

Numerical Methods for Partial Differential Equations

Prof. Ralf Hiptmair, Prof. Christoph Schwab,
Prof. H. Harbrecht, Dr. V. Gradinaru, and Dr. A. Chernov
Draft version May 21, 2010, SVN rev. 26776

(C) Seminar für Angewandte Mathematik, ETH Zürich

URL: <http://www.sam.math.ethz.ch/~hiptmair/tmp/NPDE10.pdf>

0.0

p. 1

Contents

1 Case Study: A Two-point Boundary Value Problem	19
1.1 Introduction	19
1.2 A model problem	22
1.2.1 Linear elastic string	22
1.2.2 Mass-spring model	28
1.2.3 Continuum limit	33
1.3 Variational approach	41
1.3.1 Virtual work equation	41
1.3.2 Regularity requirements	49
1.3.3 Differential equation	53
1.4 Simplified model	58
1.5 Discretization	65
	0.0
	p. 2

1.5.1	Galerkin discretization	69
1.5.1.1	Spectral Galerkin scheme	78
1.5.1.2	Linear finite elements	96
1.5.2	Collocation	114
1.5.2.1	Spectral collocation	117
1.5.2.2	Spline collocation	123
1.5.3	Finite differences	126
1.6	Convergence	130
1.6.1	Norms on function spaces	132
1.6.2	Algebraic and exponential convergence	139
2	Second-order Scalar Elliptic Boundary Value Problems	154
2.1	Equilibrium models	158
2.1.1	Taut membrane	158
2.1.2	Electrostatic fields	165
2.1.3	Quadratic minimization problems	170
2.2	Sobolev spaces	181
2.3	Variational formulations	200
2.3.1	Linear variational problems	200
2.3.2	Stability	206
2.4	Equilibrium models: Boundary value problems	215
2.5	Diffusion models (Stationary heat conduction)	227
2.6	Boundary conditions	233
2.7	Characteristics of elliptic boundary value problems	238
2.8	Second-order elliptic variational problems	242
2.9	Essential and natural boundary conditions	250

3 Finite Element Methods (FEM)	259
3.1 Galerkin discretization	262
3.2 Case study: Triangular linear FEM in two dimensions	272
3.2.1 Triangulations	274
3.2.2 Linear finite element space	276
3.2.3 Nodal basis functions	279
3.2.4 Sparse Galerkin matrix	284
3.2.5 Computation of Galerkin matrix	290
3.2.6 Computation of right hand side vector	300
3.3 Building blocks of general FEM	305
3.3.1 Meshes	306
3.3.2 Polynomials	309
3.3.3 Basis functions	313
3.4 Lagrangian FEM	318
3.4.1 Simplicial Lagrangian FEM	319
3.4.2 Tensor-product Lagrangian FEM	326
3.5 Implementation of FEM	336
3.5.1 Mesh file format	337
3.5.2 Mesh data structures [1, Sect. 1.1]	343
3.5.3 Assembly [1, Sect. 5]	349
3.5.4 Local computations and quadrature	362
3.5.5 Incorporation of essential boundary conditions	376
3.6 Parametric finite elements	384
3.6.1 Affine equivalence	384
3.6.2 Example: Quadrilateral Lagrangian finite elements	393
3.6.3 Transformation techniques	401
3.6.4 Boundary approximation	407
3.7 Linearization	409

0.0
p. 4

4 Finite Differences (FD) and Finite Volume Methods (FV)	419
4.1 Finite differences	421
4.2 Finite volume methods (FVM)	432
4.2.1 Gist of FVM	432
4.2.2 Dual meshes	435
4.2.3 Relationship of finite elements and finite volume methods	439
5 Convergence and Accuracy	448
5.1 Galerkin error estimates	449
5.2 Empirical Convergence of FEM	459
5.3 Finite element error estimates	475
5.3.1 Estimates for linear interpolation in 1D	477
5.3.2 Error estimates for linear interpolation in 2D	485
5.3.3 The Sobolev scales	500
5.3.4 Anisotropic interpolation error estimates	504
5.3.5 General approximation error estimates	514
5.4 Elliptic regularity theory	527
5.5 Variational crimes	539
5.5.1 Impact of numerical quadrature	540
5.5.2 Approximation of boundary	543
5.6 Duality techniques	547
5.6.1 Linear output functionals	547
5.6.2 Case study: Boundary flux computation	557
5.6.3 L^2 -estimates	568
5.7 Discrete maximum principle	578
	0.0
	p. 5

6 2nd-Order Linear Evolution Problems	593	
6.1 Parabolic initial-boundary value problems	596	
6.1.1 Heat equation	596	
6.1.2 Spatial variational formulation	600	
6.1.3 Method of lines	608	
6.1.4 Timestepping	612	
6.1.4.1 Single step methods	613	
6.1.4.2 Stability	617	
6.1.5 Convergence	640	
6.2 Wave equations	650	
6.2.1 Vibrating membrane	651	
6.2.2 Wave propagation	659	
6.2.3 Method of lines	666	
6.2.4 Timestepping	669	
6.2.5 CFL-condition	681	
7 Convection-Diffusion Problems	692	
7.1 Heat conduction in a fluid	692	
7.1.1 Modelling fluid flow	693	
7.1.2 Heat convection and diffusion	697	
7.1.3 Incompressible fluids	700	0.0
7.1.4 Transient heat conduction	705	p. 6

7.2	Stationary convection-diffusion problems	707
7.2.1	Singular perturbation	710
7.2.2	Upwinding	716
7.2.2.1	Upwind quadrature	727
7.2.2.2	Streamline diffusion	735
7.3	Transient convection-diffusion BVP	744
7.3.1	Method of lines	745
7.3.2	Transport equation	751
7.3.3	Lagrangian split-step method	755
7.3.3.1	Split-step timestepping	756
7.3.3.2	Particle method for advection	762
7.3.3.3	Particle mesh method	770
7.3.4	Semi-Lagrangian method	780
8	Numerical Methods for Conservation Laws	787
8.1	Conservation laws: Examples	788
8.1.1	Linear advection	789
8.1.2	Inviscid gas flow	793
8.2	Scalar conservation laws in 1D	799
8.2.1	Integral and differential form	799
8.2.2	Characteristics	805
8.2.3	Weak solutions	811
		0.0
		p. 7

8.2.4	Jump conditions	815
8.2.5	Riemann problem	818
8.2.6	Entropy condition	827
8.2.7	Properties of entropy solutions	832
8.3	Conservative finite volume discretization	836
8.3.1	Semi-discrete conservation form	839
8.3.2	Discrete conservation property	842
8.3.3	Numerical flux functions	846
8.3.3.1	Central flux	846
8.3.3.2	Lax-Friedrichs flux	851
8.3.3.3	Upwind flux	855
8.3.3.4	Godunov flux	861
8.3.4	Monotone schemes	870
8.4	Timestepping	881
8.4.1	CFL-condition	885
8.4.2	Linear stability	891
8.4.3	Convergence	905
8.5	Higher order conservative schemes	914
Index		916
Keywords	916	
Examples	927	
Definitions	932	
MATLAB-CODE	934	0.0
Symbols	935	p. 8

A Essential skills	941
A.1 Chapter ??: Prologue: A Two-point Boundary Value Problem	941
A.2 Chapter ??: Second-order scalar elliptic boundary value problems	942
A.3 Chapter ??: The Finite Element Method (FEM)	943
A.4 Chapter ??: Special elliptic boundary value problems	946
A.5 Chapter ??: Solving discrete boundary value problems	947
A.6 Chapter ??: Parabolic Boundary Value Problems	948
A.7 Chapter ??: Numerical Methods for Conservation Laws	948
A.8 Chapter ??: Adaptive Finite Element Schemes	949

Course history

- Summer semester 04, R. Hiptmair (for RW/CSE undergraduates)
- Winter semester 04/05, C. Schwab (for RW/CSE undergraduates)
- Winter semester 05/06, H. Harbrecht (for RW/CSE undergraduates)
- Winter semester 06/07, C. Schwab (for BSc RW/CSE)
- Autumn semester 07, A. Chernov (for BSc RW/CSE)
- Autumn semester 08, C. Schwab (for BSc RW/CSE)
- Autumn semester 09, V. Gradinaru (for BSc RW/CSE, Subversion Revision: 22844)
- Spring semester 10, R. Hiptmair (for BSc Computer Science, Subversion Revision: 0.0)
- Autumn semester 10, R. Hiptmair (for BSc RW/CSE) p. 9

Preamble

This lecture is a core course for

- BSc in Computational Science and Engineering (RW/CSE),
- BSC in Computer Science with focus Computational Science.

Main *skills* to be acquired in this course:

- Ability to *implement* advanced numerical methods for the solution of partial differential equations in MATLAB efficiently
- Ability to *modify* and *adapt* numerical algorithms guided by awareness of their mathematical foundations
- Ability to *select* and *assess* numerical methods in light of the predictions of theory
- Ability to *identify features* of a PDE (= partial differential equation) based model that are relevant for the selection and performance of a numerical algorithm

- Ability to *understand research publications* on theoretical and practical aspects of numerical methods for partial differential equations.

This course \neq Numerical analysis of PDE (\rightarrow mathematics curriculum)
(401-3651-00V *Numerical methods for elliptic and parabolic partial differential equations*, R. Hiptmair, Tue 15-17 HG E 5, Thu 13-15 HG E 5)

Instruction on how to apply software packages

Reading instructions

This course materials are neither a textbook nor lecture notes.
They are meant to be supplemented by explanations given in class.

Some pieces of advice:

- this document is not meant for mere reading, but for working with,

0.0

p. 11

- turn pages all the time and follow the numerous cross-references,
- study the relevant section of the course material when doing homework problems.

Practical information

Course: 401-0674-00L Numerical Methods for Partial Differential Equations

Lectures: Wed 8-10 HG E 3
Fri 10-12 HG E 5

Tutorials: Thu 13-15 HG D 7.2
Fri 15-17 HG E 21
Tue 13-15 HG E 21 (extra slot)

Lecturer: **Prof. Ralf Hiptmair**, office: HG G 58.2, e-mail: hiptmair@sam.math.ethz.ch

Assistants: **Eivind Fonn**, office: HG J 59, e-mail: eivind.fonn@sam.math.ethz.ch
(1st year PhD student at Seminar of Applied Mathematics, D-MATH)

Dr. Serban Georgescu, office: CAB F 81, e-mail: serban.georgescu@inf.ethz.ch
(Postdoctoral researcher at Chair of Computational Science, D-INFK)

Konstantinos Ritos, office: CAB F 84, e-mail: konstantinos.ritos@inf.ethz.ch
(1st year PhD student at Chair of Computational Science, D-INFK)

Rajdeep Deb, e-mail: debr@student.ethz.ch (MSc student RW/CSE)

Roman Fuchs, e-mail: rofuchs@student.ethz.ch (MSc student RW/CSE)

Assignments:

- 12 weekly assignment sheets, handed out on Thursday, discussed in tutorial classes in the following week
- “Testat” requirement: regular attempts to solve homework problems and programming exercises and attendance of at least 9 tutorial classes
- MATLAB programming exercises

Examination: “Sessionsprüfung”: computer based examination, programming & theoretical tasks

Date: Friday, August 13, 2010

Lecturer's questions for course evaluation

Course number (LV-ID): 401-0674-00L

Date of evaluation: April 23, 2010

D1: Theoretical and algorithmic aspects are well balanced in the course.

D2: (Numerical) examples in class provide useful insights and motivation.

D3: My prior knowledge in analysis was adequate for the course.

D4: The programming exercises help understand the numerical methods.

D5: The MATLAB finite element library is easy to use.

D6: The model solutions for exercise problems offer sufficient guidance.

Scoring: 6: I agree fully

5: I agree to a large extent

4: I agree partly

3: I do not quite agree

2: I disagree

1: I disagree strongly

0.0

p. 14

Evaluation of assistants:

Assistant	shortcut
Eivind Fonn	EIV
Dr. Serban Georgescu	SER
Konstantinos Ritos (Costas)	COS

Please enter the shortcut code after the LV-ID in the three separate boxes.

Evaluation results: coming soon

Reporting errors

Please report errors in the electronic lecture notes via a [wiki page](#) !

<http://elbanet.ethz.ch/wikifarm/rhiptmair/index.php?n>Main.NPDECourse>

(Password: NPDE, please choose [EDIT](#) menu to enter information)

Please supply the following information:

- (sub)section where the error has been found,
- precise location (e.g, after Equation (4), Thm. 2.3.3, etc.). Refrain from giving page numbers,
- brief description of the error.

Online discussion forum

Contribute to the forum

Numerical Methods for Partial Differential Equations

to be found at the URL

<http://forum.vis.ethz.ch/forumdisplay.php?f=79>

This forum has been set up so that you can post questions on the programming exercises that accompany the course. One of the assistants will look at the entries in this forum and

- write an answer in this forum or
- discuss the question in a consulting session and post an answer later.

A second purpose of this forum is that the assistants can collect FAQs and post answers here.

Main topics

- Second order elliptic boundary value problems
- The finite element method (FEM)
- Parabolic boundary value problems
- Special elliptic boundary value problems

- Numerical methods for conservation laws
- Adaptive finite element schemes
- Multilevel iterative solvers

Case Study: A Two-point Boundary Value Problem

1.1 Introduction

The term “partial differential equation” (PDE) usually conjures up formulas like

$$\operatorname{div}(\sqrt{1 + \|\operatorname{grad} u(\mathbf{x})\|^2} \operatorname{grad} u(\mathbf{x})) + \mathbf{v} \cdot \operatorname{grad} u = f(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d.$$

This chapter aims to rid you from this impulse and instil an appreciation that

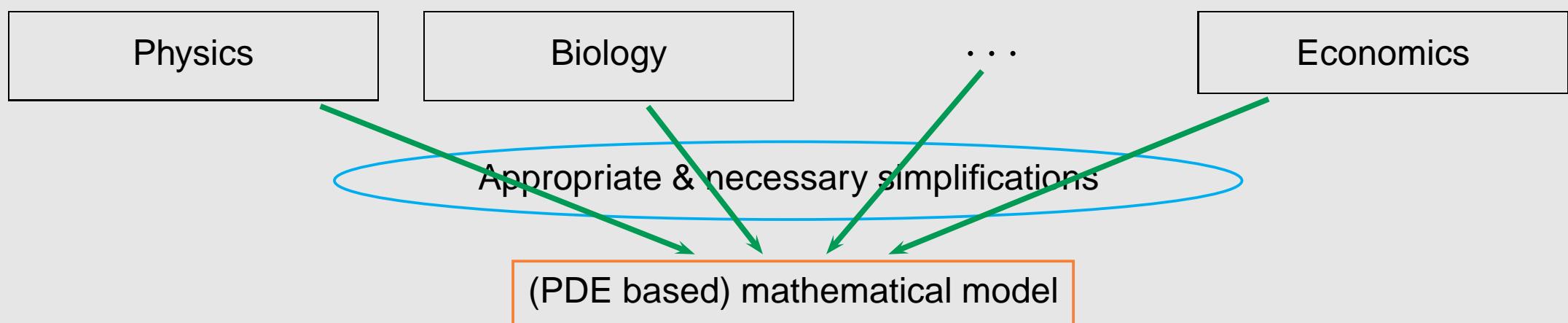
a meaningful PDE encodes structural principles
(like equilibrium, conservation, etc.)

!

The design and selection of numerical methods has to take into account these governing principles.

Remark 1.1.1 (Mathematical modelling).

Prerequisite for numerical simulation: **Mathematical modelling**



Necessary simplification:

$\left\{ \begin{array}{l} \text{system} \\ \text{phenomenon} \end{array} \right\}$ described by *a few* variables/functions in **configuration space**

The art of modelling: devise “faithful model”

Essential/relevant **traits** of $\left\{ \begin{array}{l} \text{system} \\ \text{phenomenon} \end{array} \right\}$ \longrightarrow **structural properties** of model



Remark 1.1.2 (“PDEs” for univariate functions).

The classical concept of a PDE inherently involves functions of several independent variables. However, when one embraces the concept of a PDE as encoding fundamental structural properties of a model, then simple representatives in a univariate setting can be discussed.

☞ ordinary differential equations (ODEs) offer simple specimens of important classes of PDEs!

Thus, in this chapter we examine ODEs that are related to the important class of **elliptic PDEs**.



functional analytic framework

1.2 A model problem

1.2.1 Linear elastic string

Static mechanical problem:

Deformation of elastic “1D” string (rubber band) under its own weight

Constraint: string pinned at endpoints

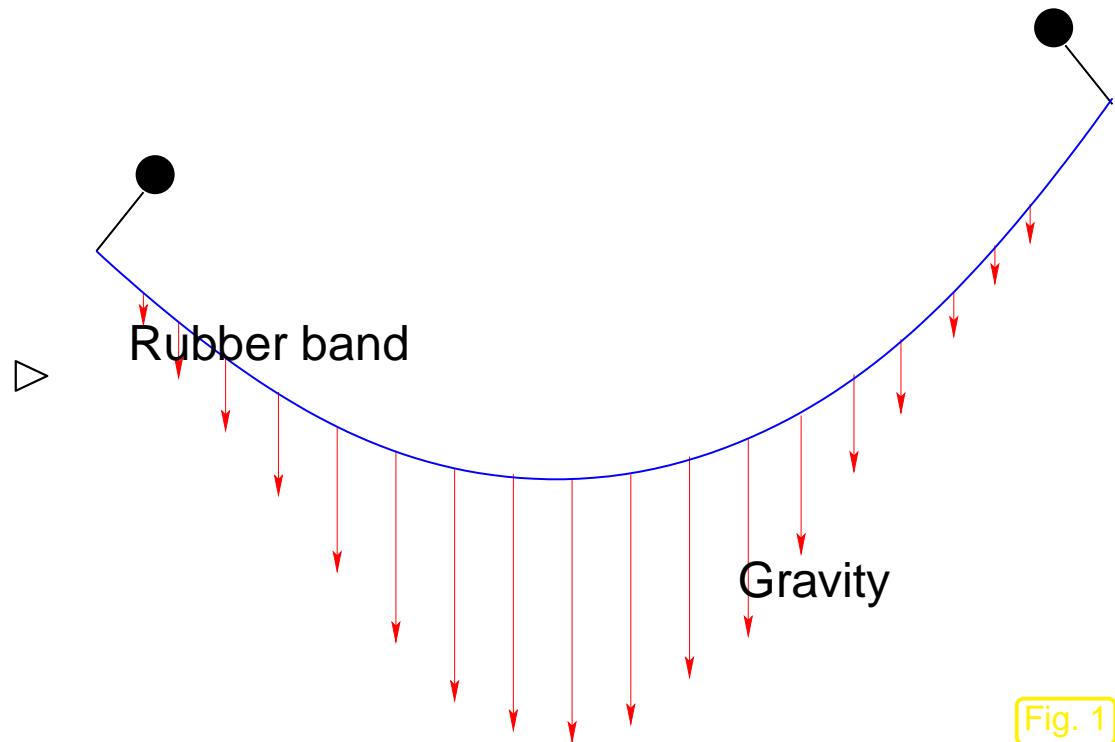


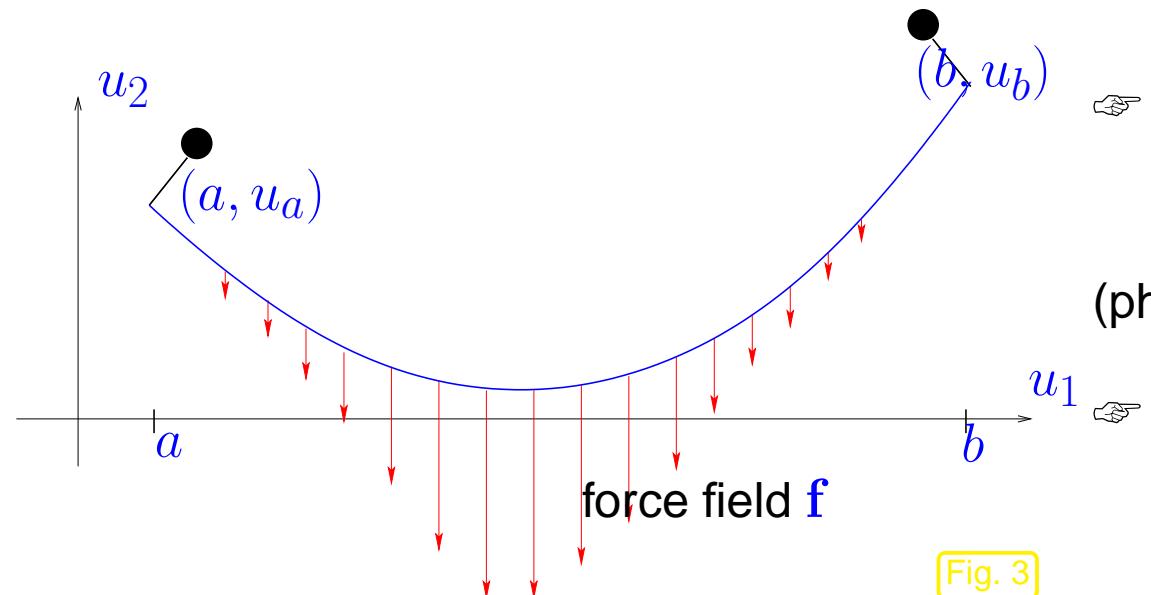
Fig. 1



Fig. 2

Sought: (Approximation of) “shape” of elastic string

Configuration space



= space of curves $[0, 1] \mapsto \mathbb{R}^2$

shape of string

curve $\mathbf{u} : [0, 1] \mapsto \mathbb{R}^2$, $\mathbf{u} = \mathbf{u}(\xi)$
(physical units $[\mathbf{u}] = 1\text{m}$)

☞ Pinning conditions (boundary conditions):

$$\mathbf{u}(0) = \begin{pmatrix} a \\ u_a \end{pmatrix} \in \mathbb{R}^2, \quad \mathbf{u}(1) = \begin{pmatrix} b \\ u_b \end{pmatrix} \in \mathbb{R}^2. \quad (1.2.1)$$

Terminology: $[0, 1] \hat{=} \text{parameter domain}$, ↗ notation Ω

Remark 1.2.2 (Parametrization of a curve). → [19, Sect. 7.4]

We consider a curve in \mathbb{R}^2 $\mathbf{u} : [0, 1] \mapsto \mathbb{R}^2$

$$\mathbf{u} \in (C^0([0, 1]))^2$$

\Updownarrow

connected curve

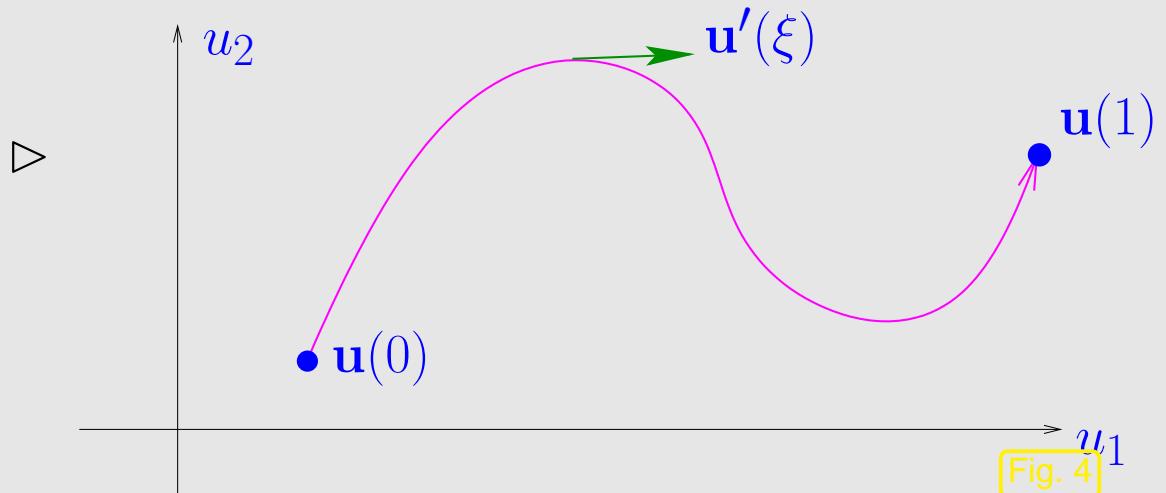


Fig. 4

notation: $C^k([a, b]) \hat{=} k$ -times continuously differentiable functions on $[a, b] \subset \mathbb{R}$, see [19, Sect. 5.4]

$(C^k([a, b]))^2 \hat{=} k$ -times continuously differentiable curves $\mathbf{u} : [a, b] \mapsto \mathbb{R}^2$, that is, if $\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$, then $u_1, u_2 \in C^k([a, b])$.

Geometric intuition: $\mathbf{u}(\xi)$ moves along the curve as ξ increases from 0 to 1.

Interpretation of curve parameter ξ : “virtual time”

$\Rightarrow \|\mathbf{u}'\| \hat{=} \text{“speed” with which curve is traversed}$

$\Rightarrow \int_0^1 \|\mathbf{u}'(\xi)\| d\xi \hat{=} \text{length of curve}$

➤ parametrization is supposed to be *locally injective*:

$$\forall \xi \in]0, 1[: \exists \epsilon > 0 : \forall \eta, |\eta - \xi| < \epsilon : \mathbf{u}(\eta) \neq \mathbf{u}(\xi) .$$

► For $\mathbf{u} \in (C^1([0, 1]))^2$ we expect $\mathbf{u}'(\xi) \neq 0$ for all $0 \leq \xi \leq 1$

✎ notation: $' \hat{=} \text{derivative w.r.t. curve parameter, here } \xi$



Remark 1.2.3 (Material coordinate).

Interpretation of curve parameter ξ :

ξ : unique identifier for each infinitesimal section of the string,
a *label* for each “material point” on the string

1.2

p. 26

- ξ $\hat{=}$ material coordinate, unrelated to “position in space” (= physical coordinate),
 ξ has no physical dimension ► $'$ does not affect dimension.



Remark 1.2.4 (Non-dimensional equations).

By fixing reference values for the basic physical units occurring in a model (“**scaling**”), one can switch to a **non-dimensional** form of the model equations.

In the case of the elastic string model the basic units are

- unit of length 1m ,
- unit of force 1N .

Thus, non-dimensional equations arise from fixing a reference length ℓ_0 and a reference force f_0 .

Below, physical units will be routinely dropped, which tacitly assumes a prior scaling.



1.2

Further problem parameters:

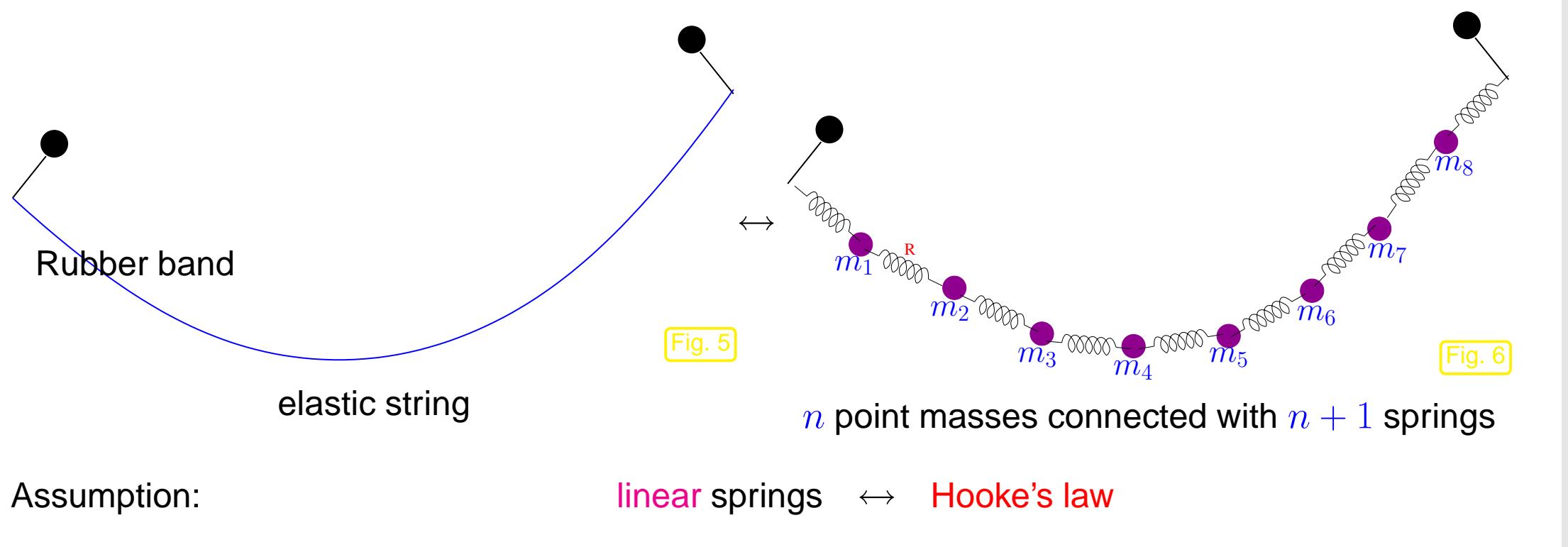
- force field $\mathbf{f} : [0, 1] \mapsto \mathbb{R}^2$, $[f] = 1\text{N}$, $\mathbf{f}(\xi) \hat{=} \text{force acting on material point } \xi$.

Special case: gravitational force $\mathbf{f}(\xi) := -g\rho(\xi) \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $0 \leq \xi \leq 1$, $g = 9.81\text{kg m s}^{-2}$
with density $\rho : [0, 1] \mapsto \mathbb{R}^+$, $[\rho] = \text{kg}$,

- local elastic material properties, see Sect. 1.2.3.

1.2.2 Mass-spring model

Idea: model string as a system of many simple components that interact in simple ways



Assumption:

linear springs \leftrightarrow Hooke's law

Force

$$F(l) = \kappa \left(\frac{l}{l_0} - 1 \right) \quad (\text{relative elongation}) . \quad (1.2.5)$$

$\hat{\kappa}$ $\hat{=}$ spring constant (stiffness), $[\kappa] = 1\text{N}$, $\kappa > 0$,

\hat{l}_0 $\hat{=}$ equilibrium length of (relaxed) spring.

► elastic energy stored in linear spring at length $l > 0$

$$E_{\text{el}} = \int_{l_0}^l F(\tau) d\tau = \frac{1}{2} \frac{\kappa}{l_0} (l - l_0)^2 , \quad [E_{\text{el}}] = 1\text{J} . \quad (1.2.6)$$

Configuration space for mass-spring model:

$\mathbf{u}^i \in \mathbb{R}^2 \hat{=} \text{position of } i\text{-th mass point}, i = 1, \dots, n$

➤ finite-dimensional configuration space $= (\mathbb{R}^2)^n$

Models, for which configurations can be described by means of finitely many real numbers are called **discrete**. Hence, the mass-spring model is a **discrete model**, see Sect. 1.5.

➤ Total elastic energy of mass-spring model in configuration $(\mathbf{u}^1, \dots, \mathbf{u}^n) \in (\mathbb{R}^2)^n$:

$$J_{\text{el}}^{(n)} = J_{\text{el}}^{(n)}(\mathbf{u}^1, \dots, \mathbf{u}^n) := \frac{1}{2} \sum_{i=0}^n \underbrace{\frac{\kappa_i}{l_i} (\|\mathbf{u}^{i+1} - \mathbf{u}^i\| - l_i)^2}_{\text{elastic energy of } i\text{-th spring}}, \quad (1.2.7)$$

where $\mathbf{u}^0 := \begin{pmatrix} a \\ u_a \end{pmatrix}$, $\mathbf{u}^{n+1} := \begin{pmatrix} b \\ u_b \end{pmatrix}$ (pinning positions (1.2.1)),
 $\kappa_i \hat{=} \text{spring constant of } i\text{-th spring}, i = 0, \dots, n$,
 $l_i > 0 \hat{=} \text{equilibrium length of } i\text{-th spring}.$

- Total potential energy of mass-spring model in configuration $(\mathbf{u}^1, \dots, \mathbf{u}^n)$ due to external force field:

$$J_f^{(n)} = J_f^{(n)}(\mathbf{u}^1, \dots, \mathbf{u}^n) := - \sum_{i=1}^n \mathbf{f}^i \cdot \mathbf{u}^i, \quad (1.2.8)$$

where $\mathbf{f}^i \hat{=} \text{force acting on } i\text{-th mass}, i = 1, \dots, n$.

☞ notation: $\mathbf{u} \cdot \mathbf{v} := \mathbf{u}^H \mathbf{v} \hat{=} \text{inner product of vectors in } \mathbb{C}^n$.

Known from classical mechanics, static case: **equilibrium principle**

systems attains configuration(s) of minimal (potential) energy

$$J^{(n)} := J_{\text{el}}^{(n)} + J_f^{(n)}$$

► equilibrium configuration $\mathbf{u}_*^1, \dots, \mathbf{u}_*^n$ of mass-spring system solves

$$(\mathbf{u}_*^1, \dots, \mathbf{u}_*^n) = \underset{(\mathbf{u}^1, \dots, \mathbf{u}^n) \in \mathbb{R}^{2n}}{\operatorname{argmin}} J^{(n)}(\mathbf{u}^1, \dots, \mathbf{u}^n). \quad (1.2.9)$$

Plot of $J^{(1)}(\mathbf{u}^1)$



Mass-spring system with only one point mass

(non-dimensional $l_1 = l_2 = 1$, $\kappa_1 = \kappa_2 = 1$,
 $\mathbf{u}^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\mathbf{u}^2 = \begin{pmatrix} 1 \\ 0.2 \end{pmatrix}$, $\mathbf{f}^1 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$)

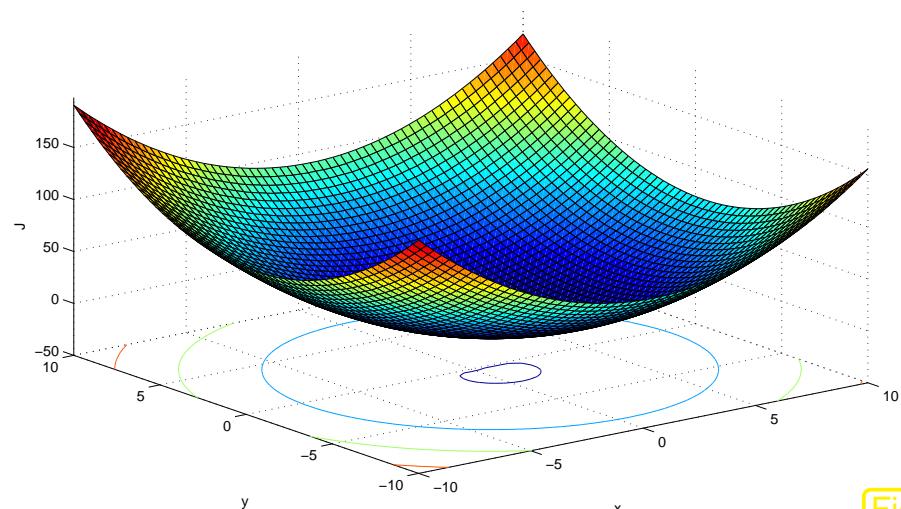
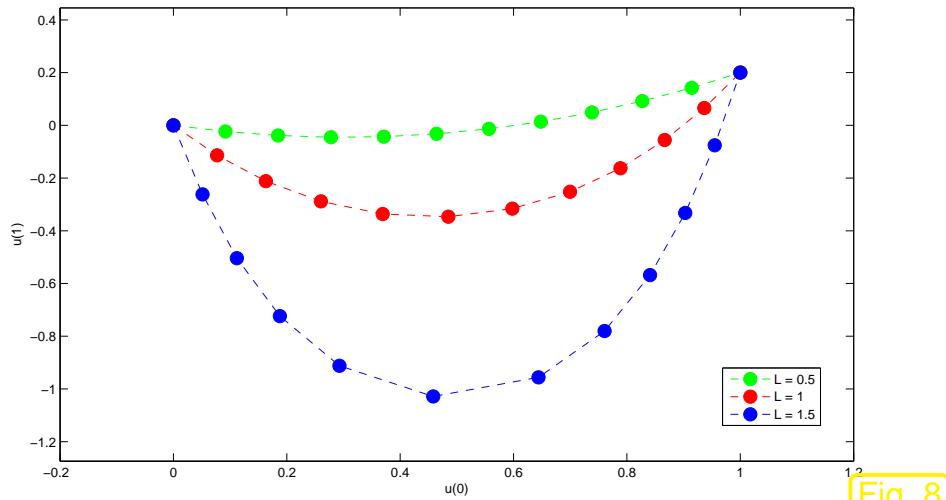


Fig. 7

Note: solutions of (1.2.9) need not be unique !

To see this, consider the case $L := \sum_{i=0}^n l_i > \|\mathbf{u}^{n+1} - \mathbf{u}^0\|$ and $\mathbf{f} \equiv 0$ (slack ensemble of springs without external forcing). In this situation many crooked arrangements of the masses will have zero total potential energy.



◁ minimal energy configuration a mass spring system for variable L .

($n = 10$, non-dimensional $\kappa_i = 1$, $l_i = L/n$, $i = 1, \dots, 10$)

Fig. 8

1.2.3 Continuum limit

Heuristics: elastic string = spring-mass system with “infinitely many infinitesimal masses” and “infinitesimally short” springs.

- Policy:
- consider sequence $(\mathcal{SMM}_n)_{n \in \mathbb{N}}$ of spring-mass systems with n masses,
 - identify material coordinate (\rightarrow Rem. 1.2.3) of point masses,
 - choose system parameters with meaningful limits,
 - derive expressions for energies as $n \rightarrow \infty$,
 - use them to define the “continuous elastic string model”.

Assumption: equal equilibrium lengths of all springs $l_i = \frac{L}{n+1}$, $L > 0$,
 ➤ $L \hat{=} \text{equilibrium length of elastic string: } L = \sum_i l_i$, $[L] = 1\text{m}$.

Equilibrium configuration of mass-spring system▷

(non-dimensional $l_i = \frac{L}{n+1}$, $\kappa_i = 1$, $\mathbf{f}_i = \frac{1}{n} \begin{pmatrix} 0 \\ -1 \end{pmatrix}$,
 $L = 1$, n varying)

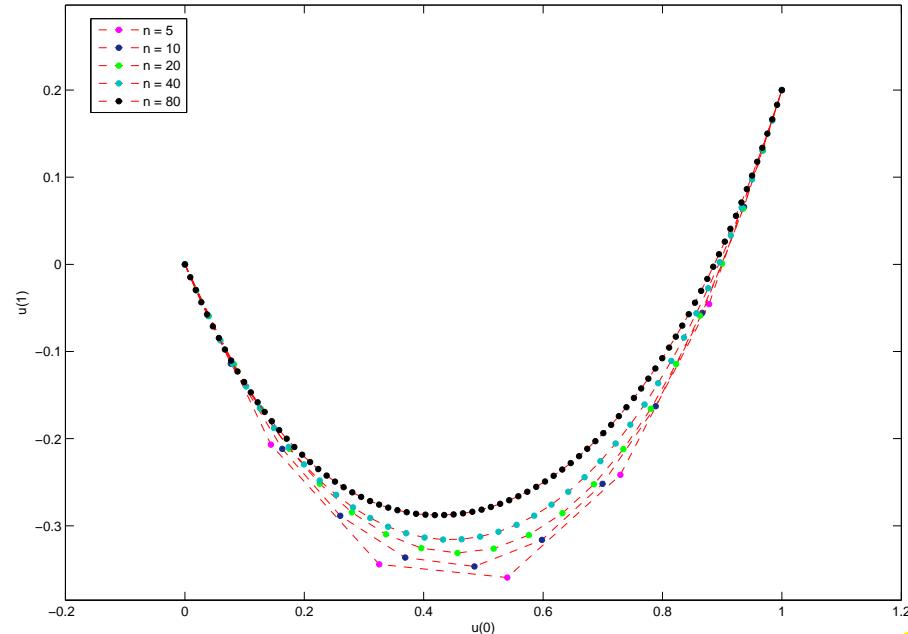


Fig. 9

- masses are uniformly spaced **on** string $\mathbf{u} : [0, 1] \mapsto \mathbb{R}^2$
- material coordinate of i -th mass in $\mathcal{SMM}_n = \xi_i^{(n)} := \frac{i}{n+1} :$ $\mathbf{u}^i := \mathbf{u}(\xi_i^{(n)})$

In the spring-mass model each spring has its own stiffness κ_i and every mass point its own force \mathbf{f}^i acting on it. When considering the “limit” of a sequence of spring-mass models, we have to detach stiffness and force from springs and masses and attach them to material points, cf. Rem. 1.2.3. In other words stiffness κ_i and force \mathbf{f}^i have to be induced by a stiffness function $\kappa(\xi)$ and force function $\mathbf{f}(\xi)$. This linkage has to be done in a way to allow for a meaningful limit $n \rightarrow \infty$ for the potential energy.

“Limit-compatible” system parameters: $(\xi_{i+1/2}^{(n)} := \frac{1}{2}(\xi_{i+1}^{(n)} + \xi_i^{(n)}))$

- $\kappa_i = \kappa(\xi_{i+1/2}^{(n)})$ with *integrable* stiffness function $\kappa : [0, 1] \mapsto \mathbb{R}^+$,

$$\xi_{i+1/2}^{(n)}$$

- $\mathbf{f}^i = \int_{\xi_{i-1/2}^{(n)}}^{\xi_{i+1/2}^{(n)}} \mathbf{f}(\xi) d\xi$ “lumped force”, integrable force field $\mathbf{f} : [0, 1] \mapsto \mathbb{R}^2$

$$\xi_{i-1/2}^{(n)}$$

- energies, see (1.2.7), (1.2.8)

$$J_{\text{el}}^{(n)}(\mathbf{u}) = \frac{1}{2} \sum_{i=0}^n \frac{n+1}{L} \kappa(\xi_{i+1/2}^{(n)}) \left(\left\| \mathbf{u}(\xi_{i+1}^{(n)}) - \mathbf{u}(\xi_i^{(n)}) \right\| - \frac{L}{n+1} \right)^2, \quad (1.2.10)$$

$$J_f^{(n)}(\mathbf{u}) = - \sum_{i=1}^n \int_{\xi_{i-1/2}^{(n)}}^{\xi_{i+1/2}^{(n)}} \mathbf{f}(\xi) d\xi \cdot \mathbf{u}(\xi_i^{(n)}) . \quad (1.2.11)$$

Assumption: $\mathbf{u} \in (C^2([0, 1]))^2$ (twice continuously differentiable)

① Simple limit for potential energy due to external force:

$$J_f(\mathbf{u}) = \lim_{n \rightarrow \infty} J_f^{(n)}(\mathbf{u}) = \lim_{n \rightarrow \infty} - \sum_{i=1}^n \int_{\xi_{i-1/2}^{(n)}}^{\xi_{i+1/2}^{(n)}} \mathbf{f}(\xi) d\xi \cdot \mathbf{u}(\xi_i^n) = - \int_0^1 \mathbf{f}(\xi) \cdot \mathbf{u}(\xi) d\xi . \quad (1.2.12)$$

② Limit of elastic energy:

Tool: Taylor expansion: for $\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \in C^2$ with derivative \mathbf{u}' , $1 \gg \eta \rightarrow 0$

$$\begin{aligned}\|\mathbf{u}(\xi + \eta) - \mathbf{u}(\xi - \eta)\| &= \sqrt{(u_1(\xi + \eta) - u_1(\xi - \eta))^2 + (u_2(\xi + \eta) - u_2(\xi - \eta))^2} \\ &= \sqrt{(2u'_1(\xi)\eta + O(\eta^3))^2 + (2u'_2(\xi)\eta + O(\eta^3))^2} \\ &= 2\eta \|\mathbf{u}'(\xi)\| \sqrt{1 + O(\eta^2)} = 2\eta \|\mathbf{u}'(\xi)\| + O(\eta^2).\end{aligned}\tag{1.2.13}$$

Apply this to (1.2.10) with $\eta = \frac{1}{2n+1}$ for $n \rightarrow \infty$

$$\begin{aligned}J_{\text{el}}^{(n)}(\mathbf{u}) &= \frac{1}{2} \sum_{i=0}^n \frac{n+1}{L} \kappa(\xi_{i+1/2}^{(n)}) \left(\frac{1}{n+1} \|\mathbf{u}'(\xi_{i+1/2}^{(n)})\| + O\left(\frac{1}{(n+1)^2}\right) - \frac{L}{n+1} \right)^2 \\ &= \frac{1}{2L} \frac{1}{n+1} \sum_{i=0}^n \kappa(\xi_{i+1/2}^{(n)}) \left(\|\mathbf{u}'(\xi_{i+1/2}^{(n)})\| + O\left(\frac{1}{n+1}\right) - L \right)^2\end{aligned}\tag{1.2.14}$$

Consideration: integral as limit of Riemann sums, see [19, Sect. 6.2]:

$$q \in C^0([0, 1]): \quad \lim_{n \rightarrow \infty} \frac{1}{n+1} \sum_{j=0}^n q\left(\frac{j+1/2}{n+1}\right) = \int_0^1 q(\xi) d\xi.\tag{1.2.15}$$

$$\Rightarrow J_{\text{el}}(\mathbf{u}) = \lim_{n \rightarrow \infty} J_{\text{el}}^{(n)}(\mathbf{u}) = \frac{1}{2L} \int_0^1 \kappa(\xi) (\|\mathbf{u}'(\xi)\| - L)^2 d\xi . \quad (1.2.16)$$

► Equilibrium condition for limit model (minimal total potential energy):

$$\mathbf{u}_* = \underset{\mathbf{u} \in (C^1([0,1]))^2 \text{ & (1.2.1)}}{\operatorname{argmin}} \underbrace{\int_0^1 \frac{\kappa(\xi)}{2L} (\|\mathbf{u}'(\xi)\| - L)^2 - \mathbf{f}(\xi) \cdot \mathbf{u}(\xi) d\xi}_{=:J(\mathbf{u})} . \quad (1.2.17)$$

total potential energy functional, $[J] = 1\text{J}$
= a minimization problem in a **function space** !

Example 1.2.18 (Tense string without external forcing).

Setting:

- no external force: $\mathbf{f} \equiv 0$
- homogeneous string: $\kappa = \kappa_0 = \text{const}$
- tense string: $L < \|\mathbf{u}(0) - \mathbf{u}(1)\|$
(\Rightarrow positive elastic energy)



[Fig. 10]

► (1.2.17) $\Leftrightarrow \mathbf{u}_* = \underset{\mathbf{u} \in (C^1([0,1]))^2 \& (1.2.1)}{\operatorname{argmin}} \frac{\kappa_0}{2L} \int_0^1 (\|\mathbf{u}'(\xi)\| - L)^2 d\xi .$ (1.2.19)

Note: in (1.2.19) \mathbf{u} enters J only through \mathbf{u}' !

Constraint on \mathbf{u}' : by triangle inequality for integrals, see [19, Sect. 6.3]

$$\ell := \|\mathbf{u}(1) - \mathbf{u}(0)\| = \left\| \int_0^1 \mathbf{u}'(\xi) d\xi \right\| \leq \int_0^1 \|\mathbf{u}'(\xi)\| d\xi . \quad (1.2.20)$$

→ Consider related minimization problem

$$w_* = \operatorname{argmin}_w \left\{ \frac{\kappa_0}{2L} \int_0^1 (w - L)^2 d\xi : \begin{array}{l} w \in (C^0([0, 1]))^2, \\ \int_0^1 w(\xi) d\xi \geq \ell \end{array} \right\}. \quad (1.2.21)$$

⇒ unique solution $w_*(\xi) = \ell$ (constant solution)

$\|\mathbf{u}'(\xi)\| = \ell$ and the boundary conditions (1.2.1) are satisfied for the **straight line solution** of (1.2.19)

$$\mathbf{u}_*(\xi) = (1 - \xi)\mathbf{u}(0) + \xi\mathbf{u}(1).$$

It is exactly the “straight string” solution that physical intuition suggests.



1.3 Variational approach

We face the task of minimizing a functional over an ∞ -dimensional function space. In this section necessary conditions for the minimizer will formally be derived in the form of variational equations. This idea is one of the cornerstone of a branch of analysis called **calculus of variations**.

1.3.1 Virtual work equation

☞ notation: $C_0^k([0, 1]) := \{v \in C^k([0, 1]): v(0) = v(1) = 0\}$, $k \in \mathbb{N}_0$

Main “idea of calculus of variations”:

$$\mathbf{u}_* \text{ solves (1.2.17)} \Rightarrow J(\mathbf{u}_*) \leq J(\mathbf{u}_* + t\mathbf{v}) \quad \forall t \in \mathbb{R}, \mathbf{v} \in (C_0^2([0, 1]))^2. \quad (1.3.1)$$

► $\varphi(t) := J(\mathbf{u}_* + t\mathbf{v})$ has global minimum for $t = 0$

► If φ differentiable, then $\frac{d\varphi}{dt}(0) = 0$

Computation of $\frac{d\varphi}{dt}(0)$ for J from (1.2.17) amounts to computing a “configurational derivative” in direction \mathbf{v} .

