L^2(\Omega)} \|u_0\|_{L^2(\Omega)} + 2 \int_0^t \|f(s)\|_{L^2(\Omega)} \int_0^s \|f(\tau)\|_{L^2(\Omega)} d\tau d\tau \\ & = (\|u_0\|_{L^2(\Omega)} + \int_0^t \|f(s)\| ds)^2. \end{aligned} \quad (5.16)$$

The latter equality follows upon noticing that

$$\|f(s)\|_{L^2(\Omega)} \int_0^s \|f(\tau)\|_{L^2(\Omega)} d\tau = \frac{d}{ds} \left(\int_0^s \|f(\tau)\|_{L^2(\Omega)} d\tau \right)^2.$$

We therefore conclude with the additional a priori estimate

$$\begin{aligned} & (\|u(t)\|_{L^2(\Omega)}^2 + 2\alpha \int_0^t \|\nabla u(s)\|_{L^2(\Omega)}^2 ds)^{\frac{1}{2}} \\ & \leq \|u_0\|_{L^2(\Omega)} + \int_0^t \|f(s)\|_{L^2(\Omega)} ds, \quad t > 0. \end{aligned} \quad (5.17)$$

We have seen that we can formulate the Galerkin problem (5.6) for problem (5.5) and that the latter, under suitable hypotheses, admits a unique solution. Similarly to what we did for problem (5.5) we can prove the following a priori (stability) estimates for the solution to problem (5.6):

$$\begin{aligned} & \|u_h(t)\|_{L^2(\Omega)}^2 + \alpha \int_0^t \|\nabla u_h(s)\|_{L^2(\Omega)}^2 ds \\ & \leq \|u_{0h}(t)\|_{L^2(\Omega)}^2 + \frac{C_\Omega^2}{\alpha} \int_0^t \|f(s)\|_{L^2(\Omega)}^2 ds, \quad t > 0. \end{aligned} \quad (5.18)$$

For its proof we can take, for every $t > 0$, $v_h = u_h(t)$ and proceed as we did to obtain (5.13). Then, by recalling that the initial data is $u_h(0) = u_{0h}$, we can deduce the following discrete counterparts of (5.15) and (5.17):

$$\|u_h(t)\|_{L^2(\Omega)} \leq \|u_{0h}(t)\|_{L^2(\Omega)} + \int_0^t \|f(s)\|_{L^2(\Omega)} ds, \quad t > 0, \quad (5.19)$$

and

$$\begin{aligned} & (\|u_h(t)\|_{L^2(\Omega)}^2 + 2\alpha \int_0^t \|\nabla u_h(s)\|_{L^2(\Omega)}^2 ds)^{\frac{1}{2}} \\ & \leq \|u_{0h}(t)\|_{L^2(\Omega)} + \int_0^t \|f(s)\|_{L^2(\Omega)} ds, \quad t > 0. \end{aligned} \quad (5.20)$$

5.3 Convergence analysis of the semi-discrete problem

Let us consider the problem (5.5) and its approximation (5.6). We want to prove the convergence of u_h to u in suitable norms.

By the coercivity hypotheses we can write

$$\begin{aligned} \alpha \|u - u_h\|_{H^1(\Omega)}^2 &\leq a(u - u_h, u - u_h) \\ &= a(u - u_h, u - v_h) + a(u - u_h, v_h - u_h) \quad \forall v_h \in V_h. \end{aligned}$$

By subtracting equation (5.6) from equation (5.5) and setting $w_h = v_h - u_h$ we have

$$\left(\frac{\partial(u - u_h)}{\partial t}, w_h \right) + a(u - u_h, w_h) = 0,$$

where $(v, w) = \int_{\Omega} vw \, d\Omega$ is the scalar product of $L^2(\Omega)$. Then

$$\alpha \|u - u_h\|_{H^1(\Omega)}^2 \leq a(u - u_h, u - v_h) - \left(\frac{\partial(u - u_h)}{\partial t}, w_h \right). \quad (5.21)$$

We analyze the two right-hand side terms separately:

- using the continuity of the form $a(\cdot, \cdot)$ and the Young inequality, we obtain

$$\begin{aligned} a(u - u_h, u - v_h) &\leq M \|u - u_h\|_{H^1(\Omega)} \|u - v_h\|_{H^1(\Omega)} \\ &\leq \frac{\alpha}{2} \|u - u_h\|_{H^1(\Omega)}^2 + \frac{M^2}{2\alpha} \|u - v_h\|_{H^1(\Omega)}^2; \end{aligned}$$

- writing w_h in the form $w_h = (v_h - u) + (u - u_h)$ we obtain

$$-\left(\frac{\partial(u - u_h)}{\partial t}, w_h \right) = \left(\frac{\partial(u - u_h)}{\partial t}, u - v_h \right) - \frac{1}{2} \frac{d}{dt} \|u - u_h\|_{L^2(\Omega)}^2. \quad (5.22)$$

Replacing these two results in (5.21), we obtain

$$\begin{aligned} &\frac{1}{2} \frac{d}{dt} \|u - u_h\|_{L^2(\Omega)}^2 + \frac{\alpha}{2} \|u - u_h\|_{H^1(\Omega)}^2 \\ &\leq \frac{M^2}{2\alpha} \|u - v_h\|_{H^1(\Omega)}^2 + \left(\frac{\partial(u - u_h)}{\partial t}, u - v_h \right). \end{aligned}$$

Multiplying both sides by 2 and integrating in time between 0 and t we find

$$\begin{aligned} &\|(u - u_h)(t)\|_{L^2(\Omega)}^2 + \alpha \int_0^t \|(u - u_h)(s)\|_{H^1(\Omega)}^2 \, ds \leq \|(u - u_h)(0)\|_{L^2(\Omega)}^2 \\ &+ \frac{M^2}{\alpha} \int_0^t \|u(s) - v_h\|_{H^1(\Omega)}^2 \, ds + 2 \int_0^t \left(\frac{\partial}{\partial t} (u - u_h)(s), u(s) - v_h \right) \, ds. \end{aligned} \quad (5.23)$$

Integrating by parts and using the Young inequality, we obtain

$$\begin{aligned} & \int_0^t \left(\frac{\partial}{\partial t}(u - u_h)(s), u(s) - v_h \right) ds = - \int_0^t \left((u - u_h)(s), \frac{\partial}{\partial t}(u(s) - v_h) \right) ds \\ & + ((u - u_h)(t), (u - v_h)(t)) - ((u - u_h)(0), (u - v_h)(0)) \\ & \leq \frac{1}{4} \int_0^t \| (u - u_h)(s) \|_{L^2(\Omega)}^2 ds + \int_0^t \left\| \frac{\partial(u(s) - v_h)}{\partial t} \right\|_{L^2(\Omega)}^2 ds + \frac{1}{4} \| (u - u_h)(t) \|_{L^2(\Omega)}^2 \\ & + \| (u - v_h)(t) \|_{L^2(\Omega)}^2 + \| (u - u_h)(0) \|_{L^2(\Omega)} \| (u - v_h)(0) \|_{L^2(\Omega)}. \end{aligned}$$

From (5.23) we thus obtain

$$\begin{aligned} & \frac{1}{2} \| (u - u_h)(t) \|_{L^2(\Omega)}^2 + \alpha \int_0^t \| (u - u_h)(s) \|_{H^1(\Omega)}^2 ds \\ & \leq \frac{M^2}{\alpha} \int_0^t \| u(s) - v_h \|_{H^1(\Omega)}^2 ds + 2 \int_0^t \left\| \frac{\partial(u(s) - v_h)}{\partial t} \right\|_{L^2(\Omega)}^2 ds \\ & + 2 \| (u - v_h)(t) \|_{L^2(\Omega)}^2 + \| (u - u_h)(0) \|_{L^2(\Omega)}^2 \\ & + 2 \| (u - u_h)(0) \|_{L^2(\Omega)} \| (u - v_h)(0) \|_{L^2(\Omega)} + \frac{1}{2} \int_0^t \| (u - u_h)(s) \|_{L^2(\Omega)}^2 ds. \end{aligned} \quad (5.24)$$

Let us now suppose that V_h is the space of finite elements of degree r , more precisely $V_h = \{v_h \in X_h^r : v_h|_{\Gamma_D} = 0\}$, and let us choose, at each t , $v_h = \Pi_h^r u(t)$, the interpolant of $u(t)$ in V_h (see (4.20)). Thanks to (4.69) we have, assuming that u is sufficiently regular,

$$h \|u - \Pi_h^r u\|_{H^1(\Omega)} + \|u - \Pi_h^r u\|_{L^2(\Omega)} \leq C_2 h^{r+1} |u|_{H^{r+1}(\Omega)}.$$

Hence, the addenda of the right-hand side of inequality (5.24) are bounded as follows:

$$\begin{aligned} E_1 &= \frac{M^2}{\alpha} \int_0^t \| u(s) - v_h \|_{H^1(\Omega)}^2 ds \leq C_1 h^{2r} \int_0^t |u(s)|_{H^{r+1}(\Omega)}^2 ds, \\ E_2 &= 2 \int_0^t \left\| \frac{\partial(u - v_h)}{\partial t}(s) \right\|_{L^2(\Omega)}^2 ds \leq C_2 h^{2r} \int_0^t \left| \frac{\partial u}{\partial t}(s) \right|_{H^r(\Omega)}^2 ds, \\ E_3 &= 2 \| (u - v_h)(t) \|_{L^2(\Omega)}^2 \leq C_3 h^{2r} |u|_{H^r(\Omega)}^2, \\ E_4 &= \| (u - u_h)(0) \|_{L^2(\Omega)}^2 + 2 \| (u - u_h)(0) \|_{L^2(\Omega)} \| (u - v_h)(0) \|_{L^2(\Omega)} \\ &\leq C_4 h^{2r} |u(0)|_{H^r(\Omega)}^2. \end{aligned}$$

Consequently,

$$E_1 + E_2 + E_3 + E_4 \leq Ch^{2r} N(u),$$

where $N(u)$ is a suitable function depending on u and on $\frac{\partial u}{\partial t}$. This way, we obtain the inequality

$$\begin{aligned} & \frac{1}{2} \| (u - u_h)(t) \|_{L^2(\Omega)}^2 + \alpha \int_0^t \| (u - u_h)(s) \|_{H^1(\Omega)}^2 ds \\ & \leq Ch^{2r} N(u) + \frac{1}{2} \int_0^t \| (u - u_h)(s) \|_{L^2(\Omega)}^2 ds \end{aligned}$$

and finally, applying the Gronwall lemma (see Sec. 2.7), we obtain the a priori error estimate for all $t > 0$

$$\|u(t) - u_h(t)\|_{L^2(\Omega)}^2 + 2\alpha \int_0^t \|u(s) - u_h(s)\|_{H^1(\Omega)}^2 ds \leq Ch^{2r} N(u) e^t. \quad (5.25)$$

By a different proof technique that does not make use of the Gronwall lemma we can prove an error inequality similar to (5.25), however without the exponential factor e^t at the right-hand side.

We proceed as follows. If we subtract (5.6) from (5.5) and set $E_h = u - u_h$, we obtain that

$$\left(\frac{\partial E_h}{\partial t}, v_h \right) + a(E_h, v_h) = 0 \quad \forall v_h \in V_h, \quad \forall t > 0.$$

If for the sake of simplicity, we suppose that $a(\cdot, \cdot)$ is symmetric, we can define the orthogonal projection operator

$$\Pi_{1,h}^r : V \rightarrow V_h : \forall w \in V, \quad a(\Pi_{1,h}^r w - w, v_h) = 0 \quad \forall v_h \in V_h. \quad (5.26)$$

Using the results seen in Chap. 3, we can prove (see [QV94, Sec. 3.5]) that there exists a constant $C > 0$ s.t., $\forall w \in V \cap H^{r+1}(\Omega)$,

$$\|\Pi_{1,h}^r w - w\|_{H^1(\Omega)} + h^{-1} \|\Pi_{1,h}^r w - w\|_{L^2(\Omega)} \leq Ch^p |w|_{H^{p+1}(\Omega)}, \quad 0 \leq p \leq r. \quad (5.27)$$

Then we set

$$E_h = \sigma_h + e_h = (u - \Pi_{1,h}^r u) + (\Pi_{1,h}^r u - u_h). \quad (5.28)$$

Note that the orthogonal projection error σ_h can be bounded by inequality (5.27) and that e_h is an element of the subspace V_h . Then

$$\left(\frac{\partial e_h}{\partial t}, v_h \right) + a(e_h, v_h) = -\left(\frac{\partial \sigma_h}{\partial t}, v_h \right) - a(\sigma_h, v_h) \quad \forall v_h \in V_h \quad \forall t > 0.$$

If we take at every $t > 0$, $v_h = e_h(t)$, and proceed as done in Sec. 5.2 to deduce the a priori estimates on the semi-discrete solution u_h , we obtain

$$\begin{aligned} \frac{1}{2} \frac{d}{dt} \|e_h(t)\|_{L^2(\Omega)}^2 &+ \alpha \|\nabla e_h(t)\|_{L^2(\Omega)}^2 \\ &\leq |a(\sigma_h(t), e_h(t))| + \left| \left(\frac{\partial}{\partial t} \sigma_h(t), e_h(t) \right) \right|. \end{aligned} \quad (5.29)$$

Using the continuity of the bilinear form $a(\cdot, \cdot)$ (M being the continuity constant) and the Young inequality, we obtain

$$|a(\sigma_h(t), e_h(t))| \leq \frac{\alpha}{4} \|\nabla e_h(t)\|_{L^2(\Omega)}^2 + \frac{M^2}{\alpha} \|\nabla \sigma_h(t)\|_{L^2(\Omega)}^2.$$

Moreover, using the Poincaré inequality and once more the Young inequality it follows that

$$\begin{aligned} \left| \left(\frac{\partial}{\partial t} \sigma_h(t), e_h(t) \right) \right| &\leq \left\| \frac{\partial}{\partial t} \sigma_h(t) \right\|_{L^2(\Omega)} C_\Omega \|\nabla e_h(t)\|_{L^2(\Omega)} \\ &\leq \frac{\alpha}{4} \|\nabla e_h(t)\|_{L^2(\Omega)}^2 + \frac{C_\Omega^2}{\alpha} \left\| \frac{\partial}{\partial t} \sigma_h(t) \right\|_{L^2(\Omega)}^2. \end{aligned}$$

Using these bounds in (5.29) we obtain, after integrating w.r.t. t :

$$\begin{aligned} &\|e_h(t)\|_{L^2(\Omega)}^2 + \alpha \int_0^t \|\nabla e_h(s)\|_{L^2(\Omega)}^2 ds \\ &\leq \|e_h(0)\|_{L^2(\Omega)}^2 + \frac{2M^2}{\alpha} \int_0^t \|\nabla \sigma_h(s)\|_{L^2(\Omega)}^2 ds \\ &\quad + \frac{2C_\Omega^2}{\alpha} \int_0^t \left\| \frac{\partial}{\partial t} \sigma_h(s) \right\|_{L^2(\Omega)}^2 ds, \quad t > 0. \end{aligned}$$

At this point we can use (5.27) to bound the errors on the right-hand side:

$$\begin{aligned} \|\nabla \sigma_h(t)\|_{L^2(\Omega)} &\leq Ch^r |u(t)|_{H^{r+1}(\Omega)}, \\ \left\| \frac{\partial}{\partial t} \sigma_h(t) \right\|_{L^2(\Omega)} &= \left\| \left(\frac{\partial u}{\partial t} - \Pi_{1,h}^r \frac{\partial u}{\partial t} \right)(t) \right\|_{L^2(\Omega)} \leq Ch^r \left\| \frac{\partial u(t)}{\partial t} \right\|_{H^{r+1}(\Omega)}. \end{aligned}$$

Finally, note that $\|e_h(0)\|_{L^2(\Omega)} \leq Ch^r |u_0|_{H^r(\Omega)}$, still using (5.27). Since, for any norm $\|\cdot\|$,

$$\|u - u_h\| \leq \|\sigma_h\| + \|e_h\|$$

(owing to 5.28), using the previous estimates we can conclude that there exists a constant $C > 0$ independent of both t and h s.t.

$$\begin{aligned} &\|u(t) - u_h(t)\|_{L^2(\Omega)}^2 + \alpha \int_0^t \|\nabla u(s) - \nabla u_h(s)\|_{L^2(\Omega)}^2 ds \\ &\leq Ch^{2r} \{ |u_0|_{H^r(\Omega)}^2 + \int_0^t |u(s)|_{H^{r+1}(\Omega)}^2 ds + \int_0^t \left\| \frac{\partial u(s)}{\partial t} \right\|_{H^{r+1}(\Omega)}^2 ds \}. \end{aligned}$$

This error inequality has a form similar to (5.25), however, as anticipated, the right-hand side does not include the exponential factor e^t any longer. Further error estimates are proven, e.g. in [QV94, Chap. 11].

5.4 Stability analysis of the θ -method

We now analyze the stability of the fully discretized problem.

Applying the θ -method to the Galerkin problem (5.6) we obtain

$$\begin{aligned} & \left(\frac{u_h^{k+1} - u_h^k}{\Delta t}, v_h \right) + a(\theta u_h^{k+1} + (1-\theta)u_h^k, v_h) \\ &= \theta F^{k+1}(v_h) + (1-\theta)F^k(v_h) \quad \forall v_h \in V_h, \end{aligned} \tag{5.30}$$

for each $k \geq 0$, with $u_h^0 = u_{0h}$; F^k indicates that the functional is evaluated at time t^k .

We will limit ourselves to the case where $F = 0$ and start to consider the case of the implicit Euler method where $\theta = 1$, that is

$$\left(\frac{u_h^{k+1} - u_h^k}{\Delta t}, v_h \right) + a(u_h^{k+1}, v_h) = 0 \quad \forall v_h \in V_h.$$

By choosing $v_h = u_h^{k+1}$, we obtain

$$(u_h^{k+1}, u_h^{k+1}) + \Delta t a(u_h^{k+1}, u_h^{k+1}) = (u_h^k, u_h^{k+1}).$$

By now exploiting the following inequalities

$$a(u_h^{k+1}, u_h^{k+1}) \geq \alpha \|u_h^{k+1}\|_V^2, \quad (u_h^k, u_h^{k+1}) \leq \frac{1}{2} \|u_h^k\|_{L^2(\Omega)}^2 + \frac{1}{2} \|u_h^{k+1}\|_{L^2(\Omega)}^2,$$

the former deriving from the coercivity of the bilinear form $a(\cdot, \cdot)$ and the latter from the Cauchy-Schwarz and Young inequalities, we obtain

$$\|u_h^{k+1}\|_{L^2(\Omega)}^2 + 2\alpha \Delta t \|u_h^{k+1}\|_V^2 \leq \|u_h^k\|_{L^2(\Omega)}^2. \tag{5.31}$$

By summing over the k index from 0 to $n - 1$ we deduce that

$$\|u_h^n\|_{L^2(\Omega)}^2 + 2\alpha \Delta t \sum_{k=0}^{n-1} \|u_h^{k+1}\|_V^2 \leq \|u_{0h}\|_{L^2(\Omega)}^2.$$

In the case where $f \neq 0$, using the discrete Gronwall lemma (see Sec. 2.7) it can be proven in a similar way that

$$\|u_h^n\|_{L^2(\Omega)}^2 + 2\alpha \Delta t \sum_{k=1}^n \|u_h^k\|_V^2 \leq C(t^n) \left(\|u_{0h}\|_{L^2(\Omega)}^2 + \sum_{k=1}^n \Delta t \|f^k\|_{L^2(\Omega)}^2 \right). \tag{5.32}$$

Such relation is similar to (5.20), provided that the integrals $\int_0^t \cdot ds$ are approximated by a composite numerical integration formula with step Δt .

Finally, observing that $\|u_h^{k+1}\|_V \geq \|u_h^{k+1}\|_{L^2(\Omega)}$, we deduce from (5.31) that, for each given $\Delta t > 0$,

$$\lim_{k \rightarrow \infty} \|u_h^k\|_{L^2(\Omega)} = 0,$$

that is the backward Euler method is absolutely stable without any restriction on the time step Δt .

Before analyzing the general case where θ is an arbitrary parameter ranging between 0 and 1, we introduce the following definition.

We say that the scalar λ is an *eigenvalue of the bilinear form* $a(\cdot, \cdot) : V \times V \mapsto \mathbb{R}$ and that $w \in V$ is its corresponding *eigenfunction* if it turns out that

$$a(w, v) = \lambda(w, v) \quad \forall v \in V.$$

If the bilinear form $a(\cdot, \cdot)$ is symmetric and coercive, it has infinite eigenvalues, all positive real, which form an infinite sequence; moreover, its eigenfunctions form a basis of the space V .

The eigenvalues and eigenfunctions of $a(\cdot, \cdot)$ can be approximated by finding the pairs $\lambda_h \in \mathbb{R}$ and $w_h \in V_h$ which satisfy

$$a(w_h, v_h) = \lambda_h(w_h, v_h) \quad \forall v_h \in V_h. \quad (5.33)$$

From an algebraic viewpoint, problem (5.33) can be formulated as follows

$$Aw = \lambda_h Mw,$$

where A is the stiffness matrix and M the mass matrix. We are therefore dealing with a *generalized eigenvalue problem*.

Such eigenvalues are all positive and as many as N_h (N_h being as usual the dimension of the subspace V_h); after ordering them in ascending order, $\lambda_h^1 \leq \lambda_h^2 \leq \dots \leq \lambda_h^{N_h}$, we have

$$\lambda_h^{N_h} \rightarrow \infty \quad \text{for } N_h \rightarrow \infty.$$

Moreover, the corresponding eigenfunctions form a basis for the subspace V_h and can be chosen in order to be *orthonormal* with respect to the scalar product of $L^2(\Omega)$. This means that, denoting by w_h^i the eigenfunction corresponding to the eigenvalue λ_h^i , we have $(w_h^i, w_h^j) = \delta_{ij} \quad \forall i, j = 1, \dots, N_h$. Thus, each function $v_h \in V_h$ can be represented as follows

$$v_h(\mathbf{x}) = \sum_{j=1}^{N_h} v_j w_h^j(\mathbf{x})$$

and, thanks to the eigenfunction orthonormality,

$$\|v_h\|_{L^2(\Omega)}^2 = \sum_{j=1}^{N_h} v_j^2. \quad (5.34)$$

Let us now consider an arbitrary $\theta \in [0, 1]$ and let us limit ourselves to the case where the bilinear form $a(\cdot, \cdot)$ is symmetric (otherwise, although the final stability result holds in general, the following proof would not be applicable, as the eigenfunctions would not necessarily form a basis). Let $\{w_h^i\}$ still denote the discrete (orthonormal) eigenfunctions of $a(\cdot, \cdot)$. Since $u_h^k \in V_h$, we can write

$$u_h^k(\mathbf{x}) = \sum_{j=1}^{N_h} u_j^k w_h^j(\mathbf{x}).$$

We observe that in this modal expansion, the u_j^k no longer represent the nodal values of u_h^k . If we now set $F = 0$ in (5.30) and take $v_h = w_h^i$, we find

$$\frac{1}{\Delta t} \sum_{j=1}^{N_h} [u_j^{k+1} - u_j^k] \left(w_h^j, w_h^i \right) + \sum_{j=1}^{N_h} [\theta u_j^{k+1} + (1 - \theta) u_j^k] a(w_h^j, w_h^i) = 0,$$

for each $i = 1, \dots, N_h$. For each pair $i, j = 1, \dots, N_h$ we have

$$a(w_h^j, w_h^i) = \lambda_h^j (w_h^j, w_h^i) = \lambda_h^j \delta_{ij} = \lambda_h^i,$$

and thus, for each $i = 1, \dots, N_h$,

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} + [\theta u_i^{k+1} + (1 - \theta) u_i^k] \lambda_h^i = 0.$$

Solving now with respect to u_i^{k+1} , we find

$$u_i^{k+1} = u_i^k \frac{1 - (1 - \theta) \lambda_h^i \Delta t}{1 + \theta \lambda_h^i \Delta t}.$$

Recalling (5.34), we can conclude that for the method to be absolutely stable, we must satisfy the inequality

$$\left| \frac{1 - (1 - \theta) \lambda_h^i \Delta t}{1 + \theta \lambda_h^i \Delta t} \right| < 1,$$

that is

$$-1 - \theta \lambda_h^i \Delta t < 1 - (1 - \theta) \lambda_h^i \Delta t < 1 + \theta \lambda_h^i \Delta t.$$

Hence,

$$-\frac{2}{\lambda_h^i \Delta t} - \theta < \theta - 1 < \theta.$$

The second inequality is always verified, while the first one can be rewritten as

$$2\theta - 1 > -\frac{2}{\lambda_h^i \Delta t}.$$

If $\theta \geq 1/2$, the left-hand side is non-negative, while the right-hand side is negative, so the inequality holds for each Δt . Instead, if $\theta < 1/2$, the inequality is satisfied (hence the method is stable) only if

$$\Delta t < \frac{2}{(1 - 2\theta)\lambda_h^i}. \quad (5.35)$$

As such relation must hold for all the eigenvalues λ_h^i of the bilinear form, it will suffice to require that it holds for the maximum among them, which we have supposed to be $\lambda_h^{N_h}$. To summarize, we have:

- if $\theta \geq 1/2$, the θ -method is unconditionally stable, i.e. it is stable for each Δt ;
- if $\theta < 1/2$, the θ -method is stable only for

$$\Delta t \leq \frac{2}{(1 - 2\theta)\lambda_h^{N_h}}.$$

Thanks to the definition of eigenvalue (5.33) and to the continuity property of $a(\cdot, \cdot)$, we deduce

$$\lambda_h^{N_h} = \frac{a(w_{N_h}, w_{N_h})}{\|w_{N_h}\|_{L^2(\Omega)}^2} \leq \frac{M\|w_{N_h}\|_V^2}{\|w_{N_h}\|_{L^2(\Omega)}^2} \leq M(1 + C^2 h^{-2}).$$

The constant $C > 0$ which appears in the latter step derives from the following *inverse inequality*

$$\exists C > 0 : \|\nabla v_h\|_{L^2(\Omega)} \leq Ch^{-1} \|v_h\|_{L^2(\Omega)} \quad \forall v_h \in V_h,$$

for whose proof we refer to [QV94, Chap. 3].

Hence, for h small enough, $\lambda_h^{N_h} \leq Ch^{-2}$. In fact, we can prove that $\lambda_h^{N_h}$ is indeed of the order of h^{-2} , that is

$$\lambda_h^{N_h} = \max_i \lambda_h^i \simeq ch^{-2}.$$

Keeping this into account, we obtain that for $\theta < 1/2$ the method is absolutely stable only if

$$\Delta t \leq C(\theta)h^2, \quad (5.36)$$

where $C(\theta)$ denotes a positive constant depending on θ .

The latter relation implies that, for $\theta < 1/2$, Δt cannot be chosen arbitrarily but is bound to the choice of h .

5.5 Convergence analysis of the θ -method

We can prove the following convergence theorem

Theorem 5.1 *Under the hypothesis that u_0 , f and the exact solution are sufficiently regular, the following a priori error estimate holds: $\forall n \geq 1$,*

$$\|u(t^n) - u_h^n\|_{L^2(\Omega)}^2 + 2\alpha\Delta t \sum_{k=1}^n \|u(t^k) - u_h^k\|_V^2 \leq C(u_0, f, u)(\Delta t^{p(\theta)} + h^{2r}),$$

where $p(\theta) = 2$ if $\theta \neq 1/2$, $p(1/2) = 4$ and C depends on its arguments but not on h and Δt .

Proof. The proof is carried out by comparing the solution of the fully discretized problem (5.30) with that of the semi-discrete problem (5.6), using the stability result (5.32) as well as the decay rate of the truncation error of the time discretization. For simplicity, we will limit ourselves to considering the backward Euler method (corresponding to $\theta = 1$)

$$\frac{1}{\Delta t}(u_h^{k+1} - u_h^k, v_h) + a(u_h^{k+1}, v_h) = (f^{k+1}, v_h) \quad \forall v_h \in V_h. \quad (5.37)$$

We refer the reader to [QV94], Sec. 11.3.1, for the proof in the general case.

Let $\Pi_{1,h}^r$ be the orthogonal projector operator introduced in (5.26). Then

$$\|u(t^k) - u_h^k\|_{L^2(\Omega)} \leq \|u(t^k) - \Pi_{1,h}^r u(t^k)\|_{L^2(\Omega)} + \|\Pi_{1,h}^r u(t^k) - u_h^k\|_{L^2(\Omega)}. \quad (5.38)$$

The first term can be estimated by referring to (5.27). To analyze the second term, having set $\varepsilon_h^k = u_h^k - \Pi_{1,h}^r u(t^k)$, we obtain

$$\frac{1}{\Delta t}(\varepsilon_h^{k+1} - \varepsilon_h^k, v_h) + a(\varepsilon_h^{k+1}, v_h) = (\delta^{k+1}, v_h) \quad \forall v_h \in V_h, \quad (5.39)$$

having set, $\forall v_h \in V_h$,

$$(\delta^{k+1}, v_h) = (f^{k+1}, v_h) - \frac{1}{\Delta t}(\Pi_{1,h}^r(u(t^{k+1}) - u(t^k)), v_h) - a(u(t^{k+1}), v_h) \quad (5.40)$$

and having exploited on the last addendum the orthogonality (5.26) of the operator $\Pi_{1,h}^r$. The sequence $\{\varepsilon_h^k, k = 0, 1, \dots\}$ satisfies problem (5.39), which is similar in all to (5.37) (provided that we take δ^{k+1} instead of f^{k+1}). By adapting the stability estimate (5.32), we obtain, for each $n \geq 1$,

$$\|\varepsilon_h^n\|_{L^2(\Omega)}^2 + 2\alpha\Delta t \sum_{k=1}^n \|\varepsilon_h^k\|_V^2 \leq C(t^n) \left(\|\varepsilon_h^0\|_{L^2(\Omega)}^2 + \sum_{k=1}^n \Delta t \|\delta^k\|_{L^2(\Omega)}^2 \right). \quad (5.41)$$

The norm associated to the initial time-level can easily be estimated: for instance, if $u_{0h} = \Pi_h^r u_0$ is the finite element interpolant of u_0 , by suitably using the estimates (4.69) and (5.27) we obtain

$$\begin{aligned}\|\varepsilon_h^0\|_{L^2(\Omega)} &= \|u_{0h} - \Pi_{1,h}^r u_0\|_{L^2(\Omega)} \\ &\leq \|\Pi_h^r u_0 - u_0\|_{L^2(\Omega)} + \|u_0 - \Pi_{1,h}^r u_0\|_{L^2(\Omega)} \leq C h^r |u_0|_{H^r(\Omega)}.\end{aligned}\quad (5.42)$$

Let us now focus on estimating the norm $\|\delta^k\|_{L^2(\Omega)}$. We note that, thanks to (5.5),

$$(f^{k+1}, v_h) - a(u(t^{k+1}), v_h) = \left(\frac{\partial u(t^{k+1})}{\partial t}, v_h \right).$$

This allows us to rewrite (5.40) as

$$\begin{aligned}(\delta^{k+1}, v_h) &= \left(\frac{\partial u(t^{k+1})}{\partial t}, v_h \right) - \frac{1}{\Delta t} (\Pi_{1,h}^r(u(t^{k+1}) - u(t^k)), v_h) \\ &= \left(\frac{\partial u(t^{k+1})}{\partial t} - \frac{u(t^{k+1}) - u(t^k)}{\Delta t}, v_h \right) + \left((I - \Pi_{1,h}^r) \left(\frac{u(t^{k+1}) - u(t^k)}{\Delta t} \right), v_h \right).\end{aligned}\quad (5.43)$$

Using the Taylor formula with the remainder in integral form, we have

$$\frac{\partial u(t^{k+1})}{\partial t} - \frac{u(t^{k+1}) - u(t^k)}{\Delta t} = \frac{1}{\Delta t} \int_{t^k}^{t^{k+1}} (s - t^k) \frac{\partial^2 u}{\partial t^2}(s) ds,\quad (5.44)$$

having made the suitable regularity requirements on the u function with respect to the temporal variable. By now using the fundamental theorem of integration and exploiting the commutativity between the projection operator $\Pi_{1,h}^r$ and the temporal derivative, we obtain

$$(I - \Pi_{1,h}^r)(u(t^{k+1}) - u(t^k)) = \int_{t^k}^{t^{k+1}} (I - \Pi_{1,h}^r) \left(\frac{\partial u}{\partial t} \right)(s) ds.\quad (5.45)$$

By choosing $v_h = \delta^{k+1}$ in (5.43), thanks to (5.44) and (5.45), we can deduce the following upper bound

$$\begin{aligned}\|\delta^{k+1}\|_{L^2(\Omega)} &\leq \left\| \frac{1}{\Delta t} \int_{t^k}^{t^{k+1}} (s - t^k) \frac{\partial^2 u}{\partial t^2}(s) ds \right\|_{L^2(\Omega)} + \left\| \frac{1}{\Delta t} \int_{t^k}^{t^{k+1}} (I - \Pi_{1,h}^r) \left(\frac{\partial u}{\partial t} \right)(s) ds \right\|_{L^2(\Omega)} \\ &\leq \int_{t^k}^{t^{k+1}} \left\| \frac{\partial^2 u}{\partial t^2}(s) \right\|_{L^2(\Omega)} ds + \frac{1}{\Delta t} \int_{t^k}^{t^{k+1}} \left\| (I - \Pi_{1,h}^r) \left(\frac{\partial u}{\partial t} \right)(s) \right\|_{L^2(\Omega)} ds.\end{aligned}\quad (5.46)$$

By reverting to the stability estimate (5.41) and exploiting (5.42) and the estimate (5.46) with suitably scaled indexes, we have

$$\begin{aligned} \|\varepsilon_h^n\|_{L^2(\Omega)}^2 &\leq C(t^n) \left(h^{2r} |u_0|_{H^r(\Omega)}^2 + \sum_{k=1}^n \Delta t \left[\left(\int_{t^{k-1}}^{t^k} \left\| \frac{\partial^2 u}{\partial t^2}(s) \right\|_{L^2(\Omega)} ds \right)^2 \right. \right. \\ &+ \left. \left. \frac{1}{\Delta t^2} \left(\int_{t^{k-1}}^{t^k} \left\| \left(I - \Pi_{1,h}^r \right) \left(\frac{\partial u}{\partial t} \right)(s) \right\|_{L^2(\Omega)} ds \right)^2 \right] \right). \end{aligned}$$

Then, using the Cauchy-Schwarz inequality and the estimate (5.27) for the projection operator $\Pi_{1,h}^r$, we obtain

$$\begin{aligned} \|\varepsilon_h^n\|_{L^2(\Omega)}^2 &\leq C(t^n) \left(h^{2r} |u_0|_{H^r(\Omega)}^2 + \sum_{k=1}^n \Delta t \left[\Delta t \int_{t^{k-1}}^{t^k} \left\| \frac{\partial^2 u}{\partial t^2}(s) \right\|_{L^2(\Omega)}^2 ds \right. \right. \\ &+ \left. \left. \frac{1}{\Delta t^2} \left(\int_{t^{k-1}}^{t^k} h^r \left| \frac{\partial u}{\partial t}(s) \right|_{H^r(\Omega)} ds \right)^2 \right] \right) \\ &\leq C(t^n) \left(h^{2r} |u_0|_{H^r(\Omega)}^2 + \Delta t^2 \sum_{k=1}^n \int_{t^{k-1}}^{t^k} \left\| \frac{\partial^2 u}{\partial t^2}(s) \right\|_{L^2(\Omega)}^2 ds \right. \\ &+ \left. \frac{1}{\Delta t} h^{2r} \sum_{k=1}^n \Delta t \int_{t^{k-1}}^{t^k} \left| \frac{\partial u}{\partial t}(s) \right|_{H^r(\Omega)}^2 ds \right). \end{aligned}$$

The result now follows using (5.38) and the estimate (5.27). \diamond

More stability and convergence estimates can be found in [Tho84].

5.6 Exercises

1. Verify that the mass matrix M introduced in (5.7) is positive definite.
2. Consider the problem:

$$\left\{ \begin{array}{ll} \frac{\partial u}{\partial t} - \frac{\partial}{\partial x} \left(\alpha \frac{\partial u}{\partial x} \right) - \beta u = 0 & \text{in } Q_T = (0, 1) \times (0, \infty), \\ u = u_0 & \text{for } x \in (0, 1), t = 0, \\ u = \eta & \text{for } x = 0, t > 0, \\ \alpha \frac{\partial u}{\partial x} + \gamma u = 0 & \text{for } x = 1, t > 0, \end{array} \right.$$

where $\alpha = \alpha(x)$, $u_0 = u_0(x)$ are given functions and $\beta, \gamma, \eta \in \mathbb{R}$ (with positive β).

- Prove existence and uniqueness of the weak solution for varying γ , providing suitable limitations on the coefficients and suitable regularity hypotheses on the functions α and u_0 .
 - Introduce the spatial semi-discretization of the problem using the Galerkin-finite element method, and carry out its stability and convergence analysis.
 - In the case where $\gamma = 0$, approximate the same problem with the explicit Euler method in time and carry out its stability analysis.
3. Consider the following problem: find $u(x, t)$, $0 \leq x \leq 1$, $t \geq 0$, such that

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial v}{\partial x} = 0, & 0 < x < 1, t > 0, \\ v + \alpha(x) \frac{\partial u}{\partial x} - \gamma(x)u = 0, & 0 < x < 1, t > 0, \\ v(1, t) = \beta(t), u(0, t) = 0, & t > 0, \\ u(x, 0) = u_0(x), & 0 < x < 1, \end{cases}$$

where $\alpha, \gamma, \beta, u_0$ are given functions.

- Introduce an approximation based on finite elements of degree two in x and the implicit Euler method in time and prove its stability.
 - How will the error behave as a function of the h and Δt parameters?
 - Suggest a way to provide an approximation for v starting from the one for u as well as its approximation error.
4. Consider the following (diffusion-transport-reaction) initial-boundary value problem: find $u : (0, 1) \times (0, T) \rightarrow \mathbb{R}$ such that

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial}{\partial x} \left(\alpha \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial x} (\beta u) + \gamma u = 0, & 0 < x < 1, 0 < t < T, \\ u = 0 & \text{for } x = 0, 0 < t < T, \\ \alpha \frac{\partial u}{\partial x} + \delta u = 0 & \text{for } x = 1, 0 < t < T, \\ u(x, 0) = u_0(x), & 0 < x < 1, t = 0, \end{cases}$$

where $\alpha = \alpha(x)$, $\beta = \beta(x)$, $\gamma = \gamma(x)$, $\delta = \delta(x)$, $u_0 = u_0(x)$, $x \in [0, 1]$ are given functions.

- Write its weak formulation.
- In addition to the hypotheses:

- $\exists \beta_0, \alpha_0, \alpha_1 > 0 : \forall x \in (0, 1) \alpha_1 \geq \alpha(x) \geq \alpha_0, \beta(x) \leq \beta_0,$
- $\frac{1}{2}\beta'(x) + \gamma(x) \geq 0 \quad \forall x \in (0, 1),$

provide possible further hypotheses on the data so that the problem is well-posed. Moreover, give an a priori estimate of the solution. Treat the same

problem with non-homogeneous Dirichlet data $u = g$ for $x = 0$ and $0 < t < T$.

- c) Consider a semi-discretization based on the linear finite elements method and prove its stability.
 - d) Finally, provide a full discretization where the temporal derivative is approximated using the implicit Euler scheme and prove its stability.
5. Consider the following fourth-order initial-boundary value problem:

find $u : \Omega \times (0, T) \rightarrow \mathbb{R}$ such that

$$\begin{cases} \frac{\partial u}{\partial t} - \operatorname{div}(\mu \nabla u) + \Delta^2 u + \sigma u = 0 & \text{in } \Omega \times (0, T), \\ u(\mathbf{x}, 0) = u_0 & \text{in } \Omega, \\ \frac{\partial u}{\partial n} = u = 0 & \text{on } \Sigma_T = \partial\Omega \times (0, T), \end{cases}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded open domain with "regular" boundary $\partial\Omega$, $\Delta^2 = \Delta\Delta$ is the bi-harmonic operator, $\mu(\mathbf{x})$, $\sigma(\mathbf{x})$ and $u_0(\mathbf{x})$ are known functions defined in Ω . It is known that

$$\sqrt{\int_{\Omega} |\Delta u|^2 d\Omega} \simeq \|u\|_{H^2(\Omega)} \quad \forall u \in H_0^2(\Omega),$$

that is the two norms are equivalent, where

$$H_0^2(\Omega) = \{u \in H^2(\Omega) : u = \partial u / \partial n = 0 \text{ on } \partial\Omega\}. \quad (5.47)$$

- a) Write its weak formulation and verify that the solution exists and is unique, formulating suitable regularity hypotheses on the data.
- b) Consider a semi-discretization based on the triangular finite elements and provide the minimum degree that such elements must have in order to adequately solve the given problem. (We note that, if \mathcal{T}_h is a triangulation of Ω and $v_h|_K$ is a polynomial for each $K \in \mathcal{T}_h$, then $v_h \in H^2(\Omega)$ if and only if $v_h \in C^1(\overline{\Omega})$, that is v_h and its first derivatives are continuous across the interfaces of the elements of \mathcal{T}_h .)

6

Generation of 1D and 2D grids

As we have seen, the finite element method for the solution of partial differential equations requires a “triangulation” of the computational domain, i.e. a partition of the domain in simpler geometric entities (for instance, triangles or quadrangles in two dimensions, tetrahedra, prisms or hexahedra in three dimensions), called the elements, which verify a number of conditions. Similar partitions stand at the base of other approximation methods, such as the finite volume method (see Chap. 9) and the spectral element method (see Chap. 10). The set of all elements is the so-called computational grid (or, simply, grid, or mesh).

In this chapter, for simplicity, we focus on the main partitioning techniques for one- and two-dimensional domains, with no ambition of completeness. When applicable, we will refer the reader to the relevant specialized literature. We will deal only with the case of polygonal domains; for computational domains with curved boundaries, the interested reader can consult [Cia78], [BS94], [GB98]. The techniques exposed for the 2D case can be extended to three-dimensional domains.

6.1 Grid generation in 1D

Suppose that the computational domain Ω be an interval (a, b) . The most elementary partition in sub-intervals is the one where the step h is constant. Having chosen the number of elements, say N , we pose $h = \frac{b-a}{N}$ and introduce the points $x_i = x_0 + ih$, with $x_0 = a$ and $i = 0, \dots, N$. Such points $\{x_i\}$ are called “vertices” to align with the two-dimensional case where they will actually be the vertices of the triangles whose union covers the domain Ω . The partition thus obtained is called grid. The latter is *uniform* as it is composed by elements of the same length.

In the more general case, we will use non-uniform grids, possibly generated according to a given law. Among the different possible procedures, we illustrate a fairly general one. Let a strictly positive function $\mathcal{H} : [a, b] \rightarrow \mathbb{R}^+$, called *spacing function*, be assigned and let us consider the problem of generating a partition of the interval $[a, b]$ having $N + 1$ vertices x_i . The value $\mathcal{H}(x)$ represents the desired

spacing in correspondence of the point x . For instance, if $\mathcal{H} = h$ (constant), with $h = (b - a)/M$ for a given integer M , we fall exactly in the preceding case of the uniform grid, with $N = M$. More generally, we compute $\mathcal{N} = \int_a^b \mathcal{H}^{-1}(x) dx$ and we set $N = \max(1, [\mathcal{N}])$, where $[\mathcal{N}]$ denotes the integral part of \mathcal{N} , i.e. the largest positive integer smaller than or equal to \mathcal{N} . Note that the resulting grid will have at least one element. We then set $\kappa = \frac{N}{\mathcal{N}}$ and look for the points x_i such that

$$\kappa \int_a^{x_i} \mathcal{H}^{-1}(x) dx = i,$$

for $i = 0, \dots, N$. The κ constant is a positive correction factor, with a value as close as possible to 1, whose purpose is to guarantee that N is indeed an integer. Indeed, we point out that, for a given \mathcal{H} , the number N of elements is itself an unknown of the problem. Instead, the \mathcal{H}^{-1} function defines a density function: to higher values of \mathcal{H}^{-1} correspond more dense nodes, and conversely, to smaller values of \mathcal{H}^{-1} correspond less frequent nodes.

Obviously, if we wish to construct a grid with a prefixed number N of elements, as well as a given variation on $[a, b]$, it would be sufficient to renormalize the spacing function so that the integral on (a, b) of the corresponding density is exactly equal to N . In any case, to compute the points x_i , it is useful to introduce the following Cauchy problem

$$y'(x) = \kappa \mathcal{H}^{-1}(x), \quad x \in (a, b), \quad \text{with } y(a) = 0.$$

The x_i points will then be defined by the relation $y(x_i) = i$, for $i = 1, \dots, N - 1$. Then, it will automatically be guaranteed that $x_0 = a$ and $x_N = b$. We will then be enabled to use a numerical solution method to find the roots of the $f_j(x) = y(x) - j$ functions, for each value of $j \in \{1, \dots, N - 1\}$ (see e.g. [QSS07]).

Besides being quite generic, this procedure can be easily extended to the generation of vertices on the curved boundary of a two-dimensional domain, as we will see in Sec. 6.4.2.

In the case where \mathcal{H} does not exhibit excessive variations in the interval (a, b) , we can also use a simplified procedure which consists in computing a set of preliminary points \tilde{x}_i , for $i = 0, \dots, N$, defined as follows:

1. Set $\tilde{x}_0 = a$ and define $\tilde{x}_i = \tilde{x}_{i-1} + \mathcal{H}(\tilde{x}_{i-1})$, $i = 1, 2, \dots$, until finding the value M such that $\tilde{x}_M \geq b$ and $\tilde{x}_{M-1} < b$;
2. if $\tilde{x}_M - b \leq b - \tilde{x}_{M-1}$ set $N = M$, otherwise define $N = M - 1$.

Then the final set of vertices are obtained by setting

$$x_i = x_{i-1} + k \mathcal{H}(\tilde{x}_{i-1}), \quad i = 1, \dots, N,$$

with $x_0 = a$ and $k = (b - x_{N-1})/(x_N - x_{N-1})$.

The MATLAB program **mesh_1d** allows to construct a grid on an interval with endpoints **a** and **b** with step specified in the macro **H**, using the previous simplified algorithm. For instance, with the following MATLAB commands:

```
a = 0; b = 1; H = '0.1';
coord = mesh_1d(a,b,H);
```

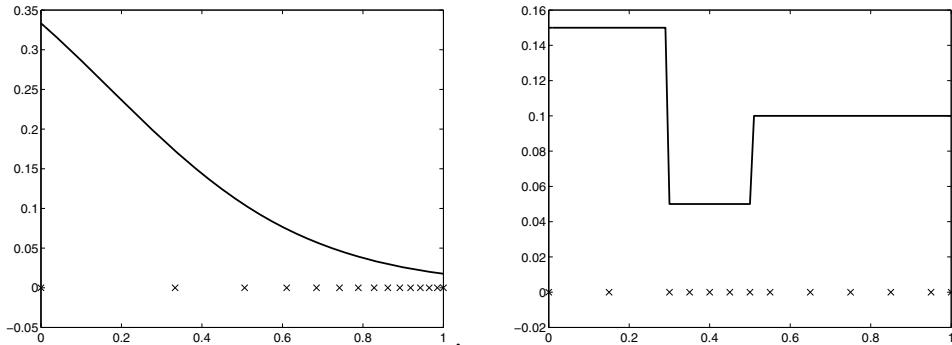


Fig. 6.1. At the left-hand side, the behavior of the grid step (on the x-axis) associated to the function $H = '1 / (\exp(4*x) + 2)'$, at the right-hand side the one relating to the function $H = '.1 * (x < 3) + .05 * (x > 5) + .05'$. The graph also reports the corresponding vertex distributions

we create a uniform grid on $[0, 1]$ with 10 sub-intervals with step $h = 0.1$.

Setting $H = '1 / (\exp(4*x) + 2)'$ we obtain a grid that intensifies when approaching the second extremum of the interval, while for $H = '.1 * (x < .3) + .05 * (x > .5) + .05'$ we obtain a grid with a discontinuously varying step (see Fig. 6.1).

Program 1 - mesh_1d: Constructs a one-dimensional grid on an interval $[a, b]$ following the spacing function H

```
function coord = mesh_1d(a,b,H)

coord = a;
while coord(end) < b
    x = coord(end);
    xnew = x + eval(H);
    coord = [coord, xnew];
end
if (coord(end) - b) > (b - coord(end-1))
    coord = coord(1:end-1);
end
coord_old = coord;
kappa = (b - coord(end-1))/(coord(end) - coord(end-1));
coord = a;
for i = 1:length(coord_old)-1
    x = coord_old(i);
    coord(i+1) = x + kappa*eval(H);
end
```

We point out that in case \mathcal{H} is determined by an error estimate, Program 1 will allow to perform grid adaptivity.

We now face the problem of constructing the grid for two-dimensional domains.

6.2 Grid of a polygonal domain

Given a bounded polygonal domain Ω in \mathbb{R}^2 , we can associate it with a grid (or partition) \mathcal{T}_h of Ω in polygons K such that

$$\overline{\Omega} = \bigcup_{K \in \mathcal{T}_h} K,$$

where $\overline{\Omega}$ is the closure of Ω , and

- $\overset{\circ}{K} \neq \emptyset \forall K \in \mathcal{T}_h$;
- $\overset{\circ}{K_1} \cap \overset{\circ}{K_2} = \emptyset$ for each $K_1, K_2 \in \mathcal{T}_h$ s.t. $K_1 \neq K_2$;
- if $F = K_1 \cap K_2 \neq \emptyset$ with $K_1, K_2 \in \mathcal{T}_h$ and $K_1 \neq K_2$, then F is either a whole edge or a vertex of the grid;
- having denoted by h_K the diameter of K for each $K \in \mathcal{T}_h$, we define $h = \max_{K \in \mathcal{T}_h} h_K$.

We have denoted by $\overset{\circ}{K} = K \setminus \partial K$ the interior of K . The grid \mathcal{T}_h is also called *mesh*, or sometimes *triangulation* (in a broad sense) of $\overline{\Omega}$.

The constraints imposed on the grid by the first two conditions are obvious: in particular, the second one requires that given two distinct elements, their interiors do not overlap. The third condition limits the admissible triangulations to the so-called *conforming* ones. To illustrate the concept, we report in Fig. 6.2 a conforming (left) and nonconforming (right) triangulation. In the remainder, we will only consider conforming triangulations. However, there exist very specific finite element approximations, not considered in the present book, which use nonconforming grids, i.e. grids that do not satisfy the third condition. These methods are therefore more flexible, at least as far as the choice of the computational grid is concerned, allowing, among other things, the coupling of grids constructed from elements of different nature, for instance triangles and quadrilaterals. The fourth condition links the parameter h to the maximal diameter of the elements of \mathcal{T}_h .

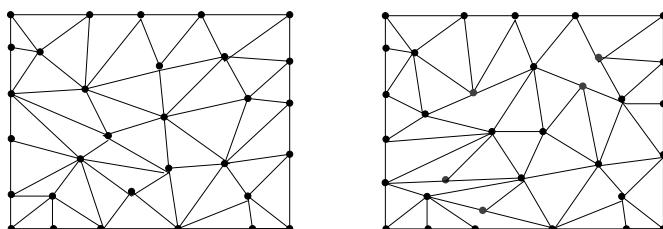


Fig. 6.2. Example of conforming (left) and nonconforming (right) grid

For reasons linked to the interpolation error theory recalled in Chap. 4, we will only consider *regular* triangulations \mathcal{T}_h , i.e. the ones for which, for each element $K \in \mathcal{T}_h$, the ratio between the diameter h_K and the sphericity ρ_K (i.e. the diameter of the inscribed circle) is less than a given constant. More precisely, the grids satisfy Property (4.37). Fig. 6.3 illustrates the meaning of diameter and sphericity for a triangular or quadrilateral element.

In actual applications, it is customary to distinguish between *structured* and *unstructured* grids. Structured grids basically use quadrangular elements and are characterized by the fact that access to the vertices adjacent to a given node (or to the elements adjacent to a given element) is immediate. Indeed, it is possible to establish a bijective relation between the vertices of the grid and the pairs of integer numbers (i, j) , $i = 1, \dots, N_i$, $j = 1, \dots, N_j$ such that, given the node of indices (i, j) , the four adjacent vertices are in correspondence of the indices $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$ and $(i, j + 1)$. The total number of vertices is therefore $N_i N_j$. An analogous association can be established between the elements of the grid and the pairs (I, J) , $I = 1, \dots, N_i - 1$, $J = 1, \dots, N_j - 1$. Moreover, it is possible to directly identify the vertices corresponding to each element, without having to explicitly memorize the connectivity matrix (the latter is the matrix which, for each element, provides its vertex numbering). Fig. 6.4 (left) illustrates such situation.

In a computer code, pairs of indices are typically replaced by a numbering formed by a single integer that is biunivocally associated to the indices described above. For instance, for the numbering of vertices, we can choose to associate the integer number $k = i + (j - 1)N_i$ to each pair (i, j) , and, conversely, we univocally associate to the vertex k the indices $i = ((k - 1) \bmod N_i) + 1$ and $j = ((k - 1) \text{ div } N_i) + 1$, where \bmod and div denote the remainder and the quotient of the integer division.

In unstructured grids, the association between an element of the grid and its vertices must instead be explicitly stored in the connectivity matrix.

Code developed for structured grids can benefit from the “structure” of the grid, and, for an equal number of elements, it will normally produce a more efficient algorithm, both in terms of memory and in terms of computational time, with respect to a similar scheme on a non-structured grid. In contrast, non-structured grids offer a greater flexibility both from the viewpoint of a triangulation of domains of complex shape and for the possibility to locally refine/derefine the grid. Fig. 6.4 (right) shows

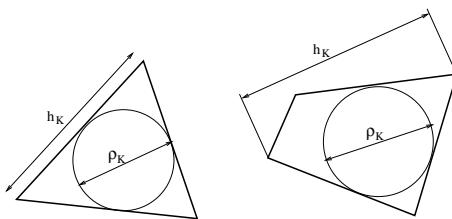


Fig. 6.3. Diameter and sphericity for a triangular (left) and quadrilateral element (right)

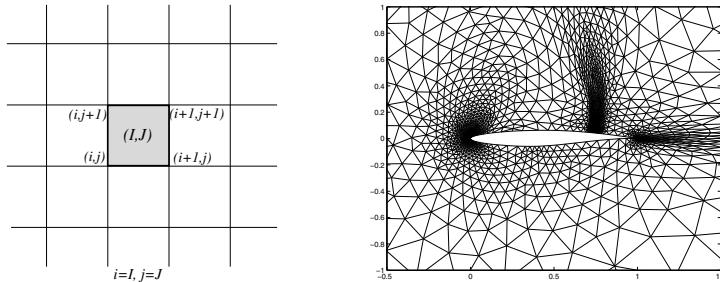


Fig. 6.4. (Left) Placement of the numbering of the vertices belonging to an element with indices (I, J) in a structured grid. (Right) A non-structured triangular grid in an external region to an airfoil, adapted to improve the accuracy of the numerical solution for a given flow condition

an example of a non-structured grid whose spacing has been adapted to the specific problem under exam. Such localized refinements would be difficult, or impossible, to obtain using a structured type of grid (unless we use nonconforming grids).

Non-structured two-dimensional grids are generally formed by triangles, although it is possible to have quadrangular non-structured grids.

6.3 Generation of structured grids

The most elementary idea to generate a structured grid on a domain Ω of arbitrary shape consists in finding a regular and invertible map \mathcal{M} between the square $\widehat{\Omega} = [0, 1] \times [0, 1]$ (which we will call reference square) and $\overline{\Omega}$. Note that the map must be regular up to the boundary (a requirement that can in some cases be relaxed). We then proceed by generating a reticulation, uniform for instance, in the reference square, then we use the mapping \mathcal{M} to transform the coordinates of the vertices in $\widehat{\Omega}$ into the corresponding ones in $\overline{\Omega}$.

There are different aspects of this procedure to be considered with due care.

1. The search for the \mathcal{M} map is often difficult. Moreover, such map is not unique. In general it is preferable that the latter be as regular as possible.
2. A uniform mesh of the reference square does not generally provide an optimal grid in Ω . Indeed, we usually want to control the distribution of vertices in Ω , and generally this can only be done by generating non-uniform grids on the reference square, whose spacing will depend both on the desired spacing in Ω and on the chosen map \mathcal{M} .
3. Even if the mapping is regular (for instance of class C^1) it is not guaranteed that the elements of the grid produced in Ω are admissible, as the latter are not the image through \mathcal{M} of the corresponding elements in $\widehat{\Omega}$. For instance, if we desire piecewise bilinear (\mathbb{Q}_1) finite elements in Ω , the edges of the latter will need to be parallel to the cartesian axes, while the image of a mesh \mathbb{Q}_1 on the reference

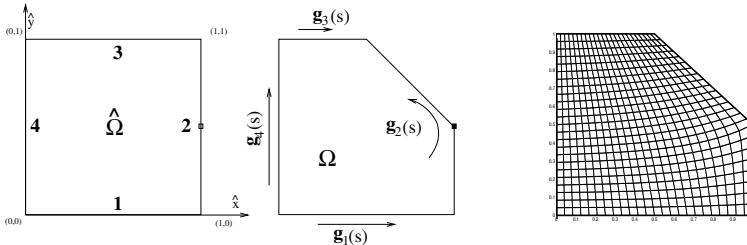


Fig. 6.5. Construction of a structured grid: at the left-hand side, identification of the map on the boundary; at the right-hand side, grid corresponding to a uniform partitioning of the reference square into 24×24 elements

square produces curve edges in Ω if the mapping is non-linear. In other words, the map is made effective only on the vertices, not on the edges, of the grid of $\hat{\Omega}$.

An option to construct the map \mathcal{M} consists in using the transfinite interpolation (10.3) that will be illustrated in Chap. 10. Such methodology is however not always easily applicable. We will therefore illustrate in the remainder a more general methodology, which we will apply to a specific example, and refer to the specific literature [TWM85, TSW99] for further examples and details.

Suppose we have a domain Ω whose boundary can be divided in four consecutive parts $\Gamma_1, \dots, \Gamma_4$, as illustrated in Fig. 6.5 for a particularly simple domain. Moreover, suppose we can describe such portions of $\partial\Omega$ via four parametric curves g_1, \dots, g_4 oriented as in the figure, where the parameter s varies between 0 and 1 on each curve. This construction allows us to create a bijective map between the sides of the reference square and the domain boundary. Indeed, we will associate each curve to the corresponding side of the square, as exemplified in Fig. 6.5. We now need to understand how to extend the mapping to the whole $\hat{\Omega}$.

Remark 6.1 Note that the curves g_i $i = 1, \dots, 4$ are generally not differentiable in all of $(0, 1)$, but can exhibit a finite number of “edges” where $\frac{dg_i}{ds}$ is undefined. In Fig. 6.5, for instance, the curve g_2 is not differentiable at the “edge” marked by a small black square. •

An option to construct the map $\mathcal{M} : (\hat{x}, \hat{y}) \mapsto (x, y)$ consists in solving the following elliptic system in $\hat{\Omega}$:

$$-\frac{\partial^2 \mathbf{x}}{\partial \hat{x}^2} - \frac{\partial^2 \mathbf{x}}{\partial \hat{y}^2} = 0 \quad \text{in } \hat{\Omega} = (0, 1)^2, \quad (6.1)$$

with boundary conditions

$$\begin{aligned} \mathbf{x}(\hat{x}, 0) &= \mathbf{g}_1(\hat{x}), \quad \mathbf{x}(\hat{x}, 1) = \mathbf{g}_3(\hat{x}), \quad \hat{x} \in (0, 1), \\ \mathbf{x}(1, \hat{y}) &= \mathbf{g}_2(\hat{y}), \quad \mathbf{x}(0, \hat{y}) = \mathbf{g}_4(\hat{y}), \quad \hat{y} \in (0, 1). \end{aligned}$$

The vertices of a grid in the reference square can then be transformed into the vertices of a grid in Ω . Note that the solution of problem (6.1) will generally be found by using a numerical method, for instance via a finite difference (or finite element) scheme. Moreover, to suitably abide to the geometry of the boundary of Ω , it is necessary to ensure that a vertex is generated at each “edge”. In Fig. 6.5 (right) we illustrate the result of the application of this methodology to the domain in Fig. 6.5 (left). It can be noted that the grid corresponding to a regular partition of the reference square is not particularly satisfactory if, for instance, we want to have a higher distribution of vertices at the edge.

Moreover, the methodology described above is not applicable to non-convex domains. Indeed, let us consider Fig. 6.6 where we show an L-shaped domain, with the corresponding boundary partition, and the grid obtained by solving problem (6.1) starting from a regular partition of the reference domain. It is evident that such grid is unacceptable.

To solve such problems, we can proceed in several (not mutually exclusive) ways:

- we use in $\widehat{\Omega}$ a non-uniform grid, that accounts for the geometric features of Ω ;
- we use a different map \mathcal{M} , obtained, for instance, by solving the following new differential problem instead of (6.1)

$$-\alpha \frac{\partial^2 \mathbf{x}}{\partial \widehat{x}^2} - \beta \frac{\partial^2 \mathbf{x}}{\partial \widehat{y}^2} + \gamma \mathbf{x} = \mathbf{f} \quad \text{in } \widehat{\Omega}, \quad (6.2)$$

where $\alpha > 0$, $\beta > 0$, $\gamma \geq 0$ and \mathbf{f} are suitable functions of \widehat{x} and \widehat{y} . Their choice will depend on the geometry of Ω and will be made in order to control the vertex distribution;

- we partition Ω in sub-domains that are triangulated separately. This technique is normally known as *blockwise structured grid generation*. If we wish the global grid to be conforming, we need to exert special care as to how to distribute the number of vertices on the boundaries of the interfaces between the different sub-domains. The problem can become extremely complex when the number of sub-domains is very large.

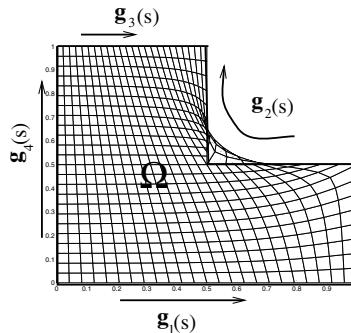


Fig. 6.6. Triangulation of a non-convex domain. Identification of the boundary map and mesh obtained by solving the elliptic problem (6.1)

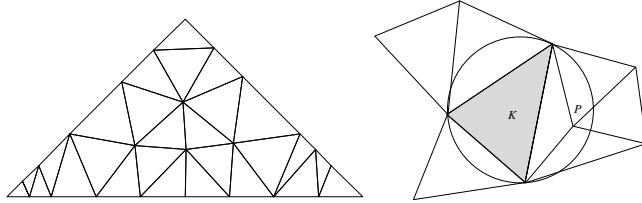


Fig. 6.7. At the left-hand side, an example of Delaunay grid on a triangular shaped convex domain. It can easily be verified that the circumscribed circle to each triangle does not include any vertex of the grid. At the right-hand side, a detail of a grid which does not satisfy the Delaunay condition: indeed, the vertex P falls inside the circle circumscribed to the triangle K

Methods of the type illustrated above are called *elliptic schemes of grid generation*, as they are based on the solution of elliptic equations, such as (6.1) and (6.2).

The interested reader is referred to the above-cited specialized literature.

6.4 Generation of non-structured grids

We will here consider the generation of non-structured grids with triangular elements. The two main algorithms used for this purpose are the *Delaunay triangulation* and the *advancing front* technique.

6.4.1 Delaunay triangulation

A triangulation of a set of n points of \mathbb{R}^2 is a Delaunay triangulation if the circumscribed circle to each triangle contains no vertex in its inside (see Fig. 6.7).

A Delaunay triangulation features the following properties:

1. given a set of points, the Delaunay triangulation is unique, except for specific situations where M points (with $M > 3$) lie on a circumference;
2. among all possible triangulations, the Delaunay triangulation is the one maximizing the minimum angle of the grid triangles (this is called the max-min regularity property);
3. the set composed by the union of triangles is the convex figure of minimum surface that encloses the given set of points (and is called convex hull).

The third property makes the Delaunay algorithm inapplicable to non-convex domains, at least in its original form.

However, there exists a variant, called *Constrained Delaunay Triangulation (CDT)*, that allows to fix *a priori* a set of the grid edges to generate: the resulting grid necessarily associates such edges to some triangle. In particular, we can *a priori* impose those edges which define the boundary of the grid.

In order to better specify the concept of CDT, we state beforehand the following definition: given two points P_1 and P_2 , we will say that these are reciprocally *visible* if

the segment P_1P_2 passes through none of the boundary sides (or, more generally, the edges we want to fix a priori). A constrained Delaunay triangulation satisfies the following property: the interior of the circumscribed circle to each triangle K contains no vertex visible from an internal point to K .

It can again be proven that such triangulation is unique and satisfies the max-min regularity property. The CDT is therefore not a proper Delaunay triangulation, as some of its triangles could contain vertices belonging to the initial set. In any case, the vertices are only the original ones specified in the set, and no further vertices are added. However, two variants are possible: the *Conforming Delaunay Triangulation* and the *Conforming Constrained Delaunay Triangulation* (or CCDT). The former is a triangulation where each triangle is a Delaunay triangulation but each edge to fix can be further subdivided in sub-segments; in this case, new vertices can be added to obtained shorter segments. The additional vertices are often necessary to guarantee the satisfaction of the max-min Delaunay property and at the same time to ensure that each prescribed side is correctly represented. The second variant represents a triangulation where the triangles are of the constrained Delaunay type. Also in this case, we can add additional vertices, and the edges to be fixed cannot be divided in smaller segments. In the latter case however, the aim is not to guarantee that the edges are preserved, but to improve the triangles quality.

Among the available software for the generation of Delaunay grids, or their variants, `Triangle` [She] allows to generate Delaunay triangulations, with the option to modulate the regularity of the resulting grids in terms of maximal and minimal angles of the triangles. The geometry is passed as an input to `Triangle` in the form of a graph, called Planar Straight Line Graph (PSLG). Such codification is written in an input file with extension `.poly`: the latter basically contains a list of vertices and edges, but can also include information on cavities or concavities present in the geometry.

A sample `.poly` file is reported below.

```
# A box with eight vertices in 2D, no attribute, one boundary marker
8 2 0 1
# Vertices of the external box
1 0 0 0
2 0 3 0
3 3 0 0
4 3 3 0
# Vertices of the internal box
5 1 1 0
6 1 2 0
7 2 1 0
8 2 2 0
# Five sides with a boundary marker
5 1
1 1 2 5 # Left side of the external box
# Sides of the square cavity

2 5 7 0
3 7 8 0
```

```

4 8 6 10
5 6 5 0
# One hole in the center of the internal box
1
1 1.5 1.5

```

The example above illustrates a geometry representing a square with a square hole in its inside. The first part of the file lists the vertices, while the second one defines the sides to fix. The first line declares that eight vertices are going to follow, that the spatial dimension of the grid is two (we are in \mathbb{R}^2), that no other attribute is associated to the vertices and that a boundary marker is defined on each point. The attributes represent possible physical properties relating to the mesh nodes, such as e.g. conductibility and viscosity values, etc. The boundary markers are integer valued flags which can be used within a computational code to assign suitable boundary conditions at different vertices. The following lines report the eight vertices, with their abscissae and ordinates, followed by the boundary marker value, zero in this case. The first line of the second part declares that the following sides will be five and that on each of them a value will be specified for the boundary marker. Then, five boundary sides follow one another, specified based on the extreme vertices of each, and on the value of the boundary marker. In the final section of the file, a hole is defined by specifying the center coordinates, in the last line, preceded by the progressive numbering (in this case, limited to 1) of the holes.

The constrained Delaunay grid associated to this geometry, say `box.poly`, is obtained via the command

```
triangle -pc box
```

The `-p` parameter declares that the input file is a `.poly`, while option `-c` prevents the remotion of the concavities, which would otherwise be automatically removed. De facto, this option forces the triangulation of the convex hull of the PSLG graph. The result will be the creation of three files, `box.1.poly`, `box.1.node` and `box.1.ele`. The first file contains the description of the sides of the produced triangulation, the second one contains the node description, and the latter defines the connectivity of the generated elements. For the sake of conciseness, we will not describe the format of these three files in detail. Finally, we point out that the numerical value, 1 in this example, that separates the name of these three files from their respective extensions, plays the role of an iteration counter: Triangle can indeed successively refine or modify the triangulations produced time after time. The resulting triangulation is depicted in Fig. 6.8. A software attached to Triangle, called Show Me, allows to visualize the outputs of Triangle. For instance, Fig 6.8 (left) is obtained via the command

```
showme box
```

To obtain a constrained conforming triangulation we must specify the command `triangle` with other parameters, such as `-q`, `-a` or `-u`. The first one imposes a constraint on the minimum angle, the second one fixes a maximum value for the surface of the triangles, while the third one forces the dimension of the triangles, typically through an external function which the user must provide. For example, via the command

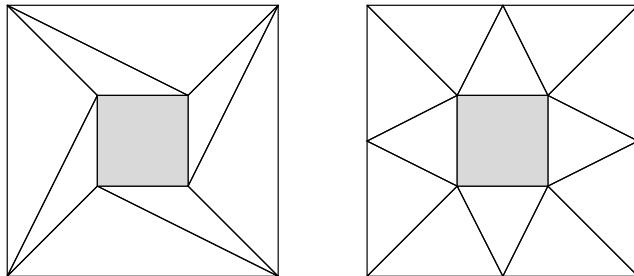


Fig. 6.8. Delaunay triangulation of a square with a square hole: CDT at the left-hand side, CCDT at the right-hand side

`triangle -pcq20 box`

we obtain the constrained conforming Delaunay triangulation reported in Fig. 6.8 (right), characterized by a minimum angle of 20° . Finally, the conforming Delaunay triangulation is obtained by further specifying the option `-D`. A more complex example is represented in Fig. 6.9. The command used

`triangle -pca0.001q30 mox`

fixes the minimum angle to 30° and the maximum surface of the generated triangles to 0.001. The initial PSLG file `mox.poly` describes the geometry via 474 vertices, as many sides and one cavity. The final mesh consists of 1595 vertices, 2708 elements and 4303 total sides, 482 of which lie on the boundary.

We refer to the wide on-line documentation and to the detailed help of `Triangle` for several further usage options of the software.

Returning to the properties of Delaunay grids, the Delaunay triangulation does not allow to control the aspect ratio (maximum over minimum edge) of the generated elements, exactly because of the above-mentioned max-min property. On the other hand, in some situations, it can be useful to generate “stretched” triangles in a given direction, for instance to well represent a boundary layer. To this end, the algorithm called *generalized Delaunay triangulation* has been developed, where the condition



Fig. 6.9. Delaunay triangulation of a complex shaped domain

on the circumscribed triangle is replaced by an analogous condition on the ellipse circumscribed to the triangle under exam. In this way, by suitably ruling the length and directions of the axes of each ellipse, we can generate elements stretched in the desired direction.

The most currently used algorithms for the generation of Delaunay grids are incremental, i.e. they generate a sequence of Delaunay grids by adding a vertex at a time. Hence, it is necessary to find procedures providing the new vertices in accordance with the desired grid spacing, succeeding in stopping such procedure as soon as the grid generated this way results to be unsatisfactory. For further details, [GB98] and [TSW99, Chap. 16] can be consulted, among others. A detailed description of the geometric properties of the constrained Delaunay triangulation, both for domains of \mathbb{R}^2 and of \mathbb{R}^3 , can be found in [BE92].

6.4.2 Advancing front technique

We coarsely describe another widely used technique used for the generation of non-structured grids, the *advancing front* technique. A necessary ingredient is the knowledge of the desired spacing to be generated for the grid elements. Let us then suppose that a spacing function \mathcal{H} , defined on $\overline{\Omega}$, provides for each point P of $\overline{\Omega}$ the dimensions of the grid desired in that point, for instance, through the diameter h_K of the elements that must be generated in a neighborhood of P . If we wanted to also control the shape aspect of the generated elements, \mathcal{H} will have a more complex shape. In fact, it will be a positive definite symmetric tensor, i.e. $\mathcal{H} : \Omega \rightarrow \mathbb{R}^{2 \times 2}$ such that, for each point P of the domain, the (perpendicular) eigenvectors of \mathcal{H} denote the direction of maximum and minimum stretching of the triangles that will need to be generated in the neighborhood of P , while the eigenvalues (more precisely, the square roots of the eigenvalue inverses), characterize the two corresponding spacings (see [GB98]). In the remainder, we will only consider the case where \mathcal{H} is a scalar function.

The first operation to perform is to generate the vertices along the domain boundary. Let us suppose that $\partial\Omega$ is described as the union of parametric curves $\mathbf{g}_i(s)$, $i = 1, \dots, N$, for instance splines or polygonal splits. For simplicity, we assume that, for each curve, the s parameter varies between 0 and 1. If we wish to generate $N_i + 1$ vertices along the curve \mathbf{g}_i it is sufficient to create a vertex for all the values of s for which the function

$$f_i(s) = \int_0^s \mathcal{H}^{-1}(\mathbf{g}_i(\tau)) \left| \frac{d\mathbf{g}_i}{ds}(\tau) \right| d\tau$$

takes integer values. More precisely, the curvilinear coordinates $s_i^{(j)}$ of the nodes to generate along the curve \mathbf{g}_i satisfy the relations

$$f_i(s_i^{(j)}) = j, \quad j = 0, \dots, N_i \text{ with the constraints } s_i^{(0)} = 0, s_i^{(N_i)} = 1.$$

The procedure is similar to the one described in Sec. 6.1. Note that the term $\left| \frac{d\mathbf{g}_i}{ds} \right|$ accounts for the intrinsic metric of the curve.

This being done, the advancing front process can start. The latter is described by a data structure that contains the list of the sides defining the boundary between the

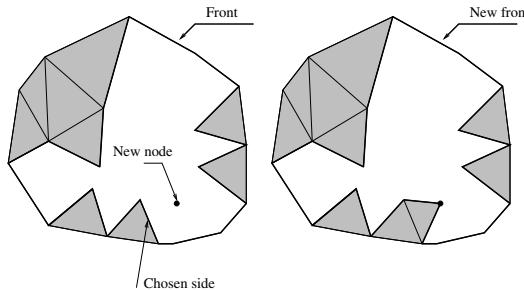


Fig. 6.10. Advancement of the front. The previously triangulated part of the domain has been shaded

already triangulated portion of Ω and the one yet to triangulate. At the beginning of the process, the front contains the boundary sides.

During the process of grid generation, each side of the front is available to create a new element, which is constructed by connecting the chosen side with a new or previously existing vertex of the grid. The choice whether to use an existing vertex or to create a new one depends on several factors, among which the compatibility between the dimension and the shape of the element that would be generated and the ones provided by the spacing function \mathcal{H} . Moreover, the new element must not intersect any side of the front.

Once the new element has been generated, its new sides will be “added” to the front so that the latter describes the new boundary between the triangulated and non-triangulated part, while the initial side is removed from the data list. This way, during the generation process, the front will progress from the already triangulated zones toward the zone yet to be triangulated (see Fig. 6.10).

The general advancing front algorithm hence consists of the following steps:

1. define the boundary of the domain to be triangulated;
2. initialize the front by a piecewise linear curve conforming to the boundary;
3. choose the side to be removed from the front based on some criterion (typically the choice of the shortest side provides good quality meshes);
4. for the side, say AB , chosen this way:
 - a) select the “candidate” vertex C , i.e. the point inside the domain distant from AB according to the desired spacing function \mathcal{H} ;
 - b) seek an already existing point C' on the front in a suitable neighborhood of C . If the search is successful, C' becomes the new candidate C point. Continue the search;
 - c) establish whether the triangle ABC intersects some other side of the front. If so, select a new candidate point from the front and start back from step 4.b);
5. add the new point C , the new edges and the new triangle ABC to the corresponding lists;
6. erase the edge AB from the front and add the new edges;
7. if the front is non-empty, continue from point 3.

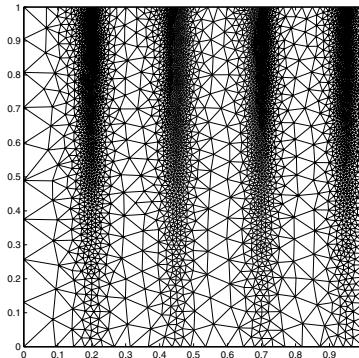


Fig. 6.11. Advancing front technique. Example of non-uniform spacing

It is obvious that if we wish the computational cost to be a linear as a function of the number of generated elements, it will be necessary to make the above-described operations as independent as possible of the number of dimensions of the grid we are generating and, in particular, of the dimensions of the advancing front. Such an objective is all but trivial, especially because operations such as the control of the intersection of a new triangle, or the search for the vertices of the front close to a generic point, span the whole front. We refer for this to the specialized literature, and in particular to Chap. 14 and 17 of [TSW99].

As previously pointed out in the algorithm description, the quality of the generated grid depends on the procedure of choice of the front edge on which to generate the new triangle. In particular, a frequently adopted technique consists in choosing the side with the smallest length: intuitively, this allows to also satisfy non-uniform spacing requirements, without risking that the zones where a more dense node distributions is required are overwritten by triangles associated to a coarser spacing. An example of mesh obtained through such technique, in correspondence of the choice $\mathcal{H}(x, y) = e^4 \sin(8\pi x) e^{-2y}$, is represented in Fig. 6.11.

By implementing the suitable tricks and data structures, the advancing algorithm provides a grid whose spacing is coherent with the requested one, with computational times nearly proportional to the number of generated elements.

The advancing front technique can also be used for the generation of quadrangular grids.

6.5 Regularization techniques

Once the grid has been generated, a post-processing can be necessary in order to improve its regularity. Some methods allow to transform the grid via operations that improve the triangles shape. In particular, we will examine regularization techniques that modify either the topological features (by diagonal exchange) or the geometrical features (by node displacement).

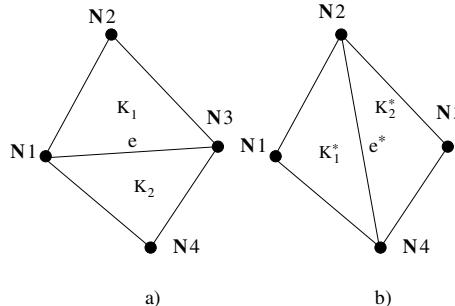


Fig. 6.12. The two configurations obtained via diagonal exchange in the convex quadrilateral formed by two adjacent elements. The two configurations are compared based on an optimality criterion

6.5.1 Diagonal exchange

The exchange of diagonals is a technique allowing to modify the topology of the grid without changing the position and number of its vertices. Such technique is based on the fact that a quadrilateral can be subdivided into a couple of triangles sharing a common side in two different ways (see Fig. 6.12).

In general, diagonal exchange is used to improve the quality of non-structured grids by following a prefixed optimality criterion. If the objective, for instance, is to avoid too wide angles, a possible criterion is to perform the exchange in the case where the sum of the angles opposite the diagonal is bigger than π .

A general scheme for a possible diagonal exchange algorithm is obtained by defining the optimality criterion at the element level, under the form of an appropriate non-negative function $S : K \rightarrow \mathbb{R}^+ \cup \{0\}$ that takes value 0 in the case where K has the “optimal” shape and dimension. For instance, we can use

$$S(K) = \left| \frac{|K|}{\sum_{i=1}^3 |e_i^K|^2} - \frac{\sqrt{3}}{12} \right|, \quad (6.3)$$

where $|K|$ denotes the size of K , e_i^K represents a generic side of K and $|e_i^K|$ is its length. Using this function, we privilege the triangles “close” to the equilateral one, for which $S(K) = 0$. Thus, we will generally obtain a grid as regular as possible, which does not take the spacing into account. With reference to Fig. 6.12, the algorithm will proceed as follows:

1. *Cycle 0*: set the exchanged side counter to zero: $swap = 0$;
2. range through all the internal sides e of the current mesh;
3. if the two triangles adjacent to e form a convex quadrilateral:
 - a) compute $G = S^2(K_1) + S^2(K_2) - [S^2(K_1^*) + S^2(K_2^*)]$;
 - b) if $G \geq \tau$, with $\tau > 0$ a prefixed tolerance, then execute the diagonal exchange (hence modify the current grid) and set $swap = swap + 1$;
4. if $swap > 0$ start back from *Cycle 0*. Otherwise, the procedure terminates.

It can easily be verified that this algorithm necessarily terminates in a finite number of steps as, to each diagonal exchange, the positive quantity $\sum_K S^2(K)$, where the sum is extended to all the triangles of the current grid, is reduced by the finite quantity G (note that, although the grid is modified, at each diagonal exchange the number of elements and sides remains unchanged).

Remark 6.2 It is not always a good option to construct the optimality function S at the element level. For instance, based on the available data structures, S can also be associated to the nodes or to the sides of the grid. •

The diagonal exchange technique is also the basis for a widely used algorithm (the Lawson algorithm) for the Delaunay triangulation. It can indeed be proven that, starting from *any* triangulation of a convex domain, the corresponding Delaunay triangulation (which we recall to be unique) can be obtained through a finite number of diagonal exchanges. Moreover, the maximum number of necessary swaps to this purpose can be determined a priori and is a function of the number of grid vertices. The technique (and convergence results) can be extended to constrained Delaunay triangulations, through a suitable modification of the algorithm. We refer to the specialized literature, for instance [GB98], for the details.

6.5.2 Node displacement

Another method to improve the quality of the grid consists in moving its points without modifying its topology. Let us consider an internal vertex P and the polygon \mathcal{K}_P constituted by the union of the grid elements containing it. The set \mathcal{K}_P is often called element “patch” associated to P and has been considered in Sec. 4.6. A regularization technique, called *Laplacian regularization*, or *barycentrization*, consists in moving P to the center of gravity of \mathcal{K}_P , that is in computing its new position \mathbf{x}_P as follows:

$$\mathbf{x}_P = |\mathcal{K}_P|^{-1} \int_{\mathcal{K}_P} \mathbf{x} d\mathbf{x}$$

(see Fig. 6.13). This procedure will obviously be iterated on all the internal vertices of the mesh and repeated several times. At convergence, the final grid is the one minimizing the quantity

$$\sum_P \int_{\mathcal{K}_P} (\mathbf{x}_P - \mathbf{x})^2 d\mathbf{x}, \quad (6.4)$$

where the sum is extended to all the internal vertices of the grid. The name of such procedure derives from the known property of harmonic functions (those having a null Laplacian) which take in a point of the domain a value equal to that of the mean on a closed curve containing the point.

The final grid will generally depend on the way the vertices are browsed. Moreover, note that this procedure can provide an unacceptable grid if \mathcal{K}_P is a concave polygon, as \mathbf{x}_P can fall out of the polygon. We then present an extension of the procedure that is suitable for generic patches of elements. We consider Fig. 6.14 which shows a concave

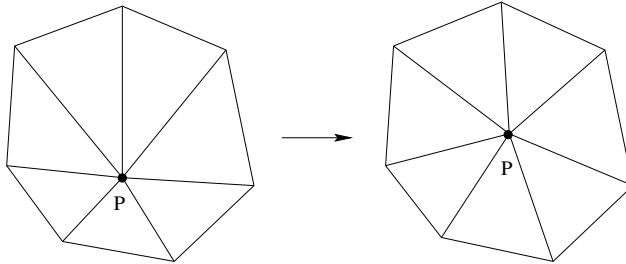


Fig. 6.13. Displacement of a point to the center of gravity of the convex polygon \mathcal{K}_P formed by the union of the elements containing P

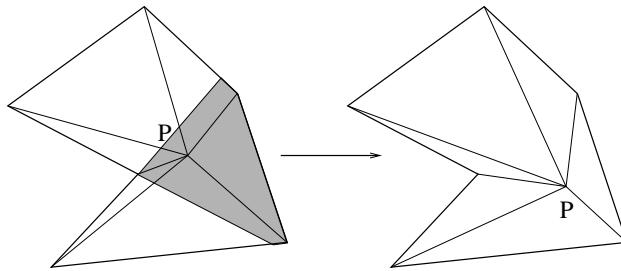


Fig. 6.14. Modification of the Laplacian regularization algorithm for concave patches. At the left-hand side, the initial patch; at the right-hand side, the modification due to regularization. We have shaded the concave polygon \mathcal{C}_P

patch \mathcal{K}_P . We define \mathcal{C}_P as the locus of points of \mathcal{K}_P “visible” to all of the boundary points of \mathcal{K}_P , that is $\mathcal{C}_P = \{A \in \mathcal{K}_P : AB \subset \mathcal{K}_P, \forall B \in \partial\mathcal{K}_P\}$; note that \mathcal{C}_P is always convex. The modification of the regularization algorithm consists in placing P not in the center of gravity of \mathcal{K}_P , but in that of \mathcal{C}_P , as illustrated in Fig. 6.14. Clearly, in the case of convex patches, we have $\mathcal{C}_P = \mathcal{K}_P$. The construction of \mathcal{C}_P can be executed in a computationally efficient manner by using suitable algorithms, whose description is out of the scope of this book.

Another option consists in displacing the vertex to the center of gravity of the boundary of \mathcal{K}_P (or \mathcal{C}_P in the case of concave patches), i.e. in setting

$$\mathbf{x}_P = |\partial\mathcal{K}_P|^{-1} \int_{\partial\mathcal{K}_P} \mathbf{x} d\mathbf{x}.$$

This is equivalent to minimizing the square of the distance between the vertex P and the sides forming the patch boundary.

A further technique, which can often be found in the literature, consists in displacing each internal vertex to the center of gravity of the vertices belonging to the associated patch, i.e. in computing the new position of each internal vertex P via

$$\mathbf{x}_P = \left(\sum_{\substack{N \in \mathcal{K}_P \\ N \neq P}} \mathbf{x}_N \right) \Big/ \left(\sum_{\substack{N \in \mathcal{K}_P \\ N \neq P}} 1 \right),$$

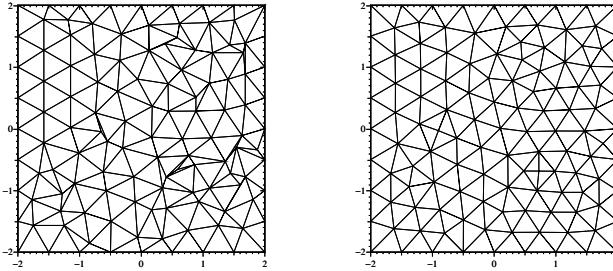


Fig. 6.15. Example of regularization through both diagonal exchange and node displacement

where the sum is extended to all the vertices N belonging to the patch. Despite being the simplest methodology, the latter often yields bad results, in particular if the distribution of vertices inside the patch is very irregular. Moreover, it is more difficult to extend it to concave patches. Thus, the two previous procedures are preferable. In Fig. 6.15 we present an example of consequent application of both of the above-described regularization techniques. Note that the regularization algorithms presented here tend to uniform the grid and therefore to destroy its thickenings or coarsenings due for instance to grid adaptivity procedures such as the ones described in Chap. 4. However, it is possible to modify them to account for a non-uniform spacing. For instance, a weighted barycentrization can be used, i.e. by setting

$$\mathbf{x}_P = \left(\int_{\mathcal{K}_P} \mu(\mathbf{x}) d\mathbf{x} \right)^{-1} \int_{\mathcal{K}_P} \mu(\mathbf{x}) \mathbf{x} d\mathbf{x},$$

where the strictly positive weighting function μ depends on the grid spacing function. In the case of non-uniform spacing, μ will take higher values in the zones where the grid must be finer. When choosing for instance $\mu = \mathcal{H}^{-1}$ the resulting grid (approximately) minimizes

$$\sum_P \int_{\mathcal{K}_P} [\mathcal{H}^{-1}(\mathbf{x})(\mathbf{x}_P - \mathbf{x})]^2 d\mathbf{x},$$

where the sum is extended to the internal vertices.

Also concerning the diagonal exchange procedure, we can take the spacing into account when evaluating the “optimal” configuration, for instance by suitably changing the definition of function $S(K)$ in (6.3).

Algorithms for the solution of linear systems

In this chapter we make a quick and elementary introduction of some of the basic algorithms that are used to solve a system of linear algebraic equations. For a more thorough presentation we advise the reader to refer to, e.g., [QSS07], Chap. 3 and 4, [Saa96] and [vdV03].

A system of m linear equations in n unknowns is a set of algebraic relations of the form

$$\sum_{j=1}^n a_{ij}x_j = b_i, \quad i = 1, \dots, m \quad (7.1)$$

x_j being the unknowns, a_{ij} the system's coefficients and b_i the known terms. The system (7.1) will more commonly be written in matrix form

$$\mathbf{Ax} = \mathbf{b}, \quad (7.2)$$

having denoted by $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{m \times n}$ the coefficient matrix, $\mathbf{b} = (b_i) \in \mathbb{R}^m$ being the known term vector (also named as the right-hand side) and $\mathbf{x} = (x_i) \in \mathbb{R}^n$ the unknown vector. We call *solution* of (7.2) any n -tuple of values x_i verifying (7.1).

In the following sections, we recall some numerical techniques for the solution of (7.2) in the case where $m = n$; we will obviously suppose that \mathbf{A} is non-singular, i.e. that $\det(\mathbf{A}) \neq 0$. Numerical methods are called *direct* if they lead to the solution of the system in a finite number of operations, or *iterative* if they require a (theoretically) infinite number.

7.1 Direct methods

The solution of a linear system can be performed through the Gauss elimination method (GEM), where the initial system $\mathbf{Ax} = \mathbf{b}$ is reconducted in n steps to an equivalent system (i.e. having the same solution) of the form $\mathbf{A}^{(n)}\mathbf{x} = \mathbf{b}^{(n)}$ where $\mathbf{A}^{(n)} = \mathbf{U}$ is a nonsingular upper triangular matrix and $\mathbf{b}^{(n)}$ is a new known term. It will be possible to solve the latter system with a computational cost of the order of n^2 operations,

through the following backward substitution algorithm:

$$\begin{aligned} x_n &= \frac{b_n^{(n)}}{u_{nn}}, \\ x_i &= \frac{1}{u_{ii}} \left(b_i^{(n)} - \sum_{j=i+1}^n u_{ij} x_j \right), \quad i = n-1, \dots, 1. \end{aligned} \tag{7.3}$$

Denoting by $A^{(1)}\mathbf{x} = \mathbf{b}^{(1)}$ the original system, the k -th step of GEM is achieved via the following formulae:

$$\begin{aligned} m_{ik} &= \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, & i = k+1, \dots, n, \\ a_{ij}^{(k+1)} &= a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, & i, j = k+1, \dots, n \\ b_i^{(k+1)} &= b_i^{(k)} - m_{ik} b_k^{(k)}, & i = k+1, \dots, n. \end{aligned} \tag{7.4}$$

We note that in this way, the elements $a_{ij}^{(k+1)}$ with $j = k$ and $i = k+1, \dots, n$ result to be null. The m_{ik} elements are called *multipliers*, while the denominators $a_{kk}^{(k)}$ are named *pivotal elements*. The GEM can obviously be achieved only if all the pivotal elements result to be non null. This happens, for instance, for symmetric positive definite matrices and for strict diagonal dominant ones. In general, it will be necessary to resort to the *pivoting* method, i.e. to the swapping of rows (and/or columns) of $A^{(k)}$, in order to ensure that the element $a_{kk}^{(k)}$ be non-null.

To complete the Gauss eliminations, we need $2(n-1)n(n+1)/3 + n(n-1)$ flops, to which we must add n^2 flops to solve the upper triangular system $\mathbf{Ux} = \mathbf{b}^{(n)}$ via the backward substitution method. Hence, about $(2n^3/3 + 2n^2)$ flops are needed to solve the linear system via the GEM. More simply, by neglecting the lower order terms with respect to n , it can be said that the Gaussian elimination process requires $2n^3/3$ flops.

It can be verified that the GEM is equivalent to factorizing the A matrix, i.e. to rewriting A as the product LU of two matrices. The U matrix, upper triangular, coincides with the matrix $A^{(n)}$ obtained at the end of the elimination process. The L matrix is lower triangular, its diagonal elements are equal to 1 while the ones located in the remaining lower triangular portion are equal to the multipliers.

Once the matrices L and U are known, the solution of the initial linear system simply involves the solution (in a sequence) of the two triangular systems

$$\mathbf{Ly} = \mathbf{b}, \quad \mathbf{Ux} = \mathbf{y}.$$

Obviously, the computational cost of the factorization process is the same as the one required by the GEM. The advantages of such a reinterpretation are evident: as L and U depend on A only and not on the known term, the same factorization can be used to solve different linear systems having the same matrix A , but a variable known term \mathbf{b} (think for instance of the discretization of a linear parabolic problem by an implicit method where at each time step it is necessary to solve a system with the same matrix

all the time, but with a different known term). Consequently, as the computational cost is concentrated in the elimination procedure (about $2n^3/3$ flops), we have in this way a considerable reduction in the number of operations when we want to solve several linear systems having the same matrix.

If A is a symmetric positive definite matrix, the LU factorization can be conveniently specialized. Indeed, there exists only one upper triangular matrix H with positive elements on the diagonal such that

$$A = H^T H. \quad (7.5)$$

Equation (7.5) is the so-called Cholesky factorization. The h_{ij} elements of H^T are given by the following formulae: $h_{11} = \sqrt{a_{11}}$ and, for $i = 2, \dots, n$,

$$\begin{aligned} h_{ij} &= \left(a_{ij} - \sum_{k=1}^{j-1} h_{ik} h_{jk} \right) / h_{jj}, \quad j = 1, \dots, i-1, \\ h_{ii} &= \left(a_{ii} - \sum_{k=1}^{i-1} h_{ik}^2 \right)^{1/2}. \end{aligned}$$

This algorithm only requires about $n^3/3$ flops, i.e. saves about twice the computing time of the LU factorization and about half the memory.

Let us now consider the particular case of a linear system with non-singular *tridiagonal* matrix A of the form

$$A = \begin{bmatrix} a_1 & c_1 & & 0 \\ b_2 & a_2 & \ddots & \\ \ddots & & & c_{n-1} \\ 0 & b_n & a_n & \end{bmatrix}.$$

In this case, the L and U matrices of the LU factorization of A are two bidiagonal matrices of the type

$$L = \begin{bmatrix} 1 & & 0 \\ \beta_2 & 1 & \\ \ddots & \ddots & \\ 0 & \beta_n & 1 \end{bmatrix}, \quad U = \begin{bmatrix} \alpha_1 & c_1 & & 0 \\ & \alpha_2 & \ddots & \\ & & \ddots & c_{n-1} \\ 0 & & & \alpha_n \end{bmatrix}.$$

The α_i and β_i unknown coefficients can be easily computed through the following equations:

$$\alpha_1 = a_1, \quad \beta_i = \frac{b_i}{\alpha_{i-1}}, \quad \alpha_i = a_i - \beta_i c_{i-1}, \quad i = 2, \dots, n.$$

This algorithm is named *Thomas algorithm* and can be seen as a particular kind of LU factorization without pivoting.

7.2 Iterative methods

Iterative methods aim at constructing the solution \mathbf{x} of a linear system as the limit of a sequence $\{\mathbf{x}^{(n)}\}$ of vectors. To obtain the single elements of the sequence, computing the residue $\mathbf{r}^{(n)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(n)}$ of the system is required. In the case where the matrix is full and of order n , the computational cost of an iterative method is therefore of the order of n^2 operations per iteration. Such cost must be compared with the approximately $2n^3/3$ operations required by a direct method. Consequently, iterative methods are competitive with direct methods only if the number of necessary iterations to reach convergence (within a given tolerance) is independent of n or depends on n in a sub-linear way.

Other considerations in the choice between an iterative method and a direct one intervene as soon as the matrix is sparse.

7.2.1 Classical iterative methods

A general strategy to construct iterative methods is based on an additive decomposition, called splitting, starting from matrix \mathbf{A} of the form $\mathbf{A}=\mathbf{P}-\mathbf{N}$, where \mathbf{P} and \mathbf{N} are two suitable matrices and \mathbf{P} is non-singular. For reasons which will become evident in the remainder, \mathbf{P} is also called *preconditioning matrix* or *preconditioner*.

Precisely, given $\mathbf{x}^{(0)}$, we obtain $\mathbf{x}^{(k)}$ for $k \geq 1$ by solving the new systems

$$\mathbf{P}\mathbf{x}^{(k+1)} = \mathbf{N}\mathbf{x}^{(k)} + \mathbf{b}, \quad k \geq 0 \quad (7.6)$$

or, equivalently,

$$\mathbf{x}^{(k+1)} = \mathbf{B}\mathbf{x}^{(k)} + \mathbf{P}^{-1}\mathbf{b}, \quad k \geq 0 \quad (7.7)$$

having denoted by $\mathbf{B} = \mathbf{P}^{-1}\mathbf{N}$ the *iteration matrix*.

We are interested in *convergent* iterative methods, i.e. such that $\lim_{k \rightarrow \infty} \mathbf{e}^{(k)} = \mathbf{0}$ for each choice of the *initial vector* $\mathbf{x}^{(0)}$, having denoted by $\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}$ the error. As with a recursive argument we find

$$\mathbf{e}^{(k)} = \mathbf{B}^k \mathbf{e}^{(0)}, \quad \forall k = 0, 1, \dots \quad (7.8)$$

We can conclude that an iterative method of the form (7.6) is convergent if and only if $\rho(\mathbf{B}) < 1$, $\rho(\mathbf{B})$ being the spectral radius of the iteration matrix \mathbf{B} , i.e. the maximum modulus of the eigenvalues of \mathbf{B} .

Equation (7.6) can also be formulated in the form

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{P}^{-1}\mathbf{r}^{(k)}, \quad (7.9)$$

having denoted by

$$\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} \quad (7.10)$$

the *residue* at step k . Equation (7.9) thus expresses the fact that to update the solution at step $k + 1$, it is necessary to solve a linear system with matrix \mathbf{P} . Hence, besides

being non-singular, P must be invertible at a low computational cost if we want to prevent the overall cost of the scheme from increasing excessively (obviously, in the limit case where P is equal to A and $N=0$, method (7.9) would converge in only one iteration, but at the cost of a direct method).

Let us now see how to accelerate the convergence of the iterative methods (7.6) by exploiting the latter form. We denote by

$$R_P = I - P^{-1}A$$

the iteration matrix associated to method (7.9). Matrix (7.9) can be generalized by introducing a suitable relaxation (or acceleration) parameter α . This way, we obtain the *stationary Richardson methods* (more simply called *Richardson* methods), of the form

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha P^{-1} \mathbf{r}^{(k)}, \quad k \geq 0. \quad (7.11)$$

More generally, supposing α to be dependent on the iteration index, we obtain the *non-stationary Richardson methods* given by

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k P^{-1} \mathbf{r}^{(k)}, \quad k \geq 0. \quad (7.12)$$

If we set $\alpha = 1$, we can recover two classical iterative methods: the Jacobi method if $P = D(A)$ (the diagonal part of A), the Gauss-Seidel method if $P = L(A)$ (the lower triangular part of A).

The iteration matrix at step k for such methods is given by

$$R(\alpha_k) = I - \alpha_k P^{-1} A,$$

(note that the latter depends on k). In the case where $P=I$, the methods under exam will be called *non preconditioned*.

We can rewrite (7.12) (and therefore also (7.11)) in a form of greater computational interest. Indeed, having set $\mathbf{z}^{(k)} = P^{-1} \mathbf{r}^{(k)}$ (the so-called *preconditioned residue*), we have that $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{z}^{(k)}$ and $\mathbf{r}^{(k+1)} = \mathbf{b} - A\mathbf{x}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k A\mathbf{z}^{(k)}$. To summarize, a non-stationary Richardson method at step $k+1$ requires the following operations:

- solving the linear system $P\mathbf{z}^{(k)} = \mathbf{r}^{(k)}$,
 - computing the acceleration parameter α_k ,
 - updating the solution $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{z}^{(k)}$,
 - updating the residue $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k A\mathbf{z}^{(k)}$.
- (7.13)

As far as convergence is concerned for the stationary Richardson method (for which $\alpha_k = \alpha$, for each $k \geq 0$) the following result holds:

Property 7.1 If P is a non-singular matrix, the stationary Richardson method (7.11) is convergent if and only if

$$\frac{2\operatorname{Re}\lambda_i}{\alpha|\lambda_i|^2} > 1 \quad \forall i = 1, \dots, n, \quad (7.14)$$

λ_i being the eigenvalues of $P^{-1}A$.

Moreover, if we suppose that $P^{-1}A$ has positive real eigenvalues, ordered in such a way that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0$, then, the stationary Richardson method (7.11) converges if and only if $0 < \alpha < 2/\lambda_1$. Having set

$$\alpha_{opt} = \frac{2}{\lambda_1 + \lambda_n} \quad (7.15)$$

the spectral radius of the iteration matrix R_α is minimal if $\alpha = \alpha_{opt}$, with

$$\rho_{opt} = \min_{\alpha} [\rho(R_\alpha)] = \frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}. \quad (7.16)$$

If P and A are both symmetric and positive definite, it can be proven that the Richardson method converges monotonically with respect to the vector norms $\|\cdot\|_2$ and $\|\cdot\|_A$. We recall that $\|\mathbf{v}\|_2 = (\sum_{i=1}^n v_i^2)^{1/2}$ and $\|\mathbf{v}\|_A = (\sum_{i,j=1}^n v_i a_{ij} v_j)^{1/2}$.

In this case, thanks to (7.16), we can relate ρ_{opt} with the condition number introduced in Sec. 4.5.2 in the following way:

$$\rho_{opt} = \frac{K_2(P^{-1}A) - 1}{K_2(P^{-1}A) + 1}, \quad \alpha_{opt} = \frac{2\|A^{-1}P\|_2}{K_2(P^{-1}A) + 1}. \quad (7.17)$$

The importance of the choice of the P preconditioner in a Richardson method can therefore be clearly understood. We refer to Chap. 4 of [QSS07] for some examples of preconditioners.

7.2.2 Gradient and conjugate gradient methods

The optimal expression of the acceleration parameter α , indicated in (7.15), results to be of little practical utility, as it requires knowing the maximal and minimal eigenvalues of the matrix $P^{-1}A$. In the particular case of positive definite symmetric matrices, it is however possible to evaluate the optimal acceleration parameter in a *dynamic* way, that is as a function of quantities computed by the method itself at step k , as we show below.

First of all, we observe that in the case where A is a symmetric positive definite matrix, solving system (7.2) is equivalent to finding the minimum $\mathbf{x} \in \mathbb{R}^n$ of the quadratic form

$$\varPhi(\mathbf{y}) = \frac{1}{2}\mathbf{y}^T A \mathbf{y} - \mathbf{y}^T \mathbf{b},$$

called *energy of the system* (7.2).

The problem is thus reconducted to determining the minimum point \mathbf{x} of Φ starting from a point $\mathbf{x}^{(0)} \in \mathbb{R}^n$ and, consequently, choosing suitable directions along which to move to approach the solution \mathbf{x} as quickly as possible. The optimal direction, joining $\mathbf{x}^{(0)}$ and \mathbf{x} , is obviously unknown a priori: we will therefore have to move from $\mathbf{x}^{(0)}$ along another direction $\mathbf{d}^{(0)}$ and fix a new point $\mathbf{x}^{(1)}$ on the latter, from which to repeat the procedure until convergence.

At the generic step k we will then determine $\mathbf{x}^{(k+1)}$ as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}, \quad (7.18)$$

α_k being the value fixing the length of the step along $\mathbf{d}^{(k)}$. The most natural idea, consisting in taking as downhill direction the maximal incline direction for Φ , given by $\mathbf{r}^{(k)} = -\nabla\Phi(\mathbf{x}^{(k)})$, leads to the *gradient or steepest descent method*.

The latter yields to the following algorithm: given $\mathbf{x}^{(0)} \in \mathbb{R}^n$, having set $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$, for $k = 0, 1, \dots$ until convergence, we compute

$$\begin{aligned} \alpha_k &= \frac{\mathbf{r}^{(k)T} \mathbf{r}^{(k)}}{\mathbf{r}^{(k)T} \mathbf{A} \mathbf{r}^{(k)}}, & \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \alpha_k \mathbf{r}^{(k)}, \\ \mathbf{r}^{(k+1)} &= \mathbf{r}^{(k)} - \alpha_k \mathbf{A} \mathbf{r}^{(k)}. \end{aligned}$$

Its preconditioned version takes the following form: given $\mathbf{x}^{(0)} \in \mathbb{R}^n$, having set $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$, $\mathbf{z}^{(0)} = P^{-1}\mathbf{r}^{(0)}$, for $k = 0, 1, \dots$ until convergence we compute

$$\begin{aligned} \alpha_k &= \frac{\mathbf{z}^{(k)T} \mathbf{r}^{(k)}}{\mathbf{z}^{(k)T} \mathbf{A} \mathbf{z}^{(k)}}, & \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \alpha_k \mathbf{z}^{(k)}. \\ \mathbf{r}^{(k+1)} &= \mathbf{r}^{(k)} - \alpha_k \mathbf{A} \mathbf{z}^{(k)}, & P\mathbf{z}^{(k+1)} &= \mathbf{r}^{(k+1)}. \end{aligned}$$

As far as the convergence properties of the descent method are concerned, the following result holds

Theorem 7.1 *Let \mathbf{A} be symmetric and positive definite, then the gradient method converges for each value of the initial datum $\mathbf{x}^{(0)}$ and*

$$\|\mathbf{e}^{(k+1)}\|_{\mathbf{A}} \leq \frac{K_2(\mathbf{A}) - 1}{K_2(\mathbf{A}) + 1} \|\mathbf{e}^{(k)}\|_{\mathbf{A}}, \quad k = 0, 1, \dots \quad (7.19)$$

where $\|\cdot\|_{\mathbf{A}}$ is the previously defined energy norm.

A similar result, with $K_2(\mathbf{A})$ replaced by $K_2(P^{-1}\mathbf{A})$, holds also in the case of the preconditioned gradient method, as long as we assume that P is also symmetric and positive definite.

An even more effective alternative consists in using the *conjugate gradient method* where the descent directions no longer coincide with that of the residue. In particular,

having set $\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$, we seek directions of the form

$$\mathbf{p}^{(k+1)} = \mathbf{r}^{(k+1)} - \beta_k \mathbf{p}^{(k)}, \quad k = 0, 1, \dots \quad (7.20)$$

where the parameters $\beta_k \in \mathbb{R}$ are to be determined so that

$$(\mathbf{A}\mathbf{p}^{(j)})^T \mathbf{p}^{(k+1)} = 0, \quad j = 0, 1, \dots, k. \quad (7.21)$$

Directions of this type are called A-orthogonal. The method in the preconditioned case then takes the form: given $\mathbf{x}^{(0)} \in \mathbb{R}^n$, having set $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$, $\mathbf{z}^{(0)} = \mathbf{P}^{-1}\mathbf{r}^{(0)}$ and $\mathbf{p}^{(0)} = \mathbf{z}^{(0)}$, the k -th iteration, with $k = 0, 1, \dots$, is

$$\begin{aligned} \alpha_k &= \frac{\mathbf{p}^{(k)T} \mathbf{r}^{(k)}}{(\mathbf{A}\mathbf{p}^{(k)})^T \mathbf{p}^{(k)}}, \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}, \\ \mathbf{r}^{(k+1)} &= \mathbf{r}^{(k)} - \alpha_k \mathbf{A}\mathbf{p}^{(k)}, \\ \mathbf{P}\mathbf{z}^{(k+1)} &= \mathbf{r}^{(k+1)}, \\ \beta_k &= \frac{(\mathbf{A}\mathbf{p}^{(k)})^T \mathbf{z}^{(k+1)}}{\mathbf{p}^{(k)T} \mathbf{A}\mathbf{p}^{(k)}}, \\ \mathbf{p}^{(k+1)} &= \mathbf{z}^{(k+1)} - \beta_k \mathbf{p}^{(k)}. \end{aligned}$$

The parameter α_k is chosen in order to guarantee that the error $\|(\mathbf{e})^{(k+1)}\|_A$ be minimized along the descent direction $\mathbf{p}^{(k)}$. The β_k parameter, instead, is chosen so that the new direction $\mathbf{p}^{(k+1)}$ be A-conjugate with $\mathbf{p}^{(k)}$, or $(\mathbf{A}\mathbf{p}^{(k)})^T \mathbf{p}^{(k+1)} = 0$. Indeed, it can be proven (thanks to the induction principle) that if the latter relation is verified, then so are all the ones in (7.21) relating to $j = 0, \dots, k-1$. For a complete derivation of the method, see e.g. [QSS07, Chap. 4] or [Saa96].

It can be proven that the conjugate gradient method converges in exact arithmetics in at most n steps and that

$$\|\mathbf{e}^{(k)}\|_A \leq \frac{2c^k}{1+c^{2k}} \|\mathbf{e}^{(0)}\|_A, \quad (7.22)$$

with

$$c = \frac{\sqrt{K_2(\mathbf{P}^{-1}\mathbf{A})} - 1}{\sqrt{K_2(\mathbf{P}^{-1}\mathbf{A})} + 1}. \quad (7.23)$$

Consequently, in the absence of roundoff errors, the CG method can be seen as a direct method as it terminates after a finite number of operations.

On the other hand, for matrices of large dimension, it is usually applied as an iterative method and is arrested when an error estimator (as for instance the relative residue) is less than a given tolerance.

Thanks to (7.23), the dependence on the reduction factor of the error on the matrix condition number is more favorable than the one of the gradient method (due to the presence of the square root of $K_2(P^{-1}A)$).

It can be noted that the number of iterations required for convergence (up to a prescribed tolerance) is proportional to $\frac{1}{2}\sqrt{K_2(P^{-1}A)}$ for the preconditioned conjugate gradient method, a decided improvement w.r.t $\frac{1}{2}K_2(P^{-1}A)$ for the preconditioned gradient method. Of course, the PCG method is more costly per iteration, both in CPU time and storage.

7.2.3 Krylov subspace methods

Generalizations of the gradient method in the case where matrix A is not symmetric lead to the so-called Krylov methods. Notable examples are the GMRES method and the conjugate bigradient method BiCG, as well as its stabilized version, the BiCGSTAB method. We refer the interested reader to [QSS07, Chap. 4], [Saa96] and [vdV03].

Here we briefly review the GMRES (generalized minimal residual) method. We start by a revisit of the Richardson method (7.13) with $P = I$; the residual at the k -th step can be related to the initial residual as

$$\mathbf{r}^{(k)} = \prod_{j=0}^{k-1} (I - \alpha_j A) \mathbf{r}^{(0)} = p_k(A) \mathbf{r}^{(0)}, \quad (7.24)$$

where $p_k(A)$ is a polynomial in A of degree k . If we introduce the space

$$K_m(A; \mathbf{v}) = \text{span}\{\mathbf{v}, A\mathbf{v}, \dots, A^{m-1}\mathbf{v}\}, \quad (7.25)$$

it immediately appears from (7.24) that $\mathbf{r}^{(k)} \in K_{k+1}(A; \mathbf{r}^{(0)})$. The space defined in (7.25) is called the *Krylov subspace* of order m associated with matrix A and vector \mathbf{v} . It is a subspace of the set spanned by all the vectors $\mathbf{u} \in \mathbb{R}^n$ that can be written as $\mathbf{u} = p_{m-1}(A)\mathbf{v}$, where p_{m-1} is a polynomial in A of degree $\leq m-1$.

Similarly, the iterate $\mathbf{x}^{(k)}$ of the Richardson method can be represented as follows

$$\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + \sum_{j=0}^{k-1} \alpha_j \mathbf{r}^{(j)},$$

whence $\mathbf{x}^{(k)}$ belongs to the space

$$W_k = \{\mathbf{v} = \mathbf{x}^{(0)} + \mathbf{y}, \mathbf{y} \in K_k(A; \mathbf{r}^{(0)})\}. \quad (7.26)$$

Notice also that $\sum_{j=0}^{k-1} \alpha_j \mathbf{r}^{(j)}$ is a polynomial in A of degree less than $k-1$. In the nonpreconditioned Richardson method we are thus looking for an approximate solution to \mathbf{x} in the space W_k . More generally, one can devise methods that search for approximate solutions of the form

$$\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + q_{k-1}(A) \mathbf{r}^{(0)}, \quad (7.27)$$

where q_{k-1} is a polynomial selected in such a way that $\mathbf{x}^{(k)}$ be, in a sense that must be made precise, the best approximation of \mathbf{x} in W_k . A method that looks for a solution of the form (7.27) with W_k defined as in (7.26) is called a *Krylov method*.

A first question concerning Krylov subspace iterations is whether the dimension of $K_m(\mathbf{A}; \mathbf{v})$ increases as the order m grows. A partial answer is provided by the following result.

Property 7.2 Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{v} \in \mathbb{R}^n$. The Krylov subspace $K_m(\mathbf{A}; \mathbf{v})$ has dimension equal to m iff the degree of \mathbf{v} with respect to \mathbf{A} , denoted by $\deg_{\mathbf{A}}(\mathbf{v})$, is not less than m ; the degree of \mathbf{v} is defined as the minimum degree of a monic nonnull polynomial p in \mathbf{A} , for which $p(\mathbf{A})\mathbf{v} = \mathbf{0}$.

The dimension of $K_m(\mathbf{A}; \mathbf{v})$ is thus equal to the minimum between m and the degree of \mathbf{v} with respect to \mathbf{A} and, as a consequence, the dimension of the Krylov subspaces is certainly a nondecreasing function of m . The degree of \mathbf{v} cannot be greater than n due to the Cayley-Hamilton theorem (see [QSS07, Sec. 1.7]).

Example 7.1 Consider the 4×4 matrix $\mathbf{A} = \text{tridiag}_4(-1, 2, -1)$. The vector $\mathbf{v} = [1, 1, 1, 1]^T$ has degree 2 with respect to \mathbf{A} since $p_2(\mathbf{A})\mathbf{v} = \mathbf{0}$ with $p_2(\mathbf{A}) = \mathbf{I}_4 - 3\mathbf{A} + \mathbf{A}^2$ (\mathbf{I}_4 is the 4×4 identity matrix), while there is no monic polynomial p_1 of degree 1 for which $p_1(\mathbf{A})\mathbf{v} = \mathbf{0}$. As a consequence, all Krylov subspaces from $K_2(\mathbf{A}; \mathbf{v})$ on, have dimension equal to 2. The vector $\mathbf{w} = [1, 1, -1, 1]^T$ has, instead, degree 4 with respect to \mathbf{A} . ■

For a fixed m , it is possible to compute an orthonormal basis for $K_m(\mathbf{A}; \mathbf{v})$ using the so-called *Arnoldi algorithm*.

Setting $\mathbf{v}_1 = \mathbf{v}/\|\mathbf{v}\|_2$, this method generates an orthonormal basis $\{\mathbf{v}_i\}$ for $K_m(\mathbf{A}; \mathbf{v}_1)$ using the Gram-Schmidt procedure (see [QSS07, Sec. 3.4.3]). For $k = 1, \dots, m$, the Arnoldi algorithm computes

$$\begin{aligned} h_{ik} &= \mathbf{v}_i^T \mathbf{A} \mathbf{v}_k, & i &= 1, 2, \dots, k, \\ \mathbf{w}_k &= \mathbf{A} \mathbf{v}_k - \sum_{i=1}^k h_{ik} \mathbf{v}_i, & h_{k+1,k} &= \|\mathbf{w}_k\|_2. \end{aligned} \tag{7.28}$$

If $\mathbf{w}_k = \mathbf{0}$ the process terminates and in such a case we say that a *breakdown* of the algorithm has occurred; otherwise, we set $\mathbf{v}_{k+1} = \mathbf{w}_k/\|\mathbf{w}_k\|_2$ and the algorithm restarts, incrementing k by 1.

It can be shown that if the method terminates at the step m then the vectors $\mathbf{v}_1, \dots, \mathbf{v}_m$ form a basis for $K_m(\mathbf{A}; \mathbf{v})$. In such a case, if we denote by $\mathbf{V}_m \in \mathbb{R}^{n \times m}$ the matrix whose columns are the vectors \mathbf{v}_i , we have

$$\mathbf{V}_m^T \mathbf{A} \mathbf{V}_m = \mathbf{H}_m, \quad \mathbf{V}_{m+1}^T \mathbf{A} \mathbf{V}_m = \widehat{\mathbf{H}}_m, \tag{7.29}$$

where $\widehat{\mathbf{H}}_m \in \mathbb{R}^{(m+1) \times m}$ is the upper Hessenberg matrix whose entries h_{ij} are given by (7.28) and $\mathbf{H}_m \in \mathbb{R}^{m \times m}$ is the restriction of $\widehat{\mathbf{H}}_m$ to the first m rows and m columns.

The algorithm terminates at an intermediate step $k < m$ iff $\deg_A(\mathbf{v}_1) = k$. As for the stability of the procedure, all the considerations valid for the Gram-Schmidt method hold. For more efficient and stable computational variants of (7.28), we refer to [Saa96].

We are now ready to solve the linear system (7.2) by a Krylov method. We look for the iterate $\mathbf{x}^{(k)}$ under the form (7.27); for a given $\mathbf{r}^{(0)}$, $\mathbf{x}^{(k)}$ is selected as being the unique element in W_k which satisfies a criterion of minimal distance from \mathbf{x} . The criterion for selecting $\mathbf{x}^{(k)}$ is precisely the distinguishing feature of a Krylov method.

The most natural idea consists of searching for $\mathbf{x}^{(k)} \in W_k$ as the vector which minimizes the Euclidean norm of the error. This approach, however, does not work in practice since $\mathbf{x}^{(k)}$ would depend on the (unknown) solution \mathbf{x} . Two alternative strategies can be pursued:

1. compute $\mathbf{x}^{(k)} \in W_k$ enforcing that the residual $\mathbf{r}^{(k)}$ is orthogonal to any vector in $K_k(A; \mathbf{r}^{(0)})$, i.e., we look for $\mathbf{x}^{(k)} \in W_k$ such that

$$\mathbf{v}^T(\mathbf{b} - A\mathbf{x}^{(k)}) = 0 \quad \forall \mathbf{v} \in K_k(A; \mathbf{r}^{(0)}); \quad (7.30)$$

2. compute $\mathbf{x}^{(k)} \in W_k$ minimizing the Euclidean norm of the residual $\|\mathbf{r}^{(k)}\|_2$, i.e.

$$\|\mathbf{b} - A\mathbf{x}^{(k)}\|_2 = \min_{\mathbf{v} \in W_k} \|\mathbf{b} - A\mathbf{v}\|_2. \quad (7.31)$$

Satisfying (7.30) leads to the Arnoldi method for linear systems (more commonly known as FOM, *full orthogonalization method*), while satisfying (7.31) yields the GMRES (*generalized minimum residual*) method.

We shall assume that k steps of the Arnoldi algorithm have been carried out, in such a way that an orthonormal basis for $K_k(A; \mathbf{r}^{(0)})$ has been generated and stored into the column vectors of the matrix V_k with $\mathbf{v}_1 = \mathbf{r}^{(0)} / \|\mathbf{r}^{(0)}\|_2$. In such a case the new iterate $\mathbf{x}^{(k)}$ can always be written as

$$\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + V_k \mathbf{z}^{(k)}, \quad (7.32)$$

where $\mathbf{z}^{(k)}$ must be selected according to a suitable criterion that we are going to specify. Consequently we have

$$\mathbf{r}^{(k)} = \mathbf{r}^{(0)} - A V_k \mathbf{z}^{(k)}. \quad (7.33)$$

Since $\mathbf{r}^{(0)} = \mathbf{v}_1 \|\mathbf{r}^{(0)}\|_2$, using (7.29), relation (7.33) becomes

$$\mathbf{r}^{(k)} = V_{k+1} (\|\mathbf{r}^{(0)}\|_2 \mathbf{e}_1 - \widehat{H}_k \mathbf{z}^{(k)}), \quad (7.34)$$

where \mathbf{e}_1 is the first unit vector of \mathbb{R}^{k+1} . Therefore, in the GMRES method the solution at step k can be computed through (7.32), provided

$$\mathbf{z}^{(k)} \text{ minimizes } \| \|\mathbf{r}^{(0)}\|_2 \mathbf{e}_1 - \widehat{H}_k \mathbf{z}^{(k)} \|_2 \quad (7.35)$$

(we note that the matrix V_{k+1} appearing in (7.34) does not alter the value of $\|\cdot\|_2$ since it is orthogonal). Having to solve at each step a least-squares problem of size k , the GMRES method will be the more effective the smaller is the number of iterations.

Similarly to the CG method, the GMRES method has the finite termination property, that is it terminates at most after n iterations, yielding the exact solution (in exact arithmetic). Indeed, the $k - th$ iterate minimizes the residual in the Krylov subspace K_k . Since every subspace is contained in the next subspace, the residual decreases monotonically. After n iterations, where n is the size of the matrix A , the Krylov space K_n is the whole of \mathbb{R}^n and hence the GMRES method arrives at the exact solution. Premature stops are due to a breakdown in the orthonormalization Arnoldi algorithm. More precisely, we have the following result.

Property 7.3 *A breakdown occurs for the GMRES method at a step m (with $m < n$) if and only if the computed solution $\mathbf{x}^{(m)}$ coincides with the exact solution to the system.*

However, the idea is that after a small number of iterations (relative to n), the vector \mathbf{x}_k is already a good approximation to the exact solution. This is confirmed by the convergence results that we report later in this section.

To improve the efficiency of the GMRES algorithm it is necessary to devise a stopping criterion which does not require the explicit evaluation of the residual at each step. This is possible, provided that the linear system with upper Hessenberg matrix \widehat{H}_k is appropriately solved.

In practice, matrix \widehat{H}_k in (7.29) is transformed into an upper triangular matrix $R_k \in \mathbb{R}^{(k+1) \times k}$ with $r_{k+1,k} = 0$ such that $Q_k^T R_k = \widehat{H}_k$, where Q_k is a matrix obtained as the product of k Givens rotations. Then, since Q_k is orthogonal, it can be seen that minimizing $\| \| \mathbf{r}^{(0)} \|_2 \mathbf{e}_1 - \widehat{H}_k \mathbf{z}^{(k)} \|_2$ is equivalent to minimize $\| \mathbf{f}_k - R_k \mathbf{z}^{(k)} \|_2$, with $\mathbf{f}_k = Q_k \| \mathbf{r}^{(0)} \|_2 \mathbf{e}_1$. It can also be shown that the $k + 1$ -th component of \mathbf{f}_k is, in absolute value, the Euclidean norm of the residual at the k -th step.

As FOM, the GMRES method entails a high computational effort and a large amount of memory, unless convergence occurs after few iterations. For this reason, two variants of the algorithm are available, one named GMRES(m) and based on the *restart* after m steps, with $\mathbf{x}(m)$ as initial guess, the other named Quasi-GMRES or QGMRES and based on stopping the Arnoldi orthogonalization process. It is worth noting that these two methods do not enjoy Property 7.3.

The convergence analysis of GMRES is not trivial, and we report just some of the more elementary results here. If A is positive definite, i.e., its symmetric part A_S has positive eigenvalues, then the k -th residual decreases according to the following bound

$$\| \mathbf{r}^{(k)} \|_2 \leq \sin^k(\beta) \| \mathbf{r}^{(0)} \|_2 , \quad (7.36)$$

where $\cos(\beta) = \lambda_{\min}(L_S)/\| L \|$ with $\beta \in [0, \pi/2]$. Moreover, GMRES(m) converges for all $m \geq 1$. In order to obtain a bound on the residual at a step $k \geq 1$, let us assume that the matrix A is diagonalizable

$$A = T \Lambda T^{-1} ,$$

where Λ is the diagonal matrix of eigenvalues, $\{\lambda_j\}_{j=1,\dots,n}$, and $T = [\omega^1, \dots, \omega^n]$ is the matrix whose columns are the right eigenvectors of A . Under these assumptions, the residual norm after k steps of GMRES satisfies

$$\|\mathbf{r}^{(k)}\| \leq K_2(T)\delta\|\mathbf{r}^{(0)}\|,$$

where $K_2(T) = \|T\|_2\|T^{-1}\|_2$ is the condition number of T and

$$\delta = \min_{p \in \mathbb{P}_k, p(0)=1} \max_{1 \leq i \leq k} |p(\lambda_i)|.$$

Moreover, suppose that the initial residual is dominated by m eigenvectors, i.e., $\mathbf{r}^0 = \sum_{j=1}^m \alpha_j \omega^j + \mathbf{e}$, with $\|\mathbf{e}\|$ small in comparison to $\|\sum_{j=1}^m \alpha_j \omega^j\|$, and assume that if some complex ω^j appears in the previous sum, then its conjugate $\bar{\omega}^j$ appears as well. Then

$$\|\mathbf{r}^{(k)}\| \leq K_2(T)c_k\|\mathbf{e}\|,$$

$$c_k = \max_{p > k} \prod_{j=1}^k \left| \frac{\lambda_p - \lambda_j}{\lambda_j} \right|.$$

Very often, c_k is of order one; hence, k steps of GMRES reduce the residual norm to the order of $\|\mathbf{e}\|$ provided that $\kappa_2(T)$ is not too large.

In general, as highlighted from the previous estimate, the eigenvalue information alone is not enough, and information on the eigensystem is also needed. If the eigensystem is orthogonal, as for normal matrices, then $K_2(T) = 1$, and the eigenvalues are descriptive for convergence. Otherwise, upper bounds for $\|\mathbf{r}^{(k)}\|$ can be provided in terms of both spectral and pseudospectral information, as well as the so-called *field of values* of A

$$\mathcal{F}(A) = \{\mathbf{v}^* A \mathbf{v} \mid \|\mathbf{v}\| = 1\}.$$

If $0 \notin \mathcal{F}(A)$, then the estimate (7.36) can be improved by replacing $\lambda_{\min}(A_S)$ with $\text{dist}(0, \mathcal{F}(A))$.

An extensive discussion of convergence of GMRES and GMRES(m) can be found in [Saa96], [Emb99], [Emb03], [TE05], and [vdV03].

The GMRES method can of course be implemented for a preconditioned system. We provide here an implementation of the preconditioned GMRES method with a left preconditioner P .

Preconditioned GMRES (PGMRES) Method. Initialize

$$\mathbf{x}^{(0)}, P\mathbf{r}^{(0)} = \mathbf{f} - A\mathbf{x}^{(0)}, \beta = \|\mathbf{r}^{(0)}\|_2, \mathbf{x}^{(1)} = \mathbf{r}^{(0)}/\beta.$$

Iterate

```

For  $j = 1, \dots, k$  Do
    Compute  $P\mathbf{w}^{(j)} = A\mathbf{x}^{(j)}$ 
    For  $i = 1, \dots, j$  Do
         $g_{ij} = (\mathbf{x}^{(i)})^T \mathbf{w}^{(j)}$ 
         $\mathbf{w}^{(j)} = \mathbf{w}^{(j)} - g_{ij}\mathbf{x}_i$ 
    End Do
     $g_{j+1,j} = \|\mathbf{w}^{(j)}\|_2$ 
    (if  $g_{j+1,j} = 0$  set  $k = j$  and Goto (1))
     $\mathbf{x}^{(j+1)} = \mathbf{w}^{(j)}/g_{j+1,j}$ 
End Do
 $V_k = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)}]$ ,  $\hat{H}_k = \{g_{ij}\}$ ,  $1 \leq j \leq k$ ,  $1 \leq i \leq j + 1$ ;
(1) Compute  $\mathbf{z}^{(k)}$ , the minimizer of  $\|\beta\mathbf{e}_1 - \hat{H}_k\mathbf{z}\|$ 
Set  $\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + V_k\mathbf{z}^{(k)}$ 

```

(7.37)

More generally, as proposed by Saad (1996), a variable preconditioner P_k can be used at the k -th iteration, yielding the so-called *flexible GMRES* method. The use of a variable preconditioner is especially interesting in those situations where the preconditioner is not explicitly given, but implicitly defined, for instance, as an approximate Jacobian in a Newton iteration or by a few steps of an inner iteration process (see Chap. 15). Another meaningful case is the one of domain decomposition preconditioners (of either Schwarz or Schur type) where the preconditioning step involves one or several substeps of local solves in the subdomains (see Chap. 17).

Several considerations for the practical implementation of GMRES, its relation with FOM, how to restart GMRES, and the Householder version of GMRES can be found in [Saa96].

Remark 7.1 (Projection methods) Denoting by Y_k and L_k two generic m -dimensional subspaces of \mathbb{R}^n , we call *projection method* a process which generates an approximate solution $\mathbf{x}^{(k)}$ at step k , enforcing that $\mathbf{x}^{(k)} \in Y_k$ and that the residual $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$ be orthogonal to L_k . If $Y_k = L_k$, the projection process is said to be *orthogonal*, *oblique* otherwise (see [Saa96]).

The Krylov subspace iterations can be regarded as being projection methods. For instance, the Arnoldi method (see [Saa96]) is an orthogonal projection method where $L_k = Y_k = K_k(A; \mathbf{r}^{(0)})$, while the GMRES method is an oblique projection method with $Y_k = K_k(A; \mathbf{r}^{(0)})$ and $L_k = AY_k$. It is worth noticing that some classical methods introduced in previous sections fall into this category. For example, the Gauss-Seidel method is an orthogonal projection method where at the k -th step $K_k(A; \mathbf{r}^{(0)}) = \text{span}\{\mathbf{e}_k\}$, with $k = 1, \dots, n$. The projection steps are carried out cyclically from 1 to n until convergence. •

Elements of finite element programming

In this chapter, we focus more deeply on a number of aspects relating to the translation of the finite element method into computer code. This *implementation* process can hide some pitfalls. Beyond the syntactic requirements of a given programming language, the need for a high computational efficiency implementation requires a codification that is generally not the immediate translation of what has been seen during theoretical presentation. Efficiency depends on many factors, including the language used and the architecture on which one works. Personal experience can play a role as fundamental as learning from a textbook. Moreover, although spending time searching for a bug in the code or for a more efficient data structure can sometimes appear to be a waste of time, it (almost) never is. For this reason, we wish to propose the present chapter as a sort of “guideline” for trials that the reader can perform on his own, rather than a chapter to be studied in the traditional sense.

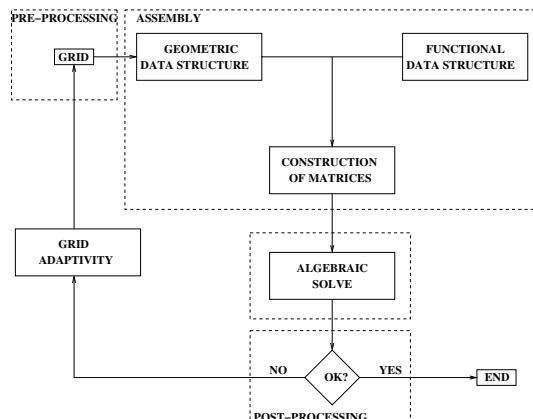


Fig. 8.1. Operational phases of a finite element code

A final note concerns the cut of the chapter: the approach followed here is to provide a *general* kind of information: obviously, each problem has specific features that can be exploited in a careful way for a yet more efficient implementation.

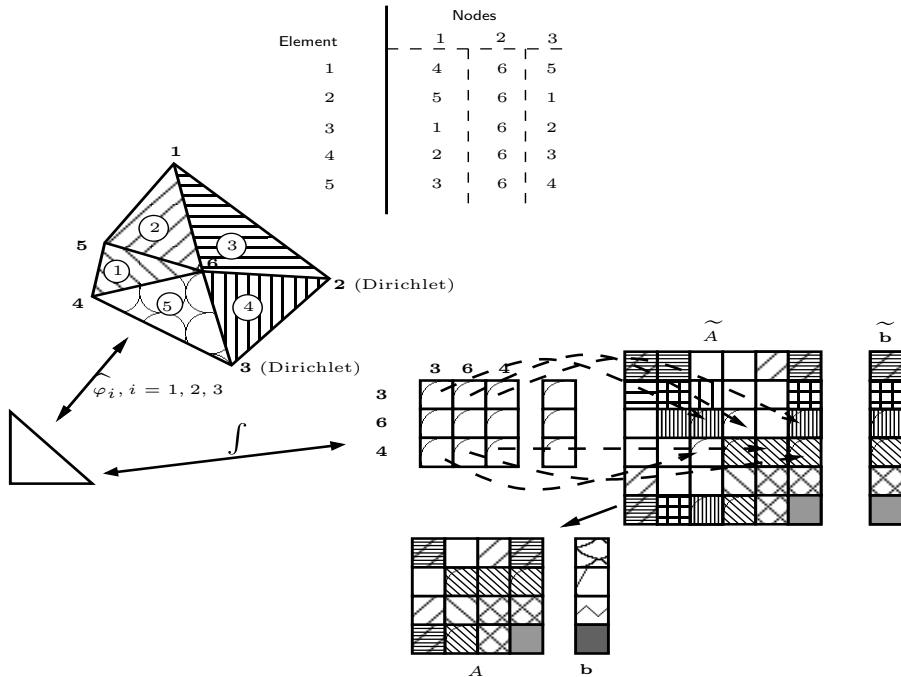


Fig. 8.2. Schematization of the assembly. The geometric and topological information (top table), suitably stored, describes the grid. Through the mapping on the reference element, we perform the computation of the discretization matrix \tilde{A} and of the known term \tilde{b} , first by proceeding element by element (local computation) and then, exploiting the additivity of the integration operation, we update the global matrix. The symbols representing each element of the matrix are obtained through the overlap of the symbols used to define each element of the mesh. Finally, we operate the prescription of the boundary conditions, which ideally remove the degrees of freedom with Dirichlet conditions, getting to the final matrix A and to the known term b . As we will see, the operation is often implemented in a different way

8.1 Operational phases of a finite element code

We can distinguish four cases in the execution of a finite element computation, which represent as many coding phases (Fig. 8.1).

1. *Pre-processing.* This phase consists in setting the problem and coding its computational domain, which, as seen in Chap. 4, requires the construction of the *mesh*

(or grid). In general, setting aside the trivial cases (for instance in one dimension), the construction of an adequate mesh is a numerical problem of considerable interest, for which *ad hoc* techniques have been developed. Generally, this operation is performed by dedicated programs or modules within a solver, where recently great effort has been devolved to the aspect of interface and interfacing with CAD (*Computer Aided Design*) software. Chap. 6 is dedicated to the fundamental techniques for grid generation.

2. *Assembly.* In this phase, we construct the “functional” data structures, starting from the “geometric” ones obtained by the mesh and by the user’s choices concerning the desired type of finite elements to be used. Moreover, based on the problem we want to solve and on its boundary conditions, we compute the stiffness matrix associated to the discretization (see Chap. 4 and 11). This operation can optionally be inserted within a temporal progress cycle if we are handling time-dependent problems (as done in Chap. 5-15) and can also be the outcome of a linearization operation in the case where we are handling nonlinear problems. In a strict sense, the term “assembly” refers to the construction of the matrix of the linear system, moving from the local computation performed on the reference element to the global one that concurs to determining the matrix associated to the discretized problem.

Fig. 8.2 summarizes the different operations during the assembly phase for the preparation of the algebraic system.

3. *Solution of the algebraic system.* The core of the solution of any finite element computation is represented by the solution of a linear system. As previously said, this will eventually be part of a temporal cycle (based on an implicit discretization method) or of an iterative cycle due to the linearization of a nonlinear problem. The choice of the solution method is generally left to the user. For this reason, it is very important that the user joins on the one hand the knowledge of the problem under exam, which, as we have seen in Chap. 4, is reflected on the structure of the matrix (for instance, symmetry and positivity), and on the other a good knowledge of the available methods to perform an optimal choice (which rarely is the default one). This is why in Chap. 7 we have recalled the main properties of numerical methods for the solution of linear systems.

Nowadays, several very efficient computational libraries exist for the solution of various types of linear systems, hence the trend in the coding phase is generally to include such libraries rather than implementing from scratch. For instance, in the following examples, the solution of linear systems is given to Aztec, a library developed at the Sandia National Laboratories in Albuquerque, New Mexico, USA (see [AZT]). However, many other libraries exist for this purpose, among which we recall PetSC (see [Pet]) and UMFPACK [UMF], TriLinos [Tri].

4. *Post-processing.* Since the amount of numerical data generated by a finite element code might be huge, their post-processing is often necessary in order to present concise results and in a usable format. However, the synthesis via images may be a non-trivial phase. In particular, an incautious post-processing for the a posteriori computation of derivative quantities (e.g. stresses from displacements, fluxes or vorticity from velocities, etc ...) can introduce unacceptable overhead errors.

Since grid generation techniques have already been addressed in Chap. 6, and the algorithms for the solution of linear systems in Chap. 7, the main focus of this chapter will be on the *Assembly* phase (Sec. 8.4).

Before dealing with this subject, in Sec. 8.2 we will deal with the coding of the quadrature formula for the numerical computation of integrals, while sparse matrix coding is discussed in Sec. 8.3.

As far as the *Post-processing* phase is concerned, we refer to specific literature, by observing that the techniques used above have been previously introduced in Chap. 4 for the computation of a posteriori estimates.

Sec. 8.6 finally reports a complete example.

8.1.1 Code in a nutshell

There are many programming languages and environments available today, characterized by different philosophies and objectives. When facing the implementation of a numerical method, it is necessary to operate a motivated choice in these respects. Amongst the most useful programming environments for the construction of prototypes, Matlab is certainly an excellent tool under many viewpoints, although, as for all interpreted languages, it is weaker under the computational efficiency profile. Another environment targeted to the solution of differential problems in 2D through the finite element method is FreeFem ++ (see www.freefem.org). This environment comprises all of the four phases indicated above in a single package (free and usable under different operating systems), with a particularly captivating syntax, which reduces the distance between coding and theoretical formulation, and in particular significantly approaches the former to the latter. This operation has a clear “educational” merit, which is to quickly produce simulations also for non trivial problems. However, the computational costs and the difficulty of implementing new strategies that require an extension of the syntax can result to be penalizing in actual cases of interest.

Traditionally, among compiled programming languages, Fortran (Fortran 77 in particular) is the one that had the most success in the numerical domain, thanks to the fact that it generates very efficient executable code. Lately, the abstraction feature that is intrinsic to the object-oriented programming philosophy has proven to be very suitable for finite element programming. The level of abstraction enabled by the breadth of application of mathematical tools seems to find an excellent counterpart in the abstraction of object oriented programming, based on the design of data types made by the user (more than on operations to perform, as in procedural programming) and on their polymorphism (see e.g. [LL00, CP00, Str00]). However, the computational cost of such an abstraction has sometimes reduced the interest for a theoretically captivating programming style, but often operationally weak for scientific kinds of problems where computational efficiency is (almost) always crucial. This has required the development of more sophisticated programming techniques (for instance, Expression Templates), allowing to avoid the interpretation cost of abstract objects to become too heavy during code execution (see e.g. [Vel95, Fur97, Pru06, DV08]). Hence, besides Fortran, languages like C++, born as an object-oriented improvement of the C language, are nowadays more and more frequent in the scientific domain;

amongst others, we recall Diffpack and FoamCFD. We will therefore refer to C++ in the code excerpts presented below. In particular, these excerpts are part of a wide library, LifeV (Life 5), developed at the CMCS research center at the Swiss Federal Institute of Technology in Lausanne, at the INRIA at Rocquencourt, Paris and at the MOX at Politecnico in Milan. This library, freely available at www.lifev.org under the general LGPL licensing, is an open code to new contributions in different application contexts (mainly in 3D) for the application of recent numerical methods in the context of advanced object-oriented programming.

An accurate examination of the code (which we will henceforth call “Programs” for simplicity) requires some basic knowledge of C++, for which we refer to [LL00]. However, as we want to use the current chapter as a basis for autonomous experiments (using one’s own favorite programming language) it is not essential to have a full knowledge of the C++ syntax to understand the text, but rather it is sufficient to be familiar with its basic syntax constructs.

8.2 Numerical computation of integrals

The effective numerical computation of the integrals in the finite element formulation is typically performed via the application of *quadrature formulae*. For an introduction to the subject of numerical quadrature, we refer to basic numerical analysis textbooks (e.g. [QSS07]). Here, it will suffice to recall that a generic quadrature formula has the form

$$\int_K f(\mathbf{x}) d\mathbf{x} \approx \sum_{iq=1}^{nqn} f(\mathbf{x}_{iq}) w_{iq}$$

where K denotes the region over which we integrate (typically an element of the finite element grid), nqn is the number of quadrature nodes for the selected formula, \mathbf{x}_{iq} are the coordinates of the *quadrature nodes* and w_{iq} are the *weights*. Typically, the accuracy of the formula and its computational cost grow with the number of quadrature nodes. As we will see in Chap. 10, Sec. 10.2.2 and 10.2.3, the formulae which guarantee the best accuracy for the same number of nodes, are the *Gaussian* ones.

The integral computation is generally performed on the reference element, on which the expression of base functions is known, through a suitable change of variable (Sec. 4.3). If we denote the coordinates of the reference space by $\hat{\mathbf{x}}_i$, integration in the reference space will require knowing the Jacobian matrix J of the geometric transformation, defined as

$$J = [J_{ij}] = \left[\frac{\partial x_i}{\partial \hat{x}_j} \right].$$

Indeed, we have

$$\int_K f(\mathbf{x}) d\mathbf{x} = \int_{\hat{K}} f(\hat{\mathbf{x}}) |J|(\hat{\mathbf{x}}) d\hat{\mathbf{x}} \approx \sum_{iq} \hat{f}(\hat{\mathbf{x}}_{iq}) |J|(\hat{\mathbf{x}}_{iq}) w_{iq} \quad (8.1)$$

where $|J|$ denotes the determinant of J and the $\{\hat{\mathbf{x}}_{iq}\}$ are the quadrature nodes on \hat{K} .

In the case of operators where the spatial derivative intervenes, it is necessary to replicate the derivation rules of composite functions:

$$\frac{\partial f}{\partial x_i} = \sum_{j=1}^d \frac{\partial f}{\partial \hat{x}_j} \frac{\partial \hat{x}_j}{\partial x_i}, \quad \nabla f = J^{-T} \nabla_{\hat{x}} \hat{f} = \frac{1}{|J|} \text{cof}(J) \nabla_x \hat{f},$$

where $\text{cof}(J)$ is the cofactor matrix of J . For instance, for the computation of the stiffness matrix, we have

$$\begin{aligned} \int_K \nabla \varphi_\alpha \nabla \varphi_\beta d\mathbf{x} &= \sum_{i,j} \int_K \frac{\partial \varphi_\alpha}{\partial x_i} \frac{\partial \varphi_\beta}{\partial x_j} d\mathbf{x} = \sum_{i,j,l,m} \int_K \frac{\partial \hat{\varphi}_\alpha}{\partial \hat{x}_l} \frac{\partial \hat{x}_l}{\partial x_i} \frac{\partial \hat{\varphi}_\beta}{\partial \hat{x}_m} \frac{\partial \hat{x}_m}{\partial x_j} |J| d\mathbf{x} \\ &\approx \sum_{i,j,l,m} \sum_{iq} \frac{\partial \hat{\varphi}_\alpha}{\partial \hat{x}_l} J_{li}^{-1} \frac{\partial \hat{\varphi}_\beta}{\partial \hat{x}_m} J_{mj}^{-1} |J| (\hat{\mathbf{x}}_{iq}) w_{iq}, \end{aligned} \quad (8.2)$$

α and β being the indexes of two generic basis functions φ_α and φ_β on K ; $\hat{\varphi}_\alpha$ and $\hat{\varphi}_\beta$ are the corresponding basis functions on \hat{K} . Note the presence of the elements of the inverse Jacobian matrix J^{-1} , due to the coordinate transformation intervening in the computation of the basis function derivatives.

The class coding a quadrature formula stores quadrature nodes and their associated weights. In the effective integral computation, we will then obtain the necessary mapping information for the actual computation, which depend on the geometry of K .

Program 2 reports the code for a 5-point quadrature formula for tetrahedra:

$$\begin{aligned} \hat{\mathbf{x}}_1 &= \left(\frac{1}{6}, \frac{1}{6}, \frac{1}{6} \right), & w_1 &= \frac{9}{20} \frac{1}{6} \\ \hat{\mathbf{x}}_2 &= \left(\frac{1}{6}, \frac{1}{6}, \frac{1}{2} \right), & w_2 &= \frac{9}{20} \frac{1}{6} \\ \hat{\mathbf{x}}_3 &= \left(\frac{1}{6}, \frac{1}{2}, \frac{1}{6} \right), & w_3 &= \frac{9}{20} \frac{1}{6} \\ \hat{\mathbf{x}}_4 &= \left(\frac{1}{2}, \frac{1}{6}, \frac{1}{6} \right), & w_4 &= \frac{9}{20} \frac{1}{6} \\ \hat{\mathbf{x}}_5 &= \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4} \right), & w_5 &= -\frac{16}{20} \frac{1}{6}. \end{aligned}$$

The factor $1/6$, appearing in the expression of the weights w_i represents the volume of the reference tetrahedron. Often, the tabbed weights in the books do not explicitly account for this factor, hence, in our case, we find the values $9/20$ and $-16/20$, but the size of the reference element must not be forgotten!

Program 2 - pt-tetra-5pt: Five-node quadrature formula on a tetrahedron: class QuadPoint defines the single quadrature node with the associated weight. The quadrature formula will be defined by an array of QuadPoint

```

class QuadPoint {
Real _coor[ 3 ];
Real _weight;
public: QuadPoint(Real x, Real y, Real z, Real weight )
{ _coor[ 0 ] = x;
  _coor[ 1 ] = y;
  _coor[ 2 ] = z;
  _weight = weight;
}
}

//Integration on Tetrahedron with a 5 node formula
const Real tet5ptx1 = 1. / 6., tet5ptx2 = 1. / 2., tet5ptx3 = 1. / 4.;

static const QuadPoint pt_tetra_5pt[ 5 ] =
{ QuadPoint( tet5ptx1,tet5ptx1, tet5ptx1, 9. / 120. ),
  QuadPoint( tet5ptx1,tet5ptx1, tet5ptx2, 9. / 120. ),
  QuadPoint( tet5ptx1, tet5ptx2, tet5ptx1, 9. / 120. ),
  QuadPoint( tet5ptx2, tet5ptx1, tet5ptx1, 9. / 120. ),
  QuadPoint( tet5ptx3, tet5ptx3, tet5ptx3, -16. / 120. )
};

```

The choice of a quadrature formula responds to two (conflicting) needs:

1. on one hand, the higher is the accuracy, the smaller the error generated by the computation of the integrals is controlled; for problems whose differential operator has constant (or polynomial) coefficients, by leveraging the notion of *exactness degree* of a quadrature formula, we can get to the point of completely vanishing the numerical integration error;
2. on the other hand, an increase of accuracy implies an increase in the number of n_{qn} nodes.

The appropriate synthesis of these two needs evidently depends on the requirements of the problem we want to solve, as well as on the accuracy and speed specifications for the computation to execute.

8.2.1 Numerical integration using barycentric coordinates

The numerical evaluation of integrals on simplices (intervals in 1D, triangles in 2D, tetrahedra in 3D) can take advantage from the use of the barycentric coordinates that were introduced in Sect. 4.4.3. To start with, we observe that the following exact integration formulae hold (see, e.g., [Aki94, Chap. 9] or [Hug00, Chap. 3])

$$\int_{\hat{K}_d} \prod_{i=0}^d \lambda_i^{n_i} d\omega = \frac{\prod_{i=0}^d n_i!}{(\sum_{i=0}^d n_i + d)!} d! |\hat{K}_d| \quad (8.3)$$

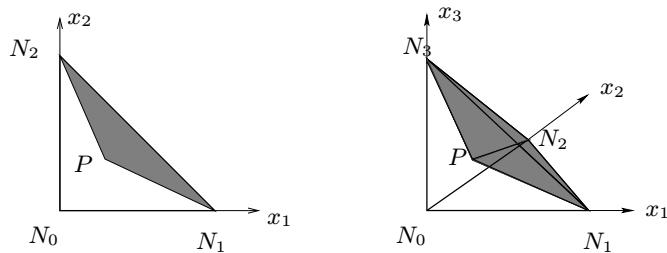


Fig. 8.3. The barycentric coordinate λ_i of point P represents the ratio between the volume of the tetrahedron having as vertexes P itself and the vertexes of the face opposite to N_i (in the figure, right, we have shadowed the tetrahedron opposite N_0 with vertices P, N_1, N_2, N_3) and the total volume of the tetrahedron

where \hat{K}_d is a d-dimensional unitary simplex, $|\hat{K}_d|$ denotes its measure, $\{n_i, 0 \leq i \leq d\}$ is a set of non-negative integers.

These formulae are useful when dealing with finite element approximation of boundary-value problems for exact computation of polynomial integrals in the characteristic Lagrangian basis functions.

For the sake of an example, Table 8.1 reports the weights and nodes for some popular quadrature formulae in 2D. Table 8.2 reports some formulae for tetrahedra. These formulae are symmetric: we must consider all possible permutations of the barycentric coordinates to obtain the full list of nodes.

For the reader's convenience, we report, next to the total number of nqn nodes, the multiplicity m of each quadrature node, i.e. the number of nodes generated by the permutations. We also provide the exactness degree r , that is the largest positive integer r for which all polynomials of degree $\leq r$ are integrated exactly by the quadrature formula at hand.

Table 8.1. Nodes and weights for the quadrature formulae on triangles. The nodes are expressed through their barycentric coordinates. The weights do not account for the measure of the reference element (which is equal to 1/2)

nqn	barycentric coordinates			m	w_j	r
1	1/3	1/3	1/3	1	1	1
3	1	0	0	3	1/3	1
3	2/3	1/3	1/3	3	1/3	1
4	1/3	1/3	1/3	1	-0.5625	2
	0.6	0.2	0.2	3	0.52083	
6	0.65902762237	0.23193336855	0.10903900907	6	1/6	2
6	0.81684757298	0.09157621351	0.09157621351	3	0.10995174366	3
	0.10810301817	0.44594849092	0.44594849092	3	0.22338158968	

Let us see two simple examples. Suppose we want to compute:

$$I = \int_K f(x, y) d\mathbf{x} = \int_{\widehat{K}} \widehat{f}(\widehat{x}, \widehat{y}) |J|(\widehat{x}, \widehat{y}) d\widehat{\mathbf{x}}.$$

Using the weights and nodes of the first row of the table, we obtain:

$$I \simeq \frac{1}{2} \widehat{f}\left(\frac{1}{3}, \frac{1}{3}\right) J\left(\frac{1}{3}, \frac{1}{3}\right) = \text{Area}(K) f(\bar{x}, \bar{y}),$$

where the coefficient $1/2$ represents the area of the reference element, \bar{x} is the node with barycentric coordinates $\lambda_1 = \lambda_2 = \lambda_3 = 1/3$ and corresponding to the center of gravity of the triangle. Hence, the corresponding formula is the well known *composite midpoint formula*.

To use the formula in the second row, we note that $m = 3$, hence we have indeed 3 quadrature nodes whose barycentric coordinates are obtained via cyclic permutation:

$$(\lambda_0 = 1, \lambda_1 = 0, \lambda_2 = 0), (\lambda_0 = 0, \lambda_1 = 1, \lambda_2 = 0), (\lambda_0 = 0, \lambda_1 = 0, \lambda_2 = 1).$$

Hence, for each triangle K we obtain

$$\begin{aligned} \int_K f(\mathbf{x}) d\mathbf{x} &\simeq \frac{1}{2} \frac{1}{3} \left[\widehat{f}(0, 0) |J|(0, 0) + \widehat{f}(1, 0) |J|(1, 0) + \widehat{f}(0, 1) |J|(0, 1) \right] \\ &= \text{Area}(K) \sum_{i=0}^2 \frac{1}{3} f(\mathbf{N}_i). \end{aligned}$$

$\mathbf{N}_0, \mathbf{N}_1, \mathbf{N}_2$, being the vertices of the K triangle, corresponding to the barycentric coordinates $(0,0)$, $(1,0)$ and $(0,1)$ respectively. The corresponding formula therefore yields the *composite trapezoidal formula*. Both formulae have exactness degree equal to 1.

Table 8.2. Nodes and weights for quadrature formulae on tetrahedra. The nodes are expressed using their barycentric coordinates. The weights do not account for the measure of the reference element (equal to 1/6)

nqn	barycentric coordinates				m	w_j	r
1	1/4	1/4	1/4	1/4	1	1	1
4	0.58541020	0.13819660	0.13819660	0.13819660	4	1/4	2
5	1/4	1/4	1/4	1/4	1	-16/20	3
	1/2	1/6	1/6	1/6	4	9/20	

Other quadrature formulae for the computation of integrals for different finite elements can be found in [Com95], [Hug00], [Str71].

Remark 8.1 When using quadrilateral or prismatic elements, nodes and weights of the quadrature formulae can be obtained as the tensor product of the Gauss quadrature formulae for the one-dimensional interval, see Sec. 10.2 (see also [CHQZ06]). •

8.3 Storage of sparse matrices

As seen in Chap. 4, finite element matrices are sparse. The distribution of non-null elements is reported in the so-called sparsity pattern (also called *graph*) of the matrix. The pattern depends on the adopted computational grid, on the chosen finite element type and on the numbering of the nodes. The efficient storage of a matrix therefore consists in a storage of its non-null elements, according to the positioning denoted by the pattern. The discretization of different differential problems, sharing however on the same computational grid and the same type of finite elements, leads to matrices with the same graph. Hence, in an object-oriented programming logic, it can be useful to separate the storage of the graph (which can become a “data type” defined by the user, i.e. a class) from the storage of the values of each matrix. This way, a matrix can be seen as a data structure for the storage of its values, together with a pointer to the graph associated to it. The pointer only stores the position in the memory where the pattern is stored, hence occupying a minimal amount of memory. Different matrices may therefore share the same graph, without useless storage duplications of the pattern (see Programs 3 and 4).

In practice, there are several techniques to efficiently store sparse matrices, i.e. the position and value of their non-null elements. It is appropriate to observe that, in this context, the adjective “efficient” does not only refer to the lower memory occupation that can be achieved, but also to the memory access speed for each element. A storage format requiring the least possible memory waste is likely to be slower in accessing a desired value. Indeed, the higher storage compactness is typically obtained after getting its position in the memory by accessing the data structures storing the graph. The more intermediate passages are necessary, the longer the access time to the desired element will be. Precisely for the need of finding the right compromise, different storage techniques have been proposed in the literature, with different prerogatives. A commented review of these can be found e.g. in [FSV05], Appendix B. We here limit ourselves to recalling a widely used format for the storage of sparse square matrices, i.e. the MSR (*Modified Sparse Row*) format. The sparsity graph of a square matrix generated by the discretization of a problem via finite elements has the property that diagonal elements are always a priori comprised between non-null elements, for the trivial reason that the support of a base function has non-empty intersection with itself. The MSR format is based on this consideration to store only the pattern of the extra-diagonal part, then using another vector to store the values of the main diagonal, ordered according to their row.

In practice, to store the matrix, we use two vectors, which we call *value* and *bindx* (row binding). To the latter, for reasons given in [FSV05], we propose to add a third vector which we call *bindy* (column binding). We denote by n the size of the matrix to be stored and by nz the number of its non-null elements.

To illustrate the MSR format we use an example (see Fig. 8.4) where $n = 5$ and $nz = 17$:

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ a & 0 & f & 0 & g \\ 0 & b & k & m & 0 \\ h & l & c & 0 & r \\ 0 & n & 0 & d & p \\ i & 0 & s & q & e \end{bmatrix} \quad \begin{array}{l} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array}$$

We point out that the numbering of rows and columns in matrices and vectors starts from 0, following C++ syntax. The vectors characterizing the MSR format are:

$$\text{value} = \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ * \\ f \\ g \\ k \\ m \\ h \\ l \\ r \\ n \\ p \\ i \\ s \\ q \end{bmatrix} \quad \begin{array}{l} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \end{array} \quad \text{bindx} = \begin{bmatrix} 6 \\ 8 \\ 10 \\ 13 \\ 15 \\ 18 \\ 2 \\ 4 \\ 2 \\ 3 \\ 0 \\ 1 \\ 4 \\ 1 \\ 4 \\ 0 \\ 2 \\ 3 \end{bmatrix}$$

The `bindx` vector is:

$$\text{bindy} = \begin{bmatrix} 10 \\ 15 \\ 11 \\ 13 \\ 6 \\ 8 \\ 16 \\ 9 \\ 17 \\ 7 \\ 12 \\ 14 \end{bmatrix} \quad \begin{array}{l} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array}$$

Let us see how these vectors are structured. At the n first positions of `value` (indexed from 0 to $4 = n - 1$) we store the diagonal values of the matrix. The index position n is left empty, while from the positions from $n + 1$ on we store the values of extradiagonal elements, ordered row by row. The overall size of `value` will therefore

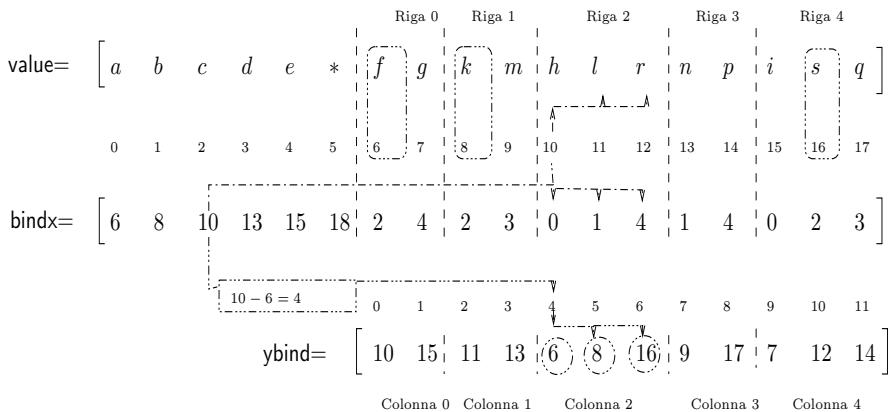


Fig. 8.4. The structures of the MSR format: the arrows denote the path to the determination of the third row and third column elements

be $nz + 1$. The bindx vector also has $nz + 1$ elements. The elements at positions $n + 1$ to $nz + 1$ contain the column indexes of the respective elements whose value is stored in value at the same position. The first $n + 1$ positions of bindx point at where the rows at positions indexed from $n + 1$ to $nz + 1$ start.

If for instance we want to access the third row elements (see Fig. 8.4), we have that:

- the diagonal element is at $\text{value}(2)$;
- the extra-diagonal elements are comprised at the positions indexed by $\text{bindx}(2)$ and $\text{bindx}(3) - 1$, or
 - the values are at $\text{value}(\text{bindx}(2)), \text{bindx}(2) + 1), \dots, \text{value}(\text{bindx}(3) - 1)$;
 - the column indexes are at $\text{bindx}(\text{bindx}(2)), \text{bindx}(\text{bindx}(2) + 1), \dots, \text{bindx}(\text{bindx}(3) - 1)$.

The $\text{bindx}(n)$ element points to a hypothetic row following the last one. This way, the number of non-null elements of the i -th row (for $1 \leq i \leq n$) is given without exceptions (meaning without needing to introduce conditional branches) by the difference $\text{bindx}(i + 1) - \text{bindx}(i)$.

If the matrix was stored using these two vectors only, we would have an easy access by row (i.e. it is easy to extract one row), while access by column would require several comparisons, in spite of efficiency. A way to make this operation easier is to enrich the MSR format of bindy. We exploit the feature of finite element matrices which is to have a symmetric pattern. Indeed, the property that two basis functions have non-disjoint support is evidently “symmetric”. This means that, if we go through the extra-diagonal elements of the row of index k and we find that if the coefficient a_{kl} is present in the pattern (i.e. it is not null), also a_{lk} will be present, which is obtained by browsing the row at index l . If the position at bindx (and value) of element a_{lk}

is stored in a “twin” vector of the portion of bindx that ranges from index $n + 1$ to nz , we have a structure that returns the elements of a desired column. Such vector is bindy: for instance, to extract the column at index 2 from the matrix it is sufficient to read the elements of bindy located between the positions $\text{bindx}(2) - (n + 1)$ and $\text{bindx}(3) - 1 - (n + 1)$ (the subtraction of index $n + 1$ only serves as shift between the indexes to which bindx points in value and those where it has to point in bindy). These elements point to the positions of bindx and value where we can find, respectively, the corresponding row indexes and the matrix values.

As the MSR format is one of the most “compact” formats for sparse matrices, it allows to save memory and is therefore used in several linear algebra libraries for large scale problems, such as Aztec (see [AZT]). However, it has the drawback of only being usable for square matrices. For more details, see [FSV05], [Saa96].

Programs 3 and 4 report the data structure (i.e. the private members) of the MSRPatt and MSRMatr classes.

Program 3 - BasePattern: Basic structure to store the pattern of the matrix in MSR format

```
class BasePattern : public PatternDefs

public:
...
protected:
...
    UInt _nnz;
    UInt _nrows;
    UInt _ncols;
;

class MSRPatt : public BasePattern

public:
...
    const Container& bindx() const return _bindx; ;
    const Container& bindy() const return _bindy; ;
;
```

Program 4 - MSRMatr: Matrices in MSR format

```
template <typename DataType>
class MSRMatr

public:
...
private:
    std::vector<DataType> _value;
    const MSRPatt *_Patt;
;
```

8.4 Assembly phase

By assembly phase we actually mean the articulation of different operations that lead to the construction of the matrix associated to the discretized problem. To this end, we need two types of information:

1. *geometric*, typically contained in the mesh file;
2. *functional*, relative to the representation of the solution via finite elements.

Functional types of information are as wide as the different types of elements that are coded. LifeV treats Lagrangian and non-lagrangian finite elements, both continuous and discontinuous. In particular, for continuous finite elements, we have:

1. Lagrangian finite elements in 1D of degree 1 and 2;
2. Lagrangian finite elements in 2D;
 - a) triangular with linear and quadratic functions;
 - b) quadrilateral with bilinear and biquadratic functions;
3. Lagrangian finite elements in 3D;
 - a) tetrahedric with linear and quadratic functions;
 - b) prismatic with bilinear and biquadratic functions.

In Fig. 8.5 we report the main reference geometries covered in the code with their local vertex numbering. Tetrahedra represent the 3D extension of the triangular elements

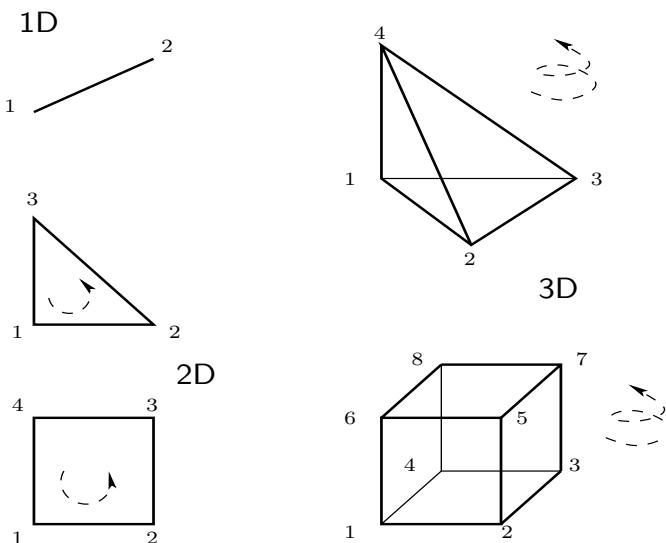


Fig. 8.5. Illustration of some reference elements available in LifeV with (conventional) local numbering of nodes

considered in Chap. 4. Prismatic elements extend in 3D the quadrilateral geometric elements in 2D which will be introduced in Chap. 10. A complete description of the construction of this kind of elements can be found in [Hug00], Chap. 3.

Geometric and functional information, suitably coded, is then used to construct the matrix of the discretized problem. As opposed to what would seem natural in the definition of Lagrangian finite elements, the construction of the matrix happens by cycling on the elements instead of nodes. The reason for this *element-oriented* approach, rather than the *node-oriented* one, is essentially linked to computational efficiency matters. The analytical expression of a base function associated to a node varies on each element sharing that node. Before the computation of the integrals, it would be necessary to cycle over the nodes and detect the analytical expression of the appropriate basis functions to each element. Hence, it would be necessary to cycle on the nodes and locate the analytic expression of the appropriate basis function for each different element, before carrying out the computation of the integrals. In terms of code, this means that the body of the cycle must be filled with *conditional branches*, that is instructions of the type if...then...elseif...then...else... within the assembly cycle. The latter are “costly” operations in computational terms, especially if they lie within a cycle (and thus are carried out several times). To realize this, it is sufficient to observe that the number of micro assembler instructions required in the compilation phase to expand a conditional branch, with respect to any other instruction, see e.g. [HVZ97]. As we will see, the element-oriented approach, exploiting the additivity of the integration operation, allows to brilliantly bypass this obstacle.