We pursue a separate treatment of energy contributions:

① Potential energy (1.2.12) due to external force:

$$\lim_{t \rightarrow 0} \frac{J_{\text{pot}}(\mathbf{u}_* + t\mathbf{v}) - J_{\text{pot}}(\mathbf{u}_*)}{t} = - \lim_{t \rightarrow 0} \frac{1}{t} \int_0^1 \mathbf{f}(\xi) \cdot t\mathbf{v}(\xi) d\xi = - \int_0^1 \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) d\xi. \quad (1.3.2)$$

② Elastic energy (1.2.16): more difficult, tool: Taylor expansion

Analogous to (1.2.13), $\mathbf{x} \in \mathbb{R}^2 \setminus \{0\}$, $\mathbf{h} \in \mathbb{R}^2$, for $\mathbb{R} \ni t \rightarrow 0$

$$\begin{aligned}\|\mathbf{x} + t\mathbf{h}\| &= \sqrt{(x_1 + th_1)^2 + (x_2 + th_2)^2} = \sqrt{\|\mathbf{x}\|^2 + 2t\mathbf{x} \cdot \mathbf{h} + t^2 \|\mathbf{h}\|^2} \\ &= \|\mathbf{x}\| \sqrt{1 + 2t \frac{\mathbf{x} \cdot \mathbf{h}}{\|\mathbf{x}\|^2} + t^2 \frac{\|\mathbf{h}\|^2}{\|\mathbf{x}\|^2}} = \|\mathbf{x}\| + t \frac{\mathbf{x} \cdot \mathbf{h}}{\|\mathbf{x}\|} + O(t^2) ,\end{aligned}\tag{1.3.3}$$

where we used

$$\sqrt{1 + \delta} = 1 + \frac{1}{2}\delta + O(\delta^2) \quad \text{for } \delta \rightarrow 0 .\tag{1.3.4}$$

Use (1.3.3) in the perturbation analysis for the elastic energy:

$$\begin{aligned}\blacktriangleright (\|\mathbf{u}'(\xi) + t\mathbf{v}'(\xi)\| - L)^2 &= \left(\|\mathbf{u}'(\xi)\| + t \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} + O(t^2) - L \right)^2 \\ &= (\|\mathbf{u}'(\xi)\| - L)^2 + 2t (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} + O(t^2) .\\ \blacktriangleright J_{\text{el}}(\mathbf{u} + t\mathbf{v}) - J_{\text{el}}(\mathbf{u}) &= \frac{t}{L} \int_0^1 \kappa(\xi) (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} + O(t^2) d\xi .\end{aligned}\tag{1.3.5}$$

►
$$\lim_{t \rightarrow 0} \frac{J_{\text{el}}(\mathbf{u}_* + t\mathbf{v}) - J_{\text{el}}(\mathbf{u}_*)}{t} = \int_0^1 \frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} d\xi . \quad (1.3.6)$$

Here we take for granted $\|\mathbf{u}'(\xi)\| \neq 0$, which is an essential property of a meaningful parameterization of the elastic string, see Rem. 1.2.2.



Necessary condition for \mathbf{u}_* solving (1.2.17)

$$\int_0^1 \frac{\kappa(\xi)}{L} (\|\mathbf{u}'_*(\xi)\| - L) \frac{\mathbf{u}'_*(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'_*(\xi)\|} - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) d\xi = 0 \quad \forall \mathbf{v} \in (C_0^2([0, 1]))^2 . \quad (1.3.7)$$

This is a

non-linear variational equation on domain $\Omega = [0, 1]$

Remark 1.3.8 (Differentiating a functional on a space of curves).

For a C^2 -function $F : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$, $d \in \mathbb{N}$, consider the functional

$$J : (C_{\text{pw}}^1([0, 1]))^d \mapsto \mathbb{R} \quad , \quad J(\mathbf{u}) := \int_0^1 F(\mathbf{u}'(\xi), \mathbf{u}(\xi)) \, d\xi .$$

By simple Taylor expansion we find

$$J(\mathbf{u} + t\mathbf{v}) = J(\mathbf{u}) + t \underbrace{\int_0^1 D_1 F(\mathbf{u}'(\xi), \mathbf{u}(\xi)) \mathbf{v}'(\xi) + D_2 F(\mathbf{u}'(\xi), \mathbf{u}(\xi)) \mathbf{v}(\xi) \, d\xi}_{\text{"directional derivative" } (D_{\mathbf{u}} J)(\mathbf{u})(\mathbf{v})} + O(t^2) .$$

Here, $D_1 F$ and $D_2 F$ are the **partial derivatives** of F w.r.t the first and second vector argument, respectively. These are *row vectors*. The derivatives \mathbf{u}' , \mathbf{v}' are just regular 1D derivatives w.r.t. the parameter ξ . They yield column vectors.



Remark 1.3.9 (Virtual work principle).

In statics, the derivation of variational equations from energy minimization (equilibrium principle, see (1.2.9)) is known as the method of **virtual work**: Small admissible changes of the equilibrium configuration of the system invariably entail active work.



Remark 1.3.10 (Non-linear variational equation).

Recall from linear algebra:

*Definition 1.3.11. Given an \mathbb{R} -vector space V , a **linear form** ℓ is a mapping $f : V \mapsto \mathbb{R}$ that satisfies*

$$\ell(\alpha u + \beta v) = \alpha\ell(u) + \beta\ell(v) \quad \forall u, v \in V, \forall \alpha, \beta \in \mathbb{R}.$$

A **bilinear form** a on V is a mapping $a : V \times V \mapsto \mathbb{R}$, for which

$$\begin{aligned} a(\alpha_1 v_1 + \beta_1 u_1, \alpha_2 v_2 + \beta_2 u_2) &= \\ &= \alpha_1 \alpha_2 a(v_1, v_2) + \alpha_1 \beta_2 a(v_1, u_2) + \beta_1 \alpha_2 a(u_1, v_2) + \beta_1 \beta_2 a(u_1, u_2) \end{aligned}$$

for all $u_i, v_i \in V, \alpha_i, \beta_i \in \mathbb{R}, i = 1, 2$.

Structure of (1.3.7): abstract non-linear variational equation

$$u \in V: \quad \mathbf{a}(u; v) = \ell(v) \quad \forall v \in V_0 , \quad (1.3.12)$$

- $V_0 \hat{=} (\text{real}) \text{ vector space of functions},$
- $V \hat{=} \text{affine space of functions}: \quad V = u_0 + V_0, \text{ with offset function } u_0 \in V,$
- $\ell \hat{=} \text{a linear mapping } V_0 \mapsto \mathbb{R}, \text{ a linear form,}$
- $\mathbf{a} \hat{=} \text{a mapping } V \times V_0 \mapsto \mathbb{R}, \text{ linear in the second argument, that is}$

$$\mathbf{a}(u; \alpha v + \beta w) = \alpha \mathbf{a}(u; v) + \beta \mathbf{a}(u; w) \quad \forall u \in V, v, w \in V_0, \alpha, \beta \in \mathbb{R} . \quad (1.3.13)$$

Terminology related to variational problem (1.3.12): V = trial space

V_0 = test space

Explanation of terminology:

1.3

p. 47

- **trial space** $\hat{=}$ the function space in which we seek the solution
- **test space** $\hat{=}$ the space of eligible **test functions** v in a variational problem like (1.3.12)

The two spaces need not be the same: $V \longleftrightarrow V_0$. For many variational problems, which are not examined in this course, they may even comprise functions with different smoothness properties.

Rewriting (1.3.12) using the offset function $u_0 \in V$:

$$(1.3.12) \Rightarrow w \in V_0: a(u_0 + w; v) = l(v) \quad \forall v \in V_0 \quad \text{and} \quad u = u_0 + w. \quad (1.3.14)$$

In concrete terms (for elastic string continuum model):

- $V_0 := (C_0^2([0, 1]))^2$,
- $V := \{\mathbf{u} \in (C^2([0, 1]))^2 : \mathbf{u}(0) = \begin{pmatrix} a \\ u_a \end{pmatrix}, \mathbf{u}(1) = \begin{pmatrix} b \\ u_b \end{pmatrix}\}$
 $= \underbrace{[\xi \mapsto (1 - \xi)\mathbf{u}(0) + \xi\mathbf{u}(1)]}_{=: \mathbf{u}_0} + V_0, \quad (1.3.15)$

- $\ell(\mathbf{v}) := \int_0^1 \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) d\xi , \quad (1.3.16)$

- $a(\mathbf{u}; \mathbf{v}) := \int_0^1 \frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} d\xi . \quad (1.3.17)$

△

1.3.2 Regularity requirements

Issue: The derivation of the continuum models (1.2.17) (\rightarrow Sect. 1.2.3) and (1.3.7) was based on the assumption $\mathbf{u} \in (C^2([0, 1]))^2$.

Is $\mathbf{u} \in (C^2([0, 1]))^2$ required to render the minimization problem (1.2.17)/variational problem 1.2.3) meaningful ?

Obvious (\rightarrow c Rem. 1.2.2):

$$\mathbf{u} \in (C^0([0, 1]))^2
(\text{string must not be torn})$$

Observation:

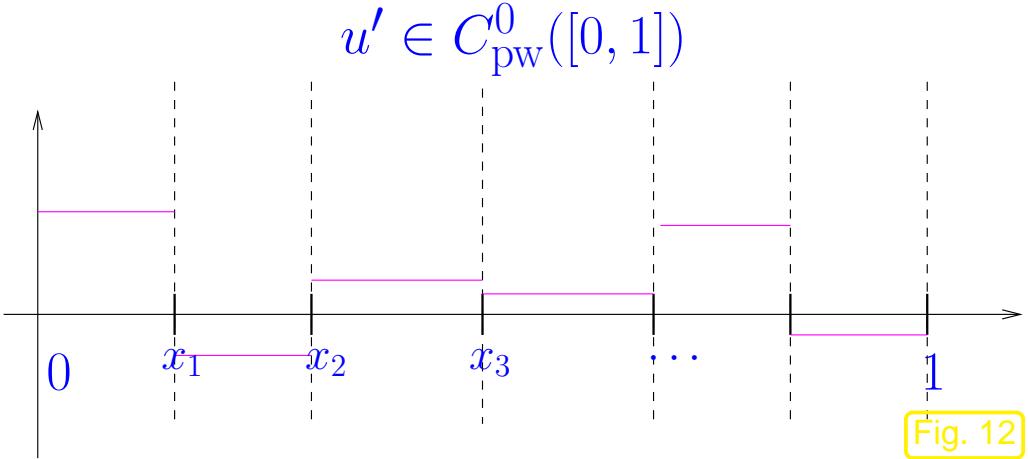
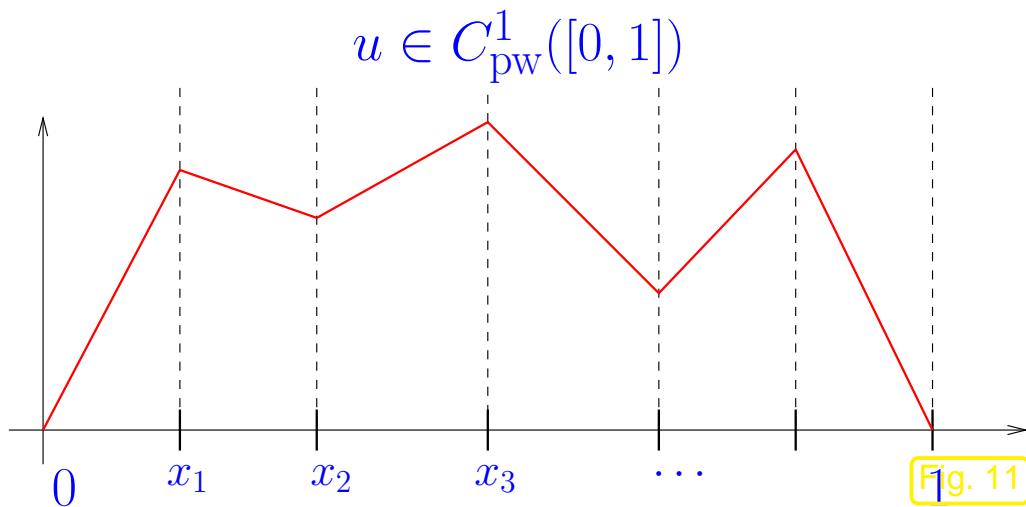
- $J(\mathbf{u})$ from (1.2.17), \mathbf{a} from (1.3.17) well defined for

merely *continuous, piecewise continuously differentiable* functions $\mathbf{u}, \mathbf{v} : [0, 1] \mapsto \mathbb{R}^2$,

➢ \mathbf{u}' will be piecewise continuous and can be integrated.

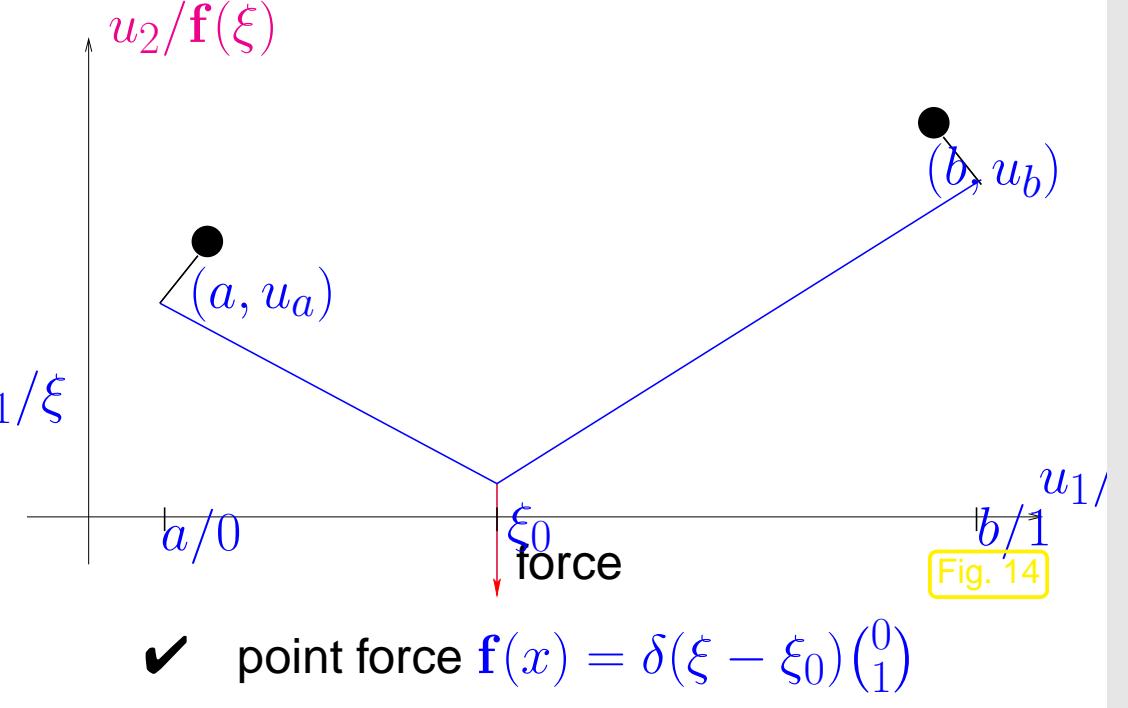
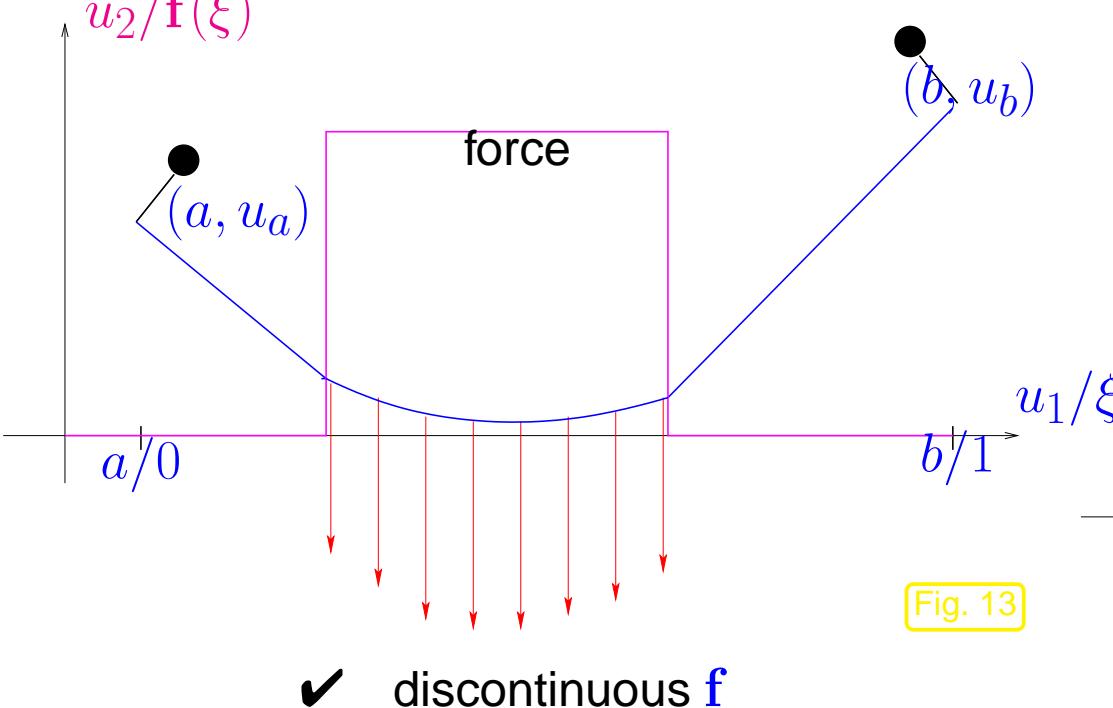
- mere integrability of κ, \mathbf{f} sufficient.

notation: $C_{\text{pw}}^k([a, b]) \hat{=} \text{globally } C^{k-1} \text{ and piecewise } k\text{-times continuously differentiable functions on } [a, b] \subset \mathbb{R}$: for each $v \in C_{\text{pw}}^k([a, b])$ there is a finite partition $\{a = \tau_0 < \tau_1 < \dots < \tau_m = b\}$ such that $v|_{[\tau_{i-1}, \tau_i]}$ can be extended to a function $\in C^k([\tau_{i-1}, \tau_i])$. $C_{\text{pw}}^0([a, b]) \hat{=} \text{piecewise continuous functions with only a finite number of discontinuities.}$

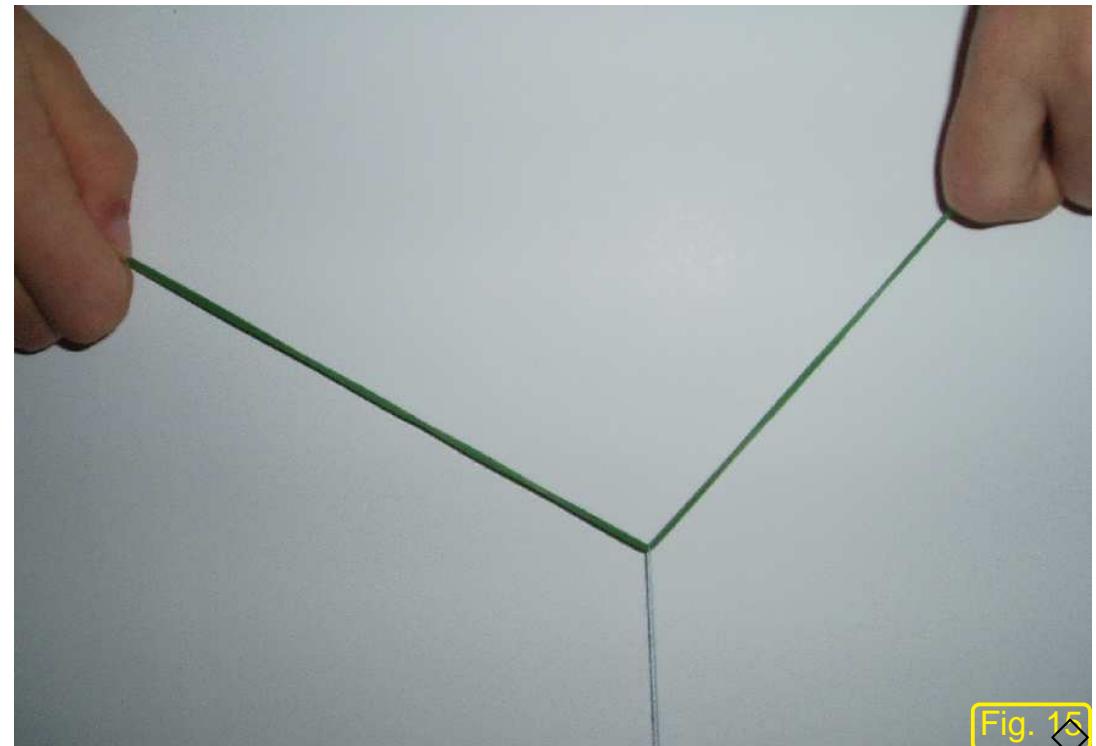


Example 1.3.18 (Non-smooth external forcing).

Setting: $\kappa = \text{const}$ (homogeneous string)



- ⇒ $\mathbf{u}_* \notin (C^2([0, 1]))^2$ physically meaningful:
- $\mathbf{u}_* \in (C^1([0, 1]))^2$ for discontinuous \mathbf{f}
- merely $\mathbf{u}_* \in (C^0([0, 1]))^2$ for point force concentrated in ξ : kink at ξ_0 !



1.3.3 Differential equation

Consider non-linear variational equation (1.3.7):

$$\int_0^1 \frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) d\xi = 0 \quad \forall \mathbf{v} \in (C_{0,\text{pw}}^1([0, 1]))^2. \quad (1.3.7)$$

Assumption: $\mathbf{u} \in (C^2([0, 1]))^2 \text{ & } \kappa \in C^1([0, 1]) \text{ & } \mathbf{f} \in (C^0([0, 1]))^2$ (1.3.19)

Recall: integration by parts formula:

$$\int_0^1 u(\xi)v'(\xi) d\xi = - \int_0^1 u'(\xi)v(\xi) d\xi + \underbrace{(u(1)v(1) - u(0)v(0))}_{\text{boundary terms}} \quad \forall u, v \in C_{\text{pw}}^1([0, 1]). \quad (1.3.20)$$

Apply to elastic energy contribution in (1.3.7):

$$\begin{aligned} & \int_0^1 \left(\frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi)}{\|\mathbf{u}'(\xi)\|} \right) \cdot \mathbf{v}'(\xi) - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi \\ &= \int_0^1 \left\{ -\frac{d}{d\xi} \left(\frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi)}{\|\mathbf{u}'(\xi)\|} \right) - \mathbf{f}(\xi) \right\} \cdot \mathbf{v}(\xi) \, d\xi . \end{aligned}$$

Note:

$$\mathbf{v}(0) = \mathbf{v}(1) = 0 \Rightarrow \text{boundary terms vanish !}$$

$$(1.3.7) \Rightarrow \int_0^1 \underbrace{\left\{ -\frac{d}{d\xi} \left(\frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi)}{\|\mathbf{u}'(\xi)\|} \right) - \mathbf{f}(\xi) \right\}}_{\in C_{pw}^0([0,1])} \cdot \mathbf{v}(\xi) \, d\xi = 0$$

$\forall \mathbf{v} \in (C_0^1([0, 1]))^2$

Lemma 1.3.21 (fundamental lemma of the calculus of variations).

Let $f \in C_{\text{pw}}^0([a, b])$, $-\infty < a < b < \infty$, satisfy

$$\int_a^b f(\xi) v(\xi) d\xi = 0 \quad \forall v \in C^k([a, b]), v(a) = v(b) = 0 .$$

for some $k \in \mathbb{N}_0$. This implies $f \equiv 0$.

$$\text{Ass. (1.3.19) \& (1.3.7)} \xrightarrow{\text{Lemma 1.3.21}} -\frac{d}{d\xi} \left(\frac{\kappa(\xi)}{L} (\|\mathbf{u}'(\xi)\| - L) \frac{\mathbf{u}'(\xi)}{\|\mathbf{u}'(\xi)\|} \right) = \mathbf{f}(\xi) \quad 0 \leq \xi \leq 1 .$$

If $\kappa \in C^1$, $\mathbf{f} \in C^0$, then a C^2 -minimizer of J /a C^2 -solution of (1.3.7) solve the 2nd-order ODE

$$\frac{d}{d\xi} \left(\kappa(\xi) (\|\mathbf{u}'\| - L) \frac{\mathbf{u}'}{\|\mathbf{u}'\|} \right) = \mathbf{f} \quad \text{on } [0; 1] . \quad (1.3.22)$$

ODE (1.3.22) + boundary conditions (1.2.1) = two-point boundary value problem
 (on domain $\Omega = [0, 1]$)

Minimization problem
 (1.2.17)

$$\mathbf{u}_* = \underset{\mathbf{v} \in V}{\operatorname{argmin}} J(\mathbf{v})$$

①

Variational problem
 (1.3.7)

$$a(\mathbf{u}; \mathbf{v}) = f(\mathbf{v}) \quad \forall \mathbf{v}$$

②

Two-point BVP

$$F(\mathbf{u}, \mathbf{u}', \mathbf{u}'') = \mathbf{f}, \\ \mathbf{u}(0), \mathbf{u}(1) \text{ fixed}.$$

- ①: equivalence (“ \Leftrightarrow ”) holds if minimization problem has unique solution
- ②: meaningful two-point BVP stipulates extra regularity (smoothness) of \mathbf{u} , see Rem. 1.3.23.

Terminology: $\left\{ \begin{array}{l} \text{minimization problem (1.2.17)} \\ \text{variational problem (1.3.7)} \end{array} \right\}$ is called the **weak form** of the string model,
 Two-point boundary value problem (1.3.22), (1.2.1) is called the **strong form** of the string model.

A solution \mathbf{u} of (1.3.22), for which all occurring derivatives are continuous is called a **classical solution** of the two-point BVP.

Minimization problem

(1.2.17):

- κ, \mathbf{f} integrable,
- \mathbf{u} piecewise C^1

=

Variational problem

(1.3.7):

- κ, \mathbf{f} integrable,
- \mathbf{u} piecewise C^1

≠

Two-point BVP:

- $\kappa \in C^1([0, 1]),$
- $\mathbf{f} \in (C^0([0, 1]))^2,$
- $\mathbf{u} \in (C^2([0, 1]))^2.$

☞ formulation as a classical two-point BVP imposes (unduly) restrictive smoothness on solution and coefficient functions.



Lemma 1.3.24 (Classical solutions are weak solutions).

For $\tilde{\kappa} \in C^1([0, 1]),$ any classical solution of (1.3.22) also solves (1.3.7).

Proof. (“Derivation of (1.3.22) reversed”)

Multiply (1.3.22) with $v \in C_{0,\text{pw}}^1([0, 1])$ and integrate over $[0, 1]$. Then push a derivative onto v by using (1.3.20). \square

1.4 Simplified model

Setting: taut string

$$L \ll \|u(0) - u(1)\| . \quad (1.4.1)$$

→ expected: $\|u'_*(\xi)\| \gg L$ for all $0 \leq \xi \leq 1$ for solution u_* of (1.2.17)

“Intuitive asymptotics”: • renormalize stiffness $\kappa \rightarrow \tilde{\kappa} := \frac{\kappa}{L}$, $[\tilde{\kappa}] = \text{Nm}^{-1}$
• suppress equilibrium length: $L = 0$ in (1.2.17).



Simplified equilibrium model:

$$\widetilde{\mathbf{u}}_* = \underset{\mathbf{u} \in (C_{\text{pw}}^1([0,1]))^2 \& (1.2.1)}{\operatorname{argmin}} \int_0^1 \frac{1}{2} \widetilde{\kappa}(\xi) \|\mathbf{u}'(\xi)\|^2 - \mathbf{f}(\xi) \cdot \mathbf{u}(\xi) d\xi . \quad (1.4.2)$$

$\underbrace{\qquad\qquad\qquad}_{=: \widetilde{J}(\mathbf{u})}$

= a quadratic minimization problem in a function space !



Corresponding variational problem: use

$$\|\mathbf{x} + t\mathbf{h}\|^2 = \|\mathbf{x}\|^2 + 2t\mathbf{x} \cdot \mathbf{h} + t^2 \|\mathbf{h}\|^2 = \|\mathbf{x}\|^2 + 2t\mathbf{x} \cdot \mathbf{h} + O(t^2) .$$



$$\lim_{t \rightarrow 0} \frac{\widetilde{J}(\mathbf{u} + t\mathbf{v}) - \widetilde{J}(\mathbf{u})}{t} = \int_0^1 \widetilde{\kappa}(\xi) \mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi) - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) d\xi = 0 , \quad \mathbf{v} \in (C_{\text{pw},0}^1([0,1]))^2 .$$

Variational equation satisfied by solution $\tilde{\mathbf{u}}_*$ of (1.4.2):

$$\int_0^1 \tilde{\kappa}(\xi) \mathbf{u}'_*(\xi) \cdot \mathbf{v}'(\xi) - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi = 0 \quad \forall \mathbf{v} \in (C_{\text{pw},0}^1([0,1]))^2. \quad (1.4.3)$$

Remark 1.4.4 (Linear variational problems). \rightarrow Rem. 1.3.10

(1.4.3) has the structure (1.3.12)

$$u \in V: \quad \mathbf{a}(u, v) = f(v) \quad \forall v \in V_0, \quad (1.4.5)$$

where now

- $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R}$ is a **bilinear form** (\rightarrow Def. 1.3.11), that is, linear in *both* arguments.



→ Corresponding two-point boundary value problem: by integration by parts, see (1.3.20),

$$\int_0^1 \tilde{\kappa}(\xi) \mathbf{u}'_*(\xi) \cdot \mathbf{v}'(\xi) - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) d\xi = \int_0^1 \left\{ -\frac{d}{d\xi} \left(\tilde{\kappa}(\xi) \frac{d}{d\xi} \mathbf{u}(\xi) \right) - \mathbf{f}(\xi) \right\} \cdot \mathbf{v}(\xi) d\xi$$

$$\forall \mathbf{v} \in (C_{pw,0}^1([0, 1]))^2.$$

Then use Lemma 1.3.21.

If $\kappa \in C^1$, $f \in C^0$, then a C^2 -solution of (1.4.3) solves the two-point BVP

$$\begin{aligned} -\frac{d}{d\xi} \left(\tilde{\kappa}(\xi) \frac{d\mathbf{u}}{d\xi}(\xi) \right) &= \mathbf{f}(\xi) , \quad 0 \leq \xi \leq 1 , \\ \mathbf{u}(0) &= \begin{pmatrix} a \\ u_a \end{pmatrix} , \quad \mathbf{u}(1) = \begin{pmatrix} b \\ u_b \end{pmatrix} . \end{aligned} \tag{1.4.6}$$

Special setting:

“gravitational force” $\mathbf{f}(\xi) = -g(\xi)\mathbf{e}_2$



(1.4.2) decouples into two minimization problems for the components of \mathbf{u} !

$$\begin{aligned} \widetilde{u}_{1,*} &= \underset{u \in C_{\text{pw}}^1([0,1]), u(0)=a, u(1)=b}{\operatorname{argmin}} \frac{1}{2} \int_0^1 \widetilde{\kappa}(\xi)(u'(\xi))^2 d\xi , \\ (1.4.2) \Rightarrow \end{aligned} \quad (1.4.7)$$

$$\widetilde{u}_{2,*} = \underset{u \in C_{\text{pw}}^1([0,1]), u(0)=u_a, u(1)=u_b}{\operatorname{argmin}} \int_0^1 \frac{1}{2} \widetilde{\kappa}(\xi)(u'(\xi))^2 + g(\xi)u(\xi) d\xi .$$

The minimization problem for $\widetilde{u}_{1,*}$ has a closed-form solution:

$$\widetilde{u}_{1,*}(\xi) = a + \frac{b-a}{\int_0^1 \widetilde{\kappa}^{-1}(\tau) d\tau} \int_0^\xi \widetilde{\kappa}^{-1}(\tau) d\tau , \quad 0 \leq \xi \leq 1 . \quad (1.4.8)$$

The minimization problem for $\widetilde{u}_{2,*}$ leads to the *linear* variational problem, cf. (1.4.3)

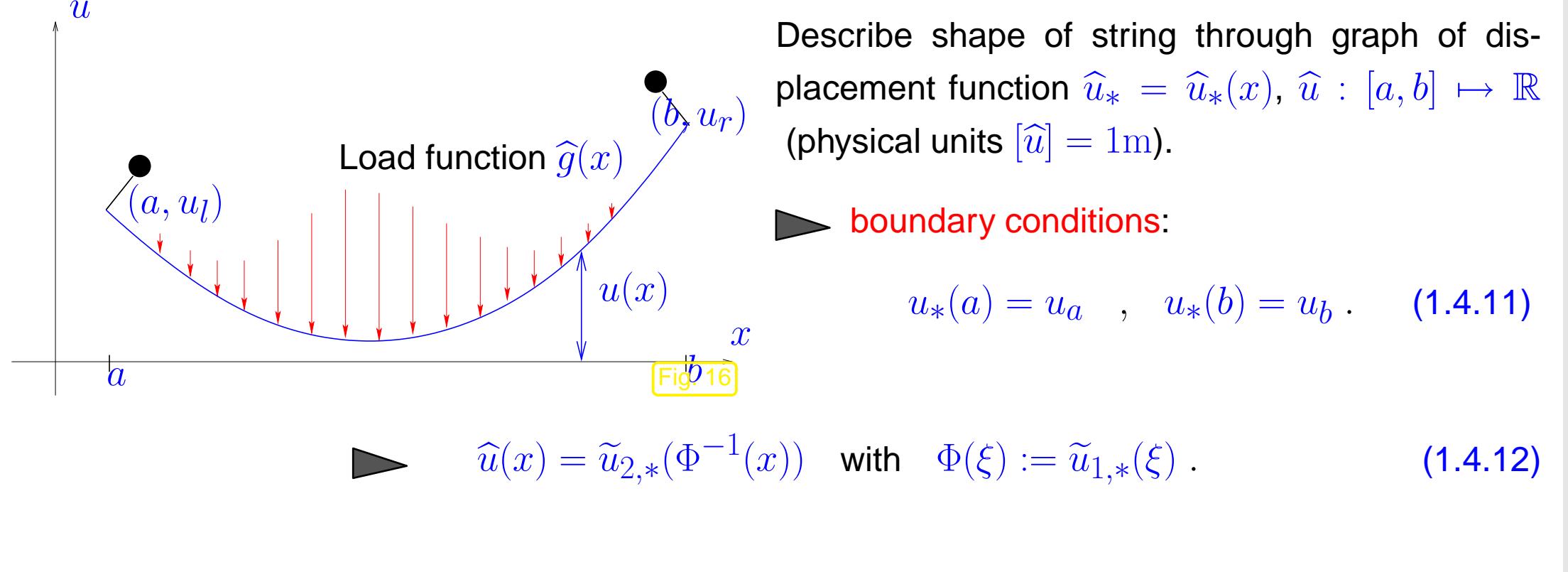
$$\begin{aligned} \widetilde{u}_{2,*} &\in C_{\text{pw}}^1([0,1]) \\ \widetilde{u}_{2,*}(0) = u_a, \quad \widetilde{u}_{2,*}(1) = u_b &: \int_0^1 \widetilde{\kappa}(\xi)\widetilde{u}'_{2,*}(\xi)v'(\xi) d\xi = - \int_0^1 g(\xi)v(\xi) d\xi \quad \forall v \in C_{0,\text{pw}}^1([0,1]) . \end{aligned} \quad (1.4.9)$$

Remark 1.4.10 (Formulation in physical space coordinate).

1.4

Focus: situation with vertical gravitational force, see (1.4.8), (1.4.9)

p. 62



Note: $\xi \mapsto \Phi(\xi)$ is monotone, $\Phi'(\xi) \neq 0$ for all $0 \leq \xi \leq 1$.

By chain rule [19, Thm. 5.1.3]:

$$v(\xi) = \hat{v}(\Phi(\xi)) \Rightarrow v'(\xi) = \frac{d\hat{v}}{dx}(x)\Phi'(\xi) , \quad x := \Phi(\xi) . \quad (1.4.13)$$

Recall: transformation formula for integrals in one dimension (substitution rule, $x := \Phi(\xi)$, “ $dx = \Phi'(\xi)d\xi$ ”):

$$q \in C_{\text{pw}}^0([0, 1]): \int_0^1 q(\xi) d\xi = \int_{a=\Phi(0)}^{b=\Phi(1)} \hat{q}(x) \left| \frac{1}{\Phi'(\Phi^{-1}(x))} \right| dx , \quad \hat{q}(x) := g(\Phi^{-1}(x)) . \quad (1.4.14)$$



\leftarrow (1.4.13) & (1.4.14)

$$\begin{aligned}
 \int_0^1 \tilde{\kappa}(\xi) \tilde{u}'_{2,*}(\xi) v'(\xi) d\xi &= \int_a^b \tilde{\kappa}(\Phi^{-1}(x)) \Phi'(\xi) \frac{d\hat{u}}{dx}(x) \Phi'(\xi) \frac{d\hat{v}}{dx}(x) \frac{1}{|\Phi'(\xi)|} dx \\
 &= \int_a^b \underbrace{\tilde{\kappa}(\Phi^{-1}(x)) |\Phi'(\Phi^{-1}(x))|}_{=: \hat{\sigma}(x)} \frac{d\hat{u}}{dx}(x) \frac{d\hat{v}}{dx}(x) dx, \\
 - \int_0^1 g(\xi) v(\xi) d\xi &= - \int_a^b \underbrace{\frac{f(\Phi^{-1}(x))}{|\Phi'(\Phi^{-1}(x))|}}_{=: \hat{g}(x), [\hat{g}] = N m^{-1}} \hat{v}(x) dx.
 \end{aligned}$$



Linear variational problem in physical space coordinate on spatial domain $\Omega = [a, b]$:

$$\begin{aligned}
 \hat{u}_* \in C_{\text{pw}}^1([a, b]), \quad : \quad & \int_a^b \hat{\sigma}(x) \frac{d\hat{u}_*}{dx}(x) \frac{d\hat{v}}{dx}(x) dx = - \int_a^b \hat{g}(x) \hat{v}(x) dx \quad \forall \hat{v} \in C_{0,\text{pw}}^1([a, b]) \\
 \hat{u}_*(a) = u_a, \quad \hat{u}_*(b) = u_b \quad : \quad & (1.4.15)
 \end{aligned}$$

(assuming $\widehat{\sigma} \in C^1([a, b])$) Two-point BVP

$$(1.4.15) \Rightarrow \begin{cases} \frac{d}{dx} \left(\widehat{\sigma}(x) \frac{d\widehat{u}_*(x)}{dx} \right) = \widehat{g}(x) , \quad a \leq x \leq b , \\ \widehat{u}_*(a) = u_a , \quad \widehat{u}_*(b) = u_b . \end{cases} \quad (1.4.16)$$



1.5 Discretization

Goal: “computation” of a/the solution $\mathbf{u} : [0, 1] \mapsto \mathbb{R}^2$ of

 $\left\{ \begin{array}{l} \text{minimization problem (1.2.17)} \\ \text{variational problem (1.3.7)} \\ \text{two-point BVP (1.3.22) \& (1.2.1)} \end{array} \right.$

a function: infinite amount of information, see [14, Rem. 7.0.3].

! Well, just provide a *formula* for \mathbf{u} (**analytic solution**):



in general elusive
for the above problems

Only option:

Numerical algorithm

Computer

approximate solution

Finitely many floating point operations

Continuous (PDE) model
("∞-dimensional")

Discretization

Discrete model

("finitely many unknowns")

as small as possible
(only a few unknowns)

as accurate as possible
(good approximation)

as faithful as as possible
(structure preserving)

► Numerical algorithms can only operate on discrete models

Remark 1.5.1 (“Physics based” discretization).

Mass-spring model (\rightarrow Sect. 1.2.2) = discretization of the minimization problem (1.2.17) describing the elastic string.

This discretization may be called “physics based”, because it is inspired by the (physical) context of the model.

Note: Other approaches to discretization discussed below will lead to equations resembling the mass-spring model, see 1.5.1.2.



This section will present a few strategies on how to derive discrete models for the problem of computing the shape of an elastic string. The different approaches start from different formulations, some target the minimization problem (1.2.17), or, equivalently, the variational problem (1.3.7), while others tackle the ODE (1.3.22) together with the boundary conditions (1.2.1).

Remark 1.5.2 (Timestepping for ODEs).

For initial value problems for ODEs, whose solutions are functions, too, we also face the problem of discretization: timestepping methods compute a finite number of approximate values of the solutions at discrete instances in time, see [14, Ch. 11].



Remark 1.5.3 (Coefficients/data in procedural form).

For the elastic string mode (\rightarrow Sect. 1.2.3) the stiffness $\kappa(\xi)$, and force field \mathbf{f} may not be available in closed form (as formulas).

Instead they are usually given in **procedural form**:

```
function k = kappa(xi);  
function f = force(xi);
```

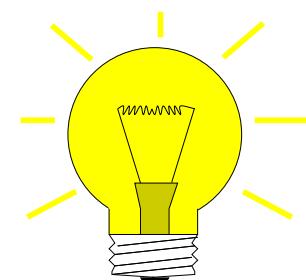
because they may be obtained

- as results of another computation,
- by interpolation from a table.

viable discretizations must be able to deal with data in procedural form!



1.5.1 Galerkin discretization



Simple idea of first step of **Galerkin discretization**

In $\left\{ \begin{array}{l} \text{minimization problem, e.g., (1.2.17)} \\ \qquad \Downarrow \\ \text{variational problem, e.g. (1.3.7)} \end{array} \right.$

replace function space V_0 with
finite dimensional subspace $V_{N,0}$

Note that a subscript tag N distinguishes “discrete functions/quantities”, that is, functions/operators etc. that are associated with a finite dimensional space. In some contexts, N will also be an integer designating the dimension of a finite dimensional space.

Formal presentation: V, V_0 : (affine) function spaces, $\dim V_0 = \infty$,
 $V_N, V_{N,0}$: subspaces $V_N \subset V, V_{N,0} \subset V_0$, $N := \dim V_{N,0}, \dim V_N < \infty$.

Galerkin discretization of minimization problem for **functional** $J : V \mapsto \mathbb{R}$:

Continuous minimization problem

$$u = \underset{v \in V}{\operatorname{argmin}} J(v) . \quad (1.5.4)$$

Galerkin disc.

Discrete minimization problem

$$u_N = \underset{v_N \in V_N}{\operatorname{argmin}} J(v_N) . \quad (1.5.5)$$

Galerkin discretization of abstract (non-linear) variational problem (1.3.12), see Rem. 1.3.10

Continuous variational problem

$$u \in V: \quad a(u; v) = f(v) \quad \forall v \in V_0 . \quad (1.5.6)$$

Galerkin disc.

Discrete variational problem

$$u_N \in V_N: \quad a(u_N; v_N) = f(v_N) \quad \forall v_N \in V_{N,0} . \quad (1.5.7)$$

Terminology: $u_N \in V_N$ satisfying (1.5.5)/(1.5.7) is called a **Galerkin solution** of (1.5.4)/(1.5.6)
 V_N is called the **(Galerkin) trial space**, $V_{N,0}$ is the **(Galerkin) test space**.

Remark 1.5.8 (Relationship between discrete minimization problem and discrete variational problem).

In Sect. 1.3.1 we discovered the **equivalence**

$$\boxed{\begin{array}{l} \text{Continuous minimization problem} \\ (1.5.4) \end{array}}$$



$$\boxed{\begin{array}{l} \text{Continuous variational problem} \\ (1.5.6) \end{array}}$$

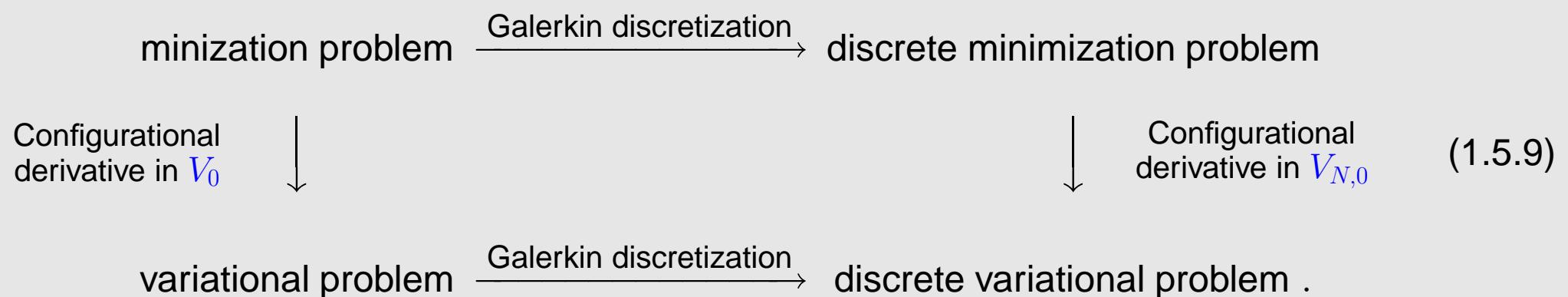
Now it seems that we have *two different* strategies for Galerkin discretization:

1. Galerking discretization via the discrete minimization problem (1.5.5),
2. Galerking discretization based on the discrete variational problem (1.5.7).

However,

the above equivalence extends to the discrete problems!

More precisely, we have the *commuting relationship*:



The commuting diagram means that the same discrete variational problem is obtained no matter whether

- the minimization problem is first restricted to a finite dimensional subspace and the result is converted into a variational problem according to the recipe of Sect. 1.3.1.
- or whether the variational problem derived from the minimization problem is restricted to the subspace.

To see this, understand that the manipulations of Sect. 1.3.1 can be carried out for infinite and finite dimensional function spaces alike.



Remark 1.5.10 (Offset functions and Galerkin discretization).

Often: $V = u_0 + V_0$, with offset function $u_0 \rightarrow$ Rem. 1.3.10

If u_0 is sufficiently simple, we may choose a trial space $V_N = u_0 + V_{N,0}$

➤ Discrete variational problem analogous to (1.3.14)

$$w_N \in V_{N,0}: \quad \mathbf{a}(u_0 + w_N; v_N) = f(v_N) \quad \forall v_N \in V_{N,0} \quad \rightarrow \quad u_N := w_N + u_0 . \quad (1.5.11)$$

In the case of a linear variational problem (\rightarrow Rem. 1.4.4), that is, a bilinear form \mathbf{a} , we have

$$(1.5.11) \Leftrightarrow \mathbf{a}(w_N, v_N) = f(v_N) - \mathbf{a}(u_0, v_N) \quad \forall v_N \in V_{N,0} . \quad (1.5.12)$$

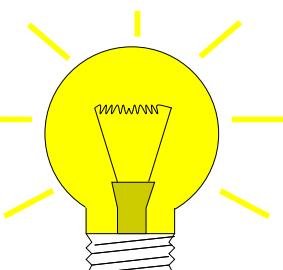
Below we will always make the assumption $V = u_0 + V_0$.



However, a computer is clueless about a concept like “finite dimensional subspace”. What it can process are arrays of floating point numbers.

Idea:

- choose **basis** $\mathfrak{B}_N = \{b_N^1, \dots, b_N^N\}$ of $V_{N,0}$: $V_{N,0} = \text{Span } \{\mathfrak{B}_N\}$
- insert basis representation into (1.5.5)/(1.5.7)


$$v_N \in V_N \Rightarrow v_N = \nu_1 b_N^1 + \dots + \nu_N b_N^N , \quad \nu_i \in \mathbb{R} . \quad (1.5.13)$$

Remark 1.5.14 (Ordered basis of test space).

Once we have chosen a basis \mathcal{B} and ordered it, as already indicated in the notation above, the test space $V_{N,0}$ can be identified with \mathbb{R}^N : a coefficient vector provides a *unique* characterization of a function $\in V_{N,0}$ (basis property). \triangle

Discrete minimization problem

$$u_N = \operatorname{argmin}_{v_N \in V} J(v_N) . \quad (1.5.5)$$

Basis
representation

Minimization problem on \mathbb{R}^N

$$\vec{\mu} = \operatorname{argmin}_{\vec{\nu} \in \mathbb{R}^N} F(\vec{\nu}) , \quad (1.5.15)$$

$$F(\vec{\nu}) := J(u_0 + \nu_1 b_N^1 + \cdots + \nu_N b_N^N) .$$

amenable to classical optimization
techniques

notation: $\vec{\nu}, \vec{\mu} \doteq$ vectors of coefficients $(\nu_i)_{i=1}^N, (\mu_i)_{i=1}^N$, in basis representation of functions $v_N, u_N \in V_N$ according to (1.5.13).

Discrete variational problem

$$u_N \in V_N: \quad \mathbf{a}(u_N; v_N) = f(v_N) \\ \forall v_N \in V_{N,0} . \quad (1.5.7)$$

Basis
representation

System of equations

$$\mathbf{a}(u_0 + \sum_{j=1}^N \mu_j b_N^j; b_N^k) = f(b_N^k) \\ \forall k = 1, \dots, N . \quad (1.5.16)$$



use techniques for linear/non-linear
systems of equations, see [14, Ch. 2],
[14, Ch. 3].