In particular, as previously pointed out in Chap. 4, the construction of the problem matrix can take place in two conceptual steps, within a cycle on the grid elements:

1. construction of the matrix and of the known term that discretize the differential operator on the element at hand (*local* matrix and vector);
2. update of the *global* matrix and known term, by exploiting the additivity of the integration operation.

There also exist different approaches to the problem: in some cases, the matrix is not constructed, but its effects are directly computed in the multiplication by a vector, as happens when the linear system is solved by iterative methods; for reasons of space, we here deal with the more standard approach.

As previously noted, the construction of the local matrix is carried out by integrating on the reference element \hat{K} , by using suitable quadrature formulae. Once the matrix and known terms have been constructed, boundary conditions are imposed; in particular, the prescription of Dirichlet conditions does not necessarily use the technique seen in Sec. 3.2.2 and 4.5, consisting of the removal of degrees of freedom associated to such conditions after the construction of the lifting.

As it can be seen, assembly is an articulated phase. In the following paragraphs we will discuss the mentioned aspects, however in little detail for reasons of space. First, we will treat the data structures for the coding of geometric (Sec. 8.4.1) and functional (Sec. 8.4.2) information. The computation of the geometric mapping between reference element and current element provides the opportunity of introducing *isoparametric elements* (Sec. 8.4.3). The effettive computation of local matrix and

known term and their use in the construction of the global system is treated in Sec. 8.4.4. Finally, in Sec. 8.4.5 we refer to implementation techniques for the lifting of the boundary datum.

8.4.1 Coding geometrical information

In terms of data structures, the *mesh* can be seen as a collection of geometric elements and topological information. The former can be constructed by aggregating classes for the definition of points (i.e. zero-dimensional geometric elements), edges (one-dimensional geometric elements), faces (2D) and finally volumes (3D). A possible interface for the coding of these geometric entities, limited to the case of points and volumes, is provided in Program 5.

Program 5 - GeoElements: Elementary and aggregate classes for the construction of geometric entities

```
//! Class for Points and Vertexes
class GeoElement0D

public:
    GeoElement0D();

    GeoElement0D( ID id, Real x, Real y, Real z, bool boundary = false );
    GeoElement0D & operator = ( GeoElement0D const & g );
;

// Class for 3D Elements
template<typename GEOSHAPE>
class GeoElement3D

public:
    GeoElement3D( ID id = 0 );

    typedef GeoElement1D<EdgeShape, MC> EdgeType;
    typedef GeoElement2D<FaceShape, MC> FaceType;
    typedef GeoElement0D PointType;
    typedef FaceType GeoBElement;

    //! Number of Vertexes per element
    static const UInt numLocalVertices;
    //! Number of Faces per element
    static const UInt numLocalFaces;
    //! Number of edges per element (Euler rule)
    static const UInt numLocalEdges;
;
```

The class defining geometric entities, which is here presented in a significantly simplified form with respect to LifeV, provides methods (which do not report for reasons of space) allowing to query the structure to obtain interesting information, such as the identifier within a list of the identifiers of adjacent geometrical entities. This is very important in defining the mesh connection, and therefore the matrix pattern.

The definition of standard types for geometric entities (denoted by the term GEOSHAPE in the code above) can be made through the introduction of suitable classes denoting the geometric structure composing the volumes of the mesh. For instance, in Program 6 we illustrate the class for tetrahedra.

Program 6 - Tetra: Class for the coding of tetrahedric elements

```
class Tetra

public:
    static const ReferenceShapes Shape = TETRA;
    static const ReferenceGeometry Geometry = VOLUME;
    static const UInt nDim = 3;
    static const UInt numVertices = 4;
    static const UInt numFaces = 4;
    static const UInt numEdges = numFaces + numVertices - 2;
;
```

Starting from these base classes, a mesh will be a class collecting the elements. In fact, it is mandatory to add the following to the geometric structure:

1. topological information allowing to characterize the elements in the grid, i.e. the connectivity among nodes, with respect to a conventional numbering of the latter. The convention for the possible elements in LifeV is illustrated in Fig. 8.5; to efficiently “visit” the elements of a grid, we can also add information of the adjacent elements to each given element;
2. specific information allowing to locate the degrees of freedom on the boundary; this simplifies the handling of the boundary condition prescription; we observe that we typically associate to each boundary geometric element an indicator that will later be associated to a specific boundary condition.

Starting from the reference geometric class, we then code the current geometric elements, according to the possible mappings treated in Sec. 8.4.3. For instance, if the mapping is affine, we obtain linear tetrahedra, as seen in Program 7.

Program 7 - LinearTetra: Class for the coding of tetrahedra obtained via affine geometric transformation of the reference element

```
class LinearTetra:
    public Tetra

public:
    typedef Tetra BasRefSha;
    typedef LinearTriangle GeoBShape;
```

```

static const UInt numPoints = 4;
static const UInt nbPtsPerVertex = 1;
static const UInt nbPtsPerEdge = 0;
static const UInt nbPtsPerFace = 0;
static const UInt nbPtsPerVolume = 0;
;
```

At this point, the code containing the member of the class identifying the mesh is in Program 8.

Program 8 - RegionMesh3D: Class for the storage of a 3D mesh

```

template<typename GEOSHAPE>
class RegionMesh3D

public:
    explicit RegionMesh3D();

//Definition of the base profiles
typedef GEOSHAPE VolumeShape;
typedef typename GEOSHAPE::GeoBShape FaceShape;
typedef typename FaceShape::GeoBShape EdgeShape;

//Geometric entities
typedef GeoElement3D<GEOSHAPE> VolumeType;
typedef GeoElement2D<FaceShape> FaceType;
typedef GeoElement1D<EdgeShape> EdgeType;
typedef GeoElement0D PointType;

//Vector of the points
typedef SimpleVect<PointType> Points;
//Vector of the volumes
typedef SimpleVect<VolumeType > Volumes;
//Vector of the boundary faces
typedef SimpleVect<FaceType> Faces;
//Vector of the edges
typedef SimpleVect<EdgeType> Edges;

typedef GEOSHAPE ElementShape;
typedef typename GEOSHAPE::GeoBShape BElementShape;
typedef GeoElement3D<GEOSHAPE> ElementType;
typedef GeoElement2D<FaceShape> BElementType;

typedef SimpleVect<VolumeType > Elements;

Points _pointList;
Volumes volumeList;
Faces faceList;
Edges edgeList;
```

```

UInt numLocalVertices() const; //Number of vertexes per element
UInt numLocalFaces() const; //Number of faces per element
UInt numLocalEdges() const; //Number of edges per element
UInt numLocalEdgesOfFace() const; //Number of edges per face

UInt numElements() const; //Total number of volumes
UInt & numElements();
UInt numBElements() const; //Number of boundary elements (=faces)
UInt & numBElements();
ElementType & element( ID const & i );
ElementType const & element( ID const & i ) const;
BElementType & bElement( ID const & i );
BElementType const & bElement( ID const & i ) const;
;

;

```

The vector of geometric elements of type “volume” declared in `Volumes` `volumeList` will contain for instance the list of vertexes that define each tetrahedron in the mesh, in the conventional order established and illustrated in Fig. 8.5.

The construction of a container for an affine tetrahedron mesh will be performed via the instruction

```
RegionMesh3D<LinearTetra> aMesh;
```

which will be followed by the reading of the Mesh to effectively fill the vectors of volumes, faces, edges and points declared in `RegionMesh3D`.

As far as the format of a mesh file is concerned, there is no universally accepted standard. Typically, we expect such a file to contain the vertex coordinates, the connectivity associating the vertexes to the geometric elements and the list of boundary elements with corresponding indicator to be used for defining boundary conditions. The values of the boundary conditions, instead, are generally assigned separately.

Remark 8.2 Multi-physics or multi-model problems are becoming a relevant component of scientific computation: think for instance of fluid-structure interaction problems or of the coupling of problems where the full (and computationally more costly) differential model is used only in a specific region of interest, by coupling it with simpler models in the remaining regions. These applications and, more generally, the need to develop parallel computation algorithms, have motivated the development of techniques for the solution of differential problems through *domain decomposition*. The interested reader can consult, e.g., [QV99], [TW05]. In this case, the resulting mesh is the collection of subdomain meshes, together with topological information about subdomain interfaces. In this textbook, for simplicity, we will refer in any case to single-domain problems only. •

8.4.2 Coding of functional information

As seen in Chap. 4, the definition of basis functions is performed on a reference element. For instance, for tetrahedra, this element coincides with the unit simplex (see Fig. 8.5). The coding of a reference element will basically include pointers to functions for determining basis functions and their derivatives. Moreover, it will be possible to enrich it by a pointer to the quadrature formula used in the computation of integrals (see Sec. 8.2), as in Program 9.

Program 9 - RefEle: Class for the storage of functional information on the reference element

```
class RefEle

protected:
    const Fct* _phi; //Pointer to basis functions
    const Fct* _dPhi; //Pointer to basis function derivatives
    const Fct* _d2Phi; //Pointer to basis function second derivatives
    const Real* _refCoor; //Reference coord: xi_1,eta_1,zeta_1,xi_2,eta_2,zeta_2, ...
    const SetOfQuadRule* _sqr; //Pointer to the set of quadrature formulae

public:
    const std::string name; //Name of the reference element
    const ReferenceShapes shape; //Geometric form of the element
    const int nbDof; //Total number of degrees of freedom
    const int nbCoor; //Number of local coordinates
;
```

In Program 10 we report the functions for the definition of linear finite elements on tetrahedra. For reasons of space, we report the code for only part of the first derivatives.

Program 10 - fctP13D: Basis functions for a linear tetrahedric element

```
Real fct1_P1_3D( cRRef x, cRRef y, cRRef z )return 1 -x - y - z;
Real fct2_P1_3D( cRRef x, cRRef, cRRef )return x;
Real fct3_P1_3D( cRRef, cRRef y, cRRef )return y;
Real fct4_P1_3D( cRRef, cRRef, cRRef z )return z;

Real derfct1_1_P1_3D( cRRef, cRRef, cRRef )return -1;
Real derfct1_2_P1_3D( cRRef, cRRef, cRRef )return -1;
...
```

Once the reference element is instantiated, functional information will be available both for the representation of the solution and for the definition of the geometric mapping between reference element and current element, as we see in the following section.

Having defined the geometric element and the type of finite elements we want to use, we are now able to construct the problem's degrees of freedom. This means assigning to each mesh element the numbering of degrees of freedom lying on the

element and the pattern of the local matrix; the latter is generally full, although it can contain null elements in any case.

A degree of freedom can require additional information, such as, in the case of Lagrangian elements, the coordinates of the node corresponding to the reference element.

8.4.3 Mapping between reference and physical element

In Chap.4 we have seen how convenient it is to write basis functions, quadrature formulae and, therefore, to compute the integrals with respect to a reference element. It can thus be interesting to examine some practical methods to construct and code such coordinate change. For further details, we refer to [Hug00]. Let us now limit ourselves to considering the case of triangular and tetrahedric elements.

A first type of coordinate transformation is the *affine* one. Basically, the mapping between $\hat{\mathbf{x}}$ and \mathbf{x} can be expressed via a matrix B and a vector \mathbf{c} (see Sec.4.5.3 and Fig. 8.6)

$$\mathbf{x} = B\hat{\mathbf{x}} + \mathbf{c}. \quad (8.4)$$

This way, we trivially have that $J = B$ (constant on each element). If the node distribution generated by the grid generator is correct, the determinant of J is always positive, which guarantees there are no degenerate cases (for instance, four vertexes on the same plane in a tetrahedron) and that there are no incorrect permutations in the nodes corresponding to the mapping. The expressions of B and \mathbf{c} can be obtained from those of the node coordinates. Indeed, let us suppose that the nodes numbered *locally* 1,2,3,4 of the reference tetrahedron correspond to the nodes of the mesh numbered as i, k, l, m , respectively.

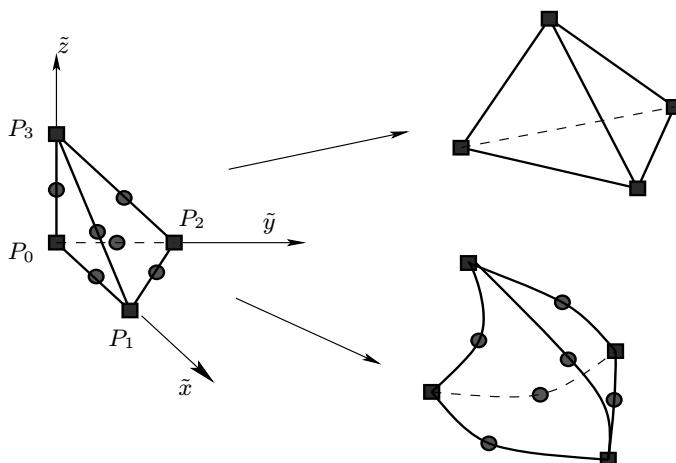


Fig. 8.6. Mapping between the reference tetrahedron and the current one. At the top right, an affine mapping; at the bottom right, a quadratic mapping

We then have:

$$\begin{cases} x_i = c_1 & y_i = c_2 & z_i = c_3 \\ x_k = b_{11} + x_i & y_k = b_{12} + y_i & z_k = b_{13} + z_i \\ x_l = b_{21} + x_i & y_l = b_{22} + y_i & z_l = b_{23} + z_i \\ x_m = b_{31} + x_i & y_m = b_{32} + y_i & z_m = b_{33} + z_i \end{cases} \quad (8.5)$$

from which we obtain the expressions for B and \mathbf{c} .

However, there exists a more efficient way to represent the transformation: being element-wise linear, the latter can be represented via the basis functions of linear Lagrangian finite elements. Indeed, we can write:

$$x = \sum_{j=0}^3 X_j \hat{\varphi}_j(\hat{x}, \hat{y}, \hat{z}), \quad y = \sum_{j=0}^3 Y_j \hat{\varphi}_j(\hat{x}, \hat{y}, \hat{z}), \quad z = \sum_{j=0}^3 Z_j \hat{\varphi}_j(\hat{x}, \hat{y}, \hat{z}). \quad (8.6)$$

The elements of the Jacobian matrix of the transformation are immediately computed:

$$J = \begin{bmatrix} \sum_{j=1}^4 X_j \frac{\partial \hat{\varphi}_j}{\partial \hat{x}} & \sum_{j=1}^4 X_j \frac{\partial \hat{\varphi}_j}{\partial \hat{y}} & \sum_{j=1}^4 X_j \frac{\partial \hat{\varphi}_j}{\partial \hat{z}} \\ \sum_{j=1}^4 Y_j \frac{\partial \hat{\varphi}_j}{\partial \hat{x}} & \sum_{j=1}^4 Y_j \frac{\partial \hat{\varphi}_j}{\partial \hat{y}} & \sum_{j=1}^4 Y_j \frac{\partial \hat{\varphi}_j}{\partial \hat{z}} \\ \sum_{j=1}^4 Z_j \frac{\partial \hat{\varphi}_j}{\partial \hat{x}} & \sum_{j=1}^4 Z_j \frac{\partial \hat{\varphi}_j}{\partial \hat{y}} & \sum_{j=1}^4 Z_j \frac{\partial \hat{\varphi}_j}{\partial \hat{z}} \end{bmatrix}. \quad (8.7)$$

When in a Lagrangian finite element the same basis functions are used for the definition of the geometric mapping, we say that we are dealing with *iso-parametric* elements (see Fig. 8.6 and 8.7). Obviously, the coincidence holds because we have chosen linear finite elements and affine geometric transformations. When we consider finite elements of degree higher than 1, we can consider two kinds of mapping:

- affine finite elements: in this case, the geometric transformation is still described by the affine transformations (8.6), although the functional information relative to the solution is described by quadratic functions of higher degree; the boundary of the discretized domain Ω_h , in this case, is still polygonal (polyhedric);
- isoparametric finite elements: the geometric transformation is described by the same basis functions used to represent the solution; hence the elements in the physical space $Oxyz$ will generally have curved sides;

The definition of a quadratic mapping starting from the tetrahedric reference element allows for instance to create tetrahedric quadratic geometric elements, coded in the class `QuadraticTetra` reported in Program 11.

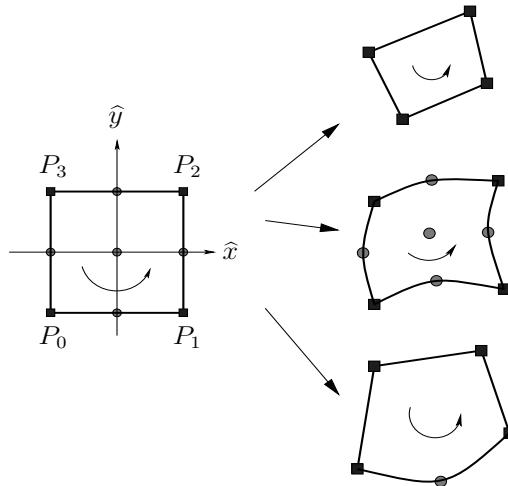


Fig. 8.7. Mapping between the reference quadrilateral and the current element: affine (top), isoparametric (center), hybrid (bottom). The latter is constructed with 5 nodes, in order to have a biquadratic transformation for the nodes of a single side

Program 11 - QuadraticTetra: Class for the definition of quadratic tetrahedric elements

```
class QuadraticTetra: public Tetra

public:
    typedef Tetra BasRefSha;
    typedef QuadraticTriangle GeoBShape;
    static const UInt numPoints = 10;
    static const UInt nbPtsPerVertex = 1;
    static const UInt nbPtsPerEdge = 1;
    static const UInt nbPtsPerFace = 0;
    static const UInt nbPtsPerVolume = 0;
;
```

Having established the type of reference element and the geometrical mappings, it is possible to construct the collection of “current” elements. The current element can be coded as in Program 12.

Program 12 - CurrentFE: Class for the definition of the current element

```
class CurrentFE

private:
    void _comp_jacobian();
    void _comp_jacobian_and_det();
```

```

void _comp_inv_jacobian_and_det();
void _comp_quad_point_coor();

template <class GEOELE>
void _update_point( const GEOELE& geole ) ;

//! compute phiDer
void _comp_phiDer();
//! compute the second derivative phiDer2
void _comp_phiDer2();
//! compute phiDer and phiDer2
void _comp_phiDerDer2();

UInt _currentId;
public:
    CurrentFE( const RefFE& _refFE, const GeoMap& _geoMap, const QuadRule& _qr );
    const int nbGeoNode;
    const int nbNode;
    const int nbCoor;
    const int nbQuadPt;
    const int nbDiag;
    const int nbUpper;
    const int nbPattern;
    const RefFE& refFE;
    const GeoMap& geoMap;
    const QuadRule& qr;
};

;

```

As it can be seen, the class contains information relating to the reference element, to the geometric mapping that generates it and to the quadrature formula that will be used for the computation of the integrals.

In particular, (8.7) proves to be very efficient in the coding phase, which we report in Program 13. It must be noticed how the computation of the jacobian is carried out at the quadrature nodes required for the integral computation (Sec. 8.2).

Program 13 - comp-jacobian: Section of the class storing the current elements which computes the jacobian of the transformation between current element and reference element

```

void CurrentFE::_comp_jacobian()

Real fctDer;
// GeoMap derivatives:
for ( int ig = 0;ig < nbQuadPt;ig++ )

    for ( int icoor = 0;icoor < nbCoor;icoor++ )

```

```

for ( int jcoor = 0;jcoor < nbCoor;jcoor++ )

    fctDer = 0.;

    for ( int j = 0;j < nbGeoNode;j++ )

        fctDer += point( j, icoor ) * dPhiGeo( j, jcoor, ig );

    jacobian( icoor, jcoor, ig ) = fctDer;

```

In the case of quadrilateral and prismatic elements, several of the previously seen concepts can be extended, by referring e.g. to bilinear or biquadratic mappings. However, it proves more complex to guarantee the invertibility of the mapping: for more details, see [FSV05].

There are cases where it can be convenient to use finite elements of different degree with respect to different coordinates. This is possible using quadrilateral structured grids, where it is possible to construct an element on which we have a biquadratic polynomial on one side, while on the other sides we have bilinear polynomials. In the case of an isoparametric coding of the geometrical mapping, this leads to having, e.g., quadrilateral elements with three straight sides and one curved sides. To this end, we point out that [Hug00], Chap. 4 reports the “incremental” implementation of a quadrilateral element that, starting from a four-node bilinear setting, is enriched by other degrees of freedom until the biquadratic 9-node element.

8.4.4 Construction of local and global systems

This phase is the core of the construction of the discretization of differential operators. As an example, let us take the code in Program 14, which constructs the discretization of the elliptic differential equation $\mu\Delta u + \sigma u = f$.

The overall operation is articulated in a cycle over all the elements of the mesh `aMesh`. After setting to zero the elementary matrix and vector, these structures are filled incrementally, first with the discretization of the stiffness (diffusion) operator and then with the mass one (reaction). The source subroutine handles the local known term vector. Then, the `assemb` subroutines handle the update of the computation in the global matrix, as previously indicated in Fig. 8.2.

In this phase, to avoid checking whether a degree of freedom is on the boundary through conditional branches within the loop, we ignore boundary conditions.

Program 14 - assemble: Code for assembling the discretization of a diffusion-reaction problem $\mu\Delta u + \sigma u = f$, where f is denoted by `sourceFct`

```

Real mu=1., sigma=0.5;
ElemMat elmat(fe.nbNode,1,1);
ElemVec elvec(fe.nbNode,1);
for(UInt i = 1; i<=aMesh.numVolumes(); i++)
    fe.updateFirstDerivQuadPt(aMesh.volumeList(i));
    //<- computes the necessary information for numerical integration

```

```

elmat.zero();
elvec.zero();
stiff(mu,elmat,fe);
mass(sigma,elmat,fe);
source(sourceFct,elvec,fe,0);
assemb_mat(A,elmat,fe,dof,0,0);
assemb_vec(F,elvec,fe,dof,0);

```

Let us see separately in detail a possible implementation of the local computation and of the global update.

Computation of the local matrices. Program 15 reports the implementation of the computation of the local matrix of the diffusion operator and of the known term.

In particular, we first assemble the diagonal contributions and then the extra-diagonal ones of the local matrix, thus looping over the quadrature nodes. The “core” loop operation is:

```

s += fe.phiDer( iloc, icoor, ig ) * fe.phiDer( jloc, icoor, ig )
    * fe.weightDet( ig )*coef;

```

The instruction

```
mat( iloc, jloc ) += s;
```

updates the term i, j of the local matrix incrementally: upon call of the following subroutine `mass()`, the contribution of the reaction operator will be summed to the previously computed one.

We proceed in a similar way in `source` for the computation of the local vector of known terms.

Program 15 - stiff-source: Subroutines for the computation of the second derivative and local-level known term computation

```

void stiff( Real coef,
            ElemMat& elmat, const CurrentFE& fe,
            const Dof& dof,
            const ScalUnknown<Vector>& U,Real t)

int iblock=0,jblock=0;
ElemMat::matrix_view mat = elmat.block( 0,0 ); //initialize local matrix
int iloc, jloc, i, icoor, ig, iu;
double s, coef_s, x, y, z;
ID eleId=fe.currentId();

// Diagonal elements
for ( i = 0;i < fe.nbDiag;i++ )

iloc = fe.patternFirst( i );s = 0;
for ( ig = 0;ig < fe.nbQuadPt;ig++ ) // numerical integration

```

```

fe.coorQuadPt(x,y,z,ig); // definition of the quadrature formula
for ( icoor = 0; icoor < fe.nbCoor; icoor++ ) // core of the assembly
  s += fe.phiDer( iloc, icoor, ig ) * fe.phiDer( iloc, icoor, ig )
    * fe.weightDet( ig ) * coef(t,x,y,z,uPt);

  mat( iloc, iloc ) += s;

//Extra-diagonal elements
for ( i = fe.nbDiag; i < fe.nbDiag + fe.nbUpper; i++ )

  iloc = fe.patternFirst( i );
  jloc = fe.patternSecond( i ); s = 0;
  for ( ig = 0; ig < fe.nbQuadPt; ig++ )

    fe.coorQuadPt(x,y,z,ig);
    for ( icoor = 0; icoor < fe.nbCoor; icoor++ )
      s += fe.phiDer( iloc, icoor, ig ) * fe.phiDer( jloc, icoor, ig ) *
        fe.weightDet( ig ) * coef;

  coef_s = s;
  mat( iloc, jloc ) += coef_s; //incremental
  mat( jloc, iloc ) += coef_s; //local matrix update
  // recall that the operator is SYMMETRIC!

void source( Real (*fct)(Real,Real,Real,Real,Real),
            ElemVec& elvec, const CurrentFE& fe,
            const Dof& dof,
            const ScalUnknown<Vector>& U,Real t)

int iblock=0;
int i, ig;
ElemVec::vector_view vec = elvec.block( iblock );
Real s;
ID eleId=fe.currentId();
int iu;

for ( i = 0; i < fe.nbNode; i++ )

  s = 0.0;
  for ( ig = 0; ig < fe.nbQuadPt; ig++ )

    s += fe.phi( i, ig ) *
      fct(t, fe.quadPt( ig, 0 ), fe.quadPt( ig, 1 ), fe.quadPt( ig, 2 )) *
        fe.weightDet( ig );

  vec( i ) += s; //known term computation

```

Update of the global matrix. Program 16 contains the update of the global matrix starting from the local ones. The crucial point is the identification of the position of the nodes that compose the current element, on which we have just computed the local matrix within the global one. This operation is performed by looking up the dof.localToGlobal Tables, which contain this type of operation.

For the known term update, we perform a similar operation. Obviously, the additivity of the integral requires the operation to be performed, gradually adding the different contributions: this explains the $+ =$ in the update of the vector (corresponding to $V[ig] = V[ig] + \text{vec}(i)$) and to the analogous one in M.setmatinc which stands for *set matrix incrementally*.

Program 16 - assemb: Assembly of the global matrix and of the known term

```
template <typename Matrix, typename DOF>
void
assemb_mat( Matrix& M, ElemMat& elmat, const CurrentFE& fe, const DOF& dof)

ElemMat::matrix_view mat = elmat.block(0,0);
UInt totdof = dof.numTotalDof();
int i, j, k;
UInt ig, jg;
UInt eleld = fe.currentId();
for ( k = 0 ; k < fe.nbPattern ; k++ )

    i = fe.patternFirst( k );
    j = fe.patternSecond( k );
    ig = dof.localToGlobal( eleld, i + 1 ) - 1;
    jg = dof.localToGlobal( eleld, j + 1 ) - 1;
    M.set_mat_inc( ig, jg, mat( i, j ) );

template <typename DOF, typename Vector, typename ElemVec>
void
assemb_vec( Vector& V, ElemVec& elvec, const CurrentFE& fe, const DOF& dof)

UInt totdof = dof.numTotalDof();
typename ElemVec::vector_view vec = elvec.block( iblock );
int i;
UInt ig;
UInt eleld = fe.currentId();
for ( i = 0 ; i < fe.nbNode ; i++ )

    ig = dof.localToGlobal( eleld, i + 1 ) - 1;
    V[ ig ] += vec( i );
```

8.4.5 Boundary conditions prescription

The need to efficiently store sparse matrices must be compensated by the need to access and manipulate the matrix itself, as we have previously noticed for the modified MSR format, for instance in the phase of setting the boundary layers. In a finite element code, the matrix is typically assembled regardless of boundary conditions, so as not to introduce conditional branches within the assembly cycle. Boundary conditions are then introduced by modifying the algebraic system. The setting of Neumann and Robin-type conditions basically translates into the computation of suitable boundary integrals (or, in one-dimensional cases, of values evaluated at the boundary). For instance, Program 17 implements the computation of integrals on the surface for Neumann-type conditions specified in function `Bcb`. The integral implies using a suitable quadrature formula allowing to update the known term `b`. The structure `bdLocalToGlobal` allows to transfer the information for each boundary element having Neumann degrees of freedom at the global known term.

Program 17 - `BcNaturalManage`: Subroutine for handling Neumann-type boundary conditions

```
template <typename VectorType, typename MeshType, typename DataType>
void bcNaturalManage( VectorType& b, const MeshType& mesh, const Dof& dof,
                      const BCBBase& BCb, CurrentBdFE& bdfem, const DataType& t )

UInt nDofF = bdfem.nbNode;
UInt totalDof = dof.numTotalDof();
UInt nComp = BCb.numberOfComponents();

const IdentifierNatural* pld;
ID ibF, idDof, icDof, gDof;
Real sum;

DataType x, y, z;
// Loop on the type of boundary conditions
for ( ID i = 1; i <= BCb.list_size(); ++i )

    pld = static_cast< const IdentifierNatural*>( BCb( i ) );
    // Number of current boundary face
    ibF = pld->id();
    // definition of information on the face
    bdfem.updateMeas( mesh.boundaryFace( ibF ) );
    // Loop on degrees of freedom per face
    for ( ID idofF = 1; idofF <= nDofF; ++idofF )

        // Loop on the involved unknown components
        for ( ID j = 1; j <= nComp; ++j )

            //global Dof
```

```

idDof = pld->bdLocalToGlobal( idofF ) + ( BCb.component( j ) - 1 ) * totalDof;
// Loop on quadrature nodes
for ( int l = 0; l < bdFem.nbQuadPt; ++l )

    bdFem.coorQuadPt( x, y, z, l ); // quadrature point coordinates
    // Contribution in the known term
    b[ idDof - 1 ] += bdFem.phi( int( idofF - 1 ), l ) * BCb( t, x, y, z, BCb.component( j ) ) *
    bdFem.weightMeas( l );

```

The handling of Dirichlet (essential) boundary conditions is more complex (see Fig. 8.2). There are various strategies for this operation, some of which are treated in [FSV05]. The most coherent approach to what is prescribed by the theory consists in removing the rows and columns relating to the nodes associated to the Dirichlet boundary conditions from the system obtained during assembly, correcting the known term by using the values of the Dirichlet datum we want to impose.

In fact, this coincides with the operation of lifting of the boundary datum through a piecewise polynomial function of the chosen degree for the finite elements via which we approximate the problem, and whose support is limited to the only layer of elements of the triangulation that face the boundary (see Fig. 4.13 in Sec. 8.4).

This way to proceed has the advantage of reducing the dimension of the problem to the effective number of degrees of freedom, however its practical implementation is problematic. Indeed, while for 1D problems, due to the natural ordering of degrees of freedom, the optional rows and columns to be removed are always the first and last one, for multi-dimensional problems it involves eliminating rows and columns whose numbering can be arbitrary, a difficult operation to handle efficiently. It must also be noted that this operation substantially modifies the *pattern* of the matrix, and this can be inconvenient in case we want to share it among several matrices in order to save memory. For this reason, we prefer to consider the Dirichlet condition to be imposed at a given node k_D as an equation of the form $u_{k_D} = g_{k_D}$ to replace the k_D -th row of the original system. To avoid modifying the matrix pattern, this substitution must be inserted by vanishing the extra-diagonal row elements, except for the diagonal one, which is set to 1, while the known term is set to g_{k_D} .

This operation only requires a row-wise access to the matrix, hence value and bindx would be sufficient. However, in such a way we would compromise the possible symmetry of the matrix. To preserve it (for instance, in order to use the conjugate gradient iterative method, or to execute a Cholesky decomposition) it is necessary to modify the columns of the matrix and the known term as well.

To this end, we may think of also modifying the k_D -th column by setting its elements (excluding the diagonal one) to zero and updating the known term in an appropriate way. From an implementation viewpoint, the “elimination” of Dirichlet degrees of freedom is basically replaced by an alteration of the matrix coefficients that however leaves its pattern unchanged. To perform this, it also requires an easy column-wise access to the matrices which, as we have seen, is enabled by vector bindy.

The effects of imposing boundary conditions through this approach are evident in Fig. 8.8. A possible implementation of this technique is reported in Program 18.

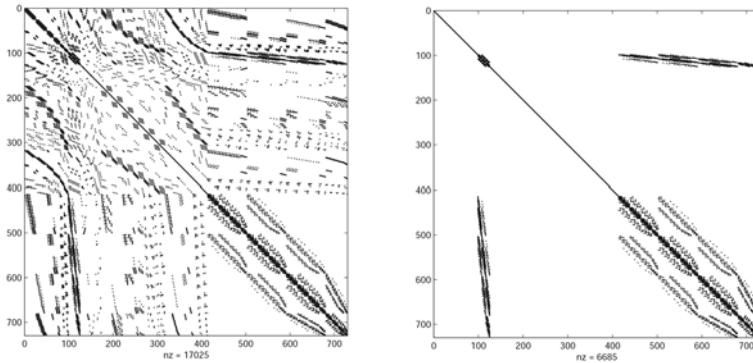


Fig. 8.8. Effects of the treatment of Dirichlet conditions on a real life case of a 3D grid via vanishing of the extra-diagonal elements of the rows and columns associated to Dirichlet degrees of freedom: before (left) and after (right) the imposition of boundary conditions

Program 18 - diagonalize: Treatment of the Dirichlet boundary conditions through vanishing of the extra-diagonal coefficients of the rows and columns associated to Dirichlet conditions

```
template <typename DataType>
void
MSRMat<DataType>::diagonalize( UInt const r, std::vector<DataType> &b,
                                 DataType datum )

    _value[r] = 1.;

    UInt istart = *(_Patt->give_bidx().begin() + r);
    UInt iend = *(_Patt->give_bidx().begin() + r + 1);

    typename std::vector<DataType>::iterator start = _value.begin() + istart;
    typename std::vector<DataType>::iterator end = _value.begin() + iend;
    UInt disp = _Patt->nRows() + 1;
    UInt row, col;

    transform( start, end, start, zero );

    for ( UInt i = istart;i < iend;++i )

        row = _Patt->give_bidx() [ i ];
        col = _Patt->give_bindy() [ i - disp ];
        b[ row ] -= _value[ col ] * datum;
        _value[ col ] = 0.;

        b[ r ] = datum;

    return ;
```

8.5 Integration in time

Among the different methods to perform the integration in time, we have analyzed the θ method in the previous chapters, but we have also pointed out a number of other methods, in particular BDF (*Backward Difference Formulas*) methods implemented in LifeV. An introduction of these methods can be found in [QSS07]. We here recall some of their basic aspects.

Given the system of ordinary differential equations:

$$M \frac{d\mathbf{u}}{dt} = \mathbf{f} - A\mathbf{u}$$

and the associated initial datum $\mathbf{u}(t = 0) = \mathbf{u}_0$, a BDF method is an implicit multi-step method of the form

$$\frac{\alpha_0}{\Delta t} M \mathbf{U}^{n+1} + A \mathbf{U}^{n+1} = \mathbf{f}^{n+1} + \sum_{j=1}^p \frac{\alpha_j}{\Delta t} \mathbf{U}^{n+1-j}, \quad (8.8)$$

for suitable $p \geq 1$, where the coefficients are determined so that:

$$\frac{\partial \mathbf{U}}{\partial t} \Big|_{t=t^{n+1}} = \frac{\alpha_0}{\Delta t} \mathbf{U}^{n+1} - \sum_{j=1}^p \frac{\alpha_j}{\Delta t} \mathbf{U}^{n+1-j} + \mathcal{O}(\Delta t^p).$$

Here, $\Delta t > 0$ is the time-step, $t^n = n\Delta t$, and \mathbf{U}^n stands for \mathbf{U} at time t^n . In Table 8.3 (left) we report the coefficients for $p = 1$ (implicit Euler method), $p = 2, 3$.

Table 8.3. α_i coefficients for the BDF methods ($p = 1, 2, 3$) and β_i coefficients for time extrapolation

p	α_0	α_1	α_2	α_3	β_0	β_1	β_2
1	1	1	–	–	1	–	–
2	3/2	2	-1/2	–	2	-1	–
3	11/6	3	-3/2	1/3	3	-3	1

In the case where matrix A is a function of \mathbf{u} , that is when problem (8.8) is nonlinear, BDF methods, being implicit, can result to be very costly, requiring at each time step the solution of the nonlinear algebraic system in \mathbf{U}^{n+1}

$$\frac{\alpha_0}{\Delta t} M \mathbf{U}^{n+1} + A(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} = \mathbf{f}^{n+1} + \sum_{j=1}^p \frac{\alpha_j}{\Delta t} \mathbf{U}^{n+1-j}.$$

A possible trade-off that significantly reduces computational costs, without moving to a completely explicit method (whose stability properties can in general be unsatisfying) is to solve the linear system

$$\frac{\alpha_0}{\Delta t} M \mathbf{U}^{n+1} + A(\mathbf{U}^*) \mathbf{U}^{n+1} = \mathbf{f}^{n+1} + \sum_{j=1}^p \frac{\alpha_j}{\Delta t} \mathbf{U}^{n+1-j}.$$

where \mathbf{U}^* approximates \mathbf{U}^{n+1} using the solutions known since the previous steps. We basically set

$$\mathbf{U}^* = \sum_{j=0}^p \beta_j \mathbf{U}^{n-j} = \mathbf{U}^{n+1} + \mathcal{O}(\Delta t^p),$$

for suitable “extrapolation” coefficients β_j . The objective is to reduce the computational costs without dramatically reducing neither the region of absolute stability of the implicit scheme nor the overall accuracy of the time advancing scheme. Table 8.3 on the right reports the β_j coefficients.

The coding of a BDF time integrator can at this point be performed using a dedicated class, reported in Program 19, whose members are:

1. the indicator of the order p which also regulates the α and β vectors;
2. vectors α and β ;
3. the unknowns matrix given by aligning the vectors $\mathbf{U}^n, \mathbf{U}^{n-1}, \dots, \mathbf{U}^{n+1-p}$. The size of each vector, i.e. the number of rows of such matrix (which has p columns) is stored in the size index.

Having assembled matrices A and M , the time advancing scheme will be performed by computing the matrix $\frac{\alpha_0}{\Delta t} M + A$, the known term $\mathbf{f}^{n+1} + \sum_{j=1}^p \frac{\alpha_j}{\Delta t} \mathbf{U}^{n+1-j}$ and solving the system (8.8). In particular, in the implementation presented in Program 19, function `time_der` computes the term $\sum_{j=1}^p \frac{\alpha_j}{\Delta t} \mathbf{U}^{n+1-j}$ by accessing the *alpha* vector and the unknowns matrix. In case the problem is nonlinear, we can resort to the β vector via the `extrap()` function.

Having computed the solution at the new time step, the unknowns matrix has to “make room for it”, by shifting all of its columns to the right, so that the first column is the solution computed just now. This operation is performed by function `shift_right`, which basically copies the next-to-last column of `unknowns` into the last one, the one-but-last one into the next-to-last one and so on until storing the solution just computed.

Program 19 - Bdf: Base class for costructing BDF time integration methods

```
class Bdf

public:
    Bdf( const UInt p );
    ~Bdf();
    void initialize_unk( Vector u0 );
    void shift_right( Vector const& u_curr );

    Vector time_der( Real dt ) const;
    Vector extrap() const;
    double coeff_der( UInt i ) const;
    double coeff_ext( UInt i ) const;
    const std::vector<Vector>& unk() const;
    void showMe() const;
```

```

private:
    UInt _M_order;
    UInt _M_size;
    Vector _M_alpha;
    Vector _M_beta;
    std::vector<Vector> _M_unknowns;
;

Bdf::Bdf( const UInt p )
{
    _M_order( p ),
    _M_size( 0 ),
    _M_alpha( p + 1 ),
    _M_beta( p )

    if ( n <= 0 || n > BDF_MAX_ORDER )

        // Error handling for requesting a wrong or non-implemented order

        switch ( p )

            case 1:
                _M_alpha[ 0 ] = 1.; // implicit Euler
                _M_alpha[ 1 ] = 1.;
                _M_beta[ 0 ] = 1.; // u at time n+1 approximated by u at time n
                break;
            case 2:
                _M_alpha[ 0 ] = 3. / 2.;
                _M_alpha[ 1 ] = 2.;
                _M_alpha[ 2 ] = -1. / 2.;
                _M_beta[ 0 ] = 2.;
                _M_beta[ 1 ] = -1.;
                break;
            case 3:
                _M_alpha[ 0 ] = 11. / 6.;
                _M_alpha[ 1 ] = 3.;
                _M_alpha[ 2 ] = -3. / 2.;
                _M_alpha[ 3 ] = 1. / 3.;
                _M_beta[ 0 ] = 3.;
                _M_beta[ 1 ] = -3.;
                _M_beta[ 2 ] = 1.;
                break;
    }

    _M_unknowns.resize( p ); //number of columns of matrix _M_unknowns
}

```

8.6 A complete example

We conclude this chapter with the listing of a program written in LifeV for the solution of the parabolic diffusion-reaction problem:

$$\begin{cases} \frac{\partial u}{\partial t} - \mu(t) \Delta u + \sigma(t) u = f, & \mathbf{x} \in \Omega, \quad 0 < t \leq 10, \\ u = g_1, & \mathbf{x} \in \Gamma_{10} \cup \Gamma_{11}, \quad 0 < t \leq 10, \\ u = g_2, & \mathbf{x} \in \Gamma_{20} \cup \Gamma_{21}, \quad 0 < t \leq 10, \\ \nabla u \cdot \mathbf{n} = 0, & \mathbf{x} \in \Gamma_{50}, \quad 0 < t \leq 10, \\ u = u_0, & \mathbf{x} \in \Omega, \quad t = 0, \end{cases}$$

where Ω is a cubic domain and $\partial\Omega = \Gamma_{10} \cup \Gamma_{11} \cup \Gamma_{20} \cup \Gamma_{21} \cup \Gamma_{50}$. Precisely, the numerical codes on the boundary portions are:

$$\begin{aligned} \Gamma_{20} : & x = 0, 0 < y < 1, 0 < z < 1; \\ \Gamma_{21} : & x = 0, (y = 0, 0 < z < 1) \cup (y = 1, 0 < z < 1) \\ & \cup (z = 0, 0 < y < 1) \cup (z = 1, 0 < y < 1); \\ \Gamma_{10} : & x = 1, 0 < y < 1, 0 < z < 1; \\ \Gamma_{11} : & x = 1, (y = 0, 0 < z < 1) \cup (y = 1, 0 < z < 1) \\ & \cup (z = 0, 0 < y < 1) \cup (z = 1, 0 < y < 1); \\ \Gamma_{50} : & \partial\Omega \setminus \{\Gamma_{20} \cup \Gamma_{21} \cup \Gamma_{10} \cup \Gamma_{11} \cup \Gamma_{50}\}. \end{aligned}$$

In particular, $\mu(t) = t^2$, $\sigma(t) = 2$, $g_1(x, y, z, t) = g_2(x, y, z, t) = t^2 + x^2$, $u_0(x, y, z) = 0$, $f = 2t + 2x^2$. The exact solution is precisely $t^2 + x^2$ and the test is proposed on a cubic grid made of 6007 elements with quadratic affine tetrahedra, for a total of 9247 degrees of freedom. The chosen time step is $\Delta t = 0.5$, the order of the chosen BDF scheme is 3.

Listing 20 contains the main program for this example and has been enriched by comments to help the reading, although obviously not everything will be clear simply after reading the preceding paragraphs. Coherently with the spirit with which this chapter has been designed, we invite the reader to try and run the code, suitably modifying it for a full understanding of its structure. Downloading LifeV from www.lifev.org it is possible to obtain other cases to be tried under the testsuite directory.

For the visualization of results, in this listing we refer to the Medit program, freely available at www.inria.fr. Fig. 8.9 reports the matrix pattern before and after the imposition of the boundary conditions, moving from 244171 to 214087 non-null elements.

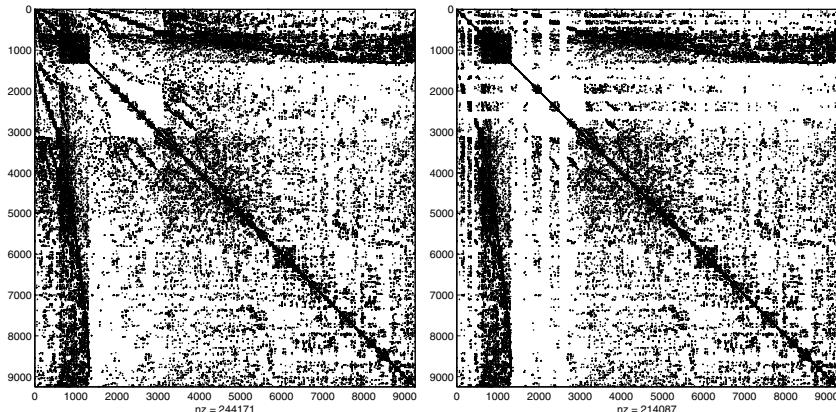


Fig. 8.9. Matrix pattern of the proposed test case before (left) and after (right) applying the boundary conditions

Program 20 - main.cpp: Solution of a parabolic problem on a cubic domain

```
// PRELIMINARY INCLUSIONS OF PARTS OF THE LIBRARY
#include <life/lifecore/GetPot.hpp>
#include "main.hpp"
#include "ud_functions.hpp"
#include <life/lifeFem/bcManage.hpp>
#include <life/lifeFem/eleMat.hpp>
#include <life/lifeFem/eleOper.hpp>
#include <life/lifeFem/bdf.hpp>
#include <life/lifeFilters/medit_wrtrs.hpp>
#include <life/lifeFilters/gmv_wrtrs.hpp>
#include <life/lifeFem/sobolevNorms.hpp>
#define P2 // We will use quadratic affine Lagrangian elements

int main()
    using namespace LifeV;
    using namespace std;

    Chrono chrono; // Utility for the computation of the execution time

    // =====
    // Definition of the boundary conditions (see main.hpp)
    // =====

    BCFFunctionBase gv1(g1); // Function g1
    BCFFunctionBase gv2(g2); // Function g2
    BCHandler BCh(2); // Two boundary conditions are imposed
    // To the two conditions, we associate the numerical codes 10 and 20
```

```

// contained in the computational grid
BCh.addBC("Dirichlet1", 10, Essential, Scalar, gv1);
BCh.addBC("Dirichlet2", 20, Essential, Scalar, gv2);

// =====
// Information on the geometric mapping and on the numerical integration
// =====
const GeoMap& geoMap = geoLinearTetra;
const QuadRule& qr = quadRuleTetra64pt;

const GeoMap& geoMapBd = geoLinearTria;
const QuadRule& qrBd = quadRuleTria3pt;

//P2 elements
const RefFE& refFE = feTetraP2;
const RefFE& refBdFE = feTriaP2;

// =====
// Structure of the mesh
// =====
RegionMesh3D<LinearTetra> aMesh;

GetPot datafile( "data" ); //information on the mesh file
// and other information is contained in a file named "data"
long int m=1;
std::string mesh_type = datafile( "mesh_type", "INRIA" );
string mesh_dir = datafile( "mesh_dir", "." );
string fname=mesh_dir+datafile( "mesh_file", "cube_6007.mesh" );

readMppFile(aMesh,fname,m); // grid reading

aMesh.updateElementEdges();
aMesh.updateElementFaces();
aMesh.showMe();

// =====
// Definition of the current finite element, equipped with
// geometric mapping and quadrature rule
// =====

CurrentFE fe(refFE,geoMap,qr);
CurrentBdFE feBd(refBdFE,geoMapBd,qrBd);

// =====
// Definition of the degrees of freedom (DOF) of the problem
// and of the specific boundary conditions
// =====

Dof dof(refFE);

```

```

dof.update(aMesh);
BCh.bdUpdate( aMesh, feBd, dof );
UInt dim = dof.numTotalDof();
dof.showMe();

// =====
// Initialization of the unknown vectors
// U and of known term F
// =====
ScalUnknown<Vector> U(dim), F(dim);
U=ZeroVector( dim );
F=ZeroVector( dim );

// =====
// Definition of the parameters for the integration in time
// always specified in "data" and read from there
// =====
Real Tfin = datafile( "bdf/endtime", 10.0 );
Real delta_t = datafile( "bdf/timestep", 0.5 );
Real t0 = 0.;
UInt ord_bdf = datafile( "bdf/order", 3 );
Bdf bdf(ord_bdf);
Real coeff=bdf.coeff_der(0)/delta_t;

bdf.showMe();

// =====
// Construction of the pattern and of the time-independent matrices
// =====
// pattern for stiff operator
MSRPatt pattA(dof);

MSRMatr<double> A(pattA);
MSRMatr<double> M(pattA);
M.zeros();

cout << "**** Matrix computation" : "<<endl;
chrono.start();
//
SourceFct sourceFct;
ElemMat elmat(fe.nbNode,1,1);
ElemVec elvec(fe.nbNode,1);
for(UInt i = 1; i<=aMesh.numVolumes(); i++)
  fe.updateJacQuadPt(aMesh.volumeList(i));
  elmat.zero();
  mass(1.,elmat,fe);
  assemb_mat(M,elmat,fe,dof,0,0); // Mass matrix M

```

```

// =====
// Definition of the parameters for the solver of the linear system
// AZTEC is used
// =====
int proc_config[AZ_PROC_SIZE];
    // Information on the processor (serial or parallel)
int options[AZ_OPTIONS_SIZE];
    // Vector of the solver type used
double params[AZ_PARAMS_SIZE];
    // Parameters of the solver used
int *data_org;
    // Utility vector
double status[AZ_STATUS_SIZE];
    // Return vector of the AZTEC call
    // indicates whether the solution has been successful
    // additional declarations for AZTEC
int *update,*external;
int *update_index;
int *extern_index;
int N_update;
//
cout << "**** Linear System Solving (AZTEC)" << endl;
AZ_set_proc_config(proc_config, AZ_NOT_MPI );
AZ_read_update(&N_update, &update, proc_config, U.size(), 1, AZ_linear);

AZ_defaults(options,params);

AZ_transform(proc_config, &external,
    (int *)pattA.giveRaw_bindx(), A.giveRaw_value(),
    update, &update_index,
    &extern_index, &data_org, N_update, NULL, NULL, NULL, NULL,
    AZ_MSR_MATRIX);

chrono.start();
init_options(options,params);

// =====
// TIME LOOP
// =====

int count=0;
bdf.initialize_unk(u0,aMesh,refFE,fe,dof,t0,delta_t,1);

for (Real t=t0+delta_t;t<=Tfin;t+=delta_t)

    A.zeros();
    F=ZeroVector( F.size() );
    // =====

```

```

// Assembly and
// Update of the known term with the solution of
// the preceding steps
// =====

Real visc=nu(t); // mu and sigma depend on time
Real s=sigma(t);
for(UInt i = 1; i<=aMesh.numVolumes(); i++)
    fe.updateFirstDerivQuadPt(aMesh.volumeList(i));
    elmat.zero();
    elvec.zero();
    mass(coeff+s,elmat,fe);
    stiff(visc,elmat,fe);
    source(sourceFct,elvec,fe,t,0);
    assemb_mat(A,elmat,fe,dof,0,0);
    assemb_vec(F,elvec,fe,dof,0);

// Handling of the right hand side
F += M*bdf.time_der(delta_t);

// =====
// Prescription of the boundary conditions
// =====

chrono.start();
A.spy("test.m");
bcManage(A,F,aMesh,dof,BCh,feBd,1.,t);
A.spy("test_bc.m");

chrono.stop();
chrono.start();
AZ_solve(U.giveVec(), F.giveVec(), options, params, NULL,
          (int *)pattA.giveRaw_bindx(), NULL, NULL, NULL,
          A.giveRaw_value(), data_org,
          status, proc_config);
chrono.stop();

// =====
// Writing of the post-processing file
// =====
count++;
index << count;
wr_medit_ascii_scalar( "U" + index.str() + ".bb", U.giveVec(), dim );
wr_medit_ascii( "U" + index.str() + ".mesh", aMesh);

// =====
// In this test case we know the analytic solution

```

```

// (specified in main.hpp)
// and we want to compute the errors in different norms
// =====
AnalyticalSol analyticSol;

Real normL2=0., normL2diff=0., normL2sol=0.;
Real normH1=0., normH1diff=0., normH1sol=0.;

for(UInt i=1; i<=aMesh.numVolumes(); ++i)
  fe.updateFirstDeriv(aMesh.volumeList(i));

normL2 += elem_L2_2(U,fe,dof);
normL2sol += elem_L2_2(analyticSol,fe,t,( UInt )U.nbcomp());
normL2diff += elem_L2_diff_2(U,analyticSol,fe, dof, t,( UInt )U.nbcomp());

normH1 += elem_H1_2(U,fe,dof);
normH1sol += elem_H1_2(analyticSol,fe,t,U.nbcomp());
normH1diff += elem_H1_diff_2(U,analyticSol,fe,dof,t,U.nbcomp());

normL2 = sqrt(normL2);
normL2sol = sqrt(normL2sol);
normL2diff = sqrt(normL2diff);

normH1 = sqrt(normH1);
normH1sol = sqrt(normH1sol);
normH1diff = sqrt(normH1diff);

bdf.shift_right(U);

// END OF TIME LOOP

return EXIT_SUCCESS;

```

The following is a part of the terminal output obtained by running the code:

```

Boundary Conditions Handler =====>
Number of BC stored 5
List =>
*****
BC Name: Wall
Flag: 50
Type: 1
Mode: 0
Number of components: 1
List of components: 1
Number of stored ID's: 0
*****

```

...

```
<=====>
Reading INRIA mesh file
Linear Tetra Mesh
#Vertices = 1322 #BVertices = 599
#Points = 1322 #Boundary Points = 599
#Volumes = 6007
*****
***** RegionMesh3D *****
*****
ID: 0 Marker Flag:1
***** COUNTERS *****
NumPoints=1322 numBPoints=599
NumVertices=1322 numBVertices=599
NumVolumes=6007 numFaces=12611
NumBFaces=1194 numEdges=7925
NumBEdges=1791
*****
***** ACTUALLY STORED *****
Points=1322 Edges= 1791
Faces= 1194 Volumes=6007
*****
```

...

*** BDF Time discretization of order 3 ***

Coefficients:

```
alpha(0) = 1.83333
alpha(1) = 3
alpha(2) = -1.5
alpha(3) = 0.333333
beta (0) = 3
beta (1) = -3
beta (2) = 1
```

3 unknown vectors of length 0

dim = 9247

Now we are at time 0.5

A has been constructed

16.93s.

*** BC Management:

1.44s.

```
*****
```

***** Preconditioned GMRES solution

```
***** ILUT( fill-in = 5.000e+00, drop = 1.000e-04)
***** without overlap
***** No scaling
***** NOTE: convergence VARIES when the total number of
***** processors is changed.
*****
```

```
iter: 0 residual = 1.000000e+00
```

```
***** ilut: The ilut factors require 3.914e+00 times
***** the memory of the overlapped subdomain matrix.
*****
```

```
iter: 1 residual = 9.674996e-04
iter: 2 residual = 1.262096e-04
iter: 3 residual = 9.806253e-06
iter: 4 residual = 5.693536e-07
iter: 5 residual = 4.168325e-08
iter: 6 residual = 3.028269e-09
iter: 7 residual = 1.919715e-10
iter: 8 residual = 1.371272e-11
```

Solution time: 1.160000 (sec.)

total iterations: 8

Flops not available for options[AZ_precond] = 14

*** Solution computed in 1.17s.

$$\begin{aligned}\|U\|_{L^2} &= 0.655108 \\ \|sol\|_{L^2} &= 0.655108 \\ \|U - sol\|_{L^2} &= 1.49398e-09 \\ \|U - sol\|_{L^2}/\|sol\|_{L^2} &= 2.28051e-09 \\ \|U\|_{H^1} &= 1.32759 \\ \|sol\|_{H^1} &= 1.32759 \\ \|U - sol\|_{H^1} &= 8.09782e-09 \\ \|U - sol\|_{H^1}/\|sol\|_{H^1} &= 6.09963e-09\end{aligned}$$

Now we are at time 1

A has been constructed

28.77s.

*** BC Management:

1.43s.

Note that the errors are to be attributed exclusively to the linear system solution: indeed, as the exact solution is a parabolic function in time and space, the choice of finite elements of degree 2 and of the BDF scheme of order 3 guarantees that the discretization errors are non-null. Fig. 8.10 illustrates the results visualized by Medit.

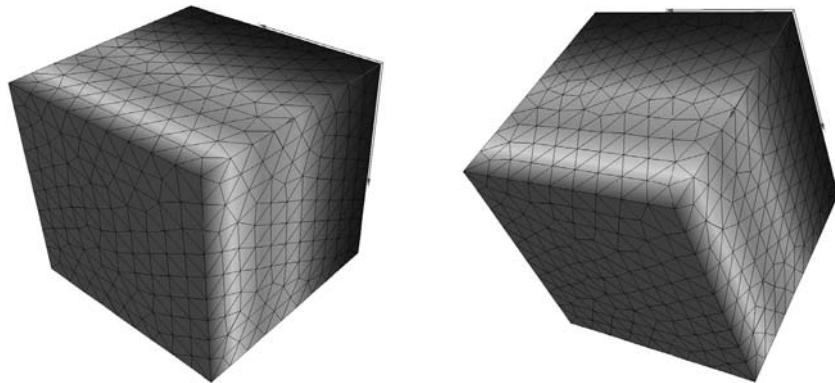


Fig. 8.10. Results of the simulation and after 5 (left) and 20 (right) time steps

The finite volume method

The *finite volume* method is a very popular method for the space discretization of partial differential problems in conservation form. For an in-depth presentation of the method, we suggest the monographs [LeV02a] and [Wes01].

As a paradigm to describe the method and illustrate its main features, let us consider the following scalar equation

$$\frac{\partial u}{\partial t} + \operatorname{div}(\mathbf{F}(u)) = s(u), \quad \mathbf{x} \in \Omega, t > 0 \quad (9.1)$$

where $u : (\mathbf{x}, t) \rightarrow \mathbb{R}$ denotes the unknown, $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ ($d = 1, 2, 3$), \mathbf{F} is a given vector function, linear or nonlinear, called flux, s is a given function called source term. If the flux \mathbf{F} contains terms depending on the first derivatives of u , the differential problem is a second-order one. The differential equation (9.1) must be completed by the initial condition $u(\mathbf{x}, 0) = u_0(\mathbf{x})$, $\mathbf{x} \in \Omega$ for $t = 0$, as well as by suitable boundary conditions, on the whole boundary $\partial\Omega$ in the case where problem (9.1) is a second-order one, or just on a subset $\partial\Omega^{in}$ of $\partial\Omega$ (the inflow boundary) in the case of first-order problems. As we will see in Chap. 14 (see Sec. 14.1 and Sec. 14.4), this type of differential equations are called *conservation laws*. Typically, the finite volume method operates on equations written in conservation form such as (9.1).

The diffusion-transport equations of Chap. 11, the pure transport equations of Chaps. 12–14, and the parabolic ones examined in Chap. 5, can all be considered as special cases of (9.1). Indeed, all the partial differential equations deriving from physical conservation laws can be expressed in conservation form.

With some additional effort, we can obviously consider the vector case, where the unknown \mathbf{u} is a vector function with p components, as well as the source term \mathbf{s} , while the flux \mathbf{F} is now a tensor with dimension $p \times d$. In particular, also the Navier-Stokes equations and Euler equations for compressible flows that will be considered in Sec. 14.4 can be rewritten in conservative form. A finite volume approximation of free-surface incompressible flows for real life applications is reported in Sec. 15.11.

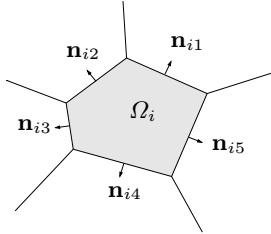


Fig. 9.1. A control volume

9.1 Some basic principles

The preliminary step towards a finite volume discretization of (9.1) consists in identifying a set of polyhedra $\Omega_i \subset \Omega$ with diameter less than h , called *control volumes* (or *control cells*), $i = 1, \dots, M$, such that $\cup_i \overline{\Omega}_i = \overline{\Omega}$ (we will here assume for simplicity that the domain Ω is polygonal, otherwise $\cup_i \overline{\Omega}_i$ will be its approximation). See Fig. 9.1 for an example of control volume in two dimensions. We will furthermore hypothesize the cells to be pairwise disjoint, this being the most commonly used case, although such restriction is not required in principle by the method.

Equation (9.1) is integrated on each Ω_i ; using the divergence theorem we obtain the system of ordinary differential equations

$$\frac{\partial}{\partial t} \int_{\Omega_i} u \, d\Omega + \int_{\partial\Omega_i} \mathbf{F}(u) \cdot \mathbf{n}_i \, d\gamma = \int_{\Omega_i} s(u) \, d\Omega, \quad i = 1, \dots, M. \quad (9.2)$$

We have denoted by \mathbf{n}_i the normal unit vector directed outward to $\partial\Omega_i$. In two dimensions, let us denote by L_i the number of straight sides of Ω_i in Fig. 9.1 ($L_i = 5$) and by \mathbf{n}_{ij} , $j = 1, \dots, L_i$, the (constant) external normal unit vector to the side l_{ij} of $\partial\Omega_i$. Then (9.2) can be rewritten as

$$\frac{\partial}{\partial t} \int_{\Omega_i} u \, d\Omega + \sum_{j=1}^{L_i} \int_{l_{ij}} \mathbf{F}(u) \cdot \mathbf{n}_{ij} \, d\gamma = \int_{\Omega_i} s(u) \, d\Omega, \quad i = 1, \dots, M. \quad (9.3)$$

Several issues have to be addressed:

- which geometrical shape should the control volumes have;
- how to represent the unknown u in each control volume, that is which are its degrees of freedom and where should they be placed;
- how to approximate the (volume and surface) integrals;
- how to represent the flux $\mathbf{F}(u)$ on each side, as a function of the values of the unknown u on the control volumes adjacent to the side.

For the construction of the control volumes, we usually start from a triangulation \mathcal{T}_h of the domain into polygons called elements, say $\{K_m\}$, of the same kind, typically

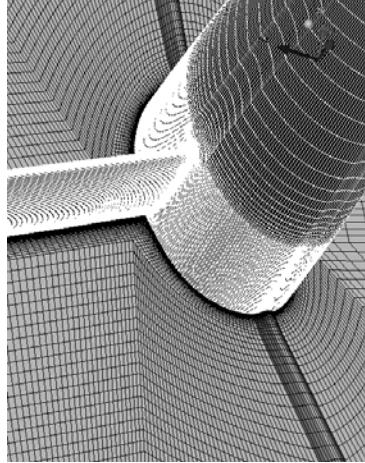


Fig. 9.2. An example of blockwise structured mesh

triangles or quadrilaterals in 2D, tetrahedra or hexahedra in 3D, as seen for instance in Chap. 4 when using finite elements. The grid can be structured, blockwise structured (with either disjoint or overlapping blocks), or unstructured. Structured grids are bounded to domains of relatively simple shape, in order for the whole domain, or each block in which it is subdivided, to be mapped into a rectangle or a cube. In Fig. 9.2 we display a block structured grid on the surface of the appendages of a yacht. Once the domain has been triangulated, two alternatives can be pursued.

In the so-called *cell-centered* method, the elements $\{K_m\}$ of \mathcal{T}_h directly serve as control volumes. Consequently, the unknowns are associated to an internal point on each element, typically the barycenter, called *node*. However, this apparently natural choice of control volumes denotes a disadvantage: as there are no nodes lying on the boundary of Ω , imposing the boundary conditions will require special arrangements, which we will examine later on. To account for such inconvenient, we can construct control volumes around the elements of \mathcal{T}_h , where we will place the unknowns. This yields to the so-called *vertex-centered* schemes.

Sometimes, in multifield problems with several unknowns, both techniques are used at the same time to place different unknowns at different nodes. In this case, we will say that *staggered grids* are used; we will present a remarkable example in Sec. 15.11, which is devoted to the discretization of Navier-Stokes equations.

A basic example on a structured quadrangular grid is reported in Fig. 9.3, where we also show the control volumes for *cell-centered* and *vertex-centered* schemes. The latter are defined by the squares

$$\Omega_i^V = \{\mathbf{x} \in \Omega : \|\mathbf{x} - \mathbf{x}_i\|_\infty < h/2\}, \quad \Omega_i = \Omega_i^V \cap \Omega,$$

$\{\mathbf{x}_i\}$ being the vertices of the $\{K_m\}$ squares of the initial grid \mathcal{T}_h , which coincide in this case with the nodes of the control volumes and h is the uniform length of the element edges.

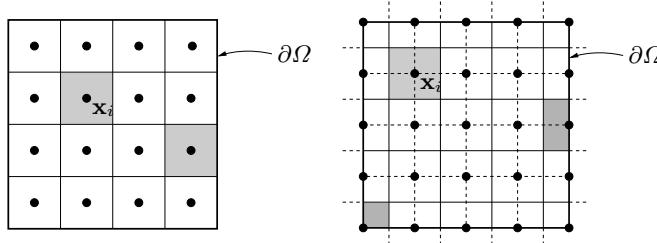


Fig. 9.3. Control volumes (in grey) generated by a partition of a square domain Ω with square finite elements of edge h . In the left figure, we report the *cell-centered* case, in the right one the *vertex-centered* case

These two choices do not exhaust the options encountered in practice. Sometimes, the variables are placed on each edge (or face, in 3D) of the grid \mathcal{T}_h , and the corresponding control volume is formed by the elements of \mathcal{T}_h adjacent to the edge (or face).

In general terms, a finite volume approach is simple to implement: the discretization cells can be chosen in a very general form, the solution is typically assumed to be a constant function in each control volume, the Neumann boundary conditions are imposed in a natural way, and the formulation itself of the problem expresses the local conservation of the amount $\int_{\Omega_i} u \, d\Omega$. The potential disadvantage is the objective difficulty in drawing high-order schemes, the need to treat essential (Dirichlet) boundary conditions, in particular for the cell-centered methods; finally, the mathematical analysis is less simple than in the case of Galerkin methods as a direct application of variational techniques used for the former is not straightforward.

9.2 Construction of control volumes for vertex-centered schemes

In the case the original triangulation \mathcal{T}_h is made of triangular unstructured elements in 2D or tetrahedric ones in 3D, the construction of control volumes around the vertices of \mathcal{T}_h is not straightforward. In principle, we could choose as a control volume Ω_i the set of all elements containing the vertex \mathbf{x}_i . However, this would generate control volumes with non-null intersection, a permitted but not desirable situation.

We can thus take advantage of some geometrical concepts. Let us consider for example a bounded polygonal domain $\Omega \subset \mathbb{R}^2$, and let $\{\mathbf{x}_i\}_{i \in \mathcal{P}}$ be a set of points, which we will call nodes, of $\overline{\Omega}$. Here \mathcal{P} denotes a set of indexes. These points are typically the ones where we intend to provide an approximation of the solution u . We associate to each node the polygon

$$\Omega_i^V = \{\mathbf{x} \in \mathbb{R}^2 : |\mathbf{x} - \mathbf{x}_i| < |\mathbf{x} - \mathbf{x}_j| \ \forall j \in \mathcal{P}, j \neq i\}, \quad (9.4)$$

with $i \in \mathcal{P}$. The set $\{\Omega_i^V, i \in \mathcal{P}\}$ is called *Voronoi diagram*, or also *Voronoi tessellation*, associated to the set of points $\{\mathbf{x}_i\}_{i \in \mathcal{P}}$; Ω_i^V is called i -th *Voronoi polygon*. For

an example, see Fig. 9.4. The polygons thus obtained are convex, but not necessarily bounded (consider for instance the ones adjacent to the boundary). Their vertices are called *Voronoi vertices*; a vertex is said *regular* when it is the meeting point of three Voronoi polygons, and *degenerate* when it is shared by at least four polygons. A Voronoi diagram having all its vertices regular is in turn called *regular*.

At this point, we can define the control volumes Ω_i introduced in the previous section as

$$\Omega_i = \Omega_i^V \cap \Omega, \quad i \in \mathcal{P}. \quad (9.5)$$

For each $i \in \mathcal{P}$, we denote by \mathcal{P}_i the set of indexes of the nodes adjacent to \mathbf{x}_i , i.e.

$$\mathcal{P}_i = \{j \in \mathcal{P} \setminus \{i\} : \partial\Omega_i \cap \partial\Omega_j \neq \emptyset\}.$$

Moreover, we denote by $l_{ij} = \partial\Omega_i \cap \partial\Omega_j$, $j \in \mathcal{P}_i$, a side of the boundary of Ω_i shared by an adjacent control volume, and by m_{ij} its length. If the Voronoi diagram is regular, we have $m_{ij} > 0$. In this case, if we connect each node \mathbf{x}_i with the nodes of \mathcal{P}_i , we obtain a triangulation of Ω coinciding with the Delaunay triangulation (see Sec. 6.4.1) of the convex hull of the nodes. In case there are degenerate vertices in the Voronoi tessellation, from this procedure we can still obtain a Delaunay triangulation provided a suitable triangulation of the polygons Ω_i constructed around the degenerate vertices is made. Clearly, if Ω is convex, the above-described process directly provides a Delaunay triangulation of Ω itself, see e.g. Fig. 9.5. The inverse procedure is also possible, noting that the vertices of the Voronoi diagram correspond to the centers of the circles circumscribed to the triangles (the circumcenters) of the corresponding Delaunay triangulation. The triangle axes thus form the sides of the tessellation. The latter therefore represents a possible set of control volumes associated to a given Delaunay triangulation (see e.g. Fig. 9.6).

The Voronoi diagram and the Delaunay triangulation are indeed in a duality relation: each vertex of the Voronoi tessellation is tied in a one-to-one correspondence to an element (triangle) of the Delaunay triangulation, and each vertex of the Delaunay triangulation is in a one-to-one correspondence with a polygon in the tessellation and therefore with a node.

There are two interesting properties which are worth highlighting. The first one is that the center of the circumscribed circle to a non-obtuse triangle K lies within the closure of K . Hence, if the Delaunay triangulation has no obtuse angles, the vertices of the corresponding Voronoi diagram are all contained in $\overline{\Omega}$. The second is that, if we denote by \mathbf{v}_i , $i = 1, 2, 3$, the vertices of the non-obtuse triangle K , and by

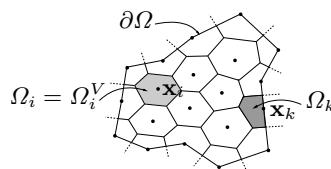


Fig. 9.4. A Voronoi diagram

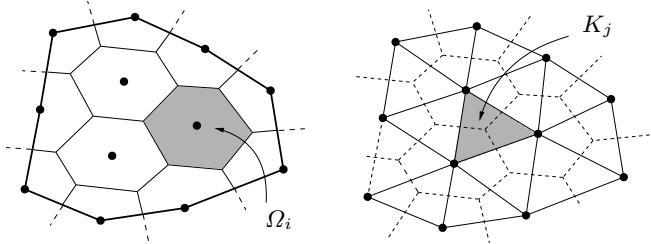


Fig. 9.5. Delaunay triangulation (right) obtained from a Voronoi diagram (left). The dots indicate the nodes $\{\mathbf{x}_i\}_{i \in \mathcal{P}}$

$\Omega_{i,K} = \Omega_i \cap K$ the portion of the control volume Ω_i included in K , then we have the following inequalities between the measures of K and $\Omega_{i,K}$,

$$\frac{1}{4} |K| \leq |\Omega_{i,K}| \leq \frac{1}{2} |K|, \quad i = 1, 2, 3. \quad (9.6)$$

An alternative to the construction based on the Voronoi diagram which does not require a Delaunay triangulation consists in starting from a triangulation \mathcal{T}_h of Ω formed by any kind of triangles, including obtuse ones. If K is the generic triangle of \mathcal{T}_h with vertices \mathbf{v}_i , $i = 1, 2, 3$, we now define

$$\Omega_{i,K} = \{\mathbf{x} \in K : \lambda_j(\mathbf{x}) < \lambda_i(\mathbf{x}), j \neq i\}$$

where λ_j are the barycentric coordinates of K (see Sec. 4.4.3 for their definition). An example is reported in Fig. 9.7. At this point, the control volumes can be defined in the following way

$$\Omega_i = \text{int}\left(\bigcup_{\{K : \mathbf{v}_i \in \partial K\}} \overline{\Omega}_{i,K}\right), \quad i \in \mathcal{P},$$

where $\text{int}(\mathcal{D})$ denotes the interior of the closed set \mathcal{D} . The family $\{\Omega_i, i \in \mathcal{P}\}$ defines the so-called *median dual grid* (sometimes also named *Donald diagram*). See Fig. 9.8 for an example. Consequently, we can define the quantities l_{ij} , m_{ij} and \mathcal{P}_i as for the Voronoi diagram. Now the l_{ij} elements are not necessarily straight segments.

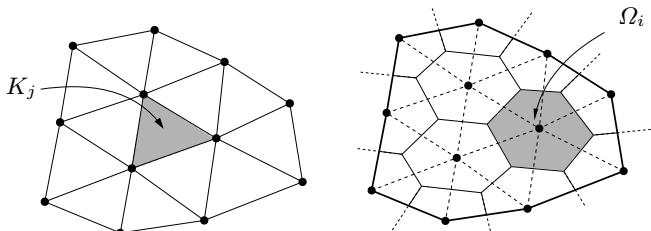


Fig. 9.6. Voronoi diagram (right) obtained starting from a Delaunay triangulation (left)

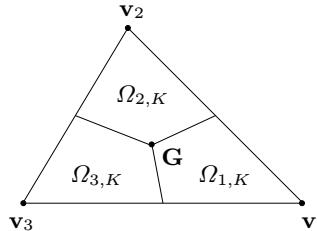


Fig. 9.7. A triangle K , its center of gravity $\mathbf{G} = \frac{1}{3}(\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3)$, and the polygons $\Omega_{i,K}$

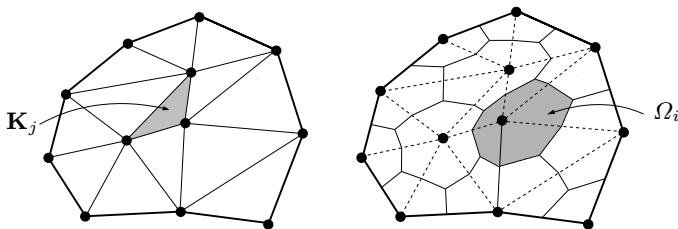


Fig. 9.8. Triangulation of the domain (left) and median dual grid, or Donald diagram (right)

9.3 Discretization of a diffusion-transport-reaction problem

Let us consider for the sake of an example equation (9.1) where

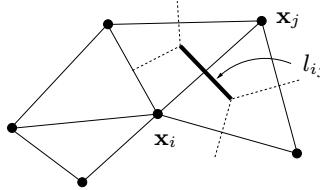
$$\mathbf{F}(u) = -\mu \nabla u + \mathbf{b} u, \quad s(u) = f - \sigma u. \quad (9.7)$$

This is a time-dependent diffusion-transport-reaction equation written in conservation form, similar to the one described at the beginning of Chap. 11. Functions f , μ , σ and \mathbf{b} are given; for these, we formulate the hypotheses made at the beginning of Chap. 11. As in the case of problem (11.1), here again we will suppose for simplicity that u satisfies a homogeneous Dirichlet boundary condition, $u = 0$ on $\partial\Omega$. Let us suppose that Ω be partitioned by a Voronoi diagram and consider the corresponding Delaunay triangulation (an instance is provided in Fig. 9.5). What follows can in fact be extended to other types of finite volumes; for that, it will be sufficient to consider the set of the inner indexes only, $\mathcal{P}_{int} = \{i \in \mathcal{P} : \mathbf{x}_i \in \Omega\}$, u being null at the boundary. Integrating the assigned equation on the control volume Ω_i as we did in (9.3) and using the divergence theorem, we find

$$\frac{\partial}{\partial t} \int_{\Omega_i} u d\Omega + \sum_{j=1}^{L_i} \int_{l_{ij}} \left(-\mu \frac{\partial u}{\partial \mathbf{n}_{ij}} + \mathbf{b} \cdot \mathbf{n}_{ij} u \right) d\gamma = \int_{\Omega_i} (f - \sigma u) d\Omega, \quad i \in \mathcal{P}_{int}. \quad (9.8)$$

In order to approximate the line integrals, a typical strategy consists in approximating the functions μ and $\mathbf{b} \cdot \mathbf{n}_{ij}$ using piecewise constants, and precisely

$$\mu|_{l_{ij}} \simeq \mu_{ij} = \text{const} > 0, \quad \mathbf{b} \cdot \mathbf{n}_{ij}|_{l_{ij}} \simeq b_{ij} = \text{const}. \quad (9.9)$$

**Fig. 9.9.** The segment l_{ij}

Such constants can represent either the value of the corresponding function at the midpoint of segment l_{ij} , or the mean value on the same side, that is

$$\mu_{ij} = \frac{1}{m_{ij}} \int_{l_{ij}} \mu \, d\gamma, \quad b_{ij} = \frac{1}{m_{ij}} \int_{l_{ij}} \mathbf{b} \cdot \mathbf{n}_{ij} \, d\gamma.$$

As far as the normal derivatives are concerned, an option consists in approximating them using incremental ratios of the type

$$\frac{\partial u}{\partial \mathbf{n}_{ij}} \simeq \frac{u(\mathbf{x}_j) - u(\mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|}$$

(see e.g. Fig. 9.9). This formula is exact if u is linear on the segment connecting \mathbf{x}_i and \mathbf{x}_j . Finally, regarding the approximation of the integral of u on l_{ij} , we replace $u|_{l_{ij}}$ by a constant obtained by a linear convex combination, that is

$$u|_{l_{ij}} \simeq \rho_{ij} u(\mathbf{x}_i) + (1 - \rho_{ij}) u(\mathbf{x}_j),$$

with $\rho_{ij} \in [0, 1]$ a parameter to be defined. Operating the previously introduced approximations and denoting by u_i the approximation of the unknown value $u(\mathbf{x}_i)$, we can derive from (9.8) the following approximate equations

$$\begin{aligned} m_i \frac{du_i}{dt} &+ \sum_{j=1}^{L_i} m_{ij} \left\{ -\mu_{ij} \frac{u_j - u_i}{\delta_{ij}} + b_{ij} [\rho_{ij} u_i + (1 - \rho_{ij}) u_j] \right\} \\ &+ m_i \sigma_i u_i = m_i f_i, \quad i \in \mathcal{P}, \end{aligned} \quad (9.10)$$

having denoted by m_i the measure of Ω_i , by σ_i and f_i the values of σ and f in \mathbf{x}_i and by δ_{ij} the distance between \mathbf{x}_i and \mathbf{x}_j . Note that (9.10) can be written in the form

$$m_i \frac{du_i}{dt} + \sum_{j=1}^{L_i} m_{ij} H_{ij}(u_i, u_j) + m_i \sigma_i u_i = m_i f_i, \quad (9.11)$$

where H_{ij} is the so-called *numerical flux* representing the contribution of the approximation of the flux through the side l_{ij} . The concept of numerical flux is relevant also in the context of finite difference schemes for hyperbolic equations, as we will see in Chap. 12 and 14. Some of the features of the numerical flux also translate into scheme properties. For instance, to have a conservative scheme, it will be necessary that $H_{ij}(u_i, u_j) = -H_{ji}(u_j, u_i)$.

9.4 Analysis of the finite volume approximation

The system of equations (9.10) can be rewritten in the form of a discrete variational problem by proceeding in the following way. For each $i = 1, \dots, \overset{\circ}{M}$, the i -th equation is multiplied by a real number v_i , then by summing over the index i , we obtain

$$\begin{aligned} \sum_{i=1}^{\overset{\circ}{M}} m_i v_i \frac{du_i}{dt} + \sum_{i=1}^{\overset{\circ}{M}} v_i \sum_{j=1}^{L_i} m_{ij} \left\{ -\mu_{ij} \frac{u_j - u_i}{\delta_{ij}} + b_{ij} [\rho_{ij} u_i + (1 - \rho_{ij}) u_j] \right\} \\ + \sum_{i=1}^{\overset{\circ}{M}} m_i \sigma_i v_i u_i = \sum_{i=1}^{\overset{\circ}{M}} m_i v_i f_i. \end{aligned} \quad (9.12)$$

Let us now denote by V_h the space of piecewise linear continuous functions with respect to the Delaunay triangulation \mathcal{T}_h , which vanish at the boundary $\partial\Omega$ (see (4.17)). From a set of values v_i we can univocally reconstruct a function $v_h \in V_h$ that interpolates such values at the nodes \mathbf{x}_i , that is

$$v_h \in V_h : v_h(\mathbf{x}_i) = v_i, \quad i = 1, \dots, \overset{\circ}{M}.$$

In a similar way, let $u_h \in V_h$ be the function interpolating the values u_i at \mathbf{x}_i . Then, (9.12) is rewritten equivalently in the following discrete variational form: for each $t > 0$, find $u_h = u_h(t) \in V_h$ s.t.

$$\left(\frac{\partial}{\partial t} u_h, v_h \right)_h + a_h(u_h, v_h) = (f, v_h)_h \quad \forall v_h \in V_h, \quad (9.13)$$

having set $(w_h, v_h)_h = \sum_{i=1}^{\overset{\circ}{M}} m_i v_i w_i$ and having denoted by $a_h(u_h, v_h)$ the bilinear form appearing at the left-hand side of (9.12). We have thus interpreted the finite volume approximation as a particular case of the generalized Galerkin method for the assigned problem. As far as the choice of the ρ_{ij} coefficients for the linear combination is concerned, an option is to use $\rho_{ij} = 1/2$, which corresponds to using a finite difference of the centered type for the convective term. This strategy is adequate when the local Péclet number is not very large, which means there is no index pair i, j verifying

$$\mu_{ij} \ll |b_{ij}| \delta_{ij}.$$

Instead, when this is verified, it is necessary to impose a more careful choice of the ρ_{ij} coefficients for the convex combination. In general, $\rho_{ij} = \varphi(\mathbb{P}_{ij})$, where φ is a function of the *local Péclet number* $\mathbb{P}_{ij} = b_{ij} \delta_{ij} / \mu_{ij}$ with values in $[0, 1]$ that can be chosen as follows: if $\varphi(z) = 1/2 [\text{sign}(z) + 1]$ we will have a stabilization of *upwind* type, while choosing $\varphi(z) = 1 - (1 - z/(e^z - 1))/z$ we will have a stabilization of *exponential-fitting* type. (A similar kind of stabilization will be used in Sec. 11.6 in the context of finite difference approximation of diffusion-transport equations.) By this choice, we can show that the bilinear form $a_h(\cdot, \cdot)$ is V_h -elliptic, uniformly with respect to h , under the usual hypothesis that the coefficients of the problem satisfy the positivity condition $1/2 \operatorname{div}(\mathbf{b}) + \sigma \geq \beta_0 = \text{const} \geq 0$.

Precisely, in this case, supposing further that $\mu \geq \mu_0 = \text{const} > 0$,

$$a_h(v_h, v_h) \geq \mu_0 |v_h|_{H^1(\Omega)}^2 + \beta_0 (v_h, v_h)_h.$$

Moreover, as $(v_h, v_h)_h$ is uniformly equivalent to the exact scalar product (v_h, v_h) for functions of V_h , this ensures the stability of problem (9.13). Finally, the method is linearly convergent with respect to h . In particular

$$\|u - u_h\|_{H^1(\Omega)} \leq C h (\|u\|_{H^2(\Omega)} + |\nabla f|_{L^\infty(\Omega)})$$

under the assumption that the norms at the right-hand side are bounded. For the proof, see e.g. [KA00]. We suggest the same reference for an analysis of other properties of the method, such as monotonicity and conservation.

9.5 Implementation of boundary conditions

As previously stated, the differential problem under exam must be completed by suitable boundary conditions. For a problem written in conservation form, natural boundary conditions would be to impose the fluxes, i.e.

$$\mathbf{F}(u) \cdot \mathbf{n} = h \quad \text{on } \Gamma_N \subset \partial\Omega.$$

For their implementation in the framework of finite volumes it is sufficient to act on the numerical flux relating to the boundary sides, imposing

$$H_{ik} = H(u_i, u_k) = h(\mathbf{x}_{ik}) \quad \text{if } l_{ik} \subset \Gamma_N,$$

where \mathbf{x}_{ik} is a suitable point (typically the midpoint) of l_{ik} .

On the other hand, essential (Dirichlet) conditions of the form

$$u = g \quad \text{on } \Gamma_D \subset \partial\Omega,$$

are immediate to implement in the context of vertex-centered schemes, as it is sufficient to add the corresponding equation for the nodes lying on Γ_D . As previously noted, the matter is more complicated for cell-centered schemes, as in this case there are no nodes on the boundary. An option is to impose the conditions *weakly*, in a similar way to what we will illustrate, although in a different context, in Sec. 13.3.1. This is a matter of suitably modifying the numerical fluxes on the sides, imposing

$$H_{ik} = H(u_i, g(\mathbf{x}_{ik})) \quad \text{if } l_{ik} \subset \Gamma_D.$$

Often however, in practice Dirichlet boundary conditions for cell-centered finite volumes are implemented using the so-called *ghost nodes*. For each side l_{ik} on the boundary, we generate additional nodes, external to the domain, to which the corresponding boundary values are assigned. This way, the computation of numerical fluxes is formally the same also for the boundary sides.

10

Spectral methods

As we have seen in Chap. 4, when we approximate boundary-value problems using the finite element method, the order of convergence is anyhow limited by the degree of the polynomials used, also in the case where solutions are very regular. In this chapter, we will introduce *spectral methods*, for which the convergence rate is only limited by the regularity of the solution of the problem (and is exponential for analytical solutions). For a detailed analysis we refer to [CHQZ06, CHQZ07, Fun92, BM92].

10.1 The spectral Galerkin method for elliptic problems

The main feature that distinguishes finite elements from spectral methods in their classical “single-domain” version, is that the latter use global polynomials on the computational domain Ω , instead of piecewise polynomials. This is however no-longer true in the case of the spectral element method.

For each positive integer N , we denote by \mathbb{Q}_N the space of polynomials with real coefficients with degree less than or equal to N with respect to *each* of the variables. Thus, in one dimension we will denote by

$$\mathbb{Q}_N(I) = \left\{ v(x) = \sum_{k=0}^N a_k x^k, \quad a_k \in \mathbb{R} \right\} \quad (10.1)$$

the space of polynomials of degree $\leq N$ on the interval $I \subset \mathbb{R}$, while in two dimensions,

$$\mathbb{Q}_N(\Omega) = \left\{ v(\mathbf{x}) = \sum_{k,m=0}^N a_{km} x_1^k x_2^m, \quad a_{km} \in \mathbb{R} \right\} \quad (10.2)$$

will denote the same space, but on the open set $\Omega \subset \mathbb{R}^2$. We note that, while in one dimension $\mathbb{Q}_N = \mathbb{P}_N$, in several dimensions this does not happen. In particular, $\dim \mathbb{Q}_N = (N+1)^2$, while, as already seen in Sec. 4.4.1, $\dim \mathbb{P}_N = (N+1)(N+2)/2$.

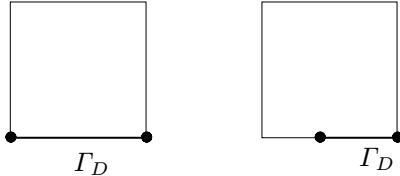


Fig. 10.1. Acceptable (left) and unacceptable (right) Dirichlet boundaries for the spectral method SM

Suppose we want to approximate the solution u of an elliptic problem which admits the variational formulation (4.1). Using a *spectral Galerkin method* (SM in short), the space V will be approximated by a space $V_N \subset \mathbb{Q}_N$ and the approximate solution will consequently be indicated by u_N . In particular, if we suppose that V be $H_{\Gamma_D}^1(\Omega)$ (the space defined in (3.27)), V_N will denote the set of polynomials of \mathbb{Q}_N that vanish on the boundary portion Γ_D where a Dirichlet condition is prescribed, that is

$$V_N = \{v_N \in \mathbb{Q}_N : v_N|_{\Gamma_D} = 0\}.$$

It is evident that $V_N \subset V$. The spectral Galerkin method SM will therefore be formulated on the subspace V_N . However, there is an issue in the definition of V_N : in the multi-dimensional case it is indeed not possible (in general) to require that a polynomial v_N vanishes only on an arbitrary part of the boundary of Ω . For instance, if Ω is the square $(-1, 1)^2$, it is impossible to construct a polynomial that is null only on a part of one side of the square without it being null on the whole side (see Fig. 10.1). This does not prevent a polynomial from vanishing on one whole side of the square or on all of its sides without necessarily being null in the whole of Ω (for instance, $v_2(\mathbf{x}) = (1 - x_1^2)(1 - x_2^2)$ is null only on the boundary of Ω).

For this reason, in the two-dimensional case we limit our attention to square domains (or, more generally, to domains that are reconductible, through appropriate transformation, to the reference square $\widehat{\Omega} = (-1, 1)^2$) and we suppose that the boundary portion Γ_D is formed by the union of one or more sides of the domain.

However, the spectral method can be extended to the case of a domain Ω composed by the union of quadrilaterals Ω_k , each of which can be reconducted to the reference square $\widehat{\Omega}$ via an invertible transformation $\varphi_k : \widehat{\Omega} \rightarrow \Omega_k$ (see Fig. 10.2), yielding to the so-called *spectral element method* (SEM), that was introduced by A. T. Patera [Pat84]. It is evident that in such a context it will be possible to require that the solution vanish on portions of the boundary given by the union of sides of the quadrilateral, but naturally not by portions of sides (see Fig. 10.2). In the SEM case, the discrete space has the following form

$$V_N^C = \{v_N \in C^0(\overline{\Omega}) : v_N|_{\Omega_k} \circ \varphi_k \in \mathbb{Q}_N(\widehat{\Omega})\}.$$

Example 10.1 A particularly important two-dimensional mapping is the one constituted by the transfinite interpolation (called *Gordon-Hall transformation* as well as *Coons patch*). The mapping φ_k is in this case expressed as a function of the invertible mappings $\pi_k^{(i)} : (-1, 1) \rightarrow$

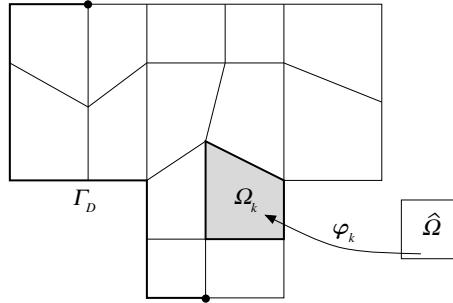


Fig. 10.2. Decomposition of the solution domain and acceptable boundary conditions for the SEM

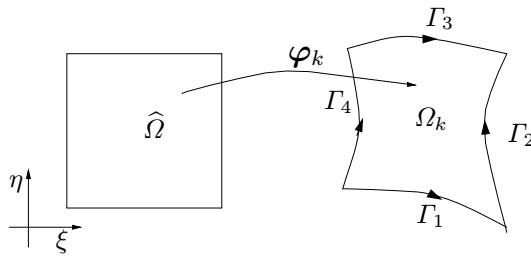


Fig. 10.3. The transformation φ_k in the case of the transfinite interpolation

Γ_i (for $i = 1, \dots, 4$) that define the four sides of the computational domain Ω_k (see Fig. 10.3). The transformation takes the following form

$$\begin{aligned} \varphi_k(\xi, \eta) &= \frac{1-\eta}{2}\pi_k^1(\xi) + \frac{1+\eta}{2}\pi_k^3(\xi) \\ &+ \frac{1-\xi}{2}[\pi_k^4(\eta) - \frac{1+\eta}{2}\pi_k^3(-1) - \frac{1-\eta}{2}\pi_k^1(-1)] \\ &+ \frac{1+\xi}{2}[\pi_k^2(\eta) - \frac{1+\eta}{2}\pi_k^2(1) - \frac{1-\eta}{2}\pi_k^2(-1)]. \end{aligned} \quad (10.3)$$

The transfinite interpolation therefore allows to consider computational domains Ω characterized by curvy domains. For more examples of transformations, see [CHQZ07]. ■

The approximation of problem (4.1) using the Galerkin spectral method (SM) is the following

$$\text{find } u_N \in V_N : \quad a(u_N, v_N) = F(v_N) \quad \forall v_N \in V_N,$$

while the spectral element one (SEM) will be

$$\text{find } u_N \in V_N^C : \quad a_C(u_N, v_N) = F_C(v_N) \quad \forall v_N \in V_N^C, \quad (10.4)$$

where

$$a_C(u_N, v_N) = \sum_k a_{\Omega_k}(u_N, v_N), \quad F_C(v_N) = \sum_k F_{\Omega_k}(v_N),$$

$a_{\Omega_k}(\cdot, \cdot)$ and $F_{\Omega_k}(\cdot)$ being the restrictions of $a(\cdot, \cdot)$ and $F(\cdot)$ to Ω_k .

Since these methods represent a special instance of the Galerkin method (4.2), the analysis made in Sec. 4.2 continues to hold and therefore, in particular, the existence, uniqueness, stability and convergence results can be applied.

Moreover, it can be proven that, for SM and SEM spectral methods, the following a priori error estimates hold:

Theorem 10.1 *Let $u \in V$ be the exact solution of the variational problem (4.1) and suppose that $u \in H^{s+1}(\Omega)$, for some $s \geq 0$. If u_N is the corresponding approximate solution obtained via SM, the following estimate holds*

$$\|u - u_N\|_{H^1(\Omega)} \leq C_s N^{-s} \|u\|_{H^{s+1}(\Omega)},$$

N being the degree of the approximating polynomials and C_s being a constant that does not depend on N , but can depend on s . If u_N is instead the solution obtained via SEM, then we have

$$\|u - u_N\|_{H^1(\Omega)} \leq C_s H^{\min(N, s)} N^{-s} \|u\|_{H^{s+1}(\Omega)},$$

H being the maximal length of the sides of the macroelements Ω_k .

It is therefore evident that, as opposed to what happens for the finite element method, a greater regularity of the solution leads to an increase in convergence rate even supposing that the polynomial degree N is fixed. In particular, if u is analytical, the order of convergence of the spectral method becomes more than algebraic, i.e. exponential: more precisely,

$$\exists \gamma > 0 : \|u - u_N\|_{H^1(\Omega)} \leq C \exp(-\gamma N).$$

Also in the case where u has finite regularity, it is however possible to obtain from the spectral method the maximal convergence rate allowed by the regularity of the exact solution: this is a clear advantage of spectral methods with respect to finite elements. The main limitation (in two or three dimensions) of classical spectral methods SM is that they can only handle simple geometries: rectangles or quadrilaterals which can be mapped into a square via an invertible transformation. However, as previously mentioned, they can be extended, via the SEM, to the case where the domain is given by the union of quadrilaterals, possibly with curvy sides.

A further disadvantage of classical spectral methods lies in the fact that the stiffness matrix A associated to them is full in the one-dimensional case, or anyhow much less sparse than the one for finite elements in more dimensions, because of the fact that the basis functions of such methods have global (and not local) support, see Sec. 10.2 and 10.3. The associated system of equations therefore results to be generally more costly to solve.

Finally, the computational cost required to compute the elements of the stiffness matrix of the right-hand side must not be underestimated, as we are dealing with high

degree polynomials. This issue is successfully addressed in the next section by using well-chosen Gaussian numerical integration.

Remark 10.1 In the course of Sec. 10.5 at the end of this chapter, we will provide the algebraic formulation of the SEM method for a one-dimensional problem. In particular, we will introduce the basis functions for the space V_N^C of composite polynomials. •

Remark 10.2 The SEM approach has a not so different formulation from the p version of the finite element method. In both cases, the number of subdomains Ω_k is fixed while the local degree of polynomials (called N in the case of SEM, p in the finite element case) is locally augmented in order to improve the accuracy of the numerical approximation. For further details, we refer the interested reader to [CHQZ07, Sch98]. •

10.2 Orthogonal polynomials and Gaussian numerical integration

In this section, we introduce the mathematical ingredients that allow to construct numerical integration formulae of gaussian type. As previously anticipated, such formulae are the basis of pseudo-spectral methods, but also of spectral element methods that make use of numerical integration formulae.

10.2.1 Orthogonal Legendre polynomials

Let us consider a function $f : (-1, 1) \rightarrow \mathbb{R}$. We recall that the space $L^2(-1, 1)$ is defined by (see Sec. 2.3.1)

$$L^2(-1, 1) = \left\{ f : (-1, 1) \rightarrow \mathbb{R} : \|f\|_{L^2(-1, 1)} = \left(\int_{-1}^1 f^2(x) dx \right)^{1/2} < \infty \right\}.$$

Its scalar product is given by

$$(f, g) = \int_{-1}^1 f(x)g(x)dx.$$

The *orthogonal Legendre polynomials* $L_k \in \mathbb{P}_k$, for $k = 0, 1, \dots$, constitute a sequence for which the following orthogonality property is satisfied

$$(L_k, L_m) = \begin{cases} 0 & \text{if } m \neq k, \\ (k + \frac{1}{2})^{-1} & \text{if } m = k. \end{cases}$$

These are linearly independent and form a basis for $L^2(-1, 1)$. Consequently, each function $f \in L^2(-1, 1)$ admits the series expansion

$$f(x) = \sum_{k=0}^{\infty} \hat{f}_k L_k(x) \tag{10.5}$$

known as *Legendre series*. This is a *modal* representation of f . The Legendre coefficients \hat{f}_k can easily be computed by exploiting the orthogonality of Legendre polynomials. Indeed, we have

$$\begin{aligned}(f, L_k) &= \int_{-1}^1 f(x)L_k(x) dx = \int_{-1}^1 \left(\sum_{i=0}^{\infty} \hat{f}_i L_i(x)L_k(x) \right) dx \\ &= \sum_{i=0}^{\infty} \left(\int_{-1}^1 L_i(x)L_k(x) dx \right) \hat{f}_i = \hat{f}_k \|L_k\|_{L^2(-1,1)}^2.\end{aligned}$$

Henceforth,

$$\hat{f}_k = (f, L_k) / \|L_k\|_{L^2(-1,1)}^2 = (k + \frac{1}{2}) \int_{-1}^1 f(x)L_k(x) dx \quad (10.6)$$

from which the so-called *Parseval identity* immediately derives

$$\|f\|_{L^2(-1,1)}^2 = \sum_{k=0}^{\infty} (\hat{f}_k)^2 \|L_k\|_{L^2(-1,1)}^2.$$

It is possible to recursively compute the Legendre polynomials via the following three term relation:

$$\begin{aligned}L_0 &= 1, \quad L_1 = x, \\ L_{k+1} &= \frac{2k+1}{k+1} x L_k - \frac{k}{k+1} L_{k-1}, \quad k = 1, 2, \dots\end{aligned}$$

(In Fig. 10.4, the graphs of the L_k polynomials, with $k = 2, \dots, 5$, are drawn). For each $f \in L^2(-1, 1)$, its Legendre series converges to f in the $L^2(-1, 1)$ norm. Denoting by

$$f_N(x) = \sum_{k=0}^N \hat{f}_k L_k(x)$$

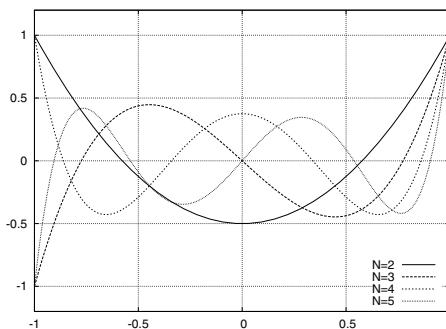


Fig. 10.4. The Legendre polynomials of degree $k = 2, 3, 4, 5$

the N -th truncation of the Legendre series of f , this means that

$$\lim_{N \rightarrow \infty} \|f - f_N\|_{L^2(-1,1)} = 0, \quad (10.7)$$

that is

$$\lim_{N \rightarrow \infty} \left\| \sum_{k=N+1}^{\infty} \hat{f}_k L_k \right\|_{L^2(-1,1)} = 0.$$

Thanks to the Parseval identity, we have that

$$\|f - f_N\|_{L^2(-1,1)}^2 = \sum_{k=N+1}^{\infty} (\hat{f}_k)^2 \|L_k\|_{L^2(-1,1)}^2 = \sum_{k=N+1}^{\infty} \frac{(\hat{f}_k)^2}{k + \frac{1}{2}},$$

hence the condition (10.7) is equivalent to

$$\lim_{N \rightarrow \infty} \sum_{k=N+1}^{\infty} \frac{(\hat{f}_k)^2}{k + \frac{1}{2}} = 0.$$

Moreover, it can be proven that, if $f \in H^s(-1, 1)$, for some $s \geq 1$, then it is possible to find a suitable constant $C_s > 0$, independent of N , such that

$$\|f - f_N\|_{L^2(-1,1)} \leq C_s \left(\frac{1}{N} \right)^s \|f^{(s)}\|_{L^2(-1,1)},$$

i.e. we have convergence of order s , with respect to $1/N$.

At this point, we can prove that f_N is the orthogonal projection of f on \mathbb{Q}_N with respect to the scalar product of $L^2(-1, 1)$, that is

$$(f - f_N, p) = 0 \quad \forall p \in \mathbb{Q}_N. \quad (10.8)$$

First of all we note that

$$(f - f_N, L_m) = \left(\sum_{k=N+1}^{\infty} \hat{f}_k L_k, L_m \right) = \sum_{k=N+1}^{\infty} \hat{f}_k (L_k, L_m).$$

Since the L_k polynomials, with $0 \leq k \leq N$, form a basis for the space \mathbb{Q}_N , every polynomial $p \in \mathbb{Q}_N$ can be expanded w.r.t. this basis. Equation (10.8) follows noticing that for $m \leq N$, $(L_k, L_m) = 0 \quad \forall k \geq N + 1$ because of the orthogonality.

In particular, from (10.8) also descends that f_N is the function which minimizes the distance of f from \mathbb{Q}_N , that is

$$\|f - f_N\|_{L^2(-1,1)} \leq \|f - p\|_{L^2(-1,1)} \quad \forall p \in \mathbb{Q}_N. \quad (10.9)$$

For this purpose, we start by observing that

$$\|f - f_N\|_{L^2(-1,1)}^2 = (f - f_N, f - f_N) = (f - f_N, f - p) + (f - f_N, p - f_N)$$

for each $p \in \mathbb{Q}_N$ and that $(f - f_N, p - f_N) = 0$ by the orthogonality property (10.8). Consequently,

$$\|f - f_N\|_{L^2(-1,1)}^2 = (f - f_N, f - p) \quad \forall p \in \mathbb{Q}_N,$$

from which, applying the Cauchy-Schwarz inequality, we obtain

$$\|f - f_N\|_{L^2(-1,1)}^2 \leq \|f - f_N\|_{L^2(-1,1)} \|f - p\|_{L^2(-1,1)} \quad \forall p \in \mathbb{Q}_N,$$

i.e. (10.9).

10.2.2 Gaussian integration

Gaussian integration formulae are the ones which, having fixed the number of quadrature nodes, allow to obtain the maximal *exactness degree* (see [QSS07]). The latter is the highest integer r s.t. all polynomials of degree less than or equal to r are integrated *exactly* by the formula at hand. We will start by introducing such formulae on the $(-1, 1)$ interval, consequently extending them to the case of a generic interval.

We denote by N the number of nodes. We call Gauss-Legendre quadrature nodes the *zeroes* $\{\bar{x}_1, \dots, \bar{x}_N\}$ of the Legendre polynomial L_N . In the presence of such a node set, we will consider the following quadrature formula (called interpolatory of Gauss-Legendre)

$$I_{N-1}^{GL} f = \int_{-1}^1 \Pi_{N-1}^{GL} f(x) dx, \quad (10.10)$$

$\Pi_{N-1}^{GL} f$ being the polynomial of degree $N-1$ interpolating f at the nodes $\bar{x}_1, \dots, \bar{x}_N$. We denote by $\bar{\psi}_k \in \mathbb{Q}_{N-1}$, $k = 1, \dots, N$, the characteristic Lagrange polynomials associated to the Gauss-Legendre nodes, that is such that

$$\bar{\psi}_k(\bar{x}_j) = \delta_{kj}, \quad j = 1, \dots, N.$$

The quadrature formula (10.10) then takes the following expression

$$\int_{-1}^1 f(x) dx \simeq I_{N-1}^{GL} f = \sum_{k=1}^N \bar{\alpha}_k f(\bar{x}_k), \quad \text{with } \bar{\alpha}_k = \int_{-1}^1 \bar{\psi}_k(x) dx,$$

and is called Gauss-Legendre quadrature formula (GL).

To find the nodes \bar{t}_k and the weights $\bar{\delta}_k$ characterizing such formula on a generic interval $[a, b]$, it will be sufficient to refer for the former to the relation

$$\bar{t}_k = \frac{b-a}{2} \bar{x}_k + \frac{a+b}{2},$$

while, for the latter, it can easily be verified that

$$\bar{\delta}_k = \frac{b-a}{2} \bar{\alpha}_k.$$

The *exactness degree* of these formulae is equal to $2N - 1$ (and is the maximum possible for formulae with $N - 1$ nodes). This means that

$$\int_a^b f(x)dx = \sum_{k=1}^N \bar{\delta}_k f(\bar{t}_k) \quad \forall f \in \mathbb{Q}_{2N-1}.$$

10.2.3 Gauss-Legendre-Lobatto formulae

A feature of the Gauss-Legendre integration formulae is to have all quadrature nodes internal to the integration interval. In the case of differential problems, this makes the imposition of boundary conditions on the extrema of the interval problematic.

To overcome such difficulty, the so-called Gauss-Lobatto formulae are introduced, particularly the Gauss-Legendre-Lobatto (GLL) formulae, whose nodes, relative to the interval $(-1, 1)$, are represented by the extrema of the interval themselves, and by the maximum and minimum points of the Legendre polynomial of degree N , i.e. by the zeroes of the first derivative of the L_N polynomial.

We denote such nodes by $\{x_0 = -1, x_1, \dots, x_{N-1}, x_N = 1\}$. Therefore, we have

$$L'_N(x_i) = 0, \quad \text{for } i = 1, \dots, N-1. \quad (10.11)$$

(In this chapter the prime denotes derivative w.r.t. x .) Let ψ_i be the corresponding characteristic polynomials, that is

$$\psi_i \in \mathbb{Q}_N : \quad \psi_i(x_j) = \delta_{ij}, \quad 0 \leq i, j \leq N, \quad (10.12)$$

whose analytical expression is given by

$$\psi_i(x) = \frac{-1}{N(N+1)} \frac{(1-x^2)L'_N(x)}{(x-x_i)L_N(x_i)}, \quad i = 0, \dots, N \quad (10.13)$$

(see Fig. 10.5 reporting the graphs of the characteristic polynomials ψ_i , for $i = 0, \dots, 4$ in the case where $N = 4$). The $\psi_i(x)$ functions are the counterpart of the

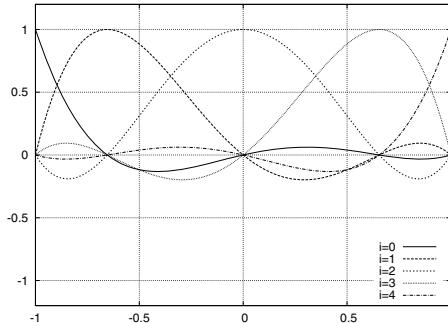


Fig. 10.5. The characteristic polynomials ψ_i , $i = 0, \dots, 4$ of degree 4 corresponding to the Gauss-Legendre-Lobatto nodes

Lagrangian basis functions $\{\varphi_i\}$ of the finite elements introduced in Sec. 4.3. For each function $f \in C^0([-1, 1])$, its interpolation polynomial $\Pi_N^{GLL} f \in \mathbb{Q}_N$ at the GLL nodes is identified by the relation

$$\Pi_N^{GLL} f(x_i) = f(x_i), \quad 0 \leq i \leq N. \quad (10.14)$$

It has the following expression

$$\Pi_N^{GLL} f(x) = \sum_{i=0}^N f(x_i) \psi_i(x). \quad (10.15)$$

It can be proven that, thanks to the non-uniform distribution of the nodes $\{x_i\}$, $\Pi_N^{GLL} f$ converges towards f when $N \rightarrow \infty$. Moreover, the following error estimate is satisfied: if $f \in H^s(-1, 1)$, for some $s \geq 1$,

$$\|f - \Pi_N^{GLL} f\|_{L^2(-1,1)} \leq C_s \left(\frac{1}{N}\right)^s \|f^{(s)}\|_{L^2(-1,1)}, \quad (10.16)$$

where C_s is a constant depending on s but not on N . More generally (see [CHQZ06]),

$$\|f - \Pi_N^{GLL} f\|_{H^k(-1,1)} \leq C_s \left(\frac{1}{N}\right)^{s-k} \|f^{(s)}\|_{L^2(-1,1)}, \quad s \geq 1, k = 0, 1. \quad (10.17)$$

In Fig. 10.6 (left), we report the convergence curves for the interpolation error of two different functions.

By using $\Pi_N^{GLL} f$ instead of $\Pi_{N-1}^{GLL} f$ we can define the following Gauss-Legendre-Lobatto (GLL) integration formula, in alternative to (10.10)

$$I_N^{GLL} f = \int_{-1}^1 \Pi_N^{GLL} f(x) dx = \sum_{k=0}^N \alpha_k f(x_k). \quad (10.18)$$

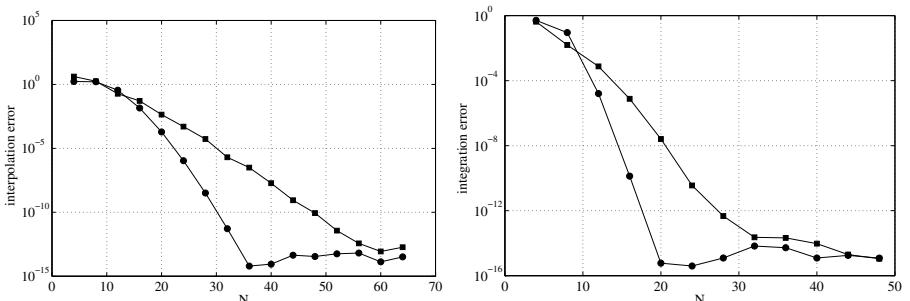


Fig. 10.6. Behavior of the interpolation (left) and integration (right) error in the GLL nodes as a function of the degree N for the two functions $f_1(x) = \cos(4\pi x)$ (●) and $f_2(x) = 4 \cos(4x) \exp(\sin(4x))$ (■) on the interval $(-1, 1)$

The new weights are $\alpha_i = \int_{-1}^1 \psi_i(x) dx$ and take the following expression

$$\alpha_i = \frac{2}{N(N+1)} \frac{1}{L_N^2(x_i)}. \quad (10.19)$$

The GLL formula has *exactness degree* equal to $2N - 1$, that is it integrates exactly all the polynomials of degree $\leq 2N - 1$,

$$\int_{-1}^1 f(x) dx = I_N^{GLL} f \quad \forall f \in \mathbb{Q}_{2N-1}. \quad (10.20)$$

This is the maximum obtainable degree when $N + 1$ nodes are used, 2 of which assigned a priori. Moreover, using the interpolation estimate (10.16), the following integration error estimate can be proven: if $f \in H^s(-1, 1)$, with $s \geq 1$,

$$\left| \int_{-1}^1 f(x) dx - I_N^{GLL} f \right| \leq C_s \left(\frac{1}{N} \right)^s \|f^{(s)}\|_{L^2(-1,1)},$$

where C_s is independent of N but can depend, in general, on s . This means that the more regular function f is, the higher is the order of convergence of the integration formula. In Fig. 10.6 (right) we report the integration error for two different functions (the same ones considered for the left graph).

If we now consider a generic interval (a, b) instead of $(-1, 1)$, nodes and weights in (a, b) take the following expression

$$t_k = \frac{b-a}{2} x_k + \frac{a+b}{2}, \quad \delta_k = \frac{b-a}{2} \alpha_k.$$

Formula (10.20) generalizes as follows

$$\int_a^b f(x) dx \simeq \sum_{k=0}^N \delta_k f(t_k). \quad (10.21)$$

The properties of exactness and accuracy remain unchanged.

10.3 G-NI methods in one dimension

Let us consider the following one-dimensional elliptic problem, with homogeneous Dirichlet data

$$\begin{cases} Lu = -(\mu u')' + \sigma u = f, & -1 < x < 1, \\ u(-1) = 0, & u(1) = 0, \end{cases} \quad (10.22)$$

with $\mu(x) \geq \mu_0 > 0$ and $\sigma(x) \geq 0$, in order to have an associated bilinear form that is coercive in $H_0^1(-1, 1)$.

The *spectral Galerkin method* SM is written as

$$\text{find } u_N \in V_N : \int_{-1}^1 \mu u'_N v'_N \, dx + \int_{-1}^1 \sigma u_N v_N \, dx = \int_{-1}^1 f v_N \, dx \quad \forall v_N \in V_N, \quad (10.23)$$

with

$$V_N = \{v_N \in \mathbb{Q}_N : v_N(-1) = v_N(1) = 0\}. \quad (10.24)$$

The G-NI (*Galerkin with Numerical Integration*) method is obtained by approximating the integrals in (10.23) via the GLL integration formulae. This amounts to substituting the scalar product (f, g) in $L^2(-1, 1)$ by the *discrete GLL scalar product* (for continuous functions)

$$(f, g)_N = \sum_{i=0}^N \alpha_i f(x_i) g(x_i), \quad (10.25)$$

where the x_i and the α_i are defined according to (10.11) and (10.19). Hence, the G-NI method is written as

$$\text{find } u_N^* \in V_N : (\mu u_N^{*\prime}, v'_N)_N + (\sigma u_N^*, v_N)_N = (f, v_N)_N \quad \forall v_N \in V_N. \quad (10.26)$$

Due to the numerical integration, in general it will be that $u_N^* \neq u_N$, that is the solutions of the SM and G-NI methods do not coincide.

However, we observe that, thanks to the exactness property (10.20), we will have

$$(f, g)_N = (f, g) \quad \text{provided that } fg \in \mathbb{Q}_{2N-1}. \quad (10.27)$$

If we consider the particular case where in (10.22) μ is a constant and $\sigma = 0$, the G-NI problem becomes

$$\mu(u_N^{*\prime}, v'_N)_N = (f, v_N)_N. \quad (10.28)$$

In some very particular cases, we can observe a coincidence between the spectral and the G-NI methods. This is for instance the case of (10.28), where f is a polynomial with degree equal at most to $N - 1$. It is simple to verify that the two methods coincide thanks to the exactness relation (10.27).

Generalizing to the case of more complex differential formulations having different boundary conditions (Neumann, or mixed), the G-NI problem is written as

$$\text{find } u_N^* \in V_N : a_N(u_N^*, v_N) = F_N(v_N) \quad \forall v_N \in V_N, \quad (10.29)$$

where $a_N(\cdot, \cdot)$ and $F_N(\cdot)$ are obtained starting from the bilinear form $a(\cdot, \cdot)$ and from the known term $F(\cdot)$ of the spectral Galerkin problem, by substituting the exact integrals with the GLL integration formulae. V_N is the space of polynomials of degree N that vanish on the boundary conditions (provided that there are any) on which Dirichlet functions are imposed.

Observe that, due of the fact that the bilinear form $a_N(\cdot, \cdot)$ and the functional $F_N(\cdot)$ are no longer the ones associated to the initial problem, what we obtain is no longer a Galerkin approximation method, thus the theoretical results pertaining to it cannot be applied (in particular, the Céa lemma, see Lemma 4.1).

In general, a method derived from a Galerkin method, either spectral or with finite elements, having substituted the exact integrals with the numerical ones will be called *generalized Galerkin method (GG)*. For the corresponding analysis we will resort to the Strang lemma (see Sec. 10.4.1 and also [Cia78, QV94]).

10.3.1 Algebraic interpretation of the G-NI method

The functions ψ_i , with $i = 1, 2, \dots, N - 1$, introduced in Sec. 10.2.3 constitute a basis for the space V_N , as they are all null in $x_0 = -1$ and $x_N = 1$. We can therefore provide for the solution u_N^* of the problem G-NI (10.29) the *nodal* representation

$$u_N^*(x) = \sum_{i=1}^{N-1} u_N^*(x_i) \psi_i(x),$$

that is, analogously with the finite element method, identify the unknowns of our problem with the values taken by u_N^* at the nodes x_i (now coinciding with the Gauss-Legendre-Lobatto nodes). Moreover, for problem (10.29) to be verified for each $v_N \in V_N$, it will be sufficient that it be verified for each function with basis ψ_i . We will therefore have

$$\sum_{j=1}^{N-1} u_N^*(x_j) a_N(\psi_j, \psi_i) = F_N(\psi_i), \quad i = 1, 2, \dots, N - 1,$$

which we can rewrite

$$\sum_{j=1}^{N-1} a_{ij} u_N^*(x_j) = f_i, \quad i = 1, 2, \dots, N - 1,$$

that is, in matrix form,

$$\mathbf{A}\mathbf{u}_N^* = \mathbf{f} \tag{10.30}$$

where

$$\mathbf{A} = (a_{ij}) \quad \text{with} \quad a_{ij} = a_N(\psi_j, \psi_i), \quad \mathbf{f} = (f_i) \quad \text{with} \quad f_i = F_N(\psi_i),$$

and where \mathbf{u}_N^* denotes the vector of unknown coefficients $u_N^*(x_j)$, for $j = 1, \dots, N - 1$. In the particular case of problem (10.26), we would obtain

$$a_{ij} = (\mu\psi'_j, \psi'_i)_N + \alpha_i \sigma(x_i) \delta_{ij}, \quad f_i = (f, \psi_i)_N = \alpha_i f(x_i),$$

for each $i, j = 1, \dots, N - 1$. The matrix in 1D is full due to the presence of the diffusive term. Indeed, the reactive term only contributes to the diagonal. In more

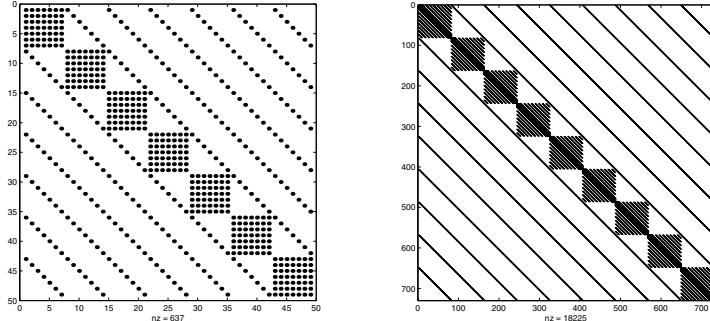


Fig. 10.7. Pattern of matrix A of the G-NI method for the 2D (left) and 3D (right) case: nz denotes the number of non-null elements in the matrix

dimensions, matrix A has a block structure, and the diagonal blocks are full. See Fig. 10.7, reporting the *pattern* relating to matrix A in 2D and 3D. Finally, we observe that the condition number of the matrix to which we would get in the absence of numerical integration results, in general, to be even larger, being $O(N^4)$. Moreover, the A matrix results to be ill-conditioned, with a condition number that turns out to be $O(N^3)$. For the solution of system (10.30) it is therefore convenient to resort, especially in 2D and 3D, to a suitably preconditioned iterative method. By choosing as a preconditioner the matrix of linear finite elements associated to the same bi-linear form $a(\cdot, \cdot)$ and to the GLL nodes, we obtain a preconditioned matrix whose conditioning is independent of N ([CHQZ06]). In the top of Fig. 10.8, we report the condition number (as a function of N) of the A matrix and of the matrix obtained by preconditioning A with different preconditioning matrices: the diagonal matrix of A , the one obtained from A through the incomplete Cholesky factorization, the one obtained using linear finite elements by approximating the integrals with the composite trapezoidal formula, and finally the exact one from finite elements. In the bottom of Fig. 10.8, we report the number

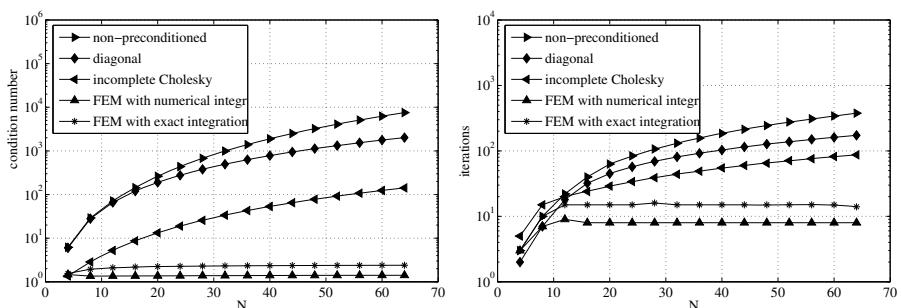


Fig. 10.8. Condition number (left) and iteration number (right), for different types of preconditioners

of necessary iterations for the conjugate gradient method to converge in the different cases.

10.3.2 Conditioning of the stiffness matrix in the G-NI method

We seek estimates for the eigenvalues λ^N of the stiffness matrix A of the G-NI method

$$A\mathbf{u} = \lambda^N \mathbf{u}.$$

In the case of the simple second derivative operator, we have $A = (a_{ij})$, with $a_{ij} = (\psi'_j, \psi'_i)_N = (\psi'_j, \psi'_i)$, ψ_j being the j -th characteristic Lagrange function associated to the node x_j . Then,

$$\lambda^N = \frac{\mathbf{u}^T A \mathbf{u}}{\mathbf{u}^T \mathbf{u}} = \frac{\|u_x^N\|_{L^2(-1,1)}^2}{\mathbf{u}^T \mathbf{u}}, \quad (10.31)$$

$u^N \in V_N$ being the only polynomial of the space V_N defined in (10.24) satisfying $u^N(x_j) = u_j$, for $j = 1, \dots, N - 1$, where $\mathbf{u} = (u_j)$. Thus, for each j , $u_j = \int_{-1}^{x_j} u_x^N(s) ds$; thanks to the Cauchy-Schwarz inequality we obtain the bound

$$|u_j| \leq \left(\int_{-1}^{x_j} |(u^N)'(s)|^2 ds \right)^{1/2} \left(\int_{-1}^{x_j} ds \right)^{1/2} \leq \sqrt{2} \|(u^N)'\|_{L^2(-1,1)}.$$

Hence

$$\mathbf{u}^T \mathbf{u} = \sum_{j=1}^{N-1} u_j^2 \leq 2(N-1) \|(u^N)'\|_{L^2(-1,1)}^2,$$

which, thanks to (10.31), provides the lower bound

$$\lambda^N \geq \frac{1}{2(N-1)}. \quad (10.32)$$

An upper bound for λ^N can be obtained by recurring to the following *inverse inequality* for algebraic polynomials (see [CHQZ06], Sec. 5.4.1)

$$\exists C > 0 : \forall p \in V_N, \quad \|p'\|_{L^2(-1,1)} \leq \sqrt{2} N \left(\int_{-1}^1 \frac{p^2(x)}{1-x^2} dx \right)^{1/2}. \quad (10.33)$$

Then

$$\|(u^N)'\|_{L^2(-1,1)}^2 \leq 2N^2 \int_{-1}^1 \frac{[u^N(x)]^2}{1-x^2} dx = 2N^2 \sum_{j=1}^{N-1} \frac{[u^N(x_j)]^2}{1-x_j^2} \alpha_j, \quad (10.34)$$

where we use the exactness of the GLL integration formula (see (10.20)), being $[u^N]^2/(1-x^2) \in \mathbb{P}_{2N-2}$. Since for the α_j coefficients the following asymptotic estimate holds: $\alpha_j/(1-x_j^2) \leq C$, for a suitable constant C independent of N , we can conclude, thanks to (10.31) and (10.34), that

$$\lambda^N \leq 2C N^2. \quad (10.35)$$

It can finally be proven that both estimates (10.32) and (10.35) are optimal as far as the asymptotic behavior with respect to N is concerned.

10.3.3 Equivalence between G-NI and collocation methods

We want to prove that there exists a precise relation between G-NI and *collocation methods*, i.e. those methods imposing the fulfillment of the differential equation only at selected points of the computational interval. Let us consider once again the homogeneous Dirichlet problem (10.22), whose associated G-NI problem is written in the form (10.26).

We would like to counterintegrate by parts equation (10.26), but in order to do that, we must first rewrite the discrete scalar products as integrals. Let $\Pi_N^{GLL} : C^0([-1, 1]) \mapsto \mathbb{Q}_N$ be the interpolation operator introduced in Sec. 10.2.3 which maps a continuous function into the corresponding interpolating polynomial at the Gauss-Legendre-Lobatto nodes.

Since the GLL integration formula uses the values of the function only at the integration nodes and since the function and its G-NI interpolant coincide thereby, we have

$$\sum_{i=0}^N \alpha_i f(x_i) = \sum_{i=0}^N \alpha_i \Pi_N^{GLL} f(x_i) = \int_{-1}^1 \Pi_N^{GLL} f(x) dx,$$

where the latter equality descends from (10.20) as $\Pi_N^{GLL} f$ is integrated exactly, being a polynomial of degree N .

The discrete scalar product can thus be reconducted to a scalar product in $L^2(-1, 1)$, in the case where one of the two functions is a polynomial of degree strictly less than N , i.e.

$$(f, g)_N = (\Pi_N^{GLL} f, g)_N = (\Pi_N^{GLL} f, g) \quad \forall g \in \mathbb{Q}_{N-1}. \quad (10.36)$$

In this case, indeed, $\Pi_N^{GLL} f \in \mathbb{Q}_N$, $(\Pi_N^{GLL} f)g \in \mathbb{Q}_{2N-1}$ and therefore the integral is computed exactly. Integrating by parts the exact integrals, we obtain¹

$$\begin{aligned} (\mu u'_N, v'_N)_N &= (\Pi_N^{GLL}(\mu u'_N), v'_N)_N = (\Pi_N^{GLL}(\mu u'_N), v'_N) \\ &= -([\Pi_N^{GLL}(\mu u'_N)]', v_N)_+ + [\Pi_N^{GLL}(\mu u'_N) v_N]_{-1}^1 \\ &= -([\Pi_N^{GLL}(\mu u'_N)]', v_N)_N, \end{aligned}$$

¹ From now on, for simplicity of notation, we will denote the solution G-NI by u_N (instead of u_N^*), since there is no longer the risk of a confusion with the spectral solution.

where the last equality is justified as v_N vanishes at the boundary and the terms which appear the scalar product yield a polynomial whose total degree is equal to $2N - 1$. At this point, we can rewrite the G-NI problem as follows

$$\text{find } u_N \in V_N : (L_N u_N, v_N)_N = (f, v_N)_N \quad \forall v_N \in V_N, \quad (10.37)$$

where we have defined

$$L_N u_N = -[\Pi_N^{GLL}(\mu u'_N)]' + \sigma u_N. \quad (10.38)$$

By now imposing that (10.37) is valid for each basis function ψ_i , we obtain

$$(L_N u_N, \psi_i)_N = (f, \psi_i)_N, \quad i = 1, 2, \dots, N - 1.$$

We now examine the i -th equation. The first term is

$$\begin{aligned} -([\Pi_N^{GLL}(\mu u'_N)]', \psi_i)_N &= -\sum_{j=0}^N \alpha_j [\Pi_N^{GLL}(\mu u'_N)]'(x_j) \psi_i(x_j) \\ &= -\alpha_i [\Pi_N^{GLL}(\mu u'_N)]'(x_i), \end{aligned}$$

since $\psi_i(x_j) = \delta_{ij}$. Analogously, for the second term, we have

$$(\sigma u_N, \psi_i)_N = \sum_{j=0}^N \alpha_j \sigma(x_j) u_N(x_j) \psi_i(x_j) = \alpha_i \sigma(x_i) u_N(x_i).$$

Finally, the right-hand side becomes

$$(f, \psi_i)_N = \sum_{j=0}^N \alpha_j f(x_j) \psi_i(x_j) = \alpha_i f(x_i).$$

Dividing by α_i the equation thus found, we obtain, to conclude, the following equivalent formulation of the G-NI problem

$$\begin{cases} L_N u_N(x_i) = f(x_i), & i = 1, 2, \dots, N - 1, \\ u_N(x_0) = 0, & u_N(x_N) = 0. \end{cases} \quad (10.39)$$

This is called a *collocation* problem as it is equivalent to placing at the internal nodes x_i the assigned differential equation (after approximating the operator L by L_N), as well as satisfying the boundary conditions at the boundary nodes.

We now introduce the *interpolation derivative*, $D_N(\Phi)$, of a continuous function Φ , as being the derivative of the interpolating polynomial $\Pi_N^{GLL}\Phi$ defined according to (10.14), i.e.

$$D_N(\Phi) = D[\Pi_N^{GLL}\Phi], \quad (10.40)$$

D being the exact differentiation symbol. If we now consider the differential operator L and replace all the derivatives with the corresponding interpolation derivatives, we

obtain a new operator, called *pseudo-spectral operator* L_N , that exactly coincides with the one defined in (10.38). It follows that the G-NI method, introduced here as a generalized Galerkin method, can also be interpreted as a collocation method that operates directly on the differential part of the problem, analogously to what happens, for instance, in the case of finite differences. In this sense, the finite differences can be considered as a less accurate version of the G-NI method as the derivatives are approximated using formulae that use a small number of nodal values.

If the initial operator had been

$$Lu = (-\mu u')' + (bu)' + \sigma u$$

the corresponding pseudo-spectral operator would have been

$$L_N u_N = -D_N(\mu u'_N) + D_N(bu_N) + \sigma u_N. \quad (10.41)$$

In the case where the boundary conditions for problem (10.22) were of Neumann type,

$$(\mu u')(-1) = g_-, \quad (\mu u')(1) = g_+,$$

the spectral Galerkin method would be formulated as follows

$$\begin{aligned} \text{find } u_N \in \mathbb{Q}_N : \quad & \int_{-1}^1 \mu u'_N v'_N \, dx + \int_{-1}^1 \sigma u_N v_N \, dx = \\ & \int_{-1}^1 f v_N \, dx + \mu(1) g_+ v_N(1) - \mu(-1) g_- v_N(-1) \quad \forall v_N \in \mathbb{Q}_N, \end{aligned}$$

while the G-NI method would become

$$\begin{aligned} \text{find } u_N \in \mathbb{Q}_N : \quad & (\mu u'_N, v'_N)_N + (\sigma u_N, v_N)_N = \\ & (f, v_N)_N + \mu(1) g_+ v_N(1) - \mu(-1) g_- v_N(-1) \quad \forall v_N \in \mathbb{Q}_N. \end{aligned}$$

Its interpretation as a collocation method becomes: find $u_N \in \mathbb{Q}_N$ s.t.

$$\begin{aligned} L_N u_N(x_i) &= f(x_i), \quad i = 1, \dots, N-1, \\ (L_N u_N(x_0) - f(x_0)) - \frac{1}{\alpha_0} ((\mu u'_N)(-1) - g_-) &= 0, \\ (L_N u_N(x_N) - f(x_N)) + \frac{1}{\alpha_N} ((\mu u'_N)(1) - g_+) &= 0, \end{aligned}$$

where L_N is defined in (10.38). Note that at the boundary nodes, the Neumann condition is satisfied up to the equation residual $L_N u_N - f$ multiplied by the coefficient of the GLL formula which is an infinitesimal of order 2 with respect to $1/N$.

In Fig. 10.9 (taken from [CHQZ06]) we report the error in the $H^1(-1, 1)$ -norm (left) and the absolute value of the difference $(\mu u'_N)(\pm 1) - g_{\pm}$ (right), that can be regarded

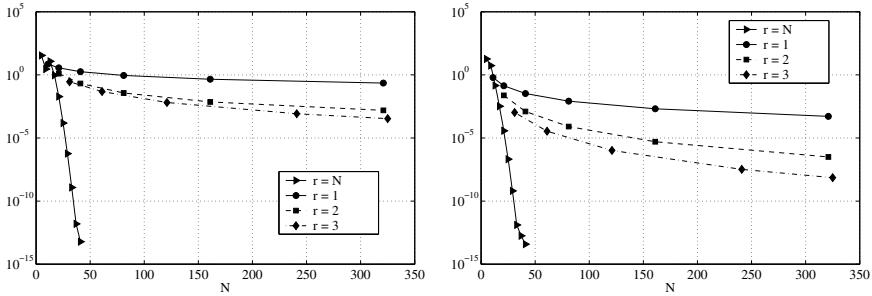


Fig. 10.9. Error in $H^1(-1, 1)$ (left) and error on the Neumann datum (right) for varying N

as the error made on the fulfillment of the Neumann boundary condition, for different values of N . Both errors decay exponentially when N increases. Moreover, we report the errors obtained by using the Galerkin finite element approximations of degree $r = 1, 2, 3$.

Finally, it can be useful to observe that the interpolation derivative (10.40) can be represented through a matrix $D \in \mathbb{R}^{(N+1) \times (N+1)}$, called *matrix of the interpolation derivative*, associating to any vector $\mathbf{v} \in \mathbb{R}^{N+1}$ of nodal values $v_i = \Phi(x_i)$, $i = 0, \dots, N$, the vector $\mathbf{w} = D\mathbf{v}$ whose components are the nodal values of the polynomial $D_N(\Phi)$, i.e. $w_i = (D_N(\Phi))(x_i)$, $i = 0, \dots, N$. The elements of D are (see [CHQZ06])

$$D_{ij} = \psi'_j(x_i) = \begin{cases} \frac{L_N(x_i)}{L_N(x_j)} \frac{1}{x_i - x_j}, & i, j = 0, \dots, N, i \neq j, \\ -\frac{(N+1)N}{4}, & i = j = 0, \\ \frac{(N+1)N}{4}, & i = j = N, \\ 0 & \text{otherwise,} \end{cases}$$

where $d_0 = d_N = 2$ and $d_j = 1$ for $j = 1, \dots, N-1$.

10.3.4 G-NI for parabolic equations

When we consider time-dependent problems, the spectral G-NI method can be used for the spatial approximation. For the discretization of the time derivative we can then apply a finite difference scheme. In this section, we consider one specific instance, the θ -method that was introduced in Sec. 5.1.

The θ -method applied to the G-NI spatial discretization of the homogeneous Dirichlet problem (5.4), defined on the space interval $-1 < x < 1$, is formulated as follows:

for each $k \geq 0$, find $u_N^k \in V_N = \{v_N \in \mathbb{Q}_N : v_N(-1) = v_N(1) = 0\}$ s.t.

$$\begin{aligned} & \left(\frac{u_N^{k+1} - u_N^k}{\Delta t}, v_N \right)_N + a_N(\theta u_N^{k+1} + (1-\theta)u_N^k, v_N) \\ &= \theta(f^{k+1}, v_N)_N + (1-\theta)(f^k, v_N)_N \quad \forall v_N \in V_N, \end{aligned}$$

with $u_N^0 = u_{0,N} \in V_N$ being a convenient approximation of u_0 (for instance, the interpolant $\Pi_N^{GLL}u_0$ introduced in (10.14)). As usual, $(\cdot, \cdot)_N$ denotes the discrete scalar product obtained using the Gauss-Legendre-Lobatto (GLL) numerical integration formula, while $a_N(\cdot, \cdot)$ is the approximation of the bilinear form $a(\cdot, \cdot)$ obtained by replacing the exact integrals with the above-mentioned numerical integration formula. By proceeding as we did in Sec. 5.4 for finite element spatial discretizations, it can be proven that also in this case, the θ -method is unconditionally stable if $\theta \geq \frac{1}{2}$, while for $\theta < \frac{1}{2}$ we have absolute stability if

$$\Delta t \leq C(\theta)N^{-4}. \quad (10.42)$$

Indeed, the proof can be achieved by repeating the same steps we followed earlier in the case of the finite element approximation. In particular, we define the eigenvalue-eigenfunction pairs (λ_j, w_N^j) of the bilinear form $a_N(\cdot, \cdot)$, for each $j = 1, \dots, N-1$, through the relation

$$w_N^j \in V_N : a_N(w_N^j, v_N) = \lambda_j (w_N^j, v_N) \quad \forall v_N \in V_N.$$

Hence

$$\lambda_j = \frac{a_N(w_N^j, w_N^j)}{\|w_N^j\|_N^2}.$$

Using the continuity of the bilinear form $a_N(\cdot, \cdot)$, we find

$$\lambda_j \leq \frac{M \|w_N^j\|_{H^1(-1,1)}^2}{\|w_N^j\|_N^2}.$$

We now recall the following inverse inequality for algebraic polynomials ([CHQZ06])

$$\exists C_I > 0 : \|v'_N\|_{L^2(-1,1)}^2 \leq C_I \|v_N\|_{L^2(-1,1)}^2 \quad \forall v_N \in \mathbb{Q}_N.$$

Then

$$\lambda_j \leq \frac{C_I^2 M N^4 \|w_N^j\|_{L^2(-1,1)}^2}{\|w_N^j\|_N^2}.$$

Recalling the equivalence property (10.54), we conclude that

$$\lambda_j \leq 3C_I^2 M N^4 \quad \forall j = 1, \dots, N-1.$$

The result (10.42) is now obtained using the stability condition (5.35) (with the finite element eigenvalues $\{\lambda_h^i\}$ replaced by the λ_j 's). Moreover, we have the following

convergence estimate, for $n \geq 1$ and $\Omega = (-1, 1)$,

$$\begin{aligned} \|u(t^n) - u_N^n\|_{L^2(\Omega)} &\leq \tilde{C}(t^n) \left[N^{-r} \left(|u_0|_{H^r(\Omega)} + \int_0^{t^n} \left| \frac{\partial u}{\partial t}(s) \right|_{H^r(\Omega)} ds \right. \right. \\ &+ \left. \left. |u(t^n)|_{H^r(\Omega)} \right) + \Delta t \int_0^{t^n} \left\| \frac{\partial^2 u}{\partial t^2}(s) \right\|_{L^2(\Omega)} ds \right]. \end{aligned}$$

For the proof, refer to [CHQZ06], Chap. 7.

10.4 Generalization to the two-dimensional case

Let us consider as a domain the unit square $\Omega = (-1, 1)^2$. Since Ω is the tensor product of the one-dimensional interval $(-1, 1)$, it is natural to choose as nodes the points \mathbf{x}_{ij} whose coordinates both coincide with the one-dimensional GLL nodes x_i ,

$$\mathbf{x}_{ij} = (x_i, x_j), \quad i, j = 0, \dots, N,$$

while we take as weights the product of the corresponding one-dimensional weights

$$\alpha_{ij} = \alpha_i \alpha_j, \quad i, j = 0, \dots, N.$$

The Gauss-Legendre-Lobatto (GLL) integration formula in two dimensions is therefore defined by

$$\int_{\Omega} f(\mathbf{x}) d\Omega \simeq \sum_{i,j=0}^N \alpha_{ij} f(\mathbf{x}_{ij}),$$

while the discrete scalar product is given by

$$(f, g)_N = \sum_{i,j=0}^N \alpha_{ij} f(\mathbf{x}_{ij}) g(\mathbf{x}_{ij}). \quad (10.43)$$

Analogously to the one-dimensional case it can be proven that the integration formula (10.43) is exact whenever the integrand function is a polynomial of degree at most $2N - 1$. In particular, this implies that

$$(f, g)_N = (f, g) \quad \forall f, g \text{ s.t. } fg \in \mathbb{Q}_{2N-1}.$$

In this section, for each N , \mathbb{Q}_N denotes the space of polynomials of degree less than or equal to N with respect to each of the variables, introduced in (10.2).

We now consider as an example the problem

$$\begin{cases} Lu = -\operatorname{div}(\mu \nabla u) + \sigma u = f & \text{in } \Omega = (-1, 1)^2, \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$

By assuming that $\mu(\mathbf{x}) \geq \mu_0 > 0$ and $\sigma(\mathbf{x}) \geq 0$, the corresponding bilinear form is coercive in $H_0^1(\Omega)$. Its G-NI approximation is given by

$$\text{find } u_N \in V_N : \quad a_N(u_N, v_N) = F_N(v_N) \quad \forall v_N \in V_N,$$

where

$$V_N = \{v \in \mathbb{Q}_N : v|_{\partial\Omega} = 0\},$$

$$a_N(u, v) = (\mu \nabla u, \nabla v)_N + (\sigma u, v)_N$$

and

$$F_N(v_N) = (f, v_N)_N.$$

As shown in the one-dimensional case, also in more dimensions it can be verified that the G-NI formulation is equivalent to a collocation method where the L operator is replaced by L_N , the pseudo-spectral operator obtained by approximating each derivative by the corresponding interpolation derivative (10.40).

In the case of spectral element methods, we will need to generalize the GLL numerical integration formula on each element Ω_k . This can be done thanks to the transformation $\varphi_k : \widehat{\Omega} \rightarrow \Omega_k$ (see Fig. 10.2). Indeed, we can first of all generate the GLL nodes on the generic element Ω_k , by setting

$$\mathbf{x}_{ij}^{(k)} = \varphi_k(\mathbf{x}_{ij}), \quad i, j = 0, \dots, N,$$

then defining the corresponding weights

$$\alpha_{ij}^{(k)} = \alpha_{ij} |\det J_k| = \alpha_{ij} \frac{|\Omega_k|}{4}, \quad i, j = 0, \dots, N,$$

having denoted by J_k the Jacobian of the transformation φ_k and by $|\Omega_k|$ the measure of Ω_k . The GLL integration formula on Ω_k hence becomes

$$\int_{\Omega_k} f(x) dx \simeq I_{N,k}^{GLL}(f) = \sum_{i,j=0}^N \alpha_{ij}^{(k)} f(\mathbf{x}_{ij}^{(k)}). \quad (10.44)$$

The spectral element formulation with gaussian numerical integration, which we will denote by SEM-NI, then becomes

$$\text{find } u_N \in V_N^C : \quad a_{C,N}(u_N, v_N) = F_{C,N}(v_N) \quad \forall v_N \in V_N^C. \quad (10.45)$$

We have set

$$a_{C,N}(u_N, v_N) = \sum_k a_{\Omega_k, N}(u_N, v_N)$$

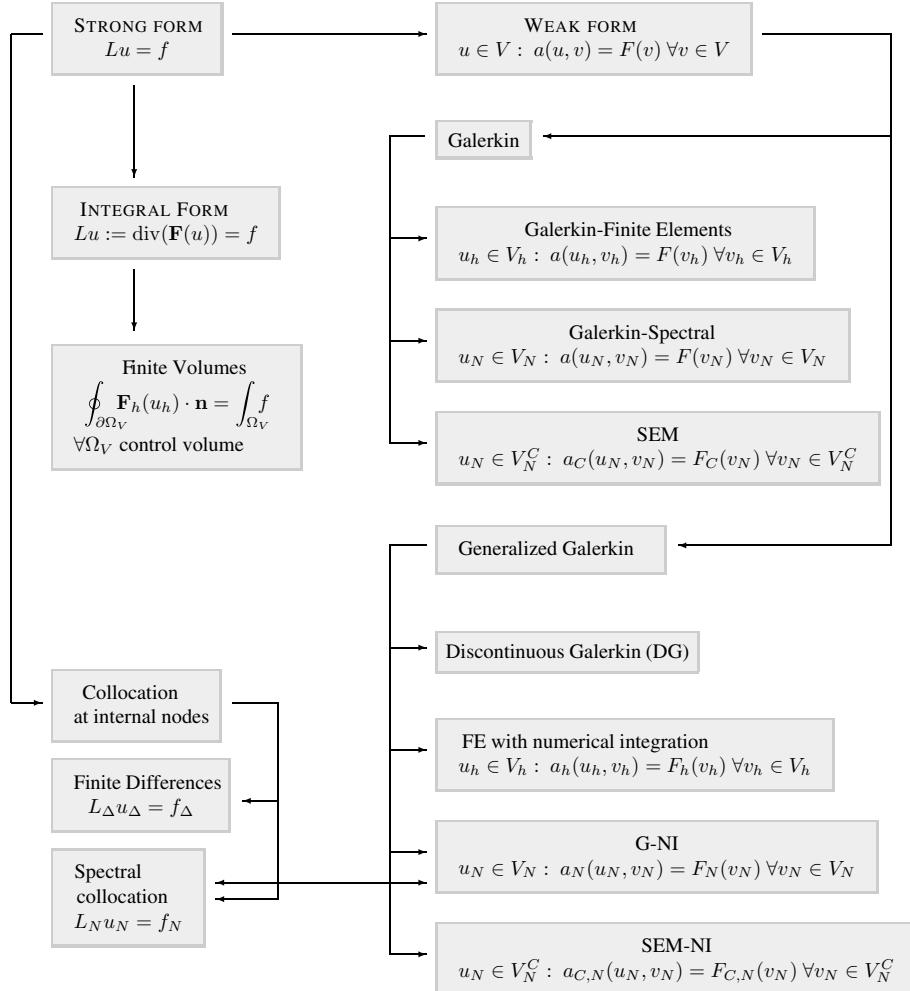


Fig. 10.10. Reference frame for the main numerical methods addressed in this book

where $a_{\Omega_k, N}(u_N, v_N)$ is the approximation of $a_{\Omega_k}(u_N, v_N)$ obtained by approximating each integral on Ω_k that appears in its bi-linear form via the GLL numerical integration formula in Ω_k (10.44). The term $F_{C, N}$ is defined in a similar way, and precisely $F_{C, N}(v_N) = \sum_k F_{\Omega_k, N}(v_N)$, where $F_{\Omega_k, N}$ is obtained, in turn, by replacing

$$\int_{\Omega_k} f v_N d\mathbf{x} \text{ with the formula } I_{N,k}^{GLL}(f v_N) \text{ for each } k.$$

Remark 10.3 Fig. 10.10 summarizes rather schematically the origin of the different approximation schemes evoked up to now. In the case of finite differences, we have denoted by L_Δ the discretization of the operator through finite difference schemes applied to the various derivatives appearing in the definition of L . •

10.4.1 Convergence of the G-NI method

As observed in the one-dimensional case, the G-NI method can be considered as a generalized Galerkin method. For the latter, the analysis of convergence is based on the following general result:

Lemma 10.1 (Strang) *Consider the problem*

$$\text{find } u \in V : \quad a(u, v) = F(v) \quad \forall v \in V, \quad (10.46)$$

where V is a Hilbert space with norm $\|\cdot\|_V$, $F \in V'$ a linear and bounded functional on V and $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$ a bilinear, continuous and coercive form on V .

Moreover, assume an approximation of (10.46) that can be formulated through the following generalized Galerkin problem

$$\text{find } u_h \in V_h : \quad a_h(u_h, v_h) = F_h(v_h) \quad \forall v_h \in V_h, \quad (10.47)$$

$\{V_h, h > 0\}$ being a family of finite-dimensional subspaces of V .

Let us suppose that the discrete bilinear form $a_h(\cdot, \cdot)$ be continuous on $V_h \times V_h$, and uniformly coercive on V_h , that is

$$\exists \alpha^* > 0 \text{ independent of } h \text{ s.t. } a_h(v_h, v_h) \geq \alpha^* \|v_h\|_V^2 \quad \forall v_h \in V_h.$$

Furthermore, let us suppose that F_h is a linear and bounded functional on V_h . Then:

1. there exists one unique solution u_h to problem (10.47);
2. such solution depends continuously on the data, i.e. we have

$$\|u_h\|_V \leq \frac{1}{\alpha^*} \sup_{v_h \in V_h \setminus \{0\}} \frac{F_h(v_h)}{\|v_h\|_V};$$

3. finally, the following a priori error estimate holds

$$\begin{aligned} \|u - u_h\|_V &\leq \inf_{w_h \in V_h} \left\{ \left(1 + \frac{M}{\alpha^*} \right) \|u - w_h\|_V \right. \\ &\quad \left. + \frac{1}{\alpha^*} \sup_{v_h \in V_h \setminus \{0\}} \frac{|a(w_h, v_h) - a_h(w_h, v_h)|}{\|v_h\|_V} \right\} \\ &\quad + \frac{1}{\alpha^*} \sup_{v_h \in V_h \setminus \{0\}} \frac{|F(v_h) - F_h(v_h)|}{\|v_h\|_V}, \end{aligned} \quad (10.48)$$

M being the continuity constant of the bilinear form $a(\cdot, \cdot)$.

Proof. The Lax-Milgram lemma hypotheses for problem (10.47) being satisfied, the solution of such problem exists and is unique. Moreover,

$$\|u_h\|_V \leq \frac{1}{\alpha^*} \|F_h\|_{V'_h},$$

$$\|F_h\|_{V'_h} = \sup_{v_h \in V_h \setminus \{0\}} \frac{F_h(v_h)}{\|v_h\|_V} \text{ being the norm of the dual space } V'_h \text{ of } V_h.$$

Let us now prove the error inequality (10.48). Let $u_h \in V_h$ be the solution of problem (10.47) and let w_h be any function of the subspace V_h . Setting $\sigma_h = u_h - w_h \in V_h$, we have:

$$\begin{aligned} \alpha^* \|\sigma_h\|_V^2 &\leq a_h(\sigma_h, \sigma_h) \quad [\text{by the coercivity of } a_h] \\ &= a_h(u_h, \sigma_h) - a_h(w_h, \sigma_h) \\ &= F_h(\sigma_h) - a_h(w_h, \sigma_h) \quad [\text{thanks to (10.47)}] \\ &= F_h(\sigma_h) - F(\sigma_h) + F(\sigma_h) - a_h(w_h, \sigma_h) \\ &= [F_h(\sigma_h) - F(\sigma_h)] + a(u, \sigma_h) - a_h(w_h, \sigma_h) \quad [\text{thanks to (10.46)}] \\ &= [F_h(\sigma_h) - F(\sigma_h)] + a(u - w_h, \sigma_h) + [a(w_h, \sigma_h) - a_h(w_h, \sigma_h)]. \end{aligned} \tag{10.49}$$

If $\sigma_h \neq 0$, (10.49) can be divided by $\alpha^* \|\sigma_h\|_V$, obtaining

$$\begin{aligned} \|\sigma_h\|_V &\leq \frac{1}{\alpha^*} \left\{ \frac{|a(u - w_h, \sigma_h)|}{\|\sigma_h\|_V} + \frac{|a(w_h, \sigma_h) - a_h(w_h, \sigma_h)|}{\|\sigma_h\|_V} \right. \\ &\quad \left. + \frac{|F_h(\sigma_h) - F(\sigma_h)|}{\|\sigma_h\|_V} \right\} \\ &\leq \frac{1}{\alpha^*} \left\{ M \|u - w_h\|_V + \sup_{v_h \in V_h \setminus \{0\}} \frac{|a(w_h, v_h) - a_h(w_h, v_h)|}{\|v_h\|_V} \right. \\ &\quad \left. + \sup_{v_h \in V_h \setminus \{0\}} \frac{|F_h(v_h) - F(v_h)|}{\|v_h\|_V} \right\} \quad [\text{by the continuity of } a]. \end{aligned}$$

If $\sigma_h = 0$ such inequality is still valid (as it states that 0 is smaller than a sum of positive terms), although the process through which it has been obtained is no longer valid.

We can now estimate the error between the solution u of (10.46) and the solution u_h of (10.47). Since

$$u - u_h = (u - w_h) - \sigma_h,$$

we obtain

$$\begin{aligned}
\|u - u_h\|_V &\leq \|u - w_h\|_V + \|\sigma_h\|_V \leq \|u - w_h\|_V \\
&+ \frac{1}{\alpha^*} \left\{ M \|u - w_h\|_V + \sup_{v_h \in V_h \setminus \{0\}} \frac{|a(w_h, v_h) - a_h(w_h, v_h)|}{\|v_h\|_V} \right. \\
&+ \left. \sup_{v_h \in V_h \setminus \{0\}} \frac{|F_h(v_h) - F(v_h)|}{\|v_h\|_V} \right\} \\
&= \left(1 + \frac{M}{\alpha^*} \right) \|u - w_h\|_V + \frac{1}{\alpha^*} \sup_{v_h \in V_h \setminus \{0\}} \frac{|a(w_h, v_h) - a_h(w_h, v_h)|}{\|v_h\|_V} \\
&+ \frac{1}{\alpha^*} \sup_{v_h \in V_h \setminus \{0\}} \frac{|F_h(v_h) - F(v_h)|}{\|v_h\|_V}.
\end{aligned}$$

If the previous inequality holds $\forall w_h \in V_h$, it also holds when taking the infimum when w_h varies in V_h . Hence, we obtain (10.48). \diamond

By observing the right-hand side of inequality (10.48), we can recognize three different contributions to the approximation error $u - u_h$: the first is the best approximation error, the second is the error deriving from the approximation of the bilinear form $a(\cdot, \cdot)$ using the discrete bilinear form $a_h(\cdot, \cdot)$, and the third is the error arising from the approximation of the linear functional $F(\cdot)$ by the discrete linear functional $F_h(\cdot)$.

Remark 10.4 Note that, if we choose in (10.49) $w_h = u_h^*$, u_h^* being the solution to the Galerkin problem

$$u_h^* \in V_h : a(u_h^*, v_h) = F(v_h) \quad \forall v_h \in V_h,$$

then term $a(u - w_h, \sigma_h)$ in (10.49) is null thanks to (10.46). It is therefore possible to obtain the following estimate, alternative to (10.48)

$$\begin{aligned}
\|u - u_h\|_V &\leq \|u - u_h^*\|_V \\
&+ \frac{1}{\alpha^*} \sup_{v_h \in V_h \setminus \{0\}} \frac{|a(u_h^*, v_h) - a_h(u_h^*, v_h)|}{\|v_h\|_V} \\
&+ \frac{1}{\alpha^*} \sup_{v_h \in V_h \setminus \{0\}} \frac{|F(v_h) - F_h(v_h)|}{\|v_h\|_V}.
\end{aligned}$$

The latter highlights the fact that the error due to the generalized Galerkin method can be bounded by the error of the Galerkin method plus the errors induced by the use of numerical integration for the computation of both $a(\cdot, \cdot)$ and $F(\cdot)$. \bullet

We now want to apply the Strang lemma to the G-NI method, to verify its convergence, by limiting ourselves, for simplicity, to the one-dimensional case. Obviously, V_h will be replaced by V_N , u_h by u_N , v_h by v_N and w_h by w_N . First of all, we begin by computing the error of the GLL numerical integration formula

$$E(g, v_N) = (g, v_N) - (g, v_N)_N,$$

g and v_N being a generic continuous function and a generic polynomial of \mathbb{Q}_N , respectively. By introducing the interpolation polynomial $\Pi_N^{GLL}g$ defined according to (10.14), we obtain

$$\begin{aligned} E(g, v_N) &= (g, v_N) - (\Pi_N^{GLL}g, v_N)_N \\ &= (g, v_N) - (\Pi_{N-1}^{GLL}g, v_N) + \underbrace{(\Pi_{N-1}^{GLL}g, v_N)}_{\in \mathbb{Q}_{2N-1}} - (\Pi_N^{GLL}g, v_N)_N \\ &= (g, v_N) - (\Pi_{N-1}^{GLL}g, v_N) \\ &\quad + (\Pi_{N-1}^{GLL}g, v_N)_N - (\Pi_N^{GLL}g, v_N)_N \quad [\text{by (10.27)}] \\ &= (g - \Pi_{N-1}^{GLL}g, v_N) + (\Pi_{N-1}^{GLL}g - \Pi_N^{GLL}g, v_N)_N. \end{aligned} \tag{10.50}$$

The first addendum of the right-hand side can be upper bounded using the Cauchy-Schwarz inequality as follows

$$|(g - \Pi_{N-1}^{GLL}g, v_N)| \leq \|g - \Pi_{N-1}^{GLL}g\|_{L^2(-1,1)} \|v_N\|_{L^2(-1,1)}. \tag{10.51}$$

To find an upper bound for the second addendum, we must first introduce the two following lemmas, for the proof of which we refer to [CHQZ06]:

Lemma 10.2 *The discrete scalar product $(\cdot, \cdot)_N$ defined in (10.25) is a scalar product on \mathbb{Q}_N and, as such, satisfies the Cauchy-Schwarz inequality*

$$|(\varphi, \psi)_N| \leq \|\varphi\|_N \|\psi\|_N, \tag{10.52}$$

where the discrete norm $\|\cdot\|_N$ is given by

$$\|\varphi\|_N = \sqrt{(\varphi, \varphi)_N} \quad \forall \varphi \in \mathbb{Q}_N. \tag{10.53}$$

Lemma 10.3 *The “continuous” norm of $L^2(-1, 1)$ and the “discrete” norm $\|\cdot\|_N$ defined in (10.53) verify the inequalities*

$$\|v_N\|_{L^2(-1,1)} \leq \|v_N\|_N \leq \sqrt{3} \|v_N\|_{L^2(-1,1)} \quad \forall v_N \in \mathbb{Q}_N, \quad (10.54)$$

hence they are uniformly equivalent on \mathbb{Q}_N .

By using first (10.53) and then (10.54) we obtain

$$\begin{aligned} |(\Pi_{N-1}^{GLL} g - \Pi_N^{GLL} g, v_N)_N| &\leq \|\Pi_{N-1}^{GLL} g - \Pi_N^{GLL} g\|_N \|v_N\|_N \\ &\leq 3 [\|\Pi_{N-1}^{GLL} g - g\|_{L^2(-1,1)} + \|\Pi_N^{GLL} g - g\|_{L^2(-1,1)}] \|v_N\|_{L^2(-1,1)}. \end{aligned}$$

Using such inequality and (10.51), from (10.50) we can obtain the following upper bound

$$|E(g, v_N)| \leq [4\|\Pi_{N-1}^{GLL} g - g\|_{L^2(-1,1)} + 3\|\Pi_N^{GLL} g - g\|_{L^2(-1,1)}] \|v_N\|_{L^2(-1,1)}.$$

Using the interpolation estimate (10.17), we have that

$$|E(g, v_N)| \leq C \left[\left(\frac{1}{N-1} \right)^s + \left(\frac{1}{N} \right)^s \right] \|g\|_{H^s(-1,1)} \|v_N\|_{L^2(-1,1)},$$

provided that $g \in H^s(-1, 1)$, for some $s \geq 1$. Finally, as for each $N \geq 2$, $1/(N-1) \leq 2/N$, the Gauss-Legendre-Lobatto integration error results to be bound as

$$|E(g, v_N)| \leq C \left(\frac{1}{N} \right)^s \|g\|_{H^s(-1,1)} \|v_N\|_{L^2(-1,1)}, \quad (10.55)$$

for each $g \in H^s(-1, 1)$ and for each polynomial $v_N \in \mathbb{Q}_N$.

At this point we are ready to evaluate the various contributions that intervene in (10.48). We anticipate that this analysis will be carried out in the case where suitable simplifying hypotheses are introduced on the differential problem under exam. We begin with the simplest term, i.e. the one associated with the functional F , supposing to consider a problem with homogeneous Dirichlet boundary conditions, in order to obtain $F(v_N) = (f, v_N)$ and $F_N(v_N) = (f, v_N)_N$. We then have, provided that $f \in H^s(-1, 1)$ for some $s \geq 1$,

$$\begin{aligned} \sup_{v_N \in V_N \setminus \{0\}} \frac{|F(v_N) - F_N(v_N)|}{\|v_N\|_V} &= \sup_{v_N \in V_N \setminus \{0\}} \frac{|(f, v_N) - (f, v_N)_N|}{\|v_N\|_V} \\ &= \sup_{v_N \in V_N \setminus \{0\}} \frac{|E(f, v_N)|}{\|v_N\|_V} \leq \sup_{v_N \in V_N \setminus \{0\}} \frac{C \left(\frac{1}{N} \right)^s \|f\|_{H^s(-1,1)} \|v_N\|_{L^2(-1,1)}}{\|v_N\|_V} \quad (10.56) \\ &\leq C \left(\frac{1}{N} \right)^s \|f\|_{H^s(-1,1)}, \end{aligned}$$

having exploited relation (10.55) and having bounded the norm in $L^2(-1, 1)$ by that in $H^s(-1, 1)$.

As of the contribution arising from the approximation of the bilinear form,

$$\sup_{v_N \in V_N \setminus \{0\}} \frac{|a(w_N, v_N) - a_N(w_N, v_N)|}{\|v_N\|_V},$$

we cannot explicitly evaluate it without referring to a particular differential problem. We then choose, as an example, the one-dimensional diffusion-reaction problem (10.22), supposing moreover that μ and σ are constant. Incidentally, such problem satisfies homogeneous Dirichlet boundary conditions, in accordance with what was requested for deriving estimate (10.56). In such case, the associated bilinear form is

$$a(u, v) = (\mu u', v') + (\sigma u, v),$$

while its G-NI approximation is given by

$$a_N(u, v) = (\mu u', v')_N + (\sigma u, v)_N.$$

We must then evaluate

$$a(w_N, v_N) - a_N(w_N, v_N) = (\mu w'_N, v'_N) - (\mu w'_N, v'_N)_N + (\sigma w_N, v_N) - (\sigma w_N, v_N)_N.$$

Since $w'_N v'_N \in \mathbb{Q}_{2N-2}$, if we suppose that μ is constant, the product $\mu w'_N v'_N$ is integrated exactly by the GLL integration formula, that is $(\mu w'_N, v'_N) - (\mu w'_N, v'_N)_N = 0$. We now observe that

$$(\sigma w_N, v_N) - (\sigma w_N, v_N)_N = E(\sigma w_N, v_N) = E(\sigma(w_N - u), v_N) + E(\sigma u, v_N),$$

and therefore, using (10.55), we obtain

$$\begin{aligned} |E(\sigma(w_N - u), v_N)| &\leq C \left(\frac{1}{N} \right) \|\sigma(w_N - u)\|_{H^1(-1, 1)} \|v_N\|_{L^2(-1, 1)}, \\ |E(\sigma u, v_N)| &\leq C \left(\frac{1}{N} \right)^s \|\sigma u\|_{H^s(-1, 1)} \|v_N\|_{L^2(-1, 1)}. \end{aligned}$$

On the other hand, since σ is also constant, setting $w_N = \Pi_N^{GLL} u$ and using (10.17), we obtain

$$\|\sigma(w_N - u)\|_{H^1(-1, 1)} \leq C \|u - \Pi_N^{GLL} u\|_{H^1(-1, 1)} \leq C \left(\frac{1}{N} \right)^{s-1} \|u\|_{H^s(-1, 1)}.$$

Henceforth,

$$\sup_{v_N \in V_N \setminus \{0\}} \frac{|a(w_N, v_N) - a_N(w_N, v_N)|}{\|v_N\|_V} \leq C^* \left(\frac{1}{N} \right)^s \|u\|_{H^s(-1, 1)}. \quad (10.57)$$

We still need to estimate the first addendum of (10.48). Having chosen $w_N = \Pi_N^{GLL} u$ and exploiting (10.17) again, we obtain that

$$\|u - w_N\|_V = \|u - \Pi_N^{GLL} u\|_{H^1(-1,1)} \leq C \left(\frac{1}{N} \right)^s \|u\|_{H^{s+1}(-1,1)} \quad (10.58)$$

provided that $u \in H^{s+1}(-1,1)$, for a suitable $s \geq 1$. To conclude, thanks to (10.56), (10.57) and (10.58), from (10.48) applied to the G-NI approximation of problem (10.22), under the previous hypotheses, we find the following error estimate

$$\|u - u_N\|_{H^1(-1,1)} \leq C \left(\frac{1}{N} \right)^s (\|f\|_{H^s(-1,1)} + \|u\|_{H^{s+1}(-1,1)}).$$

The convergence analysis just carried out for the model problem (10.22) can be generalized (with a few technical difficulties) to the case of more complex differential problems and different boundary conditions.

Example 10.2 (Problem with regularity depending on a parameter) Let us consider the following (trivial but instructive) problem

$$\begin{cases} -u'' = 0, & x \in (0,1], \\ -u'' = -\alpha(\alpha-1)(x-1)^{\alpha-2}, & x \in (1,2), \\ u(0) = 0, & u(2) = 1, \end{cases}$$

with $\alpha \in \mathbb{N}$. The exact solution is null in $(0,1)$ and equals $(x-1)^\alpha$ for $x \in (1,2)$. Thus it belongs to $H^\alpha(0,2)$, but not to $H^{\alpha+1}(0,2)$. We report in Table 10.1 the behavior of the error in $H^1(0,2)$ norm with respect to N using a G-NI method for three different values of α . As it can be seen, when the regularity increases, so does the order of convergence of the spectral method with respect to N , as stated by the theory. In the same table we report the results obtained using linear finite elements (this time N denotes the number of elements). The order of convergence of the finite element method remains linear in any case. ■

Example 10.3 Let us take the second example illustrated in Sec. 4.6.3, this time using the spectral element method. Let us consider a partition of the domain into four spectral elements of degree 8 as shown in the left of Fig. 10.11. The obtained solution (Fig. 10.11, left) does not exhibit any inaccuracy in proximity of the origin, as opposed to the solution obtained using finite elements in the absence of grid adaptivity (compare with Fig. 4.24, left). ■

Table 10.1. Behavior of the error of the G-NI spectral method for varying polynomial degree N and solution regularity index (left). Behavior of the error of the linear finite element method for varying number of intervals N and solution regularity index (right)

N	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	N	$\alpha = 2$	$\alpha = 3$
4	0.5931	0.2502	0.2041	4	0.4673	0.5768
8	0.3064	0.0609	0.0090	8	0.2456	0.3023
16	0.1566	0.0154	$7.5529 \cdot 10^{-4}$	16	0.1312	0.1467
32	0.0792	0.0039	$6.7934 \cdot 10^{-5}$	32	0.0745	0.0801

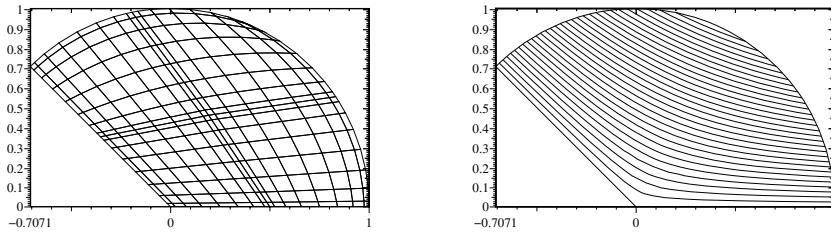


Fig. 10.11. The grid (left) and the solution isolines obtained using the spectral finite element method (right) for the problem in Example 10.3

10.5 G-NI and SEM-NI methods for a one-dimensional model problem

Let us consider the one-dimensional diffusion-reaction problem

$$-[(1 + x^2) u'(x)]' + \cos(x^2) u(x) = f(x), \quad x \in (-1, 1), \quad (10.59)$$

completed with mixed-type boundary conditions

$$u(-1) = 0, \quad u'(1) = 1.$$

The objective of this section is to discuss in detail how to formulate the G-NI and SEM-NI approximations. For the former, we will also provide the corresponding matrix formulation as well as a stability analysis.

10.5.1 The G-NI method

The weak formulation of problem (10.59) is:

$$\text{find } u \in V : a(u, v) = F(v) \quad \forall v \in V$$

$V = \{v \in H^1(-1, 1) : v(-1) = 0\}$, $a : V \times V \rightarrow \mathbb{R}$ and $F : V \rightarrow \mathbb{R}$ being the bi-linear form and the linear functional, respectively, defined by

$$\begin{aligned} a(u, v) &= \int_{-1}^1 (1 + x^2) u'(x) v'(x) dx + \int_{-1}^1 \cos(x^2) u(x) v(x) dx, \\ F(v) &= \int_{-1}^1 f(x) v(x) dx + 2 v(1). \end{aligned}$$

The spectral-Galerkin formulation (SM) takes the following form

$$\text{find } u_N \in V_N \quad \text{s.t.} \quad a(u_N, v_N) = F(v_N) \quad \forall v_N \in V_N, \quad (10.60)$$

with

$$V_N = \{v_N \in \mathbb{Q}_N : v_N(-1) = 0\} \subset V. \quad (10.61)$$

In order to obtain the corresponding G-NI formulation, it is sufficient to approximate in (10.60) all the scalar products on $L^2(-1, 1)$ with the GLL discrete scalar product defined in (10.25). We then have

$$\text{find } u_N^* \in V_N : a_N(u_N^*, v_N) = F_N(v_N) \quad \forall v_N \in V_N, \quad (10.62)$$

having set

$$\begin{aligned} a_N(u, v) &= ((1 + x^2) u', v')_N + (\cos(x^2) u, v)_N \\ &= \sum_{i=0}^N (1 + x_i^2) u'(x_i) v'(x_i) \alpha_i + \sum_{i=1}^N \cos(x_i^2) u(x_i) v(x_i) \alpha_i \end{aligned} \quad (10.63)$$

and

$$F_N(v) = (f, v)_N + 2 v(1) = \sum_{i=1}^N f(x_i) v(x_i) \alpha_i + 2 v(1). \quad (10.64)$$

Note that this requires f to be continuous. We observe that the index i of the last sum in (10.63) and of the sum in (10.64) starts from 1, instead of 0, since $v(x_0) = v(-1) = 0$. Moreover, the SM formulations (10.60) and G-NI (10.62) never coincide. Consider, for instance, the diffusive term $(1 + x^2)(u_N^*)' v'_N$: this is a polynomial of degree $2N$. Since the GLL integration formula has exactness degree $2N-1$, the discrete scalar product (10.25) will not return the exact value of the corresponding continuous scalar product $((1 + x^2)(u_N^*)', v'_N)$.

To obtain the matrix formulation of the G-NI approximation, we denote by ψ_i , for $i = 1, \dots, N$, the characteristic polynomials associated to all of the GLL nodes except to the one where a Dirichlet boundary condition is assigned, $x_0 = -1$. Such polynomials constitute a basis for the space V_N introduced in (10.61). This allows us, in the first place, to write the solution u_N^* of the G-NI formulation as

$$u_N^*(x) = \sum_{j=1}^N u_N^*(x_j) \psi_j(x).$$

Secondly, we can choose in (10.62) $v_N = \psi_i$, $i = 1, \dots, N$, obtaining

$$a_N(u_N^*, \psi_i) = F_N(\psi_i), \quad i = 1, \dots, N,$$

i.e.

$$\sum_{j=1}^N u_N^*(x_j) a_N(\psi_j, \psi_i) = F_N(\psi_i), \quad i = 1, \dots, N.$$

In matrix form,

$$A \mathbf{u}_N^* = \mathbf{f},$$

having $\mathbf{u}_N^* = (u_N^*(x_i))$, $A = (a_{ij})$, with

$$\begin{aligned} a_{ij} = a_N(\psi_j, \psi_i) &= \sum_{k=0}^N (1 + x_k^2) \psi'_j(x_k) \psi'_i(x_k) \alpha_k \\ &+ \sum_{k=1}^N \cos(x_k^2) \psi_j(x_k) \psi_i(x_k) \alpha_k \\ &= \sum_{k=0}^N (1 + x_k^2) \psi'_j(x_k) \psi'_i(x_k) \alpha_k + \cos(x_i^2) \alpha_i \delta_{ij}, \end{aligned}$$

and

$$\begin{aligned} \mathbf{f} = (f_i), \text{ con } f_i &= F_N(\psi_i) = (f, \psi_i)_N + 2 \psi_i(1) \\ &= \sum_{k=1}^N f(x_k) \psi_i(x_k) \alpha_k + 2 \psi_i(1) \\ &= \begin{cases} \alpha_i f(x_i) & \text{for } i = 1, \dots, N-1, \\ \alpha_N f(1) + 2 & \text{for } i = N. \end{cases} \end{aligned}$$

We recall that matrix A , besides being ill-conditioned, is full due to the presence of the diffusive term.

Finally, we can verify that the G-NI method (10.62) can be reformulated as a suitable collocation method. To this end, we wish to rewrite the discrete formulation (10.62) in continuous form in order to counterintegrate by parts, i.e. to reconnect to the initial differential operator. In order to do this, we will resort to the interpolation operator Π_N^{GLL} defined in (10.15), recalling in addition that the discrete scalar product (10.25) coincides with the continuous one on $L^2(-1, 1)$ if the product of the two integrand functions is a polynomial of degree $\leq 2N - 1$ (see (10.36)).

We then accurately rewrite the first addendum of $a_N(u_N^*, v_N)$, ignoring the apex $*$ to simplify the notation. Thanks to (10.36) and integrating by parts, we have

$$\begin{aligned} &((1 + x^2) u'_N, v'_N)_N \\ &= (\Pi_N^{GLL}((1 + x^2) u'_N), v'_N)_N = (\Pi_N^{GLL}((1 + x^2) u'_N), v'_N) \\ &= -([\Pi_N^{GLL}((1 + x^2) u'_N)]', v_N) + \Pi_N^{GLL}((1 + x^2) u'_N)(1) v_N(1) \\ &= -([\Pi_N^{GLL}((1 + x^2) u'_N)]', v_N)_N + \Pi_N^{GLL}((1 + x^2) u'_N)(1) v_N(1). \end{aligned}$$

Hence, we can reformulate (10.62) as

$$\begin{aligned} \text{find } u_N \in V_N &: (L_N u_N, v_N)_N = (f, v_N)_N \\ &+ (2 - \Pi_N^{GLL}((1 + x^2) u'_N)(1)) v_N(1) \quad \forall v_N \in V_N, \end{aligned} \tag{10.65}$$

with

$$L_N u_N = -[\Pi_N^{GLL}((1+x^2) u'_N)]' + \cos(x^2) u_N = -D_N((1+x^2) u'_N) + \cos(x^2) u_N,$$

D_N being the interpolation derivative introduced in (10.40). We now choose (10.65) $v_N = \psi_i$. For $i = 1, \dots, N-1$, we have

$$\begin{aligned} (L_N u_N, \psi_i)_N &= (-[\Pi_N^{GLL}((1+x^2) u'_N)]', \psi_i)_N + (\cos(x^2) u_N, \psi_i)_N \\ &= -\sum_{j=1}^{N-1} \alpha_j [\Pi_N^{GLL}((1+x^2) u'_N)]'(x_j) \psi_i(x_j) + \sum_{j=1}^{N-1} \alpha_j \cos(x_j^2) u_N(x_j) \psi_i(x_j) \\ &= -\alpha_i [\Pi_N^{GLL}((1+x^2) u'_N)]'(x_i) + \alpha_i \cos(x_i^2) u_N(x_i) = (f, \psi_i)_N \\ &= \sum_{j=1}^{N-1} \alpha_j f(x_j) \psi_i(x_j) = \alpha_i f(x_i), \end{aligned}$$

that is, exploiting the definition of the L_N operator and dividing all by α_i ,

$$L_N u_N(x_i) = f(x_i), \quad i = 1, \dots, N-1. \quad (10.66)$$

Having set $v_N = \psi_N$ in (10.65), we obtain instead

$$\begin{aligned} (L_N u_N, \psi_N)_N &= -\alpha_N [\Pi_N^{GLL}((1+x^2) u'_N)]'(x_N) + \alpha_N \cos(x_N^2) u_N(x_N) \\ &= (f, \psi_N)_N + 2 - \Pi_N^{GLL}((1+x^2) u'_N)(1) \\ &= \alpha_N f(x_N) + 2 - \Pi_N^{GLL}((1+x^2) u'_N)(1), \end{aligned}$$

or, dividing all by α_N ,

$$L_N u_N(x_N) = f(x_N) + \frac{1}{\alpha_N} (2 - \Pi_N^{GLL}((1+x^2) u'_N)(1)). \quad (10.67)$$

Equations (10.66) and (10.67) therefore provide the collocation in all the nodes (except the potential boundary ones where Dirichlet conditions are assigned) of the given differential problem, further to the approximation of the differential operator L using operator L_N .

Finally, we analyze the stability of the formulation (10.62). Since we are dealing with a generalized Galerkin type of approach, we will have to resort to the Strang lemma 10.1 which guarantees that, for the solution u_N^* di (10.62), the estimate

$$\|u_N^*\|_V \leq \frac{1}{\alpha^*} \sup_{v_N \in V_N \setminus \{0\}} \frac{|F_N(v_N)|}{\|v_N\|_V} \quad (10.68)$$

holds, α^* being the (uniform) coercivity constant associated to the discrete bilinear form $a_N(\cdot, \cdot)$. We make such result specific to problem (10.59), by computing first

of all α^* . By exploiting the definition (10.53) of the discrete norm $\|\cdot\|_N$ and the equivalence relation (10.54), we have

$$\begin{aligned} a_N(u_N, u_N) &= ((1 + x^2) u'_N, u'_N)_N + (\cos(x^2) u_N, u_N)_N \\ &\geq (u'_N, u'_N)_N + \cos(1) (u_N, u_N)_N = \|u'_N\|_N^2 + \cos(1) \|u_N\|_N^2 \\ &\geq \|u'_N\|_{L^2(-1,1)}^2 + \cos(1) \|u_N\|_{L^2(-1,1)}^2 \geq \cos(1) \|u_N\|_V^2, \end{aligned}$$

having moreover exploited the relations

$$\begin{aligned} \min_j (1 + x_j^2) &\geq \min_{x \in [-1,1]} (1 + x^2) = 1, \\ \min_j \cos(x_j^2) &\geq \min_{x \in [-1,1]} \cos(x^2) = \cos(1). \end{aligned}$$

This allows us to identify α^* using the value $\cos(1)$. At this point, we can evaluate the quotient $|F_N(v_N)|/\|v_N\|_V$ in (10.68). Indeed, we have

$$\begin{aligned} |F_N(v_N)| &= |(f, v_N)_N + 2 v_N(1)| \leq \|f\|_N \|v_N\|_N + 2 |v_N(1)| \\ &\leq \sqrt{3} \|f\|_N \|v_N\|_V + 2 \left| \int_{-1}^1 v'_N(x) dx \right| \leq \sqrt{3} \|f\|_N \|v_N\|_V + 2 \sqrt{2} \|v_N\|_V, \end{aligned}$$

having once more used the equivalence property (10.54) together with the Cauchy-Schwarz inequality in its discrete (10.52) and continuous (3.7) version. We can thus conclude that

$$\frac{|F_N(v_N)|}{\|v_N\|_V} \leq \sqrt{3} \|f\|_N + 2 \sqrt{2},$$

that is, returning to the stability estimate (10.68),

$$\|u_N^*\|_V \leq \frac{1}{\cos(1)} [\sqrt{3} \|f\|_N + 2 \sqrt{2}].$$

Finally, we note that $\|f\|_N \leq 2 \|f\|_{C^0([-1,1])} \forall f \in C^0([-1,1])$.

10.5.2 The SEM-NI method

Starting from problem (10.59), we now want to consider its SEM-NI formulation, i.e. a spectral element formulation using the integration formulae of type GLL in each element. Moreover, we propose to provide a basis for the space where such formulation will be implemented.

We first introduce a partition of the interval $(-1, 1)$ in M (≥ 2) disjoint sub-intervals $\Omega_m = (\bar{x}_{m-1}, \bar{x}_m)$, with $m = 1, \dots, M$, denoting by $h_m = \bar{x}_m - \bar{x}_{m-1}$ the width of the m -th interval, and setting $h = \max_m h_m$. The SEM formulation of problem (10.59) takes the form

$$\text{find } u_N \in V_N^C : a(u_N, v_N) = F(v_N) \quad \forall v_N \in V_N^C, \quad (10.69)$$

with

$$V_N^C = \{v_N \in C^0([-1, 1]) : v_N|_{\Omega_m} \in \mathbb{Q}_N, \forall m = 1, \dots, M, v_N(-1) = 0\}.$$

We note that the functional space V_N^C of the SEM approach loses the “global” nature that is instead typical of a SM formulation. Similarly to what happens in the case of finite element approximation, we now have piecewise polynomial functions. By exploiting the $\{\Omega_m\}$ partition, we can rewrite formulation (10.69) in the following way

$$\text{find } u_N \in V_N^C : \sum_{m=1}^M a_{\Omega_m}(u_N, v_N) = \sum_{m=1}^M F_{\Omega_m}(v_N) \quad \forall v_N \in V_N^C, \quad (10.70)$$

where

$$\begin{aligned} a_{\Omega_m}(u_N, v_N) &= a(u_N, v_N)|_{\Omega_m} \\ &= \int_{\bar{x}_{m-1}}^{\bar{x}_m} (1 + x^2) u'_N(x) v'_N(x) dx + \int_{\bar{x}_{m-1}}^{\bar{x}_m} \cos(x^2) u_N(x) v_N(x) dx, \end{aligned}$$

while

$$F_{\Omega_m}(v_N) = F(v_N)|_{\Omega_m} = \int_{\bar{x}_{m-1}}^{\bar{x}_m} f(x) v_N(x) dx + 2v_N(1)\delta_{mM}.$$

The SEM-NI formulation can be obtained at this point by replacing in (10.70) the continuous scalar products by the discrete GLL scalar product (10.25):

$$\text{find } u_N^* \in V_N^C : \sum_{m=1}^M a_{N,\Omega_m}(u_N^*, v_N) = \sum_{m=1}^M F_{N,\Omega_m}(v_N) \quad \forall v_N \in V_N^C,$$

where

$$a_{N,\Omega_m}(u, v) = ((1 + x^2) u', v')_{N,\Omega_m} + (\cos(x^2) u, v)_{N,\Omega_m},$$

$$F_{N,\Omega_m}(v) = (f, v)_{N,\Omega_m} + 2v(1)\delta_{mM},$$

$$(u, v)_{N,\Omega_m} = \sum_{i=0}^N u(x_i^{(m)}) v(x_i^{(m)}) \alpha_i^{(m)},$$

$x_i^{(m)}$ being the i -th GLL node of the sub-interval Ω_m and $\alpha_i^{(m)}$ the corresponding integration weight.

Starting from the reference element $\widehat{\Omega} = (-1, 1)$ (which, in the case under exam, coincides with the domain Ω of problem (10.59)) and denoted by

$$\varphi_m(\xi) = \frac{h_m}{2} \xi + \frac{\bar{x}_m + \bar{x}_{m-1}}{2}, \quad \xi \in [-1, 1],$$

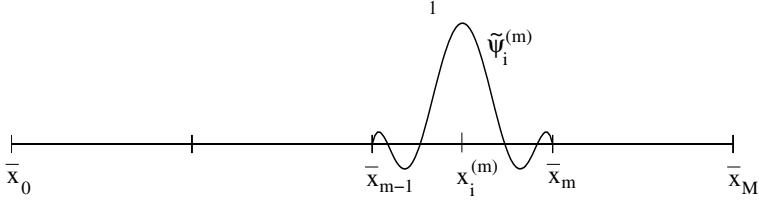


Fig. 10.12. basis function $\tilde{\psi}_i^{(m)}$ associated to the internal node $x_i^{(m)}$

the affine map from $\widehat{\Omega}$ into Ω_m , for $m = 1, \dots, M$, we will have

$$x_i^{(m)} = \varphi_m(x_i) = \alpha_i^{(m)} = \frac{h_m}{2}\alpha_i, \quad i = 0, \dots, N \quad (10.71)$$

that is $x_i^{(m)}$ and the image, through the mapping φ_m , of the i -th GLL node of $\widehat{\Omega}$.

We introduce, on each Ω_m , the set $\{\psi_i^{(m)}\}_{i=0}^N$ of basis functions, such that

$$\psi_i^{(m)}(x) = \psi_i(\varphi_m^{-1}(x)) \quad \forall x \in \Omega_m,$$

ψ_i being the characteristic polynomial introduced in (10.12) and (10.13) associated to node x_i of GLL in $\widehat{\Omega}$. Having now a basis for each sub-interval Ω_m , we can write the solution u_N of the SEM on each Ω_m as

$$u_N(x) = \sum_{i=0}^N u_i^{(m)} \psi_i^{(m)}(x) \quad \forall x \in \Omega_m, \quad (10.72)$$

having $u_i^{(m)} = u_N(x_i^{(m)})$.

Since we want to define a global basis for the space V_N^C , we start by defining the basis functions associated to the internal nodes of Ω_m , for $m = 1, \dots, M$. For this purpose, it will be sufficient to extend by zero, outside Ω_m , each basis function $\psi_i^{(m)}$, yielding

$$\tilde{\psi}_i^{(m)}(x) = \begin{cases} \psi_i^{(m)}(x), & x \in \Omega_m \\ 0, & \text{otherwise.} \end{cases}$$

These functions are in total $(N - 1)M$ and behave as shown in Fig. 10.12. For each extreme node \bar{x}_m of the Ω_m sub-domains, with $m = 1, \dots, M - 1$, we define instead the basis function

$$\psi_m^*(x) = \begin{cases} \psi_N^{(m)}(x), & x \in \Omega_m \\ \psi_0^{(m+1)}(x), & x \in \Omega_{m+1} \\ 0, & \text{otherwise,} \end{cases}$$

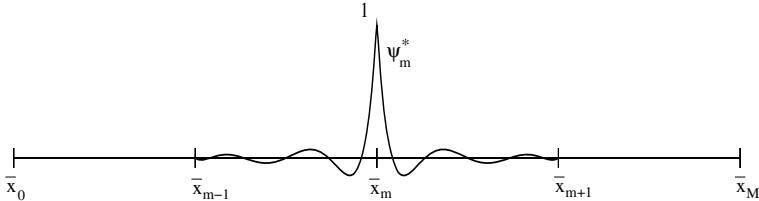


Fig. 10.13. basis function ψ_m^* associated to the internal node \bar{x}_m

obtained by “pasting” functions $\psi_N^{(m)}$ and $\psi_0^{(m+1)}$ together (see Fig. 10.13). In particular, we observe that function ψ_0^* is not needed, since a homogeneous Dirichlet condition is assigned in $\bar{x}_0 = -1$, whereas we need function ψ_M^* that we indicate with $\psi_N^{(M)}$. Thus, by the choice of boundary conditions made, there exist M basis functions associated to the endpoints of the sub-intervals Ω_m . (Had Dirichlet conditions been applied at both endpoints of Ω , we would have had the $(M - 1)$ functions ψ_m^* , $m = 1, \dots, M - 1$.)

Hence, altogether we have $n = (N - 1)M + M$ basis functions for the space V_N^C . Each function $u_N \in V_N^C$ can then be expressed in the following way

$$u_N(x) = \sum_{m=1}^M u_m^\Gamma \psi_m^*(x) + \sum_{m=1}^M \sum_{i=1}^{N-1} u_i^{(m)} \tilde{\psi}_i^{(m)}(x),$$

with $u_m^\Gamma = u_N(\bar{x}_m)$ and $u_i^{(m)}$ defined as in (10.72). This way, the Dirichlet boundary condition is respected.

10.6 Spectral methods on triangles and tetrahedra

As we have seen, the use of spectral methods on quadrilaterals in two dimensions (or parallelepipeds in three dimensions) is made possible through tensor products of one-dimensional functions (on the reference interval $[-1, 1]$) and of the one-dimensional Gaussian numerical integration formulae. Since a few years however, we are witnessing a growth of interest toward the use of spectral-type methods also on geometries that do not have tensor product structure, such as, for instance, triangles in 2D and tetrahedra, prisms or pyramids in 3D.

We briefly describe Dubiner’s pioneering idea [Dub91] to introduce polynomial bases of high degree on triangles, later extended in [KS05] to the three-dimensional case.

We consider the reference triangle

$$\widehat{T} = \{(x_1, x_2) \in \mathbb{R}^2 : -1 < x_1, x_2 ; x_1 + x_2 < 0\},$$

and the reference square

$$\widehat{Q} = \{(\xi_1, \xi_2) \in \mathbb{R}^2 : -1 < \xi_1, \xi_2 < 1\}.$$

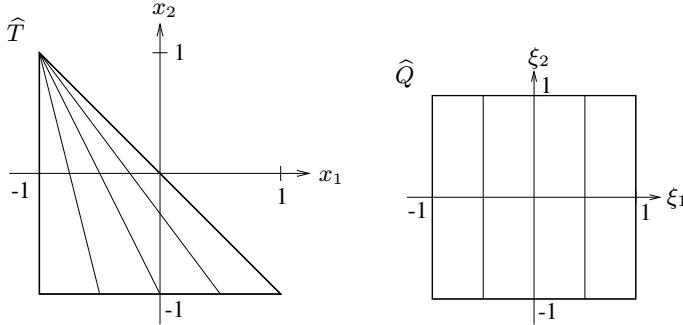


Fig. 10.14. Transformation of the reference triangle \widehat{T} on the reference square \widehat{Q} . The oblique segments are transformed into vertical segments

The transformation

$$(x_1, x_2) \rightarrow (\xi_1, \xi_2), \quad \xi_1 = 2 \frac{1+x_1}{1-x_2} - 1, \quad \xi_2 = x_2 \quad (10.73)$$

is a bijection between \widehat{T} and \widehat{Q} . Its inverse is given by

$$(\xi_1, \xi_2) \rightarrow (x_1, x_2), \quad x_1 = \frac{1}{2} (1 + \xi_1)(1 - \xi_2) - 1, \quad x_2 = \xi_2.$$

As highlighted in Fig. 10.14, the mapping $(x_1, x_2) \rightarrow (\xi_1, \xi_2)$ sends the radius in \widehat{T} issuing from the vertex $(-1, 1)$ and passing through the point $(x_1, -1)$ in the vertical segment of \widehat{Q} of equation $\xi_1 = x_1$. The latter therefore becomes singular in $(-1, 1)$. For this reason we call (ξ_1, ξ_2) the *collapsed cartesian coordinates* of the point of the triangles having coordinates (x_1, x_2) .

We denote by $\{J_k^{(\alpha, \beta)}(\xi), k \geq 0\}$ the family of Jacobi polynomials that are orthogonal with respect to the weight $w(\xi) = (1 - \xi)^\alpha (1 + \xi)^\beta$, for $\alpha, \beta \geq 0$. Hence,

$$\forall k \geq 0, \quad J_k^{(\alpha, \beta)} \in \mathbb{P}_k \quad \text{and} \quad \int_{-1}^1 J_k^{(\alpha, \beta)}(\xi) J_m^{(\alpha, \beta)}(\xi) w(\xi) d\xi = 0 \quad \forall m \neq k. \quad (10.74)$$

We observe that, for $\alpha = \beta = 0$, $J_k^{(0,0)}$ coincides with the k -th Legendre polynomial L_k . For each pair of integers $\mathbf{k} = (k_1, k_2)$ we define the so-called *warped tensor product* basis on \widehat{Q}

$$\Phi_{\mathbf{k}}(\xi_1, \xi_2) = \Psi_{k_1}(\xi_1) \Psi_{k_1, k_2}(\xi_2), \quad (10.75)$$

with $\Psi_{k_1}(\xi_1) = J_{k_1}^{(0,0)}(\xi_1)$ and $\Psi_{k_1, k_2}(\xi_2) = (1 - \xi_2)^{k_1} J_{k_2}^{(2k_1+1, 0)}(\xi_2)$. Note that $\Phi_{\mathbf{k}}$ is a polynomial of degree k_1 in ξ_1 and $k_1 + k_2$ in ξ_2 .

By now applying mapping (10.73), we find the following function defined on \widehat{T}

$$\varphi_{\mathbf{k}}(x_1, x_2) = \Phi_{\mathbf{k}}(\xi_1, \xi_2) = J_{k_1}^{(0,0)}\left(2 \frac{1+x_1}{1-x_2} - 1\right) (1 - x_2)^{k_1} J_{k_2}^{(2k_1+1, 0)}(x_2). \quad (10.76)$$

This is a polynomial of total degree $k_1 + k_2$ in the variables x_1, x_2 , i.e. $\varphi_{\mathbf{k}} \in \mathbb{P}_{k_1+k_2}(\widehat{T})$. Moreover, it can be proven that thanks to the orthogonality of the Jacobi polynomials (10.74), for each $m \neq k$,

$$\begin{aligned} \int_{\widehat{T}} \varphi_{\mathbf{k}}(x_1, x_2) \varphi_{\mathbf{m}}(x_1, x_2) dx_1 dx_2 &= \frac{1}{2} \left(\int_{-1}^1 J_{k_1}^{(0,0)}(\xi_1) J_{m_1}^{(0,0)}(\xi_1) d\xi_1 \right) \cdot \\ &\quad \left(\int_{-1}^1 J_{k_2}^{(2k_1+1,0)}(\xi_2) J_{m_2}^{(2m_1+1,0)}(\xi_2) (1-\xi_2)^{k_1+m_1+1} d\xi_2 \right) = 0. \end{aligned} \quad (10.77)$$

Hence, $\{\varphi_{\mathbf{k}} : 0 \leq k_1, k_2, k_1 + k_2 \leq N\}$ constitutes an *orthogonal (modal) basis* for the space of polynomials $\mathbb{P}_N(\widehat{T})$, with dimension $\frac{1}{2}(N+1)(N+2)$.

The orthogonality property is undoubtedly convenient as it allows to diagonalize the mass matrix (see Chap. 5). However, with the modal basis described above, the imposition of the boundary conditions (in case the computational domain is a triangle \widehat{T}), as well as satisfying the continuity conditions on the interelements in case spectral element methods with triangular elements are used, results to be uncomfortable. A possible remedy consists in *adapting* such basis, by generating a new one, which we will denote by $\{\varphi_{\mathbf{k}}^{ba}\}$; ba stands for *boundary adapted*. In order to obtain it, we will start by replacing the uni-dimensional Jacobi basis $J_k^{(\alpha,0)}(\xi)$ (with $\alpha = 0$ or $2k+1$) with the adapted basis constituted by:

- two boundary functions : $\frac{1+\xi}{2} e^{\frac{1-\xi}{2}}$;
- $(N-1)$ bubble functions : $\left(\frac{1+\xi}{2}\right)\left(\frac{1-\xi}{2}\right) J_{k-2}^{(\alpha,\beta)}(\xi)$, $k = 2, \dots, N$, for suitable fixed $\alpha, \beta \geq 1$.

These one-dimensional bases are then used as in (10.75) instead of the non-adapted Jacobi polynomials. This way, we find vertex-type, edge-type and bubble functions. Precisely:

- vertex-type functions:

$$\varPhi^{V_1}(\xi_1, \xi_2) = \left(\frac{1-\xi_1}{2}\right) \left(\frac{1-\xi_2}{2}\right) \quad (\text{vertex } V_1 = (-1, -1)),$$

$$\varPhi^{V_2}(\xi_1, \xi_2) = \left(\frac{1+\xi_1}{2}\right) \left(\frac{1-\xi_2}{2}\right) \quad (\text{vertex } V_2 = (1, -1)),$$

$$\varPhi^{V_3}(\xi_1, \xi_2) = \frac{1+\xi_2}{2} \quad (\text{vertex } V_3 = (-1, 1));$$

- edge-type functions:

$$\varPhi_{K_1}^{V_1 V_2}(\xi_1, \xi_2) = \left(\frac{1-\xi_1}{2}\right) \left(\frac{1+\xi_1}{2}\right) J_{k_1-2}^{(\beta,\beta)}(\xi_1) \left(\frac{1-\xi_2}{2}\right)^{k_1}, \quad 2 \leq k_1 \leq N,$$

$$\varPhi_{K_2}^{V_1 V_3}(\xi_1, \xi_2) = \left(\frac{1-\xi_1}{2}\right) \left(\frac{1-\xi_2}{2}\right) \left(\frac{1+\xi_2}{2}\right) J_{k_2-2}^{(\beta,\beta)}(\xi_2), \quad 2 \leq k_2 \leq N,$$

$$\varPhi_{K_2}^{V_2 V_3}(\xi_1, \xi_2) = \left(\frac{1+\xi_1}{2}\right) \left(\frac{1-\xi_2}{2}\right) \left(\frac{1+\xi_2}{2}\right) J_{k_2-2}^{(\beta,\beta)}(\xi_2), \quad 2 \leq k_2 \leq N;$$

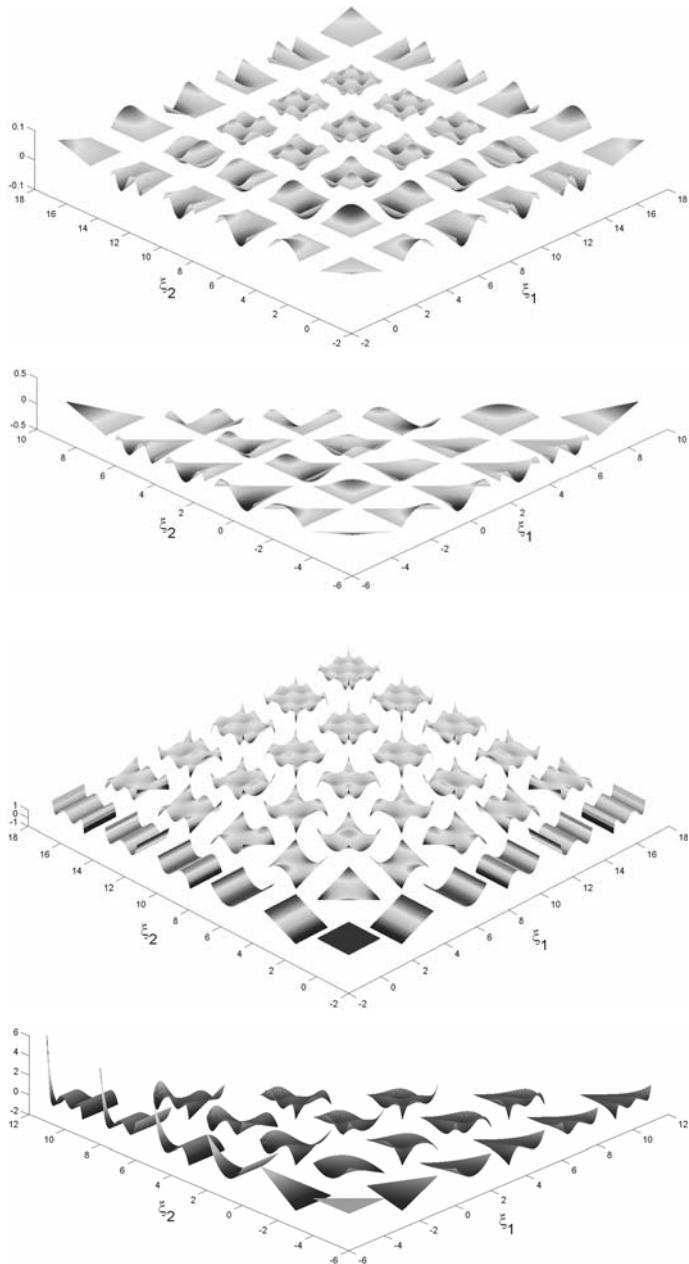


Fig. 10.15. Basis functions of degree $N = 5$: *boundary-adapted* bases on the square (first from the top) and on the triangle (second from the top) associated to the values $\beta = 1$ and $\delta = 0$; Jacobi basis $J_k^{(\alpha, \beta)}$ on the square (second from the bottom) corresponding to the values $\alpha = \beta = 0$ (Legendre case); Dubiner basis functions $\{\Phi_k\}$ on the triangle (first from the bottom)

- bubble-type functions:

$$\begin{aligned}\Phi_{k_1, k_2}^{\beta}(\xi_1, \xi_2) &= \left(\frac{1-\xi_1}{2}\right)\left(\frac{1+\xi_1}{2}\right) J_{k_1-2}^{(\beta, \beta)}(\xi_1) \cdot \\ &\quad \left(\frac{1-\xi_2}{2}\right)^{k_1} \left(\frac{1+\xi_2}{2}\right) J_{k_2-2}^{(2k_1-1+\delta, \beta)}(\xi_2),\end{aligned}$$

$$2 \leq k_1, k_2, k_1 + k_2 \leq N.$$

Although the choice $\beta = \delta = 2$ ensures the orthogonality of the bubble-functions, generally we prefer the choice $\beta = 1, \delta = 0$ as it guarantees a good sparsity of the mass and stiffness matrices and an acceptable condition number for the stiffness matrix for second-order differential operators.

In Fig. 10.15 we report some examples of bases on triangles corresponding to different choices of β and δ and different values of the degree N .

Using these modal bases, we can now set up a spectral Galerkin approximation for a boundary-value problem set on the triangle \widehat{T} , or a SEM-type method on a domain Ω partitioned in triangular elements. We refer the interested reader to [CHQZ06], [CHQZ07], [KS05].

10.7 Exercises

1. Prove inequality (10.52).
2. Prove property (10.54).
3. Write the weak formulation of problem

$$\begin{cases} -((1+x)u'(x))' + u(x) = f(x), & 0 < x < 1, \\ u(0) = \alpha, & u(1) = \beta, \end{cases}$$

and the linear system resulting from its discretization using the G-NI method.

4. Approximate the problem

$$\begin{cases} -u''(x) + u'(x) = x^2, & -1 < x < 1, \\ u(-1) = 1, & u'(1) = 0, \end{cases}$$

using the G-NI method and analyze its stability and convergence.

5. Write the G-NI approximation of the problem

$$\begin{cases} Lu(x) = -(\mu(x)u'(x))' + (b(x)u(x))' + \sigma(x)u(x) = f(x), & -1 < x < 1, \\ \mu(\pm 1)u'(\pm 1) = 0. \end{cases}$$

Prove under which conditions on the data the pseudo-spectral approximation is stable. Moreover, verify that the following relations hold:

$$L_N u_N(x_j) = f(x_j), \quad j = 1, \dots, N-1,$$

$$\mu(1) u'_N(1) = \alpha_N(f - L_N u_N)(1),$$

$$\mu(-1) u'_N(-1) = -\alpha_0(f - L_N u_N)(-1),$$

L_N being the pseudo-spectral operator defined in (10.41).

6. Consider the problem

$$\begin{cases} -\mu \Delta u + \mathbf{b} \cdot \nabla u - \sigma u = f & \text{in } \Omega = (-1, 1)^2, \\ u(\mathbf{x}) = u_0 & \text{for } x_1 = -1, \\ u(\mathbf{x}) = u_1 & \text{for } x_1 = 1, \\ \nabla u(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = 0 & \text{for } x_2 = -1 \text{ and } x_2 = 1, \end{cases}$$

where $\mathbf{x} = (x_1, x_2)^T$, \mathbf{n} is the outgoing normal from Ω , $\mu = \mu(\mathbf{x})$, $\mathbf{b} = \mathbf{b}(\mathbf{x})$, $\sigma = \sigma(\mathbf{x})$, $f = f(\mathbf{x})$ are assigned functions, and u_0 and u_1 are given constants. Provide sufficient conditions on the data to guarantee the existence and uniqueness of the weak solution, and give an a priori estimate. Approximate then the weak problem using the G-NI method, providing an analysis of its stability and convergence.

7. Prove the stability condition (10.42) in the case of the pseudo-spectral approximation of the equation (5.4) (replacing the interval $(0, 1)$ with $(-1, 1)$).

[*Solution:* follow a similar procedure to that explained in Sec. 5.4 for the finite element solution and invoke the properties reported in Lemmas 10.2 and 10.3.]

8. Consider the parabolic heat equation

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0, & -1 < x < 1, t > 0, \\ u(x, 0) = u_0(x), & -1 < x < 1, \\ u(-1, t) = u(1, t) = 0, & t > 0. \end{cases}$$

Approximate it using the G-NI method in space and the implicit Euler method in time and conduct its stability analysis.

11

Diffusion-transport-reaction equations

In this chapter, we consider problems of the following form

$$\begin{cases} -\operatorname{div}(\mu \nabla u) + \mathbf{b} \cdot \nabla u + \sigma u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \quad (11.1)$$

where μ, σ, f and \mathbf{b} are given functions (or constants). In the most general case, we will suppose that $\mu \in L^\infty(\Omega)$ with $\mu(\mathbf{x}) \geq \mu_0 > 0$, $\sigma \in L^2(\Omega)$ with $\sigma(\mathbf{x}) \geq 0$ a.e. in Ω , $\mathbf{b} \in [L^\infty(\Omega)]^2$ with $\operatorname{div}(\mathbf{b}) \in L^2(\Omega)$, and $f \in L^2(\Omega)$.

In many practical applications, the *diffusion* term $-\operatorname{div}(\mu \nabla u)$ is dominated by the *convection* term $\mathbf{b} \cdot \nabla u$ (also called *transport*) or by the *reaction* term σu (also called the *absorption* term when σ is non-negative). In such cases, as we will see, the solution can give rise to *boundary layers*, that is regions, generally close to the boundary of Ω , where the solution is characterized by strong gradients.

To derive such models, and to capture the analogy with random walk processes, see e.g. [Sal08], Chap. 2.

In this chapter we propose to analyze the conditions ensuring the existence and uniqueness of the solution to problem (11.1). We also consider the Galerkin method, illustrate its difficulties in providing stable solutions in the presence of boundary layers, and finally propose alternative discretization methods for the approximation of (11.1).

11.1 Weak problem formulation

Let $V = H_0^1(\Omega)$. Introducing the bilinear form $a : V \times V \mapsto \mathbb{R}$,

$$a(u, v) = \int_{\Omega} \mu \nabla u \cdot \nabla v \, d\Omega + \int_{\Omega} v \mathbf{b} \cdot \nabla u \, d\Omega + \int_{\Omega} \sigma u v \, d\Omega \quad \forall u, v \in V, \quad (11.2)$$

the weak formulation of problem (11.1) becomes

$$\text{find } u \in V : \quad a(u, v) = (f, v) \quad \forall v \in V. \quad (11.3)$$

In order to prove the existence and uniqueness of the solution of (11.3) we want to be in the conditions to apply the Lax-Milgram lemma.

To verify the coercivity of the bilinear form $a(\cdot, \cdot)$, we proceed separately on the single terms composing (11.2).

For the first term we have

$$\int_{\Omega} \mu \nabla v \cdot \nabla v \, d\Omega \geq \mu_0 \|\nabla v\|_{L^2(\Omega)}^2. \quad (11.4)$$

As $v \in H_0^1(\Omega)$, the Poincaré inequality holds (see (2.13))

$$\|v\|_{L^2(\Omega)} \leq C_{\Omega} \|\nabla v\|_{L^2(\Omega)}, \quad (11.5)$$

for a suitable positive constant C_{Ω} independent of v . Thus

$$\|v\|_{H^1(\Omega)}^2 = \|v\|_{L^2(\Omega)}^2 + \|\nabla v\|_{L^2(\Omega)}^2 \leq (1 + C_{\Omega}^2) \|\nabla v\|_{L^2(\Omega)}^2,$$

and therefore it follows from (11.4) that

$$\int_{\Omega} \mu \nabla v \cdot \nabla v \, d\Omega \geq \frac{\mu_0}{1 + C_{\Omega}^2} \|v\|_{H^1(\Omega)}^2.$$

We now move to the convective term. Using the Green formula (3.16) yields

$$\begin{aligned} \int_{\Omega} v \mathbf{b} \cdot \nabla v \, d\Omega &= \frac{1}{2} \int_{\Omega} \mathbf{b} \cdot \nabla(v^2) \, d\Omega = -\frac{1}{2} \int_{\Omega} v^2 \operatorname{div}(\mathbf{b}) \, d\Omega + \frac{1}{2} \int_{\partial\Omega} \mathbf{b} \cdot \mathbf{n} v^2 \, d\gamma \\ &= -\frac{1}{2} \int_{\Omega} v^2 \operatorname{div}(\mathbf{b}) \, d\Omega, \end{aligned}$$

as $v = 0$ on $\partial\Omega$, whence

$$\int_{\Omega} v \mathbf{b} \cdot \nabla v \, d\Omega + \int_{\Omega} \sigma v^2 \, d\Omega = \int_{\Omega} v^2 \left(-\frac{1}{2} \operatorname{div}(\mathbf{b}) + \sigma \right) \, d\Omega,$$

The last integral is certainly positive if we suppose that

$$-\frac{1}{2} \operatorname{div}(\mathbf{b}) + \sigma \geq 0 \quad \text{a.e. in } \Omega. \quad (11.6)$$

Consequently, the bilinear form $a(\cdot, \cdot)$ is coercive, as

$$a(v, v) \geq \alpha \|v\|_{H^1(\Omega)}^2 \quad \forall v \in V, \quad \text{with } \alpha = \frac{\mu_0}{1 + C_{\Omega}^2}. \quad (11.7)$$

Let us now prove that the bilinear form $a(\cdot, \cdot)$ is continuous, that is there exists a positive constant M such that

$$|a(u, v)| \leq M \|u\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)} \quad \forall u, v \in V. \quad (11.8)$$

The first term at the right-hand side of (11.2) can be bounded as follows

$$\begin{aligned} \left| \int_{\Omega} \mu \nabla u \cdot \nabla v \, d\Omega \right| &\leq \|\mu\|_{L^{\infty}(\Omega)} \|\nabla u\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)} \\ &\leq \|\mu\|_{L^{\infty}(\Omega)} \|u\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)}, \end{aligned} \quad (11.9)$$

having used the Hölder and Cauchy-Schwarz inequalities (see Sec. 2.5), as well as the inequality $\|\nabla w\|_{L^2(\Omega)} \leq \|w\|_{H^1(\Omega)} \forall w \in H^1(\Omega)$. For the right-hand side, proceeding in a similar way we find

$$\begin{aligned} \left| \int_{\Omega} v \mathbf{b} \cdot \nabla u \, d\Omega \right| &\leq \|\mathbf{b}\|_{L^{\infty}(\Omega)} \|v\|_{L^2(\Omega)} \|\nabla u\|_{L^2(\Omega)} \\ &\leq \|\mathbf{b}\|_{L^{\infty}(\Omega)} \|v\|_{H^1(\Omega)} \|u\|_{H^1(\Omega)}. \end{aligned} \quad (11.10)$$

Finally, for the third term we have, thanks to the Cauchy-Schwarz inequality,

$$\left| \int_{\Omega} \sigma u v \, d\Omega \right| \leq \|\sigma\|_{L^2(\Omega)} \|uv\|_{L^2(\Omega)} \leq \|\sigma\|_{L^2(\Omega)} \|u\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)}. \quad (11.11)$$

Indeed, $\|uv\|_{L^2(\Omega)} \leq \|u\|_{L^4(\Omega)} \|v\|_{L^4(\Omega)} \leq \|u\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)}$, having applied inequality (2.17) and exploited the inclusions (2.18).

Summing term-by-term (11.9), (11.10) and (11.11), the property (11.8) follows by taking, e.g.,

$$M = \|\mu\|_{L^{\infty}(\Omega)} + \|\mathbf{b}\|_{L^{\infty}(\Omega)} + \|\sigma\|_{L^2(\Omega)}. \quad (11.12)$$

On the other hand, the right-hand side of (11.3) defines a bounded and linear functional thanks to the Cauchy-Schwarz inequality and to (11.5).

As the Lax-Milgram Lemma hypotheses are verified (see Lemma 3.1), it follows that the solution of the weak problem (11.3) exists and is unique. Moreover, the following a priori estimates hold

$$\|u\|_{H^1(\Omega)} \leq \frac{1}{\alpha} \|f\|_{L^2(\Omega)}, \quad \|\nabla u\|_{L^2(\Omega)} \leq \frac{C_{\Omega}}{\mu_0} \|f\|_{L^2(\Omega)},$$

as consequences of (11.4), (11.7) and (11.5). The first is an immediate consequence of the Corollary 3.1, the second one can easily be proven starting from equation $a(u, u) = (f, u)$ and using the Cauchy-Schwarz and Poincaré inequalities as well as (11.4) and (11.6).

The Galerkin approximation of problem (11.3) is

$$\text{find } u_h \in V_h : \quad a(u_h, v_h) = (f, v_h) \quad \forall v_h \in V_h,$$

where $\{V_h, h > 0\}$ is a suitable family of subspaces of $H_0^1(\Omega)$. By replicating the proof carried out above for the exact problem (11.3), the following estimates can be

proven:

$$\|u_h\|_{H^1(\Omega)} \leq \frac{1}{\alpha} \|f\|_{L^2(\Omega)}, \quad \|\nabla u_h\|_{L^2(\Omega)} \leq \frac{C_\Omega}{\mu_0} \|f\|_{L^2(\Omega)}.$$

These prove, in particular, that the gradient of the discrete solution (as well as that of the weak solution u) could be as large as μ_0 is small.

Moreover, the Galerkin error inequality (4.10) gives

$$\|u - u_h\|_V \leq \frac{M}{\alpha} \inf_{v_h \in V_h} \|u - v_h\|_V. \quad (11.13)$$

By the definitions of α and M (see (11.7) and (11.12)), the upbounding constant M/α becomes as high (and, correspondingly, the estimate (11.13) meaningless) as the ratio $\|\mathbf{b}\|_{L^\infty(\Omega)}/\|\mu\|_{L^\infty(\Omega)}$ (or the ratio $\|\sigma\|_{L^\infty(\Omega)}/\|\mu\|_{L^\infty(\Omega)}$) grows, that is when the convective (or reactive) term dominates over the diffusive one.

In such cases the Galerkin method can yield to inaccurate solutions, unless – as we will see – an extremely small discretization step h is used.

In order to evaluate more precisely the behavior of the numerical solution provided by the Galerkin method, we analyze a one-dimensional problem.

11.2 Analysis of a one-dimensional diffusion-transport problem

Let us consider the following one-dimensional diffusion-transport problem

$$\begin{cases} -\mu u'' + bu' = 0, & 0 < x < 1, \\ u(0) = 0, & u(1) = 1, \end{cases} \quad (11.14)$$

μ and b being two positive constants.

Its weak formulation is

$$\text{find } u \in H^1(0, 1) : \quad a(u, v) = 0 \quad \forall v \in H_0^1(0, 1), \quad (11.15)$$

with $u(0) = 0$, $u(1) = 1$, and $a(u, v) = \int_0^1 (\mu u' v' + bu' v) dx$. Following what indicated in Sec. 3.2.2, we can reformulate (11.15) by introducing a suitable lifting (or extension) of the boundary data. In this particular case, we can choose $R_g = x$. Having then set $\hat{u} = u - R_g = u - x$, we can reformulate (11.15) in the following way

$$\text{find } \hat{u} \in H_0^1(0, 1) : \quad a(\hat{u}, v) = F(v) \quad \forall v \in H_0^1(0, 1), \quad (11.16)$$

being $F(v) = -a(x, v) = -\int_0^1 bv dx$ the contribution due to the data lifting.

We define the *global Péclet number* as the ratio

$$\text{Pe}_g = \frac{|b|L}{2\mu},$$

with L being the linear dimension of the domain (1 in our case). This ratio provides a measure of how the convective term dominates the diffusive one, hence it has the same role as the Reynolds number in the Navier-Stokes equations, which we will see in Chap. 15.

We start by computing the exact solution of such problem. Its associated characteristic equation is

$$-\mu\lambda^2 + b\lambda = 0,$$

and it has two roots, $\lambda_1 = 0$ and $\lambda_2 = b/\mu$. The general solution is therefore

$$u(x) = C_1 e^{\lambda_1 x} + C_2 e^{\lambda_2 x} = C_1 + C_2 e^{\frac{b}{\mu} x}.$$

By imposing the boundary conditions, we find the constants C_1 and C_2 , and therefore the solution

$$u(x) = \frac{\exp(\frac{b}{\mu}x) - 1}{\exp(\frac{b}{\mu}) - 1}.$$

Using the Taylor expansion for the exponentials, if $b/\mu \ll 1$ we obtain

$$u(x) = \frac{1 + \frac{b}{\mu}x + \dots - 1}{1 + \frac{b}{\mu} + \dots - 1} \simeq \frac{\frac{b}{\mu}x}{\frac{b}{\mu}} = x.$$

Thus, the solution is near the straight line interpolating the boundary data (which is the solution corresponding to the case $b = 0$).

Conversely, if $b/\mu \gg 1$, the exponentials are very large, hence

$$u(x) \simeq \frac{\exp(\frac{b}{\mu}x)}{\exp(\frac{b}{\mu})} = \exp\left(-\frac{b}{\mu}(1-x)\right),$$

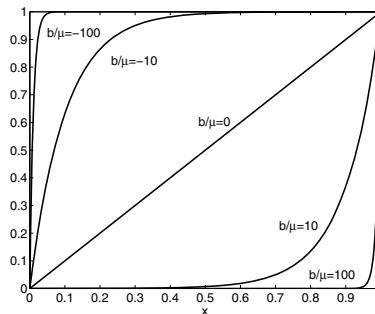


Fig. 11.1. Behavior of the solution of problem (11.14) when varying the ratio b/μ . For completeness, we also highlight the solutions relating to the case where b is negative

thus, the solution is close to zero in almost all of the interval, except for a neighborhood of the point $x = 1$, where it tends to 1 exponentially. Such neighborhood has a width of the order of μ/b and is therefore very small: the solution exhibits a *boundary layer* of width $\mathcal{O}(\frac{\mu}{b})$ in proximity of $x = 1$ (see Fig. 11.1), where the derivative behaves like b/μ , and is therefore unbounded if $\mu \rightarrow 0$.

Let us now suppose to use the Galerkin - linear finite elements method to approximate (11.15)

$$\text{find } u_h \in X_h^1 : \begin{cases} a(u_h, v_h) = 0 & \forall v_h \in \overset{\circ}{X}_h^1, \\ u_h(0) = 0, u_h(1) = 1, \end{cases} \quad (11.17)$$

where, denoting by x_i , for $i = 0, \dots, M$, the vertices of the partition introduced on $(0, 1)$, we have set, coherently with (4.14),

$$X_h^r = \{v_h \in C^0([0, 1]) : v_h \Big|_{[x_{i-1}, x_i]} \in \mathbb{P}_r, \quad i = 1, \dots, M\},$$

$$\overset{\circ}{X}_h^r = \{v_h \in X_h^r : v_h(0) = v_h(1) = 0\},$$

for $r \geq 1$. Having chosen, for each $i = 1, \dots, M-1$, $v_h = \varphi_i$ (the i -th basis function of X_h^1), we have

$$\int_0^1 \mu u'_h \varphi'_i \, dx + \int_0^1 b u'_h \varphi_i \, dx = 0,$$

that is, the support of φ_i being equal to $[x_{i-1}, x_{i+1}]$ and writing $u_h = \sum_{j=1}^{M-1} u_j \varphi_j(x)$,

$$\begin{aligned} & \mu \left[u_{i-1} \int_{x_{i-1}}^{x_i} \varphi'_{i-1} \varphi'_i \, dx + u_i \int_{x_{i-1}}^{x_{i+1}} (\varphi'_i)^2 \, dx + u_{i+1} \int_{x_i}^{x_{i+1}} \varphi'_{i+1} \varphi'_i \, dx \right] \\ & + b \left[u_{i-1} \int_{x_{i-1}}^{x_i} \varphi'_{i-1} \varphi_i \, dx + u_i \int_{x_{i-1}}^{x_{i+1}} \varphi'_i \varphi_i \, dx + u_{i+1} \int_{x_i}^{x_{i+1}} \varphi'_{i+1} \varphi_i \, dx \right] = 0, \end{aligned}$$

$\forall i = 1, \dots, M-1$. If the partition is uniform, that is $x_i = x_{i-1} + h$, with $i = 1, \dots, M$, observing that $\varphi'_i(x) = \frac{1}{h}$ if $x_{i-1} < x < x_i$, $\varphi'_i(x) = -\frac{1}{h}$ if $x_i < x < x_{i+1}$, for $i = 1, \dots, M-1$, we obtain

$$\mu \left(-u_{i-1} \frac{1}{h} + u_i \frac{2}{h} - u_{i+1} \frac{1}{h} \right) + b \left(-u_{i-1} \frac{1}{h} \frac{h}{2} + u_{i+1} \frac{1}{h} \frac{h}{2} \right) = 0,$$

that is

$$\frac{\mu}{h} (-u_{i-1} + 2u_i - u_{i+1}) + \frac{1}{2} b (u_{i+1} - u_{i-1}) = 0, \quad i = 1, \dots, M-1. \quad (11.18)$$

Reordering the terms, we find

$$\left(\frac{b}{2} - \frac{\mu}{h}\right)u_{i+1} + \frac{2\mu}{h}u_i - \left(\frac{b}{2} + \frac{\mu}{h}\right)u_{i-1} = 0, \quad i = 1, \dots, M-1.$$

Dividing by μ/h and defining the *local* (or “grid”) Péclet number

$$\text{Pe} = \frac{|b|h}{2\mu}, \quad (11.19)$$

we finally have

$$(\text{Pe} - 1)u_{i+1} + 2u_i - (\text{Pe} + 1)u_{i-1} = 0, \quad i = 1, \dots, M-1. \quad (11.20)$$

This is a linear difference equation that admits exponential solutions of the form $u_i = \rho^i$ (see [QSS07]). Replacing such expression into (11.20), we obtain

$$(\text{Pe} - 1)\rho^2 + 2\rho - (\text{Pe} + 1) = 0, \quad i = 1, \dots, M-1,$$

from which we get the two roots

$$\rho_{1,2} = \frac{-1 \pm \sqrt{1 + \text{Pe}^2 - 1}}{\text{Pe} - 1} = \begin{cases} (1 + \text{Pe})/(1 - \text{Pe}), \\ 1. \end{cases}$$

Thanks to the linearity of (11.20), the general solution of such equation takes the form

$$u_i = A_1 \rho_1^i + A_2 \rho_2^i,$$

with A_1 and A_2 two arbitrary constants. By imposing the boundary conditions $u_0 = 0$ and $u_M = 1$, we find

$$A_2 = -A_1 \text{ and } A_1 = \left(1 - \left(\frac{1 + \text{Pe}}{1 - \text{Pe}}\right)^M\right)^{-1}.$$

To conclude, the solution of problem (11.17) has the following nodal values

$$u_i = \frac{1 - \left(\frac{1 + \text{Pe}}{1 - \text{Pe}}\right)^i}{1 - \left(\frac{1 + \text{Pe}}{1 - \text{Pe}}\right)^M}, \quad i = 0, \dots, M.$$

We observe that, if $\text{Pe} > 1$, the exponential at the numerator carries a negative basis power, therefore the approximate solution becomes oscillatory, as opposed to the exact solution that is monotone! This phenomenon is displayed in Fig. 11.2 where the solution of (11.20), for different values of the local Péclet number, is compared to the exact solution for a case where the global Péclet number is equal to 50. As it can be observed, the higher the Péclet number gets, the more the behavior of the approximate

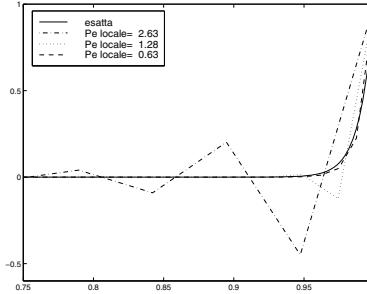


Fig. 11.2. Finite element solution of the diffusion-transport problem (11.14) with $\text{Pe}_g = 50$ for different values of the local Péclet number

solution differs from that of the exact solution, denoting oscillations that become more and more noticeable in proximity of the boundary layer.

The most obvious remedy to this misbehavior would be to choose a sufficiently small grid-size h , in order to ensure $\text{Pe} < 1$. However, this strategy is not always convenient: for instance, if $b = 1$ and $\mu = 1/5000$, we should take $h < 10^{-4}$, that is introduce at least 10000 intervals on the $(0, 1)$ interval! In particular, such strategy would require an unreasonably high number of nodal points for boundary-value problems in several dimensions. A more suitable remedy consists in using an a-priori adaptive procedure that refines the grid only in proximity of the boundary layer. Several strategies can be followed to this purpose. Amongst the most well-known, we mention the so-called type B (for Bakhvâlov) or type S (for Shishkin) grids. See e.g. [GRS07].

Alternative grid adaptive strategies, both a-priori and a-posteriori, especially useful for multidimensional problems, are those described in Sec. 4.6.

11.3 Analysis of a one-dimensional diffusion-reaction problem

Let us now consider a one-dimensional diffusion-reaction problem

$$\begin{cases} -\mu u'' + \sigma u = 0, & 0 < x < 1, \\ u(0) = 0, & u(1) = 1, \end{cases} \quad (11.21)$$

with μ and σ positive constants, whose solution is

$$u(x) = \frac{\sinh(\alpha x)}{\sinh(\alpha)} = \frac{e^{\alpha x} - e^{-\alpha x}}{e^\alpha - e^{-\alpha}}, \text{ with } \alpha = \sqrt{\sigma/\mu}.$$

Also in this case, if $\sigma/\mu \gg 1$, there is a boundary layer for $x \rightarrow 1$, with thickness of order $\sqrt{\mu/\sigma}$, where the first derivative becomes unbounded for $\mu \rightarrow 0$ (note, for instance, the exact solution for the case displayed in Fig. 11.3). Also in this case, it is interesting to define the global Péclet number, which takes the form

$$\text{Pe}_g = \frac{\sigma L^2}{6\mu},$$

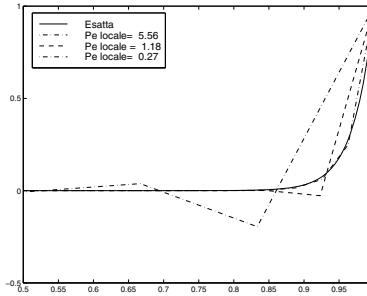


Fig. 11.3. Comparison between the numerical solution and the exact solution of the diffusion-reaction problem (11.21) with $\mathbb{P}e_g = 200$. The numerical solution has been obtained using the Galerkin-linear finite elements method on uniform grids

L still being the linear dimension of the domain (1 in our case).

The Galerkin finite element approximation of (11.21) reads

$$\text{find } u_h \in X_h^r \text{ s.t. } a(u_h, v_h) = 0 \quad \forall v_h \in \overset{\circ}{X}_h^r, \quad (11.22)$$

for $r \geq 1$, with $u_h(0) = 0$ and $u_h(1) = 0$ and $a(u_h, v_h) = \int_0^1 (\mu u'_h v'_h + \sigma u_h v_h) dx$.

Equivalently, by setting $\overset{\circ}{u}_h = u_h - x$, and $F(v_h) = -a(x, v_h) = -\int_0^1 x v_h dx$, we have

$$\text{find } \overset{\circ}{u}_h \in V_h \text{ s.t. } a(\overset{\circ}{u}_h, v_h) = F(v_h) \quad \forall v_h \in V_h, \quad (11.23)$$

with $V_h = \overset{\circ}{X}_h^r$. For the sake of simplicity, let us consider problem (11.22) with piecewise linear elements (that is $r = 1$) on a uniform partition. The equation associated to the generic basis function $v_h = \varphi_i$, $i = 1, \dots, M - 1$, is

$$\int_0^1 \mu u'_h \varphi'_i dx + \int_0^1 \sigma u_h \varphi_i dx = 0.$$

By carrying out our computation in a similar way to what we did in the previous section, and observing that

$$\int_{x_{i-1}}^{x_i} \varphi_{i-1} \varphi_i dx = \frac{h}{6}, \quad \int_{x_{i-1}}^{x_{i+1}} \varphi_i^2 dx = \frac{2}{3}h, \quad \int_{x_i}^{x_{i+1}} \varphi_i \varphi_{i+1} dx = \frac{h}{6},$$

we obtain

$$\mu \left(-u_{i-1} \frac{1}{h} + u_i \frac{2}{h} - u_{i+1} \frac{1}{h} \right) + \sigma \left(u_{i-1} \frac{h}{6} + u_i \frac{2}{3}h + u_{i+1} \frac{h}{6} \right) = 0, \quad (11.24)$$

that is

$$\left(\frac{h}{6} \sigma - \frac{\mu}{h} \right) u_{i+1} + \left(\frac{2}{3}h + \sigma \frac{2\mu}{h} \right) u_i + \left(\frac{h}{6} \sigma - \frac{\mu}{h} \right) u_{i-1} = 0.$$

Dividing by μ/h and defining the following local Péclet number

$$\text{Pe} = \frac{\sigma h^2}{6\mu}, \quad (11.25)$$

we finally have

$$(\text{Pe} - 1)u_{i+1} + 2(1 + 2\text{Pe})u_i + (\text{Pe} - 1)u_{i-1} = 0, \quad i = 1, \dots, M - 1.$$

This three-term difference equation admits the following solutions for each $i = 0, \dots, M$,

$$u_i = \frac{\left[\frac{1 + 2\text{Pe} + \sqrt{3\text{Pe}(\text{Pe} + 2)}}{1 - \text{Pe}} \right]^i - \left[\frac{1 + 2\text{Pe} - \sqrt{3\text{Pe}(\text{Pe} + 2)}}{1 - \text{Pe}} \right]^i}{\left[\frac{1 + 2\text{Pe} + \sqrt{3\text{Pe}(\text{Pe} + 2)}}{1 - \text{Pe}} \right]^M - \left[\frac{1 + 2\text{Pe} - \sqrt{3\text{Pe}(\text{Pe} + 2)}}{1 - \text{Pe}} \right]^M},$$

which once more results to be oscillatory when $\text{Pe} > 1$.

The problem is thus critical when $\frac{\sigma}{\mu} \gg 1$, that is when the diffusion coefficient is very small with respect to the reaction one (see the example reported in Fig. 11.3).

11.4 Finite elements and finite differences (FD)

We want to analyze the behavior of the finite difference method (FD, in short) applied to the solution of diffusion-transport and diffusion-reaction problems, highlighting analogies and differences with respect to the finite element method (FE, in short). We will limit ourselves to the *one-dimensional* case and we will consider a *uniform mesh*.

Let us consider once more problem (11.14) and let us approximate it via finite differences. In order to generate a local discretization error of the same entity for both terms, we will approximate the derivatives by using the following centered incremental ratios:

$$u'(x_i) = \frac{u(x_{i+1}) - u(x_{i-1})}{2h} + \mathcal{O}(h^2), \quad i = 1, \dots, M - 1, \quad (11.26)$$

$$u''(x_i) = \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} + \mathcal{O}(h^2), \quad i = 1, \dots, M - 1. \quad (11.27)$$

In both cases, as highlighted, the remainder is an infinitesimal with respect to the step size h , as it can be easily proven by invoking the truncated Taylor series (see, e.g., [QSS07]). By replacing in (11.14) the exact derivatives with these incremental ratios (thus ignoring the infinitesimal error), we find the following scheme

$$\begin{cases} -\mu \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + b \frac{u_{i+1} - u_{i-1}}{2h} = 0, & i = 1, \dots, M - 1, \\ u_0 = 0, \quad u_M = 1. \end{cases} \quad (11.28)$$

For each i , the unknown u_i provides an approximation for the nodal value $u(x_i)$. Multiplying by h , we obtain the same equation (11.18) obtained using linear finite elements on the same uniform grid.

Let us now consider the diffusion-reaction problem (11.21). Proceeding in an analogous way, its approximation using finite differences yields

$$\begin{cases} -\mu \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + \sigma u_i = 0, & i = 1, \dots, M-1, \\ u_0 = 0, \quad u_M = 1. \end{cases} \quad (11.29)$$

The above equation is different from (11.24), that has been obtained using linear finite elements, as the reaction term, appearing in (11.29) with the diagonal contribution σu_i , yields in (11.24) the sum of three different contributions

$$\sigma \left(u_{i-1} \frac{h}{6} + u_i \frac{2}{3}h + u_{i+1} \frac{h}{6} \right).$$

Hence, the two methods FE and FD are *not* equivalent in this case. We observe that the solution obtained using the FD scheme (11.29) does not display oscillations, whichever value is chosen for the discretization step h . As a matter of fact, the solution of (11.29) is

$$u_i = (\rho_1^M - \rho_2^M)^{-1}(\rho_1^i - \rho_2^i),$$

with

$$\rho_{1,2} = \frac{\gamma}{2} \pm \left(\frac{\gamma^2}{4} - 1 \right)^{\frac{1}{2}} \text{ and } \gamma = 2 + \frac{\sigma h^2}{\mu}.$$

The i -th powers now have a positive basis, guaranteeing a monotone behavior of the sequence $\{u_i\}$. This differs from what we have seen in Sec. 11.3 for the FE, for which it is necessary to choose $h \leq \sqrt{\frac{6\mu}{\sigma}}$ so to guarantee that the local Péclet number (11.25) is less than 1. See the example reported in Fig. 11.4 for a comparison between a finite element approximation and a finite difference approximation.

11.5 The mass-lumping technique

In the case of the reaction-diffusion problem, we can obtain the same result as with finite differences by using linear finite elements, provided that we resort to the so-called *mass-lumping* technique, thanks to which the *mass matrix*

$$M = (m_{ij}), \quad m_{ij} = \int_0^1 \varphi_j \varphi_i \, dx,$$

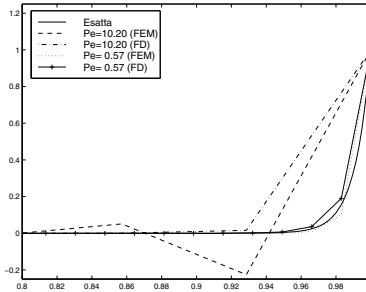


Fig. 11.4. Comparison between the numerical solutions of the one-dimensional diffusion-transport equation (11.21) with $\text{Pe}_g = 2000$ obtained using the Galerkin-linear finite element method (FEM) and the finite difference method (DF), for different values of the local Péclet number

which is tridiagonal, is approximated using a diagonal matrix M_L , called *condensed* or *lumped matrix*. To this end, we use the following trapezoidal quadrature formula on each interval (x_i, x_{i+1}) , for each $i = 0, \dots, M - 1$

$$\int_{x_i}^{x_{i+1}} f(x) dx \simeq \frac{h}{2} (f(x_i) + f(x_{i+1})).$$

Thanks to the properties of the finite element basis functions, we then find:

$$\begin{aligned} \int_{x_{i-1}}^{x_i} \varphi_{i-1} \varphi_i dx &\simeq \frac{h}{2} [\varphi_{i-1}(x_{i-1}) \varphi_i(x_{i-1}) + \varphi_{i-1}(x_i) \varphi_i(x_i)] = 0, \\ \int_{x_{i-1}}^{x_{i+1}} \varphi_i^2 dx &= 2 \int_{x_{i-1}}^{x_i} \varphi_i^2 dx \simeq 2 \frac{h}{2} [\varphi_i^2(x_{i-1}) + \varphi_i^2(x_i)] = h, \\ \int_{x_i}^{x_{i+1}} \varphi_i \varphi_{i+1} dx &\simeq \frac{h}{2} [\varphi_i(x_i) \varphi_{i+1}(x_i) + \varphi_i(x_{i+1}) \varphi_{i+1}(x_{i+1})] = 0. \end{aligned}$$

Using the previous formulae to approximate the mass matrix coefficients, we get to the following diagonal matrix M_L whose elements are the sums of the elements of each row of matrix M , i.e.

$$M_L = \text{diag}(\tilde{m}_{ii}), \quad \text{with } \tilde{m}_{ii} = \sum_{j=i-1}^{i+1} m_{ij}. \quad (11.30)$$

Note that, thanks to the following *partition of unity* property of the basis functions

$$\sum_{j=0}^M \varphi_j(x) = 1 \quad \forall x \in [0, 1], \quad (11.31)$$

the elements of matrix M_L take the following expression on the interval $[0, 1]$

$$\tilde{m}_{ii} = \int_0^1 \varphi_i \, dx, \quad i = 0, \dots, M.$$

Their values are reported in Exercise 3 for finite elements of degree 1, 2, 3.

If the terms of order zero are replaced in the following way

$$\int_0^1 \sigma u_h \varphi_i \, dx = \sigma \sum_{j=1}^{M-1} u_j \int_0^1 \varphi_j \varphi_i \, dx = \sigma \sum_{j=1}^{M-1} m_{ij} u_j \simeq \sigma \tilde{m}_{ii} u_i,$$

the finite element problem produces solutions coinciding with those of finite differences, hence they are monotone for each value of h . Moreover, replacing M with M_L does not reduce the order of accuracy of the method.

The process of mass-lumping (11.30) can be generalized to the two-dimensional case when linear elements are used. For quadratic finite elements, instead, the above-mentioned procedure consisting in summing by rows would generate a singular mass matrix M_L (see Example 11.1). An alternative diagonalization strategy consists in using the matrix $\hat{M} = \text{diag}(\hat{m}_{ii})$ with elements

$$\hat{m}_{ii} = \frac{m_{ii}}{\sum_j m_{jj}}.$$

In the one-dimensional case, for linear and quadratic finite elements, the matrices \hat{M} and M_L coincide, while they differ for cubic elements (see Exercise 3). The matrix \hat{M} is non-singular also for Lagrangian finite elements of high order, while it can turn out to be singular when using non-lagrangian finite elements, for instance when using hierarchical bases. In the latter case, we resort to more sophisticated mass-lumping procedures. Indeed, a number of diagonalization techniques able to generate non-singular matrices have been elaborated also for finite elements of high degree. See for example [CJRT01].

Example 11.1 The mass matrix for the \mathbb{P}_2 finite elements on the reference triangle with vertices $(0, 0)$, $(1, 0)$ and $(0, 1)$, is given by

$$M = \frac{1}{180} \begin{bmatrix} 6 & -1 & -1 & 0 & -4 & 0 \\ -1 & 6 & -1 & 0 & 0 & -4 \\ -1 & -1 & 6 & -4 & 0 & 0 \\ 0 & 0 & -4 & 32 & 16 & 16 \\ -4 & 0 & 0 & 16 & 32 & 16 \\ 0 & -4 & 0 & 16 & 16 & 32 \end{bmatrix},$$

while the lumped mass matrices are given by

$$M_L = \frac{1}{180} \text{diag}(0 0 0 60 60 60),$$

$$\hat{M} = \frac{1}{114} \text{diag}(6 6 6 32 32 32).$$

As it can be noticed the matrix M_L is singular. ■

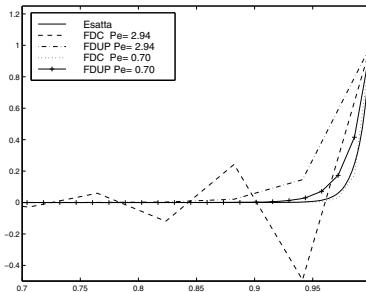


Fig. 11.5. Solution obtained using the centered (CFD) and upwind (UPFD) finite difference scheme for the one-dimensional diffusion-transport equation (11.14) with $\text{Pe}_g = 50$. Also in the presence of high local Péclet numbers, one can notice the stabilizing effect of the artificial diffusion introduced by the upwind scheme, inevitably accompanied by a loss of accuracy

The mass-lumping technique is also used in other contexts, for instance in the solution of parabolic problems (see Chap. 5) when finite-element spatial discretizations and finite-difference explicit temporal discretizations (e.g., the forward-Euler method) are used. In such case, lumping the mass matrix deriving from the discretization of the temporal derivative can conduct to the solution of a diagonal system, with corresponding reduction of the computational cost.

11.6 Decentered FD schemes and artificial diffusion

The comparative analysis with finite differences has allowed us to find a remedy to the oscillatory behavior of finite element solutions in the case of a diffusion-reaction problem. We now wish to also find a remedy for the case of the diffusion-transport problem (11.14).

Let us consider finite differences. The oscillations in the numerical solution arise from the fact that we use a centered finite difference (CFD) scheme for the discretization of the transport term. Since the latter is non-symmetric, this suggests to discretize the first derivative at a point x_i with a decentered incremental ratio where the value at x_{i-1} intervenes if the field is positive, and at x_{i+1} in the opposite case.

This technique is called *upwinding* and the resulting scheme, called *upwind scheme* (FDUP, in short) in the case $b > 0$ is written as

$$-\mu \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + b \frac{u_i - u_{i-1}}{h} = 0, \quad i = 1, \dots, M-1. \quad (11.32)$$

(See Fig. 11.5 for an example of application of the upwind scheme). The price to pay is a reduction of the order of convergence as the decentered incremental ratio introduces a local discretization error which is $\mathcal{O}(h)$ (see (11.27)) as opposed to $\mathcal{O}(h^2)$ in the CFD case.

We now observe that

$$\frac{u_i - u_{i-1}}{h} = \frac{u_{i+1} - u_{i-1}}{2h} - \frac{h}{2} \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2},$$

that is the decentered incremental ratio to approximate the first derivative can be written as the sum of a centered incremental ratio plus a term proportional to the discretization of the second derivative, still with a centered incremental ratio. Thus, the upwind scheme can be re-interpreted as a centered finite difference scheme where an *artificial diffusion* term proportional to h has been introduced. As a matter of fact, (11.32) is equivalent to

$$-\mu_h \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + b \frac{u_{i+1} - u_{i-1}}{2h} = 0, \quad i = 1, \dots, M-1, \quad (11.33)$$

where $\mu_h = \mu(1 + \mathbb{P}\text{e})$, $\mathbb{P}\text{e}$ being the local Péclet number introduced in (11.19). Scheme (11.33) corresponds to the discretization using a CFD scheme of the *perturbed problem*

$$-\mu_h u'' + bu' = 0. \quad (11.34)$$

The viscosity “correction” $\mu_h - \mu = \mu\mathbb{P}\text{e} = \frac{bh}{2}$ is called *numerical viscosity* or *artificial viscosity*. The new local Péclet number, associated to the scheme (11.33), is

$$\mathbb{P}\text{e}^* = \frac{bh}{2\mu_h} = \frac{\mathbb{P}\text{e}}{(1 + \mathbb{P}\text{e})},$$

and therefore it verifies $\mathbb{P}\text{e}^* < 1$ for all possible values of $h > 0$. As we will see in the next section, this interpretation allows to extend the upwind technique to finite elements and also to the two-dimensional case, where incidentally the notion of decentered differentiation is not obvious.

More generally, in a CFD scheme of the form (11.33) we can use the following numerical viscosity coefficient

$$\mu_h = \mu(1 + \phi(\mathbb{P}\text{e})), \quad (11.35)$$

where ϕ is a suitable function of the local Péclet number that must satisfy the property $\lim_{t \rightarrow 0^+} \phi(t) = 0$. It can be easily observed that if $\phi = 0$, we obtain the CFD method (11.28), while if $\phi(t) = t$, we obtain the upwind UPFD method (11.32) (or (11.33)). Different choices of ϕ yield to different schemes. For instance, setting

$$\phi(t) = t - 1 + B(2t),$$

where B is the so-called *Bernoulli function* defined as

$$B(t) = \frac{t}{e^t - 1} \quad \text{if } t > 0, \quad \text{and} \quad B(0) = 1,$$

we obtain the *exponential fitting* scheme, generally attributed to Scharfetter and Gummel or to Iljin (in fact, it was originally introduced by Allen and Southwell [AS55]).

Having denoted by ϕ^U , resp. ϕ^{SG} , the two functions determined by the choices $\phi(t) = t$ and $\phi(t) = t - 1 - B(2t)$, we observe that $\phi^{SG} \simeq \phi^U$ if $\text{Pe} \rightarrow +\infty$, while $\phi^{SG} = \mathcal{O}(\text{Pe}^2)$ and $\phi^U = \mathcal{O}(\text{Pe})$ if $\text{Pe} \rightarrow 0^+$ (see Fig. 11.6).

It can be verified that for each given μ and b the Scharfetter-Gummel scheme is a second order scheme (with respect to h) and, because of this, it is sometimes called upwind scheme with optimal viscosity. In fact, it can also be verified that, in the case where f is constant – more generally, it is sufficient that f is constant in each interval $[x_i, x_{i+1}]$ – the numerical solution produced by this scheme is nodally exact, that is it exactly coincides with the solution u at each discretization node inside the interval $(0, 1)$, that is we have

$$u_i = u(x_i) \quad \text{for } i = 1, \dots, M-1,$$

independently of the choice of h (see Fig. 11.7).

We observe that the local Péclet number associated with the coefficient (11.35) is

$$\text{Pe}^* = \frac{bh}{2\mu_h} = \frac{\text{Pe}}{(1 + \phi(\text{Pe}))},$$

and is therefore always less than 1, for each value of h .

Remark 11.1 The matrix associated with the upwind and the exponential fitting scheme is a M-matrix *regardless* of the value of h ; hence, the numerical solution has a monotone behavior (see [QSS07, Chap. 1]). •

11.7 Eigenvalues of the diffusion-transport equation

Let us consider the operator $Lu = -\mu u'' + bu'$ associated to the problem (11.14) in a generic interval (α, β) . Its eigenvalues, λ , verify the problem $Lu = \lambda u$, $\alpha < x < \beta$, $u(\alpha) = u(\beta) = 0$, u being an eigenfunction. Such eigenvalues, in general, will be complex because of the presence of the first-order term bu' . Supposing $\mu > 0$ constant

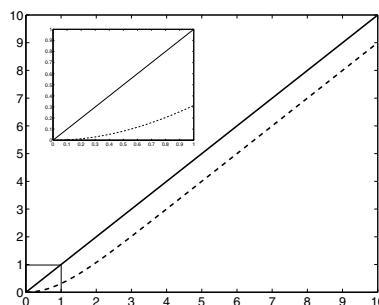


Fig. 11.6. The functions ϕ^U (solid line) and ϕ^{SG} (dashed line) versus the local Péclet number

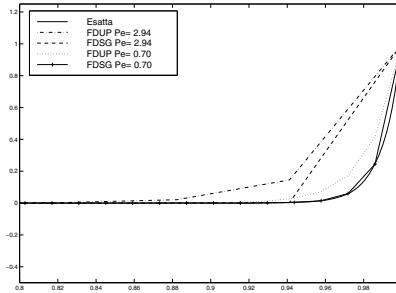


Fig. 11.7. Comparison between the exact solution and those obtained by the upwind scheme (UPFD) and the Scharfetter and Gummel one (SGFD) in the case where $\text{Pe}_g = 50$

(and b variable a priori), we have

$$\text{Re}(\lambda) = \frac{\int_{\alpha}^{\beta} Lu \bar{u} dx}{\int_{\alpha}^{\beta} |u|^2 dx} = \frac{\mu \int_{\alpha}^{\beta} |u'|^2 dx - \frac{1}{2} \int_{\alpha}^{\beta} b' |u|^2 dx}{\int_{\alpha}^{\beta} |u|^2 dx}. \quad (11.36)$$

It can be inferred that if μ is small and b' is strictly positive, the real part of λ is not necessarily positive. However, thanks to the Poincaré inequality

$$\int_{\alpha}^{\beta} |u'|^2 dx \geq C_{\alpha,\beta} \int_{\alpha}^{\beta} |u|^2 dx, \quad (11.37)$$

with $C_{\alpha,\beta}$ being a positive constant depending on $\beta - \alpha$, we deduce from (11.36) that

$$\text{Re}(\lambda) \geq C_{\alpha,\beta} \mu - \frac{1}{2} b'_{max}$$

being $b'_{max} = \max_{\alpha \leq s \leq \beta} b'(s)$. Thus, only a finite number of eigenvalues can have a negative real part. In particular, let us observe that

$$\text{Re}(\lambda) > 0 \quad \text{if } b \text{ is constant or if } b'(x) \leq 0 \quad \forall x \in [\alpha, \beta].$$

The same kind of lower bound can be obtained for the eigenvalues associated to the Galerkin-finite element approximation of the problem at hand. The latter are the solution of the problem

$$\text{find } \lambda_h \in \mathbb{C}, u_h \in V_h : \int_{\alpha}^{\beta} \mu u'_h v'_h dx + \int_{\alpha}^{\beta} b u'_h v_h dx = \lambda_h \int_{\alpha}^{\beta} u_h v_h dx \quad \forall v_h \in V_h, \quad (11.38)$$

where $V_h = \{v_h \in X_h^r : v_h(\alpha) = v_h(\beta) = 0\}$. To prove this, it suffices to take again $v_h = \bar{u}_h$ in (11.38) and proceed as previously.

We can instead obtain an upper bound by choosing again $v_h = \bar{u}_h$ in (11.38) and taking the modulus in both members:

$$|\lambda_h| \leq \frac{\mu \|u'_h\|_{L^2(\alpha,\beta)}^2 + \|b\|_{L^\infty(\alpha,\beta)} \|u'_h\|_{L^2(\alpha,\beta)} \|u_h\|_{L^2(\alpha,\beta)}}{\|u_h\|_{L^2(\alpha,\beta)}^2}.$$

By now using the *inverse inequality* (4.52) in the one-dimensional case

$$\exists C_I = C_I(r) > 0 : \forall v_h \in X_h^r, \|v'_h\|_{L^2(\alpha,\beta)} \leq C_I h^{-1} \|v_h\|_{L^2(\alpha,\beta)}, \quad (11.39)$$

we easily find that

$$|\lambda_h| \leq \mu C_I^2 h^{-2} + \|b\|_{L^\infty(\alpha,\beta)} C_I h^{-1}.$$

Should we use instead a Legendre G-NI spectral approximation of the same problem on the usual reference interval $(-1, 1)$, the eigenvalue problem would take the following form

$$\begin{aligned} \text{find } \lambda^N \in \mathbb{C}, u_N \in \mathbb{P}_N^0 : \\ (\mu u'_N, v'_N)_N + (bu'_N, v_N)_N = \lambda^N (u_N, v_N)_N \quad \forall v_N \in \mathbb{P}_N^0, \end{aligned} \quad (11.40)$$

with \mathbb{P}_N^0 now being the space of algebraic polynomials of degree N vanishing in $x = \pm 1$, and $(\cdot, \cdot)_N$ the discrete GLL scalar product defined in (10.25). We will suppose, for simplicity, that b is also constant. Taking $v_N = \bar{u}_N$, we obtain

$$\operatorname{Re}(\lambda^N) = \frac{\mu \|u'_N\|_{L^2(-1,1)}^2}{\|u_N\|_N^2},$$

thus $\operatorname{Re}(\lambda^N) > 0$. Thanks to the Poincaré inequality (11.37) (which holds in the interval $(-1, 1)$ with constant $C_{\alpha,\beta} = \pi^2/4$), we obtain the estimate from below

$$\operatorname{Re}(\lambda^N) > \mu \frac{\pi^2}{4} \frac{\|u_N\|_{L^2(-1,1)}^2}{\|u_N\|_N^2}.$$

As u_N is a polynomial of degree at most N , thanks to (10.54) we obtain

$$\operatorname{Re}(\lambda^N) > \mu \frac{\pi^2}{12}.$$

Instead, using the following inverse inequality for algebraic polynomials

$$\exists C > 0 : \forall v_N \in \mathbb{P}_N, \|v'_N\|_{L^2(-1,1)} \leq C N^2 \|v_N\|_{L^2(-1,1)} \quad (11.41)$$

(see [CHQZ06]) and once again (10.54), we find

$$\operatorname{Re}(\lambda^N) < C \mu N^4.$$

In fact, if $N > 1/\mu$, we can prove that the moduli of the eigenvalues of the diffusion-transport problem (11.40) behave like those of the pure diffusion problem, that is

$$C_1 N^{-1} \leq |\lambda^N| \leq C_2 N^{-2}.$$

For the proofs and for more details, see [CHQZ06, Sec. 4.3.3].

11.8 Stabilization methods

The Galerkin method introduced in the previous sections provides a centered approximation of the transport term. A possible way to decenter or desymmetrize such approximation consists in choosing the test functions v_h in a different space from the one to which u_h belongs: by so doing, we obtain a method called *Petrov-Galerkin*, for which the analysis based on the Céa lemma no longer holds. We will analyze this approach more in detail in Sec. 11.8.2. In this section we will deal instead with the methods of *stabilized finite elements*. As a matter of fact, as we will see, the schemes thus produced can be interpreted as special cases of Petrov-Galerkin methods.

The approximation of problem (11.16) using the Galerkin finite element method would be

$$\text{find } u_h \in V_h : \quad a(u_h, v_h) = F(v_h) \quad \forall v_h \in V_h, \quad (11.42)$$

being $V_h = \overset{\circ}{X}_h^r$, $r \geq 1$ and u_h an approximation of $\overset{\circ}{u}$. Instead, we consider the generalized Galerkin method

$$\text{find } u_h \in V_h : \quad a_h(u_h, v_h) = F_h(v_h) \quad \forall v_h \in V_h, \quad (11.43)$$

where

$$a_h(u_h, v_h) = a(u_h, v_h) + b_h(u_h, v_h) \quad \text{and} \quad F_h(v_h) = F(v_h) + G_h(v_h).$$

The additional terms $b_h(u_h, v_h)$ and $G_h(v_h)$ have the purpose of eliminating (or at least reducing) the numerical oscillations produced by the Galerkin method and are therefore named *stabilization terms*. The latter depend parametrically on h .

Remark 11.2 We want to point out that the term “*stabilization*” is in fact inexact. The Galerkin method is indeed already stable, in the sense of the continuity of the solution with respect to the data of problem (see what has been proved, e.g. in Sec. 11.1 for problem (11.1)). In this case, stabilization must be understood as the aim of reducing (ideally eliminating) the oscillations in the numerical solution when $\mathbb{P}e > 1$. •

Let us now see several possible ways to choose the stabilization terms.

11.8.1 Artificial diffusion and decentered finite element schemes

Based on what we have seen for finite differences and referring, for simplicity, to the one-dimensional case, we apply the Galerkin method to problem (11.14) by replacing

the viscosity coefficient μ with a coefficient $\mu_h = \mu(1 + \phi(\mathbb{P}\epsilon))$. This way, we end up adding to the original viscosity term μ an *artificial viscosity* equal to $\mu\phi(\mathbb{P}\epsilon)$, which depends on the discretization step h through the local Péclet number $\mathbb{P}\epsilon$.

Thus, we obtain the following approximation of (11.16)

$$\text{find } \overset{\circ}{u}_h \in V_h : \quad a_h(\overset{\circ}{u}_h, v_h) = F(v_h) \quad \forall v_h \in V_h, \quad (11.44)$$

being

$$a_h(\overset{\circ}{u}_h, v_h) = a(\overset{\circ}{u}_h, v_h) + b_h(\overset{\circ}{u}_h, v_h). \quad (11.45)$$

The additional stabilization term is given by

$$b_h(\overset{\circ}{u}_h, v_h) = \mu\phi(\mathbb{P}\epsilon) \int_0^1 \overset{\circ}{u}'_h v'_h dx. \quad (11.46)$$

Since

$$a_h(\overset{\circ}{u}_h, \overset{\circ}{u}_h) \geq \mu_h \|\overset{\circ}{u}_h\|_{H^1(\Omega)}^2$$

and $\mu_h \geq \mu$, we can state that problem (11.45) is “more coercive” (i.e. has a larger coercivity constant) than the discrete problem obtained using the standard Galerkin method, which we find by taking $a_h = a$ in (11.44).

The following result provides an a priori estimate of the error made approximating the solution of problem (11.16) with $\overset{\circ}{u}_h$.

Theorem 11.1 *Under the assumption that $u \in H^{r+1}(\Omega)$, the error between the solution of problem (11.16) and that of the approximate problem (11.44) can be upbounded as follows*

$$\begin{aligned} & \| \overset{\circ}{u} - \overset{\circ}{u}_h \|_{H^1(\Omega)} \leq \\ & C \frac{h^r}{\mu(1 + \phi(\mathbb{P}\epsilon))} \| \overset{\circ}{u} \|_{H^{r+1}(\Omega)} + \frac{\phi(\mathbb{P}\epsilon)}{1 + \phi(\mathbb{P}\epsilon)} \| \overset{\circ}{u} \|_{H^1(\Omega)}, \end{aligned} \quad (11.47)$$

with C a suitable positive constant independent of h and μ .

Proof. We can take advantage of the *Strang lemma*, previously introduced in Sec. 10.4.1, thanks to which we obtain

$$\begin{aligned} \| \overset{\circ}{u} - \overset{\circ}{u}_h \|_{H^1(\Omega)} & \leq \inf_{w_h \in V_h} \left\{ \left(1 + \frac{M}{\mu_h} \right) \| \overset{\circ}{u} - w_h \|_{H^1(\Omega)} \right. \\ & \left. + \frac{1}{\mu_h} \sup_{v_h \in V_h, v_h \neq 0} \frac{|a(w_h, v_h) - a_h(w_h, v_h)|}{\|v_h\|_{H^1(\Omega)}} \right\}. \end{aligned} \quad (11.48)$$

We choose $w_h = P_h^r \overset{\circ}{\hat{u}}$, the orthogonal projection of $\overset{\circ}{\hat{u}}$ on V_h with respect to the scalar product $\int_0^1 u' v' dx$ of $H_0^1(\Omega)$, that is

$$P_h^r \overset{\circ}{\hat{u}} \in V_h : \quad \int_0^1 (P_h^r \overset{\circ}{\hat{u}} - \overset{\circ}{\hat{u}})' v'_h dx = 0 \quad \forall v_h \in V_h.$$

It can be proven that (see [QV94, Chap. 3])

$$\|(P_h^r \overset{\circ}{\hat{u}})'\|_{L^2(\Omega)} \leq \|(\overset{\circ}{\hat{u}})'\|_{L^2(\Omega)} \quad \text{and} \quad \|P_h^r \overset{\circ}{\hat{u}} - \overset{\circ}{\hat{u}}\|_{H^1(\Omega)} \leq Ch^r \|\overset{\circ}{\hat{u}}\|_{H^{r+1}(\Omega)},$$

with C being a constant independent of h . Thus, we can upbound the first addendum of the right-hand side in (11.48) by $(C/\mu_h)h^r \|\overset{\circ}{\hat{u}}\|_{H^{r+1}(\Omega)}$.

Now, thanks to (11.45), we obtain

$$\frac{1}{\mu_h} \frac{|a(w_h, v_h) - a_h(w_h, v_h)|}{\|v_h\|_{H^1(\Omega)}} \leq \frac{\mu}{\mu_h} \phi(\mathbb{P}\mathbf{e}) \frac{1}{\|v_h\|_{H^1(\Omega)}} \left| \int_0^1 w'_h v'_h dx \right|.$$

Using the Cauchy-Schwarz inequality, and observing that

$$\|v'_h\|_{L^2(\Omega)} \leq \|v_h\|_{H^1(\Omega)} \quad \text{and that} \quad \|(P_h^r \overset{\circ}{\hat{u}})'\|_{L^2(\Omega)} \leq \|P_h^r \overset{\circ}{\hat{u}}\|_{H^1(\Omega)} \leq \|\overset{\circ}{\hat{u}}\|_{H^1(\Omega)},$$

we obtain

$$\frac{1}{\mu_h} \sup_{v_h \in V_h, v_h \neq 0} \frac{|a(P_h^r \overset{\circ}{\hat{u}}, v_h) - a_h(P_h^r \overset{\circ}{\hat{u}}, v_h)|}{\|v_h\|_{H^1(\Omega)}} \leq \frac{\phi(\mathbb{P}\mathbf{e})}{1 + \phi(\mathbb{P}\mathbf{e})} \|\overset{\circ}{\hat{u}}\|_{H^1(\Omega)}.$$

Inequality (11.47) is therefore proven. \diamond

Corollary 11.1 *For a given μ and for h tending to 0, we have*

$$\|\overset{\circ}{\hat{u}} - \overset{\circ}{\hat{u}_h}\|_{H^1(\Omega)} \leq C_1 \left[h^r \|\overset{\circ}{\hat{u}}\|_{H^{r+1}(\Omega)} + \phi(\mathbb{P}\mathbf{e}) \|\overset{\circ}{\hat{u}}\|_{H^1(\Omega)} \right], \quad (11.49)$$

where C_1 is a positive constant, independent of h , while for a given h and for μ tending to 0, we have

$$\|\overset{\circ}{\hat{u}} - \overset{\circ}{\hat{u}_h}\|_{H^1(\Omega)} \leq C_2 \left[h^{r-1} \|\overset{\circ}{\hat{u}}\|_{H^{r+1}(\Omega)} + \|\overset{\circ}{\hat{u}}\|_{H^1(\Omega)} \right], \quad (11.50)$$

where C_2 is a positive constant independent of h and μ .

Proof. We obtain (11.49) from (11.47) remembering that $\phi(\text{Pe}) \rightarrow 0$ for any given μ when $h \rightarrow 0$. To obtain (11.50) it is sufficient to observe that, in the *upwind* case, $\phi^U(\text{Pe}) = \text{Pe}$, thus

$$\mu(1 + \phi(\text{Pe})) = \mu + \frac{b}{2}h \quad \text{and} \quad \frac{\phi(\text{Pe})}{1 + \phi(\text{Pe})} = \frac{h}{h + 2\mu/b}$$

and that, in the case of the Scharfetter and Gummel method, $\phi^{SG}(\text{Pe}) \simeq \phi^U(\text{Pe})$ for a given h and μ tending to 0. \diamond

In particular, for a given μ , the stabilized method generates an error that decays linearly with respect to h (irrespectively of the degree r) when using the *upwind* viscosity, while with an artificial viscosity of the Scharfetter and Gummel type, the convergence rate becomes quadratic if $r \geq 2$. This result follows from the estimate (11.49) recalling that $\phi^U(\text{Pe}) = \mathcal{O}(h)$ while $\phi^{SG}(\text{Pe}) = \mathcal{O}(h^2)$ for a fixed μ and for $h \rightarrow 0$.

11.8.2 The Petrov-Galerkin method

An equivalent way to write the generalized Galerkin problem (11.44) with numerical viscosity is to reformulate it as a Petrov-Galerkin method, that is a method where the space of test functions is different from the space where the solution is sought. Precisely, the approximation takes the following form

$$\text{find } \overset{\circ}{u}_h \in V_h : \quad a(\overset{\circ}{u}_h, v_h) = F(v_h) \quad \forall v_h \in W_h, \quad (11.51)$$

where $W_h \neq V_h$, while the bilinear form $a(\cdot, \cdot)$ is the same as in the initial problem. It can be verified that, in the case of linear finite elements, that is for $r = 1$, problem (11.44)-(11.46) can be rewritten as (11.51) where W_h is the space generated by the functions $\psi_i(x) = \varphi_i(x) + B_i^\alpha$ (see Fig. 11.8, right). Here the $B_i^\alpha = \alpha B_i(x)$ are the so-called *bubble functions*, with

$$B_i(x) = \begin{cases} g\left(1 - \frac{x-x_{i-1}}{h}\right), & x_{i-1} \leq x \leq x_i, \\ -g\left(\frac{x-x_i}{h}\right), & x_i \leq x \leq x_{i+1}, \\ 0 & \text{otherwise,} \end{cases}$$

and $g(\xi) = 3\xi(1 - \xi)$, with $0 \leq \xi \leq 1$ (see Fig. 11.8, left) [ZT00]. In the case of *upwind* finite differences we have $\alpha = 1$, while in the case of the Scharfetter-Gummel scheme we have $\alpha = \coth(\text{Pe}) - 1/\text{Pe}$. Note that the test functions de-symmetrize themselves (with respect to the usual piecewise linear basis functions) under the effect of the convective field.

11.8.3 The artificial diffusion and streamline-diffusion methods in the two-dimensional case

The upwind artificial viscosity method can be generalized to the case where we consider a two-dimensional problem of the type (11.1). (Same remarks apply to a three

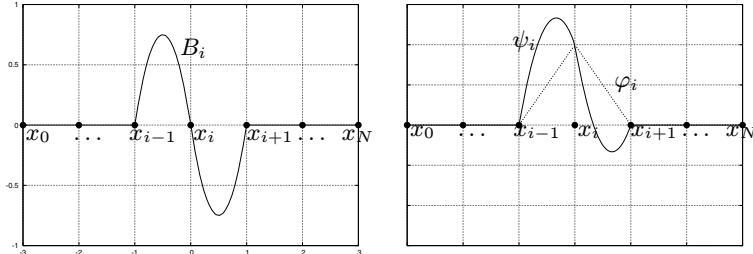


Fig. 11.8. Example of a bubble function B_i and of a basis function ψ_i of the space W_h

dimensional problem as well.) In such case, it will suffice to add to the bilinear form (11.3) a term of type

$$Qh \int_{\Omega} \nabla u_h \cdot \nabla v_h \, d\Omega \quad \text{for a chosen } Q > 0, \quad (11.52)$$

which corresponds to adding the artificial diffusion term $-Qh\Delta u$ to the initial problem (11.1). The corresponding method is called *upwind artificial diffusion*. This way, an additional diffusion is introduced, not only in the direction of the field \mathbf{b} , as it is right if the objective is to stabilize the oscillations generated by the Galerkin method, but also in the one orthogonal to the latter, which is not at all necessary. For instance, if we consider the problem

$$-\mu\Delta u + \frac{\partial u}{\partial x} = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega,$$

where the transport field is given by the vector $\mathbf{b} = [1, 0]^T$, the artificial diffusion term we wish to add would be

$$-Qh \frac{\partial^2 u}{\partial x^2} \quad \text{and not} \quad -Qh\Delta u = -Qh \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right).$$

More generally, we can add the following stabilization term

$$-Qh \operatorname{div} [(\mathbf{b} \cdot \nabla u) \mathbf{b}] = -Qh \operatorname{div} \left(\frac{\partial u}{\partial \mathbf{b}} \mathbf{b} \right), \quad \text{with } Q = |\mathbf{b}|^{-1}.$$

In the Galerkin problem, the latter yields the following term

$$b_h(u_h, v_h) = Qh(\mathbf{b} \cdot \nabla u_h, \mathbf{b} \cdot \nabla v_h) = Qh \left(\frac{\partial u_h}{\partial \mathbf{b}}, \frac{\partial v_h}{\partial \mathbf{b}} \right). \quad (11.53)$$

The resulting discrete problem, called *streamline-diffusion*, becomes

$$\text{find } u_h \in V_h : \quad a_h(u_h, v_h) = (f, v_h) \quad \forall v_h \in V_h,$$

where

$$a_h(u_h, v_h) = a(u_h, v_h) + b_h(u_h, v_h).$$

Basically, we are adding a term proportional to the second derivative in the direction of the field \mathbf{b} (also called *streamline*). Note that, in this case, the artificial viscosity coefficient is a *tensor*. As a matter of fact, the stabilization term $b_h(\cdot, \cdot)$ can be seen as the bilinear form associated to the operator $-\operatorname{div}(\boldsymbol{\mu}_a \nabla u)$ with $[\boldsymbol{\mu}_a]_{ij} = Q h b_i b_j$, b_i being the i -th component of \mathbf{b} .

Although the term (11.53) is less diffusive than (11.52), yet also for the streamline-diffusion method, the accuracy is only $\mathcal{O}(h)$. More accurate stabilization methods are described in sections (11.8.6), (11.8.7) and (11.8.8). To introduce them, we need some definitions that we will anticipate in sections (11.8.4) and (11.8.5).

11.8.4 Consistence and truncation error for the Galerkin and generalized Galerkin methods

For the generalized Galerkin problem (11.43), consider the difference between the left and right hand side when replacing the approximate solution u_h with the exact solution, i.e.

$$\tau_h(u; v_h) = a_h(u, v_h) - F_h(v_h). \quad (11.54)$$

The latter is a functional of the variable v_h , whose norm

$$\tau_h(u) = \sup_{v_h \in V_h, v_h \neq 0} \frac{|\tau_h(u; v_h)|}{\|v_h\|_V} \quad (11.55)$$

defines the *truncation error* associated to the method (11.43).

In accordance with the definitions given in Sec. 1.2, we will say that the generalized Galerkin method at hand is *consistent* if $\lim_{h \rightarrow 0} \tau_h(u) = 0$.

Moreover, we will say that it is *strongly* (or *fully*) *consistent* if the truncation error (11.55) results to be null for each value of h . The standard Galerkin method, for instance, is strongly consistent, as seen in Chap. 4, as $\forall v_h \in V_h$ we have

$$\tau_h(u; v_h) = a(u, v_h) - F(v_h) = 0.$$

However, the generalized Galerkin method is generally only consistent, as follows from the Strang lemma, as long as $a_h - a$ and $F_h - F$ “tend to zero” when h tends to zero.

As for the upwind and streamline-diffusion methods, we have

$$\begin{aligned} \tau_h(u; v_h) &= a_h(u, v_h) - F(v_h) \\ &= a_h(u, v_h) - a(u, v_h) = \begin{cases} Qh(\nabla u, \nabla v_h) & (\text{Upwind}), \\ Qh\left(\frac{\partial u}{\partial \mathbf{b}}, \frac{\partial v_h}{\partial \mathbf{b}}\right) & (\text{Streamline-Diffusion}). \end{cases} \end{aligned}$$

Hence, these are consistent but *not* strongly consistent methods.

11.8.5 Symmetric and skew-symmetric part of an operator

Let V be a Hilbert space and V' its dual. We will say that an operator $L : V \rightarrow V'$ is *symmetric* if

$${}_{V'}\langle Lu, v \rangle_V = {}_V\langle u, Lv \rangle_{V'}, \quad \forall u, v \in V,$$

skew-symmetric when

$${}_{V'}\langle Lu, v \rangle_V = -{}_V\langle u, Lv \rangle_{V'} \quad \forall u, v \in V.$$

An operator can be split into the sum of a symmetric part L_S and of a skew-symmetric part L_{SS} , i.e.

$$Lu = L_S u + L_{SS} u.$$

Let us consider, for instance, the following diffusion-transport-reaction operator

$$Lu = -\mu \Delta u + \operatorname{div}(\mathbf{b}u) + \sigma u, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, d \geq 2, \quad (11.56)$$

operating on the space $V = H_0^1(\Omega)$. Since

$$\begin{aligned} \operatorname{div}(\mathbf{b}u) &= \frac{1}{2} \operatorname{div}(\mathbf{b}u) + \frac{1}{2} \operatorname{div}(\mathbf{b}u) \\ &= \frac{1}{2} \operatorname{div}(\mathbf{b}u) + \frac{1}{2} u \operatorname{div}(\mathbf{b}) + \frac{1}{2} \mathbf{b} \cdot \nabla u, \end{aligned}$$

we can split it the following way

$$Lu = \underbrace{-\mu \Delta u + \left[\sigma + \frac{1}{2} \operatorname{div}(\mathbf{b}) \right] u}_{L_S u} + \underbrace{\frac{1}{2} [\operatorname{div}(\mathbf{b}u) + \mathbf{b} \cdot \nabla u]}_{L_{SS} u}.$$

Note that the reaction coefficient has become $\sigma^* = \sigma + \frac{1}{2} \operatorname{div}(\mathbf{b})$. We verify that the two parts into which the operator has been split are symmetric resp. skew-symmetric. Indeed, integrating twice by parts, we obtain $\forall u, v \in V$, indicating by (\cdot, \cdot) the scalar product of $L^2(\Omega)$:

$$\begin{aligned} {}_{V'}\langle L_S u, v \rangle_V &= \mu (\nabla u, \nabla v) + (\sigma^* u, v) \\ &= -\mu {}_V\langle u, \Delta v \rangle_{V'} + (u, \sigma^* v) \\ &= {}_V\langle u, L_S v \rangle_{V'}, \end{aligned}$$

$$\begin{aligned} {}_{V'}\langle L_{SS} u, v \rangle_V &= \frac{1}{2} (\operatorname{div}(\mathbf{b}u), v) + \frac{1}{2} (\mathbf{b} \cdot \nabla u, v) \\ &= -\frac{1}{2} (\mathbf{b}u, \nabla v) + \frac{1}{2} (\nabla u, \mathbf{b}v) \\ &= -\frac{1}{2} (u, \mathbf{b} \cdot \nabla v) - \frac{1}{2} (u, \operatorname{div}(\mathbf{b}v)) \\ &= -{}_V\langle u, L_{SS} v \rangle_{V'}. \end{aligned}$$

Remark 11.3 We recall that each matrix \mathbf{A} can be decomposed into the sum

$$\mathbf{A} = \mathbf{A}_S + \mathbf{A}_{SS},$$

where

$$\mathbf{A}_S = \frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$$

is a symmetric matrix, called *symmetric part* of \mathbf{A} and

$$\mathbf{A}_{SS} = \frac{1}{2}(\mathbf{A} - \mathbf{A}^T)$$

is a skew-symmetric matrix, called *skew-symmetric part* of \mathbf{A} . •

11.8.6 Strongly consistent methods (GLS, SUPG)

We consider a diffusion-transport-reaction problem that we write in the abstract form $Lu = f$ in Ω , with $u = 0$ on $\partial\Omega$. Let us consider the corresponding weak formulation

$$\text{find } u \in V = H_0^1(\Omega) : \quad a(u, v) = (f, v) \quad \forall v \in V,$$

with $a(\cdot, \cdot)$ being the bilinear form associated to L . A stabilized and strongly consistent method can be obtained by adding a further term to its Galerkin approximation (11.42), that is by considering the problem

$$\text{find } u_h \in V_h : \quad a(u_h, v_h) + \mathcal{L}_h(u_h, f; v_h) = (f, v_h) \quad \forall v_h \in V_h, \quad (11.57)$$

for a suitable form \mathcal{L}_h satisfying

$$\mathcal{L}_h(u, f; v_h) = 0 \quad \forall v_h \in V_h. \quad (11.58)$$

We observe that the form \mathcal{L}_h depends both on the approximate solution u_h and on the forcing term f . A possible choice that verifies (11.58) is

$$\mathcal{L}_h(u_h, f; v_h) = \mathcal{L}_h^{(\rho)}(u_h, f; v_h) = \sum_{K \in \mathcal{T}_h} \delta(Lu_h - f, \mathcal{S}_K^{(\rho)}(v_h))_{L^2(K)},$$

where we use the notation $(u, v)_{L^2(K)} = \int_K uv \, dK$, ρ and δ are parameters to assign and we have set

$$\mathcal{S}_K^{(\rho)}(v_h) = \frac{h_K}{|\mathbf{b}|}[L_{SS}v_h + \rho L_S v_h],$$

\mathbf{b} being the convective (or transport) field, L_S and L_{SS} the symmetric resp. skew-symmetric part of the operator L under exam and h_K the diameter of the generic element K .

To verify the consistence of (11.57), we set

$$a_h(u_h, v_h) = a(u_h, v_h) + \mathcal{L}_h^{(\rho)}(u_h, f; v_h).$$

Thanks to the definition (11.54) we obtain

$$\begin{aligned}\tau_h(u; v_h) &= a_h(u, v_h) - (f, v_h) = a(u, v_h) + \mathcal{L}_h^{(\rho)}(u, f; v_h) - (f, v_h) \\ &= \mathcal{L}_h^{(\rho)}(u, f; v_h) = 0.\end{aligned}$$

The latter equality derives from the fact that $Lu - f = 0$. Hence, $\tau_h(u) = 0$ and thus property (11.58) ensures that method (11.57) is strongly consistent. Let us now see some particular cases associated to three different choices of the ρ parameter:

- if $\rho = 1$ we obtain the method called *Galerkin Least-Squares* (GLS), where

$$\mathcal{S}_K^{(1)}(v_h) = \frac{h_K}{|\mathbf{b}|} L v_h;$$

if we take $v_h = u_h$ we see that, on each triangle, a term proportional to $\int_K (Lu_h)^2 dK$ has been added;

- if $\rho = 0$ we obtain the method named *Streamline Upwind Petrov-Galerkin* (SUPG) where

$$\mathcal{S}_K^{(0)}(v_h) = \frac{h_K}{|\mathbf{b}|} L_{SS} v_h;$$

- if $\rho = -1$ we obtain the so-called Douglas-Wang (DW) method where

$$\mathcal{S}_K^{(-1)}(v_h) = \frac{h_K}{|\mathbf{b}|} (L_{SS} - L_S) v_h.$$

We note that in the case where $\sigma^* = 0$ and we use \mathbb{P}_1 finite elements, the three previous methods coincide, as $-\Delta u_h|_K = 0 \quad \forall K \in \mathcal{T}_h$.

Let us now limit ourselves to the two most classical procedures, GLS ($\rho = 1$) and SUPG ($\rho = 0$). We define the “ ρ norm”

$$\|v\|_{(\rho)} = \{\mu \|\nabla v\|_{L^2(\Omega)}^2 + \|\sqrt{\gamma} v\|_{L^2(\Omega)}^2 + \sum_{K \in \mathcal{T}_h} \delta \left((L_{SS} + \rho L_S) v, \mathcal{S}_K^{(\rho)}(v) \right)_{L^2(K)}\}^{\frac{1}{2}},$$

where γ is a positive constant such that $+\frac{1}{2} \operatorname{div} \mathbf{b} + \sigma \geq \gamma > 0$. The following (stability) inequality holds: $\exists \alpha^*$ depending on γ and on the coercivity constant α of $a(\cdot, \cdot)$, such that

$$\|u_h\|_{(\rho)} \leq \frac{C}{\alpha^*} \|f\|_{L^2(\Omega)}, \tag{11.59}$$

where C is a suitable constant (see for instance (11.70)). Moreover, the following error estimate holds

$$\|u - u_h\|_{(\rho)} \leq Ch^{r+1/2} |u|_{H^{r+1}(\Omega)}, \tag{11.60}$$

hence the order of accuracy of the method increases when the degree r of the polynomials we employ increases, as in the standard Galerkin method. The proofs of (11.59) and (11.60) will be provided in Sec. 11.8.7.

The choice of the stabilization parameter δ , measuring the amount of artificial viscosity, is extremely important. To this end, we report in Table 11.1 the range admitted

for such parameter as a function of the chosen stabilized scheme. In the table, C_0 is the constant of the following *inverse inequality*

$$\sum_{K \in \mathcal{T}_h} h_K^2 \int_K |\Delta v_h|^2 dK \leq C_0 \|\nabla v_h\|_{L^2(\Omega)}^2 \quad \forall v_h \in X_h^r. \quad (11.61)$$

For a deeper analysis of such methods and for the proofs of the above-mentioned cases, we refer to [QV94], Chap. 8, and to [RST96]. We also suggest [Fun97] for the case of an approximation with spectral elements.

Table 11.1. Admissible values for the stabilization parameter δ

SUPG	$0 < \delta < 1/C_0$
GLS	$0 < \delta$
DW	$0 < \delta < 1/(2C_0)$

11.8.7 Analysis of the GLS method

In this section, we want to prove the stability property (11.59) and the convergence property (11.60) in the case of the GLS method (hence for $\rho = 1$).

We suppose that the differential operator L has the form (11.56), with $\mu > 0$ and $\sigma \geq 0$ constant, with homogeneous Dirichlet boundary conditions being assigned. The bilinear form $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$ associated to the L operator is therefore

$$a(u, v) = \mu \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega + \int_{\Omega} \operatorname{div}(\mathbf{b}u) v \, d\Omega + \int_{\Omega} \sigma u v \, d\Omega,$$

with $V = H_0^1(\Omega)$. For simplicity, we suppose in the following that there exist two constants γ_0 and γ_1 such that

$$0 < \gamma_0 \leq \gamma(\mathbf{x}) = \frac{1}{2} \operatorname{div}(\mathbf{b}(\mathbf{x})) + \sigma \leq \gamma_1 \quad \forall \mathbf{x} \in \Omega. \quad (11.62)$$

In this case the form $a(\cdot, \cdot)$ is coercive, as $a(v, v) \geq \mu \|\nabla v\|_{L^2(\Omega)}^2 + \gamma_0 \|v\|_{L^2(\Omega)}^2$. Following the procedure developed in Sec. 11.8.5, we can write the symmetric and skew-symmetric parts associated to L , respectively, as

$$L_S u = -\mu \Delta u + \gamma u, \quad L_{SS} u = \frac{1}{2} (\operatorname{div}(\mathbf{b}u) + \mathbf{b} \cdot \nabla u).$$

Moreover, we rewrite the stabilized formulation (11.57) by splitting $\mathcal{L}_h(u_h, f; v_h)$ in two terms, one containing u_h , the other f :

$$\text{find } u_h \in V_h : \quad a_h^{(1)}(u_h, v_h) = f_h^{(1)}(v_h) \quad \forall v_h \in V_h, \quad (11.63)$$

with

$$a_h^{(1)}(u_h, v_h) = a(u_h, v_h) + \sum_{K \in \mathcal{T}_h} \delta \left(L u_h, \frac{h_K}{|\mathbf{b}|} L v_h \right)_K \quad (11.64)$$

and

$$f_h^{(1)}(v_h) = (f, v_h) + \sum_{K \in \mathcal{T}_h} \delta \left(f, \frac{h_K}{|\mathbf{b}|} L v_h \right)_K. \quad (11.65)$$

We observe that, using these notations, the strong consistency property (11.58) is expressed via the equality

$$a_h^{(1)}(u, v_h) = f_h^{(1)}(v_h) \quad \forall v_h \in V_h. \quad (11.66)$$

We can now prove the following preliminary result.

Lemma 11.1 *For each $\delta > 0$, the bilinear form $a_h^{(1)}(\cdot, \cdot)$ defined in (11.64) satisfies the following relation*

$$\begin{aligned} a_h^{(1)}(v_h, v_h) &= \mu \|\nabla v_h\|_{L^2(\Omega)}^2 + \|\sqrt{\gamma} v_h\|_{L^2(\Omega)}^2 \\ &\quad + \sum_{K \in \mathcal{T}_h} \delta \left(\frac{h_K}{|\mathbf{b}|} L v_h, L v_h \right)_K \quad \forall v_h \in V_h. \end{aligned} \quad (11.67)$$

The previous result follows from the definition (11.64) (having chosen $v_h = u_h$) and from the hypothesis (11.62). In the case under exam, the norm $\|\cdot\|_{(1)}$, which we here denote for convenience by the symbol $\|\cdot\|_{GLS}$, becomes

$$\|v_h\|_{GLS}^2 = \mu \|\nabla v_h\|_{L^2(\Omega)}^2 + \|\sqrt{\gamma} v_h\|_{L^2(\Omega)}^2 + \sum_{K \in \mathcal{T}_h} \delta \left(\frac{h_K}{|\mathbf{b}|} L v_h, L v_h \right)_K. \quad (11.68)$$

We can prove the following stability result.

Lemma 11.2 *Let u_h be the solution provided by the GLS scheme. Then, there exists a constant $C > 0$, independent of h , such that*

$$\|u_h\|_{GLS} \leq C \|f\|_{L^2(\Omega)}.$$

Proof. We choose $v_h = u_h$ in (11.63). By exploiting Lemma 11.1 and definition (11.68), we can first write that

$$\|u_h\|_{GLS}^2 = a_h^{(1)}(u_h, u_h) = f_h^{(1)}(u_h) = (f, u_h) + \sum_{K \in \mathcal{T}_h} \delta \left(f, \frac{h_K}{|\mathbf{b}|} L u_h \right)_K. \quad (11.69)$$

We separately seek for an upper bound for the two right-hand side terms of (11.69), by suitably applying the Cauchy-Schwarz and Young inequalities. We thus obtain

$$\begin{aligned} (f, u_h) &= \left(\frac{1}{\sqrt{\gamma}} f, \sqrt{\gamma} u_h \right) \leq \left\| \frac{1}{\sqrt{\gamma}} f \right\|_{L^2(\Omega)} \left\| \sqrt{\gamma} u_h \right\|_{L^2(\Omega)} \\ &\leq \frac{1}{4} \left\| \sqrt{\gamma} u_h \right\|_{L^2(\Omega)}^2 + \left\| \frac{1}{\sqrt{\gamma}} f \right\|_{L^2(\Omega)}^2, \end{aligned}$$

$$\begin{aligned} \sum_{K \in \mathcal{T}_h} \delta \left(f, \frac{h_K}{|\mathbf{b}|} L u_h \right)_K &= \sum_{K \in \mathcal{T}_h} \left(\sqrt{\delta \frac{h_K}{|\mathbf{b}|}} f, \sqrt{\delta \frac{h_K}{|\mathbf{b}|}} L u_h \right)_K \\ &\leq \sum_{K \in \mathcal{T}_h} \left\| \sqrt{\delta \frac{h_K}{|\mathbf{b}|}} f \right\|_{L^2(K)} \left\| \sqrt{\delta \frac{h_K}{|\mathbf{b}|}} L u_h \right\|_{L^2(K)} \\ &\leq \sum_{K \in \mathcal{T}_h} \delta \left(\frac{h_K}{|\mathbf{b}|} f, f \right)_K + \frac{1}{4} \sum_{K \in \mathcal{T}_h} \delta \left(\frac{h_K}{|\mathbf{b}|} L u_h, L u_h \right)_K. \end{aligned}$$

By summing the two previous upper bounds and by exploiting again definition (11.68), we have

$$\|u_h\|_{GLS}^2 \leq \left\| \frac{1}{\sqrt{\gamma}} f \right\|_{L^2(\Omega)}^2 + \sum_{K \in \mathcal{T}_h} \delta \left(\frac{h_K}{|\mathbf{b}|} f, f \right)_K + \frac{1}{4} \|u_h\|_{GLS}^2,$$

that is, recalling that $h_K \leq h$,

$$\|u_h\|_{GLS}^2 \leq \frac{4}{3} \left[\left\| \frac{1}{\sqrt{\gamma}} f \right\|_{L^2(\Omega)}^2 + \sum_{K \in \mathcal{T}_h} \delta \left(\frac{h_K}{|\mathbf{b}|} f, f \right)_K \right] \leq C^2 \|f\|_{L^2(\Omega)}^2,$$

having set

$$C = \left(\frac{4}{3} \max_{\mathbf{x} \in \Omega} \left(\frac{1}{\gamma} + \delta \frac{h}{|\mathbf{b}|} \right) \right)^{1/2}. \quad (11.70)$$

◊

We observe that the previous result is valid with the only constraint that the stabilization parameter δ be positive.

In fact, such parameter might also vary for each element K . In this case, we would have δ_K instead of δ in (11.64) and (11.65), while the constant δ in (11.70) would have the meaning of $\max_{K \in \mathcal{T}_h} \delta_K$.

We now proceed to the analysis of convergence of the GLS method.

Theorem 11.2 Let us suppose that the space V_h satisfies the following local approximation property: for each $v \in V \cap H^{r+1}(\Omega)$, there exists a function $\hat{v}_h \in V_h$ such that

$$\|v - \hat{v}_h\|_{L^2(K)} + h_K |v - \hat{v}_h|_{H^1(K)} + h_K^2 |v - \hat{v}_h|_{H^2(K)} \leq Ch_K^{r+1} |v|_{H^{r+1}(K)} \quad (11.71)$$

for each $K \in \mathcal{T}_h$. Moreover, let us suppose that the following inequality holds for the local Péclet number of K

$$\text{Pe}_K(\mathbf{x}) = \frac{|\mathbf{b}(\mathbf{x})| h_K}{2\mu} > 1 \quad \forall \mathbf{x} \in K. \quad (11.72)$$

Finally, let us suppose that the inverse inequality (11.61) holds and that the stabilization parameter satisfies the relation $0 < \delta \leq 2C_0^{-1}$.

Then, the following estimate holds for the error associated to the GLS scheme

$$\|u_h - u\|_{GLS} \leq Ch^{r+1/2} |u|_{H^{r+1}(\Omega)}, \quad (11.73)$$

as long as $u \in H^{r+1}(\Omega)$.

Proof. First of all, we rewrite the error as follows

$$e_h = u_h - u = \sigma_h - \eta, \quad (11.74)$$

with $\sigma_h = u_h - \hat{u}_h$, $\eta = u - \hat{u}_h$, where $\hat{u}_h \in V_h$ is a function that depends on u and that satisfies property (11.71). If, for instance, $V_h = X_h^r \cap H_0^1(\Omega)$, we can choose $\hat{u}_h = \Pi_h^r u$, that is the finite element interpolant of u .

We start by estimating the norm $\|\sigma_h\|_{GLS}$. By exploiting the strong consistency of the GLS scheme given by (11.66), we obtain thanks to (11.63)

$$\|\sigma_h\|_{GLS}^2 = a_h^{(1)}(\sigma_h, \sigma_h) = a_h^{(1)}(u_h - u + \eta, \sigma_h) = a_h^{(1)}(\eta, \sigma_h).$$

Now, by definition (11.64) it follows that

$$\begin{aligned} a_h^{(1)}(\eta, \sigma_h) &= \mu \int_{\Omega} \nabla \eta \cdot \nabla \sigma_h \, d\Omega - \int_{\Omega} \eta \mathbf{b} \cdot \nabla \sigma_h \, d\Omega + \int_{\Omega} \sigma \eta \sigma_h \, d\Omega \\ &+ \sum_{K \in \mathcal{T}_h} \delta \left(L\eta, \frac{h_K}{|\mathbf{b}|} L\sigma_h \right)_K = \underbrace{\mu (\nabla \eta, \nabla \sigma_h)}_{(I)} - \underbrace{\sum_{K \in \mathcal{T}_h} (\eta, L\sigma_h)_K}_{(II)} + \underbrace{2(\gamma \eta, \sigma_h)}_{(III)} \\ &+ \underbrace{\sum_{K \in \mathcal{T}_h} (\eta, -\mu \Delta \sigma_h)_K}_{(IV)} + \underbrace{\sum_{K \in \mathcal{T}_h} \delta \left(L\eta, \frac{h_K}{|\mathbf{b}|} L\sigma_h \right)_K}_{(V)}. \end{aligned}$$

We now bound the terms (I)-(V) separately. By using the Cauchy-Schwarz and Young inequality appropriately, we obtain

$$\begin{aligned}
 \text{(I)} &= \mu(\nabla\eta, \nabla\sigma_h) \leq \frac{\mu}{4} \|\nabla\sigma_h\|_{L^2(\Omega)}^2 + \mu\|\nabla\eta\|_{L^2(\Omega)}^2, \\
 \text{(II)} &= -\sum_{K \in \mathcal{T}_h} (\eta, L\sigma_h)_K = -\sum_{K \in \mathcal{T}_h} \left(\sqrt{\frac{|\mathbf{b}|}{\delta h_K}} \eta, \sqrt{\frac{\delta h_K}{|\mathbf{b}|}} L\sigma_h \right)_K \\
 &\leq \frac{1}{4} \sum_{K \in \mathcal{T}_h} \delta \left(\frac{h_K}{|\mathbf{b}|} L\sigma_h, L\sigma_h \right)_K + \sum_{K \in \mathcal{T}_h} \left(\frac{|\mathbf{b}|}{\delta h_K} \eta, \eta \right)_K, \\
 \text{(III)} &= 2(\gamma\eta, \sigma_h) = 2(\sqrt{\gamma}\eta, \sqrt{\gamma}\sigma_h) \leq \frac{1}{2} \|\sqrt{\gamma}\sigma_h\|_{L^2(\Omega)}^2 + 2\|\sqrt{\gamma}\eta\|_{L^2(\Omega)}^2.
 \end{aligned}$$

For the term (IV), thanks again to the Cauchy-Schwarz and Young inequalities and following hypothesis (11.72) and the inverse inequality (11.61), we obtain

$$\begin{aligned}
 \text{(IV)} &= \sum_{K \in \mathcal{T}_h} (\eta, -\mu\Delta\sigma_h)_K \\
 &\leq \frac{1}{4} \sum_{K \in \mathcal{T}_h} \delta \mu^2 \left(\frac{h_K}{|\mathbf{b}|} \Delta\sigma_h, \Delta\sigma_h \right)_K + \sum_{K \in \mathcal{T}_h} \left(\frac{|\mathbf{b}|}{\delta h_K} \eta, \eta \right)_K \\
 &\leq \frac{1}{8} \delta \mu \sum_{K \in \mathcal{T}_h} h_K^2 (\Delta\sigma_h, \Delta\sigma_h)_K + \sum_{K \in \mathcal{T}_h} \left(\frac{|\mathbf{b}|}{\delta h_K} \eta, \eta \right)_K \\
 &\leq \frac{\delta C_0 \mu}{8} \|\nabla\sigma_h\|_{L^2(\Omega)}^2 + \sum_{K \in \mathcal{T}_h} \left(\frac{|\mathbf{b}|}{\delta h_K} \eta, \eta \right)_K.
 \end{aligned}$$

The term (V) can finally be bounded once again thanks to the Cauchy-Schwarz and Young inequalities as follows

$$\begin{aligned}
 \text{(V)} &= \sum_{K \in \mathcal{T}_h} \delta \left(L\eta, \frac{h_K}{|\mathbf{b}|} L\sigma_h \right)_K \\
 &\leq \frac{1}{4} \sum_{K \in \mathcal{T}_h} \delta \left(\frac{h_K}{|\mathbf{b}|} L\sigma_h, L\sigma_h \right)_K + \sum_{K \in \mathcal{T}_h} \delta \left(\frac{h_K}{|\mathbf{b}|} L\eta, L\eta \right)_K.
 \end{aligned}$$

Thanks to these upper bounds and exploiting once more the GLS norm definition (11.68), we obtain the following estimate

$$\begin{aligned}
 \|\sigma_h\|_{GLS}^2 &= a_h^{(1)}(\eta, \sigma_h) \leq \frac{1}{4} \|\sigma_h\|_{GLS}^2 \\
 &+ \frac{1}{4} \left(\|\sqrt{\gamma}\sigma_h\|_{L^2(\Omega)}^2 + \sum_{K \in \mathcal{T}_h} \delta \left(\frac{h_K}{|\mathbf{b}|} L\sigma_h, L\sigma_h \right)_K \right) + \frac{\delta C_0 \mu}{8} \|\nabla\sigma_h\|_{L^2(\Omega)}^2 \\
 &+ \mu \|\nabla\eta\|_{L^2(\Omega)}^2 + 2 \underbrace{\sum_{K \in \mathcal{T}_h} \left(\frac{|\mathbf{b}|}{\delta h_K} \eta, \eta \right)_K + 2\|\sqrt{\gamma}\eta\|_{L^2(\Omega)}^2}_{\mathcal{E}(\eta)} + \sum_{K \in \mathcal{T}_h} \delta \left(\frac{h_K}{|\mathbf{b}|} L\eta, L\eta \right)_K \\
 &\leq \frac{1}{2} \|\sigma_h\|_{GLS}^2 + \mathcal{E}(\eta),
 \end{aligned}$$

having exploited, in the last passage, the assumption that $\delta \leq 2C_0^{-1}$. Then, we can state that

$$\|\sigma_h\|_{GLS}^2 \leq 2\mathcal{E}(\eta).$$

We now estimate the term $\mathcal{E}(\eta)$, by separately bounding each of its addenda. To this end, we will basically use the local approximation property (11.71) and the requirement formulated in (11.72) on the local Péclet number \mathbb{P}_K . Moreover, we observe that the constants C , introduced in the remainder, depend neither on h nor on \mathbb{P}_K , but can depend on other quantities such as the constant γ_1 in (11.62), the reaction constant σ , the norm $\|\mathbf{b}\|_{L^\infty(\Omega)}$, the stabilization parameter δ . We then have

$$\begin{aligned} \mu \|\nabla \eta\|_{L^2(\Omega)}^2 &\leq C \mu h^{2r} |u|_{H^{r+1}(\Omega)}^2 \\ &\leq C \frac{\|\mathbf{b}\|_{L^\infty(\Omega)} h}{2} h^{2r} |u|_{H^{r+1}(\Omega)}^2 \leq C h^{2r+1} |u|_{H^{r+1}(\Omega)}^2, \end{aligned} \quad (11.75)$$

$$\begin{aligned} 2 \sum_{K \in \mathcal{T}_h} \left(\frac{|\mathbf{b}|}{\delta h_K} \eta, \eta \right)_K &\leq C \frac{\|\mathbf{b}\|_{L^\infty(\Omega)}}{\delta} \sum_{K \in \mathcal{T}_h} \frac{1}{h_K} h_K^{2(r+1)} |u|_{H^{r+1}(K)}^2 \\ &\leq C h^{2r+1} |u|_{H^{r+1}(\Omega)}^2, \end{aligned}$$

$$2 \|\sqrt{\gamma} \eta\|_{L^2(\Omega)}^2 \leq 2 \gamma_1 \|\eta\|_{L^2(\Omega)}^2 \leq C h^{2(r+1)} |u|_{H^{r+1}(\Omega)}^2, \quad (11.76)$$

having exploited, for controlling the third addendum, the assumption (11.62).

Finding an upper bound for the fourth addendum of $\mathcal{E}(\eta)$ results to be slightly more difficult: first, by elaborating on the term $L\eta$, we have

$$\begin{aligned} \sum_{K \in \mathcal{T}_h} \delta \left(\frac{h_K}{|\mathbf{b}|} L\eta, L\eta \right)_K &= \sum_{K \in \mathcal{T}_h} \delta \left\| \sqrt{\frac{h_K}{|\mathbf{b}|}} L\eta \right\|_{L^2(K)}^2 \\ &= \sum_{K \in \mathcal{T}_h} \delta \left\| -\mu \sqrt{\frac{h_K}{|\mathbf{b}|}} \Delta \eta + \sqrt{\frac{h_K}{|\mathbf{b}|}} \operatorname{div}(\mathbf{b}\eta) + \sigma \sqrt{\frac{h_K}{|\mathbf{b}|}} \eta \right\|_{L^2(K)}^2 \\ &\leq C \sum_{K \in \mathcal{T}_h} \delta \left(\left\| \mu \sqrt{\frac{h_K}{|\mathbf{b}|}} \Delta \eta \right\|_{L^2(K)}^2 + \left\| \sqrt{\frac{h_K}{|\mathbf{b}|}} \operatorname{div}(\mathbf{b}\eta) \right\|_{L^2(K)}^2 + \left\| \sigma \sqrt{\frac{h_K}{|\mathbf{b}|}} \eta \right\|_{L^2(K)}^2 \right). \end{aligned} \quad (11.77)$$

Now, with a similar computation to the one performed to obtain the estimates (11.75) and (11.76), it is easy to prove that the second and third addendum of the left-hand side of (11.77) can be bounded using a term of the form $C h^{2r+1} |u|_{H^{r+1}(\Omega)}^2$, for a suitable choice of the constant C . For the first addendum, we have

$$\begin{aligned} \sum_{K \in \mathcal{T}_h} \delta \left\| \mu \sqrt{\frac{h_K}{|\mathbf{b}|}} \Delta \eta \right\|_{L^2(K)}^2 &\leq \sum_{K \in \mathcal{T}_h} \delta \frac{h_K^2 \mu}{2} \|\Delta \eta\|_{L^2(K)}^2 \\ &\leq C \delta \|\mathbf{b}\|_{L^\infty(\Omega)} \sum_{K \in \mathcal{T}_h} h_K^3 \|\Delta \eta\|_{L^2(K)}^2 \leq C h^{2r+1} |u|_{H^{r+1}(\Omega)}^2, \end{aligned}$$

having again exploited the conditions (11.71) and (11.72). The latter bound allows us to conclude that

$$\mathcal{E}(\eta) \leq C h^{2r+1} |u|_{H^{r+1}(\Omega)}^2,$$

that is that

$$\|\sigma_h\|_{GLS} \leq C h^{r+1/2} |u|_{H^{r+1}(\Omega)}. \quad (11.78)$$

Reverting to (11.74), to obtain the desired estimate for the norm $\|u_h - u\|_{GLS}$, we still have to estimate $\|\eta\|_{GLS}$. This evidently leads to estimating three contributions as in (11.75), (11.76) and (11.77), respectively, i.e. to the estimate

$$\|\eta\|_{GLS} \leq C h^{r+1/2} |u|_{H^{r+1}(\Omega)}.$$

The desired estimate (11.73) follows by combining this result with (11.78). \diamond

11.8.8 Stabilization through bubble functions

The generalized Galerkin method considered in the previous sections yields a stable numerical solution owing to the *enrichment* of the bilinear form $a(\cdot, \cdot)$. An alternative strategy consists of adopting a *richer subspace* than the standard one V_h . The idea is then to choose both the approximate solution and the test functions in the enriched space, therefore remaining within a classical Galerkin framework.

Referring to the usual diffusion-transport-reaction problem of the form (11.1), we introduce the finite dimensional space

$$V_h^b = V_h \oplus B,$$

where $V_h = X_h^r \cap H_0^1(\Omega)$ is the usual space and B is a finite dimensional space of *bubble functions*, or

$$B = \{v_B \in H_0^1(\Omega) : v_B|_K = c_K b_K, b_K|_{\partial K} = 0, \text{ and } c_K \in \mathbb{R}\}.$$

On each element K we then add the correction term b_K , for which several different choices are possible. As we only wish to work on the initial grid \mathcal{T}_h associated to the space V_h , a standard choice leads to defining $b_K = \lambda_1 \lambda_2 \lambda_3$ where the λ_i , for $i = 0, \dots, 2$, are the barycentric coordinates on K , i.e. linear polynomials, defined on K , each of which vanishes on one of the sides of the triangle and takes the value 1 at the vertex opposed to such side. (See Sec. 4.4.3 for their definition). The function b_K coincides in this case with the so-called *cubic bubble* that takes value 0 on the boundary of K and positive values inside it (see Fig. 11.9 (left)). The c therefore results to be the only degree of freedom associated to the triangle K (it will coincide, for instance, with the largest value taken by b_K on K or with the value it takes in the center of gravity). (see Sec. 4.4.3).

Remark 11.4 In order to introduce a *computational subgrid* on the Ω domain (obtained as a suitable refinement of the mesh \mathcal{T}_h), we can adopt more complex definitions for the bubble function b_K . For instance, b_K could be a piecewise linear function,

again defined on the element K and assuming the value 0 on the boundary of the triangle (as the basis function of the linear finite elements, associated to some point inside K) (see Fig. 11.9 (right)) [EG04]. •

At this point, we can introduce the Galerkin approximation on the space V_h^b of the problem under exam, which will take the form

$$\text{find } u_h^b \in V_h^b : \quad a(u_h + u_b, v_h^b) = (f, v_h^b) \quad \forall v_h^b \in V_h^b, \quad (11.79)$$

with $a(\cdot, \cdot)$ being the bilinear form associated to the differential operator L .

We propose to rewrite (11.79) as a stabilized Galerkin scheme in V_h , by eliminating function u_b . So far we can only say that, in each element K , $u_b|_K = c_{b,K} b_K$, for a suitable (unknown) constant $c_{b,K}$, with $u_b \in B$.

We decompose both u_h^b and v_h^b as a sum of a function of V_h and a function of B , that is

$$u_h^b = u_h + u_b, \quad v_h^b = v_h + v_b.$$

We first select as a test function v_h^b in (11.79) the one identified by $v_h = 0$ and $v_b \in B$ such that

$$v_b = \begin{cases} b_K & \text{in } K, \\ 0 & \text{elsewhere.} \end{cases}$$

We then have

$$a(u_h + u_b, v_b) = a_K(u_h + c_{b,K} b_K, b_K),$$

having denoted by $a_K(\cdot, \cdot)$ the restriction of the bilinear form $a(\cdot, \cdot)$ to the element K . We can therefore rewrite (11.79) as

$$a_K(u_h, b_K) + c_{b,K} a_K(b_K, b_K) = (f, b_K)_K. \quad (11.80)$$

Exploiting the fact that b_K vanishes on the boundary of K , we can integrate by parts the first term of (11.80), obtaining $a_K(u_h, b_K) = (Lu_h, b_K)_K$, or getting the unknown value of the constant $c_{b,K}$, given by

$$c_{b,K} = \frac{(f - Lu_h, b_K)_K}{a_K(b_K, b_K)}.$$

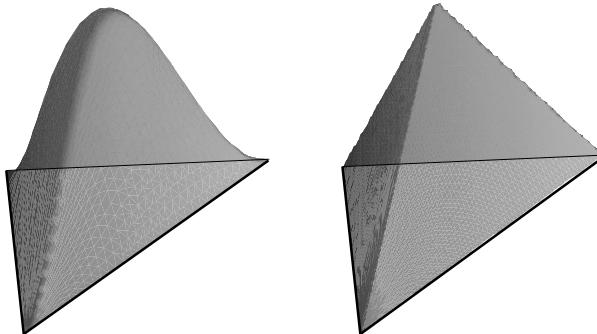


Fig. 11.9. Example of a cubic (left) and linear (right) bubble

We now choose as a test function v_h^b in (11.79) the one identified by any function $v_h \in V_h$ and by $v_b = 0$, thus obtaining

$$a(u_h, v_h) + \sum_{K \in \mathcal{T}_h} c_{b,K} a_K(b_K, v_h) = (f, v_h). \quad (11.81)$$

Let us suitably rewrite $a_K(b_K, v_h)$. By integrating by parts and exploiting the definitions of symmetric and skew-symmetric parts of the differential operator L (see Sec. 11.8.5), we have

$$\begin{aligned} a_K(b_K, v_h) &= \int_K \mu \nabla b_K \cdot \nabla v_h \, dK + \int_K \mathbf{b} \cdot \nabla b_K v_h \, dK + \int_K \sigma b_K v_h \, dK \\ &= - \int_K \mu b_K \Delta v_h \, dK + \int_{\partial K} \mu b_K \nabla v_h \cdot \mathbf{n} \, d\gamma - \int_K b_K \nabla v_h \cdot \mathbf{b} \, dK \\ &+ \int_{\partial K} \mathbf{b} \cdot \mathbf{n} v_h b_K \, d\gamma + \int_K \sigma b_K v_h \, dK = (b_K, (L_S - L_{SS})v_h)_K. \end{aligned}$$

We have exploited the property that the bubble function b_K vanishes on the boundary of the element K , and moreover that $\operatorname{div}(\mathbf{b}) = 0$. In a very analogous manner we can rewrite the denominator of the constant $c_{b,K}$ in the following way

$$a_K(b_K, b_K) = (L_S b_K, b_K)_K.$$

Reverting to (11.81), we thus have

$$a(u_h, v_h) + a_B(u_h, f; v_h) = (f, v_h) \quad \forall v_h \in V_h,$$

where

$$a_B(u_h, f; v_h) = \sum_{K \in \mathcal{T}_h} \frac{(Lu_h - f, b_K)_K (L_{SS}v_h - L_S v_h, b_K)_K}{(L_S b_K, b_K)_K}.$$

We have therefore found a stabilized Galerkin scheme, which can be formulated in the strongly consistent form (11.57). In the case where \mathbf{b} is constant, we can identify it using a sort of generalized Douglas-Wang method.

By choosing a convenient bubble b_K and following an analogous procedure to the one illustrated above, it is possible to also define generalized SUPG and GLS methods (see [BFHR97]). Similar strategies based on the so-called *subgrid viscosity* can be successfully used. See [EG04] for an extensive analysis.

11.9 Some numerical tests

We now present some numerical solutions obtained using the finite elements methods for the following two-dimensional diffusion-transport problem

$$\begin{cases} -\mu \Delta u + \mathbf{b} \cdot \nabla u = 1 & \text{in } \Omega = (0, 1) \times (0, 1), \\ u = 0 & \text{on } \partial\Omega, \end{cases} \quad (11.82)$$

where $\mathbf{b} = (1, 1)^T$. Note that the solution is characterized by a boundary layer near the sides $x = 1$ and $y = 1$. We have considered two different viscosity values: $\mu = 10^{-3}$ and $\mu = 10^{-5}$. We compare the solutions obtained using the standard and GLS Galerkin method, respectively, for both problems, by making two different choices for the uniform discretization step h : $1/20$ and $1/80$, respectively. The cross combinations of the two values for μ and h yield four different values for the local Péclet number Pe . As it can be observed by analyzing Fig. 11.10-11.13 (bearing in mind the different vertical scales), for growing Péclet numbers, the solution provided by the standard Galerkin method denotes stronger and stronger fluctuations. The latter eventually completely overcome the numerical solution (see Fig. 11.13). On the other hand, the GLS method is able to provide an acceptable numerical solution even for extremely high values of Pe (even though it develops an *over-shoot* at point $(1, 1)$).

11.10 An example of goal-oriented adaptivity

As anticipated in Remark 4.10, the a posteriori analysis presented in Sec. 4.6.5 for the control of a suitable functional of the error can be extended to differential problems of various kinds assuming a suitable redefinition of the local residue (4.96) and of the generalized jump (4.92). A grid adaptation results indeed to be particularly useful when dealing with diffusion-transport problems with dominant transport. Here, an accurate placement of the mesh triangles, e.g. at the (internal or boundary) layers can dramatically reduce the computational cost.

Let us consider problem (11.1) with $\mu = 10^{-3}$, $\mathbf{b} = (y, -x)^T$, σ and f identically null, and Ω coinciding with the *L*-shaped domain (reported in Fig. 11.14) described by the relation $(0, 4)^2 \setminus (0, 2)^2$. Let us suppose to assign a homogeneous Neumann condition on the sides $\{x = 4\}$ and $\{y = 0\}$, a non-homogeneous Dirichlet condition ($u = 1$) on the edge $\{x = 0\}$, and a homogeneous Dirichlet condition on the remaining parts of the boundary. The solution u of (11.1) thus results to be characterized by two internal layers having a circular shape. In order to test the sensitivity of

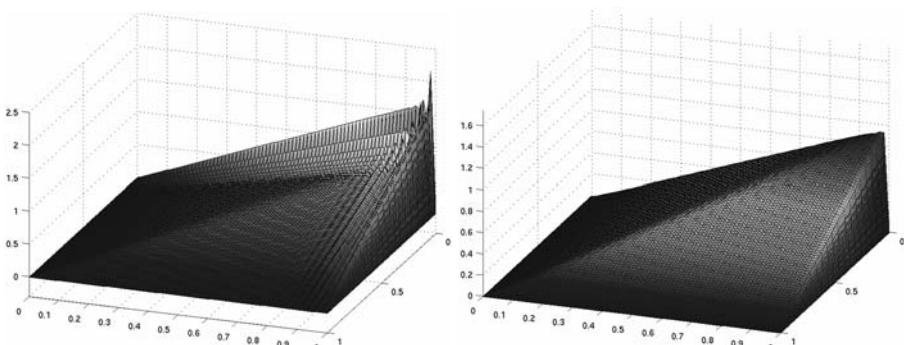


Fig. 11.10. Approximation of problem (11.82) with $\mu = 10^{-3}$, $h = 1/80$, using the standard (left) and GLS (right) Galerkin method. The corresponding local Péclet number is $\text{Pe} = 8.84$

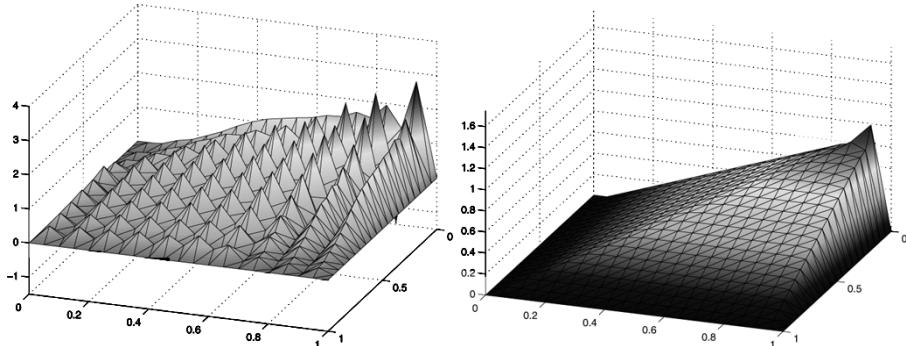


Fig. 11.11. Approximation of problem (11.82) with $\mu = 10^{-3}$, $h = 1/20$, using the standard (left) and GLS (right) Galerkin method. The corresponding local Péclet number is $\text{Pe} = 35.35$

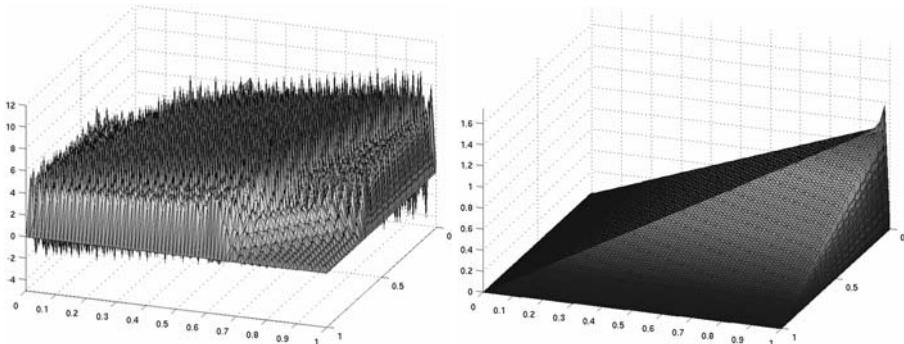


Fig. 11.12. Approximation of problem (11.82) with $\mu = 10^{-5}$, $h = 1/80$, using the standard (left) and GLS (right) Galerkin method. The corresponding local Péclet number is $\text{Pe} = 883.88$

the adapted grid with respect to the specific choice made for the functional J , let us consider the two following options:

$$J(v) = J_1(v) = \int_{\Gamma_1} \mathbf{b} \cdot \mathbf{n} v \, ds, \quad \text{with} \quad \Gamma_1 = \{x = 4\} \cup \{y = 0\},$$

for the control of the outgoing normal flow through the edges $\{x = 4\}$ and $\{y = 0\}$, and

$$J(v) = J_2(v) = \int_{\Gamma_2} \mathbf{b} \cdot \mathbf{n} v \, ds, \quad \text{with} \quad \Gamma_2 = \{x = 4\},$$

in the case where we are still interested in controlling the flow, but only through the $\{x = 4\}$ side. Starting from a quasi-uniform initial grid of 1024 elements, we show in Fig. 11.14 the (anisotropic) grids obtained for the choice $J = J_1$ (left) resp. $J = J_2$ (right), at the fourth and second iteration of the adaptive process. As it can be observed, while both boundary layers are responsible for the flow through Γ_1 , with a consequent

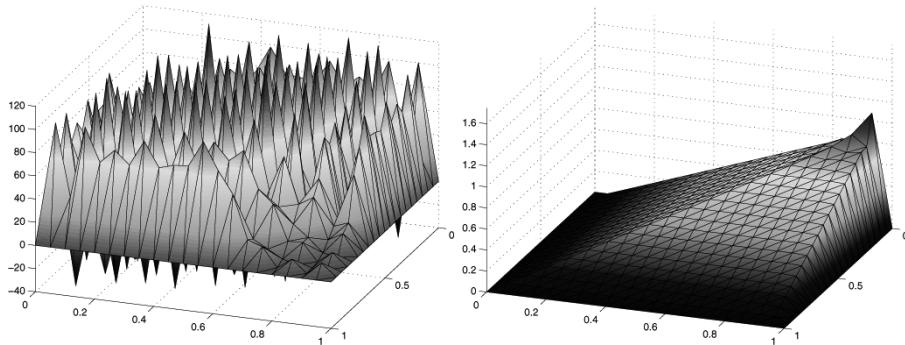


Fig. 11.13. Approximation of problem (11.82) with $\mu = 10^{-5}$, $h = 1/20$, using the standard (left) and GLS (right) Galerkin method. The corresponding local Péclet number is $\text{Pe} = 3535.5$

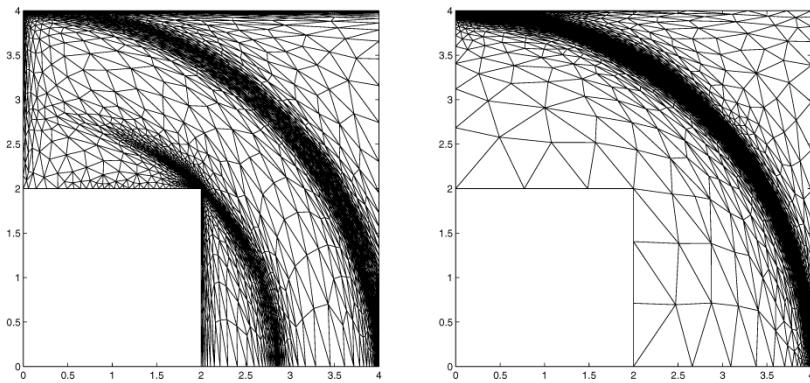


Fig. 11.14. Fourth adapted grid for the functional J_1 (left); second adapted grid for the functional J_2 (right)

refinement of the grid in correspondence of the two layers, only the upper layer is “recognized” as carrying information to the flow along Γ_2 . Finally, note the strongly anisotropic nature of the mesh in the figure, that is not only the refinement but also the correct orientation of the grid triangles in order to follow the directional properties (the boundary layers) of the solution. For further details, refer to [FMP04].

11.11 Exercises

1. Split in its symmetric and skew-symmetric parts the one-dimensional diffusion-transport-reaction operator

$$Lu = -\mu u'' + bu' + \sigma u.$$

2. Split in its symmetric and skew-symmetric parts the diffusion-transport operator written in the non divergence form

$$Lu = -\mu \Delta u + \mathbf{b} \cdot \nabla u.$$

3. Prove that the one-dimensional linear, quadratic and cubic finite elements yield, in the reference interval $[0, 1]$, the following condensed matrices, obtained via the *mass-lumping* technique:

$$r = 1 \quad M_L = \widehat{M} = \frac{1}{2} \text{diag}(1 \ 1),$$

$$r = 2 \quad M_L = \widehat{M} = \frac{1}{6} \text{diag}(1 \ 4 \ 1),$$

$$r = 3 \quad \begin{cases} M_L = \frac{1}{8} \text{diag}(1 \ 3 \ 3 \ 1), \\ \widehat{M} = \frac{1}{1552} \text{diag}(128 \ 648 \ 648 \ 128) = \text{diag}\left(\frac{8}{97}, \frac{81}{194}, \frac{81}{194}, \frac{8}{97}\right). \end{cases}$$

4. Consider the problem

$$\begin{cases} -\epsilon u''(x) + bu'(x) = 1, & 0 < x < 1, \\ u(0) = \alpha, & u(1) = \beta, \end{cases}$$

where $\epsilon > 0$ and $\alpha, \beta, b \in \mathbb{R}$ are given. Find its finite element formulation with upwind artificial viscosity. Discuss its stability and convergence properties and compare them with that of Galerkin-linear finite elements formulation.

5. Consider the problem

$$\begin{cases} -\epsilon u''(x) + u'(x) = 1, & 0 < x < 1, \\ u(0) = 0, & u'(1) = 1, \end{cases}$$

with $\epsilon > 0$ being given. Write its weak formulation and its approximation of the Galerkin-finite element type. Verify that the scheme is stable and explain such result.

6. Consider the problem

$$\begin{cases} -\text{div}(\mu \nabla u) + \text{div}(\beta u) + \sigma u = f & \text{in } \Omega, \\ -\gamma \cdot \mathbf{n} + \mu \nabla u \cdot \mathbf{n} = 0 & \text{on } \Gamma_N, \\ u = 0 & \text{on } \Gamma_D, \end{cases}$$

where Ω is an open subset of \mathbb{R}^2 with boundary $\Gamma = \Gamma_D \cup \Gamma_N$, $\Gamma_D \cap \Gamma_N = \emptyset$, \mathbf{n} is the normal outgoing vector to Γ , $\mu = \mu(\mathbf{x}) > \mu_0 > 0$, $\sigma = \sigma(\mathbf{x}) > 0$, $f = f(\mathbf{x})$ are given scalar functions, $\beta = \beta(\mathbf{x})$, $\gamma = \gamma(\mathbf{x})$ are given vector functions.

Approximate it using the Galerkin-linear finite element method. State under which hypotheses on the coefficients μ , σ and β the method results to be inaccurate and suggest the relevant remedies in the different cases.

7. Consider the diffusion-transport one-dimensional problem

$$\begin{cases} -(\mu u' - \psi' u)' = 1, & 0 < x < 1, \\ u(0) = u(1) = 0, \end{cases} \quad (11.83)$$

where μ is a positive constant and ψ a given function.

- a) Study the existence and uniqueness of problem (11.83) by introducing suitable hypotheses on the function ψ and propose a stable numerical approximation with finite elements.
 - b) Consider the variable change $u = \rho e^{\psi/\mu}$, ρ being an auxiliary unknown function. Study the existence and uniqueness of the weak solution of the problem (11.83) in the new unknown ρ and provide its numerical approximation using the finite elements method.
 - c) Compare the two approaches followed in (a) and (b), both from the abstract viewpoint and from the numerical one.
8. Consider the diffusion-transport-reaction problem

$$\begin{cases} -\Delta u + \operatorname{div}(\mathbf{b}u) + u = 0 & \text{in } \Omega \subset \mathbb{R}^2, \\ u = \varphi & \text{on } \Gamma_D, \\ \frac{\partial u}{\partial n} = 0 & \text{on } \Gamma_N, \end{cases}$$

where Ω is an open bounded domain, $\partial\Omega = \Gamma_D \cup \Gamma_N$, $\Gamma_D \neq \emptyset$.

Prove the existence and uniqueness of the solution by making suitable regularity assumptions on the data $\mathbf{b} = (b_1(\mathbf{x}), b_2(\mathbf{x}))^T$ ($\mathbf{x} \in \Omega$) and $\varphi = \varphi(\mathbf{x})$ ($\mathbf{x} \in \Gamma_D$).

In the case where $|\mathbf{b}| \gg 1$, approximate the same problem with the artificial diffusion-finite elements and SUPG-finite elements methods, discussing advantages and disadvantages with respect to the Galerkin finite element method.

9. Consider the problem

$$\begin{cases} -\sum_{i,j=1}^2 \frac{\partial^2 u}{\partial x_i \partial x_j} + \beta \frac{\partial^2 u}{\partial x_1^2} + \gamma \frac{\partial^2 u}{\partial x_1 \partial x_2} + \delta \frac{\partial^2 u}{\partial x_2^2} + \eta \frac{\partial u}{\partial x_1} = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases}$$

where $\beta, \gamma, \delta, \eta$ are given coefficients and f is a given function of $\mathbf{x} = (x_1, x_2) \in \Omega$.

- a) Find the conditions on the data that ensure the existence and uniqueness of a weak solution.
- b) Provide an approximation using the Galerkin finite element method and analyze its convergence.
- c) Under which conditions on the data is the Galerkin problem symmetric?
In such case, provide suitable methods for the solution of the associated algebraical problem.

12

Finite differences for hyperbolic equations

In this chapter, we will deal with time-dependent problems of hyperbolic type. For their derivation and for an in-depth analysis see e.g. [Sal08], Chap. 4. We will limit ourselves to considering the numerical approximation using the finite difference method, which was historically the first one to be applied to this type of equations. To introduce in a simple way the basic concepts of the theory, most of our presentation will concern problems depending on a single space variable. Finite element approximations will be addressed in Chap. 13, the extension to nonlinear problems in Chap. 14.

12.1 A scalar transport problem

Let us consider the following scalar hyperbolic problem

$$\begin{cases} \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0, & x \in \mathbb{R}, t > 0, \\ u(x, 0) = u_0(x), & x \in \mathbb{R}, \end{cases} \quad (12.1)$$

where $a \in \mathbb{R} \setminus \{0\}$. The solution of such problem is a wave travelling at velocity a , given by

$$u(x, t) = u_0(x - at), \quad t \geq 0.$$

We consider the curves $x(t)$ in the plane (x, t) , solutions of the following ordinary differential equations

$$\begin{cases} \frac{dx}{dt} = a, & t > 0, \\ x(0) = x_0, \end{cases}$$

for varying values of $x_0 \in \mathbb{R}$.

Such curves are called *characteristic lines* (often simply characteristics) and the solution along these lines remains constant as

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} \frac{dx}{dt} = 0.$$

In the case of the more general problem

$$\begin{cases} \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} + a_0 u = f, & x \in \mathbb{R}, t > 0, \\ u(x, 0) = u_0(x), & x \in \mathbb{R}, \end{cases} \quad (12.2)$$

where a , a_0 , and f are given functions of the (x, t) variables, the characteristic lines $x(t)$ are the solutions of the Cauchy problem

$$\begin{cases} \frac{dx}{dt} = a(x, t), & t > 0, \\ x(0) = x_0. \end{cases}$$

In such case, the solutions of (12.2) satisfy the following relation

$$\frac{d}{dt}u(x(t), t) = f(x(t), t) - a_0(x(t), t)u(x(t), t).$$

It is therefore possible to extract the solution u by solving an ordinary differential equation on each characteristic curve (this approach leads to the so-called *characteristic method*).

Let us now consider problem (12.1) in a bounded interval. For instance, let us suppose $x \in [0, 1]$ and $a > 0$. As u is constant on the characteristics, from Fig. 12.1 we deduce that the value of the solution at point P coincides with the value of u_0 at the foot P_0 of the characteristic outgoing from P . Instead, the characteristic outgoing from point Q , intersects the straight line $x = 0$ for $t > 0$. The point $x = 0$ is therefore an inflow point and must necessarily be assigned the value of u . Note that if $a < 0$, the inflow point would be $x = 1$.

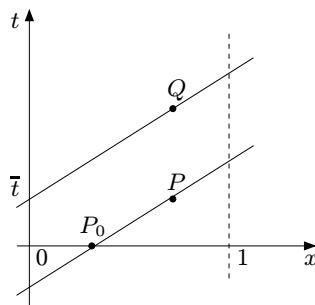


Fig. 12.1. Examples of characteristic lines (of straight lines in this case) issuing from points P and Q

By referring to problem (12.1) it is useful to observe that if u_0 were a discontinuous function at x_0 , then such discontinuity would propagate along the characteristic outgoing from x_0 (this process can be rigorously formalized from a mathematical viewpoint, by introducing the concept of *weak solution* for hyperbolic problems). In order to regularize the discontinuity, one could approximate the initial datum u_0 with a sequence of regular functions $u_0^\varepsilon(x)$, $\varepsilon > 0$. However, this procedure is only effective if the hyperbolic problem is linear. The solutions of non-linear hyperbolic problems can indeed develop discontinuities also for regular initial data (as we will see in Chap. 14). In this case, the strategy (which also inspires numerical methods) is to regularize the differential equation itself rather than the initial datum. In such perspective, we can consider the following diffusion-transport equation

$$\frac{\partial u^\varepsilon}{\partial t} + a \frac{\partial u^\varepsilon}{\partial x} = \varepsilon \frac{\partial^2 u^\varepsilon}{\partial x^2}, \quad x \in \mathbb{R}, t > 0,$$

for small values of $\varepsilon > 0$, which can be regarded as a parabolic regularization of equation (12.1). If we set $u^\varepsilon(x, 0) = u_0(x)$, we can prove that

$$\lim_{\varepsilon \rightarrow 0^+} u^\varepsilon(x, t) = u_0(x - at), \quad t > 0, \quad x \in \mathbb{R}.$$

12.1.1 An a priori estimate

Let us now return to the transport-reaction problem (12.2) on a bounded interval

$$\begin{cases} \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} + a_0 u = f, & x \in (\alpha, \beta), t > 0, \\ u(x, 0) = u_0(x), & x \in [\alpha, \beta], \\ u(\alpha, t) = \varphi(t), & t > 0, \end{cases} \quad (12.3)$$

where $a(x)$, $f(x, t)$ and $\varphi(t)$ are assigned functions; we have made the assumption that $a(x) > 0$, so that $x = \alpha$ is the inflow point (where to impose the boundary condition), while $x = \beta$ is the outflow point.

By multiplying the first equation of (12.3) by u , integrating with respect to x and using the formula of integration by parts, we obtain for each $t > 0$

$$\frac{1}{2} \frac{d}{dt} \int_\alpha^\beta u^2 dx + \int_\alpha^\beta (a_0 - \frac{1}{2} a_x) u^2 dx + \frac{1}{2} (au^2)(\beta) - \frac{1}{2} (au^2)(\alpha) = \int_\alpha^\beta f u dx.$$

By supposing that there exists a $\mu_0 \geq 0$ s.t.

$$a_0 - \frac{1}{2} a_x \geq \mu_0 \quad \forall x \in [\alpha, \beta],$$

we find

$$\frac{1}{2} \frac{d}{dt} \|u(t)\|_{L^2(\alpha, \beta)}^2 + \mu_0 \|u(t)\|_{L^2(\alpha, \beta)}^2 + \frac{1}{2} (au^2)(\beta) \leq \int_\alpha^\beta f u dx + \frac{1}{2} a(\alpha) \varphi^2(t).$$

If f and φ are identically null, then

$$\|u(t)\|_{L^2(\alpha,\beta)} \leq \|u_0\|_{L^2(\alpha,\beta)} \quad \forall t > 0.$$

In the case of the more general problem (12.2), if we suppose that $\mu_0 > 0$, thanks to the Cauchy-Schwarz and Young inequalities we have

$$\int_{\alpha}^{\beta} f u \, dx \leq \|f\|_{L^2(\alpha,\beta)} \|u\|_{L^2(\alpha,\beta)} \leq \frac{\mu_0}{2} \|u\|_{L^2(\alpha,\beta)}^2 + \frac{1}{2\mu_0} \|f\|_{L^2(\alpha,\beta)}^2.$$

Integrating over time, we get the following a priori estimate

$$\begin{aligned} \|u(t)\|_{L^2(\alpha,\beta)}^2 &+ \mu_0 \int_0^t \|u(s)\|_{L^2(\alpha,\beta)}^2 \, ds + a(\beta) \int_0^t u^2(\beta, s) \, ds \\ &\leq \|u_0\|_{L^2(\alpha,\beta)}^2 + a(\alpha) \int_0^t \varphi^2(s) \, ds + \frac{1}{\mu_0} \int_0^t \|f\|_{L^2(\alpha,\beta)}^2 \, ds. \end{aligned}$$

An alternative estimate that does not require differentiability of $a(x)$ but uses the hypothesis that $a_0 \leq a(x) \leq a_1$ for two suitable positive constants a_0 and a_1 can be obtained by multiplying the equation by a^{-1} ,

$$a^{-1} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = a^{-1} f.$$

By now multiplying by u and integrating between α and β we obtain, after a few simple steps,

$$\frac{1}{2} \frac{d}{dt} \int_{\alpha}^{\beta} a^{-1}(x) u^2(x, t) \, dx + \frac{1}{2} u^2(\beta, t) = \int_{\alpha}^{\beta} a^{-1}(x) f(x, t) u(x, t) \, dx + \frac{1}{2} \varphi^2(t).$$

If $f = 0$ we immediately obtain

$$\|u(t)\|_a^2 + \int_0^t u^2(\beta, s) \, ds = \|u_0\|_a^2 + \int_0^t \varphi^2(s) \, ds, \quad t > 0.$$

We have defined

$$\|v\|_a = \left(\int_{\alpha}^{\beta} a^{-1}(x) v^2(x) \, dx \right)^{\frac{1}{2}}.$$

Thanks to the lower and upper bound of a^{-1} , the latter is an equivalent norm to that of $L^2(\alpha, \beta)$. On the other hand, if $f \neq 0$, we can proceed as follows

$$\|u(t)\|_a^2 + \int_0^t u^2(\beta, s) \, ds \leq \|u_0\|_a^2 + \int_0^t \varphi^2(s) \, ds + \int_0^t \|f\|_a^2 \, ds + \int_0^t \|u(s)\|_a^2 \, ds,$$

having used the Cauchy-Schwarz inequality.