The choice of the basis \mathfrak{B} has no impact on the (set of) Galerkin solutions of (1.5.7)!

Below, we apply Galerkin approaches to

- (1.4.15) as an example for the treatment of a *linear* variational problem:

$$\begin{aligned} u \in C_{\text{pw}}^1([a, b]), \\ u(a) = u_a, \quad u(b) = u_b \end{aligned} : \quad \int_a^b \sigma(x) \frac{du}{dx}(x) \frac{dv}{dx}(x) dx = - \int_a^b g(x)v(x) dx \quad \forall v \in C_{0,\text{pw}}^1([a, b]) .$$

(1.4.15)

Here:

spatial domain $\Omega = [a, b]$, linear offset function $u_0(x) = \frac{b-x}{b-a}u_a + \frac{x-a}{b-a}u_b$,
 function space $V_0 = C_{0,\text{pw}}^1([a, b])$.

- (1.3.7) to demonstrate its use in the case of a non-linear variational equation:

$$\begin{aligned} \mathbf{u} \in C_{\text{pw}}^1([0, 1]) \\ \mathbf{u}(0), \mathbf{u}(1) \text{ from (1.2.1)} \end{aligned} : \quad \int_0^1 \frac{\kappa(\xi)}{L} \left(\|\mathbf{u}'(\xi)\| - L \right) \frac{\mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi)}{\|\mathbf{u}'(\xi)\|} - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) d\xi = 0$$

$\forall \mathbf{v} \in (C_{0,\text{pw}}^1([0, 1]))^2 .$ (1.3.7)

Here: parameter domain $\Omega = [0, 1]$, linear offset function $\mathbf{u}_0(\xi) = \xi \mathbf{u}(0) + (1 - \xi) \mathbf{u}(1)$,
 function space $V_0 = (C_{0,\text{pw}}^1([a, b]))^2$.

1.5.1.1 Spectral Galerkin scheme

A simple function space (widely used for interpolation, see [14, Ch. 8], and approximation, see [14, Sec. 8.4]): for interval $\Omega \subset \mathbb{R}$

$$V_{N,0} = \mathcal{P}_p(\mathbb{R}) \cap C_0^0(\Omega)$$

$\hat{=}$ space of univariate **polynomials** of degree $\leq p$ vanishing at endpoints of Ω ,

(1.5.17)

► $N := \dim V_N = p - 1$

[14, Sect. 8.1] for more information.

Obvious: choice (1.5.17) guarantees $V_N \subset C_{\text{pw},0}^1(\Omega)$ (even $V_{N,0} \subset C^\infty(\Omega)$)

Please note that $V_{N,0}$ is a space of *global* polynomials on Ω .

Example 1.5.18 (Spectral Galerkin discretization of linear variational problem).

Targetted: linear variational problem (1.4.15) with

- $a = 0, b = 1 \Rightarrow$ domain $\Omega =]0, 1[$,
- constant coefficient function $\sigma \equiv 1$,
- load $g(x) = -4\pi(\cos(2\pi x^2) - 4\pi x^2 \sin(2\pi x^2))$,
- boundary values $u_a = u_b = 0$.

► $u(x) = \sin(2\pi x^2), \quad 0 < x < 1$.
because $\frac{d^2u}{dx^2}(x) = g(x)$.

► Concrete variational problem

$$u \in C_{0,\text{pw}}^1([0, 1]): \quad \int_0^1 \frac{du}{dx}(x) \frac{dv}{dx}(x) dx = - \int_0^1 g(x)v(x) dx \quad \forall v \in C_{0,\text{pw}}^1([0, 1]). \quad (1.5.19)$$

Polynomial spectral Galerkin discretization, degree $p \in \{4, 5, 6\}$.

Plots of approximate/exact solutions

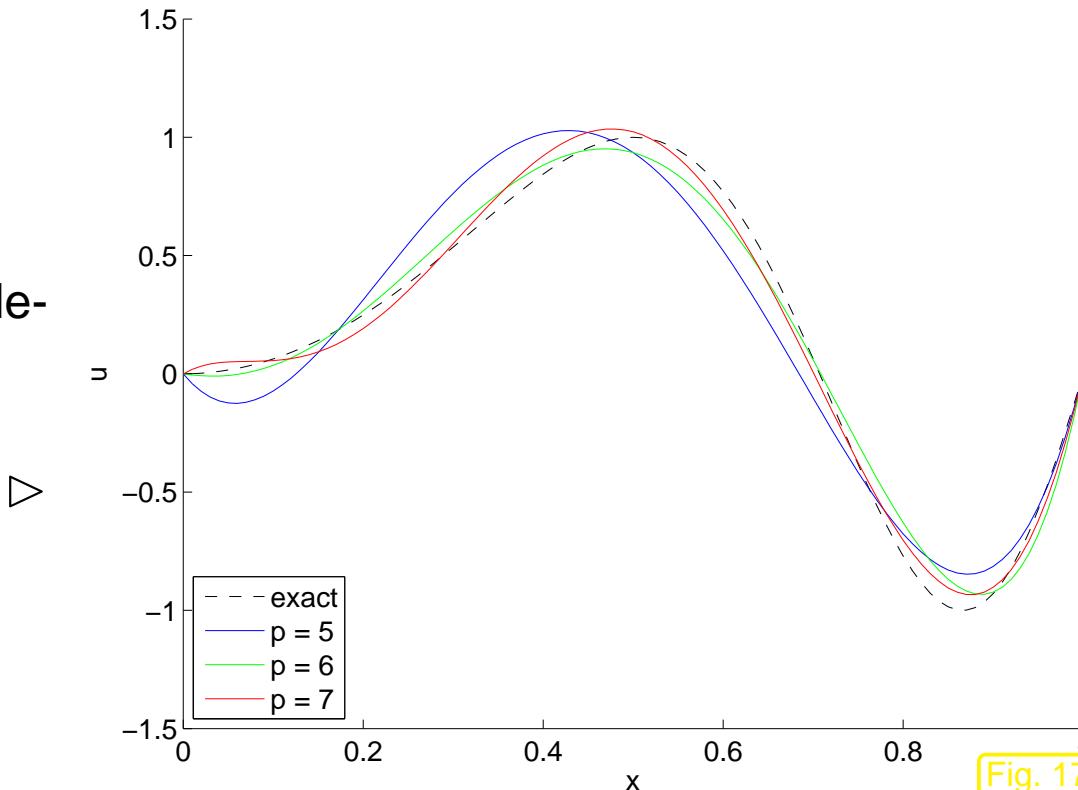


Fig. 17

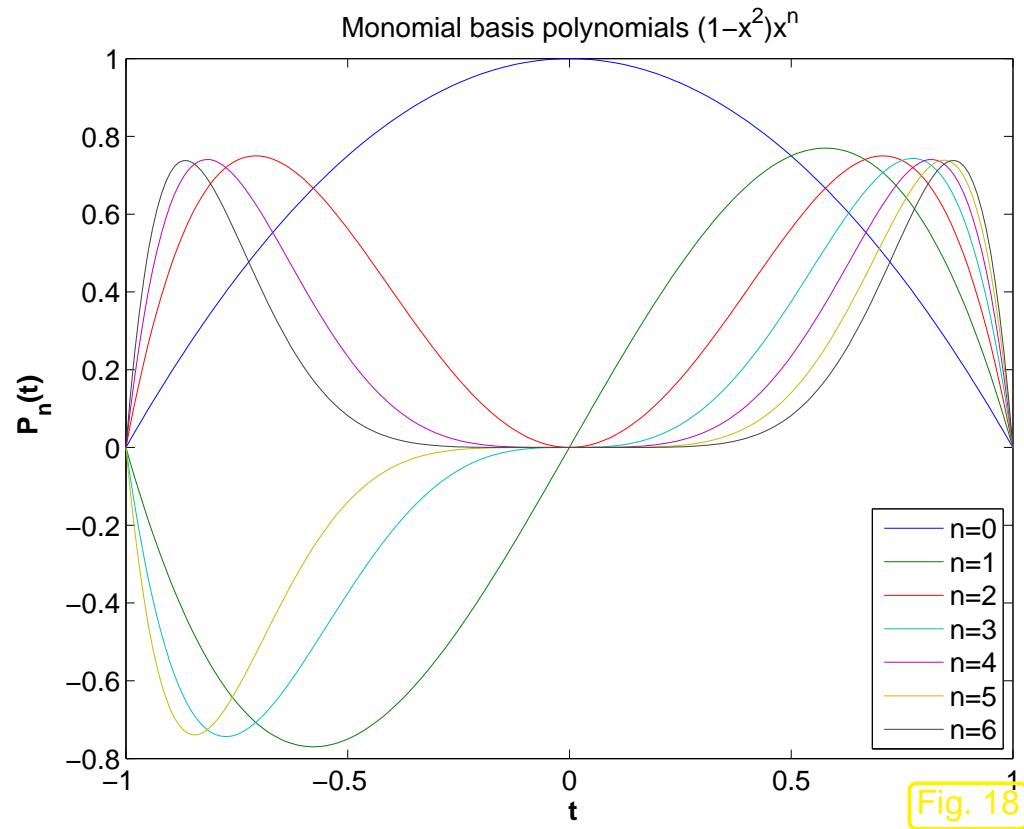


Remark 1.5.20 (Choice of basis for polynomial spectral Galerkin methods).

Sought: (ordered) basis of $V_{N,0} := C_0^1([-1, 1]) \cap \mathcal{P}_p(\mathbb{R})$

① “Tempting”: monomial-type basis

$$V_{N,0} = \text{Span} \left\{ 1 - x^2, x(1 - x^2), x^2(1 - x^2), \dots, x^{p-2}(1 - x^2) \right\}. \quad (1.5.21)$$



▷ Monomial basis polynomials

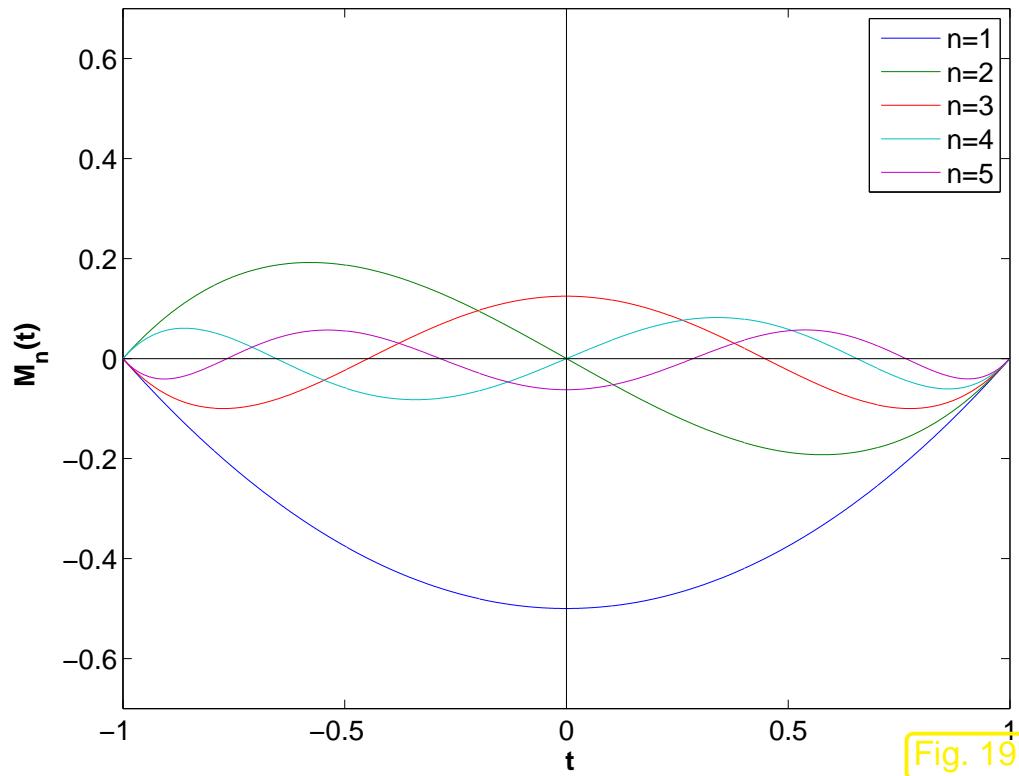
Beware: ill-conditioned !

② “Popular”: integrated Legendre polynomials

$$V_{N,0} = \text{Span} \left\{ x \mapsto M_n(x) := \int_{-1}^x P_n(\tau) d\tau, \quad n = 1, \dots, p-1 \right\}, \quad (1.5.22)$$

where $P_n \hat{=} n$ -th Legendre polynomial.

Integrated Legendre polynomials



◀ integrated Legendre polynomials
 M_1, \dots, M_5

Fig. 19

Definition 1.5.23 (Legendre polynomials). → [14, Def. 10.4.2]

The n -th **Legendre polynomial** P_n , $n \in \mathbb{N}_0$, is defined by (Rodriguez formula)

$$P_n(x) := \frac{1}{n!2^n} \frac{d^n}{dx^n} [(x^2 - 1)^n].$$

Legendre polynomials

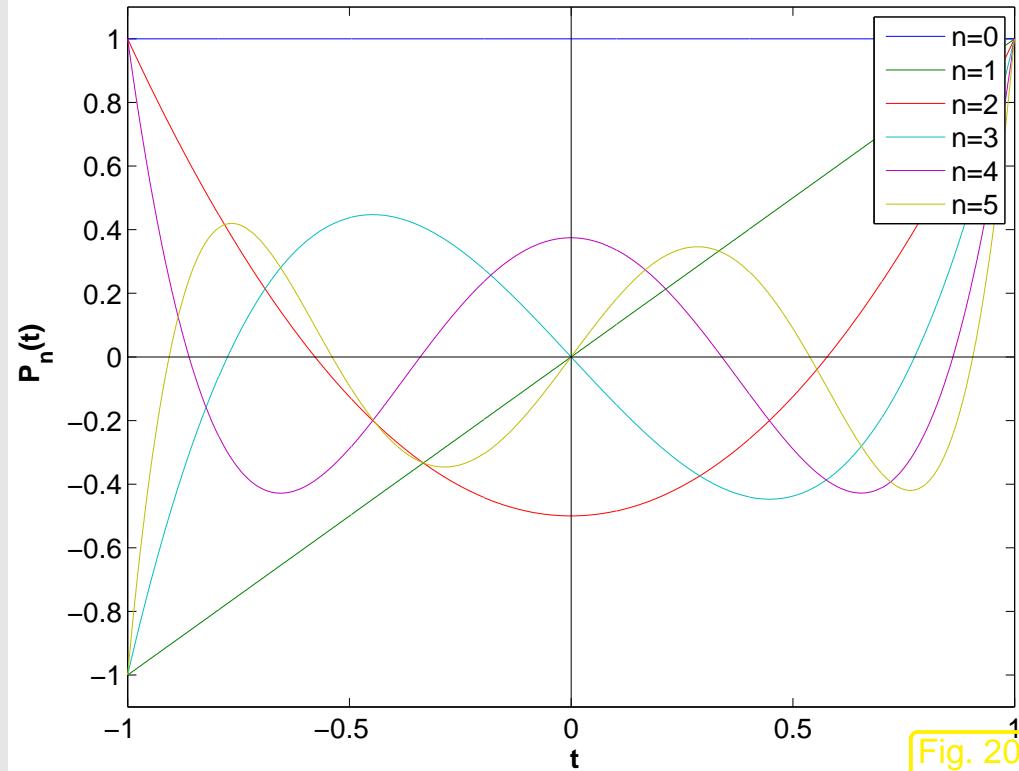


Fig. 20

Legendre polynomials P_0, \dots, P_5

$$P_0(x) = 1 ,$$

$$P_1(x) = x ,$$

$$P_2(x) = \frac{3}{2}x^2 - \frac{1}{2} ,$$

$$P_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x ,$$

$$P_4(x) = \frac{35}{8}x^4 - \frac{15}{4}x^2 + \frac{3}{8} .$$

Some facts about Legendre polynomials:

- Symmetry:

$$P_n \text{ is } \begin{cases} \text{even} \\ \text{odd} \end{cases} \quad \text{for} \quad \begin{cases} \text{even } n \\ \text{odd } n \end{cases}, \quad P_n(1) = 1 , \quad P_n(-1) = (-1)^n . \quad (1.5.24)$$

- Orthogonality

$$\int_{-1}^1 P_n(x) P_m(x) dx = \begin{cases} \frac{2}{2n+1} & , \text{ if } m = n , \\ 0 & \text{else.} \end{cases} \quad (1.5.25)$$

• 3-term recursion

$$P_{n+1}(x) := \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x) , \quad P_0 := 1 , \quad P_1(x) := x . \quad (1.5.26)$$

This formula paves the way for the efficient evaluation of all Legendre polynomials at many (quadrature) points, see [14, Code 10.4.2].

• Representation of derivatives and primitives, cf. Code 1.5.28:

$$P_n(x) = (\frac{d}{dx} P_{n+1}(x) - \frac{d}{dx} P_{n-1}(x)) / (2n+1) , \quad n \in \mathbb{N} , \quad (1.5.27)$$

► $M_n(x) = \frac{1}{2n+1} (P_{n+1}(x) - P_{n-1}(x)) \quad \text{and} \quad \frac{dM_n}{dx} = P_n . \quad (1.5.28)$



Code 1.5.29: Computation of (integrated) Legendre polynomials using (1.5.26) and (1.5.28)

```
1 function [V,M] = intlegpol(n,x)
2 % Computes values of the first n+1 Legendre polynomials (returned in matrix V)
3 % and the first n-1 integrated Legendre polynomials (returned in matrix M)
4 % in the points xj passed in the row vector x.
```

```

5 % Uses the recursion formulas (1.5.26) and (1.5.28)
6 V = ones( size(x) ); V = [V; x];
7 for j=1:n-1, V = [V; ( (2*j+1)/(j+1) ).*x.*V(end, : ) -
8   j/(j+1)*V(end-1, : )]; end
9 M = diag(1./(2*(1:n-1)+1))* (V(3:n+1, : ) - V(1:n-1, : ) );

```

Remark 1.5.30 (Transformation of basis functions).

On a “general domain $\Omega = [a, b]$ ”, we obtain the basis function by a so-called affine transformation of the basis functions on $[0, 1]$, cf. [14, Rem. 10.1.1], e.g., in the case of integrated Legendre polynomials as basis functions on $\Omega = [a, b]$ we use the basis functions

$$b_N^i(x) = M_i \left(2 \frac{x-a}{b-a} - 1 \right), \quad a \leq x \leq b. \quad (1.5.31)$$

Note the effect of this transformation on the derivative (chain rule!):

$$\frac{db_N^i}{dx}(x) = \frac{dM_i}{dx} \left(2 \frac{x-a}{b-a} - 1 \right) \cdot \frac{2}{b-a} = P_i \left(2 \frac{x-a}{b-a} - 1 \right) \cdot \frac{2}{b-a}. \quad (1.5.32)$$

Remark 1.5.33 (Spectral Galerkin discretization with quadrature).

Consider the linear variational problem, cf. (1.4.15),

$$u \in C_{0,\text{pw}}^1([a, b]): \int_a^b \sigma(x) \frac{du}{dx}(x) \frac{dv}{dx}(x) dx = \int_a^b g(x)v(x) dx \quad \forall v \in C_{0,\text{pw}}^1([a, b]) . \quad (1.5.34)$$

Assume: σ, g only given in procedural form, see Rem. 1.5.3.

- Analytic evaluation of integrals becomes impossible even if u, v polynomials !

Only remaning option:

Numerical quadrature, see [14, Ch. 10]

- Replace integral with m -point quadrature formula on $[a, b]$, $m \in \mathbb{N} \rightarrow$ [14, Sect. 10.1]:

$$\int_a^b f(t) dt \approx Q_n(f) := \sum_{j=1}^m \omega_j^m f(\zeta_j^m) . \quad (1.5.35)$$

ω_j^n : quadrature weights , ζ_j^n : quadrature nodes $\in [a, b]$.



(1.5.34) > discrete variational problem with quadrature:

$$u_N \in V_N: \sum_{j=1}^m \omega_j^m \sigma(\zeta_j^m) \frac{du_N}{dx}(\zeta_j^m) \frac{dv_N}{dx}(\zeta_j^m) = \sum_{j=1}^m \omega_j^m g(\zeta_j^m) v(\zeta_j^m) \quad \forall v \in V_N . \quad (1.5.36)$$

Popular (global) quadrature formulas: **Gauss quadrature** → [14, Sect. 10.4]

Important: Accuracy of quadrature formula and computational cost (no. m of quadrature nodes) have to be balanced.



Remark 1.5.37 (Implementation of spectral Galerkin discretization for linear 2nd-order two-point BVP).

Setting:

- linear variational problem (1.5.34) $\Rightarrow u_0 = 0$,
- coefficients σ, g in procedural form, see Rem. 1.5.3,
- approximation of integrals by p -point Gaussian quadrature formula,
- polynomial spectral Galerkin discretization, degree $\leq p, p \geq 2$,
- basis \mathfrak{B} : integrated Legendre polynomials, see (1.5.22):

$$V_{N,0} = \text{Span} \{M_n, n = 1, \dots, p - 1\}, \quad M_n \hat{=} \text{integrated Legendre polynomials}.$$

Trial expression, cf. (1.5.13)

$$u_N = \mu_1 M_1 + \mu_2 M_2 + \dots + \mu_N M_N, \quad \mu_i \in \mathbb{R}, \quad N := p - 1.$$

Note: by definition $\frac{d}{dx} M_n = P_n$.

From (1.5.36) with (1.5.37)

$$\sum_{j=1}^m \omega_j^m \sigma(\zeta_j^m) \sum_{l=1}^N \mu_l P_l(\zeta_j^m) P_k(\zeta_j^m) = \underbrace{\sum_{j=1}^m \omega_j^m g(\zeta_j^m) M_k(\zeta_j^m)}_{=: \varphi_k}, \quad k = 1, \dots, N. \quad (1.5.38)$$

↑

$$\sum_{l=1}^N \left(\sum_{j=1}^m \omega_j^m \sigma(\zeta_j^m) P_l(\zeta_j^m) P_k(\zeta_j^m) \right) \mu_l = \varphi_k, \quad k = 1, \dots, N. \quad (1.5.39)$$

↑

$$\boxed{\mathbf{A}\vec{\mu} = \vec{\varphi}}$$

with

$$(\mathbf{A})_{kl} := \sum_{j=1}^m \omega_j^m \sigma(\zeta_j^m) P_l(\zeta_j^m) P_k(\zeta_j^m), \quad k, l = 1, \dots, N, \quad (1.5.40)$$

$$\vec{\mu} = (\mu_l)_{l=1}^N \in \mathbb{R}^N, \quad \vec{\varphi} = (\varphi_k)_{k=1}^N \in \mathbb{R}^N.$$

A linear system of equations !

The Galerkin discretization of a *linear* variational problem leads to a *linear* system of equations.

Code 1.5.41: Polynomial spectral Galerkin solution of (1.5.34)

```
1 function u = lin2pbvpspecgal(sigma,g,N,x)
```

```
2 % Polynomial spectral Galerkin discretization of linear 2nd-order two-point BVP
```

```

3   %  $-\frac{d}{dx}(\sigma(x)\frac{du}{dx}) = g(x)$ ,  $u(0) = u(1) = 0$ 
4   % on  $\Omega = [0, 1]$ . Trial space of dimension  $N$ .
5   % Values of approximate solution in points  $x_j$  are returned in the row vector  $u$ 
6 m = N+1;                                % Number of quadrature nodes
7 [zeta,w] = gaussquad(m); % Obtain Gauss quadrature nodes w.r.t  $[-1, 1]$ 
8 % Compute values of (integrated) Legendre polynomials at Gauss nodes
9 [V,M] = intlegpol(N+1,zeta');
10 omega = w' .* sigma((zeta'+1)/2)*2;        % Modified quadrature weights
11 A = V(2:N+1,:)*diag(omega)*V(2:N+1,:)''; % Assemble Galerkin matrix
12 phi = M*(0.5*w' .*g((zeta'+1)/2))';       % Assemble right hand side
13 mu = A\phi;                                % Solve linear system
14 % Compute values of integrated Legendre polynomials at output points
15 [V,M] = intlegpol(N+1,2*x-1); u = mu'*M;

```

Code 1.5.42: MATLAB driver script creating plots of Ex. 1.5.18

```

1 % MATLAB script: Driver routine for polynomial spectral Galerkin
2 % discretization
3 clear all;
4 % Coefficient functions (function handles, see MATLAB help)
5 sigma = @(x) ones(size(x));
6 g = @(x) -4*pi*(cos(2*pi*x.^2)-4*pi*x.^2.*sin(2*pi*x.^2));
7 % Evaluation points
8 x = 0:0.01:1;
9 % Computation with trial space of dimension 4,5,6
10 N = 4; U = [lin2pbvpspecgal(sigma,g,N,x);
11 lin2pbvpspecgal(sigma,g,N+1,x); lin2pbvpspecgal(sigma,g,N+2,x)];

```

```

9 % Graphical output
10 figure('name','Polynomial spectral Galerkin');
11 plot(x,U); hold on;
12 plot(x,sin(2*pi*x.^2),'g--','linewidth',2);
13 xlabel('{\bf x}', 'fontsize',14);
14 ylabel('{\bf u}', 'fontsize',14);
15 legend('N=4','N=5','N=6','u(x)', 'location','southwest');
16 print -depsc2 '../.../Slides/NPDEPics/specgallinsol.eps';

```



Example 1.5.43 (Implementation of spectral Galerkin discretization for elastic string problem).

Targetted: *non-linear* variational equation on domain $\Omega = [0, 1]$

$$\int_0^1 \frac{\kappa(\xi)}{L} \left(1 - \frac{L}{\|\mathbf{u}'(\xi)\|} \right) \mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi) - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi = 0 \quad \forall \mathbf{v} \in (C_{0,\text{pw}}^1([0, 1]))^2. \quad (1.3.7)$$

- Data κ, \mathbf{f} given in procedural form, see Rem. 1.5.3.
- Spectral Galerkin discretization, basis $\mathfrak{B} = \{M_n\}_{n=1}^K$, $K \in \mathbb{N}$, consists of integrated Legendre polynomials, see (1.5.22) \Rightarrow basis representation, cf. (1.5.37)

$$\mathbf{u}_N(\xi) = \underbrace{\mathbf{u}(0)(1 - \xi) + \mathbf{u}(1)\xi}_{=: \mathbf{u}_0(\xi) \text{ (offset function)}} + \begin{pmatrix} \mu_1 \\ \mu_{K+1} \end{pmatrix} M_1(\xi) + \cdots + \begin{pmatrix} \mu_K \\ \mu_{2K} \end{pmatrix} M_K(\xi) . \quad (1.5.44)$$

- Approximate evaluation of integrals by m -point Gaussian quadrature on $[0, 1]$, $m := K + 1$ below: nodes ζ_j , weights ω_j , $j = 1, \dots, m$.

In analogy to (1.5.38) we arrive at the *non-linear system of equations*: $(M'_k = P_k!)$

$$\sum_{j=1}^m s_j \left(b - a + \sum_{l=1}^K \mu_l P_l(\zeta_j) \right) \cdot P_k(\zeta_j) = \sum_{j=1}^m \omega_j f_1(\zeta_j) \cdot M_k(\zeta_j) , \quad k = 1, \dots, K ,$$

$$\sum_{j=1}^m s_j \left(u_b - u_a + \sum_{l=1}^K \mu_{K+l} P_l(\zeta_j) \right) \cdot P_k(\zeta_j) = \sum_{j=1}^m \omega_j f_2(\zeta_j) \cdot M_k(\zeta_j) , \quad k = 1, \dots, K ,$$

with $s_j := \omega_j \kappa(\zeta_j) \left(\frac{1}{L} - \frac{1}{\|\mathbf{u}'_N(\zeta_j)\|} \right) .$

Code 1.5.47: Polynomial spectral Galerkin discretization of elastic string variational problem

```

1 function [vu,figsol] = stringspecgal(kappa,f,L,u0,u1,K,xi,tol)
2 % Solving the non-linear variational problem (1.3.7) for the elastic string by
3 % means of polynomial
4 % spectral Galerkin discretization based on K integrated Legendre polynomials.
5 % Approximate
6 % evaluation of integrals by means of Gaussian quadrature.
7 % kappa, f are handles of type @(xi) providing the coefficient function
8 % κ and the force field f. The column vectors u0 and u1 pass the
9 % pinning points. M is the number of mesh cells, tol specifies the tolerance
10 % for the
11 % fixed point iteration. return value: 2 × length(xi)-matrix of node
12 % positions for curve parameter values passed in the row vector xi.
13 if (nargin < 8), tol = 1E-2; end
14 m = K+1; % Number of quadrature nodes
15 [zeta,w] = gaussquad(m); % Obtain Gauss quadrature nodes w.r.t [-1,1]
16 % Compute values of (integrated) Legendre polynomials at Gauss nodes and
17 % evaluation points
18 [V,M] = intlegpol(K+1,zeta');
19 [Vx,Mx] = intlegpol(K+1,2*xi-1); Mx = [1-xi;Mx;xi]; %
20 % Compute right hand side based on m-point Gaussian quadrature on [0,1].
21 force = f((zeta'+1)/2); phi = M*(0.5*[w';w'].*force)';
22 sv = kappa((zeta'+1)/2); % Values of coefficient function κ at Gauss
23 % points in [0,1].
24 % mu is an 2 × (K + 2)-matrix, containing the vectorial basis expansion
25 % coefficients
26 % of uN. The first and last column are contributions of the two functions

```

```

21 %  $\xi \mapsto (1 - \xi)$  and  $\xi \mapsto \xi$ , which represent the offset function.
22 % Initial guess for fixed point iteration: straight string
23 mu = [u0, zeros(2,K), u1];
24 figsol = figure; hold on;
25 for k=1:100 % loop for fixed point iteration, maximum 100 iterations
26 % Plot shape of string
27 vu = mu*Mx; plot(vu(1,:),vu(2,:),'--g'); drawnow;
28 title(sprintf('K = %d, iteration # %d',K,k));
29 xlabel('{\bf x_1}'); ylabel('{\bf x_2}');
30 % Compute values of derivatives of  $u_N$  and  $\|u'_N\|$  at Gauss points
31 up = mu(:,2:K)*V(2:K,:)+repmat(u1-u0,1,m);
32 lup = sqrt(up(1,:).^2 + up(2,:).^2);
33 s = 0.5*(w').*sv.* (1/L - 1./lup); % Initialization of  $s_j$ 
34 % Modification of right hand side due to offset function
35 phi1 = phi(:,1) + (2*(u1(1)-u0(1))*V(2:K+1,:)*s');
36 phi2 = phi(:,2) + (2*(u1(2)-u0(2))*V(2:K+1,:)*s');
37 % Assemble  $K \times K$ -matrix blocks  $R$  of linear system
38 R = 4*V(2:K+1,:)*diag(s)*V(2:K+1,:)';
39 mu_new = [u0, [(R\phi1)'; (R\phi2)'], u1];
40 % Check simple termination criterion for fixed point iteration.
41 if (norm(mu_new - mu,'fro') < tol*norm(mu_new,'fro')/K)
42 vu = mu*Mx; fig = plot(vu(1,:),vu(2,:),'r--');
43 legend(fig,'spectral Galerkin
        solution','location','southeast'); break; end

```

```
44 mu = mu_new;  
45 end
```

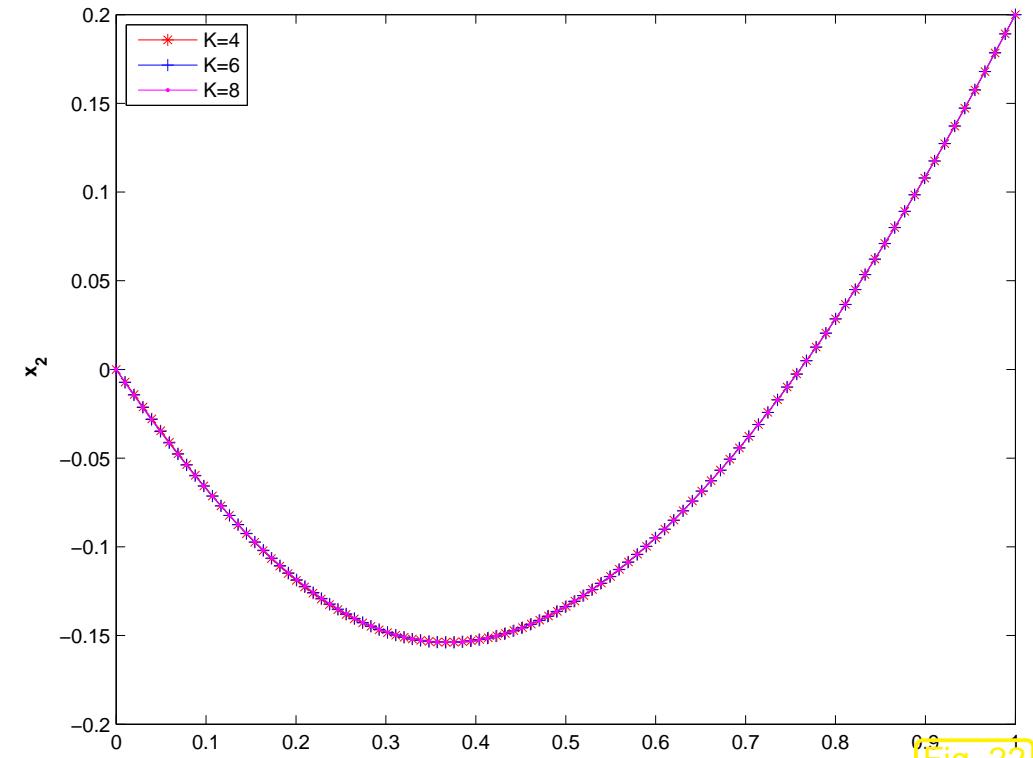
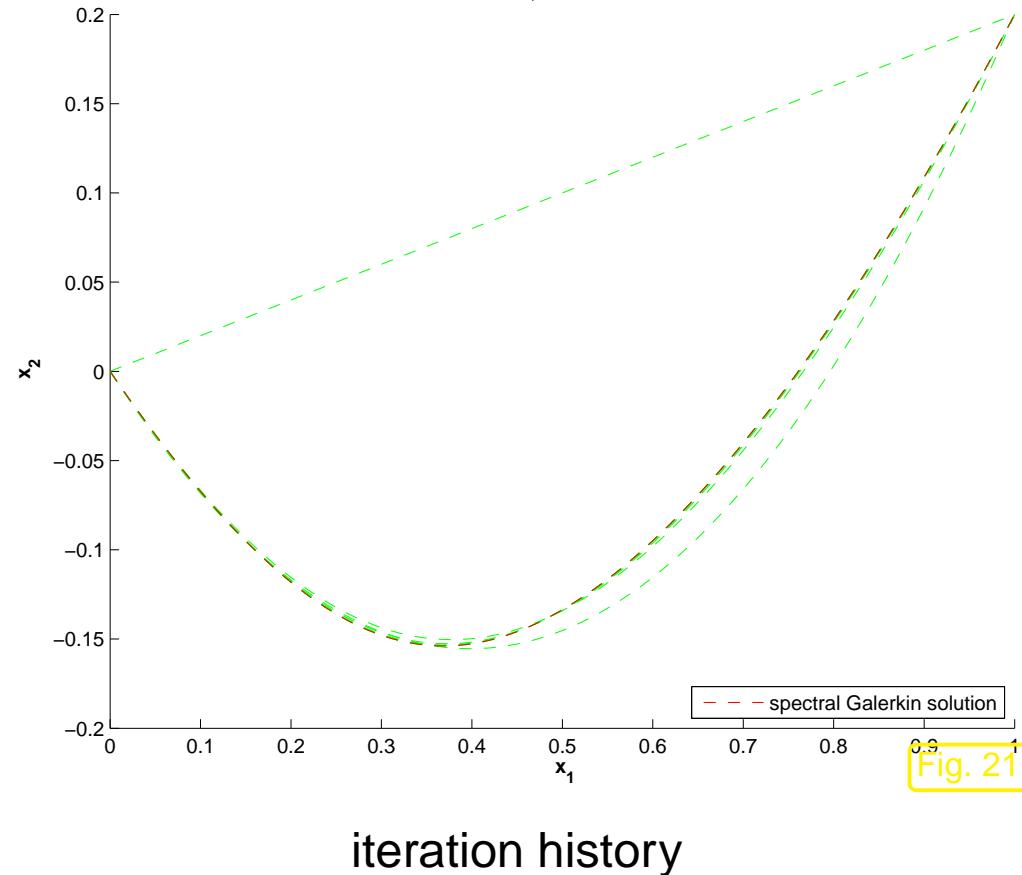


Example 1.5.48 (Spectral Galerkin discretization for elastic string simulation).

Test of polynomial spectral Galerkin method for elastic string problem, algorithm of Ex. 1.5.43, Code 1.5.
with

- pinning positions $\mathbf{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\mathbf{u}(1) = \begin{pmatrix} 1 \\ 0.2 \end{pmatrix}$,
- equilibrium length $L = 0.5$,
- constant coefficient function $\kappa \equiv 1\text{N}$,
- gravitational force field $\mathbf{f}(\xi) = -\begin{pmatrix} 0 \\ 2 \end{pmatrix}$.

K = 8, iteration #7



1.5.1.2 Linear finite elements

Two ways to approximate functions by polynomials:

global polynomials
[14, Ch. 8]

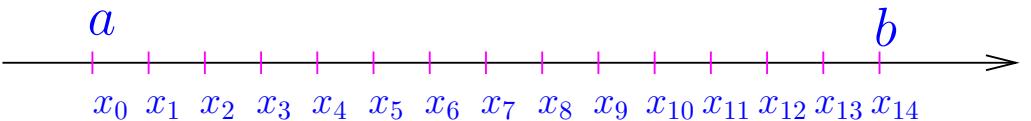
↔
piecewise polynomials
[14, Ch. 9]

The spectral polynomial Galerkin approach presented in Sect. 1.5.1.1 relies on global polynomials.
Now let us examine the use of *piecewise polynomials*.

Preliminaries: piecewise polynomials have to be defined w.r.t. partitioning of the domain $\Omega \subset \mathbb{R}$

➤ $\Omega = [a, b]$ equipped with **nodes** ($M \in \mathbb{N}$)

$\mathcal{X} := \{a = x_0 < x_1 < \dots < x_{M-1} < x_M = b\}$.



➤ **mesh/grid**

$\mathcal{M} := \{]x_{j-1}, x_j[: 1 \leq j \leq M\}$.

Special case:

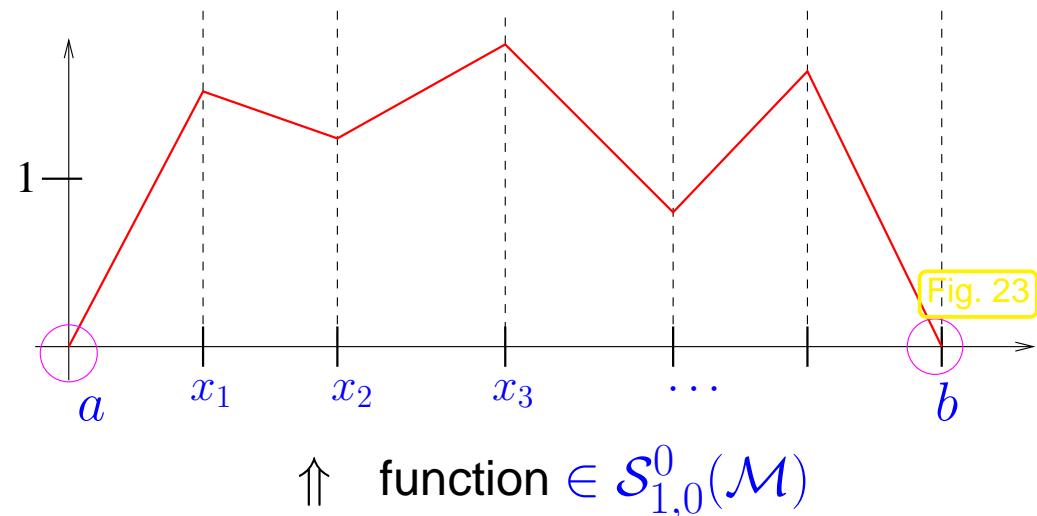
equidistant mesh: $x_j := a + jh$, $h := \frac{b-a}{M}$.

☞ $[x_{j-1}, x_j]$, $j = 1, \dots, M$, $\hat{=}$ **cells** of \mathcal{M} ,

cell size $h_j := |x_j - x_{j-1}|$, $j = 1, \dots, M$

meshwidth $h_{\mathcal{M}} := \max_j |x_j - x_{j-1}|$

Recall from Sect. 1.3.2: merely continuous, piecewise C^1 trial and test functions provide valid trial/test functions!



Simplest choice for test space

$$V_N = \mathcal{S}_{1,0}^0(\mathcal{M})$$

$$:= \left\{ v \in C^0([0, 1]): v|_{[x_{i-1}, x_i]} \text{ linear, } i = 1, \dots, M, v(a) = v(b) = 0 \right\}$$

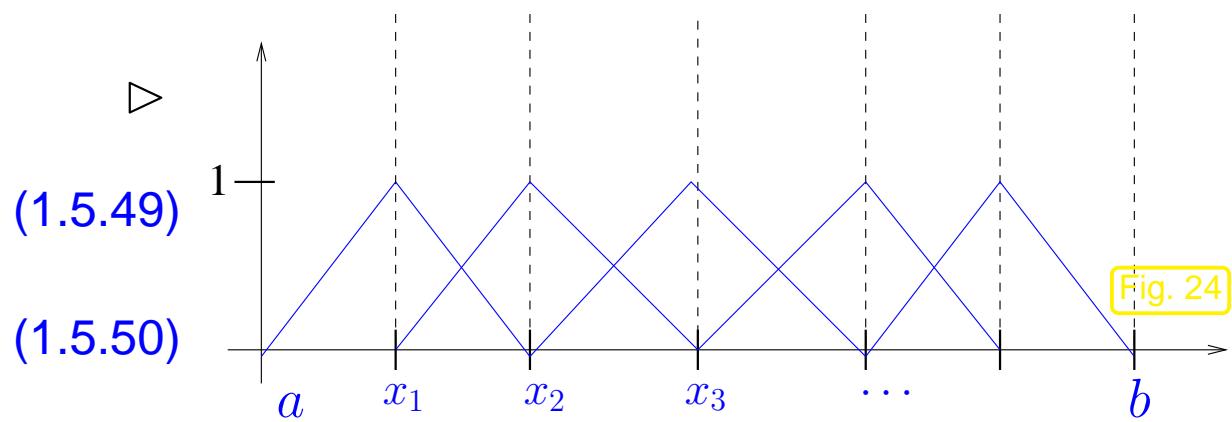
$$N := \dim V_N = M - 1$$

Choice of (ordered) basis \mathfrak{B}_N of V_N ?

1D “tent functions”

$$\mathfrak{B} = \{b_N^1, \dots, b_N^{M-1}\},$$

$$b_N^j(x_i) = \delta_{ij} := \begin{cases} 1 & , \text{if } i = j , \\ 0 & , \text{if } i \neq j , \end{cases}$$



►
$$\frac{db_N^j}{dx}(x) = \begin{cases} \frac{1}{h_j} & , \text{ if } x_{j-1} \leq x \leq x_j , \\ -\frac{1}{h_{j+1}} & , \text{ if } x_j < x \leq x_{j+1} , \\ 0 & \text{elsewhere.} \end{cases} \quad (\text{piecewise derivative!}) \quad (1.5.51)$$

Remark 1.5.52 (Benefit of variational formulation of BVPs).

The possibility of using simple piecewise linear trial and test functions is a clear benefit of the variational formulation that can accommodate merely piecewise continuously differentiable functions, see Sect. 1.3.2.

Below, in Sect. 1.5.2 we will learn about a method that targets the strong form of the 2-point BVP and, thus, has to impose more regularity on the trial functions.



① simplest case: linear variational problem with constant stiffness coefficient

$$u \in C_0^1([a, b]): \quad \int_a^b \frac{du}{dx}(x) \frac{dv}{dx}(x) dx = \int_a^b g(x)v(x) dx \quad \forall v \in C_0^1([a, b]) .$$

Discrete variational problem with $u_N = \mu_1 b_N^1 + \dots + \mu_N b_N^N$:

$$\int_a^b \sum_{l=1}^N \mu_l \frac{db_N^l}{dx}(x) \frac{db_N^k}{dx}(x) dx = \int_a^b g(x) b_N^k(x) dx \quad k = 1, \dots, N .$$

\Updownarrow

$$\sum_{l=1}^N \left(\int_a^b \frac{db_N^l}{dx}(x) \frac{db_N^k}{dx}(x) dx \right) \mu_l = \underbrace{\int_a^b g(x) b_N^k(x) dx}_{=: \varphi_k}, \quad k = 1, \dots, N .$$

\Updownarrow

$\mathbf{A}\vec{\mu} = \vec{\varphi}$ with $(\mathbf{A})_{kl} := \int_a^b \frac{db_N^l}{dx}(x) \frac{db_N^k}{dx}(x) dx, \quad k, l = 1, \dots, N ,$

$$\vec{\mu} = (\mu_l)_{l=1}^N \in \mathbb{R}^N , \quad \vec{\varphi} = (\varphi_k)_{k=1}^N \in \mathbb{R}^N .$$

1.5

A **linear system of equations**, cf. Rem. 1.5.37!

p. 100

- ▷ system matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{M-1, M-1}$, $a_{ij} := \int_a^b \frac{db_N^i}{dx}(x) \frac{db_N^j}{dx}(x) dx$, $1 \leq i, j \leq N$
- ▷ r.h.s. vector $\vec{\varphi} \in \mathbb{R}^{M-1}$, $\varphi_k := \int_a^b g(x) b_N^k(x) dx$, $k = 1, \dots, N$.

piecewise derivatives

The detailed computations start with the evident fact that

$$|i - j| \geq 2 \quad \Rightarrow \quad \frac{b_N^j}{dx}(x) \cdot \frac{b_N^i}{dx}(x) = 0 \quad \forall x \in [a, b],$$

because there is *no overlap* of the *supports* of the two basis functions.

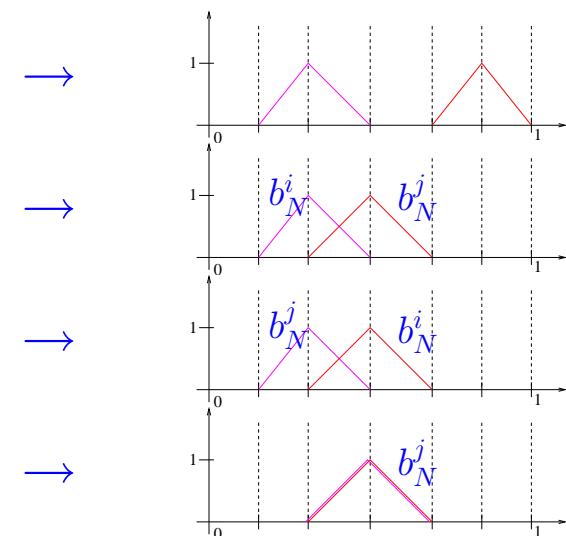
Definition 1.5.53 (Support of a function).

The *support* of a function $f : \Omega \mapsto \mathbb{R}$ is defined as

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

In addition, we use that the gradients of the tent functions are piecewise constant, see (1.5.51).

$$\int_0^1 \frac{db_N^j(x)}{dx} \frac{db_N^i(x)}{dx} dx = \begin{cases} 0 & , \text{ if } |i - j| \geq 2 \\ -\frac{1}{h_{i+1}} & , \text{ if } j = i + 1 \\ -\frac{1}{h_i} & , \text{ if } j = i - 1 \\ \frac{1}{h_i} + \frac{1}{h_{i+1}} & , \text{ if } 1 \leq i = j \leq M - 1 \end{cases}$$



→ **A** symmetric, positive definite and tridiagonal:

$$\mathbf{A} = \begin{pmatrix} \frac{1}{h_1} + \frac{1}{h_2} & -\frac{1}{h_2} & 0 & & & 0 \\ -\frac{1}{h_2} & \frac{1}{h_2} + \frac{1}{h_3} & -\frac{1}{h_3} & & & \\ 0 & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & 0 \\ 0 & & & 0 & -\frac{1}{h_{M-1}} & \frac{1}{h_{M-1}} + \frac{1}{h_M} \end{pmatrix} \quad (1.5.54)$$

☞ notation: $h_j := |x_j - x_{j-1}|$ local meshwidth, cell size

e.g, composite trapezoidal rule: $\varphi_k = \int_0^1 g(x) b_N^k(x) dx \approx \frac{1}{2}(h_k + h_{k+1})g(x_k), \quad 1 \leq k \leq N.$

$$(1.5.55)$$

For equidistant mesh with uniform cell size $h > 0$ we arrive at the linear system of equations:

$$\frac{1}{h} \begin{pmatrix} 2 & -1 & 0 & & & 0 \\ -1 & 2 & -1 & & & \\ 0 & \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & 0 & \\ & & -1 & 2 & -1 & \\ 0 & & 0 & -1 & 2 & \end{pmatrix} \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_N \end{pmatrix} = h \begin{pmatrix} g(x_1) \\ \vdots \\ g(x_N) \end{pmatrix}. \quad (1.5.56)$$

② case: linear variational problem with variable stiffness, cf. (1.4.15)

$$u \in C_{0,\text{pw}}^1([a, b]): \int_a^b \sigma(x) \frac{du}{dx}(x) \frac{dv}{dx}(x) dx = \int_a^b g(x)v(x) dx \quad \forall v \in C_{0,\text{pw}}^1([a, b]).$$

Discrete variational problem with $u_N = \mu_1 b_N^1 + \dots + \mu_N b_N^N$:

$$\int_a^b \sigma(x) \sum_{l=1}^N \mu_l \frac{db_N^l}{dx}(x) \frac{db_N^k}{dx}(x) dx = \int_a^b g(x) b_N^k(x) dx \quad k = 1, \dots, N . \quad (1.5.57)$$

Here:

numerical quadrature required for both integrals

Choice: • composite midpoint rule for left hand side integral → [14, Sect. 10.3]

$$\int_a^b f(x) dx \approx \sum_{j=1}^M h_j f(m_j) , \quad m_j := \frac{1}{2}(x_j + x_{j-1}) . \quad (1.5.58)$$

• composite trapezoidal rule [14, Eq. 10.3.2] for right hand side integral, see (1.5.55).