By now applying the Gronwall lemma (see Lemma 2.2) we obtain, for each $t > 0$,

$$\|u(t)\|_a^2 + \int_0^t u^2(\beta, s) ds \leq e^t \left(\|u_0\|_a^2 + \int_0^t \varphi^2(s) ds + \int_0^t \|f\|_a^2 ds \right). \quad (12.4)$$

12.2 Systems of linear hyperbolic equations

Let us consider a linear system of the form

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + A \frac{\partial \mathbf{u}}{\partial x} = \mathbf{0}, & x \in \mathbb{R}, t > 0, \\ \mathbf{u}(0, x) = \mathbf{u}_0(x), & x \in \mathbb{R}, \end{cases} \quad (12.5)$$

where $\mathbf{u} : [0, \infty) \times \mathbb{R} \rightarrow \mathbb{R}^p$, $A : \mathbb{R} \rightarrow \mathbb{R}^{p \times p}$ is a given matrix, and $\mathbf{u}_0 : \mathbb{R} \rightarrow \mathbb{R}^p$ is the initial datum.

Let us first consider the case where the coefficients of A are constant (i.e. independent of both x and t). The system (12.5) is called *hyperbolic* if A can be diagonalized and has real eigenvalues. In such case, there exists a non-singular matrix $T : \mathbb{R} \rightarrow \mathbb{R}^{p \times p}$ such that

$$A = T \Lambda T^{-1},$$

where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_p)$, with $\lambda_i \in \mathbb{R}$ for $i = 1, \dots, p$, is the diagonal matrix of the eigenvalues of A while $T = [\boldsymbol{\omega}^1, \boldsymbol{\omega}^2, \dots, \boldsymbol{\omega}^p]$ is the matrix whose column vectors are the right eigenvectors of A , that is

$$A \boldsymbol{\omega}^k = \lambda_k \boldsymbol{\omega}^k, \quad k = 1, \dots, p.$$

Through this similarity transformation, it is possible to rewrite the system (12.5) in the form

$$\frac{\partial \mathbf{w}}{\partial t} + \Lambda \frac{\partial \mathbf{w}}{\partial x} = \mathbf{0}, \quad (12.6)$$

where $\mathbf{w} = T^{-1}\mathbf{u}$ are called *characteristic variables*. In this way, we obtain p independent equations of the form

$$\frac{\partial w_k}{\partial t} + \lambda_k \frac{\partial w_k}{\partial x} = 0, \quad k = 1, \dots, p,$$

analogous in all to the one of problem (12.1) (provided that we suppose a_0 and f null). The solution w_k is therefore constant along each *characteristic curve* $x = x(t)$, solution of the Cauchy problem

$$\begin{cases} \frac{dx}{dt} = \lambda_k, & t > 0, \\ x(0) = x_0. \end{cases} \quad (12.7)$$

Since the λ_k are constant, the characteristic curves are in fact the lines $x(t) = x_0 + \lambda_k t$ and the solutions feature the form $w_k(x, t) = \psi_k(x - \lambda_k t)$, where ψ_k is a function of a single variable, determined by the initial conditions. In the case of problem (12.5), we have that $\psi_k(x) = w_k(x, 0)$, thus the solution $\mathbf{u} = \mathbf{T}\mathbf{w}$ will be of the form

$$\mathbf{u}(x, t) = \sum_{k=1}^p w_k(x - \lambda_k t, 0) \boldsymbol{\omega}^k.$$

As we can see, the latter is composed by p travelling, non-interacting waves.

As in a strictly hyperbolic system p different characteristic lines exit each point (\bar{x}, \bar{t}) of the plane (x, t) , $u(\bar{x}, \bar{t})$ will only depend on the initial datum at the points $\bar{x} - \lambda_k \bar{t}$, for $k = 1, \dots, p$. For this reason, the set of the p points that form the feet of the characteristics outgoing point (\bar{x}, \bar{t}) , that is

$$D(\bar{x}, \bar{t}) = \{x \in \mathbb{R} \mid x = \bar{x} - \lambda_k \bar{t}, k = 1, \dots, p\}, \quad (12.8)$$

is called *domain of dependence* at point (\bar{x}, \bar{t}) of the solution \mathbf{u} .

In case we consider a bounded interval (α, β) instead of the whole real line, the sign of λ_k , $k = 1, \dots, p$, denotes the inflow point for each of the characteristic variables. The ψ_k function in the case of a problem set on a bounded interval will be determined not only by the initial conditions, but also by the boundary conditions provided to the inflow of each characteristic variable. Having considered a point (\bar{x}, \bar{t}) with $\bar{x} \in (\alpha, \beta)$ and $\bar{t} > 0$, if $\bar{x} - \lambda_k \bar{t} \in (\alpha, \beta)$ then $w_k(\bar{x}, \bar{t})$ is determined by the initial condition, in particular we have $w_k(\bar{x}, \bar{t}) = w_k(\bar{x} - \lambda_k \bar{t}, 0)$. Conversely, if $\bar{x} - \lambda_k \bar{t} \notin (\alpha, \beta)$ then the value of $w_k(\bar{x}, \bar{t})$ will depend on the boundary condition (see Fig. 12.2):

$$\begin{aligned} \text{if } \lambda_k > 0, \quad w_k(\bar{x}, \bar{t}) &= w_k(\alpha, \frac{\bar{x} - \alpha}{\lambda_k}), \\ \text{if } \lambda_k < 0, \quad w_k(\bar{x}, \bar{t}) &= w_k(\beta, \frac{\bar{x} - \beta}{\lambda_k}). \end{aligned}$$

As a consequence, the number of positive eigenvalues determines the number of boundary conditions to be assigned at $x = \alpha$, while at $x = \beta$ we will need to assign as many conditions as is the number of negative eigenvalues.

In the case where the coefficients of the matrix A in (12.5) are functions of x and t , we denote respectively by

$$L = \begin{bmatrix} I_1^T \\ \vdots \\ I_p^T \end{bmatrix} \quad \text{and} \quad R = [r_1 \dots r_p],$$

the matrices containing the left resp. right eigenvectors of A , whose elements satisfy the relations

$$Ar_k = \lambda_k r_k, \quad I_k^T A = \lambda_k I_k^T,$$

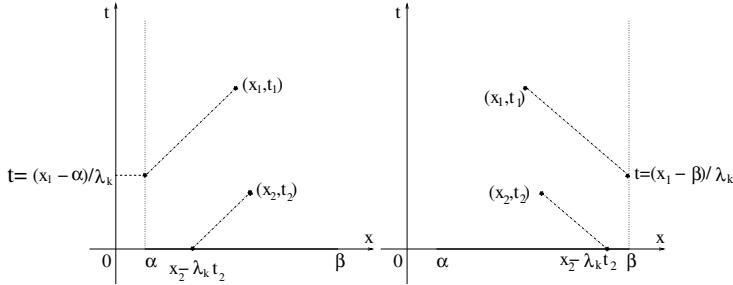


Fig. 12.2. The value of w_k at a point in the plane (x, t) depends either on the boundary condition or on the initial condition, based on the value of $x - \lambda_k t$. Both the positive (right) and negative (left) λ_k cases are reported

that is

$$AR = RA, \quad LA = AL.$$

Without loss of generality, we can suppose that $LR = I$. Let us now suppose that there exists a vector function \mathbf{w} satisfying the relations

$$\frac{\partial \mathbf{w}}{\partial \mathbf{u}} = R^{-1}, \quad \text{that is} \quad \frac{\partial \mathbf{u}_k}{\partial \mathbf{w}} = \mathbf{r}_k, \quad k = 1, \dots, p.$$

Proceeding as we did initially, we obtain

$$R^{-1} \frac{\partial \mathbf{u}}{\partial t} + A R^{-1} \frac{\partial \mathbf{u}}{\partial x} = 0$$

hence the new diagonal system (12.6). By reintroducing the characteristic curves (12.7) (the latter will no longer be straight lines as the eigenvalues λ_k vary for different values of x and t), \mathbf{w} is constant along them. The components of \mathbf{w} are therefore still called characteristic variables. As $R^{-1} = L$ (thanks to the normalization relation) we obtain

$$\frac{\partial w_k}{\partial \mathbf{u}} \cdot \mathbf{r}_m = \mathbf{l}_k \cdot \mathbf{r}_m = \delta_{km}, \quad k, m = 1, \dots, p.$$

The functions w_k , $k = 1, \dots, p$ are called *Riemann invariants* of the hyperbolic system.

12.2.1 The wave equation

Let us consider the following second order hyperbolic equation

$$\frac{\partial^2 u}{\partial t^2} - \gamma^2 \frac{\partial^2 u}{\partial x^2} = f, \quad x \in (\alpha, \beta), \quad t > 0. \quad (12.9)$$

Let

$$u(x, 0) = u_0(x) \quad \text{and} \quad \frac{\partial u}{\partial t}(x, 0) = v_0(x), \quad x \in (\alpha, \beta),$$

be the initial data and let us suppose, moreover, that u be identically null at the boundary

$$u(\alpha, t) = 0 \quad \text{and} \quad u(\beta, t) = 0, \quad t > 0. \quad (12.10)$$

In this case, u can represent the vertical displacement of a vibrating elastic chord with length $\beta - \alpha$, fixed at the endpoints, and γ is a coefficient that depends on the specific mass of the chord and on its tension. The chord is subject to a vertical force whose density is f . The functions $u_0(x)$ and $v_0(x)$ describe the initial displacement resp. velocity of the chord.

For simplicity of notation, we denote by u_t the derivative $\frac{\partial u}{\partial t}$, by u_x the derivative $\frac{\partial u}{\partial x}$ and we use similar notations for the second derivatives.

Let us now suppose that f be null. From equation (12.9) we can deduce that in this case, the kinetic energy of the system is preserved, that is (see Exercise 1)

$$\|u_t(t)\|_{L^2(\alpha, \beta)}^2 + \gamma^2 \|u_x(t)\|_{L^2(\alpha, \beta)}^2 = \|v_0\|_{L^2(\alpha, \beta)}^2 + \gamma^2 \|u_{0x}\|_{L^2(\alpha, \beta)}^2. \quad (12.11)$$

With the change of variables

$$\omega_1 = u_x, \quad \omega_2 = u_t,$$

the wave equation (12.9) becomes the following first-order system

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + A \frac{\partial \boldsymbol{\omega}}{\partial x} = \mathbf{f}, \quad x \in (\alpha, \beta), \quad t > 0, \quad (12.12)$$

where

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & -1 \\ -\gamma^2 & 0 \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} 0 \\ f \end{bmatrix},$$

whose initial conditions are $\omega_1(x, 0) = u'_0(x)$ and $\omega_2(x, 0) = v_0(x)$.

Since the eigenvalues of A are two distinct real numbers $\pm\gamma$ (representing the wave propagation rates), the system (12.12) is hyperbolic.

Note that, also in this case, to regular initial data correspond regular solutions, while discontinuities in the initial data will propagate along the characteristic lines $\frac{dx}{dt} = \pm\gamma$.

12.3 The finite difference method

Out of simplicity, we will now consider the case of problem (12.1). To numerically solve the latter, we can use spatio-temporal discretizations based on the finite difference method. In this case, the half-plane $\{t > 0\}$ is discretized choosing a temporal step Δt , a spatial discretization step h and defining the gridpoints (x_j, t^n) in the following way

$$x_j = jh, \quad j \in \mathbb{Z}, \quad t^n = n\Delta t, \quad n \in \mathbb{N}.$$

Let

$$\lambda = \Delta t/h,$$

and let us define

$$x_{j+1/2} = x_j + h/2.$$

We seek discrete solutions u_j^n which approximate $u(x_j, t^n)$ for each j and n .

The hyperbolic initial value problems are often discretized in time using explicit methods. Of course, this imposes restrictions on the values of λ that implicit methods generally don't have. For instance, let us consider problem (12.1). Any explicit finite difference method can be written in the form

$$u_j^{n+1} = u_j^n - \lambda(H_{j+1/2}^n - H_{j-1/2}^n), \quad (12.13)$$

where $H_{j+1/2}^n = H(u_j^n, u_{j+1}^n)$ for a suitable function $H(\cdot, \cdot)$ called *numerical flux*.

The numerical scheme (12.13) is basically the outcome of the following consideration. Suppose that a is constant and let us write equation (12.1) in conservation form

$$\frac{\partial u}{\partial t} + \frac{\partial(au)}{\partial x} = 0,$$

au being the *flux* associated to the equation. By integrating in space, we obtain

$$\int_{x_{j-1/2}}^{x_{j+1/2}} \frac{\partial u}{\partial t} dx + [au]_{x_{j-1/2}}^{x_{j+1/2}} = 0, \quad j \in \mathbb{Z},$$

that is

$$\frac{\partial}{\partial t} U_j + \frac{(au)(x_{j+\frac{1}{2}}) - (au)(x_{j-\frac{1}{2}})}{h} = 0, \quad \text{where } U_j = h^{-1} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x) dx.$$

Equation (12.13) can now be interpreted as an approximation where the temporal derivative is discretized using the forward Euler finite difference scheme, U_j is replaced by u_j and $H_{j+1/2}$ is a suitable approximation of $(au)(x_{j+\frac{1}{2}})$.

12.3.1 Discretization of the scalar equation

In the context of explicit methods, the numerical methods are distinguished based on how the numerical flux H is chosen. In particular, we cite the following methods:

- **forward/centered Euler (FE/C)**

$$u_j^{n+1} = u_j^n - \frac{\lambda}{2} a(u_{j+1}^n - u_{j-1}^n), \quad (12.14)$$

that takes the form (12.13) provided we define

$$H_{j+1/2} = \frac{1}{2}a(u_{j+1} + u_j). \quad (12.15)$$

– **Lax-Friedrichs (LF)**

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{\lambda}{2}a(u_{j+1}^n - u_{j-1}^n), \quad (12.16)$$

also of the form (12.13) with

$$H_{j+1/2} = \frac{1}{2}[a(u_{j+1} + u_j) - \lambda^{-1}(u_{j+1} - u_j)]. \quad (12.17)$$

– **Lax-Wendroff (LW)**

$$u_j^{n+1} = u_j^n - \frac{\lambda}{2}a(u_{j+1}^n - u_{j-1}^n) + \frac{\lambda^2}{2}a^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n), \quad (12.18)$$

that can be rewritten in the form (12.13) provided that we take

$$H_{j+1/2} = \frac{1}{2}[a(u_{j+1} + u_j) - \lambda a^2(u_{j+1} - u_j)]. \quad (12.19)$$

– **Upwind (or forward/decentered Euler) (U)**

$$u_j^{n+1} = u_j^n - \frac{\lambda}{2}a(u_{j+1}^n - u_{j-1}^n) + \frac{\lambda}{2}|a|(u_{j+1}^n - 2u_j^n + u_{j-1}^n), \quad (12.20)$$

corresponding to the form (12.13) provided that we choose

$$H_{j+1/2} = \frac{1}{2}[a(u_{j+1} + u_j) - |a|(u_{j+1} - u_j)]. \quad (12.21)$$

The LF method represents a modification of the FE/C method consisting in replacing the nodal value u_j^n in (12.14) with the average of the previous nodal value u_{j-1}^n and of the following one, u_{j+1}^n .

The LW method can be derived by first applying the Taylor development with respect to the temporal variable

$$u^{n+1} = u^n + (\partial_t u)^n \Delta t + (\partial_{tt} u)^n \frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3),$$

where $(\partial_t u)^n$ denotes the partial derivative of u at time t^n . Then, using equation (12.1), we replace $\partial_t u$ by $-a\partial_x u$, and $\partial_{tt} u$ by $a^2\partial_{xx} u$. Neglecting the remainder $\mathcal{O}(\Delta t^3)$ and approximating the spatial derivatives with centered finite differences, we get to (12.18). Finally, the U method is obtained by discretizing the convective term $a\partial_x u$ of the equation with the upwind finite difference, as seen in Chap. 11, Sec. 11.6.

All of the previously introduced schemes are explicit. An example of implicit method is the following:

– Backward/centered Euler (BE/C)

$$u_j^{n+1} + \frac{\lambda}{2}a(u_{j+1}^{n+1} - u_{j-1}^{n+1}) = u_j^n. \quad (12.22)$$

Naturally, the implicit schemes can also be rewritten in a general form that is similar to (12.13) where H^n is replaced by H^{n+1} . In the specific case, the numerical flux will again be defined by (12.15).

The advantage of the (12.13) formulation is that it can easily be extended to the case of more general hyperbolic problems.

In particular, we will examine the case of linear systems in Chap. 12.3.2. The extension to the case of non-linear hyperbolic equations will instead be considered in Sec. 14.2. Finally, we point out the following schemes for approximating the wave equation (12.9), again in the $f = 0$ case:

– Leap-Frog

$$u_j^{n+1} - 2u_j^n + u_j^{n-1} = (\gamma\lambda)^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n). \quad (12.23)$$

– Newmark

$$u_j^{n+1} - 2u_j^n + u_j^{n-1} = \frac{(\gamma\lambda)^2}{4}(w_j^{n-1} + 2w_j^n + w_j^{n+1}), \quad (12.24)$$

where $w_j^n = u_{j+1}^n - 2u_j^n + u_{j-1}^n$.

12.3.2 Discretization of linear hyperbolic systems

Let us consider the linear system (12.5). Generalizing (12.13), a numerical scheme for a finite difference approximation can be written in the form

$$\mathbf{u}_j^{n+1} = \mathbf{u}_j^n - \lambda(\mathbf{H}_{j+1/2}^n - \mathbf{H}_{j-1/2}^n),$$

where \mathbf{u}_j^n is the vector approximating $\mathbf{u}(x_j, t^n)$. Now, $\mathbf{H}_{j+1/2}$ is a *vector numerical flux*. Its formal expression can easily be derived by generalizing with respect to the scalar case and replacing in (12.15), (12.17), (12.19), (12.21), a , a^2 , and $|a|$ respectively with \mathbf{A} , \mathbf{A}^2 , and $|\mathbf{A}|$, being

$$|\mathbf{A}| = \mathbf{T}|\Lambda|\mathbf{T}^{-1},$$

where $|\Lambda| = \text{diag}(|\lambda_1|, \dots, |\lambda_p|)$ and \mathbf{T} is the matrix of eigenvectors of \mathbf{A} .

For instance, transforming system (12.5) in p independent transport equations and approximating each of these with an upwind scheme for scalar equations, we obtain the following upwind numerical scheme for the initial system

$$\mathbf{u}_j^{n+1} = \mathbf{u}_j^n - \frac{\lambda}{2}\mathbf{A}(\mathbf{u}_{j+1}^n - \mathbf{u}_{j-1}^n) + \frac{\lambda}{2}|\mathbf{A}|(\mathbf{u}_{j+1}^n - 2\mathbf{u}_j^n + \mathbf{u}_{j-1}^n).$$

The numerical flux of such scheme is

$$\mathbf{H}_{j+\frac{1}{2}} = \frac{1}{2}[\mathbf{A}(\mathbf{u}_{j+1} + \mathbf{u}_j) - |\mathbf{A}|(\mathbf{u}_{j+1} - \mathbf{u}_j)].$$

The Lax-Wendroff method becomes

$$\mathbf{u}_j^{n+1} = \mathbf{u}_j^n - \frac{1}{2}\lambda A(\mathbf{u}_{j+1}^n - \mathbf{u}_{j-1}^n) + \frac{1}{2}\lambda^2 A^2(\mathbf{u}_{j+1}^n - 2\mathbf{u}_j^n + \mathbf{u}_{j-1}^n),$$

and its numerical flux is

$$\mathbf{H}_{j+\frac{1}{2}} = \frac{1}{2}[A(\mathbf{u}_{j+1} - \mathbf{u}_j) - \lambda A^2(\mathbf{u}_{j+1} - \mathbf{u}_j)].$$

12.3.3 Boundary treatment

In case we want to discretize the hyperbolic equation (12.3) on a bounded interval, we will obviously need to use the inflow node $x = \alpha$ to impose the boundary condition, say $u_0^{n+1} = \varphi(t^{n+1})$, while in all the other nodes x_j , $1 \leq j \leq m$ (including the outflow node $x_m = \beta$) we will write the finite difference scheme.

However, we must observe that the schemes using a centered discretization of the space derivative require a particular treatment at x_m . Indeed, they would require using the value u_{m+1} , the latter being unavailable as it relates to the point with coordinates $\beta + h$ which lies outside the integration interval. The problem can be solved in various ways. An option is to only use the upwind decentered discretization on the last node, as such discretization does not require knowing the datum in x_{m+1} ; this approach however is only a first-order one. Alternatively, the value u_m^{n+1} can be obtained through an extrapolation from the values available at the internal nodes. An example could be an extrapolation along the characteristic lines applied to a scheme for which $\lambda a \leq 1$; this provides $u_m^{n+1} = u_{m-1}^n \lambda a + u_m^n(1 - \lambda a)$.

A further option consists in applying the centered finite difference scheme to the outflow node x_m as well and use, in place of u_{m+1}^n , an approximation based on a constant extrapolation ($u_{m+1}^n = u_m^n$), or on a linear one ($u_{m+1}^n = 2u_m^n - u_{m-1}^n$).

This matter becomes more problematic in the case of hyperbolic systems, where we must resort to compatibility equations. To gain a more in-depth view of these aspects and to analyze their possible instabilities deriving from the numerical boundary treatment, the reader can refer to Strickwerda [Str89], [QV94, Chap. 14] and [LeV07].

12.4 Analysis of the finite difference methods

We analyze the consistency, stability and convergence properties of the finite difference methods we introduced previously.

12.4.1 Consistency and convergence

For a given numerical scheme, the local truncation error is the error generated by expecting the exact solution to verify the numerical scheme itself.

For instance, in the case of scheme (12.14), having denoted by u the solution of the exact problem (12.1), we can define the truncation error at point (x_j, t^n) as follows

$$\tau_j^n = \frac{u(x_j, t^{n+1}) - u(x_j, t^n)}{\Delta t} + a \frac{u(x_{j+1}, t^n) - u(x_{j-1}, t^n)}{2h}.$$

If the *truncation error*

$$\tau(\Delta t, h) = \max_{j,n} |\tau_j^n|$$

tends to zero when Δt and h tend to zero, independently, then the numerical scheme will be said to be *consistent*.

Moreover, we will say that a numerical scheme is *accurate to the order p in time* and *to the order q in space* (for suitable integers p and q), if for a sufficiently regular solution of the exact problem, we have

$$\tau(\Delta t, h) = \mathcal{O}(\Delta t^p + h^q).$$

Using the Taylor developments suitably, we can then see that the truncation error of the previously introduced methods behaves as follows:

- **Euler (forward or backward) / centered:** $\mathcal{O}(\Delta t + h^2)$;
- **Upwind:** $\mathcal{O}(\Delta t + h)$;
- **Lax-Friedrichs :** $\mathcal{O}(\frac{h^2}{\Delta t} + \Delta t + h^2)$;
- **Lax-Wendroff :** $\mathcal{O}(\Delta t^2 + h^2 + h^2 \Delta t)$.

Finally, we will say that a scheme is *convergent* (in the maximum norm) if

$$\lim_{\Delta t, h \rightarrow 0} (\max_{j,n} |u(x_j, t^n) - u_j^n|) = 0.$$

Obviously, we can also consider weaker norms, such as $\|\cdot\|_{\Delta,1}$ and $\|\cdot\|_{\Delta,2}$ which we will introduce in (12.26).

12.4.2 Stability

We will say that a numerical method for a linear hyperbolic problem is *stable* if for each time T , there exists a constant $C_T > 0$ (possibly dependent on T) such that for each $h > 0$, there exists $\delta_0 > 0$ (possibly dependent on h) s.t. for each $0 < \Delta t < \delta_0$ we have

$$\|\mathbf{u}^n\|_{\Delta} \leq C_T \|\mathbf{u}^0\|_{\Delta}, \quad (12.25)$$

for each n such that $n\Delta t \leq T$, and for each initial datum \mathbf{u}_0 . Note that C_T should not depend on Δt and h . Often (always, in the case of explicit methods) we will have stability only if the temporal step is sufficiently small with respect to the spatial one, that is for $\delta_0 = \delta_0(h)$.

The notation $\|\cdot\|_{\Delta}$ denotes a suitable discrete norm, for instance

$$\|\mathbf{v}\|_{\Delta,p} = \left(h \sum_{j=-\infty}^{\infty} |v_j|^p \right)^{\frac{1}{p}} \quad \text{for } p = 1, 2, \quad \|\mathbf{v}\|_{\Delta,\infty} = \sup_j |v_j|. \quad (12.26)$$

Note how $\|\mathbf{v}\|_{\Delta,p}$ represents an approximation of the $L^p(\mathbb{R})$ norm, for $p = 1, 2$ or $+\infty$.

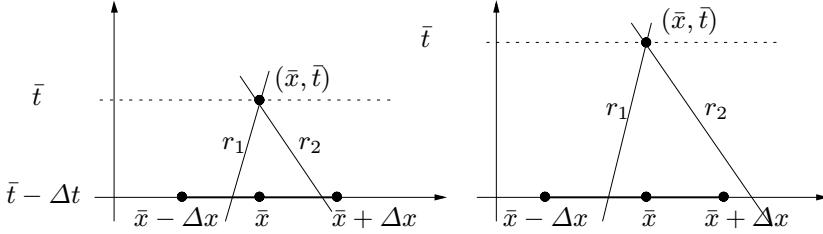


Fig. 12.3. Geometric interpretation of the CFL condition for a system with $p = 2$, where $r_i = \bar{x} - \lambda_i(t - \bar{t})$ $i = 1, 2$. The CFL condition is satisfied in the left case, and violated in the right case

The implicit backward/centered Euler scheme (12.22) is stable in the $\|\cdot\|_{\Delta,2}$ norm for any choice of the Δt and h parameters (see Exercise 2).

A scheme is called *strongly stable* with respect to the $\|\cdot\|_\Delta$ norm if

$$\|\mathbf{u}^n\|_\Delta \leq \|\mathbf{u}^{n-1}\|_\Delta, \quad (12.27)$$

for each n such that $n\Delta t \leq T$, and for each initial datum \mathbf{u}_0 , which implies that (12.25) is verified with $C_T = 1$.

Remark 12.1 In the context of hyperbolic problems, solutions for long time intervals (i.e. for $T \gg 1$) are often sought. Such cases usually require a strongly stable scheme, as this guarantees that the numerical solution is limited for each value of T . •

As we will see, a necessary condition for the stability of an explicit numerical scheme of the form (12.13) is that the temporal and spatial discretization steps be linked by the following relation

$$|a\lambda| \leq 1, \text{ or } \Delta t \leq \frac{h}{|a|} \quad (12.28)$$

called *CFL condition* (from Courant, Friedrichs and Lewy). The number $a\lambda$ is commonly called *CFL number*; this is an a-dimensional quantity (a being a velocity).

The geometrical interpretation of the CFL stability condition is the following. In a finite difference scheme, the value of u_j^{n+1} generally depends on the values u_{j+i}^n of u^n at the three points x_{j+i} , $i = -1, 0, 1$. Proceeding backwards, we deduce that the solution u_j^{n+1} will only depend on the initial data at the points x_{j+i} , for $i = -(n+1), \dots, (n+1)$ (see Fig. 12.3).

Denoting by *numerical domain of dependence* $D_{\Delta t}(x_j, t^n)$ the domain of dependency of u_j^n , which will therefore be called numerical dependency domain of u_j^n , the former will verify

$$D_{\Delta t}(x_j, t^n) \subset \{x \in \mathbb{R} : |x - x_j| \leq nh = \frac{t^n}{\lambda}\}.$$

Consequently, for each given point (\bar{x}, \bar{t}) we have

$$D_{\Delta t}(\bar{x}, \bar{t}) \subset \{x \in \mathbb{R} : |x - \bar{x}| \leq \frac{\bar{t}}{\lambda}\}.$$

In particular, taking the limit for $\Delta t \rightarrow 0$, and fixing λ , the numerical dependency domain becomes

$$D_0(\bar{x}, \bar{t}) = \{x \in \mathbb{R} : |x - \bar{x}| \leq \frac{\bar{t}}{\lambda}\}.$$

The condition (12.28) is then equivalent to the inclusion

$$D(\bar{x}, \bar{t}) \subset D_0(\bar{x}, \bar{t}), \quad (12.29)$$

where $D(\bar{x}, \bar{t})$ is the dependency domain of the exact solution defined in (12.8). Note that in the scalar case, $p = 1$ and $\lambda_1 = a$.

Remark 12.2 The CFL condition establishes, in particular, that there exist no explicit finite difference schemes, unconditionally stable and consistent for hyperbolic initial value problems. Indeed, if the CFL condition were violated, there would exist at least a point x^* in the dependency domain that does not belong to the numerical dependency domain. Then, changing the initial datum to x^* will only modify the exact solution and not the numerical one. This implies a non-convergence of the method and therefore also its instability. Indeed, for a consistent method, the Lax-Richtmyer equivalence theorem states that stability is a necessary and sufficient condition for its convergence. •

Remark 12.3 In the case where $a = a(x, t)$ is no longer constant in (12.1), the CFL condition becomes

$$\Delta t \leq \frac{h}{\sup_{x \in \mathbb{R}, t > 0} |a(x, t)|},$$

and even if the spatial discretization step varies, we have

$$\Delta t \leq \min_k \frac{h_k}{\sup_{x \in (x_k, x_{k+1}), t > 0} |a(x, t)|},$$

as $h_k = x_{k+1} - x_k$. •

Referring to the hyperbolic system (12.5), the stability condition CFL, analogous in all to (12.28), will be

$$\left| \lambda_k \frac{\Delta t}{h} \right| \leq 1, \quad k = 1, \dots, p, \quad \text{or, equivalently,} \quad \Delta t \leq \frac{h}{\max_k |\lambda_k|},$$

where $\{\lambda_k, k = 1 \dots, p\}$ are the eigenvalues of A.

This condition as well can be written in the form (12.29). The latter expresses the requirement that each line of the form $x = \bar{x} - \lambda_k(\bar{t} - t)$, $k = 1, \dots, p$, must intersect the horizontal line $t = \bar{t} - \Delta t$ at points $x^{(k)}$ which lie within the numerical dependency domain.

Theorem 12.1 *If the CFL condition (12.28) is satisfied, the upwind, Lax-Friedrichs and Lax-Wendroff schemes are strongly stable in the norm $\|\cdot\|_{\Delta,1}$.*

Proof. To prove the stability of the upwind scheme (12.20) we rewrite it in the following form (having supposed $a > 0$)

$$u_j^{n+1} = u_j^n - \lambda a(u_j^n - u_{j-1}^n).$$

Then

$$\|\mathbf{u}^{n+1}\|_{\Delta,1} \leq h \sum_j |(1 - \lambda a)u_j^n| + h \sum_j |\lambda a u_{j-1}^n|.$$

Under the hypothesis (12.28) both values λa and $1 - \lambda a$ are non-negative. Hence,

$$\|\mathbf{u}^{n+1}\|_{\Delta,1} \leq h(1 - \lambda a) \sum_j |u_j^n| + h \lambda a \sum_j |u_{j-1}^n| = \|\mathbf{u}^n\|_{\Delta,1},$$

that is, inequality (12.25) holds with $C_T = 1$. The scheme is therefore strongly stable with respect to the norm $\|\cdot\|_{\Delta} = \|\cdot\|_{\Delta,1}$.

For the Lax-Friedrichs scheme, always under the CFL condition (12.28), we derive from (12.16) that

$$u_j^{n+1} = \frac{1}{2}(1 - \lambda a)u_{j+1}^n + \frac{1}{2}(1 + \lambda a)u_{j-1}^n,$$

so

$$\begin{aligned} \|\mathbf{u}^{n+1}\|_{\Delta,1} &\leq \frac{1}{2}h \left[\sum_j |(1 - \lambda a)u_{j+1}^n| + \sum_j |(1 + \lambda a)u_{j-1}^n| \right] \\ &\leq \frac{1}{2}(1 - \lambda a)\|\mathbf{u}^n\|_{\Delta,1} + \frac{1}{2}(1 + \lambda a)\|\mathbf{u}^n\|_{\Delta,1} = \|\mathbf{u}^n\|_{\Delta,1}. \end{aligned}$$

For the Lax-Wendroff scheme, the proof is analogous (see e.g. [QV94, Chap. 14] or [Str89]). \diamond

Finally, we can prove that, if the CFL condition is verified, the upwind scheme satisfies

$$\|\mathbf{u}^n\|_{\Delta,\infty} \leq \|\mathbf{u}^0\|_{\Delta,\infty} \quad \forall n \geq 0, \tag{12.30}$$

i.e. it is strongly stable in the $\|\cdot\|_{\Delta,\infty}$ norm. The relation (12.30) is called *discrete maximum principle* (see Exercise 4).

Theorem 12.2 *The backward Euler scheme BE/C is strongly stable in the norm $\|\cdot\|_{\Delta,2}$, with no restriction on Δt . The forward Euler scheme FE/C, instead, is never strongly stable. However, it is stable with constant $C_T = e^{T/2}$ provided that we assume that Δt satisfies the following condition (more restrictive than the CFL condition)*

$$\Delta t \leq \left(\frac{h}{a}\right)^2. \quad (12.31)$$

Proof. We observe that

$$(B - A)B = \frac{1}{2}(B^2 - A^2 + (B - A)^2) \quad \forall A, B \in \mathbb{R}. \quad (12.32)$$

As a matter of fact

$$(B - A)B = (B - A)^2 + (B - A)A = \frac{1}{2}((B - A)^2 + (B - A)(B + A)).$$

Multiplying (12.22) by u_j^{n+1} we find

$$(u_j^{n+1})^2 + (u_j^{n+1} - u_j^n)^2 = (u_j^n)^2 - \lambda a(u_{j+1}^{n+1} - u_{j-1}^{n+1})u_j^{n+1}.$$

Observing that

$$\sum_{j \in \mathbb{Z}} (u_{j+1}^{n+1} - u_{j-1}^{n+1})u_j^{n+1} = 0 \quad (12.33)$$

(being a telescopic sum), we immediately obtain that $\|\mathbf{u}^{n+1}\|_{\Delta,2}^2 \leq \|\mathbf{u}^n\|_{\Delta,2}^2$, which is the result sought for the BE/C scheme.

Let us now move to the FE/C scheme and multiply (12.14) by u_j^n . Observing that

$$(B - A)A = \frac{1}{2}(B^2 - A^2 - (B - A)^2) \quad \forall A, B \in \mathbb{R}, \quad (12.34)$$

we find

$$(u_j^{n+1})^2 = (u_j^n)^2 + (u_j^{n+1} - u_j^n)^2 - \lambda a(u_{j+1}^n - u_{j-1}^n)u_j^n.$$

On the other hand, we obtain once again from (12.14) that

$$u_j^{n+1} - u_j^n = -\frac{\lambda a}{2}(u_{j+1}^n - u_{j-1}^n),$$

and therefore

$$(u_j^{n+1})^2 = (u_j^n)^2 + \left(\frac{\lambda a}{2}\right)^2 (u_{j+1}^n - u_{j-1}^n)^2 - \lambda a(u_{j+1}^n - u_{j-1}^n)u_j^n.$$

Now summing on j and observing that the last addendum yields a telescopic sum (hence it does not provide any contribution) we obtain, after multiplying by h ,

$$\|\mathbf{u}^{n+1}\|_{\Delta,2}^2 = \|\mathbf{u}^n\|_{\Delta,2}^2 + \left(\frac{\lambda a}{2}\right)^2 h \sum_{j \in \mathbb{Z}} (u_{j+1}^n - u_{j-1}^n)^2,$$

from which we infer that there is no value of Δt for which the method is strongly stable. However, as

$$(u_{j+1}^n - u_{j-1}^n)^2 \leq 2 [(u_{j+1}^n)^2 + (u_{j-1}^n)^2],$$

we find that, under the hypothesis (12.31),

$$\|\mathbf{u}^{n+1}\|_{\Delta,2}^2 \leq (1 + \lambda^2 a^2) \|\mathbf{u}^n\|_{\Delta,2}^2 \leq (1 + \Delta t) \|\mathbf{u}^n\|_{\Delta,2}^2.$$

By recursion, we find

$$\|\mathbf{u}^n\|_{\Delta,2}^2 \leq (1 + \Delta t)^n \|\mathbf{u}^0\|_{\Delta,2}^2 \leq e^T \|\mathbf{u}^0\|_{\Delta,2}^2,$$

where we have used inequality

$$(1 + \Delta t)^n \leq e^{n \Delta t} \leq e^T \quad \forall n \text{ s.t. } t^n \leq T.$$

We conclude that

$$\|\mathbf{u}^n\|_{\Delta,2} \leq e^{T/2} \|\mathbf{u}^0\|_{\Delta,2},$$

which is the stability result sought for the FE/C scheme. \diamond

12.4.3 Von Neumann analysis and amplification coefficients

The stability of a scheme in the norm $\|\cdot\|_{\Delta,2}$ can be also studied by the von Neumann analysis. To this purpose, we hypothesize that the function $u_0(x)$ is 2π -periodic and thus it can be written as a Fourier series as follows

$$u_0(x) = \sum_{k=-\infty}^{\infty} \alpha_k e^{ikx}, \tag{12.35}$$

where

$$\alpha_k = \frac{1}{2\pi} \int_0^{2\pi} u_0(x) e^{-ikx} dx$$

is the k -th Fourier coefficient. Hence,

$$u_j^0 = u_0(x_j) = \sum_{k=-\infty}^{\infty} \alpha_k e^{ikjh}, \quad j = 0, \pm 1, \pm 2, \dots$$

Table 12.1. Amplification coefficient for the different numerical schemes in Sec. 12.3.1. We recall that $\lambda = \Delta t/h$

Scheme	γ_k
Forward/centered Euler	$1 - ia\lambda \sin(kh)$
Backward/centered Euler	$(1 + ia\lambda \sin(kh))^{-1}$
Upwind	$1 - a \lambda(1 - e^{-ikh})$
Lax-Friedrichs	$\cos kh - ia\lambda \sin(kh)$
Lax-Wendroff	$1 - ia\lambda \sin(kh) - a^2\lambda^2(1 - \cos(kh))$

It can be verified that applying any of the difference schemes seen in Sec. 12.3.1 we get the following relation

$$u_j^n = \sum_{k=-\infty}^{\infty} \alpha_k e^{ikjh} \gamma_k^n, \quad j = 0, \pm 1, \pm 2, \dots, \quad n \geq 1. \quad (12.36)$$

The number $\gamma_k \in \mathbb{C}$ is called *amplification coefficient* of the k -th frequency (or harmonic), and characterizes the scheme under exam. For instance, in the case of the forward centered Euler method (FE/C) we find

$$\begin{aligned} u_j^1 &= \sum_{k=-\infty}^{\infty} \alpha_k e^{ikjh} \left(1 - \frac{a\Delta t}{2h} (e^{ikh} - e^{-ikh}) \right) \\ &= \sum_{k=-\infty}^{\infty} \alpha_k e^{ikjh} \left(1 - \frac{a\Delta t}{h} i \sin(kh) \right). \end{aligned}$$

Hence,

$$\gamma_k = 1 - \frac{a\Delta t}{h} i \sin(kh) \quad \text{and thus} \quad |\gamma_k| = \left\{ 1 + \left(\frac{a\Delta t}{h} \sin(kh) \right)^2 \right\}^{\frac{1}{2}}.$$

As there exist values of k for which $|\gamma_k| > 1$, there is no value of Δt and h for which the scheme is strongly stable.

Proceeding in a similar way for the other schemes, we find the coefficients reported in Table 12.1.

We will now see how the von Neumann analysis can be applied to study the stability of a numerical scheme with respect to the $\|\cdot\|_{\Delta,2}$ norm and to ascertain its dissipation and dispersion characteristics.

To this purpose, we prove the following result:

Theorem 12.3 *If there exists a number $\beta \geq 0$, and a positive integer m such that, for suitable choices of Δt and h , we have $|\gamma_k| \leq (1 + \beta\Delta t)^{\frac{1}{m}}$ for each k , then the scheme is stable with respect to the norm $\|\cdot\|_{\Delta,2}$ with a stability constant $C_T = e^{\beta T/m}$. In particular, if we can take $\beta = 0$ (and therefore $|\gamma_k| \leq 1 \forall k$) then the scheme is strongly stable with respect to the same norm.*

Proof. We will suppose that problem (12.1) is formulated on the interval $[0, 2\pi]$. In such interval, let us consider $N + 1$ equidistant nodes,

$$x_j = jh, \quad j = 0, \dots, N, \quad \text{with} \quad h = \frac{2\pi}{N},$$

(N being an even positive integer) where to satisfy the numerical scheme (12.13). Moreover, we will suppose for simplicity that the initial datum u_0 be periodic. As the numerical scheme only depends on the values of u_0 at the x_j nodes, we can replace u_0 by the Fourier polynomial of order $N/2$,

$$\tilde{u}_0(x) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \alpha_k e^{ikx} \quad (12.37)$$

which interpolates it at the nodes. Note that \tilde{u}_0 is a periodic function with period 2π . We will have, thanks to (12.36),

$$u_j^0 = u_0(x_j) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \alpha_k e^{ikjh}, \quad u_j^n = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \alpha_k \gamma_k^n e^{ikjh}.$$

We note that

$$\|\mathbf{u}^n\|_{\Delta,2}^2 = h \sum_{j=0}^{N-1} \sum_{k,m=-\frac{N}{2}}^{\frac{N}{2}-1} \alpha_k \bar{\alpha}_m (\gamma_k \bar{\gamma}_m)^n e^{i(k-m)jh}.$$

As

$$h \sum_{j=0}^{N-1} e^{i(k-m)jh} = 2\pi \delta_{km}, \quad -\frac{N}{2} \leq k, m \leq \frac{N}{2} - 1,$$

(see e.g. [QSS07, Lemma 10.2]) we find

$$\|\mathbf{u}^n\|_{\Delta,2}^2 = 2\pi \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} |\alpha_k|^2 |\gamma_k|^{2n}.$$

Thanks to the assumption made on $|\gamma_k|$ we have

$$\|\mathbf{u}^n\|_{\Delta,2}^2 \leq (1 + \beta \Delta t)^{\frac{2n}{m}} 2\pi \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} |\alpha_k|^2 = (1 + \beta \Delta t)^{\frac{2n}{m}} \|\mathbf{u}^0\|_{\Delta,2}^2 \quad \forall n \geq 0.$$

As $1 + \beta \Delta t \leq e^{\beta \Delta t}$, we deduce that

$$\|\mathbf{u}^n\|_{\Delta,2} \leq e^{\frac{\beta \Delta t n}{m}} \|\mathbf{u}^0\|_{\Delta,2} = e^{\frac{\beta T}{m}} \|\mathbf{u}^0\|_{\Delta,2} \quad \forall n \text{ s.t. } n \Delta t \leq T.$$

This proves the theorem. \diamond

Remark 12.4 Should strong stability be required, the condition $|\gamma_k| \leq 1$ indicated in Theorem 12.3 is also necessary. •

In the case of the upwind scheme (12.20), as

$$|\gamma_k|^2 = [1 - |a|\lambda(1 - \cos kh)]^2 + a^2\lambda^2 \sin^2 kh, \quad k \in \mathbb{Z},$$

we obtain

$$|\gamma_k| \leq 1 \text{ if } \Delta t \leq \frac{h}{|a|}, \quad k \in \mathbb{Z}, \quad (12.38)$$

that is we find that the CFL condition guarantees the strong stability in the $\|\cdot\|_{\Delta,2}$ norm.

Proceeding in a similar way, we can verify that (12.38) also holds for the Lax-Friedrichs scheme.

The centered backward Euler scheme BE/C instead is unconditionally strongly stable in the $\|\cdot\|_{\Delta,2}$ norm, as $|\gamma_k| \leq 1$ for each k and for each possible choice of Δt and h , as we previously obtained in Theorem 12.2 by following a different procedure.

In the case of the centered forward Euler method FE/C we have

$$|\gamma_k|^2 = 1 + \frac{a^2 \Delta t^2}{h^2} \sin^2(kh) \leq 1 + \frac{a^2 \Delta t^2}{h^2}, \quad k \in \mathbb{Z}.$$

If $\beta > 0$ is a constant such that

$$\Delta t \leq \beta \frac{h^2}{a^2} \quad (12.39)$$

then $|\gamma_k| \leq (1 + \beta \Delta t)^{1/2}$. Hence, applying Theorem 12.3 (with $m = 2$) we deduce that the FE/C scheme is stable, albeit with a more restrictive CFL condition, as previously obtained following a different path in Theorem 12.2.

We can find a strong stability condition for the centered forward Euler method applied to the transport-reaction equation

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} + a_0 u = 0, \quad (12.40)$$

with $a_0 > 0$. In this case we have for each $k \in \mathbb{Z}$

$$|\gamma_k|^2 = 1 - 2a_0 \Delta t + \Delta t^2 a_0^2 + \lambda^2 \sin^2(kh) \leq 1 - 2a_0 \Delta t + \Delta t^2 a_0^2 + \left(\frac{a \Delta t}{h}\right)^2$$

and thus the scheme is strongly stable in the $\|\cdot\|_{\Delta,2}$ norm under the condition

$$\Delta t < \frac{2a_0}{a_0^2 + h^{-2}a^2}. \quad (12.41)$$

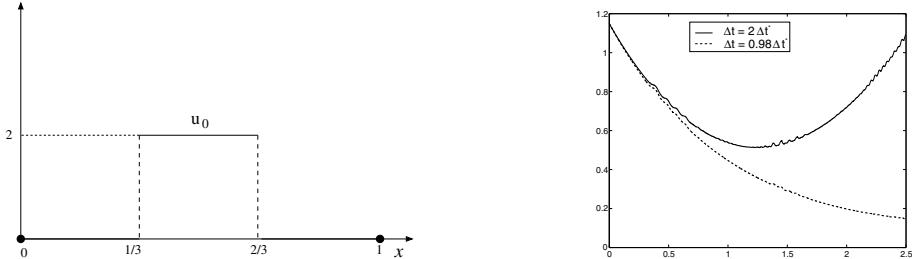


Fig. 12.4. The figure on the right displays the behavior of $\|u^n\|_{\Delta,2}$, where u^n is the solution of equation (12.40) (with $a = a_0 = 1$) obtained using the FE/C method, for two values of Δt , one smaller and one greater than the critical value Δt^* . On the left, the initial datum used

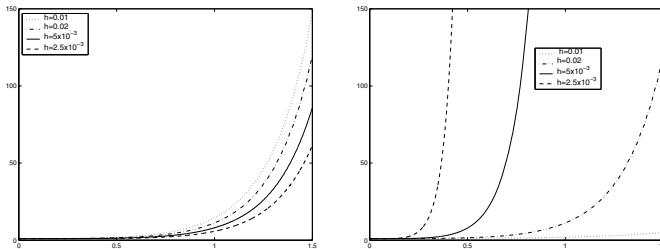


Fig. 12.5. Behavior of $\|u^n\|_{\Delta,2}$ where u^n is the solution obtained using the FE/C method, in the $a_0 = 0$ case and for different values of h . On the left, the case where Δt satisfies the stability condition (12.39). On the right, the results obtained by maintaining the CFL number constant and equal to 0.1, violating the condition (12.39)

Example 12.1 In order to numerically verify the stability condition (12.41), we have considered equation (12.40) in the $(0, 1)$ interval with periodic boundary conditions. We have chosen $a = a_0 = 1$ and the initial datum u_0 equal to 2 in the $(1/3, 2/3)$ interval and 0 elsewhere. As the initial datum is a square wave, its Fourier development has all its α_k coefficients not null. On the right of Fig. 12.4, we report $\|u^n\|_{\Delta,2}$ in the time interval $(0, 2.5)$ for two values of Δt , one larger and one smaller than the critical value $\Delta t^* = 2/(1 + h^{-2})$, provided by (12.41). Note that for $\Delta t < \Delta t^*$ the norm is decreasing, while, in the opposite case, after an initial decrease it grows exponentially. Fig. 12.5 shows the result for $a_0 = 0$ obtained with FE/C using the same initial datum. In the figure on the left, we display the behavior of $\|u^n\|_{\Delta,2}$ for different values of h and using $\Delta t = 10h^2$, that is varying the time step based on the restriction provided by inequality (12.39) and taking $\beta = 10$. Note how the norm of the solution remains bounded for decreasing values of h . At the right-hand side of the same figure, we illustrate the result obtained for the same values of h taking as condition $\Delta t = 0.1h$, which corresponds to a constant CFL number equal to 0.1. In this case, the discrete norm of the numerical solution blows up as h decreases, as expected. ■

12.4.4 Dissipation and dispersion

Besides allowing to enquire about the stability of a numerical scheme, the analysis of amplification coefficients is also useful to study its dissipation and dispersion properties.

To understand what this is about, let us consider the exact solution of the problem (12.1); for such solution, we have the following relation

$$u(x, t^n) = u_0(x - an\Delta t), \quad n \geq 0, \quad x \in \mathbb{R},$$

as $t^n = n\Delta t$. In particular, using (12.35) we obtain

$$u(x_j, t^n) = \sum_{k=-\infty}^{\infty} \alpha_k e^{ikjh} (g_k)^n \quad \text{with} \quad g_k = e^{-ia k \Delta t}. \quad (12.42)$$

Comparing (12.42) with (12.36) we can note that the amplification coefficient γ_k (generated by the specific numerical scheme) is the correspondent of g_k .

We observe that $|g_k| = 1$ for each $k \in \mathbb{Z}$, while it must be $|\gamma_k| \leq 1$ in order to guarantee the strong stability of the scheme. Thus, γ_k is a *dissipative* coefficient. The smaller $|\gamma_k|$ is, the larger will be the reduction of the amplitude α_k and, consequently, the larger will be the dissipation of the numerical scheme.

The ratio $\epsilon_a(k) = \frac{|\gamma_k|}{|g_k|}$ is called *amplification error* (or *dissipation error*) of the k -th harmonic associated to the numerical scheme (and in our case it coincides with the amplification coefficient).

Having set

$$\phi_k = kh,$$

as $k\Delta t = \lambda\phi_k$ we obtain

$$g_k = e^{-ia\lambda\phi_k}. \quad (12.43)$$

The real number ϕ_k , here expressed in radians, is called *phase angle* of the k -th harmonic. We rewrite γ_k in the following way

$$\gamma_k = |\gamma_k| e^{-i\omega\Delta t} = |\gamma_k| e^{-i\frac{\omega}{k}\lambda\phi_k},$$

and comparing such relation to (12.43), we can deduce that the ratio $\frac{\omega}{k}$ represents the *propagation rate* of the numerical scheme, relatively to the k -th harmonic. The ratio

$$\epsilon_d(k) = \frac{\omega}{ka} = \frac{\omega h}{\phi_k a}$$

between such rate and the propagation rate a of the exact solution is called *dispersion error* ϵ_d relative to the k -th harmonic.

The amplification (or dissipation) error and the dispersion error for the numerical schemes analyzed up to now vary as a function of the phase angle ϕ_k and of the CFL number $a\lambda$, as reported in Fig. 12.6. For symmetry reasons we have considered the

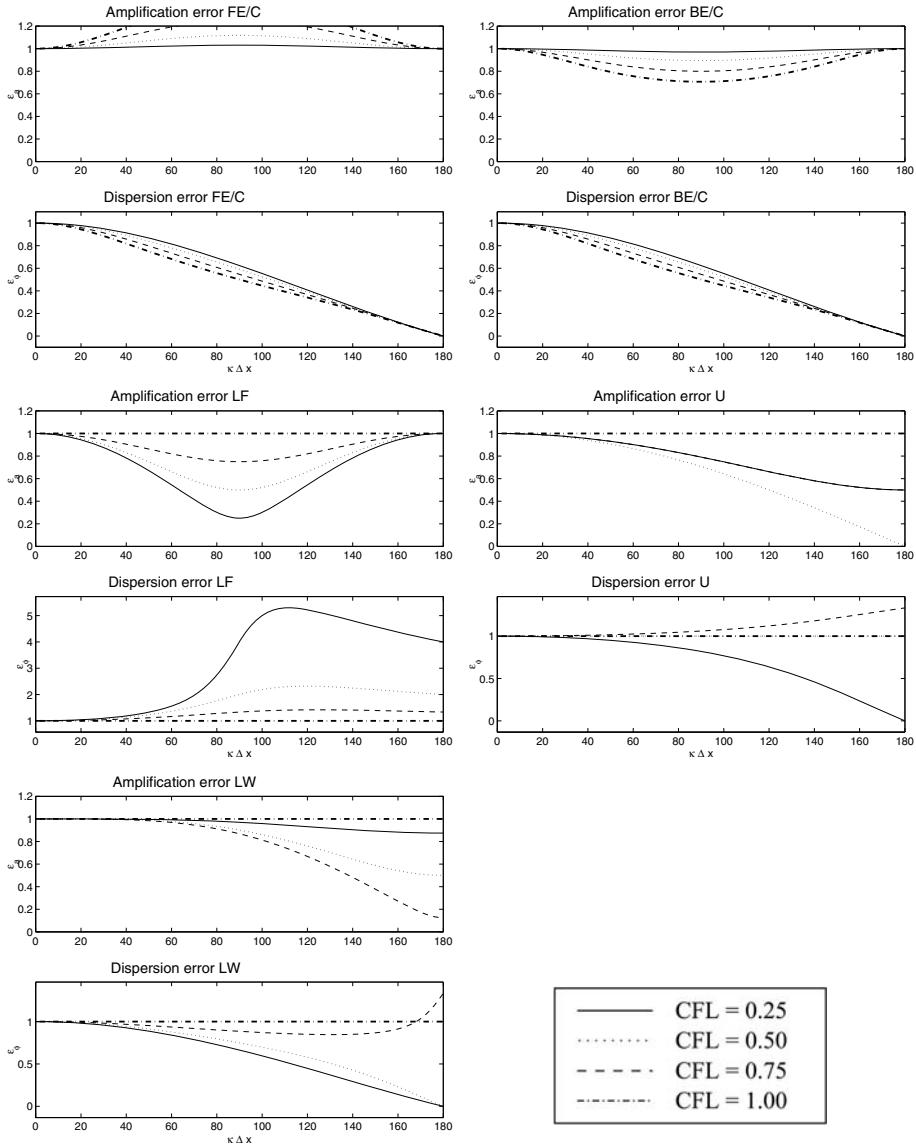


Fig. 12.6. Amplification and dispersion errors for different numerical schemes as a function of the phase angle $\phi_k = kh$ and for different values of the CFL number

interval $0 \leq \phi_k \leq \pi$ and we have used degrees instead of radians in the abscissa to indicate ϕ_k . Note how the forward/centered Euler scheme denotes a curve of the amplification factor with values above one for all the CFL schemes we have considered, in accordance with the fact that such scheme is never strongly stable.

Example 12.2 In Fig. 12.7 we compare the obtained numerical results by solving equation (12.1) with $a = 1$ and initial datum u_0 composed by a packet of two sinusoidal waves of equal length l centered at the origin ($x = 0$). In the figures on the left $l = 20h$, while in the right ones we have $l = 8h$. As $k = \frac{2\pi}{l}$, we have $\phi_k = \frac{2\pi}{l}h$ and therefore the values of the phase angle of the wave packet are $\phi_k = \pi/20$ at the left and $\phi_k = \pi/8$ at the right. The numerical solution has been computed for the value 0.75 of the CFL number, using the different (stable) schemes illustrated previously. We can note how the dissipative effect is very strong at high frequencies ($\phi_k = \pi/4$) and in particular for the first-order upwind, backward/centered Euler and Lax-Friedrichs methods.

In order to appreciate the dispersion effects, the solution for $\phi_k = \pi/4$ after 8 time steps is reported in Fig. 12.8. We can note how the Lax-Wendroff method is the least dissipative. Moreover, by attentively observing the position of the numerical wave crests with respect to those of the numerical solution, we can verify that the Lax-Friedrichs method features a positive dispersion error. Indeed, the numerical wave results to anticipate the exact one. The upwind method is also weakly dispersive for a CFL number equal to 0.75, while the dispersion of the Lax-Friedrichs and backward Euler methods is evident (even after only 8 time steps!). ■

12.5 Equivalent equations

To each numerical scheme, we can associate a family of differential equations, called equivalent equations.

12.5.1 The upwind scheme case

Let us first focus on the upwind scheme. Suppose there exists a regular function $v(x, t)$ satisfying the difference equation (12.20) at each point $(x, t) \in \mathbb{R} \times \mathbb{R}^+$ (and not only at the grid nodes (x_j, t^n) !). We can then write (in the case where $a > 0$)

$$\frac{v(x, t + \Delta t) - v(x, t)}{\Delta t} + a \frac{v(x, t) - v(x - h, t)}{h} = 0. \quad (12.44)$$

Using the Taylor developments with respect to x and t relative to the point (x, t) and supposing that v is of class C^4 with respect to x and t , we can write

$$\begin{aligned} \frac{v(x, t + \Delta t) - v(x, t)}{\Delta t} &= v_t + \frac{\Delta t}{2} v_{tt} + \frac{\Delta t^2}{6} v_{ttt} + \mathcal{O}(\Delta t^3), \\ a \frac{v(x, t) - v(x - h, t)}{h} &= av_x - \frac{ah}{2} v_{xx} + \frac{ah^2}{6} v_{xxx} + \mathcal{O}(h^3), \end{aligned}$$

where the right-hand side derivatives are all evaluated at point (x, t) . Thanks to (12.44) we deduce that, at each point (x, t) , the v function satisfies the relation

$$v_t + av_x = R^U + \mathcal{O}(\Delta t^3 + h^3), \quad (12.45)$$

with

$$R^U = \frac{1}{2}(ah v_{xx} - \Delta t v_{tt}) - \frac{1}{6}(ah^2 v_{xxx} + \Delta t^2 v_{ttt}).$$

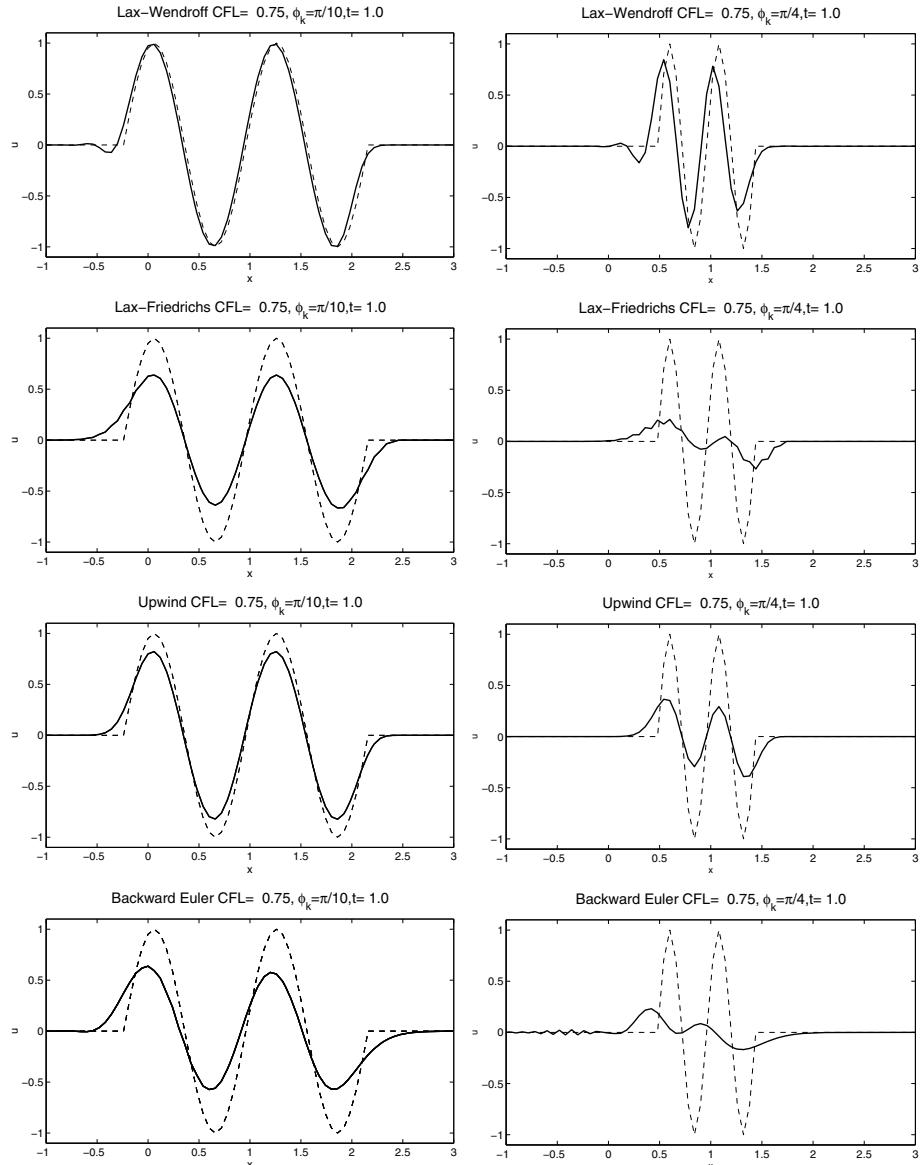


Fig. 12.7. Numerical solution of the convective transport equation of a sinusoidal wave packet with different wavelengths ($l = 20h$ at the left, $l = 8h$ at the right) obtained with different numerical schemes. The numerical solution for $t = 1$ is displayed in solid line, while the exact solution at the same time instant is displayed in etched line

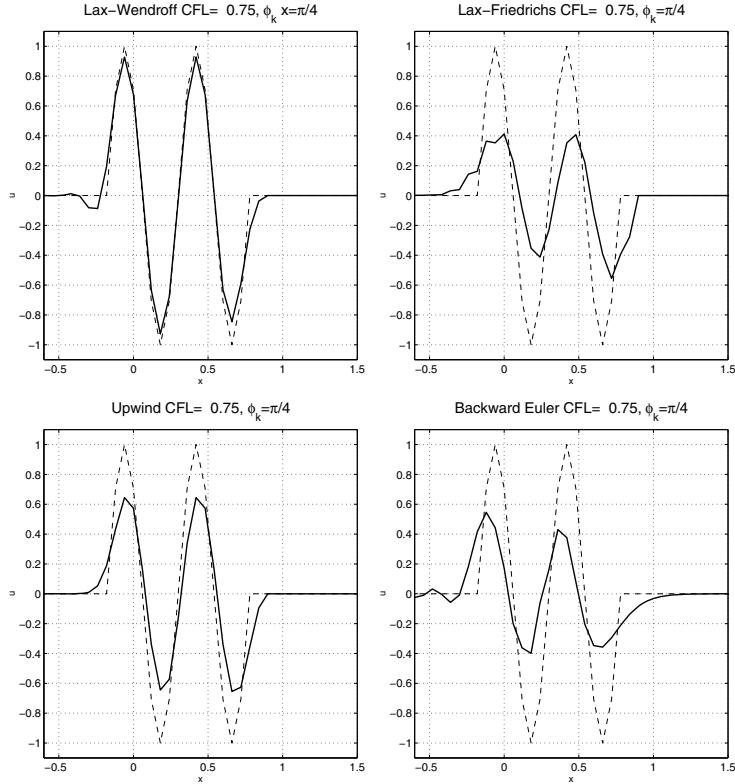


Fig. 12.8. Numerical solution of the convective transport of a packet of sinusoidal waves. The solid line represents the solution after 8 time steps. The etched line represents the corresponding exact solution at the same time level

Formally differentiating such equation with respect to t , we find

$$v_{tt} + av_{xt} = R_t^U + \mathcal{O}(\Delta t^3 + h^3).$$

Instead, differentiating it with respect to x , we have

$$v_{xt} + av_{xx} = R_x^U + \mathcal{O}(\Delta t^3 + h^3). \quad (12.46)$$

Hence,

$$v_{tt} = a^2 v_{xx} + R_t^U - aR_x^U + \mathcal{O}(\Delta t^3 + h^3) \quad (12.47)$$

which allows to obtain from (12.45)

$$v_t + av_x = \mu v_{xx} - \frac{1}{6}(ah^2 v_{xxx} + \Delta t^2 v_{ttt}) - \frac{\Delta t}{2}(R_t^U - aR_x^U) + \mathcal{O}(\Delta t^3 + h^3), \quad (12.48)$$

having set

$$\mu = \frac{1}{2}ah(1 - (a\lambda)) \quad (12.49)$$

and, as usual, $\lambda = \Delta t/h$. Now, formally differentiating (12.47) with respect to t , hence (12.46) with respect to x , we find

$$\begin{aligned} v_{ttt} &= a^2 v_{xxt} + R_{tt}^U - aR_{xt}^U + \mathcal{O}(\Delta t^3 + h^3) \\ &= -a^3 v_{xxx} + a^2 R_{xx}^U + R_{tt}^U - aR_{xt}^U + \mathcal{O}(\Delta t^3 + h^3). \end{aligned} \quad (12.50)$$

Moreover, we have that

$$\begin{aligned} R_t^U &= \frac{1}{2}ah v_{xxt} - \frac{\Delta t}{2}v_{ttt} - \frac{ah^2}{6}v_{xxxt} - \frac{\Delta t^2}{6}v_{tttt}, \\ R_x^U &= \frac{1}{2}ah v_{xxx} - \frac{\Delta t}{2}v_{txx} - \frac{ah^2}{6}v_{xxxx} - \frac{\Delta t^2}{6}v_{tttx}. \end{aligned} \quad (12.51)$$

Using the relations (12.50) and (12.51) in (12.48) we obtain

$$\begin{aligned} v_t + av_x &= \mu v_{xx} - \frac{ah^2}{6} \left(1 - \frac{a^2 \Delta t^2}{h^2} - \frac{3a \Delta t}{2h} \right) v_{xxx} \\ &\quad + \underbrace{\frac{\Delta t}{4} (\Delta t v_{ttt} - ah v_{xxt} - a \Delta t v_{txx})}_{(A)} \\ &\quad + \frac{\Delta t}{12} (\Delta t^2 v_{tttt} - a \Delta t^2 v_{tttx} + ah^2 v_{xxxt} - a^2 h^2 v_{xxxx}) \\ &\quad - \frac{a^2 \Delta t^2}{6} R_{xx}^U - \frac{\Delta t^2}{6} R_{tt}^U + a \frac{\Delta t^2}{6} R_{xt}^U + \mathcal{O}(\Delta t^3 + h^3). \end{aligned} \quad (12.52)$$

Let us now focus on the third derivatives of v contained in the term (A). Thanks to (12.50), (12.46) and (12.47), respectively, we find:

$$v_{ttt} = -a^3 v_{xxx} + r_1,$$

$$v_{xxt} = -a v_{xxx} + r_2,$$

$$v_{txx} = a^2 v_{xxx} + r_3,$$

where r_1 , r_2 and r_3 are terms containing derivatives of v of order no less than four, as well as terms of order $\mathcal{O}(\Delta t^3 + h^3)$. (Note that it follows from the definition of R^U that its derivatives of order two are expressed through derivatives of v of order no less than four.) Regrouping the coefficients that multiply v_{xxx} , we therefore deduce from (12.52) that

$$v_t + av_x = \mu v_{xx} + \nu v_{xxx} + R_4(v) + \mathcal{O}(\Delta t^3 + h^3), \quad (12.53)$$

having set

$$\nu = -\frac{ah^2}{6} (1 - 3a\lambda + 2(a\lambda)^2), \quad (12.54)$$

and having indicated with $R_4(v)$ the set of terms containing the derivatives of v of order no less than four.

We can conclude that the v function satisfies, respectively, the equations:

$$v_t + av_x = 0 \quad (12.55)$$

if we neglect the terms containing derivatives of order above the first;

$$v_t + av_x = \mu v_{xx} \quad (12.56)$$

if we neglect the terms containing derivatives of order above the second;

$$v_t + av_x = \mu v_{xx} + \nu v_{xxx} \quad (12.57)$$

if we neglect the derivatives of order above the third. The coefficients μ and ν have been defined in (12.49) and (12.54). Equations (12.55), (12.56) and (12.57) are called *equivalent equations* (at the first, second resp. third order) relative to the upwind scheme.

12.5.2 The Lax-Friedrichs and Lax-Wendroff case

Proceeding in a similar way, we can derive the equivalent equations of any numerical scheme. For instance, in the case of the Lax-Friedrichs scheme, having denoted by v a hypothetic function that verifies the equation (12.16) at each point (x, t) , having observed that

$$\begin{aligned} \frac{1}{2}(v(x+h, t) + v(x-h, t)) &= v + \frac{h^2}{2}v_{xx} + \mathcal{O}(h^4), \\ \frac{1}{2}(v(x+h, t) - v(x-h, t)) &= h v_x + \frac{h^3}{6}v_{xxx} + \mathcal{O}(h^4), \end{aligned}$$

we obtain

$$v_t + av_x = R^{LF} + \mathcal{O}\left(\frac{h^4}{\Delta t} + \Delta t^3\right), \quad (12.58)$$

having set

$$R^{LF} = \frac{h^2}{2\Delta t}(v_{xx} - \lambda^2 v_{tt}) - \frac{ah^2}{6}(v_{xxx} + \frac{\lambda^2}{a}v_{ttt}).$$

Proceeding as we did previously, tedious computation allows us to deduce from (12.58) the equivalent equations (12.55)-(12.57), in this case having however

$$\mu = \frac{h^2}{2\Delta t}(1 - (a\lambda)^2), \quad \nu = \frac{ah^2}{3}(1 - (a\lambda)^2).$$

In the case of the *Lax-Wendroff* scheme, the equivalent equations are characterized by the following parameters

$$\mu = 0, \quad \nu = \frac{ah^2}{6}((a\lambda)^2 - 1).$$

12.5.3 On the meaning of coefficients in equivalent equations

In general, in the equivalent equations, the term μv_{xx} represents a dissipation, while νv_{xxx} represents a dispersion. We can provide a heuristic proof of this by examining the solution to the problem

$$\begin{cases} v_t + av_x = \mu v_{xx} + \nu v_{xxx}, & x \in \mathbb{R}, t > 0, \\ v(x, 0) = e^{ikx}, & k \in \mathbb{Z}. \end{cases} \quad (12.59)$$

By applying the Fourier transform we find, if $\mu = \nu = 0$,

$$v(x, t) = e^{ik(x-at)},$$

while for μ and ν arbitrary real numbers (with $\mu > 0$) we have

$$v(x, t) = e^{-\mu k^2 t} e^{ik[x-(a+\nu k^2)t]}.$$

Comparing these two relations, we see that for growing μ , the modulus of the solution gets smaller. Such effect gets more remarkable as the frequency k increases (a phenomenon we have already registered in the previous section, albeit with partly different arguments).

The term μv_{xx} in (12.59) therefore has a dissipative effect on the solution. In turn, ν modifies the propagation rate of the solution, increasing it in the $\nu > 0$ case, and decreasing it if $\nu < 0$. Also in this case, the effect is more notable at high frequencies. Hence, the third derivative term νv_{xxx} introduces a dispersive effect.

In general, in the equivalent equation, even order spatial derivatives represent diffusive terms, while odd order derivatives represent dispersive terms. For first-order schemes (such as the upwind scheme) the dispersive effect is often barely visible, as it is disguised by the dissipative one. Taking Δt and h of the same order, from (12.56) and (12.57) we evince that $\nu \ll \mu$ for $h \rightarrow 0$, as $\nu = O(h^2)$ and $\mu = O(h)$. In particular, if the CFL number is $\frac{1}{2}$, the third-order equivalent equation of the upwind method features a null dispersion, in accordance with the numerical results seen in the previous section.

Conversely, the dispersive effect is evident for the Lax-Friedrichs scheme, as well as for the Lax-Wendroff scheme which, being of the second order, does not feature a dissipative term of type μv_{xx} . However, being stable, the latter cannot avoid being dissipative. Indeed, the fourth-order equivalent equation for the Lax-Wendroff scheme is

$$v_t + av_x = \frac{ah^2}{6} [(a\lambda)^2 - 1] v_{xxx} - \frac{ah^3}{6} a\lambda [1 - (a\lambda)^2] v_{xxxx},$$

where the last term is dissipative if $|a\lambda| < 1$, as it can easily be verified by applying the Fourier transform. We then recover, also for the Lax-Wendroff scheme, the CFL condition.

12.5.4 Equivalent equations and error analysis

The technique applied to obtain the equivalent equation denotes a strong analogy with the so-called *backward analysis* that we encounter during the numerical solution of

linear systems, where the computed (not exact) solution is interpreted as the exact solution of a perturbed linear system (see [QSS07, Chap. 3]). As a matter of fact, the perturbed system plays a similar role to that of the equivalent equation.

Moreover, we observe that an error analysis of a numerical scheme can be deduced from the knowledge of the equivalent equation associated to it. Indeed, by generically denoting by $r = \mu v_{xx} + \nu v_{xxx}$ the right-hand side of the equivalent equation, by comparison with (12.1) we obtain the error equation

$$e_t + ae_x = r,$$

where $e = v - u$. Multiplying such equation by e and integrating in space and time (between 0 and t) we obtain

$$\|e(t)\|_{L^2(\mathbb{R})} \leq C(t) \left(\|e(0)\|_{L^2(\mathbb{R})} + \sqrt{\int_0^t \|r(s)\|_{L^2(\mathbb{R})}^2 ds} \right), \quad t > 0$$

having used the a priori estimate (12.4). We can assume $e(0) = 0$ and therefore observe that $\|e(t)\|_{L^2(\mathbb{R})}$ tends to zero (for h and Δt tending to zero) at order 1 for the upwind or Lax-Friedrichs schemes, and at order 2 for the Lax-Wendroff scheme (having supposed v to be sufficiently regular).

12.6 Exercises

- Verify that the solution to the problem (12.9)-(12.10) (with $f = 0$) satisfies the identity (12.11).

[*Solution:* Multiplying (12.9) by u_t and integrating in space we obtain

$$0 = \int_{\alpha}^{\beta} u_{tt} u_t dx - \int_{\alpha}^{\beta} \gamma^2 u_{xx} u_t dx = \frac{1}{2} \int_{\alpha}^{\beta} [(u_t)^2]_t dx + \int_{\alpha}^{\beta} \gamma^2 u_x u_{xt} dx - [\gamma^2 u_x u_t]_{\alpha}^{\beta}. \quad (12.60)$$

As

$$\int_{\alpha}^{\beta} u_{tt} u_t dx = \frac{1}{2} \int_{\alpha}^{\beta} [(u_t)^2]_t dx \quad \text{e} \quad \int_{\alpha}^{\beta} \gamma^2 u_x u_{xt} dx = \frac{1}{2} \int_{\alpha}^{\beta} \gamma^2 [(u_x)^2]_t dx,$$

integrating (12.60) in time, we have

$$\int_{\alpha}^{\beta} u_t^2(t) dx + \int_{\alpha}^{\beta} \gamma^2 u_x^2(t) dx - \int_{\alpha}^{\beta} v_0^2 dx - \int_{\alpha}^{\beta} u_{0x}^2 dx = 0. \quad (12.61)$$

Hence, (12.11) immediately follows from the latter relation.]

2. Verify that the solution provided by the backward/centered Euler scheme (12.22) is unconditionally stable; more precisely,

$$\|\mathbf{u}\|_{\Delta,2} \leq \|\mathbf{u}^0\|_{\Delta,2} \quad \forall \Delta t, h > 0.$$

[*Solution:* Note that, thanks to (12.32),

$$(u_j^{n+1} - u_j^n)u_j^{n+1} \geq \frac{1}{2}(|u_j^{n+1}|^2 - |u_j^n|^2) \quad \forall j, n.$$

Then, multiplying (12.22) by u_j^{n+1} , summing over the index j and using (12.33) we find

$$\sum_j |u_j^{n+1}|^2 \leq \sum_j |u_j^n|^2 \quad \forall n \geq 0,$$

from which the result follows.]

3. Prove (12.30)

[*Solution:* We note that, in the case where $a > 0$, the upwind scheme can be rewritten in the form

$$u_j^{n+1} = (1 - a\lambda)u_j^n + a\lambda u_{j-1}^n.$$

Under the hypothesis (12.28) both coefficients $a\lambda$ and $1 - a\lambda$ are non-negative, hence

$$\min(u_j^n, u_{j-1}^n) \leq u_j^{n+1} \leq \max(u_j^n, u_{j-1}^n).$$

Then

$$\inf_{l \in \mathbb{Z}} \{u_l^0\} \leq u_j^n \leq \sup_{l \in \mathbb{Z}} \{u_l^0\} \quad \forall j \in \mathbb{Z}, \forall n \geq 0,$$

from which (12.30) follows.]

4. Study the accuracy of the Lax-Friedrichs scheme (12.16) for the solution of problem (12.1).
 5. Study the accuracy of the Lax-Wendroff scheme (12.18) for the solution of problem (12.1).

13

Finite elements and spectral methods for hyperbolic equations

In this chapter, we will illustrate how to apply Galerkin methods, and in particular the finite element method and the spectral one, to the spatial and/or temporal discretization of scalar hyperbolic equations. We will treat both the continuous as well as discontinuous finite element cases.

Let us consider the transport problem (12.3) and let us set for simplicity $(\alpha, \beta) = (0, 1)$, $\varphi = 0$. Moreover, let us suppose that a is a positive constant and a_0 a non-negative constant.

To start with, we proceed with a spatial discretization based on continuous finite elements. We therefore attempt a semidiscretization of the following form:

$\forall t > 0$, find $u_h = u_h(t) \in V_h$ s.t.

$$\left(\frac{\partial u_h}{\partial t}, v_h \right) + a \left(\frac{\partial u_h}{\partial x}, v_h \right) + a_0 (u_h, v_h) = (f, v_h) \quad \forall v_h \in V_h, \quad (13.1)$$

u_h^0 being the approximation of the initial datum. We have set

$$V_h = \{v_h \in X_h^r : v_h(0) = 0\}, \quad r \geq 1.$$

The space X_h^r is defined as in (4.14), provided that we replace (a, b) with $(0, 1)$.

13.1 Temporal discretization

For the temporal discretization of problem (13.1) we will use finite difference schemes such as those introduced in Chap. 12.

As usual, we will denote by u_h^n , $n \geq 0$, the approximation of u_h at time $t^n = n\Delta t$.

13.1.1 The forward and backward Euler schemes

In case we use the forward Euler scheme, the discrete problem becomes:

$\forall n \geq 0$, find $u_h^{n+1} \in V_h$ such that

$$\frac{1}{\Delta t} (u_h^{n+1} - u_h^n, v_h) + a \left(\frac{\partial u_h^n}{\partial x}, v_h \right) + a_0 (u_h^n, v_h) = (f^n, v_h) \quad \forall v_h \in V_h, \quad (13.2)$$

where $(u, v) = \int_0^1 u(x)v(x)dx$ denotes as usual the scalar product of $L^2(0, 1)$.

In the case of the backward Euler method, instead of (13.2) we will have

$$\frac{1}{\Delta t} (u_h^{n+1} - u_h^n, v_h) + a \left(\frac{\partial u_h^{n+1}}{\partial x}, v_h \right) + a_0 (u_h^{n+1}, v_h) = (f^{n+1}, v_h) \quad \forall v_h \in V_h. \quad (13.3)$$

Theorem 13.1 *The backward Euler scheme is strongly stable with no restriction on Δt . Instead, the forward Euler method, is strongly stable only for $a_0 > 0$, provided we suppose that*

$$\Delta t \leq \frac{2a_0}{(aCh^{-1} + a_0)^2} \quad (13.4)$$

for a given constant $C = C(r)$.

Proof. Choosing $v_h = u_h^n$ in (13.2), we obtain (in the $f = 0$ case)

$$(u_h^{n+1} - u_h^n, u_h^n) + \Delta t a \left(\frac{\partial u_h^n}{\partial x}, u_h^n \right) + \Delta t a_0 \|u_h^n\|_{L^2(0,1)}^2 = 0.$$

For the first term, we use the identity

$$(v - w, w) = \frac{1}{2} \left(\|v\|_{L^2(0,1)}^2 - \|w\|_{L^2(0,1)}^2 - \|v - w\|_{L^2(0,1)}^2 \right) \quad \forall v, w \in L^2(0, 1) \quad (13.5)$$

which generalizes (12.34). For the second addendum, integrating by parts and using the boundary conditions, we find

$$\left(\frac{\partial u_h^n}{\partial x}, u_h^n \right) = \frac{1}{2} (u_h^n(1))^2.$$

Thus, we obtain

$$\begin{aligned} & \|u_h^{n+1}\|_{L^2(0,1)}^2 + a \Delta t (u_h^n(1))^2 + 2a_0 \Delta t \|u_h^n\|_{L^2(0,1)}^2 \\ &= \|u_h^n\|_{L^2(0,1)}^2 + \|u_h^{n+1} - u_h^n\|_{L^2(0,1)}^2. \end{aligned} \quad (13.6)$$

We now seek an estimate for the term $\|u_h^{n+1} - u_h^n\|_{L^2(0,1)}^2$. To this end, setting in (13.2) $v_h = u_h^{n+1} - u_h^n$, we obtain

$$\begin{aligned} \|u_h^{n+1} - u_h^n\|_{L^2(0,1)}^2 &\leq \Delta t a \left| \left(\frac{\partial u_h^n}{\partial x}, u_h^{n+1} - u_h^n \right) \right| + \Delta t a_0 |(u_h^n, u_h^{n+1} - u_h^n)| \\ &\leq \Delta t \left[a \left\| \frac{\partial u_h^n}{\partial x} \right\|_{L^2(0,1)} + a_0 \|u_h^n\|_{L^2(0,1)} \right] \|u_h^{n+1} - u_h^n\|_{L^2(0,1)}. \end{aligned}$$

By now using the inverse inequality (11.39) (referred to the interval $(0, 1)$), we obtain

$$\|u_h^{n+1} - u_h^n\|_{L^2(0,1)} \leq \Delta t (aC_I h^{-1} + a_0) \|u_h^n\|_{L^2(0,1)}.$$

Finally, (13.6) becomes

$$\begin{aligned} & \|u_h^{n+1}\|_{L^2(0,1)}^2 + a\Delta t(u_h^n(1))^2 \\ & + \Delta t [2a_0 - \Delta t(aC_I h^{-1} + a_0)^2] \|u_h^n\|_{L^2(0,1)}^2 \leq \|u_h^n\|_{L^2(0,1)}^2. \end{aligned} \quad (13.7)$$

If (13.4) is satisfied, then $\|u_h^{n+1}\|_{L^2(0,1)} \leq \|u_h^n\|_{L^2(0,1)}$ and we therefore have strong stability in $L^2(0, 1)$ norm.

In the case where $a_0 = 0$ the obtained stability condition is meaningless. However, if we suppose that

$$\Delta t \leq \frac{Kh^2}{a^2 C_I^2},$$

for a given constant $K > 0$, then we can apply the discrete Gronwall lemma (see Lemma 2.3) to (13.7) and we find that the method is stable with a stability constant which in this case depends on the final time T . Precisely,

$$\|u_h^n\|_{L^2(0,1)} \leq \exp(Kt^n) \|u_h^0\|_{L^2(0,1)} \leq \exp(KT) \|u_h^0\|_{L^2(0,1)}.$$

In the case of the backward Euler method (13.3), we choose instead $v_h = u_h^{n+1}$. By then using the relation

$$(v - w, v) = \frac{1}{2} (\|v\|_{L^2(0,1)}^2 - \|w\|_{L^2(0,1)}^2 + \|v - w\|_{L^2(0,1)}^2) \quad \forall v, w \in L^2(0, 1) \quad (13.8)$$

which generalizes (12.32), we find,

$$(1 + 2a_0\Delta t) \|u_h^{n+1}\|_{L^2(0,1)}^2 + a\Delta t(u_h^{n+1}(1))^2 \leq \|u_h^n\|_{L^2(0,1)}^2. \quad (13.9)$$

Hence, we have strong stability in $L^2(0, 1)$, unconditioned (that is for each Δt) and for each $a_0 \geq 0$. \diamond

13.1.2 The upwind, Lax-Friedrichs and Lax-Wendroff schemes

The generalization to the finite elements case of the Lax-Friedrichs (LF), Lax-Wendroff (LW) and upwind (U) finite difference schemes can be made in different ways.

We start by observing that (12.16), (12.18), and (12.20) can be rewritten in the following common form

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a \frac{u_{j+1}^n - u_{j-1}^n}{2h} - \mu \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2} + a_0 u_j^n = 0. \quad (13.10)$$

(Note however that $a_0 = 0$ in (12.16), (12.18) and (12.20).) The second term is the discretization via centered finite differences of the convective term $au_x(t^n)$, while the

third one is a numerical diffusion term and corresponds to the discretization via finite differences of $-\mu u_{xx}(t^n)$. The numerical viscosity coefficient μ is given by

$$\mu = \begin{cases} h^2/2\Delta t & (\text{LF}), \\ a^2\Delta t/2 & (\text{LW}), \\ ah/2 & (\text{U}). \end{cases} \quad (13.11)$$

Equation (13.10) suggests the following finite element version for the approximation of problem (12.3): $\forall n \geq 0$, find $u_h^{n+1} \in V_h$ such that

$$\begin{aligned} \frac{1}{\Delta t} (u_h^{n+1} - u_h^n, v_h) + a \left(\frac{\partial u_h^n}{\partial x}, v_h \right) + a_0 (u_h^n, v_h) \\ + \mu \left(\frac{\partial u_h^n}{\partial x}, \frac{\partial v_h}{\partial x} \right) - \mu \gamma \frac{\partial u_h^n}{\partial x}(1)v_h(1) = (f^n, v_h) \quad \forall v_h \in V_h, \end{aligned} \quad (13.12)$$

where $\gamma = 1, 0$ depending on whether or not we want to take the boundary contribution into account in the integration by parts of the numerical viscosity term.

For the stability analysis, in the case $\gamma = 0$, $a_0 = 0$, $a > 0$, let us set $v_h = u_h^{n+1} - u_h^n$, in order to obtain, thanks to inequality (4.52)

$$\|u_h^{n+1} - u_h^n\|_{L^2(0,1)} \leq \Delta t(a + \mu C_I h^{-1}) \left\| \frac{\partial u_h^n}{\partial x} \right\|_{L^2(0,1)}.$$

Having now set $v_h = u_h^n$, thanks to (13.5) we obtain

$$\begin{aligned} \|u_h^{n+1}\|_{L^2(0,1)}^2 &= \|u_h^n\|_{L^2(0,1)}^2 + a\Delta t(u_h^n(1))^2 + 2\Delta t\mu \left\| \frac{\partial u_h^n}{\partial x} \right\|_{L^2(0,1)}^2 \\ &= \|u_h^{n+1} - u_h^n\|_{L^2(0,1)}^2 \leq \Delta t^2(a + \mu C_I h^{-1})^2 \left\| \frac{\partial u_h^n}{\partial x} \right\|_{L^2(0,1)}^2. \end{aligned}$$

A sufficient condition for strong stability (i.e. to obtain an estimate such as (12.27), with respect to the $\|\cdot\|_{L^2(0,1)}$ norm) is therefore

$$\Delta t \leq \frac{2\mu}{(a + \mu C_I h^{-1})^2}.$$

Thanks to (13.11), in the case of the upwind method this is equivalent to

$$\Delta t \leq \frac{h}{a} \left(\frac{1}{1 + C_I/2} \right)^2.$$

In the case of linear finite elements, $C_I \simeq 2\sqrt{3}$, therefore we deduce that

$$\frac{a\Delta t}{h} \lesssim \left(\frac{1}{1 + \sqrt{3}} \right)^2.$$

The stability analysis we have just developed is based on the *energy method* and, in this case, leads to sub-optimal results. A better indicator can be obtained by resorting to the von Neumann analysis, as done in Sec. 12.4.3. To this end we observe that, in the case of linear finite elements with constant spacing h , (13.12) with $f = 0$ can be rewritten in the following way for each internal node x_j :

$$\begin{aligned} \frac{1}{6}(u_{j+1}^{n+1} + 4u_j^{n+1} + u_{j-1}^{n+1}) + \frac{\lambda a}{2}(u_{j+1}^n - u_{j-1}^n) + \frac{a_0}{6}(u_{j+1}^n + 4u_j^n + u_{j-1}^n) \\ - \mu \Delta t \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2} = \frac{1}{6}(u_{j+1}^n + 4u_j^n + u_{j-1}^n). \end{aligned} \quad (13.13)$$

By comparing such relation to (13.10), we can note that the difference only resides in the term arising from the temporal derivative and from the term of order zero, and has to be attributed to the presence of the mass matrix in the case of finite elements. On the other hand, we have previously seen in Sec. 11.5 that we can apply the mass-lumping technique to approximate the mass matrix using a diagonal matrix. By proceeding this way, the scheme (13.13) can effectively be reduced to (13.10) (see Exercise 1).

Remark 13.1 Note that the provided relations (13.13) refer to the internal nodes. The approach used to handle boundary conditions with the finite element method generally yields different relations than those obtained via the finite difference method. •

These observations allow us to extend all the schemes seen in Sec. 12.3.1 to analogous schemes, generated by discretizations in space with continuous linear finite elements. To this end, it will be sufficient to replace the term $u_j^{n+1} - u_j^n$ with

$$\frac{1}{6}[(u_{j-1}^{n+1} - u_{j-1}^n) + 4(u_j^{n+1} - u_j^n) + (u_{j+1}^{n+1} - u_{j+1}^n)].$$

Thus, the general scheme (12.13) is replaced by

$$\frac{1}{6}(u_{j-1}^{n+1} + 4u_j^{n+1} + u_{j+1}^{n+1}) = \frac{1}{6}(u_{j-1}^n + 4u_j^n + u_{j+1}^n) - \lambda(H_{j+1/2}^{n*} - H_{j-1/2}^{n*}), \quad (13.14)$$

where

$$H_{j+1/2}^{n*} = \begin{cases} H_{j+1/2}^n & \text{for explicit time-advancing schemes,} \\ H_{j+1/2}^{n+1} & \text{for implicit time-advancing schemes.} \end{cases}$$

Note that, even if we adopted a numerical flux corresponding to an explicit time-advancing scheme, the resulting scheme would no longer yield to a diagonal system (indeed, it becomes a tridiagonal one) because of the mass matrix terms. It could therefore appear that the use of an explicit time-advancing scheme for finite elements is inconvenient with respect to a similar full finite difference scheme. However, such a scheme has interesting features. In particular, let us consider its amplification and dispersion coefficients, using the von Neumann analysis illustrated in Sec. 12.4.3. To this end, let us suppose that the differential equation be defined on all of \mathbb{R} , or,

alternatively, let us consider a bounded interval, however imposing periodic boundary conditions. In either case, we can assume that relation (13.14) holds for all values of the index j . A simple computation leads us to writing the following relation between the amplification coefficient γ_k of a finite difference scheme (see Table 12.1) and the amplification coefficient γ_k^{FEM} of the corresponding finite element scheme

$$\gamma_k^{\text{FEM}} = \frac{3\gamma_k - 1 + \cos(\phi_k)}{2 + \cos(\phi_k)}, \quad (13.15)$$

where we denote again with ϕ_k the phase angle relative to the k -th harmonic (see Sec. 12.4.3).

We can thus compute the amplification and dispersion errors, which are reported in Fig. 13.1. Comparing them with the analogous errors relating to the corresponding finite difference scheme (reported in Fig. 12.6) we can make the following remarks. The forward Euler scheme is still unconditionally unstable (in the sense of strong stability). The upwind scheme (FEM) is strongly stable if the CFL number is less than $\frac{1}{3}$ (hence, a less restrictive result than the one found using the energy method), while the Lax-Friedrichs (FEM) method *never satisfies* the condition $\gamma_k^{\text{FEM}} \leq 1$ (in this case, in accordance with the result that we would find using the energy method). More generally, we can say that in the case of schemes with an explicit temporal treatment the “finite element” version requires more restrictive stability conditions than the corresponding finite difference one. In particular, for the Lax-Wendroff finite element scheme, that we will denote with LW (FEM), the CFL number must now be less than $\frac{1}{\sqrt{3}}$, instead of 1 as in the finite differences case. However, the LW (FEM) scheme (for the CFL values for which it is stable), results to be slightly less diffusive and dispersive than the equivalent finite difference scheme for a wide range of values of the phase angle $\phi_k = kh$. The implicit Euler scheme remains unconditionally stable also in the FEM version (coherently with what we obtained using the energy method in Sec. 13.1.1).

Example 13.1 The previous conclusions have been experimentally verified as follows. We have repeated the case of Fig. 12.7 (right), where we have now considered a CFL value of 0.5. The numerical solutions obtained via the classical Lax-Wendroff method (LW) and via LW (FEM) for $t = 2$ are reported in Fig. 13.2. We can note how the LW (FEM) scheme provides a solution that is more accurate and especially more in phase with the exact solution. This result is confirmed by the value of the $\|\cdot\|_{\Delta,2}$ norm of the error in the two cases. Indeed, by calling u the exact solution and u_{LW} resp. $u_{\text{LW(FEM)}}$ the one obtained using the two numerical schemes, $\|u_{\text{LW}} - u\|_{\Delta,2} = 0.78$, $\|u_{\text{LW(FEM)}} - u\|_{\Delta,2} = 0.49$.

Further tests conducted by using non periodic boundary conditions confirm the stability properties previously derived. ■

13.2 Taylor-Galerkin schemes

We now illustrate a class of finite element schemes named “Taylor-Galerkin” schemes. These are derived in a similar way to the Lax-Wendroff scheme, and we will indeed see that the LW (FEM) version is part of their class.

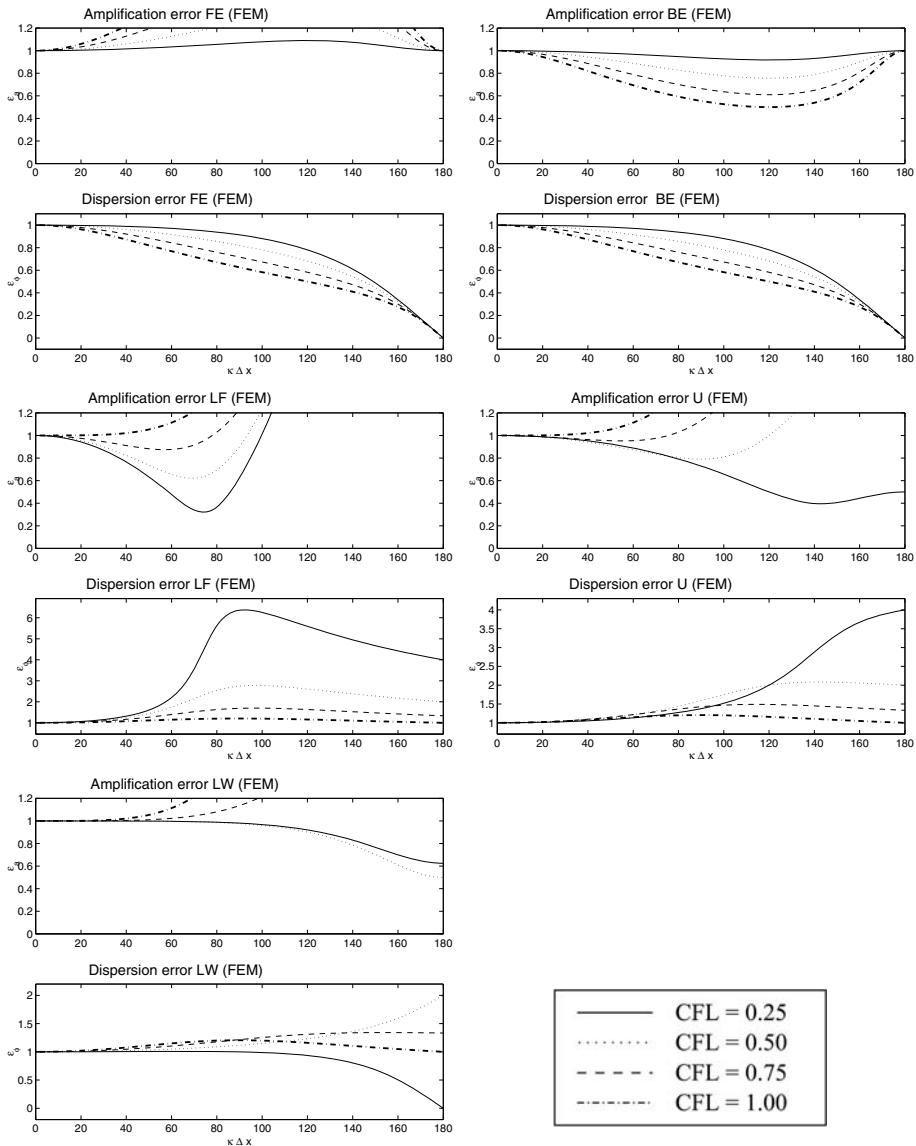


Fig. 13.1. Amplification and dispersion errors for several finite element schemes obtained from the general scheme (13.14)

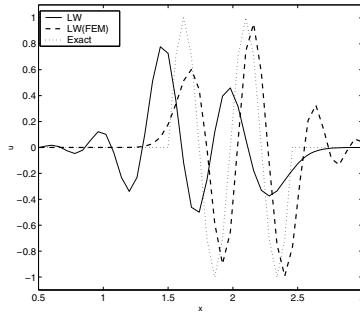


Fig. 13.2. Comparison between the solution obtained via the Lax-Wendroff finite difference scheme (LW) and its finite element version (LW (FEM)) ($\phi_k = \pi/4$, $t = 2$)

For simplicity, we will refer to the pure transport problem (12.1). The Taylor-Galerkin method consists in combining the Taylor formula truncated to the first order

$$u(x, t^{n+1}) = u(x, t^n) + \Delta t \frac{\partial u}{\partial t}(x, t^n) + \int_{t^n}^{t^{n+1}} (s - t^n) \frac{\partial^2 u}{\partial t^2}(x, s) ds \quad (13.16)$$

with equation (12.1), thanks to which we obtain

$$\frac{\partial u}{\partial t} = -a \frac{\partial u}{\partial x},$$

and, by formal derivation,

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial t} \left(-a \frac{\partial u}{\partial x} \right) = -a \frac{\partial}{\partial x} \frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}.$$

From (13.16) we then obtain

$$u(x, t^{n+1}) = u(x, t^n) - a \Delta t \frac{\partial u}{\partial x}(x, t^n) + a^2 \int_{t^n}^{t^{n+1}} (s - t^n) \frac{\partial^2 u}{\partial x^2}(x, s) ds. \quad (13.17)$$

We approximate the integral in the following way

$$\int_{t^n}^{t^{n+1}} (s - t^n) \frac{\partial^2 u}{\partial x^2}(x, s) ds \approx \frac{\Delta t^2}{2} \left[\theta \frac{\partial^2 u}{\partial x^2}(x, t^n) + (1 - \theta) \frac{\partial^2 u}{\partial x^2}(x, t^{n+1}) \right], \quad (13.18)$$

obtained by evaluating the first factor in $s = t^n + \frac{\Delta t}{2}$ and the second one through a linear combination using $\theta \in [0, 1]$ as a parameter of its values in $s = t^n$ and $s = t^{n+1}$. We denote by $u^n(x)$ the approximating function $u(x, t^n)$.

Let us consider two remarkable situations. If $\theta = 1$, the resulting semi-discretized scheme is explicit in time and is written as

$$u^{n+1} = u^n - a \Delta t \frac{\partial u^n}{\partial x} + \frac{a^2 \Delta t^2}{2} \frac{\partial^2 u^n}{\partial x^2}.$$

If we now discretize in space by finite differences or finite elements, we re-encounter the previously examined LW and LW (FEM) schemes.

Instead, if we take $\theta = \frac{2}{3}$, the approximation error in (13.18) becomes $O(\Delta t^4)$ (supposing that u has the required regularity). De facto, such choice corresponds to approximating $\frac{\partial^2 u}{\partial x^2}$ between t^n and t^{n+1} with its linear interpolant. The resulting semi-discretized scheme is written

$$\left[1 - \frac{a^2 \Delta t^2}{6} \frac{\partial^2}{\partial x^2} \right] u^{n+1} = u^n - a \Delta t \frac{\partial u^n}{\partial x} + \frac{a^2 \Delta t^2}{3} \frac{\partial^2 u^n}{\partial x^2}, \quad (13.19)$$

and the truncation error of the semi-discretized scheme in time (13.19) is $\mathcal{O}(\Delta t^3)$.

At this point, a discretization in space using the finite element method leads to the following scheme, called Taylor-Galerkin (TG):

for $n = 0, 1, \dots$ find $u_h^{n+1} \in V_h$ such that

$$\begin{aligned} A(u_h^{n+1}, v_h) &= (u_h^n, v_h) - a \Delta t \left(\frac{\partial u_h^n}{\partial x}, v_h \right) - \frac{a^2 \Delta t^2}{3} \left(\frac{\partial u_h^n}{\partial x}, \frac{\partial v_h}{\partial x} \right) \\ &\quad + \gamma \frac{a^2 \Delta t^2}{3} \frac{\partial u_h^n}{\partial x}(1) v_h(1) \quad \forall v_h \in V_h, \end{aligned} \quad (13.20)$$

where

$$A(u_h^{n+1}, v_h) = (u_h^{n+1}, v_h) + \frac{a^2 \Delta t^2}{6} \left(\frac{\partial u_h^{n+1}}{\partial x}, \frac{\partial v_h}{\partial x} \right) - \gamma \frac{a^2 \Delta t^2}{6} \frac{\partial u_h^{n+1}}{\partial x}(1) v_h(1),$$

and $\gamma = 1, 0$ depending on whether or not we want to take into account the boundary contribution in the integration by parts of the second derivative term.

The latter yields a linear system whose matrix is

$$A = M + \frac{a^2 (\Delta t)^2}{6} K,$$

M being the mass matrix and K being the stiffness matrix, potentially taking the boundary contribution as well into account (if $\gamma = 1$).

In the case of linear finite elements, the von Neumann analysis leads to the following amplification factor for the scheme (13.20)

$$\gamma_k = \frac{2 + \cos(kh) - 2a^2 \lambda^2(1 - \cos(kh)) + 3ia\lambda \sin(kh)}{2 + \cos(kh) + a^2 \lambda^2(1 - \cos(kh))}. \quad (13.21)$$

It can be proven that the scheme is strongly stable in the $\|\cdot\|_{\Delta,2}$ norm under the CFL condition $\frac{a\Delta t}{h} \leq 1$. Thus, it has a *less restrictive* stability condition than the Lax-Wendroff (FEM) scheme.

Fig. 13.3 shows the behavior of the amplification and dispersion error for the scheme (13.20), as a function of the phase angle, analogously to what we have seen for other schemes in Sec. 12.4.4.

In the case of linear finite elements the truncation error of the TG scheme results to be $\mathcal{O}(\Delta t^3) + \mathcal{O}(h^2) + \mathcal{O}(h^2 \Delta t)$.

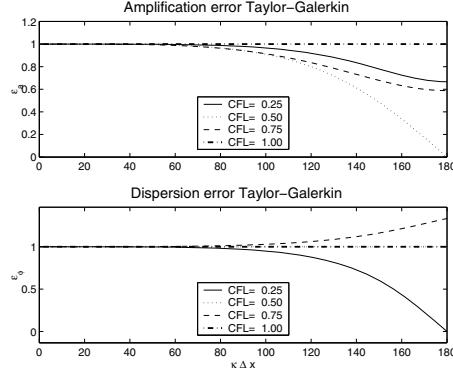


Fig. 13.3. Amplification (top) and dispersion (bottom) error of the Taylor-Galerkin scheme (13.20), as a function of the phase angle $\phi_k = kh$ and for different values of the CFL number

Example 13.2 To compare the accuracy of the schemes presented in the last two sections, we have considered the problem

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0, & x \in (0, 0.5), t > 0, \\ u(x, 0) = u_0(x), & x \in (0, 0.5), \end{cases}$$

with periodic boundary conditions, $u(0, t) = u(0.5, t)$, for $t > 0$. The initial datum is $u_0(x) = 2 \cos(4\pi x) + \sin(20\pi x)$, and is illustrated in Fig. 13.4 (left). The latter superposes two harmonics, one with low frequency one and one with high frequency.

We have considered the Taylor-Galerkin, Lax-Wendroff (FEM), (finite difference) Lax-Wendroff and upwind schemes. In Fig. 13.4 (right), we show the error in discrete norm $\|u - u_h\|_{\Delta,2}$ obtained at time $t = 1$ for different values of Δt and at a fixed CFL number

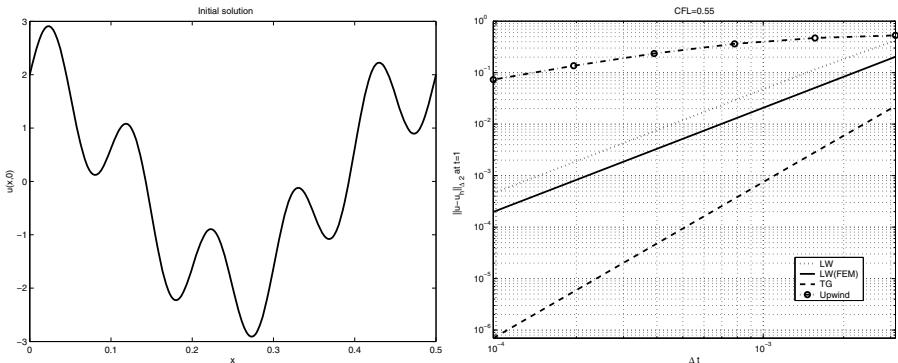


Fig. 13.4. Initial condition u_0 for the simulation of example 13.2 (left) and error $\|u - u_h\|_{\Delta,2}$ at $t = 1$ for varying Δt at fixed CFL for different numerical schemes (right)

of 0.55. We can note a better convergence of the Taylor-Galerkin scheme, while the two versions of the Lax-Wendroff scheme show the same order of convergence, but with a smaller error for the finite element version with respect to the finite difference one. The upwind scheme is less accurate: it features a larger absolute error and a lower convergence rate. Moreover, it can be verified that for a fixed CFL, the error of the upwind scheme is $\mathcal{O}(\Delta t)$, that of both variants of the Lax-Wendroff scheme is $\mathcal{O}(\Delta t^2)$, while the error of the Taylor-Galerkin scheme is $\mathcal{O}(\Delta t^3)$. ■

We report in Fig. 13.5 and 13.6 the numerical approximations and corresponding errors in the maximum norm for the transport problem

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} = 0, & x \in (0, 2\pi), t > 0 \\ u(x, 0) = \sin(\pi \cos(x)), & x \in (0, 2\pi) \end{cases}$$

and periodic boundary conditions. Such approximations are obtained using finite differences of order 2 and 4 (ufd2, ufd4), compact finite differences of order 4 and 6 (ucp4, ucp6), and by the Galerkin spectral method using Fourier basis (ugal). For the

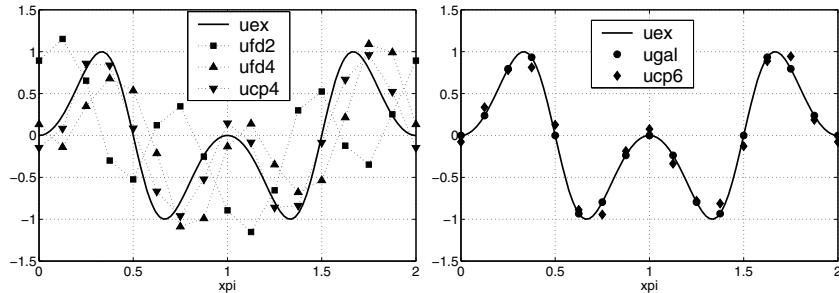


Fig. 13.5. Approximation of the solution of a wave propagation problem using finite difference methods (of order 2, 4), compact finite difference methods (of order 4 and 6) and with the Fourier Galerkin spectral method (from [CHQZ06])

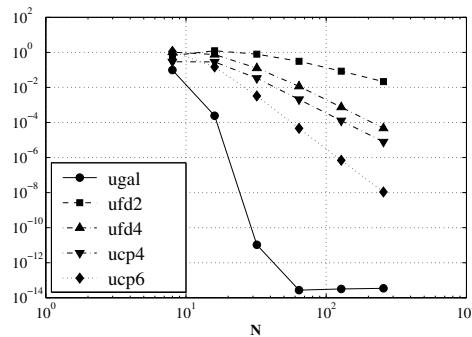


Fig. 13.6. Behavior of the error in the maximum norm for the different numerical methods reported in Fig. 13.5 (from [CHQZ06])

sake of comparison, we also report the exact solution $u(x, t) = \sin(\pi \cos(x + t))$ (uex).

13.3 The multi-dimensional case

Let us now move to the multi-dimensional case and let us consider the following first-order, linear and scalar hyperbolic transport-reaction problem in the domain $\Omega \subset \mathbb{R}^d$, with $d = 2, 3$

$$\begin{cases} \frac{\partial u}{\partial t} + \mathbf{a} \cdot \nabla u + a_0 u = f, & \mathbf{x} \in \Omega, t > 0 \\ u = \varphi, & \mathbf{x} \in \partial\Omega^{in}, t > 0, \\ u|_{t=0} = u_0, & \mathbf{x} \in \Omega, \end{cases} \quad (13.22)$$

where $\mathbf{a} = \mathbf{a}(\mathbf{x})$, $a_0 = a_0(\mathbf{x}, t)$ (optionally null), $f = f(\mathbf{x}, t)$, $\varphi = \varphi(\mathbf{x}, t)$ and $u_0 = u_0(\mathbf{x})$ are given functions. The inflow boundary $\partial\Omega^{in}$ is defined by

$$\partial\Omega^{in} = \{\mathbf{x} \in \partial\Omega : \mathbf{a}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0\}, \quad (13.23)$$

\mathbf{n} being the outward unit normal vector to $\partial\Omega$.

For simplicity, we have supposed that \mathbf{a} does not depend on t ; this way, the inflow boundary $\partial\Omega^{in}$ does not change with time.

13.3.1 Semi-discretization: strong and weak treatment of the boundary conditions

To obtain a semi-discrete approximation of problem (13.22), similar to that used in the one-dimensional case (13.1), we define the spaces

$$V_h = X_h^r, \quad V_h^{in} = \{v_h \in V_h : v_h|_{\partial\Omega^{in}} = 0\},$$

where r is an integer ≥ 1 and X_h^r has been introduced in (4.38).

We denote by $u_{0,h}$ and φ_h two suitable finite element approximations of u_0 and φ , respectively, and we consider the problem: for each $t > 0$ find $u_h(t) \in V_h$ such that

$$\begin{cases} \int_{\Omega} \frac{\partial u_h(t)}{\partial t} v_h \, d\Omega + \int_{\Omega} \mathbf{a} \cdot \nabla u_h(t) v_h \, d\Omega + \int_{\Omega} a_0(t) u_h(t) v_h \, d\Omega \\ = \int_{\Omega} f(t) v_h \, d\Omega \quad \forall v_h \in V_h^{in}, \\ u_h(t) = \varphi_h(t) \quad \text{on } \partial\Omega^{in}, \end{cases} \quad (13.24)$$

with $u_h(0) = u_{0,h} \in V_h$.

To obtain a stability estimate, we assume for simplicity that φ , and therefore φ_h , is identically null. In this case $u_h(t) \in V_h^{in}$, and taking, for every t , $v_h = u_h(t)$, we get the following inequality

$$\begin{aligned} \|u_h(t)\|_{L^2(\Omega)}^2 &+ \int_0^t \mu_0 \|u_h(\tau)\|_{L^2(\Omega)}^2 d\tau + \int_0^t \int_{\partial\Omega \setminus \partial\Omega^{in}} \mathbf{a} \cdot \mathbf{n} u_h^2(\tau) d\gamma d\tau \\ &\leq \|u_{0,h}\|_{L^2(\Omega)}^2 + \int_0^t \frac{1}{\mu_0} \|f(\tau)\|_{L^2(\Omega)}^2 d\tau. \end{aligned} \quad (13.25)$$

We have assumed that there exists a positive constant μ_0 s.t., for all $t > 0$ and for each \mathbf{x} in Ω

$$0 < \mu_0 \leq \mu(\mathbf{x}, t) = a_0(\mathbf{x}, t) - \frac{1}{2} \operatorname{div} \mathbf{a}(\mathbf{x}). \quad (13.26)$$

In the case where such hypothesis is not verified (for instance if \mathbf{a} is a constant field and $a_0 = 0$), then by using the Gronwall Lemma 2.2 we obtain

$$\begin{aligned} \|u_h(t)\|_{L^2(\Omega)}^2 &+ \int_0^t \int_{\partial\Omega \setminus \partial\Omega^{in}} \mathbf{a} \cdot \mathbf{n} u_h^2(\tau) d\gamma d\tau \\ &\leq \left(\|u_{0,h}\|_{L^2(\Omega)}^2 + \int_0^t \|f(\tau)\|_{L^2(\Omega)}^2 d\tau \right) \exp \int_0^t [1 + 2\mu^*(\tau)] d\tau, \end{aligned} \quad (13.27)$$

where we have set $\mu^*(t) = \max_{\mathbf{x} \in \bar{\Omega}} |\mu(\mathbf{x}, t)|$.

Supposing for simplicity that $f = 0$, if $u_0 \in H^{r+1}(\Omega)$ we have the following convergence result

$$\begin{aligned} \max_{t \in [0, T]} \|u(t) - u_h(t)\|_{L^2(\Omega)} &+ \left(\int_0^T \int_{\partial\Omega} |\mathbf{a} \cdot \mathbf{n}| |u(t) - u_h(t)|^2 d\gamma dt \right)^{1/2} \\ &\leq Ch^r \|u_0\|_{H^{r+1}(\Omega)}. \end{aligned}$$

For the proofs, we refer to [QV94, Chap. 14], [Joh87] and to the references cited thereby.

In problem (13.24) the boundary condition has been imposed in a *strong* (or essential) way. An alternative option is the *weak* (or natural) treatment that derives from the integration by parts of the transport term in the first equation in (13.24), where we now consider $v_h \in V_h$ (i.e. we no longer require that the test function be null on the inflow boundary). We obtain

$$\begin{aligned} &\int_{\Omega} \frac{\partial u_h(t)}{\partial t} v_h d\Omega - \int_{\Omega} \operatorname{div}(\mathbf{a} v_h) u_h(t) d\Omega \\ &+ \int_{\Omega} a_0 u_h(t) v_h d\Omega + \int_{\partial\Omega} \mathbf{a} \cdot \mathbf{n} u_h(t) v_h d\gamma = \int_{\Omega} f(t) v_h d\Omega. \end{aligned}$$

The boundary condition is imposed by replacing u_h with φ_h on the inflow boundary part, obtaining

$$\begin{aligned} & \int_{\Omega} \frac{\partial u_h(t)}{\partial t} v_h \, d\Omega - \int_{\Omega} \operatorname{div}(\mathbf{a} v_h) u_h(t) \, d\Omega \\ & + \int_{\Omega} a_0 u_h(t) v_h \, d\Omega + \int_{\partial\Omega \setminus \partial\Omega^{in}} \mathbf{a} \cdot \mathbf{n} u_h(t) v_h \, d\gamma \\ & = \int_{\Omega} f(t) v_h \, d\Omega - \int_{\partial\Omega^{in}} \mathbf{a} \cdot \mathbf{n} \varphi_h(t) v_h \, d\gamma \quad \forall v_h \in V_h. \end{aligned} \quad (13.28)$$

Clearly, the solution u_h found in this way only satisfies the boundary condition in an approximate way.

A further option consists in counter-integrating (13.28) by parts, thus getting to the following formulation: for each $t > 0$, find $u_h(t) \in V_h$ such that

$$\begin{aligned} & \int_{\Omega} \frac{\partial u_h(t)}{\partial t} v_h \, d\Omega + \int_{\Omega} \mathbf{a} \cdot \nabla u_h(t) v_h \, d\Omega + \int_{\partial\Omega^{in}} v_h (\varphi_h(t) - u_h(t)) \mathbf{a} \cdot \mathbf{n} \, d\gamma \\ & = \int_{\Omega} f(t) v_h \, d\Omega \quad \forall v_h \in V_h. \end{aligned} \quad (13.29)$$

We note that the (13.28) and (13.29) formulations are equivalent: the only difference is the way boundary terms are highlighted. In particular, the boundary integral in formulation (13.29) can be interpreted as a penalization term with which we evaluate how different u_h is from the data φ_h on the inflow boundary. Assuming that hypothesis (13.26) is still true, having chosen $v_h = u_h(t)$ in (13.29), integrating the convective term by parts and using the Cauchy-Schwarz and Young inequalities we get the following stability estimate

$$\begin{aligned} & \|u_h(t)\|_{L^2(\Omega)}^2 + \int_0^t \mu_0 \|u_h(\tau)\|_{L^2(\Omega)}^2 \, d\tau + \int_0^t \int_{\partial\Omega \setminus \partial\Omega^{in}} \mathbf{a} \cdot \mathbf{n} u_h^2(\tau) \, d\gamma \, d\tau \\ & \leq \|u_{0,h}\|_{L^2(\Omega)}^2 + \int_0^t \int_{\partial\Omega^{in}} |\mathbf{a} \cdot \mathbf{n}| \varphi_h^2(\tau) \, d\gamma \, d\tau + \int_0^t \frac{1}{\mu_0} \|f(\tau)\|_{L^2(\Omega)}^2 \, d\tau. \end{aligned} \quad (13.30)$$

In absence of hypothesis (13.26), inequality (13.30) would change in an analogous way to what we have previously seen, provided we use the Gronwall Lemma 2.2 as done to derive (13.27).

Remark 13.2 In the case where the boundary condition for problem (13.22) takes the form $\mathbf{a} \cdot \mathbf{n} u = \psi$, we could again impose it weakly by adding a penalization term, that

in such case would take the form

$$\int_{\partial\Omega^{in}} (\psi_h(t) - \mathbf{a} \cdot \mathbf{n} u_h(t)) v_h \, d\gamma,$$

ψ_h being a suitable finite element approximation of the datum ψ . •

Alternatively to the strong and weak imposition of the boundary conditions, i.e. to formulations (13.24) and (13.29), we could adopt a Petrov-Galerkin approach by imposing in a strong way the condition $u_h(t) = \varphi_h(t)$ on the inflow boundary $\partial\Omega^{in}$ and requiring $v_h = 0$ on the outflow boundary $\partial\Omega^{out}$, yielding the following discrete formulation. Set $V_h^{out} = \{v_h \in V_h : v_h|_{\partial\Omega^{out}} = 0\}$ then, for each $t > 0$, find $u_h(t) \in V_h = X_h^r$ such that

$$\left\{ \begin{array}{l} \int_{\Omega} \frac{\partial u_h(t)}{\partial t} v_h \, d\Omega + \int_{\Omega} (\mathbf{a} \cdot \nabla u_h(t)) v_h \, d\Omega + \int_{\Omega} a_0(t) u_h(t) v_h \, d\Omega \\ \qquad \qquad \qquad = \int_{\Omega} f(t) v_h \, d\Omega \quad \forall v_h \in V_h^{out}, \\ u_h(t) = \varphi_h(t) \quad \text{on } \partial\Omega^{in}. \end{array} \right.$$

We recall that for a Petrov-Galerkin formulation, the well-posedness analysis cannot be based on the Lax-Milgram lemma any longer.

Instead, if the inflow condition were imposed in a weak way, we would have the following formulation:

for each $t > 0$, find $u_h(t) \in V_h = X_h^r$ such that, for each $v_h \in V_h^{out}$,

$$\begin{aligned} & \int_{\Omega} \frac{\partial u_h(t)}{\partial t} v_h \, d\Omega - \int_{\Omega} \operatorname{div}(\mathbf{a} v_h) u_h(t) \, d\Omega + \int_{\Omega} a_0(t) u_h(t) v_h \, d\Omega \\ &= \int_{\Omega} f(t) v_h \, d\Omega - \int_{\partial\Omega^{in}} \mathbf{a} \cdot \mathbf{n} \varphi_h(t) v_h \, d\gamma. \end{aligned}$$

For further details, the reader can refer to [QV94, Chap. 14].

13.3.2 Temporal discretization

For an illustrative purpose, let us limit ourselves to considering the Galerkin semi-discrete problem (13.24). Using the backward Euler scheme for the temporal discretization, we get to the following fully-discrete problem:

$\forall n \geq 0$ find $u_h^n \in V_h$ s.t.

$$\begin{cases} \frac{1}{\Delta t} \int_{\Omega} (u_h^{n+1} - u_h^n) v_h \, d\Omega + \int_{\Omega} \mathbf{a} \cdot \nabla u_h^{n+1} v_h \, d\Omega + \int_{\Omega} a_0^{n+1} u_h^{n+1} v_h \, d\Omega \\ = \int_{\Omega} f^{n+1} v_h \, d\Omega \quad \forall v_h \in V_h^{in}, \\ u_h^{n+1} = \varphi_h^{n+1} \text{ on } \partial\Omega^{in}, \end{cases}$$

with $u_h^0 = u_{0,h} \in V_h$ being a suitable approximation in V_h of the initial datum u_0 .

Let us limit ourselves to the homogeneous case, where $f = 0$ and $\varphi_h = 0$ (in this case $u_h^n \in V_h^{in}$ for every $n \geq 0$). Having set $v_h = u_h^{n+1}$ and using identities (13.8) and (13.26), we obtain, for each $n \geq 0$

$$\begin{aligned} \frac{1}{2\Delta t} \left(\|u_h^{n+1}\|_{L^2(\Omega)}^2 - \|u_h^n\|_{L^2(\Omega)}^2 \right) + \frac{1}{2} \int_{\partial\Omega \setminus \partial\Omega^{in}} \mathbf{a} \cdot \mathbf{n} (u_h^{n+1})^2 \, d\gamma + \\ \mu_0 \|u_h^{n+1}\|_{L^2(\Omega)}^2 \leq 0. \end{aligned}$$

For each $m \geq 1$, summing over n from 0 to $m-1$ we obtain

$$\begin{aligned} \|u_h^m\|_{L^2(\Omega)}^2 + 2\Delta t \left(\mu_0 \sum_{n=0}^m \|u_h^n\|_{L^2(\Omega)}^2 + \frac{1}{2} \sum_{n=0}^m \int_{\partial\Omega \setminus \partial\Omega^{in}} \mathbf{a} \cdot \mathbf{n} (u_h^n)^2 \, d\gamma \right) \\ \leq \|u_{0,h}\|_{L^2(\Omega)}^2. \end{aligned}$$

In particular, as $\mathbf{a} \cdot \mathbf{n} \geq 0$ on $\partial\Omega \setminus \partial\Omega^{in}$, we conclude that

$$\|u_h^m\|_{L^2(\Omega)} \leq \|u_{0,h}\|_{L^2(\Omega)} \quad \forall m \geq 0.$$

As expected, this method is strongly stable, with no condition on Δt and h . We now consider the discretization in time using the forward Euler method

$$\begin{cases} \frac{1}{\Delta t} \int_{\Omega} (u_h^{n+1} - u_h^n) v_h \, d\Omega + \int_{\Omega} \mathbf{a} \cdot \nabla u_h^n v_h \, d\Omega + \int_{\Omega} a_0^n u_h^n v_h \, d\Omega \\ = \int_{\Omega} f^n v_h \, d\Omega \quad \forall v_h \in V_h, \\ u_h^{n+1} = \varphi_h^{n+1} \text{ on } \partial\Omega^{in}. \end{cases} \quad (13.31)$$

We suppose again that $f = 0$, $\varphi = 0$ and that the condition (13.26) is verified. Moreover, we suppose that $\|\mathbf{a}\|_{L^\infty(\Omega)} < \infty$ and that, for each $t > 0$, $\|a_0\|_{L^\infty(\Omega)} < \infty$. Setting $v_h = u_h^n$, exploiting identity (13.5) and integrating the convective term by parts, we obtain

$$\begin{aligned} \frac{1}{2\Delta t} \left(\|u_h^{n+1}\|_{L^2(\Omega)}^2 - \|u_h^n\|_{L^2(\Omega)}^2 - \|u_h^{n+1} - u_h^n\|_{L^2(\Omega)}^2 \right) \\ + \int_{\partial\Omega \setminus \partial\Omega^{in}} \mathbf{a} \cdot \mathbf{n} (u_h^n)^2 \, d\gamma + \left(-\frac{1}{2} \operatorname{div}(\mathbf{a}) + a_0^n, (u_h^n)^2 \right) = 0, \end{aligned}$$

and then, after a few steps,

$$\begin{aligned} \|u_h^{n+1}\|_{L^2(\Omega)}^2 &+ 2\Delta t \int_{\partial\Omega \setminus \partial\Omega^{in}} \mathbf{a} \cdot \mathbf{n} (u_h^n)^2 d\gamma + 2\Delta t \mu_0 \|u_h^n\|_{L^2(\Omega)}^2 \\ &\leq \|u_h^n\|_{L^2(\Omega)}^2 + \|u_h^{n+1} - u_h^n\|_{L^2(\Omega)}^2. \end{aligned} \quad (13.32)$$

It is now necessary to control the term $\|u_h^{n+1} - u_h^n\|_{L^2(\Omega)}^2$. To this end, we set $v_h = u_h^{n+1} - u_h^n$ in (13.31) and obtain

$$\begin{aligned} \|u_h^{n+1} - u_h^n\|_{L^2(\Omega)}^2 &= -\Delta t (\mathbf{a} \nabla u_h^n, u_h^{n+1} - u_h^n) - \Delta t (a_0^n u_h^n, u_h^{n+1} - u_h^n) \\ &\leq \Delta t \|\mathbf{a}\|_{L^\infty(\Omega)} |(\nabla u_h^n, u_h^{n+1} - u_h^n)| + \Delta t \|a_0^n\|_{L^\infty(\Omega)} |(u_h^n, u_h^{n+1} - u_h^n)| \\ &\leq \Delta t \|\mathbf{a}\|_{L^\infty(\Omega)} \|\nabla u_h^n\|_{L^2(\Omega)} \|u_h^{n+1} - u_h^n\|_{L^2(\Omega)} + \\ &\quad \Delta t \|a_0^n\|_{L^\infty(\Omega)} \|u_h^n\|_{L^2(\Omega)} \|u_h^{n+1} - u_h^n\|_{L^2(\Omega)}. \end{aligned}$$

Using the inverse inequality (4.52), we obtain

$$\begin{aligned} \|u_h^{n+1} - u_h^n\|_{L^2(\Omega)}^2 &\leq \Delta t (C_I h^{-1} \|\mathbf{a}\|_{L^\infty(\Omega)} + \\ &\quad \|a_0^n\|_{L^\infty(\Omega)}) \|u_h^n\|_{L^2(\Omega)} \|u_h^{n+1} - u_h^n\|_{L^2(\Omega)}, \end{aligned}$$

and then

$$\|u_h^{n+1} - u_h^n\|_{L^2(\Omega)} \leq \Delta t (C_I h^{-1} \|\mathbf{a}\|_{L^\infty(\Omega)} + \|a_0^n\|_{L^\infty(\Omega)}) \|u_h^n\|_{L^2(\Omega)}.$$

Using such result to find an upper bound to the term in (13.32), we have

$$\begin{aligned} &\|u_h^{n+1}\|_{L^2(\Omega)}^2 + 2\Delta t \int_{\partial\Omega \setminus \partial\Omega^{in}} \mathbf{a} \cdot \mathbf{n} (u_h^n)^2 d\Omega + \\ &\Delta t \left[2\mu_0 - \Delta t (C_I h^{-1} \|\mathbf{a}\|_{L^\infty(\Omega)} + \|a_0^n\|_{L^\infty(\Omega)})^2 \right] \|u_h^n\|_{L^2(\Omega)}^2 \\ &\leq \|u_h^n\|_{L^2(\Omega)}^2. \end{aligned}$$

The integral on $\partial\Omega \setminus \partial\Omega^{in}$ is positive because of the hypotheses on the boundary conditions; hence, if

$$\Delta t \leq \frac{2\mu_0}{(C_I h^{-1} \|\mathbf{a}\|_{L^\infty(\Omega)} + \|a_0^n\|_{L^\infty(\Omega)})^2} \quad (13.33)$$

we have $\|u_h^{n+1}\|_{L^2(\Omega)} \leq \|u_h^n\|_{L^2(\Omega)}$, that is the scheme is strongly stable. Note that the stability condition (13.33) is of parabolic type, similar to the one found in (12.31) for the case of finite difference discretizations.

Remark 13.3 In the case where \mathbf{a} is constant and $a_0 = 0$ we have that $\mu_0 = 0$ and the stability condition (13.33) can never be satisfied by a positive Δt . Thus, the result in (13.33) does not contradict the one we have previously found for the forward Euler scheme. •

13.4 Discontinuous finite elements

An alternative approach to the one adopted so far is based on the use of *discontinuous* finite elements. The resulting method is called the discontinuous Galerkin method (DG in short). This choice is motivated by the fact that, as we have previously observed, the solutions of (even linear) hyperbolic problems can be discontinuous.

For a given mesh \mathcal{T}_h of Ω , the space of discontinuous finite elements is

$$W_h = Y_h^r = \{v_h \in L^2(\Omega) \mid v_{h|K} \in \mathbb{P}_r, \forall K \in \mathcal{T}_h\}, \quad (13.34)$$

that is the space of piecewise polynomial functions of degree less than or equal to r , with $r \geq 0$, which are not necessarily continuous across the finite element interfaces.

13.4.1 The one-dimensional case

In the case of the one-dimensional problem (12.3), the DG finite element method takes the following form: $\forall t > 0$, find a function $u_h = u_h(t) \in W_h$ such that

$$\begin{aligned} & \int_{\alpha}^{\beta} \frac{\partial u_h(t)}{\partial t} v_h \, dx \\ & + \sum_{i=0}^{m-1} \left[\int_{x_i}^{x_{i+1}} \left(a \frac{\partial u_h(t)}{\partial x} + a_0 u_h(t) \right) v_h \, dx \right. \\ & \quad \left. + a(x_i)(u_h^+(t) - U_h^-(t))(x_i)v_h^+(x_i) \right] \\ & = \int_{\alpha}^{\beta} f(t)v_h \, dx \quad \forall v_h \in W_h, \end{aligned} \quad (13.35)$$

where we have supposed that $a(x)$ is a continuous function. We have set, for each $t > 0$,

$$U_h^-(t)(x_i) = \begin{cases} u_h^-(t)(x_i), & i = 1, \dots, m-1, \\ \varphi_h(t)(x_0), & \end{cases} \quad (13.36)$$

where $\{x_i, i = 0, \dots, m\}$ are the nodes, $x_0 = \alpha, x_m = \beta$, h is the maximal distance between two consecutive nodes, $v_h^+(x_i)$ denotes the right limit of v_h in x_i , $v_h^-(x_i)$ the left one. For simplicity of notation, the dependence of u_h and f on t will often be understood when this does not yield to ambiguities.

We now derive a stability estimate for the solution u_h of (13.35), supposing, for simplicity, that the forcing term f be identically null. Having then chosen $v_h = u_h$ in

(13.35), we have (setting $\Omega = (\alpha, \beta)$)

$$\frac{1}{2} \frac{d}{dt} \|u_h\|_{L^2(\Omega)}^2 + \sum_{i=0}^{m-1} \left[\int_{x_i}^{x_{i+1}} \left(\frac{a}{2} \frac{\partial}{\partial x} (u_h)^2 + a_0 u_h^2 \right) dx + a(x_i) (u_h^+ - U_h^-)(x_i) u_h^+(x_i) \right] = 0.$$

Now, integrating the convective term by parts, we have

$$\begin{aligned} & \frac{1}{2} \frac{d}{dt} \|u_h\|_{L^2(\Omega)}^2 + \sum_{i=0}^{m-1} \int_{x_i}^{x_{i+1}} \left(a_0 - \frac{\partial}{\partial x} \left(\frac{a}{2} \right) \right) u_h^2 dx + \\ & \sum_{i=0}^{m-1} \left[\frac{a}{2} (x_{i+1}) (u_h^-(x_{i+1}))^2 + \frac{a}{2} (x_i) (u_h^+(x_i))^2 - a(x_i) U_h^-(x_i) u_h^+(x_i) \right] = 0. \end{aligned} \quad (13.37)$$

Isolating the contribution associated to node x_0 and exploiting definition (13.36), we can rewrite the second sum in the previous equation as

$$\begin{aligned} & \sum_{i=0}^{m-1} \left[\frac{a}{2} (x_{i+1}) (u_h^-(x_{i+1}))^2 + \frac{a}{2} (x_i) (u_h^+(x_i))^2 - a(x_i) U_h^-(x_i) u_h^+(x_i) \right] \\ & = \frac{a}{2} (x_0) (u_h^+(x_0))^2 - a(x_0) \varphi_h(x_0) u_h^+(x_0) + \frac{a}{2} (x_m) (u_h^-(x_m))^2 + \\ & \sum_{i=1}^{m-1} \left[\frac{a}{2} (x_i) \left((u_h^-(x_i))^2 + (u_h^+(x_i))^2 \right) - a(x_i) u_h^-(x_i) u_h^+(x_i) \right] \\ & = \frac{a}{2} (x_0) (u_h^+(x_0))^2 - a(x_0) \varphi_h(x_0) u_h^+(x_0) + \\ & \frac{a}{2} (x_m) (u_h^-(x_m))^2 + \sum_{i=1}^{m-1} \frac{a}{2} (x_i) [u_h(x_i)]^2, \end{aligned} \quad (13.38)$$

having denoted by $[u_h(x_i)] = u_h^+(x_i) - u_h^-(x_i)$ the jump of function u_h at node x_i . We now suppose, analogously to the multi-dimensional case (see (13.26)), that

$$\exists \gamma \geq 0 \text{ s.t. } a_0 - \frac{\partial}{\partial x} \left(\frac{a}{2} \right) \geq \gamma. \quad (13.39)$$

Returning to (13.37) and using the relation (13.38) and the Cauchy-Schwarz and Young inequalities, we have

$$\begin{aligned} & \frac{1}{2} \frac{d}{dt} \|u_h\|_{L^2(\Omega)}^2 + \gamma \|u_h\|_{L^2(\Omega)}^2 + \sum_{i=1}^{m-1} \frac{a}{2} (x_i) [u_h(x_i)]^2 + \frac{a}{2} (x_0) (u_h^+(x_0))^2 + \\ & \frac{a}{2} (x_m) (u_h^-(x_m))^2 = a(x_0) \varphi_h(x_0) u_h^+(x_0) \leq \frac{a}{2} (x_0) \varphi_h^2(x_0) + \frac{a}{2} (x_0) (u_h^+(x_0))^2, \end{aligned}$$

that is, integrating with respect to time as well, $\forall t > 0$,

$$\begin{aligned} \|u_h(t)\|_{L^2(\Omega)}^2 + 2\gamma \int_0^t \|u_h(t)\|_{L^2(\Omega)}^2 dt + \sum_{i=1}^{m-1} a(x_i) \int_0^t [u_h(x_i, t)]^2 dt + \\ a(x_m) (u_h^-(x_m))^2 \leq \|u_{0,h}\|_{L^2(\Omega)}^2 + a(x_0) \int_0^t \varphi_h^2(x_0, t) dt. \end{aligned} \quad (13.40)$$

Such estimate represents the desired stability result.

Note that, in case the forcing term is no longer null, we can replicate the previous analysis by suitably using the Gronwall Lemma 2.2 to handle the contribution of f . This would lead to an estimate similar to (13.40), however this time the right-hand side of the inequality would become

$$e^t \left(\|u_{0,h}\|_{L^2(\Omega)}^2 + a(x_0) \int_0^t \varphi_h^2(x_0, t) dt + \int_0^t (f(\tau))^2 d\tau \right). \quad (13.41)$$

In the case where the γ constant in inequality (13.39) is strictly positive, we could avoid using the Gronwall lemma, getting an estimate such as (13.40) where in the first term 2γ is replaced by γ , while the second term takes the form (13.41), however without the exponential e^t .

Because of the discontinuity of test functions, (13.35) can be rewritten in an equivalent way as follows, $\forall i = 0, \dots, m-1$,

$$\begin{aligned} \int_{x_i}^{x_{i+1}} \left(\frac{\partial u_h}{\partial t} + a \frac{\partial u_h}{\partial x} + a_0 u_h \right) v_h dx + a(u_h^+ - U_h^-)(x_i) v_h^+(x_i) \\ = \int_{\alpha}^{\beta} f v_h dx \quad \forall v_h \in \mathbb{P}_r(I_i), \end{aligned} \quad (13.42)$$

with $I_i = [x_i, x_{i+1}]$. In other terms, the approximation via discontinuous finite elements yields to element-wise “independent” relations, the only term connecting an element and its neighbors is the jump term $(u_h^+ - U_h^-)$ that can also be interpreted as the attribution of the boundary datum on the inflow boundary of the element under exam.

We then have a set of small problems to be solved in each element, precisely $r+1$ equations for each interval $[x_i, x_{i+1}]$. Let us write them in compact form as

$$M_h \dot{\mathbf{u}}_h(t) + L_h \mathbf{u}_h(t) = \mathbf{f}_h(t) \quad \forall t > 0, \quad \mathbf{u}_h(0) = \mathbf{u}_{0,h}, \quad (13.43)$$

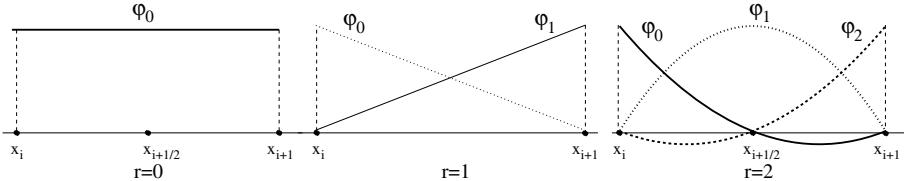


Fig. 13.7. The Lagrange bases for $r = 0$, $r = 1$ and $r = 2$

M_h being the mass matrix, L_h the matrix associated to the bilinear form and to the jump relation, \mathbf{f}_h the known term:

$$(M_h)_{pq} = \int_{x_i}^{x_{i+1}} \varphi_p \varphi_q \, dx, \quad (L_h)_{pq} = \int_{x_i}^{x_{i+1}} (a\varphi_{q,x} + a_0\varphi_q) \varphi_p \, dx + (a\varphi_q \varphi_p)(x_i),$$

$$(\mathbf{f}_h)_p = \int_{x_i}^{x_{i+1}} f \varphi_p \, dx + aU_h^-(x_i)\varphi_p(x_i), \quad p, q = 0, \dots, r.$$

We have denoted by $\{\varphi_q, q = 0, \dots, r\}$ a basis for $\mathbb{P}_r([x_i, x_{i+1}])$ and by $\mathbf{u}_h(t)$ the coefficients of $u_h(x, t)|_{[x_i, x_{i+1}]}$ in the development with respect to the basis $\{\varphi_q\}$. If we take the Lagrange basis we will have, for instance, the functions reported in Fig. 13.7 (for the $r = 0$, $r = 1$ and $r = 2$ cases) and the values of $\{\mathbf{u}_h(t)\}$ are the ones taken by $u_h(t)$ at nodes ($x_{i+1/2}$ for $r = 0$, x_i and x_{i+1} for $r = 1$, x_i , $x_{i+1/2}$ and x_{i+1} for $r = 2$). Note that all the previous functions are identically null outside the interval $[x_i, x_{i+1}]$. Also, in the case of discontinuous finite elements it is perfectly acceptable to use polynomials of degree $r = 0$, in which case the transport term $a \frac{\partial u_h}{\partial x}$ will provide a null contribution on each element.

Aiming at diagonalizing the mass matrix, it can be interesting to use as a basis for $\mathbb{P}_r([x_i, x_{i+1}])$ the Legendre polynomials $\varphi_q(x) = L_q(2(x - x_i)/h_i)$, $h_i = x_i - x_i$ and $\{L_q, q = 0, 1, \dots\}$ being the orthogonal Legendre polynomials defined over the interval $[-1, 1]$, that we have introduced in Sec. 10.2.2. Indeed, in such a way we obtain $(M_h)_{pq} = \frac{h_i}{2p+1} \delta_{pq}$, $p, q = 0, \dots, r$. Obviously, in this case the unknown values $\{\mathbf{u}_h(t)\}$ will no longer be interpretable as nodal values of $u_h(t)$, but rather as the Legendre coefficients of the expansion of $u_h(t)$ with respect to the new basis.

The diagonalization of the mass matrix turns out to be particularly interesting when we use explicit time advancing schemes (such as e.g. second- and third-order Runge-Kutta schemes, introduced in Chap. 14). In this case, indeed, we will have a fully explicit problem to solve on each small interval.

For illustrative purposes, we present below some numerical results obtained for problem

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0, & x \in (-5, 5), \quad t > 0, \\ u(-5, t) = 0, & t > 0, \end{cases} \quad (13.44)$$

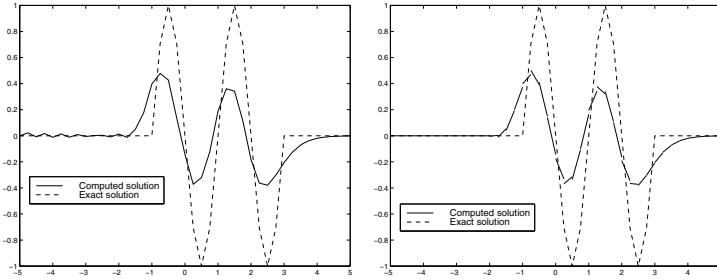


Fig. 13.8. Solution at time $t = 1$ of problem (13.44) with $\phi_k = \pi/2$, $h = 0.25$, obtained using continuous (left) and discontinuous (right) linear finite elements and backward Euler time discretization

using the initial condition

$$u(x, 0) = \begin{cases} \sin(\pi x), & x \in (-2, 2), \\ 0 & \text{otherwise.} \end{cases} \quad (13.45)$$

The problem has been discretized using linear finite elements in space, both continuous and discontinuous. For the temporal discretization, we have used the backward Euler scheme in both cases. We have chosen $h = 0.25$ and a time step $\Delta t = h$; for such value of h the phase number associated to the sinusoidal wave is $\phi_k = \pi/2$.

In Fig. 13.8 we report the numerical solution at time $t = 1$ together with the corresponding exact solution. We can note the strong numerical diffusion of the scheme that, however, denotes small oscillations in the posterior part in the case of continuous elements. Furthermore, we can observe that the numerical solution obtained using discontinuous finite elements, although being discontinuous, it no longer features an oscillatory behavior in the posterior part.

Let us now consider the following problem

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0, & x \in (0, 1), \quad t > 0, \\ u(0, t) = 1, & t > 0, \\ u(x, 0) = 0, & x \in [0, 1], \end{cases} \quad (13.46)$$

which represents the transport of a discontinuity entering the domain. We have considered continuous linear finite elements, with both strong and weak treatment of the boundary conditions, as well as discontinuous linear finite elements. This time as well, we have used the backward Euler method for the temporal discretization. The grid-size is $h = 0.025$ and the time step is $\Delta t = h$.

The results at time $t = 0.5$ are represented in Fig. 13.9. We can note how the Dirichlet datum is well represented also by schemes with weak boundary treatment. To this end, for the case of continuous finite elements with weak boundary treatment, we have computed the behavior of $|u_h(0) - u(0)|$ for $t = 0.1$ for several values of h , Δt being constant. We can note a linear convergence to zero with respect to h .

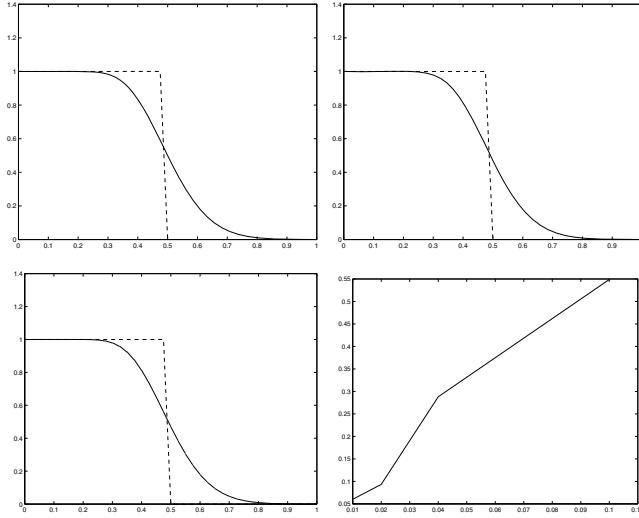


Fig. 13.9. Solution to problem (13.46) for $t = 0.5$ with $h = 0.025$ obtained using continuous linear finite elements and strong (top left) and weak (top right) treatment of the boundary Dirichlet condition, while discontinuous elements in space have been used in the bottom left case. Finally, we show in the bottom left the behavior of $|u_h(0) - u(0)|$ as a function of h for $t = 0.1$, in weak treatment of the Dirichlet condition

13.4.2 The multi-dimensional case

Let us now consider the case of the multi-dimensional case (13.22). Let W_h be the space of discontinuous piecewise polynomials of degree r on each element $K \in \mathcal{T}_h$, introduced in (13.34). The discontinuous Galerkin (DG) finite element semi-discretization of problem (13.22) becomes: for each $t > 0$ find $u_h(t) \in W_h$ such that

$$\begin{aligned} & \int_{\Omega} \frac{\partial u_h(t)}{\partial t} v_h d\Omega + \sum_{K \in \mathcal{T}_h} \left[a_K(u_h(t), v_h) - \int_{\partial K^{in}} \mathbf{a} \cdot \mathbf{n}_K [u_h(t)] v_h^+ d\gamma \right] \\ &= \int_{\Omega} f(t) v_h d\Omega \quad \forall v_h \in W_h, \end{aligned} \tag{13.47}$$

with $u_h(0) = u_{0,h}$, where \mathbf{n}_K denotes the outward normal unit vector on ∂K , and

$$\partial K^{in} = \{\mathbf{x} \in \partial K : \mathbf{a}(\mathbf{x}) \cdot \mathbf{n}_K(\mathbf{x}) < 0\}.$$

The bilinear form a_K is defined in the following way

$$a_K(u, v) = \int_K (\mathbf{a} \cdot \nabla u v + a_0 u v) d\mathbf{x},$$

while

$$[u_h(\mathbf{x})] = \begin{cases} u_h^+(\mathbf{x}) - u_h^-(\mathbf{x}), & \mathbf{x} \notin \partial\Omega^{in}, \\ u_h^+(\mathbf{x}) - \varphi_h(\mathbf{x}), & \mathbf{x} \in \partial\Omega^{in}, \end{cases}$$

$\partial\Omega^{in}$ being the inflow boundary (13.23) and with

$$u_h^\pm(\mathbf{x}) = \lim_{s \rightarrow 0^\pm} u_h(\mathbf{x} + s\mathbf{a}), \quad \mathbf{x} \in \partial K.$$

For each $t > 0$, the stability estimate obtained for problem (13.47) is (thanks to the hypothesis (13.26))

$$\begin{aligned} \|u_h(t)\|_{L^2(\Omega)}^2 &+ \int_0^t \left(\mu_0 \|u_h(\tau)\|_{L^2(\Omega)}^2 + \sum_{K \in \mathcal{T}_h} \int_{\partial K^{in}} |\mathbf{a} \cdot \mathbf{n}_K| [u_h(\tau)]^2 \right) d\tau \\ &\leq C \left[\|u_{0,h}\|_{L^2(\Omega)}^2 + \int_0^t \left(\|f(\tau)\|_{L^2(\Omega)}^2 + |\varphi_h|_{\mathbf{a}, \partial\Omega^{in}}^2 \right) d\tau \right], \end{aligned}$$

having introduced, for each subset Γ of $\partial\Omega$ of positive measure, the seminorm

$$|v|_{\mathbf{a}, \Gamma} = \left(\int_{\Gamma} |\mathbf{a} \cdot \mathbf{n}| v^2 d\gamma \right)^{1/2}.$$

Supposing for simplicity that $f = 0$, $\varphi = 0$, and that $u_0 \in H^{r+1}(\Omega)$, we can prove the following a priori error estimate

$$\begin{aligned} \max_{t \in [0, T]} \|u(t) - u_h(t)\|_{L^2(\Omega)} &+ \left(\int_0^T \sum_{K \in \mathcal{T}_h} \int_{\partial K^{in}} |\mathbf{a} \cdot \mathbf{n}_K| [u(t) - u_h(t)]^2 dt \right)^{\frac{1}{2}} \\ &\leq Ch^{r+1/2} \|u_0\|_{H^{r+1}(\Omega)}. \end{aligned} \tag{13.48}$$

For the proofs, we refer to [QV94, Chap. 14], [Joh87], and to the references cited thereby.

Other formulations are possible, based on different forms of stabilization. Let us consider a diffusion and reaction problem such as (13.22) but written in conservative form

$$\frac{\partial u}{\partial t} + \operatorname{div}(\mathbf{a}u) + a_0 u = f, \quad \mathbf{x} \in \Omega, \quad t > 0. \tag{13.49}$$

Having now set

$$a_K(u_h, v_h) = \int_K (-u_h(\mathbf{a} \cdot \nabla v_h) + a_0 u_h v_h) d\mathbf{x},$$

we consider the following approximation: for each $t > 0$, find $u_h(t) \in W_h$ such that, $\forall v_h \in W_h$,

$$\begin{aligned} & \int_{\Omega} \frac{\partial u_h(t)}{\partial t} v_h d\Omega + \sum_{K \in \mathcal{T}_h} a_K(u_h(t), v_h) + \sum_{e \not\subset \partial\Omega^{in}} \int_e \{\mathbf{a} u_h(t)\} [\![v_h]\!] d\gamma \\ & + \sum_{e \not\subset \partial\Omega} \int_e c_e(\gamma) [\![u_h(t)]\!] [\![v_h]\!] d\gamma \\ & = \int_{\Omega} f(t) v_h d\Omega - \sum_{e \subset \partial\Omega^{in}} \int_e (\mathbf{a} \cdot \mathbf{n}) \varphi(t) v_h d\gamma. \end{aligned} \quad (13.50)$$

The notations are the following: we denote by e any side of the grid \mathcal{T}_h shared by two triangles, say K_1 and K_2 . For each scalar function ψ , piecewise regular on the mesh, with $\psi^i = \psi|_{K_i}$, we have defined its jump on e as follows:

$$[\![\psi]\!] = \psi^1 \mathbf{n}_1 + \psi^2 \mathbf{n}_2,$$

\mathbf{n}_i being the outward normal unit vector to element K_i . Instead, if $\boldsymbol{\sigma}$ is a vector function, then its average on e is defined as

$$\{\boldsymbol{\sigma}\} = \frac{1}{2} (\boldsymbol{\sigma}^1 + \boldsymbol{\sigma}^2).$$

Note that the jump $[\![\psi]\!]$ through e of a scalar function ψ is a vector parallel to the normal to e .

These definitions do not depend on the ordering of the elements.

If e is a side belonging to the boundary $\partial\Omega$, then

$$[\![\psi]\!] = \psi \mathbf{n}, \quad \{\boldsymbol{\sigma}\} = \boldsymbol{\sigma}.$$

Concerning $c_e(\gamma)$, this is a non-negative function which will typically be chosen to be constant on each side. Choosing, for instance, $c_e = |\mathbf{a} \cdot \mathbf{n}|/2$ on each internal side, $c_e = -\mathbf{a} \cdot \mathbf{n}/2$ on $\partial\Omega^{in}$, $c_e = \mathbf{a} \cdot \mathbf{n}/2$ on $\partial\Omega^{out}$, the formulation in (13.50) is reduced to the standard upwind formulation

$$\begin{aligned} & \int_{\Omega} \frac{\partial u_h(t)}{\partial t} v_h d\Omega + \sum_{K \in \mathcal{T}_h} a_K(u_h(t), v_h) + \sum_{e \not\subset \partial\Omega^{in}} \int_e \{\mathbf{a} u_h(t)\}_{up} [\![v_h]\!] d\gamma \\ & = \int_{\Omega} f(t) v_h d\Omega - \sum_{e \subset \partial\Omega^{in}} \int_e (\mathbf{a} \cdot \mathbf{n}) \varphi(t) v_h d\gamma \quad \forall v_h \in W_h. \end{aligned} \quad (13.51)$$

Here $\{\mathbf{a} u_h\}_{up}$ denotes the upwind value of $\mathbf{a} u_h$, that is coincides with $\mathbf{a} u_h^1$ if $\mathbf{a} \cdot \mathbf{n}_1 > 0$, with $\mathbf{a} u_h^2$ if $\mathbf{a} \cdot \mathbf{n}_1 < 0$, and finally with $\mathbf{a} \{u_h\}$ if $\mathbf{a} \cdot \mathbf{n}_1 = 0$. Finally, if \mathbf{a} is constant (or divergence free), then $\operatorname{div}(\mathbf{a} u_h) = \mathbf{a} \cdot \nabla u_h$ and (13.51) coincides with (13.47). The formulation (13.50) is called discontinuous Galerkin method with *jump stabilization*. The latter is stable if $c_e \geq \theta_0 |\mathbf{a} \cdot \mathbf{n}_e|$ (for a suitable $\theta_0 > 0$) for each internal side e ,

and also convergent with optimal order. Indeed, in the case of the stationary problem it can be proven that

$$\|u - u_h\|_{L^2(\Omega)}^2 + \sum_{e \in \mathcal{T}_h} \|\sqrt{c_e} [u - u_h]\|_{L^2(e)}^2 \leq C h^{2r+1} \|u\|_{H^{r+1}(\Omega)}^2.$$

For the proof and for other formulations with jump stabilization, including the case of advection-diffusion equations, we refer the reader to [BMS04].

13.5 Approximation using spectral methods

In this section, we will briefly discuss the approximation of hyperbolic problems with spectral methods. For simplicity, we will limit our discussion to one-dimensional problems. We will first treat the G-NI approximation in a single interval, then the SEM approximation corresponding to a decomposition in sub-intervals where we use discontinuous polynomials when we move from an interval to its neighbors. This provides a generalization of discontinuous finite elements, in the case where we consider polynomials of “high” degree on each element, and the integrals on each element are approximated using the GLL integration formula (10.18).

13.5.1 The G-NI method in a single interval

Let us consider the first-order hyperbolic transport-reaction problem (12.3) and let us suppose that $(\alpha, \beta) = (-1, 1)$. Then we approximate in space by a spectral collocation method, with strong imposition of the boundary conditions. Having denoted by $\{x_0 = -1, x_1, \dots, x_N = 1\}$ the GLL nodes introduced in Sec. 10.2.3, the semi-discretized problem is:

for each $t > 0$, find $u_N(t) \in \mathbb{Q}_N$ (the space of polynomials (10.1)) such that

$$\begin{cases} \left(\frac{\partial u_N}{\partial t} + a \frac{\partial u_N}{\partial x} + a_0 u_N \right)(x_j, t) = f(x_j, t), & j = 1, \dots, N, \\ u_N(-1, t) = \varphi(t), \\ u_N(x_j, 0) = u_0(x_j), & j = 0, \dots, N. \end{cases} \quad (13.52)$$

Suitably using the discrete GLL scalar product defined in (10.25), the G-NI approximation of problem (13.52) becomes: for each $t > 0$, find $u_N(t) \in \mathbb{Q}_N$ such that

$$\begin{cases} \left(\frac{\partial u_N(t)}{\partial t}, v_N \right)_N + \left(a \frac{\partial u_N(t)}{\partial x}, v_N \right)_N + (a_0 u_N(t), v_N)_N = (f(t), v_N)_N \\ u_N(-1, t) = \varphi(t), \\ u_N(x, 0) = u_{0,N}, \end{cases} \quad \forall v_N \in \mathbb{Q}_N^-, \quad (13.53)$$

where $u_{0,N} \in \mathbb{Q}_N$ is a suitable approximation of u_0 , and having set $\mathbb{Q}_N^- = \{v_N \in \mathbb{Q}_N : v_N(-1) = 0\}$. At the inflow, the solution u_N then satisfies the imposed condition at each time $t > 0$, while the test functions vanish.

In fact, the solutions of problems (13.52) and (13.53) coincide if $u_{0,N}$ in (13.53) is chosen as the interpolated $\Pi_N^{GLL} u_0$. To prove this, it is sufficient to choose in (13.53) v_N coinciding with the characteristic polynomial ψ_j (defined in (10.12), (10.13)) associated to the GLL node x_j , for each $j = 1, \dots, N$.

Let us now derive a stability estimate for the formulation (13.53) in the norm (10.53) induced from the discrete scalar product (10.25). For simplicity, we choose a homogeneous inflow datum, that is $\varphi(t) = 0$, for each t , and a and a_0 constant. Having chosen, for each $t > 0$, $v_N = u_N(t)$, we obtain

$$\frac{1}{2} \frac{\partial}{\partial t} \|u_N(t)\|_N^2 + \frac{a}{2} \int_{-1}^1 \frac{\partial u_N^2(t)}{\partial x} dx + a_0 \|u_N(t)\|_N^2 = (f(t), u_N(t))_N.$$

Suitably rewriting the convective term, integrating with respect to time and using the Young inequality, we have

$$\begin{aligned} \|u_N(t)\|_N^2 &+ a \int_0^t (u_N(1, \tau))^2 d\tau + 2a_0 \int_0^t \|u_N(\tau)\|_N^2 d\tau \\ &= \|u_{0,N}\|_N^2 + 2 \int_0^t (f(\tau), u_N(\tau))_N d\tau \\ &\leq \|u_{0,N}\|_N^2 + a_0 \int_0^t \|u_N(\tau)\|_N^2 d\tau + \frac{1}{a_0} \int_0^t \|f(\tau)\|_N^2 d\tau, \end{aligned}$$

that is

$$\begin{aligned} \|u_N(t)\|_N^2 &+ a \int_0^t (u_N(1, \tau))^2 d\tau + a_0 \int_0^t \|u_N(\tau)\|_N^2 d\tau \\ &\leq \|u_{0,N}\|_N^2 + \frac{1}{a_0} \int_0^t \|f(\tau)\|_N^2 d\tau. \end{aligned} \tag{13.54}$$

The norm of the initial data can be bounded as follows

$$\|u_{0,N}\|_N^2 \leq \|u_{0,N}\|_{L^\infty(-1,1)}^2 \left(\sum_{i=0}^N \alpha_i \right) = 2 \|u_{0,N}\|_{L^\infty(-1,1)}^2,$$

and a similar bound holds for $\|f(\tau)\|_N^2$ provided that f be a continuous function. Hence, reverting to (13.54) and using inequality (10.54) to bound the norms of the

left-hand side, we deduce

$$\begin{aligned} \|u_N(t)\|_{L^2(-1,1)}^2 &+ a \int_0^t (u_N(1,\tau))^2 d\tau + a_0 \int_0^t \|u_N(\tau)\|_{L^2(-1,1)}^2 d\tau \\ &\leq 2 \|u_{0,N}\|_{L^\infty(-1,1)}^2 + \frac{2}{a_0} \int_0^t \|f(\tau)\|_{L^2(-1,1)}^2 d\tau. \end{aligned}$$

The reinterpretation of the G-NI method as a collocation method is less immediate in the case where the convective term a is not constant and we start from a conservative formulation of the differential equation in (13.52), that is the second term on the left-hand side is replaced by $\partial(au)/\partial x$. In such case, we can show again that the G-NI approximation is equivalent to the collocation approximation where the convective term is replaced by $\partial(\Pi_N^{GLL}(au_N))/\partial x$, i.e. by the interpolation derivative (10.40).

Also in the case of a G-NI approximation, we can resort to a weak imposition of the boundary conditions. Such approach is more flexible than the one considered above and more suitable for the generalization to multi-dimensional problems or systems of equations. As we have seen in the previous section, the starting point for a weak imposition of boundary conditions is a suitable integration by parts of the transport terms. Referring to the one-dimensional problem (13.52), we have (if a is constant)

$$\begin{aligned} \int_{-1}^1 a \frac{\partial u(t)}{\partial x} v dx &= - \int_{-1}^1 a u(t) \frac{\partial v}{\partial x} dx + [a u(t) v]_{-1}^1 \\ &= - \int_{-1}^1 a u(t) \frac{\partial v}{\partial x} dx + a u(1,t) v(1) - a \varphi(t) v(-1). \end{aligned}$$

Thanks to the above identity, we can immediately formulate the G-NI approximation to problem (13.52) with a weak treatment of boundary conditions:
for each $t > 0$, find $u_N(t) \in \mathbb{Q}_N$ such that

$$\begin{aligned} \left(\frac{\partial u_N(t)}{\partial t}, v_N \right)_N - \left(a u_N(t), \frac{\partial v_N}{\partial x} \right)_N + \left(a_0 u_N(t), v_N \right)_N \\ + a u_N(1,t) v_N(1) = (f(t), v_N)_N + a \varphi(t) v_N(-1) \quad \forall v_N \in \mathbb{Q}_N, \end{aligned} \tag{13.55}$$

with $u_N(x, 0) = u_{0,N}(x)$. We note that both the solution u_N and the test function v_N are free at the boundary.

An equivalent formulation to (13.55) is obtained by suitably counter-integrating the convective term by parts:

for each $t > 0$, find $u_N(t) \in \mathbb{Q}_N$ such that

$$\begin{aligned} \left(\frac{\partial u_N(t)}{\partial t}, v_N \right)_N + \left(a \frac{\partial u_N(t)}{\partial x}, v_N \right)_N + \left(a_0 u_N(t), v_N \right)_N \\ + a (u_N(-1,t) - \varphi(t)) v_N(-1) = (f, v_N)_N \quad \forall v_N \in \mathbb{Q}_N. \end{aligned} \tag{13.56}$$

It is now possible to reinterpret such weak formulation as a suitable collocation method. To this end, it is sufficient to choose in (13.56) the test function v_N coinciding with the characteristic polynomials (10.12), (10.13) associated to the GLL nodes. Considering first the internal and outflow nodes, and choosing therefore $v_N = \psi_i$, with $i = 1, \dots, N$, we have

$$\left(\frac{\partial u_N}{\partial t} + a \frac{\partial u_N}{\partial x} + a_0 u_N \right)(x_i, t) = f(x_i, t), \quad (13.57)$$

having previously simplified the weight α_i common to all the terms of the equation. On the other hand, by choosing $v_N = \psi_0$ we obtain the following relation at the inflow node

$$\begin{aligned} & \left(\frac{\partial u_N}{\partial t} + a \frac{\partial u_N}{\partial x} + a_0 u_N \right)(-1, t) \\ & + \frac{1}{\alpha_0} a (u_N(-1, t) - \varphi(t)) = f(-1, t), \end{aligned} \quad (13.58)$$

$\alpha_0 = 2/(N^2 + N)$ being the GLL weight associated to node $x_0 = -1$. From equations (13.57) and (13.58) it then follows that a reformulation in terms of collocation is possible at all the GLL nodes except for the inflow node, for which we find the relation

$$a (u_N(-1, t) - \varphi(t)) = \alpha_0 \left(f - \frac{\partial u_N}{\partial t} - a \frac{\partial u_N}{\partial x} - a_0 u_N \right)(-1, t). \quad (13.59)$$

The latter can be interpreted as the fulfillment of the boundary condition of the differential problem (13.52) up to the residue associated to the u_N approximation. Such condition is therefore satisfied exactly only at the limit, for $N \rightarrow \infty$ (i.e. in a natural way).

In accordance with what we previously noted, the formulation (13.56) would be complicated for instance in case of a non-constant convective field a . Indeed,

$$-\left(a u_N(t), \frac{\partial v_N}{\partial x} \right)_N = \left(a \frac{\partial u_N(t)}{\partial x}, v_N \right)_N - a u_N(1, t) v_N(1) + a \varphi(t) v_N(-1),$$

would not be true as, in this case, the product $a u_N(t) \frac{\partial v_N}{\partial x}$ no longer identifies a polynomial of degree $2N - 1$, so the exactness of the numerical integration formula would not hold in this case. It is therefore necessary to apply the interpolation operator Π_N^{GLL} , introduced in Sec. 10.2.3, before counter-integrating by parts, yielding

$$\begin{aligned} & -\left(a u_N(t), \frac{\partial v_N}{\partial x} \right)_N = -\left(\Pi_N^{GLL}(a u_N(t)), \frac{\partial v_N}{\partial x} \right)_N \\ & = -\left(\Pi_N^{GLL}(a u_N(t)), \frac{\partial v_N}{\partial x} \right) \\ & = \left(\frac{\partial \Pi_N^{GLL}(a u_N(t))}{\partial x}, v_N \right) - [(a u_N(t)) v_N]_{-1}^1. \end{aligned}$$

In this case, the formulation (13.56) then becomes:

for each $t > 0$, find $u_N(t) \in \mathbb{Q}_N$ such that

$$\begin{aligned} & \left(\frac{\partial u_N(t)}{\partial t}, v_N \right)_N + \left(\frac{\partial \Pi_N^{GLL}(a u_N(t))}{\partial x}, v_N \right)_N + (a_0 u_N(t), v_N)_N \\ & + a(t) (u_N(-1, t) - \varphi(t)) v_N(-1) = (f(t), v_N)_N \quad \forall v_N \in \mathbb{Q}_N, \end{aligned} \quad (13.60)$$

with $u_N(x, 0) = u_{0,N}(x)$. Also the collocation reinterpretation of formulation (13.56), represented by relations (13.57) and (13.59), will need to be modified with the introduction of the interpolation operator Π_N^{GLL} (that is by replacing the exact derivative with the interpolation derivative). Precisely, we obtain

$$\left(\frac{\partial u_N}{\partial t} + \frac{\partial \Pi_N^{GLL}(a u_N)}{\partial x} + a_0 u_N \right)(x_i, t) = f(x_i, t),$$

for $i = 1, \dots, N$, and

$$a(-1) (u_N(-1, t) - \varphi(t)) = \alpha_0 \left(f - \frac{\partial u_N}{\partial t} - \frac{\partial \Pi_N^{GLL}(a u_N)}{\partial x} - a_0 u_N \right)(-1, t),$$

at the inflow node $x = -1$.