Assumption: $\sigma \in C_{\text{pw}}^0([a, b])$ with jumps *only* at grid nodes x_j

$$\begin{aligned}
 & \sum_{l=1}^N \underbrace{\left(\sum_{j=1}^M h_j \sigma(m_j) \frac{db_N^l}{dx}(m_j) \frac{db_N^k}{dx}(m_j) \right)}_{=(\mathbf{A})_{k,l}} \mu_l = \underbrace{\frac{1}{2}(h_{k+1} + h_k)g(x_k)}_{=: \varphi_k}, \quad k = 1, \dots, N, \\
 & \quad \Updownarrow \\
 & \quad \mathbf{A}\vec{\mu} = \vec{\varphi}.
 \end{aligned}$$

Resulting linear system of equations equidistant mesh with uniform cell size $h > 0$

$$\begin{aligned}
 & \frac{1}{h} \begin{pmatrix} \sigma_1 + \sigma_2 & -\sigma_2 & 0 \\ -\sigma_2 & \sigma_2 + \sigma_3 & -\sigma_3 \\ 0 & \ddots & \ddots & \ddots \\ & \ddots & \ddots & \ddots & 0 \\ & & -\sigma_{M-2} & \sigma_{M-2} + \sigma_{M-1} & -\sigma_{M-1} \\ & & 0 & -\sigma_{M-1} & \sigma_{M-1} + \sigma_M \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ \mu_1 \\ \vdots \\ \mu_N \end{pmatrix} \\
 & = h \begin{pmatrix} g(x_1) \\ \vdots \\ g(x_N) \end{pmatrix}, \quad (1.5.59)
 \end{aligned}$$

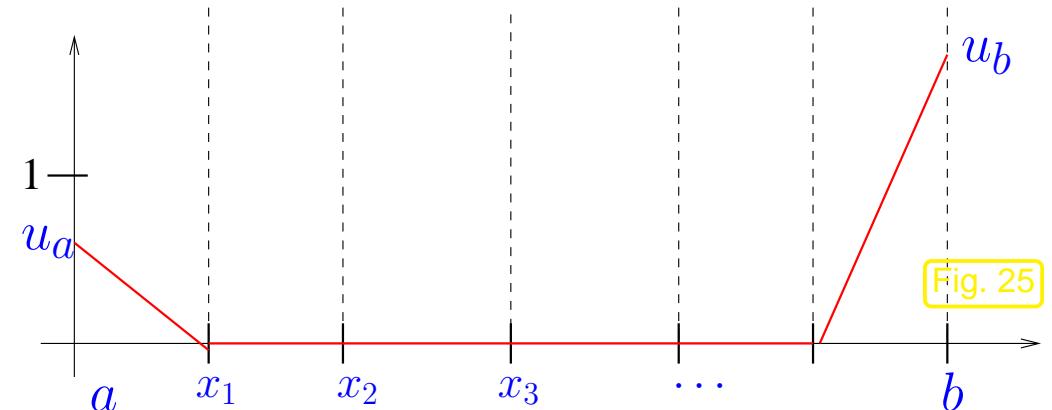
with $\sigma_j = \sigma(m_j)$, $j = 1, \dots, m$.

Remark 1.5.60 (Offset function for finite element Galerkin discretization).

In the case of general boundary conditions

$$u(a) = u_a, \quad u(b) = u_b$$

use *piecewise linear* offset function



$$u_0(x) = \begin{cases} u_a(1 - \frac{x-a}{h_1}) & , \text{ if } a \leq x \leq x_1 , \\ u_b(1 - \frac{b-x}{h_M}) & , \text{ if } x_{M-1} \leq x \leq b , \\ 0 & \text{elsewhere.} \end{cases} \quad (1.5.61)$$



Example 1.5.62 (Linear finite element Galerkin discretization for elastic string model).

Targetted: *non-linear* variational equation on domain $\Omega = [0, 1]$

$$\int_0^1 \frac{\kappa(\xi)}{L} \left(1 - \frac{L}{\|\mathbf{u}'(\xi)\|} \right) \mathbf{u}'(\xi) \cdot \mathbf{v}'(\xi) - \mathbf{f}(\xi) \cdot \mathbf{v}(\xi) \, d\xi = 0 \quad \forall \mathbf{v} \in (C_{0,\text{pw}}^1([0, 1]))^2 . \quad (1.3.7)$$

- Data κ, \mathbf{f} given in procedural form, see Rem. 1.5.3.

- trial space $V_{N,0} = (\mathcal{S}_{1,0}^0(\mathcal{M}))^2$ on equidistant mesh \mathcal{M} , meshwidth $h := \frac{1}{M}$.
- Basis: 1D tent functions, lexikographic ordering

$$\mathfrak{B} = \left\{ \begin{pmatrix} b_N^1 \\ 0 \end{pmatrix}, \begin{pmatrix} b_N^2 \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} b_N^{M-1} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ b_N^1 \end{pmatrix}, \begin{pmatrix} 0 \\ b_N^2 \end{pmatrix}, \dots, \begin{pmatrix} 0 \\ b_N^{M-1} \end{pmatrix} \right\} .$$

- Evaluation of right hand side by composite trapezoidal rule (1.5.55).
- Evaluation left hand side by composite midpoint rule (1.5.58).

Preliminary consideration: the derivative of

$$\mathbf{u}_N := \mu_1 \begin{pmatrix} b_N^1 \\ 0 \end{pmatrix} + \dots + \mu_{M-1} \begin{pmatrix} b_N^{M-1} \\ 0 \end{pmatrix} + \mu_M \begin{pmatrix} 0 \\ b_N^1 \end{pmatrix} + \dots + \mu_{2M-2} \begin{pmatrix} 0 \\ b_N^{M-1} \end{pmatrix} \quad (1.5.63)$$

is piecewise constant on \mathcal{M} :

$$\text{in }]x_{j-1}, x_j[: \quad s_j(\vec{\mu}) := \mathbf{u}'_N(\xi) = \frac{\mathbf{u}(x_j) - \mathbf{u}(x_{j-1})}{h} \quad (1.5.64)$$

$$= \frac{1}{h} \cdot \begin{cases} \begin{pmatrix} \mu_j - \mu_{j-1} \\ \mu_{j+M-1} - \mu_{j+M-2} \end{pmatrix} & , \text{ if } 2 \leq j \leq M-1 , \\ \begin{pmatrix} \mu_1 \\ \mu_M \end{pmatrix} - \mathbf{u}(0) & , \text{ if } j = 1 , \\ \mathbf{u}(1) - \begin{pmatrix} \mu_{M-1} \\ \mu_{2M-2} \end{pmatrix} & , \text{ if } j = M . \end{cases}$$

Set: $r_j = r_j(\vec{\mu}) := h \frac{\kappa(m_j)}{L} \left(1 - \frac{L}{\|s_j(\vec{\mu})\|} \right)$

Single row non-linear system of equations arising from Galerkin finite element discretization:

$$\text{row 1: } (r_1 + r_2)\mu_1 - r_2\mu_2 = hf_1(h) + r_1a , \quad (1.5.65)$$

$$\text{row } j: -r_j\mu_j + (r_j + r_{j+1})\mu_{j+1} - r_{j+1}\mu_{j+2} = f_1(jh) , \quad 2 \leq j < M - 1 , \quad (1.5.66)$$

$$\text{row } M - 1: -r_{M-1}\mu_{M-2} + (r_{M-1} + r_M)\mu_{M-1} = hf_1((M-1)h) + r_Mb , \quad (1.5.67)$$

$$\text{row } M: (r_1 + r_2(\vec{\mu}))\mu_M - r_2\mu_{M+1} = hf_2(h) + r_1u_a , \quad (1.5.68)$$

$$\text{row } j: -r_j\mu_{j+M-1} + (r_j + r_{j+1})\mu_{j+M} - r_{j+1}\mu_{j+M+1} = f_2(jh) , \quad 2 \leq j < M - 1 , \quad (1.5.69)$$

$$\text{row } M - 1: -r_{M-1}\mu_{2M-3} + (r_{M-1} + r_M)\mu_{2M-2} = hf_2((M-1)h) + r_Mu_b . \quad (1.5.70)$$

Here the dependence $r_j = r_j(\vec{\mu})$ has been suppressed to simplify the notation.

Please study the derivation of (1.5.59) in order to understand how (1.5.65)-(1.5.70) arise.

These equations can be written in a more compact form:

$$(1.5.65)-(1.5.70) \Leftrightarrow \begin{pmatrix} \mathbf{R}(\vec{\mu}) & 0 \\ 0 & \mathbf{R}(\vec{\mu}) \end{pmatrix} \vec{\mu} = \begin{pmatrix} \vec{\varphi}_1 \\ \vec{\varphi}_2 \end{pmatrix}. \quad (1.5.71)$$

with

$$\mathbf{R}(\vec{\mu}) := \begin{pmatrix} r_1 + r_2 & -r_2 & 0 & & & 0 \\ -r_2 & r_2 + r_3 & -r_3 & & & \\ 0 & \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \ddots & 0 \\ & & -r_{M-2} & r_{M-2} + r_{M-1} & -r_{M-1} & \\ 0 & & 0 & -r_{M-1} & r_{M-1} + r_M & \end{pmatrix} \in \mathbb{R}^{M-1, M-1},$$

$$(\vec{\varphi}_1)_j := h f_1(hj), \quad (\vec{\varphi}_2)_j := h f_2(hj), \quad j = 1, \dots, M-1.$$

Iterative solution of (1.5.71) by **fixed point iteration**, see Ex. 1.5.43

Initial guess $\vec{\mu}^{(0)} \in \mathbb{R}^N$; $k = 0$;

repeat

$k \leftarrow k + 1$;

Solve the *linear* system of equations

$$\begin{pmatrix} \mathbf{R}(\vec{\mu}^{(k-1)}) & 0 \\ 0 & \mathbf{R}(\vec{\mu}^{(k-1)}) \end{pmatrix} \vec{\mu}^{(k)} = \begin{pmatrix} \vec{\varphi}_1 \\ \vec{\varphi}_2 \end{pmatrix};$$

until $\left\| \vec{\mu}^{(k)} - \vec{\mu}^{(k-1)} \right\| \leq \text{tol} \cdot \left\| \vec{\mu}^{(k)} \right\|$

Code 1.5.72: Linear finite element discretization of elastic string variational problem

```
1 function [vu,Jrec,figsol,figerg] =
2     stringlinfem(kappa,f,L,u0,u1,M,tol)
3     % Solving the non-linear variational problem (1.3.7) for the elastic string by
4     means of piecewise
5     % linear finite elements on an equidistant mesh with  $M - 1$  interior nodes.
6     % kappa, f are handles of type @(xi) providing the coefficient function
7     %  $\kappa$  and the force field f. u0 and u1 pass the pinning points.
8     % M is the number of mesh cells, tol specifies the tolerance for the fixed
9     point
10    % iteration. return value:  $2 \times (M + 1)$ -matrix of node positions
11
12    if (nargin < 7), tol = 1E-2; end
13
14    h = 1/M;                                % meshwidth
15    phi = h*f(h*(1:M-1));                  % Right hand side vector
16
17
18    % Initial guess: straight string, condition  $L > \|u(0) - u(1)\|$ .
19    if (L >= norm(u1-u0)), error ('String must be tense'); end
20
21    vu_new = u0*(1-(0:1/M:1))+u1*(0:1/M:1);
22    % Meaning of components of vu: vu(1,2:M)  $\leftrightarrow \mu_1, \dots, \mu_{M-1}$ , vu(2,2:M)  $\leftrightarrow$ 
23     $\mu_M, \dots, \mu_{2M-2}$ .
24    figsol = figure; Jrec = []; hold on;
25
26    for k=1:100      % loop for fixed point iteration, maximum 100 iterations
27        vu = vu_new;
28        % Plot shape of string
```

```

20 plot(vu(1,:),vu(2,:),'--g'); drawnow;
21 title(sprintf('M = %d, iteration #%d',M,k));
22 xlabel('{\bf x_1}'); ylabel('{\bf x_2}');
23 %Compute the cell values  $s_j$ ,  $r_j$ ,  $j = 1, \dots, M$ , see (1.5.64).
24 d = (vu(:,2:end) - vu(:,1:end-1))/h;
25 s = sqrt(d(1,:).^2 + d(2,:).^2);
26 r = kappa(h*((1:M)-0.5)).*(1/L - 1./s)/h;
27 % Compute total potential energy
28 Jel = h/(2*L)*kappa(h*((1:M)-0.5))*((s-L).^2)';
29 Jf = - (phi(1,:)*vu(1,2:M)' + phi(2,:)*vu(2,2:M)')';
30 Jrec = [Jrec; k, Jel, Jf, Jel+Jf];
31 % Assemble triadiagonal matrix  $R = R(\vec{\mu})$ 
32 R = gallery('tridiag',-r(2:M-1),r(1:M-1)+r(2:M),-r(2:M-1));
33 % modify right hand side in order to take into account pinning conditions
34 phil = phi(1,:); phil(1) = phil(1) + r(1)*u0(1); phil(M-1) =
    phil(M-1) + r(M)*u1(1);
35 phi2 = phi(2,:); phi2(1) = phi2(1) + r(1)*u0(2); phi2(M-1) =
    phi2(M-1) + r(M)*u1(2);
36 % Solve linear system and compute new iterate
37 vu_new = [u0,[(R\phil)';(R\phi2)'],u1];
38 % Check simple termination criterion for fixed point iteration.
39 if (norm(vu_new - vu,'fro') < tol*norm(vu_new,'fro')/M)
    plot(vu(1,:),vu(2,:),'r-*'); break; end
40
41 end

```

```

42 % Plot of total potential energy in the course of the iteration
43 figerg = figure('name','total potential energy');
44 title(sprintf('elastic string, M = %d',M));
45 plot(Jrec(:,1),Jrec(:,4),'m-*',Jrec(:,1),Jrec(:,2),'b-+',Jrec(:,1),Jrec(:,3),'r-.');
46 xlabel('{\bf no. of iteration step}'); ylabel('{\bf energy}');
47 legend('total potential energy','elastic energy','energy in force field','location','east');

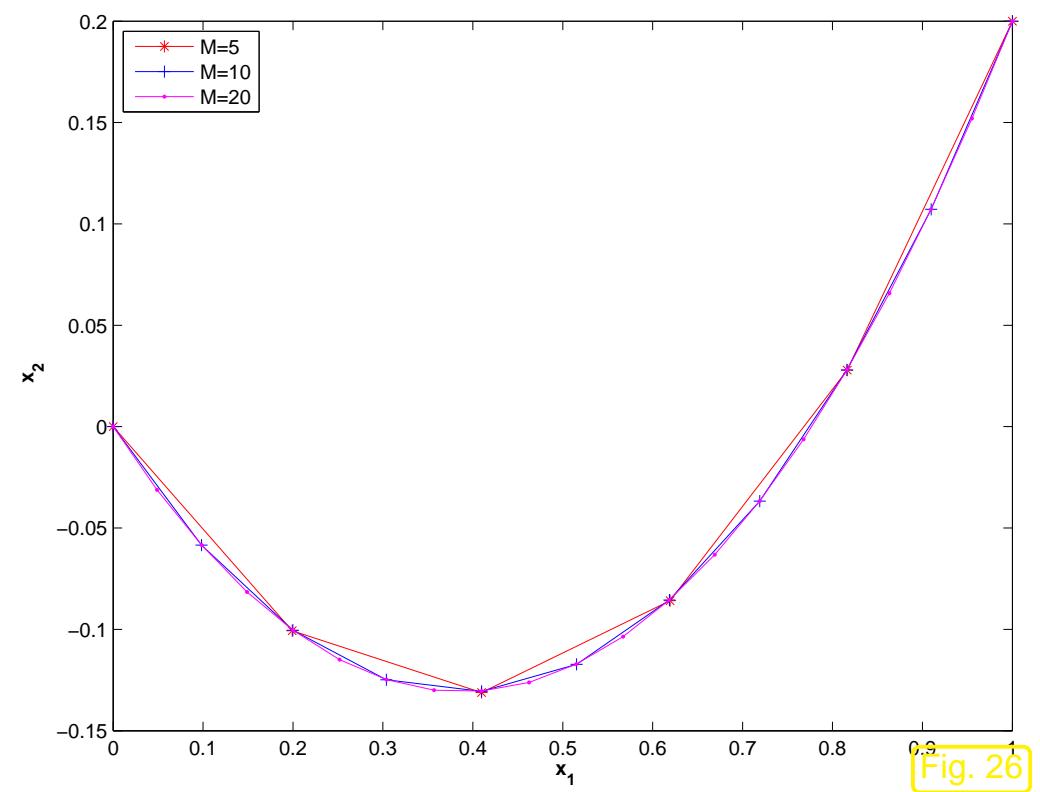
```



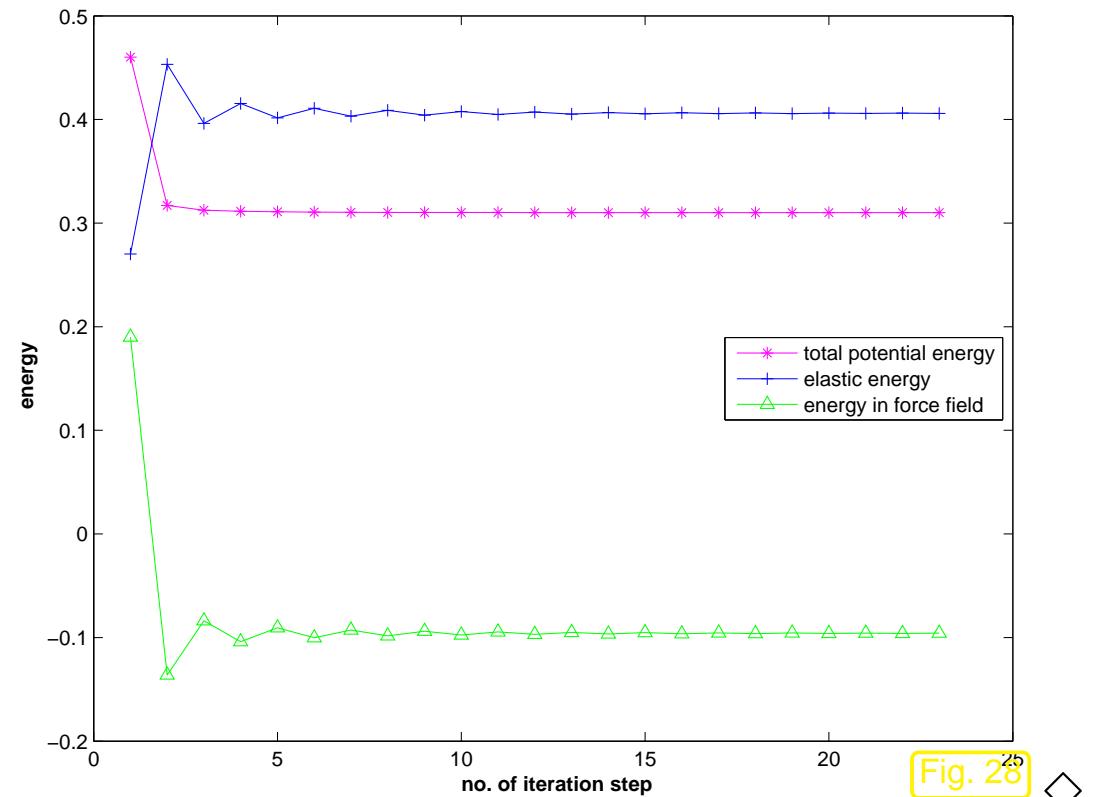
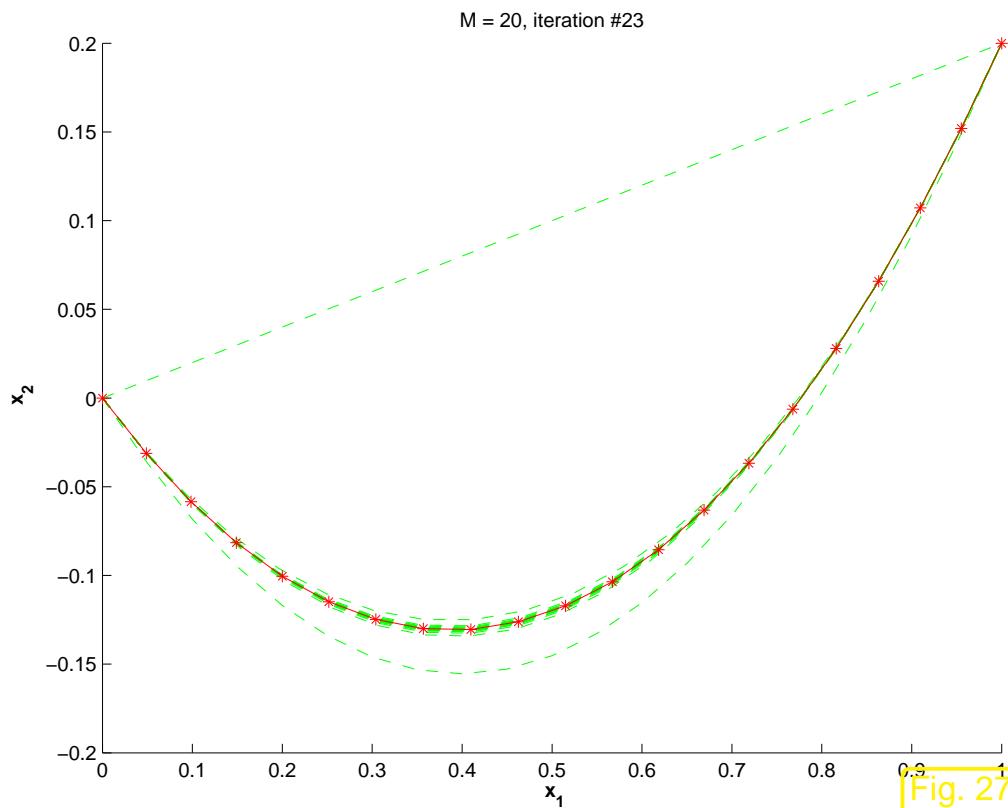
Example 1.5.73 (Elastic string shape by finite element discretization).

- Linear finite element discretization of (1.3.7), see Ex. 1.5.62, Code 1.5.71.
- $\kappa \equiv 1$, $L = 0.5$, $\mathbf{u}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\mathbf{u}(1) = \begin{pmatrix} 1 \\ 0.2 \end{pmatrix}$
- gravitational force field $\mathbf{f}(\xi) = -\begin{pmatrix} 0 \\ 2 \end{pmatrix}$.

Piecewise linear finite element solution of (1.3.7),
equidistant meshes with M cells, $M = 5, 10, 20 \triangleright$



Convergence of fixed point iteration ($M = 20$):



1.5.2 Collocation

Targetted:

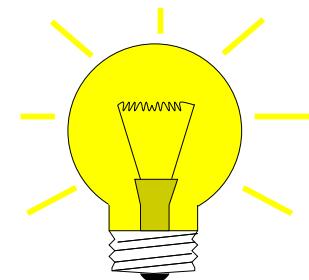
Two-point BVP = ODE $\mathcal{L}(u) = f$ + boundary conditions

Note: In contrast to the Galerkin approach, collocation techniques do not tackle the weak form of a boundary value problem, but rather the “classical”/strong form.

- Idea:
- ① seek solution in **finite-dimensional** trial space $V_{N,0}$, $N := \dim V_{N,0} < \infty$
 - ② pick **collocation nodes** $\mathcal{N} := \{x_1, \dots, x_N\} \subset \Omega$ such that \mathbf{x}

“point evaluation”
$$\begin{cases} V_{N,0} \mapsto \mathbb{R}^N \\ v \mapsto (v(x_j))_{j=1}^N \end{cases} \quad (1.5.74)$$

is a *bijection* linear mapping.



Collocation conditions: $u_N \in V_N: \quad \mathcal{L}(u_0 + u_N)(x_j) = f(x_j), \quad j = 1, \dots, N.$

↑
offset function, cf. Rem. 1.5.10

- ③ choose ordered **basis** $\mathfrak{B} = \{b_N^1, \dots, b_N^N\}$ of $V_{N,0}$ & plug basis representation

$$u_N = u_0 + \mu_1 b_N^1 + \dots + \mu_N b_N^N \quad (u_0 \hat{=} \text{offset function})$$

into collocation conditions (1.5.75)

► $\vec{\mu} = (\mu_l)_{l=1}^N: \quad \mathcal{L}(u_0 + \mu_1 b_N^1 + \dots + \mu_N b_N^N)(x_j) = f(x_j), \quad j = 1, \dots, N. \quad (1.5.76)$

In general: (1.5.76) is a non-linear system of equation (N equations for N unknowns μ_1, \dots, μ_N).

Note: bijectivity of point evaluation (1.5.74) \Rightarrow

$$\#\{\text{points}\} = \dim V_{N,0}$$

Below: detailed discussion for *linear* two point boundary value problem

$$\mathcal{L}(u) := -\frac{d}{dx} \left(\sigma(x) \frac{du}{dx}(x) \right) = g(x) , \quad a \leq x \leq b , \quad (1.5.77)$$

$$u(a) = u_a , \quad u(b) = u_b , \quad (1.5.78)$$

on domain $\Omega = [a, b]$, related to variational problem (1.4.15).

Remark 1.5.79 (Smoothness requirements for collocation trial space).

For two-point BVP (1.5.77) consider space $V_{N,0} := \mathcal{S}_{1,0}^0(\mathcal{M})$ of \mathcal{M} -piecewise linear finite element functions. → Sect. 1.5.1.2

Note: $v_N \in \mathcal{S}_{1,0}^0(\mathcal{M})$ is *not differentiable* in nodes x_j of the mesh.

- Natural choice collocation points = nodes of the mesh is *not possible!*
(because for $v_N \in \mathcal{S}_{1,0}^0(\mathcal{M})$ the function $\mathcal{L}(v_N)$ is discontinuous in the nodes of the mesh)
- Assuming $\sigma \in C^1([a, b])$ global continuity of $\mathcal{L}(v_N)$ entails $V_{N,0} \subset C^2([a, b])$, cf.
Sect. 1.5.2.2. △

1.5.2.1 Spectral collocation

Focus: *linear* two point boundary value problem (1.5.77)

trial space for polynomial spectral collocation:

$$V_{N,0} = \mathcal{P}_p(\mathbb{R}) \cap C_0^2([a, b]) , \quad p \geq 2 . \quad (1.5.80)$$

- = polynomials of degree $\leq p$, vanishing at endpoints of domain, $N := \dim V_{N,0} = p - 1$.
- > same trial space as for polynomial spectral Galerkin approach, see Sect. 1.5.1.1.

Discussion: polynomial spectral collocation for two-point BVP (1.5.77)

- offset function $u_0(x) := \frac{b-x}{b-a}u_a + \frac{x-a}{b-a}u_b$.
- Basis $\mathfrak{B} := \{b_N^j := M_j\}$ consisting of integrated Legendre polynomials, see (1.5.22).

\mathcal{L} from (1.5.77) is a linear differential operator!

Note:

Terminology: A differential operator is a mapping on a function space involving only values of the function argument and some of its derivatives in the same point.

A differential operator \mathcal{L} is linear, if

$$\mathcal{L}(\alpha u + \beta v) = \alpha \mathcal{L}(u) + \beta \mathcal{L}(v) \quad \forall \alpha, \beta \in \mathbb{R}, \quad \forall \text{functions } u, v \quad (1.5.81)$$

$$(1.5.76) \quad \stackrel{(1.5.81)}{\implies} \quad \sum_{l=1}^N \mathcal{L}(b_N^l)(x_k) \mu_l = f(x_k) - \mathcal{L}(u_0)(x_k), \quad k = 1, \dots, N. \quad (1.5.82)$$

$$\begin{array}{c} \uparrow \\ \mathbf{A}\vec{\mu} = \vec{\varphi}, \quad (\mathbf{A})_{k,l} := \mathcal{L}(b_N^l)(x_k), \quad k, l \in \{1, \dots, N\}, \\ \varphi_k := f(x_k) - \mathcal{L}(u_0)(x_k), \quad k \in \{1, \dots, N\}. \end{array} \quad (1.5.83)$$

An $N \times N$ linear system of equations

For BVPs featuring linear differential operators, collocation invariably leads to a linear system of equations for the unknown coefficients of the basis representation of the collocation solution.

Remark 1.5.84 (Bases for polynomial polynomial spectral collocation).

Same choices as for spectral Galerkin methods, see Rem. 1.5.20.



Remark 1.5.85 (Collocation points for polynomial spectral collocation).

Rule of thumb (without further explanation, see [12]):

choose collocation points x_j , $j = 1, \dots, N$ such that the induced Lagrangian interpolation operator (\rightarrow [14, Thm. 8.2.2]) has a small ∞ -norm, see [14, Lemma 8.2.5].

► Popular choice (due to [14, Eq. 8.5.6]): **Chebychev nodes**

$$x_k := a + \frac{1}{2}(b - a) \left(\cos\left(\frac{2k - 1}{2N} \pi\right) + 1 \right), \quad k = 1, \dots, N. \quad (1.5.86)$$



Code 1.5.87: Computation of derivatives of Legendre polynomials using (1.5.27)

```
1 function [V,M,D] = dilegpol(n,x)
2 % Computes values of the first n+1 Legendre polynomials (returned in matrix V)
```

1.5

p. 120

```

3 % the first n-1 integrated Legendre polynomials (returned in matrix M), and
4 % first n+1 first derivatives of Legendre polynomials in the points xj passed
5 % in the row vector x.
6 % Uses the recursion formulas (1.5.26) and (1.5.22)
7 V = ones(size(x)); V = [V; x];
8 % recursion (1.5.26) for Legendre polynomials
9 for j=1:n-1, V = [V; ((2*j+1)/(j+1)).*x.*V(end,:)-
10   j/(j+1)*V(end-1,:)]; end
11 % Formula (1.5.22) for integrated Legendre polynomials
12 M = diag(1./(2*(1:n-1)+1))*(V(3:n+1,:)-V(1:n-1,:));
13 % Recursion formula (1.5.27) for derivatives of Legendre polynomials
14 if (nargout < 3)
15   D = [zeros(size(x)); ones(size(x))];
16   for j=1:n-1, D = [D; (2*j+1)*V(j+1,:)+D(j,:)]; end
end

```

Code 1.5.88: Spectral collocation for linear 2nd-order two-point BVP

```

1 function u = linspeccol(g,N,x)
2 % Polynomial spectral collocation discretization of linear 2nd-order two-point
3 % BVP
4 %  $-\frac{d^2u}{dx^2} = g(x)$ ,  $u(0) = u(1) = 0$ 
5 % on  $\Omega = [0, 1]$ . Trial space of dimension N, collocation in Chebychev nodes.
6 % Values of approximate solution in points  $x_j$  are returned in the row vector u
7 cn = cos((2*(1:N)-1)*pi/(2*N)); % Chebychev nodes, see (1.5.86)

```

```

7 | [V,M,D] = dilegpol(N+1,cn); % Obtain values of (2nd
| derivatives) of  $M_m$ 
8 | mu = (-4*D(2:N+1,:))' \ (g(0.5*(cn+1))'); % Solve collocation system
9 | % Compute values of integrated Legendre polynomials at output points
10 | [V,M] = dilegpol(N+1,2*x-1); u = mu' *M;

```

Example 1.5.89 (Polynomial spectral collocation for 2-point BVP).

Setting of Ex. 1.5.18, spectral polynomial collocation, on , $N = 5, 7, 10$, basis from integrated Legendre polynomials, plot of solution u_N .

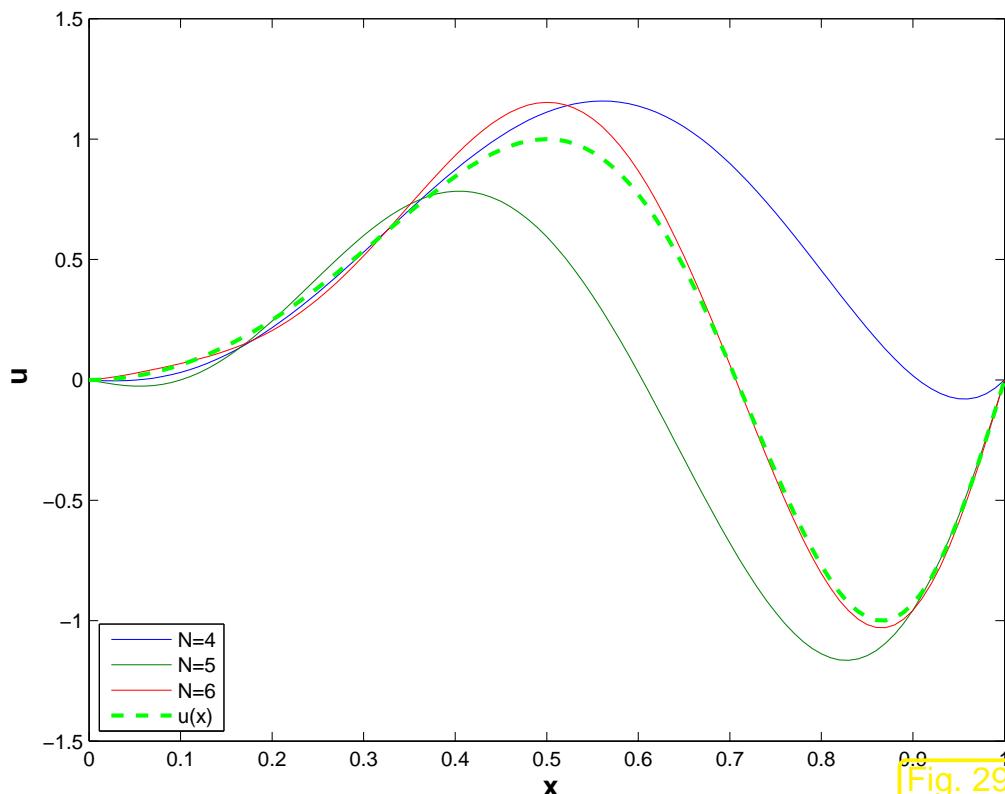


Fig. 29

Collocation in Chebychev nodes

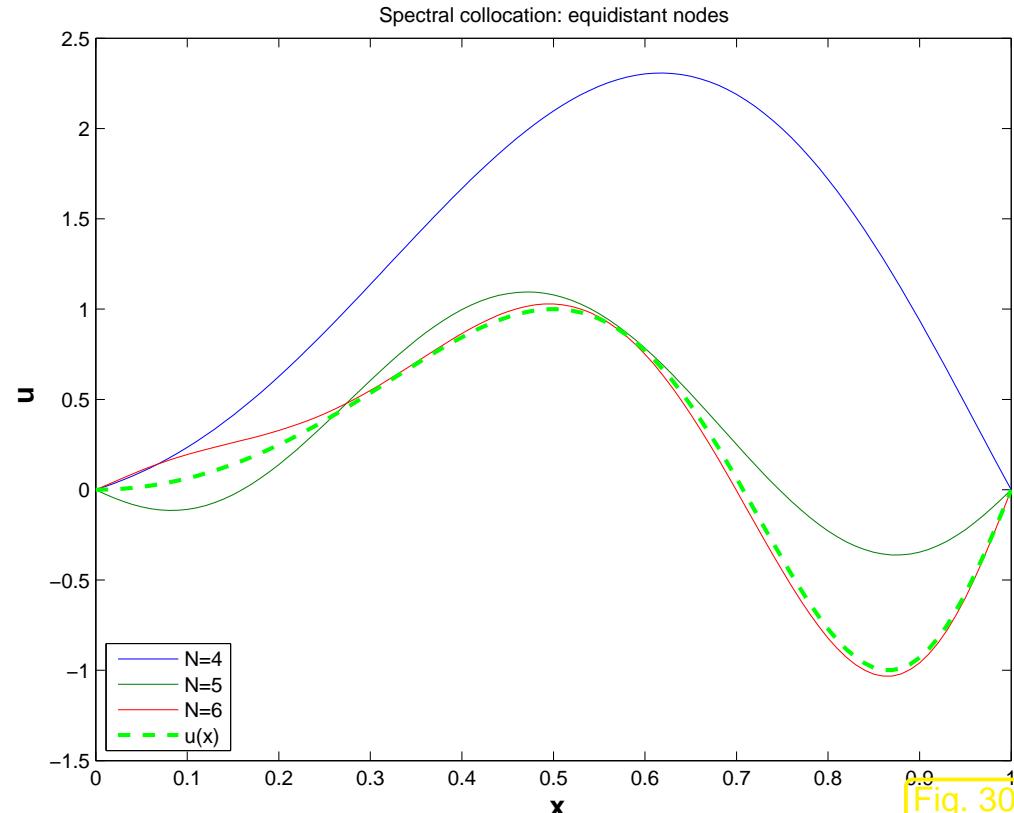


Fig. 30

Collocation in equidistant nodes



1.5.2.2 Spline collocation

Analogous to Sect. 1.5.1.2: now collocation based on *piecewise polynomials*

Rem. 1.5.79 ➤ for BVP (1.5.77) smoothness $V_{N,0} \subset C^2([a,b])$ is required.

Which piecewise polynomial spaces offer this kind of smoothness ?

Recall [14, Def. 9.4.1], cf. [14, Sect. 9.4.1]:

Definition 1.5.90 (Cubic spline).

$s :]a, b[\mapsto \mathbb{R}$ is a **cubic spline** function w.r.t. the **node set** $\mathcal{T} := \{a = x_0 < x_1 < x_2 < \dots < x_{M-1} < x_M = b\}$, if

- (i) $s \in C^2([a, b])$ (twice continuously differentiable),
- (ii) $s|_{]x_{j-1}, x_j[} \in \mathcal{P}_3(\mathbb{R})$ (**piecewise cubic polynomial**)

Known:

$$\dim \mathcal{S}_{3,\mathcal{T}} = \#\mathcal{T} + 2 = M + 3$$

► Trial space for collocation for 2-point BVP (1.5.77)

natural cubic splines: $V_{N,0} := \left\{ s \in \mathcal{S}_{3,\mathcal{T}} : \begin{array}{l} s''(a) = s''(b) = 0 \\ s(a) = s(b) = 0 \end{array} \right\} \Rightarrow \dim N := V_N = M - 1$,

Choice of collocation nodes:

collocation nodes for cubic spline collocation = spline nodes x_j : $\mathcal{N} = \mathcal{T}$

Example 1.5.91 (Cubic spline collocation discretization of 2-point BVP).

Setting of Ex. 1.5.18

Cubic spline collocation with equidistant nodes,

$$M = 5, 7, 12$$

Solution u_N ▷

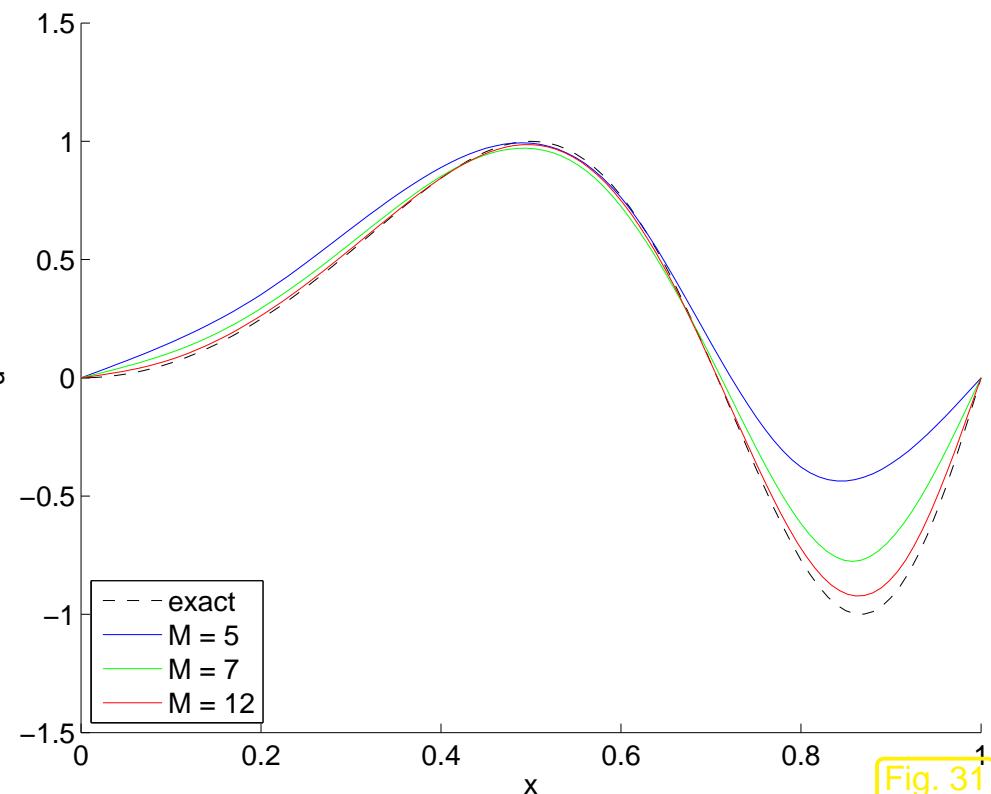


Fig. 31



1.5.3 Finite differences

Focus: 2nd-order linear two-point BVP

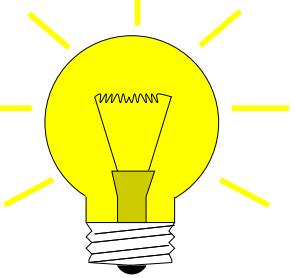
$$\mathcal{L}(u) := -\frac{d}{dx} \left(\sigma(x) \frac{du}{dx}(x) \right) = g(x) , \quad a \leq x \leq b , \quad (1.5.77)$$

$$u(a) = u_a \quad , \quad u(b) = u_b \quad ,$$

Idea:

Replace derivatives \longrightarrow difference quotients

(in finitely many special points = nodes of a mesh)

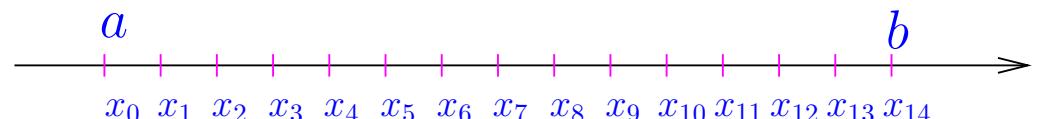


E.g.
$$\frac{d^2u}{dx^2}(x) \approx \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}, \quad h > 0 \text{ "small" .} \quad (1.5.92)$$

Setting as in Sect. 1.5.1.2:

➤ $\Omega = [a, b]$ equipped with **nodes** ($M \in \mathbb{N}$)

$$\mathcal{X} := \{a = x_0 < x_1 < \dots < x_{M-1} < x_M = b\}.$$



➤ **mesh/grid**

$$\mathcal{M} := \{]x_{j-1}, x_j[: 1 \leq j \leq M\}.$$

Special case:

equidistant mesh: $x_j := a + jh, \quad h := \frac{b-a}{M}.$

☞ $[x_{j-1}, x_j], j = 1, \dots, M,$ $\hat{=}$ **cells** of $\mathcal{M},$ **cell size** $h_j := |x_j - x_{j-1}|, j = 1, \dots, M$
meshwidth $h_{\mathcal{M}} := \max_j |x_j - x_{j-1}|$

① replacement of outer derivative ($x_{j-1/2} = \frac{1}{2}(x_j + x_{j-1})$):

$$\frac{d}{dx} \left(\sigma(x) \frac{du}{dx}(x) \right)_{|x=x_j} \approx \frac{2}{h_{j-1} + h_j} \left(\sigma(x_{j+1/2}) \frac{du}{dx}(x_{j+1/2}) - \sigma(x_{j-1/2}) \frac{du}{dx}(x_{j+1/2}) \right).$$

② replacement of inner derivative, e.g.,

$$\frac{du}{dx}(x_{j+1/2}) \approx \frac{u(x_{j+1}) - u(x_j)}{h_j}.$$



$$-\frac{d}{dx} \left(\sigma(x) \frac{du}{dx}(x) \right)_{|x=x_j} = \frac{\sigma(x_{j-1/2}) \frac{u(x_j) - u(x_{j-1})}{h_{j-1}} - \sigma(x_{j+1/2}) \frac{u(x_{j+1}) - u(x_j)}{h_j}}{\frac{1}{2}(h_{j-1} + h_j)}. \quad (1.5.93)$$

On equidistant mesh, $h_j = j$, $j = 1, \dots, M$:

$$\begin{aligned} & -\frac{d}{dx} \left(\sigma(x) \frac{du}{dx}(x) \right)_{|x=x_j} \\ &= \frac{1}{h^2} \left(-\sigma(x_{j+1/2}) u(x_{j+1}) + (\sigma(x_{j+1/2}) + \sigma(x_{j-1/2})) u(x_j) - \sigma(x_{j-1/2}) u(x_{j-1}) \right). \quad (1.5.94) \end{aligned}$$

Unknowns in finite difference method:

$$\mu_l = u(x_l), \quad l = 1, \dots, M-1$$

$$-\frac{d}{dx} \left(\sigma(x) \frac{du}{dx}(x) \right) = g(x) , a \leq x \leq b .$$

\blacktriangleleft restriction to \mathcal{X} , use (1.5.94)

$$\frac{-\sigma(x_{j+1/2})\mu_{j+1} + (\sigma(x_{j+1/2}) + \sigma(x_{j-1/2}))\mu_j - \sigma(x_{j-1/2})\mu_{j-1}}{h^2} = g(x_j) , \quad j = 1, \dots, M-1 . \quad (1.5.95)$$

\Updownarrow

$$(\mathbf{A})_{jl} = h^{-2} \cdot \begin{cases} 0 & , \text{if } |j-l| > 1 , \\ -\sigma(x_{j+1/2}) & , \text{if } j = l-1 , \\ \sigma(x_{j-1/2}) + \sigma(x_{j+1/2}) & , \text{if } j = l , \\ -\sigma(x_{l+1/2}) & , \text{if } l = j-1 . \end{cases} \quad (1.5.96)$$

$\boxed{\mathbf{A}\vec{\mu} = \vec{\varphi}}$, with

$$\varphi_j = \begin{cases} g(x_1) + \sigma(x_{1/2})u_a & , \text{if } j = 1 , \\ g(x_j) & , \text{if } 1 < j < M-1 , \\ g(x_{M-1}) + \sigma(x_{M-1/2})u_b & , \text{if } j = M-1 . \end{cases}$$

An $(M-1) \times (M-1)$ linear system of equations

(Up to scaling with h) the finite difference approach and the linear finite element Galerkin scheme (\rightarrow Sect. 1.5.1.2) yield the same system matrix for the BVP (1.5.77) and its associated variational problem (1.4.15), cf. (1.5.96) and (1.5.59).

1.6 Convergence

For elastic string model (1.2.17)/(1.3.7), taut string model in physical space (1.4.15) with exact solution $\mathbf{u} : [0, 1] \mapsto \mathbb{R}^2$ or $u : [a, b] \mapsto \mathbb{R}$, respectively:

$$\begin{array}{ccc} \text{Discretization schemes} & \longrightarrow & \text{Approximate solution} \\ (\text{Galerkin approach, Sect. 1.5.1} \\ \text{collocation methods, Sect. 1.5.2}) & & \mathbf{u}_N : [0, 1] \mapsto \mathbb{R}^2 / u_N : [a, b] \mapsto \mathbb{R} \\ & & (\text{functions } \in V_N) \end{array}$$

Desirable: approximation u_N “close to” exact solution u : *rigorous meaning ?*

↑
How to measure **discretization error** $u - u_N$?

Remark 1.6.1 (Grid functions).

Note: for finite differences (\rightarrow Sect. 1.5.3) we get no solution function, only **grid function** $\mathcal{X} \mapsto \mathbb{R}$ (“point values”)

- ☞ reconstruction of a function through **postprocessing**, e.g., linear interpolation



Remark 1.6.2.

We encountered the issues of *convergence of approximate solutions* before:

- Numerical quadrature [14, Ch. 10]: study of **asymptotic** behavior of quadrature error
- Numerical integration [14, Ch. 11]: discretization error of single step methods



1.6.1 Norms on function spaces

Tools for measuring discretization errors: **norms** on function spaces/grid function spaces

Reminder → [14, Sect. 2.5.1]

Definition 1.6.3 (Norm).

A *norm* $\|\cdot\|_V$ on an \mathbb{R} -vector space V is a mapping $\|\cdot\|_V : V \mapsto \mathbb{R}_0^+$, such that

$$(\text{definiteness}) \quad \|v\|_V = 0 \iff v = 0 \quad \forall v \in V \tag{N1}$$

$$(\text{homogeneity}) \quad \|\lambda v\|_V = |\lambda| \|v\|_V \quad \forall \lambda \in \mathbb{R}, \forall v \in V, \tag{N2}$$

$$(\text{triangle inequality}) \quad \|w + v\|_V \leq \|w\|_V + \|v\|_V \quad \forall w, v \in V. \tag{N3}$$

Next: important norms on function spaces, cf. [14, Eq. 8.2.7], [14, Eq. 8.2.8], [14, Eq. 8.2.9]:

Definition 1.6.4 (Supremum norm).

The **supremum norm** of an (essentially) bounded function $\mathbf{u} : \Omega \mapsto \mathbb{R}^n$ is defined as

$$\|\mathbf{u}\|_{\infty} (\|\mathbf{u}\|_{L^{\infty}(\Omega)}) := \sup_{x \in \Omega} \|\mathbf{u}(x)\|, \quad \mathbf{u} \in (L^{\infty}(\Omega))^n. \quad (1.6.5)$$

- $L^{\infty}(\Omega)$ denotes the vector space of essentially bounded functions. It is the instance for $p = \infty$ of an L^p -space.
- The notation $\|\cdot\|_{\infty}$ hints at the relationship between the supremum norm of functions and the maximum norm for vectors in \mathbb{R}^n .
- For $n = 1$ the Euclidean vector norm in the definition reduces to the modulus $|u(x)|$.
- The norm $\|\mathbf{u} - \mathbf{u}_N\|_{L^{\infty}(\Omega)}$ measures the maximum distance of the function values of \mathbf{u} and \mathbf{u}_N .

Definition 1.6.6 (Mean square norm/ L^2 -norm). \rightarrow Def. 2.2.5

For a function $\mathbf{u} \in (C_{\text{pw}}^0(\Omega))^n$ the **mean square norm/ L^2 -norm** is given by

$$\|\mathbf{u}\|_0 \left(\|\mathbf{u}\|_{L^2(\Omega)} \right) := \left(\int_{\Omega} \|\mathbf{u}(x)\|^2 \, dx \right)^{1/2}, \quad \mathbf{u} \in (L^2(\Omega))^n.$$

- $L^2(\Omega)$ designates the vector space of square integrable functions, another L^p -space (for $p = 2$) and a **Hilbert space**.
- The “0” in the notation $\|\cdot\|_0$ refers to the absence of derivatives in the definition of the norm.
- Obviously, the L^2 -norm is **weaker** than the supremum norm:

$$\|v\|_{L^2([a,b])} \leq \sqrt{|b-a|} \|v\|_{L^\infty([a,b])} \quad \forall v \in C_{\text{pw}}^0([a,b]).$$

In particular, the L^2 -norm of the discretization error may be small despite large deviations of \mathbf{u}_N from \mathbf{u} , provided that these deviations are very much *localized*.

We consider the model for a homogeneous taut string in physical space, see (1.4.15), with associated total potential energy functional

$$J(u) := \int_a^b \frac{1}{2} \left| \frac{du}{dx}(x) \right|^2 + \widehat{g}(x)u(x) dx , \quad u \in C_{0,\text{pw}}^1([a, b]) , \quad (1.6.8)$$

where, for the sake of simplicity, we assume $u_a = u_b = 0$.

A manifestly relevant error quantity of interest is the **deviation of energies**

$$E_J := |J(u) - J(u_N)| .$$

We adopt the concise notations introduced for abstract (linear) variational problems in Rems. 1.3.10, 1.4.4:

$$\begin{aligned} \mathbf{a}(u, v) &:= \int_a^b \frac{du}{dx}(x) \frac{dv}{dx}(x) dx , \\ J(u) &= \frac{1}{2}\mathbf{a}(u, u) - \ell(u) , \\ \ell(v) &:= - \int_a^b \widehat{g}(x)v(x) dx , \end{aligned}$$

where \mathbf{a} is a bilinear form, see Def. 1.3.11.

Assumption: $u_N \in V_{N,0} \hat{=} \text{Galerkin solution based on discrete trial space } V_{N,0} \subset V_0$.

$$\begin{aligned} & \mathbf{a}(u, v) = \ell(v) \quad \forall v \in V_0 := C_{0,\text{pw}}^1([a, b]) , \\ & \mathbf{a}(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{N,0} \subset V_0 . \end{aligned} \quad (1.6.9)$$

We can use the defining variational equations for u and u_N to express

$$J(u) - J(u_N) = -\frac{1}{2}(\mathbf{a}(u, u) - \mathbf{a}(u_N, u_N)) \stackrel{(*)}{=} -\frac{1}{2}\mathbf{a}(u + u_N, u - u_N) . \quad (1.6.10)$$

($*$): a straightforward consequence of the bilinearity of \mathbf{a} , see Def. 1.3.11, c.f. $a^2 - b^2 = (a+b)(a-b)$ for $a, b \in \mathbb{R}$.

Concretely,

$$\begin{aligned} |J(u) - J(u_N)| &= \frac{1}{2} \left| \int_a^b \frac{d}{dx}(u + u_N) \cdot \frac{d}{dx}(u - u_N) dx \right| \\ &\stackrel{(*)}{\leq} \frac{1}{2} \left(\int_a^b \left| \frac{d}{dx}(u + u_N) \right|^2 dx \right)^{1/2} \left(\int_a^b \left| \frac{d}{dx}(u - u_N) \right|^2 dx \right)^{1/2} . \end{aligned} \quad (1.6.11)$$

($*$): due to Cauchy-Schwarz inequality (2.2.15)

Definition 1.6.12 (H^1 -seminorm). → Def. 2.2.12

For a function $u \in C_{\text{pw}}^1([a, b])$ the H^1 -seminorm reads

$$|u|_{H^1([a,b])}^2 := \int_a^b |\frac{du}{dx}(x)|^2 dx . \quad (1.6.13)$$

- $|\cdot|_{H^1([a,b])}$ is merely a **semi-norm**, because it only satisfies norm axioms (N2) and (N3), but fails to be definite: $|\cdot|_{H^1([a,b])} = 0$ for constant functions.
- In the setting of the homogeneous taut string model, we have

$$|u|_{H^1(iab)}^2 = a(u, u) \quad \blacktriangleright \quad |\cdot|_{H^1([a,b])} \text{ is called the } \text{energy norm} \text{ for the model.}$$

More explanations in Sect. 2.1.3.

- On $C_{0,\text{pw}}^1([a, b])$ the semi-norm $|\cdot|_{H^1(iab)}$ is a genuine norm → Def. 1.6.3.

From (1.6.11)

$$\|u - u_N\|_{H^1(\Omega)} \leq \epsilon \quad \blacktriangleright \quad |J(u) - J(u_N)| \leq |u + u_N|_{H^1(\Omega)} \|u - u_N\|_{H^1(\Omega)} \quad (1.6.14)$$
$$\stackrel{(N3)}{\leq} (2|u|_{H^1(\Omega)} + \epsilon) \epsilon .$$

- estimate of the energy norm of the discretization error paves the way for bounding the energy deviation.



Remark 1.6.15 (Norms on grid function spaces).

To measure the discretization error for finite difference schemes (\rightarrow Sect. 1.5.3) one may resort to **mesh dependent norms**

$$(\text{discrete}) \quad l^2\text{-norm} \quad : \quad \|\vec{\mu}\|_{l^2(\mathcal{X})}^2 := \sum_{j=0}^M \frac{1}{2}(h_j + h_{j+1}) |\mu_j|^2 , \quad (1.6.16)$$

(under convention $h_0 := 0, h_{M+1} := 0$) ,

(discrete) maximum norm : $\|\vec{\mu}\|_{l^\infty(\mathcal{X})} := \max_{j=0,\dots,M} |\mu_j| . \quad (1.6.17)$

1.6.2 Algebraic and exponential convergence

Crucial: convergence is an *asymptotic notion* !

sequence of discrete models \Rightarrow sequence of approximate solutions $(u_N^{(i)})_{i \in \mathbb{N}}$
 \Rightarrow study sequence $(\|u_N^{(i)} - u\|)_{i \in \mathbb{N}}$

created by *variation of a discretization parameter*:

Discretization parameters:

- *meshwidth* $h > 0$ for finite differences (\rightarrow Sect. 1.5.3), p.w. linear finite elements (\rightarrow Sect. 1.5.1.2), spline collocation (\rightarrow Sect. 1.5.2.2)
- *polynomial degree* for spectral collocation (\rightarrow Sect. 1.5.2.1),
spectral Galerkin discretization (\rightarrow Sect. 1.5.1.1)

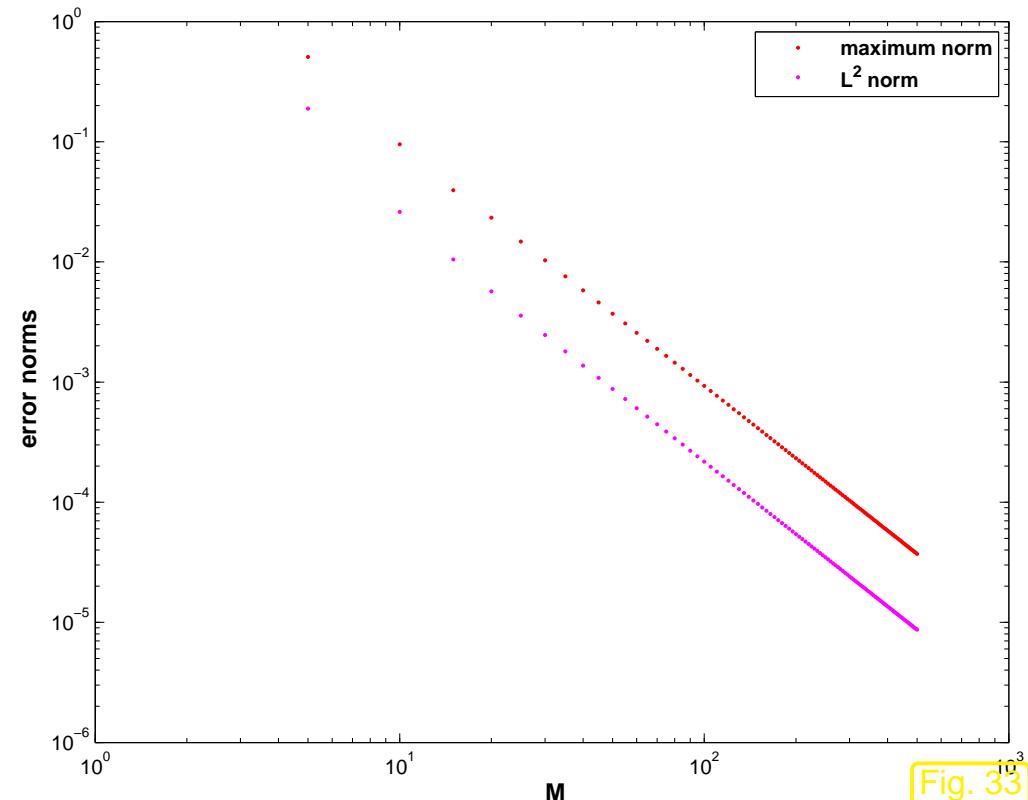
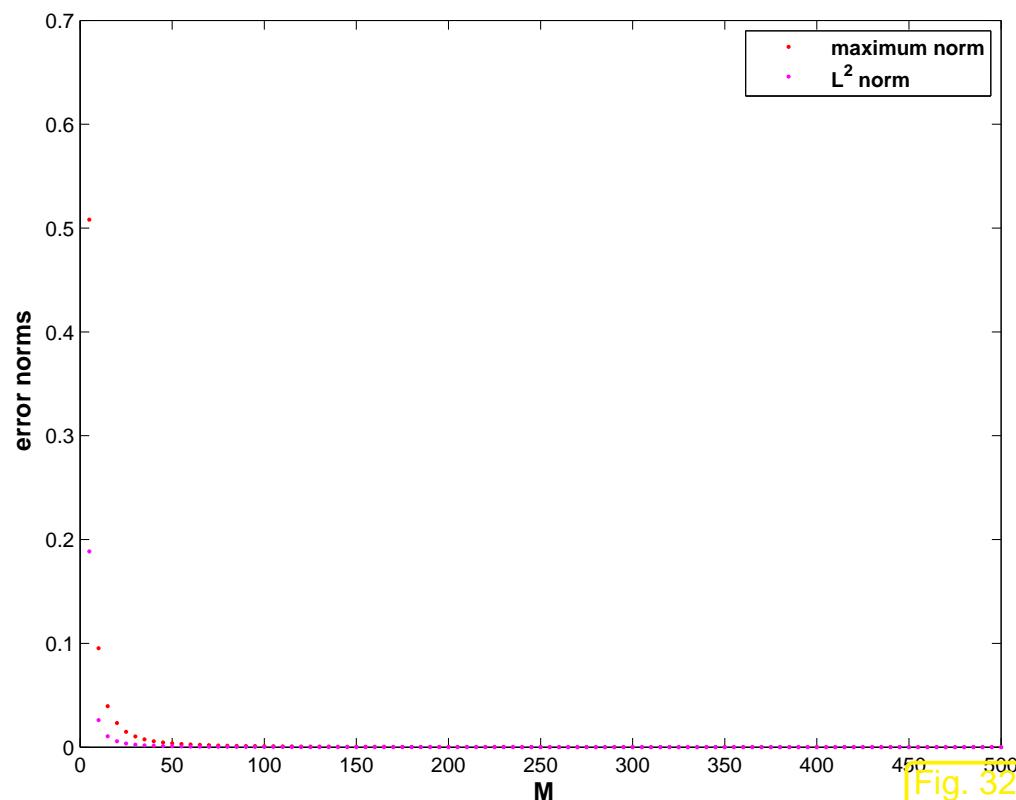
Example 1.6.18 (Numerical studies of convergence).

Focus: Linear 2-point boundary value problem $-\frac{d^2u}{dx^2} = g(x)$, $u(0) = u(1) = 0$ on $\Omega =]0, 1[$,
variational form (1.5.19),

exact solution $u(x) = \sin(2\pi x^2)$ (\rightarrow setting of Ex. 1.5.18)

① finite difference discretization on equidistant mesh, meshwidth $h > 0$ (\rightarrow Sect. 1.5.3)

Monitored: maximum norm (1.6.17), l^2 -norm (1.6.16) of pointwise discretization error



② Spectral collocation, polynomial degree $p \in \mathbb{N}$ → Sect. 1.5.2.1

Monitored: supremum norm (1.6.5), L^2 -norm (1.6.6) of discretization error $u - u_N$ (*approximated* by trapezoidal rule on fine grid with 10^4 points)

Spectral collocation, Chebychev points

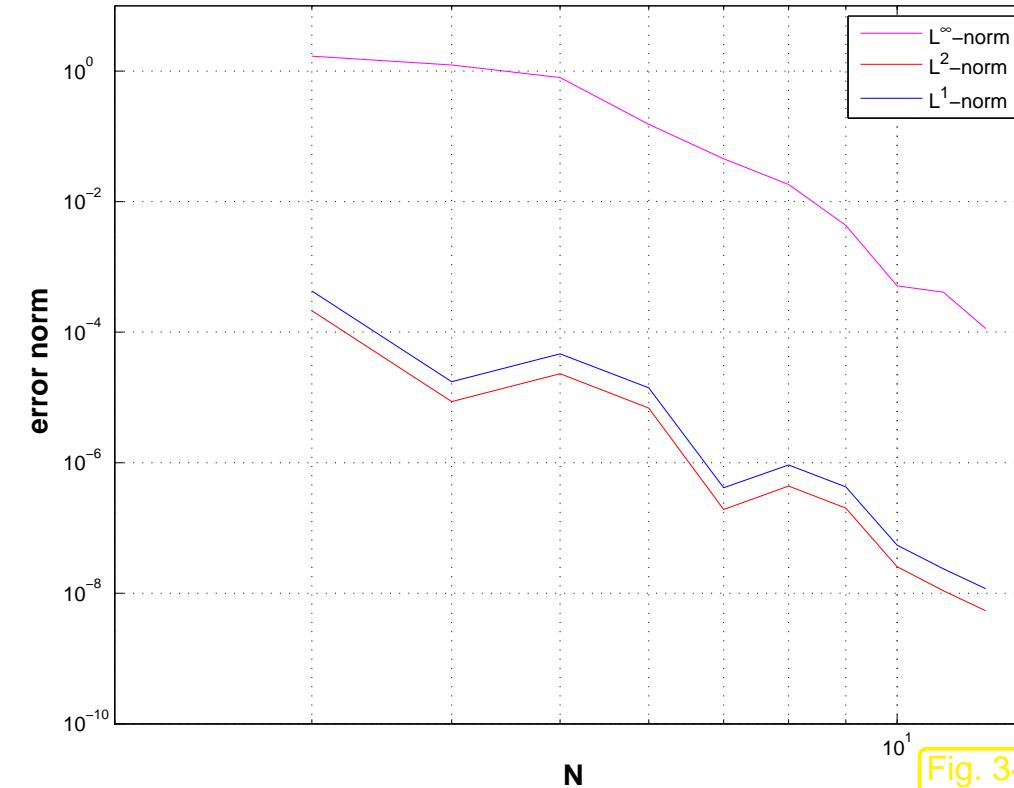


Fig. 34

Spectral collocation, Chebychev points

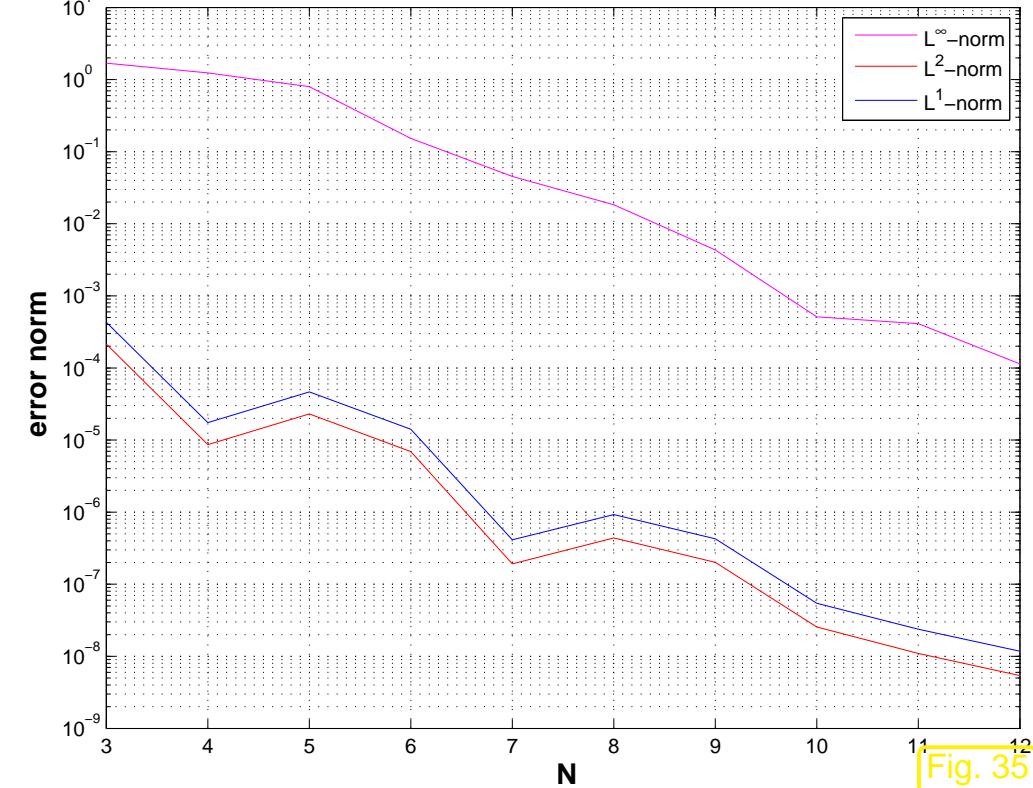
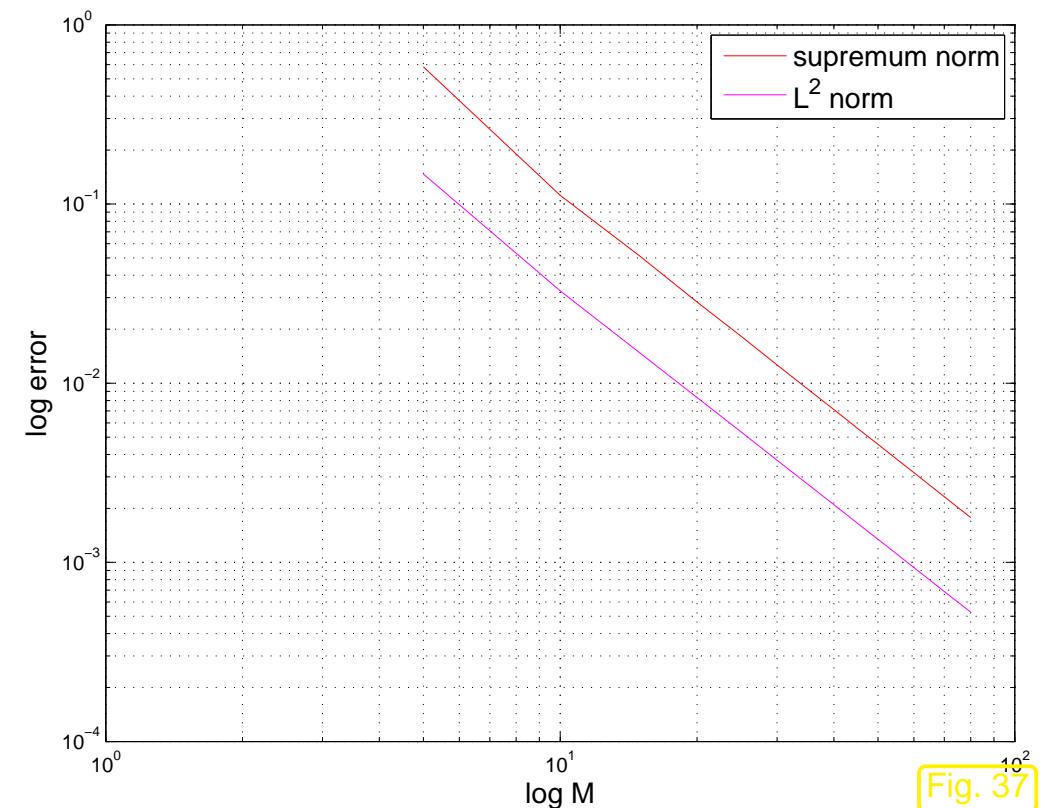
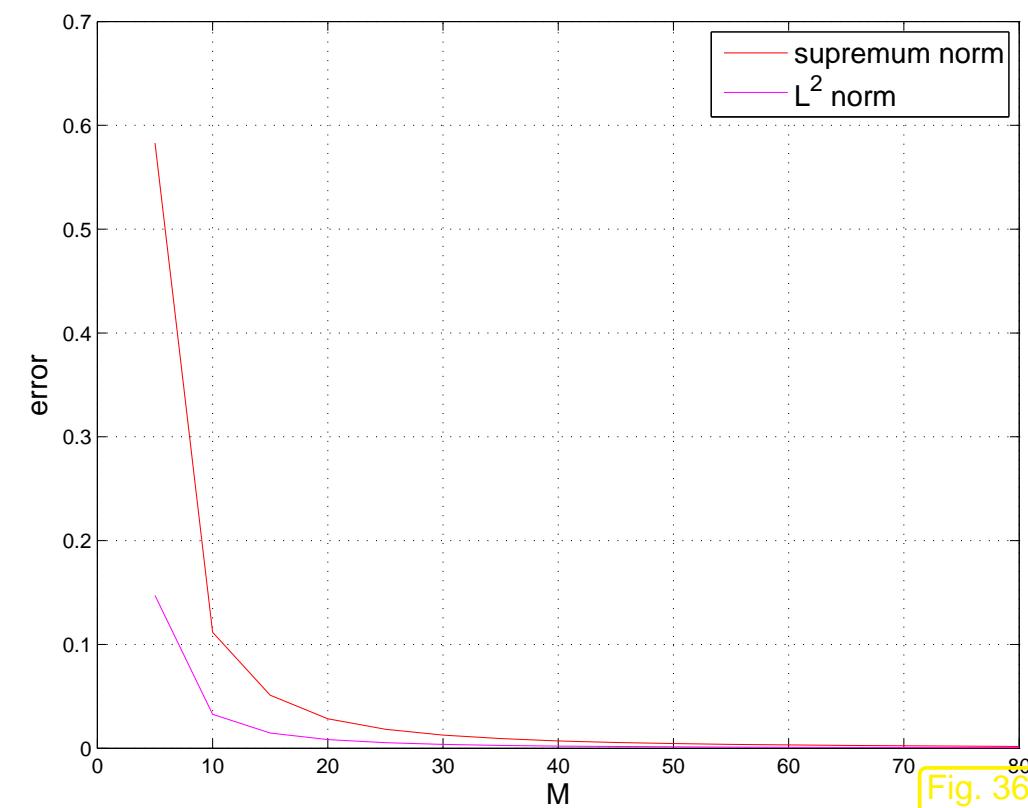


Fig. 35

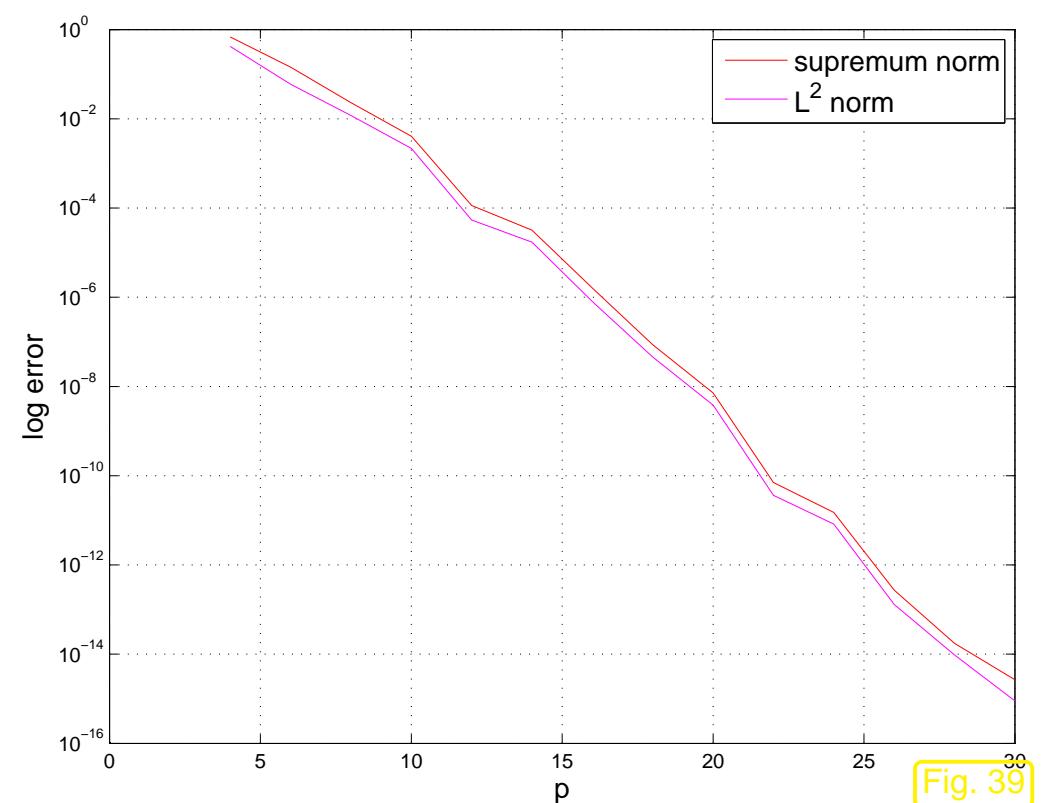
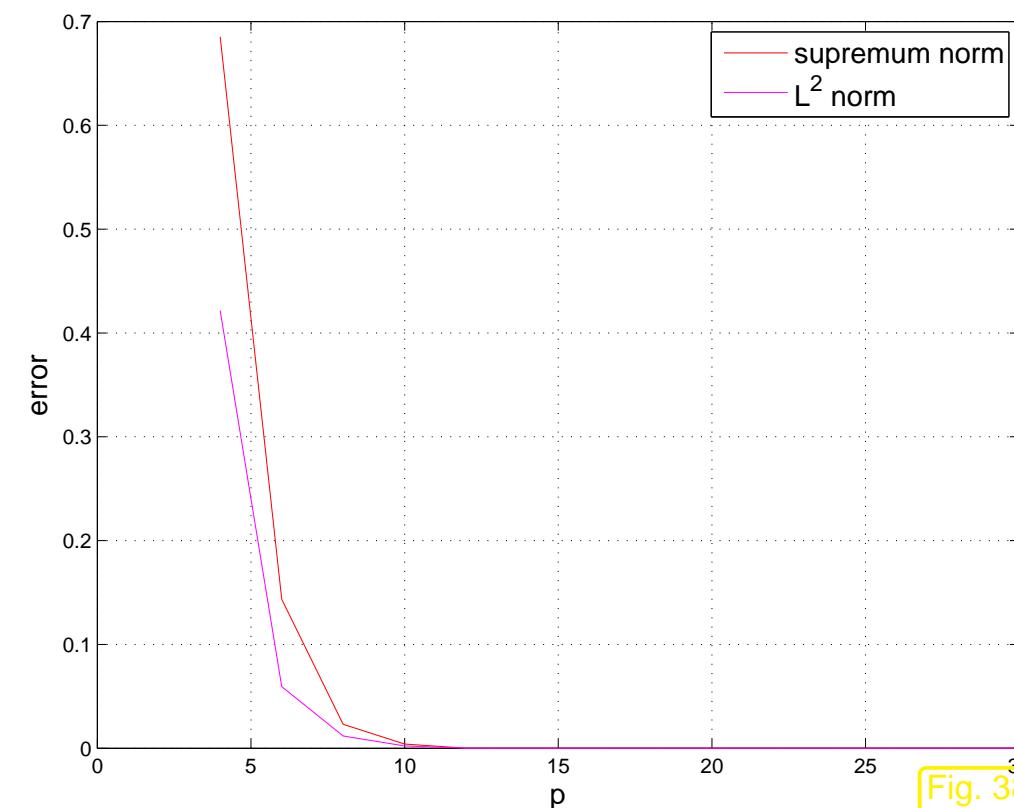
③ Spline collocation on equidistant mesh, meshwidth $h > 0$ (\rightarrow Sect. 1.5.2.2)

Monitored: supremum norm (1.6.5), L^2 -norm (1.6.6) of $u - u_N$ (approximated by sampling on fine grid with 10^4 points)



④ Spectral Galerkin based on degree $p \in \mathbb{N}$ polynomials → Sect. 1.5.1.1

Monitored: supremum norm (1.6.5), L^2 -norm (1.6.6) of discretization error $\|u - u_N\|$ (*approximated* by trapezoidal rule on fine grid with 10^4 points)



Observation:

- ▷ ‘Empiric convergence’ in all cases
- ▷ different qualitative behavior (of norm of discretization error)



Unified view:

Study $\|u - u_N\|$ as function of number N of unknowns (degrees of freedom)

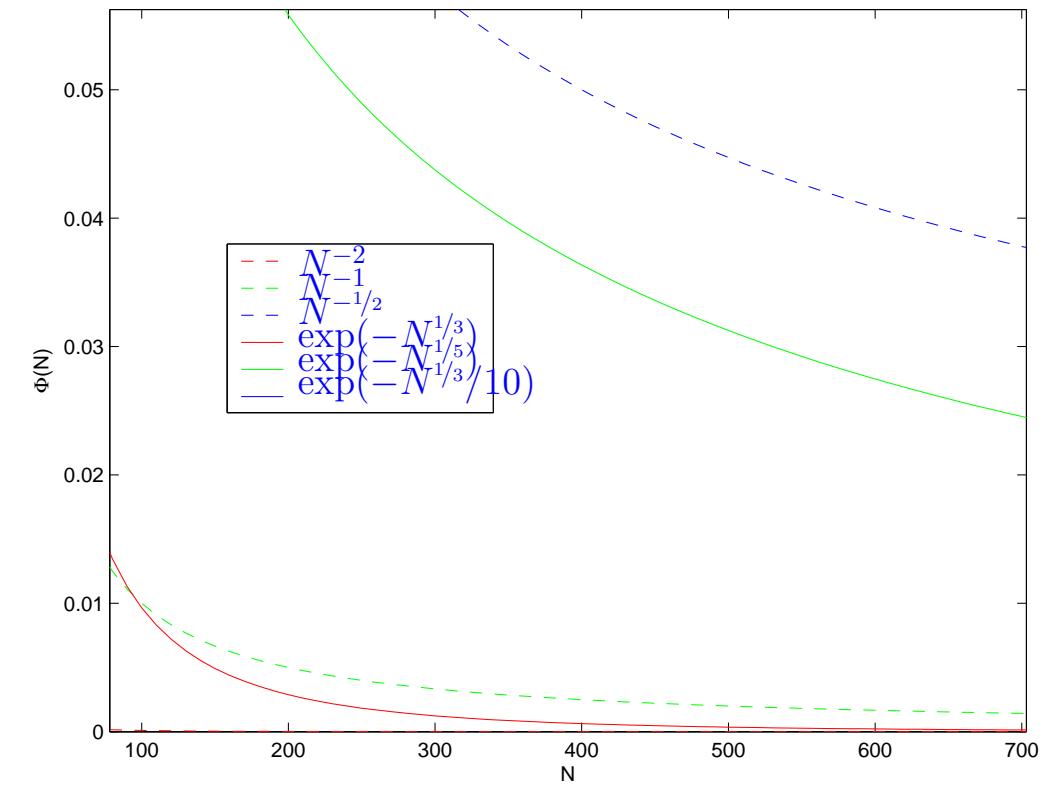
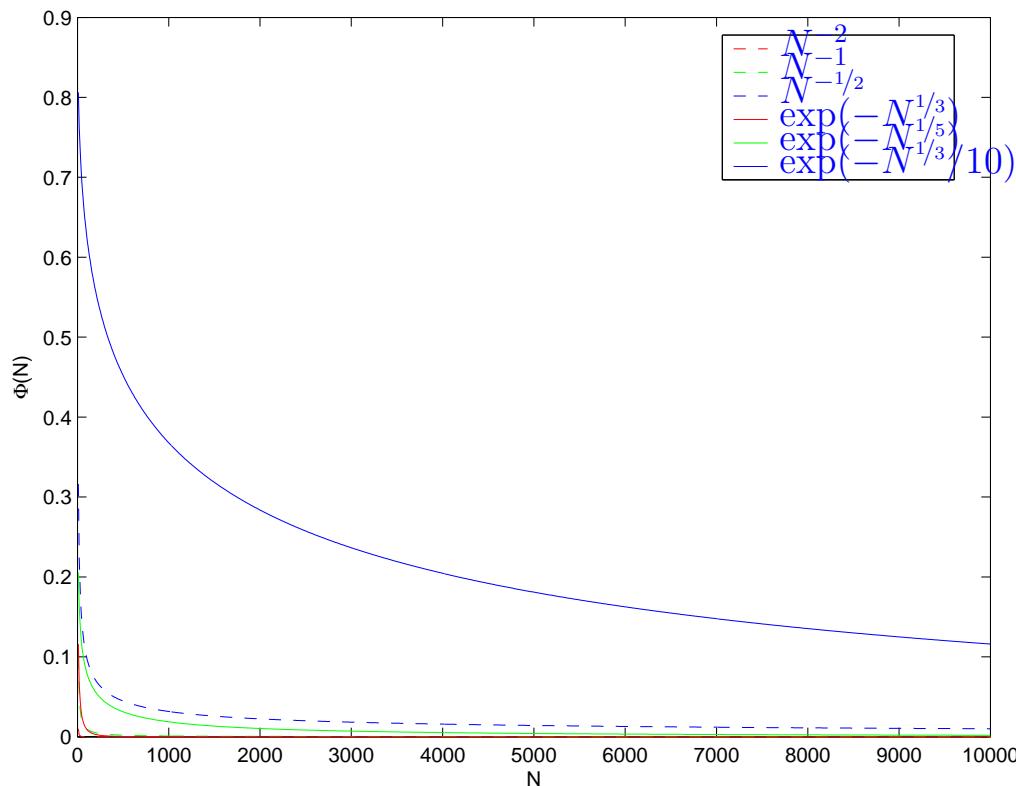
measure for costs incurred by method

Definition 1.6.19 (Convergence rate). → [14, Sect. 8.4], [14, Eq. 8.4.1]

$$\begin{aligned} \|u - u_N\| = O(N^{-\alpha}), \alpha > 0 &\iff \text{algebraic convergence with rate } \alpha \\ \|u - u_N\| = O(\exp(-\gamma N^\delta)), \text{ with } \gamma, \delta > 0 &\iff \text{exponential convergence} \end{aligned}$$

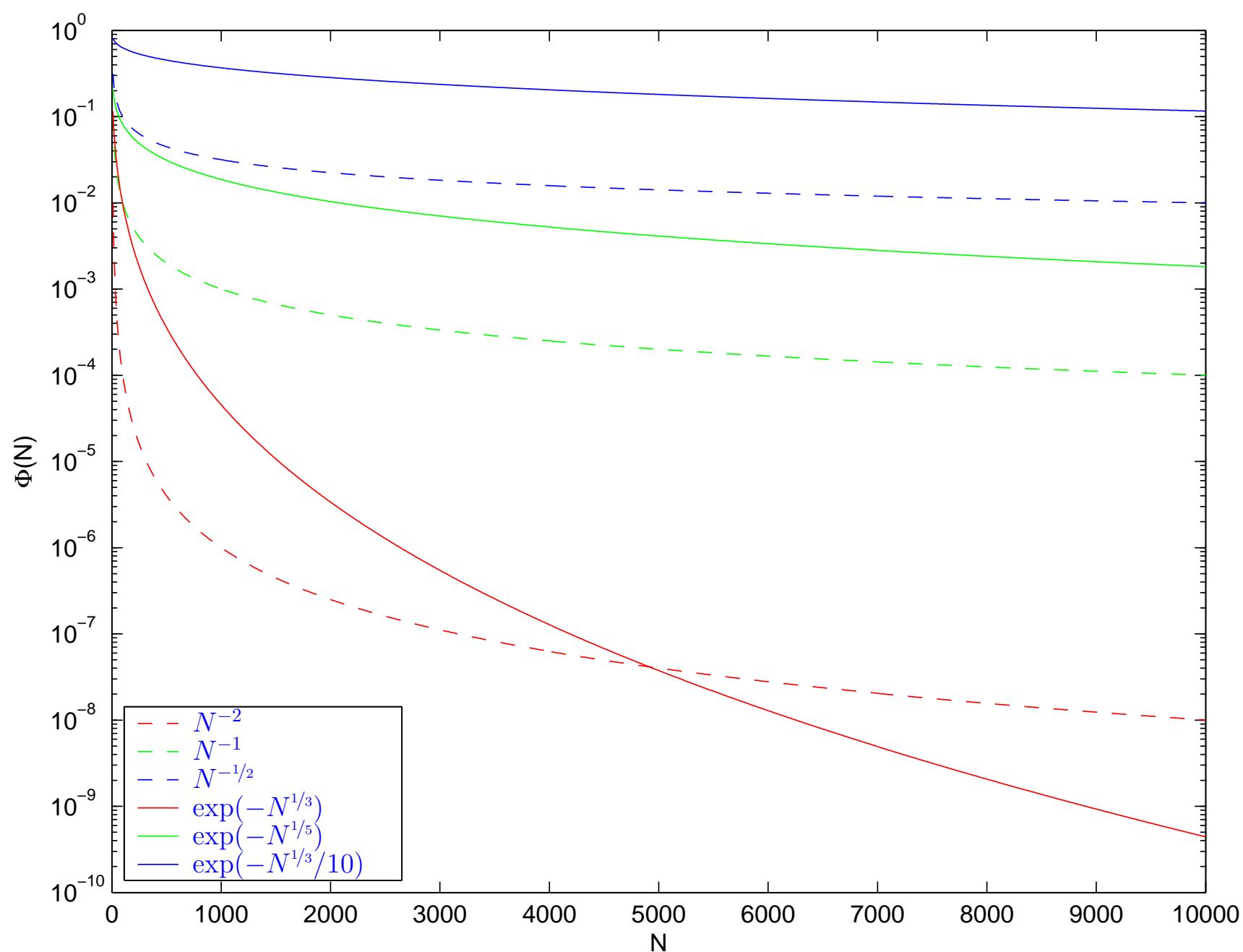
☞ recall notation (Landau- O):

$$f(N) = O(g(N)) \iff \begin{array}{l} \exists N_0 > 0, \exists C > 0 \text{ independent of } N \\ \text{such that } |f(N)| \leq Cg(N) \text{ for } N > N_0. \end{array} \quad (1.6.20)$$

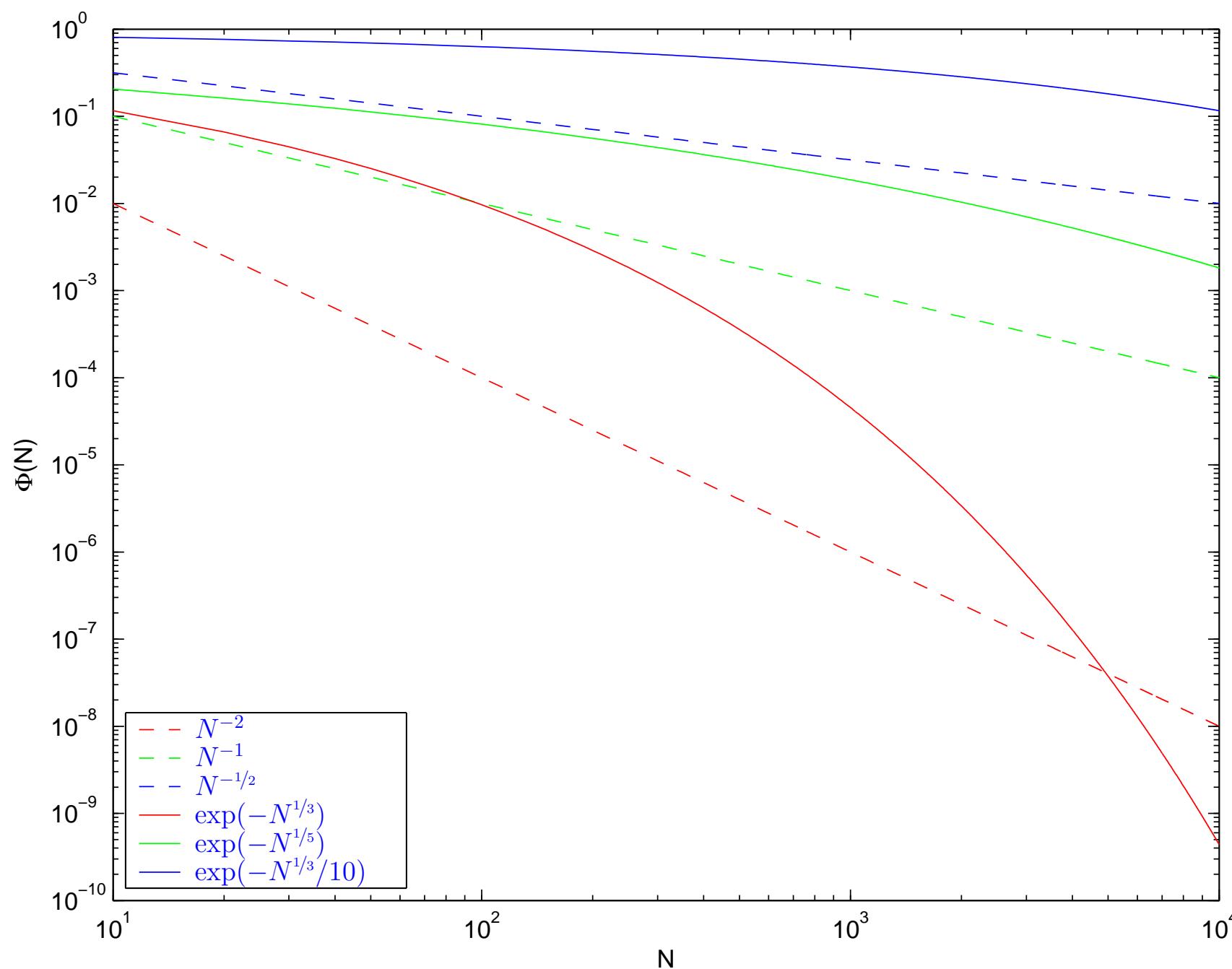


Linear plot of qualitative convergence behavior: algebraic/exponential convergence rates

Exponential convergence will always win (asymptotically)



Log-linear plot of decrease of discretization error for algebraic/exponential convergence rates



Log-log plot of decrease of discretization error for algebraic/exponential convergence rates

Remark 1.6.21 (Exploring convergence experimentally). → [14, Rem. 8.4.4]

How to determine qualitative asymptotic convergence from raw norms of discretization error?

Given: data tuples (N_i, ϵ_i) , $i = 1, 2, 3, \dots$, $N_i \hat{=} \text{problem sizes}$, $\epsilon_i \hat{=} \text{error norms}$

1. Conjecture: algebraic convergence: $\epsilon_i \approx C N_i^{-\alpha}$

$$\log(\epsilon_i) \approx \log(C) - \alpha \log N_i \quad (\text{affine linear in log-log scale}).$$

► linear regression on data $(\log N_i, \log \epsilon_i)$, $i = 1, 2, 3, \dots$ to determine rate α .

2. Conjecture: exponential convergence: $\epsilon_i \approx C \exp(-\gamma N_i^\delta)$

$$\log \epsilon_i \approx \log(C) - \gamma N_i^\delta.$$

► non-linear least squares fit (→ [14, Sect. ??]) to determine δ :

$$(c, \gamma, \delta) = \operatorname{argmin} \left\{ \sum_i |\log \epsilon_i - c + \gamma N_i^\delta|^2 \right\},$$

residual ↔ validity of conjecture. This can be done by a short MATLAB code (→ exercise)



Example 1.6.22 (Asymptotic nature of convergence).