13.5.2 The DG-SEM-NI method

As anticipated, we will introduce in this section an approximation based on a partition in sub-intervals, in each of which the G-NI method is used. Moreover, the solution will be discontinuous between an interval and its neighbors. This explains the DG (*discontinuous Galerkin*), SEM (*spectral element method*), NI (*numerical integration*) acronym.

Let us reconsider problem (13.52) on the generic interval (α, β) . On the latter, we introduce a partition in M subintervals $\Omega_m = (\bar{x}_{m-1}, \bar{x}_m)$ with $m = 1, \dots, M$. Let

$$W_{N,M} = \{v \in L^2(\alpha, \beta) : v|_{\Omega_m} \in \mathbb{Q}_N, \forall m = 1, \dots, M\}$$

be the space of piecewise polynomials of degree $N (\geq 1)$ on each sub-interval. We observe that the continuity is not necessarily guaranteed in correspondence of the points $\{\bar{x}_i\}$. Thus, we can formulate the following approximation of problem (13.52): for each $t > 0$, find $u_{N,M}(t) \in W_{N,M}$ such that

$$\begin{aligned} & \sum_{m=1}^M \left[\left(\frac{\partial u_{N,M}}{\partial t}, v_{N,M} \right)_{N,\Omega_m} + \left(a \frac{\partial u_{N,M}}{\partial x}, v_{N,M} \right)_{N,\Omega_m} + (a_0 u_{N,M}, v_{N,M})_{N,\Omega_m} \right. \\ & \left. + a(\bar{x}_{m-1}) (u_{N,M}^+ - U_{N,M}^-)(\bar{x}_{m-1}) v_{N,M}^+(\bar{x}_{m-1}) \right] = \sum_{m=1}^M (f, v_{N,M})_{N,\Omega_m} \end{aligned} \quad (13.61)$$

for all $v_{N,M} \in W_{N,M}$, with

$$U_{N,M}^-(\bar{x}_i) = \begin{cases} u_{N,M}^-(\bar{x}_i), & i = 1, \dots, M-1, \\ \varphi(\bar{x}_0), & \text{for } i = 0, \end{cases} \quad (13.62)$$

and where $(\cdot, \cdot)_{N,\Omega_m}$ denotes the approximation via the GLL formula (10.25) of the scalar product L^2 restrained to the element Ω_m . To simplify the notations we have omitted to explicitly indicate the dependence on t of $u_{N,M}$ and f . Given the discontinuous nature of the test functions, we can reformulate equation (13.61) on each of the M sub-intervals, by choosing the test function $v_{N,M}$ so that $v_{N,M}|_{[\alpha,\beta] \setminus \Omega_m} = 0$. Proceeding this way, we obtain

$$\begin{aligned} & \left(\frac{\partial u_{N,M}}{\partial t}, v_{N,M} \right)_{N,\Omega_m} + \left(a \frac{\partial u_{N,M}}{\partial x}, v_{N,M} \right)_{N,\Omega_m} + (a_0 u_{N,M}, v_{N,M})_{N,\Omega_m} \\ & + a(\bar{x}_{m-1}) (u_{N,M}^+ - U_{N,M}^-)(\bar{x}_{m-1}) v_{N,M}^+(\bar{x}_{m-1}) = (f, v_{N,M})_{N,\Omega_m}, \end{aligned}$$

for each $m = 1, \dots, M$. We note that, for $m = 1$, the term

$$a(\bar{x}_0) (u_{N,M}^+ - \varphi)(\bar{x}_0) v_{N,M}^+(\bar{x}_0)$$

can be regarded as the imposition in weak form of the inflow boundary condition. On the other hand for $m = 2, \dots, M$, the term

$$a(\bar{x}_{m-1}) (u_{N,M}^+ - U_{N,M}^-)(\bar{x}_{m-1}) v_{N,M}^+(\bar{x}_{m-1}),$$

can be interpreted as a penalization term that provides a weak imposition of the continuity of the solution $u_{N,M}$ at the extrema \bar{x}_i , $i = 1, \dots, M-1$.

We now want to interpret the formulation (13.61) as a suitable collocation method. To this end, we introduce on each sub-interval Ω_m , the $N+1$ GLL nodes $x_j^{(m)}$, with $j = 0, \dots, N$, and we denote by $\alpha_j^{(m)}$ the corresponding weights (see (10.71)). We now identify the test function $v_{N,M}$ in (13.61) as the characteristic Lagrangian polynomial $\psi_j^{(m)} \in \mathbb{P}^N(\Omega_m)$ associated to node $x_j^{(m)}$ and extended by zero outside the domain Ω_m . Given the presence of the jump term, we will have a non-univocal rewriting for equation (13.61). We start by considering the characteristic polynomials associated to the nodes $x_j^{(m)}$, with $j = 1, \dots, N-1$, and $m = 1, \dots, M$. In this case we will have no contribution of the penalization term, yielding

$$\left[\frac{\partial u_{N,M}}{\partial t} + a \frac{\partial u_{N,M}}{\partial x} + a_0 u_{N,M} \right](x_j^{(m)}) = f(x_j^{(m)}).$$

For this choice of nodes we thus find exactly the collocation of the differential problem (13.52).

Instead, in the case where function $\psi_j^{(m)}$ is associated to a node of the partition $\{\bar{x}_i\}$, that is $j = 0$, with $m = 1, \dots, M$ we have

$$\begin{aligned} & \alpha_0^{(m)} \left[\frac{\partial u_{N,M}}{\partial t} + a \frac{\partial u_{N,M}}{\partial x} + a_0 u_{N,M} \right](x_0^{(m)}) \\ & + a(x_0^{(m)}) (u_{N,M}^+ - U_{N,M}^-)(x_0^{(m)}) = \alpha_0^{(m)} f(x_0^{(m)}), \end{aligned} \quad (13.63)$$

recalling that $U_{N,M}^-(x_0^{(1)}) = \varphi(\bar{x}_0)$. We have implicitly adopted the convention that the sub-interval Ω_m should not include \bar{x}_m , as the discontinuous nature of the adopted method would take us to processing twice each node \bar{x}_i , with $i = 1, \dots, M - 1$. Equation (13.63) can be rewritten as

$$\left[\frac{\partial u_{N,M}}{\partial t} + a \frac{\partial u_{N,M}}{\partial x} + a_0 u_{N,M} - f \right](x_0^{(m)}) = -\frac{a(x_0^{(m)})}{\alpha_0^{(m)}} (u_{N,M}^+ - U_{N,M}^-)(x_0^{(m)}).$$

We observe that while the left-hand side represents the residue of the equation at node $x_0^{(m)}$, the right-hand side one is, up to a multiplicative factor, the residue of the weak imposition of the continuity of $u_{N,M}$ in $x_0^{(m)}$.

13.6 Numerical treatment of boundary conditions for hyperbolic systems

We have seen different strategies to impose the inflow boundary conditions for the scalar transport equation. When considering hyperbolic systems, the numerical treatment of boundary conditions requires more attention. We will illustrate this issue on a linear system with constant coefficients in one dimension,

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + A \frac{\partial \mathbf{u}}{\partial x} = \mathbf{0}, & -1 < x < 1, \quad t > 0, \\ \mathbf{u}(x, 0) = \mathbf{u}_0(x), & -1 < x < 1, \end{cases} \quad (13.64)$$

completed with suitable boundary conditions. Following [CHQZ07], we choose the case of a system made of two hyperbolic equations, identifying in (13.64) \mathbf{u} with the vector $(u, v)^T$ and A with the matrix

$$A = \begin{bmatrix} -1/2 & -1 \\ -1 & -1/2 \end{bmatrix},$$

whose eigenvalues are $-3/2$ and $1/2$. We make the choice

$$u(x, 0) = \sin(2x) + \cos(2x), \quad v(x, 0) = \sin(2x) - \cos(2x)$$

for the initial conditions and

$$\begin{aligned} u(-1, t) &= \sin(-2 + 3t) + \cos(-2 - t) = \varphi(t), \\ v(1, t) &= \sin(2 + 3t) + \cos(2 - t) = \psi(t) \end{aligned} \quad (13.65)$$

for the boundary conditions.

Let us now consider the (right) eigenvector matrix

$$W = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix},$$

whose inverse is

$$W^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Exploiting the diagonalization

$$\Lambda = W^{-1} A W = \begin{bmatrix} -3/2 & 0 \\ 0 & 1/2 \end{bmatrix},$$

we can rewrite the differential equation in (13.64) in terms of the characteristic variables

$$\mathbf{z} = W^{-1} \mathbf{u} = \begin{bmatrix} u + v \\ u - v \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \quad (13.66)$$

as

$$\frac{\partial \mathbf{z}}{\partial t} + \Lambda \frac{\partial \mathbf{z}}{\partial x} = \mathbf{0}. \quad (13.67)$$

The characteristic variable z_1 propagates toward the left at rate $3/2$, while z_2 propagates toward the right at rate $1/2$.

This suggests to assign a condition for z_1 at $x = 1$ and one for z_2 at $x = -1$. The boundary values of z_1 and z_2 can be generated by using the boundary conditions for u and v as follows. From relation (13.66), we have

$$\mathbf{u} = W \mathbf{z} = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1/2(z_1 + z_2) \\ 1/2(z_1 - z_2) \end{bmatrix},$$

that is, exploiting the boundary values (13.65) assigned for u and v ,

$$\frac{1}{2}(z_1 + z_2)(-1, t) = \varphi(t), \quad \frac{1}{2}(z_1 - z_2)(1, t) = \psi(t). \quad (13.68)$$

The conclusion is that, in spite of the diagonal structure of system (13.67), the characteristic variables are in fact coupled by the boundary conditions (13.68).

We are therefore confronted with the problem of how to handle, from a numerical viewpoint, the boundary conditions for systems like (13.64). Indeed, difficulties can arise even from the discretization of the corresponding scalar problem (for a constant > 0)

$$\begin{cases} \frac{\partial z}{\partial t} + a \frac{\partial z}{\partial x} = 0, & -1 < x < 1, \quad t > 0, \\ z(-1, t) = \phi(t), & t > 0, \\ z(x, 0) = z_0(x), & -1 < x < 1, \end{cases} \quad (13.69)$$

if we do not use an appropriate discretization scheme. We will illustrate the procedure for a spectral approximation method. As a matter of fact, a correct treatment of the boundary conditions for high order methods is even more vital than for a finite element or finite difference method, as with spectral methods boundary errors would be propagated inwards with an infinite rate.

Having introduced the partition $x_0 = -1 < x_1 < \dots < x_{N-1} < x_N = 1$ of the interval $[-1, 1]$, if we decide to use, e.g., a finite difference scheme, we encounter problems essentially in getting the value of z at the outflow node x_N , unless we use the first order upwind scheme. As a matter of fact, higher order FD schemes such as the centered finite difference scheme would not be able to provide us with such an approximation unless we introduce additional nodes outside the definition interval $(-1, 1)$.

In contrast, a spectral discretization does not involve any boundary problem. For instance, the collocation scheme corresponding to problem (13.69) can be written as follows:

$\forall n \geq 0$, find $z_N^n \in \mathbb{Q}_N$ such that

$$\begin{cases} \frac{z_N^{n+1}(x_i) - z_N^n(x_i)}{\Delta t} + a \frac{\partial z_N^n}{\partial x}(x_i) = 0, & i = 1, \dots, N, \\ z_N^{n+1}(x_0) = \phi(t^{n+1}). \end{cases}$$

One equation results to be associated to each node, be it an internal or a boundary one, and the outflow node is treated as any other internal node. However, when moving to system (13.64), two unknowns and two equations are associated at each internal node x_i , with $i = 1, \dots, N-1$, while at the boundary nodes x_0 and x_N we still have two unknowns but a single equation. Thus, we will need additional conditions for these points: in general, at the extremum $x = -1$ we will need as many conditions as the positive eigenvalues, while for $x = 1$ we will need to provide as many additional conditions as the negative eigenvalues.

Let us look for a solution to this problem by drawing inspiration from the spectral Galerkin method. Let us suppose we apply a collocation method to system (13.64); then, we want to find $\mathbf{u}_N = (u_{N,1}, u_{N,2})^T \in (\mathbb{Q}_N)^2$ such that

$$\frac{\partial \mathbf{u}_N}{\partial t}(x_i) + A \frac{\partial \mathbf{u}_N}{\partial x}(x_i) = \mathbf{0}, \quad i = 1, \dots, N-1, \quad (13.70)$$

and with

$$u_{N,1}(x_0, t) = \varphi(t), \quad u_{N,2}(x_N, t) = \psi(t). \quad (13.71)$$

The simplest idea to obtain the two missing equations for $u_{N,1}$ and $u_{N,2}$ at x_N resp. x_0 , is to exploit the vector equation (13.70) together with the known vectors $\varphi(t)$ and $\psi(t)$ in (13.71). The solution computed this way results however to be strongly unstable.

To seek an alternative approach, the idea is to add to the $2(N-1)$ collocation relations (13.70) and to the “physical” boundary conditions (13.71), the equations of the outgoing characteristics at points x_0 and x_N . More in detail, the characteristic outgoing from the domain at point $x_0 = -1$ is the one associated to the negative eigenvalue of matrix A , and has equation

$$\frac{\partial z_1}{\partial t}(x_0) - \frac{3}{2} \frac{\partial z_1}{\partial x}(x_0) = 0, \quad (13.72)$$

while the one associated with the point $x_N = 1$ is highlighted by the positive eigenvalue $1/2$ and is given by

$$\frac{\partial z_2}{\partial t}(x_N) + \frac{1}{2} \frac{\partial z_2}{\partial x}(x_N) = 0. \quad (13.73)$$

The choice of the outgoing characteristic is motivated by the fact that the latter carries information from the inside of the domain to the corresponding outflow point, where it makes sense to impose the differential equation.

Equations (13.72) and (13.73) allow us to have a closed system of $2N + 2$ equations in the $2N + 2$ unknowns $u_{N,1}(x_i, t) = u_N(x_i, t)$, $u_{N,2}(x_i, t) = v_N(x_i, t)$, with $i = 0, \dots, N$.

For completeness, we can rewrite the characteristic equations (13.72) and (13.73) in terms of the unknowns u_N and v_N , as

$$\frac{\partial(u_N + v_N)}{\partial t}(x_0) - \frac{3}{2} \frac{\partial(u_N + v_N)}{\partial x}(x_0) = 0,$$

and

$$\frac{\partial(u_N - v_N)}{\partial t}(x_N) + \frac{1}{2} \frac{\partial(u_N - v_N)}{\partial x}(x_N) = 0,$$

respectively, or in matrix terms as

$$\begin{aligned} [W_{11}^{-1} \ W_{12}^{-1}] \left[\frac{\partial \mathbf{u}_N}{\partial t}(x_0) + A \frac{\partial \mathbf{u}_N}{\partial x}(x_0) \right] &= 0, \\ [W_{21}^{-1} \ W_{22}^{-1}] \left[\frac{\partial \mathbf{u}_N}{\partial t}(x_N) + A \frac{\partial \mathbf{u}_N}{\partial x}(x_N) \right] &= 0. \end{aligned} \quad (13.74)$$

Such additional equations are called *compatibility* equations: they represent a linear combination of the differential equations of the problem at the boundary points with coefficients given by the components of matrix W^{-1} .

Remark 13.4 Due to their global nature, spectral methods (either collocation, Galerkin, or G-NI) propagate immediately and on the whole domain every possible numerical perturbation introduced at the boundary. As such, spectral methods represent a good testbed for investigating the suitability of numerical strategies for the boundary treatment of hyperbolic systems. •

13.6.1 Weak treatment of boundary conditions

We now want to generalize the approach based on compatibility equations moving from pointwise relations, such as (13.74), to integral relations, that are more suitable for numerical approximations such as, e.g., finite elements or G-NI.

Let us again consider the constant coefficient system (13.64) and the notations used in Sec. 13.6. Let A be a real, symmetric and non-singular matrix of order d , Λ the diagonal matrix whose diagonal entries are the real eigenvalues of A , and W the square matrix whose columns are the (right) eigenvectors of A . Let us suppose that W is

orthogonal, which guarantees that $\Lambda = W^T AW$. The characteristic variables, defined as $\mathbf{z} = W^T \mathbf{u}$, satisfy the diagonal system (13.67). We introduce the splitting $\Lambda = \text{diag}(\Lambda^+, \Lambda^-)$ of the eigenvalue matrix, respectively grouping the positive eigenvalues (Λ^+) and the negative ones (Λ^-). Both such sub-matrices result to be diagonal, Λ^+ positive definite of order p , Λ^- negative definite of order $n = d - p$.

Analogously, we can rewrite \mathbf{z} as $\mathbf{z} = (\mathbf{z}^+, \mathbf{z}^-)^T$, having denoted by \mathbf{z}^+ (\mathbf{z}^- , respectively) the characteristic variables that are constant along the characteristic lines with positive (respectively, negative) slope, that is that move towards the right (respectively, left) on the (x, t) reference frame. In correspondence of the right extremum $x = 1$, \mathbf{z}^+ is associated to the outgoing characteristic variables while \mathbf{z}^- is associated to the incoming ones. Clearly, the roles are switched at the left boundary point $x = -1$.

A simple case occurs when, as boundary conditions, we assign the values of the incoming characteristics at both the domain extrema, that is p conditions at $x = -1$ and n conditions at $x = 1$. In this case, (13.67) represents a full-fledged decoupled system. Much more frequently however, the values of suitable linear combinations of the physical variables are assigned at both boundary points. Re-reading them in terms of the \mathbf{z} variables, these yield linear combinations of the characteristic variables. None of the outgoing characteristics will in principle be determined by these combinations as the resulting values will generally be incompatible with the ones propagated inwards from the hyperbolic system. In contrast, the boundary conditions should allow to determine the incoming characteristic variables as a function of the outgoing ones and of the problem data.

For the sake of illustration, let us consider the following boundary conditions

$$B_L \mathbf{u}(-1, t) = \mathbf{g}_L(t), \quad B_R \mathbf{u}(1, t) = \mathbf{g}_R(t), \quad t > 0, \quad (13.75)$$

where \mathbf{g}_L and \mathbf{g}_R are assigned vectors and B_L, B_R are suitable matrices. At the left extremum, $x = -1$ we have p incoming characteristics, then B_L will have dimension $p \times d$. Setting $C_L = B_L W$ and using the splitting $\mathbf{z} = (\mathbf{z}^+, \mathbf{z}^-)^T$ introduced for \mathbf{z} and the corresponding splitting $W = (W^+, W^-)^T$ for the eigenvector matrix, we have

$$C_L \mathbf{z}(-1, t) = C_L^+ \mathbf{z}^+(-1, t) + C_L^- \mathbf{z}^-(-1, t) = \mathbf{g}_L(t),$$

where $C_L^+ = B_L W^+$ is a $p \times p$ matrix, while $C_L^- = B_L W^-$ has dimension $p \times n$. We formulate the requirement that matrix C_L^+ be non-singular. Then the incoming characteristic at the $x = -1$ extremum can be obtained by

$$\mathbf{z}^+(-1, t) = S_L \mathbf{z}^-(-1, t) + \mathbf{z}_L(t), \quad (13.76)$$

$S_L = -(C_L^+)^{-1} C_L^-$ being a $p \times n$ matrix and $\mathbf{z}_L(t) = (C_L^+)^{-1} \mathbf{g}_L(t)$. In a similar way, we can assign at the right extremum $x = 1$ the incoming characteristic variable as

$$\mathbf{z}^-(1, t) = S_R \mathbf{z}^+(1, t) + \mathbf{z}_R(t), \quad (13.77)$$

S_R being a $n \times p$ matrix.

Matrices S_L and S_R are called *reflection matrices*.

The hyperbolic system (13.64) will thus be completed by the boundary conditions

(13.75) or, equivalently, by conditions (13.76)-(13.77).

Let us see which advantages can be brought by such a choice for boundary conditions. We start from the weak formulation of problem (13.64), integrating by parts the term containing the space derivative

$$\int_{-1}^1 \mathbf{v}^T \frac{\partial \mathbf{u}}{\partial t} dx - \int_{-1}^1 \left(\frac{\partial \mathbf{v}}{\partial x} \right)^T A \mathbf{u} dx + [\mathbf{v}^T A \mathbf{u}]_{-1}^1 = 0,$$

for each $t > 0$, \mathbf{v} being an arbitrary, differentiable test function. We want to rewrite the boundary term $[\mathbf{v}^T A \mathbf{u}]_{-1}^1$ by exploiting the boundary equations (13.76) - (13.77). Introducing the characteristic variable $W^T \mathbf{v} = \mathbf{y} = (\mathbf{y}^+, \mathbf{y}^-)^T$ associated to the test function \mathbf{v} , we will have

$$\mathbf{v}^T A \mathbf{u} = \mathbf{y}^T \Lambda \mathbf{z} = (\mathbf{y}^+)^T \Lambda^+ \mathbf{z}^+ + (\mathbf{y}^-)^T \Lambda^- \mathbf{z}^-.$$

Using the relations (13.76)-(13.77), it then follows that

$$\begin{aligned} & \int_{-1}^1 \mathbf{v}^T \frac{\partial \mathbf{u}}{\partial t} dx - \int_{-1}^1 \left(\frac{\partial \mathbf{v}}{\partial x} \right)^T A \mathbf{u} dx \\ & - (\mathbf{y}^+)^T (-1, t) \Lambda^+ S_L \mathbf{z}^-(-1, t) - (\mathbf{y}^-)^T (-1, t) \Lambda^- \mathbf{z}^-(-1, t) \\ & + (\mathbf{y}^+)^T (1, t) \Lambda^+ \mathbf{z}^+(1, t) + (\mathbf{y}^-)^T (1, t) \Lambda^- S_R \mathbf{z}^+(1, t) \\ & = (\mathbf{y}^+)^T (-1, t) \Lambda^+ \mathbf{z}_L(t) - (\mathbf{y}^-)^T (1, t) \Lambda^- \mathbf{z}_R(t). \end{aligned} \quad (13.78)$$

We observe that the boundary conditions (13.76)-(13.77) are naturally incorporated in the right-hand side of the system. Moreover, integrating again by parts, it is possible to obtain an equivalent formulation to (13.78) where the boundary conditions are imposed in a weak way

$$\begin{aligned} & \int_{-1}^1 \mathbf{v}^T \frac{\partial \mathbf{u}}{\partial t} dx + \int_{-1}^1 \mathbf{v}^T A \frac{\partial \mathbf{u}}{\partial x} dx \\ & + (\mathbf{y}^+)^T (-1, t) \Lambda^+ (\mathbf{z}^+(-1, t) - S_L \mathbf{z}^-(-1, t)) \\ & - (\mathbf{y}^-)^T (1, t) \Lambda^- (\mathbf{z}^-(1, t) - S_R \mathbf{z}^+(1, t)) \\ & = (\mathbf{y}^+)^T (-1, t) \Lambda^+ \mathbf{z}_L(t) - (\mathbf{y}^-)^T (1, t) \Lambda^- \mathbf{z}_R(t). \end{aligned} \quad (13.79)$$

Finally, we recall that the following assumption, called *dissipation hypothesis*, is usually made on the reflection matrices S_L and S_R

$$\|S_L\| \|S_R\| < 1. \quad (13.80)$$

The matrix norm in (13.80) must be understood as the euclidean norm of a rectangular matrix, that is the square root of the maximum eigenvalue of $S_L^T S_L$ and that of $S_R^T S_R$, respectively.

This assumption is sufficient to guarantee the stability of the previous scheme in the L^2 norm. Formulation (13.78) (or (13.79)) is suitable for Galerkin approximations such as the Galerkin-finite elements, the spectral Galerkin method, the spectral method with Gaussian numerical integration in a single domain (G-NI), the spectral element version, in both cases of continuous (SEM-NI) or discontinuous (DG-SEM-NI) spectral elements.

13.7 Exercises

1. Prove that the discretization with continuous linear finite elements (13.13) coincides with the finite difference one (12.22) in the case where the mass matrix is diagonalized using the mass lumping technique.
[Solution: use the partition of unity property (11.31) as in Sec. 11.5.]
2. Prove the stability inequalities provided in Sec. (13.4) for the semi-discretization based on finite elements.
3. Prove relation (13.13).
4. Discretize system (13.78) using the continuous spectral element method, SEM-NI, and the discontinuous one, DG-SEM-NI.

14

Nonlinear hyperbolic problems

In this chapter, we introduce some examples of nonlinear hyperbolic problems. We will point out some characteristic properties of such problems, the most relevant being their possibility to generate discontinuous solutions also in the case of continuous initial and boundary data. The numerical approximation of these problems is an all but simple task. Here, we will simply limit ourselves to point out how finite difference and finite element schemes can be applied in the case of one-dimensional equations.

For a more complete discussion, we refer to [LeV07], [GR96], [Bre00], [Tor99], [Kro97].

14.1 Scalar equations

Let us consider the following equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} F(u) = 0, \quad x \in \mathbb{R}, \quad t > 0, \quad (14.1)$$

where F is a nonlinear function of u called *flux* of u as, on each interval (α, β) of \mathbb{R} , it satisfies the following relation

$$\frac{d}{dt} \int_{\alpha}^{\beta} u(x, t) dx = F(u(t, \alpha)) - F(u(t, \beta)).$$

For this reason, (14.1) expresses a *conservation law*. A typical example is the Burgers equation

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0. \quad (14.2)$$

This equation was already considered in Example 1.3, and corresponds to (14.1) where the flux is $F(u) = u^2/2$. Its characteristic curves are obtained by solving $x'(t) = u$. However, since u is constant on the characteristics, we obtain $x'(t) = \text{constant}$, so the characteristics are straight lines. The latter are defined in the plane (x, t) by the map $t \rightarrow (x + tu_0(x), t)$, and the solution to (14.2) is implicitly defined by $u(x + tu_0(x)) =$

$u_0(x)$, $t < t_c$, t_c being the very first time where such characteristics intersect. For instance, if $u_0(x) = (1 + x^2)^{-1}$, then $t_c = 8/\sqrt{27}$.

Indeed, if $u'_0(x)$ is negative at some point, having set

$$t_c = -\frac{1}{\min u'_0(x)},$$

for $t > t_c$ there can be no classical solution (i.e. of class C^1), as

$$\lim_{t \rightarrow t_c^-} \left(\inf_{x \in \mathbb{R}} \frac{\partial u}{\partial x}(x, t) \right) = -\infty$$

Let us consider Fig. 14.1: note how for $t = t_c$ the solution denotes a discontinuity.

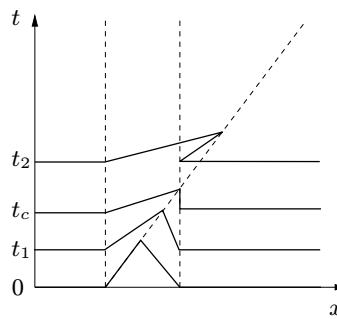


Fig. 14.1. Development of the singularity at the critical time t_c

To account for this uniqueness loss, we introduce the notion of *weak solution* of a hyperbolic equation: we say that u is a weak solution of (14.1) if it satisfies the differential relation (14.1) at all points $x \in \mathbb{R}$ except for those where it is discontinuous. In the latter, we no longer expect (14.1) to hold (it would make no sense to differentiate a discontinuous function), rather we require the following *Rankine-Hugoniot* condition to be verified

$$F(u_r) - F(u_l) = \sigma(u_r - u_l), \quad (14.3)$$

where u_r and u_l respectively denote the right and left limit of u at the discontinuity point, and σ is the speed of propagation of the discontinuity. Condition (14.3) therefore expresses the fact that the jump of the flux is proportional to the jump of the solution.

Weak solutions are not necessarily unique: among them, the physically correct one is the so-called *entropic solution*. As we will see at the end of this section, in the case of the Burgers equation, the entropic solution is obtained as the limit, for $\varepsilon \rightarrow 0$, of the solution $u^\varepsilon(x, t)$ of the equation having a viscous perturbation term

$$\frac{\partial u^\varepsilon}{\partial t} + \frac{\partial}{\partial x} F(u^\varepsilon) = \varepsilon \frac{\partial^2 u^\varepsilon}{\partial x^2}, \quad x \in \mathbb{R}, \quad t > 0,$$

with $u^\varepsilon(x, 0) = u_0(x)$.

In general, we can say that:

- if $F(u)$ is differentiable, a discontinuity that propagates at rate σ given by (14.3) satisfies the entropy condition if

$$F'(u_l) \geq \sigma \geq F'(u_r);$$

- if $F(u)$ is not differentiable, a discontinuity that propagates at rate σ given by (14.3) satisfies the entropy condition if

$$\frac{F(u) - F(u_l)}{u - u_l} \geq \sigma \geq \frac{F(u) - F(u_r)}{u - u_r},$$

for each u between u_l and u_r .

Example 14.1 Let us consider the Burgers equation with the following initial condition

$$u_0(x) = \begin{cases} u_l & \text{if } x < 0, \\ u_r & \text{if } x > 0, \end{cases}$$

where u_r and u_l are two constants. If $u_l > u_r$, then there exists a unique weak solution (which is also entropic)

$$u(x, t) = \begin{cases} u_l, & x < \sigma t, \\ u_r, & x > \sigma t, \end{cases} \quad (14.4)$$

where $\sigma = (u_l + u_r)/2$ is the propagation rate of the discontinuity (also called *shock*). In this case the characteristics “enter” the shock (see Fig. 14.2).

In the $u_l < u_r$ case, there are infinitely many weak solutions: one still has the form (14.4), but in this case the characteristics *exit* the discontinuity (see Fig. 14.3). Such solution is unstable, i.e. small perturbations on the data substantially change the solution itself. Another weak solution is

$$u(x, t) = \begin{cases} u_l & \text{if } x < u_l t, \\ \frac{x}{t} & \text{if } u_l t \leq x \leq u_r t, \\ u_r & \text{if } x > u_r t. \end{cases}$$

Such solution, describing a rarefaction wave, is entropic in contrast to the previous one (see Fig. 14.4). ■

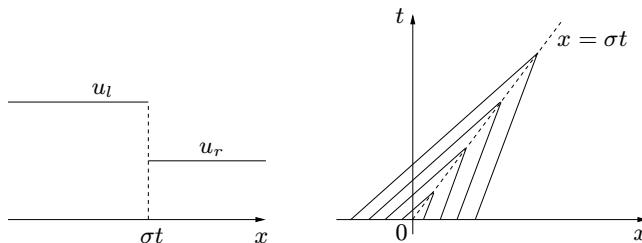
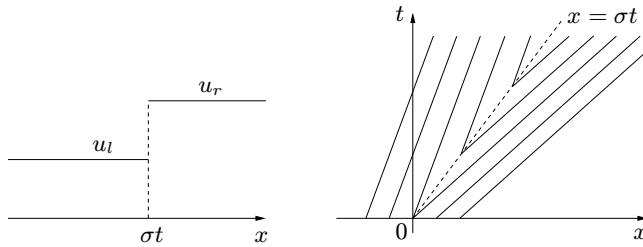
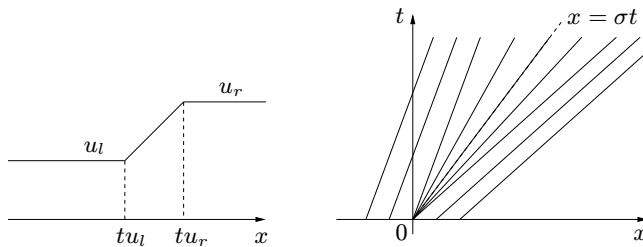


Fig. 14.2. Entropic solution to the Burgers equation

**Fig. 14.3.** Non-entropic solution to the Burgers equation**Fig. 14.4.** Rarefaction wave

We say that a hyperbolic problem (14.1) has an *entropy function* if there exist a strictly convex function $\eta = \eta(u)$ and a function $\Psi = \Psi(u)$ such that

$$\Psi'(u) = \eta'(u)F'(u), \quad (14.5)$$

where the apex ' denotes the derivative with respect to the argument u . The η function is called *entropy* and Ψ is called *entropy flux*. We recall that a function η is said to be convex if for each distinct u and w and for each $\theta \in (0, 1)$, we have

$$\eta(u + \theta(w - u)) < (1 - \theta)\eta(u) + \theta\eta(w).$$

If η has a continuous second derivative, this is equivalent to requiring that $\eta'' > 0$.

Remark 14.1 The one presented here is a “mathematical” definition of entropy. In the case where (14.1) governs a physical phenomenon, it is often possible to define a “thermodynamic entropy”. The latter effectively also results to be an entropy according to the definition given above. •

The quasi-linear form of (14.1) is given by

$$\frac{\partial u}{\partial t} + F'(u)\frac{\partial u}{\partial x} = 0. \quad (14.6)$$

If u is sufficiently regular, it can easily be verified by multiplying (14.6) by $\eta'(u)$ that η and Ψ satisfy the following *conservation law*

$$\frac{\partial \eta}{\partial t}(u) + \frac{\partial \Psi}{\partial x}(u) = 0. \quad (14.7)$$

For a scalar equation it is generally possible to find different pairs of functions η and Ψ that satisfy the given conditions.

The operations carried out to derive (14.7) make sense only if u is regular, in particular if there are no discontinuities in the solution. However, we can find the conditions to be verified by the entropy variable at a discontinuity in the solution of (14.1) when such equation represents the limit for $\epsilon \rightarrow 0^+$ of the following regularized equation (called *viscosity equation*)

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x}(u) = \epsilon \frac{\partial^2 u}{\partial x^2}. \quad (14.8)$$

The solution of (14.8) is regular for each $\epsilon > 0$, and performing the same manipulations used previously we can write

$$\frac{\partial \eta}{\partial t}(u) + \frac{\partial \Psi}{\partial x}(u) = \epsilon \eta'(u) \frac{\partial^2 u}{\partial x^2} = \epsilon \frac{\partial}{\partial x} \left[\eta'(u) \frac{\partial u}{\partial x} \right] - \epsilon \eta''(u) \left(\frac{\partial u}{\partial x} \right)^2.$$

By now integrating on a generic rectangle $[x_1, x_2] \times [t_1, t_2]$ we obtain

$$\int_{t_1}^{t_2} \int_{x_1}^{x_2} \left[\frac{\partial \eta}{\partial t}(u) + \frac{\partial \Psi}{\partial x}(u) \right] dx dt = \epsilon \int_{t_1}^{t_2} \left[\eta'(u(x_2, t)) \frac{\partial u}{\partial x}(x_2, t) \right. \\ \left. - \eta'(u(x_1, t)) \frac{\partial u}{\partial x}(x_1, t) \right] dt - \epsilon \int_{t_1}^{t_2} \int_{x_1}^{x_2} \eta''(u) \left(\frac{\partial u}{\partial x} \right)^2 dx dt = R_1(\epsilon) + R_2(\epsilon),$$

where we have set

$$R_1(\epsilon) = \epsilon \int_{t_1}^{t_2} \left[\eta'(u(x_2, t)) \frac{\partial u}{\partial x}(x_2, t) - \eta'(u(x_1, t)) \frac{\partial u}{\partial x}(x_1, t) \right] dt, \\ R_2(\epsilon) = -\epsilon \int_{t_1}^{t_2} \int_{x_1}^{x_2} \eta''(u) \left(\frac{\partial u}{\partial x} \right)^2 dx dt.$$

We have

$$\lim_{\epsilon \rightarrow 0^+} R_1(\epsilon) = 0,$$

while if the solution for $\epsilon \rightarrow 0^+$ of the modified problem denotes a discontinuity along a curve of the (x, t) plane, we have

$$\lim_{\epsilon \rightarrow 0^+} R_2(\epsilon) \neq 0,$$

as the integral containing the term $(\frac{\partial u}{\partial x})^2$ is, in general, unbounded.

On the other hand $R_2(\epsilon) \leq 0$ for each $\epsilon > 0$, with $\partial^2 \eta / \partial u^2 > 0$, hence the weak boundary solution for $\epsilon \rightarrow 0^+$ satisfies

$$\int_{t_1}^{t_2} \int_{x_1}^{x_2} \left[\frac{\partial \eta}{\partial t}(u) + \frac{\partial \Psi}{\partial x}(u) \right] dx dt \leq 0 \quad \forall x_1, x_2, t_1, t_2. \quad (14.9)$$

In other words

$$\frac{\partial \eta}{\partial t}(u) + \frac{\partial \Psi}{\partial x}(u) \leq 0, \quad x \in \mathbb{R}, \quad t > 0$$

in a weak sense.

There is obviously a relation between what we have just seen and the notion of entropic solution. If the differential equation allows for an entropy function η , then a weak solution is an entropic solution if and only if η satisfies (14.9). In other words, the entropic solutions are the limit, as $\epsilon \rightarrow 0^+$, of the solution of the regularized problem (14.8).

14.2 Finite difference approximation

Let us return to the nonlinear hyperbolic equation (14.1), with initial condition

$$u(x, 0) = u_0(x), \quad x \in \mathbb{R}.$$

We denote by $a(u) = F'(u)$ its characteristic rate.

Also for this problem, we can use an explicit finite difference scheme of the form (12.13). The functional interpretation of $H_{j+1/2}^n = H(u_j^n, u_{j+1}^n)$ is

$$H_{j+1/2}^n \simeq \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} F(u(x_{j+1/2}, t)) dt,$$

so $H_{j+1/2}^n$ approximates the mean flux through $x_{j+1/2}$ in the time interval $[t^n, t^{n+1}]$. To have consistency, the numerical flux $H(\cdot, \cdot)$ must verify

$$H(\bar{u}, \bar{u}) = F(\bar{u}), \tag{14.10}$$

in the case where \bar{u} is a constant. Under hypothesis (14.10), thanks to a classical result by Lax and Wendroff, the u functions such that

$$u(x_j, t^n) = \lim_{\Delta t, h \rightarrow 0} u_j^n$$

are weak solutions of the original problem (14.1).

Unfortunately however, it is not guaranteed that the solutions obtained in this manner satisfy the entropy condition (i.e. it is not said that weak solutions are also entropic). In order to “recover” the entropic solutions, numerical schemes must introduce a suitable numerical diffusion, as suggested by the analysis of Sec. 14.1. To this end, we rewrite (12.13) in the form

$$u_j^{n+1} = G(u_{j-1}^n, u_j^n, u_{j+1}^n) \tag{14.11}$$

and we introduce some definitions. The numerical scheme (14.11) is called:

- *monotone* if G is a monotonically increasing function of each of its arguments;
- *bounded* if there exists $C > 0$ such that $\sup_{j,n} |u_j^n| \leq C$;
- *stable* if $\forall h > 0$, $\exists \delta_0 > 0$ (possibly dependent on h) s.t. for each $0 < \Delta t < \delta_0$, if \mathbf{u}^n and \mathbf{v}^n are the finite difference solutions obtained starting from the two initial data \mathbf{u}^0 and \mathbf{v}^0 , then

$$\|\mathbf{u}^n - \mathbf{v}^n\|_\Delta \leq C_T \|\mathbf{u}^0 - \mathbf{v}^0\|_\Delta , \tag{14.12}$$

for each $n \geq 0$ s.t. $n\Delta t \leq T$ and for any choice of the initial data \mathbf{u}^0 and \mathbf{v}^0 . The constant $C_T > 0$ is independent of Δt and h , and $\|\cdot\|_\Delta$ is a suitable discrete norm, such as those introduced in (12.26). Note that for linear problems, this definition is equivalent to (12.25). We say that the numerical scheme is *strongly stable* when in (14.12) we can take $C_T = 1$ for each $T > 0$.

For example, using $F_j = F(u_j)$ for simplicity of notation, the Lax-Friedrichs scheme for problem (14.1) is realized through the general scheme (12.13) where we take

$$H_{j+1/2} = \frac{1}{2} \left[F_{j+1} + F_j - \frac{1}{\lambda}(u_{j+1} - u_j) \right].$$

This method is consistent, stable and monotone provided that the following condition (analogous to the CFL condition seen previously in the linear case) holds

$$|F'(u_j^n)| \frac{\Delta t}{h} \leq 1 \quad \forall j \in \mathbb{Z}, \quad \forall n \in \mathbb{N}. \quad (14.13)$$

A classical result due to N.N. Kuznetsov establishes that monotone schemes of the type (14.11) are bounded, stable, convergent to the entropic solution, and are at most first order accurate with respect to both time and space, that is there exists a constant $C > 0$ s.t.

$$\max_{j,n} |u_j^n - u(x_j, t^n)| \leq C(\Delta t + h).$$

These schemes are generally too dissipative and do not generate accurate solutions except when using very fine grids.

Higher order schemes (called *high order shock capturing schemes*) can be developed using techniques that allow to calibrate the numerical dissipation as a function of the local regularity of the solution. By so doing one can correctly solve the discontinuities (ensuring the convergence of entropic solution and avoiding spurious oscillations) by using a minimal numerical dissipation. This is a complex topic and cannot be addressed with too much synthesis. For an in-depth analysis, we refer to [LeV02b], [LeV07], [GR96] and [Hir88].

14.3 Approximation by discontinuous finite elements

For the discretization of problem (14.1) we now consider the space approximation based on discontinuous Galerkin (DG) finite elements. Using the same notations introduced in Sec. 13.4, we seek, for each $t > 0$, $u_h(t) \in W_h$ such that we have $\forall j = 0, \dots, m-1$ and $\forall v_h \in \mathbb{P}_r(I_j)$,

$$\int_{I_j} \frac{\partial u_h}{\partial t} v_h dx - \int_{I_j} F(u_h) \frac{\partial v_h}{\partial x} dx + H_{j+1}(u_h) v_h^-(x_{j+1}) - H_j(u_h) v_h^+(x_j) = 0, \quad (14.14)$$

with $I_j = [x_j, x_{j+1}]$. The initial datum u_h^0 is provided by the relations

$$\int_{I_j} u_h^0 v_h dx = \int_{I_j} u_0 v_h dx, \quad j = 0, \dots, m-1.$$

Function H_j now denotes the nonlinear flux at node x_j and depends on the values of u_h at x_j , that is

$$H_j(u_h(t)) = H(u_h^-(x_j, t), u_h^+(x_j, t)), \quad (14.15)$$

for a suitable numerical flux $H(\cdot, \cdot)$. If $j = 0$ we will have to set $u_h^-(x_0, t) = \phi(t)$, which is the boundary datum at the left extremum (assuming of course that this is the inflow point).

We note that there exist various options for the choice of function H . However, we want such choices to yield in (14.14) to schemes that can be regarded as perturbations of *monotone* schemes. Indeed, as noted in the previous section, the latter are stable and convergent to the entropic solution albeit being only first order accurate. More precisely, we pretend (14.14) to be a monotone scheme when $r = 0$. In this case, having denoted by $u_h^{(j)}$ the *constant value* of u_h on I_j , (14.14) becomes

$$h_j \frac{\partial}{\partial t} u_h^{(j)}(t) + H(u_h^{(j)}(t), u_h^{(j+1)}(t)) - H(u_h^{(j-1)}(t), u_h^{(j)}(t)) = 0, \quad (14.16)$$

with initial datum $u_h^{0,(j)} = h_j^{-1} \int_{x_j}^{x_{j+1}} u_0 dx$ in the interval I_j , $j = 0, \dots, m-1$. We have denoted by $h_j = x_{j+1} - x_j$ the width of I_j .

In order for scheme (14.16) to be monotone, the flux H must be monotone, which is equivalent to saying that $H(v, w)$ is:

- a Lipschitz function of its arguments;
- a function not decreasing in v and not increasing in w . Symbolically, $H(\uparrow, \downarrow)$;
- consistent with the flux F , i.e. $H(\bar{u}, \bar{u}) = F(\bar{u})$, for any constant value \bar{u} .

Three classical examples of monotone fluxes are the following:

1. Godunov Flux

$$H(v, w) = \begin{cases} \min_{v \leq u \leq w} F(u) & \text{if } v \leq w, \\ \max_{w \leq u \leq v} F(u) & \text{if } v > w; \end{cases}$$

2. Engquist-Osher Flux

$$H(v, w) = \int_0^v \max(F'(u), 0) du + \int_0^w \min(F'(u), 0) du + F(0);$$

3. Lax-Friedrichs Flux

$$H(v, w) = \frac{1}{2} [F(v) + F(w) - \delta(w - v)], \quad \delta = \max_{\inf_x u_0(x) \leq u \leq \sup_x u_0(x)} |F'(u)|.$$

The Godunov flux is the one yielding the least amount of numerical dissipation, the Lax-Friedrichs is the cheapest to evaluate. However, numerical experience suggests that if the degree r increases, the choice of the H flux has no significant consequences on the quality of the approximation.

In the linear case, where $F(u) = au$, all of the previous fluxes coincide and are equal to the upwind flux

$$H(v, w) = a \frac{v + w}{2} - \frac{|a|}{2}(w - v). \quad (14.17)$$

In this case we observe that the scheme (14.14) exactly coincides with the one introduced in (13.42) when $a > 0$. Indeed, having set $a_0 = 0$ and $f = 0$ in (13.42) and integrating by parts we obtain, for each $j = 1, \dots, m-1$

$$\begin{aligned} & \int_{I_j} \frac{\partial u_h}{\partial t} v_h \, dx - \int_{I_j} (au_h) \frac{\partial v_h}{\partial x} \, dx \\ & + (au_h)^-(x_{j+1}) v_h^-(x_{j+1}) - (au_h)^-(x_j) v_h^+(x_j) = 0, \end{aligned} \quad (14.18)$$

i.e. (14.14), keeping in mind that in the case under exam $au_h = F(u_h)$ and, $\forall j = 1, \dots, m-1$,

$$(au_h)^-(x_j) = a \frac{u_h^-(x_j) + u_h^+(x_j)}{2} - \frac{a}{2}(u_h^+(x_j) - u_h^-(x_j)) = H_j(u_h).$$

Verification in the $j = 0$ case is obvious.

We have the following stability result

$$\|u_h(t)\|_{L^2(\alpha, \beta)}^2 + \theta(u_h(t)) \leq \|u_h^0\|_{L^2(\alpha, \beta)}^2$$

having set $[u_h]_j = u_h^+(x_j) - u_h^-(x_j)$, and

$$\theta(u_h(t)) = |a| \int_0^t \sum_{j=1}^{m-1} [u_h(t)]_j^2 dt.$$

Note how jumps are also controlled by the initial datum. Convergence analysis provides the following result (under the assumption that $u_0 \in H^{r+1}(\alpha, \beta)$)

$$\|u(t) - u_h(t)\|_{L^2(\alpha, \beta)} \leq Ch^{r+1/2}|u_0|_{H^{r+1}(\alpha, \beta)}, \quad (14.19)$$

hence a convergence order ($= r + 1/2$) larger than the one ($= r$) we would have using continuous finite elements, as previously encountered in the linear case (see (13.48)). In the nonlinear case and for $r = 0$, defining the seminorm

$$|v|_{TV(\alpha, \beta)} = \sum_{j=0}^{m-1} |v_{j+1} - v_j|, \quad v \in W_h,$$

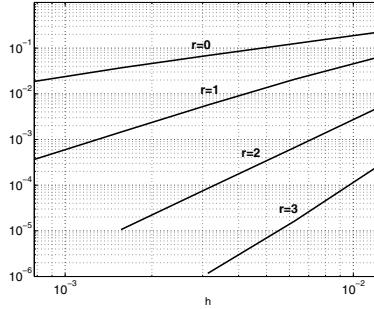


Fig. 14.5. Error $\|u - u_h\|_{L^2(0,1)}$ obtained by solving a linear transport problem with regular initial datum using discontinuous finite elements of degree $r = 0, 1, 2, 3$. The error has been computed at time $t = 0.01$. The time advancing scheme is the third-order Runge-Kutta scheme with time step $\Delta t = 5 \times 10^{-4}$

and taking the Engquist-Osher numerical flux in (14.16), we have the following result (due to N.N. Kuznestov)

$$\|u(t) - u_h(t)\|_{L^1(\alpha, \beta)} \leq \|u_0 - u_h^0\|_{L^1(\alpha, \beta)} + C|u_0|_{TV(\alpha, \beta)}\sqrt{t h}.$$

Moreover, $|u_h(t)|_{TV(\alpha, \beta)} \leq |u_h^0|_{TV(\alpha, \beta)} \leq |u_0|_{TV(\alpha, \beta)}$.

For the temporal discretization, we first write scheme (14.14) in the algebraic form

$$\begin{aligned} M_h \dot{\mathbf{u}}_h(t) &= L_h(\mathbf{u}_h(t), t), \quad t \in (0, T), \\ \mathbf{u}_h(0) &= \mathbf{u}_h^0, \end{aligned}$$

$\mathbf{u}_h(t)$ being the degrees of freedom vector, $L_h(\mathbf{u}_h(t), t)$ the vector resulting from the discretization of the flux term $-\frac{\partial F}{\partial x}$ and M_h the mass matrix. M_h is a block diagonal matrix whose j -th block is the mass matrix corresponding to the I_j element (as previously observed, the latter is diagonal if we resort to the Legendre polynomial basis, which is orthogonal).

For the temporal discretization, in addition to the previously discussed Euler schemes, we can resort to the following second-order Runge-Kutta method:

$$\begin{aligned} M_h(\mathbf{u}_h^* - \mathbf{u}_h^n) &= \Delta t L_h(\mathbf{u}_h^n, t^n), \\ M_h(\mathbf{u}_h^{**} - \mathbf{u}_h^*) &= \Delta t L_h(\mathbf{u}_h^*, t^{n+1}), \\ \mathbf{u}_h^{n+1} &= \frac{1}{2}(\mathbf{u}_h^n + \mathbf{u}_h^{**}). \end{aligned}$$

In the case of the linear problem (where $F(u) = au$), using $r = 1$ this scheme is stable in the $\|\cdot\|_{L^2(\alpha, \beta)}$ norm provided that the condition

$$\Delta t \leq \frac{1}{3} \frac{h}{|a|}$$

is satisfied. For an arbitrary r , numerical evidence shows that a scheme of order $2r + 1$ must be used, in which case we have stability under the condition

$$\Delta t \leq \frac{1}{2r+1} \frac{h}{|a|}.$$

We report the 3rd order Runge-Kutta scheme, to be used preferably when $r = 1$:

$$\begin{aligned} M_h(\mathbf{u}_h^* - \mathbf{u}_h^n) &= \Delta t L_h(\mathbf{u}_h^n, t^n), \\ M_h(\mathbf{u}_h^{**} - (\frac{3}{4}\mathbf{u}_h^n + \frac{1}{4}\mathbf{u}_h^*)) &= \frac{1}{4}\Delta t L_h(\mathbf{u}_h^*, t^{n+1}), \\ M_h(u_h^{n+1} - (\frac{1}{3}\mathbf{u}_h^n + \frac{2}{3}\mathbf{u}_h^{**})) &= \frac{2}{3}\Delta t L_h(\mathbf{u}_h^{**}, t^{n+1/2}). \end{aligned} \quad (14.20)$$

Example 14.2 Let us reconsider the problem of Example 13.2, which we solve with the discontinuous finite element method, using the third-order Runge-Kutta scheme for the temporal discretization. Our scope is to experimentally verify (14.19). To this end, we used a very small time step, $\Delta t = 5 \times 10^{-4}$, and 5 decreasing values for step h obtained by repeatedly halving the initial value $h = 12.5 \times 10^{-3}$. We have compared the error in $L^2(0, 1)$ norm at time $t = 0.01$ for elements of degree r equal to 0, 1, 2 and 3. The result is reported in logarithmic scale in Fig. 14.5. This is in accordance with the theory by which the error tends to zero as $h^{r+1/2}$. Indeed, for $r = 1$ in this particular case convergence is faster than predicted in theory: the numerical data provides an order of convergence very close to 2. In the $r > 1$ case we have not reported the results for values smaller than h as for such values (and for the selected Δt) the problem is numerically unstable. ■

Example 14.3 Let us consider the same linear transport problem of the previous example, now using as the initial datum the square wave illustrated at the left-hand side in Fig. 14.6. As the initial datum is discontinuous, the use of high-degree elements does not improve the convergence order, which results to be very close to the theoretical value of 1/2 for all of the considered values of r . In Fig. 14.7 we show the oscillations in proximity of the discontinuity of the solution in the $r = 2$ case, responsible of the convergence degradation, while the solution for $r = 0$ denotes no oscillation. ■

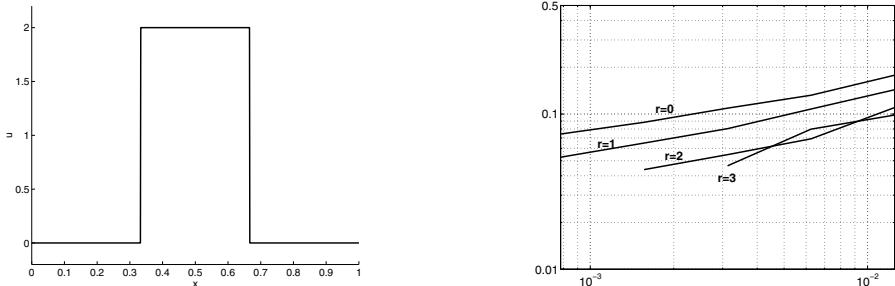


Fig. 14.6. Error $\|u - u_h\|_{L^2(0,1)}$ obtained by solving a linear transport problem with initial datum illustrated in the left figure. We have used discontinuous finite elements of degree r of values 0, resp. 1, 2 and 3. The error has been computed at time $t = 0.01$. The temporal progression scheme is the third-order Runge-Kutta scheme with $\Delta t = 5 \times 10^{-4}$

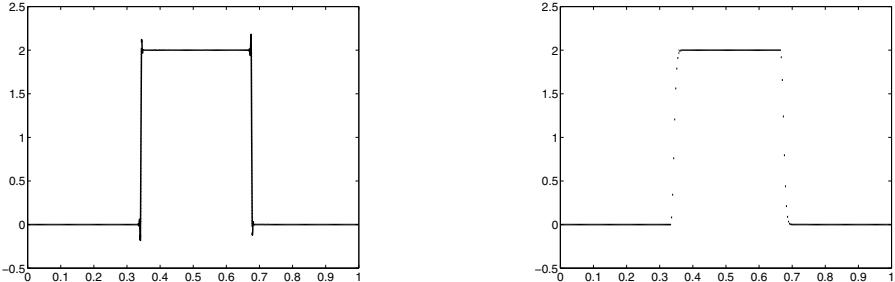


Fig. 14.7. Solution at time $t = 0.01$ and for $h = 3.125 \times 10^{-3}$ for the test case of Fig. 14.6. At the left-hand side, the $r = 3$ case: note the presence of oscillations at the discontinuities, while elsewhere the solution is accurate. At the right-hand side, we show the solution obtained when using the same spatial and temporal discretization for $r = 0$

In the case of the nonlinear problem, using the second-order Runge-Kutta scheme with $r = 0$ under the condition (14.13) we obtain

$$|u_h^n|_{TV(\alpha,\beta)} \leq |u_0|_{TV(\alpha,\beta)},$$

i.e. strong stability in the $|\cdot|_{TV(\alpha,\beta)}$ norm.

When we do not resort to monotone schemes, it is much more difficult to obtain strong stability. In this case, we can limit ourselves to guaranteeing that the total variation of the *local means* be uniformly bounded. (See [Coc98].)

Example 14.4 This example illustrates a typical feature of nonlinear problems, that is how discontinuities can show up even starting from a regular initial datum. To this end, we consider the Burgers equation (14.2) in the $(0, 1)$ interval, with initial datum (see Fig. 14.8)

$$u_0(x) = \begin{cases} 1, & 0 \leq x \leq \frac{5}{12}, \\ 54(2x - \frac{5}{6})^3 - 27(2x - \frac{5}{6})^2 + 1, & \frac{5}{12} < x < \frac{7}{12}, \\ 0, & \frac{7}{12} \leq x \leq 1. \end{cases}$$

It can be easily verified that u_0 , illustrated in Fig. 14.8, is of class $C^1(0, 1)$.

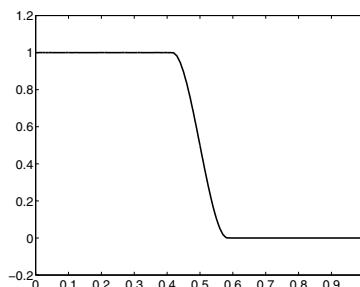


Fig. 14.8. Initial solution u_0 of the first test case of the Burgers problem

We have then considered the numerical solution obtained with the discontinuous Galerkin method, using the third-order Runge-Kutta scheme with a time step of $\Delta t = 10^{-3}$ and $h = 0.01$, for $r = 0$, $r = 1$ and $r = 2$. Fig. 14.9 shows the solution at time $t = 0.5$ obtained with such schemes. We can note the development of a discontinuity, which the numerical scheme solves without oscillations in the case $r = 0$, while for larger values of r we have oscillations in proximity of the discontinuity ■

To eliminate the oscillations in proximity of the solution discontinuities, we can use the *flux limiters* technique, whose description goes beyond the scope of this book. To this end, we refer the reader to the previously cited bibliography. We limit ourselves to saying that the third-order Runge-Kutta scheme (14.20) is modified as follows

$$\begin{aligned}\mathbf{u}_h^* &= \Lambda_h (\mathbf{u}_h^n + \Delta t M_h^{-1} L_h(\mathbf{u}_h^n, t^n)), \\ \mathbf{u}_h^{**} &= \Lambda_h \left(\frac{3}{4} \mathbf{u}_h^n + \frac{1}{4} \mathbf{u}_h^* + \frac{1}{4} \Delta t M_h^{-1} L_h(\mathbf{u}_h^*, t^{n+1}) \right), \\ \mathbf{u}_h^{n+1} &= \Lambda_h \left(\frac{1}{3} \mathbf{u}_h^n + \frac{2}{3} \mathbf{u}_h^{**} + \frac{2}{3} \Delta t M_h^{-1} L_h(\mathbf{u}_h^{**}, t^{n+1/2}) \right),\end{aligned}$$

Λ_h being the flux limiter, that is a function depending also on the variations of the computed solutions, i.e. of the difference between the values of two adjacent nodes. This is equal to the identity operator where the solution is regular, while it limits its variations if these provoke the occurrence of high-frequency oscillations in the numerical solution. Clearly Λ_h must be constructed in a suitable way, in particular

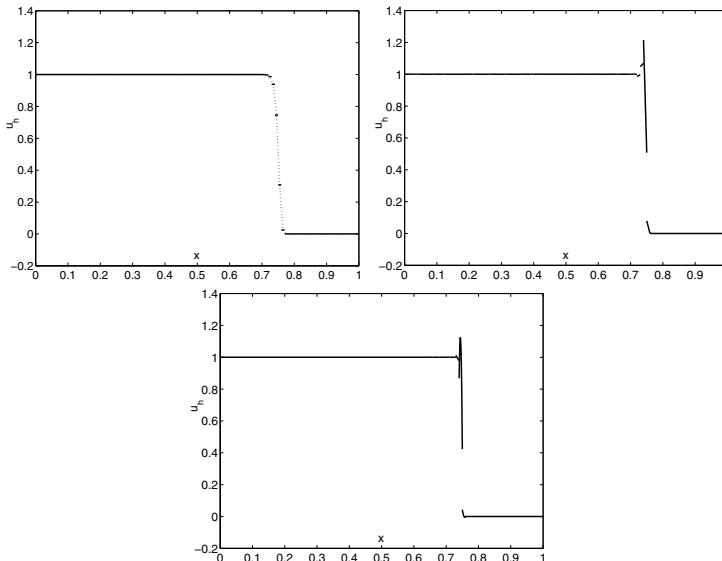


Fig. 14.9. Solution at time $t = 0.5$ of the first test case of the Burgers problem. Comparison between the numerical solution for $r = 0$ (top left), $r = 1$ (top right) and $r = 2$ (bottom). For the $r = 0$ case, the piecewise constant discrete solution has been highlighted by connecting with a dashed line the values at the midpoint of each element

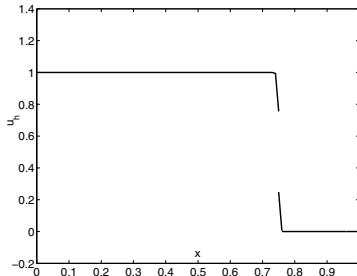


Fig. 14.10. Solution at time $t = 0.5$ for the first test case of the Burgers problem. It has been obtained for $r = 1$ and by applying the flux limiters technique to regularize the numerical solution near the discontinuities

it has to maintain the properties of consistency and conservation of the scheme and must differ as little as possible from the identity operator so as to prevent accuracy degradation.

For the sake of an example, we report in Fig. 14.10 the result obtained with linear discontinuous finite elements ($r = 1$) for the same test case of Fig. 14.9 applying the flux limiters technique. The obtained numerical solution results to be more regular, although slightly more diffusive than that of Fig. 14.9.

Example 14.5 Let us now consider a second problem, where the initial datum is that of Fig. 14.11, obtained by mirroring with respect to the line $x = 0.5$ the datum of the previous test case. By keeping all the remaining parameters of the numerical simulation unchanged, we once again examine the solution at $t = 0.5$. The latter is illustrated in Fig. 14.12. In this case, the solution remains continuous; in fact, with this initial condition, the characteristic lines (which in the case of the Burgers equation are straight lines in the plane (x, t) with slope $\arctan u^{-1}$) never cross. The enlargement allows to qualitatively appreciate the better accuracy of the solution obtained for $r = 2$ with respect to the one obtained for $r = 1$. ■

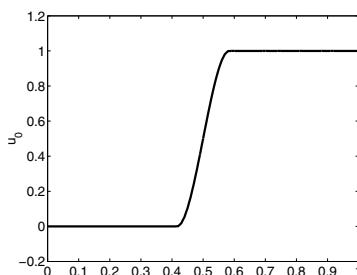


Fig. 14.11. Initial solution u_0 for the second test case of the Burgers problem

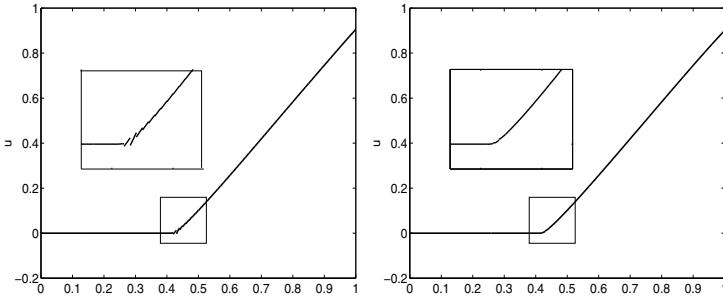


Fig. 14.12. Solution at time $t = 0.5$ for the second test case of the Burgers problem. Comparison between the solution obtained for $r = 1$ (left) and the one obtained for $r = 2$ (right). In the box, we illustrate an enlargement of the numerical solution which allows to qualitatively grasp the improved accuracy obtained for $r = 2$

14.4 Nonlinear hyperbolic systems

In this last section, we briefly address the case of nonlinear hyperbolic systems. A classical example is provided by the Euler equations which are obtained from the following Navier-Stokes equations (for compressible fluids) in \mathbb{R}^d , $d = 1, 2, 3$:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \sum_{j=1}^d \frac{\partial(\rho u_j)}{\partial x_j} &= 0, \\ \frac{\partial(\rho u_i)}{\partial t} + \sum_{j=1}^d \left[\frac{\partial(\rho u_i u_j + \delta_{ij} p)}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \right] &= 0, \quad i = 1, \dots, d, \\ \frac{\partial \rho e}{\partial t} + \sum_{j=1}^d \left[\frac{\partial(\rho h u_j)}{\partial x_j} - \frac{\partial(\sum_{i=1}^d u_i \tau_{ij} + q_j)}{\partial x_j} \right] &= 0. \end{aligned} \quad (14.21)$$

The variables have the following meaning: $\mathbf{u} = (u_1, \dots, u_d)^T$ is the vector of velocities, ρ is the density, p the pressure, $e_i + \frac{1}{2}|\mathbf{u}|^2$ the total energy by mass unit, equal to the sum of the internal energy e_i and of the kinetic energy of the fluid, $h = e + p/\rho$ the total enthalpy per mass unit, \mathbf{q} the thermal flux and finally

$$\tau_{ij} = \mu \left[\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) - \frac{2}{3} \delta_{ij} \operatorname{div} \mathbf{u} \right]$$

the stress tensor (μ being the molecular viscosity of the fluid).

The equations in the above system describe the conservation of mass, momentum and energy, respectively. To complete the system, it is necessary to link e to the variables ρ, p, \mathbf{u} , by defining a law

$$e = \Phi(\rho, p, \mathbf{u}).$$

The latter is normally derived from the state equations of the fluid under exam. In particular, the state equations of the ideal gas

$$p = \rho R T, \quad e_i = C_v T,$$

where $R = C_p - C_v$ is the gas constant and T is its temperature, provide

$$e = \frac{p}{\rho(\gamma - 1)} + \frac{1}{2}|\mathbf{u}|^2,$$

where $\gamma = C_p/C_v$ is the ratio between the specific heats at constant pressure and volume, respectively. The thermal flux \mathbf{q} is usually linked to the temperature gradient via the Fick law

$$\mathbf{q} = -\kappa \nabla T = -\frac{\kappa}{C_v} \nabla(e - \frac{1}{2}|\mathbf{u}|^2),$$

κ being the conductivity of the fluid under exam.

If $\mu = 0$ and $\kappa = 0$, we obtain the Euler equations for non-viscous fluids. The interested reader can find them in specialized fluid dynamics textbooks, or in the textbooks on nonlinear hyperbolic systems, such as for instance [Hir88] or [GR96]. Such equations can be written in compact form in the following way

$$\frac{\partial \mathbf{w}}{\partial t} + \text{Div}F(\mathbf{w}) = \mathbf{0}, \quad (14.22)$$

with $\mathbf{w} = (\rho, \rho\mathbf{u}, \rho e)^T$ being the vector of the so-called *conservative variables*. The flux matrix $F(\mathbf{w})$, a nonlinear function of \mathbf{w} , can be obtained from (14.21). For instance, if $d = 2$, we have

$$F(\mathbf{w}) = \begin{bmatrix} \rho u_1 & \rho u_2 \\ \rho u_1^2 + p & \rho u_1 u_2 \\ \rho u_1 u_2 & \rho u_2^2 + p \\ \rho h u_1 & \rho h u_2 \end{bmatrix}.$$

Finally, in (14.22) Div denotes the divergence operator of a tensor: if $\boldsymbol{\tau}$ is a tensor with components (τ_{ij}) , its divergence is a vector with components

$$(\text{Div}(\boldsymbol{\tau}))_k = \sum_{j=1}^d \frac{\partial}{\partial x_j} (\tau_{kj}), \quad k = 1, \dots, d. \quad (14.23)$$

The form (14.22) is called *conservation form of the Euler equations*. Indeed, by integrating it on any region $\Omega \subset \mathbb{R}^d$ and using the Gauss theorem, we obtain

$$\frac{d}{dt} \int_{\Omega} \mathbf{w} d\Omega + \int_{\partial\Omega} F(\mathbf{w}) \cdot \mathbf{n} d\gamma = 0.$$

This is interpreted by saying that *the variation in time of \mathbf{w} in Ω is balanced by the variation of the fluxes through the boundary of Ω* ; (14.22) is thus a conservation law.

The Navier-Stokes equations can also be written in conservative form as follows

$$\frac{\partial \mathbf{w}}{\partial t} + \text{Div}F(\mathbf{w}) = \text{Div}G(\mathbf{w}),$$

where $G(\mathbf{w})$ are the so-called *viscous fluxes*. Remaining in the $d = 2$ case, these are given by

$$G(\mathbf{w}) = \begin{bmatrix} 0 & 0 \\ \tau_{11} & \tau_{12} \\ \tau_{21} & \tau_{22} \\ \rho hu_1 + \mathbf{u} \cdot \boldsymbol{\tau}_1 + q_1 & \rho hu_2 + \mathbf{u} \cdot \boldsymbol{\tau}_2 + q_2 \end{bmatrix}$$

where $\boldsymbol{\tau}_1 = (\tau_{11}, \tau_{21})^T$ and $\boldsymbol{\tau}_2 = (\tau_{12}, \tau_{22})^T$.

We now rewrite system (14.22) in the form

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{i=1}^d \frac{\partial F_i(\mathbf{w})}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial x_i} = 0. \quad (14.24)$$

This is a particular case of the following problem

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{i=1}^d A_i(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial x_i} = \mathbf{0}, \quad (14.25)$$

called *quasi-linear form*. If the matrix $A^\alpha(\mathbf{w}) = \sum_{i=1}^d \alpha_i A_i(\mathbf{w})$ can be diagonalized for all real values of $\{\alpha_1, \dots, \alpha_d\}$ and its eigenvalues are real and distinct, then system (14.25) is said to be strictly hyperbolic.

Example 14.6 A simple example of a strictly hyperbolic problem is provided by the so-called *p-system*:

$$\begin{aligned} \frac{\partial v}{\partial t} - \frac{\partial u}{\partial x} &= 0, \\ \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} p(v) &= 0. \end{aligned}$$

If $p'(v) < 0$, the two eigenvalues of the jacobian matrix

$$A(\mathbf{w}) = \begin{pmatrix} 0 & -1 \\ p'(v) & 0 \end{pmatrix}$$

are

$$\lambda_1(v) = -\sqrt{-p'(v)} < 0 < \lambda_2(v) = +\sqrt{-p'(v)}. \quad \blacksquare$$

Example 14.7 For the one-dimensional Euler system (i.e. with $d = 1$) we have: $\mathbf{w} = (\rho, \rho u, e)^T$ and $F(\mathbf{w}) = (\rho u, \rho u^2 + p, u(e + p))^T$. The eigenvalues of matrix $A_1(\mathbf{w})$ of the system are $u - c, u, u + c$ where $c = \sqrt{\gamma \frac{p}{\rho}}$ is the speed of sound. As $u, c \in \mathbb{R}$, the eigenvalues are real and distinct and therefore the one-dimensional Euler system is strictly hyperbolic. \blacksquare

The remarks made on the discontinuities of the solution in the scalar case can be extended to the case of systems, by introducing here as well the notion of weak solution. The entropy condition can be extended to the case of systems, using for

instance the proposal from Lax. We observe that in the case of the one-dimensional system (14.22) the Rankine-Hugoniot jump conditions are written in the form

$$F(\mathbf{w}_+) - F(\mathbf{w}_-) = \sigma(\mathbf{w}_+ - \mathbf{w}_-),$$

\mathbf{w}_\pm being the two constant states that represent the values of the unknowns through the discontinuity and σ representing once again the rate at which the discontinuity propagates. Using the fundamental theorem of calculus, such relation is written in the form

$$\begin{aligned} \sigma(\mathbf{w}_+ - \mathbf{w}_-) &= \int_0^1 DF(\theta \mathbf{w}_+ + (1-\theta)\mathbf{w}_-) \cdot (\mathbf{w}_+ - \mathbf{w}_-) d\theta \\ &= A(\mathbf{w}_-, \mathbf{w}_+) \cdot (\mathbf{w}_+ - \mathbf{w}_-), \end{aligned} \quad (14.26)$$

where the matrix

$$A(\mathbf{w}_-, \mathbf{w}_+) = \int_0^1 DF(\theta \mathbf{w}_+ + (1-\theta)\mathbf{w}_-) d\theta$$

represents the mean value of the Jacobian of F (denoted by DF) along the segment connecting \mathbf{w}_- with \mathbf{w}_+ . Relation (14.26) shows that, at each point where a discontinuity occurs, the difference between the right and left state $\mathbf{w}_+ - \mathbf{w}_-$ is an eigenvector of the matrix $A(\mathbf{w}_-, \mathbf{w}_+)$, while the rate of the jump σ coincides with the corresponding eigenvalue $\lambda = \lambda(\mathbf{w}_-, \mathbf{w}_+)$. Calling $\lambda_i(\mathbf{w})$ the i -th eigenvalue of matrix

$$A(\mathbf{w}) = DF(\mathbf{w}),$$

the Lax *admissibility condition* requires that

$$\lambda_i(\mathbf{w}_+) \leq \sigma \leq \lambda_i(\mathbf{w}_-) \quad \text{for each } i.$$

Intuitively, this means that the velocity at which a shock of the i -th family travels must exceed the velocity $\lambda_i(\mathbf{w}_+)$ of the waves that are immediately ahead of the shock, and be less than the velocity $\lambda_i(\mathbf{w}_-)$ of the waves behind the shock.

In the case of hyperbolic systems of m equations ($m > 1$), the entropy η and its corresponding flux Ψ are still scalar functions and relation (14.5) becomes

$$\nabla \Psi(\mathbf{u}) = \frac{d\mathbf{F}}{d\mathbf{u}}(\mathbf{u}) \cdot \nabla \eta(\mathbf{u}), \quad (14.27)$$

which represents a system of m equations and 2 unknowns (η and Ψ). If $m > 2$, such system may have no solutions.

Remark 14.2 In the case of the Euler equations, the entropy function exists also in the $m = 3$ case. •

15

Navier-Stokes equations

Navier-Stokes equations describe the motion of a fluid with constant density ρ in a domain $\Omega \subset \mathbb{R}^d$ (with $d = 2, 3$). They write as follows

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} - \operatorname{div} [\nu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mathbf{f}, & \mathbf{x} \in \Omega, t > 0, \\ \operatorname{div} \mathbf{u} = 0, & \mathbf{x} \in \Omega, t > 0, \end{cases} \quad (15.1)$$

\mathbf{u} being the fluid velocity, p the pressure divided by the density (which will simply be called "pressure"), $\nu = \frac{\mu}{\rho}$ the kinematic viscosity, μ the dynamic viscosity, and \mathbf{f} a forcing term per unit mass that we suppose to belong to the space $L^2(\mathbb{R}^+; [L^2(\Omega)]^d)$ (see Sec. 5.2). The first equation is that of conservation of linear momentum, the second one that of conservation of mass, which is also called the continuity equation. The term $(\mathbf{u} \cdot \nabla) \mathbf{u}$ describes the process of convective transport, while $-\operatorname{div} [\nu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)]$ the process of molecular diffusion. System (15.1) can be derived by the analogous system for compressible flows introduced in Chap. 14 by assuming ρ constant, using the continuity equation (that in the current assumption takes the simplified form $\operatorname{div} \mathbf{u} = 0$) to simplify the various terms, and finally dividing the equation by ρ . Note that in the incompressible case (15.2) the energy equation has disappeared. Indeed, even though such an equation can still be written for incompressible flows, its solution can be carried out independently once the velocity field is obtained from the solution of (15.1).

When ν is constant, from the continuity equation we obtain

$$\operatorname{div} [\nu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] = \nu (\Delta \mathbf{u} + \nabla \operatorname{div} \mathbf{u}) = \nu \Delta \mathbf{u}$$

whence the system (15.1) can be written in the equivalent form

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mathbf{f}, & \mathbf{x} \in \Omega, t > 0, \\ \operatorname{div} \mathbf{u} = 0, & \mathbf{x} \in \Omega, t > 0, \end{cases} \quad (15.2)$$

which is the one that we will consider in this chapter.

Equations (15.2) are often called incompressible Navier-Stokes equations. More in general, fluids satisfying the *incompressibility condition* $\operatorname{div} \mathbf{u} = 0$ are said to be incompressible. Constant density fluids necessarily satisfy this condition, however there exist incompressible fluids featuring variable density (e.g., stratified fluids) that are governed by a different system of equations in which the density ρ explicitly shows up. This case will be faced in Sec. 15.9.

In order for problem (15.2) to be well posed it is necessary to assign the initial condition

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega, \quad (15.3)$$

where \mathbf{u}_0 is a given divergence-free vector field, together with suitable boundary conditions, such as, e.g., $\forall t > 0$,

$$\begin{cases} \mathbf{u}(\mathbf{x}, t) = \varphi(\mathbf{x}, t) & \forall \mathbf{x} \in \Gamma_D, \\ \left(\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} \right) (\mathbf{x}, t) = \psi(\mathbf{x}, t) & \forall \mathbf{x} \in \Gamma_N, \end{cases} \quad (15.4)$$

where φ and ψ are given vector functions, while Γ_D and Γ_N provide a partition of the domain boundary $\partial\Omega$, that is $\Gamma_D \cup \Gamma_N = \partial\Omega$, $\overset{\circ}{\Gamma}_D \cap \overset{\circ}{\Gamma}_N = \emptyset$. Finally, as usual \mathbf{n} is the outward unit normal vector to $\partial\Omega$. Should we use the alternative formulation (15.1), the second equation in (15.4) must be replaced by

$$\left[\frac{1}{2} \nu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \cdot \mathbf{n} - p \mathbf{n} \right] (\mathbf{x}, t) = \psi(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Gamma_N.$$

Further considerations on boundary conditions are made later in Sec. 15.9.2.

Denoting with u_i , $i = 1, \dots, d$ the components of the vector \mathbf{u} w.r.t. a cartesian frame, with f_i those of \mathbf{f} , system (15.2) can be written componentwise as

$$\begin{cases} \frac{\partial u_i}{\partial t} - \nu \Delta u_i + \sum_{j=1}^d u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} = f_i, & i = 1, \dots, d, \\ \sum_{j=1}^d \frac{\partial u_j}{\partial x_j} = 0. \end{cases}$$

In the two-dimensional case, Navier-Stokes equations with the boundary conditions previously indicated yield well-posed problems. This means that if all of the data (initial condition, forcing term, boundary data) are smooth enough, then the solution is continuous together with its derivatives and does not develop singularities in time. Things may go differently in three dimensions, where existence and uniqueness of classical solutions have been proven only locally in time (that is for a sufficiently small time interval). In the following section we will introduce the weak formulation of Navier-Stokes equations: in such case a global in time existence result has been proven. However, the issue of uniqueness (which is related to that of regularity) is still open, and is actually the central issue of Navier-Stokes theory.

Remark 15.1 The Navier-Stokes equations have been written in terms of the *primitive variables* \mathbf{u} and p . Other set of variables can however be used, for instance, in the two-dimensional case it is often used the couple given by the vorticity ω and the streamfunction ψ , that are related to the velocity as follows

$$\omega = \text{rot}\mathbf{u} = \frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2}, \quad \mathbf{u} = \begin{bmatrix} \frac{\partial \psi}{\partial x_2} \\ -\frac{\partial \psi}{\partial x_1} \end{bmatrix}.$$

The various formulations are in fact equivalent from a mathematical standpoint, however they give rise to different numerical methods. See, e.g., [Qua93]. •

15.1 Weak formulation of Navier-Stokes equations

A weak formulation of problem (15.2)-(15.4) can be obtained by formally proceeding as follows. Let us multiply the first equation of (15.2) by a test function \mathbf{v} belonging to a suitable space V that will be specified later on, and integrate it on Ω

$$\int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} d\Omega - \int_{\Omega} \nu \Delta \mathbf{u} \cdot \mathbf{v} d\Omega + \int_{\Omega} [(\mathbf{u} \cdot \nabla) \mathbf{u}] \cdot \mathbf{v} d\Omega + \int_{\Omega} \nabla p \cdot \mathbf{v} d\Omega = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega.$$

Using the Green formulae (3.16) and (3.17) we find:

$$\begin{aligned} - \int_{\Omega} \nu \Delta \mathbf{u} \cdot \mathbf{v} d\Omega &= \int_{\Omega} \nu \nabla \mathbf{u} \cdot \nabla \mathbf{v} d\Omega - \int_{\partial\Omega} \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \cdot \mathbf{v} d\gamma, \\ \int_{\Omega} \nabla p \cdot \mathbf{v} d\Omega &= - \int_{\Omega} p \operatorname{div} \mathbf{v} d\Omega + \int_{\partial\Omega} p \mathbf{v} \cdot \mathbf{n} d\gamma. \end{aligned}$$

Using these relations in the first of (15.2), we obtain

$$\begin{aligned} \int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} d\Omega + \int_{\Omega} \nu \nabla \mathbf{u} \cdot \nabla \mathbf{v} d\Omega + \int_{\Omega} [(\mathbf{u} \cdot \nabla) \mathbf{u}] \cdot \mathbf{v} d\Omega - \int_{\Omega} p \operatorname{div} \mathbf{v} d\Omega \\ = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega + \int_{\partial\Omega} \left(\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} \right) \cdot \mathbf{v} d\gamma \quad \forall \mathbf{v} \in V. \end{aligned} \tag{15.5}$$

(All the boundary integrals should indeed be regarded as duality pairings.)

Similarly, by multiplying the second equation of (15.2) by a test function q , belonging to a suitable space Q to be specified, then integrating on Ω it follows

$$\int_{\Omega} q \operatorname{div} \mathbf{u} d\Omega = 0 \quad \forall q \in Q. \tag{15.6}$$

As customarily, V is chosen in such a way that the test functions vanish on that boundary portion where a Dirichlet data is prescribed on \mathbf{u} , that is

$$V = [\mathbf{H}_{\Gamma_D}^1(\Omega)]^d = \{\mathbf{v} \in [\mathbf{H}^1(\Omega)]^d : \mathbf{v}|_{\Gamma_D} = \mathbf{0}\}. \quad (15.7)$$

It will coincide with $[\mathbf{H}_0^1(\Omega)]^d$ if $\Gamma_D = \partial\Omega$. If $\Gamma_N \neq \emptyset$, then it can be chosen $Q = \mathbf{L}^2(\Omega)$. Moreover, if $t > 0$, $\mathbf{u}(t) \in [\mathbf{H}^1(\Omega)]^d$, with $\mathbf{u}(t) = \varphi(t)$ on Γ_D , $\mathbf{u}(0) = \mathbf{u}_0$ and $p(t) \in Q$.

Having chosen these functional spaces, we can note first of all that

$$\int_{\partial\Omega} \left(\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} \right) \cdot \mathbf{v} d\gamma = \int_{\Gamma_N} \psi \cdot \mathbf{v} d\gamma \quad \forall \mathbf{v} \in V.$$

Moreover, all the integrals involving bilinear terms are finite. More precisely, by using the vector notation $\mathbf{H}^k(\Omega) = [\mathbf{H}^k(\Omega)]^d$, $\mathbf{L}^p(\Omega) = [\mathbf{L}^p(\Omega)]^d$, $k \geq 1$, $1 \leq p < \infty$, we find:

$$\begin{aligned} \left| \nu \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} \right| &\leq \nu |\mathbf{u}|_{\mathbf{H}^1(\Omega)} |\mathbf{v}|_{\mathbf{H}^1(\Omega)}, \\ \left| \int_{\Omega} p \operatorname{div} \mathbf{v} d\Omega \right| &\leq \|p\|_{\mathbf{L}^2(\Omega)} |\mathbf{v}|_{\mathbf{H}^1(\Omega)}, \\ \left| \int_{\Omega} q \nabla \mathbf{u} \cdot \mathbf{d} d\Omega \right| &\leq \|q\|_{\mathbf{L}^2(\Omega)} |\mathbf{u}|_{\mathbf{H}^1(\Omega)}. \end{aligned}$$

For every function $\mathbf{v} \in \mathbf{H}^1(\Omega)$, we denote by

$$\|\mathbf{v}\|_{\mathbf{H}^1(\Omega)} = \left(\sum_{k=1}^d \|v_k\|_{\mathbf{H}^1(\Omega)}^2 \right)^{1/2}$$

its norm and by

$$|\mathbf{v}|_{\mathbf{H}^1(\Omega)} = \left(\sum_{k=1}^d |v_k|_{\mathbf{H}^1(\Omega)}^2 \right)^{1/2}$$

its seminorm. The notation $\|\mathbf{v}\|_{\mathbf{L}^p(\Omega)}$, $1 \leq p < \infty$, has a similar meaning. Same symbols will be used in case of tensor functions. Thanks to Poincaré inequality, $|\mathbf{v}|_{\mathbf{H}^1(\Omega)}$ is equivalent to the norm $\|\mathbf{v}\|_{\mathbf{H}^1(\Omega)}$ for all functions belonging to V .

Also the integral involving the trilinear term is finite. To see it, let us start by recalling the following result (see (2.18); for its proof, see [Ada75]): if $d \leq 3$, $\exists C > 0$ s.t.

$\forall \mathbf{v} \in \mathbf{H}^1(\Omega)$ we have $\mathbf{v} \in \mathbf{L}^4(\Omega)$ and moreover $\|\mathbf{v}\|_{\mathbf{L}^4(\Omega)} \leq C \|\mathbf{v}\|_{\mathbf{H}^1(\Omega)}$.

Using the following three-term Hölder inequality

$$\left| \int_{\Omega} fgh d\Omega \right| \leq \|f\|_{\mathbf{L}^p(\Omega)} \|g\|_{\mathbf{L}^q(\Omega)} \|h\|_{\mathbf{L}^r(\Omega)},$$

valid for all $p, q, r > 1$ s.t. $p^{-1} + q^{-1} + r^{-1} = 1$, we conclude that

$$\left| \int_{\Omega} [(\mathbf{u} \cdot \nabla) \mathbf{u}] \cdot \mathbf{v} d\Omega \right| \leq \|\nabla \mathbf{u}\|_{\mathbf{L}^2(\Omega)} \|\mathbf{u}\|_{\mathbf{L}^4(\Omega)} \|\mathbf{v}\|_{\mathbf{L}^4(\Omega)} \leq C^2 \|\mathbf{u}\|_{\mathbf{H}^1(\Omega)}^2 \|\mathbf{v}\|_{\mathbf{H}^1(\Omega)}.$$

As of uniqueness of solution, let us consider again the Navier-Stokes equations in strong form (15.2) (similar considerations can be made on the weak form (15.5), (15.6)). If $\Gamma_D = \partial\Omega$, that is only boundary conditions of Dirichlet type are imposed, the pressure would only appear through its gradient; in such a case, should (\mathbf{u}, p) be a solution of (15.2), for any possible constant c the couple $(\mathbf{u}, p+c)$ would be a solution too, since $\nabla(p+c) = \nabla p$. To avoid such indeterminacy one could fix a priori the value of p at one given point \mathbf{x}_0 of the domain Ω , that is set $p(\mathbf{x}_0) = p_0$, or, alternatively, requires the pressure to have null average, i.e., $\int_{\Omega} p \, d\Omega = 0$. The former condition requires to prescribe a pointwise value for the pressure, but this is inconsistent with our ansatz that $p \in L^2(\Omega)$. (We anticipate, however, that this is admissible at the numerical level when we look for a continuous finite dimensional pressure.) For this reason we assume from now on that the pressure be average free. More specifically, we will consider the following pressure space

$$Q = L_0^2(\Omega) = \{p \in L^2(\Omega) : \int_{\Omega} p \, d\Omega = 0\}.$$

Further, we observe that if $\Gamma_D = \partial\Omega$, the prescribed Dirichlet data φ must be compatible with the incompressibility constraint; indeed,

$$\int_{\partial\Omega} \varphi \cdot \mathbf{n} \, d\gamma = \int_{\Omega} \operatorname{div} \mathbf{u} \, d\Omega = 0.$$

In the case in which Γ_N is not empty, that is we consider either Neumann or mixed Dirichlet-Neumann boundary conditions, the problem of pressure indeterminacy (up to an additive constant) previously mentioned, no longer exists, as on Γ_N the unknown p appears without derivatives. In this case we can take $Q = L^2(\Omega)$. In conclusion, from now on it will be understood that

$$Q = L^2(\Omega) \quad \text{if } \Gamma_N \neq \emptyset, \quad Q = L_0^2(\Omega) \quad \text{if } \Gamma_N = \emptyset. \quad (15.8)$$

The weak formulation of the system (15.2), (15.3), (15.4) is therefore:

find $\mathbf{u} \in L^2(\mathbb{R}^+; [H^1(\Omega)]^d) \cap C^0(\mathbb{R}^+; [L^2(\Omega)]^d)$, $p \in L^2(\mathbb{R}^+; Q)$ s.t.

$$\left\{ \begin{array}{l} \int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} \, d\Omega + \nu \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, d\Omega + \int_{\Omega} [(\mathbf{u} \cdot \nabla) \mathbf{u}] \cdot \mathbf{v} \, d\Omega - \int_{\Omega} p \operatorname{div} \mathbf{v} \, d\Omega \\ \qquad \qquad \qquad = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\Omega + \int_{\Gamma_N} \psi \cdot \mathbf{v} \, d\gamma \quad \forall \mathbf{v} \in V, \\ \int_{\Omega} q \operatorname{div} \mathbf{u} \, d\Omega = 0 \quad \forall q \in Q, \end{array} \right. \quad (15.9)$$

with $\mathbf{u}|_{\Gamma_D} = \boldsymbol{\varphi}_D$ and $\mathbf{u}|_{t=0} = \mathbf{u}_0$. The space V is the one in (15.7) while Q is the space introduced in (15.8). The spaces of time-dependent functions for \mathbf{u} and p have been introduced in Sec. 2.7.

As we have already anticipated, existence of solutions can be proven for this problem for both dimensions $d = 2$ and $d = 3$, whereas uniqueness has been proven only in the case $d = 2$ for sufficiently small data (see, e.g., [Tem01] and [Sal08]).

Let us define the Reynolds number

$$Re = \frac{|\mathbf{U}|L}{\nu},$$

where L is a representative length of the domain Ω (e.g. the length of a channel wherein the fluid flow is studied) and \mathbf{U} a representative fluid velocity.

The Reynolds number measures the extent at which convection dominates over diffusion. When $Re \ll 1$ the convective term $(\mathbf{u} \cdot \nabla)\mathbf{u}$ can be omitted, and Navier-Stokes equations reduce to the so-called Stokes equations that will be investigated later in this chapter. On the other side, if Re is large, problems may arise concerning uniqueness of solution, the existence of stationary and stable solutions, the possible existence of strange attractors, the transition towards turbulent flows. See [Tem01], [FMRT01].

When fluctuations of flow velocity occur at very small spatial and temporal scales, their numerical approximation becomes very difficult if not impossible. In those cases one typically resorts to the so-called *turbulence models*: the latter allow the approximate description of this flow behavior through either algebraic or differential equations. This subject, however, will not be addressed in this monograph. The interested readers may consult, e.g., [Wil98] for a description of the physical aspects of turbulence flows, [HYR08] for multiscale analysis of incompressible flows, [Le 05] for modelling aspects of multiscale systems, [MP94] for the analysis of one of the most widely used turbulence models, the so-called $\kappa - \epsilon$ model, while [Sag06] and [BIL06] provide the analysis of the so-called *Large Eddy* model, which is more computationally expensive but in principle better suited to provide a more realistic description of turbulent flow fields.

Finally, let us mention the Euler equations introduced in (14.21), which are used for both compressible or incompressible flows in those cases in which the viscosity can be neglected. Formally speaking, this corresponds to taking the Reynolds number equal to infinity.

By eliminating the pressure, Navier-Stokes equations can be rewritten in *reduced form* in the sole variable \mathbf{u} . This can be shown by starting from the weak formulation (15.9) and using the following subspaces of the functional space $[H^1(\Omega)]^d$,

$$V_{\text{div}} = \{\mathbf{v} \in [H^1(\Omega)]^d : \operatorname{div} \mathbf{v} = 0\}, \quad V_{\text{div}}^0 = \{\mathbf{v} \in V_{\text{div}} : \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D\}.$$

Upon requiring the test function \mathbf{v} in the momentum equation in (15.9) to belong to the space V_{div} , the term associated to the pressure gradient vanishes, whence we find the following reduced problem for the velocity

find $\mathbf{u} \in L^2(\mathbb{R}^+; V_{\text{div}}) \cap C^0(\mathbb{R}^+; [L^2(\Omega)]^d)$ s.t.

$$\begin{aligned} \int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} \, d\Omega &+ \nu \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, d\Omega + \int_{\Omega} [(\mathbf{u} \cdot \nabla) \mathbf{u}] \cdot \mathbf{v} \, d\Omega \\ &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\Omega + \int_{\Gamma_N} \psi \cdot \mathbf{v} \, d\gamma \quad \forall \mathbf{v} \in V_{\text{div}}^0, \end{aligned} \quad (15.10)$$

with $\mathbf{u}|_{\Gamma_D} = \varphi_D$ and $\mathbf{u}|_{t=0} = \mathbf{u}_0$. Since this problem is a (nonlinear) parabolic one, its analysis can be carried out using techniques similar to those applied in Chap. 5. (See, e.g., [Sal08].)

Obviously, should \mathbf{u} be a solution of (15.9), then it would also be a solution to (15.10). Conversely, the following theorem holds. For its proof, see, e.g., [QV94].

Theorem 15.1 *Let $\Omega \subset \mathbb{R}^d$ be a domain with Lipschitz continuous boundary $\partial\Omega$. Let \mathbf{u} be a solution to the reduced problem (15.10). Then there exists a unique function $p \in L^2(\mathbb{R}^+; Q)$ s.t. (\mathbf{u}, p) is a solution to (15.9).*

Once the reduced problem is solved, there exists therefore a unique way to recover the pressure p , and henceforth the complete solution of the original Navier-Stokes problem (15.9).

In practice, however, this approach can be quite unsuitable from a numerical viewpoint. Indeed, in a Galerkin spatial approximation framework, it would require the construction of finite dimensional subspace, say $V_{\text{div},h}$, of *divergence-free* velocity functions. In this regard, see, e.g., [BF91a] for finite element approximations, and [CHQZ06] for spectral approximations. Moreover, the result of Theorem 15.1 is not constructive, as it does not provide a way to build the pressure solution p . For these reasons it is generally preferred to directly approximate the complete coupled problem (15.9).

15.2 Stokes equations and their approximation

In this section we will consider the following *generalized Stokes problem* with homogeneous Dirichlet boundary conditions

$$\left\{ \begin{array}{ll} \alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega, \\ \operatorname{div} \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u} = \mathbf{0} & \text{on } \partial\Omega, \end{array} \right. \quad (15.11)$$

for a given coefficient $\alpha \geq 0$. This problem describes the motion of an incompressible viscous flow in which the (quadratic) convective term has been neglected, a simplification that is acceptable when $Re \ll 1$. However, one can generate a problem like

(15.11) also while using an implicit temporal discretization of the Navier-Stokes equations, as we will see in Sec. 15.7.

From an analytical standpoint, Navier-Stokes equations can be regarded as a *compact perturbation* of the Stokes equations, as they differ from the latter solely because of the presence of the convective term, which is first order whereas the diffusive term is second order. On the other hand, it is fair to say that this term can have a fundamental impact on the solution behavior when the Reynolds number Re is very large, as already pointed out.

The weak formulation of the (15.11) problem reads:

find $\mathbf{u} \in V$ and $p \in Q$ such that

$$\begin{cases} \int_{\Omega} (\alpha \mathbf{u} \cdot \mathbf{v} + \nu \nabla \mathbf{u} \cdot \nabla \mathbf{v}) d\Omega - \int_{\Omega} p \operatorname{div} \mathbf{v} d\Omega = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega & \forall \mathbf{v} \in V, \\ \int_{\Omega} q \operatorname{div} \mathbf{u} d\Omega = 0 & \forall q \in Q, \end{cases} \quad (15.12)$$

where $V = [H_0^1(\Omega)]^d$ and $Q = L_0^2(\Omega)$. (We recall that we are addressing a homogeneous Dirichlet problem for the velocity field \mathbf{u} and that we are looking for a pressure having null average.) Now define the bilinear forms $a : V \times V \mapsto \mathbb{R}$ and $b : V \times Q \mapsto \mathbb{R}$ as follows:

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} (\alpha \mathbf{u} \cdot \mathbf{v} + \nu \nabla \mathbf{u} \cdot \nabla \mathbf{v}) d\Omega, \\ b(\mathbf{u}, q) &= - \int_{\Omega} q \operatorname{div} \mathbf{u} d\Omega. \end{aligned} \quad (15.13)$$

With these notations, problem (15.12) becomes

find $(\mathbf{u}, p) \in V \times Q$ s.t.

$$\begin{cases} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = (\mathbf{f}, \mathbf{v}) & \forall \mathbf{v} \in V, \\ b(\mathbf{u}, q) = 0 & \forall q \in Q, \end{cases} \quad (15.14)$$

where $(\mathbf{f}, \mathbf{v}) = \sum_{i=1}^d \int_{\Omega} f_i v_i d\Omega$.

Should we consider non-homogeneous boundary conditions, as indicated in (15.4), the weak formulation of the Stokes problem would become:

find $(\overset{\circ}{\mathbf{u}}, p) \in V \times Q$ s.t.

$$\begin{cases} a(\overset{\circ}{\mathbf{u}}, \mathbf{v}) + b(\mathbf{v}, p) = \mathbf{F}(\mathbf{v}) & \forall \mathbf{v} \in V, \\ b(\overset{\circ}{\mathbf{u}}, q) = G(q) & \forall q \in Q, \end{cases} \quad (15.15)$$

where V and Q are the spaces introduced in (15.7) and (15.8), respectively. Having denoted with $\mathbf{R}\varphi \in [H^1(\Omega)]^d$ a lifting of the boundary datum φ , we have set $\bar{\mathbf{u}} = \mathbf{u} - \mathbf{R}\varphi$, while the new terms on the right hand side have the following expression:

$$\mathbf{F}(\mathbf{v}) = (\mathbf{f}, \mathbf{v}) + \int_{\Gamma_N} \psi \mathbf{v} \, d\gamma - a(\mathbf{R}\varphi, \mathbf{v}), \quad G(q) = -b(\mathbf{R}\varphi, q). \quad (15.16)$$

The following result holds:

Theorem 15.2 *The couple (\mathbf{u}, p) solves the Stokes problem (15.14) iff it is a saddle-point of the Lagrangian functional*

$$\mathcal{L}(\mathbf{v}, q) = \frac{1}{2}a(\mathbf{v}, \mathbf{v}) + b(\mathbf{v}, q) - (\mathbf{f}, \mathbf{v}),$$

that is iff

$$\mathcal{L}(\mathbf{u}, p) = \min_{\mathbf{v} \in V} \max_{q \in Q} \mathcal{L}(\mathbf{v}, q).$$

The pressure q henceforth plays the role of Lagrange multiplier associated to the divergence-free constraint. As done in Sec. 15.1 for the Navier-Stokes equations, it is possible to formally eliminating the variable p from the Stokes equations, thus yielding the following reduced Stokes problem (in weak form)

$$\text{find } \mathbf{u} \in V_{\text{div}}^0 \quad \text{s.t.} \quad a(\mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in V_{\text{div}}^0. \quad (15.17)$$

This is an elliptic problem for the vector variable \mathbf{u} . Existence and uniqueness can be proven by virtue of Lax-Milgram Lemma 3.1. As a matter of fact, V_{div}^0 is a Hilbert space with respect to the norm $\|\nabla \mathbf{v}\|_{L^2(\Omega)}$. This because the divergence operator is continuous from V into $L^2(\Omega)$, thus V_{div}^0 is a closed subspace of the space V . Moreover, the bilinear form $a(\cdot, \cdot)$ is continuous and coercive in V_{div}^0 , and $\mathbf{f} \in V_{\text{div}}'$. Using Cauchy-Schwarz and Poincaré inequalities, the following estimates can be drawn by taking $\mathbf{v} = \mathbf{u}$ in (15.17)):

$$\begin{aligned} \frac{\alpha}{2} \|\mathbf{u}\|_{L^2(\Omega)}^2 + \nu \|\nabla \mathbf{u}\|_{L^2(\Omega)}^2 &\leq \frac{1}{2\alpha} \|\mathbf{f}\|_{L^2(\Omega)}^2, & \text{if } \alpha \neq 0, \\ \|\nabla \mathbf{u}\|_{L^2(\Omega)} &\leq \frac{C_\Omega}{\nu} \|f\|_{L^2(\Omega)}, & \text{if } \alpha = 0, \end{aligned}$$

where C_Ω is the constant of the Poincaré inequality (2.13). Note that the pressure has disappeared from (15.17). However, still from (15.17) we can infer that the vector $\mathbf{w} = \alpha \mathbf{u} - \nu \Delta \mathbf{u} - \mathbf{f}$, regarded as a linear functional of $H^{-1}(\Omega)$, vanishes when applied to any vector function of V_{div}^0 . Thanks to this property, it can be proven that there exists a unique function $p \in Q$ such that $\mathbf{w} = \nabla p$, that is p satisfies the first equation of (15.11) in the distributional sense (see, e.g., [QV94]). The couple (\mathbf{u}, p) is therefore the unique solution of the weak problem (15.14).

The Galerkin approximation of problem (15.14) has the following form:
find $(\mathbf{u}_h, p_h) \in V_h \times Q_h$ s.t.

$$\begin{cases} a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = (\mathbf{f}, \mathbf{v}_h) & \forall \mathbf{v}_h \in V_h, \\ b(\mathbf{u}_h, q_h) = 0 & \forall q_h \in Q_h, \end{cases} \quad (15.18)$$

where $\{V_h \subset V\}$ and $\{Q_h \subset Q\}$ represent two families of finite dimensional subspaces depending on a real discretization parameter h . Should we consider instead problem (15.15)-(15.16) corresponding to non-homogeneous boundary data (15.4), the above formulation needs to be modified by using on the right hand side of the first equation $\mathbf{F}(\mathbf{v}_h)$, and on that of the second equation $G(q_h)$. These new functionals can be obtained from (15.16) upon replacing $\mathbf{R}\varphi$ with the interpolant of φ at the nodes of Γ_D (and vanishing at all the other nodes), and ψ with its interpolant at the nodes sitting on Γ_N . The algebraic formulation of problem (15.18) will be addressed in Sec. 15.4.

The following theorem is due to F. Brezzi [Bre74], and guarantees uniqueness and solution of problem (15.18):

Theorem 15.3 *The Galerkin approximation (15.18) admits one and only one solution if the following conditions hold:*

1. *The bilinear form $a(\cdot, \cdot)$ is:*

a) *coercive, that is $\exists \alpha > 0$ (possibly depending on h) s.t.*

$$a(\mathbf{v}_h, \mathbf{v}_h) \geq \alpha \|\mathbf{v}_h\|_V^2 \quad \forall \mathbf{v}_h \in V_h^*,$$

where $V_h^* = \{\mathbf{v}_h \in V_h : b(\mathbf{v}_h, q_h) = 0 \forall q_h \in Q_h\}$;

b) *continuous, that is $\exists \gamma > 0$ s.t.*

$$|a(\mathbf{u}_h, \mathbf{v}_h)| \leq \gamma \|\mathbf{u}_h\|_V \|\mathbf{v}_h\|_V \quad \forall \mathbf{u}_h, \mathbf{v}_h \in V_h.$$

2. *The bilinear form $b(\cdot, \cdot)$ is continuous, that is $\exists \delta > 0$ s.t.*

$$|b(\mathbf{v}_h, q_h)| \leq \delta \|\mathbf{v}_h\|_V \|q_h\|_Q \quad \forall \mathbf{v}_h \in V_h, q_h \in Q_h.$$

3. *Finally, there exists a positive constant β (possibly depending on h) s.t.*

$$\forall q_h \in Q_h, \exists \mathbf{v}_h \in V_h : b(\mathbf{v}_h, q_h) \geq \beta \|\mathbf{v}_h\|_{\mathbf{H}^1(\Omega)} \|q_h\|_{L^2(\Omega)}. \quad (15.19)$$

Under the previous assumptions the following a priori estimates hold on the discrete solution:

$$\|\mathbf{u}_h\|_V \leq \frac{1}{\alpha} \|\mathbf{f}\|_{V'},$$

$$\|p_h\|_Q \leq \frac{1}{\beta} \left(1 + \frac{\gamma}{\alpha}\right) \|\mathbf{f}\|_{V'},$$

where, as usual, V' denotes the dual space of V . Moreover, the following convergence results hold:

$$\begin{aligned}\|\mathbf{u} - \mathbf{u}_h\|_V &\leq \left(1 + \frac{\gamma}{\beta}\right) \left(1 + \frac{\gamma}{\alpha}\right) \inf_{\mathbf{v}_h \in V_h} \|\mathbf{u} - \mathbf{v}_h\|_V + \frac{\delta}{\alpha} \inf_{q_h \in Q_h} \|p - q_h\|_Q, \\ \|p - p_h\|_Q &\leq \frac{\gamma}{\beta} \left(1 + \frac{\gamma}{\alpha}\right) \inf_{\mathbf{v}_h \in V_h} \|\mathbf{u} - \mathbf{v}_h\|_V \\ &\quad + \left(1 + \frac{\delta}{\beta} + \frac{\delta\gamma}{\alpha\beta}\right) \inf_{q_h \in Q_h} \|p - q_h\|_Q.\end{aligned}$$

It is worth noticing that condition (15.19) is equivalent to assume that there exists a positive constant β s.t.

$$\inf_{q_h \in Q_h, q_h \neq 0} \sup_{\mathbf{v}_h \in V_h, \mathbf{v}_h \neq 0} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{\mathbf{H}^1(\Omega)} \|q_h\|_{L^2(\Omega)}} \geq \beta. \quad (15.20)$$

For such a reason it is often called the *inf-sup condition*.

The proof of this theorem requires non-elementary tools of functional analysis. It will be given in Sec. 15.3 for a saddle-point problem that is more general than Stokes' problem. In this perspective, Theorem 15.3 can be regarded as a special case of Theorems 15.5 and 15.6. The reader uninterested to these theoretical aspects can skip the next section and jump directly to Sec. 15.4.

15.3 Saddle-point problems

Scope of this section is the study of problems (15.14) and (15.18) and the proof of a convergence result for the solution of the discrete problem (15.18) to that of problem (15.14). With this aim we will recast those formulations within a more abstract framework that will eventually allow the use of the theory proposed in [Bre74], [Bab71].

15.3.1 Problem formulation

Let X and M be two Hilbert spaces endowed respectively with the norms $\|\cdot\|_X$ and $\|\cdot\|_M$. Denoting with X' and M' the corresponding dual spaces (that is the spaces of linear and bounded functionals defined respectively on X and M), we introduce the bilinear forms $a(\cdot, \cdot) : X \times X \rightarrow \mathbb{R}$ and $b(\cdot, \cdot) : X \times M \rightarrow \mathbb{R}$ that we suppose to be continuous, that is there exist two constants $\gamma, \delta > 0$ s.t.

$$|a(w, v)| \leq \gamma \|w\|_X \|v\|_X, \quad |b(w, \mu)| \leq \delta \|w\|_X \|\mu\|_M, \quad (15.21)$$

for all $w, v \in X$ and $\mu \in M$.

Consider now the following constrained problem: find $(u, \eta) \in X \times M$ s.t.

$$\begin{cases} a(u, v) + b(v, \eta) = \langle l, v \rangle & \forall v \in X, \\ b(u, \mu) = \langle \sigma, \mu \rangle & \forall \mu \in M, \end{cases} \quad (15.22)$$

where $l \in X'$ and $\sigma \in M'$ are two assigned linear functionals, while $\langle \cdot, \cdot \rangle$ denotes the duality between X and X' or that between M and M' .

The formulation (15.22) is general enough to include the formulation (15.14) of the Stokes problem, that of a generic constrained problem w.r.t. the bilinear form $a(\cdot, \cdot)$ (with η representing the constraint), or again the formulation which is obtained when mixed finite element approximations are used for various kind of boundary-value problems, for instance those of linear elasticity (see, e.g., [BF91a], [QV94]).

Problem (15.22) can be conveniently restated in operator form. With this aim, we associate the bilinear forms $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ with the operators $A \in \mathcal{L}(X, X')$ and $B \in \mathcal{L}(X, M')$, respectively, defined through the following relations:

$$\begin{aligned} \langle Aw, v \rangle &= a(w, v) & \forall w, v \in X, \\ \langle Bv, \mu \rangle &= b(v, \mu) & \forall v \in X, \mu \in M. \end{aligned}$$

Duality is between X' and X in the former relation, M' and M in the latter. In accordance with notations introduced in Sec. 4.5.3, we denote by $\mathcal{L}(V, W)$ the space of linear and bounded functionals between V and W .

Let $B^T \in \mathcal{L}(M, X')$ be the adjoint operator of B befinde by

$$\langle B^T \mu, v \rangle = \langle Bv, \mu \rangle = b(v, \mu) \quad \forall v \in X, \mu \in M. \quad (15.23)$$

The former duality holds between X' and X , the latter between M' and M . (This operator was denoted by the symbol B' in Sec. 2.6, see the general definition (2.19). Here, however, it is denoted by B^T to conform with the classical notation used in the framework of saddle-point problems.)

In operator form, the saddle-point problem (15.22) can be restated as follows:
find $(u, \eta) \in X \times M$ s.t.

$$\begin{cases} Au + B^T \eta = l & \text{in } X', \\ Bu = \sigma & \text{in } M'. \end{cases} \quad (15.24)$$

15.3.2 Problem analysis

In order to analyze problem (15.24), we introduce the affine manifold

$$X^\sigma = \{v \in X : b(v, \mu) = \langle \sigma, \mu \rangle \forall \mu \in M\}. \quad (15.25)$$

The space X^0 identifies the kernel of B , that is

$$X^0 = \{v \in X : b(v, \mu) = 0 \forall \mu \in M\} = \ker(B).$$

This is a closed subspace of X . We can therefore associate (15.22) with the following reduced problem

$$\text{find } u \in X^\sigma \quad \text{s.t.} \quad a(u, v) = \langle l, v \rangle \quad \forall v \in X^0. \quad (15.26)$$

Should (u, η) be a solution of (15.22), then u would be a solution to (15.26). In the following we will introduce suitable conditions that allow the converse to hold too. Moreover, we would like to prove uniqueness for the solution of (15.26). This would allow us to conclude with an existence and uniqueness result for the original saddle-point problem (15.22).

We denote by X_{polar}^0 the polar set of X^0 , that is

$$X_{polar}^0 = \{g \in X' : \langle g, v \rangle = 0 \quad \forall v \in X^0\}.$$

Since $X^0 = \ker(B)$, we have $X_{polar}^0 = (\ker(B))_{polar}$. The space X is a direct sum of X^0 and its orthogonal space $(X^0)^\perp$,

$$X = X^0 \oplus (X^0)^\perp.$$

A graphical representation of these spaces is provided in Fig. 15.1. Since, in general, $\ker(B)$ is not empty, we cannot state that B is an isomorphism between X and M' . Thus we aim at providing a condition which guarantees that B is an isomorphism between $(X^0)^\perp$ and M' (and, similarly, that B^T is an isomorphism between M and X_{polar}^0).

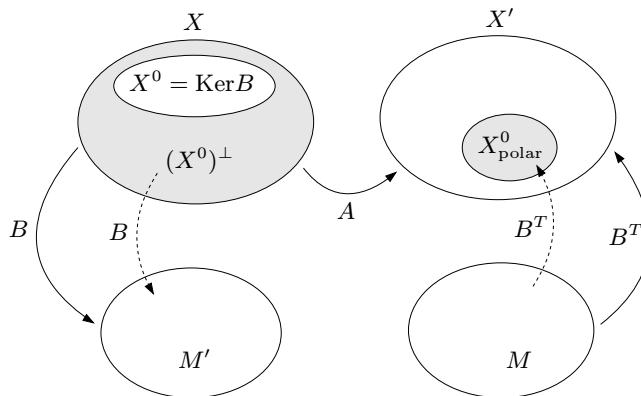


Fig. 15.1. The spaces X and M , the dual spaces X' and M' , the subspaces X^0 , $(X^0)^\perp$ and X_{polar}^0 , and the operators A , B and B^T . Dashed lines indicate isomorphisms

Lemma 15.1 *The three following statements are equivalent:*

a. *there exists a constant $\beta^* > 0$ s.t. the following compatibility condition holds*

$$\forall \mu \in M \quad \exists v \in X, \text{ with } v \neq 0 : b(v, \mu) \geq \beta^* \|v\|_X \|\mu\|_M; \quad (15.27)$$

b. *the operator B^T is an isomorphism from M onto X_{polar}^0 ; moreover,*

$$\|B^T \mu\|_{X'} = \sup_{v \in V, v \neq 0} \frac{\langle B^T \mu, v \rangle}{\|v\|_X} \geq \beta^* \|\mu\|_M \quad \forall \mu \in M; \quad (15.28)$$

c. *the operator B is an isomorphism from $(X^0)^\perp$ onto M' ; moreover,*

$$\|Bv\|_{M'} = \sup_{\mu \in M, \mu \neq 0} \frac{\langle Bv, \mu \rangle}{\|\mu\|_M} \geq \beta^* \|v\|_X \quad \forall v \in (X^0)^\perp. \quad (15.29)$$

Proof. First of all we prove the equivalence between a. and b.. Owing to the definition (15.23) of B^T , the two inequalities (15.27) and (15.28) do coincide. Let us now prove that B^T is an isomorphism between M and X_{polar}^0 . From (15.28) it follows that B^T is an injective operator from M into its range $\mathcal{R}(B^T)$, with continuous inverse. Then $\mathcal{R}(B^T)$ is a closed subspace of X' . It remains to prove that $\mathcal{R}(B^T) = X_{polar}^0$. From the *closed range theorem* (see, e.g., [Yos74]), we have

$$\mathcal{R}(B^T) = (\ker(B))_{polar},$$

whence $\mathcal{R}(B^T) = X_{polar}^0$, which is the desired result.

Let us now prove the equivalence between b. and c.. The space X_{polar}^0 can be identified with the dual space of $(X^0)^\perp$. As a matter of fact, to every $g \in ((X^0)^\perp)'$, we can associate a functional $\hat{g} \in X'$ which satisfies the relation

$$\langle \hat{g}, v \rangle = \langle g, P^\perp v \rangle \quad \forall v \in X,$$

where P^\perp denotes the orthogonal projection of X onto $(X^0)^\perp$, that is

$$\forall v \in X, \quad P^\perp v \in (X^0)^\perp : (P^\perp v - v, w)_X = 0 \quad \forall w \in (X^0)^\perp.$$

Clearly, $\hat{g} \in X_{polar}^0$ and it can be verified that $g \rightarrow \hat{g}$ is an isometric bijection between $((X^0)^\perp)'$ and X_{polar}^0 . Consequently, B^T is an isomorphism from M onto $((X^0)^\perp)'$ satisfying the relation

$$\|(B^T)^{-1}\|_{\mathcal{L}(X_{polar}^0, M)} \leq \frac{1}{\beta^*}$$

if and only if B is an isomorphism from $(X^0)^\perp$ onto M' satisfying the relation

$$\|B^{-1}\|_{\mathcal{L}(M', (X^0)^\perp)} \leq \frac{1}{\beta^*}.$$

This completes our proof. \diamond

At this point we can prove that problem (15.22) is well posed.

Theorem 15.4 *Let the bilinear form $a(\cdot, \cdot)$ satisfy the continuity condition (15.21) and be coercive on the space X^0 , that is*

$$\exists \alpha > 0 : a(v, v) \geq \alpha \|v\|_X^2 \quad \forall v \in X^0. \quad (15.30)$$

Suppose moreover that the bilinear form $b(\cdot, \cdot)$ satisfy the continuity condition (15.21) as well as the compatibility condition (15.27).

Then for every $l \in X'$ and $\sigma \in M'$, there exists a unique solution u of problem (15.26); furthermore, there exists a unique function $\eta \in M$ s.t. (u, η) is the unique solution to the original saddle-point problem (15.22).

Moreover, the map $(l, \sigma) \rightarrow (u, \eta)$ is an isomorphism from $X' \times M'$ onto $X \times M$ and the following a priori estimates hold:

$$\|u\|_X \leq \frac{1}{\alpha} \left[\|l\|_{X'} + \frac{\alpha + \gamma}{\beta^*} \|\sigma\|_{M'} \right], \quad (15.31)$$

$$\|\eta\|_M \leq \frac{1}{\beta^*} \left[\left(1 + \frac{\gamma}{\alpha} \right) \|l\|_{X'} + \frac{\gamma(\alpha + \gamma)}{\alpha \beta^*} \|\sigma\|_{M'} \right]. \quad (15.32)$$

The constants α , β^ and γ are defined in (15.30), (15.27) and (15.21), respectively. The symbols $\|\cdot\|_{X'}$ and $\|\cdot\|_{M'}$ indicate the norms of the dual spaces, and are defined as in (15.28) and (15.29), respectively.*

Proof. Uniqueness of the solution to (15.26) directly follows from the coercivity property (15.30). Let us now prove existence. From assumption (15.27) and the equivalence result stated at point c. of Lemma 15.1, we can infer that there exists a unique function $u^\sigma \in (X^0)^\perp$ s.t. $Bu^\sigma = \sigma$, and, moreover,

$$\|u^\sigma\|_X \leq \frac{1}{\beta^*} \|\sigma\|_{M'}. \quad (15.33)$$

The saddle-point problem (15.26) can be restated as follows

$$\text{find } \tilde{u} \in X^0 \quad \text{s.t.} \quad a(\tilde{u}, v) = \langle l, v \rangle - a(u^\sigma, v) \quad \forall v \in X^0. \quad (15.34)$$

The solution u to problem (15.26) is identified by the relation $u = \tilde{u} + u^\sigma$. At this point, existence and uniqueness of the solution \tilde{u} of problem (15.34) follows by the Lax-Milgram Lemma, together with the a priori estimate

$$\|\tilde{u}\|_X \leq \frac{1}{\alpha} \left(\|l\|_{X'} + \gamma \|u^\sigma\|_X \right),$$

that is, thanks to (15.33),

$$\|\tilde{u}\|_X \leq \frac{1}{\alpha} \left(\|l\|_{X'} + \frac{\gamma}{\beta^*} \|\sigma\|_{M'} \right). \quad (15.35)$$

Uniqueness of the u component of the solution to problem (15.22) is therefore a direct consequence of the uniqueness of $\tilde{u} \in X^0$ and $u^\sigma \in (X^0)^\perp$, while the stability estimate (15.31) follows again from the combination of (15.35) with (15.33).

We focus now on the η component of the solution. Since (15.34) can be restated as

$$\langle Au - l, v \rangle = 0 \quad \forall v \in X^0,$$

it follows that $(Au - l) \in X_{polar}^0$, thus we can exploit point b. of Lemma 15.1 and conclude that there exists a unique $\eta \in M$ such that $Au - l = -B^T \eta$, that is (u, η) is a solution of problem (15.22) and η satisfies the inequality

$$\|\eta\|_M \leq \frac{1}{\beta^*} \|Au - l\|_{X'}. \quad (15.36)$$

We have already noticed that every solution (u, η) to (15.22) yields a solution u to the reduced problem (15.26), whence the uniqueness of the solution of (15.22). Finally, the a priori estimate (15.32) follows from (15.36), by noting that

$$\|\eta\|_M \leq \frac{1}{\beta^*} [\|A\|_{\mathcal{L}(X, X')} \|u\|_X + \|l\|_{X'}],$$

and using the already proven a priori estimate (15.31) on u . \diamond

15.3.3 Galerkin approximation, stability and convergence analysis

To introduce a Galerkin approximation of the abstract saddle-point problem (15.22), we consider two families of finite dimensional subspaces X_h and M_h of the spaces X and M , respectively. They can be either finite element piecewise polynomial spaces, or global polynomial (spectral) spaces, or spectral element subspaces.

We look for the solution to the following problem:

given $l \in X'$ and $\sigma \in M'$, find $(u_h, \eta_h) \in X_h \times M_h$ s.t.

$$\begin{cases} a(u_h, v_h) + b(v_h, \eta_h) = \langle l, v_h \rangle & \forall v_h \in X_h, \\ b(u_h, \mu_h) = \langle \sigma, \mu_h \rangle & \forall \mu_h \in M_h. \end{cases} \quad (15.37)$$

By following what we did on the continuous problem, we can introduce the subspace

$$X_h^\sigma = \{v_h \in X_h : b(v_h, \mu_h) = \langle \sigma, \mu_h \rangle \forall \mu_h \in M_h\} \quad (15.38)$$

which allows us to introduce the following finite dimensional counterpart of the reduced formulation (15.26)

$$\text{find } u_h \in X_h^\sigma \quad \text{s.t. } a(u_h, v_h) = \langle l, v_h \rangle \quad \forall v_h \in X_h^0. \quad (15.39)$$

Since, in general, M_h is different than M , the space (15.38) is not necessarily a subspace of X^σ .

Clearly, every solution (u_h, η_h) of (15.37) yields a solution u_h for the reduced problem (15.39). In this section we look for conditions that allow us to prove that the converse statement is also true, together with a result of stability and convergence for the solution of problem (15.37).

We start by proving the discrete counterpart of Theorem 15.4.

Theorem 15.5 (Existence, uniqueness and stability) *Suppose that the bilinear form $a(\cdot, \cdot)$ satisfy the continuity property (15.21) and that it be coercive on the space X_h^0 , that is there exists a constant $\alpha_h > 0$ s.t.*

$$a(v_h, v_h) \geq \alpha_h \|v_h\|_X^2 \quad \forall v_h \in X_h^0. \quad (15.40)$$

Moreover, suppose that the bilinear form $b(\cdot, \cdot)$ satisfy the continuity condition (15.21) and that the following discrete compatibility condition holds: there exists a constant $\beta_h > 0$ s.t.

$$\forall \mu_h \in M_h \quad \exists v_h \in X_h, \quad v_h \neq 0 : b(v_h, \mu_h) \geq \beta_h \|v_h\|_X \|\mu_h\|_M. \quad (15.41)$$

Then, for every $l \in X'$ and $\sigma \in M'$, there exists a unique solution (u_h, η_h) of problem (15.37) which satisfies the following stability conditions:

$$\|u_h\|_X \leq \frac{1}{\alpha_h} \left[\|l\|_{X'} + \frac{\alpha_h + \gamma}{\beta_h} \|\sigma\|_{M'} \right], \quad (15.42)$$

$$\|\eta_h\|_M \leq \frac{1}{\beta_h} \left[\left(1 + \frac{\gamma}{\alpha_h} \right) \|l\|_{X'} + \frac{\gamma(\alpha_h + \gamma)}{\alpha_h \beta_h} \|\sigma\|_{M'} \right]. \quad (15.43)$$

Proof. The proof can be obtained by repeating that of Theorem 15.4, considering however X_h instead of X , M_h instead of M , and simply noting that

$$\|l\|_{X'_h} \leq \|l\|_{X'}, \quad \|\sigma\|_{M'_h} \leq \|\sigma\|_{M'}. \quad \diamond$$

The coercivity condition (15.30) does not necessarily guarantees (15.40) as $X_h^0 \not\subset X^0$, neither the compatibility condition (15.27) in general implies the discrete compatibility condition (15.41), due to the fact that X_h is a proper subspace of X . Moreover, in the case in which the constants α_h and β_h in (15.40) and (15.41) are independent of h , the inequalities (15.42) and (15.43) provide the desired stability result.

Condition (15.41) represents the well known *inf-sup* or *LBB* condition (see [BF91a]). (The condition (15.19) (or (15.20)) is just a special case.)

We move now to the convergence result.

Theorem 15.6 (Convergence) *Let the assumptions of Theorems 15.4 and 15.5 be satisfied. Then the solutions (u, η) and (u_h, η_h) of problems (15.22) and (15.37), respectively, satisfy the following error estimates:*

$$\|u - u_h\|_X \leq \left(1 + \frac{\gamma}{\alpha_h}\right) \inf_{v_h^* \in X_h^\sigma} \|u - v_h^*\|_X + \frac{\delta}{\alpha_h} \inf_{\mu_h \in M_h} \|\eta - \mu_h\|_M, \quad (15.44)$$

$$\begin{aligned} \|\eta - \eta_h\|_M &\leq \frac{\gamma}{\beta_h} \left(1 + \frac{\gamma}{\alpha_h}\right) \inf_{v_h^* \in X_h^\sigma} \|u - v_h^*\|_X \\ &+ \left(1 + \frac{\delta}{\beta_h} + \frac{\gamma\delta}{\alpha_h\beta_h}\right) \inf_{\mu_h \in M_h} \|\eta - \mu_h\|_M, \end{aligned} \quad (15.45)$$

where γ , δ , α_h and β_h are respectively defined by the relations (15.21), (15.40) and (15.41). Moreover, the following error estimate holds

$$\inf_{v_h^* \in X_h^\sigma} \|u - v_h^*\|_X \leq \left(1 + \frac{\delta}{\beta_h}\right) \inf_{v_h \in X_h} \|u - v_h\|_X. \quad (15.46)$$

Proof. Consider $v_h \in X_h$, $v_h^* \in X_h^\sigma$ and $\mu_h \in M_h$. By subtracting (15.37)₁ from (15.22)₁, then adding and subtracting the quantities $a(v_h^*, v_h)$ and $b(v_h, \mu_h)$, we find

$$a(u_h - v_h^*, v_h) + b(v_h, \eta_h - \mu_h) = a(u - v_h^*, v_h) + b(v_h, \eta - \mu_h).$$

Let us now choose $v_h = u_h - v_h^* \in X_h^0$. From the definition of the space X_h^0 and using (15.40) and (15.21), we find the bound

$$\|u_h - v_h^*\|_X \leq \frac{1}{\alpha_h} \left(\gamma \|u - v_h^*\|_X + \delta \|\eta - \mu_h\|_M \right)$$

from which the estimate (15.44) immediately follows, as

$$\|u - u_h\|_X \leq \|u - v_h^*\|_X + \|u_h - v_h^*\|_X.$$

Let us prove now the estimate (15.45). Owing to the compatibility condition (15.41), for every $\mu_h \in M_h$ we can write

$$\|\eta_h - \mu_h\|_M \leq \frac{1}{\beta_h} \sup_{v_h \in X_h, v_h \neq 0} \frac{b(v_h, \eta_h - \mu_h)}{\|v_h\|_X}. \quad (15.47)$$

On the other hand, by subtracting side by side (15.37)₁ from (15.22)₁, then adding and subtracting the quantity $b(v_h, \mu_h)$, we obtain

$$b(v_h, \eta_h - \mu_h) = a(u - u_h, v_h) + b(v_h, \eta - \mu_h).$$

Using this identity in (15.47) as well as the continuity inequalities (15.21), it follows that

$$\|\eta_h - \mu_h\|_M \leq \frac{1}{\beta_h} \left(\gamma \|u - u_h\|_X + \delta \|\eta - \mu_h\|_M \right).$$

This yields the desired result, provided we use the error estimate (15.44) that was previously derived for the u variable.

Finally, let us prove (15.46). Property (15.41) allows us to use the discrete version of Lemma 15.1 (now applied in the finite dimensional subspaces). Then, owing to the discrete counterpart of (15.29), for every $v_h \in X_h$ we can find a unique function $z_h \in (X_h^0)^\perp$ such that

$$b(z_h, \mu_h) = b(u - v_h, \mu_h) \quad \forall \mu_h \in M_h,$$

and, moreover,

$$\|z_h\|_X \leq \frac{\delta}{\beta_h} \|u - v_h\|_X.$$

The function $v_h^* = z_h + v_h$ belongs to X_h^σ , as $b(u, \mu_h) = \langle \sigma, \mu_h \rangle$ for all $\mu_h \in M_h$. Moreover,

$$\|u - v_h^*\|_X \leq \|u - v_h\|_X + \|z_h\|_X \leq \left(1 + \frac{\delta}{\beta_h}\right) \|u - v_h\|_X,$$

whence the estimate (15.46) follows. \diamond

The inequalities (15.44) and (15.45) yield error estimates with optimal convergence rate, provided that the constants α_h and β_h in (15.40) and (15.41) be bounded from below by two constants α and β independent of h . Let us also remark that the inequality (15.44) holds even if the compatibility conditions (15.27) and (15.41) are not satisfied.

Remark 15.2 (Spurious pressure modes) The compatibility condition (15.41) is essential to guarantee uniqueness of the η_h component of the solution. Indeed, if (15.41) does not hold, then one could find functions $\mu_h^* \in M_h$, $\mu_h^* \neq 0$, such that

$$b(v_h, \mu_h^*) = 0 \quad \forall v_h \in X_h.$$

Consequently, should (u_h, η_h) be a solution to problem (15.37), then $(u_h, \eta_h + \tau \mu_h^*)$, for all $\tau \in \mathbb{R}$, would be a solution too.

Any such function μ_h^* is called *spurious mode*, or, more specifically, *pressure* spurious mode when it does refer to the Stokes problem (15.18) in which functions μ_h represent discrete pressures. Numerical instabilities can arise since the discrete problem (15.37) is unable to detect such spurious modes. \bullet

For a given couple of finite dimensional spaces X_h and M_h , proving that the discrete compatibility condition (15.41) holds with a constant β_h independent of h is not always easy. Several practical criteria are available, among which we mention those due to Fortin ([For77]), Boland and Nicolaides ([BN83]), and Verfürth ([Ver84]). (See [BF91b].)

15.4 Algebraic formulation of the Stokes problem

Let us investigate the structure of the algebraic system associated to the Galerkin approximation (15.18) to the Stokes problem (or, more generally, to a discrete saddle-point problem like (15.37)). Denote with

$$\{\varphi_j \in V_h\}, \quad \{\phi_k \in Q_h\},$$

the basis functions of the spaces V_h and Q_h , respectively. Let us expand the discrete solutions \mathbf{u}_h and p_h with respect to such bases,

$$\mathbf{u}_h(\mathbf{x}) = \sum_{j=1}^N u_j \varphi_j(\mathbf{x}), \quad p_h(\mathbf{x}) = \sum_{k=1}^M p_k \phi_k(\mathbf{x}), \quad (15.48)$$

having set $N = \dim V_h$ and $M = \dim Q_h$. By choosing as test functions in (15.18) the same basis functions we obtain the following blockwise linear system

$$\begin{cases} \mathbf{A}\mathbf{U} + \mathbf{B}^T \mathbf{P} = \mathbf{F}, \\ \mathbf{B}\mathbf{U} = \mathbf{0}, \end{cases} \quad (15.49)$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ and $\mathbf{B} \in \mathbb{R}^{M \times N}$ are the matrices related respectively to the bilinear forms $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$, whose elements are given by

$$\mathbf{A} = [a_{ij}] = [a(\varphi_j, \varphi_i)], \quad \mathbf{B} = [b_{km}] = [b(\varphi_m, \phi_k)],$$

while \mathbf{U} and \mathbf{P} are the vectors of the unknowns,

$$\mathbf{U} = [u_j], \quad \mathbf{P} = [p_j].$$

The $(N + M) \times (N + M)$ matrix

$$\mathbf{S} = \begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \quad (15.50)$$

is *block symmetric* (as \mathbf{A} is symmetric) and *indefinite*, featuring real eigenvalues with variable sign (either positive and negative). \mathbf{S} is non-singular iff no eigenvalue is null, a property that follows from the *inf-sup* condition (15.20). To prove the latter statement we proceed as follows.

Since \mathbf{A} is non-singular – being associated to the coercive bilinear form $a(\cdot, \cdot)$ – from the first of (15.49) we can formally obtain \mathbf{U} as

$$\mathbf{U} = \mathbf{A}^{-1}(\mathbf{F} - \mathbf{B}^T \mathbf{P}). \quad (15.51)$$

Using (15.51) in the second equation of (15.49) yields

$$\mathbf{R}\mathbf{P} = \mathbf{B}\mathbf{A}^{-1}\mathbf{F}, \quad \text{where} \quad \mathbf{R} = \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T. \quad (15.52)$$

This corresponds to having carried out a block gaussian elimination on system (15.50).

This way we obtain a reduced system for the sole unknown \mathbf{P} (the pressure), which admits a unique solution in case matrix R is non-singular. Since A is non-singular and positive definite, the latter condition is satisfied iff matrix B^T has a null kernel, that is

$$\ker B^T = \{\mathbf{0}\}. \quad (15.53)$$

On the other hand, since A is non-singular, from the existence and uniqueness of \mathbf{P} we entail that there exists a unique vector \mathbf{U} which satisfies (15.51).

In conclusion, system (15.49) admits a unique solution (\mathbf{U}, \mathbf{P}) if and only if condition (15.53) holds. On the other hand, the latter algebraic condition is in fact equivalent to the *inf-sup* condition (15.20) (see Exercise 1).

Let us consider again the Remark 15.2 concerning the general saddle-point problem and suppose that the *inf-sup* condition (15.20) does not hold. In this case we can state that

$$\exists q_h^* \in Q_h : \quad b(\mathbf{v}_h, q_h^*) = 0 \quad \forall \mathbf{v}_h \in V_h. \quad (15.54)$$

Consequently, if (\mathbf{u}_h, p_h) is a solution to the Stokes problem (15.18), then $(\mathbf{u}_h, p_h + q_h^*)$ is a solution too, as

$$\begin{aligned} a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h + q_h^*) &= a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) + b(\mathbf{v}_h, q_h^*) \\ &= a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = (\mathbf{f}, \mathbf{v}_h) \quad \forall \mathbf{v}_h \in V_h. \end{aligned}$$

Functions q_h^* which fail to satisfy the *inf-sup* condition are transparent to the Galerkin problem(15.18). For this reason, as already observed, they are called spurious pressure modes, or even *parasitic modes*. Their presence inhibits the pressure solution to be unique, yielding numerical instabilities. For this reason, those finite dimensional subspaces that violate the compatibility condition (15.20) are said to be *unstable*, or *incompatible*.

Two strategies are generally adopted:

- choose spaces V_h and Q_h that satisfy the *inf-sup* condition;
- stabilize (either a priori or a posteriori) the finite dimensional problem by eliminating the spurious modes.

Let us analyze the first type of strategy. To start with, we will consider the case of finite element spaces. To characterize Q_h and V_h it suffices to choose on every element of the triangulation the degrees of freedom for velocity and pressure. The weak formulation does not require a continuous pressure. We will therefore start considering the case of *discontinuous pressures*.

As Stokes equations are first order w.r.t. p and second order w.r.t. \mathbf{u} , generally speaking it makes sense to use piecewise polynomials of degree $k \geq 1$ for the velocity space V_h and of degree $k - 1$ for the space Q_h .

In particular, we might want to use piecewise linear \mathbb{P}_1 finite elements for each velocity component, and piecewise constant \mathbb{P}_0 finite elements for the pressure (see

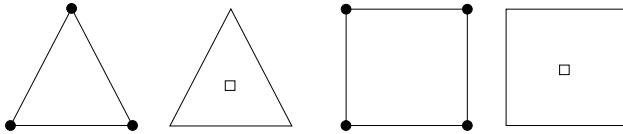


Fig. 15.2. Case of discontinuous pressure: choices that do not satisfy the *inf-sup* condition, on triangles (left), and on quadrilaterals (right)

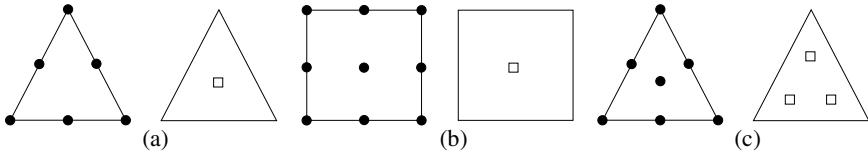


Fig. 15.3. Case of discontinuous pressure: choices that do satisfy the *inf-sup* condition: on triangles, (a), and on quadrilaterals, (b). Also the couple (c), known as Crouzeix-Raviart elements, satisfies the *inf-sup* condition

Fig. 15.2 in which, as in all those that will follow, by means of the symbol \square we indicate the degrees of freedom for the pressure, whereas the symbol \bullet identifies those for each velocity component). In fact, although being quite natural, this choice does not pass the *inf-sup* condition (15.20) (see Exercise 3).

When looking for a compatible couple of spaces, we note that the larger the velocity space V_h , the higher the probability that the *inf-sup* condition be satisfied. Otherwise said, the space V_h should be “rich” enough w.r.t. the space Q_h . In Fig. 15.3 we report three different choices of spaces that fulfill the *inf-sup* condition, still in the case of continuous velocity and discontinuous pressure. Choice (a) is made by $\mathbb{P}_2 - \mathbb{P}_0$ elements, (b) by $\mathbb{Q}_2 - \mathbb{P}_0$ elements, while choice (c) by piecewise linear discontinuous elements for the pressure, while the velocity components are made by piecewise quadratic continuous elements enriched by a cubic bubble function on each triangle – these are the so-called Crouzeix-Raviart elements.

In Fig. 15.4(a), (b) we report two choices of incompatible finite elements in the case of continuous pressure. They consist of piecewise linear elements on triangles (resp. bilinear on quadrilaterals) for both velocity and pressure. More in general, finite elements of the same polynomial degree $k \geq 1$ for both velocity and pressures are unstable. On the same figure, the elements displayed in (c) and (d) are instead stable. In both cases, pressure is a piecewise linear continuous function, whereas velocities are piecewise linear polynomials on everyone of the four sub-triangles (case (c)), or piecewise linear polynomials enriched by a cubic bubble function (case (d)). The pair $\mathbb{P}_2 - \mathbb{P}_1$ (continuous piecewise quadratic velocities and continuous piecewise linear pressure) is stable. This is the lowest degree representative of the family of the so-called Taylor-Hood elements $\mathbb{P}_k - \mathbb{P}_{k-1}$, $k \geq 2$ (continuous velocities and continuous pressure), that are *inf-sup* stable. For the proof of the stability results here mentioned, as well for the convergence analysis, the reader can refer to [BF91a].

Should spectral methods be used, using equal order polynomial spaces for both velocity and pressure yields subspaces that violate the *inf-sup* condition. Compatible

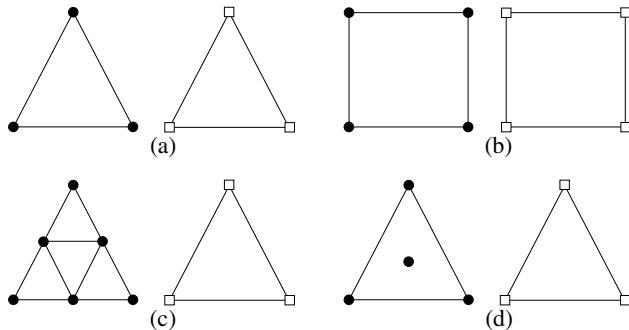


Fig. 15.4. Case of continuous pressure: the couples (a) and (b) do not satisfy the *inf-sup* condition. The elements used for the velocity components in (c) are known as \mathbb{P}_1 -*iso* \mathbb{P}_2 finite elements, whereas couple (d) is called *mini-element*

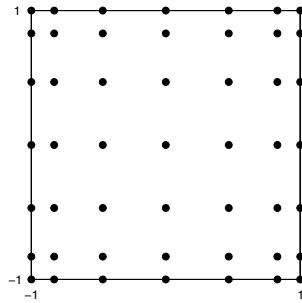


Fig. 15.5. The $(N + 1)^2$ Gauss-Legendre-Lobatto (GLL) nodes (here $N = 6$), hosting the degrees of freedom of the velocity components

spectral spaces can instead be obtained by using, e.g., polynomials of degree $N (\geq 2)$ for each velocity component, and degree $N - 2$ for the pressure, yielding the so-called $\mathbb{Q}_N - \mathbb{Q}_{N-2}$ approximation. The degrees of freedom for each velocity component are represented by the $(N + 1)^2$ GLL nodes (see Fig. 15.5).

For the pressure, at least two sets of interpolation nodes can be used: either the subset represented by the $(N - 1)^2$ internal nodes of the set of $(N + 1)^2$ Gauss-Legendre nodes (Fig. 15.6, left), or the $(N - 1)^2$ Gauss nodes (Fig. 15.6, right). This choice stands at the base of a spectral-type approximation, such as collocation, G-NI (Galerkin with numerical integration), or SEM-NI (spectral element with numerical integration) (see [CHQZ07]).

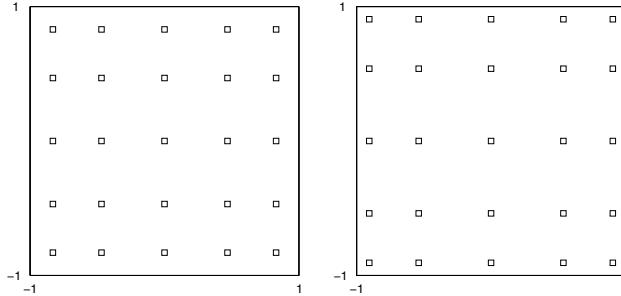


Fig. 15.6. The $(N-1)^2$ internal Gauss-Legendre-Lobatto (GLL) nodes (left) and the $(N-1)^2$ Gauss-Legendre (GL) nodes (right) (here for $N = 6$), hosting the degrees of freedom of the pressure

15.5 An example of stabilized problem

We have seen that finite element or spectral methods that make use of equal degree polynomials for both velocity and pressure do not fulfill the *inf-sup* condition and are therefore “unstable”. However, stabilizing them is possible by SUPG or GLS techniques like those encountered in Chap. 11 in the framework of the numerical approximation of advection-diffusion equations.

For a general discussion on stabilization techniques for Stokes equations, the reader can refer e.g. to [BF91a]. Here we limit ourselves to show how the GLS stabilization can be applied on problem(15.18) in case piecewise continuous linear finite elements are used for velocity components as well as for the pressure

$$V_h = [\overset{\circ}{X}_h^1]^2, \quad Q_h = \{q_h \in X_h^1 : \int_{\Omega} q_h \, d\Omega = 0\}.$$

This choice is urged by the need of keeping the global number of degrees of freedom as low as possible, especially when dealing with three-dimensional problems. We set therefore $W_h = V_h \times Q_h$ and, instead of (15.18), the following problem is considered (we restrict ourselves to the case where $\alpha = 0$):

$$\text{find } (\mathbf{u}_h, p_h) \in W_h : A_h(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h) = (\mathbf{f}_h, \mathbf{v}_h) \quad \forall (\mathbf{v}_h, q_h) \in W_h. \quad (15.55)$$

We have set

$$A_h : W_h \times W_h \rightarrow \mathbb{R},$$

$$A_h(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h) = a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) - b(\mathbf{u}_h, q_h)$$

$$+ \delta \sum_{K \in \mathcal{T}_h} h_K^2 \int_K (-\nu \Delta \mathbf{u}_h + \nabla p_h - \mathbf{f})(-\nu \Delta \mathbf{v}_h + \nabla q_h) \, dK,$$

and we have denoted with δ a positive parameter that must be conveniently chosen. This is a strongly consistent approximation of problem (15.11): as a matter of fact, the

additional term, which depends on the residual of the discrete momentum equation, is null when calculated on the exact solution as, thanks to (15.12), $-\nu \Delta \mathbf{u} + \nabla p - \mathbf{f} = \mathbf{0}$. (Note that, in this specific case, $\Delta \mathbf{u}_{h|K} = \Delta \mathbf{v}_{h|K} = \mathbf{0} \forall K \in \mathcal{T}_h$ as we are using piecewise linear finite element functions.).

From the identity

$$A_h(\mathbf{u}_h, p_h; \mathbf{u}_h, p_h) = \nu \|\nabla \mathbf{u}_h\|_{\mathbf{L}^2(\Omega)}^2 + \delta \sum_{K \in \mathcal{T}_h} h_K^2 \|\nabla p_h\|_{\mathbf{L}^2(K)}^2, \quad (15.56)$$

we deduce that the kernel of the bilinear form A_h reduces to the sole null vector, whence problem (15.55) admits one and only one solution. The latter satisfies the stability inequality

$$\nu \|\nabla \mathbf{u}_h\|_{\mathbf{L}^2(\Omega)}^2 + \delta \sum_{K \in \mathcal{T}_h} h_K^2 \|\nabla p_h\|_{\mathbf{L}^2(K)}^2 \leq C \|\mathbf{f}\|_{\mathbf{L}^2(\Omega)}^2, \quad (15.57)$$

C being a constant that depends on ν but not on h (see Exercise 7).

By applying Strang lemma 10.1 we can now show that the solution to the generalized Galerkin problem (15.55) satisfies the following error estimate

$$\|\mathbf{u} - \mathbf{u}_h\|_{\mathbf{H}^1(\Omega)} + \left(\delta \sum_{K \in \mathcal{T}_h} h_K^2 \|\nabla p - \nabla p_h\|_{\mathbf{L}^2(K)}^2 \right)^{1/2} \leq Ch.$$

Still using the notations of Sec. 15.2, we can show that (15.55) admits the following matrix form

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{G} \end{bmatrix}. \quad (15.58)$$

This system differs from the one (15.49) without stabilization because of the presence of the non-null block occupying the position (2,2), which is associated to the stabilization term. More precisely,

$$C = (c_{km}) \ , \ c_{km} = \delta \sum_{K \in \mathcal{T}_h} h_K^2 \int_K \nabla \phi_m \cdot \nabla \phi_k \, dK, \quad k, m = 1, \dots, M,$$

while the components of the right hand side \mathbf{G} are

$$g_k = -\delta \sum_{K \in \mathcal{T}_h} h_K^2 \int_K \mathbf{f} \cdot \nabla \phi_k \, dK, \quad k = 1, \dots, M.$$

In this case, the system reduced to the pressure unknown reads

$$\mathbf{R}\mathbf{P} = \mathbf{B}\mathbf{A}^{-1}\mathbf{F} - \mathbf{G}.$$

Differently than (15.52), this time it is $\mathbf{R} = \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{C}$. Matrix \mathbf{R} is non-singular as \mathbf{C} is a positive definite matrix.

15.6 A numerical example

We want to solve the stationary Navier-Stokes equations in the square domain $\Omega = (0, 1) \times (0, 1)$ with the following Dirichlet conditions

$$\begin{aligned}\mathbf{u} &= \mathbf{0}, & \mathbf{x} \in \partial\Omega \setminus \Gamma, \\ \mathbf{u} &= (1, 0)^T, & \mathbf{x} \in \Gamma,\end{aligned}\tag{15.59}$$

where $\Gamma = \{\mathbf{x} = (x_1, x_2)^T \in \partial\Omega : x_2 = 1\}$. This problem is known as flow in a lid driven cavity. We will use continuous piecewise bilinear $\mathbb{Q}_1 - \mathbb{Q}_1$ polynomials on rectangular finite elements. As we know, these spaces do not fulfill the compatibility condition approximation; in Fig. 15.7, left, we display the spurious pressure modes that are generated by this Galerkin approximation. On the same figure, at the right, we have drawn the pressure isolines obtained using a GLS stabilization (that was addressed in the previous section) on the same kind of finite elements. The pressure is now free of numerical oscillations. Still for the stabilized problem, in Fig. 15.8 we display the streamlines for two different values of the Reynolds number, $Re = 1000$ and $Re = 5000$. The stabilization term cures simultaneously the pressure instabilities (by getting rid of the spurious modes) and potential instabilities of the pure Galerkin method that develop when diffusion is dominated by convection, an issue that we have extensively addressed in Chap. 11.

For exactly the same problem, we consider as well a spectral G-NI approximation in which the pressure as well as each velocity component are polynomials of \mathbb{Q}_N (with $N = 32$). As previously observed, this choice of spaces does not fulfill the *inf-sup* condition, thus it generates pressure spurious modes that are clearly visible in Fig. 15.9, left. A GLS stabilization, similar to that previously used for finite elements, can be set up for the G-NI method too. The corresponding solution is now stable and free

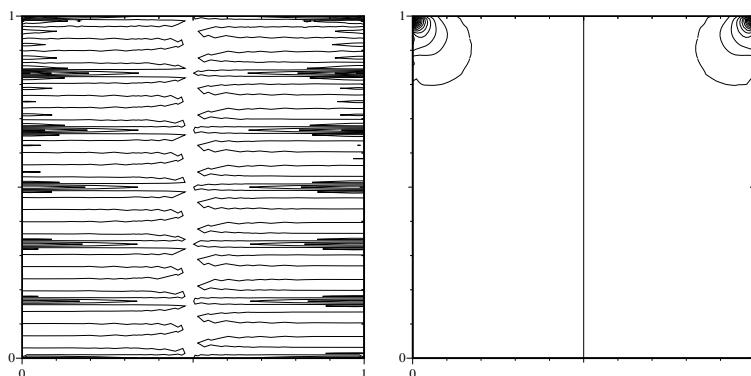


Fig. 15.7. Pressure isolines for the numerical approximation of the lid driven cavity problem. Stabilized GLS approximation (on the right); the vertical line corresponds to the null value of the pressure. Non-stabilized approximation (on the left); the presence of a spurious numerical pressure is evident

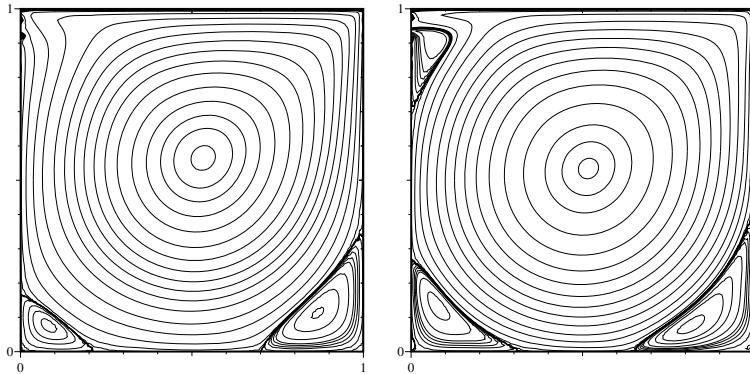


Fig. 15.8. Streamlines of the numerical solution of the lid driven cavity problem corresponding to two different values of the Reynolds number: $Re = 1000$, left, and $Re = 5000$, right

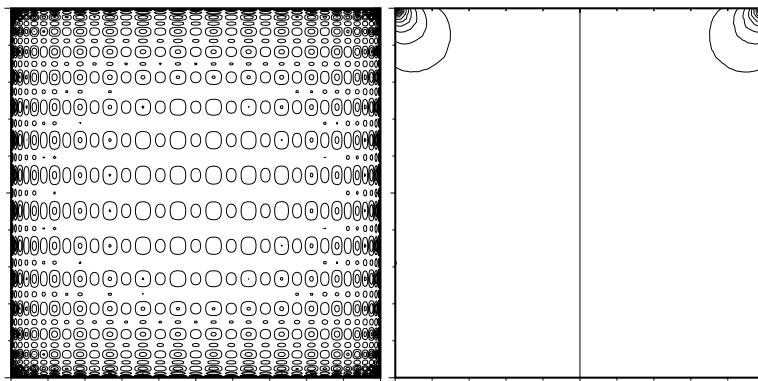


Fig. 15.9. Pressure isolines obtained by the pure spectral G-NI method (on the left), and by the GLS stabilized spectral G-NI method (on the right). In either case, polynomials of the same degree, $N = 32$, are used for both pressure and velocity. As expected, the pure G-NI method yields spurious pressure solutions. The test case is still the same lid driven cavity problem previously approximated by bilinear finite elements

of spurious pressure modes, as it can be evicted from the pressure isolines displayed on the right hand of the same figure.

15.7 Time discretization of Navier-Stokes equations

Let us now revert to the Navier-Stokes equations (15.2) and focus on the issue of time discretization. To avoid unnecessary cumbersome notation, from now on we will assume that $\Gamma_D = \partial\Omega$ and $\varphi = \mathbf{0}$ in (15.3), whence the velocity space becomes $V = [H_0^1(\Omega)]^d$. The space discretization of Navier-Stokes equations yields the following problem:

for every $t > 0$, find $(\mathbf{u}_h(t), p_h(t)) \in V_h \times Q_h$ s.t.

$$\left\{ \begin{array}{l} \left(\frac{\partial \mathbf{u}_h(t)}{\partial t}, \mathbf{v}_h \right) + a(\mathbf{u}_h(t), \mathbf{v}_h) + c(\mathbf{u}_h(t), \mathbf{u}_h(t), \mathbf{v}_h) + b(\mathbf{v}_h, p_h(t)) \\ = (\mathbf{f}_h(t), \mathbf{v}_h) \quad \forall \mathbf{v}_h \in V_h, \\ b(\mathbf{u}_h(t), q_h) = 0 \quad \forall q_h \in Q_h, \end{array} \right. \quad (15.60)$$

where, as customarily, $\{V_h \subset V\}$ and $\{Q_h \subset Q\}$ are two families of finite dimensional subspaces of the velocity and pressure functional spaces, respectively. The trilinear form $c(\cdot, \cdot, \cdot)$, defined by

$$c(\mathbf{w}, \mathbf{z}, \mathbf{v}) = \int_{\Omega} [(\mathbf{w} \cdot \nabla) \mathbf{z}] \cdot \mathbf{v} \, d\Omega \quad \forall \mathbf{w}, \mathbf{z}, \mathbf{v} \in V,$$

is associated to the nonlinear convective term, while $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ are the same as in (15.13) (setting however $\alpha = 0$).

Problem (15.60) is in fact a system of nonlinear differential algebraic equations. By using notations already employed in the previous sections, it can be restated in compact form as follows

$$\left\{ \begin{array}{l} M \frac{d\mathbf{u}(t)}{dt} + A\mathbf{u}(t) + C(\mathbf{u}(t))\mathbf{u}(t) + B^T \mathbf{p}(t) = \mathbf{f}(t), \\ B\mathbf{u}(t) = \mathbf{0}, \end{array} \right. \quad (15.61)$$

with $\mathbf{u}(0) = \mathbf{u}_0$. $C(\mathbf{u}(t))$ is in fact a matrix depending on $\mathbf{u}(t)$, whose generic coefficient is $c_{mi}(t) = c(\mathbf{u}(t), \varphi_i, \varphi_m)$. For the temporal discretization of this system let us use, for instance, the θ -method, that was introduced in Sec. 5.1 for parabolic equations. Upon setting

$$\begin{aligned} \mathbf{u}_{\theta}^{n+1} &= \theta \mathbf{u}^{n+1} + (1 - \theta) \mathbf{u}^n, \\ \mathbf{p}_{\theta}^{n+1} &= \theta \mathbf{p}^{n+1} + (1 - \theta) \mathbf{p}^n, \\ \mathbf{f}_{\theta}^{n+1} &= \mathbf{f}(\theta t^{n+1} + (1 - \theta) t^n), \end{aligned}$$

we obtain the following system of algebraic equations

$$\left\{ \begin{array}{l} M \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + A\mathbf{u}_{\theta}^{n+1} + C(\mathbf{u}_{\theta}^{n+1})\mathbf{u}_{\theta}^{n+1} + B^T \mathbf{p}_{\theta}^{n+1} = \mathbf{f}_{\theta}^{n+1}, \\ B\mathbf{u}^{n+1} = \mathbf{0}. \end{array} \right. \quad (15.62)$$

Unless for the special case $\theta = 0$, which corresponds to the forward Euler method, the solution of this system is quite involved. A possible alternative to resort to a *semi-implicit* scheme, in which the linear part of the equation is advanced implicitly, whereas the nonlinear terms explicitly. By so doing, if $\theta \geq 1/2$, the resulting scheme

is unconditionally stable, whereas it must undergo a stability restriction on the time step Δt (depending on h and ν) in all other cases. We further elaborate on this issue on the next section. Later, in Sec. 15.7.2 and 15.7.3 we will instead address other temporal discretization schemes. For more details, results and bibliographical references, see, e.g., [QV94, Chap. 13].

15.7.1 Finite difference methods

We consider at first an explicit temporal discretization of the first equation in (15.61), corresponding to the choice $\theta = 0$ in (15.62). If we suppose that all quantities be known at the time level t^n , we can write the associated problem at time t^{n+1} as follows

$$\begin{cases} \mathbf{M}\mathbf{u}^{n+1} = H(\mathbf{u}^n, \mathbf{p}^n, \mathbf{f}^n), \\ \mathbf{B}\mathbf{u}^{n+1} = \mathbf{0}, \end{cases}$$

where \mathbf{M} is the *mass matrix* whose entries are

$$m_{ij} = \int_{\Omega} \varphi_i \varphi_j \, d\Omega.$$

This is an overdetermined system for the unknown vector \mathbf{u}^{n+1} , whereas it does not allow the determination of the pressure \mathbf{p}^{n+1} . However, should we replace \mathbf{p}^n by \mathbf{p}^{n+1} in the momentum equation, we would obtain the new linear system

$$\begin{cases} \frac{1}{\Delta t} \mathbf{M}\mathbf{u}^{n+1} + \mathbf{B}^T \mathbf{p}^{n+1} = \mathbf{G}, \\ \mathbf{B}\mathbf{u}^{n+1} = \mathbf{0}, \end{cases} \quad (15.63)$$

being \mathbf{G} a suitable known vector. This system corresponds to a *semi-explicit* discretization of (15.60). Since \mathbf{M} is symmetric and positive definite, if condition (15.53) is satisfied, then the reduced system $\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T \mathbf{p}^{n+1} = \mathbf{B}\mathbf{M}^{-1}\mathbf{G}$ is non-singular. Once solved, the velocity vector \mathbf{u}^{n+1} can be recovered from the first equation of (15.63). This discretization method is temporally stable provided the time step satisfies the following limitation

$$\Delta t \leq C \min \left(\frac{h^2}{\nu}, \frac{h}{\max_{\mathbf{x} \in \Omega} |\mathbf{u}^n(\mathbf{x})|} \right).$$

Let us now consider an *implicit* discretization of (15.60), for instance that based on the backward Euler method which corresponds to choosing $\theta = 1$ in (15.62). As already observed, this scheme is unconditionally stable. It yields a nonlinear algebraic system which can be regarded as the finite element space approximation to the steady Navier-Stokes problem

$$\begin{cases} -\nu \Delta \mathbf{u}^{n+1} + (\mathbf{u}^{n+1} \cdot \nabla) \mathbf{u}^{n+1} + \nabla p^{n+1} + \frac{\mathbf{u}^{n+1}}{\Delta t} = \tilde{\mathbf{f}}, \\ \operatorname{div} \mathbf{u}^{n+1} = 0. \end{cases}$$

The solution of such nonlinear algebraic system can be achieved by a Newton-Krylov techniques, that is by using a Krylov method (e.g. GMRES or BiCGStab) for the solution of the linear system that is obtained at each Newton iteration step (see, e.g., [Saa96] or [QV94, Chap. 2]). We recall that Newton method is based on the full linearization of the convective term, $\mathbf{u}_k^{n+1} \cdot \nabla \mathbf{u}_k^{n+1} + \mathbf{u}_{k+1}^{n+1} \cdot \nabla \mathbf{u}_k^{n+1}$. A popular approach consists of starting Newton iterations after few Picard iterations in which the convective term is evaluated as follows: $\mathbf{u}_k^{n+1} \cdot \nabla \mathbf{u}_{k+1}^{n+1}$.

This approach entails three nested cycles:

- temporal iteration: $t^n \rightarrow t^{n+1}$,
- Newton iteration: $\mathbf{x}_k^{n+1} \rightarrow \mathbf{x}_{k+1}^{n+1}$,
- Krylov iteration: $[\mathbf{x}_k^{n+1}]_j \rightarrow [\mathbf{x}_k^{n+1}]_{j+1}$,

where for simplicity we have noted with \mathbf{x}^n the couple $(\mathbf{u}^n, \mathbf{p}^n)$. Obviously, the goal is that the following convergence result holds

$$\lim_{k \rightarrow \infty} \lim_{j \rightarrow \infty} [\mathbf{x}_k^{n+1}]_j = \begin{bmatrix} \mathbf{u}^{n+1} \\ \mathbf{p}^{n+1} \end{bmatrix}.$$

Finally, let us operate a *semi-implicit*, temporal discretization, consisting of treating explicitly the nonlinear convective term. The following algebraic linear system, whose form is similar to (15.49), is obtained in this case

$$\begin{cases} \frac{1}{\Delta t} \mathbf{M} \mathbf{u}^{n+1} + \mathbf{A} \mathbf{u}^{n+1} + \mathbf{B}^T \mathbf{p}^{n+1} = \mathbf{G}, \\ \mathbf{B} \mathbf{u}^{n+1} = \mathbf{0}, \end{cases} \quad (15.64)$$

where \mathbf{G} is a suitable known vector. In this case the stability restriction on the time step takes the following form

$$\Delta t \leq C \frac{h}{\max_{\mathbf{x} \in \Omega} |\mathbf{u}^n(\mathbf{x})|}. \quad (15.65)$$

In all cases, optimal error estimates can be proven.

15.7.2 Characteristics (or Lagrangian) methods

The *material derivative* (also called Lagrangian derivative) of the velocity vector field is defined as

$$\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u}.$$

Characteristics methods are based on approximating the material derivative, e.g. by the backward Euler method

$$\frac{D\mathbf{u}}{Dt}(\mathbf{x}) \approx \frac{\mathbf{u}^{n+1}(\mathbf{x}) - \mathbf{u}^n(\mathbf{x}_p)}{\Delta t},$$

where \mathbf{x}_p is the *foot* (at time t^n) of the characteristic issuing from \mathbf{x} at time t^{n+1} . A system of ordinary differential equations has to be solved to follow backward the characteristic line \mathbf{X} issuing from the point \mathbf{x}

$$\begin{cases} \frac{d\mathbf{X}}{dt}(t; s, \mathbf{x}) = \mathbf{u}(t, \mathbf{X}(t; s, \mathbf{x})), & t \in (t^n, t^{n+1}), \\ \mathbf{X}(s; s, \mathbf{x}) = \mathbf{x}, \end{cases}$$

having set $s = t^{n+1}$.

The main difficulty relies on the retrieval of the characteristic lines. The first problem is how to suitably approximate the velocity field $\mathbf{u}(\mathbf{t})$ for $t \in (t^n, t^{n+1})$, as \mathbf{u}^{n+1} is unknown. With this aim, the simplest way consists of using a forward Euler scheme for the discretization of the material derivative. The second difficulty stems from the fact that a characteristic line may cross several elements of the computational grid. An algorithm is therefore necessary to locate the element wherein the characteristic foot falls, or to detect those cases in which the latter hits a boundary edge. With the previous discretization of the material derivative, at every time level t^{n+1} the momentum equation becomes (formally speaking)

$$\frac{\mathbf{u}^{n+1}(\mathbf{x}) - \mathbf{u}^n(\mathbf{x}_p)}{\Delta t} - \nu \Delta \mathbf{u}^{n+1}(\mathbf{x}) + \nabla p^{n+1}(\mathbf{x}) = \mathbf{f}^{n+1}(\mathbf{x}).$$

If used in the framework of piecewise linear finite elements in space, this scheme is unconditionally stable. Moreover, it satisfies the error estimate

$$\|\mathbf{u}(t^n) - \mathbf{u}^n\|_{L^2(\Omega)} \leq C(h + \Delta t + h^2/\Delta t) \quad \forall n \geq 1,$$

for a positive constant C independent of ν . Characteristic-based time discretization strategies for spectral methods are reviewed in [CHQZ07, Chap. 3].

15.7.3 Fractional step methods

Let us consider an abstract time dependent problem,

$$\frac{\partial w}{\partial t} + Lw = f,$$

where L is a differential operator that splits into the sum of two operators, L_1 and L_2 , that is

$$Lv = L_1 v + L_2 v.$$

Fractional step methods allow the temporal advancement from time t^n to t^{n+1} in two steps (or more). At first only the operator L_1 is advanced in time implicitly, then the solution so obtained is corrected by performing a second step in which only the other operator, L_2 , is accounted for. This is why these kind of methods are also named *operator splitting*.

In principle, by separating the two operators L_1 and L_2 , a complex problem is split into two simpler problems, each one with its own feature. In this respect, the operators

L_1 and L_2 can be chosen on the ground of physical considerations: diffusion can be split from transport, for instance. In fact, also the solution of Navier-Stokes equations by the characteristic method can be regarded as a fractional step method whose first step operator is expressed by the Lagrangian derivative.

A simple, albeit not optimal fractional step scheme, is the following, known as *Yanenko splitting*:

1. compute the solution \tilde{w} of the equation

$$\frac{\tilde{w} - w^n}{\Delta t} + L_1 \tilde{w} = 0;$$

2. compute the solution w^{n+1} of the equation

$$\frac{w^{n+1} - \tilde{w}}{\Delta t} + L_2 w^{n+1} = f^n.$$

By eliminating \tilde{w} , the following problem is found for w^{n+1}

$$\frac{w^{n+1} - w^n}{\Delta t} + L w^{n+1} = f^n + \Delta t L_1 (f^n - L_2 w^{n+1}).$$

In the case where both L_1 and L_2 are elliptic operators, this scheme is unconditionally stable w.r.t. Δt .