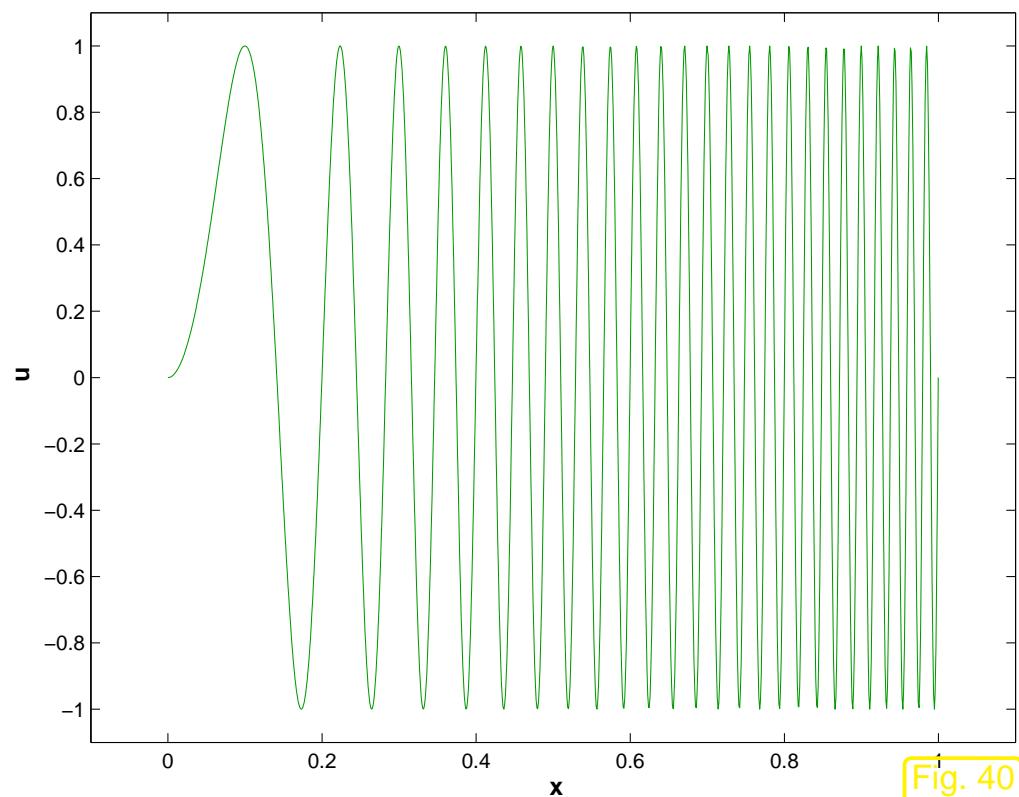


Fig. 40

- 2-point BVP $-\frac{d^2u}{dx^2} = g(x)$, $u(0) = u(1) = 0$,
 $\Omega =]0, 1[$,
 $\triangleleft u(x) = \sin(50\pi x^2)$
 - ① finite difference discretization on equidistant mesh, meshwidth $h > 0$ (\rightarrow Sect. 1.5.3)
 - ② Spectral Galerkin based on degree $p \in \mathbb{N}$ polynomials \rightarrow Sect. 1.5.1.1
- Evaluations as in Ex. 1.6.18

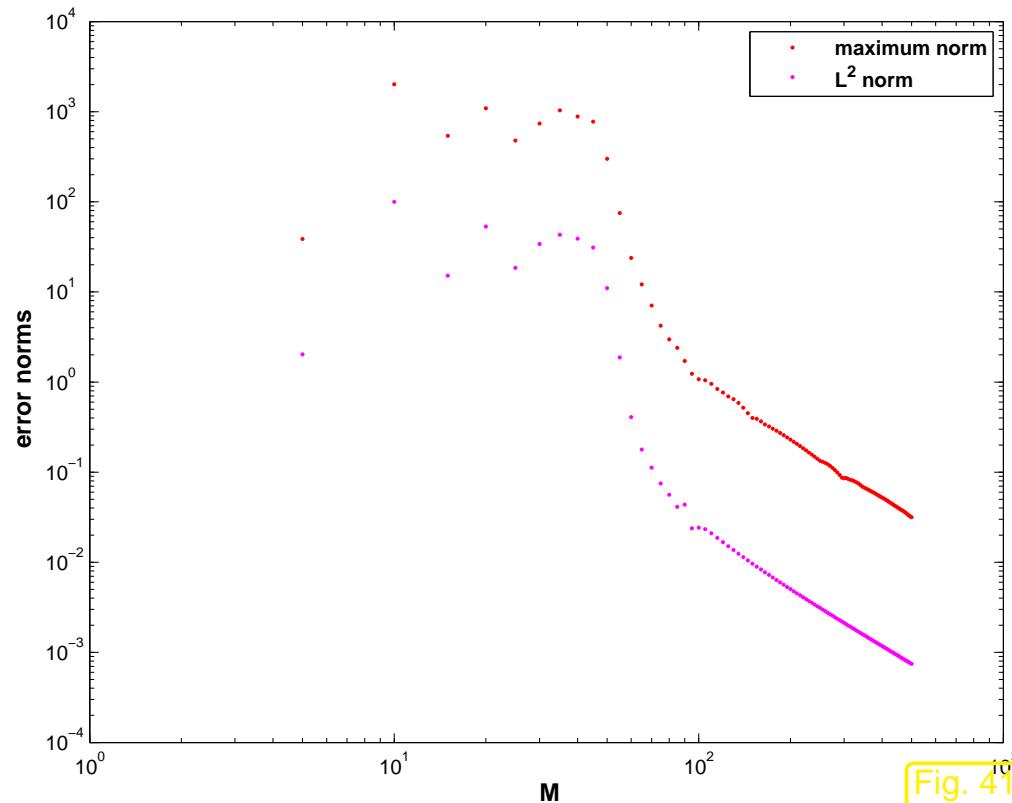


Fig. 41

① Finite Difference Method

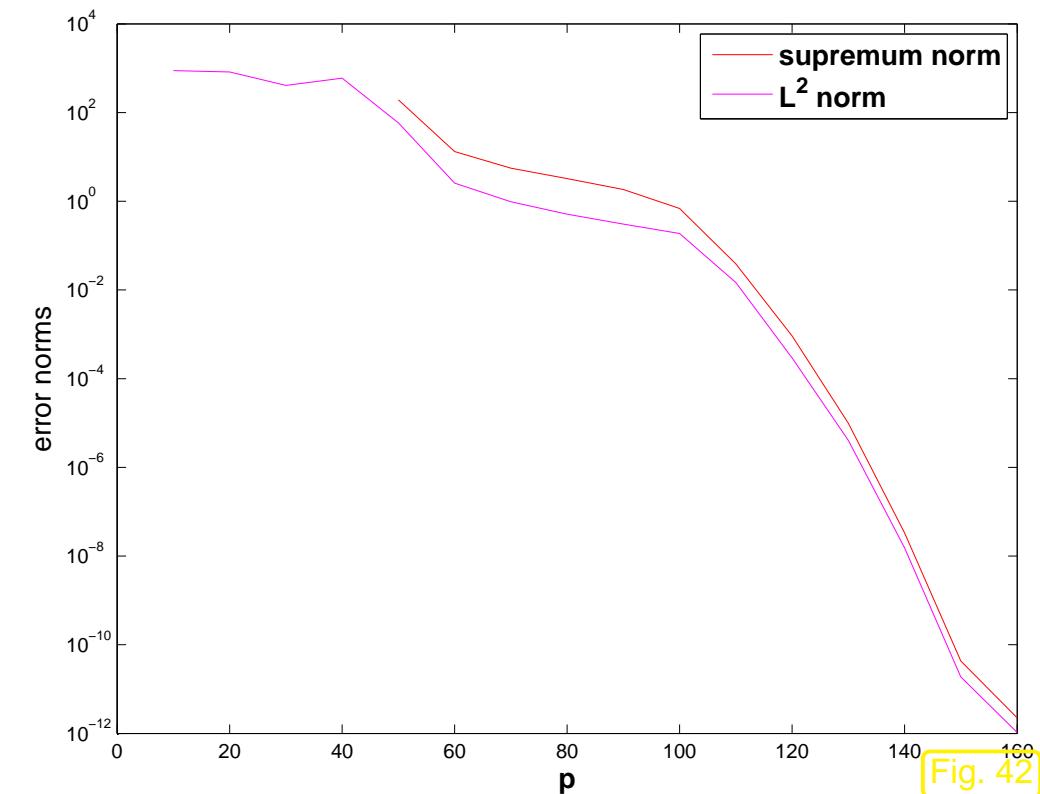


Fig. 42

② Spectral Galerkin Method

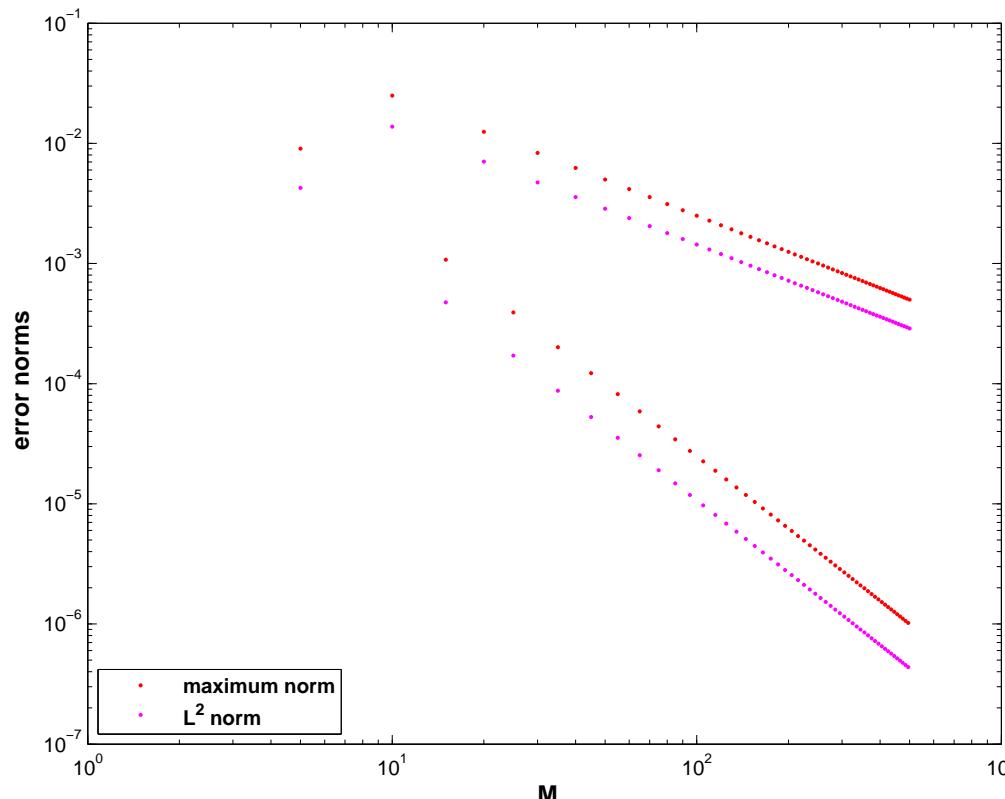


Qualitative asymptotic convergence also depends on data !

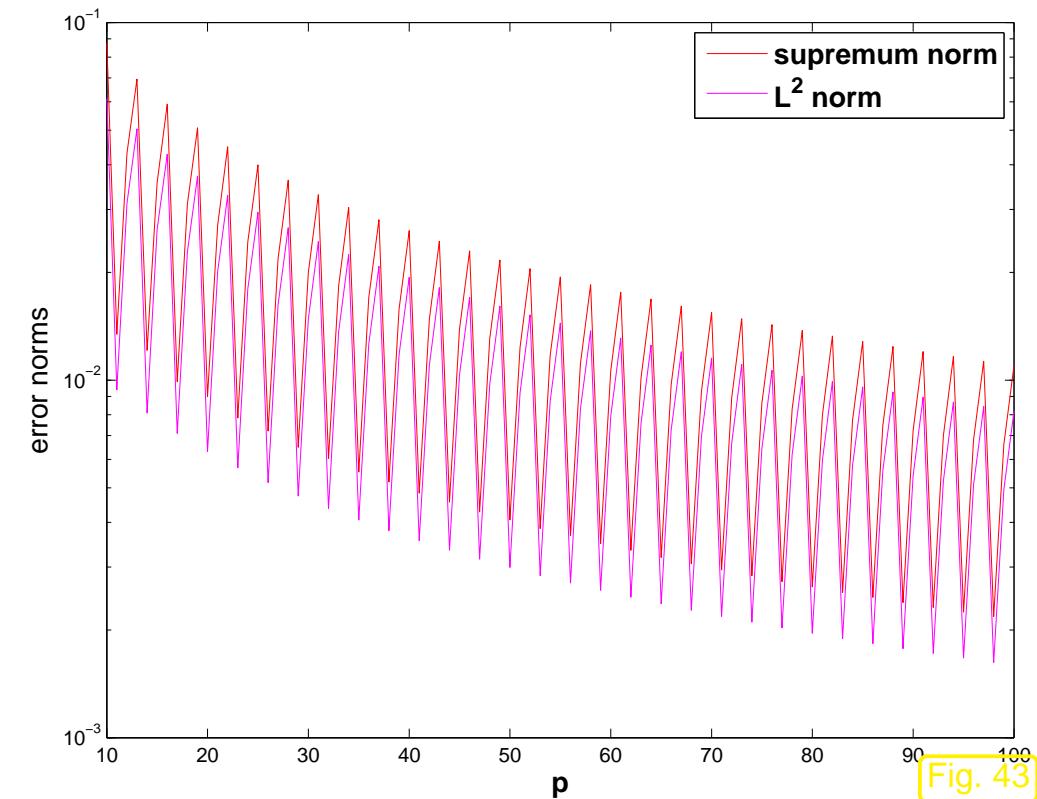
Example 1.6.23 (Convergence and smoothness of solution).

- $\Omega =]0, 1[$ (for finite differences), $\Omega =]-1, 1[$ (for spectral Galerkin),
exact solution of 2-point BVP for ODE $-\frac{d^2u}{dx^2} = g(x)$,

$$u(x) = \begin{cases} \frac{3}{4} - x^2 & , \text{ if } |x| < \frac{1}{2} , \\ 1 - |x| & , \text{ if } |x| \geq \frac{1}{2} . \end{cases} \quad \leftrightarrow \quad g(x) = \begin{cases} 2 & , \text{ if } |x| < \frac{1}{2} , \\ 0 & \text{elsewhere} . \end{cases}$$



① Finite Difference Method



② Spectral Galerkin Method

- Observations:
- no more exponential convergence of spectral Galerkin
 - FD: different rate of algebraic convergence for even/odd M !



Second-order Scalar Elliptic Boundary Value Problems

Preface

The previous chapter discussed the transformation of a minimization problem on a function space via a variational problem to a differential equation. To begin with, in Sect. 2.1–Sect. 2.4, this chapter revisits this theme for models that naturally rely on function spaces over domains in two and three spatial dimensions. Thus the transformation leads to genuine partial differential equations.

Sect. 2.2 ventures into the realm of Sobolev spaces, which provide the framework for rigorous mathematical investigation of variational equations. However, we will approach Sobolev spaces as “spaces

of physically meaningful solutions” or “spaces of solutions with finite energy”. From this perspective dealing with Sobolev spaces will be reduced to dealing with their norms.

In Sect. 2.5, we change tack and consider a physical phenomenon (heat conduction) where modelling naturally leads to partial differential equations. On this occasion, we embark on a general discussion of boundary conditions in Sect. 2.6.

Then the fundamental class of second-order elliptic boundary value problems is introduced. Appealing to “intuitive knowledge” about the physical systems underlying the models, key properties of their solutions are presented in Sect. 2.7.

In 2.6 Sect. in the context of stationary heat conduction we introduce the whole range of standard boundary conditions for 2nd-order elliptic boundary value problems. Their discussion in variational context will be resumed in Sect. 2.9.

The traditional concept of a boundary value problem for a partial differential equation:

Boundary value problem (BVP)

Given a partial differential operator \mathcal{L} , a domain $\Omega \subset \mathbb{R}^d$, a boundary differential operator \mathcal{B} , boundary data g , and a source term f , seek a function $u : \Omega \mapsto \mathbb{R}^n$ such that

$$\begin{aligned}\mathcal{L}(u) &= f \quad \text{in } \Omega , \\ \mathcal{B}(u) &= g \quad \text{on part of (or all) boundary } \partial\Omega .\end{aligned}$$

Terminology:

boundary value problem is scalar $\Leftrightarrow n = 1$
(in this case the unknown is a real valued function)

What does **elliptic** mean ?

Mathematical theory of PDEs distinguishes three main classes of boundary value problems (**BVPs**)
for partial differential equations (**PDE**):

- Elliptic BVPs (\Rightarrow “equilibrium problems”, as discussed in Sects. 1.2.3, 2.1.1, 2.1.2)
- Parabolic initial boundary value problems (IBVPs) (\Rightarrow evolution towards equilibrium)
- Hyperbolic IBVPs, among them wave propagation problems and conservation laws (\Rightarrow transport/propagation)

The rigorous mathematical definition is complicated and often fails to reveal fundamental properties of, e.g., solutions that are intuitively clear against the backdrop of the physics modelled by a certain PDE. Further discussion of classification in [3, § 1] and [11, Ch. 1].

\Rightarrow In the spirit of Sect. 1.1

Structural properties of a BPV inherited from the modelled system are more important than formal mathematical classification.

2.1 Equilibrium models

We only consider stationary systems. Then, frequently, see Sect. 1.2.2

equilibrium = minimal energy configuration of a system

Example: elastic string model of Sect. 1.2 (minimization of energy functional $J(\mathbf{u})$, see (1.2.17))

Now we study minimization problems for energy functional on spaces of functions $\Omega \mapsto \mathbb{R}$, where $\Omega \subset \mathbb{R}^d$ is a bounded (spatial) domain and $d = 2, 3$.

2.1.1 Taut membrane

Recall: energy functional for pinned *taut* string under gravitational load \hat{g} , see (1.4.7), in terms of displacement, see Fig. 16:

$$J(u) := \frac{1}{2} \int_a^b \hat{\sigma}(x) \left| \frac{du}{dx}(x) \right|^2 - \hat{g}(x)u(x) dx , \quad u \in C_{\text{pw}}^1([a, b]) , \\ u(a) = u_a , u(b) = u_b .$$

“2D generalization” of an elastic string \Rightarrow elastic membrane.

Taut drum membranes



[Fig. 44]

Shape of membrane



Graph of $u : \Omega \mapsto \mathbb{R}$

“membrane” on spatial domain $\Omega =]0, 1[^2$
($\text{---} \hat{=} \text{ boundary data}$)

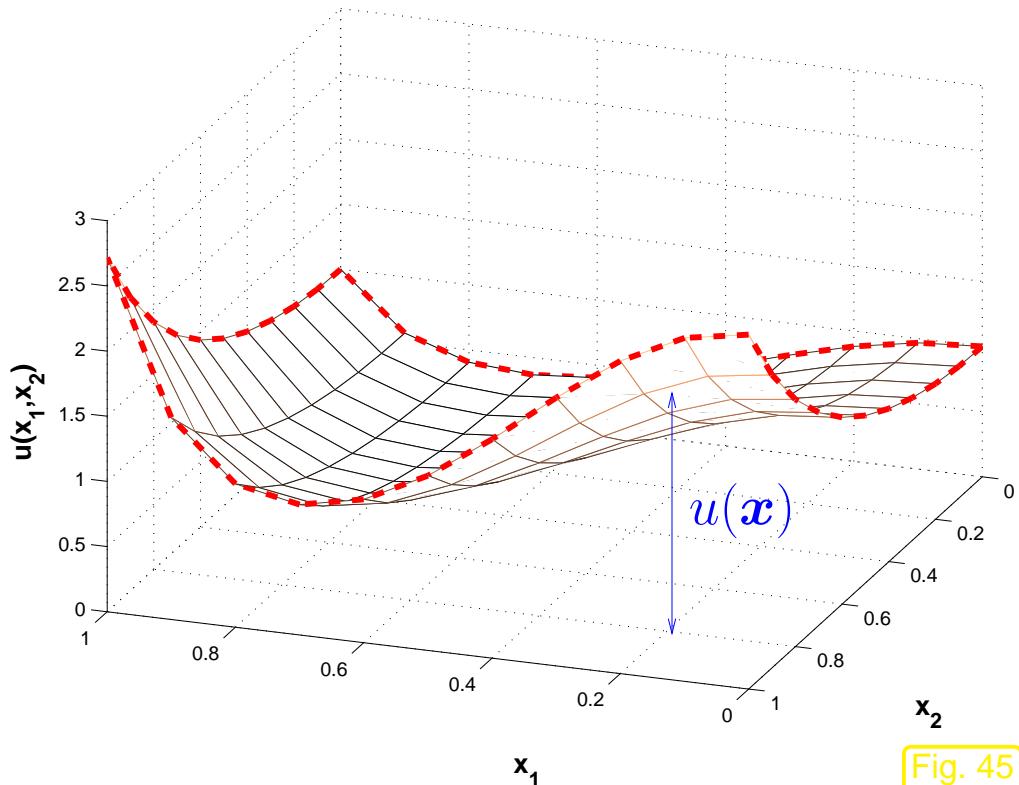


Fig. 45

Remark 2.1.1 (Spatial domains).

General assumptions on spatial domains $\Omega \subset \mathbb{R}^d$:

☞ $d = 1, 2, 3 \hat{=} \text{"dimension" of domain}$

• Ω is bounded

$$\text{diam}(\Omega) := \sup\{\|\mathbf{x} - \mathbf{y}\| : \mathbf{x}, \mathbf{y} \in \Omega\} < \infty ,$$

• Ω has piecewise smooth boundary $\partial\Omega$

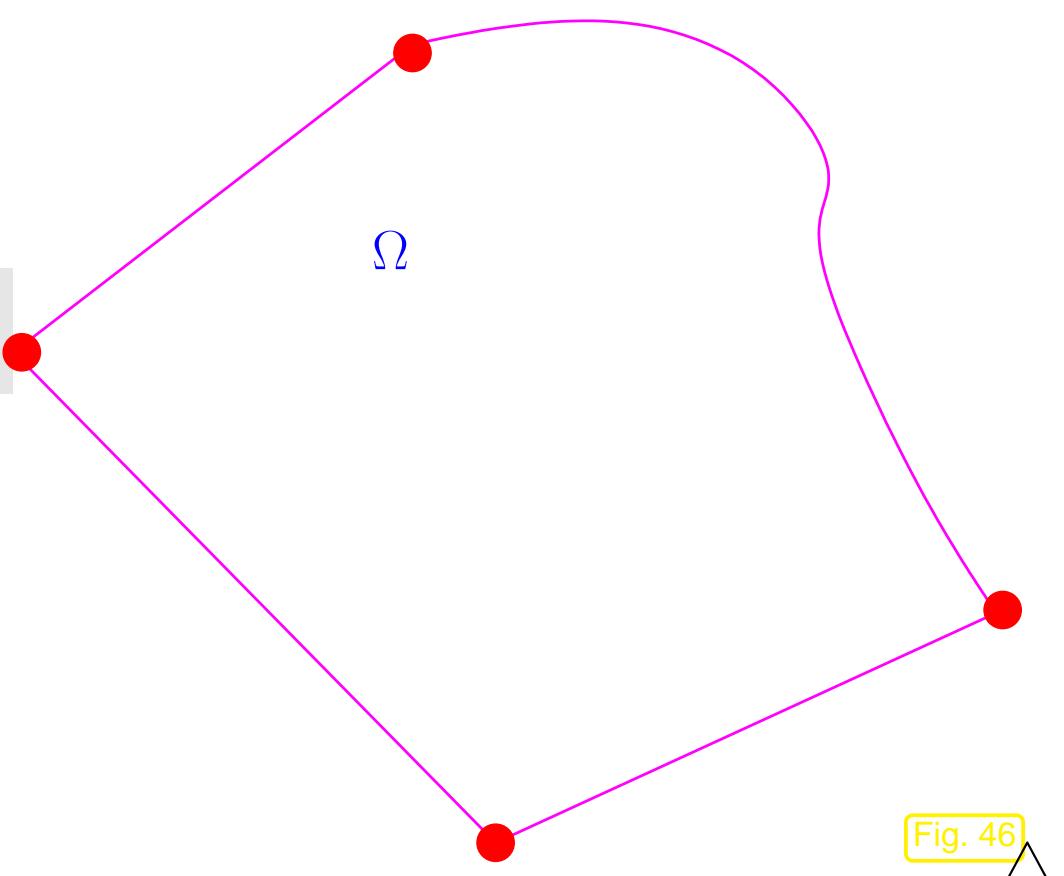


Fig. 46

Pinning conditions (**boundary conditions**), cf. (1.2.1), (1.4.11):

$$\begin{array}{lll} \text{fix} & u(\mathbf{x}) = g(\mathbf{x}) & \mathbf{x} \in \partial\Omega \\ & \Updownarrow & \\ & u|_{\partial\Omega} = g & \text{on } \partial\Omega . \end{array} \quad \text{for some } g \in C^0(\partial\Omega) . \quad (2.1.2)$$

☞ notation: $\partial\Omega \hat{=} \text{boundary of } \Omega$

(2.1.2) means that the displacement of the membrane over $\partial\Omega$ is provided by a prescribed *continuous* function $g : \partial\Omega \mapsto \mathbb{R}$: the membrane is clamped into a rigid frame.

Intuition:

g has to be continuous, unless the membrane is to be torn!
(Further discussion in Rem. 2.9.4)

► configuration space $V = \left\{ \begin{array}{l} \text{continuous functions } u \in C^0(\Omega), \\ \text{with } u|_{\partial\Omega} = g. \end{array} \right\}$

Think of the membrane as a grid of taut strings. Together with Rem. 1.4.10 this justifies the following expression for its total potential energy.

Potential energy of a taut membrane (described by $u \in C^0(\Omega)$) under vertical loading:

$$J_M(u) := \int_{\Omega} \frac{1}{2} \sigma(x) \|\mathbf{grad} u\|^2 - f(x)u(x) dx , \quad (2.1.3)$$

elastic energy

potential energy in force field

Note that

$$\sigma(\mathbf{x}) \|\mathbf{grad} u\|^2 = \sigma(x_1, x_2) \left| \frac{\partial u}{\partial x_1}(x_1, x_2) \right|^2 + \sigma(x_1, x_2) \left| \frac{\partial u}{\partial x_2}(x_1, x_2) \right|^2,$$

which justifies calling the taut membrane a “two-dimensional string under tension”.

with

- $u : \Omega \mapsto \mathbb{R}$ $\hat{=}$ displacement function, see Fig. 45, $[u] = \text{m}$,
- $f : \Omega \mapsto \mathbb{R}$ $\hat{=}$ force **density** (pressure), $[f] = \text{N m}^{-2}$,
- $\sigma : \Omega \mapsto \mathbb{R}^+$ $\hat{=}$ stiffness, $[\sigma] = \text{J}$.

Displacement of taut membrane in **equilibrium** achieves minimal potential energy, cf. (1.2.17)

$$u_* = \operatorname{argmin}_{u \in V} J_M(u). \quad (2.1.4)$$

Remark 2.1.5 (Minimal regularity of membrane displacement).

Smoothness required for \underline{u} , \underline{f} to render $J_M(u)$ from (2.1.3) meaningful, cf. Sect. 1.3.2:

- $\underline{u} \in C_{\text{pw}}^1(\Omega)$ is sufficient for displacement \underline{u} ,
- $\sigma, f \in C_{\text{pw}}^0(\Omega)$ already allows integration.



2.1.2 Electrostatic fields

- metal body in metal box
- prescribed voltage drop body—box

Sought: electric field $\mathbf{E} : \Omega \mapsto \mathbb{R}^3$ in $\Omega \subset \mathbb{R}^3$
($\Omega \hat{=} \text{blue region}$ ▷)

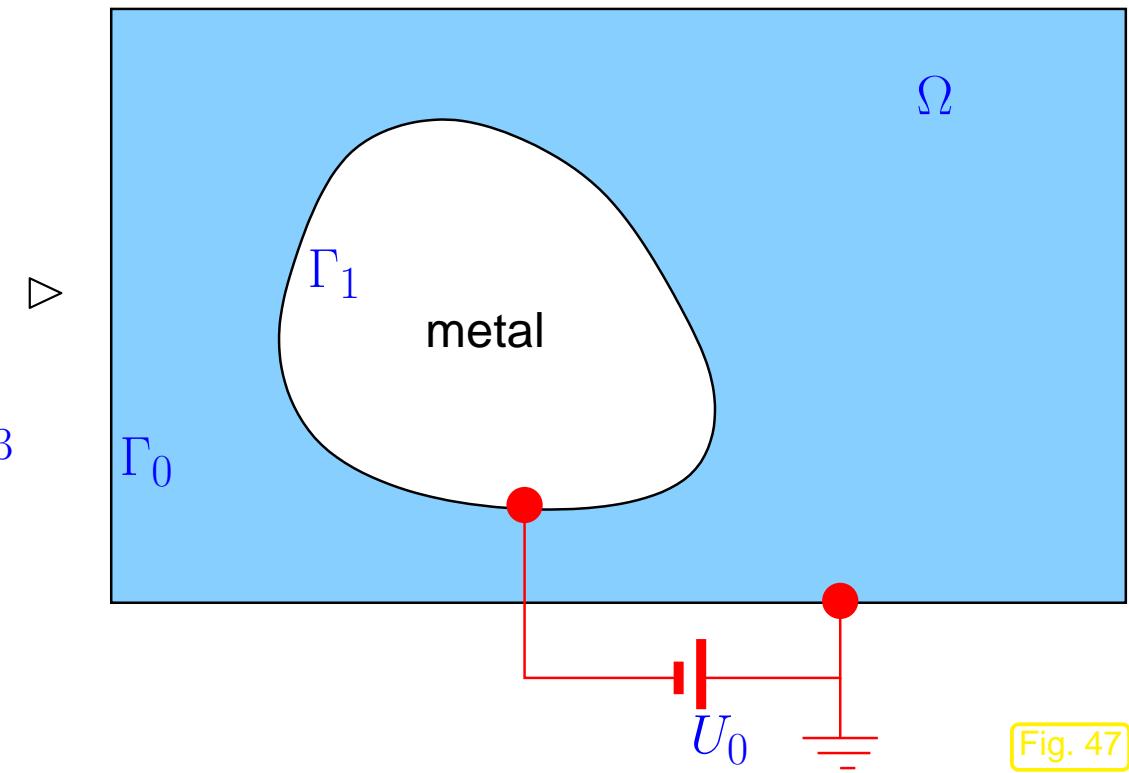
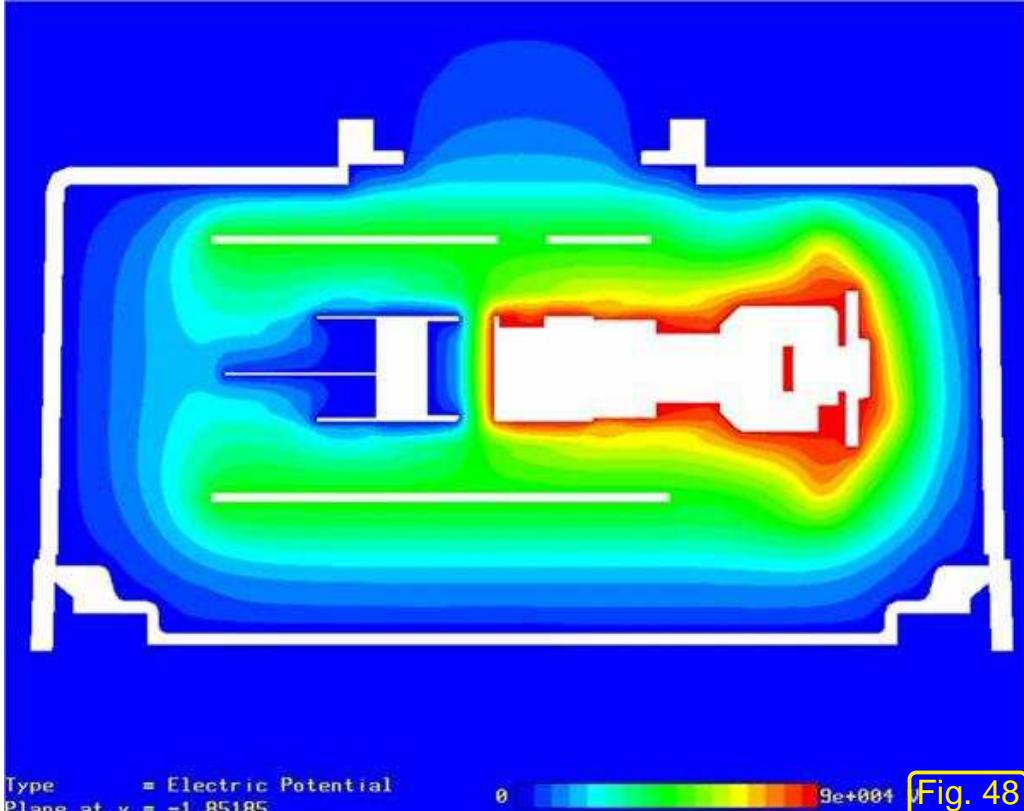


Fig. 47



From Maxwell's equations, static case:

$$\mathbf{E} = - \operatorname{grad} u , \quad (2.1.6)$$

where $u : \Omega \mapsto \mathbb{R} \hat{=} \text{electric (scalar) potential}$,
 $[u] = 1\text{V}$

◁ Electric potential in technical device ▷

Recall the definition of the **gradient** of a function $F : \Omega \subset \mathbb{R}^d \mapsto \mathbb{R}, F(\mathbf{x}) = F(x_1, \dots, x_d)$, see [19, Kap. 7], [14, Eq. 4.1.3]:

$$\operatorname{grad} F(\mathbf{x}) := \begin{pmatrix} \frac{\partial F}{\partial x_1} \\ \vdots \\ \frac{\partial F}{\partial x_d} \end{pmatrix} .$$

Note: the gradient at \mathbf{x} is a column vector of first *partial derivatives*,
 read $\operatorname{grad} F(\mathbf{x})$ as $(\operatorname{grad} F)(\mathbf{x})$; $\operatorname{grad} F$ is a vector valued function $\Omega \mapsto \mathbb{R}^d$.

Also in use (but not in this course) is the “ ∇ -notation”: $\nabla F(\mathbf{x}) := \text{grad } F(\mathbf{x})$.

Electromagnetic field energy: (electrostatic setting)

$$J_E(\mathbf{E}) = \frac{1}{2} \int_{\Omega} (\boldsymbol{\epsilon}(\mathbf{x}) \mathbf{E}(\mathbf{x})) \cdot \mathbf{E}(\mathbf{x}) \, d\mathbf{x} = \frac{1}{2} \int_{\Omega} (\boldsymbol{\epsilon}(\mathbf{x}) \text{grad } u(\mathbf{x})) \cdot \text{grad } u(\mathbf{x}) \, d\mathbf{x}, \quad (2.1.7)$$

where $\boldsymbol{\epsilon} : \Omega \mapsto \mathbb{R}^{3,3} \hat{=} \text{dielectric tensor}$, $\boldsymbol{\epsilon}(\mathbf{x})$ symmetric, $[\boldsymbol{\epsilon}] = \frac{\text{As}}{\text{Vm}}$.

- Symmetry of the dielectric tensor can always be assumed: if $\boldsymbol{\epsilon}(\mathbf{x})$ was not symmetric, then replacing it with $\frac{1}{2}(\boldsymbol{\epsilon}(\mathbf{x})^T + \boldsymbol{\epsilon}(\mathbf{x}))$ will yield exactly the same field energy.
- In terms of partial derivatives and tensor components $\boldsymbol{\epsilon}(\mathbf{x}) = (\epsilon_{ij})_{i,j=1}^3$ we have

$$(\boldsymbol{\epsilon}(\mathbf{x}) \text{grad } u(\mathbf{x})) \cdot \text{grad } u(\mathbf{x}) = \sum_{i=1}^3 \sum_{j=1}^3 \epsilon_{ij}(\mathbf{x}) \frac{\partial u}{\partial x_i}(\mathbf{x}) \frac{\partial u}{\partial x_j}(\mathbf{x}).$$

Fundamental property of dielectric tensor (for “normal” materials):

$$\exists 0 < \epsilon^- \leq \epsilon^+ < \infty: \quad \epsilon^- \|\mathbf{z}\|^2 \leq (\boldsymbol{\epsilon}(\mathbf{x})\mathbf{z}) \cdot \mathbf{z} \leq \epsilon^+ \|\mathbf{z}\|^2 \quad \forall \mathbf{z} \in \mathbb{R}^3, \forall \mathbf{x} \in \Omega . \quad (2.1.8)$$

Terminology: (2.1.8) \Leftrightarrow $\boldsymbol{\epsilon}$ is bounded and uniformly positive definite

Definition 2.1.9 (Uniformly positive (definite) tensor field).

An matrix-valued function $\mathbf{A} : \Omega \mapsto \mathbb{R}^{n,n}$, $n \in \mathbb{N}$, is called **uniformly positive definite**, if

$$\exists \alpha^- > 0: \quad (\mathbf{A}(\mathbf{x})\mathbf{z}) \cdot \mathbf{z} \geq \alpha^- \|\mathbf{z}\|^2 \quad \forall \mathbf{z} \in \mathbb{R}^n \quad (2.1.10)$$

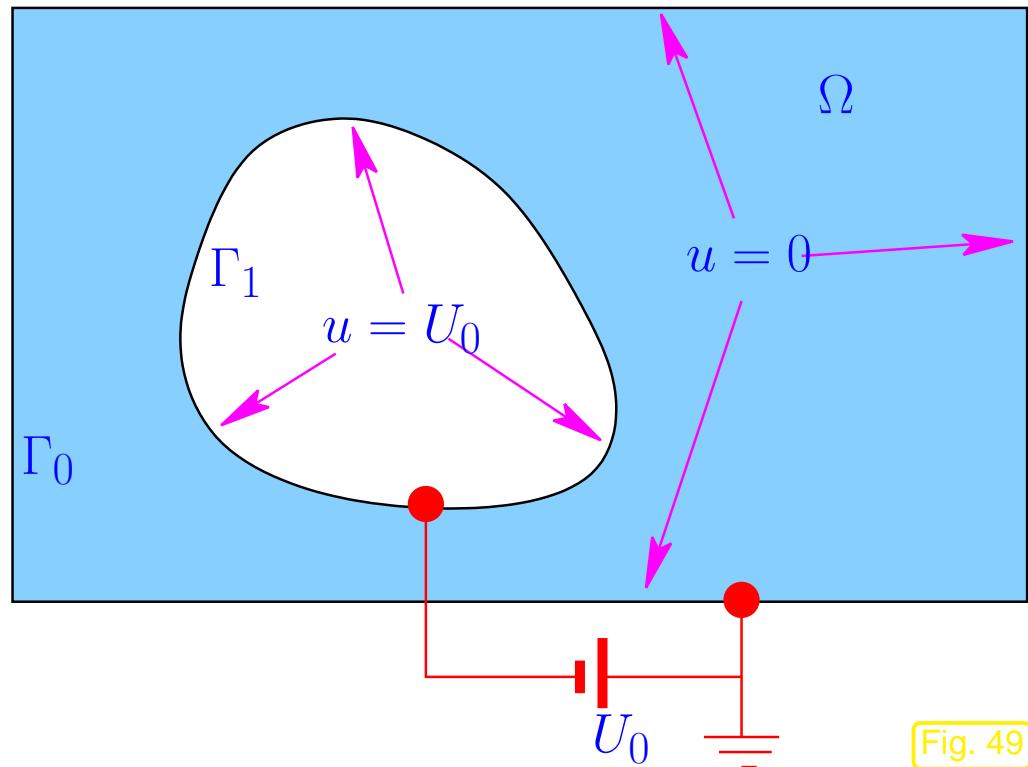
for almost all $\mathbf{x} \in \Omega$, that is, only with the exception of a set of volume zero.

If $\mathbf{A}(\mathbf{x})$ is symmetric, then we have the equivalence, cf. [14, Rem. 4.1.9],

$$(2.1.10) \Leftrightarrow \mathbf{A}(\mathbf{x}) \text{ s.p.d.} (\rightarrow [14, Def. 2.7.1]) \quad \text{and} \quad \lambda_{\min}(\mathbf{A}(\mathbf{x})) \geq \alpha^- .$$

What is the set/space V of admissible electric scalar potentials ?

Recall: in electrostatics surfaces of conducting bodies are **equipotential surfaces**



In the situation of Fig. 47:

Boundary conditions

$$\begin{aligned} u &= 0 \quad \text{on } \Gamma_0 , \\ u &= U_0 \quad \text{on } \Gamma_1 . \end{aligned} \tag{2.1.11}$$

$$V = \left\{ u \in C_{\text{pw}}^1(\Omega) , u \text{ satisfies (2.1.11)} \right\} .$$

to render $J_E(u)$ well defined, cf. Sect. 1.3.2.

Below, the notation $u = U$ will designate the boundary conditions (2.1.11).

$$u_* = \operatorname{argmin}_{u \in V} J_E(u) . \quad (2.1.12)$$

2.1.3 Quadratic minimization problems

Structure of minimization problems (equilibrium problems) encountered above:

$$\text{Sect. 2.1.1} \Rightarrow u_* = \operatorname{argmin}_{\substack{u \in C_{\text{pw}}^1(\Omega) \\ u=g \text{ on } \partial\Omega}} \underbrace{\frac{1}{2} \int_{\Omega} \sigma(\mathbf{x}) \|\mathbf{grad} u(\mathbf{x})\|^2 - f(\mathbf{x})u(\mathbf{x}) \, d\mathbf{x}}_{=:J_M(u), \text{ see (2.1.3)}} , \quad (2.1.13)$$

$$\text{Sect. 2.1.2} \Rightarrow u_* = \operatorname{argmin}_{\substack{u \in C_{\text{pw}}^1(\Omega) \\ u=U \text{ on } \partial\Omega}} \underbrace{\frac{1}{2} \int_{\Omega} (\epsilon(\mathbf{x}) \mathbf{grad} u(\mathbf{x})) \cdot \mathbf{grad} u(\mathbf{x}) \, d\mathbf{x}}_{=:J_E(u), \text{ see (2.1.7)}} . \quad (2.1.14)$$

Evidently, (2.1.13) and (2.1.14) share a common structure. It is the *same* structure we have already come across in the minimization problem (1.4.2) for the taut string model in Sect. 1.4.

Definition 2.1.15 (Quadratic functional).

A *quadratic functional* on a real vector space V_0 is a mapping $J : V_0 \mapsto \mathbb{R}$ of the form

$$J(u) := \frac{1}{2}a(u, u) + \ell(u) + c, \quad u \in V_0, \quad (2.1.16)$$

where $a : V_0 \times V_0 \mapsto \mathbb{R}$ is a *symmetric bilinear form* (\rightarrow Def. 1.3.11), $\ell : V_0 \mapsto \mathbb{R}$ a *linear form*, and $c \in \mathbb{R}$.

Recall: A bilinear form $a : V_0 \times V_0 \mapsto \mathbb{R}$ is *symmetric*, if

$$a(u, v) = a(v, u) \quad \forall u, v \in V_0. \quad (2.1.17)$$

Definition 2.1.18 (Quadratic minimization problem).

A *minimization problem*

$$w_* = \underset{w \in V_0}{\operatorname{argmin}} J(w)$$

is called a *quadratic minimization problem*, if J is a quadratic functional on a real vector space V_0 .

Hey, both (2.1.13) and (2.1.14) are no genuine quadratic minimization problems, because they are posed over affine spaces (= “vector space + offset function”, cf. (1.3.12))!

“Offset function trick”, c.f. (1.3.14), resolves the mismatch: for quadratic form J from (2.1.16)

$$\begin{aligned} J(u + u_0) &= \frac{1}{2}\mathbf{a}(u + u_0, u + u_0) + \ell(u + u_0) + c \\ &= \frac{1}{2}\mathbf{a}(u, u) + \underbrace{\mathbf{a}(u, u_0) + \ell(u)}_{=: \tilde{\ell}(u)} + \underbrace{\ell(u_0) + c}_{=: \tilde{c}} =: \tilde{J}(u) , \end{aligned}$$

due to the bilinearity of \mathbf{a} and the linearity of ℓ .

► $\underset{u \in u_0 + V_0}{\operatorname{argmin}} J(u) = u_0 + \underset{w \in V_0}{\operatorname{argmin}} J(w + u_0) = u_0 + \underset{w \in V_0}{\operatorname{argmin}} \tilde{J}(w) . \quad (2.1.19)$

For a discussion of quadratic functionals on $\mathbb{R}^n \rightarrow [14, \text{Sect. 4.1.1}]$

Both (2.1.13) and (2.1.14) involve quadratic functionals. To see this apply the “offset function trick” from (2.1.19) in this concrete case: write $u = u_0 + w$ with an offset function u_0 that satisfies the boundary conditions and $w \in C_{0,\text{pw}}^1(\Omega)$, cf. (1.3.14).

(2.1.13) \rightarrow quadratic minimization problem (\rightarrow Def. 2.1.18) with, cf. (2.1.16),

$$a(w, v) = \int_{\Omega} \sigma(x) \operatorname{grad} w(x) \cdot \operatorname{grad} v(x) dx, \quad \ell(v) := a(u_0, v) - \int_{\Omega} f(x)v(x) dx. \quad (2.1.20)$$

(2.1.14) \rightarrow quadratic minimization problem (\rightarrow Def. 2.1.18) with, cf. (2.1.16),

$$a(w, v) = \int_{\Omega} \operatorname{grad} w(x)^T \epsilon(x) \operatorname{grad} v(x) dx, \quad \ell(v) := a(u_0, v). \quad (2.1.21)$$

In both cases: $V_0 = C_{0,\text{pw}}^1(\Omega)$

Can we conclude existence and uniqueness of solutions of the minimization problems (2.1.13) and (2.1.14) ?

Let us first tackle the issue of **uniqueness**:

Definition 2.1.22 (Positive definite bilinear form).

A (symmetric) bilinear form $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R}$ on a real vector space V_0 is **positive definite**, if

$$u \in V_0 \setminus \{0\} \iff \mathbf{a}(u, u) > 0.$$

For the special case $V_0 = \mathbb{R}^n$ any matrix $\mathbf{A} \in \mathbb{R}^{n,n}$ induces a bilinear form via

$$\mathbf{a}(\mathbf{u}, \mathbf{v}) := \mathbf{u}^T \mathbf{A} \mathbf{v} = (\mathbf{A} \mathbf{v}) \cdot \mathbf{u}, \quad \mathbf{u}, \mathbf{v} \in \mathbb{R}^n. \quad (2.1.23)$$

This connects the concept of a symmetric positive definite bilinear form to the more familiar concept of s.p.d. matrices (\rightarrow [14, Def. 2.7.1])

$$\mathbf{A} \text{ s.p.d.} \iff \mathbf{a} \text{ from (2.1.23) is symmetric, positive definite.}$$

Definition 2.1.24 (Energy norm). cf. [14, Def. 4.1.1]

A symmetric positive definite bilinear form $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R}$ (\rightarrow Def. 2.1.22) induces the **energy norm**

$$\|u\|_{\mathbf{a}} := (\mathbf{a}(u, u))^{1/2} .$$

Origin of the term “energy norm” is clear from the connection with potential energy (e.g., in membrane model and in the case of electrostatic fields, see (2.1.20), (2.1.21)), see above.

Next, we have to verify the norm axioms (N1), (N2), and (N3) from Def. 1.6.3:

- (N1) is immediate from Def. 2.1.22,
- (N2) follows from bilinearity of \mathbf{a} ,
- (N3) is a consequence of the **Cauchy-Schwarz** inequality: for any symmetric positive definite bilinear form

$$|\mathbf{a}(u, v)| \leq (\mathbf{a}(u, u))^{1/2} (\mathbf{a}(v, v))^{1/2} . \quad (2.1.25)$$

Example 2.1.26 (Quadratic functionals with positive definite bilinear form in 2D).

Analogy between quadratic functionals with positive definite bilinear form and parabolas:

$$\begin{array}{c} J(v) = \frac{1}{2}a(v, v) - \ell(v) \\ \uparrow \qquad \uparrow \qquad \uparrow \\ f(x) = \frac{1}{2}ax^2 + bx \end{array}$$

with $a > 0$!

graph of quadratic functional $\mathbb{R}^2 \mapsto \mathbb{R}$

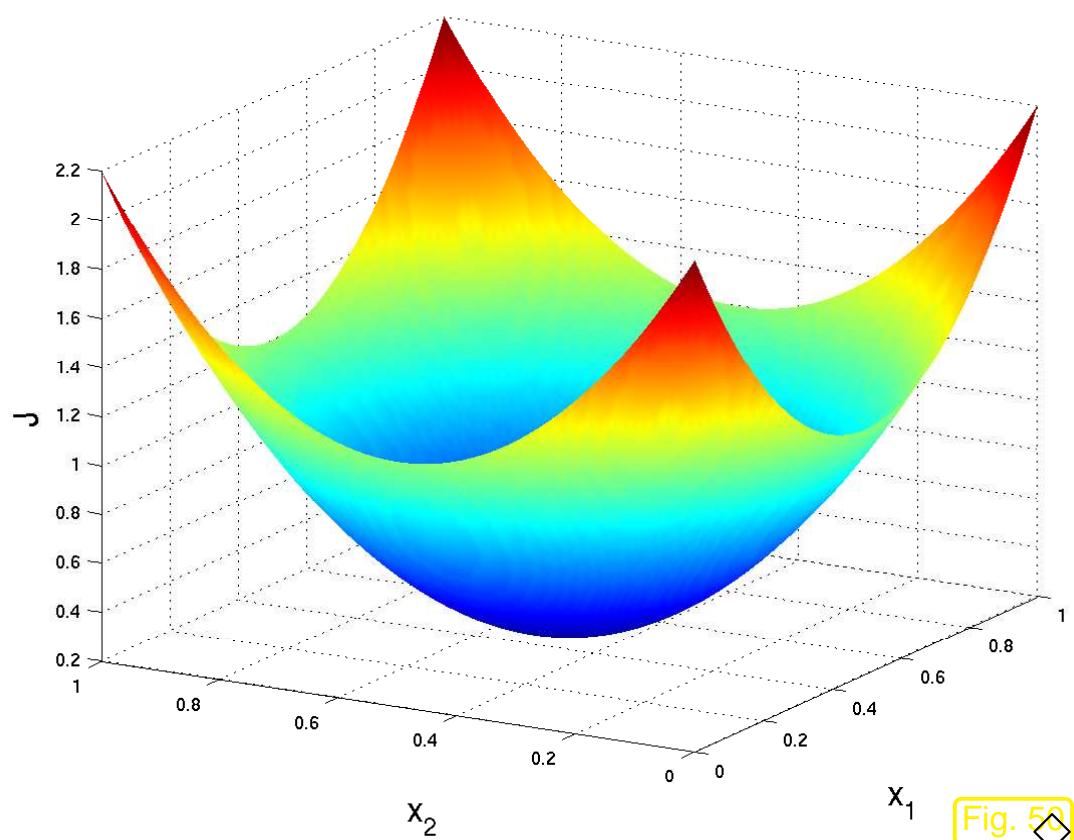


Fig. 58

Theorem 2.1.27 (Uniqueness of solutions of quadratic minimization problems).