This strategy can be applied to the Navier-Stokes equations (15.2), choosing L_1 as $L_1(\mathbf{w}) = -\nu \Delta \mathbf{w} + (\mathbf{w} \cdot \nabla) \mathbf{w}$ whereas L_2 is the operator associated to the remaining terms of the Navier-Stokes problem. This way we have split the main difficulties arising when treating Navier-Stokes equations, the nonlinear part from that imposing the incompressibility constraint. The corresponding fractional step scheme reads:

1. solve the diffusion-transport equation for the $\tilde{\mathbf{u}}^{n+1}$ velocity

$$\begin{cases} \frac{\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^n}{\Delta t} - \nu \Delta \tilde{\mathbf{u}}^{n+1} + (\mathbf{u}^* \cdot \nabla) \mathbf{u}^{**} = \mathbf{f}^{n+1} & \text{in } \Omega, \\ \tilde{\mathbf{u}}^{n+1} = \mathbf{0} & \text{on } \partial\Omega; \end{cases} \quad (15.66)$$

2. solve the following coupled problem for the two unknowns \mathbf{u}^{n+1} and p^{n+1}

$$\begin{cases} \frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}}{\Delta t} + \nabla p^{n+1} = \mathbf{0} & \text{in } \Omega, \\ \operatorname{div} \mathbf{u}^{n+1} = 0 & \text{in } \Omega, \\ \mathbf{u}^{n+1} \cdot \mathbf{n} = 0 & \text{on } \partial\Omega, \end{cases} \quad (15.67)$$

where \mathbf{u}^* and \mathbf{u}^{**} can be either $\tilde{\mathbf{u}}^{n+1}$ or \mathbf{u}^n depending on the fact that the nonlinear convective terms be treated explicitly, implicitly or semi-implicitly. In such a way, in the first step an intermediate velocity $\tilde{\mathbf{u}}^{n+1}$ is calculated, then it is corrected in the second step in order to satisfy the incompressibility constraint. The diffusion-transport

problem of the first step can be successfully faced by using the approximation techniques investigated in Chap. 11.

More involved is the numerical treatment of the problem associated with the second step. By formally applying the divergence operator to the first equation, we obtain

$$\operatorname{div} \frac{\mathbf{u}^{n+1}}{\Delta t} - \operatorname{div} \frac{\tilde{\mathbf{u}}^{n+1}}{\Delta t} + \Delta p^{n+1} = 0,$$

that is an elliptic boundary-value problem with Neumann boundary conditions

$$\begin{cases} -\Delta p^{n+1} = -\operatorname{div} \frac{\tilde{\mathbf{u}}^{n+1}}{\Delta t} & \text{in } \Omega, \\ \frac{\partial p^{n+1}}{\partial n} = 0 & \text{on } \partial\Omega. \end{cases} \quad (15.68)$$

The Neumann condition follows from the condition $\mathbf{u}^{n+1} \cdot \mathbf{n} = 0$ on $\partial\Omega$, see (15.67). From the solution of (15.68) we obtain p^{n+1} and thus \mathbf{u}^{n+1} by using the first equation of (15.67),

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}}^{n+1} - \Delta t \nabla p^{n+1} \quad \text{in } \Omega. \quad (15.69)$$

This is precisely the correction to operate on the velocity field in order to fulfill the divergence-free constraint.

In conclusion, at first we solve the scalar elliptic problem (15.66) to yield the intermediate velocity $\tilde{\mathbf{u}}^{n+1}$, then the elliptic problem (15.68) yields the pressure unknown p^{n+1} , and finally we obtain the new velocity field \mathbf{u}^{n+1} through the explicit correction equation (15.69).

Let us now investigate the main features of this method.

Assume that we take $\mathbf{u}^* = \mathbf{u}^{**} = \mathbf{u}^n$ in the first step; after space discretization, we get a linear system as

$$\left(\frac{1}{\Delta t} M + A \right) \tilde{\mathbf{u}}^{n+1} = \tilde{\mathbf{f}}^{n+1}.$$

The main limitation of this approach consists in the fact that, having treated explicitly the convective term, the solution undergoes a stability restriction on the time step like (15.65). On the other hand, because of this explicit treatment, this linear system naturally splits into d independent systems of smaller size, one for each spatial component of the velocity field.

Should we use instead an implicit time advancing scheme, like the one that we would get by setting $\mathbf{u}^* = \mathbf{u}^{**} = \tilde{\mathbf{u}}^{n+1}$, we would obtain an unconditionally stable scheme, however with a more involved coupling of all the spatial components due to the nonlinear convective term. This nonlinear algebraic system can be solved by, e.g., a Newton-Krylov method, similar to the one that we have introduced in Sec. 15.7.1. In the second step of the method, we enforce a boundary condition only on the normal component of the velocity field, however we lack any control on the behavior of the tangential component of the same velocity at the boundary. This generates a so-called *splitting error*:

although the solution is divergence-free, by failing to satisfy the physical boundary condition on the tangential velocity component yields the onset of a pressure boundary layer of width $\sqrt{\nu \Delta t}$.

The method just described is due to Chorin and Temam, and is also called *projection method*. The reason can be found in the celebrated Helmholtz-Weyl decomposition theorem:

Theorem 15.7 Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, be a domain with Lipschitz boundary. Then, for every $\mathbf{v} \in [L^2(\Omega)]^d$, there exist two (unique) functions \mathbf{w}, \mathbf{z} ,

$$\mathbf{w} \in H_{\text{div}}^0 = \{ \mathbf{v} \in [L^2(\Omega)]^d : \operatorname{div} \mathbf{v} = 0 \text{ in } \Omega, \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega \},$$

$$\mathbf{z} \in [L^2(\Omega)]^d, \quad \operatorname{rot} \mathbf{z} = \mathbf{0} \quad (\text{thus } \mathbf{z} = \nabla \psi, \text{ for a suitable } \psi \in H^1(\Omega))$$

such that

$$\mathbf{v} = \mathbf{w} + \mathbf{z}.$$

Owing to this result, any function $\mathbf{v} \in [L^2(\Omega)]^d$ can be univocally represented as being the sum of a solenoidal (that is, divergence-free) field and of an irrotational field (that is, the gradient of a suitable scalar function).

As a matter of fact, after the first step (15.66) in which the preliminary velocity $\tilde{\mathbf{u}}^{n+1}$ is obtained from \mathbf{u}^n by solving the momentum equation, in the course of the second step a solenoidal field \mathbf{u}^{n+1} is constructed in (15.69), with $\mathbf{u}^{n+1} \cdot \mathbf{n} = 0$ on $\partial\Omega$. This solenoidal field is the projection of $\tilde{\mathbf{u}}^{n+1}$, and is obtained by applying the decomposition theorem with the following identifications: $\mathbf{w} = \tilde{\mathbf{u}}^{n+1}$, $\mathbf{v} = \mathbf{u}^{n+1}$, $\psi = -\Delta t p^{n+1}$.

The name projection method is due to the fact that

$$\int_{\Omega} \mathbf{u}^{n+1} \cdot \psi \, d\Omega = \int_{\Omega} \tilde{\mathbf{u}}^{n+1} \cdot \psi \, d\Omega \quad \forall \psi \in H_{\text{div}}^0,$$

that is \mathbf{u}^{n+1} is the projection, with respect to the scalar product of $L^2(\Omega)$, of $\tilde{\mathbf{u}}^{n+1}$ on the space H_{div}^0 .

Remark 15.3 Several variants of the projection method have been set up with the aim of reducing the splitting error on the pressure, not only for the finite element method but also for higher order spectral or spectral element space approximations. The interested reader can refer to, e.g., [QV94], [Qua93], [Pro97], [CHQZ07, Chap. 3] and [KS05]. •

Example 15.1 In Fig. 15.10 we display the isolines of the modulus of velocity corresponding to the solution of Navier-Stokes equations in a two dimensional domain $\Omega = (0, 17) \times (0, 10)$ with five circular holes. This can be regarded as the orthogonal section of a three dimensional domain with 5 cylinders. A non-homogeneous Dirichlet condition, $\mathbf{u} = [\arctan(20(5 - |5 - y|)), 0]^T$, is assigned at the inflow, a homogeneous Dirichlet condition is prescribed on the horizontal side

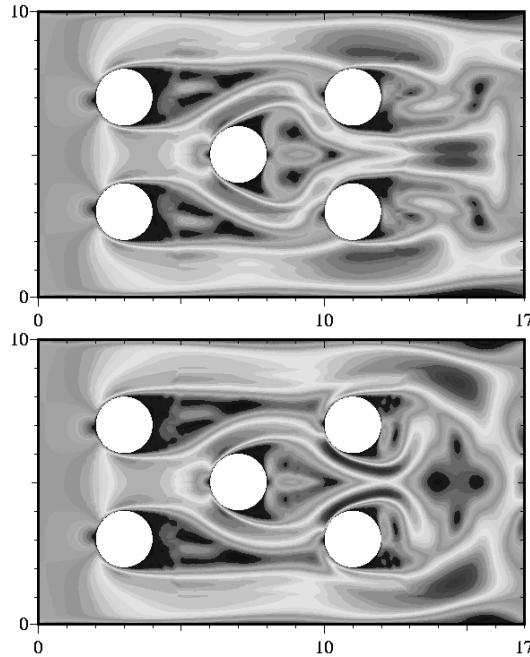


Fig. 15.10. Isolines of the modulus of the velocity vector for the test case of Example 15.1 at the time levels $t = 10.5$ (above) and $t = 11.4$ (below)

as well as on the border of the cylinders, while at the outflow the normal component of the stress tensor is set to zero. For the space discretization the stabilized spectral element method was used, with 114 spectral elements, and polynomials of degree 7 for both the pressure and the velocity components on every element, and a second order BDF2 scheme for temporal discretization (see Sec. 8.5 and also [QSS07]). ■

15.8 Algebraic factorization methods and preconditioners for saddle-point systems

An alternative approach for the solution of systems like (15.49) is the one based on the use of inexact (or incomplete) factorizations of the system matrix (15.50). We remind that these systems can be obtained by the approximate solution of Stokes equations, from that of Navier-Stokes equations (after using one of the linearization approaches described in Sec. 15.7.1, 15.7.2 or 15.7.3), or, more generally, from the approximation of saddle-point problems, as shown in Sec. 15.3. Let us also point out that, in all these cases, the space discretization method can be based on either one of the method discussed thus far (finite elements, finite differences, finite volumes, spectral methods, etc...).

Generally speaking, we will suppose to deal with an algebraic system of the following form

$$\begin{bmatrix} C & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} U \\ P \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix} \quad (15.70)$$

where C coincides with matrix A in the case of system (15.49), with $\frac{1}{\Delta t}M + A$ in case of system (15.64), while more in general it could be given by $\frac{\alpha}{\Delta t}M + A + \delta D$, with D being the matrix associated to the pressure gradient operator, in case a linearization or a semi-implicit treatment are applied to the convective term. In the latter case the coefficients α and δ would depend on the specific linearization or semi-implicit method adopted.

Also in this case we can associate (15.70) with the Schur complement

$$RP = BC^{-1}F, \quad \text{with } R = BC^{-1}B^T, \quad (15.71)$$

which reduces to (15.52) in the case we start from the Stokes system (15.49) instead than from (15.70). We start noticing that the condition number of R depends on the *inf-sup* constant β_h , see (15.41), as well as on the continuity constant δ (see (15.21)). More precisely, in the case of the stationary problem (15.52), the following relations hold

$$\beta_h = \sqrt{\lambda_{\min}}, \quad \delta \geq \sqrt{\lambda_{\max}}$$

where λ_{\min} and λ_{\max} are the eigenvalues of R (see [QV94, Sec. 9.2.1]). Thus, $\text{cond}(R) \leq \delta^2/\beta_h^2$. In the time dependent case we get a system like (15.70); in this case the condition number of R also depends on Δt as well as on the way the convective term has been discretized.

A possible strategy for the solution of (15.52) consists in solving the Schur complement system (15.71) by an iterative method: the conjugate gradient method if $C = A$ (as A is symmetric), otherwise the GMRES or the Bi-CGStab method when $\delta \neq 0$. The use of a convenient preconditioner is mandatory. For a more general discussion, see, e.g., [ESW05], [BGL05] and [QV94] for the case of finite element discretizations, and [CHQZ07] for that of discretization based on spectral methods.

We start by observing that the matrix of system (15.70), that we denote by S , can be written as the LU product of two block triangular matrices,

$$S = \begin{bmatrix} I & 0 \\ BC^{-1} & I \end{bmatrix} \begin{bmatrix} C & B^T \\ 0 & -R \end{bmatrix}.$$

Each one of the two matrices

$$P_D = \begin{bmatrix} C & 0 \\ 0 & -R \end{bmatrix} \quad \text{or} \quad P_T = \begin{bmatrix} C & B^T \\ 0 & -R \end{bmatrix}$$

provides an optimal preconditioner for S , a *block diagonal* preconditioner (P_D), and a *block triangular* preconditioner (P_T). Unfortunately, they are both computationally

expensive because of the presence on the diagonal of the Schur complement R , which, on its turn, contains the inverse of matrix C . Alternatively, we can use their approximants

$$\widehat{P}_D = \begin{bmatrix} \widehat{C} & 0 \\ 0 & -\widehat{R} \end{bmatrix} \quad \text{or} \quad \widehat{P}_T = \begin{bmatrix} \widehat{C} & B^T \\ 0 & -\widehat{R} \end{bmatrix}$$

where \widehat{C} and \widehat{R} are two inexpensive approximations of C and R , respectively. \widehat{C} can be built up from optimal preconditioners of the stiffness matrix, like those that will be introduced in Chap. 17.

The *pressure correction diffusion* preconditioner (PCD) makes use of the following approximation of R

$$\widehat{R}_{PCD} = A_P C_P^{-1} M_P,$$

where M_P is the pressure mass matrix, A_P the pressure Laplacian matrix, C_P the convection-diffusion pressure matrix. The term “pressure” here means that these matrices are generated by using the basis functions $\{\varphi_k, k = 1, \dots, M\}$ of the finite dimensional pressure subspace Q_h . This preconditioner is spectrally equivalent to $B M^{-1} B^T$, where M is the velocity mass matrix. See [ESW05]. Application of this preconditioner requires the action of one Poisson pressure solve, a mass matrix solve, and a matrix-vector product with F_P . Boundary conditions should be taken into account while constructing A_P and C_P .

The *least-squares commutator* preconditioner (LSC) is

$$\widehat{R}_{LSC} = (B \widehat{M}_V^{-1} B^T) (B \widehat{M}_V^{-1} C \widehat{M}_V^{-1} B^T)^{-1} (B \widehat{M}_V^{-1} B^T),$$

where \widehat{M}_V is the diagonal matrix obtained from the velocity mass matrix M by disregarding the extra-diagonal terms. Application of this preconditioner entails two Poisson solves. The convergence of Krylov iterations with the LSC preconditioner is independent of the grid-size and mildly dependent on the Reynolds number. See [EHS⁺06].

The *augmented Lagrangian* preconditioner (AL), introduced in [BO06] reads

$$\widehat{R}_{AL} = (\nu \widehat{M}_P^{-1} + \gamma W^{-1})^{-1}$$

where \widehat{M}_P is a diagonal matrix that approximates M_P , W is a suitably chosen matrix that, in the simplest case, is also given by \widehat{M}_P , ν is the flow viscosity and γ is a positive parameter (usually taken to be 1). This preconditioner requires the original system (15.70) to be modified by replacing the $(1, 1)$ block by $C + \gamma B^T W^{-1} B$, which is consistent because $B u = 0$. The new term $\gamma B^T W^{-1} B$ introduces a coupling between the velocity vector components. Convergence, however, is independent of both the grid-size and the Reynolds number.

Finally, let us remark that direct algebraic preconditioners based on *incomplete LU factorization* (ILU) of the global matrix S can be used, in combination with suitable reordering of the unknowns. An in-depth discussion is made in [uRVS08].

A different LU factorization of S ,

$$S = \begin{bmatrix} C & 0 \\ B & -R \end{bmatrix} \begin{bmatrix} I & C^{-1}B^T \\ 0 & I \end{bmatrix} \quad (15.72)$$

stands at the base of the so-called *SIMPLE* preconditioner introduced in [Pat80], and obtained by replacing C^{-1} in both factors L and U by a triangular matrix D^{-1} (for instance, D could be the diagonal of C). More precisely,

$$P_{SIMPLE} = \begin{bmatrix} C & 0 \\ B & -\hat{R} \end{bmatrix} \begin{bmatrix} I & D^{-1}B^T \\ 0 & I \end{bmatrix} = \hat{L}\hat{U},$$

with $\hat{R} = BD^{-1}B^T$.

Convergence of preconditioned iterative methods deteriorates when the grid-size h decreases and/or the Reynolds number increases.

Note that one step of application of P_{SIMPLE} , say

$$P_{SIMPLE}\mathbf{w} = \mathbf{r} \quad (15.73)$$

with $\mathbf{r} = [\mathbf{r}_u, \mathbf{r}_p]$ and $\mathbf{w} = [\mathbf{u}, \mathbf{p}]$, yields $\hat{L}\mathbf{w}^* = \mathbf{r}$, then $\hat{U}\mathbf{w} = \mathbf{w}^*$, that is, setting $\mathbf{w}^* = [\mathbf{u}^*, \mathbf{p}^*]$:

$$C\mathbf{u}^* = \mathbf{r}_u, \quad (15.74)$$

$$\hat{R} = B\mathbf{u}^* - \mathbf{r}_p, \quad (15.75)$$

$$\mathbf{u} = \mathbf{u}^* - D^{-1}B^T\mathbf{p}^*. \quad (15.76)$$

This requires a C -solve for the velocity and a pressure Poisson solve (for matrix $BD^{-1}B^T$).

Several generalizations of the *SIMPLE* preconditioner have been proposed, in particular *SIMPLER*, h -*SIMPLE* and *MSIMPLER*. Application of $P_{SIMPLER}$ instead of P_{SIMPLE} in (15.73) yields the following steps:

$$\hat{R}\mathbf{p}^0 = BD^{-1}\mathbf{r}_u - \mathbf{r}_p, \quad (15.77)$$

$$C\mathbf{u}^* = \mathbf{r}_u - B^T\mathbf{p}^0, \quad (15.78)$$

$$\hat{R}\mathbf{p}^* = B\mathbf{u}^* - \mathbf{r}_p, \quad (15.79)$$

$$\mathbf{u} = \mathbf{u}^* - D^{-1}B^T\mathbf{p}^*, \quad (15.80)$$

$$\mathbf{p} = \mathbf{p}^* + \omega\mathbf{p}^0, \quad (15.81)$$

$$(15.82)$$

with $\omega \in]0, 1]$ being a possible relaxation parameter ($\omega = 1$ in *SIMPLER*, $\omega \neq 1$ in *SIMPLER*(ω)). It therefore involves two pressure Poisson solves and one C -velocity solve; however, in general it enjoys faster convergence than *SIMPLE*.

The preconditioner *hSIMPLE* (*h*=hybrid) is based on a combined application of *SIMPLE* and *SIMPLER* preconditioner. Finally, the preconditioner *MSIMPLER* makes use of the same steps (15.77)–(15.81) as *SIMPLER*, however the approximate Schur complement $\widehat{R} = BD^{-1}B^T$ is replaced by the least-squares commutator \widehat{R}_{LSC} . The convergence is better than other variants of *SIMPLE*.

For more discussion and a comparative analysis see [uRVS09] and also [Wes01].

The $\widehat{L}\widehat{U}$ factorization used in P_{SIMPLE} can be regarded as a special case of a more general family of inexact or algebraic factorizations that read as follows

$$\widehat{S} = \widehat{L}\widehat{U} = \begin{bmatrix} C & 0 \\ B & -B\mathcal{L}B^T \end{bmatrix} \begin{bmatrix} I & \mathcal{U}B^T \\ 0 & I \end{bmatrix}. \quad (15.83)$$

Here \mathcal{L} and \mathcal{U} represent two (not necessarily coincident) possible approximations of C^{-1} . Using this inexact factorization, the solution of the linear system

$$\widehat{S} \begin{bmatrix} \widehat{\mathbf{u}} \\ \widehat{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}$$

can be accomplished through the following steps:

$$\begin{aligned} \text{step } \widehat{L} : \quad & \left\{ \begin{array}{ll} Cu^* = \mathbf{F} & \text{(intermediate velocity)} \\ -B\mathcal{L}B^T \widehat{\mathbf{p}} = -Bu^* & \text{(pressure)} \end{array} \right. \\ \text{step } \widehat{U} : \quad & \widehat{\mathbf{u}} = \mathbf{u}^* - \mathcal{U}B^T \widehat{\mathbf{p}} \quad \text{(final velocity).} \end{aligned}$$

When used in connection with time-dependent (either Stokes or Navier-Stokes) problems, e.g. like (15.64), two different possibilities have been exploited in [QSV00]:

$$\mathcal{L} = \mathcal{U} = \left(\frac{1}{\Delta t} M \right)^{-1}, \quad (15.84)$$

$$\mathcal{L} = \left(\frac{1}{\Delta t} M \right)^{-1} \quad \text{and} \quad \mathcal{U} = C^{-1}. \quad (15.85)$$

The former (15.84) is named as *Chorin-Temam algebraic approximation* because the steps \widehat{L} and \widehat{U} previously described can be regarded as the algebraic counterpart of the Chorin-Temam fractional step method previously described (see Sec. 15.7.3).

The second choice, (15.85), is said to be the *Yosida approximation* as it can be interpreted as a Yosida regularization of the Schur complement ([Ven98]).

The potential advantage of this strategy w.r.t. that based on differential fractional step methods described above is that it does not require any special care about boundary conditions. The latter are implicitly accounted for in the algebraic formulation (15.70) and no further requirement is needed in the course of the \widehat{L} and \widehat{U} steps.

Several generalizations of the inexact factorization technique (15.83) are possible, based on different choices of the factors \mathcal{L} and \mathcal{U} . In case the time dependent Navier-Stokes equations are discretized in time by high order (≥ 2) temporal schemes, inexact factors are chosen in a way so that the time discretization order is maintained. See [GSV06, SV05, Ger08].

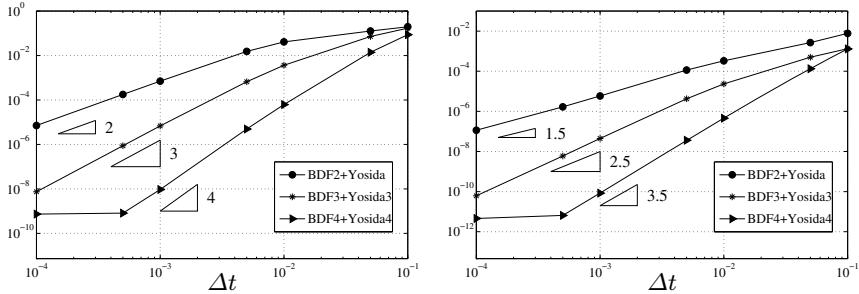


Fig. 15.11. Velocity errors E_u on the left; pressure errors E_p on the right

In Fig. 15.11 we display the error behavior corresponding to the approximation of the time dependent Navier-Stokes equations on the domain $\Omega = (0, 1)^2$ using the spectral element method (SEM) with 4×4 square elements with side-length $H = 0.25$ and polynomials of degree $N = 8$ for the velocity components and $N = 6$ for the pressure. The exact solution is $\mathbf{u}(x, y, t) = (\sin(x) \sin(y + t), \cos(x) \cos(y + t))^T$, $p(x, y, t) = \cos(x) \sin(y + t)$. The temporal discretization is based on implicit backward differentiation formulae of order 2 (BDF2), 3 (BDF3), and 4 (BDF4) (see [QSS07]), then on inexact (Yosida) algebraic factorizations of order 2, 3, and 4, respectively. Denoting by (\mathbf{u}_N^n, p_N^n) the numerical solution at the time level t^n , the errors on velocity and pressure are defined as

$$E_u = \left(\Delta t \sum_{n=0}^{N_T} \|\mathbf{u}(t^n) - \mathbf{u}_N^n\|_{H^1(\Omega)}^2 \right)^{1/2} \quad \text{and}$$

$$E_p = \left(\Delta t \sum_{n=0}^{N_T} \|p(t^n) - p_N^n\|_{L^2(\Omega)}^2 \right)^{1/2}$$

Errors on velocity are infinitesimal with respect to Δt of order 2, 3, and 4, respectively, whereas those on pressure are of order $3/2$, $5/2$ and $7/2$, respectively.

15.9 Free surface flow problems

Free surface flows can manifest under various situations and manyfold shapes. A free surface is generated every time that two immiscible fluids get in contact. They can give raise to jets, [LR98], bubbles [HB76], [TF88], droplets [Max76] and films. This kind of fluids are encountered in a variety of different applications, such as waves in rivers, lakes and oceans [Bla02], [Qu02] and the way they interact with solid media (boats, coasts, etc.) [Wya00], [KMI⁺83], injection, moulding and extrusion of polymers and liquid metals [Cab03], chemical reactors or bioreactors, etc. Depending upon spatial and temporal scales involved, processes like heat transfer, surface tension, laminar to turbulent transition, compressibility and chemical reactions, interaction with solids, might have a relevant impact on the flow behavior. In what follows we will focus on

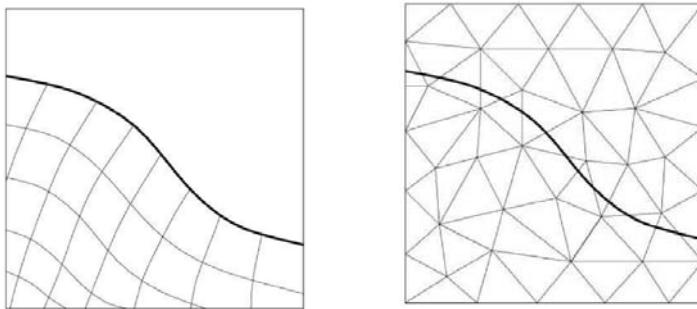


Fig. 15.12. Two typical grids in two dimensions for front tracking methods (left) and front capturing methods (right). The thick line represents the free surface

laminar flows for viscous newtonian fluids, subject to surface tension; in these circumstances, the flow can be described by the incompressible Navier-Stokes equations.

When modeling this kind of fluids, two different approaches can be pursued:

- *Front tracking methods.* These methods consider the free surface as being the boundary of a moving domain on which suitable boundary conditions are specified. At the interior of the domain, a conventional fluid model is used; special attention however should be paid to the fact that the domain is not fixed. On the other side of the domain, the fluid, e.g. the air, is usually neglected, or otherwise modeled in a simplified fashion without explicitly solving it (see, e.g., [MP97]).
- *Front capturing methods.* The two fluids are in fact considered as a single fluid in a domain with fixed boundaries whose properties like density and viscosity vary as piecewise constant functions. The discontinuity line is in fact the free surface (see, e.g., [HW65], [HN81]).

For a review on numerical methods for free-boundary problems, see [Hou95].

In what follows we will consider front capturing methods. More precisely, we will derive a mathematical model for the case of a general fluid with variable density and viscosity, which would therefore be appropriate to model the flow of two fluids separated by a free surface.

15.9.1 Navier-Stokes equations with variable density and viscosity

We consider the general case of a viscous incompressible flow whose density ρ and dynamical viscosity μ vary both in space and in time. Within a given spatial domain $\Omega \subset \mathbb{R}^d$, the evolution of the fluid velocity $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ and pressure $p = p(\mathbf{x}, t)$ are modeled by the following equations

$$\rho \partial_t \mathbf{u} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} - \operatorname{div} (2\mu \mathbf{D}(\mathbf{u})) + \nabla p = \mathbf{f}, \quad \mathbf{x} \in \Omega, t > 0, \quad (15.86)$$

$$\operatorname{div} \mathbf{u} = 0, \quad \mathbf{x} \in \Omega, t > 0, \quad (15.87)$$

in which $(\mathbf{D}(\mathbf{v})) = \frac{\nabla \mathbf{v} + \nabla \mathbf{v}^T}{2}$ is the *symmetric gradient* of \mathbf{v} , also called *rate of deformation tensor*, while \mathbf{f} denotes a volumetric force, for instance the gravity force. (Here ∂_t stands for $\partial/\partial t$.)

These equations must be supplemented by suitable initial and boundary conditions. In case ρ is constant we retrieve the form (15.1). Note that incompressibility is not in contradiction with variable density. Incompressibility means that one fluid parcel does not change volume and thus density, whereas variable density means that different fluid parcels may have different densities. The last two terms of the left hand side (15.86) can be rewritten as $-\operatorname{div}\mathbf{T}(\mathbf{u}, p)$, where

$$\mathbf{T}(\mathbf{u}, p) = 2\mu\mathbf{D}(\mathbf{u}) - \mathbf{I}p$$

is the *stress tensor* while \mathbf{I} is the $d \times d$ identity tensor. The divergence of a tensor has been introduced in (14.23). A complete derivation of this model can be found, e.g., in [LL59].

An equation for the density function ρ can be obtained from the mass balance equation

$$\begin{aligned} D_t \rho &= \partial_t \rho + \mathbf{u} \cdot \nabla \rho = 0, & \mathbf{x} \in \Omega, t > 0, \\ \rho|_{t=0} &= \rho_0, & \mathbf{x} \in \Omega, \end{aligned} \quad (15.88)$$

where D_t indicates the material, or Lagrangian, derivative, see Sec. 15.7.2. In those cases in which viscosity μ can be expressed in terms of the density, that is $\mu = \mu(\rho)$, this relation, together with (15.88), provides the model for the evolution of ρ and μ . Models adapted to the special case of a flow of two fluids are described in Sec. 15.9.3.

The analysis of the coupled problem (15.86)-(15.87)-(15.88) is a hard topic. We refer the reader to [Lio96]. A global existence result can be proven if $\mathbf{f} = \rho \mathbf{g}$ and $\sigma = 0$. This proof requires Ω to be a smooth, bounded, connected open subset of \mathbb{R}^d , and that homogeneous Dirichlet boundary conditions (i.e., with $\mathbf{g}_D = \mathbf{0}$) are imposed on the whole boundary. If the initial and source data satisfy

$$\begin{aligned} \rho_0 &\geq 0 \quad \text{a.e. in } \Omega, \quad \rho_0 \in L^\infty(\Omega), \quad \rho_0 \mathbf{u}_0 \in L^2(\Omega)^d, \quad \rho_0 |\mathbf{u}_0|^2 \in L^1(\Omega), \\ \text{and } \mathbf{g} &\in L^2(\Omega \times (0, T))^d, \end{aligned}$$

then there exist global weak solutions which satisfy

$$\begin{aligned} \rho &\in L^\infty(\Omega \times (0, T)), \quad \rho \in C([0, \infty); L^p(\Omega)) \quad \forall p \in [1, \infty); \\ \mathbf{u} &\in [L^2(0, T; H_0^1(\Omega))]^d, \quad \nabla \mathbf{u} \in [L^2(\Omega \times (0, T))]^{d \times d}; \\ \rho |\mathbf{u}|^2 &\in L^\infty(0, T; L^1(\Omega)). \end{aligned}$$

Another result by Tanaka [Tan93] treats the case where the surface tension coefficient σ is different from zero but constant. Under some (stronger) regularity assumptions on the initial data, it has been proven that a global solution exists for sufficiently small initial data and external forces. Moreover, local (in time) uniqueness is proved.

15.9.2 Boundary conditions

Let us generalize the discussion on boundary conditions that we have carried out at the beginning of this chapter to the case of the more general formulation (15.86), (15.87) of Navier-Stokes equations. We still consider a splitting of the boundary $\partial\Omega$ of the domain Ω into a finite number of components and impose therein appropriate boundary conditions. Several kind of conditions are admissible: for a general discussion see, e.g., [QV94] and the references therein. In the following we will limit to describe the most commonly used conditions for free surface flows.

The Dirichlet boundary conditions prescribe the value of the velocity vector on a boundary subset Γ_D

$$\mathbf{u} = \varphi \quad \text{on} \quad \Gamma_D \subset \partial\Omega. \quad (15.89)$$

They are used either for imposing a velocity profile on the *inflow* boundary, or to model a solid boundary moving with a prescribed velocity. In the latter case they are said to be *no-slip* boundary condition, as they impose the fluid not to slip but to stick at the wall.

As we have already noted, when Dirichlet boundary conditions are specified on the entire boundary $\partial\Omega$, the pressure is not uniquely defined. In this case, if (\mathbf{u}, p) is a solution of (15.86), (15.87) and (15.89), then $(\mathbf{u}, p + c)$, $c \in \mathbb{R}$ is also a solution of the same set of equations. Using the Gauss theorem, from equation (15.87), it follows that \mathbf{g}_D has to satisfy the compatibility condition

$$\int_{\partial\Omega} \mathbf{g}_D \cdot \mathbf{n} \, d\gamma = 0.$$

Neumann boundary conditions prescribe a force ψ_N per unit area as the normal component of the stress tensor

$$\mathbf{T}(\mathbf{u}, p)\mathbf{n} = 2\mu\mathbf{D}(\mathbf{u})\mathbf{n} - p\mathbf{n} = \psi \quad \text{on} \quad \Gamma_N \subset \partial\Omega, \quad (15.90)$$

where \mathbf{n} is the outer unit normal on Γ_N . When $\psi_N = \mathbf{0}$ the subset Γ_N is called a *free outflow*. For vanishing velocity gradients, the force ψ_N corresponds to the pressure on the boundary. See also [HRT96] for more details about the interpretation and implications of this type of boundary conditions. Neumann boundary conditions are used to model a given force per unit area \mathbf{g}_N on the boundary.

Mixed boundary conditions prescribe values of the normal component of the velocity field, as well as on the tangential component of the normal stresses, that is:

$$\begin{aligned} \mathbf{u} \cdot \mathbf{n} &= \varphi \cdot \mathbf{n} && \text{on } \Gamma_D, \\ (\mathbf{T}(\mathbf{u}, p)\mathbf{n}) \cdot \boldsymbol{\tau} &= (2\mu\mathbf{D}(\mathbf{u})\mathbf{n}) \cdot \boldsymbol{\tau} = 0 && \text{on } \Gamma_N, \quad \forall \boldsymbol{\tau} : \boldsymbol{\tau} \cdot \mathbf{n} = 0. \end{aligned}$$

The choice $\varphi = \mathbf{0}$ models symmetry of the solution along Γ_D , but also free slip on Γ_D without penetration. In this case we talk about *free-slip* boundary conditions.

In some situations, a smooth transition from slip to no-slip boundary conditions is desired. This can be realized by imposing Dirichlet boundary conditions in the normal

direction, as for the free slip boundary conditions, and to replace the boundary condition in the tangential direction by Robin boundary conditions, a linear combination of Dirichlet and Neumann boundary conditions:

$$\begin{aligned} \mathbf{u} \cdot \mathbf{n} &= \varphi \cdot \mathbf{n} && \text{on } \Gamma_D, \\ (\omega C_\tau \mathbf{u} + (1 - \omega)(\mathbf{T}(\mathbf{u}, p)\mathbf{n})) \cdot \boldsymbol{\tau} &= \\ (\omega C_\tau \mathbf{u} + (1 - \omega)(2\mu \mathbf{D}(\mathbf{u})\mathbf{n})) \cdot \boldsymbol{\tau} &= \omega C_\tau \mathbf{g}_D \cdot \boldsymbol{\tau} && \forall \boldsymbol{\tau} : \boldsymbol{\tau} \cdot \mathbf{n} = 0. \end{aligned}$$

Here, $\omega \in [0, 1]$ determines the regime. For $\omega = 0$, we have free-slip boundary conditions, whereas for $\omega = 1$, we have no-slip boundary conditions. In practice, ω can be a smooth function of space and time, with values in $[0, 1]$, allowing thus a smooth transition between the two cases. This holds for $\varphi = \mathbf{0}$, but transition boundary conditions cover also the general Dirichlet case for $\varphi \neq \mathbf{0}$ and $\omega = 1$. The weight C_τ can be seen as conversion factor between velocities and force per unit area. This type of boundary conditions has been studied in more details in [Joe05].

15.9.3 Application to free surface flows

A free surface flow can be modeled by (15.86)-(15.87). In this perspective, the free surface is an interface denoted by $\Gamma(t)$, cutting the domain Ω into two open subdomains $\Omega^+(t)$ and $\Omega^-(t)$. The initial position of the interface is known, $\Gamma(0) = \Gamma_0$, and the interface moves with the fluid velocity \mathbf{u} . On each subdomain, we have the constant densities and viscosities denoted by ρ^+ , ρ^- , μ^+ and μ^- . We require $\rho^\pm > 0$ and $\mu^\pm > 0$.

Density and viscosity are then globally defined as follows:

$$\rho(\mathbf{x}, t) = \begin{cases} \rho^- & \mathbf{x} \in \Omega^-(t) \\ \rho^+ & \mathbf{x} \in \Omega^+(t), \end{cases} \quad \mu(\mathbf{x}, t) = \begin{cases} \mu^- & \mathbf{x} \in \Omega^-(t) \\ \mu^+ & \mathbf{x} \in \Omega^+(t). \end{cases}$$

In order to model buoyancy effects, the gravitation force $\mathbf{f} = \rho \mathbf{g}$, where \mathbf{g} is the vector of gravity acceleration, has to be introduced into the right hand side.

As the viscosity is discontinuous across the interface, equation (15.86) can hold strongly only at the interior of the two subdomains. The latter must therefore be coupled with suitable interface conditions (see, e.g., [Smo01]).

We denote by \mathbf{n}_Γ the interface unit normal pointing from Ω^- into Ω^+ and by κ the interface curvature, defined as

$$\kappa = \sum_{i=1}^{d-1} \frac{1}{R_{\boldsymbol{\tau}_i}}, \quad (15.91)$$

where $R_{\boldsymbol{\tau}_i}$ are the radii of curvature along the principal vectors $\boldsymbol{\tau}_i$ which span the tangential space to the interface Γ . The sign of $R_{\boldsymbol{\tau}_i}$ is such that $R_{\boldsymbol{\tau}_i} \mathbf{n}_\Gamma$ points from Γ to the center of the circle approximating Γ locally.

The jump of a quantity v across the interface is denoted by $[v]_\Gamma$ and defined as

$$\begin{aligned} [v]_\Gamma(\mathbf{x}, t) &= \lim_{\epsilon \rightarrow 0^+} (v(\mathbf{x} + \epsilon \mathbf{n}_\Gamma, t) - v(\mathbf{x} - \epsilon \mathbf{n}_\Gamma, t)) \\ &= v|_{\Omega^+(t)}(\mathbf{x}, t) - v|_{\Omega^-(t)}(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Gamma(t). \end{aligned}$$

The interface conditions then read

$$[\mathbf{u}]_\Gamma = \mathbf{0}, \quad (15.92)$$

$$[\mathbf{T}(\mathbf{u}, p)\mathbf{n}_\Gamma]_\Gamma = [2\mu\mathbf{D}(\mathbf{u})\mathbf{n}_\Gamma - p\mathbf{n}_\Gamma]_\Gamma = \sigma\kappa\mathbf{n}_\Gamma. \quad (15.93)$$

Equation (15.92) is called the *kinematic interface condition*. It expresses the property that all components of the velocity are continuous. In fact the normal component has to be continuous because there is no flow through the interface, whereas the tangential component(s) have to be continuous because both fluids are assumed viscous ($\mu^+ > 0$ and $\mu^- > 0$).

Equation (15.93) is referred to as the *dynamic interface condition*. It expresses the property that the normal stress jumps by the amount of the surface tension force. This force is proportional to the interface curvature and pointing to the direction of the interface normal. The surface tension coefficient σ depends on the fluid pairing, and in general also on temperature. We will assume it to be constant as all heat transfer effects are neglected.

Note that the evolution of the interface has to be compatible with the mass conservation equation (15.88). Mathematically, this equation has to be understood in the weak sense, i.e., in the sense of distributions, as the density is discontinuous across the interface and its derivatives can by consequence only be interpreted weakly.

As this form of the mass conservation equation is often not convenient for numerical simulations, other equivalent models that describe the evolution of the interface $\Gamma(t)$ have been introduced. A short overview is presented in Sec. 15.10.

15.10 Interface evolution modeling

We give here a short overview of different approaches for modelling the evolution of an interface $\Gamma(t)$ in a fixed domain Ω .

15.10.1 Explicit interface descriptions

An interface can be represented explicitly by a set of marker points or line segments (in 2D, surface segments in 3D) on the interface and transported by the fluid velocity.

In the case of marker points, introduced in [HW65], the connectivity of the interface between the points is not known and has to be reconstructed whenever needed. In order to simplify this task, additional markers are usually placed near the interface, marking Ω^+ or Ω^- . The advection of the markers is simple, and connectivity can change easily. However it is still somewhat cumbersome to reconstruct the interface from the marker distribution. Typically, it is also necessary to redistribute the markers, to introduce new ones or to discard existing ones.

Several markers can be connected to define a line or surface, either straight (plane) or curved, e.g. by nurbs. A set of such geometrical objects can now define the surface. Its evolution is modeled by the evolution of the constituting objects, and thus by the markers defining them. The connectivity of the interface is thereby conserved, which

solves the difficulty of pure marker methods, and brings a new drawback in turn: topological changes of the interface are allowed by the underlying physics but not by this description. Sophisticated procedures have to be applied to detect and handle interface breakup correctly.

15.10.2 Implicit interface descriptions

In front capturing methods, the interface is represented implicitly by the value of a scalar function $\phi : \Omega \times (0, T) \rightarrow \mathbb{R}$ that encodes at each point \mathbf{x} to which subset it belongs: $\Omega^+(t)$ or $\Omega^-(t)$. A transport equation solved for ϕ then describes the evolution of the interface. By this feature, all implicit interface models share the advantage that topology changes of the interface are possible in the model, and that these happen without special intervention.

Volume of fluid methods. The *volume of fluid methods* (VOF) were originally introduced by Hirt and Nichols [HN81]. Let ϕ be a piecewise constant function s.t.

$$\phi(\mathbf{x}, t) = \begin{cases} 1, & \mathbf{x} \in \Omega^+(t), \\ 0, & \mathbf{x} \in \Omega^-(t); \end{cases}$$

the interface $\Gamma(t)$ is thus located at the discontinuity of the function ϕ , while density and viscosity are simply defined as

$$\begin{aligned} \rho &= \rho^- + (\rho^+ - \rho^-)\phi, \\ \mu &= \mu^- + (\mu^+ - \mu^-)\phi. \end{aligned} \tag{15.94}$$

The transport equation is usually discretized with cell centered finite volume methods, approximating ϕ by a constant value in each grid cell (see Sec. 9.1 and 15.12). Due to discretization errors and diffusive transport schemes, the approximation ϕ will take values between 0 and 1, which by the virtue of equation (15.94) can be (and usually are) interpreted as the volume fraction of the fluid occupying Ω^+ . This explains the name *volume of fluid*. Volume fractions between 0 and 1 actually represent a mixture of the two fluids. As the fluids are assumed immiscible, this behaviour is not desired, especially because mixing effects may not stay concentrated near the interface but spread over the whole domain Ω . Like this, the supposedly sharp interface becomes more and more diffuse. Several techniques exist to limit this problem. Elaborate procedures have been developed for the reconstruction of normals and curvature of a diffuse interface.

Volume of fluid methods have the advantage that applying a conservative discretization of the transport equation ensures mass conservation of the fluid, because the relation (15.94) between ϕ and ρ is linear.

Level set methods. In order to circumvent the problems with volume of fluid methods, Dervieux and Thomasset [DT80] proposed in 1980 to define the interface as the zero level set of a continuous *pseudo-density* function and to apply this method to flow problems. Their approach was then studied more systematically in [OS88] and subsequent publications, where the term *level set method* was coined. Their first application

to flow problems was by Mulder, Osher and Sethian in 1992 [MOS92]. In contrast with volume of fluid approaches, these methods allow to keep the interface sharp, as ϕ is defined as a *continuous* function such that

$$\begin{aligned}\phi(\mathbf{x}, t) &> 0, \quad \mathbf{x} \in \Omega^+(t), \\ \phi(\mathbf{x}, t) &< 0, \quad \mathbf{x} \in \Omega^-(t), \\ \phi(\mathbf{x}, t) &= 0, \quad \mathbf{x} \in \Gamma(t).\end{aligned}$$

The function ϕ is called *level set function*, because the interface $\Gamma(t)$ is its zero level set, its isoline or isosurface associated to the value zero

$$\Gamma(t) = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}, t) = 0\}. \quad (15.95)$$

The density and the viscosity can now be expressed in function of ϕ as

$$\rho = \rho^- + (\rho^+ - \rho^-)H(\phi), \quad (15.96)$$

$$\mu = \mu^- + (\mu^+ - \mu^-)H(\phi), \quad (15.97)$$

where $H(\cdot)$ is the Heaviside function

$$H(\xi) = \begin{cases} 0, & \xi < 0 \\ 1, & \xi > 0. \end{cases}$$

By construction, the interface stays sharp in a level set model, and the immiscible fluids do not start to mix. Also, the determination of the normals and the curvature of the interface are more straightforward and very natural. In turn, as the relation (15.96) is not linear, applying a conservative discretization of the transport equation for ϕ does not ensure mass conservation of the fluid after discretization. This is not a big problem however, as the mass error still disappears with grid refinement and is outweighed by advantages of the level set formulation.

The evolution of the free surface is described by an advection equation for the level set function

$$\begin{aligned}\partial_t \phi + \mathbf{u} \cdot \nabla \phi &= 0 && \text{in } \Omega \times (0, T), \\ \phi &= \phi_0 && \text{in } \Omega \text{ at } t = 0, \\ \phi &= \phi_{in} && \text{on } \partial \Sigma_{in} \times (0, T),\end{aligned} \quad (15.98)$$

where Σ_{in} is the inflow boundary

$$\Sigma_{in} = \{(\mathbf{x}, t) \in \partial \Omega \times (0, T) : \mathbf{u}(\mathbf{x}, t) \cdot \mathbf{n} < 0\}.$$

The flow equations (15.86)-(15.87) and the level set equation (15.98) are therefore coupled. Equation (15.98) can be derived as follows [MOS92]: let $\bar{\mathbf{x}}(t)$ be the path of a point on the interface $\Gamma(t)$. This point moves with the fluid, thus $D_t \bar{\mathbf{x}}(t) = \mathbf{u}(\bar{\mathbf{x}}(t), t)$. Since the function ϕ is always zero on the moving interface, we must have

$$\phi(\bar{\mathbf{x}}(t), t) = 0.$$

Deriving with respect to time and applying the chain rule, we obtain

$$\partial_t \phi + \nabla \phi \cdot \mathbf{u} = 0 \quad \text{on } \Gamma(t) \quad \forall t \in (0, T). \quad (15.99)$$

If we consider instead a path of a point in Ω^\pm , we may require $\phi(\bar{\mathbf{x}}(t), t) = \pm c$, $c > 0$, in order to ensure that the sign of $\phi(\bar{\mathbf{x}}, t)$ does not change and that $\bar{\mathbf{x}}(t) \in \Omega^\pm(t)$ for all t thereby.

In this way, equation (15.99) generalizes to the whole domain Ω , which gives us equation (15.98).

We can now verify that mass conservation is satisfied: using (15.96), we obtain formally

$$\begin{aligned} \partial_t \rho + \mathbf{u} \cdot \nabla \rho &= (\rho^+ - \rho^-)(\partial_t H(\phi) + \mathbf{u} \cdot \nabla H(\phi)) \\ &= (\rho^+ - \rho^-)\delta(\phi)(\partial_t \phi + \mathbf{u} \cdot \nabla \phi) \end{aligned} \quad (15.100)$$

where $\delta(\cdot)$ denotes the Dirac delta function. By equation (15.98), the third factor in (15.100) is zero. Hence equation (15.88) holds and the mass conservation is satisfied by the level set interface evolution model.

Interface related quantities. In the context of two fluid flow, the interface normal and curvature are of particular interest. Namely the surface tension is proportional to the curvature and acting in the normal direction.

We give here an intuitive derivation of these quantities in function of ϕ , without going into the details of differential geometry. See, e.g., [Spi99] for a detailed and rigorous derivation.

The unit normal \mathbf{n}_Γ is orthogonal to all tangential directions $\boldsymbol{\tau}$, which in turn are characterized by the fact that the directional derivative of ϕ in any tangential direction must vanish:

$$0 = \partial_{\boldsymbol{\tau}} \phi = \nabla \phi \cdot \boldsymbol{\tau} \quad \text{on } \Gamma.$$

The gradient of ϕ is thus orthogonal to all tangential directions, and we can define the interface unit normal by normalizing it

$$\mathbf{n}_\Gamma = \frac{\nabla \phi}{|\nabla \phi|}. \quad (15.101)$$

Note that by this definition, \mathbf{n}_Γ points from Ω^- into Ω^+ . Moreover, as ϕ is defined not only on the interface but in the whole domain, the expression for the normal generalizes naturally to the entire domain, too.

In order to derive the expression for the curvature, we need to consider the principal tangential direction(s) $\boldsymbol{\tau}_i$, $i = 1 \dots d - 1$. These are the directions in which the interface is approximated by a circle (cylinder), i.e., the directional derivative of \mathbf{n}_Γ in direction $\boldsymbol{\tau}_i$ has itself direction $\boldsymbol{\tau}_i$

$$\partial_{\boldsymbol{\tau}_i} \mathbf{n}_\Gamma = \nabla \mathbf{n}_\Gamma \boldsymbol{\tau}_i = -\kappa_i \boldsymbol{\tau}_i, \quad \kappa_i \in \mathbb{R}, \quad i = 1 \dots d - 1. \quad (15.102)$$

The bigger $|\kappa_i|$, the more curved is the surface in this direction, and the κ_i are in fact called *principal curvatures*. It follows from straightforward computations that $\kappa_i =$

$(R_{\tau_i})^{-1}$, where the values R_{τ_i} are the radii of the approximating circles (cylinders) as in equation (15.91).

We can see from equation (15.102) that the $d - 1$ values $-\kappa_i$ are eigenvalues of the $d \times d$ -tensor $\nabla \mathbf{n}_\Gamma$. By (15.101), \mathbf{n}_Γ is (essentially) a gradient field which is smooth near the interface. The rank two tensor $\nabla \mathbf{n}_\Gamma$ is thus (essentially) a tensor of second derivatives of a smooth function and thereby symmetric. So it has one more real eigenvalue, whose associated eigenvector must be \mathbf{n}_Γ , because the eigenvectors of a symmetric tensor are orthogonal. It is easy to see that the respective eigenvalue is zero

$$(\nabla \mathbf{n}_\Gamma \mathbf{n}_\Gamma)_i = \sum_{j=1}^d (\partial_{x_i} n_j) n_j = \sum_{j=1}^d \frac{1}{2} \partial_{x_i} (n_j^2) = \frac{1}{2} \partial_{x_i} |\mathbf{n}_\Gamma|^2 = 0,$$

as $|\mathbf{n}_\Gamma| = 1$ by construction (15.101).

Starting from equation (15.91), we obtain for the curvature

$$\kappa = \sum_{i=1}^{d-1} \frac{1}{R_{\tau_i}} = \sum_{i=1}^{d-1} \kappa_i = -\text{tr}(\nabla \mathbf{n}_\Gamma) = -\nabla \cdot \mathbf{n}_\Gamma,$$

and using equation (15.101), we get

$$\kappa = -\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right). \quad (15.103)$$

Initial Condition. We know the position of the interface at $t = 0$, Γ_0 , nevertheless the associated level set function ϕ_0 is not uniquely defined. The freedom of choice can be used to simplify further subsequent tasks. We notice that steep gradients of ϕ make the numerical solution of equation (15.98) more difficult (see e.g. [QV94]), whereas flat gradients decrease the numerical stability when determining Γ from ϕ . A good compromise is thus the further constraint $|\nabla \phi| = 1$.

A function which fulfills this constraint is the distance function

$$\text{dist}(\mathbf{x}; \Gamma) = \min_{\mathbf{y} \in \Gamma} |\mathbf{x} - \mathbf{y}|,$$

which at each point \mathbf{x} takes the value of the closest Euclidean distance from \mathbf{x} to Γ . Multiplying this function by -1 on Ω^- , we obtain the *signed distance function*

$$\text{sdist}(\mathbf{x}; \Gamma) = \begin{cases} \text{dist}(\mathbf{x}; \Gamma), & \mathbf{x} \in \Omega^+, \\ 0, & \mathbf{x} \in \Gamma, \\ -\text{dist}(\mathbf{x}; \Gamma), & \mathbf{x} \in \Omega^-. \end{cases}$$

It is thus usual and reasonable to choose ϕ_0 representing an initial interface Γ_0 as $\phi_0(\mathbf{x}) = \text{sdist}(\mathbf{x}; \Gamma_0)$.

Since $|\nabla \phi| = 1$, the expressions of the interface normal and curvature simplify further to

$$\mathbf{n}_\Gamma = \nabla \phi \quad \text{and} \quad \kappa = -\nabla \cdot \nabla \phi = -\Delta \phi.$$

Reinitialization. Unfortunately, the property $|\nabla\phi| = 1$ is not preserved under advection of ϕ with the fluid velocity \mathbf{u} . This is not a problem as long as $|\nabla\phi|$ does not stay too far from 1, which however cannot be guaranteed in general. Two different strategies can be followed to cope with this issue.

One approach is to determine an advection velocity field that gives the same interface motion as the fluid velocity field, while preserving the distance property. In fact such a velocity field exists and is called *extension velocity*, as it is constructed by extending the velocity prescribed on the interface to the whole domain; efficient algorithms are described in [AS99].

Alternatively, we can still use the fluid velocity \mathbf{u} for advecting the level set function ϕ , and intervene when $|\nabla\phi|$ becomes too large or too small. The action to be taken in this case is known as *reinitialization*, as the procedure is partially the same as for initialization with the initial condition. Suppose we decide to reinitialize at time $t = t_r$:

1. given $\phi(\cdot, t_r)$, find $\Gamma(t_r) = \{\mathbf{x} : \phi(\mathbf{x}, t_r) = 0\}$;
2. replace $\phi(\cdot, t_r)$ by $\text{sdist}(\cdot, \Gamma(t_r))$.

Interestingly, it turns out that the problem of finding the extension velocity is closely related to the problem of reinitializing ϕ to a signed distance function. The same algorithms can be used and the same computational cost has to be expected. Two conceptual differences favour the reinitialization approach though: firstly, the extension velocities have to be computed at every time step, whereas reinitialization can be performed only when necessary, which results in a global reduction of the computational costs. Secondly, the approximated extension velocities will only approximately conserve the distance property and may not guarantee that reinitialization is unnecessary.

Algorithmic details about the efficient construction of an approximation to the signed distance function, especially for the three-dimensional case, can be found in [Win07].

15.11 Finite volume approximation

The finite volume approach is widely used for the solution of problems described by differential equations, with applications in different engineering fields. In particular, the most frequently used commercial codes in the field of fluid dynamics adopt finite volume schemes for the solution of Navier-Stokes equations. The latter are often coupled with models of turbulence, transition, combustion, transport and reaction of chemical species.

When applied to incompressible Navier-Stokes equations, the saddle-point nature of the problem makes the choice of control volumes critical. The most natural choice, with coinciding velocity and pressure nodes, can generate spurious pressure modes. The reason is similar to what was previously noticed for Galerkin finite element approximations: discrete spaces which implicitly underly the choice of control volumes must satisfy a compatibility condition if we want the problem to be well-posed.

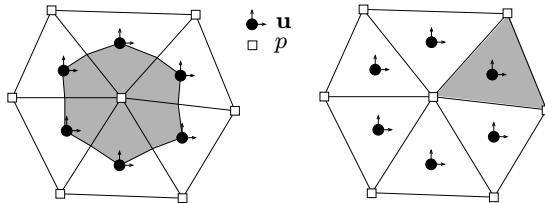


Fig. 15.13. A staggered finite volume grid for velocity and pressure. At the left-hand side we sketch the control volumes for the continuity equation, at the right-hand side the ones used for momentum equations

For this reason, it is commonplace to adopt different control volumes, and henceforth nodes, for velocity and pressure. An example is illustrated in Fig. 15.13, where we display a possible choice of nodes for the velocity components and of the ones for pressure (on the staggered grid), as well as the corresponding control volumes. The control volumes corresponding to the velocity are used for the discretization of momentum equations, while the pressure ones are used for the continuity equation. We recall that the latter does not contain the temporal derivative term.

Alternatively, we can adopt stabilization techniques similar to the ones seen in Sec. 15.5, that allow to place the velocity and pressure nodes on the same grid. The interested reader can consult the monographs [FP02], [Kro97], [Pat80] and [VM96] for further details.

In Fig. 15.14 we display the vorticity field of an incompressible flow around 5 cylinders at two time instants (this is the same problem described in example 15.1, see Fig. 15.10) with a Reynolds number of 200. In this case the Navier-Stokes equations are solved by a cell-centered finite volume discretization. The computational grid used here features 103932 elements and a time step $\Delta t = 0.001$.

Let us also report the simulation of the hydrodynamic flow around an America's Cup sailing boat in upwind regime, targeted at studying the efficiency of its appendages (bulb, keel and winglets) (see Fig. 15.15, left). The computational grid used

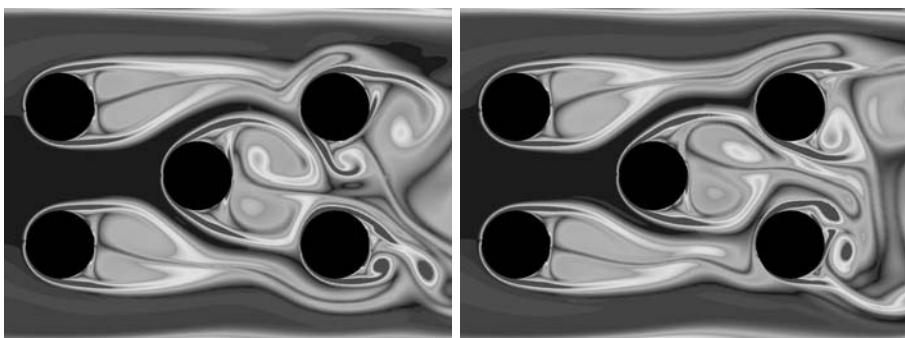


Fig. 15.14. Vorticity field of an incompressible flow around 5 cylinders at time instants $t = 100$ (left) and $t = 102$ (right), $Re = 200$

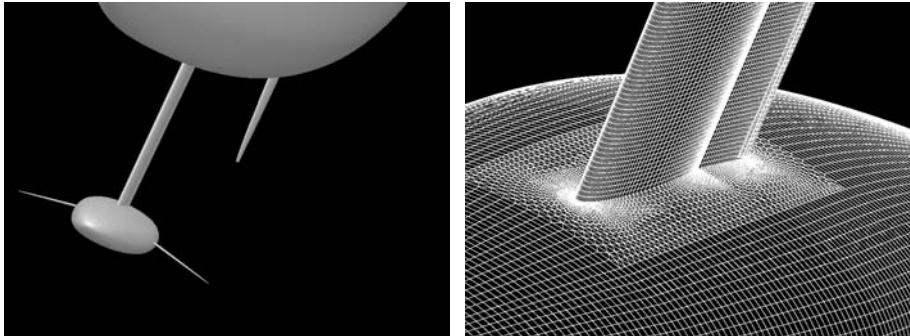


Fig. 15.15. Geometry of the hull and appendages (left) and detail of the surface grid at the bulb-keel intersection (right)

in this case is *hybrid*, with surface elements of triangular and quadrangular shape, and volume elements of tetrahedral, hexahedral, prismatic and pyramidal shape (see Fig. 15.15, right).

The mathematical model is the one illustrated in Sec. 15.9 for free-surface fluids. The Navier-Stokes equations, however, are coupled with a $k - \epsilon$ turbulence model [MP94], through an approach of type RANS (*Reynolds Averaged Navier-Stokes*). The problem's unknowns are the values of the variables (velocity, pressure and turbulent quantities) at the center of control volumes, which in this case correspond to the volume elements of the grid.

The time-dependent Navier-Stokes equations are advanced in time using a fractional-step scheme, as described in Sec. 15.7.3. As previously pointed out, the choice of placing velocity and pressure at the same points makes it necessary to adopt a suitable stabilization of the equations [RC83]. For the computation of the free surface, we have used both the *volume-of-fluid* method and the one based on the *level set* technique, described in Sec. 15.10.2, the latter being more costly from a computational viewpoint but less dissipative.

Simulations of this kind can require grids with very many elements, in the cases where one wants to reproduce complex fluid dynamics phenomena such as the turbulent flow around complex geometries, or the presence of regions of flow separation. The grid used in this case is composed of 5 million cells and yields to an algebraic system with more than 30 million unknowns. Problems of this size are generally solved by resorting to parallel computation techniques based on domain decomposition methods (see Chap. 17) in order to distribute the computation over several processors.

The analysis of pressure distributions and of wall shear stresses, as well as the visualization of the 3D flow through streamlines (see Fig. 15.16 and 15.17) are indeed very useful in the phase of the hydrodynamic project aiming at the optimization of the boat's performances (see e.g. [PQ05], [PQ07], [DPQ08]).

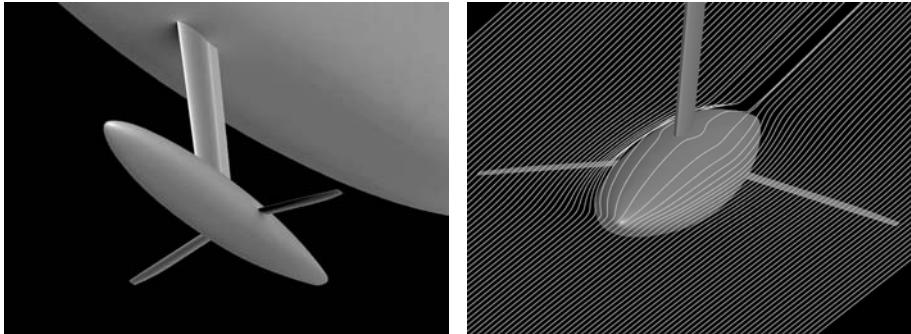


Fig. 15.16. Surface pressure distribution (left) and streamlines around the hull appendages (right)

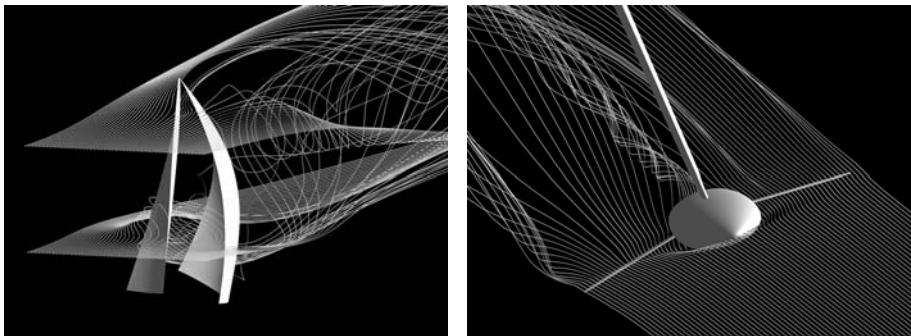


Fig. 15.17. Current lines around the sails during downwind navigation (left) and streamlines around the hull appendages (right)

15.12 Exercises

1. Prove that condition (15.53) is equivalent to the *inf-sup* condition (15.20).
[*Solution:* note that condition (15.53) is violated iff $\exists \mathbf{p}^* \neq \mathbf{0}$ with $\mathbf{p}^* \in \mathbb{R}^M$ such that $\mathbf{B}^T \mathbf{p}^* = \mathbf{0}$ or, equivalently, that $\exists p_h^* \in \mathbb{Q}_h$ such that $b(\varphi_n, p_h^*) = 0 \quad \forall n = 1, \dots, N$. This is equivalent to $b(\mathbf{v}_h, p_h^*) = 0 \quad \forall \mathbf{v}_h \in V_h$, which in turn is equivalent to violate (15.20).]
2. Prove that a necessary condition in order that (15.53) be satisfied is that $N \geq M$.

[*Solution:* we have $N = \text{rank}(B) + \dim(\ker B)$, while $M = \text{rank}(B^T) + \dim(\ker B^T) = \text{rank}(B^T) = \text{rank}(B)$. Consequently, we have $N - M = \dim(\ker B) \geq 0$, thus the condition $N \geq M$ is necessary for the solution to be unique.]

3. Show that the finite element couple $\mathbb{P}_1 - \mathbb{P}_0$ for velocity and pressure does not satisfy the *inf-sup* condition.

[*Solution:* we restrict ourselves to a two-dimensional Dirichlet problem, and consider a simple uniform triangulation made of $2n^2$ triangles, $n \geq 2$, like the one displayed in Fig. 15.18, left. This triangulation carries $M = 2n^2 - 1$ degrees of freedom for the pressure (one value for every triangle but one, as our pressure we must have null average), $N = 2(n - 1)^2$ for the velocity field (which correspond to the values of two components at all of the internal vertices). Thus the necessary condition $N \geq M$ proven in Exercise 2 is not fulfilled in the current case.]

4. Show that on a grid made by rectangles, the finite element couple $\mathbb{Q}_1 - \mathbb{Q}_0$ of bilinear polynomials for the velocity components and constant pressure on each rectangle does not satisfy the *inf-sup* condition.

[*Solution:* consider a square computational domain and a uniform cartesian grid made of $n \times n$ squares as in Fig. 15.18 right). There are $(n - 1)^2$ internal nodes carrying $N = 2(n - 1)^2$ degrees of freedom for the velocity and $M = n^2 - 1$ for the pressure. The necessary condition is therefore satisfied as far as $n \geq 3$. We therefore proceed to a direct verification that the *inf-sup* condition does not hold. Let h be the uniform size of the element edges and denote by $q_{i \pm 1/2, j \pm 1/2}$ the value at the midpoint $(x_{i \pm 1/2}, y_{j \pm 1/2}) = (x_i \pm h/2, y_j \pm h/2)$ of a given function q . Let K_{ij} be the ij -th square of the grid. A simple calculation shows that

$$\begin{aligned} \int_{\Omega} q_h \operatorname{div} \mathbf{u}_h \, d\Omega &= \frac{h}{2} \sum_{i,j=1}^{n-1} u_{ij} (q_{i-1/2, j-1/2} + q_{i-1/2, j+1/2} \\ &\quad - q_{i+1/2, j-1/2} - q_{i+1/2, j+1/2}) \\ &\quad + v_{ij} (q_{i-1/2, j-1/2} - q_{i-1/2, j+1/2} + q_{i+1/2, j-1/2} - q_{i+1/2, j+1/2}). \end{aligned}$$

Clearly, any element-wise constant function p^* whose value is 1 on the black elements and -1 on the white ones of Fig. 15.18, right, is a spurious pressure.]

5. Consider the steady Stokes problem with non-homogeneous Dirichlet boundary conditions

$$\begin{cases} -\nu \Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega \subset \mathbb{R}^2, \\ \operatorname{div} \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma = \partial\Omega, \end{cases}$$

where \mathbf{g} is a given vector function. Show that $\int_{\Gamma} \mathbf{g} \cdot \mathbf{n} = 0$ is a necessary condition for the existence of a weak solution. Show that the right hand side of the weak form of the momentum equation identifies an element of V' , the dual of the space $V = [\mathbf{H}_0^1(\Omega)]^d$.

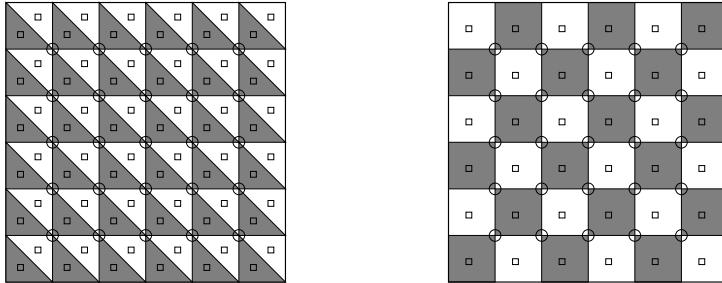


Fig. 15.18. Uniform grid for a finite element discretization using $\mathbb{P}_1 - \mathbb{P}_0$ (left) and $\mathbb{Q}_1 - \mathbb{Q}_0$ (right) finite elements with spurious pressure modes

6. Repeat the previous exercise for the non-homogeneous Navier-Stokes problem

$$\begin{cases} (\mathbf{u} \cdot \nabla) \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega \subset \mathbb{R}^2, \\ \operatorname{div} \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma = \partial\Omega. \end{cases}$$

7. Prove the a priori estimate (15.57).

[*Solution:* choose $\mathbf{v}_h = \mathbf{u}_h$ and $q_h = p_h$ as test functions in (15.55). Then apply Cauchy-Schwarz inequality, Young inequality and Poincaré inequality to bound the right hand side.]

16

Optimal control of partial differential equations

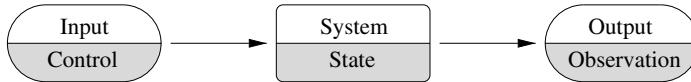
In this chapter we will introduce the basic concepts of optimal control for linear elliptic partial differential equations. At first we present the classical theory in functional spaces “à la J.L.Lions”, see [Lio71] and [Lio72]; then we will address the methodology based on the use of the Lagrangian functional (see, e.g., [Mau81], [BKR00] and [Jam88]). Finally, we will show two different numerical approaches for control problems, based on the Galerkin finite element method.

This is intended to be an elementary introduction to this fascinating and complex subject. The interested reader is advised to consult more specialized monographs such as, e.g., [Lio71], [AWB71], [ATF87], [Ago03], [BKR00], [Gun03b], [Jam88], [APV98], [MP01], [FCZ04], [DZ06], [Zua05]. For the basic concepts of Functional Analysis here used, see Chap. 2 and also [Ada75], [BG87], [Bre86], [Rud91], [Sal08] and [TL58].

16.1 Definition of optimal control problems

In abstract terms, a control problem can be expressed by the paradigm illustrated in Fig. 16.1. There is a system expressed by a *state* problem that can be either an algebraic problem, an initial-value problem for ordinary differential equations, or a boundary-value problem for partial differential equations. Its solution, that will generically be denoted by y , depends on a variable u representing the *control* that can be exerted on the system. The goal of a control problem is to find the control u in such a way that a suitable output variable, denoted by z and called observed variable (which is a function of u through y), could take a desired "value" z_d , the so-called *observation*, or target.

The problem is said to be *controllable* if a control u exists such that the observed variable z matches *exactly* the desired value z_d . Not all systems are controllable (see, in this respect, the review paper [Zua06]): take for instance the simple case in which the state problem is the linear algebraic system $Ay = \mathbf{b}$, where A is a given $n \times n$ non singular matrix and \mathbf{b} a given vector of \mathbb{R}^n . Assume moreover that the observation be represented by one solution component, say the first one, and the control be one of

**Fig. 16.1.** The essential ingredients of a control problem

the components of the right hand side, say the last one. The question therefore reads: "Find $u \in \mathbb{R}$ s.t. the solution of the linear system $Ay = \mathbf{b} + [0, \dots, 0, u]^T$ satisfy $y_1 = y_1^*$ ", being y_1^* a given value. In general, this problem admits no solution.

For this reason it is often preferred to replace the problem of controllability by one of *optimization*: by so doing one does not pretend the output variable z to be exactly equal to the observation z_d , rather that the difference between z and z_d (in a suitable sense) be minimum. Therefore, control and optimization are two intimately related concepts, as we will see later on in this chapter.

As already noticed, we will limit to the case of systems governed by elliptic PDEs. With this aim, we start by introducing the mathematical entities that enter in the control problem.

- The *control function* u . It belongs to a functional space \mathcal{U}_{ad} , called the space of *admissible controls*. In general, $\mathcal{U}_{ad} \subseteq \mathcal{U}$, being \mathcal{U} a functional space apt to describe the role assumed by u in the given state equation. If $\mathcal{U}_{ad} = \mathcal{U}$ the control problem is *unconstrained*; otherwise, if $\mathcal{U}_{ad} \subsetneq \mathcal{U}$ the control problem is said to be *constrained*.
- The *state* of the system $y(u) \in \mathcal{V}$ (a suitable functional space), which is a function depending on the control u , that satisfies the *equation of state*

$$Ay(u) = f, \quad (16.1)$$

where $A : \mathcal{V} \mapsto \mathcal{V}'$ is a differential operator (linear or not). This problem describes a physical problem subject to suitable boundary conditions. As we will see, the control function can enter in the right hand side, on the boundary data, or in the coefficients of the differential operator.

- The *observation function*, denoted by $z(u)$, also depending on the control u through y and a suitable operator $C : \mathcal{V} \rightarrow \mathcal{Z}$,

$$z(u) = Cy(u).$$

This function belongs to the space \mathcal{Z} of the observed functions and must "approach" the observation function z_d . As a matter of fact, optimizing the system (16.1) means to find the control function u such that the function $z(u)$ be "as close as possible" to the observation function z_d . This objective will be achieved through a minimization process that we are going to describe.

- Define a *cost functional* $J(u)$, defined on the space \mathcal{U}_{ad}

$$u \in \mathcal{U}_{ad} \mapsto J(u) \in \mathbb{R} \quad \text{with } J(u) \geq 0.$$

In general, J will depend on u (also) through $z(u)$, that is $J(u) = \tilde{J}(u, z(u))$, for a suitable functional $\tilde{J} : \mathcal{U}_{ad} \times \mathcal{Z} \rightarrow \mathbb{R}$.

The optimal control problem can be formulated in either way:

- i) find $u \in \mathcal{U}_{ad}$ s.t.

$$J(u) = \inf J(v) \quad \forall v \in \mathcal{U}_{ad}; \quad (16.2)$$

- ii) find $u \in \mathcal{U}_{ad}$ s.t. the following inequality holds

$$J(u) \leq J(v) \quad \forall v \in \mathcal{U}_{ad}. \quad (16.3)$$

The function u that satisfies (16.2) (or (16.3)) is called *optimal control* of system (16.1).

Before analyzing the existence and uniqueness properties of the control problem and characterizing the condition of optimality, let us consider a simple finite dimensional example.

16.2 A control problem for linear systems

Let A be a $n \times n$ non-singular matrix and B a $n \times q$ matrix. Moreover, let \mathbf{f} be a vector of \mathbb{R}^n , \mathbf{u} a vector of \mathbb{R}^q representing the control. The vector $\mathbf{y} = \mathbf{y}(u) \in \mathbb{R}^n$ which represents the state satisfies the following linear system

$$A\mathbf{y} = \mathbf{f} + B\mathbf{u}. \quad (16.4)$$

We look for a control \mathbf{u} s.t. the following linear functional to be minimized

$$J(\mathbf{u}) = \|\mathbf{z}(\mathbf{u}) - \mathbf{z}_d\|_{\mathbb{R}^m}^2 + \|\mathbf{u}\|_N^2. \quad (16.5)$$

In this equation, \mathbf{z}_d is a given vector (the target) of \mathbb{R}^m , $\mathbf{z}(\mathbf{u}) = C\mathbf{y}(\mathbf{u})$ is the vector to be observed, where C is a $m \times n$ matrix, $\|\mathbf{u}\|_N = (N\mathbf{u}, \mathbf{u})_{\mathbb{R}^q}^{1/2}$ is the N norm of \mathbf{u} , N being a given symmetric and positive definite matrix of dimension $q \times q$.

Upon interpreting the term $\|\mathbf{u}\|_N^2$ as the energy associated to the control, the problem is therefore: how to choose the control so that the observation $\mathbf{z}(\mathbf{u})$ be close to the target \mathbf{z}_d and its energy be small.

Note that

$$J(\mathbf{u}) = (CA^{-1}(\mathbf{f} + B\mathbf{u}) - \mathbf{z}_d, CA^{-1}(\mathbf{f} + B\mathbf{u}) - \mathbf{z}_d)_{\mathbb{R}^m} + (N\mathbf{u}, \mathbf{u})_{\mathbb{R}^q}. \quad (16.6)$$

The cost functional $J(\mathbf{u})$ is therefore a quadratic function of \mathbf{u} which has on \mathbb{R}^q a global minimum. The latter is characterized by the condition

$$J'(\mathbf{u})\mathbf{h} = 0 \quad \forall \mathbf{h} \in \mathbb{R}^q \quad (16.7)$$

where $J'(\mathbf{u})\mathbf{h}$ is the directional derivative along the direction \mathbf{h} computed at the "point" \mathbf{u} , that is (see Definition 2.6 of Chap. 2)

$$J'(\mathbf{u})\mathbf{h} = \lim_{t \rightarrow 0} \frac{J(\mathbf{u} + t\mathbf{h}) - J(\mathbf{u})}{t}.$$

Since

$$A\mathbf{y}'(\mathbf{u})\mathbf{h} = B\mathbf{h} \quad \text{and} \quad \mathbf{z}'(\mathbf{u})\mathbf{h} = C\mathbf{y}'(\mathbf{u})\mathbf{h}$$

for all \mathbf{u} and \mathbf{h} , from (16.6) we obtain

$$\begin{aligned} J'(\mathbf{u})\mathbf{h} &= 2[(\mathbf{z}'(\mathbf{u})\mathbf{h}, \mathbf{z}(\mathbf{u}) - \mathbf{z}_d)_{\mathbb{R}^m} + (N\mathbf{u}, \mathbf{h})_{\mathbb{R}^q}] \\ &= 2[(CA^{-1}B\mathbf{h}, C\mathbf{y}(\mathbf{u}) - \mathbf{z}_d)_{\mathbb{R}^m} + (N\mathbf{u}, \mathbf{h})_{\mathbb{R}^q}]. \end{aligned} \quad (16.8)$$

Let us introduce the solution $\mathbf{p} = \mathbf{p}(\mathbf{u}) \in \mathbb{R}^n$ of the following system, that is called the *adjoint state* of (16.4)

$$A^T \mathbf{p}(\mathbf{u}) = C^T(C\mathbf{y}(\mathbf{u}) - \mathbf{z}_d). \quad (16.9)$$

From (16.8) we deduce

$$J'(\mathbf{u})\mathbf{h} = 2[(B\mathbf{h}, \mathbf{p}(\mathbf{u}))_{\mathbb{R}^n} + (N\mathbf{u}, \mathbf{h})_{\mathbb{R}^q}],$$

that is

$$J'(\mathbf{u}) = 2[B^T \mathbf{p}(\mathbf{u}) + N\mathbf{u}]. \quad (16.10)$$

Since J attains its minimum at that point \mathbf{u} for which $J'(\mathbf{u}) = \mathbf{0}$, we can conclude that the three-field system

$$\begin{cases} A\mathbf{y} = \mathbf{f} + B\mathbf{u}, \\ A^T \mathbf{p} = C^T(C\mathbf{y} - \mathbf{z}_d), \\ B^T \mathbf{p} + N\mathbf{u} = \mathbf{0}, \end{cases} \quad (16.11)$$

admits a unique solution $(\mathbf{u}, \mathbf{y}, \mathbf{p}) \in \mathbb{R}^q \times \mathbb{R}^n \times \mathbb{R}^n$, and that \mathbf{u} is the unique optimal control of the original system.

In the next section we will introduce several examples of optimal control problems for the Laplace equation.

16.3 Some examples of optimal control problems for the Laplace equation

Consider for simplicity the case where the elliptic operator A is the Laplacian. We define two different families of optimal control problems: the distributed control and the boundary control.

- *Distributed control.* Let us introduce the *state* problem

$$\begin{cases} -\Delta y = f + u & \text{in } \Omega, \\ y = 0 & \text{on } \Gamma = \partial\Omega, \end{cases} \quad (16.12)$$

where Ω is a domain in \mathbb{R}^n , $y \in \mathcal{V} = H_0^1(\Omega)$ is the state variable, $f \in L^2(\Omega)$ is a given source term, and $u \in \mathcal{U}_{ad} = L^2(\Omega)$ is the control function. We can consider two different kind of cost functional:

- *on the domain*, for instance

$$J(u) = \int_{\Omega} (y(u) - z_d)^2 d\Omega, \quad (16.13)$$

- *on the boundary*, for instance (provided $y(u)$ is sufficiently regular)

$$J(u) = \int_{\Gamma} \left(\frac{\partial y(u)}{\partial n} - z_{d_\Gamma} \right)^2 d\gamma.$$

The functions z_d and z_{d_Γ} are two prescribed observation (or target) functions.

- *Boundary control*. Consider now the following *state* problem

$$\begin{cases} -\Delta y = f & \text{in } \Omega, \\ y = u & \text{on } \Gamma_D, \\ \frac{\partial y}{\partial n} = 0 & \text{on } \Gamma_N, \end{cases} \quad (16.14)$$

with $\Gamma_D \cup \Gamma_N = \partial\Omega$ and $\overset{\circ}{\Gamma}_D \cap \overset{\circ}{\Gamma}_N = \emptyset$. The *control* $u \in H^{\frac{1}{2}}(\Gamma_D)$ is defined on the Dirichlet boundary. Two different kind of cost functional can be considered:

- *on the domain*, as in (16.13);
- *on the boundary*, for instance

$$J(u) = \int_{\Gamma_N} (y(u) - z_{d_{\Gamma_N}})^2 d\gamma.$$

Also $z_{d_{\Gamma_N}}$ represents a given observation function.

16.4 On the minimization of linear functionals

In this section we recall some results about the existence and uniqueness of extrema of linear functionals, with special interest for those associated to control problems addressed in this chapter. For more results see, e.g., [Lio71], [BG87], [Bre86] and [TL58].

We consider a Hilbert space \mathcal{U} , endowed with a scalar product (\cdot, \cdot) , and a bilinear form π

$$u, v \mapsto \pi(u, v) \quad \forall u, v \in \mathcal{U}, \quad (16.15)$$

that we assume to be symmetric, continuous and coercive. The norm induced on \mathcal{U} by the scalar product will be denoted by $\|w\| = \sqrt{(w, w)}$. Let

$$v \mapsto F(v) \quad \forall v \in \mathcal{U}, \quad (16.16)$$

be a linear and bounded functional on \mathcal{U} . Finally, let \mathcal{U}_{ad} be a closed subspace of \mathcal{U} of the admissible control functions, and consider the following cost functional

$$J(v) = \pi(v, v) - 2F(v) \quad \forall v \in \mathcal{U}_{ad}. \quad (16.17)$$

The following result holds:

Theorem 16.1 *Under the previous assumptions, there exists a unique $u \in \mathcal{U}_{ad}$ s.t.*

$$J(u) = \inf J(v) \quad \forall v \in \mathcal{U}_{ad}, \quad (16.18)$$

where $J(v)$ is defined in (16.17); u is called optimal control.

Moreover:

- (i) The function $u \in \mathcal{U}_{ad}$ satisfies the variational inequality

$$\pi(u, v - u) \geq F(v - u) \quad \forall v \in \mathcal{U}_{ad}. \quad (16.19)$$

- (ii) In the case in which $\mathcal{U}_{ad} = \mathcal{U}$ (that is we consider a non-constrained optimization problem), owing to the Lax–Milgram Lemma 3.1, u satisfies the following Euler equation associated to (16.18):

$$\pi(u, w) = F(w) \quad \forall w \in \mathcal{U}. \quad (16.20)$$

- (iii) In the case in which \mathcal{U}_{ad} is a closed convex cone with vertex at the origin 0^a , u satisfies

$$\pi(u, v) \geq F(v) \quad \forall v \in \mathcal{U}_{ad} \quad \text{and} \quad \pi(u, u) = F(u). \quad (16.21)$$

- (iv) In the case in which the map $v \mapsto F(v)$ is strictly convex and differentiable, J is not necessarily quadratic, yet it satisfies the property: $J(v) \rightarrow \infty$ when $\|v\|_{\mathcal{U}} \rightarrow \infty \forall v \in \mathcal{U}_{ad}$, then the unique function $u \in \mathcal{U}_{ad}$ which satisfies condition (16.18) is characterized by the variational inequality

$$J'(u)(v - u) \geq 0 \quad \forall v \in \mathcal{U}_{ad} \quad (16.22)$$

or, equivalently,

$$J'(v)(v - u) \geq 0 \quad \forall u \in \mathcal{U}_{ad}. \quad (16.23)$$

(The symbol J' denotes the Gâteaux derivative of J , see Definition 2.5 of Chap. 2.)

^a A linear metric space W is a *closed convex cone with vertex in the origin 0* if: (1) $0 \in W$, (2) $\forall x \in W \Rightarrow kx \in W \ \forall k \geq 0$, (3) $\forall x, y \in W \Rightarrow x + y \in W$, (4) W is closed.

Proof. For a complete proof see, e.g., [Lio71, Ch.1,Thm 1.1]. Here we start proving (16.19). If u minimizes (16.18), then for all $v \in \mathcal{U}_{ad}$ and every $0 < \vartheta < 1$, $J(u) \leq J((1 - \vartheta)u + \vartheta v)$, thus $\frac{1}{\vartheta}[J(u + \vartheta(v - u)) - J(u)] \geq 0$. This inequality still holds in the limit when $\vartheta \rightarrow 0$ (provided this limit exists), whence

$$J'(u)(v - u) \geq 0 \quad \forall v \in \mathcal{U}_{ad}. \quad (16.24)$$

The inequality (16.19) follows by recalling the definition (16.17) of J .

The viceversa holds as well (whence (16.18) and (16.19) are in fact equivalent). Indeed, should u satisfy (16.19), and thus (16.24), thanks to the convexity of the map $v \mapsto J(v)$ for every $0 < \vartheta < 1$ one has

$$J(v) - J(w) \geq \frac{1}{\vartheta} [J((1 - \vartheta)w + v) - J(w)] \quad \forall v, w \in \mathcal{U}_{ad}.$$

Taking the limit as $\vartheta \rightarrow 0$ we obtain

$$J(v) - J(w) \geq J'(w)(v - w).$$

Taking $w = u$ and using (16.24), we obtain that $J(v) \geq J(u)$, that is (16.18).

To prove (16.20) it is sufficient to choose $v = u \pm w \in \mathcal{U}$ in (16.19).

Let us now prove (16.21). The first inequality can be obtained by replacing v with $v + u$ in (16.19). Setting now $v = 0$ in (16.19) we can obtain $\pi(u, u) \leq F(u)$. By combining the latter inequality with the first inequality in (16.21) we obtain the second equation in (16.21). The viceversa (that is, (16.21) yields (16.19)) is obvious.

For the proof of (16.22) and (16.23), see [Lio71, Ch. 1, Thm 1.4]. \diamond

Remark 16.1 If $J(v)$ is differentiable w.r.t. v , $\forall v \in \mathcal{U}$, then for every minimizing function $u \in \mathcal{U}$ of J (provided it does exist) it is $J'(u) = 0$. Moreover, under the assumptions of Theorem 16.1 (step (iv)), there exists at least one minimizing element $u \in \mathcal{U}$. \bullet

We can summarize by saying that the solution $u \in \mathcal{U}_{ad}$ of the minimization problem satisfies the following (equivalent) conditions:

- i) $J(u) = \inf J(v) \quad \forall v \in \mathcal{U}_{ad},$
- ii) $J(u) \leq J(v) \quad \forall v \in \mathcal{U}_{ad},$
- iii) $J'(u)(v - u) \geq 0 \quad \forall v \in \mathcal{U}_{ad},$
- iv) $J'(v)(v - u) \geq 0 \quad \forall u \in \mathcal{U}_{ad}.$

Before closing this section, consider the abstract problem to find $u \in \mathcal{U}_{ad}$ satisfying the variational inequality (16.19) (when $\pi(\cdot, \cdot)$ is not symmetric, this problem does not correspond to a problem of minimization in the calculus of variations).

Theorem 16.2 *If there exists a constant $c > 0$ s.t.*

$$\pi(v_1 - v_2, v_1 - v_2) \geq c \|v_1 - v_2\|^2 \quad \forall v_1, v_2 \in \mathcal{U}_{ad}, \quad (16.25)$$

then there exists a unique function $u \in \mathcal{U}_{ad}$ which satisfies (16.19).

For the proof, see [Lio71, Ch. 1, Thm 2.1].

16.5 The theory of optimal control for elliptic problems

In this section we illustrate some existence and uniqueness results for the solution of a control problem governed by a linear elliptic equation (the equation of state). For the sake of simplicity we confine ourselves to the case of *distributed control* (see Sec. 16.3); however, similar results hold for boundary control problems as well, see [Lio71].

Let \mathcal{V} and \mathcal{H} be two Hilbert spaces, \mathcal{V}' the dual of \mathcal{V} and \mathcal{H}' that of \mathcal{H} , and assume that \mathcal{V} be dense in \mathcal{H} with continuous injection. We recall that in this case, property (2.10) of Chap. 2 holds. In addition, denote by (\cdot, \cdot) the scalar product of \mathcal{H} and suppose that $a(u, v)$ be a bilinear, continuous and coercive form on \mathcal{V} (but not necessarily symmetric). Under these assumptions, the Lax–Milgram lemma guarantees that there exists a unique solution $y \in \mathcal{V}$ of problem

$$a(y, \varphi) = (f, \varphi) \quad \forall \varphi \in \mathcal{V}. \quad (16.26)$$

By introducing the operator A associated with the bilinear form $a(\cdot, \cdot)$ (see (3.39))

$$A \in \mathcal{L}(\mathcal{V}, \mathcal{V}') : \nu' \langle A\varphi, \psi \rangle_{\mathcal{V}} = a(\varphi, \psi) \quad \forall \varphi, \psi \in \mathcal{V},$$

problem (16.26) becomes (in operator form)

$$Ay = f \quad \text{in } \mathcal{V}'. \quad (16.27)$$

This is the equation of state of a physical system governed by the operator A , and will be supplemented by a distributed control term.

For that, let \mathcal{U} be a Hilbert space of control functions, and B an operator belonging to the space $\mathcal{L}(\mathcal{U}, \mathcal{V}')$. For every control function u the *equation of state* of the system is

$$Ay(u) = f + Bu \quad \text{in } \mathcal{V}', \quad (16.28)$$

or, in weak form,

$$y(u) \in \mathcal{V} : a(y(u), \varphi) = (f, \varphi) + b(u, \varphi) \quad \forall \varphi \in \mathcal{V}, \quad (16.29)$$

where $b(\cdot, \cdot)$ is the bilinear form associated with the operator B , that is

$$b(u, \varphi) = \nu' \langle Bu, \varphi \rangle_{\mathcal{V}} \quad \forall u \in \mathcal{U}, \quad \forall \varphi \in \mathcal{V}. \quad (16.30)$$

Let us denote with \mathcal{Z} the Hilbert space of observation functions, and introduce the *equation of observation*

$$z(u) = Cy(u), \quad (16.31)$$

for a suitable operator $C \in \mathcal{L}(\mathcal{V}, \mathcal{Z})$. At last, let us define the *cost functional*

$$J(y(u), u) = \|Cy(u) - z_d\|_{\mathcal{Z}}^2 + (Nu, u)_{\mathcal{U}}, \quad (16.32)$$

that we will indicate with the shorthand notation $J(u)$. Here $N \in \mathcal{L}(\mathcal{U}, \mathcal{U})$ is a symmetric positive definite form s.t.

$$(Nu, u)_{\mathcal{U}} \geq \nu \|u\|_{\mathcal{U}}^2 \quad \forall u \in \mathcal{U}, \quad (16.33)$$

where $\nu > 0$ and $z_d \in \mathcal{Z}$ is the desired (target) observation.

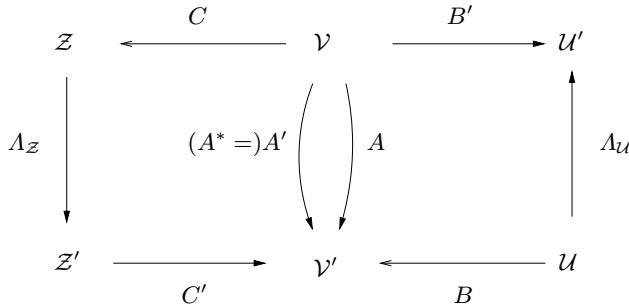


Fig. 16.2. Functional spaces and operators involved in the control problem statement

The optimal control problem consists of finding $u \in \mathcal{U}_{ad} \subseteq \mathcal{U}$ such that

$$J(u) = \inf J(v) \quad \forall v \in \mathcal{U}_{ad}. \quad (16.34)$$

An overview of spaces and operators involved in the above definition of optimal control problem is depicted in Fig. 16.2.

Remark 16.2 When minimizing (16.32), in fact one minimizes a balance between two terms. The former enforces that the observation $z(u)$ be closed to the desired value (the target) z_d . The latter penalizes the use of a control u that is “too expensive”. Heuristically speaking, we are trying to lead $z(u)$ towards z_d by a reduced effort. Note that this theory applies also if the form N is null, however in this case one can only prove the existence of an optimal control, but not its uniqueness. •

In order to apply the abstract theoretical results stated in Sec. 16.4, by noticing that the map $u \mapsto y(u)$ from \mathcal{U} in \mathcal{V} is affine, we can rewrite (16.32) as follows

$$J(u) = \|C[y(u) - y(0)] + Cy(0) - z_d\|_{\mathcal{Z}}^2 + (Nu, u)_{\mathcal{U}}. \quad (16.35)$$

Let us now define the bilinear form π that is continuous in \mathcal{U} and the functional F as follows, for all $u, v \in \mathcal{U}$:

$$\pi(u, v) = (C[y(u) - y(0)], C[y(v) - y(0)])_{\mathcal{Z}} + (Nu, v)_{\mathcal{U}},$$

$$F(v) = (z_d - Cy(0), C[y(v) - y(0)])_{\mathcal{Z}}.$$

Then

$$J(v) = \pi(v, v) - 2F(v) + \|z_d - Cy(0)\|_{\mathcal{Z}}^2.$$

Since $\|C[y(v) - y(0)]\|_{\mathcal{Z}}^2 \geq 0$, owing to (16.33) we obtain

$$\pi(v, v) \geq \nu \|v\|_{\mathcal{U}}^2 \quad \forall v \in \mathcal{U}.$$

By so doing we have casted the control problem in the general formulation addressed in Sec. 16.4. Then Theorem 16.1 guarantees existence and uniqueness of the control function $u \in \mathcal{U}_{ad}$.

At this stage we would like to study the structure of the equations useful to the *solution* of the control problem. Thanks to Theorem 16.1, and since A is an isomorphism between \mathcal{V} and \mathcal{V}' (see Definition 2.4), we have

$$y(u) = A^{-1}(f + Bu),$$

whence $y'(u)\psi = A^{-1}B\psi$ and therefore

$$y'(u)(v - u) = A^{-1}B(v - u) = y(v) - y(u).$$

Since the optimal control must satisfy (16.22), dividing by 2 the inequality (16.22) we obtain, thanks to (16.35)

$$(Cy(u) - z_d, C[y(v) - y(u)])_{\mathcal{Z}} + (Nu, v - u)_{\mathcal{U}} \geq 0 \quad \forall v \in \mathcal{U}_{ad}. \quad (16.36)$$

Let now $C' \in \mathcal{L}(\mathcal{Z}', \mathcal{V}')$ be the adjoint of the operator $C \in \mathcal{L}(\mathcal{V}, \mathcal{Z})$ (see (2.19)). Then

$$z \langle Cy, v \rangle_{\mathcal{Z}'} = v \langle y, C'v \rangle_{\mathcal{V}'} \quad \forall y \in \mathcal{V}, \forall v \in \mathcal{Z}',$$

hence (16.36) becomes

$$v' \langle C' \Lambda_{\mathcal{Z}}(Cy(u) - z_d), y(v) - y(u) \rangle_{\mathcal{V}} + (Nu, v - u)_{\mathcal{U}} \geq 0 \quad \forall v \in \mathcal{U}_{ad}, \quad (16.37)$$

where $\Lambda_{\mathcal{Z}}$ denotes the canonical Riesz isomorphism from \mathcal{Z} in \mathcal{Z}' (see (2.5)). Let us now introduce the adjoint operator $A' \in \mathcal{L}(\mathcal{V}, \mathcal{V}')$ of A such that

$$v' \langle A'\varphi, \psi \rangle_{\mathcal{V}} = v \langle \varphi, A\psi \rangle_{\mathcal{V}'} \quad \forall \varphi, \psi \in \mathcal{V}.$$

This operator was denoted with the symbol A^* in Sec 3.6 (see the Lagrange identity (3.42)). Owing to (3.40) and (3.41) we obtain

$$v' \langle A'\varphi, \psi \rangle_{\mathcal{V}} = a(\psi, \varphi) \quad \forall \varphi, \psi \in \mathcal{V}. \quad (16.38)$$

We define *adjoint state* (or *adjoint variable*) $p(u) \in \mathcal{V}$ the solution of the *adjoint equation*

$$A'p(u) = C' \Lambda_{\mathcal{Z}}(Cy(u) - z_d), \quad (16.39)$$

with $u \in \mathcal{U}$. Thanks to (16.39), we obtain

$$\begin{aligned} & v' \langle C' \Lambda_{\mathcal{Z}}(Cy(u) - z_d), y(v) - y(u) \rangle_{\mathcal{V}} \\ &= v' \langle A'p(u), y(v) - y(u) \rangle_{\mathcal{V}} \\ &= (\text{thanks to the definition of } A') \quad v \langle p(u), A(y(v) - y(u)) \rangle_{\mathcal{V}'} \\ &= (\text{thanks to (16.28)}) \quad v \langle p(u), B(v - u) \rangle_{\mathcal{V}'} = u' \langle B'p(u), v - u \rangle_{\mathcal{U}}, \end{aligned}$$

$B' \in \mathcal{L}(\mathcal{V}, \mathcal{U}')$ is the adjoint operator of B (see (2.19)). It follows that, by the help of the Riesz canonical isomorphism $\Lambda_{\mathcal{U}}$ of \mathcal{U} in \mathcal{U}' (see again (2.5)), inequality (16.22) can be rewritten as

$$J'(u)(v - u) = (\Lambda_{\mathcal{U}}^{-1}B'p(u) + Nu, v - u)_{\mathcal{U}} \geq 0 \quad \forall v \in \mathcal{U}_{ad}. \quad (16.40)$$

In the case of unconstrained control, that is when $\mathcal{U}_{ad} = \mathcal{U}$, the latter inequality becomes in fact an equation, that is

$$B'p(u) + \Lambda_{\mathcal{U}}Nu = 0. \quad (16.41)$$

This follows by taking $v = u - (\Lambda_{\mathcal{U}}^{-1}B'p(u) + Nu)$ in (16.40). In the case where $\mathcal{V} \subset \mathcal{U}$ the previous equation implies that

$$b(v, p) + n(u, v) = 0 \quad \forall v \in \mathcal{V}.$$

The final result is reported in the following Theorem ([Lio71, Ch. 2, Thm 1.4]).

Theorem 16.3 *A necessary and sufficient condition for the existence of an optimal control $u \in \mathcal{U}_{ad}$ is that the following equations and inequalities hold (see (16.28),(16.39),(16.40)):*

$$\begin{cases} y = y(u) \in \mathcal{V}, & Ay(u) = f + Bu, \\ p = p(u) \in \mathcal{V}, & A'p(u) = C'\Lambda(Cy(u) - z_d), \\ u \in \mathcal{U}_{ad}, & (\Lambda_{\mathcal{U}}^{-1}B'p(u) + Nu, v - u)_{\mathcal{U}} \geq 0 \quad \forall v \in \mathcal{U}_{ad}, \end{cases} \quad (16.42)$$

or, in weak form:

$$\begin{cases} y = y(u) \in \mathcal{V}, & a(y(u), \varphi) = (f, \varphi) + b(u, \varphi) \quad \forall \varphi \in \mathcal{V}, \\ p = p(u) \in \mathcal{V}, & a(\psi, p(u)) = (Cy(u) - z_d, C\psi)_{\mathcal{Z}} \quad \forall \psi \in \mathcal{V}, \\ u \in \mathcal{U}_{ad}, & (\Lambda_{\mathcal{U}}^{-1}B'p(u) + Nu, v - u)_{\mathcal{U}} \geq 0 \quad \forall v \in \mathcal{U}_{ad}. \end{cases} \quad (16.43)$$

This is called the optimality system.

If N is symmetric and positive definite, then the control u is unique; on the other hand, if $N = 0$ and \mathcal{U}_{ad} is bounded, then there exists at least one solution, and the family of optimal controls forms a closed and convex subset \mathcal{X} of \mathcal{U}_{ad} .

The third condition of (16.42) can be expressed as follows

$$(\Lambda_{\mathcal{U}}^{-1}B'p(u) + Nu, u)_{\mathcal{U}} = \inf_{v \in \mathcal{U}_{ad}} (\Lambda_{\mathcal{U}}^{-1}B'p(u) + Nu, v)_{\mathcal{U}}. \quad (16.44)$$

The derivative of the cost functional w.r.t. u can be expressed in terms of the adjoint state $p(u)$ as follows

$$\frac{1}{2}J'(u) = B'p(u) + \Lambda_{\mathcal{U}}Nu. \quad (16.45)$$

Remark 16.3 Apart from the term depending on the form N , J' can be obtained from the adjoint variable p through the operator B' . This result will stand at the base of the

numerical methods which are useful to determine an approximate control function. If $\mathcal{U}_{ad} = \mathcal{U}$, then the optimal control therefore satisfies

$$Nu = -A_{\mathcal{U}}^{-1}B'p(u). \quad (16.46)$$

Thanks to the identity (2.23) we have

$$A_{\mathcal{U}}^{-1}B' = B^T A_{\mathcal{V}}, \quad (16.47)$$

where $A_{\mathcal{V}}$ is the Riesz canonical isomorphism from \mathcal{V} into \mathcal{V}' and $B^T : \mathcal{V}' \rightarrow \mathcal{U}$ is the transpose operator of B introduced in (2.21). •

16.6 Some examples of optimal control problems

In this section we introduce three examples of optimal control problems.

16.6.1 A Dirichlet problem with distributed control

Let us recover the example of the distributed control (16.12) and consider the following cost functional to be minimized

$$J(v) = \int_{\Omega} (y(v) - z_d)^2 d\Omega + (Nv, v), \quad (16.48)$$

in which, for instance, we can set $N = \nu I$, $\nu > 0$. In this case $\mathcal{V} = H_0^1(\Omega)$, $\mathcal{H} = L^2(\Omega)$, $\mathcal{U} = \mathcal{H}$ (then $(Nv, v) = (Nv, v)_{\mathcal{U}}$) therefore $A_{\mathcal{U}}$ is the identity operator. Moreover, B is the identity operator, C is the injection operator of \mathcal{V} in \mathcal{H} , $\mathcal{Z} = \mathcal{H}$ and therefore $A_{\mathcal{Z}}$ is the identity operator. Finally, $a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v d\Omega$. Owing to Theorem 16.3 we obtain, with $A = -\Delta$, the following *optimality system*:

$$\begin{cases} y(u) \in H_0^1(\Omega) & : Ay(u) = f + u \quad \text{in } \Omega, \\ p(u) \in H_0^1(\Omega) & : A'p(u) = y(u) - z_d \quad \text{in } \Omega, \\ u \in \mathcal{U}_{ad} & : \int_{\Omega} (p(u) + Nu)(v - u) d\Omega \geq 0 \quad \forall v \in \mathcal{U}_{ad}. \end{cases} \quad (16.49)$$

In the unconstrained case in which $\mathcal{U}_{ad} = \mathcal{U}$ ($= L^2(\Omega)$), the last inequality reduces to the equation

$$p(u) + Nu = 0,$$

as we can see by taking $v = u - (p(u) + Nu)$.

The first two equations of (16.49) provide a system for the two variables y and p

$$\begin{cases} Ay + N^{-1}p = f \quad \text{in } \Omega, & y = 0 \quad \text{on } \partial\Omega, \\ A'p - y = -z_d \quad \text{in } \Omega, & p = 0 \quad \text{on } \partial\Omega, \end{cases}$$

whose solution provides the optimal control $u = -N^{-1}p$.

If Ω has a smooth boundary, from the elliptic regularity property, both y and p belong to the space $H^2(\Omega)$. Since N^{-1} maps $H^2(\Omega)$ in itself, the optimal control u belongs to $H^2(\Omega)$ too.

16.6.2 A Neumann problem with distributed control

Consider now the problem

$$\begin{cases} Ay(u) = f + u & \text{in } \Omega, \\ \frac{\partial y(u)}{\partial n_A} = g & \text{on } \partial\Omega, \end{cases} \quad (16.50)$$

where A is an elliptic operator and $\frac{\partial}{\partial n_A}$ is the conormal derivative associated with A (see (3.34)). The cost functional to be minimized is the same as that introduced in (16.48). In this case $\mathcal{V} = H^1(\Omega)$, $\mathcal{H} = L^2(\Omega)$, $\mathcal{U} = \mathcal{H}$, B is the identity operator, C is the injection map of \mathcal{V} into \mathcal{H} ,

$$a(\psi, \varphi) = \nu' \langle A\psi, \varphi \rangle_{\mathcal{V}}, \quad F(\varphi) = \int_{\Omega} f\varphi \, d\Omega + \int_{\partial\Omega} g\varphi \, d\gamma,$$

for given $f \in L^2(\Omega)$ and $g \in H^{-1/2}(\partial\Omega)$.

If $A\varphi = -\Delta\varphi + \beta\varphi$, with $\beta > 0$, then

$$a(\psi, \varphi) = \int_{\Omega} \nabla\psi \cdot \nabla\varphi \, d\Omega + \int_{\Omega} \beta\psi\varphi \, d\Omega.$$

The variational formulation of the state problem (16.50) is

$$\text{find } y(u) \in H^1(\Omega) : a(y(u), \varphi) = F(\varphi) \quad \forall \varphi \in H^1(\Omega). \quad (16.51)$$

The adjoint problem is the following Neumann problem

$$\begin{cases} A'p(u) = y(u) - z_d & \text{in } \Omega, \\ \frac{\partial p(u)}{\partial n_{A'}} = 0 & \text{on } \partial\Omega. \end{cases} \quad (16.52)$$

The optimal control can be obtained by solving the system formed by (16.50), (16.52), and

$$u \in \mathcal{U}_{ad} : \int_{\Omega} (p(u) + Nu)(v - u) \, d\Omega \geq 0 \quad \forall v \in \mathcal{U}_{ad}. \quad (16.53)$$

16.6.3 A Neumann problem with boundary control

Consider now the problem

$$\begin{cases} Ay(u) = f & \text{in } \Omega, \\ \frac{\partial y(u)}{\partial n_A} = g + u & \text{on } \partial\Omega, \end{cases} \quad (16.54)$$

where the operator is the same as before and the cost functional is still that of (16.48). In this case,

$$\mathcal{V} = H^1(\Omega), \quad \mathcal{H} = L^2(\Omega), \quad \mathcal{U} = H^{-1/2}(\partial\Omega).$$

For all $u \in \mathcal{U}$, $Bu \in \mathcal{V}'$ is given by $\nu' \langle Bu, \varphi \rangle_{\mathcal{V}} = \int_{\partial\Omega} u\varphi \, d\gamma$, C is the injection map of \mathcal{V} in \mathcal{H} .

The weak formulation of (16.54) is

$$\text{find } y(u) \in \mathcal{H}^1(\Omega) : a(y(u), \varphi) = \int_{\Omega} f\varphi \, d\Omega + \int_{\partial\Omega} (g+u)\varphi \, d\gamma \quad \forall \varphi \in \mathcal{H}^1(\Omega).$$

The adjoint problem is still given by (16.52), whereas the variational inequality yielding the optimal control is the third of (16.42). The interpretation of this inequality is far from trivial. If we choose

$$(u, v)_{\mathcal{U}} = \int_{\partial\Omega} (-\Delta_{\partial\Omega})^{-1/4}u \, (-\Delta_{\partial\Omega})^{-1/4}v \, d\gamma = \int_{\partial\Omega} (-\Delta_{\partial\Omega})^{-1/2}u \, v \, d\gamma,$$

as scalar product in \mathcal{U} , where $-\Delta_{\partial\Omega}$ is the Laplace–Beltrami operator (see, e.g., [QV94]), it can be proven that the third inequality of (16.42) is equivalent to (see [Lio71, Ch. 1, Sec.2.4])

$$\int_{\partial\Omega} (p(u)|_{\partial\Omega} + (-\Delta_{\partial\Omega})^{-1/2}N u)(v - u) \, d\gamma \geq 0 \quad \forall v \in \mathcal{U}_{ad}.$$

In Tables 16.1 and 16.2 we summarize the main conclusions that were drawn for the problems just considered.

16.7 Numerical tests

In this section we present some numerical tests for the solution of 1D optimal control problems similar to those summarized in Tables 16.1 and 16.2.

For all the numerical simulations we consider the domain $\Omega = (0, 1)$, a simple diffusion-reaction operator A

$$Ay = -\mu y'' + \gamma y,$$

and the very same cost functional considered in the Tables, with a regularization coefficient $\nu = 10^{-2}$ (unless otherwise specified). We discretize both the state and adjoint

Table 16.1. Summary on Dirichlet control problems

<i>Dirichlet conditions</i>	<i>Distributed Observation</i>	<i>Boundary Observation</i>
Distributed Control	$\begin{cases} Ay = f + u & \text{in } \Omega \\ y = 0 & \text{on } \partial\Omega \end{cases}$ $J(y, u) = \int_{\Omega} (y - z_d)^2 d\Omega + \nu \int_{\Omega} u^2 d\Omega$	$\begin{cases} Ay = f + u & \text{in } \Omega \\ y = 0 & \text{on } \partial\Omega \end{cases}$ $J(y, u) = \int_{\partial\Omega} \left(\frac{\partial y}{\partial n_A} - z_d \right)^2 d\gamma + \nu \int_{\Omega} u^2 d\Omega$
	$\begin{cases} A'p = y - z_d & \text{in } \Omega \\ p = 0 & \text{on } \partial\Omega \end{cases}$ $\frac{1}{2} J'(u) = p(u) + \nu u \quad \text{in } \Omega$	$\begin{cases} A'p = 0 & \text{in } \Omega \\ p = -\left(\frac{\partial y}{\partial n_A} - z_d \right) & \text{on } \partial\Omega \end{cases}$ $\frac{1}{2} J'(u) = p(u) + \nu u \quad \text{in } \Omega$
Boundary Control	$\begin{cases} Ay = f & \text{in } \Omega \\ y = u & \text{on } \partial\Omega \end{cases}$ $J(y, u) = \int_{\Omega} (y - z_d)^2 d\Omega + \nu \int_{\partial\Omega} u^2 d\gamma$	$\begin{cases} Ay = f & \text{in } \Omega \\ y = u & \text{on } \partial\Omega \end{cases}$ $J(y, u) = \int_{\partial\Omega} \left(\frac{\partial y}{\partial n_A} - z_d \right)^2 d\gamma + \nu \int_{\partial\Omega} u^2 d\gamma$
	$\begin{cases} A'p = y - z_d & \text{in } \Omega \\ p = 0 & \text{on } \partial\Omega \end{cases}$ $\frac{1}{2} J'(u) = -\frac{\partial p}{\partial n_{A'}} + \nu u \quad \text{on } \partial\Omega$	$\begin{cases} A'p = 0 & \text{in } \Omega \\ p = -\left(\frac{\partial y}{\partial n_A} - z_d \right) & \text{on } \partial\Omega \end{cases}$ $\frac{1}{2} J'(u) = -\frac{\partial p}{\partial n_{A'}} + \nu u \quad \text{on } \partial\Omega$

problems by means of piecewise linear finite elements, with grid-size $h = 10^{-2}$; for solving the minimization problem we use the conjugate gradient method with an acceleration parameter τ^k initialized with $\tau^0 = \bar{\tau}$ and then, when necessary for the convergence, reduced by 2 at every subsequent step. This satisfies the Armijo rule (see Sec. 16.9). The tolerance for the iterative method tol is fixed equal to 10^{-3} , with the following stopping criterium $\|J'(u^k)\| < tol \|J'(u^0)\|$.

Table 16.2. Summary on Neumann control problems

<i>Neumann conditions</i>	<i>Distributed Observation</i>	<i>Boundary Observation</i>
Distributed Control	$\begin{cases} Ay = f + u & \text{in } \Omega \\ \frac{\partial y}{\partial n_A} = g & \text{on } \partial\Omega \end{cases}$	$\begin{cases} Ay = f + u & \text{in } \Omega \\ \frac{\partial y}{\partial n_A} = g & \text{on } \partial\Omega \end{cases}$
	$J(y, u) = \int_{\Omega} (y - z_d)^2 d\Omega + \nu \int_{\Omega} u^2 d\Omega$	$J(y, u) = \int_{\partial\Omega} (y - z_d)^2 d\gamma + \nu \int_{\Omega} u^2 d\Omega$
	$\begin{cases} A'p = y - z_d & \text{in } \Omega \\ \frac{\partial p}{\partial n_{A'}} = 0 & \text{on } \partial\Omega \end{cases}$	$\begin{cases} A'p = 0 & \text{in } \Omega \\ \frac{\partial p}{\partial n_{A'}} = y - z_d & \text{on } \partial\Omega \end{cases}$
	$\frac{1}{2} J'(u) = p + \nu u \quad \text{in } \Omega$	$\frac{1}{2} J'(u) = p + \nu u \quad \text{in } \Omega$
Boundary Control	$\begin{cases} Ay = f & \text{in } \Omega \\ \frac{\partial y}{\partial n_A} = g + u & \text{on } \partial\Omega \end{cases}$	$\begin{cases} Ay = f & \text{in } \Omega \\ \frac{\partial y}{\partial n_A} = g + u & \text{on } \partial\Omega \end{cases}$
	$J(y, u) = \int_{\Omega} (y - z_d)^2 d\Omega + \nu \int_{\partial\Omega} u^2 d\gamma$	$J(y, u) = \int_{\partial\Omega} (y - z_d)^2 d\Omega + \nu \int_{\partial\Omega} u^2 d\gamma$
	$\begin{cases} A'p = y - z_d & \text{in } \Omega \\ \frac{\partial p}{\partial n_{A'}} = 0 & \text{on } \partial\Omega \end{cases}$	$\begin{cases} A'p = 0 & \text{in } \Omega \\ \frac{\partial p}{\partial n_{A'}} = y - z_d & \text{on } \partial\Omega \end{cases}$
	$\frac{1}{2} J'(u) = p + \nu u \quad \text{on } \partial\Omega$	$\frac{1}{2} J'(u) = p + \nu u \quad \text{on } \partial\Omega$

Table 16.3. Case D1. Number of iterations and optimal cost functional corresponding to different values of ν

ν	it	J	ν	it	J	ν	it	J
10^{-2}	11	0.0202	10^{-3}	71	0.0047	10^{-4}	349	0.0011

- Case D1 (Table 16.1 top-left): distributed control and observation, with Dirichlet boundary conditions. We assume

$$\mu = 1, \quad \gamma = 0, \quad f = 1, \quad u^0 = 0, \quad z_d = \begin{cases} x & x \leq 0.5 \\ 1-x & x > 0.5 \end{cases}, \quad \bar{\tau} = 10.$$

The value of the cost functional for u^0 is $J^0 = 0.0396$, after 11 iterations we obtain the optimal cost functional $J = 0.0202$. In Fig. 16.3 we report the state variable for the initial and optimal control u and the desired function z_d (left), and the optimal control function (right).

As displayed in Table 16.3, the number of iterations increases as ν decreases. In the same Table, we also report, for the sake of comparison, the values of the cost functional J corresponding to the optimal value of u for different values of ν .

In Fig. 16.4 we report the optimal state (left) and the control functions (right) obtained for different values of ν .

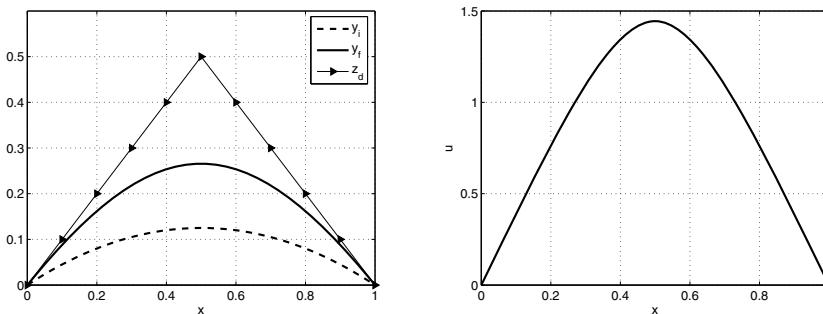


Fig. 16.3. Case D1. Initial and optimal state variables and the desired function (left); optimal control function (right)

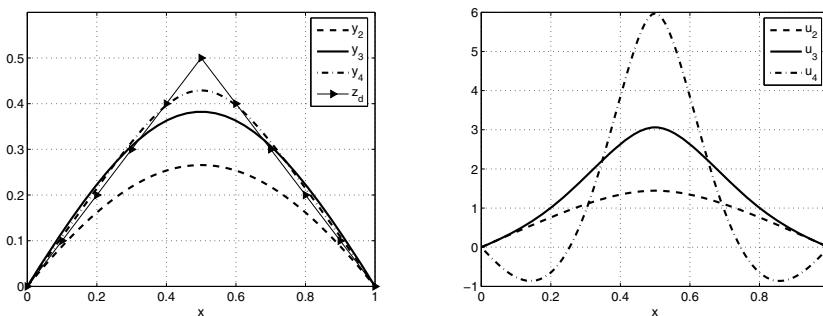


Fig. 16.4. Case D1. Optimal state variables y_2 (for $\nu = 1e-2$), y_3 (for $\nu = 1e-3$), y_4 (for $\nu = 1e-4$) and desired function z_d (left); optimal control u_2 (for $\nu = 1e-2$), u_3 (for $\nu = 1e-3$), u_4 (for $\nu = 1e-4$) (right)

- Case D2 (Table 16.1 top-right): distributed control and boundary observation, with Dirichlet boundary condition. We consider $\mu = 1$, $\gamma = 0$, $f = 1$, $u^0 = 0$, while the target function z_d is s.t. $z_d(0) = -1$ and $z_d(1) = -4$; finally, $\bar{\tau} = 0.1$. At the initial step we have $J = 12.5401$ and after 89 iterations $J = 0.04305$; we can observe how the normal derivative of the state variable is “near” the desired value $z_d \left[\mu \frac{\partial y}{\partial n}(0), \mu \frac{\partial y}{\partial n}(1) \right] = [-1.0511, -3.8695]$. In Fig. 16.5 we report the initial and optimal state (left) and the corresponding optimal control function (right).
- Case D3 (Table 16.1 bottom-left): boundary control and distributed observation, with Dirichlet boundary condition. We consider $\mu = 1$, $\gamma = 0$, $f = 1$, the initial control u^0 s.t. $u^0(0) = u^0(1) = 0$, $z_d = -1 - 3x$ and $\bar{\tau} = 0.1$. The initial functional value is $J = 0.4204$, after 55 iterations we have $J = 0.0364$ and the optimal control on the boundary is $[u(0), u(1)] = [0.6638, 0.5541]$. In Fig. 16.6, left, we report the initial and final state variables and the desired observation function.
- Case D4 (Table 16.1 bottom-right): boundary control and observation, with Dirichlet boundary condition. We assume $\mu = 1$, $\gamma = 0$, $f = 1$, the initial control u^0 s.t. $u^0(0) = u^0(1) = 0$, while the target function z_d is s.t. $z_d(0) = -1$ and $z_d(1) = -4$; finally, $\bar{\tau} = 0.1$. For $it = 0$ the cost functional value is $J = 12.5401$,

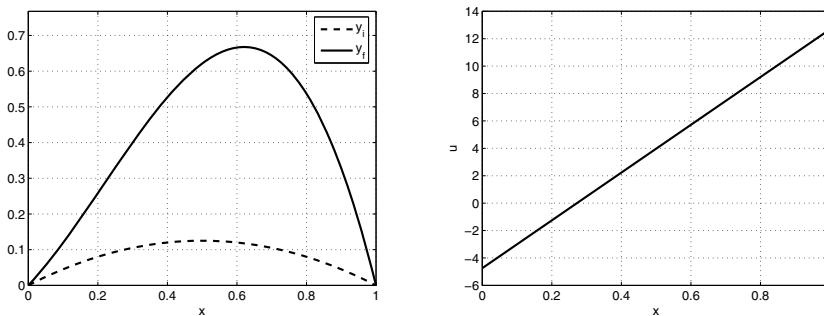


Fig. 16.5. Case D2. Initial and optimal state variables (left); optimal control variable (right)

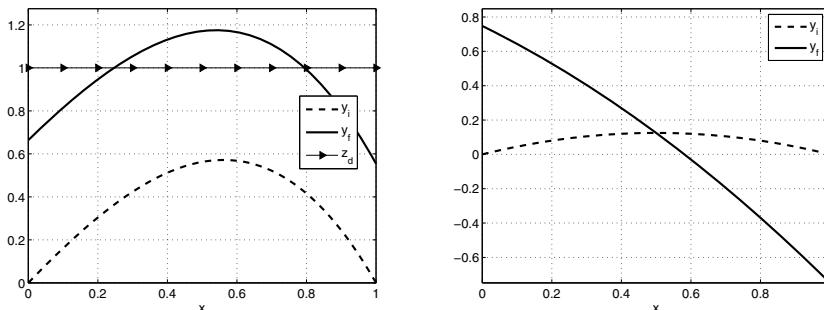


Fig. 16.6. At left, Case D3. Initial and optimal state variables and desired observation function. At right, Case D4. Initial and optimal state variables

after only 4 iterations $J = 8.0513$ and the optimal control on the boundary is $[u(0), u(1)] = [0.7481, -0.7481]$. In Fig. 16.6, right, we report the state variable.

- Case N1 (Table 16.2 top-left): distributed control and observation, with Neumann boundary condition. We consider $\mu = 1$, $\gamma = 1$, $f = 0$, $g = -1$, $u^0 = 0$, $z_d = 1$, $\bar{\tau} = 0.1$. At the initial step we have $J = 9.0053$, after 18 iterations the cost functional value is $J = 0.0944$. In Fig. 16.7 we report the state variable for the Neumann problem (left) and the final optimal control (right).
- Case N2 (Table 16.2 top-right): distributed control and boundary observation, with Neumann boundary condition. We consider $\mu = 1$, $\gamma = 1$, $f = 0$, the function g s.t. $g(0) = -1$ and $g(1) = -4$, $u^0 = 0$, while the target function z_d is s.t. $z_d(0) = z_d(1) = 1$; finally, $\bar{\tau} = 0.1$. For $it = 0$, $J = 83.1329$, after 153 iterations $J = 0.6280$, the optimal state on the boundary is $[y(0), y(1)] = [1.1613, 0.7750]$. In Fig. 16.8 we report the state variable (left) and the optimal control variable (right).
- Case N3 (Table 16.2 bottom-left): boundary control and distributed observation, with Neumann boundary condition. We assume $\mu = 1$, $\gamma = 1$, $f = 0$, the initial control u^0 s.t. $u^0(0) = u^0(1) = 0$, $z_d = -1 - 3x$, $\bar{\tau} = 0.1$. The initial cost

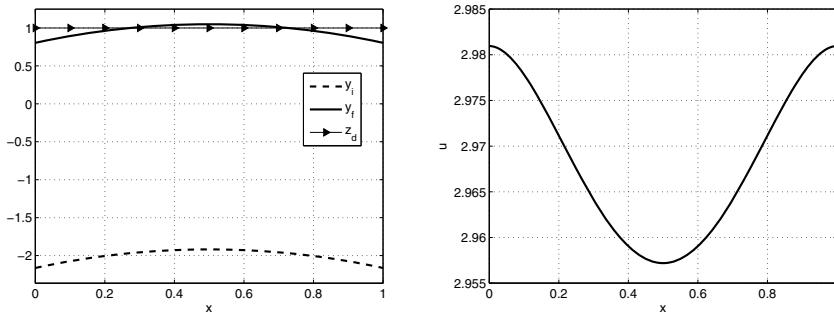


Fig. 16.7. Case N1. Initial and optimal state variables and the desired function (left); optimal control variable (right)

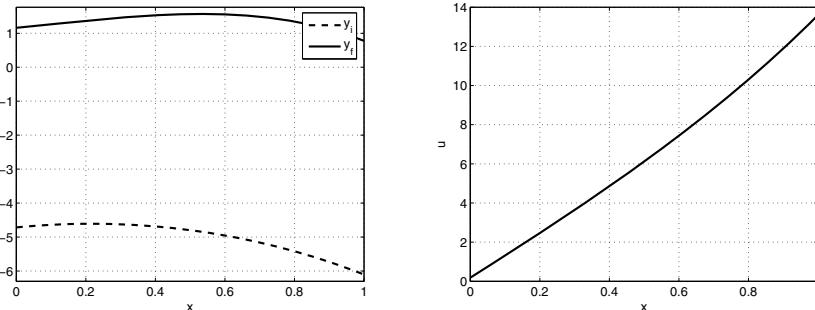


Fig. 16.8. Case N2. Initial and optimal state variables (left); optimal control variable (right)

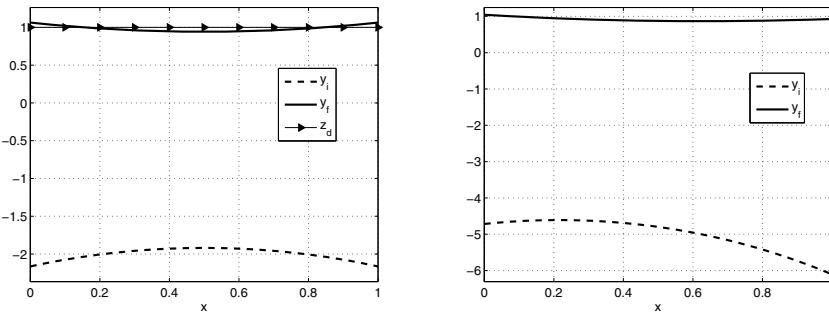


Fig. 16.9. At left, Case N3. Initial and optimal state variables and desired observation function. At right, Case N4. Initial and optimal state variables.

functional value is $J = 9.0053$, after 9 iterations we have $J = 0.0461$, and the optimal control is $[u(0), u(1)] = [1.4910, 1.4910]$. In Fig. 16.9, left, we report the state variable for the initial and optimal control u , together with the desired observation function.

- Case N4 (Table 16.2 bottom-right): boundary control and observation, with Neumann boundary condition. We consider $\mu = 1$, $\gamma = 1$, $f = 0$, the function g s.t. $g(0) = -1$ and $g(1) = -4$, the initial control u^0 s.t. $u^0(0) = u^0(1) = 0$, while the target function z_d s.t. $z_d(0) = z_d(1) = 1$; finally, $\bar{\tau} = 0.1$. At the initial step $J = 83.1329$, after 37 iterations we have $J = 0.2196$, the optimal control is $[u(0), u(1)] = [1.5817, 4.3299]$ and the observed state on the boundary is $[y(0), y(1)] = [1.0445, 0.9282]$. In Fig. 16.9, right, we report the initial and optimal state variables.

16.8 Lagrangian formulation of control problems

In this section we see another methodological approach for the solution of optimal control problems, based on Lagrange multipliers; this approach better highlights the role played by the adjoint variable.

16.8.1 Constrained optimization in \mathbb{R}^n

Let us start by a simple example, the constrained optimization in \mathbb{R}^n . Given two functions $f, g \in C^1(X)$, where X is an open set of \mathbb{R}^n , we look for the extrema of f subject to the constraint that they belong to the set

$$E_0 = \{\mathbf{x} \in \mathbb{R}^n : g(\mathbf{x}) = 0\}.$$

For simplicity of exposition we are considering the case where the constraint g is a scalar function. We give the following definitions of *regular* points and of *constrained critical* points::

Definition 16.1 A point \mathbf{x}_0 is said to be a regular point of E_0 if

$$g(\mathbf{x}_0) = 0 \quad \text{and} \quad \nabla g(\mathbf{x}_0) \neq \mathbf{0}.$$

Definition 16.2 A point $\mathbf{x}_0 \in X$ is a constrained critical point if:

- i) \mathbf{x}_0 is a regular point of E_0 ;
- ii) the directional derivative of f along the tangential direction to the constraint g is null in \mathbf{x}_0 .

On the basis of these definitions the following result holds

Theorem 16.4 A regular point \mathbf{x}_0 of E_0 is a constrained critical point if and only if there exists $\lambda_0 \in \mathbb{R}$ such that

$$\nabla f(\mathbf{x}_0) = \lambda_0 \nabla g(\mathbf{x}_0).$$

We introduce the function $\mathcal{L} : X \times \mathbb{R} \mapsto \mathbb{R}$, called the *Lagrangian* (or Lagrangian functional),

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x}).$$

From Theorem 16.4 we deduce that \mathbf{x}_0 is a constrained critical point if and only if $(\mathbf{x}_0, \lambda_0)$ is a (free) critical point for \mathcal{L} . The number λ_0 is called *Lagrange multiplier* and can be obtained, together with \mathbf{x}_0 , by solving the system

$$\nabla \mathcal{L}(\mathbf{x}, \lambda) = \mathbf{0},$$

that is

$$\begin{cases} \mathcal{L}_{\mathbf{x}} = \nabla f - \lambda \nabla g = \mathbf{0}, \\ \mathcal{L}_{\lambda} = -g = 0. \end{cases}$$

16.8.2 The solution approach based on the Lagrangian

In this section we will extend the theory developed in Sec. 16.8.1 to optimal control problems. Also in this context, the Lagrangian approach can be used as an alternative to the approach “à la Lions” (see, for example, [BKR00]).

It is actually used to integrate the techniques of grid adaptivity based on a posteriori error estimates with optimal control problems (see [BKR00] and [Ded04]).

The approach based on Lagrange multipliers is also widely used to solve shape optimization problems in which the control is represented by the shape of the computational domain. The optimal control function u is therefore a function defined on the

boundary (or on a subset of it) which describes the optimal displacement from its original position. The interested reader can consult, e.g., [Jam88], [MP01], [SZ91]. In Sec. 16.8.1 the approach based on the Lagrangian allows the determination of the extrema of a function f undergoing the constraint expressed by the function g . Instead, in control problems we look for a *function* $u \in \mathcal{U}_{ad} \subset \mathcal{U}$ satisfying the minimization problem (16.34), $y(u)$ being the solution of the state equation (16.28). As usual, A is an elliptic differential operator applied to the state variable, while B is an operator applied to the control function in the state equation. This problem can be regarded as a constrained minimization problem, provided we state a suitable correspondence between the cost functional J and the function f of Sec. 16.8.1), the equation of state and the constraint equation $g = 0$, and, finally, between the control function u and the extremum \mathbf{x} .

We assume, for the sake of simplicity, that $\mathcal{U}_{ad} = \mathcal{U}$. For a more general treatment, see, e.g., [Gun03b].

The solution of the optimal control problem can therefore be regarded as the search of (unconstrained) critical “points” of the following Lagrangian functional

$$\mathcal{L}(y, p, u) = J(y(u), u) + \nu \langle p, f + Bu - Ay(u) \rangle_{\mathcal{V}'}, \quad (16.55)$$

where p is the *Lagrange multiplier*. In this framework the (unconstrained) critical points are represented by the functions y , p and u . The problem therefore becomes

$$\text{find } (y, p, u) \in \mathcal{V} \times \mathcal{V} \times \mathcal{U} : \nabla \mathcal{L}(y, p, u)[(\psi, \varphi, \phi)] = 0 \quad \forall (\psi, \varphi, \phi) \in \mathcal{V} \times \mathcal{V} \times \mathcal{U}$$

that is

$$\begin{cases} \mathcal{L}_p(y, p, u)[\psi] = 0 & \forall \psi \in \mathcal{V}, \\ \mathcal{L}_y(y, p, u)[\varphi] = 0 & \forall \varphi \in \mathcal{V}, \\ \mathcal{L}_u(y, p, u)[\phi] = 0 & \forall \phi \in \mathcal{U}. \end{cases} \quad (16.56)$$

We have used the abridged notation \mathcal{L}_y to indicate the Gâteaux derivative of \mathcal{L} with respect to y (see Definition 2.6). The notations \mathcal{L}_p and \mathcal{L}_u have a similar meaning.

Consider now as an example a state equation of elliptic type with two linear operators A and B that we rewrite in the weak form (16.29); given $u \in \mathcal{U}$ and $f \in \mathcal{H}$,

$$\text{find } y = y(u) \in \mathcal{V} : a(y, \varphi) = (f, \varphi) + b(u, \varphi) \quad \forall \varphi \in \mathcal{V}. \quad (16.57)$$

The bilinear form $a(\cdot, \cdot)$ is associated with the operator A , $b(\cdot, \cdot)$ with B . The cost functional to be minimized can be expressed as follows

$$J(y(u), u) = \|Cy(u) - z_d\|_{\mathcal{Z}}^2 + n(u, u), \quad (16.58)$$

where C is the operator that maps the state variable into the space \mathcal{Z} of the observed functions, z_d is the target observation and $n(\cdot, \cdot)$ is a symmetric bilinear form. Thus far, no assumption was made on the boundary conditions, the kind of control (either distributed or on the boundary) and the kind of observation. This was made on purpose in order to consider a very general framework. In weak form, (16.55) becomes

$$\mathcal{L}(y, p, u) = J(y(u), u) + b(u, p) + (f, p) - a(y, p).$$

Since

$$\begin{aligned}\mathcal{L}_p(y, p, u)[\varphi] &= (f, \varphi) + b(u, \varphi) - a(y, \varphi) & \forall \varphi \in \mathcal{V}, \\ \mathcal{L}_y(y, p, u)[\psi] &= 2(Cy - z_d, C\psi)_z - a(\psi, p) & \forall \psi \in \mathcal{V}, \\ \mathcal{L}_u(y, p, u)[\phi] &= b(\phi, p) + 2n(u, \phi) & \forall \phi \in \mathcal{U},\end{aligned}\tag{16.59}$$

(16.56) yields the *optimality system*

$$\left\{ \begin{array}{ll} y \in \mathcal{V} : a(y, \varphi) = b(u, \varphi) + (f, \varphi) & \forall \varphi \in \mathcal{V}, \\ p \in \mathcal{V} : a(\psi, p) = 2(Cy - z_d, C\psi)_z & \forall \psi \in \mathcal{V}, \\ u \in \mathcal{U} : b(\phi, p) + 2n(u, \phi) = 0 & \forall \phi \in \mathcal{U}. \end{array} \right.\tag{16.60}$$

It is worth noticing that, upon rescaling p as follows: $\tilde{p} = p/2$, the variables (y, \tilde{p}, u) obtained from (16.60) actually satisfy the system (16.43) obtained in the framework of the Lions' approach. Note that the annulment of \mathcal{L}_p yields the state equation (in weak form), that of \mathcal{L}_y generates the equation for the Lagrange multiplier (which can be identified with the adjoint equation), and that of \mathcal{L}_u yields the so-called *sensitivity* equation expressing the condition that the optimum is achieved. The adjoint variable, that is the Lagrange multiplier, is associated to the *sensitivity* of the cost functional J with respect to the variation of the observation function, and therefore to the variation of the control u .

It turns out to be very useful to express the Gâteaux derivative of the Lagrangian \mathcal{L}_u in terms of the derivative of the cost functional J w.r.t. u , (16.60), according to what we have seen in Sec. 2.2. This correspondence is guaranteed by the Riesz representation theorem (see Theorem 2.1). Indeed, since $\mathcal{L}_u[\phi]$ is a linear and bounded functional, and ϕ belongs to the Hilbert space \mathcal{U} , case by case we can compute J' , that is, from the third of (16.59), and from (16.45)

$$\mathcal{L}_u[\phi] = (J'(u), \phi)_{\mathcal{U}} = b(\phi, p) + 2n(u, \phi).$$

It is worth noticing how the adjoint equation is generated. According to Lions' theory the adjoint equation is based on the use of the adjoint operator (see equation (16.39)), whereas when using the approach based on the Lagrangian we obtain it by differentiating \mathcal{L} with respect to the state variable. The adjoint variable $p(u)$, when computed on the optimal control u , corresponds to the Lagrange multiplier. In general, the Lions method and the method based on the use of the Lagrangian do not lead to the same definition of adjoint problem, henceforth they give rise to numerical methods which can behave differently. For a correct solution of the optimal control problem is therefore essential to be coherent with the kind of approach that we are considering. Another crucial issue is the derivation of the boundary conditions for the adjoint problem; the two different approaches may lead to different kind of boundary conditions. More particularly, the approach based on the Lagrangian yields the boundary conditions for the adjoint problem automatically, while this is not the case for the other approach.

16.9 Iterative solution of the optimal control problem

For the numerical solution of optimal control problems, two different paradigms can be adopted:

- (*optimize-then-discretize*) we first apply the iterative method, then we discretize the various steps of the algorithm, or
- (*discretize-then-optimize*) we first discretize our optimal control problem and then we apply an iterative algorithm to solve its discrete version.

This discussion is deferred until Sec. 16.12. In this section we illustrate the way an iterative algorithm can be used to generate a sequence that hopefully converges to the optimal control function u .

As previously discussed, an optimal control problem can be formulated according to the Lions approach, yielding the set of equations (16.43), or by means of the Lagrangian, in which case the equations to be solved are (16.60). In either case we end up with the optimality system:

- i) the state equation;
- ii) the adjoint equation;
- iii) the equation expressing the optimality condition.

In particular, the third equation is related to the variation of the cost functional, either explicitly in the Lions approach, or implicitly (through the Riesz representation theorem) when using the Lagrangian approach. Indeed, in the case of linear elliptic equations previously examined we obtain, respectively:

- $\frac{1}{2}J'(u) = B'p(u) + \Lambda_{\mathcal{U}}Nu;$
- $\mathcal{L}_u[\phi] = b(\phi, p) + 2n(u, \phi) \quad \forall \phi \in \mathcal{U}.$

In order to simplify our notation, in the remainder of this section we will use the symbol J' not only to indicate the derivative of the cost functional but also that of the Lagrangian, $\mathcal{L}_u[\phi]$. The evaluation of J' at a given point of the control region $(\Omega, \Gamma$, or one of their subsets) provides an indication of the sensitivity of the cost functional J (at that very point) with respect to the variations of the control function u . Otherwise said, an infinitesimal variation δu of the control about a given value u , entails, up to infinitesimals of higher order, a variation δJ of the cost functional that is proportional to $J'(u)$. This suggests the use of the following *steepest descent* iterative algorithm. If u^k denotes the value of the control function at step k , the control function at the next step, $k + 1$, can be obtained as follows:

$$u^{k+1} = u^k - \tau^k J'(u^k), \quad (16.61)$$

where J' represents the *descent direction*, and τ^k the *acceleration parameter*. Although not necessarily the most efficient, the choice (16.61) is however pedagogically useful to understand the role played by J' and therefore from the adjoint variable p . A method for the search of an optimal control can therefore be devised in terms of the following iterative algorithm:

1. Find the expression of the adjoint equation and of the derivative J' by either one of the two approaches (Lions' or Lagrangian);
2. Provide an initial guess u^0 of the control function u ;
3. Solve the *equation of state* in y using the above guess;
4. Being known the state variable and the observation target z_d , evaluate the *cost functional* J ;
5. Solve the *adjoint equation* for p , being known y and z_d ;
6. Being known the adjoint variable, compute J' ;
7. If the chosen stopping test is fulfilled (up to a given *tolerance*) then exit (jump to point 10), otherwise continue;
8. Compute the parameter(s) for the acceleration of the iterative algorithm (for instance τ^k);
9. Compute the new *control function*, e.g. through equation (16.61), and return to point 3.;
10. Take the last computed variable to generate the "converged" unknowns u , y and p .

In Fig. 16.10 we display a flow chart that illustrates the above algorithm.

Remark 16.4 A convenient stopping test can be built on the measure of the distance, in a suitable norm, between the observed variable z and the (desired) target observation z_d , say

$$\|z^k - z_d\|_{\mathcal{Z}} \leq Tol.$$

However, in general this does not guarantee that $J(u^k) \rightarrow 0$ as $k \rightarrow \infty$, that is, at convergence the cost functional J might be non-null. A different stopping criterion is based on the evaluation of the derivative of the cost functional

$$\|J'(u^k)\|_{\mathcal{U}'} \leq Tol.$$

The value of the tolerance must be sufficiently small w.r.t. both the value of $\|J'(u^0)\|$ on the initial control and the vicinity to the target observation that we want to achieve. •

Remark 16.5 The adjoint variable is defined on the whole computational domain. For the evaluation of $J'(u)$ it would be necessary to operate a *restriction* of the adjoint variable p on that portion of the domain, or of its boundary, on which the control function u is defined. See, for several examples, Tables 16.1 and 16.2. •

The descent iterative method requires the determination of a suitable parameter τ^k . The latter should guarantee that the convergence of the cost functional to its minimum

$$J_* = \inf_{u \in \mathcal{U}} J(u) \geq 0,$$

be monotone, that is

$$J(u^k - \tau^k J'(u^k)) < J(u^k).$$

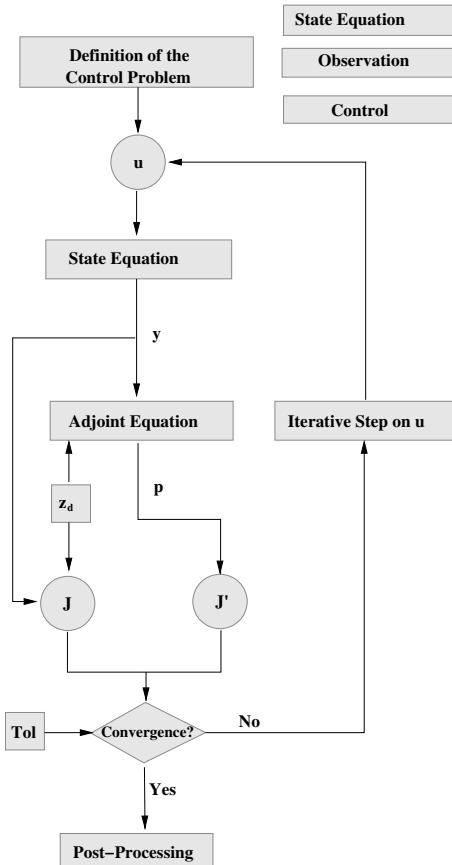


Fig. 16.10. Flow chart of a possible iterative algorithm for the solution of an optimal control problem

In those cases in which the value of J_* is known (e.g., $J_* = 0$), then the parameter can be chosen as follows (see, e.g., [Ago03] and [Vas81])

$$\tau^k = \frac{(J(u^k) - J_*)}{\|J'(u^k)\|_{U'}^2}. \quad (16.62)$$

As an example, consider the following control problem

$$\begin{cases} Ay = f + Bu, \\ \inf J(u), \quad \text{where } J(u) = \nu \|u\|_U^2 + \|Cy - z_d\|_Z^2, \quad \nu \geq 0. \end{cases}$$

The previous iterative method becomes:

$$\begin{cases} Ay^k = f + Bu^k, \\ A^*p^k = C'(Cy^k - z_d), \\ u^{k+1} = u^k - \tau^k 2(\nu u^k + B'p^k). \end{cases}$$

If $\text{Ker}(B'A^{*-1}C') = \{0\}$ this problem admits a solution, moreover $J(u) \rightarrow 0$ if $\nu \rightarrow 0$. Thus, if $\nu \simeq 0^+$ we can assume that $J_* \simeq 0$, whence, thanks to (16.62),

$$\tau^k = \frac{J(u^k)}{\|J'(u^k)\|_{\mathcal{U}'}^2} = \frac{\nu \|u^k\|_{\mathcal{U}}^2 + \|Cy^k - z_d\|_{\mathcal{Z}}^2}{\|2\nu u^k + B'p^k\|_{\mathcal{U}'}^2}. \quad (16.63)$$

When considering a *discretized* optimal control problem, for instance using the Galerkin–finite element method (as we will see in Sec. 16.12, instead of looking for the minimum of $J(u)$, with $J : \mathcal{U} \mapsto \mathbb{R}$, one looks for the minimum of $J(\mathbf{u})$, where $J : \mathbb{R}^n \mapsto \mathbb{R}$ and $\mathbf{u} \in \mathbb{R}^n$ is the vector whose components are the nodal values of the control $u \in \mathcal{U}$). We will make this assumption in the remainder of this section.

As previously noted, the steepest descent method (16.61) is one among several iterative algorithms that could be used for the solution of an optimal control problem. As a matter of fact, this method is a special case of *gradient methods* which write

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \tau^k \mathbf{d}^k, \quad (16.64)$$

where \mathbf{d}^k represents a descent direction, that is a vector that satisfies

$$\mathbf{d}^{kT} \cdot J'(\mathbf{u}^k) < 0 \quad \text{if } \nabla J(\mathbf{u}^k) \neq \mathbf{0}.$$

Depending upon the criterion that is followed to choose \mathbf{d}^k we obtain several special cases:

- *Newton method*, for which

$$\mathbf{d}^k = -H(\mathbf{u}^k)^{-1} \nabla J(\mathbf{u}^k),$$

where $H(\mathbf{u}^k)$ is the Hessian matrix of $J(\mathbf{u})$ computed at $\mathbf{u} = \mathbf{u}^k$;

- *quasi–Newton methods*, for which

$$\mathbf{d}^k = -B_k \nabla J(\mathbf{u}^k),$$

where B_k is an approximation of the inverse of $H(\mathbf{u}^k)$;

- *conjugate gradient method*, for which

$$\mathbf{d}^k = -\nabla J(\mathbf{u}^k) + \beta_k \mathbf{d}^{k-1},$$

where β_k is a scalar to be chosen in such a way that $\mathbf{d}^{kT} \mathbf{d}^{k-1} = 0$. (See also Chap. 7.)

Once \mathbf{d}^k is computed, the parameter τ^k should be chosen in such a way to guarantee the monotonicity property

$$J(\mathbf{u}^k + \tau^k \mathbf{d}^k) < J(\mathbf{u}^k). \quad (16.65)$$

A more stringent requirement is that the following scalar minimization problem is solved

$$\text{find } \tau^k : \phi(\tau^k) = J(\mathbf{u}^k + \tau^k \mathbf{d}^k) \text{ minimum;}$$

this would guarantee the following orthogonality property

$$\mathbf{d}^{k^T} \cdot \nabla J(\mathbf{u}^k) = 0.$$

Often, for the computation of τ^k is based on heuristic methods. One way is to start from a relatively large value of the parameter τ^k , which is then halved until when (16.65) is verified. However, this approach is not always successful. The idea is therefore to adopt more stringent criteria than (16.65) when choosing τ^k , with the aim of achieving a high convergence rate from one hand and, from another hand, to avoid too small steps. The first goal is achieved by requiring that

$$J(\mathbf{u}^k) - J(\mathbf{u}^k + \tau^k \mathbf{d}^k) \geq -\sigma \tau^k \mathbf{d}^{k^T} \cdot \nabla J(\mathbf{u}^k), \quad (16.66)$$

for a suitable $\sigma \in (0, 1/2)$. This inequality enforces that the average rate of decrease of J at \mathbf{u}^{k+1} along the direction \mathbf{d}^k be at least equal to a given fraction of the initial decrease rate at \mathbf{u}^k . On the other hand, too small steps are avoided by requiring that

$$|\mathbf{d}^{k^T} \cdot \nabla J(\mathbf{u}^k + \tau^k \mathbf{d}^k)| \leq \beta |\mathbf{d}^{k^T} \cdot \nabla J(\mathbf{u}^k)|, \quad (16.67)$$

for a suitable $\beta \in (\sigma, 1)$, so to guarantee that (16.66) be satisfied too. In practice, $\sigma \in [10^{-5}, 10^{-1}]$ and $\beta \in [10^{-1}, 1/2]$. Several strategies can be chosen for the choice of τ^k that are compatible with the conditions (16.66) and (16.67). A popular one is based on the *Armijo formulae* (see, e.g. [MP01]). For fixed $\sigma \in (0, 1/2)$, $\beta \in (0, 1)$ and $\bar{\tau} > 0$, one chooses $\tau^k = \beta^{m_k} \bar{\tau}$, m_k being the first non-negative integer for which (16.66) is satisfied. One can even take $\tau^k = \bar{\tau}$ for all k , at least in those cases in which the evaluation of the cost functional J is very involved.

For a more comprehensive discussion on this issue see, e.g., [GMSW89] [KPTZ00], [MP01] and [NW06].

16.10 Numerical examples

In this section we illustrate two examples of control problems inspired by real life applications. Both problems are analyzed by means of the Lagrangian approach outlined in Sec. 16.8.2; for simplicity the optimal control function is in fact a scalar value.

16.10.1 Heat dissipation by a thermal fin

Thermal fins are used to dissipate the heat produced by some devices with the goal to maintain their temperature below some limit values. Typically, they are used for electronic devices such as transistors; when active and depending on the electrical power, the latter could incur in failure with a higher frequency when the operational temperature increases. This represents a major issue while designing the dissipator, which often is used in combination with a fan able to considerably improve the thermal dissipation via forced convection, thus containing the temperature of the device. For further deepening we refer the reader e.g. to [C07]; for a further example in the field of parametrized problems see [OP07].

In our example, we aim at regulating the intensity of the forced convection associated with the fan in order to keep the temperature of the transistor as close as possible to a desired value. The control is represented by the scalar coefficient of forced convection, while the observation is the temperature on the boundary of the thermal fin which is in contact with the transistor.

Let us consider the following state problem, whose solution y (in Kelvin degrees [K]) represents the temperature in the thermal fin

$$\begin{cases} -\nabla \cdot (k \nabla y) = 0 & \text{in } \Omega, \\ -k \frac{\partial y}{\partial n} = -q & \text{on } \Gamma_N, \\ -k \frac{\partial y}{\partial n} = (h + U)(y - y_\infty) & \text{on } \Gamma_R = \partial\Omega \setminus \Gamma_N, \end{cases} \quad (16.68)$$

where the domain Ω and its boundary are reported in Fig. 16.11. The coefficient k ($[W/(mm K)]$) represents the thermal conductivity (aluminium is considered in this case), while h and U (our control variable) the natural and forced convection coefficients ($[W/(mm^2 K)]$), respectively. Let us remark that when the fan is active, then the value of U is greater than zero; if $U = 0$ heat dissipation is due only to natural convection. The temperature y_∞ corresponds to the temperature of the air far away from the dissipator, while q ($[W/mm^2]$) stands for heat for surface unit emitted by the transistor and entering in the thermal fin through the boundary Γ_N .

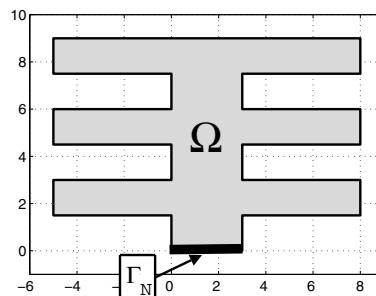


Fig. 16.11. Thermal fin: computational domain; unit measure in mm

The weak form of problem (16.68) reads for a given $U \in \mathcal{U} = \mathbb{R}$

$$\text{find } y \in \mathcal{V} : a(y, \varphi; U) = b(U, \varphi) \quad \forall \varphi \in \mathcal{V}, \quad (16.69)$$

being $\mathcal{V} = H^1(\Omega)$, $a(\varphi, \psi; \phi) = \int_{\Omega} k \nabla \varphi \cdot \nabla \psi \, d\Omega + \int_{\Gamma_R} (h + \phi) \varphi \psi \, d\gamma$ and $b(\phi, \psi) = \int_{\Gamma_R} (h + \phi) y_{\infty} \psi \, d\gamma + \int_{\Gamma_N} q \psi \, d\gamma$. Existence and uniqueness of the solution of the Robin-Neumann problem (16.69) is ensured by the Peetre-Tartar lemma (see Remark 3.5).

The optimal control problem consists in finding the value of the forced convection coefficient U s.t. the following cost functional $J(y, U)$ is minimum, being $y \in \mathcal{V}$ solution of (16.69)

$$J(y, U) = \nu_1 \int_{\Gamma_N} (y - z_d)^2 \, d\gamma + \nu_2 U^2. \quad (16.70)$$

This leads to keeping the temperature of the transistor as close as possible to the desired value z_d [K] and the forced convection coefficient close to zero depending on the value of the coefficient $\nu_2 > 0$; in particular we assume $\nu_1 = 1 / \int_{\Gamma_N} z_d^2 \, d\gamma$ and $\nu_2 = \nu_2^0 / h^2$, for a suitable ν_2^0 .

The analysis of the problem is carried out by means of the Lagrangian functional $\mathcal{L}(y, p, U) = J(y, U) + b(p; U) - a(y, p; U)$. In particular, we obtain via differentiation of $\mathcal{L}(\cdot)$ the following adjoint equation for a given $U \in \mathbb{R}$ and the corresponding $y = y(U) \in \mathcal{V}$

$$\text{find } p \in \mathcal{V} : a(\psi, p; U) = c(y, \psi) \quad \forall \psi \in \mathcal{V}, \quad (16.71)$$

where $c(\varphi, \psi) = 2\nu_1 \int_{\Gamma_N} (\varphi - z_d) \psi \, d\gamma$. Similarly, from the optimality condition we deduce that

$$J'(U) = 2\nu_2 U - \int_{\Gamma_R} (y(U) - y_{\infty}) p(U) \, d\gamma. \quad (16.72)$$

We assume now $k = 2.20 \text{ W}/(\text{mm K})$, $h = 15.0 \cdot 10^{-6} \text{ W}/(\text{mm}^2 \text{ K})$, $y_{\infty} = 298.15 \text{ K}$ ($= 25^\circ\text{C}$), $z_d = 353.15 \text{ K}$ ($= 80^\circ\text{C}$) and $\nu_2^0 = 10^{-3}$. The problem is solved by means of the finite element method with piecewise quadratic basis functions on a triangular mesh with 1608 elements and 934 d.o.f. The steepest descent

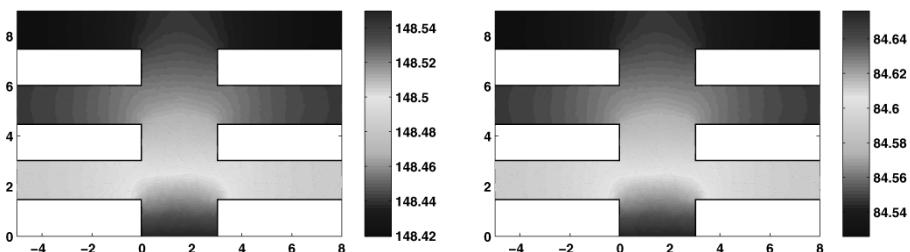


Fig. 16.12. Thermal fin: state solution (temperature [$^\circ\text{C}$]), at the initial step (natural convection) (left) and at the optimum (right)

iterative method is used for the optimization with $\tau^k = \tau = 10^{-9}$ (see (16.61)); the iterative procedure is stopped when $|J'(U^k)|/|J'(U^0)| < tol = 10^{-6}$. At the initial step we consider natural convection for the dissipation of the heat s.t. $U = 0.0$ to which corresponds a cost functional $J = 0.0377$. The optimum is reached after 132 iterations yielding the optimal cost functional $J = 0.00132$ for the optimal value of the forced convection coefficient $U = 16.1 \cdot 10^{-6} W/(mm^2 K)$. Ideally, the fan should be designed in order to warrant this value of the forced convection coefficient. In Fig. 16.12 we display the state solution at the initial step and that at the optimum; we observe that the temperature on Γ_N is not equal to z_d , being the coefficient $\nu_2^0 \neq 0$.

16.10.2 Thermal pollution in a river

Industrial activities are often related with pollution phenomena which need to be properly taken into account while designing a new plant or planning its operations. In this field, thermal pollution could often affect a river or a channel used with the purpose of cooling the hot liquids produced by industrial plants, thus affecting the vital processing of the flora and fauna.

In this case the goal could consist in regulating the heat emission in a branch of a river in order to maintain the temperature of the river close to a desired threshold without considerably affecting the ideal heat emission rate of the plant.

We introduce the following state problem, whose solution y represents the temperature in the channels and branches of the river considered

$$\begin{cases} \nabla \cdot (-k\nabla y + \mathbf{V}y) = f\chi_1 + U\chi_2 & \text{in } \Omega, \\ y = 0 & \text{on } \Gamma_{IN}, \\ (-k\nabla y + \mathbf{V}y) \cdot \mathbf{n} = 0 & \text{on } \Gamma_N, \end{cases} \quad (16.73)$$

where the domain Ω and the boundary Γ_{IN} are indicated in Fig. 16.13, while $\Gamma_N = \partial\Omega \setminus \Gamma_{IN}$ (note that the outflow boundary Γ_{OUT} displayed in Fig. 16.13 is part of Γ_N). χ_1 , χ_2 and χ_{OBS} represent the characteristic functions of the subdomains Ω_1 ,

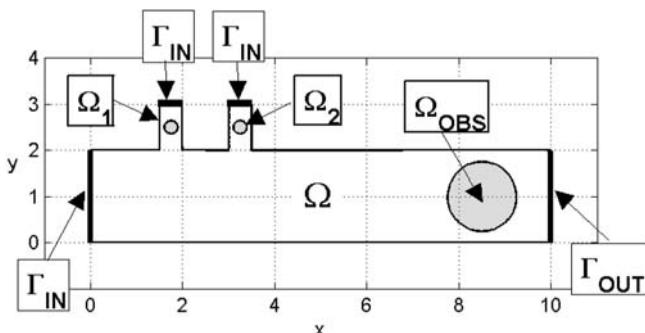


Fig. 16.13. Pollution in the river: computational domain

Ω_2 and Ω_{OBS} , respectively. Dimensionless quantities are considered for this test case. The coefficient k is the thermal diffusivity coefficient, which also accounts for the contribution to the diffusion of turbulence phenomena, while \mathbf{V} is the advection field which describes the motion of the water in the domain Ω (we comment later on the way to find it). The source term $f \in \mathbb{R}$ and the control $U \in \mathcal{U} = \mathbb{R}$ represent the heat emission rates from two industrial plants; f is given, whereas U has to be determined on the basis of the solution of the optimal control problem. In particular, we want the following cost functional to be minimized

$$J(y, U) = \int_{\Omega_{OBS}} (y - z_d)^2 d\Omega + \nu(U - U_d)^2, \quad (16.74)$$

where z_d is the desired temperature in Ω_{OBS} , U_d the ideal heat emission rate and $\nu > 0$ is conveniently chosen.

The optimal control problem is set up by means of the Lagrangian approach. With this aim, (16.73) is rewritten in weak form, for a given U , as

$$\text{find } y \in \mathcal{V} : a(y, \varphi) = b(U, \varphi) \quad \forall \varphi \in \mathcal{V}, \quad (16.75)$$

where $\mathcal{V} = H_{\Gamma_{IN}}^1(\Omega)$, $a(\varphi, \psi) = \int_{\Omega} k \nabla \varphi \cdot \nabla \psi d\Omega$ and $b(U, \psi) = f \int_{\Omega_1} \psi d\Omega + U \int_{\Omega_2} \psi d\Omega$. Existence and uniqueness of the solution of problem (16.75) is proven as indicated in Sec. 3.4.

The Lagrangian functional is $\mathcal{L}(y, p, U) = J(y, U) + b(U, p) - a(y, p)$. By differentiation of $\mathcal{L}(\cdot)$ w.r.t. $y \in \mathcal{V}$ we obtain the adjoint equation

$$\text{find } p \in \mathcal{V} : a(\psi, p) = c(y, \psi) \quad \forall \psi \in \mathcal{V}, \quad (16.76)$$

where $c(\varphi, \psi) = 2 \int_{\Omega_{OBS}} (\varphi - z_d) \psi d\Omega$. Similarly, we deduce the following derivative of the cost functional

$$J'(U) = 2\nu(U - U_d) + \int_{\Omega_2} p(U) d\Omega. \quad (16.77)$$

We assume now $k = 0.01$, $f = 10.0$, $z_d = 0$ and $U_d = f$. The advection field \mathbf{V} is deduced by solving the Navier–Stokes equations (see Chap. 15) in the domain Ω , with the following boundary conditions: on Γ_{IN} a parabolic profile is prescribed for the velocity, with a maximum velocity equal to 1; on Γ_{OUT} no stress conditions are assumed in the normal direction, together with the slip condition $\mathbf{V} \cdot \mathbf{n} = 0$; finally, no-slip conditions are prescribed on $\partial\Omega \setminus (\Gamma_{IN} \cup \Gamma_{OUT})$. Notations are those displayed in Fig. 16.13. The Reynolds number is equal to $\text{Re} = 500$. The Navier–Stokes problem is solved by means of the Taylor–Hood $\mathbb{P}^2\text{--}\mathbb{P}^1$ (see Sec. 15.4) pairs of finite elements on a mesh composed by 32248 triangles and 15989 nodes. In Fig. 16.14 we report the intensity of the advection field \mathbf{V} and the corresponding streamlines.

The optimal control problem is solved by means of the finite element method with \mathbb{P}^2 basis functions on a triangular mesh with 32812 elements and 16771 d.o.f., using the steepest descent method for the functional optimization; we select $\tau^k = \tau = 5$ (see (16.61)) and the stopping criterium is $|J'(U^k)| / |J'(U^0)| < \text{tol} = 10^{-6}$. The advection field \mathbf{V} obtained by solving the Navier–Stokes equations is interpolated on this

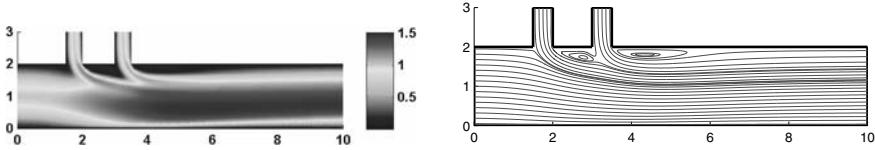


Fig. 16.14. Pollution in the river: intensity of the advection field \mathbf{V} , modulus (left) and streamlines (right)

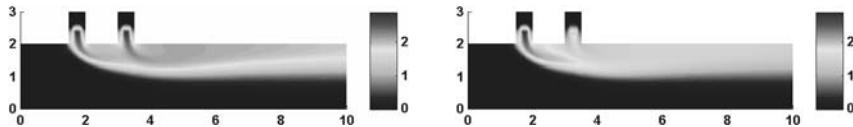


Fig. 16.15. Pollution in the river: state solution (temperature), at the initial step ($U = U_d$) (left) and at the optimum (right)

new mesh. At the initial step we assume that $U = U_d$, thus obtaining a cost functional $J = 1.884$. The optimal solution is obtained after 15 iterations, the corresponding optimal cost functional is $J = 1.817$ obtained for an optimal heat emission rate of $U = 6.685$. In practice the heat from the plant in Ω_2 should be reduced in order to maintain the temperature in Ω_{OBS} low. In Fig. 16.15 we report the state solutions y (temperature) before and after optimization.

16.11 A few considerations about observability and controllability

A few considerations can be made upon the behavior of iterative methods w.r.t. the kind of optimal control problem that we are solving, more particularly on which kind of variable z we are observing, and which kind of control function u we are using. Briefly, on the relationship between *observability* and *controllability*. We warn the reader that the conclusions that we are going to draw are not suffragated by a general theory, nor they apply to any kind of numerical discretization method.

- *Where we observe.* In general, optimal control problems based on an observation variable distributed in the domain enjoy higher convergence rate than those for which the observation variable is concentrated on the domain boundary. Within the same discretization error tolerance, in the former case the number of iterations can be up to one order of magnitude lower than in the latter.
- *Where we control.* In general, the optimization process is more robust if also the control function is distributed in the domain (as a source term to the state equation, or as coefficient of the differential operator governing the state equation), rather than being concentrated on the domain boundary. More precisely, the convergence rate is higher and its sensitivity to the choice of the acceleration parameter lower for distributed control problems than for boundary control ones, provided of course all of the other parameters are the same.

- *What we observe.* Also the kind of variable that we observe affects the convergence behavior of the iterative scheme. For instance, observing the state variable is less critical than observing either its gradients or some of its higher order derivatives. The latter circumstance occurs quite commonly. e.g., in fluid dynamics problems, when for potential problems one observes the velocity field, or for Navier-Stokes equations one observes the fluid vorticity or its stresses.
- *Shape optimization.* Shape optimization problems are a special class of optimal control problems: as a matter of fact, in this case the control function is not only *on* the boundary, it is *the* boundary itself. The cost functional to be minimized is called *shape functional* as it depends on the domain itself, that is one looks for

$$J(\Omega_{opt}) \leq J(\Omega) \quad \forall \Omega \in \mathcal{D}_{ad},$$

where \mathcal{D}_{ad} is a set of admissible domains. Shape optimization problems are difficult to analyze theoretically and to solve numerically. The numerical grid needs to be changed at every iteration. Besides, non-admissible boundary shapes might be generated in the course of the iterations, unless additional geometrical constraints are imposed. Moreover, special stabilization and regularization techniques might be necessary to prevent numerical oscillations in the case of especially complex problems. More in general, shape optimization problems are more sensitive to the variation of the various parameters that characterize the control problem.

- *Adjoint problem and state problem.* For steady elliptic problems like those considered in this chapter, the use of the adjoint problem provides the gradient of the cost functional at the same computational cost as that of the state problem. This can be considered as an approach alternative to those based on inexact or automatic differentiation of the cost functional. In the case of shape optimization problems the use of the adjoint problem allows a computational saving with respect to the method based on the shape sensitivity analysis, as the latter depends on the (often prohibitive) number of parameters that characterize the shape (the control points). See, e.g., [KAJ02].

16.12 Two alternative paradigms for numerical approximation

Let us start by considering a simple example that illustrates some additional difficulties that arise when solving an optimal control problem numerically. For a more insight analysis of the numerical discretization of optimal control problems, see, e.g., [FCZ03] and [Gun03b].

Consider again the state equation (16.1) and assume that the optimal control problem be

“find $u \in \mathcal{U}_{ad}$ such that

$$J(u) \leq J(v) \quad \forall v \in \mathcal{U}_{ad}, \tag{16.78}$$

where J is a given cost functional”.

Now the question is

“How can this problem be conveniently approximated?”

As already anticipated at the beginning of Sec. 16.8, at least two alternative strategies can be pursued:

- **Discretize–then–optimize**

According to this strategy, we discretize at first the control space \mathcal{U}_{ad} by a finite dimensional space $\mathcal{U}_{ad,h}$, and the state equation (16.1) by a discrete one that, in short, we write as follows

$$A_h y_h(u_h) = f_h. \quad (16.79)$$

In a finite element context, the parameter h would denote the finite element grid-size. We assume that the choice of the discrete control space and the discretized state equation be such that a “discrete state” $y_h(v_h)$ exists for every “admissible” discrete control $v_h \in \mathcal{U}_{ad,h}$.

At this stage we look for a discrete optimal control, that is for a function $u_h \in \mathcal{U}_{ad,h}$ such that

$$J(u_h) \leq J(v_h) \quad \forall v_h \in \mathcal{U}_{ad,h}, \quad (16.80)$$

or, more precisely,

$$J(y_h(u_h), u_h) \leq J(y_h(v_h), v_h) \quad \forall v_h \in \mathcal{U}_{ad,h}. \quad (16.81)$$

This corresponds to the following scheme

MODEL → DISCRETIZATION → CONTROL

for which the “*discretize–then–optimize*” expression was coined.

- **Optimize–then–Discretize**

Alternatively we could proceed as follows. We start from the control problem (16.1), (16.78) and we write down the corresponding *optimality system* based on the Euler–Lagrange equations :

$$\begin{aligned} Ay(u) &= f, \\ A'p &= G(y(u)), \end{aligned} \quad (16.82)$$

for a suitable G which depends on the state $y(u)$ and represents the right hand side of the adjoint problem, plus an additional equation (formally corresponding to the third equation of (16.56)) relating the three variables y , p and u which we write in shorthand form

$$Q(y, p, u) = 0. \quad (16.83)$$

At this stage we discretize the system (16.82), (16.83) and solve it numerically. This corresponds to the following procedure:

MODEL → CONTROL → DISCRETIZATION

for which the expression “*optimize–then–discretize*” is used. With respect to the former approach, here we have interchanged the last two steps.

The two strategies do not necessarily yield the same results. For instance, in [IZ99] it is shown that if the state equation is a dynamic problem that provides the vibrations of an elastic structure and a finite element approximation is used, then the first strategy yields wrong results. This can be attributed to the lack of accuracy of the finite element method for high frequency solutions of the wave equation (see [Zie00]).

On the other side, it has been observed that for several shape optimization problems in optimal design, the former strategy should be preferable; see, e.g., [MP01] and [Pir84].

The strategy of choice does certainly depend on the nature of the differential problem at hand. In this respect, control problems governed by elliptic or parabolic PDEs are less problematic than those governed by hyperbolic equations because of their intrinsic dissipative nature. See for a discussion [Zua03]. The reader should however keep abreast: many important developments are expected in this field in the next coming years.

16.13 A numerical approximation of an optimal control problem for advection–diffusion equations

In this section we consider an optimal control problem for an advection–diffusion equation formulated by the Lagrangian approach. For its numerical discretization the two different strategies: "discretize–then–optimize" and "optimize–then–discretize", will be considered. The numerical approximation will be based on stabilized finite element methods, as seen in Chap. 11. Besides, an a–posteriori error analysis will be carried out, according to the guidelines illustrated in Chap. 4. For more details we refer to [QRDQ06], [DQ05], and the references therein.

We consider the following advection–diffusion boundary–value problem

$$\begin{cases} L(y) = -\nabla \cdot (\mu \nabla y) + \mathbf{V} \cdot \nabla y = u & \text{in } \Omega, \\ y = 0 & \text{on } \Gamma_D, \\ \mu \frac{\partial y}{\partial n} = 0 & \text{on } \Gamma_N, \end{cases} \quad (16.84)$$

where Ω is a two–dimensional domain, Γ_D and Γ_N provide a disjoint partition of the domain boundary $\partial\Omega$, $u \in L^2(\Omega)$ is the control variable while μ and \mathbf{V} are two given functions (the former being a positive viscosity). Here $\Gamma_D = \{\mathbf{x} \in \partial\Omega : \mathbf{V}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0\}$ is the inflow boundary, $\mathbf{n}(\mathbf{x})$ is the outward normal unit vector, while $\Gamma_N = \partial\Omega \setminus \Gamma_D$ is the outflow boundary.

We assume that the observation function be restricted to a subdomain $D \subseteq \Omega$ and that the optimal control problem reads

$$\text{find } u : J(u) = \int_D (gy(u) - z_d)^2 \, dD \text{ minimum,} \quad (16.85)$$

where $g \in C^\infty(\Omega)$ maps the variable y into the space of observations, and z_d is the desired observation (the target). By setting

$$\mathcal{V} = H_{\Gamma_D}^1 = \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\} \text{ and } \mathcal{U} = L^2(\Omega),$$

the Lagrangian functional introduced in Sec. 16.8 becomes

$$\mathcal{L}(y, p, u) = J(u) + F(p; u) - a(y, p), \quad (16.86)$$

where:

$$a(\varphi, \psi) = \int_{\Omega} \mu \nabla \varphi \cdot \nabla \psi \, d\Omega + \int_{\Omega} \mathbf{V} \cdot \nabla \varphi \, \psi \, d\Omega, \quad (16.87)$$

$$F(\varphi; u) = \int_{\Omega} u \varphi \, d\Omega. \quad (16.88)$$

By differentiating \mathcal{L} w.r.t. the state variable y , we obtain the adjoint equation (in weak form)

$$\text{find } p \in \mathcal{V} : a^{ad}(p, \psi) = F^{ad}(\psi; \varphi) \quad \forall \psi \in \mathcal{V}, \quad (16.89)$$

where:

$$a^{ad}(p, \psi) = \int_{\Omega} \mu \nabla p \cdot \nabla \psi \, d\Omega + \int_{\Omega} \mathbf{V} \cdot \nabla \psi \, p \, d\Omega, \quad (16.90)$$

$$F^{ad}(\psi; y) = \int_D 2(g y - z_d) g \, \psi \, dD. \quad (16.91)$$

Its differential (distributional) counterpart reads

$$\begin{cases} L^{ad}(p) = -\nabla \cdot (\mu \nabla p + \mathbf{V} p) = \chi_D g (g y - z_d) & \text{in } \Omega, \\ p = 0 & \text{on } \Gamma_D, \\ \mu \frac{\partial p}{\partial n} + \mathbf{V} \cdot \mathbf{n} \, p = 0 & \text{on } \Gamma_N, \end{cases} \quad (16.92)$$

where χ_D denotes the characteristic function of the region D . By differentiating \mathcal{L} w.r.t. the control function u , we obtain the optimality equation (see the third equation of (16.56)), that is

$$\int_{\Omega} \phi \, p \, d\Omega = 0 \quad \forall \phi \in L^2(\Omega). \quad (16.93)$$

This equation provides the sensitivity $J'(u)$ of the cost functional w.r.t. the control variable. Denoting for simplicity this sensitivity by δu , in this case we obtain $\delta u = p(u) = p$. Finally, by differentiating \mathcal{L} w.r.t. the adjoint variable p , as usual we obtain the state equation (in weak form)

$$\text{find } y \in \mathcal{V} : a(y, \varphi) = F(\varphi; u) \quad \forall \varphi \in \mathcal{V}. \quad (16.94)$$

16.13.1 The strategies “optimize–then–discretize” and “discretize–then–optimize”

From a numerical viewpoint, the minimization algorithm introduced in Sec. 16.9, requires, at every step, the numerical approximation of both the state and the adjoint boundary–value problems. For both problems we can use the stabilized Galerkin–Least–Squares finite element formulations introduced in Sec. 11.8.6. The corresponding discretized equations respectively read:

$$\text{find } y_h \in \mathcal{V}_h : a(y_h, \varphi_h) + \bar{s}_h(y_h, \varphi_h) = F(\varphi_h; u_h) \quad \forall \varphi_h \in \mathcal{V}_h, \quad (16.95)$$

$$\bar{s}_h(y_h, \varphi_h) = \sum_{K \in \mathcal{T}_h} \delta_K \int_K R(y_h; u_h) L(\varphi_h) dK, \quad (16.96)$$

$$\text{find } p_h \in \mathcal{V}_h : a^{ad}(p_h, \psi_h) + \bar{s}_h^{ad}(p_h, \psi_h) = F^{ad}(\psi_h; y_h) \quad \forall \psi_h \in \mathcal{V}_h, \quad (16.97)$$

$$\bar{s}_h^{ad}(p_h, \psi_h) = \sum_{K \in \mathcal{T}_h} \delta_K \int_K R^{ad}(p_h; y_h) L^{ad}(\psi_h) dK, \quad (16.98)$$

where δ_K is a stabilization parameter, $R(y; u) = L(y) - u$, $R^{ad}(p; y) = L^{ad}(p) - G(y)$, with $G(y) = 2\chi_D g(g y - z_d)$. This is the paradigm “optimize–then–discretize”; see Sec. 16.12 and, for the specific problem at hand, [Bec01], [CH01], [Gun03b].

In the paradigm “discretize–then–optimize”, the one that we will adopt in the following, we first discretize (by the same stabilized GLS formulation (Eq.(16.95) and (16.96)), next we introduce the discrete Lagrangian functional

$$\mathcal{L}_h(y_h, p_h, u_h) = J(y_h, u_h) + F(p_h; u_h) - a(y_h, p_h) - \bar{s}_h(y_h, p_h). \quad (16.99)$$

At this stage, by differentiating w.r.t. y_h , we obtain the discrete adjoint equation (16.97), however this time the stabilization term is different, precisely

$$\bar{s}_h^{ad}(p_h, \psi_h) = \sum_{K \in \mathcal{T}_h} \delta_K \int_K L(\psi_h) L(p_h) dK. \quad (16.100)$$

Now, by differentiating \mathcal{L}_h w.r.t. u_h and using the Riesz representation theorem (Theorem 2.1), we obtain, noting that $u_h \in \mathcal{V}_h$,

$$\delta u_h = p_h + \sum_{K \in \mathcal{T}_h} \delta_K \int_K L(p_h) dK.$$

In particular, the new stabilized Lagrangian now reads [DQ05]

$$\mathcal{L}_h^s(y_h, p_h, u_h) = \mathcal{L}(y_h, p_h, u_h) + S_h(y_h, p_h, u_h), \quad (16.101)$$

where we have set

$$S_h(y, p, u) = \sum_{K \in \mathcal{T}_h} \delta_K \int_K R(y; u) R^{ad}(p; y) dK. \quad (16.102)$$

By differentiating \mathcal{L}_h^s we obtain the new discretized state and adjoint problems, which can still be written under the forms (16.95) and (16.97), however this time the stabilization terms read, respectively, as follows:

$$s_h(y_h, \varphi_h; u_h) = - \sum_{K \in \mathcal{T}_h} \delta_K \int_K R(y_h; u_h) L^{ad}(\varphi_h) dK, \quad (16.103)$$

$$s_h^{ad}(p_h, \psi_h; y_h) = - \sum_{K \in \mathcal{T}_h} \delta_K \int_K (R^{ad}(p_h; y_h) L(\psi_h) - R(y_h; u_h) G'(\psi_h)) dK, \quad (16.104)$$

having set $G'(\psi) = 2\chi_D g^2 \psi$. Finally, the sensitivity of the cost functional becomes now

$$\delta u_h(p_h, y_h) = p_h - \sum_{K \in \mathcal{T}_h} \delta_K R^{ad}(p_h; y_h). \quad (16.105)$$

16.13.2 A posteriori error estimates

With the aim of obtaining a suitable a posteriori error estimate for the optimal control problem we use as error indicator the error on the cost functional, as done in [BKR00]. Moreover, we will split this error into two terms, that we will identify as *iteration error* and *discretization error*. In particular, for the a discretization error we will make use of duality principle advocated in [BKR00] for the grid adaptivity.

Iteration error and discretization error. At every step k of the iterative algorithm for the minimization of the cost functional we consider the error

$$\varepsilon^{(k)} = J(y^*, u^*) - J(y_h^k, u_h^k), \quad (16.106)$$

where the apex $*$ identifies the variables corresponding to the optimal value of the control, while y_h^k denotes the discrete state variable at the step k . (The variables y_h^k and u_h^k have a similar meaning.) We define *discretization error* $\varepsilon_D^{(k)}$ [DQ05] the component of the total error $\varepsilon^{(k)}$ arising from step k , whereas we call *iteration error* $\varepsilon_{IT}^{(k)}$ [DQ05] the component of $\varepsilon^{(k)}$ that represents the difference between the value of the cost functional computed on the exact variables at step k and the value $J^* = J(y^*, u^*)$ of the cost functional at the optimum. In conclusion, the total error $\varepsilon^{(k)}$ (16.106) can be written as

$$\varepsilon^{(k)} = (J(y^*, u^*) - J(y^k, u^k)) + (J(y^k, u^k) - J(y_h^k, u_h^k)) = \varepsilon_{IT}^{(k)} + \varepsilon_D^{(k)}. \quad (16.107)$$

In the following we will apply an a posteriori error estimate only on $\varepsilon_D^{(k)}$, that is the only part of $\varepsilon^{(k)}$ that can be reduced by a grid refinement procedure. Since the gradient of $\mathcal{L}(\mathbf{x})$, $\mathbf{x} = (y, p, u)$, is linear w.r.t. \mathbf{x} , when using the algorithm (16.61) with $\tau^k = \tau = \text{constant}$, the iteration error $\varepsilon_{IT}^{(k)}$ becomes $\varepsilon_{IT}^{(k)} = \frac{1}{2} (\delta u(p^k, u^k), u^* - u^k)$, which in the current case becomes ([DQ05])

$$\varepsilon_{IT}^{(k)} = -\frac{1}{2}\tau \|p^k\|_{L^2(\Omega)}^2 - \frac{1}{2}\tau \sum_{r=k+1}^{\infty} (p^k, p^r)_{L^2(\Omega)}. \quad (16.108)$$

Since the iteration error cannot be exactly defined by this formula, we will approximate $\varepsilon_{IT}^{(k)}$ as

$$|\varepsilon_{IT}^{(k)}| \approx \frac{1}{2}\tau\|p^k\|_{L^2(\Omega)}^2,$$

or, more simply,

$$|\varepsilon_{IT}^{(k)}| \approx \|p^k\|_{L^2(\Omega)}^2,$$

which yields the usual stopping criterium

$$|\varepsilon_{IT}^{(k)}| \approx \|\delta u(p^k)\|_{L^2(\Omega)}. \quad (16.109)$$

In practice, $\varepsilon_{IT}^{(k)}$ is computed on the discrete variables, that is $|\varepsilon_{IT}^{(k)}| \approx \|\delta u_h(p_h^k)\|$. Should at a given iteration k the grid be adaptively refined, denoting with $\mathbf{x}_h = (y_h, p_h, u_h)$ the variables computed on the old grid (before the refinement) \mathcal{T}_h , and with $\mathbf{x}_{h,\text{ref}} = (y_h^{\text{ref}}, p_h^{\text{ref}}, u_h^{\text{ref}})$ those with the refined grid $\mathcal{T}_{h,\text{ref}}$. Then at the step k the discretization error associated with the grid $\mathcal{T}_{h,\text{ref}}$ is lower than that associated to \mathcal{T}_h . However, the discretization error $\varepsilon_{IT}^{(k)}$ computed on $\mathbf{x}_{h,\text{ref}}$, be lower than the iteration error computed on \mathbf{x}_h .

A posteriori error estimate and adaptive strategy. The a posteriori error estimate for the discretization error $\varepsilon_D^{(k)}$ can be characterized as follows ([DQ05]).

Theorem 16.5 For a linear advection-diffusion control problem at hand, with stabilized Lagrangian \mathcal{L}_h^s (Eq.(16.101) and Eq.(16.102)), the discretization error at step k of the iterative optimization algorithm can be written as

$$\varepsilon_D^{(k)} = \frac{1}{2}(\delta u(p^k, u^k), u^k - u_h^k) + \frac{1}{2}\nabla \mathcal{L}_h^s(\mathbf{x}_h^k) \cdot (\mathbf{x}^k - \mathbf{x}_h^k) + \Lambda_h(\mathbf{x}_h^k), \quad (16.110)$$

where $\mathbf{x}_h^k = (y_h^k, p_h^k, u_h^k)$ represents the GLS linear finite element solution and $\Lambda_h(\mathbf{x}_h^k) = S_h(\mathbf{x}_h^k) + s_h(y_h^k, p_h^k; u_h^k)$, being $s_h(w_h^k, p_h^k; u_h^k)$ the stabilization term (16.103).

By adapting (16.110) to the specific problem at hand and expressing the contributions on the different finite elements $K \in \mathcal{T}_h$ ([BKR00]), the following estimate can be obtained

$$|\varepsilon_D^{(k)}| \leq \eta_D^{(k)} = \frac{1}{2} \sum_{K \in \mathcal{T}_h} \{ (\omega_K^p \rho_K^y + \omega_K^y \rho_K^p + \omega_K^u \rho_K^u) + \lambda_K \}, \quad (16.111)$$

where:

$$\begin{aligned}
\rho_K^y &= \|R(y_h^k; u_h^k)\|_K + h_K^{-\frac{1}{2}} \|r(y_h^k)\|_{\partial K}, \\
\omega_K^p &= \|(p^k - p_h^k) - \delta_K L^{ad}(p^k - p_h^k) + \delta_K G'(y^k - y_h^k)\|_K + h_K^{\frac{1}{2}} \|p^k - p_h^k\|_{\partial K}, \\
\rho_K^p &= \|R^{ad}(p_h^k; y_h^k)\|_K + h_K^{-\frac{1}{2}} \|r^{ad}(p_h^k)\|_{\partial K}, \\
\omega_K^y &= \|(y^k - y_h^k) - \delta_K L(y^k - y_h^k)\|_K + h_K^{\frac{1}{2}} \|y^k - y_h^k\|_{\partial K}, \\
\omega_K^u &= \|\delta u_h(p_h^k, y_h^k) + \delta u(p^k)\|_K = \|p^k + p_h^k - \delta_K R^{ad}(p_h^k; y_h^k)\|_K, \\
\omega_K^u &= \|u^k - u_h^k\|_K, \\
\lambda_K &= 2\delta_K \|R(y_h^k; u_h^k)\|_K \|G(y_h^k)\|_K, \\
r(y_h^k) &= \begin{cases} -\frac{1}{2} \left[\mu \frac{\partial y_h^k}{\partial n} \right], & \text{on } \partial K \setminus \partial \Omega, \\ -\mu \frac{\partial y_h^k}{\partial n}, & \text{on } \partial K \in \Gamma_N, \end{cases} \\
r^{ad}(p_h^k) &= \begin{cases} -\frac{1}{2} \left[\mu \frac{\partial p_h^k}{\partial n} + \mathbf{V} \cdot \mathbf{n} p_h^k \right], & \text{on } \partial K \setminus \partial \Omega, \\ -\left(\mu \frac{\partial p_h^k}{\partial n} + \mathbf{V} \cdot \mathbf{n} p_h^k \right), & \text{on } \partial K \in \Gamma_N. \end{cases}
\end{aligned} \tag{16.112}$$

As usual, ∂K denotes the boundary of $K \in \mathcal{T}_h$, and $[\cdot]$ the jump operator across ∂K .

For a practical use of the estimate (16.111) it is necessary to evaluate y^k , p^k and u^k . At this scope we replace y^k and p^k by their quadratic reconstructions $(y_h^k)^q$ and $(p_h^k)^q$, while u^k is replaced by $(u_h^k)^q = u_h^k - \tau(\delta u_h((p_h^k)^q, (y_h^k)^q) - \delta u_h(p_h^k, y_h^k))$, according to the steepest descent method with constant acceleration parameter $\tau^k = \tau$. Consider the following adaptive strategy for the iterative optimization algorithm:

1. use a coarse grid and iterate until the tolerance on the iterative error is achieved Tol_{IT} ;
2. we adapt the grid, by equilibrating the error on the different elements $K \in \mathcal{T}_h$, according to the estimate $\eta_D^{(k)}$ (16.111), until convergence on the discretization error within a tolerance Tol_D ;
3. make a new evaluation of the variables as well as $\varepsilon_{IT}^{(k)}$ on the refined grid: return to point 1 and repeat the procedure if $\varepsilon_{IT}^{(k)} \geq Tol_{IT}$, otherwise stop the algorithm if $\varepsilon_{IT}^{(k)} < Tol_{IT}$.

16.13.3 A test problem on control of pollutant emission

As an example, we are going to apply the a posteriori estimate on the discretization error $\eta_D^{(k)}$ (16.111) and the strategy illustrated in Sec. 16.13.2 to a test case on emission of pollutants into the atmosphere. The specific problem is how to regulate the emission from industrial chimneys with the scope of keeping the pollutant concentration in a certain critical area below a prescribed admissible threshold.

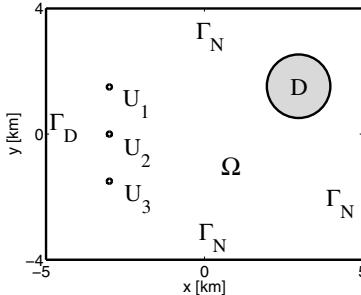


Fig. 16.16. Computational domain for the test problem on pollutant control

With this aim we consider a state equation that is given by a quasi-3D advection-diffusion boundary-value problem [DQ05]. The pollutant concentration y at the source (the emission height) $z = H$ is described by (16.84), whereas the concentration at ground is obtained by applying the projection function $g(x, y)$. The form of the diffusion coefficients $\mu(x, y)$ and by $g(x, y)$ depend from either the distance H of the source from the ground, and on class of atmospheric stability (stable, neutral or unstable). We will refer to a neutral atmosphere and, with reference to the domain illustrated in Fig.16.16, we assume that the wind field be $\mathbf{V} = V_x \hat{x} + V_y \hat{y}$, con $V_x = V \cos(\frac{\pi}{30})$ and $V_y = V \sin(\frac{\pi}{30})$, being $V = 2.5 \text{ m/s}$ the wind intensity. Moreover, we assume that there are three chimneys (represented by the three aligned small circles in Fig.16.16, all at the same height $H = 100 \text{ m}$, and that maximum discharge allowed from every chimney is $u_{max} = 800 \text{ g/s}$. We assume that the pollutant emitted be SO_2 and we fix at $z_d = 100 \mu\text{g/m}^3$ (the target observation in our control problem) the desired concentration in the region of observation, a circular region of the computational domain that we indicate by D in Fig.16.16. In (16.84) we have considered the case of a distributed control u on the whole computational domain Ω , whereas in the current example $u = \sum_{i=1}^N u_i \chi_i$, where χ_i is the characteristic function of the (tiny) region U_i which represents the location of the i -th chimney.

In Fig.16.17a we display the concentration at ground corresponding to the highest possible discharge ($u_{max} = 800 \text{ g/s}$) from each chimney, while in Fig.16.17b we display the concentration that we have after having applied the optimal control procedure (the cost functional being the square of the $L^2(D)$ norm of the distance from the target concentration z_d). We observe that the "optimal" emission rates become $u_1 = 0.0837 \cdot u_{max}$, $u_2 = 0.0908 \cdot u_{max}$ and $u_3 = 1.00 \cdot u_{max}$.

In Fig.16.18a we report the grid obtained by the a posteriori estimate on $\eta_D^{(k)}$, whereas in Fig.16.18b the one obtained by the following indicator *energy norm* error indicator $(\eta_E)^{(k)} = \sum_{K \in \mathcal{T}_h} h_K \rho_K^y$. For symbol definitions see (16.112).

These results are then compared with those that are obtained with a very fine grid of about 80000 elements. The grid adaptivity driven by the error indicator $\eta_D^{(k)}$ tends to concentrate nodes in those areas that are more relevant for the optimal control. This is confirmed by comparing the errors on the cost functional using the same number of gridpoints. For instance, the indicator $\eta_D^{(k)}$ yields an error of about 20%, against the

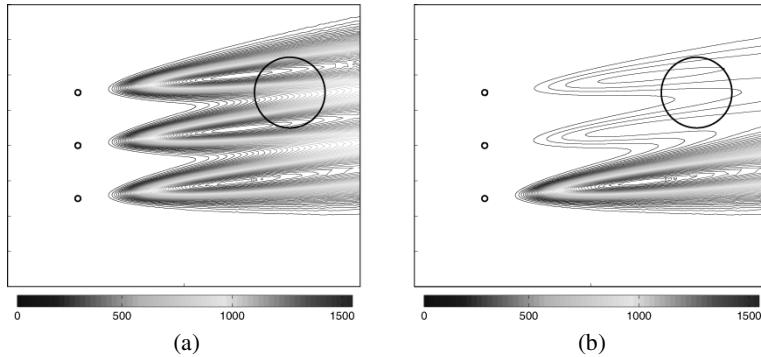


Fig. 16.17. Pollutant concentration measured in $\mu\text{g}/\text{m}^3$ at the ground before (a) after (b) regulating the emissions from the chimneys

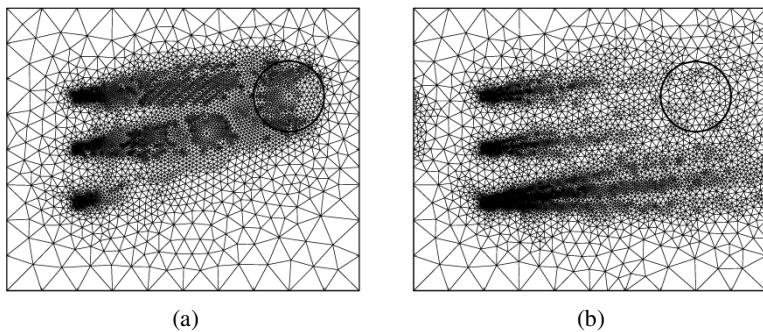


Fig. 16.18. Adapted grids (of about 14000 elements) obtained using the error indicator $\eta_D^{(j)}$ (see (16.111)) (a) and the one $(\eta_E)^{(j)}$

55% that would be obtained using the indicator $(\eta_E)^{(k)}$ using a grid of about 4000 elements, while on a grid of about 14000 elements it would be the 6% against the 15%.

16.14 Exercises

1. Consider the optimal control problem with boundary control

$$\begin{cases} -\nabla \cdot (\alpha \nabla y) + \beta \cdot \nabla y + \gamma y = f & \text{in } \Omega = (0, 1)^2, \\ \frac{\partial y}{\partial n} = u & \text{on } \partial\Omega, \end{cases} \quad (16.113)$$

being $u \in L^2(\Omega)$ the control function and $f \in L^2(\Omega)$ a given function. Consider the cost functional

$$J(u) = \frac{1}{2} \|\eta y - z_d\|_{L^2(\Omega)}^2 + \nu \|u\|_{L^2(\partial\Omega)}^2, \quad (16.114)$$

with $\eta \in L^\infty(\Omega)$.

Provide the equations (equation of state, adjoint equation and equation of optimality) of the optimal control problem (16.113)-(16.114) based on the Lagrangian approach, and those based on the Lions' approach.

2. Consider the optimal control problem

$$\begin{cases} -\nabla \cdot (\alpha \nabla y) + \beta \cdot \nabla y + \gamma y = f + c u & \text{in } \Omega = (0, 1)^2, \\ \frac{\partial y}{\partial n} = g & \text{on } \partial\Omega, \end{cases} \quad (16.115)$$

where $u \in L^2(\Omega)$ is a distributed control, c a given constant, $f \in L^2(\Omega)$ and $g \in H^{-1/2}(\partial\Omega)$ two given functions. Consider the cost functional

$$J(u) = \frac{1}{2} \|\eta y - z_d\|_{L^2(\Omega)}^2 + \nu \|u\|_{L^2(\Omega)}^2, \quad (16.116)$$

with $\eta \in L^\infty(\Omega)$.

Provide the formulation of the optimal control problem (16.115)-(16.116) by the Lagrangian based approach, then the one based on the Lions' formulation.

3. Reformulate the Dirichlet boundary control problem (see Table 16.1. down-left) as a Dirichlet distributed control problem (see Table 16.1, top-left), through a lifting $Gu \in H^1(\Omega)$ of the boundary data.

[*Solution:* Set $y = Gu + y^\circ$ where $G : H^{1/2}(\partial\Omega) \rightarrow H^1(\Omega)$ is the lifting operator, re-write the state equation as

$$\Delta y^\circ = f + Bu \quad \text{in } \Omega, \quad y^\circ = 0 \quad \text{on } \partial\Omega,$$

with the operator $B = -\Delta G$.]

Domain decomposition methods

In this chapter we will introduce the domain decomposition method (DD, in short). In its most common version, DD can be used in the framework of any discretization method for partial differential equations (such as, e.g. finite elements, finite volumes, finite differences, or spectral element methods) to make their algebraic solution more efficient on parallel computer platforms. In addition, DD methods allow the reformulation of any given boundary-value problem on a partition of the computational domain into subdomains. As such, it provides a very convenient framework for the solution of *heterogeneous* or multiphysics problems, i.e. those that are governed by differential equations of different kinds in different subregions of the computational domain.

The basic idea behind DD methods consists in subdividing the computational domain Ω , on which a boundary-value problem is set, into two or more subdomains on which discretized problems of smaller dimension are to be solved, with the further potential advantage of using parallel solution algorithms. More in particular, there are two ways of subdividing the computational domain into subdomains: one with disjoint subdomains, the others with overlapping subdomains (for an example, see Fig. 17.1). Correspondingly, different DD algorithms will be set up.

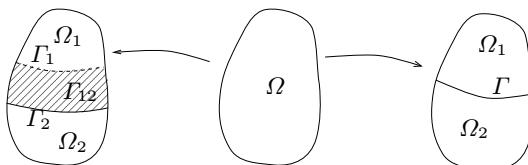


Fig. 17.1. Two examples of subdivision of the domain Ω , with and without overlap

For reference lectures on DD methods we refer to [BGS96, QV99, TW05].

17.1 Three classical iterative DD methods

In this section we introduce three different iterative schemes starting from the model problem: find $u : \Omega \rightarrow \mathbb{R}$ s.t.

$$\begin{cases} Lu = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \quad (17.1)$$

L being a generic second order elliptic operator. Its weak formulation reads

$$\text{find } u \in V = H_0^1(\Omega) : a(u, v) = (f, v) \quad \forall v \in V, \quad (17.2)$$

being $a(\cdot, \cdot)$ the bilinear form associated with L .

17.1.1 Schwarz method

Consider a decomposition of the domain Ω in two subdomains Ω_1 and Ω_2 s.t. $\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2$, $\Omega_1 \cap \Omega_2 = \Gamma_{12} \neq \emptyset$ (see Fig. 17.1) and let $\Gamma_i = \partial\Omega_i \setminus (\partial\Omega \cap \partial\Omega_i)$. Consider the following iterative method. Given $u_2^{(0)}$ on Γ_1 , solve the following problems for $k \geq 1$:

$$\begin{cases} Lu_1^{(k)} = f & \text{in } \Omega_1, \\ u_1^{(k)} = u_2^{(k-1)} & \text{on } \Gamma_1, \\ u_1^{(k)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma_1, \end{cases} \quad (17.3)$$

$$\begin{cases} Lu_2^{(k)} = f & \text{in } \Omega_2, \\ u_2^{(k)} = \begin{cases} u_1^{(k)} & \text{on } \Gamma_2, \\ u_1^{(k-1)} & \text{on } \Gamma_1, \end{cases} \\ u_2^{(k)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma_2. \end{cases} \quad (17.4)$$

In the case in which one chooses $u_1^{(k)}$ on Γ_2 in (17.4) the method is named *multiplicative Schwarz*, whereas that in which we choose $u_1^{(k-1)}$ is named *additive Schwarz*. The reason will be clarified in Sect. 17.6. We have thus two elliptic boundary-value problems with Dirichlet conditions for the two subdomains Ω_1 and Ω_2 , and we wish the two sequences $\{u_1^{(k)}\}$ and $\{u_2^{(k)}\}$ to converge to the restrictions of the solution u of problem (17.1), that is

$$\lim_{k \rightarrow \infty} u_1^{(k)} = u|_{\Omega_1} \quad \text{and} \quad \lim_{k \rightarrow \infty} u_2^{(k)} = u|_{\Omega_2}.$$

It can be proven that the Schwarz method applied to problem (17.1) always converges, with a rate that increases as the measure $|\Gamma_{12}|$ of the overlapping region Γ_{12} increases. Let us show this result on a simple one-dimensional case.

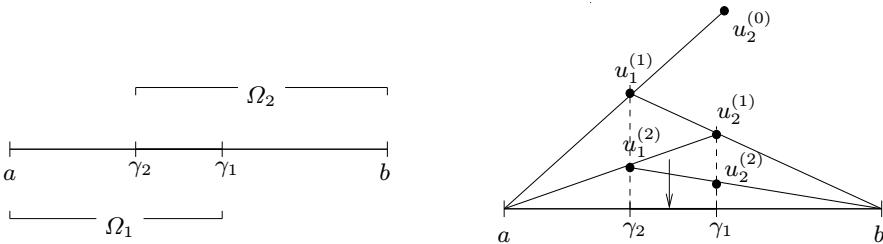


Fig. 17.2. Example of a decomposition with overlap in dimension 1 (left). A few iterations of the multiplicative Schwarz method for problem (17.7) (right)

Example 17.1 Let $\Omega = (a, b)$ and let $\gamma_1, \gamma_2 \in (a, b)$ be such that $a < \gamma_2 < \gamma_1 < b$ (see Fig. 17.2). The two problems (17.3) and (17.4) become:

$$\begin{cases} Lu_1^{(k)} = f, & a < x < \gamma_1, \\ u_1^{(k)} = u_2^{(k-1)}, & x = \gamma_1, \\ u_1^{(k)} = 0, & x = a, \end{cases} \quad (17.5)$$

$$\begin{cases} Lu_2^{(k)} = f, & \gamma_2 < x < b, \\ u_2^{(k)} = u_1^{(k)}, & x = \gamma_2, \\ u_2^{(k)} = 0, & x = b. \end{cases} \quad (17.6)$$

To show that this scheme converges, let us bound ourselves to the simpler problem

$$\begin{cases} -u''(x) = 0, & a < x < b, \\ u(a) = u(b) = 0, \end{cases} \quad (17.7)$$

that is the model problem (17.1) with $L = -d^2/dx^2$ and $f = 0$, whose solution clearly is $u = 0$ in (a, b) . This is not restrictive since at every step the error: $u - u_1^{(k)}$ in Ω_1 , $u - u_2^{(k)}$ in Ω_2 , satisfies a problem like (17.5)-(17.6) with null forcing term.

Let $k = 1$; since $(u_1^{(1)})'' = 0$, $u_1^{(1)}(x)$ is a linear function; moreover, it vanishes at $x = a$ and takes the value $u_2^{(0)}$ at $x = \gamma_1$. As we know the value of $u_1^{(1)}$ at γ_2 , we can solve the problem (17.6) which, in its turn, features a linear solution. Then we proceed in a similar manner. In Fig. 17.2 we show a few iterations: we clearly see that the method converges, moreover the convergence rate reduces as the length of the interval (γ_2, γ_1) gets smaller. ■

The Schwarz iterative method (17.3)-(17.4) requires at each iteration the solution of two subproblems with boundary conditions of the same kind as those of the original problem: indeed, by starting with a Dirichlet boundary-value problem in Ω we end up with two subproblems with Dirichlet conditions on the boundary of Ω_1 and Ω_2 .

Should the differential problem (17.1) had been completed by a Neumann boundary condition on the whole boundary $\partial\Omega$, we would have been led to the solution of a mixed Dirichlet-Neumann boundary-value problem on either subdomain Ω_1 and Ω_2 .

17.1.2 Dirichlet-Neumann method

Let us partition the domain Ω in two disjoint subdomains (as in Fig. 17.1): let then Ω_1 and Ω_2 be two subdomains providing a partition of Ω , i.e. $\overline{\Omega}_1 \cup \overline{\Omega}_2 = \overline{\Omega}$, $\overline{\Omega}_1 \cap \overline{\Omega}_2 = \Gamma$ and $\Omega_1 \cap \Omega_2 = \emptyset$. We denote by \mathbf{n}_i the outward unit normal vector to Ω_i and will use the following notational convention: $\mathbf{n} = \mathbf{n}_1 = -\mathbf{n}_2$.

The following result holds (for its proof see [QV99]):

Theorem 17.1 (of equivalence) *The solution u of problem (17.1) is such that $u|_{\Omega_i} = u_i$ for $i = 1, 2$, where u_i is the solution to the problem*

$$\begin{cases} Lu_i = f & \text{in } \Omega_i, \\ u_i = 0 & \text{on } \partial\Omega_i \setminus \Gamma, \end{cases} \quad (17.8)$$

with interface conditions

$$u_1 = u_2 \quad (17.9)$$

and

$$\frac{\partial u_1}{\partial n_L} = \frac{\partial u_2}{\partial n_L} \quad (17.10)$$

on Γ , having denoted with $\partial/\partial n_L$ the conormal derivative (see (3.34)).

Thanks to this result we can split problem (17.1) by assigning the interface conditions (17.9)-(17.10) the role of “boundary conditions” for the two subproblems on the interface Γ . In particular, we can set up the following *Dirichlet-Neumann* (DN) iterative algorithm : given $u_2^{(0)}$ on Γ , for $k \geq 1$ solve the problems:

$$\begin{cases} Lu_1^{(k)} = f & \text{in } \Omega_1, \\ u_1^{(k)} = u_2^{(k-1)} & \text{on } \Gamma, \\ u_1^{(k)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma, \end{cases} \quad (17.11)$$

$$\begin{cases} Lu_2^{(k)} = f & \text{in } \Omega_2, \\ \frac{\partial u_2^{(k)}}{\partial n} = \frac{\partial u_1^{(k)}}{\partial n} & \text{on } \Gamma, \\ u_2^{(k)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma. \end{cases} \quad (17.12)$$

Condition (17.9) has generated a Dirichlet boundary condition on Γ for the subproblem in Ω_1 whereas (17.10) has generated a Neumann boundary condition on Γ for the subproblem in Ω_2 .

Differently than Schwarz's method, the DN algorithm yields a Neumann boundary-value problem on the subdomain Ω_2 . The equivalence theorem 17.1 guarantees that when the two sequences $\{u_1^{(k)}\}$ and $\{u_2^{(k)}\}$ converge, then their limit will be the solution to the exact problem (17.1). The DN algorithm is therefore *consistent*. Its

convergence however is not always guaranteed, as we can see on the following simple example.

Example 17.2 Let $\Omega = (a, b)$, $\gamma \in (a, b)$, $L = -d^2/dx^2$ and $f = 0$. At every $k \geq 1$ the DN algorithm generates the two subproblems:

$$\begin{cases} -(u_1^{(k)})'' = 0, & a < x < \gamma, \\ u_1^{(k)} = 0, & x = a, \\ u_1^{(k)} = u_2^{(k-1)}, & x = \gamma, \end{cases} \quad (17.13)$$

$$\begin{cases} -(u_2^{(k)})'' = 0, & \gamma < x < b, \\ (u_2^{(k)})' = (u_1^{(k)})', & x = \gamma, \\ u_2^{(k)} = 0, & x = b. \end{cases} \quad (17.14)$$

Proceeding as done in Example 17.1, we can prove that the two sequences converge only if $\gamma > (a + b)/2$, as shown graphically in Fig. 17.3. ■

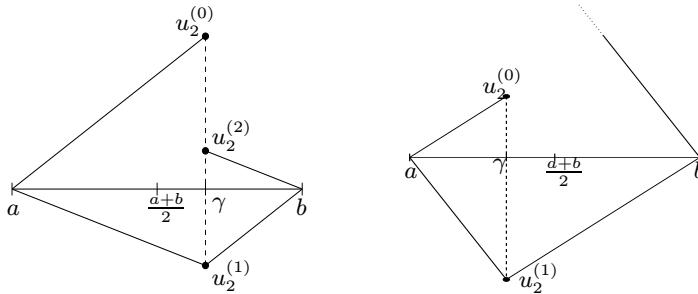


Fig. 17.3. Example of converging (left) and diverging (right) iterations for the DN method in 1D

In general, for a problem in arbitrary dimension $d > 1$, the measure of the “Dirichlet” subdomain Ω_1 must be larger than that of the “Neumann” one Ω_2 in order to guarantee the convergence of (17.11)-(17.12). This however yields a severe constraint to fulfill, especially if several subdomains will be used.

To overcome such limitation, a variant of the DN algorithm can be set up by replacing the Dirichlet condition (17.11)₂ in the first subdomain by

$$u_1^{(k)} = \theta u_2^{(k-1)} + (1 - \theta)u_1^{(k-1)} \quad \text{on } \Gamma, \quad (17.15)$$

that is by introducing a *relaxation* which depends on a positive parameter θ . In such a way it is always possible to reduce the error between two subsequent iterates.

In the case displayed in Fig. 17.3 we can easily verify that, by choosing

$$\theta_{opt} = -\frac{u_1^{(k-1)}}{u_2^{(k-1)} - u_1^{(k-1)}}, \quad (17.16)$$

the algorithm converges to the exact solution in a single iteration.

More in general, it can be proven that in any dimension $d \geq 1$, there exists a suitable value $\theta_{\max} < 1$ such that the DN algorithm converges for any possible choice of the relaxation parameter θ in the interval $(0, \theta_{\max})$.

17.1.3 Neumann-Neumann algorithm

Consider again a partition of Ω into two disjoint subdomains and denote by λ the (unknown) value of the solution u at their interface Γ . Consider the following iterative algorithm: for any given $\lambda^{(0)}$ on Γ , for $k \geq 0$ and $i = 1, 2$ solve the following problems:

$$\begin{cases} -\Delta u_i^{(k+1)} = f & \text{in } \Omega_i, \\ u_i^{(k+1)} = \lambda^{(k)} & \text{on } \Gamma, \\ u_i^{(k+1)} = 0 & \text{on } \partial\Omega_i \setminus \Gamma, \end{cases} \quad (17.17)$$

$$\begin{cases} -\Delta \psi_i^{(k+1)} = 0 & \text{in } \Omega_i, \\ \frac{\partial \psi_i^{(k+1)}}{\partial n} = \frac{\partial u_1^{(k+1)}}{\partial n} - \frac{\partial u_2^{(k+1)}}{\partial n} & \text{on } \Gamma, \\ \psi_i^{(k+1)} = 0 & \text{on } \partial\Omega_i \setminus \Gamma, \end{cases} \quad (17.18)$$

with

$$\lambda^{(k+1)} = \lambda^{(k)} - \theta \left(\sigma_1 \psi_{1|\Gamma}^{(k+1)} - \sigma_2 \psi_{2|\Gamma}^{(k+1)} \right), \quad (17.19)$$

where θ is a positive acceleration parameter, while σ_1 and σ_2 are two positive coefficients. This iterative algorithm is named *Neumann-Neumann* (NN). Note that in the first stage (17.17) we care about the continuity on Γ of the functions $u_1^{(k+1)}$ and $u_2^{(k+1)}$ but not that of their derivatives. The latter are addressed in the second stage (17.18), (17.19) by means of the correcting functions $\psi_1^{(k+1)}$ and $\psi_2^{(k+1)}$.

17.1.4 Robin-Robin algorithm

At last, we consider the following iterative algorithm, named *Robin-Robin* (RR). For every $k \geq 0$ solve the following problems:

$$\begin{cases} -\Delta u_1^{(k+1)} = f & \text{in } \Omega_1, \\ u_1^{(k+1)} = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega, \\ \frac{\partial u_1^{(k+1)}}{\partial n} + \gamma_1 u_1^{(k+1)} = \frac{\partial u_2^{(k)}}{\partial n} + \gamma_1 u_2^{(k)} & \text{on } \Gamma, \end{cases} \quad (17.20)$$

then

$$\begin{cases} -\Delta u_2^{(k+1)} = f & \text{in } \Omega_2, \\ u_2^{(k+1)} = 0 & \text{on } \partial\Omega_2 \cap \partial\Omega, \\ \frac{\partial u_2^{(k+1)}}{\partial n} + \gamma_2 u_2^{(k+1)} = \frac{\partial u_1^{(k+1)}}{\partial n} + \gamma_2 u_1^{(k+1)} & \text{on } \Gamma, \end{cases} \quad (17.21)$$

where u_0 is assigned and γ_1, γ_2 are non-negative acceleration parameters that satisfy $\gamma_1 + \gamma_2 > 0$. Aiming at the algorithm parallelization, in (17.21) we could use $u_1^{(k)}$ instead of $u_1^{(k+1)}$, provided in such a case an initial value for u_1^0 is assigned as well.

17.2 Multi-domain formulation of Poisson problem and interface conditions

In this section, for the sake of exposition, we choose $L = -\Delta$ and consider the Poisson problem with homogeneous Dirichlet boundary conditions (3.13). Generalization to an arbitrary second order elliptic operator with different boundary conditions is in order.

In the case addressed in Sec. 17.1.2 of a domain partitioned into two disjoint sub-domains, the equivalence Theorem 17.1 allows the following *multidomain formulation* of problem (17.1), in which $u_i = u|_{\Omega_i}$, $i = 1, 2$:

$$\begin{cases} -\Delta u_1 = f & \text{in } \Omega_1, \\ u_1 = 0 & \text{on } \partial\Omega_1 \setminus \Gamma, \\ -\Delta u_2 = f & \text{in } \Omega_2, \\ u_2 = 0 & \text{on } \partial\Omega_2 \setminus \Gamma, \\ u_1 = u_2 & \text{on } \Gamma, \\ \frac{\partial u_1}{\partial n} = \frac{\partial u_2}{\partial n} & \text{on } \Gamma. \end{cases} \quad (17.22)$$

17.2.1 The Steklov-Poincaré operator

We denote again by λ the unknown value of the solution u of problem (3.13) on the interface Γ , that is $\lambda = u|_{\Gamma}$. Should we know a priori the value λ on Γ , we could solve the following two independent boundary-value problems with Dirichlet condition on Γ ($i = 1, 2$):

$$\begin{cases} -\Delta w_i = f & \text{in } \Omega_i, \\ w_i = 0 & \text{on } \partial\Omega_i \setminus \Gamma, \\ w_i = \lambda & \text{on } \Gamma. \end{cases} \quad (17.23)$$

With the aim of obtaining the value λ on Γ , let us split w_i as follows

$$w_i = w_i^* + u_i^0,$$

where w_i^* and u_i^0 represent the solutions of the following problems ($i = 1, 2$):

$$\begin{cases} -\Delta w_i^* = f & \text{in } \Omega_i, \\ w_i^* = 0 & \text{on } \partial\Omega_i \cap \partial\Omega, \\ w_i^* = 0 & \text{on } \Gamma, \end{cases} \quad (17.24)$$

and

$$\begin{cases} -\Delta u_i^0 = 0 & \text{in } \Omega_i, \\ u_i^0 = 0 & \text{on } \partial\Omega_i \cap \partial\Omega, \\ u_i^0 = \lambda & \text{on } \Gamma, \end{cases} \quad (17.25)$$

respectively. Note that the functions w_i^* depend solely on the source data f , while u_i^0 solely on the value λ on Γ , henceforth we can write $w_i^* = G_i f$ and $u_i^0 = H_i \lambda$. Both operators G_i and H_i are linear; H_i is the so-called harmonic extension operator of λ on the domain Ω_i .

By a formal comparison of problem (17.22) with problem (17.23), we infer that the equality

$$u_i = w_i^* + u_i^0, \quad i = 1, 2,$$

holds iff the condition (17.22)₆ on the normal derivatives on Γ is satisfied, that is iff

$$\frac{\partial w_1}{\partial n} = \frac{\partial w_2}{\partial n} \quad \text{on } \Gamma.$$

By using the previously introduced notations the latter condition can be reformulated as

$$\frac{\partial}{\partial n}(G_1 f + H_1 \lambda) = \frac{\partial}{\partial n}(G_2 f + H_2 \lambda),$$

and therefore

$$\left(\frac{\partial H_1}{\partial n} - \frac{\partial H_2}{\partial n} \right) \lambda = \left(\frac{\partial G_2}{\partial n} - \frac{\partial G_1}{\partial n} \right) f \quad \text{on } \Gamma.$$

In this way we have obtained an equation for the unknown λ on the interface Γ , named *Steklov-Poincaré equation*, that can be rewritten in compact form as

$$S\lambda = \chi \quad \text{on } \Gamma. \quad (17.26)$$

S is the *Steklov-Poincaré* pseudo-differential operator; its formal definition is

$$S\mu = \frac{\partial}{\partial n} H_1 \mu - \frac{\partial}{\partial n} H_2 \mu = \sum_{i=1}^2 \frac{\partial}{\partial n_i} H_i \mu = \sum_{i=1}^2 S_i \mu, \quad (17.27)$$

while χ is a linear functional which depends on f

$$\chi = \frac{\partial}{\partial n} G_2 f - \frac{\partial}{\partial n} G_1 f = - \sum_{i=1}^2 \frac{\partial}{\partial n_i} G_i f. \quad (17.28)$$

The operator

$$S_i : \mu \rightarrow S_i \mu = \frac{\partial}{\partial n_i} (H_i \mu) \Big|_{\Gamma}, \quad i = 1, 2, \quad (17.29)$$

is called local Steklov-Poincaré operator. Note that S , S_1 and S_2 operate between the trace space

$$\Lambda = \{ \mu \mid \exists v \in V : \mu = v|_{\Gamma} \} \quad (17.30)$$

(that is $H_{00}^{1/2}(\Gamma)$, see [QV99]), and its dual Λ' , whereas $\chi \in \Lambda'$.

Example 17.3 With the aim of providing a practical (elementary) example of operator S , let us consider a simple one-dimensional problem. Let $\Omega = (a, b) \subset \mathbb{R}$ as shown in Fig. 17.4 and $Lu = -u''$. By subdividing Ω in two disjoint subdomains, the interface Γ reduces to a single point $\gamma \in (a, b)$, and the Steklov-Poincaré operator S becomes

$$S\lambda = \left(\frac{dH_1}{dx} - \frac{dH_2}{dx} \right) \lambda = \left(\frac{1}{l_1} + \frac{1}{l_2} \right) \lambda,$$

with $l_1 = \gamma - a$ and $l_2 = b - \gamma$. ■

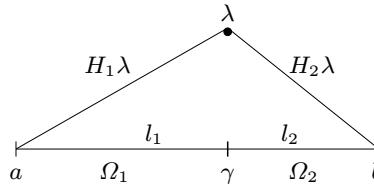


Fig. 17.4. Harmonic extensions in one dimension

17.2.2 Equivalence between Dirichlet-Neumann and Richardson methods

The Dirichlet-Neumann (DN) method introduced in Sec. 17.1.2 can be reinterpreted as a (preconditioned) Richardson method for the solution of the Steklov-Poincaré interface equation. To check this statement, consider again, for the sake of simplicity, a domain Ω partitioned into two disjoint subdomains Ω_1 and Ω_2 with interface Γ .

Then we re-write the DN algorithm (17.11), (17.12), (17.15) in the case of the operator $L = -\Delta$: for a given λ^0 , for $k \geq 1$ solve:

$$\begin{cases} -\Delta u_1^{(k)} = f_1 & \text{in } \Omega_1, \\ u_1^{(k)} = \lambda^{(k-1)} & \text{on } \Gamma, \\ u_1^{(k)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma, \end{cases} \quad (17.31)$$

$$\begin{cases} -\Delta u_2^{(k)} = f_2 & \text{in } \Omega_2, \\ \frac{\partial u_2^{(k)}}{\partial n_2} = \frac{\partial u_1^{(k)}}{\partial n_2} & \text{on } \Gamma, \\ u_2^{(k)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma, \end{cases} \quad (17.32)$$

$$\lambda^{(k)} = \theta u_2^{(k)}|_{\Gamma} + (1 - \theta) \lambda^{(k-1)}. \quad (17.33)$$

The following result holds:

Theorem 17.2 *The Dirichlet-Neumann iterative algorithm (17.31)-(17.33) is equivalent to the preconditioned Richardson algorithm*

$$P_{DN}(\lambda^{(k)} - \lambda^{(k-1)}) = \theta(\chi - S\lambda^{(k-1)}). \quad (17.34)$$

The preconditioning operator is $P_{DN} = S_2 = \partial(H_2\mu)/\partial n_2$.

Proof. The solution $u_1^{(k)}$ of (17.31) can be written as

$$u_1^{(k)} = H_1 \lambda^{(k-1)} + G_1 f_1. \quad (17.35)$$

Since $G_2 f_2$ satisfies the differential problem

$$\begin{cases} -\Delta(G_2 f_2) = f_2 & \text{in } \Omega_2, \\ G_2 f_2 = 0 & \text{on } \partial\Omega_2, \end{cases}$$

thanks to (17.32) the function $u_2^{(k)} - G_2 f_2$ satisfies the differential problem

$$\begin{cases} -\Delta(u_2^{(k)} - G_2 f_2) = 0 & \text{in } \Omega_2, \\ \frac{\partial}{\partial n_2}(u_2^{(k)} - G_2 f_2) = -\frac{\partial u_1^{(k)}}{\partial n} + \frac{\partial}{\partial n}(G_2 f_2) & \text{on } \Gamma, \\ u_2^{(k)} - G_2 f_2 = 0 & \text{on } \partial\Omega_2 \setminus \Gamma. \end{cases} \quad (17.36)$$

In particular $u_2^{(k)}|_{\Gamma} = (u_2^{(k)} - G_2 f_2)|_{\Gamma}$. Since the operator S_i (17.29) maps a Dirichlet data to a Neumann data on Γ , its inverse S_i^{-1} transforms a Neumann data in a Dirichlet one on Γ .

Otherwise said, $S_2^{-1}\eta = w_2|_{\Gamma}$, where w_2 is the solution of

$$\begin{cases} -\Delta w_2 = 0 & \text{in } \Omega_2, \\ \frac{\partial w_2}{\partial n} = \eta & \text{on } \Gamma, \\ w_2 = 0 & \text{on } \partial\Omega_2 \setminus \Gamma. \end{cases} \quad (17.37)$$

Setting now

$$\eta = -\frac{\partial u_1^{(k)}}{\partial n} + \frac{\partial}{\partial n}(G_2 f_2),$$

and comparing (17.36) with (17.37), we conclude that

$$u_2^{(k)}|_{\Gamma} = (u_2^{(k)} - G_2 f_2)|_{\Gamma} = S_2^{-1} \left(-\frac{\partial u_1^{(k)}}{\partial n} + \frac{\partial}{\partial n}(G_2 f_2) \right).$$

On the other hand, owing to (17.35) and to the definition (17.28) of χ , we obtain

$$\begin{aligned} u_2^{(k)}|_{\Gamma} &= S_2^{-1} \left(-\frac{\partial}{\partial n}(H_1 \lambda^{(k-1)}) - \frac{\partial}{\partial n}(G_1 f_1) + \frac{\partial}{\partial n}(G_2 f_2) \right) \\ &= S_2^{-1}(-S_1 \lambda^{(k-1)} + \chi). \end{aligned}$$

Using (17.33) we can therefore write

$$\lambda^{(k)} = \theta \left[S_2^{-1}(-S_1 \lambda^{(k-1)} + \chi) \right] + (1 - \theta) \lambda^{(k-1)},$$

that is

$$\lambda^{(k)} - \lambda^{(k-1)} = \theta \left[S_2^{-1}(-S_1 \lambda^{(k-1)} + \chi) - \lambda^{(k-1)} \right].$$

Since $-S_1 = S_2 - S$, we finally obtain

$$\begin{aligned} \lambda^{(k)} - \lambda^{(k-1)} &= \theta \left[S_2^{-1}((S_2 - S)\lambda^{(k-1)} + \chi) - \lambda^{(k-1)} \right] \\ &= \theta S_2^{-1}(\chi - S\lambda^{(k-1)}), \end{aligned}$$

that is (17.34). \diamond

Using an argument similar to that used for the proof of Theorem 17.2, the Neumann-Neumann algorithm (17.17) - (17.19) can also be interpreted as a preconditioned Richardson algorithm

$$P_{NN}(\lambda^{(k)} - \lambda^{(k-1)}) = \theta(\chi - S\lambda^{(k-1)}),$$

this time however the preconditioner being $P_{NN} = (\sigma_1 S_1^{-1} + \sigma_2 S_2^{-1})^{-1}$.

Consider at last the Robin-Robin iterative algorithm (17.20) - (17.21). Denoting by $\mu_i^{(k)} \in \Lambda$ the approximation at step k of the trace of $u_i^{(k)}$ on the interface Γ , $i = 1, 2$, it can be proven that (17.20) - (17.21) is equivalent to the following alternating direction (ADI) algorithm:

$$\begin{aligned} (\gamma_1 i_{\Lambda} + S_1) \mu_1^{(k)} &= \chi + (\gamma_1 i_{\Lambda} + S_2) \mu_2^{(k-1)}, \\ (\gamma_2 i_{\Lambda} + S_2) \mu_2^{(k)} &= \chi + (\gamma_2 i_{\Lambda} + S_1) \mu_1^{(k-1)}, \end{aligned}$$

where $i_{\Lambda} : \Lambda \rightarrow \Lambda'$ here denotes the Riesz isomorphism between the Hilbert space Λ and its dual Λ' (see (2.5)).

Should, for a convenient choice of the two parameters γ_1 and γ_2 , the algorithm converge to two limit functions μ_1 and μ_2 , then $\mu_1 = \mu_2 = \lambda$, the latter function being the solution to the Steklov-Poincaré equation (17.26).

17.3 Multidomain formulation of the finite element approximation of the Poisson problem

What seen thus far can be regarded as propedeutical to numerical solution of boundary-value problems. In this section we will see how the previous ideas can be reshaped in the framework of a numerical discretization method. Although we will only address the case of finite element discretization, this is however not restrictive. We refer, e.g., to [CHQZ07] for the case of spectral or spectral element discretizations.

Consider the Poisson problem (3.13), its weak formulation (3.18) and its Galerkin finite element approximation (4.40) on a triangulation \mathcal{T}_h . Recall that $V_h = \overset{\circ}{X}_h^r = \{v_h \in X_h^r : v_h|_{\partial\Omega} = 0\}$ is the space of finite element functions of degree r vanishing on $\partial\Omega$, whose basis is $\{\varphi_j\}_{j=1}^{N_h}$ (see Sec. 4.5.1).

For the finite element nodes in the domain Ω we consider the following partition: let $\{x_j^{(1)}, 1 \leq j \leq N_1\}$ be the nodes located in the subdomain Ω_1 , $\{x_j^{(2)}, 1 \leq j \leq N_2\}$ those in Ω_2 and, finally, $\{x_j^{(\Gamma)}, 1 \leq j \leq N_\Gamma\}$ those lying on the interface Γ . Let us split the basis functions accordingly: $\varphi_j^{(1)}$ will denote those associated to the nodes $x_j^{(1)}$, $\varphi_j^{(2)}$ those associated with the nodes $x_j^{(2)}$, and $\varphi_j^{(\Gamma)}$ those associated with the nodes $x_j^{(\Gamma)}$ lying on the interface. This yields

$$\varphi_j^{(\alpha)}(x_i^{(\beta)}) = \delta_{ij}\delta_{\alpha\beta}, \quad 1 \leq i \leq N_\alpha, \quad 1 \leq j \leq N_\beta, \quad (17.38)$$

with $\alpha, \beta = 1, 2, \Gamma$; δ_{ij} is the Kronecker symbol.

By letting v_h in (4.40) to coincide with a test function, (4.40) can be given the following equivalent formulation: find $u_h \in V_h$ s.t.

$$\begin{cases} a(u_h, \varphi_i^{(1)}) = F(\varphi_i^{(1)}) & \forall i = 1, \dots, N_1, \\ a(u_h, \varphi_j^{(2)}) = F(\varphi_j^{(2)}) & \forall j = 1, \dots, N_2, \\ a(u_h, \varphi_k^{(\Gamma)}) = F(\varphi_k^{(\Gamma)}) & \forall k = 1, \dots, N_\Gamma, \end{cases} \quad (17.39)$$

having set $F(v) = \int_\Omega fv d\Omega$. Let now

$$a_i(v, w) = \int_{\Omega_i} \nabla v \cdot \nabla w d\Omega \quad \forall v, w \in V, i = 1, 2$$

be the restriction of the bilinear form $a(., .)$ to the subdomain Ω_i and define $V_{i,h} = \{v \in H^1(\Omega_i) \mid v = 0 \text{ on } \partial\Omega_i \setminus \Gamma\}$ ($i = 1, 2$). Similarly we set $F_i(v) = \int_{\Omega_i} fv d\Omega$ and denote by $u_h^{(i)} = u_h|_{\Omega_i}$ the restriction of u_h to the subdomain Ω_i , with $i = 1, 2$. Problem (17.39) can be rewritten in the equivalent form: find $u_h^{(1)} \in V_{1,h}$, $u_h^{(2)} \in V_{2,h}$ such that

$$\begin{cases} a_1(u_h^{(1)}, \varphi_i^{(1)}) = F_1(\varphi_i^{(1)}) & \forall i = 1, \dots, N_1, \\ a_2(u_h^{(2)}, \varphi_j^{(2)}) = F_2(\varphi_j^{(2)}) & \forall j = 1, \dots, N_2 \\ a_1(u_h^{(1)}, \varphi_k^{(\Gamma)}|_{\Omega_1}) + a_2(u_h^{(2)}, \varphi_k^{(\Gamma)}|_{\Omega_2}) \\ = F_1(\varphi_k^{(\Gamma)}|_{\Omega_1}) + F_2(\varphi_k^{(\Gamma)}|_{\Omega_2}) & \forall k = 1, \dots, N_\Gamma. \end{cases} \quad (17.40)$$

The interface continuity condition (17.22)₅ is automatically satisfied thanks to the continuity of the functions $u_h^{(i)}$. Moreover, equations (17.40)₁-(17.40)₃ correspond to the finite element discretization of equations (17.22)₁-(17.22)₆, respectively. In particular, the third of equations (17.40) must be regarded as the discrete counterpart of condition (17.22)₆ expressing the continuity of normal derivatives on Γ .

Let us expand the solution u_h with respect to the basis functions V_h

$$\begin{aligned} u_h(x) &= \sum_{j=1}^{N_1} u_h(x_j^{(1)}) \varphi_j^{(1)}(x) + \sum_{j=1}^{N_2} u_h(x_j^{(2)}) \varphi_j^{(2)}(x) \\ &+ \sum_{j=1}^{N_\Gamma} u_h(x_j^{(\Gamma)}) \varphi_j^{(\Gamma)}(x). \end{aligned} \quad (17.41)$$

From now on, the nodal values $u_h(x_j^{(\alpha)})$, for $\alpha = 1, 2, \Gamma$ and $j = 1, \dots, N_\alpha$, which are the expansion coefficients, will be indicated with the shorthand notation $u_j^{(\alpha)}$.

Using (17.41), we can rewrite (17.40) as follows:

$$\begin{cases} \sum_{j=1}^{N_1} u_j^{(1)} a_1(\varphi_j^{(1)}, \varphi_i^{(1)}) + \sum_{j=1}^{N_\Gamma} u_j^{(\Gamma)} a_1(\varphi_j^{(\Gamma)}, \varphi_i^{(1)}) = F_1(\varphi_i^{(1)}) & \forall i = 1, \dots, N_1, \\ \sum_{j=1}^{N_2} u_j^{(2)} a_2(\varphi_j^{(2)}, \varphi_i^{(2)}) + \sum_{j=1}^{N_\Gamma} u_j^{(\Gamma)} a_2(\varphi_j^{(\Gamma)}, \varphi_i^{(2)}) = F_2(\varphi_i^{(2)}) & \forall i = 1, \dots, N_2, \\ \sum_{j=1}^{N_\Gamma} u_j^{(\Gamma)} [a_1(\varphi_j^{(\Gamma)}, \varphi_i^{(\Gamma)}) + a_2(\varphi_j^{(\Gamma)}, \varphi_i^{(\Gamma)})] \\ + \sum_{j=1}^{N_1} u_j^{(1)} a_1(\varphi_j^{(1)}, \varphi_i^{(\Gamma)}) + \sum_{j=1}^{N_2} u_j^{(2)} a_2(\varphi_j^{(2)}, \varphi_i^{(\Gamma)}) \\ = F_1(\varphi_i^{(\Gamma)}|_{\Omega_1}) + F_2(\varphi_i^{(\Gamma)}|_{\Omega_2}) & \forall i = 1, \dots, N_\Gamma. \end{cases} \quad (17.42)$$

Let us introduce the following arrays:

$$\begin{aligned} (A_{11})_{ij} &= a_1(\varphi_j^{(1)}, \varphi_i^{(1)}), & (A_{1\Gamma})_{ij} &= a_1(\varphi_j^{(\Gamma)}, \varphi_i^{(1)}), \\ (A_{22})_{ij} &= a_2(\varphi_j^{(2)}, \varphi_i^{(2)}), & (A_{2\Gamma})_{ij} &= a_2(\varphi_j^{(\Gamma)}, \varphi_i^{(2)}), \\ (A_{\Gamma\Gamma})_{ij} &= a_1(\varphi_j^{(\Gamma)}, \varphi_i^{(\Gamma)}), & (A_{\Gamma\Gamma})_{ij} &= a_2(\varphi_j^{(\Gamma)}, \varphi_i^{(\Gamma)}), \\ (A_{\Gamma 1})_{ij} &= a_1(\varphi_j^{(1)}, \varphi_i^{(\Gamma)}), & (A_{\Gamma 2})_{ij} &= a_2(\varphi_j^{(2)}, \varphi_i^{(\Gamma)}), \\ (\mathbf{f}_1)_i &= F_1(\varphi_i^{(1)}), & (\mathbf{f}_2)_i &= F_2(\varphi_i^{(2)}), \\ (\mathbf{f}_1^\Gamma)_i &= F_1(\varphi_i^{(\Gamma)}), & (\mathbf{f}_2^\Gamma)_i &= F_2(\varphi_i^{(\Gamma)}), \end{aligned}$$

then set

$$\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \boldsymbol{\lambda})^T, \text{ with } \mathbf{u}_1 = \left(u_j^{(1)} \right), \mathbf{u}_2 = \left(u_j^{(2)} \right) \text{ and } \boldsymbol{\lambda} = \left(u_j^{(\Gamma)} \right). \quad (17.43)$$

Problem (17.42) can be casted in the following algebraic form

$$\begin{cases} A_{11}\mathbf{u}_1 + A_{1\Gamma}\boldsymbol{\lambda} = \mathbf{f}_1, \\ A_{22}\mathbf{u}_2 + A_{2\Gamma}\boldsymbol{\lambda} = \mathbf{f}_2, \\ A_{\Gamma 1}\mathbf{u}_1 + A_{\Gamma 2}\mathbf{u}_2 + \left(A_{\Gamma\Gamma}^{(1)} + A_{\Gamma\Gamma}^{(2)} \right) \boldsymbol{\lambda} = \mathbf{f}_1^\Gamma + \mathbf{f}_2^\Gamma, \end{cases} \quad (17.44)$$

or, equivalently,

$$A\mathbf{u} = \mathbf{f}, \text{ that is } \begin{bmatrix} A_{11} & 0 & A_{1\Gamma} \\ 0 & A_{22} & A_{2\Gamma} \\ A_{\Gamma 1} & A_{\Gamma 2} & A_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_\Gamma \end{bmatrix}, \quad (17.45)$$

having set $A_{\Gamma\Gamma} = \left(A_{\Gamma\Gamma}^{(1)} + A_{\Gamma\Gamma}^{(2)} \right)$ and $\mathbf{f}_\Gamma = \mathbf{f}_1^\Gamma + \mathbf{f}_2^\Gamma$. (17.45) is nothing but a block-wise representation of the finite element system (4.46), the blocks being determined by the partition (17.43) of the vector of unknowns.

17.3.1 The Schur complement

Consider now the Steklov-Poincaré interface equation (17.26) and look for its finite element counterpart. Since $\boldsymbol{\lambda}$ represents the unknown value of u on Γ , its finite element correspondent is the vector $\boldsymbol{\lambda}$ of the values of u_h at the interface nodes.

By gaussian elimination operated on system (17.45), we can obtain a new reduced system on the sole unknown $\boldsymbol{\lambda}$.

Matrices A_{11} and A_{22} are invertible since they are associated with two homogeneous Dirichlet boundary-value problems for the Laplace operator, hence

$$\mathbf{u}_1 = A_{11}^{-1}(\mathbf{f}_1 - A_{1\Gamma}\boldsymbol{\lambda}) \quad \text{and} \quad \mathbf{u}_2 = A_{22}^{-1}(\mathbf{f}_2 - A_{2\Gamma}\boldsymbol{\lambda}). \quad (17.46)$$

From the third equation in (17.44), we obtain

$$\begin{aligned} & \left[\left(A_{\Gamma\Gamma}^{(1)} - A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma} \right) + \left(A_{\Gamma\Gamma}^{(2)} - A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma} \right) \right] \boldsymbol{\lambda} \\ &= \mathbf{f}_\Gamma - A_{\Gamma 1}A_{11}^{-1}\mathbf{f}_1 - A_{\Gamma 2}A_{22}^{-1}\mathbf{f}_2. \end{aligned} \quad (17.47)$$

Using the following definitions:

$$\Sigma = \Sigma_1 + \Sigma_2, \quad \Sigma_i = A_{\Gamma\Gamma}^{(i)} - A_{\Gamma i}A_{ii}^{-1}A_{i\Gamma}, \quad i = 1, 2, \quad (17.48)$$

and

$$\boldsymbol{\chi}_\Gamma = \mathbf{f}_\Gamma - A_{\Gamma 1}A_{11}^{-1}\mathbf{f}_1 - A_{\Gamma 2}A_{22}^{-1}\mathbf{f}_2, \quad (17.49)$$

(17.47) becomes

$$\Sigma\boldsymbol{\lambda} = \boldsymbol{\chi}_\Gamma. \quad (17.50)$$

Since Σ and χ_Γ approximate S and χ , respectively, (17.50) can be considered as a finite element approximation to the Steklov-Poincaré equation (17.26). Matrix Σ is the so-called *Schur complement* of A with respect to \mathbf{u}_1 and \mathbf{u}_2 , whereas matrices Σ_i are the Schur complements related to the subdomains Ω_i ($i = 1, 2$).

Once system (17.50) is solved w.r.t the unknown λ , by virtue of (17.46) we can compute \mathbf{u}_1 and \mathbf{u}_2 . This computation amounts to solve numerically two Poisson problems on the two subdomains Ω_1 and Ω_2 , with Dirichlet boundary conditions $u_h^{(i)}|_\Gamma = \lambda_h$ ($i = 1, 2$) on the interface Γ .

The Schur complement Σ inherits some of the properties of its generating matrix A , as stated by the following result:

Lemma 17.1 *Matrix Σ satisfies the following properties:*

1. if A is singular, so is Σ ;
2. if A (respectively, A_{ii}) is symmetric, then Σ (respectively, Σ_i) is symmetric too;
3. if A is positive definite, so is Σ .

Recall that the condition number of the finite element stiffness matrix A satisfies $K_2(A) \simeq C h^{-2}$ (see (4.50)). As of Σ , it can be proven that

$$K_2(\Sigma) \simeq C h^{-1}. \quad (17.51)$$

In the specific case under consideration, A (and therefore Σ , thanks to Lemma 17.1) is symmetric and positive definite. It is therefore convenient to use the conjugate gradient method (with a suitable preconditioner) for the solution of system (17.50). At every iteration, the computation of the residue will involve the finite element solution of two independent Dirichlet boundary-value problems on the subdomains Ω_i .

17.3.2 The discrete Steklov-Poincaré operator

In this section we will find the discrete operator associated with the Schur complement. With this aim, besides the space $V_{i,h}$ previously introduced, we will need the one $V_{i,h}^0$ generated by the functions $\{\varphi_j^{(i)}\}$ exclusively associated to the internal nodes of the subdomain Ω_i , and the space Λ_h generated by the set of functions $\{\varphi_j^{(\Gamma)}\}$.

We have $\Lambda_h = \{\mu_h \mid \exists v_h \in V_h : v_h|_\Gamma = \mu_h\}$, whence Λ_h represents a finite element subspace of the trace functions space Λ introduced in (17.30).

Consider now the following problem: find $H_{i,h}\eta_h \in V_{i,h}$, with $H_{i,h}\eta_h = \eta_h$ on Γ , s.t.

$$\int_{\Omega_i} \nabla(H_{i,h}\eta_h) \cdot \nabla v_h d\Omega_i = 0 \quad \forall v_h \in V_{i,h}^0. \quad (17.52)$$

Clearly, $H_{i,h}\eta_h$ represents a finite element approximation of the harmonic extension $H_i\eta_h$, and the operator $H_{i,h} : \eta_h \rightarrow H_{i,h}\eta_h$ can be regarded as an approximation of

H_i . By expanding $H_{i,h}\eta_h$ in terms of the basis functions

$$H_{i,h}\eta_h = \sum_{j=1}^{N_i} u_j^{(i)} \varphi_j^{(i)} + \sum_{k=1}^{N_\Gamma} \eta_k \varphi_k^{(\Gamma)}|_{\Omega_i},$$

we can rewrite (17.52) in matrix form

$$A_{ii}\mathbf{u}^{(i)} = -A_{i\Gamma}\boldsymbol{\eta}. \quad (17.53)$$

The following result, called *the uniform discrete extension theorem*, holds:

Theorem 17.3 *There exist two constants $\hat{C}_1, \hat{C}_2 > 0$, independent of h , s.t.*

$$\hat{C}_1 \|\eta_h\|_A \leq \|H_{i,h}\eta_h\|_{H^1(\Omega_i)} \leq \hat{C}_2 \|\eta_h\|_A \quad \forall \eta_h \in A_h \quad i = 1, 2. \quad (17.54)$$

Consequently, there exist two constants $K_1, K_2 > 0$, independent of h , s.t.

$$K_1 \|H_{1,h}\eta_h\|_{H^1(\Omega_1)} \leq \|H_{2,h}\eta_h\|_{H^1(\Omega_2)} \leq K_2 \|H_{1,h}\eta_h\|_{H^1(\Omega_1)} \quad \forall \eta_h \in A_h. \quad (17.55)$$

For the proof see, e.g., [QV99].

Now for $i = 1, 2$ the (local) discrete Steklov-Poincaré operator is defined as follows: $S_{i,h} : A_h \rightarrow A'_h$,

$$\langle S_{i,h}\eta_h, \mu_h \rangle = \int_{\Omega_i} \nabla(H_{i,h}\eta_h) \cdot \nabla(H_{i,h}\mu_h) d\Omega_i \quad \forall \eta_h, \mu_h \in A_h, \quad (17.56)$$

then we define the (global) discrete Steklov-Poincaré operator as $S_h = S_{1,h} + S_{2,h}$.

Lemma 17.2 *The local discrete Steklov-Poincaré operator can be expressed in terms of the local Schur complement as*

$$\langle S_{i,h}\eta_h, \mu_h \rangle = \boldsymbol{\mu}^T \Sigma_i \boldsymbol{\eta} \quad \forall \eta_h, \mu_h \in A_h, i = 1, 2, \quad (17.57)$$

where

$$\eta_h = \sum_{k=1}^{N_\Gamma} \eta_k \varphi_k^{(\Gamma)}|_\Gamma, \quad \mu_h = \sum_{k=1}^{N_\Gamma} \mu_k \varphi_k^{(\Gamma)}|_\Gamma$$

and

$$\boldsymbol{\eta} = (\eta_1, \dots, \eta_{N_\Gamma})^T, \quad \boldsymbol{\mu} = (\mu_1, \dots, \mu_{N_\Gamma})^T.$$

Therefore the global discrete Steklov-Poincaré operator $S_h = S_{1,h} + S_{2,h}$ satisfies the relation

$$\langle S_h\eta_h, \mu_h \rangle = \boldsymbol{\mu}^T \Sigma \boldsymbol{\eta} \quad \forall \eta_h, \mu_h \in A_h. \quad (17.58)$$

Proof. For $i = 1, 2$ we have

$$\begin{aligned}
\langle S_{i,h} \eta_h, \mu_h \rangle &= a_i(H_{i,h} \eta_h, H_{i,h} \mu_h) \\
&= a_i \left(\sum_{j=1}^{N_\Gamma} u_j \varphi_j^{(i)} + \sum_{k=1}^{N_\Gamma} \eta_k \varphi_k^{(\Gamma)}|_{\Omega_i}, \sum_{l=1}^{N_\Gamma} w_l \varphi_l^{(i)} + \sum_{m=1}^{N_\Gamma} \mu_m \varphi_m^{(\Gamma)}|_{\Omega_i} \right) \\
&= \sum_{j,l=1}^{N_\Gamma} w_l a_i(\varphi_j^{(i)}, \varphi_l^{(i)}) u_j + \sum_{j,m=1}^{N_\Gamma} \mu_m a_i(\varphi_j^{(i)}, \varphi_m^{(\Gamma)}|_{\Omega_i}) u_j \\
&\quad + \sum_{k,l=1}^{N_\Gamma} w_l a_i(\varphi_k^{(\Gamma)}|_{\Omega_i}, \varphi_l^{(i)}) \eta_k + \sum_{k,m=1}^{N_\Gamma} \mu_m a_i(\varphi_k^{(\Gamma)}|_{\Omega_i}, \varphi_m^{(\Gamma)}|_{\Omega_i}) \eta_k \\
&= \mathbf{w}^T A_{ii} \mathbf{u} + \boldsymbol{\mu}^T A_{i\Gamma} \mathbf{u} + \mathbf{w}^T A_{i\Gamma} \boldsymbol{\eta} + \boldsymbol{\mu}^T A_{\Gamma\Gamma}^{(i)} \boldsymbol{\eta}.
\end{aligned}$$

Thanks to (17.53) we obtain

$$\begin{aligned}
\langle S_{i,h} \eta_h, \mu_h \rangle &= -\mathbf{w}^T A_{i\Gamma} \boldsymbol{\eta} - \boldsymbol{\mu}^T A_{\Gamma i} A_{ii}^{-1} A_{i\Gamma} \boldsymbol{\eta} + \mathbf{w}^T A_{i\Gamma} \boldsymbol{\eta} + \boldsymbol{\mu}^T A_{\Gamma\Gamma}^{(i)} \boldsymbol{\eta} \\
&= \boldsymbol{\mu}^T \left(A_{\Gamma\Gamma}^{(i)} - A_{\Gamma i} A_{ii}^{-1} A_{i\Gamma} \right) \boldsymbol{\eta} \\
&= \boldsymbol{\mu}^T \Sigma_i \boldsymbol{\eta}.
\end{aligned}$$

◇

From Theorem 17.3 and thanks to the representation (17.56), we deduce that there exist two constants $\hat{K}_1, \hat{K}_2 > 0$, independent of h , s.t.

$$\hat{K}_1 \langle S_{1,h} \mu_h, \mu_h \rangle \leq \langle S_{2,h} \mu_h, \mu_h \rangle \leq \hat{K}_2 \langle S_{1,h} \mu_h, \mu_h \rangle \quad \forall \mu_h \in \Lambda_h. \quad (17.59)$$

Thanks to (17.57) we can infer that there exist two constants $\tilde{K}_1, \tilde{K}_2 > 0$, independent of h , s.t.

$$\tilde{K}_1 (\boldsymbol{\mu}^T \Sigma_1 \boldsymbol{\mu}) \leq \boldsymbol{\mu}^T \Sigma_2 \boldsymbol{\mu} \leq \tilde{K}_2 (\boldsymbol{\mu}^T \Sigma_1 \boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathbb{R}^{N_\Gamma}. \quad (17.60)$$

This amounts to say that the two matrices Σ_1 and Σ_2 are spectrally equivalent, that is their spectral condition number features the same asymptotic behavior w.r.t h . Henceforth, both Σ_1 and Σ_2 provide an optimal preconditioner of the Schur complement Σ , that is there exists a constant C , independent of h , s.t.

$$K_2(\Sigma_i^{-1} \Sigma) \leq C, \quad i = 1, 2. \quad (17.61)$$

As we will see in Sec. 17.3.3, this property allows us to prove that the discrete version of the Dirichlet-Neumann algorithm converges with a rate independent of h . A similar result holds for the discrete Neumann-Neumann algorithm.

17.3.3 Equivalence between the Dirichlet-Neumann algorithm and a preconditioned Richardson algorithm in the discrete case

Let us now prove the analogue of the equivalence theorem 17.2 in the algebraic case. The finite element approximation of the Dirichlet problem (17.31) has the following algebraic form

$$A_{11}\mathbf{u}_1^{(k)} = \mathbf{f}_1 - A_{1\Gamma}\boldsymbol{\lambda}^{(k-1)}, \quad (17.62)$$

whereas that of the Neumann problem (17.32) reads

$$\begin{bmatrix} A_{22} & A_{2\Gamma} \\ A_{\Gamma 2} & A_{\Gamma\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} \mathbf{u}_2^{(k)} \\ \boldsymbol{\lambda}^{(k-1/2)} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_2 \\ \mathbf{f}_\Gamma - A_{\Gamma 1}\mathbf{u}_1^{(k)} - A_{\Gamma\Gamma}^{(1)}\boldsymbol{\lambda}^{(k-1)} \end{bmatrix}. \quad (17.63)$$

In its turn, (17.33) becomes

$$\boldsymbol{\lambda}^{(k)} = \theta\boldsymbol{\lambda}^{(k-1/2)} + (1-\theta)\boldsymbol{\lambda}^{(k-1)}. \quad (17.64)$$

By eliminating $\mathbf{u}_2^{(k)}$ from (17.63) we obtain

$$\left(A_{\Gamma\Gamma}^{(2)} - A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma} \right) \boldsymbol{\lambda}^{(k-1/2)} = \mathbf{f}_\Gamma - A_{\Gamma 1}\mathbf{u}_1^{(k)} - A_{\Gamma\Gamma}^{(1)}\boldsymbol{\lambda}^{(k-1)} - A_{\Gamma 2}A_{22}^{-1}\mathbf{f}_2.$$

By the definition (17.48) of Σ_2 and by (17.62), one has

$$\Sigma_2\boldsymbol{\lambda}^{(k-1/2)} = \mathbf{f}_\Gamma - A_{\Gamma 1}A_{11}^{-1}\mathbf{f}_1 - A_{\Gamma 2}A_{22}^{-1}\mathbf{f}_2 - \left(A_{\Gamma\Gamma}^{(1)} - A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma} \right) \boldsymbol{\lambda}^{(k-1)},$$

that is, owing to the definition (17.48) of Σ_1 and to (17.49),

$$\boldsymbol{\lambda}^{(k-1/2)} = \Sigma_2^{-1} \left(\boldsymbol{\chi}_\Gamma - \Sigma_1\boldsymbol{\lambda}^{(k-1)} \right).$$

Now, by virtue of (17.64) we deduce

$$\boldsymbol{\lambda}^{(k)} = \theta\Sigma_2^{-1} \left(\boldsymbol{\chi}_\Gamma - \Sigma_1\boldsymbol{\lambda}^{(k-1)} \right) + (1-\theta)\boldsymbol{\lambda}^{(k-1)},$$

that is, since $-\Sigma_1 = -\Sigma + \Sigma_2$,

$$\boldsymbol{\lambda}^{(k)} = \theta\Sigma_2^{-1} \left(\boldsymbol{\chi}_\Gamma - \Sigma\boldsymbol{\lambda}^{(k-1)} + \Sigma_2\boldsymbol{\lambda}^{(k-1)} \right) + (1-\theta)\boldsymbol{\lambda}^{(k-1)}$$

whence

$$\Sigma_2(\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k-1)}) = \theta(\boldsymbol{\chi}_\Gamma - \Sigma\boldsymbol{\lambda}^{(k-1)}).$$

The latter is nothing but a Richardson iteration on the system (17.50) using the local Schur complement Σ_2 as preconditioner.

Remark 17.1 The Richardson preconditioner induced by the Dirichlet-Neumann algorithm is in fact the local Schur complement associated to that subdomain on which we solve a Neumann problem. So, in the so-called *Neumann-Dirichlet* algorithm, in which at every iteration we solve a Dirichlet problem in Ω_2 and a Neumann one in Ω_1 , the preconditioner of the associated Richardson algorithm would be Σ_1 and not Σ_2 . •

Remark 17.2 An analogous result can be proven for the discrete version of the Neumann-Neumann algorithm introduced in Sec. 17.1.3. Precisely, the Neumann-Neumann algorithm is equivalent to the Richardson algorithm applied to system (17.50) with a preconditioner whose inverse is given by $P_h^{-1} = \sigma_1 \Sigma_1^{-1} + \sigma_2 \Sigma_2^{-1}$, σ_1 and σ_2 being the coefficients used for the (discrete) interface equation which corresponds to (17.19). Moreover we can prove that there exists a constant $C > 0$, independent of h , s.t.

$$K_2((\sigma_1 \Sigma_1^{-1} + \sigma_2 \Sigma_2^{-1}) \Sigma) \leq C.$$

Proceeding in a similar way we can show that the discrete version of the Robin-Robin algorithm (17.20)-(17.21) is also equivalent to a Richardson algorithm for (17.50), using this time as preconditioner the matrix $(\gamma_1 + \gamma_2)^{-1}(\gamma_1 I + \Sigma_1)(\gamma_2 I + \Sigma_2)$. •

Let us recall that a matrix P_h is an optimal preconditioner for Σ if the condition number of $P_h^{-1} \Sigma$ is bounded uniformly w.r.t the dimension N of the matrix Σ (and therefore from h in the case in which Σ arises from a finite element discretization).

We can therefore summarize by saying that for the solution of system $\Sigma \lambda = \chi_\Gamma$, we can make use of the following preconditioners, all of them being optimal:

$$P_h = \begin{cases} \Sigma_2 & \text{for the Dirichlet-Neumann algorithm,} \\ \Sigma_1 & \text{for the Neumann-Dirichlet algorithm,} \\ (\sigma_1 \Sigma_1^{-1} + \sigma_2 \Sigma_2^{-1})^{-1} & \text{for the Neumann-Neumann algorithm,} \\ (\gamma_1 + \gamma_2)^{-1}(\gamma_1 I + \Sigma_1)(\gamma_2 I + \Sigma_2) & \text{for the Robin-Robin algorithm.} \end{cases} \quad (17.65)$$

From the convergence theory of Richardson method we know that if both Σ and P_h are symmetric and positive definite, one has $\|\lambda^n - \lambda\|_\Sigma \leq \rho^n \|\lambda^0 - \lambda\|_\Sigma$, $n \geq 0$, being $\|\mathbf{v}\|_\Sigma = (\mathbf{v}^T \Sigma \mathbf{v})^{1/2}$. The optimal convergence rate is given by

$$\rho = \frac{K_2(P_h^{-1} \Sigma) - 1}{K_2(P_h^{-1} \Sigma) + 1},$$

and is therefore independent of h .

17.4 Generalization to the case of many subdomains

To generalize the previous DD algorithms to the case in which the domain Ω is partitioned into an arbitrary number $M > 2$ of subdomains we proceed as follows.

Let Ω_i , $i = 1, \dots, M$, denote a family of disjoint subdomains s.t. $\cup \overline{\Omega}_i = \overline{\Omega}$, $\Gamma_i = \partial \Omega_i \setminus \partial \Omega$ and $\Gamma = \cup \Gamma_i$ (the skeleton).

Let us consider the Poisson problem (3.13). In the current case the equivalence Theorem 17.1 generalizes as follows:

$$\begin{cases} -\Delta u_i = f & \text{in } \Omega_i, \\ u_i = u_k & \text{on } \Gamma_{ik}, \quad \forall k \in \mathcal{A}(i), \\ \frac{\partial u_i}{\partial n_i} = \frac{\partial u_k}{\partial n_i} & \text{on } \Gamma_{ik}, \quad \forall k \in \mathcal{A}(i), \\ u_i = 0 & \text{on } \partial\Omega_i \cap \partial\Omega, \end{cases} \quad (17.66)$$

for $i = 1, \dots, M$, being $\Gamma_{ik} = \partial\Omega_i \cap \partial\Omega_k \neq \emptyset$, $\mathcal{A}(i)$ the set of indices k s.t. Ω_k is adjacent to Ω_i ; as usual, \mathbf{n}_i denotes the outward unit normal vector to Ω_i .

Assume now that (3.13) has been approximated by the finite element method. Following the ideas presented in Sec. 17.3 and denoting by $\mathbf{u} = (\mathbf{u}_I, \mathbf{u}_\Gamma)^T$ the vector of unknowns split in two subvectors, the one (\mathbf{u}_I) related with the internal nodes, and that (\mathbf{u}_Γ) related with the nodes lying on the skeleton Γ , the finite element algebraic system can be reformulated in blockwise form as follows

$$\begin{bmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{u}_I \\ \mathbf{u}_\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{f}_I \\ \mathbf{f}_\Gamma \end{bmatrix}, \quad (17.67)$$

being $A_{\Gamma I} = A_{I\Gamma}^T$. Matrix $A_{I\Gamma}$ is banded, while A_{II} has the block diagonal form

$$A_{II} = \begin{bmatrix} A_{11} & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & A_{MM} \end{bmatrix}. \quad (17.68)$$

We are using the following notations:

$$\begin{aligned} (A_{ii})_{lj} &= a_i(\varphi_j, \varphi_l), \quad 1 \leq l, j \leq N_i, \\ (A_{\Gamma\Gamma}^{(i)})_{sr} &= a_i(\psi_r, \psi_s), \quad 1 \leq r, s \leq N_{\Gamma_i}, \\ (A_{i\Gamma})_{lr} &= a_i(\psi_r, \varphi_l), \quad 1 \leq r \leq N_{\Gamma_i}, \quad 1 \leq l \leq N_i, \end{aligned}$$

where N_i is the number of nodes internal to Ω_i , N_{Γ_i} that of the nodes sitting on the interface Γ_i , φ_j and ψ_r the basis functions associated with the internal and interface nodes, respectively.

Let us remark that on every subdomain Ω_i the matrix

$$A_i = \begin{bmatrix} A_{ii} & A_{i\Gamma} \\ A_{\Gamma i} & A_{\Gamma\Gamma}^{(i)} \end{bmatrix} \quad (17.69)$$

represents the local finite element stiffness matrix associated to a Neumann problem on Ω_i . Since A_{II} is non-singular, from (17.67) we can formally derive

$$\mathbf{u}_I = A_{II}^{-1}(\mathbf{f}_I - A_{I\Gamma}\mathbf{u}_\Gamma). \quad (17.70)$$

By eliminating the unknown \mathbf{u}_I from system (17.67), it follows

$$A_{\Gamma\Gamma}\mathbf{u}_\Gamma = \mathbf{f}_\Gamma - A_{\Gamma I}A_{II}^{-1}(\mathbf{f}_I - A_{II}\mathbf{u}_\Gamma),$$

that is

$$(A_{\Gamma\Gamma} - A_{\Gamma I}A_{II}^{-1}A_{II})\mathbf{u}_\Gamma = \mathbf{f}_\Gamma - A_{\Gamma I}A_{II}^{-1}\mathbf{f}_I. \quad (17.71)$$

Setting now

$$\Sigma = A_{\Gamma\Gamma} - A_{\Gamma I}A_{II}^{-1}A_{II} \text{ and } \chi_\Gamma = \mathbf{f}_\Gamma - A_{\Gamma I}A_{II}^{-1}\mathbf{f}_I,$$

and denoting, as usual, $\lambda = \mathbf{u}_\Gamma$, (17.71) becomes

$$\Sigma\lambda = \chi_\Gamma. \quad (17.72)$$

This is the Schur complement system in the multidomain case. It can be regarded as a finite element approximation of the interface Steklov-Poincaré problem in the case of M subdomains.

The local Schur complements are defined as

$$\Sigma_i = A_{\Gamma\Gamma}^{(i)} - A_{\Gamma i}A_{ii}^{-1}A_{i\Gamma}, \quad i = 1, \dots, M,$$

hence

$$\Sigma = \Sigma_1 + \dots + \Sigma_M.$$

A general algorithm to solve the finite element Poisson problem in Ω could be formulated as follows:

1. compute the solution of (17.72) to obtain the value of λ on the skeleton Γ ;
2. then solve (17.70); since A_{II} is block-diagonal, this step yields the solution of M independent subproblems of reduced dimension, $A_{ii}\mathbf{u}_I^i = \mathbf{g}^i$, $i = 1, \dots, M$, which can therefore be carried out in parallel.

About the condition number of Σ , the following estimate can be proven: there exists a constant $C > 0$, independent of h e H , s.t.

$$K_2(\Sigma) \leq C \frac{H}{h H_{min}^2}, \quad (17.73)$$

H being the maximum diameter of the subdomains and H_{min} the minimum one.

17.4.1 Some numerical results

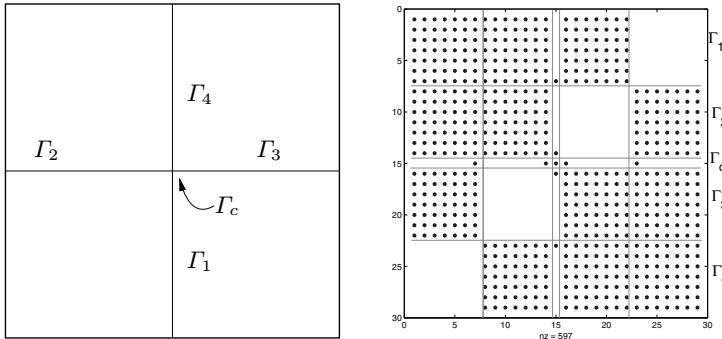
Consider the Poisson problem (3.13) on the domain $\Omega = (0, 1)^2$ whose finite element approximation was given in (4.40).

Let us partition Ω into M disjoint squares Ω_i whose sidelength is H , s.t. $\cup_{i=1}^M \overline{\Omega_i} = \overline{\Omega}$. An example with four subdomains is displayed in Fig. 17.5 (left).

In Table 17.1 we report the numerical values of $K_2(\Sigma)$ for the problem at hand, for several values of the finite element grid-size h ; it grows linearly with $1/h$ and

Table 17.1. Condition number of the Schur complement Σ

$K_2(\Sigma)$	$H = 1/2$	$H = 1/4$	$H = 1/8$
$h=1/8$	9.77	14.83	25.27
$h=1/16$	21.49	35.25	58.60
$h=1/32$	44.09	75.10	137.73
$h=1/64$	91.98	155.19	290.43

**Fig. 17.5.** Example of partition of $\Omega = (0, 1)^2$ into four squared subdomains (left). Pattern of the Schur complement Σ (right) corresponding to the domain partition displayed on the left

with $1/H$, as predicted by the formula (17.73) (here $H_{min} = H$). In Fig. 17.5 (right) we display the *pattern* of the Schur complement matrix Σ in the particular case of $h = 1/8$ and $H = 1/2$. The matrix has a blockwise structure that accounts for the interfaces $\Gamma_1, \Gamma_2, \Gamma_3$ and Γ_4 , plus the contribution arising from the crosspoint Γ_c . Since Σ is a *dense* matrix, when solving the linear system (17.72) the explicit computation of its entries is not convenient. Instead, we can use the following **Algorithm 14.1** to compute the matrix-vector product $\Sigma \mathbf{x}_\Gamma$, for any vector \mathbf{x}_Γ (and therefore the residue at every step of an iterative algorithm). We have denoted by $R_{\Gamma_i} : \Gamma \rightarrow \Gamma_i = \partial\Omega_i \setminus \partial\Omega$ a suitable restriction operator, while $\mathbf{x} \leftarrow \mathbf{y}$ indicates the algebraic operation $\mathbf{x} = \mathbf{x} + \mathbf{y}$.

Algorithm 14.1 (Schur complement multiplication by a vector)

Given \mathbf{x}_Γ , compute $\mathbf{y}_\Gamma = \Sigma \mathbf{x}_\Gamma$ as follows:

- Set $\mathbf{y}_\Gamma = \mathbf{0}$
- For $i = 1, \dots, M$ Do in parallel:
 - $\mathbf{x}_i = R_{\Gamma_i} \mathbf{x}_\Gamma$
 - $\mathbf{z}_i = A_{i\Gamma_i} \mathbf{x}_i$
 - $\mathbf{z}_i \leftarrow A_{ii}^{-1} \mathbf{z}_i$
 - sum up in the local vector $\mathbf{y}_{\Gamma_i} \leftarrow A_{\Gamma_i \Gamma_i} \mathbf{x}_i - A_{\Gamma_i i} \mathbf{z}_i$
 - sum up in the global vector $\mathbf{y}_\Gamma \leftarrow R_{\Gamma_i}^T \mathbf{y}_{\Gamma_i}$
- EndFor

Since no communication is required among the subdomains, this is a fully parallel algorithm.

Before using for the first time the Schur complement, a start-up phase, described in **Algorithm 14.2**, is requested. Note that this is an *off-line* procedure.

Algorithm 14.2 (start-up phase for the solution of the Schur complement system)

Given \mathbf{x}_Γ , compute $\mathbf{y}_\Gamma = \Sigma \mathbf{x}_\Gamma$ as follows:

a. For $i = 1, \dots, M$ Do in parallel:

b. Compute the entries of A_i

c. Reorder A_i as

$$A_i = \begin{pmatrix} A_{ii} & A_{i\Gamma_i} \\ A_{\Gamma_i i} & A_{\Gamma_i \Gamma_i} \end{pmatrix}$$

then extract the submatrices A_{ii} , $A_{i\Gamma_i}$, $A_{\Gamma_i i}$ and $A_{\Gamma_i \Gamma_i}$

d. Compute the (either LU or Cholesky) factorization of A_{ii}

e. EndFor

17.5 DD preconditioners in case of many subdomains

Before introducing the preconditioners for the Schur complement in the case in which Ω is partitioned in many subdomains we recall the following definition:

Definition 17.1 A preconditioner P_h of Σ is said to be scalable if the condition number of the preconditioned matrix $P_h^{-1}\Sigma$ is independent of the number of subdomains.

Iterative methods using scalable preconditioners allow henceforth to achieve convergence rates independent of the subdomain number. This is a very desirable property in those cases where a large number of subdomains is used.

Let R_i be a *restriction operator* which, to any vector \mathbf{v}_h of nodal values on the global domain Ω , associates its restriction to the subdomain Ω_i

$$R_i : \mathbf{v}_h|_\Omega \rightarrow \mathbf{v}_h^i|_{\Omega_i \cup \Gamma_i}.$$

Let moreover

$$R_i^T : \mathbf{v}_h^i|_{\Omega_i \cup \Gamma_i} \rightarrow \mathbf{v}_h|_\Omega$$

be the *prolongation (or extension-by-zero) operator*. In algebraic form R_i can be represented by a matrix that coincides with the identity matrix in correspondence with the subdomain Ω_i

$$R_i = \left[\begin{array}{ccc|c|cc} 0 & \dots & 0 & 1 & & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & & 1 & 0 & \dots & 0 \end{array} \right]_{\Omega_i}$$

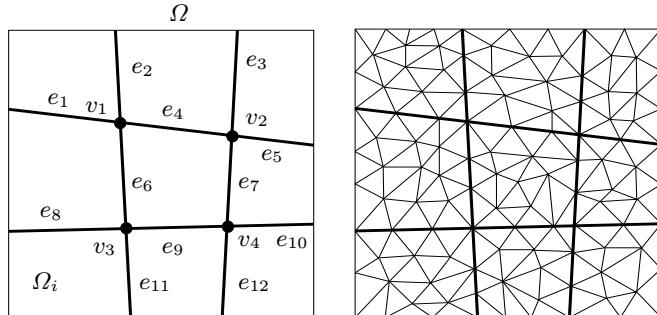


Fig. 17.6. A decomposition into 9 subdomains (left) with a fine triangulation in small triangles and a coarse triangulation in large quadrilaterals (the 9 subdomains) (right)

Similarly we can define the restriction and prolongation operators R_{Γ_i} and $R_{\Gamma_i}^T$, respectively, that act on the vector of interface nodal values. A possible preconditioner for Σ is

$$P_h = \sum_{i=1}^M R_{\Gamma_i}^T \Sigma_i R_{\Gamma_i}.$$

More in general, the strategy consists of combining the contributions of local subdomain preconditioners with that of a global contribution referring to a coarse grid whose elements are the subdomains themselves. This idea can be formalized through the following relation that provides the inverse of the preconditioner

$$(P_h)^{-1} = \sum_{i=1}^M R_{\Gamma_i}^T P_{i,h}^{-1} R_{\Gamma_i} + R_{\Gamma}^T P_H^{-1} R_{\Gamma}.$$

As usual we have denoted by H the maximum value of the diameters H_i of the subdomains Ω_i ; moreover, $P_{i,h}$ is either the local Schur complement Σ_i , or (more frequently) a suitable preconditioner of Σ_i , while R_{Γ} and P_H refer to operators that act on the global scale (that of the coarse grid).

Many different choices are possible for the local Schur complemet preconditioner $P_{i,h}$; they will give rise to different condition numbers of the preconditioned matrix $P_h^{-1} \Sigma$.

17.5.1 Jacobi preconditioner

Let $\{e_1, \dots, e_m\}$ be the set of edges and $\{v_1, \dots, v_n\}$ that of vertices of a partition of Ω into subdomains (see Fig. 17.6 for an example).

The Schur complement Σ features the following blockwise representation

$$\Sigma = \begin{bmatrix} \Sigma_{ee} & \Sigma_{ev} \\ \hline \Sigma_{ev}^T & \Sigma_{vv} \end{bmatrix},$$

having set

$$\Sigma_{ee} = \begin{bmatrix} \Sigma_{e_1 e_1} & \dots & \Sigma_{e_1 e_m} \\ \vdots & \ddots & \vdots \\ \Sigma_{e_m e_1} & \dots & \Sigma_{e_m e_m} \end{bmatrix}, \quad \Sigma_{ev} = \begin{bmatrix} \Sigma_{e_1 v_1} & \dots & \Sigma_{e_1 v_n} \\ \vdots & \ddots & \vdots \\ \Sigma_{e_m v_1} & \dots & \Sigma_{e_m v_n} \end{bmatrix}$$

and

$$\Sigma_{vv} = \begin{bmatrix} \Sigma_{v_1 v_1} & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \Sigma_{v_n v_n} \end{bmatrix}.$$

The *Jacobi preconditioner* of the Schur complement Σ is a block diagonal matrix defined by

$$P_h^J = \left[\begin{array}{c|c} \hat{\Sigma}_{ee} & 0 \\ \hline 0 & \Sigma_{vv} \end{array} \right]$$

where $\hat{\Sigma}_{ee}$ is either Σ_{ee} or a suitable approximation of it. This preconditioner does not account for the interaction between the basis functions associated with edges and those associated with vertices. The matrix $\hat{\Sigma}_{ee}$ is also diagonal

$$\hat{\Sigma}_{ee} = \begin{bmatrix} \hat{\Sigma}_{e_1 e_1} & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \hat{\Sigma}_{e_m e_m} \end{bmatrix}.$$

Here $\hat{\Sigma}_{e_k e_k}$ denotes $\Sigma_{e_k e_k}$ or a suitable approximation of it.

The preconditioner P_h^J can also be expressed in terms of restriction and prolongation operators as follows

$$(P_h^J)^{-1} = \sum_{k=1}^m R_{e_k}^T \hat{\Sigma}_{e_k e_k}^{-1} R_{e_k} + R_v^T \Sigma_{vv}^{-1} R_v, \quad (17.74)$$

where R_{e_k} and R_v denote edge and vertices restriction operators, respectively.

Regarding the condition number of the preconditioned Schur complement, there exists a constant $C > 0$, independent of both h and H , s.t.

$$K_2((P_h^J)^{-1} \Sigma) \leq CH^{-2} \left(1 + \log \frac{H}{h}\right)^2.$$

Should the conjugate gradient method be used to solve the preconditioned Schur complement system (17.72) with preconditioner P_h^J , the number of iterations necessary to converge (within a prescribed tolerance) would be proportional to H^{-1} . The presence of H indicates that the Jacobi preconditioner is not scalable.

Moreover, we notice that the presence of the logarithmic term $\log(H/h)$ introduces a relation between the size of the subdomains and the size of the computational grid \mathcal{T}_h . This generates a propagation of information among subdomains characterized by a finite (rather than infinite) speed of propagation.

17.5.2 Bramble-Pasciak-Schatz preconditioner

With the aim of accelerating the speed of propagation of information among subdomains we can devise a mechanism of global coupling among subdomains. As already anticipated, the family of subdomains can be regarded as a *coarse* grid, say \mathcal{T}_H , of the original domain. For instance, in Fig. 17.6 \mathcal{T}_H is made of 9 (macro) elements and 4 internal nodes. It identifies a stiffness matrix of piecewise bilinear elements, say A_H , of dimension 4×4 which guarantees a global coupling in Ω . We can now introduce a restriction operator that, for simplicity, we indicate $R_H : \Gamma_h \rightarrow \Gamma_H$. More precisely, this operator transforms a vector of nodal values on the skeleton Γ_h into a vector of nodal values on the internal vertices of the coarse grid (4 in the case at hand). Its transpose R_H^T is an extension operator. The matrix P_h^{BPS} , whose inverse is

$$(P_h^{BPS})^{-1} = \sum_{k=1}^m R_{e_k}^T \hat{\Sigma}_{e_k e_k}^{-1} R_{e_k} + R_H^T A_H^{-1} R_H , \quad (17.75)$$

is named Bramble-Pasciak-Schatz preconditioner. The main difference with Jacobi preconditioner (17.74) is due to the presence of the global (coarse-grid) stiffness matrix A_H instead of the diagonal vertex matrix Σ_{vv} . The following results hold:

$$\begin{aligned} K_2((P_h^{BPS})^{-1} \Sigma) &\leq C \left(1 + \log \frac{H}{h}\right)^2 \text{ in 2D,} \\ K_2((P_h^{BPS})^{-1} \Sigma) &\leq C \frac{H}{h} \text{ in 3D.} \end{aligned}$$

Note that the factor H^{-2} does not show up anymore. The number of iterations of the conjugate gradient method with preconditioner P_h^{BPS} is now proportional to $\log(H/h)$ in 2D and to $(H/h)^{1/2}$ in 3D.

17.5.3 Neumann-Neumann preconditioner

Although the Bramble-Pasciak-Schatz preconditioner has better properties than Jacobi's, yet in 3D the condition number of the preconditioned Schur complement still contains a linear dependence on H/h .

In this respect, a further improvement is achievable using the so-called Neumann-Neumann preconditioner, whose inverse has the following expression

$$(P_h^{NN})^{-1} = \sum_{i=1}^M R_{\Gamma_i}^T D_i \Sigma_i^* D_i R_{\Gamma_i} . \quad (17.76)$$

Here R_{Γ_i} still denotes the restriction from the nodal values on the whole skeleton Γ to those on the local interface Γ_i , whereas Σ_i^* is either Σ_i^{-1} (should the local inverse exist) or an approximation of Σ_i^{-1} , e.g. the pseudo-inverse Σ_i^+ of Σ_i . The matrix

$$D_i = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix}$$

is a diagonal matrix of positive weights $d_j > 0$, for $j = 1, \dots, n$, n being the number of nodes on Γ_i . More precisely, d_j coincides with the inverse of the number of subdomains that share the j -th node. If we still consider the 4 internal vertices of Fig. 17.6, we will have $d_j = 1/4$, for $j = 1, \dots, 4$.

For the preconditioner (17.76) the following estimate (similar to that of Jacobi preconditioner) holds: there exists a constant $C > 0$, independent of both h and H , s.t.

$$K_2((P_h^{NN})^{-1}\Sigma) \leq CH^{-2} \left(1 + \log \frac{H}{h}\right)^2.$$

The presence of D_i and R_{Γ_i} in (17.76) only entails matrix-matrix multiplications. On the other hand, if $\Sigma_i^* = \Sigma_i^{-1}$, applying Σ_i^{-1} to a given vector can be reconduted to the use of local inverses. As a matter of fact, let \mathbf{q} be a vector whose components are the nodal values on the local interface Γ_i ; then

$$\Sigma_i^{-1}\mathbf{q} = [0, I]A_i^{-1}[0, I]^T\mathbf{q}.$$

In particular, $[0, I]^T\mathbf{q} = [0, \mathbf{q}]^T$, and the matrix-vector product

$$\underbrace{\begin{bmatrix} & \text{internal} \\ & \text{nodes} \\ \hline & \text{boundary nodes} \end{bmatrix}}_{A_i^{-1}} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{q} \end{bmatrix}$$

corresponds to the solution on Ω_i of the Neumann boundary-value problem

$$\begin{cases} -\Delta w_i = 0 & \text{in } \Omega_i, \\ \frac{\partial w_i}{\partial n} = q & \text{on } \Gamma_i. \end{cases} \quad (17.77)$$

Algorithm 14.3 (Neumann-Neumann preconditioner)

Given a vector \mathbf{r}_Γ , compute $\mathbf{z}_\Gamma = (P_h^{NN})^{-1}\mathbf{r}_\Gamma$ as follows:

- a. Set $\mathbf{z}_\Gamma = \mathbf{0}$
- b. For $i = 1, \dots, M$ Do in parallel:
 - c. restrict the residue on Ω_i : $\mathbf{r}_i = R_{\Gamma_i}\mathbf{r}_\Gamma$
 - d. compute $\mathbf{z}_i = [0, I]A_i^{-1}[0, \mathbf{r}_i]^T$
 - e. Sum up the global residue: $\mathbf{z}_\Gamma \leftarrow R_{\Gamma_i}^T\mathbf{z}_i$
- f. EndFor

Also in this case a start-up phase is required, consisting in the preparation for the solution of linear systems with local stiffness matrices A_i . Note that in the case of the model problem (3.13), A_i is singular if Ω_i is an internal subdomain, that is if $\partial\Omega_i \setminus \partial\Omega = \emptyset$. One of the following strategies should be adopted:

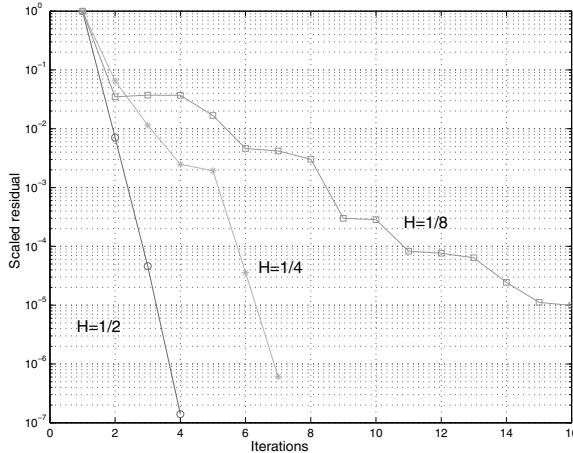


Fig. 17.7. Convergence history for the preconditioned conjugate gradient method with preconditioner P_h^{NN} when $h = 1/32$

Table 17.2. Condition number of the preconditioned matrix $(P_h^{NN})^{-1} \Sigma$

$K_2((P_h^{NN})^{-1} \Sigma)$	$H = 1/2$	$H = 1/4$	$H = 1/8$	$H = 1/16$
$h = 1/16$	2.55	15.20	47.60	—
$h = 1/32$	3.45	20.67	76.46	194.65
$h = 1/64$	4.53	26.25	105.38	316.54
$h = 1/128$	5.79	31.95	134.02	438.02

1. compute a (either LU or Cholesky) factorization of $A_i + \epsilon I$, for a given $\epsilon > 0$ sufficiently small;
2. compute a factorization of $A_i + \frac{1}{H^2} M_i$, where M_i is the mass matrix whose entries are

$$(M_i)_{k,j} = \int_{\Omega_i} \varphi_k \varphi_j d\Omega_i;$$

3. compute the singular-value decomposition of A_i .

The matrix Σ_i^* is defined accordingly. In our numerical results we have adopted the third approach.

The convergence history of the preconditioned conjugate gradient method with preconditioner P_h^{NN} in the case $h = 1/32$ is displayed in Fig. 17.7. In Table 17.2 we report the values of the condition number of $(P_h^{NN})^{-1} \Sigma$ for several values of H .

As already pointed out, the Neumann-Neumann preconditioner of the Schur complement matrix is not scalable. A substantial improvement of (17.76) can be achieved by adding a coarse grid correction mechanism, yielding the following new preconditioner

$$(P_h^{BNN})^{-1} = \Sigma_H^{-1} + (I - \Sigma_H^{-1} \Sigma)(P_h^{NN})^{-1}(I - \Sigma \Sigma_H^{-1}) \quad (17.78)$$

in which we have used the shorthand notation $\Sigma_H^{-1} = R_\Gamma^T A_H^{-1} R_\Gamma$.

The matrix P_h^{BNN} is called *balanced Neumann-Neumann preconditioner*.

It can be proven that there exists a constant $C > 0$, independent of h and H , s.t.

$$K_2 \left((P_h^{BNN})^{-1} \Sigma \right) \leq C \left(1 + \log \frac{H}{h} \right)^2$$

both in 2D and 3D. The balanced Neumann-Neumann preconditioner therefore guarantees optimal scalability up to a light logarithmic dependence on H and h .

It is worth noticing that, within an iterative algorithm, for any given residue $\mathbf{r}^{(k)} = \chi_{\Gamma} - \Sigma \boldsymbol{\lambda}^{(k)}$, the vector $\mathbf{z}^{(k)} = (P_h^{BNN})^{-1} \mathbf{r}^{(k)}$ can be computed as follows

$$\mathbf{z}^{(k)} = \mathbf{z}^{(k,1/4)} + (P_h^{NN})^{-1} \mathbf{z}^{(k,1/2)} + \Sigma_H^{-1} \mathbf{z}^{(k,3/4)},$$

where we have defined $\mathbf{z}^{(k,1/4)} = \Sigma_H \mathbf{r}^{(k)}$, then $\mathbf{z}^{(k,1/2)} = \mathbf{r}^{(k)} - \Sigma \mathbf{z}^{(k,1/4)}$ and finally $\mathbf{z}^{(k,3/4)} = -\Sigma (P_h^{NN})^{-1} \mathbf{z}^{(k,1/2)}$.

On the other hand, the coarse grid matrix A_H that is a constituent of Σ_H can be built up using the Algorithm 14.4:

Algorithm 14.4 (construction of the coarse matrix for preconditioner P_h^{BNN})

- Build the restriction operator \bar{R}_0 that returns, for every subdomain, the weighted sum of the values at all the nodes at the boundary of that subdomain
For every node the corresponding weight is given by the inverse of the number of subdomains sharing that node
- Build up the matrix $A_H = \bar{R}_0 \Sigma \bar{R}_0^T$

Step a. of this Algorithm is computationally very cheap, whereas step b. requires several (e.g., ℓ) matrix-vector products involving the Schur complement matrix Σ . Since Σ is never built explicitly, this involves the finite element solution of $\ell \times M$ Dirichlet boundary value problems to generate A_H . Observe moreover that the restriction operator introduced at step a. implicitly defines a coarse space whose functions are piecewise constant on every Γ_i . For this reason the balanced Neumann-Neumann preconditioner is especially convenient when either the finite element grid or the subdomain partition (or both) are unstructured, as in Fig. 17.8).

By a comparison of the results obtained using the Neumann-Neumann preconditioner (with and without balancing), the following conclusions can be drawn:

- although featuring a better condition number than A , Σ is still ill-conditioned. The use of a suitable preconditioner is therefore mandatory;
- the Neumann-Neumann preconditioner can be satisfactorily used for partitions featuring a moderate number of subdomains;
- the balancing Neumann-Neumann preconditioner is almost optimally scalable and therefore recommandable for partitions with a large number of subdomains.

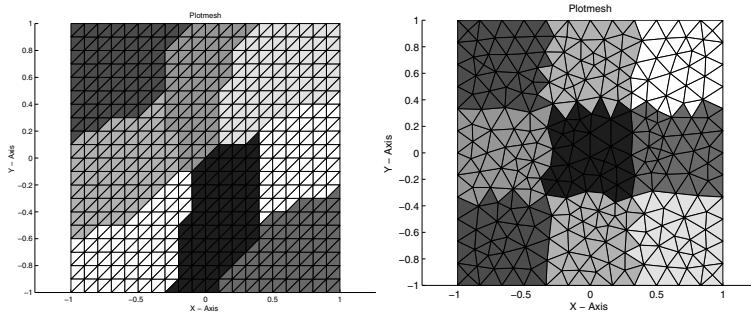


Fig. 17.8. Example of an unstructured subdomain partition in 8 subdomains for a finite element grid which is either structured (left) or unstructured (right)

Table 17.3. Condition number of $(P_h^{BNN})^{-1} \Sigma$ for several values of H

$K_2((P_h^{BNN})^{-1} \Sigma)$	$H = 1/2$	$H = 1/4$	$H = 1/8$	$H = 1/16$
$h = 1/16$	1.67	1.48	1.27	–
$h = 1/32$	2.17	2.03	1.47	1.29
$h = 1/64$	2.78	2.76	2.08	1.55
$h = 1/128$	3.51	3.67	2.81	2.07

17.6 Schwarz iterative methods

Schwarz method, in its original form described in Sec. 17.1.1, was proposed by H.Schwarz [Sch69] as an iterative scheme to prove existence of solutions to elliptic equations set in domains whose shape inhibits a direct application of Fourier series. Two elementary examples are displayed in Fig. 17.9. This method is still used in some quarters as solution method for elliptic equations in arbitrarily shaped domains. However, nowadays it is mostly used in a somehow different version, that of DD preconditioner of conjugate gradient (or, more generally, Krylov) iterations for the solution of algebraic systems arising from finite element (or other kind of) discretizations of boundary-value problems.

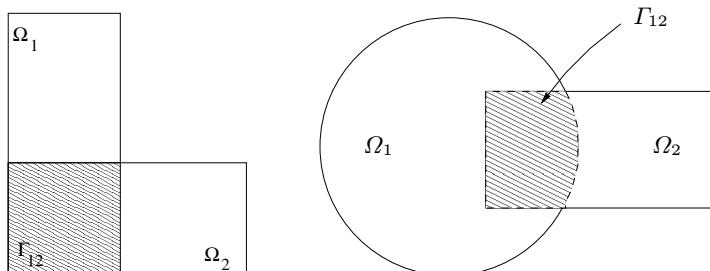


Fig. 17.9. Two examples for which the Schwarz method in its classical form applies

As seen in Sec. 17.1.1, the distinctive feature of Schwarz method is that it is based on an overlapping subdivision of the original domain. Let us still denote $\{\Omega_m\}$ these subdomains.

To start with, in the following subsection we will show how the Schwarz method can be formulated as an iterative algorithm to solve the algebraic system associated with the finite element discretization of problem (17.1).

17.6.1 Algebraic form of Schwarz method for finite element discretizations

Consider as usual a finite element triangulation \mathcal{T}_h of the domain Ω . Then assume that Ω is decomposed in two overlapping subdomains, Ω_1 and Ω_2 , as shown in Fig. 17.1 (left).

Denote with N_h the total number of nodes of the triangulation that are internal to Ω (i.e., they don't sit on its boundary), and with N_1 and N_2 , respectively, those internal to Ω_1 and Ω_2 , as done in Sec. 17.3. Note that $N_h \leq N_1 + N_2$ and that equality holds only if the overlap reduces to a single layer of elements. Indeed, if we denote with $I = \{1, \dots, N_h\}$ the set of indices of the nodes of Ω , and with I_1 and I_2 those associated with the internal nodes of Ω_1 and Ω_2 , respectively, one has $I = I_1 \cup I_2$, while $I_1 \cap I_2 \neq \emptyset$ unless the overlap consists of a single layer of elements.

Let us order the nodes in such a way that the first block corresponds to those in $\Omega_1 \setminus \Omega_2$, the second to those in $\Omega_1 \cap \Omega_2$, and the third to those in $\Omega_2 \setminus \Omega_1$. The stiffness matrix A of the finite element discretization contains two submatrices, A_1 and A_2 , corresponding to the local stiffness matrices in Ω_1 e Ω_2 , respectively (see Fig. 17.10). They are related to A as follows

$$A_1 = R_1 A R_1^T \in \mathbb{R}^{N_1 \times N_1} \quad \text{and} \quad A_2 = R_2 A R_2^T \in \mathbb{R}^{N_2 \times N_2}, \quad (17.79)$$

being R_i and R_i^T , for $i = 1, 2$, the restriction and prolongation operators, respectively. The matrix representation of the latter is

$$R_1^T = \begin{bmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \\ \mathbf{0} & & \end{bmatrix} \in \mathbb{R}^{N_h \times N_1}, \quad R_2^T = \begin{bmatrix} & & \\ & \mathbf{0} & \\ & 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{N_h \times N_2}. \quad (17.80)$$

If \mathbf{v} is a vector of \mathbb{R}^{N_h} , then $R_1 \mathbf{v}$ is a vector of \mathbb{R}^{N_1} whose components coincide with the first N_1 components of \mathbf{v} . Should \mathbf{v} instead be a vector of \mathbb{R}^{N_1} , then $R_1^T \mathbf{v}$ would be a vector of dimension N_h whose last $N_h - N_1$ components are all zero.

By using these definitions, an iteration of the multiplicative Schwarz method applied to system $A\mathbf{u} = \mathbf{f}$ can be expressed as follows:

$$\mathbf{u}^{(k+1/2)} = \mathbf{u}^{(k)} + R_1^T A_1^{-1} R_1 (\mathbf{f} - A\mathbf{u}^{(k)}), \quad (17.81)$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k+1/2)} + R_2^T A_2^{-1} R_2 (\mathbf{f} - A\mathbf{u}^{(k+1/2)}). \quad (17.82)$$

$$A = \begin{bmatrix} & & N_1 \\ & A_1 & \\ & & \\ \cdots & & & \\ & & & \\ & & & A_2 \\ & & & \\ & & & \\ & & & N_2 \end{bmatrix}_{N_h}$$

Fig. 17.10. The submatrices A_1 and A_2 of the stiffness matrix A

Equivalently, by setting

$$P_i = R_i^T A^{-1} R_i A, \quad i = 1, 2, \quad (17.83)$$

we have

$$\mathbf{u}^{(k+1/2)} = (I - P_1)\mathbf{u}^{(k)} + P_1\mathbf{u},$$

$$\mathbf{u}^{(k+1)} = (I - P_2)\mathbf{u}^{(k+1/2)} + P_2\mathbf{u} = (I - P_2)(I - P_1)\mathbf{u}^{(k)} + (P_1 + P_2 - P_2 P_1)\mathbf{u}.$$

Similarly, an iteration of the additive Schwarz method reads

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + (R_1^T A_1^{-1} R_1 + R_2^T A_2^{-1} R_2)(\mathbf{f} - A\mathbf{u}^{(k)}), \quad (17.84)$$

that is

$$\mathbf{u}^{(k+1)} = (I - P_1 - P_2)\mathbf{u}^{(k)} + (P_1 + P_2)\mathbf{u}. \quad (17.85)$$

Introducing the matrices

$$Q_i = R_i^T A_i^{-1} R_i = P_i A^{-1}, \quad i = 1, 2,$$

from (17.81) and (17.82) we derive the following recursive formula for the multiplicative Schwarz method

$$\begin{aligned} \mathbf{u}^{(k+1)} &= \mathbf{u}^{(k)} + Q_1(\mathbf{f} - A\mathbf{u}^{(k)}) + Q_2[\mathbf{f} - A(\mathbf{u}^{(k)} + Q_1(\mathbf{f} - A\mathbf{u}^{(k)}))] \\ &= \mathbf{u}^{(k)} + (Q_1 + Q_2 - Q_2 A Q_1)(\mathbf{f} - A\mathbf{u}^{(k)}), \end{aligned}$$

whereas for the additive Schwarz method we obtain from (17.84) that

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + (Q_1 + Q_2)(\mathbf{f} - A\mathbf{u}^{(k)}). \quad (17.86)$$

This last formula can easily be extended to the case of a decomposition of Ω into $M \geq 2$ overlapping subdomains $\{\Omega_i\}$ (see Fig. 17.11 for an example). In this case we have

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \left(\sum_{i=1}^M Q_i \right)(\mathbf{f} - A\mathbf{u}^{(k)}). \quad (17.87)$$

17.6.2 Schwarz preconditioners

Denoting with

$$P_{as} = \left(\sum_{i=1}^M Q_i \right)^{-1}, \quad (17.88)$$

from (17.87) it follows that an iteration of the additive Schwarz method corresponds to an iteration of the preconditioned Richardson method applied to the solution of the linear system $A\mathbf{u} = \mathbf{f}$ using P_{as} as preconditioner. For this reason the matrix P_{as} is named *additive Schwarz preconditioner*.

Equivalently, one iteration of the additive Schwarz method corresponds to an iteration by the Richardson method on the preconditioned linear system $Q_a \mathbf{u} = \mathbf{g}_a$, with $\mathbf{g}_a = P_{as}^{-1} \mathbf{f}$, and the preconditioned matrix Q_a is

$$Q_a = P_{as}^{-1} A = \sum_{i=1}^M P_i.$$

By proceeding similarly, using the multiplicative Schwarz method would yield the following preconditioned matrix

$$Q_M = P_{ms}^{-1} A = I - (I - P_M) \dots (I - P_1).$$

Lemma 17.3 Matrices P_i defined in (17.83) are symmetric and non-negative w.r.t the following scalar product induced by A

$$(\mathbf{w}, \mathbf{v})_A = (\mathbf{Aw}, \mathbf{v}) \quad \forall \mathbf{w}, \mathbf{v} \in \mathbb{R}^{N_h}.$$

Proof. For $i = 1, 2$, we have

$$\begin{aligned} (P_i \mathbf{w}, \mathbf{v})_A &= (AP_i \mathbf{w}, \mathbf{v}) = (R_i^T A_i^{-1} R_i A \mathbf{w}, \mathbf{v}) = (A \mathbf{w}, R_i^T A_i^{-1} R_i A \mathbf{v}) \\ &= (\mathbf{w}, P_i \mathbf{v})_A \quad \forall \mathbf{v}, \mathbf{w} \in \mathbb{R}^{N_h}. \end{aligned}$$

Moreover, $\forall \mathbf{v} \in \mathbb{R}^{N_h}$,

$$(P_i \mathbf{v}, \mathbf{v})_A = (AP_i \mathbf{v}, \mathbf{v}) = (R_i^T A_i^{-1} R_i A \mathbf{v}, \mathbf{v}) = (A_i^{-1} R_i A \mathbf{v}, R_i A \mathbf{v}) \geq 0.$$

◇

Lemma 17.4 The preconditioned matrix Q_a of the additive Schwarz method is symmetric and positive definite w.r.t the scalar product induced by A .

Proof. Let us first prove the symmetry: for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{N_h}$, since A and P_i are both symmetric, we obtain

$$\begin{aligned}(Q_a \mathbf{u}, \mathbf{v})_A &= (AQ_a \mathbf{u}, \mathbf{v}) = (Q_a \mathbf{u}, A\mathbf{v}) = \sum_i (P_i \mathbf{u}, A\mathbf{v}) \\ &= \sum_i (P_i \mathbf{u}, \mathbf{v})_A = \sum_i (\mathbf{u}, P_i \mathbf{v})_A = (\mathbf{u}, Q_a \mathbf{v})_A.\end{aligned}$$

Concerning the positivity, choosing in the former identities $\mathbf{u} = \mathbf{v}$, we obtain

$$(Q_a \mathbf{v}, \mathbf{v})_A = \sum_i (P_i \mathbf{v}, \mathbf{v})_A = \sum_i (R_i^T A_i^{-1} R_i A \mathbf{v}, A \mathbf{v}) = \sum_i (A_i^{-1} \mathbf{q}_i, \mathbf{q}_i) \geq 0,$$

having set $\mathbf{q}_i = R_i A \mathbf{v}$. It follows that $(Q_a \mathbf{v}, \mathbf{v})_A = 0$ iff $\mathbf{q}_i = \mathbf{0}$ for every i , that is iff $A \mathbf{v} = \mathbf{0}$. Since A is positive definite, this holds iff $\mathbf{v} = \mathbf{0}$. \diamond

Owing to the previous properties we can deduce that a more efficient iterative method can be generated by replacing the preconditioned Richardson iterations with the preconditioned conjugate gradient iterations, yet using the same additive Schwarz preconditioner P_{as} . Unfortunately, this preconditioner is not scalable. In fact, the condition number of the preconditioned matrix Q_a can only be bounded as

$$K_2(P_{as}^{-1} A) \leq C \frac{1}{\delta H}, \quad (17.89)$$

being C a constant independent of h , H and δ ; here δ is a characteristic linear measure of the overlapping regions and, as usual, $H = \max_{i=1,\dots,M} \{\text{diam}(\Omega_i)\}$. This is due to the fact that the exchange of information only occurs among neighbouring subdomains, as the application of $(P_{as})^{-1}$ involves only local solvers. This limitation can be overcome by introducing, also in the current context, a global coarse solver defined on the whole domain Ω and apt at guaranteeing a global communication among all of the subdomains. This leads to devise two-level domain decomposition strategies, see Sec. 17.6.3.

Let us address some algorithmic aspects. Let us subdivide the domain Ω in M subdomains $\{\Omega_i\}_{i=1}^M$ s.t. $\bigcup_{i=1}^M \overline{\Omega}_i = \overline{\Omega}$. Neighbouring subdomains share an overlapping region of size at least equal to $\delta = \xi h$, for a suitable $\xi \in \mathbb{N}$. In particular, $\xi = 1$ corresponds to the case of minimum overlap, that is the overlapping strip reduces to a single layer of finite elements. The following algorithm can be used.

Algorithm 14.5 (introduction of overlapping subdomains)

- Build a triangulation \mathcal{T}_h of the computational domain Ω
- Subdivide \mathcal{T}_h in M disjoint subdomains $\{\hat{\Omega}_i\}_{i=1}^M$
s.t. $\bigcup_{i=1}^M \overline{\hat{\Omega}}_i = \overline{\Omega}$
- Extend every subdomain $\hat{\Omega}_i$ by adding all the strips of finite elements of \mathcal{T}_h within a distance δ from $\hat{\Omega}_i$. These extended subdomains identify the family of overlapping subdomains Ω_i

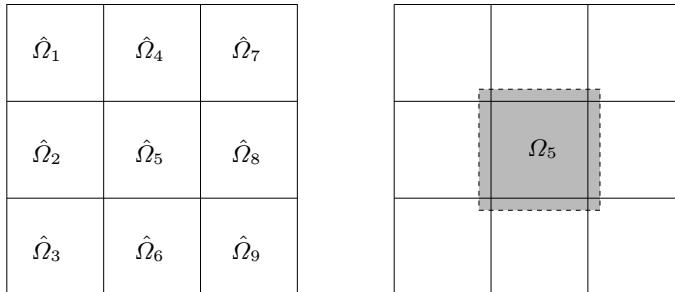


Fig. 17.11. Partition of a rectangular region Ω in 9 disjoint subregions $\hat{\Omega}_i$ (on the left), and an example of an extended subdomain Ω_5 (on the right)

In Fig. 17.11 a rectangular two-dimensional domain is subdivided into 9 disjoint subdomains $\hat{\Omega}_i$ (on the left); also shown is one of the extended (overlapping) subdomains (on the right).

To apply the Schwarz preconditioner (17.88) we can proceed as indicated in **Algorithm 14.5**. We recall that N_i is the number of internal nodes of Ω_i , R_i^T and R_i are the prolongation and restriction matrices, respectively, introduced in (17.80) and A_i are the local stiffness matrices introduced in (17.79). In Fig. 17.12 we display an example of sparsity pattern of R_i .

Algorithm 14.6 (start-up phase for the application of P_{as})

- Build on every subdomain in Ω_i the matrices R_i and R_i^T
- Build the stiffness matrix A corresponding to the finite element discretization on the grid \mathcal{T}_h
- On every Ω_i build the local submatrices $A_i = R_i A R_i^T$
- On every Ω_i set up the code for the solution of a linear system with matrix A_i .
For instance, compute a suitable (exact or incomplete) LU or Cholesky factorization of A_i

A few general comments on **Algorithm 14.5** and **Algorithm 14.6** are in order:

- steps a. and b. of algorithm 14.5 can be carried out in reverse order, that is we could first subdivide the computational domain into subdomains (based, for instance, on physical considerations), then set up a triangulation;

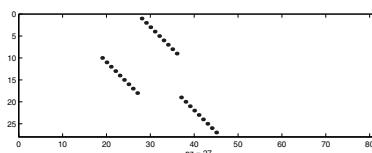


Fig. 17.12. The sparsity pattern of the matrix R_i for a partition of the domain in 4 subdomains

- depending upon the general code structure, steps b. and c. of the algorithm 14.6 could be glued together with the scope of optimizing memory requirements and CPU time.

In other circumstances we could interchange steps b. and c., that is the local stiffness matrices A_i can be built at first (using the single processors), then assembled to construct the global stiffness matrix A .

Indeed, a crucial factor for an efficient use of a parallel computer platform is keeping data locality since in most cases the time necessary for moving data among processors can be higher than that needed for computation.

Other codes (e.g. AztecOO, Trilinos, IFPACK) instead move from the global stiffness matrix distributed rowise and deduce the local stiffness matrices A_i without performing matrix-matrix products but simply using the column indices. In MATLAB, however, it seems more convenient to build A at first, next the restriction matrices R_i , and finally to carry out matrix multiplications $R_i A R_i^T$ to generate the A_i .

In Table 17.4 we analyze the case of a decomposition with minimum overlap ($\delta = h$), considering several values for the number M of subdomains. The subdomains Ω_i are overlapping squares of area H^2 . Note that the theoretical estimate (17.89) is satisfied by our results.

Table 17.4. Condition number of $P_{as}^{-1} A$ for several values of h and H

$K_2(P_{as}^{-1} A)$	$H = 1/2$	$H = 1/4$	$H = 1/8$	$H = 1/16$
$h = 1/16$	15.95	27.09	52.08	—
$h = 1/32$	31.69	54.52	104.85	207.67
$h = 1/64$	63.98	109.22	210.07	416.09
$h = 1/128$	127.99	218.48	420.04	832.57

17.6.3 Two-level Schwarz preconditioners

As anticipated in Sec. 17.6.2, the main limitation of Schwarz methods is to propagate information only among neighbouring subdomains. As for the Neumann-Neumann method, a possible remedy consists of introducing a coarse grid mechanism that allows for a sudden information diffusion on the whole domain Ω . The idea is still that of considering the subdomains as macro-elements of a new coarse grid \mathcal{T}_H and to build a corresponding stiffness matrix A_H . The matrix

$$Q_H = R_H^T A_H^{-1} R_H,$$

where R_H is the restriction operator from the fine to the coarse grid, represents the coarse level correction for the new two-level preconditioner. More precisely, setting for notational convenience $Q_0 = Q_H$, the two-level preconditioner P_{cas} is defined through its inverse as

$$P_{cas}^{-1} = \sum_{i=0}^M Q_i. \quad (17.90)$$

The following result can be proven: there exists a constant $C > 0$, independent of both h and H , s.t.

$$K_2(P_{cas}^{-1}A) \leq C\left(1 + \frac{H}{\delta}\right).$$

For “generous” overlap, that is if δ is a fraction of H , the preconditioner P_{cas} is scalable. Consequently, conjugate gradient iterations on the original finite element system using the preconditioner P_{cas} converges with a rate independent of h and H (and therefore of the number of subdomains). Moreover, thanks to the additive structure (17.90), the preconditioning step is fully parallel as it involves the solution of M independent systems, one per each local matrix A_i .

The use of P_{cas} involves the same kind of operations required by P_{as} , plus those of the following algorithm.

Algorithm 14.7 (start-up phase for the use of P_{cas})

- a. Execute **Algorithm 14.6**
- b. Define a coarse level triangulation \mathcal{T}_H whose elements are of the order of H , then set $n_0 = \dim(V_0)$. Suppose that \mathcal{T}_h be nested in \mathcal{T}_H . (See Fig. 17.13 for an example.)
- c. Build the restriction matrix $R_0 \in \mathbb{R}^{n_0 \times N_h}$ whose elements are

$$R_0(i, j) = \Phi_i(\mathbf{x}_j),$$

where Φ_i is the basis function associated to the node i of the coarse grid, while by \mathbf{x}_j we indicate the coordinates of the j -th node on the fine grid

- d. Build the coarse matrix A_H . This can be done by discretizing the original problem on the coarse grid \mathcal{T}_H , that is by computing

$$A_H(i, j) = a(\Phi_j, \Phi_i) = \int_{\Omega} \sum_{\ell=1}^d \frac{\partial \Phi_i}{\partial x_\ell} \frac{\partial \Phi_j}{\partial x_\ell},$$

or, otherwise, by setting

$$A_H = R_H A R_H^T.$$

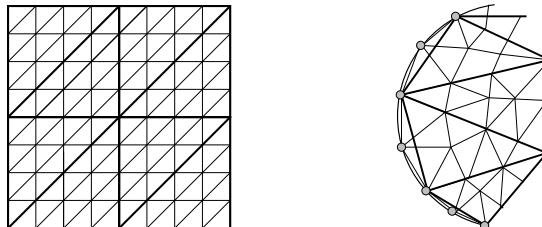


Fig. 17.13. On the left, example of a coarse grid for a 2D domain, based on a structured mesh. The triangles of the fine grid has thin edges; thick edges identify the boundaries of the coarse grid elements. On the right, a similar construction is displayed, this time for an unstructured fine grid

For a computational domain with a simple shape (like the one we are considering) one typically generates the coarse grid \mathcal{T}_H first, and then, by multiple refinements, the fine grid \mathcal{T}_h . In other cases, when the domain has a complex shape and/or a non structured fine grid \mathcal{T}_h is already available, the generation of a coarse grid might be difficult or computationally expensive. A first option would be to generate \mathcal{T}_H by successive derefinitions of the fine grid, in which case the nodes of the coarse grid will represent a subset of those of the fine grid. This approach, however, might not be very efficient in 3D.

Alternatively, one could generate the two (not necessarily nested) grids \mathcal{T}_h and \mathcal{T}_H independently, then generate the corresponding restriction and prolongation operators from the fine to the coarse grid, R_H and R_H^T .

The final implementation of P_{cas} could therefore be made as follows:

Algorithm 14.8 (P_{cas} solve)

For any given vector \mathbf{r} , the computation of $\mathbf{z} = P_{cas}^{-1}\mathbf{r}$ can be carried out as follows:

- a. Set $\mathbf{z} = \mathbf{0}$
- b. For $i = 1, \dots, M$ Do in parallel:
 - c. restrict the residue on Ω_i : $\mathbf{r}_i = R_i \mathbf{r}$
 - d. compute $\mathbf{z}_i : A_i \mathbf{z}_i = \mathbf{r}_i$
 - e. add to the global residue: $\mathbf{z} \leftarrow R_i^T \mathbf{z}_i$
- f. EndFor
- g. Compute the coarse grid contribution $\mathbf{z}_H : A_H \mathbf{z}_H = R_H \mathbf{r}$
- h. Add to the global residue: $\mathbf{z} \leftarrow R_H^T \mathbf{z}_H$

In Table 17.5 we report the condition number of $P_{cas}^{-1}A$ in the case of a minimum overlap $\delta = h$. Note that the condition number is almost the same on each NW-SE diagonal (i.e. for fixed values of the ratio H/δ).

Table 17.5. Condition number of $P_{cas}^{-1}A$ for several values of h and H

$K_2(P_{cas}^{-1}A)$	$H = 1/4$	$H = 1/8$	$H = 1/16$	$H = 1/32$
$h = 1/32$	7.03	4.94	–	–
$h = 1/64$	12.73	7.59	4.98	–
$h = 1/128$	23.62	13.17	7.66	4.99
$h = 1/256$	45.33	24.34	13.28	–

An alternative approach to the coarse grid correction can be devised as follows. Suppose that the coefficients of the restriction matrix be given by

$$\hat{R}_H(i, j) = \begin{cases} 1 & \text{if the } j\text{-th node is in } \Omega_i, \\ 0 & \text{otherwise,} \end{cases}$$

then we set $\hat{A}_H = \hat{R}_H A \hat{R}_H^T$. This procedure is named *aggregation* because the elements of \hat{A}_H are obtained by simply summing up the entries of A . Note that we don't need to construct a coarse grid in this case. The corresponding preconditioner, denoted by P_{aggre} , has an inverse that reads

$$P_{\text{aggre}}^{-1} = \hat{R}_H^T \hat{A}_H^{-1} \hat{R}_H + P_{\text{as}}.$$

It can be proven that

$$K_2(P_{\text{aggre}}^{-1} A) \leq C \left(1 + \frac{H}{\delta} \right).$$

In Table 17.6 we report several numerical values of the condition number for different values of h and H .

Table 17.6. Condition number of $P_{\text{aggre}}^{-1} A$ for several values of h and H

$P_{\text{aggre}}^{-1} A$	$H = 1/4$	$H = 1/8$	$H = 1/16$
$h = 1/16$	13.37	8.87	–
$h = 1/32$	26.93	17.71	9.82
$h = 1/64$	54.33	35.21	19.70
$h = 1/128$	109.39	70.22	39.07

If $H/\delta = \text{constant}$, this two-level preconditioner is either optimal and scalable, that is the condition number of the preconditioned stiffness matrix is independent of both h and H .

We can conclude this section with the following practical indications:

- for decompositions with a small number of subdomains, the single level Schwarz preconditioner P_{as} is very efficient;
- when the number M of subdomains gets large, using two-level preconditioners becomes crucial; aggregation techniques can be adopted, in alternative to the use of a coarse grid in those cases in which the generation of the latter is difficult.

17.7 An abstract convergence result

The analysis of overlapping and non-overlapping domain decomposition preconditioners is based on the following abstract theory, due to P.L. Lions, J. Bramble, M. Dryja, O. Widlund.

Let V_h be a Hilbert space of finite dimension. In our applications, V_h is one of the finite element spaces or spectral element spaces. Let V_h be decomposed as follows:

$$V_h = V_0 + V_1 + \cdots + V_M.$$

Let $F \in V'$ and $a : V \times V \rightarrow \mathbb{R}$ be a symmetric, continuous and coercive bilinear form. Consider the problem

$$\text{find } u_h \in V_h : a(u_h, v_h) = F(v_h) \quad \forall v_h \in V_h.$$

Let $P_i : V_h \rightarrow V_i$ be a projection operator defined by

$$b_i(P_i u_h, v_h) = a(u_h, v_h) \quad \forall v_h \in V_i$$

with $b_i : V_i \times V_i \rightarrow \mathbb{R}$ being a local symmetric, continuous and coercive bilinear form on each subspace V_i . Assume that the following properties hold.

a. **Stable subspace decomposition:**

$\exists C_0 > 0$ s.t. every $u_h \in V_h$ admits a decomposition $u_h = \sum_{i=1}^M u_i$ with $u_i \in V_i$ and

$$\sum_{i=0}^M b_i(u_i, u_i) \leq C_0^2 a(u_h, u_h);$$

b. **strengthened Cauchy-Schwarz inequality:**

$\exists \epsilon_{ij} \in [0, 1]$, $i, j = 1, \dots, M$ s.t.

$$a(u_i, u_i) \leq \epsilon_{ij} \sqrt{a(u_i, u_i)} \sqrt{a(u_j, u_j)} \quad \forall u_i \in V_i, u_j \in V_j;$$

c. **local stability:**

$\exists \omega \geq 1$ s.t. $\forall i = 0, \dots, M$

$$a(u_i, u_i) \leq \omega b_i(u_i, u_i) \quad \forall u_i \in Range(P_i) \subset V_i.$$

Then, $\forall u_h \in V_h$,

$$C_0^{-2} a(u_h, u_h) \leq a(P_{as} u_h, u_h) \leq \omega(\rho(E) + 1) a(u_h, u_h)$$

where $\rho(E)$ is the spectral radius of the matrix $E = (\epsilon_{ij})$, and $P_{as} = P_0 + \dots + P_M$ is the domain decomposition preconditioner.

For the proof, see e.g. [TW05].

17.8 Interface conditions for other differential problems

Theorem 17.1 in Sec. 17.1.2 allows a second order elliptic problem (17.1) to be reformulated in a DD version thanks to suitable interface conditions (17.9) and (17.10). On the other hand, as we have extensively discussed, such reformulation sets the ground for several iterative algorithms on disjoint DD partitions. They comprise Dirichlet-Neumann, Neumann-Neumann, Robin-Robin algorithms and, more generally, all of the preconditioned iterative algorithms of the Schur complement system (17.50) using suitable DD preconditioners.

In this section we consider other kind of boundary-value problems and formulate the associated interface conditions. Table 17.7 displays the interface conditions for these problems. For more details, analysis and investigation of associated iterative DD algorithms, the interested reader can consult [QV99].

Here we limit ourselves to provide a few additional insights in the case of advection and Stokes equations.

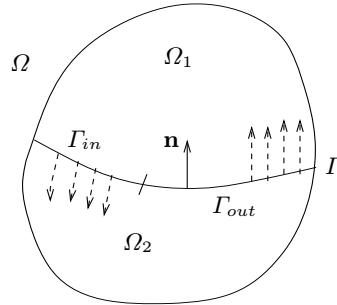


Fig. 17.14. Domain partition and interface splitting for the advection problem (17.91)

Advection (transport) problems. Consider the differential problem

$$Lu = \nabla \cdot (\mathbf{b}u) + a_0 u = f \quad \text{in } \Omega, \quad (17.91)$$

supplemented by suitable conditions on the boundary $\partial\Omega$. Consider a partition of the computational domain Ω into two disjoint subdomains whose interface is Γ . Let us partition the latter as follows (see Fig. 17.14): $\Gamma = \Gamma_{in} \cup \Gamma_{out}$, where

$$\Gamma_{in} = \{x \in \Gamma \mid \mathbf{b}(x) \cdot \mathbf{n}(x) > 0\} \quad \text{and} \quad \Gamma_{out} = \Gamma \setminus \Gamma_{in}.$$

Example 17.4 The Dirichlet-Neumann method for the problem at hand could be generalized as follows: being given two functions $u_1^{(0)}, u_2^{(0)}$ on Γ , $\forall k \geq 0$ solve:

$$\begin{cases} Lu_1^{(k+1)} = f & \text{in } \Omega_1, \\ (\mathbf{b} \cdot \mathbf{n})u_1^{(k+1)} = (\mathbf{b} \cdot \mathbf{n})u_2^{(k)} & \text{on } \Gamma_{out}, \\ \\ Lu_2^{(k+1)} = f & \text{in } \Omega_2, \\ (\mathbf{b} \cdot \mathbf{n})u_2^{(k+1)} = \theta(\mathbf{b} \cdot \mathbf{n})u_1^{(k)} + (1 - \theta)(\mathbf{b} \cdot \mathbf{n})u_2^{(k)} & \text{on } \Gamma_{in}. \end{cases}$$

where $\theta > 0$ denotes a suitable relaxation parameter. The adaptation to the case of a finite element discretization is straightforward. ■

Stokes problem. The Stokes equations (15.11) feature two fields of variables: fluid velocity and fluid pressure. When considering a DD partition, at subdomain interface only the velocity field is requested to be continuous. Pressure needs not necessarily be continuous, since in the weak formulation of the Stokes equations it is “only” requested to be in L^2 . Moreover, on the interface Γ the continuity of the normal Cauchy stress $\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n}$ needs only be satisfied in weak (natural) form.

Example 17.5 A Dirichlet-Neumann algorithm for the Stokes problem would entail at each iteration the solution of the following subproblems (we use the short-hand notation \mathcal{S} to indicate

Table 17.7. Interface continuity conditions for several kind of differential operators; D stands for Dirichlet condition, N for Neumann

Operator	Problem	D	N
Laplace	$-\Delta u = f,$	u	$\frac{\partial u}{\partial n}$
Elasticity	$-\nabla \cdot (\sigma(\mathbf{u})) = \mathbf{f},$ with $\sigma_{kj} = \hat{\mu}(D_k u_j + D_j u_k) + \hat{\lambda} \operatorname{div} \mathbf{u} \delta_{kj},$ \mathbf{u} in-plane membrane displacement		$\sigma(\mathbf{u}) \cdot \mathbf{n}$
Transport-diffusion	$-\sum_{kj} D_k (A_{kj} D_j u) + \operatorname{div}(\mathbf{b} u) + a_0 u = f$ u		$\frac{\partial u}{\partial n_L}$ $= \sum_k a_{kj} D_j u \cdot n_k$
Transport	$\operatorname{div}(\mathbf{b} u) + a_0 u = f$		$\mathbf{b} \cdot \mathbf{n} u$
Incompressible viscous flows	$-\operatorname{div} \mathbf{T}(\mathbf{u}, p) + (\mathbf{u}^* \cdot \nabla) \mathbf{u} = \mathbf{f},$ $\operatorname{div} \mathbf{u} = 0,$ with $\mathbf{T}_{kj} = \nu(D_k u_j + D_j u_k) - p \delta_{kj},$ \mathbf{u} $\mathbf{u}^* = \begin{cases} 0 & \text{(Stokes equations)} \\ \mathbf{u}_\infty & \text{(Oseen equations)} \\ \mathbf{u} & \text{(Navier-Stokes equations)} \end{cases}$		$\mathbf{T}(\mathbf{u}, p) \cdot \mathbf{n}$
Compressible viscous flows	$\alpha \mathbf{u} - \operatorname{div} \hat{\mathbf{T}}(\mathbf{u}, \sigma) = \mathbf{f},$ $\alpha \sigma + \operatorname{div} \mathbf{u} = g,$ with $\hat{\mathbf{T}}_{kj} = \nu(D_k u_j + D_j u_k) - \beta \sigma \delta_{kj} + \left(g - \frac{2\nu}{d}\right) \operatorname{div} \mathbf{u} \delta_{kj},$ $\rho = \text{fluid density} = \log \sigma$	\mathbf{u}	$\hat{\mathbf{T}}(\mathbf{u}, \sigma) \cdot \mathbf{n}$
Compressible inviscid flows	$\alpha \mathbf{u} + \beta \nabla \sigma = \mathbf{f},$ $\alpha \sigma + \operatorname{div} \mathbf{u} = 0$	$\mathbf{u} \cdot \mathbf{n}$	σ
Maxwell (harmonic regime)	$\operatorname{rot} \left(\frac{1}{\mu} \operatorname{rot} \mathbf{E} \right)$ $-\alpha^2 \varepsilon \mathbf{E} + i \alpha \sigma \mathbf{E} = \mathbf{f}$	$\mathbf{n} \times \mathbf{E}$	$\mathbf{n} \times \left(\frac{1}{\mu} \operatorname{rot} \mathbf{E} \right)$

the Stokes operator):

$$\begin{cases} \mathcal{S}(\mathbf{u}_2^{(k+1)}, p_2^{(k+1)}) = \mathbf{f} & \text{in } \Omega_2, \\ \nu \frac{\partial \mathbf{u}_2^{(k+1)}}{\partial n} - p_2^{(k+1)} = \nu \frac{\partial \mathbf{u}_1^{(k)}}{\partial n} - p_1^{(k)} & \text{on } \Gamma, \\ \mathbf{u}_2^{(k+1)} = \mathbf{0} & \text{on } \partial\Omega_2 \setminus \Gamma, \end{cases} \quad (17.92)$$

$$\begin{cases} \mathcal{S}(\mathbf{u}_1^{(k+1)}, p_1^{(k+1)}) = \mathbf{f} & \text{in } \Omega_1, \\ \mathbf{u}_1^{(k+1)} = \theta \mathbf{u}_2^{(k+1)} + (1 - \theta) \mathbf{u}_1^{(k)} & \text{on } \Gamma, \\ \mathbf{u}_1^{(k+1)} = \mathbf{0} & \text{on } \partial\Omega_1 \setminus \Gamma. \end{cases} \quad (17.93)$$

Should the boundary conditions of the original problem be prescribed on the velocity field, e.g. $\mathbf{u} = \mathbf{0}$, pressure p would be defined only up to an additive constant, which could be fixed by, e.g., imposing the constraint $\int_{\Omega} p \, d\Omega = 0$.

To fulfill this constraint we can proceed as follows. When solving the Neumann problem (17.92) on the subdomain Ω_2 , both the velocity $\mathbf{u}_2^{(k+1)}$ and the pressure $p_2^{(k+1)}$ are univocally determined. When solving the Dirichlet problem (17.93) on Ω_1 , the pressure is defined only up to an additive constant; we fix it by imposing the additional equation

$$\int_{\Omega_1} p_1^{(k+1)} \, d\Omega_1 = - \int_{\Omega_2} p_2^{(k+1)} \, d\Omega_2.$$

Should the four sequences $\{\mathbf{u}_1^{(k)}\}$, $\{\mathbf{u}_2^{(k)}\}$, $\{p_1^{(k)}\}$ and $\{p_2^{(k)}\}$ converge, the null average condition on the pressure would be automatically verified. ■

Example 17.6 Suppose now that the Schwarz iterative method is used on an overlapping subdomain decomposition of the domain like that on Fig. 17.1, left. At every step we have to solve two Dirichlet problems for the Stokes equations:

$$\begin{cases} \mathcal{S}(\mathbf{u}_1^{(k+1)}, p_1^{(k+1)}) = \mathbf{f} & \text{in } \Omega_1, \\ \mathbf{u}_1^{(k+1)} = \mathbf{u}_2^{(k)} & \text{on } \Gamma_1, \\ \mathbf{u}_1^{(k+1)} = \mathbf{0} & \text{on } \partial\Omega_1 \setminus \Gamma_1, \end{cases} \quad (17.94)$$

$$\begin{cases} \mathcal{S}(\mathbf{u}_2^{(k+1)}, p_2^{(k+1)}) = \mathbf{f} & \text{in } \Omega_2, \\ \mathbf{u}_2^{(k+1)} = \mathbf{u}_1^{(k+1)} & \text{on } \Gamma_2, \\ \mathbf{u}_2^{(k+1)} = \mathbf{0} & \text{on } \partial\Omega_2 \setminus \Gamma_2. \end{cases} \quad (17.95)$$

No continuity is required on the pressure field at subdomain boundaries.

The constraint on the fluid velocity to be divergence free on the whole domain Ω requires special care. Indeed, after solving (17.94), we have $\operatorname{div} \mathbf{u}_1^{(k+1)} = 0$ in Ω_1 , hence, thanks to the Green formula,

$$\int_{\partial\Omega_1} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n} \, d\gamma = 0.$$

This relation implies a similar relation for $\mathbf{u}_2^{(k)}$ in (17.94)₂; indeed

$$0 = \int_{\partial\Omega_1} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n} \, d\gamma = \int_{\Gamma_1} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n} \, d\gamma = \int_{\Gamma_1} \mathbf{u}_2^{(k)} \cdot \mathbf{n} \, d\gamma. \quad (17.96)$$

At the very first iteration we can select $\mathbf{u}_2^{(0)}$ in such a way that the compatibility condition (17.96) be satisfied, however this control is lost, a priori, in the course of the subsequent iterations. For the same reason, the solution of (17.95) yields the compatibility condition

$$\int_{\Gamma_2} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n} d\gamma = 0. \quad (17.97)$$

Fortunately, Schwarz method automatically guarantees that this condition holds. Indeed, in $\Gamma_{12} = \Omega_1 \cap \Omega_2$ we have $\operatorname{div} \mathbf{u}_1^{(k+1)} = 0$, moreover on $\Gamma_{12} \setminus (\Gamma_1 \cup \Gamma_2)$, $\mathbf{u}_1^{(k+1)} = \mathbf{0}$ because of the given homogeneous Dirichlet boundary conditions. Thus

$$0 = \int_{\partial\Gamma_{12}} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n} d\gamma = \int_{\Gamma_1} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n} d\gamma + \int_{\Gamma_2} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n} d\gamma.$$

The first integral on the right hand side vanishes because of (17.96), therefore (17.97) is satisfied. ■

17.9 Exercises

1. Consider the one-dimensional advection-transport-reaction problem

$$\begin{cases} -(\alpha u_x)_x + (\beta u)_x + \gamma u = f & \text{in } \Omega = (a, b) \\ u(a) = 0, \quad \alpha u_x(b) - \beta u(b) = g, \end{cases} \quad (17.98)$$

with α and $\gamma \in L^\infty(a, b)$, $\beta \in W^{1,\infty}(a, b)$ and $f \in L^2(a, b)$.

- a) Write the additive Schwarz iterative method, then the multiplicative one, on the two overlapping intervals $\Omega_1 = (a, \gamma_2)$ and $\Omega_2 = (\gamma_1, b)$, with $a < \gamma_1 < \gamma_2 < b$.
 - b) Interpret these methods as suitable Richardson algorithms to solve the given differential problem.
 - c) In case we approximate (17.98) by the finite element method, write the corresponding additive Schwarz preconditioner, with and without coarse-grid component. Then provide an estimate of the condition number of the pre-conditioned matrix, in both cases.
2. Consider the one-dimensional diffusion-transport-reaction problem

$$\begin{cases} -(\alpha u_x)_x + (\beta u)_x + \delta u = f & \text{in } \Omega = (a, b) \\ \alpha u_x(a) - \beta u(a) = g, \quad u_x(b) = 0, \end{cases} \quad (17.99)$$

with α and $\gamma \in L^\infty(a, b)$, $\alpha(x) \geq \alpha_0 > 0$, $\beta \in W^{1,\infty}(a, b)$, $f \in L^2(a, b)$ and g a given real number.

- a) Consider two disjoined subdomains of Ω , $\Omega_1 = (a, \gamma)$ and $\Omega_2 = (\gamma, b)$, with $a < \gamma < b$. Formulate problem (17.99) using the Steklov-Poincaré operator, both in differential and variational form. Analyze the properties of this operator starting from those of the bilinear form associated with problem (17.99).
- b) Apply the Dirichlet-Neumann method to problem (17.99) using the same domain partition introduced at point a).
- c) In case of finite element approximation, derive the expression of the Dirichlet-Neumann preconditioner of the Schur complement matrix.
3. Consider the one-dimensional Poisson problem

$$\begin{cases} -u_{xx}(x) = f(x) & \text{in } \Omega = (0, 1) \\ u(0) = 0, \quad u_x(1) = 0, \end{cases} \quad (17.100)$$

with $f \in L^2(\Omega)$.

- a) If \mathcal{T}_h indicates a partition of the interval Ω with step-size h , write the Galerkin-finite element approximation of problem (17.100).
- b) Consider now a partition of Ω into the subintervals $\Omega_1 = (0, \gamma)$ and $\Omega_2 = (\gamma, 1)$, being $0 < \gamma < 1$ a node of the partition \mathcal{T}_h (See Fig. 17.15). Write the algebraic blockwise form of the Galerkin-finite element stiffness matrix relative to this subdomain partition.

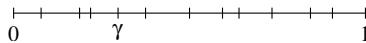


Fig. 17.15. Subdomain partition \mathcal{T}_h of the interval $(0, 1)$

- c) Derive the discrete Steklov-Poincaré interface equation which corresponds to the DD formulation at point b). Which is the dimension of the Schur complement?
- d) Consider now two overlapping subdomains $\Omega_1 = (0, \gamma_2)$ and $\Omega_2 = (\gamma_1, 1)$, with $0 < \gamma_1 < \gamma_2 < 1$, the overlap being reduced to a single finite element of the partition \mathcal{T}_h (see Fig. 17.16). Provide the algebraic formulation of the additive Schwarz iterative method.

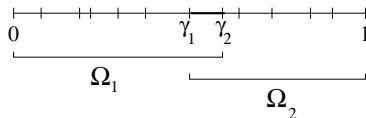


Fig. 17.16. Overlapping decomposition of the interval $(0, 1)$

- e) Provide the general expression of the two-level additive Schwarz preconditioner, by assuming as coarse matrix A_H that associated with only two elements, as displayed in Fig. 17.17.

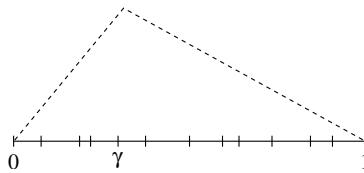


Fig. 17.17. Coarse-grid partition made of two macro elements for the construction of matrix A_H and Lagrangian characteristic function associated with the node γ

4. Consider the diffusion-transport-reaction problem

$$\begin{cases} Lu = -\nabla \cdot (\alpha \nabla u) + \nabla \cdot (\beta u) + \gamma u = f & \text{in } \Omega = (0, 2) \times (0, 1), \\ u = 0 & \text{on } \Gamma_D, \\ \alpha \frac{\partial u}{\partial n} + \delta u = 0 & \text{on } \Gamma_R, \end{cases} \quad (17.101)$$

with $\alpha = \alpha(\mathbf{x})$, $\beta = \beta(\mathbf{x})$, $\gamma = \gamma(\mathbf{x})$, $\delta = \delta(\mathbf{x})$ and $f = f(\mathbf{x})$ being given functions, and $\partial\Omega = \overline{\Gamma}_D \cup \overline{\Gamma}_R$, with $\overline{\Gamma}_D \cap \overline{\Gamma}_R = \emptyset$.

Let Ω in (17.101) be partitioned into two disjoined subdomains $\Omega_1 = (0, 1) \times (0, 1)$ and $\Omega_2 = (1, 2) \times (0, 1)$.

- a) Formulate problem (17.101) in terms of the Steklov-Poincaré operator, both in differential and variational form.
- b) Apply the Dirichlet-Neumann method to problem (17.101) using the same decomposition introduced before.
- c) Prove the equivalence between the Dirichlet-Neumann method at point b) and a suitable preconditioned Richardson operator, after setting $\alpha = 1$, $\beta = 0$, $\gamma = 1$ and $\Gamma_R = \emptyset$ in (17.101). Do the same for the Neumann-Neumann method.

5. Consider the two-dimensional diffusion-transport-reaction problem

$$\begin{cases} Lu = -\nabla \cdot (\mu \nabla u) + \mathbf{b} \cdot \nabla u + \sigma u = f & \text{in } \Omega = (a, c) \times (d, e), \\ u = 0 & \text{on } \partial\Omega. \end{cases} \quad (17.102)$$

Consider a decomposition of Ω made of the overlapping subdomains $\Omega_3 = (a, f) \times (d, e)$ and $\Omega_4 = (g, c) \times (d, e)$, with $g < f$. On such a decomposition, write for problem (17.102) the Schwarz method in both multiplicative and additive versions. Then interpret these methods as suitable preconditioned Richardson iterative algorithms. Finally, comment on the convergence properties of these methods.

Reduced basis approximation for parametrized partial differential equations

In this chapter we describe the basic ideas of reduced basis (RB) approximation methods for rapid and reliable evaluation of *input-output relationships* in which the *output* is expressed as a functional of a *field variable* that is the solution of an *input-parametrized* partial differential equation (PDE). We shall focus on linear output functionals and affinely parametrized linear elliptic coercive PDEs; however the methodology is much more generally applicable as we discuss in Sec. 18.6 at the end of this chapter. The combination with an efficient a posteriori error estimation is a key factor for RB methods to be computationally successful.

Parametrized PDEs model several processes that are relevant in applications, such as, e.g., unsteady and steady heat and mass transfer, acoustics, and solid and fluid mechanics, but also electromagnetics or even finance. The *input-parameter* vector may characterize either the geometric configuration, some physical properties, or boundary conditions and source terms. The *outputs of interest* might be the maximum system temperature, an added mass coefficient, a crack stress intensity factor, an effective constitutive property, an acoustic waveguide transmission loss, or a channel flowrate or pressure drop, just to mention a few. Finally, the *field variables* that connect the input parameters to the outputs can represent a distribution function, temperature or concentration, displacement, pressure, or velocity.

In this chapter we shall consider the case of linear functional outputs of affinely parametrized linear elliptic coercive PDEs. This class of problems — relatively simple, yet relevant to many important applications in transport (e.g., conduction and convection-diffusion) and continuum mechanics (e.g., linear elasticity) — proves a convenient expository vehicle for the methodology.

Although our focus here is on the affine linear elliptic coercive case, the reduced basis approximation and a posteriori error estimation methodology is much more general. Furthermore, most of the basic concepts introduced in the affine linear elliptic coercive case are equally crucial — with suitable extension — to more general equations, see e.g. [RHP08] and [PR07].

The RB methodology we describe in this chapter is motivated by, optimized for, and applied within, two particular contexts: the *real-time context* (e.g., parameter-estimation [GNV⁺07] or control [QRQ06]); and the *many-query context* (e.g., design

optimization or multi-model/scale simulation). Both are crucial to computational engineering. We note, however, that the RB methods we describe do not replace, but rather build upon and are measured (as regards accuracy) relative to, a finite element model: the reduced basis approximates not the exact solution but rather a “given” finite element discretization of (typically) very large dimension N_h . In short, we pursue an algorithmic collaboration rather than an algorithmic competition with the finite element method.