If the bilinear form $a : V_0 \times V_0 \mapsto \mathbb{R}$ is **positive definite** (\rightarrow Def. 2.1.22), then any solution of

$$u_* = \underset{u \in V_0}{\operatorname{argmin}} J(u) , \quad J(u) = \frac{1}{2}a(u, u) + \ell(u) + c ,$$

is unique for any linear form $\ell : V_0 \mapsto \mathbb{R}$.

Proof. As in the proof of [14, Lemma 4.1.2], straightforward computations show

$$J(u) - J(u_*) = \frac{1}{2} \|u - u_*\|_a^2 .$$

The assertion of the theorem follows from norm axiom (N1), which holds for the energy norm. \square

Under the assumptions of the theorem, the quadratic functional J is **convex**, which is easily seen by considering the second derivative of the function

$$\varphi(t) := J(u + tv) \Rightarrow \ddot{\varphi}(t) = a(v, v) > 0 , \text{ if } v \neq 0 .$$

?

Is $\mathbf{a}(u, v) := \int_{\Omega} (\epsilon(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x}$ positive definite on $V_0 := C_{0,\text{pw}}^1(\Omega)$?

①: Since ϵ bounded and uniformly positive definite (\rightarrow Def. 2.1.9, (2.1.8))

$$\epsilon^- \int_{\Omega} \|\operatorname{grad} u(\mathbf{x})\|^2 \, d\mathbf{x} \leq \mathbf{a}(u, u) \leq \epsilon^+ \int_{\Omega} \|\operatorname{grad} u(\mathbf{x})\|^2 \, d\mathbf{x} \quad \forall u. \quad (2.1.28)$$

Hence, it is sufficient to examine the simpler bilinear form

$$\mathbf{d}(u, v) := \int_{\Omega} \operatorname{grad} u(\mathbf{x}) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x}, \quad u, v \in C_{0,\text{pw}}^1(\Omega). \quad (2.1.29)$$

②: Obviously $\mathbf{d}(u, u) = 0 \Rightarrow \operatorname{grad} u = 0 \Rightarrow u \equiv \text{const in } \Omega$

Observe: $u = 0 \text{ on } \partial\Omega \Rightarrow u = 0$



Zero boundary conditions are essential; otherwise one could add constants to the arguments of a without changing its value.

What about existence?

In a finite dimensional setting this is not a moot point, see Fig. 50 for a “visual proof”.

However, infinite dimensional spaces hold a lot of surprises and existence of solutions of quadratic minimization problems becomes a subtle issue, even if the bilinear form is positive definite.

Example 2.1.30 (Non-existence of solutions of positive definite quadratic minimization problem).

We consider the quadratic functional

$$J(u) := \int_0^1 \frac{1}{2} u^2(x) - u(x) dx = \frac{1}{2} \int_0^1 (u(\xi) - 1)^2 - 1 dx ,$$

on the space

$$V_0 := C_{0,\text{pw}}^0([0, 1])$$

It fits the abstract form from Def. 2.1.15 with

$$\mathbf{a}(u, v) = \int_0^1 u(x)v(x) dx , \quad \ell(v) = \int_0^1 v(x) dx .$$

The function $\varphi(\xi) = \frac{1}{2}\xi^2 - \xi = \frac{1}{2}\xi(1 - 2\xi) = \frac{1}{2}(\xi - 1)^2 - \frac{1}{2}$ has a global minimum at $\xi = 1$ and $\varphi(\xi) - \varphi(1) = \frac{1}{2}(\xi - 1)^2$.



$$|\eta - 1| > |\xi - 1| \Rightarrow \varphi(\eta) > \varphi(\xi).$$

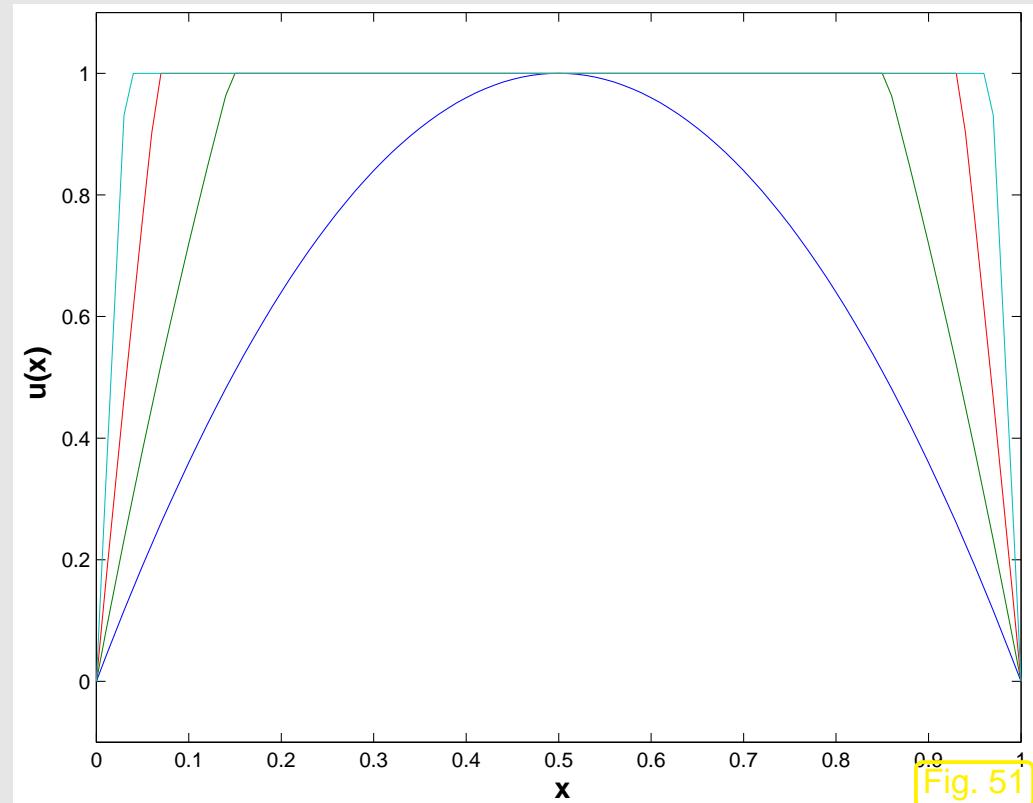
Assume that $u \in V_0$ is a global minimizer of J .

Then

$$w(x) := \min\{1, 2 \max\{u(x), 0\}\}, \\ 0 \leq x \leq 1,$$

is another function $\in C_{0,\text{pw}}^0([0, 1])$, which satisfies

$$u(x) \neq 1 \Rightarrow |w(x) - 1| < |u(x) - 1| \\ \Rightarrow J(w) < J(u) !$$



Hence, whenever we think we have found a minimizer $\in C_{0,\text{pw}}^0([0, 1])$, the formula provides another eligible function for which the value of the functional is even smaller!

The problem in this example seems to be that we have chosen “too small” a function space, c.f. Sect. 2.2 below.



2.2 Sobolev spaces

Mathematical theory is much concerned about proving existence of suitably defined solutions for minimization problems. As demonstrated in Ex. 2.1.30 this can encounter profound problems.

In this section we will learn about a class of *abstract function spaces* that has been devised to deal with the question of existence of solutions of quadratic minimization problems like (2.1.13) and (2.1.14). We can only catch of glimps of the considerations; thorough investigation is done in the mathematical field of *functional analysis*.

Consider a quadratic minimization problem (\rightarrow Def. 2.1.18) for a quadratic functional (\rightarrow Def. 2.1.15)

$$J : V_0 \mapsto \mathbb{R} , \quad J(u) = \frac{1}{2}\mathbf{a}(u, u) + \ell(u) + c ,$$

based on a symmetric positive definite (s.p.d.) bilinear form \mathbf{a} \rightarrow Def. 2.1.22.

It is clear that $J(V_0)$ is bounded from below, if

$$\exists C > 0: |\ell(u)| \leq C \|u\|_{\mathbf{a}} \quad \forall u \in V_0 , \quad (2.2.1)$$

where $\|\cdot\|_{\mathbf{a}}$ is the energy norm induced by \mathbf{a} , see Def. 2.1.24:

$$J(u) = \frac{1}{2}\mathbf{a}(u, u) - \ell(u) \geq \frac{1}{2}\|u\|_{\mathbf{a}}^2 - C \|u\|_{\mathbf{a}} \geq -\frac{1}{2}C^2 .$$

Remark: In mathematical terms (2.2.1) means that ℓ is **continuous** w.r.t. $\|\cdot\|_{\mathbf{a}}$

Under these conditions, the quadratic minimization problem for J should have a (unique, due to Thm. 2.1.27) solution, if it is considered on a space that is “large enough”.

Idea: for a quadratic minimization problem (\rightarrow Def. 2.1.18) with

- symmetric positive definite (s.p.d.) bilinear form a ,
- a linear form ℓ that is continuous w.r.t. $\|\cdot\|_a$, see (2.2.1),

posed over a function space follow the advice:

consider it on the largest space of functions for which a still makes sense !

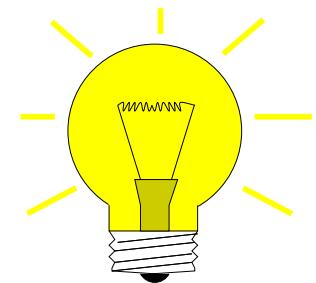
(and which complies with boundary conditions)

Choose “ $V_0 := \{\text{functions } v \text{ on } \Omega: a(v, v) < \infty\}$ ”

Example 2.2.2 (Space of square integrable functions). \rightarrow Ex. 2.1.30

Quadratic functional (related to J from Ex. 2.1.30):

$$J(u) := \int_{\Omega} \frac{1}{2} |u(\mathbf{x})|^2 - u(\mathbf{x}) \, d\mathbf{x} . \quad \left(u \in C_{\text{pw}}^0(\Omega) ? \right) \quad (2.2.3)$$



We follow the above recipe, which suggests to choose

► $V_0 := \{v : \Omega \mapsto \mathbb{R} \text{ integrable: } \int_{\Omega} |v(\boldsymbol{x})|^2 d\boldsymbol{x} < \infty\}$ (2.2.4)

Definition 2.2.5 (Space $L^2(\Omega)$).

*The function space defined in (2.2.4) is the **space of square-integrable functions** on Ω and denoted by $L^2(\Omega)$.*

It is a normed space with norm

$$\left(\|v\|_0 := \right) \|v\|_{L^2(\Omega)} := \left(\int_{\Omega} |v(\boldsymbol{x})|^2 d\boldsymbol{x} \right)^{1/2}.$$

Notation: $L^2(\Omega)$ ← superscript “2”, because square in the definition of norm $\|\cdot\|_0$

Note: obviously $C_{\text{pw}}^0(\Omega) \subset L^2(\Omega)$.

2.2



Remark 2.2.6 (Boundary conditions and $L^2(\Omega)$).

Ex. 2.1.30 vs. Ex. 2.2.2: Crying foul! (boundary conditions $u(0) = u(1) = 0$ in Ex. 2.1.30, but none in Ex. 2.2.2!)

Consider $u \in C^0([0, 1])$ and try to impose boundary values $u_0, u_1 \in \mathbb{R}$ by “altering” u :

$$\tilde{u}(x) = \begin{cases} u(x) + (1 - nx)(u_0 - u(0)) & , \text{ for } 0 \leq x \leq \frac{1}{n} , \\ u(x) & , \text{ for } \frac{1}{n} < x < 1 - \frac{1}{n} , \\ u(x) - n(1 - \frac{1}{n} - x)(u_1 - u(1)) & , \text{ for } 1 - \frac{1}{n} < x \leq 1 . \end{cases}$$

► $\tilde{u}(0) = u_0$, $\tilde{u}(1) = u_1$, $\|\tilde{u} - u\|_{L^2([0,1])}^2 = \frac{1}{3n}(u_0 + u_1 - u(0) - u(1)) \rightarrow 0$ for $n \rightarrow \infty$.

► Tiny perturbations of a function $u \in L^2([0, 1])$ (in terms of changing its L^2 -norm) can make it attain any value at $x = 0$ and $x = 1$.

Boundary conditions cannot be imposed in $L^2(\Omega)$!



Remark 2.2.7 (Quadratic minimization problems on Hilbert spaces).

On the function space $V_0 = L^2(\Omega)$ the quadratic minimization problem for the quadratic functional from (2.2.3) can be shown to possess a solution. Instrumental in the proof is the fact that $L^2(\Omega)$ is a **Hilbert space**, that is, a *complete* normed space.

This theory is beyond the scope of this course. For more explanations see [10, Ch. 5 and Sect. 6.2].



Now consider a quadratic minimization problem for the functional, *c.f.* (2.1.13),

$$J(u) := \int_{\Omega} \frac{1}{2} \|\mathbf{grad} u\|^2 - f(\mathbf{x})u(\mathbf{x}) \, d\mathbf{x} \quad \left(u \in C_{0,\text{pw}}^1(\Omega) ? \right) \quad (2.2.8)$$

What is the natural function space for this minimization problem? Again, we follow the above recipe, which suggests that we choose

► $V_0 := \{v : \Omega \mapsto \mathbb{R} \text{ integrable: } v = 0 \text{ on } \partial\Omega, \int_{\Omega} |\operatorname{grad} v(\boldsymbol{x})|^2 d\boldsymbol{x} < \infty\}$ (2.2.9)

Definition 2.2.10 (Sobolev space $H_0^1(\Omega)$).

The space defined in (2.2.9) is the **Sobolev space** $H_0^1(\Omega)$ with norm

$$|v|_{H^1(\Omega)} := \left(\int_{\Omega} \|\operatorname{grad} v\|^2 d\boldsymbol{x} \right)^{1/2}.$$

Notation: $H_0^1(\Omega)$

- ← superscript “1”, because first derivatives occur in norm
- ← subscript “0”, because zero on $\partial\Omega$

Note: $|\cdot|_{H^1(\Omega)}$ is the **energy norm** (\rightarrow Def. 2.1.24) associated with the bilinear form in the quadratic functional J from (2.2.8), cf. (2.1.16).

→ See Rem. 1.6.7 for a discussion of the relevance of the energy norm.

Remark 2.2.11 (Boundary conditions in $H_0^1(\Omega)$).

Rem. 2.2.6 explained why imposing boundary conditions on functions in $L^2(\Omega)$ does not make sense.

Yet, in (2.2.9) zero boundary conditions are required for v !

Discussion parallel to Rem. 2.2.6, but now with the norm $|\cdot|_{H^1(\Omega)}$ in mind: Consider $u \in C^1([0, 1])$ and try to impose boundary values $u_0, u_1 \in \mathbb{R}$ by “altering” u :

$$\tilde{u}(x) = \begin{cases} u(x) + (1 - nx)(u_0 - u(0)) & , \text{ for } 0 \leq x \leq \frac{1}{n}, \\ u(x) & , \text{ for } \frac{1}{n} < x < 1 - \frac{1}{n}, \\ u(x) - n(1 - \frac{1}{n} - x)(u_1 - u(1)) & , \text{ for } 1 - \frac{1}{n} < x \leq 1. \end{cases}$$

► $\tilde{u}(0) = u_0$, $\tilde{u}(1) = u_1$, BUT $|\tilde{u} - u|_{H^1(]0,1[)}^2 = n(u_0 + u_1 - u(0) - u(1)) \rightarrow \infty$ for $n \rightarrow \infty$

► Enforcing boundary values at $x = 0$ and $x = 1$ cannot be done without significantly changing the “energy” of the function.

However, the solutions of the quadratic minimization problems (2.1.13), (2.1.14) are to satisfy non-zero boundary conditions. They are sought in a larger Sobolev space, which arises from $H_0^1(\Omega)$ by dispensing with the requirement “ $v = 0$ on $\partial\Omega$ ”.

Definition 2.2.12 (Sobolev space $H^1(\Omega)$).

The Sobolev space

$$H^1(\Omega) := \{v : \Omega \mapsto \mathbb{R} \text{ integrable: } \int_{\Omega} |\operatorname{grad} v(x)|^2 dx < \infty\}$$

is a normed function space with norm

$$\|v\|_{H^1(\Omega)}^2 := \|v\|_0^2 + |v|_{H^1(\Omega)}^2.$$

- $H^1(\Omega)$ is the “maximal function space” on which both J_M and J_E from (2.1.13), (2.1.14) are defined.

Remark 2.2.13 ($|\cdot|_{H^1(\Omega)}$ -seminorm).

Note that $|\cdot|_{H^1(\Omega)}$ alone is no longer a norm on $H^1(\Omega)$, because for $v \equiv \text{const}$ obviously $|v|_{H^1(\Omega)} = 0$, which violates (N1). \triangle

In the introduction to this section we saw that a quadratic functional with s.p.d. bilinear form \mathbf{a} is bounded from below, if its linear form ℓ satisfies the continuity (2.2.1). Now, we discuss this for the quadratic functional J from (2.2.8) in lieu of J_M and J_E .

The quadratic functional J from (2.2.8) involves the linear form

$$\ell(u) := \int_{\Omega} f(\mathbf{x}) u(\mathbf{x}) \, d\mathbf{x} . \quad (2.2.14)$$

$f \doteq$ load function $\Rightarrow f \in C_{pw}^0(\Omega)$ should be admitted.

Crucial question:

Is ℓ from (2.2.14) continuous on $H_0^1(\Omega)$?
 \Updownarrow (c.f. (2.2.1))

$$\exists C > 0: |\ell(u)| \leq C|u|_{H^1(\Omega)} \quad \forall u \in H_0^1(\Omega) ? .$$

To begin with, we use the Cauchy-Schwarz inequality (2.1.25) for integrals, which implies

$$|\ell(u)| = \left| \int_{\Omega} f(\mathbf{x}) u(\mathbf{x}) \, d\mathbf{x} \right| \leq \left(\int_{\Omega} |f(\mathbf{x})|^2 \, d\mathbf{x} \right)^{1/2} \left(\int_{\Omega} |u(\mathbf{x})|^2 \, d\mathbf{x} \right)^{1/2} = \underbrace{\|f\|_0}_{<\infty} \|u\|_0 . \quad (2.2.15)$$

This reduces the problem to bounding $\|u\|_0$ in terms of $|u|_{H^1(\Omega)}$.

Theorem 2.2.16 (First Poincaré-Friedrichs inequality).

If $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, is bounded, then

$$\|u\|_0 \leq \text{diam}(\Omega) \|\mathbf{grad} u\|_0 \quad \forall u \in H_0^1(\Omega) .$$

Proof. The proof employs a powerful technique in the theoretical treatment of function spaces: exploit **density** of smooth functions (which, by itself, is a deep result).

It boils down to the insight:

In order to establish inequalities between continuous functionals on Sobolev spaces of functions on Ω it often suffices to show the target inequality for smooth functions in $C_0^\infty(\Omega)$ or $C^\infty(\Omega)$, respectively.

☞ notation: $C_0^\infty(\Omega) \hat{=} \text{smooth functions with (compact) support} (\rightarrow \text{Def. 1.5.53}) \text{ inside } \Omega$

In the concrete case (note the zero boundary values inherent in the definition of $H_0^1(\Omega)$) we have to establish the first Poincaré-Friedrichs inequality for functions $u \in C_0^\infty(\Omega)$ only.

For the sake of simplicity the proof is elaborated for $d = 1$, $\Omega = [0, 1]$. It merely employs elementary results from calculus throughout, namely the Cauchy-Schwarz inequality (2.2.15) and the fundamental theorem of calculus [19, Satz 6.3.4], see (2.4.1):

$$\forall u \in C_0^\infty([0, 1]): \quad u(x) = \underbrace{u(0)}_{=0} + \int_0^x \frac{du}{dx}(\tau) d\tau, \quad 0 \leq x \leq 1.$$

► $\|u\|_0^2 = \int_0^1 \left| \int_0^x \frac{du}{dx}(\tau) d\tau \right|^2 dx \stackrel{(2.2.15)}{\leq} \int_0^1 \left(\int_0^x 1 d\tau \cdot \int_0^x \left| \frac{du}{dx}(\tau) \right|^2 d\tau \right) dx \leq \left\| \frac{du}{dx} \right\|_0^2.$

Taking the square root finished the proof in 1D. □

► If $f \in L^2(\Omega)$, then $\ell(u) = \int_\Omega f u \, dx$ is a continuous linear functional on $H_0^1(\Omega)$.

Here “continuity” has to be read as

$$\exists C > 0: |\ell(u)| \leq C |u|_{H_0^1(\Omega)}, \quad \forall u \in H_0^1(\Omega), \tag{2.2.1}$$

Most concrete results about Sobolev spaces boil down to relationships between their norms. The spaces themselves remain intangible, but the norms are very concrete and can be computed and manipulated as demonstrated above.

Do not be afraid of Sobolev spaces!

It is only the norms that matter for us, the ‘spaces’ are irrelevant!

Sobolev spaces = “concept of convenience”: the minimization problem seeks its own function space.

Minimization problem

$$u = \operatorname{argmin}_{v: \Omega \rightarrow \mathbb{R}} J(v)$$



“Maxmimal” function space
on which J is defined
(Sobolev space)

Then, why do you bother me with these uncanny “Sobolev spaces” after all ?

- Anyone involved in CSE must be able to understand mathematical publications on numerical methods for PDEs, Those regularly resort to the concept of Sobolev spaces to express their findings.
- The statement that a function belongs to a certain Sobolev space can be regarded as a concise way of describing quite a few of its essential properties.

Let us elucidate the second point:

Theorem 2.2.17 (Compatibility conditions for piecewise smooth functions in $H^1(\Omega)$).

Let Ω be partitioned into sub-domains Ω_1 and Ω_2 . A function that is continuously differentiable in both sub-domains and continuous up to their boundary, belongs to $H^1(\Omega)$, if and only if u is continuous on Ω .

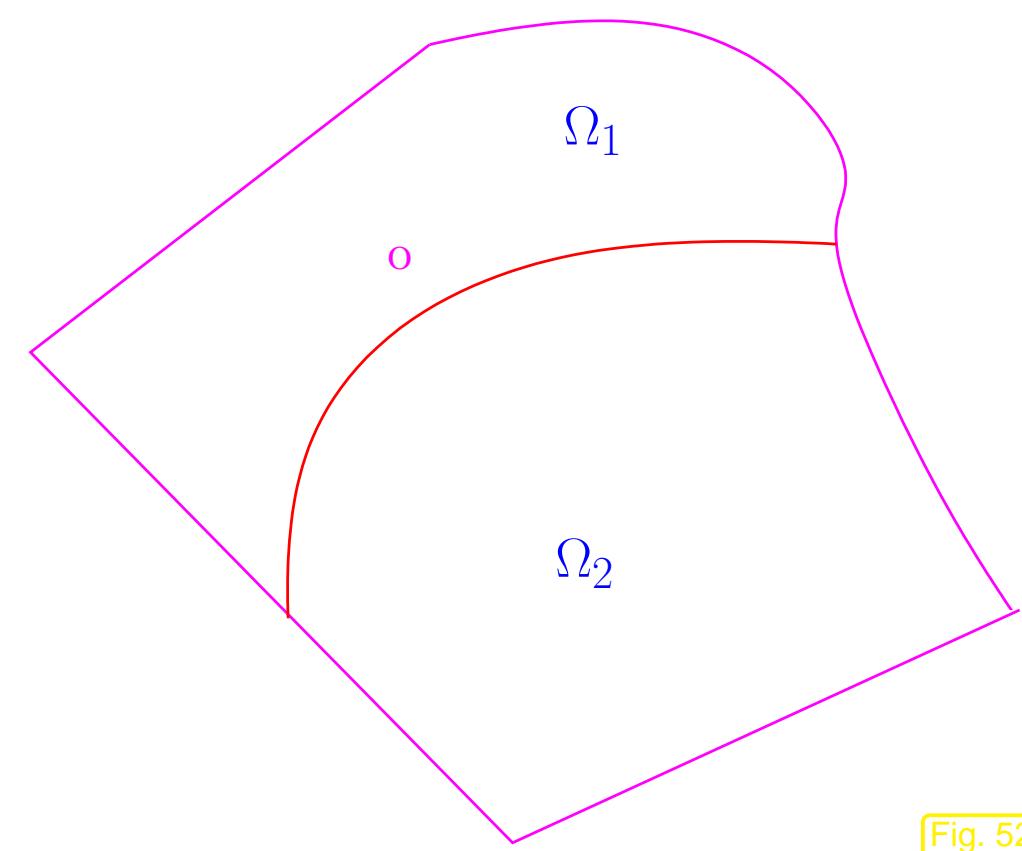
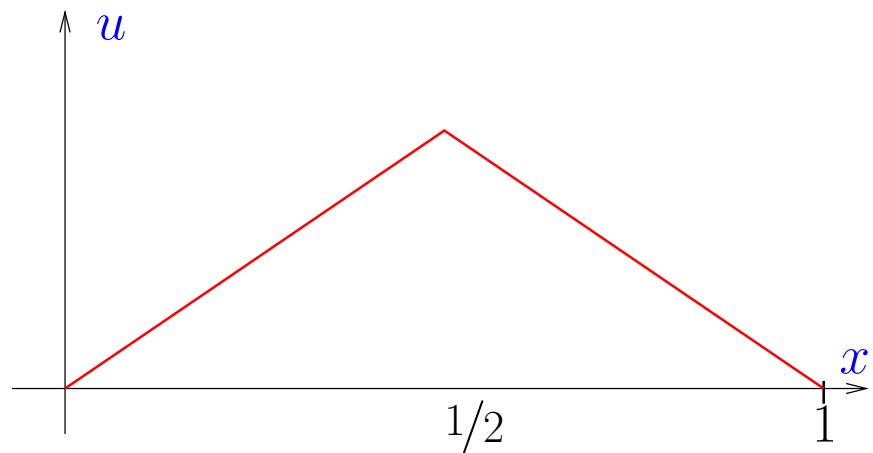


Fig. 52

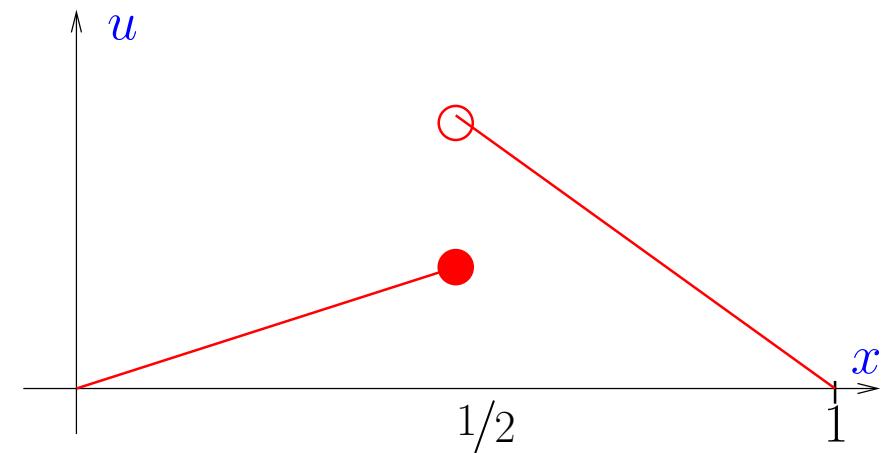
The proof of this theorem requires the notion of *weak derivatives* that will not be introduced in this course.

Example 2.2.18 (Piecewise linear functions (not) in $H_0^1(]0, 1[)$).

We conclude from Thm. 2.2.17:



$$u \in H_0^1(]0, 1[)$$



$$u \notin H_0^1(]0, 1[)$$

◇

From Thm. 2.2.17 we conclude

$$C_{\text{pw}}^1([a, b]) \subset H^1(]a, b[) \text{ and } C_{0, \text{pw}}^1([a, b]) \subset H_0^1(]a, b[)$$

2.2

p. 197

Thm. 2.2.17 provides a simple recipe for computing the norm $|u|_{H^1(\Omega)}$ of a piecewise C^1 -function that is continuous in all of Ω .

Corollary 2.2.19 (H^1 -norm of piecewise smooth functions).

Under the assumptions of Thm. 2.2.17 we have for a continuous, piecewise smooth function $u \in C^0(\Omega)$

$$|u|_{H^1(\Omega)}^2 = |u|_{H^1(\Omega_1)}^2 + |u|_{H^1(\Omega_2)}^2 = \int_{\Omega_1} |\mathbf{grad} u(\mathbf{x})|^2 d\mathbf{x} + \int_{\Omega_2} |\mathbf{grad} u(\mathbf{x})|^2 d\mathbf{x}.$$

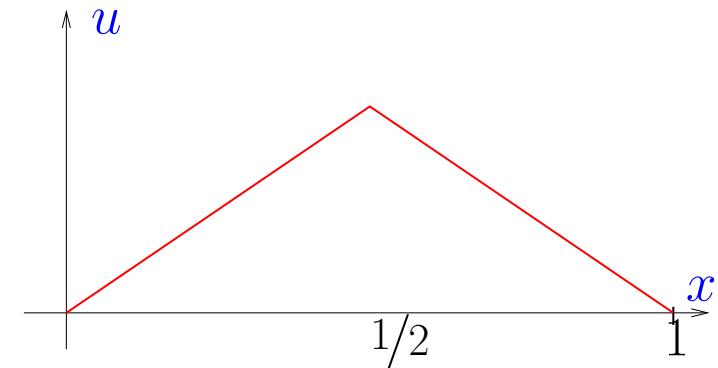
Actually, this is not new, see Sect. 1.3.2: earlier we already evaluated the elastic energy functionals (1.2.16), (1.4.2) for functions in $C_{\text{pw}}^1([0, 1])$ by “piecewise differentiation” followed by integration of the resulting discontinuous function.

Example 2.2.20 (Non-differentiable function in $H_0^1(]0, 1[)$).

$d = 1, \Omega =]0, 1[$:

“Tent function”

$$u(x) = \begin{cases} 2x & \text{for } 0 < x < 1/2, \\ 2(1-x) & \text{for } 1/2 < x < 1. \end{cases}$$



Compute

$$|u|_{H^1(\Omega)}^2 = \int_0^1 |u'(x)|^2 dx = 4 < \infty.$$

► Example for a $u \in H_0^1(]0, 1[)$, which is not globally differentiable.



If you are still feeling uneasy when dealing with Sobolev spaces, do not hesitate to think of the following replacements

$$L^2(\Omega) \rightarrow C_{\text{pw}}^0(\Omega) , \quad H_0^1(\Omega) \rightarrow C_{0,\text{pw}}^1(\Omega) .$$

2.3 Variational formulations

2.3.1 Linear variational problems

Recall: derivation of variational formulation (1.4.3) from taut string minimization problem (1.4.2) in Sect. 1.4.

No surprise: (2.1.13) & (2.1.14) are amenable to the same approach:

Calculus of variations → Sect. 1.3.1: “Directional derivative” of J_E :

$$\begin{aligned} J_E(u + tv) - J_E(u) &= \frac{1}{2} \int_{\Omega} (\epsilon(\mathbf{x}) \mathbf{grad}(u + tv)) \cdot \mathbf{grad}(u + tv) d\mathbf{x} \\ &\stackrel{(*)}{=} \frac{1}{2} \int_{\Omega} (\epsilon(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} u + 2t(\epsilon(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} v + \\ &\quad t^2(\epsilon(\mathbf{x}) \mathbf{grad} v) \cdot \mathbf{grad} v - (\epsilon(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} u d\mathbf{x} \\ &= t \int_{\Omega} (\epsilon(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} v d\mathbf{x} + O(t^2) \quad \text{for } t \rightarrow 0 . \end{aligned}$$

(*): due to the symmetry of $\epsilon(\mathbf{x})$: $(\epsilon \mathbf{grad} u) \cdot \mathbf{grad} v = (\epsilon \mathbf{grad} v) \cdot \mathbf{grad} u$!

►
$$\lim_{t \rightarrow 0} \frac{J_E(u + tv) - J_E(u)}{t} = \int_{\Omega} (\epsilon(\mathbf{x}) \mathbf{grad} u(\mathbf{x})) \cdot \mathbf{grad} v(\mathbf{x}) d\mathbf{x} ,$$

for perturbation functions

$v \in H_0^1(\Omega)$, see Def. 2.2.10

The requirement $v = 0$ on $\partial\Omega$ reflects the fact that we may not perturb u on the boundary, lest the prescribed boundary values be violated.

As explained in Sect. 1.3.1 (“idea of calculus of variations”), this leads to the following variational problem equivalent to (2.1.14)

$$\begin{aligned} u \in H^1(\Omega) , \\ u = U \text{ on } \partial\Omega : \int_{\Omega} (\epsilon(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} = 0 \quad \forall v \in H_0^1(\Omega) . \end{aligned} \quad (2.3.1)$$

For the membrane problem (2.1.13) we arrive at

$$\begin{aligned} u \in H^1(\Omega) , \\ u = g \text{ on } \partial\Omega : \int_{\Omega} \sigma(\mathbf{x}) \operatorname{grad} u(\mathbf{x}) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \quad \forall v \in H_0^1(\Omega) . \end{aligned} \quad (2.3.2)$$

Both, (2.3.1) and (2.3.2) have a common structure, expressed in the following variational problem:

Variational formulation of 2nd-order elliptic (Dirichlet) minimization problems:

$$\begin{array}{l} u \in H^1(\Omega), \\ u = g \text{ on } \partial\Omega \end{array} : \int_{\Omega} (\alpha(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega). \quad (2.3.3)$$

Symmetric uniformly positive definite **material tensor** $\alpha : \Omega \mapsto \mathbb{R}^{d,d}$

The attribute “Dirichlet” refers to a setting, in which the function u is prescribed on the entire boundary.

Some more explanations and terminology:

- $\Omega \subset \mathbb{R}^d, d = 2, 3 \hat{=} \text{(spatial) domain, bounded, piecewise smooth boundary}$
- $g \in C^0(\partial\Omega) \hat{=} \text{boundary values (Dirichlet data)}$
- $f \in C_{\text{pw}}^0(\Omega) \hat{=} \text{loading function, source function}$
- $\alpha : \Omega \mapsto \mathbb{R}^{d,d} \hat{=} \text{material tensor, stiffness function, diffusion coefficient}$
(uniformly positive definite, bounded \rightarrow Def. 2.1.9)

$$\exists 0 < \alpha^- \leq \alpha^+ : \quad \alpha^- \|z\|^2 \leq (\boldsymbol{\alpha}(\mathbf{x})z) \cdot z \leq \alpha^+ \|z\|^2 \quad \forall z \in \mathbb{R}^d , \quad (2.3.4)$$

for almost all $\mathbf{x} \in \Omega$.

Rewriting (2.3.3), using **offset function** u_0 with $u_0 = g$ on $\partial\Omega$, cf. (2.1.19),

$$\begin{aligned} w \in H_0^1(\Omega) : \quad & \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} w(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} \\ &= \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) - (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u_0(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega) . \end{aligned} \quad (2.3.5)$$

→ (2.3.5) is a **linear variational problem**, see Rem. 1.4.4

We can lift the above discussion to an abstract level:

Variational formulation of a quadratic minimization problem (→ Def. 2.1.18)

$$J(u) := \frac{1}{2}\mathbf{a}(u, u) + \ell(u) + c \quad \Rightarrow \quad J(u + tv) = J(u) + t(\mathbf{a}(u, v) + \ell(v)) + \frac{1}{2}t^2\mathbf{a}(v, v) , \quad 2.3$$

for all $u, v \in V_0$.

- For a quadratic functional (\rightarrow Def. 2.1.18) on real vector space V_0

$$\lim_{t \rightarrow 0} \frac{J(u + tv) - J(u)}{t} = \mathbf{a}(u, v) + \ell(v) . \quad (2.3.6)$$

- Linear variational problem (\rightarrow Rem. 1.4.4) arising from quadratic minimization problem for functional $J(u) := \frac{1}{2}\mathbf{a}(u, u) + \ell(u) + c$:

$$w \in V_0: \quad \mathbf{a}(w, v) + \ell(v) = 0 \quad \forall v \in V_0 . \quad (2.3.7)$$

Concretely, for (2.3.5): $V_0 = H_0^1(\Omega)$ and

$$\mathbf{a}(u, v) = \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} w(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} , \quad (2.3.8)$$

$$\ell(v) = - \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) + (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u_0(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} . \quad (2.3.9)$$

2.3.2 Stability

Notion of **stability** for a (linear) variational problem (2.3.7):

Lipschitz continuity of (linear) mapping data ℓ \mapsto solution w

\longleftrightarrow Is there/what is a constant $C_{\text{stab}} > 0$ such that

$$\|w\|_X \leq C_{\text{stab}} \|\ell\|_Y \quad , \text{ where } w \text{ solves (2.3.7)}, \quad (2.3.10)$$

with **suitable/relevant** norms $\|\cdot\|_X$, $\|\cdot\|_Y$? These norms will be suggested by the modelling background. Their choice will determine existence and value of C_{stab} .

Remark 2.3.11 (Sensitivity of linear variational problems).

Recall a notion introduced in [14, Sect. 2.5.5]:

Sensitivity of a problem (for given data) gauges
impact of small perturbations of the data on the result.

Remember: “Problem” = mapping from data space to solution space, see [14, Sect. 2.5.2].

Here, we define the “problem” as the mapping

$$\left\{ \begin{array}{l} \{\text{linear forms on } V_0\} \mapsto V_0 \\ \ell \mapsto w \in V_0: \quad a(w, v) = -\ell(v) \quad \forall v \in V_0 . \end{array} \right. \quad (2.3.12)$$

Undesirable: “sensitive dependence of solution on data”, that is small (in the norm of the data space) perturbations of ℓ translate into huge (in the norm of the solution space) or even “infinite” perturbations of the solution. In this case of an “**ill-posed problem**” inevitable data errors (e.g., due to non-exact measurements) will thwart any attempt to compute an “accurate” (in the norm of the solution space) solution.

Desirable: Lipschitz continuity of problem map with small Lipschitz constant (**well-posed problem**).

Note: the problem map (2.3.12) is **linear** and its Lipschitz constant is given by the smallest value for C_{stab} in (2.3.10).



Consider the particular choice (2.3.8).

How to choose the norms $\|\cdot\|_X$ (on solution space) and $\|\cdot\|_Y$ (on data space) ?

Norm on solution space: **energy norm**: $\|\cdot\|_a$

Norm on r.h.s: Mean square norm (L^2 -norm, \rightarrow Def. 2.2.5) for f ,
 H^1 -semi-norm (\rightarrow Def. 2.2.12) for u_0

What will be the impact of a perturbation of ℓ , if we use these norms?

First use the Cauchy-Schwarz inequality (2.2.15) and the uniform positivity (\rightarrow Def. 2.1.9) of α , see
(2.3.4):

2.3

$$\begin{aligned}
|\ell(v)| &\leq \|f\|_0 \|v\|_0 + \alpha^+ \|\mathbf{grad} u_0\|_0 \|\mathbf{grad} v\|_0 \\
&\leq \left(\|f\|_0^2 + (\alpha^+)^2 \|\mathbf{grad} u_0\|_0^2 \right)^{1/2} \|v\|_{H^1(\Omega)} \quad \forall v \in H^1(\Omega) . \quad (2.3.13)
\end{aligned}$$

Next, we appeal to the lower estimate in (2.3.4) and the first Poincaré-Friedrichs inequality of Thm. 2.2.16:

$$\|v\|_{H^1(\Omega)} \leq \sqrt{1 + \text{diam}^2(\Omega)} |v|_{H^1(\Omega)} \leq \sqrt{\frac{1 + \text{diam}^2(\Omega)}{\alpha^-}} \|v\|_{\mathbf{a}} . \quad (2.3.14)$$

Combine (2.3.13) and (2.3.14),

$$|\ell(v)| \leq \underbrace{\left(\|f\|_0^2 + (\alpha^+)^2 |u_0|_{H^1(\Omega)}^2 \right)^{1/2}}_{=: K(f, u_0)} \sqrt{\frac{1 + \text{diam}^2(\Omega)}{\alpha^-}} \|v\|_{\mathbf{a}} .$$

This enters the estimate for the perturbation of the solution:

$$\begin{aligned}
\mathbf{a}(w, v) &= -\ell(v) \quad \forall v \in V_0 , \\
\mathbf{a}(w + \delta w, v) &= -(\ell + \delta \ell)(v) \quad \forall v \in V_0 .
\end{aligned}$$

$\xrightarrow{\mathbf{a} \text{ bilinear}}$

$$\mathbf{a}(\delta w, v) = -\delta \ell(v) \quad \forall v \in V_0 ,$$

$\xrightarrow{(2.3.10)}$

$$\|\delta w\|_{\mathbf{a}} = \sqrt{\mathbf{a}(\delta w, \delta w)} = \sqrt{|\delta \ell(\delta w)|} \leq (K(\delta f, \delta u_0) \|\delta w\|_{\mathbf{a}})^{1/2} ,$$

\Rightarrow

$$\|\delta w\|_{\mathbf{a}} \leq K(\delta f, \delta u_0) .$$

As in Rem. 1.6.7 for associated quadratic energy functional J :

$$|J(w + \delta w) - J(w)| = \frac{1}{2}|\mathbf{a}(2w + \delta w, \delta w)| \leq \frac{1}{2} \|2w + \delta w\|_{\mathbf{a}} \|\delta w\|_{\mathbf{a}} . \quad (2.3.15)$$



Perturbation estimates in energy norm directly translate into perturbation estimates for the equilibrium energy!

Remark 2.3.16 (Needle loading).

Now we inspect a striking manifestation of instability for a 2nd-order elliptic variational problem caused by a right hand side functional that fails to satisfy (2.2.1).

Consider the taut membrane model, see Sect. 2.1.1 for details, (2.1.13) for the related minimization problem, and (2.3.2) for the associated variational equation.

Let us assume that a needle is poked at the membrane: loading by a force f “concentrated in a point \mathbf{y} ”, often denoted by $f = \delta_{\mathbf{y}}$, $\mathbf{y} \in \Omega$, where δ is the so-called **Dirac delta function** (delta distribution).

In the variational formulation this can be taken into account as follows ($u|_{\partial\Omega} = 0$, $\sigma \equiv 1$ is assumed):

$$u \in H_0^1(\Omega): \underbrace{\int_{\Omega} \mathbf{grad} u(\mathbf{x}) \cdot \mathbf{grad} v(\mathbf{x}) \, d\mathbf{x}}_{=:a(u,v)} = \underbrace{v(\mathbf{y})}_{=:l(v)} \quad \forall v \in H_0^1(\Omega) . \quad (2.3.17)$$

Recall the discussion of Sect. 2.2: is the linear functional ℓ on the right hand side continuous w.r.t. the $H_0^1(\Omega)$ -norm (= energy norm, see Def. 2.1.24) in the sense of (2.2.1)?

Consider the function $v(\mathbf{x}) = \log |\log \|\mathbf{x}\||$, $\mathbf{x} \neq 0$, on $\Omega = \{\mathbf{x} \in \mathbb{R}^2: \|\mathbf{x}\| < \frac{1}{2}\}$.

First, we express this function in **polar coordinates**

$$(r, \varphi)$$

$$x_1 = r \cos \varphi \quad , \quad x_2 = r \sin \varphi \quad \blacktriangleright \quad v(r, \varphi) = \log |\log r| . \quad (2.3.18)$$

Then we recall the expression for the gradient in polar coordinates

$$\mathbf{grad} v(r, \varphi) = \frac{\partial v}{\partial r}(r, \varphi) \mathbf{e}_r + \frac{1}{r} \frac{\partial v}{\partial \varphi}(r, \varphi) \mathbf{e}_\varphi , \quad (2.3.19)$$

where \mathbf{e}_r and \mathbf{e}_φ are orthogonal unit vectors in the polar coordinate directions.

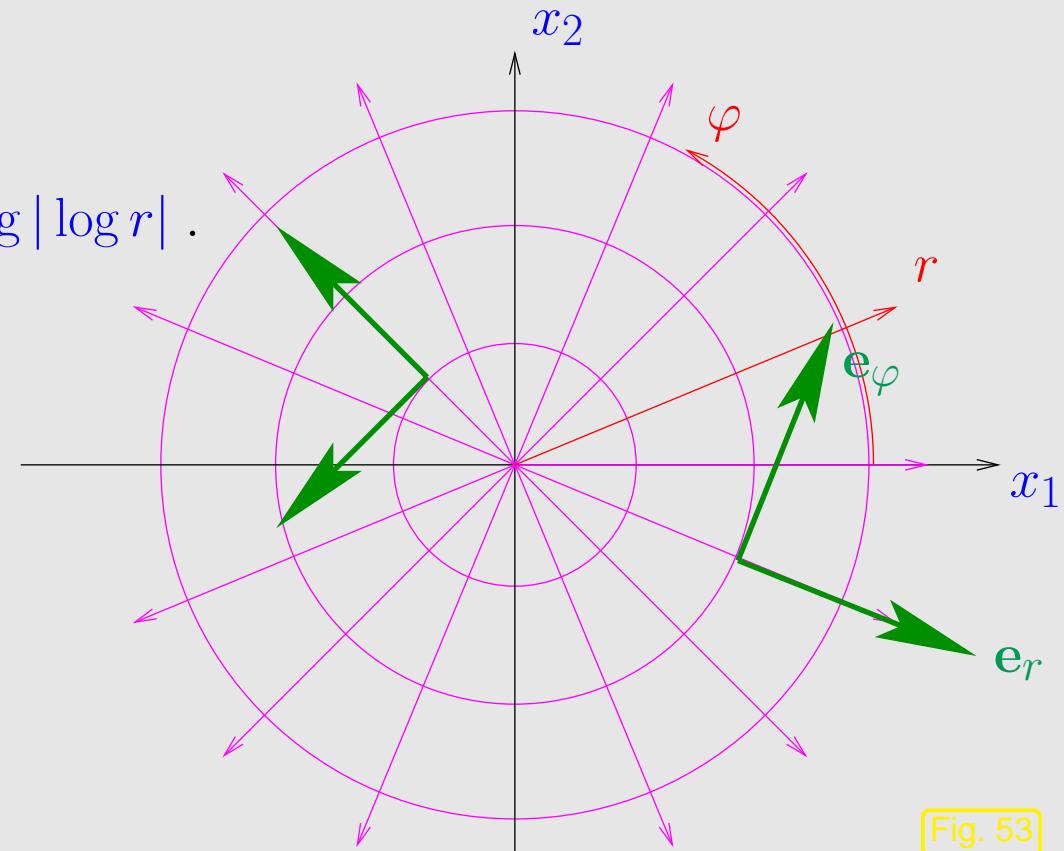


Fig. 53

Also recall integration in polar coordinates, see [19, Bsp. 8.5.3]:

$$\int_{\Omega} v(\mathbf{x}) d\mathbf{x} = \int_0^{1/2} \int_0^{2\pi} v(r, \varphi) r d\varphi dr .$$

Using these formulas we try to compute $|v|_{H^1(\Omega)}$,

$$\begin{aligned}\int_{\Omega} \|\mathbf{grad} v(\mathbf{x})\|^2 d\mathbf{x} &= \int_0^{1/2} \int_0^{2\pi} \left\| -\frac{1}{\log r} \mathbf{e}_r \right\|^2 r d\varphi dr = 2\pi \int_0^{1/2} \frac{1}{\log^2 r} \cdot \frac{1}{r} dr \\ &= [-1/\log r]_0^{1/[2]} = \frac{1}{\log 2} < \infty ,\end{aligned}$$

because the improper integral exists. This means that v has “finite elastic energy”, that is $v \in H^1(\Omega)$, see Def. 2.2.12.

On the other hand, $v(0) = \infty$!

 $H^1(\Omega)$ contains unbounded functions !

Corollary 2.3.20 (Point evaluation on $H^1(\Omega)$).

The point evaluation $v \mapsto v(\mathbf{y})$, $\mathbf{y} \in \Omega$ is not a continuous linear form on $H^1(\Omega)$.

This is the mathematics behind the observation that a needle can easily prick a taut membrane: a point load leads to configurations with “infinite elastic energy”.



Another implication of Cor. 2.3.20:

The quadratic functional $J(u) := \int_{\Omega} \|\mathbf{grad} u\|^2 \, d\mathbf{x} - u(\mathbf{y}), \mathbf{y} \in \Omega$
is *not* bounded from below on $H_0^1(\Omega)$!

Thus, it is clear that the attempt to minimize J will run into difficulties. Yet, this is the quadratic functional underlying the variational problem (2.3.17).



2.4 Equilibrium models: Boundary value problems

Recall the derivation of an ODE from a variational problem on a 1D domain (interval) in Sect. 1.3.3:

Tool:

Integration by parts (1.3.20)

This section elucidates how to extend this approach to domains $\Omega \subset \mathbb{R}^d$, $d \geq 1$ (usually $d = 2, 3$).

Crucial issue: Integration by parts in higher dimensions ?