We provide here a brief chapter outline. In Sec. 18.1 we describe the affine linear elliptic coercive setting; in Sec. 18.2 we consider admissible classes of piecewise-affine geometry and coefficient parametric variation; both in Sec. 18.1 and 18.2 we introduce several “working examples” which shall serve to illustrate the formulation and methodology. Sec. 18.3 is strictly devoted to reduced basis methodology, in particular in Sec. 18.3.1 we discuss RB Galerkin projection and optimality; in Sec. 18.3.2 we describe greedy sampling procedures for optimal space identification; in Sec. 18.4.1 and Sec. 18.4.2 we investigate the convergence theory and practice for one parameter and for many parameters, respectively.

In Sec. 18.5 we present rigorous and relatively sharp a posteriori output error bounds for RB approximations but without developing the coercivity-constant lower bounds required by the a posteriori error estimation procedures. In Sec. 18.6 we make comments on historical background and future perspectives of RB methodology.

18.1 Elliptic coercive parametric PDEs

In an abstract form, we consider the following problem: Given $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^P$, for an integer $P \geq 1$, evaluate

$$s^e(\boldsymbol{\mu}) = L(u^e(\boldsymbol{\mu})) ,$$

where $u^e(\boldsymbol{\mu}) \in V^e(\Omega)$ satisfies

$$a(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = F(v) \quad \forall v \in V^e . \quad (18.1)$$

The superscript e refers to “exact.” Here $\boldsymbol{\mu}$ is the input parameter — a P -tuple; \mathcal{D} is the parameter domain — a subset of \mathbb{R}^P ; s^e is the scalar output; L is the linear output functional, that is a linear and bounded functional on V^e ; u^e is the field variable; Ω is bounded spatial domain in \mathbb{R}^d (for $d = 2$ or 3) with Lipschitz boundary $\partial\Omega$; V^e is a Hilbert space; and a and F are the bilinear and linear forms, respectively, associated with our PDE.

We shall exclusively consider second-order elliptic partial differential equations, in which case $V^e = H_{\Gamma_D}^1(\Omega)$ is the subspace of $H_0^1(\Omega)$, the one of functions that vanish on the part of $\partial\Omega$ where Dirichlet boundary data are prescribed for u^e (see (3.25)).

We shall assume that the bilinear form $a(\cdot, \cdot; \boldsymbol{\mu}): V^e \times V^e \rightarrow \mathbb{R}$ is continuous (with continuity constant $\gamma^e(\boldsymbol{\mu})$) and coercive (with coercivity constant $\alpha^e(\boldsymbol{\mu})$) over V^e for all $\boldsymbol{\mu}$ in \mathcal{D} . We further assume that F is a bounded linear functional over V^e . Under these standard hypotheses on a and F , (18.1) admits a unique solution, thanks to Lax-Milgram lemma.

We assume that (i) a is symmetric and furthermore (ii) $L = F$. The last assumption is made for simplification and it means that we are in the so called *compliant* case [PR07], a situation occurring quite frequently in engineering problems (see Sec. 18.1.1). extension to the non-compliant case in which now a may be non-symmetric and L may be any bounded linear functional over V^e is described in [RHP08].

We shall make one last assumption, crucial to the enhancement of computational efficiency: the parametric bilinear form a is *affine* w.r.t. the parameter μ , by which we mean

$$a(w, v; \mu) = \sum_{q=1}^Q \Theta^q(\mu) a^q(w, v) \quad \forall v, w \in V^e, \mu \in \mathcal{D}. \quad (18.2)$$

Here, for $q = 1, \dots, Q$, $\Theta^q: \mathcal{D} \rightarrow \mathbb{R}$ is a μ -dependent function, whereas $a^q: V^e \times V^e \rightarrow \mathbb{R}$ is μ -independent. In actual practice, F may also depend affinely on the parameter: in this case, $F(v; \mu)$ may be expressed as a sum of Q^f products of parameter-dependent functions and parameter-independent bounded linear functionals on V^e .

We next proceed to the finite dimensional approximation of problem (18.1) by any kind of Galerkin method, for instance the finite element method: Given $\mu \in \mathcal{D} \subset \mathbb{R}^P$, evaluate

$$s^{N_h}(\mu) = F(u^{N_h}(\mu))$$

(recall our compliance assumption: $L = F$), where $u^{N_h}(\mu) \in V^{N_h} \subset V^e$ satisfies

$$a(u^{N_h}(\mu), v; \mu) = F(v) \quad \forall v \in V^{N_h}. \quad (18.3)$$

Here $V^{N_h} \subset V^e$ is a sequence of FE approximation spaces indexed by $\dim(V^{N_h}) = N_h$. It follows directly from our assumptions on a , F , and V^{N_h} that (18.3) admits a unique solution. Our RB field and RB output shall approximate, for given N_h , the FE solution $u^{N_h}(\mu)$ and FE output $s^{N_h}(\mu)$ (hence, indirectly, $u^e(\mu)$ and $s^e(\mu)$).

We can now define the energy inner product and the energy norm for elements of V^e :

$$(w, v)_\mu = a(w, v; \mu) \quad \forall w, v \in V^e, \quad (18.4)$$

$$\|w\|_\mu = (w, w)_\mu^{1/2} \quad \forall w \in V^e. \quad (18.5)$$

Next, for given $\bar{\mu} \in \mathcal{D}$ and non-negative real τ ,

$$(w, v)_V = (w, v)_{\bar{\mu}} + \tau(w, v)_{L^2(\Omega)} \quad \forall w, v \in V^e, \quad (18.6)$$

$$\|w\|_V = (w, w)_V^{1/2} \quad \forall w \in V^e,$$

shall define our V inner product and norm, respectively.

Finally, we can now define more precisely our coercivity and continuity constants (and coercivity and continuity conditions). In particular, we define the exact and FE coercivity constants as

$$\alpha^e(\mu) = \inf_{w \in V^e} \frac{a(w, w; \mu)}{\|w\|_V^2}, \quad (18.7)$$

and

$$\alpha^{N_h}(\boldsymbol{\mu}) = \inf_{w \in V^{N_h}} \frac{a(w, w; \boldsymbol{\mu})}{\|w\|_V^2}, \quad (18.8)$$

respectively. It follows from the coercivity hypothesis that there exists a positive constant α_0 s.t. $\alpha^e(\boldsymbol{\mu}) \geq \alpha_0 > 0 \forall \boldsymbol{\mu} \in \mathcal{D}$, and that $\alpha^{N_h}(\boldsymbol{\mu}) \geq \alpha^e(\boldsymbol{\mu}) \forall \boldsymbol{\mu} \in \mathcal{D}$. Similarly, we define the exact and FE continuity constants as

$$\gamma^e(\boldsymbol{\mu}) = \sup_{w \in V^e} \sup_{v \in V^e} \frac{a(w, v; \boldsymbol{\mu})}{\|w\|_V \|v\|_V}, \quad (18.9)$$

and

$$\gamma^{N_h}(\boldsymbol{\mu}) = \sup_{w \in V^{N_h}} \sup_{v \in V^{N_h}} \frac{a(w, v; \boldsymbol{\mu})}{\|w\|_V \|v\|_V}, \quad (18.10)$$

respectively. It follows from the continuity hypothesis on a that $\gamma^e(\boldsymbol{\mu})$ is finite $\forall \boldsymbol{\mu} \in \mathcal{D}$, and that $\gamma^{N_h}(\boldsymbol{\mu}) \leq \gamma^e(\boldsymbol{\mu}) \forall \boldsymbol{\mu} \in \mathcal{D}$.

18.1.1 An illustrative example

We consider a simple example in which there are only physical parameters; a second one will be introduced later to illustrate how to treat geometrical parameters. We note that in all cases we provide the formulation for the “exact” problem (superscript e); the FE approximation is then derived from the exact statement following the procedures described earlier. Note also that all problems are presented in non-dimensional form.

Thermal block. We consider heat conduction in a square domain Ω . The square comprises $B_1 \times B_2$ blocks: each block is a different region with different thermal conductivity; the geometry is depicted in Fig. 18.1. Here

$$\overline{\Omega} = \bigcup_{i=1}^{P+1} \overline{\mathcal{R}_i},$$

where the \mathcal{R}_i , $i = 1, \dots, P+1$, correspond to the regions associated with the respective blocks/conductivities. Inhomogeneous Neumann (non-zero flux) boundary conditions are imposed on Γ_{base} ; homogeneous Dirichlet (temperature) conditions are imposed on Γ_{top} ; and homogeneous (zero flux) Neuman conditions are imposed on the two vertical sides. The output of interest is the average temperature over Γ_{base} [Arp66].

The parameters are then the conductivities in the first $B_1 B_2 - 1$ blocks (with the blocks numbered as shown in Figure 18.1); the conductivity of the last block, which serves for normalization, is unity. Hence $P = B_1 B_2 - 1$ and $\boldsymbol{\mu} = (\mu_1, \dots, \mu_P)$, where μ_p is the conductivity of block p ; the parameter domain is then given by $\mathcal{D} = [\mu^{\min}, \mu^{\max}]^P$ with $\mu^{\min} = 1/\sqrt{\mu_r}$, $\mu^{\max} = \sqrt{\mu_r}$ for $\mu_r = 100$ (hence $\mu^{\max}/\mu^{\min} = 100$).

We immediately recover the abstract statement of (18.1): we identify $V^e = \{v \in H^1(\Omega) \mid v|_{\Gamma_{\text{top}}} = 0\}$, which imposes the Dirichlet conditions;

$$a(w, v; \boldsymbol{\mu}) = \sum_{i=1}^P \mu_i \int_{\mathcal{R}_i} \nabla w \cdot \nabla v d\Omega + \int_{\mathcal{R}_{P+1}} \nabla w \cdot \nabla v d\Omega, \quad (18.11)$$

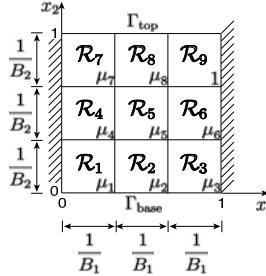


Fig. 18.1. Thermal block problem for $B_1 = B_2 = 3$

which is associated to the Laplace operator with homogeneous Neumann conditions (as well as internal flux continuity conditions), and

$$F(v) = F^{\text{Neu}}(v) = \int_{\Gamma_{\text{base}}} v d\gamma, \quad (18.12)$$

which imposes the inhomogeneous Neumann conditions.

The problem is coercive, symmetric, and compliant (the functional (18.12) indeed yields the mean temperature). Moreover, F is indeed independent of μ . The dependence on the parameters is affine, see (18.11): in this case no geometric transformations are required, though the splitting in subregions still serve to accommodate the discontinuous PDE coefficients. We directly observe that $Q = P + 1$ with $\Theta^q(\mu) = \mu_q$, $1 \leq q \leq P$, $\Theta^{P+1} = 1$, and

$$a^q(w, v) = \int_{\mathcal{R}_q} \nabla w \cdot \nabla v d\Omega, \quad 1 \leq q \leq P + 1.$$

This problem serves to illustrate the convergence rate of the RB discretization, both for $P = 1$, and for $P = 8$.

18.2 Geometric parametrization

We introduce the family of geometric parametric variations consistent with our affine restriction (18.2). Then, in Sec. 18.2.4 we describe the general class of scalar problems that undergo the abstract formulation of Sec. 18.1. For simplicity, we consider only the scalar case; the vector case permits an analogous treatment.

18.2.1 Affine geometry precondition

The RB method requires that Ω be a *parameter-independent* domain: if we wish to consider linear combinations of pre-computed solutions (also called “snapshots”), the latter must be defined relative to a common spatial configuration. Thus to permit geometric variation we must interpret Ω , our parameter-independent reference domain,

as the pre-image of $\Omega_o(\boldsymbol{\mu})$, the parameter-dependent “actual” or “original” domain of interest. The geometric transformation will yield variable (parameter dependent) coefficients of linear and bilinear forms in the reference domain that, under suitable hypotheses to be discussed below, will take the requisite affine form (18.2).

We shall assume that, for all $\boldsymbol{\mu}$ in \mathcal{D} , $\Omega_o(\boldsymbol{\mu})$ is expressed as a domain decomposition such that

$$\overline{\Omega}_o(\boldsymbol{\mu}) = \cup_{k=1}^{K_{\text{dom}}} \overline{\Omega}_o^k(\boldsymbol{\mu}), \quad (18.13)$$

where the $\Omega_o^k(\boldsymbol{\mu})$, $1 \leq k \leq K_{\text{dom}}$, are mutually non-overlapping open subdomains¹,

$$\Omega_o^k(\boldsymbol{\mu}) \cap \Omega_o^{k'}(\boldsymbol{\mu}) = \emptyset, \quad 1 \leq k < k' \leq K_{\text{dom}}. \quad (18.14)$$

See for an explanatory example Fig. 18.3 and 18.4. We now choose a value $\boldsymbol{\mu}_{\text{ref}} \in \mathcal{D}$ and define our reference domain as $\Omega = \Omega_o(\boldsymbol{\mu}_{\text{ref}})$. It immediately follows from (18.13) and (18.14), that

$$\overline{\Omega} = \cup_{k=1}^{K_{\text{dom}}} \overline{\Omega}^k, \quad (18.15)$$

$$\Omega^k \cap \Omega^{k'} = \emptyset, \quad 1 \leq k < k' \leq K_{\text{dom}}, \quad (18.16)$$

for $\Omega^k = \Omega_o^k(\boldsymbol{\mu}_{\text{ref}})$, $1 \leq k \leq K_{\text{dom}}$.

We will build our FE approximation on a very fine FE subtriangulation of the coarse domain decomposition partition. The latter can be called, for simplicity, the RB triangulation of Ω . (Recall that both the FE and RB approximations are defined over the reference domain.) Note that we purposely define K_{dom} with respect to the *exact* problem, rather than the FE approximation: K_{dom} can not depend on the FE subgrid to be meaningful. This FE subtriangulation ensures that the FE approximation accurately treats the perhaps discontinuous coefficients (arising from property and geometry variation) associated with the different regions; the subtriangulation also plays an important role in the generation of the affine representation (18.2).

We emphasize that the choice of $\boldsymbol{\mu}_{\text{ref}}$ only affects the accuracy of the underlying FE approximation upon which the RB discretization and *a posteriori* error estimator is built: typically a value of $\boldsymbol{\mu}_{\text{ref}}$ at the “center” of \mathcal{D} minimizes distortion and reduces the size N_h of the finite element problem necessary to yield a given acceptable FE error over \mathcal{D} .

We can treat any original domain $\Omega_o(\boldsymbol{\mu})$ that admits a domain partition (18.13)–(18.14) for which $\forall \boldsymbol{\mu} \in \mathcal{D}$,

$$\overline{\Omega}_o^k(\boldsymbol{\mu}) = \mathcal{T}^k(\overline{\Omega}^k; \boldsymbol{\mu}), \quad 1 \leq k \leq K_{\text{dom}}, \quad (18.17)$$

for *affine* mappings $\mathcal{T}^k(\cdot; \boldsymbol{\mu})$: $\Omega^k \rightarrow \Omega_o^k(\boldsymbol{\mu})$, $1 \leq k \leq K_{\text{dom}}$, that are:

(i) individually *bijective* (they induce the same subdivision from either side of an interface), and

¹ Typically the different subdomains correspond to different materials and hence material properties, or more generally different (discontinuously varying in space) PDE coefficients; however the subdomains may also be introduced for algorithmic purposes to ensure well-behaved mappings, as discussed in the next section.

(ii) collectively *continuous*,

$$\mathcal{T}^k(\boldsymbol{x}; \boldsymbol{\mu}) = \mathcal{T}^{k'}(\boldsymbol{x}; \boldsymbol{\mu}) \quad \forall \boldsymbol{x} \in \overline{\Omega}^k \cap \overline{\Omega}^{k'}, 1 \leq k < k' \leq K_{\text{dom}}. \quad (18.18)$$

The affine geometry precondition is a necessary condition for affine parameter dependence as defined in (18.2).

We now define the bijective affine mappings more explicitly: for $1 \leq k \leq K_{\text{dom}}$, any $\boldsymbol{\mu}$ in \mathcal{D} , and any $\boldsymbol{x} \in \Omega^k$,

$$\mathcal{T}_i^k(\boldsymbol{x}; \boldsymbol{\mu}) = C_i^k(\boldsymbol{\mu}) + \sum_{j=1}^d G_{i,j}^k(\boldsymbol{\mu}) x_j, \quad 1 \leq i \leq d, \quad (18.19)$$

for given $C^k: \mathcal{D} \rightarrow \mathbb{R}^d$ and $G^k: \mathcal{D} \rightarrow \mathbb{R}^{d \times d}$. We can then define the associated Jacobians

$$J^k(\boldsymbol{\mu}) = |\det(G^k(\boldsymbol{\mu}))|, \quad 1 \leq k \leq K_{\text{dom}}, \quad (18.20)$$

where \det denotes determinant; note the Jacobian is constant in space over each subdomain. We further define, for any $\boldsymbol{\mu} \in \mathcal{D}$,

$$D^k(\boldsymbol{\mu}) = (G^k(\boldsymbol{\mu}))^{-1}, \quad 1 \leq k \leq K_{\text{dom}}; \quad (18.21)$$

this matrix shall prove convenient in subsequent transformations involving derivatives.

We may interpret our local mappings in terms of a global transformation. In particular, for any $\boldsymbol{\mu} \in \mathcal{D}$, the local mappings (18.17) induce a global bijective *piecewise-affine* transformation $\mathcal{T}(\cdot; \boldsymbol{\mu}): \Omega \rightarrow \Omega_o(\boldsymbol{\mu})$: for any $\boldsymbol{\mu} \in \mathcal{D}$,

$$\mathcal{T}(\boldsymbol{x}; \boldsymbol{\mu}) = \mathcal{T}^k(\boldsymbol{x}; \boldsymbol{\mu}), \quad k = \min_{\{k' \in \{1, \dots, K_{\text{dom}}\} \mid \boldsymbol{x} \in \overline{\Omega}^{k'}\}} k'; \quad (18.22)$$

note the one-to-one property of this mapping (and, hence the arbitrariness of our “min” choice in (18.22)) is ensured by the interface condition (18.18). We can further demonstrate that these global *continuous* mappings are compatible with our second-order PDE variational formulation: for any $\boldsymbol{\mu} \in \mathcal{D}$, given any $w_o \in H^1(\Omega_o(\boldsymbol{\mu}))$, $w = w_o \circ \mathcal{T} \in H^1(\Omega)$; this ensures that our mapped problem on the reference domain is of the classical conforming type.

Although this concludes the formal exposition of admissible geometry variations, the application of these conditions requires familiarity with the scope of the affine mappings (18.19). We first consider a single subdomain in Sec. 18.2.2, then the case of multiple subdomains in Sec. 18.2.3: we give a prescriptive definition of a family of admissible domains, and we briefly summarize an algorithm to identify the associated domain decomposition and affine mappings. Finally, in Sec. 18.2.4 we discuss the incorporation of these affine mappings into our weak form.

18.2.2 Affine mappings: single subdomain

As we consider a single subdomain in this section, we shall suppress the subdomain superscript for clarity of exposition.

We first rewrite our affine transformation (18.19), for simplicity, without the sub-domain superscript: for any given $\boldsymbol{\mu} \in \mathcal{D}$, the reference domain Ω induces the parameter-dependent geometry of interest, $\Omega_o(\boldsymbol{\mu})$, through the affine mapping

$$\mathcal{T}_i(\mathbf{x}; \boldsymbol{\mu}) = C_i(\boldsymbol{\mu}) + \sum_{j=1}^d G_{ij}(\boldsymbol{\mu}) x_j, \quad 1 \leq i \leq d; \quad (18.23)$$

we shall refer to $C(\boldsymbol{\mu}) \in \mathbb{R}^d$ and $G(\boldsymbol{\mu}) \in \mathbb{R}^{d \times d}$ as the “mapping coefficients.” Under our assumption that the mapping is invertible we know that our Jacobian, $J(\boldsymbol{\mu})$ of (18.20), is strictly positive, and that the derivative transformation matrix, $D(\boldsymbol{\mu}) = (G(\boldsymbol{\mu}))^{-1}$ of (18.21), is well defined.

We recall that, in two dimensions, an affine transformation maps *straight lines* to *straight lines* and in fact parallel lines to parallel lines and indeed parallel lines of equal length to parallel lines of equal length: it follows that a triangle maps onto a triangle, that a parallelogram maps onto a parallelogram. We also recall that an affine transformation maps *ellipses* to *ellipses*. These “straight line” and “ellipse” properties are crucial for the descriptions of domains relevant in the engineering context.

Basic transformations. The affine transformation (18.23) is completely defined, for $d = 2$, by the $d(d + 1) = 6$ mapping coefficients $C(\boldsymbol{\mu}) \in \mathbb{R}^2$ and $G(\boldsymbol{\mu}) \in \mathbb{R}^{2 \times 2}$.

It immediately follows that, for any $\boldsymbol{\mu} \in \mathcal{D}$, we can uniquely identify $C(\boldsymbol{\mu})$ and $G(\boldsymbol{\mu})$ from the relationship between 3 non-colinear points — or nodes — in Ω , $(\bar{z}^1, \bar{z}^2, \bar{z}^3) = ((\bar{z}_1^1, \bar{z}_2^1), (\bar{z}_1^2, \bar{z}_2^2), (\bar{z}_1^3, \bar{z}_2^3))$, and 3 parametrized image nodes in $\Omega_o(\boldsymbol{\mu})$, $(\bar{z}_o^1(\boldsymbol{\mu}), \bar{z}_o^2(\boldsymbol{\mu}), \bar{z}_o^3(\boldsymbol{\mu})) = ((\bar{z}_{o1}^1, \bar{z}_{o2}^1), (\bar{z}_{o1}^2, \bar{z}_{o2}^2), (\bar{z}_{o1}^3, \bar{z}_{o2}^3))(\boldsymbol{\mu})$. Note that, being the affine transformation bijective, the image nodes are therefore also non-colinear.

In particular, for given $\boldsymbol{\mu} \in \mathcal{D}$, application of (18.23) to the selected nodes yields

$$\bar{z}_{oi}^m(\boldsymbol{\mu}) = C_i(\boldsymbol{\mu}) + \sum_{j=1}^2 G_{ij}(\boldsymbol{\mu}) \bar{z}_j^m, \quad 1 \leq i \leq 2, \quad 1 \leq m \leq 3; \quad (18.24)$$

(18.24) constitutes 6 independent equations by which to determine the 6 mapping coefficients. (In three-space dimensions, we must follow the “trajectories” of the 3 coordinates of 4 pre-image/image points: this yields 12 equations for the 12 mapping coefficients.)

To be more explicit in our construction, we first form the matrix $\mathbb{B} \in \mathbb{R}^{6 \times 6}$,

$$\mathbb{B} = \begin{bmatrix} 1 & 0 & \bar{z}_1^1 & \bar{z}_2^1 & 0 & 0 \\ 0 & 1 & 0 & 0 & \bar{z}_1^1 & \bar{z}_2^1 \\ 1 & 0 & \bar{z}_1^2 & \bar{z}_2^2 & 0 & 0 \\ 0 & 1 & 0 & 0 & \bar{z}_1^2 & \bar{z}_2^2 \\ 1 & 0 & \bar{z}_1^3 & \bar{z}_2^3 & 0 & 0 \\ 0 & 1 & 0 & 0 & \bar{z}_1^3 & \bar{z}_2^3 \end{bmatrix}. \quad (18.25)$$

We further introduce the vector $\mathbf{z}(\boldsymbol{\mu})$ of image nodal locations,

$$\mathbf{z}(\boldsymbol{\mu}) = [\bar{z}_{o1}^1(\boldsymbol{\mu}), \bar{z}_{o2}^1(\boldsymbol{\mu}), \bar{z}_{o1}^2(\boldsymbol{\mu}), \bar{z}_{o2}^2(\boldsymbol{\mu}), \bar{z}_{o1}^3(\boldsymbol{\mu}), \bar{z}_{o2}^3(\boldsymbol{\mu})]^T. \quad (18.26)$$

The solution of the linear system (18.24) can then be succinctly expressed as

$$[C_1(\boldsymbol{\mu}), C_2(\boldsymbol{\mu}), G_{11}(\boldsymbol{\mu}), G_{12}(\boldsymbol{\mu}), G_{21}(\boldsymbol{\mu}), G_{22}(\boldsymbol{\mu})]^T = (\mathbb{B})^{-1} \mathbf{z}(\boldsymbol{\mu}); \quad (18.27)$$

note that \mathbb{B} is non-singular under our hypothesis of non-collinear pre-image nodes.

The matrix \mathbb{B} is independent of $\boldsymbol{\mu}$; the parametric dependence derives from $\mathbf{z}(\boldsymbol{\mu})$. In particular, the $\boldsymbol{\mu}$ dependence of the geometry enters through the parametrized locations of the image nodes $(\bar{z}_o^1(\boldsymbol{\mu}), \bar{z}_o^2(\boldsymbol{\mu}), \bar{z}_o^3(\boldsymbol{\mu}))$ as represented in $\mathbf{z}(\boldsymbol{\mu})$. To illustrate how the parametric dependence propagates from the (desired) parametrized domain to the mapping coefficients we will provide an example. (In Sec. 18.2.4 we consider how the parametric dependence further propagates from the mapping coefficients to the affine expansion of the bilinear form, (18.2), associated with our PDE.)

We note that parallelograms and, in three space dimensions, parallelepipeds are the most intuitive subdomains by which to effect transformations — invoking the usual translation, dilation, rotation, and shear primitives. However, it is standard triangles, elliptical triangles, and more general “curvy” triangles [RHP08] which admit symbolic and numerical automation, and which are thus the building blocks of choice in general multi-subdomain software (e.g., [HNRP08]).

Triangles are indeed the “workhorses” in our geometric decompositions. To demonstrate the application of the technology to triangular domains, we consider a single parameter, $\boldsymbol{\mu} = \boldsymbol{\mu}_1 \in \mathcal{D} = [0.5, 2]$. We take for Ω the triangle with vertices (counter-clockwise) $(0, 0), (1, 0), (1, 1)$; these vertices shall also serve as the pre-image nodes, and hence $\bar{z}^1 = (0, 0), \bar{z}^2 = (1, 0), \bar{z}^3 = (1, 1)$. We take for $\Omega_o(\boldsymbol{\mu})$ the triangle with vertices (counter-clockwise) $(0, 0), (\boldsymbol{\mu}_1, 0), (1, 1)$; these vertices shall also serve as the image nodes, and hence $\bar{z}_o^1(\boldsymbol{\mu}_1) = (0, 0), \bar{z}_o^2(\boldsymbol{\mu}_1) = (\boldsymbol{\mu}_1, 0), \bar{z}_o^3(\boldsymbol{\mu}_1) = (1, 1)$. (Note for triangles, our three points uniquely define not only the transformation but also the reference and parametrized domains.) As already noted in Section 18.2.1, the pre-image nodes correspond to the image nodes for a particular value of the parameter: in our example here, $\boldsymbol{\mu}_{1\text{ref}} = 1$ such that $(\bar{z}^1, \bar{z}^2, \bar{z}^3) = (\bar{z}_o^1(\boldsymbol{\mu}_{1\text{ref}}), \bar{z}_o^2(\boldsymbol{\mu}_{1\text{ref}}), \bar{z}_o^3(\boldsymbol{\mu}_{1\text{ref}}))$. The domains Ω and $\Omega_o(\boldsymbol{\mu})$ are depicted in Figure 18.2(a) and Figure 18.2(b), respectively.

By a direct calculation, we find from (18.27) that

$$\begin{bmatrix} C_1(\boldsymbol{\mu}_1) \\ C_2(\boldsymbol{\mu}_1) \\ G_{11}(\boldsymbol{\mu}_1) \\ G_{12}(\boldsymbol{\mu}_1) \\ G_{21}(\boldsymbol{\mu}_1) \\ G_{22}(\boldsymbol{\mu}_1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{\mu}_1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad (18.28)$$

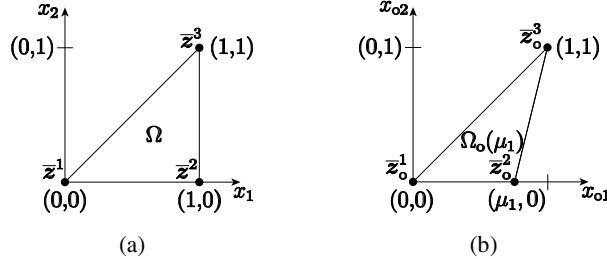


Fig. 18.2. (a) Reference domain Ω , and (b) actual (or original) domain $\Omega_o(\mu_1)$

and hence $C(\mu_1) = 0$ since there is no translation)

$$G(\mu_1) = \begin{bmatrix} \mu_1 & 1 - \mu_1 \\ 0 & 1 \end{bmatrix}. \quad (18.29)$$

It directly follows from (18.20) and (18.21) that

$$J(\mu_1) = \mu_1, \quad (18.30)$$

and

$$D(\mu_1) = \begin{bmatrix} \frac{1}{\mu_1} & -\frac{1-\mu_1}{\mu_1} \\ 0 & 1 \end{bmatrix}. \quad (18.31)$$

Note that D is *not* linear in μ_1 . (We also note that a translation would affect $C(\mu_1)$ but not $G(\mu_1)$, $J(\mu_1)$, or $D(\mu_1)$.)

We can readily construct an affine map from *any* reference triangle Ω in \mathbb{R}^2 , onto *any* desired triangle $\Omega_o(\mu)$ in \mathbb{R}^2 : it will prove most convenient to choose for our nodes $(\bar{z}^1, \bar{z}^2, \bar{z}^3)$ and $(\bar{z}_o^1(\mu), \bar{z}_o^2(\mu), \bar{z}_o^3(\mu))$.

18.2.3 Piecewise affine mappings: multiple subdomains

A single affine mapping can treat only a very limited family of parametrized domains $\Omega_o(\mu)$. However, piecewise affine mappings — in our case based on a domain decomposition in standard, elliptical, and curvy triangles [RHP08] — can address a much larger class of geometric variations. We restrict attention here to two space dimensions.

We shall consider domains for which the boundary and internal interfaces can be represented by straight edges. For the elliptical arcs or an analogous family of “curvy-edge” domains built from curvy triangles see [RHP08].

There are three steps to the multi-domain mapping process. First (our emphasis in this section), we shall generate an “RB triangulation” of the *reference* domain Ω , (18.15), that is compatible with the mapping continuity condition (18.18); second, as already developed in the previous section, we will construct the parameter-dependent

affine mappings (18.19) for each subdomain following the procedure (18.25)–(18.27); third, as described in the next section, we will translate these parametric mappings into PDE coefficients, and then “optimize” to arrive at an economical affine expansion (18.2). The algorithm shares some features with classical FE triangulation. For its implementation see [HNRP08].

Unfortunately, this procedure does *not* guarantee – even for very simple parametric domains $\Omega_o(\mu)$ with straight-edge boundaries decomposed in standard triangles – that the Jacobians of the associated affine mappings will remain strictly positive for all $\mu \in \mathcal{D}$, and in particular for μ far from μ_{ref} . Equivalently, our algorithm does *not* guarantee that a valid domain partition of Ω , (18.15)–(18.16), will induce a valid domain partition of $\Omega_o(\mu)$, (18.13)–(18.14), for all $\mu \in \mathcal{D}$.

Furthermore, even if the Jacobian does not vanish, small Jacobians corresponding to excessive distortion will lead to FE approximations that are at best inefficient and at worst very ill-conditioned.

We further caution that even well-behaved/non-singular mappings can be quite inefficient as regards ultimate performance of the reduced basis approximation. Inefficient RB triangulations are characterized by many parametrically “dissimilar” triangles (i.e. with different D and G) that, in turn, generate many distinct affine mappings (18.19): we obtain a large value for Q in (18.2) and ultimately (as we shall see) poor Offline and Online RB performance.² In contrast, efficient RB triangulations are characterized by relatively few parametrically dissimilar triangles that, in turn, generate relatively few distinct affine mappings (18.19) – in particular, relatively few distinct J and D : we obtain a smaller value for Q in (18.2) and ultimately better Offline and Online RB performance.

Fortunately, proper selection of the initial control point/edge data perhaps supplemented by the introduction of artificial regions – regions motivated by mapping considerations rather than physical/mathematical considerations – can ensure both well-behaved/non-singular and efficient transformations; in most cases, good, and bad, choices are rather self-evident. We must first understand the connection between the domain decomposition and associated mapping coefficients developed here and the final affine representation of the PDE bilinear form (18.2).

18.2.4 Bilinear forms

As already indicated, we shall consider here only the scalar case; the vector case, for instance that arising from linear elasticity, allows an analogous treatment.

Formulation on “original” domain. Our problem is initially posed on the “original” domain $\Omega_o(\mu)$, which we assume realizes the affine geometry precondition as described in the previous section. We shall assume for simplicity that $V_o^e(\mu) = H_0^1(\Omega_o(\mu))$, which corresponds to homogeneous Dirichlet boundary conditions over the entire boundary $\partial\Omega_o(\mu)$; we subsequently discuss natural (Neumann and Robin) conditions.

² As discussed in Sec. 18.3.1 and 18.5.3, the RB Offline computational complexity scales as Q , and the RB Online computational complexity scales as Q^2 .

Given $\boldsymbol{\mu} \in \mathcal{D}$, we evaluate

$$s_o^e(\boldsymbol{\mu}) = F_o(u_o^e(\boldsymbol{\mu})) ,$$

where $u_o^e(\boldsymbol{\mu}) \in V_o^e(\boldsymbol{\mu})$ satisfies

$$a_o(u_o^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = F_o(v) \quad \forall v \in V_o^e(\boldsymbol{\mu}) .$$

We now place conditions on a_o and F_o such that, in conjunction with the affine geometry precondition, we are ensured an affine expansion of the bilinear form.

In particular, we require $a_o(\cdot, \cdot; \boldsymbol{\mu}): H^1(\Omega_o(\boldsymbol{\mu})) \times H^1(\Omega_o(\boldsymbol{\mu})) \rightarrow \mathbb{R}$ to be expressed as

$$a_o(w, v; \boldsymbol{\mu}) = \sum_{\ell=1}^{K_{\text{dom}}} \int_{\Omega_o^\ell(\boldsymbol{\mu})} \begin{bmatrix} \frac{\partial w}{\partial x_{o1}} & \frac{\partial w}{\partial x_{o2}} & w \end{bmatrix} \mathcal{K}_{o,\ell} ij(\boldsymbol{\mu}) \begin{bmatrix} \frac{\partial v}{\partial x_{o1}} \\ \frac{\partial v}{\partial x_{o2}} \\ v \end{bmatrix} d\Omega_o , \quad (18.32)$$

where $\boldsymbol{x}_o = (x_{o1}, x_{o2})$ denotes a point in $\Omega_o(\boldsymbol{\mu})$. Here, for $1 \leq \ell \leq K_{\text{dom}}$, $\mathcal{K}_{o,\ell}: \mathcal{D} \rightarrow \mathbb{R}^{3 \times 3}$ is a given symmetric positive definite matrix, which in turn ensures coercivity of our bilinear form; the upper 2×2 principal submatrix of $\mathcal{K}_{o,\ell}$ is the usual tensor conductivity/diffusivity; the $(3, 3)$ element of $\mathcal{K}_{o,\ell}$ represents the identity operator, leading to the mass matrix. The $(3, 1), (3, 2)$ (and $(1, 3), (2, 3)$) elements of $\mathcal{K}_{o,\ell}$ permit first derivative (or convective) terms.

Similarly, we require that $F_o: H^1(\Omega_o(\boldsymbol{\mu})) \rightarrow \mathbb{R}$ can be expressed as

$$F_o(v) = \sum_{\ell=1}^{K_{\text{dom}}} \int_{\Omega_o^\ell(\boldsymbol{\mu})} \mathcal{F}_{o,\ell}(\boldsymbol{\mu}) v d\Omega_o ,$$

where, for $1 \leq \ell \leq K_{\text{dom}}$, $\mathcal{F}_{o,\ell}: \mathcal{D} \rightarrow \mathbb{R}$.

Formulation on reference domain. We now apply standard techniques to transform the problem statement over the original domain to an equivalent problem statement over the reference domain: given $\boldsymbol{\mu} \in \mathcal{D}$, we find

$$s^e(\boldsymbol{\mu}) = F(u^e(\boldsymbol{\mu})) ,$$

where $u^e(\boldsymbol{\mu}) \in V^e = H_0^1(\Omega)$ satisfies

$$a(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = F(v) \quad \forall v \in V^e .$$

We may then identify $s^e(\boldsymbol{\mu}) = s_o^e(\boldsymbol{\mu})$ and $u^e(\boldsymbol{\mu}) = u_o^e(\boldsymbol{\mu}) \circ \mathcal{T}(\cdot; \boldsymbol{\mu})$.

The transformed bilinear form a can be expressed as

$$a(w, v; \boldsymbol{\mu}) = \sum_{k=1}^{K_{\text{dom}}} \int_{\Omega_k} \begin{bmatrix} \frac{\partial w}{\partial x_1} & \frac{\partial w}{\partial x_2} & w \end{bmatrix} \mathcal{K}_{ij}^k(\boldsymbol{\mu}) \begin{bmatrix} \frac{\partial v}{\partial x_1} \\ \frac{\partial v}{\partial x_2} \\ v \end{bmatrix} d\Omega , \quad (18.33)$$

where $\boldsymbol{x} = (x_1, x_2)$ denotes a point in Ω . Here the $\mathcal{K}^k: \mathcal{D} \rightarrow \mathbb{R}^{3 \times 3}$, $1 \leq k \leq K_{\text{dom}}$, are symmetric positive definite matrices given by

$$\mathcal{K}^k(\boldsymbol{\mu}) = J^k(\boldsymbol{\mu}) \mathcal{G}^k(\boldsymbol{\mu}) \mathcal{K}_{o,\ell}(\boldsymbol{\mu}) (\mathcal{G}^k(\boldsymbol{\mu}))^T, \quad (18.34)$$

for $1 \leq k \leq K_{\text{dom}}$; the $\mathcal{G}^k: \mathcal{D} \rightarrow \mathbb{R}^{3 \times 3}$, $1 \leq k \leq K_{\text{dom}}$, are given by

$$\mathcal{G}^k(\boldsymbol{\mu}) = \begin{pmatrix} D^k(\boldsymbol{\mu}) & 0 \\ 0 & 0 & 1 \end{pmatrix}; \quad (18.35)$$

$J^k(\boldsymbol{\mu})$ and $D^k(\boldsymbol{\mu})$, $1 \leq k \leq K_{\text{dom}}$, are given by (18.20) and (18.21), respectively.

Similarly, the transformed linear form can be expressed as

$$F(v) = \sum_{k=1}^{K_{\text{dom}}} \int_{\Omega^k} \mathcal{F}^k(\boldsymbol{\mu}) v d\Omega.$$

Here $\mathcal{F}^k: \mathcal{D} \rightarrow \mathbb{R}$, $1 \leq k \leq K_{\text{dom}}$, is given by

$$\mathcal{F}^k = J^k(\boldsymbol{\mu}) \mathcal{F}_{o,\ell}(\boldsymbol{\mu}), \quad 1 \leq k \leq K_{\text{dom}}.$$

We note that, in general, the $\mathcal{K}^k(\boldsymbol{\mu})$ and $\mathcal{F}^k(\boldsymbol{\mu})$, $1 \leq k \leq K_{\text{dom}}$, will be different for each subdomain Ω^k . The differences can arise either due to coefficient variation, or to geometry variation, or both. We thus require, as already indicated earlier, that the FE approximation be built upon a subtriangulation of the RB triangulation: discontinuities in PDE coefficients are thereby restricted to element edges to ensure rapid convergence; and identification/extraction of the terms in the affine expansion (18.2) is more readily effected, as we now discuss.

Affine form. We focus here on a , though F admits a similar treatment. We simply expand the form (18.33) by considering in turn each subdomain Ω^k and each entry of the diffusivity tensor \mathcal{K}_{ij}^k , $1 \leq i, j \leq 3$, $1 \leq k \leq K_{\text{dom}}$. Thus,

$$\begin{aligned} a(w, v; \boldsymbol{\mu}) &= \mathcal{K}_{11}^1(\boldsymbol{\mu}) \int_{\Omega^1} \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1} d\Omega \\ &\quad + \mathcal{K}_{12}^1(\boldsymbol{\mu}) \int_{\Omega^1} \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_2} d\Omega + \cdots + \mathcal{K}_{33}^{K_{\text{dom}}}(\boldsymbol{\mu}) \int_{\Omega^{K_{\text{dom}}}} w v d\Omega. \end{aligned} \quad (18.36)$$

We can then identify each component in the affine expansion: for each term in (18.36), the pre-factor represents $\Theta^q(\boldsymbol{\mu})$, while the integral represents a^q .

Taking into account the symmetry of the bilinear form, such that only the (1, 1), (1, 2) (= (2, 1)), (2, 2), and (3, 3) entries of $\mathcal{K}_{o,\ell}(\boldsymbol{\mu})$ — and hence $\mathcal{K}^k(\boldsymbol{\mu})$ — must be accommodated, there are at most $Q = 4K$ terms in the affine expansion. The $\Theta^q(\boldsymbol{\mu})$ are given by, for the obvious numbering scheme $\Theta^1(\boldsymbol{\mu}) = \mathcal{K}_{11}^1(\boldsymbol{\mu})$, $\Theta^2(\boldsymbol{\mu}) = \mathcal{K}_{12}^1(\boldsymbol{\mu})$, \dots , $\Theta^5(\boldsymbol{\mu}) = \mathcal{K}_{11}^2(\boldsymbol{\mu})$, \dots , $\Theta^Q(\boldsymbol{\mu}) = \mathcal{K}_{33}^{K_{\text{dom}}}(\boldsymbol{\mu})$; the $a^q(w, v)$ are given by

$$\begin{aligned}
a^1(w, v) &= \int_{\Omega^1} \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1} d\Omega, \\
a^2(w, v) &= \int_{\Omega^1} \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_2} d\Omega, \\
&\vdots \\
a^5(w, v) &= \int_{\Omega^2} \frac{\partial w}{\partial x_1} \frac{\partial v}{\partial x_1} d\Omega, \\
&\vdots \\
a^Q(w, v) &= \int_{\Omega^{K_{\text{dom}}}} w v d\Omega.
\end{aligned}$$

This identification constitutes a constructive proof that the affine geometry precondition and the property/coefficients variation permitted by (18.32) do indeed yield a bilinear form which can be expressed in the requisite affine form (18.2).

We close with a discussion of generality. In fact, the conditions we provide are sufficient but not necessary. For example, we can permit affine polynomial dependence on \boldsymbol{x}_o in both $\mathcal{K}_{o,k}(\boldsymbol{x}_o; \boldsymbol{\mu})$ and $\mathcal{F}_{o,k}(\boldsymbol{x}_o; \boldsymbol{\mu})$ and still ensure an affine development, (18.2); furthermore, in the absence of geometric variation, $\mathcal{K}_{o,\ell}(\boldsymbol{x}_o; \boldsymbol{\mu})$ and $\mathcal{F}_{o,\ell}(\boldsymbol{x}_o; \boldsymbol{\mu})$ can take on *any* “separable” form in $\boldsymbol{x}, \boldsymbol{\mu}$. However, the affine expansion (18.2) is by no means completely general: for more complicated data parametric dependencies, non-affine techniques [BNMP04, GMNP07, Roz09] must be invoked.

18.2.5 A second illustrative example

We now consider an example to illustrate geometrical parametrization, its treatment and how the choice of control points can affect not only well-posedness but also efficiency of the affine representation (18.2). We shall consider the Laplace operator with isotropic diffusivity corresponding to $\mathcal{K}_{o,\ell,11} = \mathcal{K}_{o,\ell,22} = 1$ and all other entries of $\mathcal{K}_{o,\ell}$ zero for $1 \leq \ell \leq K_{\text{dom}}$. We consider the original domain $\Omega_o(\boldsymbol{\mu}) = (-2, 2) \times (-2, 2) \setminus (-\mu_1, \mu_1) \times (-\mu_2, \mu_2)$: a square with a variable rectangular hole. The two ($P = 2$) parameters correspond to the dimensions of the rectangular hole; the parameter domain is given by $\mathcal{D} = [0.5, 1.5] \times [0.5, 1.5]$. We choose $\boldsymbol{\mu}_{\text{ref}} = (1.0, 1.0)$.

In the first instance, we choose the User-provided control points/edges as shown in Figure 18.3(a), which yields the $K_{\text{dom}} = 8$ RB triangulation of Ω shown in Figure 18.3(b). There are $Q = 10$ different terms in our affine expansion (18.2) for the Laplacian. There is some symmetry in the RB triangulation, and this reduces the number of terms in the affine expansion for the Laplacian from the maximum possible of 24 to 10.

In the second instance, we choose the User-provided control points/edges as shown in Figure 18.4(a), which yields the $K_{\text{dom}} = 16$ RB triangulation shown in Figure 18.4(b). There are $Q = 6$ different terms in our affine expansion (18.2) for the Laplacian. The control points of Fig. 18.4(a) carry one drawback: the effect of the hole

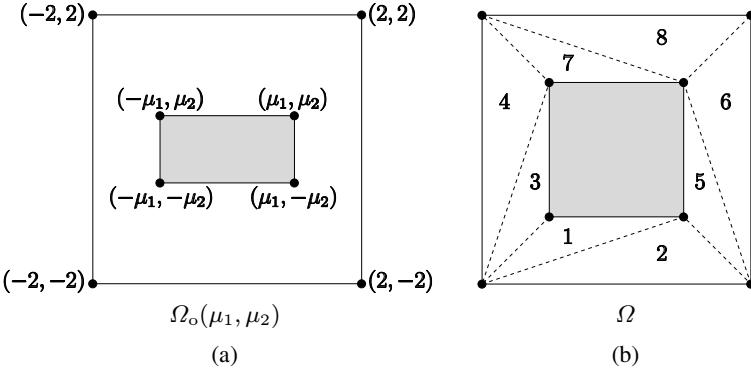


Fig. 18.3. (a) Original domain $\Omega_o(\mu)$ and initial control points/edges indicated, and (b) $K_{\text{dom}} = 8$ RB triangulations, $\overline{\Omega} = \cup_{k=1}^{K_{\text{dom}}} \overline{\Omega}^k$ (the numbers refer to subdomains)

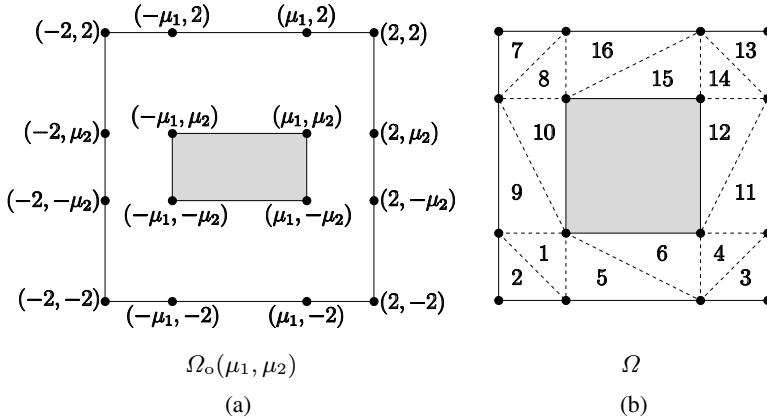
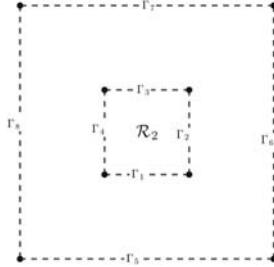


Fig. 18.4. (a) Original domain $\Omega_o(\mu)$ and initial control points/edges indicated, and (b) $K_{\text{dom}} = 16$ RB triangulations, $\overline{\Omega} = \cup_{k=1}^{K_{\text{dom}}} \overline{\Omega}^k$

“propagates” to the outer boundary, and hence would not be efficient if $\Omega_o(\mu)$ in fact represents a *region* within a larger more complex domain; in the latter case, the control points of Figure 18.3(a), which localize the geometric variation, are preferred.

We take this geometrical configuration and we build upon a second working problem, by considering the performance of a device designed for heat exchange: a square section with a parametrized rectangular cavity inside, whose internal walls $\Gamma_1 - \Gamma_4$ are subject to a constant heat flux to be dissipated (Neumann conditions). The outer walls $\Gamma_5 - \Gamma_8$ of the square are characterized by a heat exchange rate with a fluid surrounding the device (Robin conditions). We model the flowing air through a simple convection heat transfer coefficient, i.e. the *Biot* number, used as third parameter for the model problem $\mu_3 \in [.01, 1]$, so that $\mathcal{D} = [0.5, 1.5] \times [0.5, 1.5] \times [.01, 1]$. The steady-state temperature distribution is governed by a Laplace equation. Our interest

**Fig. 18.5.** Domain boundaries

is in the conduction temperature distribution at the inner walls. From the engineering point of view, this problem illustrates the application of conduction analysis to an important class of cooling problems, for example for electronic components and systems, see [RP08]. Figure 18.5 represents the schematic domain and its boundaries. We recall (18.1), in this case — and in fact, in all cases involving geometric variation — it shall prove more convenient to define the constituents with respect to the “original” parameter-dependent domain. In particular, we identify $V_o^e(\boldsymbol{\mu}) = H^1(\Omega_o(\boldsymbol{\mu}))$ and

$$a_o(w, v; \boldsymbol{\mu}) = \int_{\Omega_o(\boldsymbol{\mu})} \nabla w \cdot \nabla v d\Omega + \mu_3 \left(\int_{\Gamma_{o,5}} w d\Omega + \int_{\Gamma_{o,6}} w d\Omega + \int_{\Gamma_{o,7}} w d\Omega + \int_{\Gamma_{o,8}} w d\Omega \right),$$

which represents the bilinear form associated to the Laplacian, and

$$F_o(v) = \int_{\Gamma_{o,1}} v d\gamma + \int_{\Gamma_{o,2}} v d\gamma + \int_{\Gamma_{o,3}} v d\gamma + \int_{\Gamma_{o,4}} v d\gamma,$$

which imposes the constant heat flux on the interior walls. The problem is clearly coercive, symmetric, and compliant as $s_o(\boldsymbol{\mu}) = F_o(u(\boldsymbol{\mu}))$.

We then construct the associated affine mappings according to the approach described in Sec. 18.2.2; we next effect the re-formulation on the reference domain, as described in Sec. 18.2.4; finally, we extract the terms in the affine expansion (18.2), following the process defined in the previous section.

18.3 The reduced basis method

We are eventually ready to illustrate the essence of RB methodology.

18.3.1 Reduced basis approximation and spaces

We assume that we are given a FE approximation space V^{N_h} of dimension N_h . In order to define a particular reduced basis space, we start by considering a fixed N_h . We then introduce, given a positive integer N_{\max} , an associated sequence of what shall

ultimately be reduced basis approximation spaces: for $N = 1, \dots, N_{\max}$, $X_N^{N_h}$ is a N -dimensional subspace of V^{N_h} ; we further suppose that

$$V_1^{N_h} \subset V_2^{N_h} \subset \cdots V_{N_{\max}}^{N_h} \subset V^{N_h}. \quad (18.37)$$

As we shall see, the nested or *hierarchical* condition (18.37) is important in ensuring memory efficiency of the resulting RB approximation. As we will mention in Sec. 18.6, there are several classical RB proposals — Taylor, Lagrange [Por85], and Hermite [IR98b] spaces — as well as several more recent alternatives — such as Proper Orthogonal Decomposition (POD) spaces [Caz97, Gun03a, KV02, Loe55]. All of these spaces “focus” in one fashion or another on a low-dimensional, smooth parametric manifold, $\mathcal{M}^{N_h} = \{u(\boldsymbol{\mu}) \mid \boldsymbol{\mu} \in \mathcal{D}\}$: the set of fields engendered as the input varies over the parameter domain. In the case of single parameter, the parametrically induced manifold is a one-dimensional filament within the infinite dimensional space which characterizes *general* solutions to the given PDE. Clearly, generic approximation spaces are unnecessarily rich and hence unnecessarily expensive within the parametric framework. Much of what we present — in particular, all the discussion of this section related to optimality, discrete equations, conditioning, and Offline-Online procedures, and all that of Sec. 18.5 related to *a posteriori* error estimation — shall be relevant to any of these reduced basis spaces/approximations.

However, some of what we shall present, in particular related to sampling strategies in Sec. 18.3.2, is restricted to the particular reduced basis space which shall be our primary focus: the Lagrange reduced basis spaces [Por85], which we denote by $W_N^{N_h}$. In order to define a (hierarchical) sequence of Lagrange spaces $W_N^{N_h}, 1 \leq N \leq N_{\max}$, we first introduce a master set of parameter points $\boldsymbol{\mu}^n \in \mathcal{D}, 1 \leq n \leq N_{\max}$. We then define, for given $N \in \{1, \dots, N_{\max}\}$, the Lagrange parameter samples

$$S_N = \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\}, \quad (18.38)$$

and associated Lagrange RB spaces

$$W_N^{N_h} = \text{span}\{u^{N_h}(\boldsymbol{\mu}^n), 1 \leq n \leq N\}. \quad (18.39)$$

We observe that, by construction, these Lagrange spaces $W_N^{N_h}$ satisfy (18.37): the samples (18.38) are nested, that is $S_1 = \{\boldsymbol{\mu}^1\} \subset S_2 = \{\boldsymbol{\mu}^1, \boldsymbol{\mu}^2\} \subset \cdots \subset S_{N_{\max}}$; the Lagrange RB spaces (18.39) are hierarchical, that is $W_1^{N_h} = \text{span}\{u^{N_h}(\boldsymbol{\mu}^1)\} \subset W_2^{N_h} = \text{span}\{u^{N_h}(\boldsymbol{\mu}^1), u^{N_h}(\boldsymbol{\mu}^2)\} \subset \cdots \subset W_{N_{\max}}^{N_h}$.

The $u^{N_h}(\boldsymbol{\mu}^n), 1 \leq n \leq N_{\max}$, are often referred to as *snapshots* of the parametric manifold \mathcal{M}^{N_h} . For reasons that will become clear subsequently, these snapshots are more precisely referred to as *retained* snapshots. We depict the retained snapshots graphically in Fig. 18.6. It is clear that, if indeed the manifold is low-dimensional and smooth (a point we return to later), then we would expect to well approximate any member of the manifold — any solution $u^{N_h}(\boldsymbol{\mu})$ for some $\boldsymbol{\mu}$ in \mathcal{D} — in terms of relatively few retained snapshots. However, we must first ensure that we can choose a good combination of the available retained snapshots (Sec. 18.3.1), that we can represent the retained snapshots in a stable RB basis and efficiently obtain the associated RB basis coefficients (Sec. 18.3.1), and finally that we can choose our retained snapshots — in essence, the parameter sample $S_{N_{\max}}$ — optimally (Sec. 18.3.2).

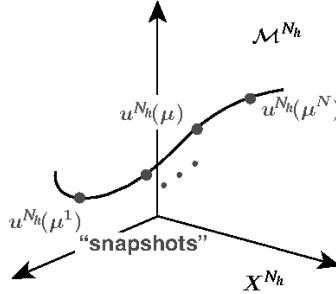


Fig. 18.6. The “snapshots” $u^{N_h}(\mu^n)$, $1 \leq n \leq N$, on the parametric manifold \mathcal{M}^{N_h}

Galerkin projection. For our particular class of equations, Galerkin projection is arguably the best approach. Given $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$s_N^{N_h}(\boldsymbol{\mu}) = F(u_N^{N_h}(\boldsymbol{\mu})) ,$$

where $u_N^{N_h}(\boldsymbol{\mu}) \in V_N^{N_h} \subset V^{N_h}$, indicating here a general RB space (not necessarily a Lagrange space) with $V_N^{N_h}$, satisfies

$$a(u_N^{N_h}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = F(v) \quad \forall v \in V_N^{N_h} . \quad (18.40)$$

We emphasize that our ultimate interest is the output prediction: the field variable serves as an intermediary. We immediately obtain (from Céa’s lemma) the classical optimality result in the energy norm (18.5)

$$\|u^{N_h}(\boldsymbol{\mu}) - u_N^{N_h}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} \leq \inf_{w \in V_N^{N_h}} \|u^{N_h}(\boldsymbol{\mu}) - w\|_{\boldsymbol{\mu}} ; \quad (18.41)$$

in this norm, the Galerkin procedure automatically selects the *best* combination of snapshots. It is also readily derived that

$$s^{N_h}(\boldsymbol{\mu}) - s_N^{N_h}(\boldsymbol{\mu}) = \|u^{N_h}(\boldsymbol{\mu}) - u_N^{N_h}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2 ; \quad (18.42)$$

the output converges as the square of the energy error. Although this latter result depends critically on the compliance assumption, extension *via* adjoint approximations to the non-compliant case is possible; see [RHP08].

We now consider the discrete equations associated with the Galerkin approximation (18.40). We must first choose an appropriate basis for our space: incorrect choice of the RB basis can lead to very poorly conditioned systems; this is immediately apparent in the Lagrange case — if $W_N^{N_h}$ provides rapid convergence then, by construction, the snapshots of (18.39) will be increasingly co-linear as N increases. Towards this end, we apply the Gram-Schmidt process [Mey00, TI97] in the $(\cdot, \cdot)_V$ inner product to our snapshots $u^{N_h}(\boldsymbol{\mu}^n)$, $1 \leq n \leq N_{\max}$, to obtain mutually orthonormal functions $\zeta_n^{N_h}$, $1 \leq n \leq N_{\max}$: $(\zeta_n^{N_h}, \zeta_m^{N_h})_V = \delta_{nm}$, $1 \leq n, m \leq N_{\max}$, where δ_{nm} is the Kronecker-delta symbol. We then choose the sets $\{\zeta_n^{N_h}\}_{n=1,\dots,N}$ as our bases for $W_N^{N_h}$, $1 \leq N \leq N_{\max}$.

We now insert

$$u_N^{N_h}(\boldsymbol{\mu}) = \sum_{m=1}^N u_{N,m}(\boldsymbol{\mu}) \zeta_m^{N_h}, \quad (18.43)$$

and $v = \zeta_n^{N_h}$, $1 \leq n \leq N$, into (18.40) to obtain the RB algebraic system

$$\sum_{m=1}^N a(\zeta_m^{N_h}, \zeta_n^{N_h}; \boldsymbol{\mu}) u_{N,m}(\boldsymbol{\mu}) = F(\zeta_n^{N_h}), \quad 1 \leq n \leq N, \quad (18.44)$$

for the RB coefficients $u_{N,m}(\boldsymbol{\mu})$, $1 \leq m \leq N$; we can subsequently evaluate the RB output prediction as

$$s_N^{N_h}(\boldsymbol{\mu}) = \sum_{m=1}^N u_{N,m}(\boldsymbol{\mu}) F(\zeta_m^{N_h}). \quad (18.45)$$

By using the Rayleigh quotient as done in (4.51), it can be readily proven [PR07] that the condition number of the matrix $a(\zeta_m^{N_h}, \zeta_n^{N_h}; \boldsymbol{\mu})$, $1 \leq n, m \leq N$, is bounded by $\gamma^e(\boldsymbol{\mu})/\alpha^e(\boldsymbol{\mu})$, independently of N and N_h , owing to the orthogonality of the $\{\zeta_n^{N_h}\}$ and to (18.7) and (18.9).

Offline-Online computational procedure. The system (18.44) is nominally of small size: a set of N linear algebraic equations in N unknowns. However, the formation of the associated stiffness matrix, and indeed the right-hand-side vector, involves entities $\zeta_n^{N_h}$, $1 \leq n \leq N$, associated with our N_h -dimensional FE approximation space. If we must invoke FE fields in order to form the RB stiffness matrix *for each new value of $\boldsymbol{\mu}$* the marginal cost per input-output evaluation $\boldsymbol{\mu} \rightarrow s_N(\boldsymbol{\mu})$ will remain unacceptably large.

Fortunately, we can appeal to affine parameter dependence to construct very efficient Offline-Online procedures, as we now discuss. In particular, we note that system (18.44) can be expressed, thanks to (18.2), as

$$\sum_{m=1}^N \left(\sum_{q=1}^Q \Theta^q(\boldsymbol{\mu}) a^q(\zeta_m^{N_h}, \zeta_n^{N_h}) \right) u_{N,m}(\boldsymbol{\mu}) = F(\zeta_n^{N_h}), \quad 1 \leq n \leq N. \quad (18.46)$$

We observe that the ζ^{N_h} are now isolated in terms that are independent of $\boldsymbol{\mu}$ and hence that can be *pre-computed* in an Offline-Online procedure.

In the Offline stage, we first compute the $u^{N_h}(\boldsymbol{\mu}^n)$, $1 \leq n \leq N_{\max}$, and subsequently the $\zeta_n^{N_h}$, $1 \leq n \leq N_{\max}$; we then form and store the terms

$$F(\zeta_n^{N_h}), \quad 1 \leq n \leq N_{\max}, \quad (18.47)$$

and

$$a^q(\zeta_m^{N_h}, \zeta_n^{N_h}), \quad 1 \leq n, m \leq N_{\max}, \quad 1 \leq q \leq Q. \quad (18.48)$$

The Offline operation count depends on N_{\max} , Q , and N_h .

In the Online stage, we retrieve (18.48) to form

$$\sum_{q=1}^Q \Theta^q(\boldsymbol{\mu}) a^q(\zeta_m^{N_h}, \zeta_n^{N_h}), \quad 1 \leq n, m \leq N; \quad (18.49)$$

we solve the resulting $N \times N$ stiffness system (18.46) to obtain the $u_{N,m}(\boldsymbol{\mu})$, $1 \leq m \leq N$; and finally we access (18.47) to evaluate the output (18.45). The Online operation count is $O(QN^2)$ to perform the sum (18.49), $O(N^3)$ to invert (18.46) — note that the RB stiffness matrix is full, and finally $O(N)$ to effect the inner product (18.45). The Online storage (the data archived in the Offline stage) is thanks to the hierarchical condition (18.37) only $O(QN_{\max}^2) + O(N_{\max})$: for any given N , we may extract the necessary RB $N \times N$ matrices (respectively, N -vectors) as principal submatrices (respectively, principal subvectors) of the corresponding $N_{\max} \times N_{\max}$ (respectively, N_{\max}) quantities.

The Online cost (operation count and storage) to evaluate $\boldsymbol{\mu} \rightarrow s_N^{N_h}(\boldsymbol{\mu})$ is thus independent of N_h . The implications are two-fold: first, if N is indeed small, we will achieve very fast response in the real-time and many-query contexts; second, we may choose N_h very conservatively — to make sure that the error between the exact and FE predictions is very small — without adversely affecting the Online marginal cost.

We now turn to a more detailed discussion of sampling and (in Section 18.4) convergence in order to understand how, to a certain extent, why, we can achieve high accuracy for N independent of N_h and indeed $N \ll N_h$.

18.3.2 Sampling strategies

We first indicate a few preliminaries, then we provide one example of sampling strategies.

Let Ξ be a generic finite sample of points in \mathcal{D} that will serve as surrogate for \mathcal{D} in the calculation of errors (and, in Section 18.5, error bounds) over the parameter domain. Typically these samples are chosen by Monte Carlo methods with respect to a uniform or log-uniform density: Ξ is however sufficiently large to ensure that the reported results are insensitive to further refinement of the parameter sample.

Given a function $y: \mathcal{D} \rightarrow \mathbb{R}$, we define

$$\|y\|_{L^\infty(\Xi)} = \max_{\boldsymbol{\mu} \in \Xi} |y(\boldsymbol{\mu})|, \|y\|_{L^p(\Xi)} = \left(|\Xi|^{-1} \sum_{\boldsymbol{\mu} \in \Xi} |y|^p(\boldsymbol{\mu}) \right)^{1/p}.$$

Here $|\Xi|$ denotes the cardinality of the test sample Ξ . Given a function $z: \mathcal{D} \rightarrow V^{N_h}$ (or V^e), we then define

$$\|z\|_{L^\infty(\Xi; V)} = \max_{\boldsymbol{\mu} \in \Xi} \|z(\boldsymbol{\mu})\|_V, \|z\|_{L^p(\Xi; V)} = \left(|\Xi|^{-1} \sum_{\boldsymbol{\mu} \in \Xi} \|z(\boldsymbol{\mu})\|_V^p \right)^{1/p}.$$

We denote the particular samples which shall serve to select our RB space – or train our RB approximation – by Ξ_{train} . The cardinality of Ξ_{train} will be denoted $|\Xi_{\text{train}}| = n_{\text{train}}$. We note that although the test samples Ξ serve primarily to understand and assess the quality of the RB approximation and a posteriori error estimators, the train samples Ξ_{train} serve to *generate* the RB approximation. The choice of n_{train} and Ξ_{train} thus have important Offline and Online computational implications.

We now illustrate a sample strategy particular to RB Lagrange spaces. The method, introduced in [PR07, RHP08], can be viewed as a heuristic (more precisely,

sub-optimal) solution to the $L^\infty(\Xi_{\text{train}}; X)$ optimization problem analogous to the $L^2(\Xi_{\text{train}}; X)$ POD optimization problem.

We are given Ξ_{train} and N_{\max} , as well as $S_1 = \{\boldsymbol{\mu}^1\}$, $W_1^{N_h \text{ Greedy}} = \text{span}\{u^{N_h}(\boldsymbol{\mu}^1)\}$. In actual practice we may set N_{\max} either directly, or indirectly through a prescribed error tolerance. Then, for $N = 2, \dots, N_{\max}$, we find

$$\boldsymbol{\mu}^N = \arg \max_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \Delta_{N-1}(\boldsymbol{\mu}),$$

set $S_N = S_{N-1} \cup \{\boldsymbol{\mu}^N\}$, and update $W_N^{N_h \text{ Greedy}} = W_{N-1}^{N_h \text{ Greedy}} \cup \text{span}\{u^{N_h}(\boldsymbol{\mu}^N)\}$. As we shall describe in detail in Section 18.5, $\Delta_N(\boldsymbol{\mu})$ is a sharp, asymptotically inexpensive a posteriori error bound for $\|u^{N_h}(\boldsymbol{\mu}) - u_N^{N_h}(\boldsymbol{\mu})\|_V$.

Roughly, at iteration N this algorithm, called greedy Lagrange RB algorithm, appends to the *retained* snapshots that particular candidate snapshot — over all candidate snapshots $u^{N_h}(\boldsymbol{\mu})$, $\boldsymbol{\mu} \in \Xi_{\text{train}}$ — which is predicted by the a posteriori error bound to be the least well approximated by the RB prediction associated to $W_{N-1}^{N_h \text{ Greedy}}$.

An analogous greedy procedure can be developed also in the energy norm [RHP08] to build $W_N^{N_h \text{ Greedy, en}}$, being this norm particularly relevant in the compliant case, since the error in the energy norm is directly related to the error in the output (see Section 18.3.1).

18.4 Convergence of RB approximations

In this section we illustrate some convergence results for problems depending on one or several parameters.

18.4.1 A priori convergence theory: single parameter case: $P = 1$

We present from [MPT02b, MPT02a, PR07] an a priori theory for RB approximations associated with specific *non-hierarchical* Lagrange spaces $W^{N_h \text{ ln}}$, $1 \leq N \leq N_{\max}$, given by

$$W_N^{N_h \text{ ln}} = \text{span}\{u^{N_h}(\boldsymbol{\mu}_N^n), 1 \leq n \leq N\}, \quad (18.50)$$

for the parameter points given by

$$\boldsymbol{\mu}_N^n = \boldsymbol{\mu}^{\min} \exp \left\{ \frac{n-1}{N-1} \ln \left(\frac{\boldsymbol{\mu}^{\max}}{\boldsymbol{\mu}^{\min}} \right) \right\}, \quad 1 \leq n \leq N, \quad 1 \leq N \leq N_{\max}. \quad (18.51)$$

We denote the corresponding RB approximation by $u_N^{N_h \text{ ln}}$.

The a priori theory described below suggests that the spaces (18.50) — which we shall denote “equi-ln” spaces — contain certainly optimality properties, though we shall observe that the more automatic greedy sample selection procedure do as just as well (and perhaps even better for larger N). We note the analysis presented here in fact is relevant to a large class of single parameter coercive problems.

We consider the thermal block problem of Sec. 18.1.1 for the case in which $B_1 = 2$, $B_2 = 1$, as shown in Figure 18.7. The governing equations are then given by (18.11)

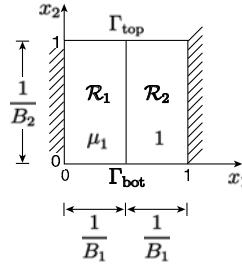


Fig. 18.7. Thermal block problem: $B_1 = 2, B_2 = 1$

and (18.12) for two blocks/regions \mathcal{R}_1 and \mathcal{R}_2 , the single parameter $\mu = \mu = \mu_1$ representing the conductivity of region \mathcal{R}_1 (the conductivity of region \mathcal{R}_2 is unity), and the parameter domain $\mathcal{D} = [\mu^{\min}, \mu^{\max}] = [1/\sqrt{\mu_r}, \sqrt{\mu_r}]$ for $\mu_r = 100$; the associated affine expansion (18.2) now comprises only $Q = 2$ terms.

Proposition 18.1. *Given general data F (of which F^{Neu} of (18.12) is a particular example), we obtain that for any $N \geq 1 + C_{\mu_r}$, and for all $\mu \in \mathcal{D}$,*

$$\frac{\|u^{N_h}(\mu) - u_N^{N_h \ln(\mu)}\|_\mu}{\|u^{N_h}(\mu)\|_\mu} \leq \exp \left\{ - \frac{N-1}{C_{\mu_r}} \right\}, \quad (18.52)$$

where $C_{\mu_r} = [2e \ln \mu_r]$ and $[]$ returns the smallest integer greater than or equal to its real argument.

Note we can directly derive from (18.52) and (18.42) a bound on the relative (compli-ant) output error.

The proof is a “parameter” version of the standard (finite element) variational arguments of Chap. 4. In particular, we first invoke (18.41); we then take as our candidate w a high-order polynomial interpolant *in the parameter* μ of $u^{N_h}(\mu)$; we next apply the standard Lagrange interpolant remainder formula; finally, we appeal to an eigenfunction expansion to bound the parametric (sensitivity) derivatives and optimize the order of the polynomial interpolant. For the complete proof and more considerations, see [PR07]. We note that the RB convergence estimate (18.52), relative to the model problem we have considered, relies on parameter smoothness and not on the FE grid (through N_h); the exponent in the convergence rate depends on N and logarithmically on μ_r .

18.4.2 Convergence: $P > 1$

As already highlighted in the previous section, the key to RB convergence in higher parameter dimensions is the role of the PDE and field variable in determining appropriate sample points and combinations of snapshots. We illustrate the process schematically in Fig. 18.8: the RB field approximation, *via* the PDE residual, yields the error bound; the error bound, in turn, facilitates the greedy selection of good sample points; the

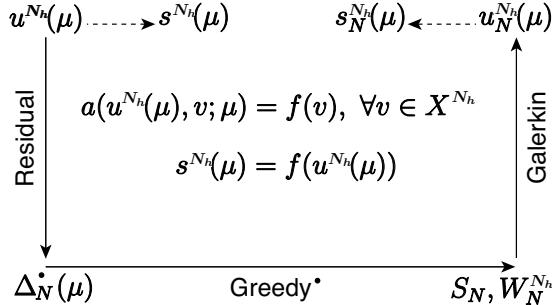


Fig. 18.8. Schematic of the RB approximation process

Galerkin projection then provides the optimal combination of retained snapshots; finally, the RB output approximation — application of the output functional — inherits the good properties of the RB field variable (18.42). The Greedy sample points are quite non-intuitive and very far from the obvious (and inefficient) uniform distribution. In general, however, we do observe clustering near the boundaries of \mathcal{D} , as we might expect from classical approximation theory.

The computational success of the (implicit) complicated process described by Figure 18.8 is in fact also responsible for the failure, at present, to provide any general *a priori* convergence theory: we can not construct an “optimal” approximant (like the piecewise polynomial interpolant in FE error analysis) since *a priori* we can neither predict an efficient sample nor construct an effective parametric interpolant.

We can anticipate that for a good set of points and from Galerkin a good combination of retained snapshots, we should obtain rapid convergence: $u^{N_h}(\mu) \in V^{N_h}$ — the field we wish to approximate by the RB method — perforce resides on the parametrically induced low-dimensional, *smooth* manifold $\mathcal{M}^{N_h} = \{u^{N_h}(\mu) | \mu \in \mathcal{D}\}$ ³; the essential role of parametric smoothness — already exploited in Sec. 18.4.1 for the single-parameter case — was identified in [FR83] and [Por85]. However, it is not obvious that a good set of points (and hence a good Lagrange RB space) must exist, and even less obvious that the greedy algorithm will identify any good set of points. At present, we have only empirical evidence, as described below. Note in all cases we consider RB approximations associated with the spaces $W_N^{N_h} = W_N^{N_h \text{ Greedy}, \text{en}}$.

Thermal block: $B_1 = B_2 = 3$. We first consider the thermal block problem introduced in Sec. 18.1.1 and (for $B_1 = B_2 = 3$) depicted in Fig. 18.1; note that now there are $P = 8$ parameters. For problems in one parameter, it is simple to choose “sufficiently rich” test and train samples; in the current situation, with $P = 8$ parameters, it is very difficult to afford — even with the greedy algorithm — a sufficiently rich test/train

³ As regards smoothness, we note that for $\Theta^q \in C^\infty(\mathcal{D})$, $1 \leq q \leq Q$, it can be shown under our coercivity, continuity, and affine hypotheses of Sec. 18.1 that $\|D^\sigma u^{N_h}(\mu)\|_X$ is bounded by a constant $C^{|\sigma|}$ (independent of N_h) for all $\mu \in \mathcal{D}$; here $D^\sigma u^{N_h}(\mu)$ refers to the σ multi-index derivative of u^{N_h} with respect to μ .

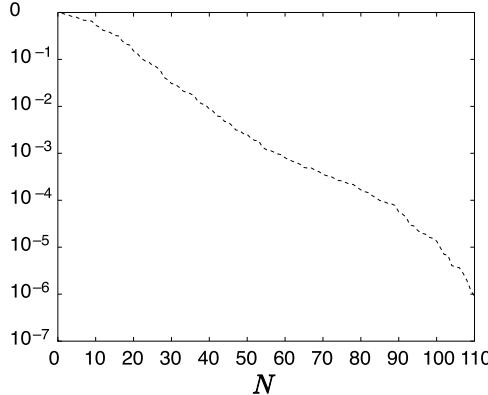


Fig. 18.9. Thermal block problem for $B_1 = B_2 = 3$: (upper bound for the) $L^\infty(\Xi_{\text{train}})$ relative energy error, (18.53), as a function of N

sample. We choose for Ξ_{train} a log-uniform random sample of size $n_{\text{train}} = 5000$; note that, in any event, we always have recourse to our a posteriori error bounds for any new $\mu \in \mathcal{D}$ used Online.

We display in Fig. 18.9 the error measure

$$\max_{\mu \in \Xi_{\text{train}}} \left(\frac{\Delta_N^{\text{en}}(\mu)}{\|u_N^{N_h}(\mu)\|_\mu} \right) \quad (18.53)$$

as a function of N ; here $\Delta_N^{\text{en}}(\mu)$ is an upper bound for $\|u^{N_h}(\mu) - u_N^{N_h}(\mu)\|_\mu$ and $\|u_N^{N_h}(\mu)\|_\mu$ is a lower bound for $\|u^{N_h}(\mu)\|_\mu$, and hence $\Delta_N^{\text{en}}(\mu)/\|u_N^{N_h}(\mu)\|_\mu$ is in fact an upper bound for the relative error in the energy norm.

We observe in Fig. 18.9 that, despite the rather large parameter dimension, and extensive parameter domain, the RB approximation still converges very rapidly w.r.t. N . We achieve an accuracy of $1E-2$ in the relative energy error (and hence an accuracy of $1E-4$ in the relative output error) with only $N \approx 40$ degrees of freedom. Results are largely insensitive to N_h for sufficiently large N_h (and any fixed N): it follows that the reduced basis can replicate an *arbitrarily rich* finite element approximation to *any desired accuracy* for N independent of N_h .

A Second example. We next consider the example of Sec. 18.2.5. For this problem, with only $P = 3$ parameters, we can now visualize the greedy-predicted sample. We show in Fig. 18.10 the sample $S_{N_{\text{max}}}$ obtained by application of the (energy version of the) greedy algorithm of Sec. 18.3.2 for Ξ_{train} a log-uniform random sample of size $n_{\text{train}} = 3000$. Clearly, the point distribution is very far from being uniform: there is some clustering near the boundaries of the parameter domain, however the interior of the domain is very sparsely populated.

We plot in Fig. 18.11 the quantity $\max_{\mu \in \Xi_{\text{train}}} (\Delta_N^{\text{en}}(\mu)/\|u_N^{N_h}(\mu)\|_\mu)$ for the Lagrange RB approximations associated with the sample of Fig. 18.10. We again ob-

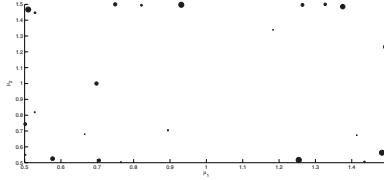


Fig. 18.10. Heat transfer in a cavity: greedy (energy version) sample $S_{N_{\max}}$ distribution (μ_1, μ_2); note the value of μ_3 ($0.01 \leq \mu_3 \leq 1$) is proportional to the dot thickness

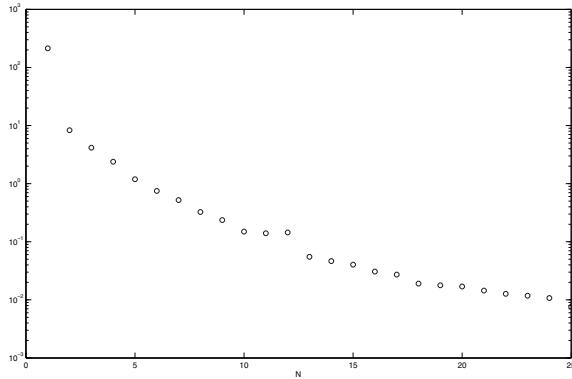


Fig. 18.11. Heat transfer in a cavity: $\max_{\mu \in \Xi_{\text{train}}} (\Delta_N^{\text{en}}(\mu) / \|u_N^{N_h}(\mu)\| \mu)$ as a function of N for the Lagrange RB approximations associated with the sample of Fig. 18.10; here Ξ_{train} is a log-uniform random sample of size $n_{\text{train}} = 3000$.

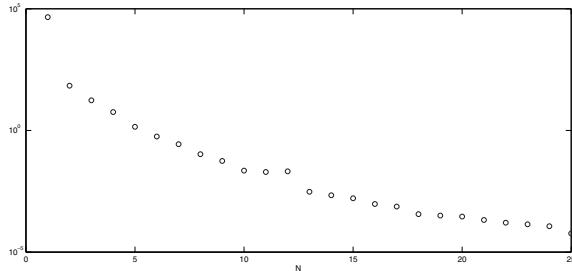


Fig. 18.12. Heat transfer in a cavity: (upper bound for the) $L^\infty(\Xi_{\text{train}})$ relative output error, (18.54), as a function of N

serve very rapid, exponential convergence. We first display in Fig. 18.12 the error measure

$$\max_{\mu \in \Xi_{\text{train}}} \frac{\Delta_N^s(\mu)}{s_N^{N_h}(\mu)} ; \quad (18.54)$$

here $\Delta_N^s(\mu)$ is an upper bound for $|s^{N_h}(\mu) - s_N^{N_h}(\mu)|$ and $s_N^{N_h}(\mu)$ is a lower bound for $s^{N_h}(\mu)$, and therefore $\Delta_N^s(\mu)/s_N^{N_h}(\mu)$ is in fact an upper bound for the relative

error in the output. We observe that we again achieve rapid convergence: to obtain a relative output error of $1\text{E}-4$, we require only $N \approx 25$ points.

A last remark about the spatial dimensionality which plays little role in RB convergence: it follows that the relative efficiency of the RB approach — relative to direct FE evaluation — increases with increasing spatial dimension [RHP08].

18.5 A posteriori error estimation

Effective a posteriori error bounds for the quantity of interest — our output — are crucial both for the efficiency and the reliability of RB approximations. As regards *efficiency* (related to the concept of “adaptivity” within the FE context), error bounds play a role in both the Offline and Online stages. In the greedy algorithm of Sec. 18.3.2, the application of error bounds (as surrogates for the actual error) permits significantly larger training samples $\Xi_{\text{train}} \subset \mathcal{D}$ at greatly reduced Offline computational cost. These more extensive training samples in turn engender RB approximations which provide high accuracy at greatly reduced Online computational cost. The error bounds also serve directly in the Online stage — to find the smallest RB dimension N that achieves the requisite accuracy — to further optimize Online performance. In short, a posteriori error estimation allows control of the error which in turn permits reduction of the computational effort.

We should emphasize that a posteriori output error bounds are particularly important for RB approximations. First, RB approximations are *ad hoc*: each problem is different as regards discretization. Second, RB approximations are typically pre-asymptotic concerning the convergence error: we will choose N quite small — before any “tail” in the convergence rate. Third, the RB basis functions can not be directly related to any spatial or temporal scales: physical intuition is of little value. And fourth and finally, the RB approach is typically applied in the real-time context: there is no time for Offline verification; errors are immediately manifested and often in deleterious ways. There is, thus, even greater need for a posteriori error estimation in the RB context than in the much more studied FE context (see Chap. 4).

The motivations for error estimation in turn place requirements on our error bounds. First, the error bounds must be *rigorous* — valid for all N and for all parameter values in the parameter domain \mathcal{D} : non-rigorous error “indicators” may suffice for adaptivity, but not for reliability. Second, the bounds must be reasonably *sharp*: an overly conservative error bound can yield inefficient approximations (N too large) or suboptimal engineering results (unnecessary safety margins); design should be dictated by the output and not the output error. And third, the bounds must be very *efficient*: the Online operation count and storage to compute the RB error bounds — the marginal or asymptotic average cost — must be independent of N_h .

18.5.1 Preliminaries

The central equation in a posteriori theory is the error residual relationship (see Sec. 4.6.2). In particular, it follows from the problem statements for $u^{N_h}(\boldsymbol{\mu})$, (18.3), and $u_N^{N_h}(\boldsymbol{\mu})$, (18.40), that the error $e^{N_h}(\boldsymbol{\mu}) = e(\boldsymbol{\mu}) = u^{N_h}(\boldsymbol{\mu}) - u_N^{N_h}(\boldsymbol{\mu}) \in V_h^N$ satisfies

$$a(e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = r(v; \boldsymbol{\mu}) \quad \forall v \in V^{N_h}. \quad (18.55)$$

Here $r(v; \boldsymbol{\mu}) \in (V^{N_h})'$, the dual space to V^{N_h} , is the residual,

$$r(v; \boldsymbol{\mu}) = F(v; \boldsymbol{\mu}) - a(u_N^{N_h}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) \quad \forall v \in V^{N_h}. \quad (18.56)$$

(18.55) directly follows from the definition (18.56), $F(v; \boldsymbol{\mu}) = a(u^{N_h}(\boldsymbol{\mu}), v; \boldsymbol{\mu})$, $\forall v \in V^{N_h}$, bilinearity of a , and the definition of $e(\boldsymbol{\mu})$.

It shall prove convenient to introduce the Riesz representation of $r(v; \boldsymbol{\mu})$ (see Theorem 2.1): $\hat{e}(\boldsymbol{\mu}) \in V^{N_h}$ satisfies

$$(\hat{e}(\boldsymbol{\mu}), v)_V = r(v; \boldsymbol{\mu}) \quad \forall v \in V^{N_h}. \quad (18.57)$$

We can thus also write the error residual equation (18.55) as

$$a(e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = (\hat{e}(\boldsymbol{\mu}), v)_V \quad \forall v \in V^{N_h}. \quad (18.58)$$

It also follows that

$$\|r(\cdot; \boldsymbol{\mu})\|_{(V^{N_h})'} = \sup_{v \in V^{N_h}} \frac{r(v; \boldsymbol{\mu})}{\|v\|_V} = \|\hat{e}(\boldsymbol{\mu})\|_V; \quad (18.59)$$

the evaluation of the dual norm of the residual through the Riesz representation theorem is central to the Offline-Online procedures developed in Section 18.5.3 below.

We recall the definition of the exact and FE coercivity constants, (18.7) and (18.8), respectively. We shall require a lower bound to the coercivity constant $\alpha_{LB}^{N_h}(\boldsymbol{\mu})$, $\alpha_{LB}^{N_h}: \mathcal{D} \rightarrow \mathbb{R}$, such that (i) $0 < \alpha_{LB}^{N_h}(\boldsymbol{\mu}) \leq \alpha^{N_h}(\boldsymbol{\mu}) \forall \boldsymbol{\mu} \in \mathcal{D}$, and (ii) the Online computational time to evaluate $\boldsymbol{\mu} \rightarrow \alpha_{LB}^{N_h}(\boldsymbol{\mu})$ is independent of N_h . In Section 18.5.3 we summarize a methodology [HRSP07] to construct the requisite lower bound.

18.5.2 Error bounds

We define error estimators for the energy norm and output as

$$\Delta_N^{\text{en}}(\boldsymbol{\mu}) = \|\hat{e}(\boldsymbol{\mu})\|_V / (\alpha_{LB}^{N_h}(\boldsymbol{\mu}))^{1/2},$$

and

$$\Delta_N^s(\boldsymbol{\mu}) = \|\hat{e}(\boldsymbol{\mu})\|_V^2 / \alpha_{LB}^{N_h}(\boldsymbol{\mu}),$$

respectively. We next introduce the effectivities associated with these error estimators as

$$\eta_N^{\text{en}}(\boldsymbol{\mu}) = \Delta_N^{\text{en}}(\boldsymbol{\mu}) / \|u^{N_h}(\boldsymbol{\mu}) - u_N^{N_h}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}},$$

and

$$\eta_N^s(\boldsymbol{\mu}) = \Delta_N^s(\boldsymbol{\mu}) / (s^{N_h}(\boldsymbol{\mu}) - s_N^{N_h}(\boldsymbol{\mu})) ,$$

respectively.

Clearly, the effectivities are a measure of the quality of the proposed estimator: for rigor, we shall insist upon effectivities ≥ 1 ; for sharpness, we desire effectivities as close to unity as possible. It has been demonstrated [RHP08, PR07] that for any $N = 1, \dots, N_{\max}$, the effectivities satisfy

$$1 \leq \eta_N^{\text{en}}(\boldsymbol{\mu}) \leq \sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{\alpha_{\text{LB}}^{N_h}(\boldsymbol{\mu})}} \quad \forall \boldsymbol{\mu} \in \mathcal{D} , \quad (18.60)$$

$$1 \leq \eta_N^s(\boldsymbol{\mu}) \leq \frac{\gamma^e(\boldsymbol{\mu})}{\alpha_{\text{LB}}^{N_h}(\boldsymbol{\mu})} \quad \forall \boldsymbol{\mu} \in \mathcal{D} . \quad (18.61)$$

Similar results can be obtained for $\Delta_N(\boldsymbol{\mu})$, the a posteriori error bound in the V norm.

It is important to observe that the effectivity upper bounds, (18.60) and (18.61), are *independent* of N , and hence stable with respect to *RB refinement*. Furthermore, it is sometimes possible (see Sec. 18.5.3) to provide a rigorous lower bound for $\alpha_{\text{LB}}^{N_h}(\boldsymbol{\mu})$ that depends only on $\boldsymbol{\mu}$: in this case we obtain an upper bound for the effectivity which is not only independent of N but also independent of N_h , and hence stable with respect to *FE refinement*.

18.5.3 Offline-online computational procedure

The error bounds of the previous section are of no utility without an accompanying Offline-Online computational approach.

The computationally crucial component of all the error bounds of the previous section is $\|\hat{e}(\boldsymbol{\mu})\|_V$, the dual norm of the residual.

To develop an Offline-Online procedure for the dual norm of the residual we first expand the residual (18.56) according to (18.43) and (18.2):

$$\begin{aligned} r(v; \boldsymbol{\mu}) &= F(v) - a(u_N^{N_h}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) \\ &= F(v) - a\left(\sum_{n=1}^N u_{Nn}(\boldsymbol{\mu}) \zeta_n^{N_h}, v; \boldsymbol{\mu}\right) \\ &= F(v) - \sum_{n=1}^N u_{Nn}(\boldsymbol{\mu}) a(\zeta_n^{N_h}, v; \boldsymbol{\mu}) \\ &= F(v) - \sum_{n=1}^N u_{Nn}(\boldsymbol{\mu}) \sum_{q=1}^Q \Theta^q(\boldsymbol{\mu}) a^q(\zeta_n^{N_h}, v) . \end{aligned} \quad (18.62)$$

If we insert (18.62) in (18.57) and apply linear superposition, we obtain

$$(\hat{e}(\boldsymbol{\mu}), v)_V = F(v) - \sum_{q=1}^Q \sum_{n=1}^N \Theta^q(\boldsymbol{\mu}) u_{Nn}(\boldsymbol{\mu}) a^q(\zeta_n^{N_h}, v) ,$$

or

$$\hat{e}(\boldsymbol{\mu}) = \mathcal{C} + \sum_{q=1}^Q \sum_{n=1}^N \Theta^q(\boldsymbol{\mu}) u_{Nn}(\boldsymbol{\mu}) \mathcal{L}_n^q,$$

where $\mathcal{C} \in V^{N_h}$, $(\mathcal{C}, v)_V = f(v) \forall v \in V^{N_h}$, and $\mathcal{L}_n^q \in V^{N_h}$ satisfies $(\mathcal{L}_n^q, v)_V = -a^q(\zeta_n^{N_h}, v) \forall v \in V^{N_h}$, $1 \leq n \leq N$, $1 \leq q \leq Q$.

We thus obtain

$$\begin{aligned} & \|\hat{e}(\boldsymbol{\mu})\|_V^2 \\ &= (\mathcal{C}, \mathcal{C})_V + \sum_{q=1}^Q \sum_{n=1}^N \Theta^q(\boldsymbol{\mu}) u_{Nn}(\boldsymbol{\mu}) \left\{ \right. \\ & \quad \left. 2(\mathcal{C}, \mathcal{L}_n^q)_V + \sum_{q'=1}^Q \sum_{n'=1}^N \Theta^{q'}(\boldsymbol{\mu}) u_{Nn'}(\boldsymbol{\mu}) (\mathcal{L}_n^q, \mathcal{L}_{n'}^{q'})_V \right\}, \end{aligned} \quad (18.63)$$

from which we can directly calculate the requisite dual norm of the residual through (18.59).

The Offline-Online decomposition is now clear. In the Offline stage we form the parameter-independent quantities. In particular, we compute the FE solutions $\mathcal{C}, \mathcal{L}_n^q$, $1 \leq n \leq N_{\max}$, $1 \leq q \leq Q$, and form/store $(\mathcal{C}, \mathcal{C})_V$, $(\mathcal{C}, \mathcal{L}_n^q)_V$, $(\mathcal{L}_n^q, \mathcal{L}_n^{q'})_V$, $1 \leq n, n' \leq N_{\max}$, $1 \leq q, q' \leq Q$. Note that a single matrix factorization suffices to obtain all $1 + QN_{\max}$ FE solutions. The Offline operation count depends on N_{\max} , Q , and N_h .

In the Online stage, given any “new” value of $\boldsymbol{\mu}$ and therefore $\Theta^q(\boldsymbol{\mu})$, $1 \leq q \leq Q$, $u_{Nn}(\boldsymbol{\mu})$, $1 \leq n \leq N$ we simply retrieve the stored quantities $(\mathcal{C}, \mathcal{C})_V$, $(\mathcal{C}, \mathcal{L}_n^q)_V$, $(\mathcal{L}_n^q, \mathcal{L}_n^{q'})_V$, $1 \leq n, n' \leq N$, $1 \leq q, q' \leq Q$, and then evaluate the sum (18.63). The Online operation count, and hence also the marginal cost and asymptotic average cost, is $O(Q^2 N^2)$ and is *independent of N_h* .⁴ Note that with hierarchical spaces the necessary quantities for any $N \in \{1, \dots, N_{\max}\}$ can be simply extracted from the corresponding quantities for $N = N_{\max}$.

Lower bounds for the coercivity constant. As introduced in Sec. 18.5, our a posteriori error analysis of reduced basis approximations to affinely parametrized partial differential equations requires a lower bound for the coercivity constant.

In essence, the discrete coercivity constant (18.8) is a generalized minimum eigenvalue [PR07]. There are many classical techniques for the estimation of minimum eigenvalues or minimum singular values.

An approach to the construction of lower bounds for coercivity (and, in the non-coercive case, inf-sup stability) constants is the Successive Constraint Method (SCM) introduced in [HRSP07]. The method — based on an Offline-Online strategy relevant in the many-query and real-time context — reduces the Online (real-time) calculation to a small Linear Program for which the operation count is *independent of N_h* , see for details in [RHP08].

⁴ It thus follows that the a posteriori error estimation contribution to the cost of the greedy algorithm of Section 18.3.2 is $O(QN_{\max}N_h) + O(Q^2 N_{\max}^2 N_h) + O(n_{\text{train}} Q^2 N_{\max}^3)$: we may thus choose N_h and n_{train} *independently* (and large).

18.6 Historical perspective, background and extensions

We end this chapter by putting RB methods in perspective. As we have seen, RB discretization is, in brief, a Galerkin projection on an N -dimensional approximation space. Initial work grew out of two related streams of inquiry: from the need for more effective, many-query design evaluation [FM71] and from the need for more efficient parameter continuation methods for nonlinear problems depending on a parameter [ASB78, Noo81, Noo82, NP80].

These early approaches were soon extended to (i) general finite-dimensional systems as well as certain classes of PDEs (and ODEs) [BR95, FR83, Lee91, NBS84, PL87, Rhe81, Rhe93], and (ii) a variety of different reduced basis approximation spaces — in particular Taylor and Lagrange [Por85] and more recently Hermite [IR98b] expansions. Further extensions were concerned with different applications and classes of equations, such as fluid dynamics and the incompressible Navier-Stokes equations [Gun89, IR98a, IR98b, IR01, Pet89].

In these early methods, the approximation spaces were rather local and typically low-dimensional in parameter (often a single parameter). This was primarily due to the absence of a posteriori error estimators and effective sampling procedures. Indeed, in more global, higher-dimensional parameter domains the *ad hoc* reduced basis predictions “far” from any sample points can not necessarily be trusted, and hence a posteriori error estimators are crucial to reliability. Moreover, sophisticated sampling strategies for parameters are crucial to convergence and computational efficiency.

Much current effort is thus devoted to development of (i) a posteriori error estimation procedures and in particular rigorous error bounds for outputs of interest [PRV⁺02], and (ii) effective sampling strategies in particular for many parameter case [NVP05, Roz05b, VPRP03]. The a posteriori error bounds are of course indispensable for rigorous certification of any particular reduced basis output prediction. However, the error estimators can also play an important role in efficient and effective sampling procedures: the inexpensive error bounds allow one first, to explore much larger subsets of the parameter domain in search of most representative or best “snapshots,” and second, to determine when we have *just enough* basis functions.

We note here that greedy sampling methods are similar in objective to, but very different in approach from, more well-known POD methods [Gun03a, KV02, LA00, Rav00, Sir87, WP02]. The former are applied in the (*multi*-dimensional) parameter domain, while the latter are most often applied in the (*one*-dimensional) temporal domain. However, POD techniques can be applied within the parametric RB context [BTDW03, CBS00, GPS07]. A brief comparison of greedy and POD approaches — computational cost and performance — is reported in [PR07, RHP08].

The reduced basis approach can also be readily applied to the more general case of affine linear elliptic *non-coercive* problems. The special issues associated with *saddle-point problems* [BF91a], in particular the Stokes equations of incompressible flow, are addressed for divergence-free spaces in [Gun89, IR98b, Pet89] and non-divergence-free spaces in [Roz05a, RV07]. The exploration of the “parameter + time” framework in the context of affine linear parabolic PDEs — such as the heat equation and the convection-diffusion equation — is carried out in [Gre05, GP05].

The reduced basis methodology, in both the elliptic and parabolic cases, can also be extended to problems with non-affine parametric variation. The strategy consists of reducing the nonaffine operator and data to approximate affine form, and then apply the methods developed for affine operators described in this chapter. However, this reduction must be done efficiently in order to avoid a proliferation of parametric functions and a corresponding degradation of Online response time. This extension is based on the so-called *empirical interpolation method* [BNMP04]: a collateral RB space for the nonaffine coefficient functions; an interpolation system that avoids costly (N_h -dependent) projections; and several a posterior error estimators. The empirical interpolation method within the context of RB treatment of elliptic and parabolic PDEs with nonaffine coefficient functions is considered in [GMNP07, Roz09]; the resulting approximations preserve the usual Offline-Online efficiency — the complexity of the Online stage is independent of N_h .

The reduced basis approach and associated Offline-Online procedures can be applied without serious computational difficulties to quadratic nonlinearities [VPRP03]. Much work focuses on the stationary incompressible Navier-Stokes equations [BF91b, GR86, Gun89]: suitable stable approximations are considered in [Gun89, IR98b, Pet89, QR07, RV07]; rigorous a posteriori error estimation is considered in [Dep08, NVP05, VP05, VPP03, NRHP08].

Needless to say, this is a fast growing research area in which substantial progresses are expected in the coming years.

18.7 Exercises

1. Consider an example of anisotropic conductivity. Referring to the illustrative example of Sec. 18.1.1 (the thermal block), reformulate the problem considering anisotropic heat conduction phenomena so that

$$a^q(w, v; \boldsymbol{\mu}) = \sum_{i,j=1}^2 \mu_{x_i x_j}^q \int_{\mathcal{R}_q} \frac{\partial w}{\partial x_i} \frac{\partial v}{\partial x_j} d\Omega \quad (18.64)$$

where $\{\mu_{x_i x_j}^q, i, j = 1, 2\}$ represent the conductivities modelling an anisotropic heat transfer in the region \mathcal{R}_q of the material of the thermal block. Provide a weak formulation of the parametrized problem; express $a(w, v; \boldsymbol{\mu})$ by an affine decomposition like (18.2) and indicate the total number of parameters P and the quantity Q for the cases a) $B_1 = B_2 = 3$, b) $B_1 = B_2 = 5$ and c) $B_1 = 3$ and $B_2 = 5$.

2. a) Consider a geometrical parametrization of the thermal block of Sec. 18.1.1 based on the formulation already provided in the text with $P = 8$ physical parameters (i.e. μ_s for $s = 1, \dots, 8$), representing isotropic heat transfer in

each sub-block, made up of different materials. Take into consideration the thermal block configuration given by $B_1 \times B_2 = 3 \times 3$ sub-blocks with additional $P_g = 6$ geometrical parameters (i.e. $\mu_{x_1}^i$ and $\mu_{x_2}^i$ for $i = 1, 2, 3$). This corresponds to having split the block so that the length of the thermal block is given by $\mu_{x_1}^1 + \mu_{x_1}^2 + \mu_{x_1}^3 = 1$ in x_1 direction and $\mu_{x_2}^1 + \mu_{x_2}^2 + \mu_{x_2}^3 = 1$ in x_2 direction; in this way the first sub-block \mathcal{R}_1 (see Fig. 18.1) has dimension $\mu_{x_1}^1 \times \mu_{x_2}^1$, whereas the sub-block \mathcal{R}_9 has dimension $\mu_{x_1}^3 \times \mu_{x_2}^3$. Write the complete formulation for $a(w, v; \boldsymbol{\mu})$ in the form given by (18.2) and indicate the number of forms $a^q(w, v; \boldsymbol{\mu})$ (i.e. Q). Then report the complete formulation for all $\Theta^q(\boldsymbol{\mu})$ parameter-dependent functions, in terms of physical parameters (i.e. μ_s for $s = 1, \dots, 8$) and geometrical parameters (i.e. $\mu_{x_1}^i$ and $\mu_{x_2}^i$ for $i = 1, 2, 3$).

- b) Propose also a range of variation for each geometrical parameter by respecting the given constraints (i.e. $\mu_{x_1}^1 + \mu_{x_1}^2 + \mu_{x_1}^3 = 1$ and $\mu_{x_2}^1 + \mu_{x_2}^2 + \mu_{x_2}^3 = 1$) and the consistency of the thermal block configuration, and by avoiding a geometrical degeneration of some elements \mathcal{R}_q .
3. Consider an advection-diffusion example in a rectangular domain $\Omega_o(\boldsymbol{\mu}) = [0, L[\times]0, 1[$ representing a channel with imposed Couette velocity profile $(x_{o2}, 0)$. Homogeneous Neumann (zero flux) boundary conditions are imposed on the bottom wall $\Gamma_{o,\text{bot}}$; homogeneous Dirichlet conditions are imposed on the top wall $\Gamma_{o,\text{top}}$ and on the “inflow” left boundary $\Gamma_{o,\text{in}}$; finally homogeneous Neumann conditions are imposed on the “outflow” right boundary $\Gamma_{o,\text{out}}$. The output of interest is the integral of the temperature over the heated (bottom) surface $\Gamma_{o,\text{bot}}$.

We consider two parameters $\boldsymbol{\mu} = (\mu_1, \mu_2)$: μ_1 is the channel length L , and μ_2 is the (global) Péclet number Pe ; the parameter domain is given by $\mathcal{D} = [1, 10] \times [0.1, 100]$. We now choose $\boldsymbol{\mu}_{\text{ref}} = (1, 1)$, which in turn defines the reference domain $\Omega = \Omega_o(\boldsymbol{\mu}_{\text{ref}})$.

If in terms of the original domain the bilinear form and the functional are given by

$$a_o(w, v; \boldsymbol{\mu}) = \int_{\Omega_o(L)} x_{o2} \frac{\partial w}{\partial x_1} v d\Omega + \frac{1}{\text{Pe}} \int_{\Omega_o(L)} \nabla w \cdot \nabla v d\Omega \quad (18.65)$$

and

$$F_o(v) = L_o(v) = \int_{\Gamma_{o,\text{bot}}(L)} v d\gamma, \quad (18.66)$$

rewrite the bilinear form $a(w, v, \boldsymbol{\mu})$ and the functional $F(v; \boldsymbol{\mu})$ in the reference domain formulation.

(This problem is considered in the last Sec. of [RHP08] and as one of the worked problems at <http://augustine.mit.edu>.)

- 4. a) Consider the steady Stokes problem of Sec. 15.2 (eq.(15.12)) and the lid driven cavity flow of Sec. 15.6. Take as possible geometrical parametrization for the cavity the aspect ratio $\mu = L/D$, where L is the length and D the height of the cavity, respectively. Write the parametrization and weak formulation of the problem in the reference domain.

- b) The reduced basis approximation spaces for a Stokes problem are given by $W_{N\mathbf{u}}^{N_h} = \{\mathbf{u}^{N_h}(\mu^n), \boldsymbol{\sigma}^{N_h}(\mu^n), 1 \leq n \leq N\}$ for the velocity and $W_{Np}^{N_h} = \{p^{N_h}(\mu^n), 1 \leq n \leq N\}$ for the pressure, where, for N selected μ^n , $\mathbf{u}^{N_h}(\mu^n)$ and $p^{N_h}(\mu^n)$ represent the finite elements solutions for velocity and pressure, respectively, and $\boldsymbol{\sigma}^{N_h}(\mu^n)$ is the solution of an auxiliary problem, called *supremizer* problem, which reads in the original domain

$$\int_{\Omega_o} \nabla \boldsymbol{\sigma}^{N_h} \cdot \nabla \mathbf{v} d\Omega = \int_{\Omega_o} p^{N_h}(\mu^n) \operatorname{div} \mathbf{v} d\Omega, \quad i = 1, 2. \quad (18.67)$$

The enrichment of the velocity space by the supremizer guarantees the stability of the RB approximation and the fulfillment of an equivalent *inf-sup* condition (see [RV07]). Write the reduced basis formulation for the Stokes problem; observe that the algebraic system obtained from the RB Galerkin projection (like (18.44) for the scalar case) features a block structure: moreover observe that this time matrices are full, contrarily to what happens with the finite element method.

- c) Do the same exercise considering the steady version of Navier-Stokes equations (Sec. 15.1) by including also the affine transformation and the subsequent parametrization on the trilinear convective term (Sec. 15.7) $c(\mathbf{w}, \mathbf{z}, \mathbf{v}; \boldsymbol{\mu})$, as considered, for example, in [QR07].

References

- [Ada75] Adams R. A. (1975) *Sobolev Spaces*. Academic Press, New York.
- [AFG⁺00] Almeida R. C., Feijóo R. A., Galeão A. C., Padra C., and Silva R. S. (2000) Adaptive finite element computational fluid dynamics using an anisotropic error estimator. *Comput. Methods Appl. Mech. Engrg.* 182: 379–400.
- [Ago03] Agoshkov V. (2003) *Optimal Control Methods and Adjoint Equations in Mathematical Physics Problems*. Institute of Numerical Mathematics, Russian Academy of Science, Moscow.
- [Aki94] Akin J. E. (1994) *Finite Elements for Analysis and Design*. Academic Press, London.
- [AO00] Ainsworth M. and Oden J. T. (2000) *A posteriori error estimation in finite element analysis*. Pure and Applied Mathematics. John Wiley and Sons, New York.
- [Ape99] Apel T. (1999) *Anisotropic Finite Elements: Local Estimates and Applications*. Book Series: Advances in Numerical Mathematics. Teubner, Stuttgart.
- [APV98] Achdou Y., Pironneau O., and Valentin F. (1998) *Équations aux dérivées partielles et applications*, chapter Shape control versus boundary control, pages 1–18. Éd. Sci. Méd. Elsevier, Paris.
- [Arp66] Arpaci V. S. (1966) *Conduction heat transfer*. Addison-Wesley, New York.
- [AS55] Allen D. N. G. and Southwell R. V. (1955) Relaxation methods applied to determine the motion, in two dimensions, of a viscous fluid past a fixed cylinder. *Quart. J. Mech. Appl. Math.* 8: 129–145.
- [AS99] Adalsteinsson D. and Sethian J. A. (1999) The fast construction of extension velocities in level set methods. *J. Comput. Phys.* 148(1): 2–22.
- [ASB78] Almroth B. O., Stern P., and Brogan F. A. (1978) Automatic choice of global shape functions in structural analysis. *AIAA Journal* 16: 525–528.
- [ATF87] Alekseev V., Tikhomninov V., and Fomin S. (1987) *Optimal Control*. Consultants Bureau, New York.
- [Aub67] Aubin J. P. (1967) Behavior of the error of the approximate solutions of boundary value problems for linear elliptic operators by Galerkin's and finite difference methods. *Ann. Scuola Norm. Sup. Pisa* 21: 599–637.
- [AWB71] Aziz A., Wingate J., and Balas M. (1971) *Control Theory of Systems Governed by Partial Differential Equations*. Academic Press, New York.
- [AZT] www.cs.sandia.gov/CRF/aztec1.html.
- [Bab71] Babuška I. (1971) Error bounds for the finite element method. *Numer. Math.* 16: 322–333.

- [BDR92] Babuška I., Durán R., and Rodríguez R. (1992) Analysis of the efficiency of an a posteriori error estimator for linear triangular finite elements. *SIAM J. Numer. Anal.* 29(4): 947–964.
- [BE92] Bern M. and Eppstein D. (1992) Mesh generation and optimal triangulation. In Du D.-Z. and Hwang F. (eds) *Computing in Euclidean Geometry*. World Scientific, Singapore.
- [Bec01] Becker R. (2001) Mesh adaptation for stationary flow control. *J. Math. Fluid Mech.* 3: 317–341.
- [BF91a] Brezzi F. and Fortin M. (1991) *Mixed and Hybrid Finite Element Methods*. Springer-Verlag, New York.
- [BF91b] Brezzi F. and Fortin M. (1991) *Mixed and Hybrid Finite Element Methods*, volume 15 of *Springer Series in Computational Mathematics*. Springer-Verlag, New York.
- [BFHR97] Brezzi F., Franca L. P., Hughes T. J. R., and Russo A. (1997) $b = \int g$. *Comput. Methods Appl. Mech. Engrg.* 145: 329–339.
- [BG87] Brezzi F. and Gilardi G. (1987) *Functional Analysis and Functional Spaces*. McGraw Hill, New York.
- [BG98] Bernardi C. and Girault V. (1998) A local regularisation operator for triangular and quadrilateral finite elements. *SIAM J. Numer. Anal.* 35(5): 1893–1916.
- [BGL05] Benzi M., Golub G. H., and Liesen J. (2005) Numerical solution of saddle-point problems. *Acta Numer.* 14: 1–137.
- [BGS96] Bjørstad P., Gropp P., and Smith B. (1996) *Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Univ. Cambridge Press, Cambridge.
- [BIL06] Berselli L. C., Iliescu T., and Layton W. J. (2006) *Mathematics of Large Eddy Simulation of Turbulent Flows*. Springer, Berlin Heidelberg.
- [BKR00] Becker R., Kapp H., and Rannacher R. (2000) Adaptive finite element methods for optimal control of partial differential equations: Basic concepts. *SIAM, J. Control Opt.* 39(1): 113–132.
- [Bla02] Blanckaert K. (2002) *Flow and turbulence in sharp open-channel bends*. PhD thesis, École Polytechnique Fédérale de Lausanne.
- [BM92] Bernardi C. and Maday Y. (1992) *Approximations Spectrales de Problèmes aux Limites Elliptiques*. Springer-Verlag, Berlin Heidelberg.
- [BMMP06] Bottasso C. L., Maisano G., Micheletti S., and Perotto S. (2006) On some new recovery based a posteriori error estimators. *Comput. Methods Appl. Mech. Engrg.* 195(37–40): 4794–4815.
- [BMS04] Brezzi F., Marini L. D., and Süli E. (2004) Discontinuous Galerkin methods for first-order hyperbolic problems. *Math. Models Methods Appl. Sci.* 14: 1893–1903.
- [BN83] Boland J. and Nicolaides R. (1983) Stability of finite elements under divergence constraints. *SIAM J. Numer. Anal.* 20: 722–731.
- [BNMP04] Barrault M., Nguyen N. C., Maday Y., and Patera A. T. (2004) An “empirical interpolation” method: Application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris, Série I.* 339: 667–672.
- [BO06] Benzi M. and Olshanskii M. (2006) An augmented lagrangian-based approach to the oseen problem. *SIAM J. on Scientific Computing* 28 (6): 2095–2113.
- [BR95] Barrett A. and Reddien G. (1995) On the reduced basis method. *Z. Angew. Math. Mech.* 75(7): 543–549.
- [Bre74] Brezzi F. (1974) On the existence, uniqueness and approximation of saddle-point problems arising from Lagrange multipliers. *R.A.I.R.O. Anal. Numér.* 8: 129–151.

- [Bre86] Brezis H. (1986) *Analisi Funzionale*. Liguori, Napoli.
- [Bre00] Bressan A. (2000) *Hyperbolic Systems of Conservation Laws: The One-dimensional Cauchy Problem*. Oxford Lecture Series in Mathematics and its Applications. The Clarendon Press Oxford University Press, New York.
- [BS94] Brenner S. C. and Scott L. R. (1994) *The Mathematical Theory of Finite Element Methods*. Springer-Verlag, New York.
- [BTDW03] Bui-Thanh T., Damodaran M., and Willcox K. (2003) Proper orthogonal decomposition extensions for parametric applications in transonic aerodynamics (AIAA Paper 2003-4213). In *Proceedings of the 15th AIAA Computational Fluid Dynamics Conference*.
- [Cab03] Caboussat A. (2003) *Analysis and numerical simulation of free surface flows*. PhD thesis, École Polytechnique Fédérale de Lausanne.
- [Caz97] Cazemier W. (1997) *Proper Orthogonal Decomposition and Low Dimensional Models for Turbulent Flows*. University of Groningen.
- [CBS00] Christensen E., Brøns M., and Sørensen J. (2000) Evaluation of proper orthogonal decomposition-based decomposition techniques applied to parameter-dependent nonturbulent flows. *SIAM J. Scientific Computing* 21(4): 1419–1434.
- [Ç07] Çengel Y. (2007) *Introduction to Thermodynamics and heat transfer*. McGraw-Hill, New York.
- [CH01] Collis S. and Heinkenschloss M. (2001) Analysis of the streamline upwind/petrov galerkin method applied to the solution of optimal control problems. *CAAM report TR02-01*.
- [CHQZ06] Canuto C., Hussaini M., Quarteroni A., and Zang T. A. (2006) *Spectral Methods. Fundamentals in Single Domains*. Springer-Verlag, Berlin Heidelberg.
- [CHQZ07] Canuto C., Hussaini M. Y., Quarteroni A., and Zang T. A. (2007) *Spectral Methods. Evolution to Complex Geometries and Application to Fluid Dynamics*. Springer-Verlag, Berlin Heidelberg.
- [Cia78] Ciarlet P. G. (1978) *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam.
- [CJRT01] Cohen G., Joly P., Roberts J. E., and Tordjman N. (2001) Higher order triangular finite elements with mass lumping for the wave equation. *SIAM J. Numer. Anal.* 38(6): 2047–2078 (electronic).
- [Clé75] Clément P. (1975) Approximation by finite element functions using local regularization. *RAIRO, Anal. Numér* 2 pages 77–84.
- [Coc98] Cockburn B. (1998) An introduction to the discontinuous Galerkin method for convection-dominated problems. In Quarteroni A. (ed) *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, volume 1697 of *LNM*, pages 151–268. Springer-Verlag, Berlin Heidelberg.
- [Com95] Comincioli V. (1995) *Analisi Numerica. Metodi Modelli Applicazioni*. McGraw-Hill, Milano.
- [CP00] Cremonesi P. and Psaila G. (2000) *Introduzione ragionata al C/C++*. Progetto Leonardo, Esculapio, Bologna.
- [CZ87] C.Zienkiewicz O. and Zhu J. Z. (1987) A simple error estimator and adaptive procedure for practical engineering analysis. *Int. J. Numer. Meth. Engng.* 24: 337–357.
- [Ded04] Dedé L. (2004) *Controllo Ottimale e Adattività per Equazioni alle Derivate Parziali e Applicazioni*. Tesi di Laurea, Politecnico di Milano.
- [Dep08] Deparis S. (2008) Reduced basis error bound computation of parameter-dependent Navier-Stokes equations by the natural norm approach. *SIAM Journal of Numerical Analysis* 46(4): 2039–2067.

- [DPQ08] Detomi D., Parolini N., and Quarteroni A. (2008) Mathematics in the wind. *MOX Reports* (25). see the web page <http://mox.polimi.it>.
- [DQ05] Dedè L. and Quarteroni A. (2005) Optimal control and numerical adaptivity for advection–diffusion equations. *M2AN Math. Model. Numer. Anal.* 39(5): 1019–1040.
- [DT80] Dervieux A. and Thomasset F. (1980) *Approximation Methods for Navier–Stokes Problems*, volume 771 of *Lecture Notes in Mathematics*, chapter A finite element method for the simulation of Rayleigh-Taylor instability, pages 145–158. Springer-Verlag, Berlin.
- [Dub91] Dubiner M. (1991) Spectral methods on triangles and other domains. *J. Sci. Comput.* 6: 345–390.
- [DV02] Darmofal D. L. and Venditti D. A. (2002) Grid adaptation for functional outputs: application to two-dimensional inviscid flows. *J. Comput. Phys.* 176: 40–69.
- [DV08] DiPietro D. A. and Veneziani A. (2008) Expression templates implementation of continuous and discontinuous Galerkin methods. *Computing and Visualization in Science* 12.
- [DZ06] Dáger R. and Zuazua E. (2006) *Wave Propagation, Observation and Control in 1-d Flexible Multi-Structures*. Mathématiques et Applications. Springer, Paris.
- [EG04] Ern A. and Guermond J. L. (2004) *Theory and Practice of Finite Elements*, volume 159 of *Applied Mathematics Sciences*. Springer-Verlag, New York.
- [EHS⁺06] Elman H. C., Howte V. E., Shadid J., Shuttleworth R., and Tuminaro R. (2006) Block preconditioners based on approximate commutators. *SIAM J. on Scientific Computing* 27 (5): 1651–1668.
- [EJ88] Eriksson E. and Johnson C. (1988) An adaptive method for linear elliptic problems. *Math. Comp.* 50: 361–383.
- [Emb99] Embree M. (1999) *Convergence of Krylov subspace methods for non-normal matrices*. PhD thesis, Oxford University Computing Laboratories.
- [Emb03] Embree M. (2003) The tortoise and the hare restart GMRES. *SIAM Rev.* 45 (2): 259–266.
- [ESW05] Elman H., Silvester D., and Wathen A. (2005) *Finite Elements and Fast Iterative Solvers*. Oxford Science Publications, Oxford.
- [Eva98] Evans L. (1998) *Partial differential equations*. American Mathematical Society, Providence.
- [FCZ03] Fernández-Cara E. and Zuazua E. (2003) Control theory: History, mathematical achievements and perspectives. *Bol. Soc. Esp. Mat. Apl.* 26: 79–140.
- [FCZ04] Fernández-Cara E. and Zuazua E. (2004) On the history and perspectives of control theory. *Matapli* 74: 47–73.
- [FM71] Fox R. L. and Miura H. (1971) An approximate analysis technique for design calculations. *AIAA Journal* 9(1): 177–179.
- [FMP04] Formaggia L., Micheletti S., and Perotto S. (2004) Anisotropic mesh adaptation in computational fluid dynamics: application to the advection-diffusion-reaction and the stokes problems. *Appl. Numer. Math.* 51(4): 511–533.
- [FMRT01] Foias C., Manley O., Rosa R., and Temam R. (2001) *Navier-Stokes Equations and Turbulence*. Cambridge Univ. Press, Cambridge.
- [For77] Fortin M. (1977) An analysis of the convergence of mixed finite element methods. *R.A.I.R.O. Anal. Numér.* 11.
- [FP01] Formaggia L. and Perotto S. (2001) New anisotropic a priori error estimates. *Numer. Math.* 89: 641–667.
- [FP02] Ferziger J. H. and Peric M. (2002) *Computational Methods for Fluid Dynamics*. Springer, Berlin, III edition.

- [FR83] Fink J. P. and Rheinboldt W. C. (1983) On the error behavior of the reduced basis technique for nonlinear finite element approximations. *Z. Angew. Math. Mech.* 63(1): 21–28.
- [FSV05] Formaggia L., Saleri F., and Veneziani A. (2005) *Applicazioni ed esercizi di modellistica numerica per problemi differenziali*. Springer Italia, Milano.
- [Fun92] Funaro D. (1992) *Polynomial Approximation of Differential Equations*. Springer-Verlag, Berlin Heidelberg.
- [Fun97] Funaro D. (1997) *Spectral Elements for Transport-Dominated Equations*. Springer-Verlag, Berlin Heidelberg.
- [Fur97] Furnish G. (May/June 1997) Disambiguated glommable expression templates. *Computers in Physics* 11(3): 263–269.
- [GB98] George P. L. and Borouchaki H. (1998) *Delaunay Triangulation and Meshing*. Editions Hermès, Paris.
- [Ger08] Gervasio P. (2008) Convergence analysis of high order algebraic fractional step schemes for timedeependent stokes equations. *SINUM*.
- [GMNP07] Grepl M. A., Maday Y., Nguyen N. C., and Patera A. T. (2007) Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *M2AN (Math. Model. Numer. Anal.)*.
- [GMSW89] Gill P., Murray W., Saunders M., and Wright M. (1989) *Constrained nonlinear programming*. Elsevier Handbooks In Operations Research And Management Science Optimization. Elsevier North-Holland, Inc., New York.
- [GNV⁺07] Grepl M. A., Nguyen N. C., Veroy K., Patera A. T., and Liu G. R. (2007) Certified rapid solution of partial differential equations for real-time parameter estimation and optimization. In Biegler L. T., Ghattas O., Heinkenschloss M., Keyes D., and van Wandeers B. (eds) *Proceedings of the 2nd Sandia Workshop of PDE-Constrained Optimization: Real-Time PDE-Constrained Optimization*, SIAM Computational Science and Engineering Book Series, pages 197–216.
- [GP05] Grepl M. A. and Patera A. T. (2005) *A Posteriori* error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *M2AN (Math. Model. Numer. Anal.)* 39(1): 157–181.
- [GPS07] Gunzburger M. D., Peterson J., and Shadid J. N. (2007) Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data. *Comp. Meth. Applied Mech.* 196: 1030–1047.
- [GR86] Girault V. and Raviart P. (1986) *Finite Element Approximation of the Navier-Stokes Equations*. Springer-Verlag, Berlin.
- [GR96] Godlewski E. and Raviart P. A. (1996) *Hyperbolic Systems of Conservation Laws*, volume 118. Springer-Verlag, New York.
- [Gre05] Grepl M. (May 2005) *Reduced-Basis Approximations and A Posteriori Error Estimation for Parabolic Partial Differential Equations*. PhD thesis, Massachusetts Institute of Technology.
- [Gri76] Grisvard P. (1976) *Behaviour of the solutions of an elliptic boundary value problem in a polygonal or polyhedral domain*. Numerical Solution of Partial Differential Equations, III. Academic Press, New York.
- [GRS07] Grossmann C., Ross H., and Stynes M. (2007) *Numerical treatment of Partial Differential Equations*. Springer, Heidelberg, Heidelberg.
- [GSV06] Gervasio P., Saleri F., and Veneziani A. (2006) Algebraic fractional-step schemes with spectral methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.* 214(1): 347–365.
- [Gun89] Gunzburger M. D. (1989) *Finite Element Methods for Viscous Incompressible Flows*. Academic Press, Boston.

- [Gun03a] Gunzburger M. D. (2003) *Perspectives in Flow Control and Optimization*. Advances in Design and Control. SIAM.
- [Gun03b] Gunzburger M. (2003) *Perspectives in Flow Control and Optimization. Advances in Design and Control*. SIAM, Philadelphia.
- [HB76] Hnat J. and Buckmaster J. (1976) Spherical cap bubbles and skirt formation. *Phys. Fluids* 19: 162–194.
- [Hir88] Hirsh C. (1988) *Numerical Computation of Internal and External Flows*, volume 1. John Wiley and Sons, Chichester.
- [HN81] Hirt C. W. and Nichols B. D. (1981) Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comp. Phys.* 39: 201–225.
- [HNRP08] Huynh D. B. P., Nguyen C. N., Rozza G., and Patera A. T. (2007-08) *rbMIT Software*: http://augustine.mit.edu/methodology/methodology_rbMIT_System.htm. ©MIT, Tech. Lic. Office 12600, Cambridge, MA.
- [Hou95] Hou T. Y. (1995) Numerical solutions to free boundary problems. *ACTA Numerica* 4: 335–415.
- [HRSP07] Huynh D. B. P., Rozza G., Sen S., and Patera A. T. (2007) A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *C. R. Acad. Sci. Paris, Analyse Numérique* 345(8): 473–478.
- [HRT96] Heywood J. G., Rannacher R., and Turek S. (1996) Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations. *Internat. J. Numer. Methods Fluids* 22(5): 325–352.
- [Hug00] Hughes T. J. R. (2000) *The Finite Element Method. Linear Static and Dynamic Finite Element Analysis*. Dover Publishers, New York.
- [HVZ97] Hamacher V. C., Vranesic Z. G., and Zaky S. G. (1997) *Introduzione all’architettura dei calcolatori*. McGraw Hill Italia, Milano.
- [HW65] Harlow F. H. and Welch J. E. (1965) Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids* 8(12): 2182–2189.
- [HYR08] Hou T. Y., Yang D. P., and Ran H. (2008) Multiscale analysis and computation for the 3d incompressible navier-stokes equations. *SIAM Multiscale Modeling and Simulation* 6 (4): 1317–1346.
- [IR98a] Ito K. and Ravindran S. S. (1998) A reduced basis method for control problems governed by PDEs. In Desch W., Kappel F., and Kunisch K. (eds) *Control and Estimation of Distributed Parameter Systems*, pages 153–168. Birkhäuser.
- [IR98b] Ito K. and Ravindran S. S. (1998) A reduced-order method for simulation and control of fluid flows. *Journal of Computational Physics* 143(2): 403–425.
- [IR01] Ito K. and Ravindran S. S. (2001) Reduced basis method for optimal control of unsteady viscous flows. *International Journal of Computational Fluid Dynamics* 15(2): 97–113.
- [IZ99] Infante J. and Zuazua E. (1999) Boundary observability for the space semi-discretizations of the 1-d wave equation. *M2AN Math. Model. Numer. Anal.* 33(2): 407–438.
- [Jam88] Jameson A. (1988) Optimum aerodynamic design using CFD and control theory. *AIAA Paper 95-1729-CP* pages 233–260.
- [Joe05] Joerg M. (2005) Numerical investigations of wall boundary conditions for two-fluid flows. Master’s thesis, École Polytechnique Fédérale de Lausanne.
- [Joh87] Johnson C. (1987) *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Cambridge.

- [KA00] Knabner P. and Angermann L. (2000) *Numerical Methods for Elliptic and Parabolic Partial Differential Equations*, volume 44 of *TAM*. Springer-Verlag, New York.
- [KAJ02] Kim S., Alonso J., and Jameson A. (2002) Design optimization of hight-lift configurations using a viscous continuos adjoint method. *AIAA paper, 40th AIAA Aerospace Sciences Meeting and Exhibit, Jan 14-17 2002* 0844.
- [KF89] Kolmogorov A. and Fomin S. (1989) *Elements of the Theory of Functions and Functional Analysis*. V.M. Tikhomirov, Nauka - Moscow.
- [KMI⁺83] Kajitani H., Miyata H., Ikehata M., Tanaka H., Adachi H., Nanimatzu M., and Ogiwara S. (1983) Summary of the cooperative experiment on Wigley parabolic model in Japan. In *Proc. of the 2nd DTNSRDC Workshop on Ship Wave Resistance Computations (Bethesda, USA)*, pages 5–35.
- [KPTZ00] Kawohl B., Pironneau O., Tartar L., and Zolesio J. (2000) *Optimal Shape Design*. Springer-Verlag, Berlin.
- [Kro97] Kroener D. (1997) *Numerical Schemes for Conservation Laws*. Wiley-Teubner, Chichester.
- [KS05] Karniadakis G. E. and Sherwin S. J. (2005) *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, New York, II edition.
- [KV02] Kunisch K. and Volkwein S. (2002) Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Num. Analysis* 40(2): 492–515.
- [LA00] LeGresley P. A. and Alonso J. J. (2000) Airfoil design optimization using reduced order models based on proper orthogonal decomposition. In *Fluids 2000 Conference and Exhibit, Denver, CO*. Paper 2000-2545.
- [Le 05] Le Bris C. (2005) *Systèmes multiéchelles: modélisation et simulation*, volume 47 of *Mathématiques et Applications*. Springer, Paris.
- [Lee91] Lee M. Y. L. (1991) Estimation of the error in the reduced-basis method solution of differential algebraic equations. *SIAM Journal of Numerical Analysis* 28: 512–528.
- [LeV02a] LeVeque R. J. (2002) *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics.
- [LeV02b] LeVeque R. J. (2002) *Numerical Methods for Conservation Laws*. Birkhäuser Verlag, Basel, II edition.
- [LeV07] LeVeque R. J. (2007) *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM, Philadelphia.
- [Lio71] Lions J. (1971) *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York.
- [Lio72] Lions J. (1972) *Some Aspects of the Optimal Control of Distributed Parameter Systems*. SIAM, Philadelphia.
- [Lio96] Lions P.-L. (1996) *Mathematical topics in fluid mechanics. Vol. 1*, volume 3 of *Oxford Lecture Series in Mathematics and its Applications*. The Clarendon Press Oxford University Press, New York. Incompressible models, Oxford Science Publications.
- [LL59] Landau L. D. and Lifshitz E. M. (1959) *Fluid mechanics*. Translated from the Russian by J. B. Sykes and W. H. Reid. Course of Theoretical Physics, Vol. 6. Pergamon Press, London.
- [LL00] Lippman S. B. and Lajoie J. (2000) *C++ Corso di Programmazione*. Addison Wesley Longman Italia, Milano, III edition.
- [LM68] Lions J. L. and Magenes E. (1968) *Quelques Méthodes des Résolution des Problèmes aux Limites non Linéaires*. Dunod, Paris.

- [Loe55] Loeve M. M. (1955) *Probability Theory*. Van Nostrand, Toronto-New York-London.
- [LR98] Lin S. P. and Reitz R. D. (1998) Drop and spray formation from a liquid jet. *Annu. Rev. Fluid Mech.* 30: 85–105.
- [LW94] Li X. D. and Wiberg N. E. (1994) A posteriori error estimate by element patch post-processing, adaptive analysis in energy and L_2 norms. *Comp. Struct.* 53: 907–919.
- [Mar95] Marchuk G. I. (1995) *Adjoint Equations and Analysis of Complex Systems*. Kluwer Academic Publishers, Dordrecht.
- [Mau81] Maurer H. (1981) First and second order sufficient optimality conditions in mathematical programming and optimal control. *Mathematical Programming Study* 14: 163–177.
- [Max76] Maxworthy T. (1976) Experiments on collisions between solitary waves. *Journal of Fluid Mechanics* 76: 177–185.
- [Mey00] Meyer C. D. (2000) *Matrix Analysis and Applied Linear Algebra*. SIAM.
- [MOS92] Mulder W., Osher S., and Sethian J. (1992) Computing interface motion in compressible gas dynamics. *Journal of Computational Physics* 100(2): 209–228.
- [MP94] Mohammadi B. and Pironneau O. (1994) *Analysis of the K-Epsilon Turbulence Model*. John Wiley & Sons, Chichester.
- [MP97] Muzaferija S. and Peric M. (1997) Computation of free-surface flows using finite volume method and moving grids. *Numer. Heat Trans., Part B* 32: 369–384.
- [MP01] Mohammadi B. and Pironneau O. (2001) *Applied Shape Optimization for Fluids*. Clarendon Press, Oxford.
- [MPT02a] Maday Y., Patera A. T., and Turinici G. (2002) Global *a priori* convergence theory for reduced-basis approximation of single-parameter symmetric coercive elliptic partial differential equations. *C. R. Acad. Sci. Paris, Série I* 335(3): 289–294.
- [MPT02b] Maday Y., Patera A., and Turinici G. (2002) *A Priori* convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. *Journal of Scientific Computing* 17(1-4): 437–446.
- [NBS84] Noor A. K., Balch C. D., and Shubit M. A. (1984) Reduction methods for non-linear steady-state thermal analysis. *Int. J. Num. Meth. Engrg.* 20: 1323–1348.
- [Nit68] Nitsche J. A. (1968) Ein kriterium für die quasi-optimalität des Ritzchen Verfahrens. *Numer. Math.* 11: 346–348.
- [Noo81] Noor A. K. (1981) Recent advances in reduction methods for nonlinear problems. *Comput. Struct.* 13: 31–44.
- [Noo82] Noor A. K. (1982) On making large nonlinear problems small. *Comp. Meth. Appl. Mech. Engrg.* 34: 955–985.
- [NP80] Noor A. K. and Peters J. M. (1980) Reduced basis technique for nonlinear analysis of structures. *AIAA Journal* 18(4): 455–462.
- [NRHP08] Nguyen N. C., Rozza G., Huynh D. B. P., and Patera A. T. (2008) Reduced basis approximation and a posteriori error estimation for parametrized parabolic pdes; Application to real-time Bayesian parameter estimation. In Biegler L., Biros G., Ghattas O., Heinkenschloss M., Keyes D., Mallick B., Tenorio L., van Bloemen Waanders B., and Willcox K. (eds) *Computational Methods for Large Scale Inverse Problems and Uncertainty Quantification*. John Wiley and Sons, Chichester.
- [NVP05] Nguyen N. C., Veroy K., and Patera A. T. (2005) Certified real-time solution of parametrized partial differential equations. In Yip S. (ed) *Handbook of Materials Modeling*, pages 1523–1558. Springer, Netherlands.
- [NW06] Nocedal J. and Wright S. (2006) *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York.

- [NZ04] Naga A. and Zhang Z. (2004) A posteriori error estimates based on the polynomial preserving recovery. *SIAM J. Numer. Anal.* 42: 1780–1800.
- [OP07] Oliveira I. and Patera A. (2007) Reduced-basis techniques for rapid reliable optimization of systems described by affinely parametrized coercive elliptic partial differential equations. *Optimization and Engineering* 8(1): 43–65.
- [OS88] Osher S. and Sethian J. A. (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* 79(1): 12–49.
- [Pat80] Patankar S. V. (1980) *Numerical Heat Transfer and Fluid Flow*. Hemisphere, Washington.
- [Pat84] Patera A. (1984) A spectral element method for fluid dynamics: laminar flow in a channel expansion. *J. Comput. Phys.* 54: 468–488.
www.mcs.anl.gov/petsc/.
- [Pet89] Peterson J. S. (1989) The reduced basis method for incompressible viscous flow calculations. *SIAM J. Sci. Stat. Comput.* 10(4): 777–786.
- [Pir84] Pironneau O. (1984) *Optimal Shape Design for Elliptic Systems*. Springer-Verlag, New York.
- [PL87] Porsching T. A. and Lee M. Y. L. (1987) The reduced-basis method for initial value problems. *SIAM Journal of Numerical Analysis* 24: 1277–1287.
- [Por85] Porsching T. A. (1985) Estimation of the error in the reduced basis method solution of nonlinear equations. *Mathematics of Computation* 45(172): 487–496.
- [PQ05] Parolini N. and Quarteroni A. (2005) Mathematical models and numerical simulations for the america’s cup. *Comput. Methods Appl. Mech. Engrg.* 194(9–11): 1001–1026.
- [PQ07] Parolini N. and Quarteroni A. (2007) Modelling and numerical simulation for yacht engineering. In *Proceedings of the 26th Symposium on Naval Hydrodynamics*. Strategic Analysis, Inc., Arlington, VA.
- [PR07] Patera A. T. and Rozza G. (2006–2007) *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*. Copyright MIT. To appear in MIT Pappalardo Monographs in Mechanical Engineering.
- [Pro97] Prohl A. (1997) *Projection and Quasi-Compressibility Methods for Solving the Incompressible Navier-Stokes Equations*. Advances in Numerical Mathematics. B.G. Teubner, Stuttgart.
- [Pru06] Prud’homme C. (2006) A domain specific embedded language in c++ for automatic differentiation, projection, integration and variational formulations. *Scientific Programming* 14(2): 81–110.
- [PRV⁺02] Prud’homme C., Rovas D., Veroy K., Maday Y., Patera A., and Turinici G. (2002) Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bounds methods. *Journal of Fluids Engineering* 124(1): 70–80.
- [PWY90] Pawlak T. P., Wheeler M. J., and Yunus S. M. (1990) Application of the Zienkiewicz-Zhu error estimator for plate and shell analysis. *Int. J. Numer. Methods Eng.* 29: 1281–1298.
- [QR07] Quarteroni A. and Rozza G. (2007) Numerical solution of parametrized Navier-Stokes equations by reduced basis method. *Num. Meth. PDEs* 23: 923–948.
- [QRDQ06] Quarteroni A., Rozza G., Dedè L., and Quaini A. (2006) Numerical approximation of a control problem for advection–diffusion processes. System modeling and optimization. *IFIP Int. Fed. Inf. Process.* 199: 261–273.
- [QRQ06] Quarteroni A., Rozza G., and Quaini A. (2006) Reduced basis method for optimal control af advection-diffusion processes. In Fitzgibbon W., Hoppe R., Periaux J.,

- Pironneau O., and Vassilevski Y. (eds) *Advances in Numerical Mathematics*, pages 193–216. Moscow, Institute of Numerical Mathematics, Russian Academy of Sciences and Houston, Department of Mathematics, University of Houston.
- [QSS07] Quarteroni A., Sacco R., and Saleri F. (2007) *Numerical Mathematics*. Springer, Berlin Heidelberg, II edition.
- [QSV00] Quarteroni A., Saleri F., and Veneziani A. (2000) Factorization methods for the numerical approximation of Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.* 188(1-3): 505–526.
- [Qu02] Qu Z. (2002) *Unsteady open-channel flow over a mobile bed*. PhD thesis, École Polytechnique Fédérale de Lausanne.
- [Qua93] Quartapelle L. (1993) *Numerical Solution of the Incompressible Navier-Stokes Equations*. Birkhäuser Verlag, Basel.
- [QV94] Quarteroni A. and Valli A. (1994) *Numerical Approximation of Partial Differential Equations*. Springer, Berlin Heidelberg.
- [QV99] Quarteroni A. and Valli A. (1999) *Domain Decomposition Methods for Partial Differential Equations*. Oxford Science Publications, Oxford.
- [Ran99] Rannacher R. (1999) Error control in finite element computations. An introduction to error estimation and mesh-size adaptation. In *Error control and adaptivity in scientific computing (Antalya, 1998)*, pages 247–278. Kluwer Acad. Publ., Dordrecht.
- [Rav00] Ravindran S. S. (2000) A reduced order approach to optimal control of fluids flow using proper orthogonal decomposition. *Int. J. of Numerical Methods in Fluids* 34(5): 425–448.
- [RC83] Rhee C. M. and Chow W. L. (1983) Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal* 21(11): 1525–1532.
- [Rhe81] Rheinboldt W. C. (1981) Numerical analysis of continuation methods for nonlinear structural problems. *Computers and Structures* 13(1-3): 103–113.
- [Rhe93] Rheinboldt W. C. (1993) On the theory and error estimation of the reduced basis method for multi-parameter problems. *Nonlinear Analysis, Theory, Methods and Applications* 21(11): 849–858.
- [RHP08] Rozza G., Huynh D. B. P., and Patera A. T. (2008) Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics. *Archives Computational Methods in Engineering* 15(3): 229–275.
- [Rod94] Rodríguez R. (1994) Some remarks on Zienkiewicz-Zhu estimator. *Numer. Methods Part. Diff. Eq.* 10: 625–635.
- [Roz05a] Rozza G. (2005) Real-time reduced basis techniques for arterial bypass geometries. In Bathe K. (ed) *Computational Fluid and Solid Mechanics*, pages 1283–1287. Elsevier. Proceedings of the Third M.I.T. Conference on Computational Fluid and Solid Mechanics, June 14-17, 2005.
- [Roz05b] Rozza G. (2005) Reduced-basis methods for elliptic equations in sub-domains with *a posteriori* error bounds and adaptivity. *Appl. Numer. Math.* 55(4): 403–424.
- [Roz09] Rozza G. (2009) Reduced basis method for Stokes equations in domains with non-affine parametric dependence. *Comp. Vis. Science* 12: 23–35.
- [RP08] Rozza G. and Patera A. T. (2008) *Worked Problems with rbMIT Software: <http://augustine.mit.edu/workedProblems.htm>*. ©MIT, Cambridge, MA.
- [RR04] Renardy M. and Rogers R. C. (2004) *An Introduction to Partial Differential Equations*. Springer-Verlag, New York, II edition.
- [RST96] Ross H. G., Stynes M., and Tobiska L. (1996) *Numerical Methods for Singularly Perturbed Differential Equations. Convection-Diffusion and Flow Problems*. Springer-Verlag, Berlin Heidelberg.

- [Rud91] Rudin W. (1991) *Analyse Relle et Complexe*. Masson, Paris.
- [RV07] Rozza G. and Veroy K. (2007) On the stability of reduced basis method for Stokes equations in parametrized domains. *Comp. Meth. Appl. Mech. and Eng.* 196: 1244–1260.
- [Saa96] Saad Y. (1996) *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston.
- [Sag06] Sagaut P. (2006) *Large Eddy Simulation for Incompressible Flows: an Introduction*. Springer-Verlag, Berlin Heidelberg, III edition.
- [Sal08] Salsa S. (2008) *Partial Differential Equations in Action - From Modelling to Theory*. Springer, Milan.
- [Sch69] Schwarz H. (1869) ber einige abbildungsdufgaben. *J. Reine Agew. Math.* 70: 105–120.
- [Sch98] Schwab C. (1998) *p and hp- Finite Element Methods*. Oxford Science Publication, Oxford.
- [SF73] Strang G. and Fix G. J. (1973) *An Analysis of the Finite Element Method*. Wellesley-Cambridge Press, Wellesley, MA.
- [She] Shewchuk J. R. www.cs.cmu.edu/~quake/triangle.html.
- [Sir87] Sirovich L. (1987) Turbulence and the dynamics of coherent structures, part 1: Coherent structures. *Quarterly of Applied Mathematics* 45(3): 561–571.
- [Smo01] Smolianski A. (2001) *Numerical Modeling of Two-Fluid Interfacial Flows*. PhD thesis, University of Jyvskyl.
- [Spi99] Spivak M. (1999) *A comprehensive introduction to differential geometry. Vol. II*. Publish or Perish Inc., Houston, Tex., III edition.
- [Str71] Stroud A. H. (1971) *Approximate calculation of multiple integrals*. Prentice-Hall, Inc., Englewood Cliffs, N.J.
- [Str89] Strickwerda J. C. (1989) *Finite Difference Schemes and Partial Differential Equations*. Wadsworth & Brooks/Cole, Pacific Grove.
- [Str00] Strostrom B. (2000) *C++ Linguaggio, Libreria Standard, Principi di Programmazione*. Addison Wesley Longman Italia, Milano, III edition.
- [SV05] Saleri F. and Veneziani A. (2005) Pressure correction algebraic splitting methods for the incompressible Navier-Stokes equations. *SIAM J. Numer. Anal.* 43(1): 174–194.
- [SZ91] Sokolowski J. and Zolesio J. (1991) *Introduction to Shape Optimization (Shape Sensitivity Analysis)*. Springer-Verlag, New York.
- [Tan93] Tanaka N. (1993) Global existence of two phase nonhomogeneous viscous incompressible fluid flow. *Comm. Partial Differential Equations* 18(1-2): 41–81.
- [TE05] Trefethen L. and Embree M. (2005) *Spectra and pseudospectra. The behavior of nonnormal matrices and operators*. Princeton University Press, Princeton.
- [Tem01] Temam R. (2001) *Navier Stokes Equations*. North-Holland, Amsterdam.
- [TF88] Tsuchiya K. and Fan L.-S. (1988) Near-wake structure of a single gas bubble in a two-dimensional liquid-solid fluidized bed: vortex shedding and wake size variation. *Chem. Engrg. Sci.* 43(5): 1167–1181.
- [Tho84] Thomee V. (1984) *Galerkin Finite Element Methods for Parabolic Problems*. Springer, Berlin and Heidelberg.
- [TI97] Trefethen L. and III D. B. (1997) *Numerical Linear Algebra*. SIAM.
- [TL58] Taylor A. and Lay D. (1958) *Introduction to Functional Analysis*. J.Wiley & Sons, New York.
- [Tor99] Toro E. (1999) *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer-Verlag, Berlin.

- [Tri] software.sandia.gov/trilinos/.
- [TSW99] Thompson J. F., Soni B. K., and Weatherill N. P. (eds) (1999) *Handook of Grid Generation*. CRC Press, Boca Raton.
- [TW05] Toselli A. and Widlund O. (2005) *Domain Decomposition Methods - Algorithms and Theory*. Springer-Verlag, Berlin Heidelberg.
- [TWM85] Thompson J. F., Warsi Z. U. A., and Mastin C. W. (1985) *Numerical Grid Generation, Foundations and Applications*. North-Holland, New York.
- [UMF] www.cise.ufl.edu/research/sparse/umfpack/.
- [uRVS08] ur Rehman M., Vuik C., and Segal G. (2008) A comparison of preconditioners for incompressible navier-stokes solvers. *Int. J. Numer. Meth. Fluids* 57: 1731–1751.
- [uRVS09] ur Rehman M., Vuik C., and Segal G. (2009) Preconditioners for the steady incompressible navier-stokes problem. *Int. J. Applied Math.* 38 (4).
- [Vas81] Vasiliev F. (1981) *Methods for Solving the Extremum Problems*. Nauka, Moscow.
- [vdV03] van der Vorst H. A. (2003) *Iterative Krylov Methods for Large Linear Systems*. Cambridge University Press, Cambridge.
- [Vel95] Veldhuizen T. (1995) Expression templates. *C++ Report Magazine* 7(5): 26–31. see also the web page <http://osl.iu.edu/tveldhui>.
- [Ven98] Veneziani A. (1998) *Mathematical and Numerical Modeling of Blood Flow Problems*. PhD thesis, Università degli Studi di Milano.
- [Ver84] Verfürth R. (1984) Error estimates for a mixed finite elements approximation of the stokes equations. *R.A.I.R.O. Anal. Numér.* 18: 175–182.
- [Ver96] Verfürth R. (1996) *A Review of a Posteriori Error Estimation and Adaptive Mesh Refinement Techniques*. Wiley-Teubner, New York.
- [VM96] Versteeg H. and Malalasekra W. (1996) *An Introduction to Computational Fluid Dynamics: the Finite Volume Method Approach*. Prentice-Hall.
- [VP05] Veroy K. and Patera A. T. (2005) Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations; Rigorous reduced-basis *a posteriori* error bounds. *International Journal for Numerical Methods in Fluids* 47: 773–788.
- [VPP03] Veroy K., Prud'homme C., and Patera A. T. (2003) Reduced-basis approximation of the viscous Burgers equation: Rigorous *a posteriori* error bounds. *C. R. Acad. Sci. Paris, Série I* 337(9): 619–624.
- [VPRP03] Veroy K., Prud'homme C., Rovas D. V., and Patera A. T. (2003) *A Posteriori* error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*. Paper 2003-3847.
- [Wes01] Wesseling P. (2001) *Principles of Computational Fluid Dynamics*. Springer-Verlag, Berlin Heidelberg New York.
- [Wil98] Wilcox D. C. (1998) *Turbulence Modeling in CFD*. DCW Industries, La Cañada, CA, II edition.
- [Win07] Winkelmann C. (2007) *Interior penalty finite element approximation of Navier-Stokes equations and application to free surface flows*. PhD thesis, École Polytechnique Fédérale de Lausanne.
- [WP02] Willcox K. and Peraire J. (2002) Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal* 40(11): 2323–2330.
- [Wya00] Wyatt D. C. (2000) Development and assessment of a nonlinear wave prediction methodology for surface vessels. *Journal of ship research* 44: 96.
- [Yos74] Yosida K. (1974) *Functional Analysis*. Springer-Verlag, Berlin Heidelberg.
- [Zie00] Zienkiewicz O. (2000) Achievements and some unsolved problems of the finite element method. *Int. J. Numer. Meth. Eng.* 47: 9–28.

- [ZT00] Zienkiewicz O. C. and Taylor R. L. (2000) *The Finite Element Method, Vol. 1, The Basis*. Butterworth-Heinemann, Oxford, V edition.
- [Zua03] Zuazua E. (2003) Propagation, observation, control and numerical approximation of waves. *Bol. Soc. Esp. Mat. Apl.* 25: 55–126.
- [Zua05] Zuazua E. (2005) Propagation, observation, and control of waves approximated by finite difference methods. *SIAM Review* 47 (2): 197–243.
- [Zua06] Zuazua E. (2006) Controllability and observability of partial differential equations: Some results and open problems. In Dafermos C. and Feireisl E. (eds) *Handbook of Differential Equations: Evolutionary Differential Equations*, volume 3, pages 527–621. Elsevier Science.
- [ZZ92] Zienkiewicz O. C. and Zhu J. Z. (1992) The superconvergent patch recovery and a posteriori error estimates. I: The recovery technique. *Int. J. Numer. Meth. Engng.* 33: 1331–1364.

Index

- adaptivity
 - a posteriori, 103, 496
 - a priori, 100
 - goal-oriented, 307
 - of type h , 99
 - of type p , 70, 100
- adjoint
 - operator, *see* operator adjoint
 - problem, *see* problem adjoint
 - state, 460, 466, 467
- affine geometry, 551
- affine mapping, 552, 554
 - piecewise, 556
- algorithm
 - diagonal exchange, 154
 - Laplacian regularization, 155
 - Lawson, 155
 - steepest descent, 480
 - Thomas, 161
- analysis
 - backward, 342
 - Von Neumann, 330, 349
- Arnoldi algorithm, 168
- assembly, 175
 - element-oriented, 187
 - node-oriented, 187
- Aubin-Nitsche trick, 98, 110
- baricentration, *see* Laplacian regularization algorithm
- boundary conditions
 - Dirichlet, 405
 - dual, 54, 56
 - essential, 46
 - free slip, 443
 - natural, 46
 - Neumann, 405, 433
 - non-slip, 443
 - primal, 54
 - Robin, 40
 - Robin-Dirichlet, 52
 - weak treatment, 226
- boundary layers, 271
- boundary observation, 471, 472
- breakdown, 168, 170
- Cauchy-Schwarz inequality, 25
- CFL
 - condition, 326, 327, 389
 - number, 326, 335, 336
- characteristic
 - function, 18
 - Lagrangian functions, 67, 80
 - lines, 7, 314, 318
 - of the Burgers equation, 383
 - rate, 388
 - variables, 317
- coefficient
 - amplification, 331, 349
 - dispersion, 349
 - dissipation, 335
- computational cost, 5
- condition
 - CFL, 326, 327, 389
 - entropy, 385
 - incompressibility, 402

- inf-sup, 410, 411, 422
- Lax admissibility, 400
- Rankine-Hugoniot, 384, 400
- conservation law, 217, 383, 398
 - of the entropy, 386
- consistence, 4, 324
- strong, 4, 64, 294, 296
- control
 - boundary, 461, 470
 - distributed, 460, 464, 468, 469
 - function, 481
 - optimal, 457, 459, 462
 - volume, 218
- controllability, 489
- convergence, 4, 64, 66, 126, 324
 - rate, 5
- coordinates
 - barycentric, 78, 222
- degree
 - of a vector, 168
- degrees of freedom, 67, 68
 - finite element, 77
- derivative
 - conormal, 48, 49
 - Fréchet, 14, 462
 - Gâteaux, 14, 478, 479
 - in the sense of distributions, 18
 - interpolation, 243, 245, 260, 372, 374
 - Lagrangian, 442
 - material, 430
 - normal, 32
- diffusion
 - artificial, 285
 - numerical, *see* artificial diffusion
- Dirac delta, 16, 33
- dispersion, 335, 342
- dissipation, 335, 342
- distributed observation, 471, 472
- distributions, 15
- domain, 23
- domain of dependence, 318
 - numerical, 326
- Donald
 - diagram, 222
- element
 - finite, 76
 - diameter, 81, 142
 - Lagrangian, 77
 - sphericity, 143
- reference, 77
- elliptic regularity, 96, 98
- entropy, 386
 - flux, 386
- equation
 - adjoint, 466, 479, 481
 - Burgers, 2, 383–385, 394, 396
 - diffusion-transport, 5, 432
 - discriminant, 5
 - elliptic, 5
 - Euler, 462
 - heat, 2, 3, 5, 120
 - homogeneous, 1
 - hyperbolic, 5
 - nonlinear, 388
 - Kortevég-de-Vries, 10
 - observation, 464
 - parabolic, 5
 - Plateau's, 9
 - Poisson, 31
 - potential, 2, 5
 - quasi-linear, 1, 399
 - semi-linear, 1
 - state, 397, 458, 464, 481
 - transport, 2
 - transport-reaction, 333
 - transportation, 3
 - viscosity, 387
 - wave, 2, 5, 9, 319
- equations
 - compatibility, 379
 - equivalent, 337
 - Euler, 397, 400, 406
 - in conservative form, 398
 - Euler-Lagrange, 491
 - Navier-Stokes, 401, 450, 576
 - for compressible fluids, 397
 - primitive variables, 403
 - reduced form, 406
 - weak formulation, 405
 - Stokes, 407
 - strictly hyperbolic, 399

- error
 - a posteriori estimate, 103, 106, 110, 112, 495
 - goal-oriented, 112
 - recovery-based, 112
 - residual-based, 106
 - a priori estimate, 75, 95, 96, 100, 128, 134, 250, 301, 368, 418
 - amplification, 335, 336, 354
 - approximation, 65
 - discretization, 495
 - dispersion, 335, 336, 354
 - interpolation
 - estimate, 92, 236
 - iteration, 495
 - truncation, 294, 325
- exactness degree, 179, 234, 235, 237
- factorization
 - Cholesky, 123, 161
 - LU, 160
 - tridiagonal, 161
- Fick law, 398
- field of values, 171
- finite differences, 244, 280, 320
- finite elements, 531
 - \mathbb{P}_1 - $\text{iso}\mathbb{P}_2$, 423
 - compatible, 421
 - Crouzeix-Raviart, 422
 - discontinuous, 362, 367, 389
 - hierarchical, 70
 - implementation, 173
 - isoparametric, 194
 - Lagrangian, 67
 - linear, 67, 71
 - mini-element, 423
 - quadratic, 68
 - sphericity, 81
 - stabilized, 289
- flow
 - turbulent, 406
- flux
 - limiters, 395
 - numerical, 224, 226, 321, 323, 324, 388, 390
 - Engquist-Osher, 390
 - Godunov, 390
 - Lax-Friedrichs, 390
 - monotone, 390
 - upwind, 391
- thermal, 397
- viscous, 399
- form, 12
 - bilinear, 12
 - affine, 549, 559
 - eigenfunction, 131
 - eigenvalue, 131
 - parametric, 549, 557
 - coercive, 13
 - continuous, 12
 - positive, 13
 - quadratic, 8
 - definite, 8
 - degenerate, 8
 - indefinite, 8
 - quasi-linear, 399
 - symmetric, 12
 - weakly coercive, 120
- formula
 - Green, *see* Green formula
- formulae
 - Armijo, 484
- formulation
 - conservative, 53
 - non-conservative, 53
 - strong, 31
 - weak, 33
- free surface, 441, 452
- function
 - basis, 67, 70
 - Bernoulli, 285
 - bubble, 70, 292, 304
 - characteristic, 20
 - compact support, 15
 - control, 458
 - Heaviside, 19
 - observation, 458, 461
 - spacing, 102, 139, 151
- functional, 11
 - bounded, 11
 - cost, 458, 464, 478
 - Lagrangian, 409
 - linear, 11
 - norm, 11
- Galerkin orthogonality, 64, 103, 110, 112
- Gaussian integration
 - Gauss-Legendre, 234
 - exactness, 235
 - Gauss-Legendre-Lobatto, 235, 247

- Gordon-Hall transformation, 228
- Green formula, 41, 42, 59, 403
- grid, *see also* triangulation
 - anisotropic, 81, 95
 - blockwise structured, 146
 - derefinement, 102
 - non-structured, 147
 - quasi uniform, 86
 - refinement, 101
 - regular, 81
 - structured, 143, 144
- hierarchical basis, 70
- hyperbolic system, 317, 320, 323
- inequality
 - Cauchy-Schwarz, 25, 36, 42, 253
 - strengthened, 540
 - Hölder, 25, 404
 - inverse, 86, 133, 241, 288, 298
 - Korn, 59
 - Poincaré, 22, 272, 287
 - Young, 124
- input-parameter, 547
- integral
 - general, 3
 - particular, 3
- interpolation, 73, 88
 - error estimates, 74, 93
 - operator, 73, 88
 - transfinite, 228
- iterative algorithms
 - generalized minimum residual (GMRES)
 - flexible, 172
- Krylov method, *see* method Krylov
- Krylov subspace, 167
- Lagrange
 - identity, 52, 55
 - multiplier, 477, 478
- Lagrangian
 - basis, 67, 69
 - finite elements, 77
 - functional, 478
 - stabilized, 494
- Legendre series, 232
- lemma
 - Bramble-Hilbert, 91
 - Céa, 64
 - Deny-Lions, 91
 - Gronwall, 28, 128, 317, 357
 - discrete, 28
 - Lax-Milgram, 50, 59, 61, 63
 - Strang, 250, 253, 290, 425
- many-query calculation, 547
- mass-lumping, 122, 281, 310, 349
- matrix
 - graph, 182
 - interpolation derivative, 245
 - iteration, 162
 - mass, 121
 - preconditioning, 88, 162
 - reflection, 380
 - sparse, MSR format, 182
 - sparsity pattern, 182
 - stiffness, 62, 83, 121
 - conditioning, 86
 - symmetric positive definite, 164
- method
 - algebraic factorization, 435
 - BDF, 204
 - BiCGSTAB, 167
 - characteristics, 314, 430
 - Chorin-Temam, 434, 439
 - collocation, 242
 - conjugate gradient
 - convergence, 166
 - consistent, 4, 294, 325
 - convergent, 4, 325
 - Crank-Nicolson, 122
 - Dirichlet-Neumann, 504, 518, 519, 541
 - Discontinuous Galerkin (DG), 362, 367, 389
 - jump stabilization, 369
 - SEM-NI, 374
 - domain decomposition, 191, 452, 501
 - Douglas-Wang (DW), 296, 297
 - empirical interpolation, 577
 - Euler
 - backward, 122, 346, 359, 366, 429
 - backward/centered, 323
 - explicit, *see* forward Euler method
 - forward, 122, 345, 350, 360, 428
 - forward/centered, 321
 - forward/decentered, *see* upwind method
 - implicit, *see* backward Euler method

- exponential fitting, *see* Scharfetter and Gummel method
 - (FV), 225
- finite volumes, 217
 - a priori estimate, 226
 - cell-centered, 219
 - ghost nodes, 226
 - staggered-grids, 219, 451
 - vertex-centered, 219
- FOM, 169
- fractional step, 431
- front capturing, 441
- front tracking, 441
- G-NI, 237, 238, 242, 245, 250, 257, 288, 370
- Galerkin, 61, 63, 273, 276, 410, 564
 - convergence, 66
 - generalized, 239, 250, 289
 - Least Squares (GLS), 296–298, 424, 494
 - projection, 564
 - spectral, 227, 238
- GEM, 159
- GMRES, 169, 170
 - convergence, 170
 - flexible, 172
 - with restart, 170
- gradient, 165, 483
 - conjugate, 165, 166, 483
- Lax-Friedrichs, 322, 328, 342, 389
 - (FEM), 350
- Lax-Wendroff, 322, 324, 328, 342
 - (FEM), 350
- leap-frog, 323
- level set, 452
- Neumann-Dirichlet, 519
- Neumann-Neumann, 506, 511, 519
- Newmark, 323
- Newton, 483
 - numerical
 - bounded, 388
 - monotone, 388, 390
 - operator splitting, 431
 - Yanenko, 432
 - Petrov-Galerkin, 289, 292
 - POD (proper order decomposition), 563, 576
 - projection, 433, 434
 - quasi-Newton, 483
- reduced basis (RB), 547
 - a posteriori error, 570, 572
 - coercivity constant, 550, 575
 - continuity constant, 549
 - convergence, 567, 568
 - greedy Lagrange space, 566
 - hierarchical space, 563
 - non-hierarchical space, 567
 - residual, 573, 574
 - sampling, 566
 - snapshot, 563, 567
 - space, 562, 563, 571, 576
- relaxation, 505
- Richardson, 163, 167, 510, 518
- Robin-Robin, 506, 511, 519
- Runge-Kutta
 - 2nd order, 392
 - 3rd order, 393
- Scharfetter and Gummel, 285, 292
- Schwarz, 502, 531, 536
 - additive, 502, 532
 - multiplicative, 502, 532
- SEM, 228, 230
- SEM-NI, 261
- semi-implicit, 430
- shock capturing, 389
- spectral element, 228
- streamline-diffusion, 292
- Streamline-Upwind Petrov-Galerkin (SUPG), 296, 297, 424
- Taylor-Galerkin, 350
- upwind, 284, 322, 323, 328, 369
 - (FEM), 350, 369
 - FV, 225
- volume of fluid, 452
- Yosida, 439
- modal
- basis, 266
 - boundary-adapted, 266, 267
- representation, 232
- nodes, 67, 69, 81, 82
- norm
 - A, 87
 - discrete, 253
 - energy, 65, 75, 88
- number
 - CFL, 326, 335, 336, 350
 - condition, 85, 515, 517, 521, 525, 538

- Péclet
 - global, 274, 278
 - grid, *see* local Péclet number
 - local, 225, 277, 280, 285, 286
- Reynolds, 406
- observability, 489
- offline-online computational procedure, 565, 574
- operator
 - adjoint, 26, 51, 55, 466
 - bilaplacian, 57
 - dual, *see* adjoint, 54
 - extension, 523
 - interpolation, 88, 242
 - Clément, 105
 - Laplace, 2
 - Laplace-Beltrami, 470
 - lifting, 39, 46, 84
 - normal, 52
 - primal, 54
 - pseudo-spectral, 244
 - restriction, 523
 - self-adjoint, 52
 - skew-symmetric, 295
 - Steklov-Poincaré, 507, 508, 516
 - symmetric, 295
 - trace, 23
 - transpose, 26
- optimality system, 467, 468, 479, 480
- output, 547
 - compliant, 549
- parameter
 - geometrical, 560
- Parseval identity, 232
- partition of unity, 79, 282
- PDE
 - coefficient
 - discontinuous, 559
 - phase angle, 335, 350
 - point
 - constrained critical, 476
 - regular, 476
 - polygon, 23
 - polyhedron, 23
 - polynomials
 - Jacobi, 266
 - Legendre, 231, 234, 365
 - preconditioner, 88, 162
 - additive Schwarz, 533
 - augmented Lagrangian, 437
 - Bramble-Pasciak-Schatz, 526
 - Jacobi, 525
 - MSIMPLER, 438
 - Neumann-Neumann, 526, 527
 - balanced, 529
 - SIMPLE, 438
 - SIMPLER, 438
 - principal symbol, 8
 - principle
 - discrete maximum, 328
 - virtual work, 38
 - problem
 - adjoint, 51, 96, 97, 110
 - advection, 541
 - collocation, 243
 - control, 457
 - constrained, 458
 - discretized, 490
 - unconstrained, 458
 - controllable, 457
 - diffusion-reaction, 207, 255, 278
 - diffusion-transport, 271, 306, 311
 - diffusion-transport-reaction, 137, 223, 311
 - Dirichlet, 31, 41, 84, 468
 - dual, *see* adjoint
 - elliptic, 61
 - fourth-order, 58, 138
 - free surface, 440
 - Galerkin, *see* Galerkin method
 - generalized eigenvalue, 131
 - generalized Stokes, 407
 - heat, 269
 - heterogeneous, 501
 - linear elasticity, 58
 - mixed, 40
 - Neumann, 32, 39, 469
 - optimal design, 492
 - optimization, 458
 - Poisson, 63, 82, 512
 - Robin, 40
 - Stokes, 541
 - algebraic formulation, 420
 - Galerkin approximation, 410
 - stabilized finite elements, 424
 - stable finite elements, 421
 - weak form, 408

- transport, 313
- transport-reaction, 315, 356
- variational, 37, 43
- product
 - discrete scalar, 238
 - warped tensor, 265
- programming
 - object-oriented, 176
- quadrature formula
 - composite midpoint, 181
 - composite trapezoid, 181
- random walk, 271
- real-time calculation, 547
- reconstruction
 - of the gradient, 101
- reduced basis method, *see* method, reduced basis
- reference domain, 552, 556, 558
- residue, 162
 - local, 106
 - preconditioned, 163
- Riemann invariants, 319
- rod
 - thin, 114
 - vibrating, 9
- saddle-point, 409
- scheme, *see* method
- Schur complement, 515, 522
- semi-discretization, 121
- seminorm, 22
- shape optimization, 490
- simplex, 78
- solution
 - classical, 384
 - entropic, 384, 388
 - of the Burgers equation, 385
 - weak, 315, 384
- space
 - dual, 11
 - of distributions, 16
 - Sobolev, 36
- splitting, 162
- spurious modes, *see* Stokes problem, spurious solutions
- stability, 5, 64, 325, 388
 - absolute, 131, 246
 - strong, 326, 328, 329, 331, 333, 360, 389, 394
- subspace
 - Krylov, 167
- support
 - compact, 15
 - of a function, 15, 84
- term
 - diffusion, 401
 - transport, 401
- theorem
 - closed range, 414
 - divergence, 41
 - equivalence, 5, 327
 - extension, 516
 - Helmholtz-Weyl, 434
 - Lax-Richtmyer, *see* equivalence theorem
 - Riesz, 12, 50
 - trace, 23
- θ -method, 122, 130
- triangulation, 142
 - advancing front, 151
 - conforming, 142
 - Delaunay, 147, 221
 - generalized, 150
 - median dual, 222
 - reduced basis, 560
 - regular, 143
- turbulence models, 406
 - RANS, 452
- unisolvent set, 77
- variational inequality, 462
- vertices, 67
- viscosity
 - artificial, 290, 292, 294
 - numerical, *see* artificial viscosity
 - subgrid, 306
- Voronoi
 - diagram, 220
 - polygon, 220
 - tessellation, 220
 - vertices, 221
- Zienkiewicz-Zhu estimator, 111