Remember the origin of integration by parts: fundamental theorem of calculus [19, Satz 6.3.4]: for $F \in C_{\text{pw}}^1([a, b])$, $a, b \in \mathbb{R}$,

$$\int_a^b F'(x) \, dx = F(b) - F(a) , \quad (2.4.1)$$

where ' $'$ stands for differentiation w.r.t x . This formula is combined with the product rule [19, Satz 5.2.1 (ii)]

$$F(x) = f(x) \cdot g(x) \Rightarrow F'(x) = f'(x)g(x) + f(x)g'(x) . \quad (2.4.2)$$

► $\int_a^b f'(x)g(x) + f(x)g'(x) \, dx = f(b)g(b) - f(a)g(a) ,$

which amounts to (1.3.20).

Lemma 2.4.3 (General product rule).

For all $\mathbf{j} \in (C^1(\Omega))^d$, $v \in C^1(\Omega)$ holds

$$\operatorname{div}(\mathbf{j}v) = v \operatorname{div} \mathbf{j} + \mathbf{j} \cdot \operatorname{grad} v . \quad (2.4.4)$$

An important *differential operator*, see [19, Def. 8.8.1]:

divergence of a C^1 -vector field $\mathbf{j} = (f_1, \dots, f_d)^T : \Omega \mapsto \mathbb{R}^d$

$$\operatorname{div} \mathbf{j}(\mathbf{x}) := \frac{\partial f_1}{\partial x_1}(\mathbf{x}) + \cdots + \frac{\partial f_d}{\partial x_d}(\mathbf{x}) , \quad \mathbf{x} \in \Omega .$$

A truly fundamental result from differential geometry provides a multidimensional analogue of the fundamental theorem of calculus:

Theorem 2.4.5 (Gauss' theorem). → [19, Sect. 8.8]

With $\mathbf{n} : \partial\Omega \mapsto \mathbb{R}^d$ denoting the **exterior unit normal vectorfield** on $\partial\Omega$ and dS indicating integration over a surface, we have

$$\int_{\Omega} \operatorname{div} \mathbf{j}(\mathbf{x}) d\mathbf{x} = \int_{\partial\Omega} \mathbf{j}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) dS(\mathbf{x}) \quad \forall \mathbf{j} \in (C_{\text{pw}}^1(\Omega))^d. \quad (2.4.6)$$

Note: In (2.4.6) integration again allows to relax smoothness requirements, cf. Sect. 1.3.2.

Theorem 2.4.7 (Green's first formula).

For all vector fields $\mathbf{j} \in (C_{\text{pw}}^1(\Omega))^d$ and functions $v \in C_{\text{pw}}^1(\Omega)$ holds

$$\int_{\Omega} \mathbf{j} \cdot \operatorname{grad} v d\mathbf{x} = - \int_{\Omega} \operatorname{div} \mathbf{j} v d\mathbf{x} + \int_{\partial\Omega} \mathbf{j} \cdot \mathbf{n} v dS. \quad (2.4.8)$$

Note that the dependence on the integration variable \mathbf{x} is suppressed in the formula (2.4.8) to achieve a more compact notation. The first Green formula could also have been written as

$$\int_{\Omega} \mathbf{j}(\mathbf{x}) \cdot (\operatorname{grad} v)(\mathbf{x}) d\mathbf{x} = - \int_{\Omega} (\operatorname{div} \mathbf{j})(\mathbf{x}) v(\mathbf{x}) d\mathbf{x} + \int_{\partial\Omega} \mathbf{j}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) v(\mathbf{x}) dS(\mathbf{x}). \quad (2.4.8)$$

Proof. (of Thm. 2.4.7) Straightforward from Lemma 2.4.3 and Thm. 2.4.5. □

Now we apply Green's first formula to the variational problem (2.3.3), which covers the membrane model and electrostatics:

The role of \mathbf{j} in (2.4.8) is played by the vector field $\alpha \operatorname{grad} u : \Omega \mapsto \mathbb{R}^d$.

$$\int_{\Omega} \underbrace{\alpha(\mathbf{x}) \operatorname{grad} u(\mathbf{x})}_{=: \mathbf{j}(\mathbf{x})} \cdot \operatorname{grad} v(\mathbf{x}) d\mathbf{x}$$

$$= - \int_{\Omega} \operatorname{div}(\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) v(\mathbf{x}) \, d\mathbf{x} + \int_{\partial\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x}) v(\mathbf{x}) \, dS(\mathbf{x}) .$$



$$\begin{aligned}
 (2.3.3) \quad \blacktriangleright \quad & - \int_{\Omega} \operatorname{div}(\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) v(\mathbf{x}) \, d\mathbf{x} \\
 & + \int_{\partial\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x}) v(\mathbf{x}) \, dS(\mathbf{x}) \\
 & \underbrace{\qquad\qquad\qquad}_{=0, \text{ since } v|_{\partial\Omega}=0} \\
 & = \int_{\Omega} f(\mathbf{x}) v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in C_{0,\text{pw}}^1(\Omega) ,
 \end{aligned} \tag{2.4.9}$$

where we have to assume that

$u, \boldsymbol{\alpha}$ are sufficiently smooth:

$$\boldsymbol{\alpha} \operatorname{grad} u \in C_{\text{pw}}^1(\Omega)$$



$$\int_{\Omega} (\operatorname{div}(\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) + f(\mathbf{x})) v(\mathbf{x}) \, d\mathbf{x} = 0 \quad \forall v \in C_{0,\text{pw}}^1(\Omega) .$$

Now we can invoke the multidimensional analogue of the fundamental lemma of the calculus of variations, see Lemm 1.3.21

Lemma 2.4.10 (Fundamental lemma of calculus of variations in higher dimensions).

If $f \in L^2(\Omega)$ satisfies

$$\int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} = 0 \quad \forall v \in C_0^\infty(\Omega) ,$$

then $f \equiv 0$ can be concluded.

(2.3.3)

$\alpha \operatorname{grad} u \in C_{\text{pw}}^1(\Omega)$



Partial differential equations (PDE)

$$-\operatorname{div}(\alpha(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \Omega .$$

(2.4.11)

Again, for the sake of brevity, dependence $\mathbf{grad} u = \mathbf{grad} u(\mathbf{x})$, $f = f(\mathbf{x})$ is not made explicit in the PDE in (2.4.11).

Remark 2.4.12 (Laplace operator).

If α agrees with a positive *constant*, by rescaling of (2.5.6) we can achieve

$$-\Delta u = f \quad \text{in } \Omega . \tag{2.4.13}$$

$$\Delta = \operatorname{div} \circ \mathbf{grad} = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2} = \text{Laplace operator}$$

► (2.4.13) is called **Poisson equation**,

$\Delta u = 0$ in Ω is called **Laplace equation**



Finally:

$$\begin{array}{c} \text{PDE (2.4.11)} \quad + \quad \text{boundary conditions} \\ \downarrow \qquad \qquad \qquad \downarrow \\ -\operatorname{div}(\boldsymbol{\alpha}(\boldsymbol{x}) \operatorname{grad} u) = f \quad \text{in } \Omega \quad , \quad u = g \quad \text{on } \partial\Omega . \end{array} \quad (2.4.14)$$

(2.4.14) = second-order elliptic BVP with Dirichlet boundary conditions

Short name for BVPs of the type (2.4.14): “Dirichlet problem”

Remark 2.4.15 (Extra smoothness requirement for PDE formulation).

Same situation as in Sect. 1.3.3, cf. Assumption (1.3.19):

Transition from variational equation to PDE requires
extra assumptions on smoothness of solution and coefficients.

Remark 2.4.16 (Membrane with free boundary values).

(Graph description of membrane shape by u :
 $\Omega \mapsto \mathbb{R}$, see Sect. 2.1.1)

Now: membrane clamped only on a part
 $\Gamma_0 \subset \partial\Omega$ of its edge.

— : prescribed boundary values here
 — : “free boundary”

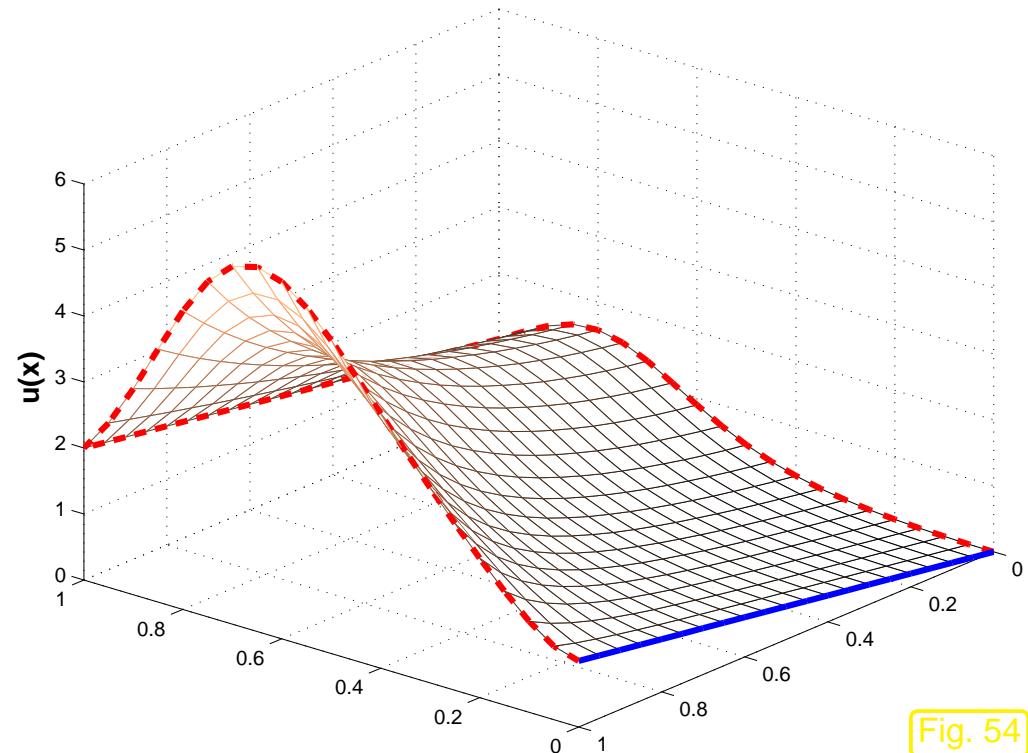


Fig. 54

► Configuration space $V := \{u \in H^1(\Omega) : u|_{\Gamma_0} = g\} \rightarrow \text{Def. 2.2.12}$

Total potential energy as in (2.1.3):

$$J_M(u) := \int_{\Omega} \frac{1}{2} \sigma(\mathbf{x}) \|\mathbf{grad} u\|^2 - f(\mathbf{x}) u(\mathbf{x}) \, d\mathbf{x} . \quad (2.1.3)$$

► test space in variational formulation

$$V_0 := \{u \in H^1(\Omega) : u|_{\Gamma_0} = 0\}$$

Variational formulation, c.f. (2.3.2)

$$\begin{aligned} & u \in H^1(\Omega), \\ & u = g \text{ on } \Gamma_0 \end{aligned} \quad \int_{\Omega} \sigma(\boldsymbol{x}) \operatorname{grad} u(\boldsymbol{x}) \cdot \operatorname{grad} v(\boldsymbol{x}) \, d\boldsymbol{x} = \int_{\Omega} f(\boldsymbol{x}) v(\boldsymbol{x}) \quad \forall v \in V_0. \quad (2.4.17)$$

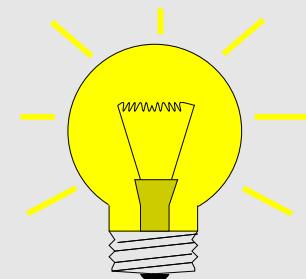
☞ Application of Green's first formula (2.4.8) to (2.4.17) leads to

$$\begin{aligned} & - \int_{\Omega} (\operatorname{div}(\sigma(\boldsymbol{x}) \operatorname{grad} u(\boldsymbol{x})) + f(\boldsymbol{x})) v(\boldsymbol{x}) \, d\boldsymbol{x} \\ & \quad + \int_{\partial\Omega \setminus \Gamma_0} ((\sigma(\boldsymbol{x}) \operatorname{grad} u(\boldsymbol{x})) \cdot \boldsymbol{n}(\boldsymbol{x})) v(\boldsymbol{x}) \, dS(\boldsymbol{x}) \quad \forall v \in V_0. \quad (2.4.18) \end{aligned}$$

Note that, unlike above, the boundary integral term cannot be dropped entirely, because $v \neq 0$ on $\partial\Omega \setminus \Gamma_0$.

Assumption (\rightarrow Rem. 2.4.15): extra smoothness $u \in C_{\text{pw}}^2(\Omega)$, $\sigma \in C_{\text{pw}}^1(\Omega)$

How to deal with the boundary term ?



Idea: ① First restrict test function v to $C_0^\infty(\Omega)$



Boundary term vanishes !

Then, apply Lemma 2.4.10



$$\operatorname{div}(\sigma(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) + f(\mathbf{x}) = 0 \quad \text{in } \Omega . \quad (2.4.19)$$

② Then test with generic $v \in V_0$, while *making use of* (2.4.19):

$$\int_{\partial\Omega \setminus \Gamma_0} ((\sigma(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x})) v(\mathbf{x}) \, dS(\mathbf{x}) = 0 \quad \forall v \in V_0 .$$

Lemma 2.4.10 on $\partial\Omega \setminus \Gamma_0$
 \implies

$$(\sigma(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x}) = 0 \quad \text{on } \partial\Omega \setminus \Gamma_0 . \quad (2.4.20)$$

When removing pinning conditions on $\partial\Omega \setminus \Gamma_0$ the equilibrium conditions imply the (homogeneous) **Neumann boundary conditions** $(\sigma(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x}) = 0$ on $\partial\Omega \setminus \Gamma_0$.



Boundary value problem for membrane clamped at $\Gamma_0 \subset \partial\Omega$

$$-\operatorname{div}(\sigma(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \Omega, \quad \begin{aligned} u &= g && \text{on } \Gamma_0, \\ (\sigma(\mathbf{x}) \operatorname{grad} u) \cdot \mathbf{n} &= 0 && \text{on } \partial\Omega \setminus \Gamma_0. \end{aligned} \quad (2.4.21)$$

(2.4.21) = Second-order elliptic BVP with Neumann boundary conditions on $\partial\Omega \setminus \Gamma_0$

Short name for BVPs of the type (2.4.21): “Mixed Neumann–Dirichlet problem”



2.5 Diffusion models (Stationary heat conduction)

Now we look at a class of physical phenomena, for which models are based on two building blocks

2.5

p. 227

1. a conservation principle (of mass, energy, etc.),
2. a potential driven flux of the conserved quantity.

Mathematical modelling for these phenomena naturally involves partial differential equations in the first steps, which are supplemented with boundary conditions. Hence, second-order elliptic boundary value problems arise first, while variational formulations are deduced from them, thus reversing the order of steps followed for equilibrium models in Sects. 2.1–2.4.

In order to keep the presentation concrete, the discussion will target heat conduction, about which everybody should have a sound “intuitive grasp”.

notation: $\Omega \subset \mathbb{R}^3$: bounded open region occupied by solid object
($\hat{=}$ $\Omega \rightarrow$ computational domain)

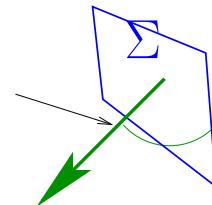
Fundamental concept:

heat flux, modelled by vector field $\mathbf{j} : \Omega \mapsto \mathbb{R}^3$

Heat flux = power flux: $[\mathbf{j}] = \frac{\text{W}}{\text{m}^2}$

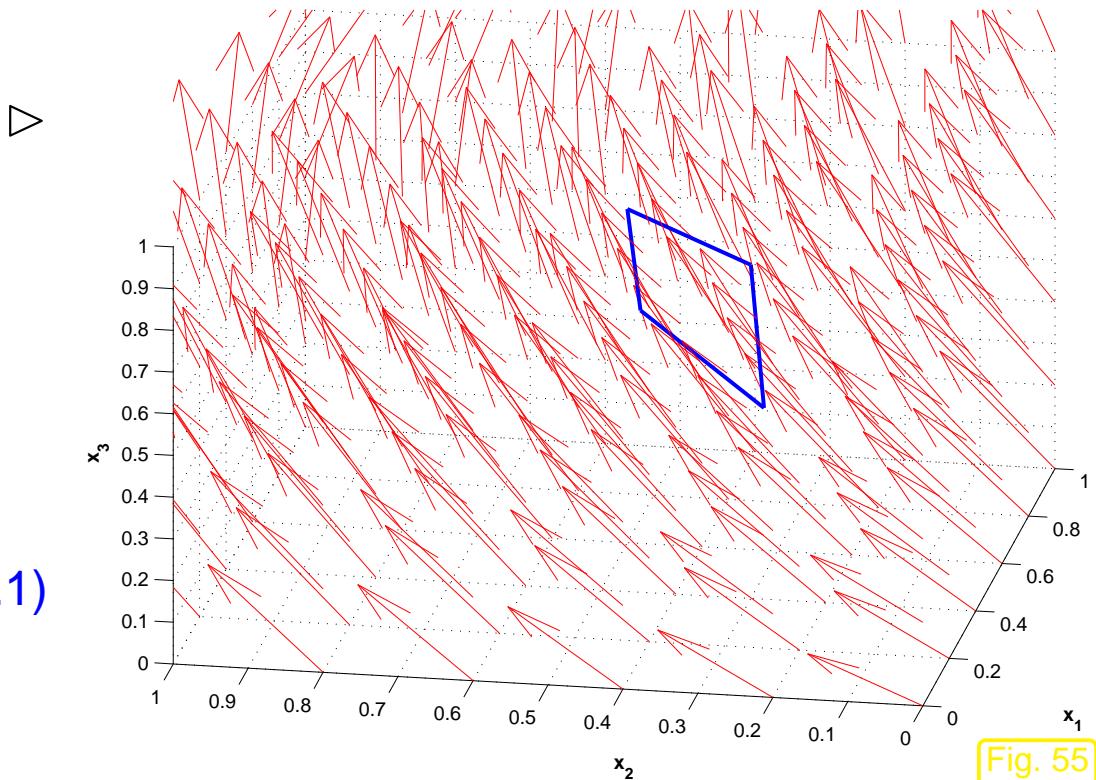
Vector field $\mathbf{j} : \Omega :=]0, 1[^2 \rightarrow \mathbb{R}^3$

normal vector \mathbf{n}



Total heat flux through oriented surface $\Sigma \subset \mathbb{R}^3$

Power $P_\Sigma = \int_\Sigma \mathbf{j} \cdot \mathbf{n} dS . \quad (2.5.1)$



P_Σ : directed total power flowing through the oriented surface Σ per unit time. Note that the sign of P_Σ will change when flipping the normal of Σ !

Conservation of energy

$$\int_{\partial V} \mathbf{j} \cdot \mathbf{n} dS = \int_V f dx \quad \text{for all "control volumes" } V . \quad (2.5.2)$$

power flux through surface of V heat production inside V

f = heat source/sink ($[f] = \frac{\text{W}}{\text{m}^3}$), $f = f(\mathbf{x})$ and f can be discontinuous ($f \in C_{\text{pw}}^0(\Omega)$)

Intuition:

- heat flows from hot zones to cold zones
- the larger the temperature difference, the stronger the heat flow

Experimental evidence supports this intuition and, for many materials, yields the following quantitative relationship:

Fourier's law

$$\mathbf{j}(\mathbf{x}) = -\kappa(\mathbf{x}) \operatorname{grad} u(\mathbf{x}) , \quad \mathbf{x} \in \Omega . \quad (2.5.3)$$

j	= heat flux	$([j] = 1 \frac{W}{m^2})$
u	= temperature	$([u] = 1K)$
κ	= heat conductivity	$([\kappa] = 1 \frac{W}{Km})$



(2.5.3) \Rightarrow Heat flow from hot to cold regions **linearly proportional** to gradient of temperature

Some facts about the heat conductivity:

- κ :
- $\kappa = \kappa(x)$ for **non-homogeneous** materials (spatially varying heat conductivity)
 - κ can even be discontinuous for composite materials
 - κ may be $\mathbb{R}^{3,3}$ -valued (heat conductivity tensor)

The most general form of the heat conductivity (tensor) enjoys the very same properties as the dielectric tensor introduced in Sect. 2.1.2:

From thermodynamic principles, cf. (2.1.8):

$$\exists \kappa^-, \kappa^+ > 0: \quad 0 < \kappa^- \leq \kappa(\mathbf{x}) \leq \kappa^+ < \infty \quad \text{for almost all } \mathbf{x} \in \Omega . \quad (2.5.4)$$

Terminology: (2.5.4) \leftrightarrow κ is bounded and uniformly positive, see Def. 2.1.9.

From 2.5.2 by Gauss' theorem Thm. 2.4.5

$$\int_V \operatorname{div} \mathbf{j}(\mathbf{x}) d\mathbf{x} = \int_V f(\mathbf{x}) d\mathbf{x} \quad \text{for all "control volumes" } V \subset \Omega .$$

Now appeal to another version of the fundamental lemma of the calculus of variations, see Lemma 2.4.10, this time sporting piecewise constant test functions.

► local form of energy conservation:

$$\operatorname{div} \mathbf{j} = f \quad \text{in } \Omega . \quad (2.5.5)$$

Combine equations (2.5.5) & (2.5.3)

$$\mathbf{j} = -\kappa(\mathbf{x}) \operatorname{grad} u \quad (2.5.3)$$

+

$$\operatorname{div} \mathbf{j} = f \quad (2.5.5)$$



$$-\operatorname{div}(\kappa(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \Omega . \quad (2.5.6)$$

► *Linear scalar second order elliptic PDE* (for unknown temperature u)

2.6 Boundary conditions

In the examples from Sects. 2.1.1, 2.1.2 we fixed the value of the unknown function $u : \Omega \mapsto \mathbb{R}$ on the boundary $\partial\Omega$: **Dirichlet boundary conditions** in (2.4.14).

Exception: free edge of taut membrane, see Rem. 2.4.16: **Neumann boundary conditions** in (2.4.21).

In this section we resume the discussion of boundary conditions and examine them for stationary heat conduction, see previous section. This has the advantage that for this everyday physical phenomenon boundary conditions have a very clear intuitive meaning.

Boundary conditions on surface/boundary $\partial\Omega$ of Ω :

(i) Temperature u is fixed: with $g : \partial\Omega \mapsto \mathbb{R}$ prescribed

$$u = g \quad \text{on } \partial\Omega . \quad (2.6.1)$$



Dirichlet boundary conditions

(ii) Heat flux \mathbf{j} through $\partial\Omega$ is fixed: with $h : \partial\Omega \mapsto \mathbb{R}$ prescribed ($\mathbf{n} : \partial\Omega \mapsto \mathbb{R}^3$ exterior unit normal vectorfield) on $\partial\Omega$

$$\mathbf{j} \cdot \mathbf{n} = -h \quad \text{on } \partial\Omega . \quad (2.6.2)$$



Neumann boundary conditions

(iii) Heat flux through $\partial\Omega$ depends on (local) temperature: with increasing function $\Psi : \mathbb{R} \mapsto \mathbb{R}$

$$\mathbf{j} \cdot \mathbf{n} = \Psi(u) \quad \text{on } \partial\Omega \quad (2.6.3)$$



radiation boundary conditions

Example 2.6.4 (Convective cooling (simple model)).

Heat is carried away from the surface of the body by a fluid at bulk temperature u_0 . A crude model assumes that the heat flux depends *linearly* on the temperature difference between the surface of Ω and the bulk temperature of the fluid.

$$\mathbf{j} \cdot \mathbf{n} = q(u - u_0) \quad \text{on } \partial\Omega , \quad \text{where } 0 < q^- \leq q(\mathbf{x}) \leq q^+ < \infty \quad \text{for almost all } \mathbf{x} \in \partial\Omega .$$



Example 2.6.5 (Radiative cooling (simple model)).

A hot body emits electromagnetic radiation (blackbody emission), which drains thermal energy. The radiative energy loss is roughly proportional to the 4th power of the temperature difference between the surface temperature of the body and the ambient temperature.

$$\mathbf{j} \cdot \mathbf{n} = \alpha |u - u_0| (u - u_0)^3 \quad \text{on } \partial\Omega , \quad \text{with } \alpha > 0$$

→ Non-linear boundary condition

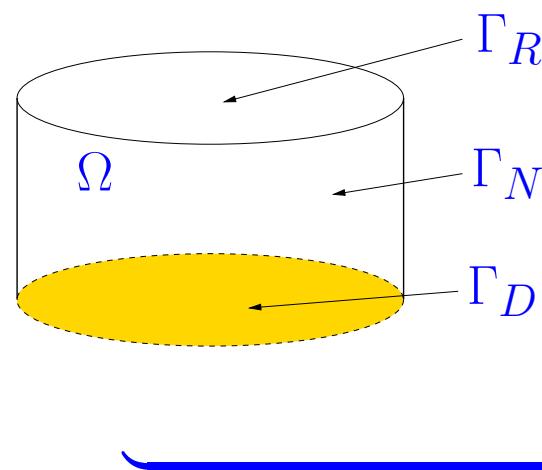


Terminology: If $g = 0$ or $h = 0$ → **homogeneous** Dirichlet or Neumann boundary conditions

Remark 2.6.6 (Mixed boundary conditions).

Different boundary conditions can be prescribed on different parts of $\partial\Omega$

(→ **mixed boundary conditions**, cf. Rem. 2.4.16)



Example 2.6.7 (“Wrapped rock on a stove”).

- Non-homogeneous Dirichlet boundary conditions on $\Gamma_D \subset \partial\Omega$
- Homogeneous Neumann boundary conditions on $\Gamma_N \subset \partial\Omega$
- Convective cooling boundary conditions on $\Gamma_R \subset \partial\Omega$

Partition: $\partial\Omega = \overline{\Gamma}_D \cup \overline{\Gamma}_N \cup \overline{\Gamma}_R$, $\Gamma_D, \Gamma_N, \Gamma_R$ mutually disjoint



– $\operatorname{div}(\kappa(\mathbf{x}) \operatorname{grad} u) = f$ + boundary conditions \Rightarrow **elliptic boundary value problem (BVP)**

For second order elliptic boundary value problems **exactly one** boundary condition is needed on any part of $\partial\Omega$.

Remark 2.6.8 (Linear BVP).

Observe that the solution mapping $\begin{pmatrix} f \\ g \end{pmatrix} \mapsto u$ for (2.5.6), (2.6.1) is linear.

This means that if u_i solves the Dirichlet problem with source function f_i and Dirichlet data g_i , $i = 1, 2$, then $u_1 + u_2$ solves (2.5.6) & (2.6.1) for source $f_1 + f_2$ and boundary values $g_1 + g_2$. \triangle

2.7 Characteristics of elliptic boundary value problems

Qualitative insights gained from heat conduction model:

- continuity: the temperature u must be continuous (jump in $u \rightarrow j = \infty$).

- normal component of \mathbf{j} across surfaces inside Ω must be continuous
(jump in $\mathbf{j} \cdot \mathbf{n} \rightarrow$ heat source f of infinite intensity).
- interior smoothness of u : u smooth where f and D smooth.
- non-locality: local alterations in f, g, h affect u everywhere in Ω .
- quasi-locality: If local changes in f, g, h confined to $\Omega' \subset \Omega$, their effects decay away from Ω' .
- maximum principle: (in the absence of heat sources extremal temperatures are on the boundary)

if $f \equiv 0$, then $\inf_{\mathbf{y} \in \partial\Omega} u(\mathbf{y}) \leq u(\mathbf{x}) \leq \sup_{\mathbf{y} \in \partial\Omega} u(\mathbf{y}) \quad \text{for all } \mathbf{x} \in \Omega$

Typical features of solutions of elliptic boundary value problems

Example 2.7.1 (Scalar elliptic boundary value problem in one space dimension).

Poisson equation \rightarrow (2.4.13) in 1D:

$$-u'' = f$$

➤ f discontinuous, piecewise $C^0 \Rightarrow u \in C^1$, piecewise C^2



Example 2.7.2 (Smoothness of solution of scalar elliptic boundary value problem).

$$\begin{aligned} -\Delta u &= f(\mathbf{x}) \quad \text{in } \Omega := [0, 1]^2, \quad u = 0 \quad \text{on } \partial\Omega, \\ f(\mathbf{x}) &:= \text{sign}(\sin(2\pi k_1 x_1) \sin(2\pi k_2 x_2)), \quad \mathbf{x} \in \Omega, \quad k_1, k_2 \in \mathbb{N}. \end{aligned} \quad (2.7.3)$$

Approximate solution computed by means of linear Lagrangian finite elements + lumping
 (→ Sect. ??, details in Sect. ??, ??)

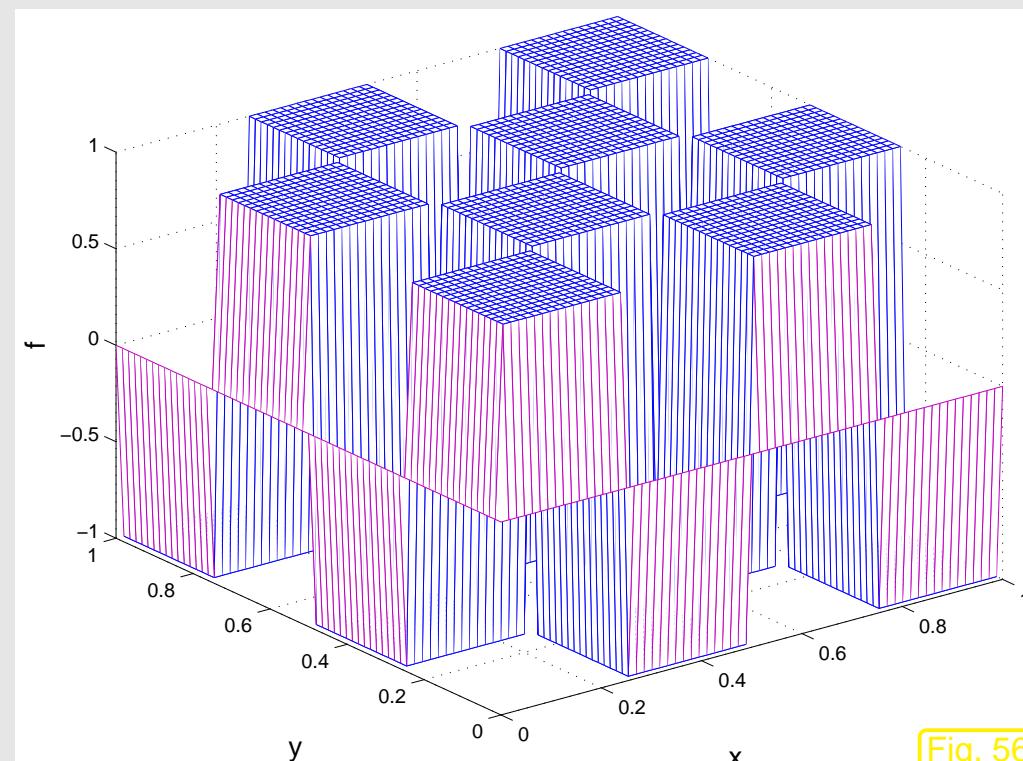


Fig. 56

Source term $f(\mathbf{x})$, $k_1 = k_2 = 2$

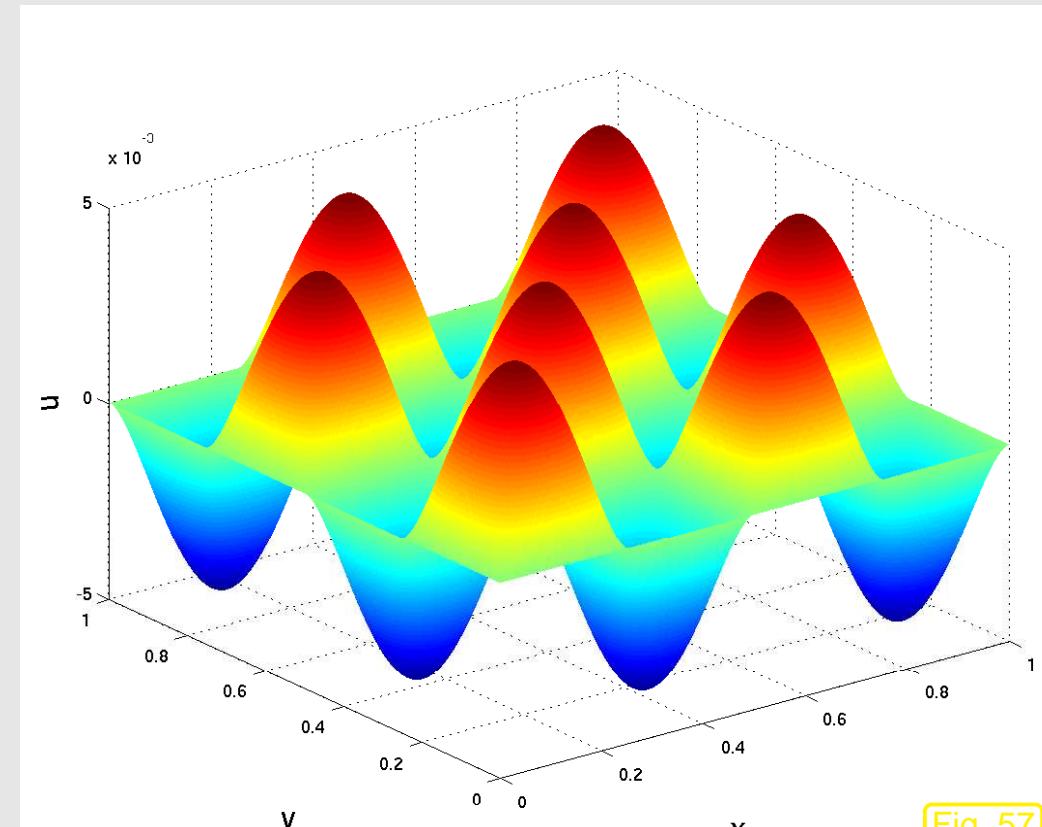


Fig. 57

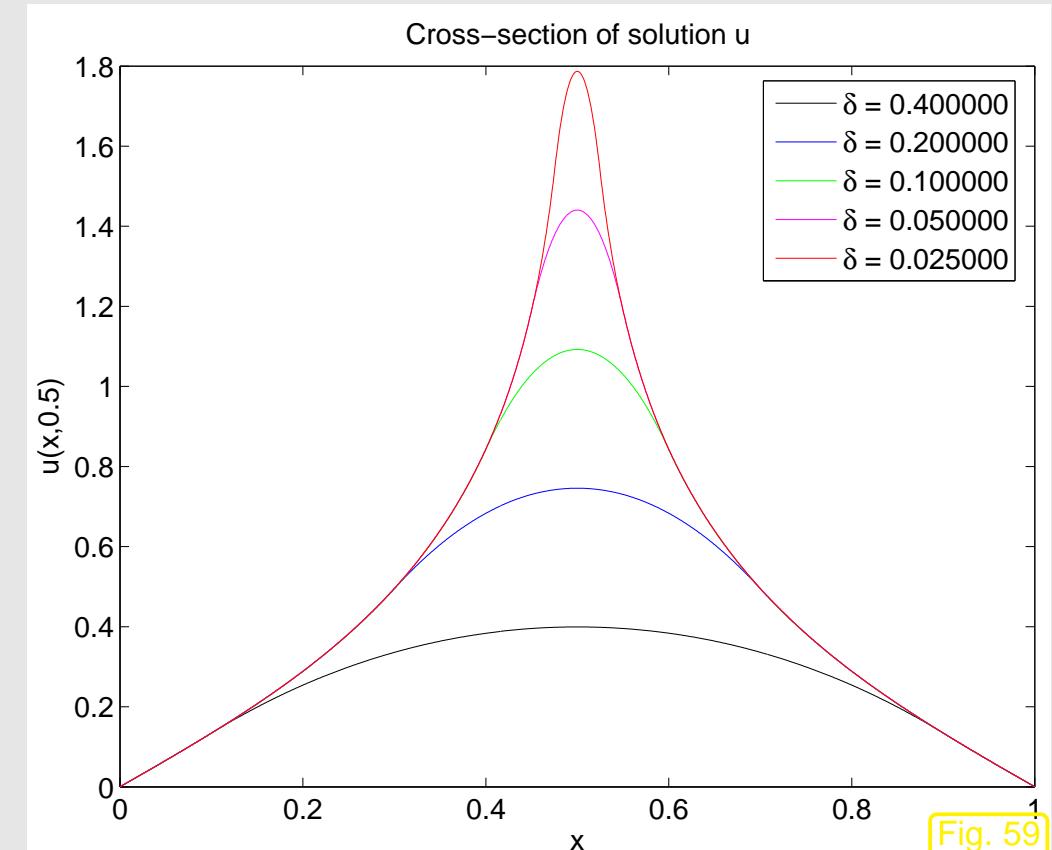
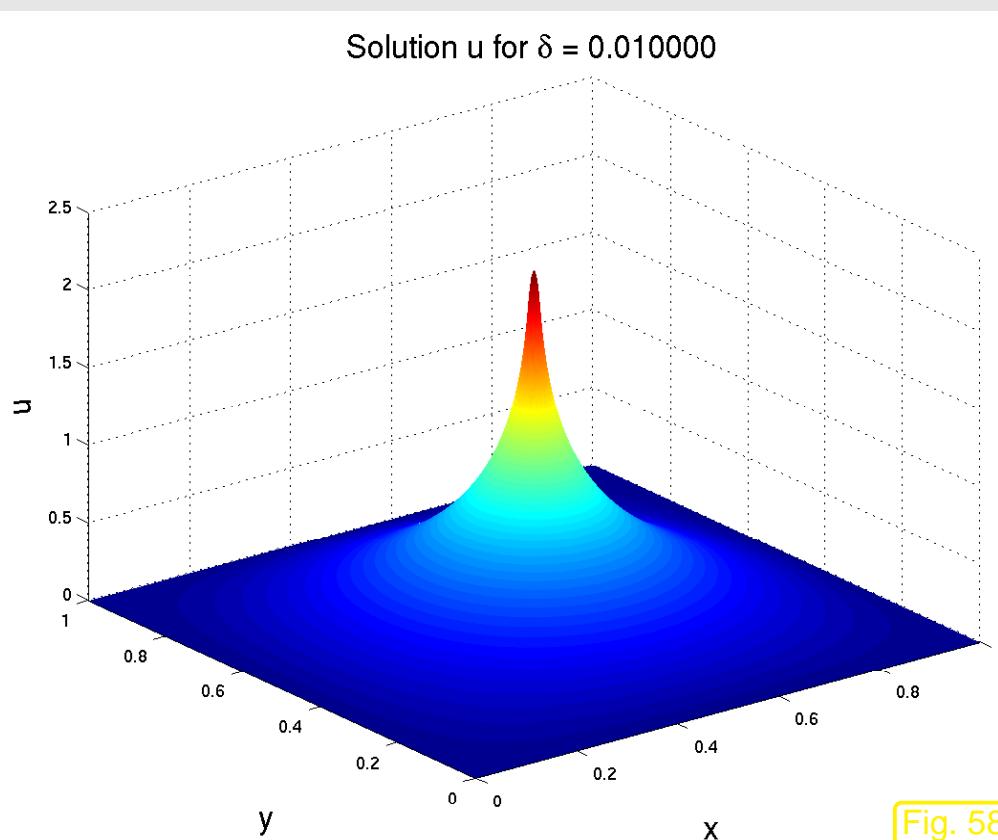
Solution of (2.7.3)

➤ “Smooth” u despite “rough” f !

Example 2.7.4 (Quasi-locality of solution of scalar elliptic boundary value problem).

$$-\Delta u = f_\delta(\mathbf{x}) \quad \text{in } \Omega := [0, 1]^2, \quad u = 0 \quad \text{on } \partial\Omega, \quad (2.7.5)$$

$$f_\delta(\mathbf{x}) = \begin{cases} \delta^{-2} & , \text{if } \left\| \mathbf{x} - \left(\frac{1}{2}, \frac{1}{2} \right) \right\|_2 \leq \delta, \\ 0 & \text{elsewhere.} \end{cases}, \quad \delta > 0. \quad (2.7.6)$$





2.8 Second-order elliptic variational problems

In Ch. 1 and Sects. 2.1–2.4 we pursued the derivation:

$$\begin{array}{ccc} \text{Minimization problem} & \Rightarrow & \text{Variational problem} \\ (\text{e.g., (2.1.4), (2.1.12)}) & & (\text{e.g., (2.3.1), (2.3.2)}) \end{array} \quad \Rightarrow \quad \begin{array}{c} \text{BVP for PDE} \\ (\text{e.g., (2.4.14), (2.4.21)}) \end{array}$$

Now we are proceeding in the opposite direction:

$$\begin{array}{ccc} \text{PDE} & + & \text{boundary conditions} \\ (\text{e.g. (2.5.6)}) & + & (\text{e.g., (2.6.1), (2.6.2), (2.6.3)}) \end{array} \quad \Rightarrow \quad \text{variational problem}$$

2.8

p. 242

Formal approach:

STEP 1: *test PDE with smooth functions*

(do not test, where the solution is known, e.g., on the boundary)

STEP 2: *integrate over domain*

STEP 3: *perform integration by parts*

(e.g. by using Green's first formula, Thm. 2.4.7)

STEP 4: [optional] *incorporate boundary conditions into boundary terms*

Example 2.8.1 (Variational formulation for heat conduction with Dirichlet boundary conditions).

$$\text{BVP: } -\operatorname{div}(\kappa(\boldsymbol{x}) \operatorname{grad} u) = f \quad \text{in } \Omega , \quad u = g \quad \text{on } \partial\Omega . \quad (2.8.2)$$

STEP 1 & 2:

test with $v \in C_0^\infty(\Omega)$

►
$$-\int_{\Omega} \operatorname{div}(\kappa(\boldsymbol{x}) \operatorname{grad} u) v \, d\boldsymbol{x} = \int_{\Omega} f v \, d\boldsymbol{x} . \quad (2.8.3)$$

Note: $v|_{\partial\Omega} = 0$ for test function, because u already fixed on $\partial\Omega$.

STEP 3: use **Green's formula** from Thm. 2.4.7 on $\Omega \subset \mathbb{R}^d$ (multidimensional integration by parts):

Apply (2.4.8) to (2.8.3) with $\mathbf{j} := \kappa(\boldsymbol{x}) \operatorname{grad} u$:

►
$$\int_{\Omega} \kappa(\boldsymbol{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\boldsymbol{x} - \underbrace{\int_{\partial\Omega} \kappa(\boldsymbol{x}) \operatorname{grad} u \cdot \mathbf{n} v \, dS}_{=0, \text{because } v|_{\partial\Omega}=0} = \int_{\Omega} f v \, d\boldsymbol{x} \quad \forall v \in C_0^\infty(\Omega) .$$

This gives the variational formulation after we switch to “maximal admissible function spaces”
(Sobolev spaces, see Sect. 2.2)

Variational form of (2.8.2): seek

$$\begin{aligned} u \in H^1(\Omega) \\ u = g \text{ on } \partial\Omega : \quad \int_{\Omega} \kappa(\boldsymbol{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\boldsymbol{x} = \int_{\Omega} f v \, d\boldsymbol{x} \quad \forall v \in H_0^1(\Omega). \end{aligned} \quad (2.8.4)$$



Example 2.8.5 (Variational formulation: heat conduction with general radiation boundary conditions).

BVP: $-\operatorname{div}(\kappa(\boldsymbol{x}) \operatorname{grad} u) = f \quad \text{in } \Omega, \quad -\kappa(\boldsymbol{x}) \operatorname{grad} u \cdot \boldsymbol{n} = \Psi(u) \quad \text{on } \partial\Omega.$ (2.8.6)

STEP 1 & 2: $u|_{\partial\Omega}$ not fixed \Rightarrow test with $v \in C^\infty(\overline{\Omega})$

► $-\int_{\Omega} \operatorname{div}(\kappa(\boldsymbol{x}) \operatorname{grad} u) v \, d\boldsymbol{x} = \int_{\Omega} f v \, d\boldsymbol{x} \quad \forall v \in C^\infty(\overline{\Omega}).$

STEP 3 & 4: apply Green's first formula (2.4.8) and incorporate boundary conditions:

►
$$\int_{\Omega} \kappa(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\mathbf{x} - \int_{\partial\Omega} \underbrace{-\kappa(\mathbf{x}) \operatorname{grad} u \cdot \mathbf{n}}_{=\Psi(u) \text{ (STEP 4)}} v \, dS = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in C^{\infty}(\bar{\Omega}).$$



Variational formulation of (2.8.6): seek

$$u \in H^1(\Omega): \quad \int_{\Omega} \kappa(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\mathbf{x} + \int_{\partial\Omega} \Psi(u) v \, dS = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H^1(\Omega). \quad (2.8.7)$$



Theorem 2.8.8. If $\kappa \in C^1(\bar{\Omega})$, classical solutions $u \in C^2(\bar{\Omega})$ of the boundary value problems (2.8.2) and (2.8.6) also solve the associated variational problems.

Proof. Apply Theorem 2.4.7 as in the derivation of the weak formulations.

Example 2.8.9 (Variational formulation for Neumann problem).

2nd-order elliptic (inhomogeneous) **Neumann problem**

$$\text{BVP:} \quad \begin{aligned} -\operatorname{div}(\kappa(\boldsymbol{x}) \operatorname{grad} u) &= f && \text{in } \Omega, \\ \kappa(\boldsymbol{x}) \operatorname{grad} u \cdot \boldsymbol{n} &= h(\boldsymbol{x}) && \text{on } \partial\Omega. \end{aligned} \quad (2.8.10)$$

We confront Neumann boundary conditions (2.6.2) (prescribed heat flux) on the whole boundary.

Variational formulation derived as in Ex. 2.8.5, with $\Psi(u) = -h$.

$$u \in H^1(\Omega): \quad \int_{\Omega} \kappa(\boldsymbol{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\boldsymbol{x} - \int_{\partial\Omega} h v \, dS = \int_{\Omega} f v \, d\boldsymbol{x} \quad \forall v \in H^1(\Omega). \quad (2.8.11)$$

Observation: when we test (2.8.11) with $v \equiv 1$  $-\int_{\partial\Omega} h \, dS = \int_{\Omega} f \, d\boldsymbol{x}$ (2.8.12)

This is a **compatibility condition** for the existence of (variational) solutions of the Neumann problem!

Interpretation of (2.8.12) against the backdrop of the stationary heat conduction model:

conservation of energy \rightarrow (2.5.2): Heat generated inside Ω ($\leftrightarrow f$) must be offset by heat flux through $\partial\Omega$ ($\rightarrow h$).



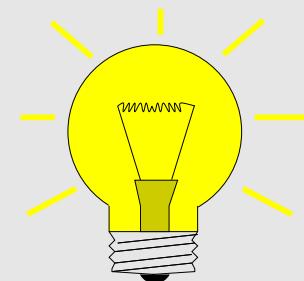
Remark 2.8.13 (Uniqueness of solutions of Neumann problem).

Observation: if compatibility condition (2.8.12) holds true, then

$$v \in H^1(\Omega) \text{ solves (2.8.11)} \iff v + \gamma \text{ solves (2.8.11)} \quad \forall \gamma \in \mathbb{R} ,$$

we say, “the solution is unique only up to constants”.

Complementary observation: $\mathbf{a}(u, v) := \int_{\Omega} \kappa(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\mathbf{x}$ is *not* s.p.d (\rightarrow Def. 2.1.22) on $H^1(\Omega)$.



Idea: Restore uniqueness of solutions by

enforcing average temperature to be zero

$$\int_{\Omega} u(\mathbf{x}) \, d\mathbf{x} = 0$$

This amounts to posing the variational problem (2.8.11) over the **constrained** function space

$$H_*^1(\Omega) := \{v \in H^1(\Omega): \int_{\Omega} v(\mathbf{x}) \, d\mathbf{x} = 0\} . \quad (2.8.14)$$

The norm on $H_*^1(\Omega)$ is the same as on $H_0^1(\Omega)$, see Def. 2.2.12. Obviously (why ?), the norm property (N1) is satisfied. These arguments also show that \mathbf{a} is s.p.d (\rightarrow Def. 2.1.22) on $H_*^1(\Omega)$.

► Variational formulation of Neumann problem:

$$u \in H_*^1(\Omega): \int_{\Omega} \kappa(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} + \int_{\partial\Omega} h v \, dS \quad \forall v \in H_*^1(\Omega) . \quad (2.8.15)$$

2.9 Essential and natural boundary conditions

Synopsis:

- 2nd-order elliptic Dirichlet problem:

$$-\operatorname{div}(\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \Omega \quad , \quad u = g \quad \text{on } \partial\Omega . \quad (2.4.14)$$

with variational formulation

$$\begin{aligned} & u \in H^1(\Omega) \quad , \\ & u = g \text{ on } \partial\Omega \quad , \quad \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega) . \end{aligned} \quad (2.3.3)$$

- 2nd-order elliptic Neumann problem:

$$-\operatorname{div}(\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \Omega \quad , \quad (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u) \cdot \mathbf{n} = -h \quad \text{on } \partial\Omega . \quad (2.9.1)$$

with variational formulation

$$u \in H_*^1(\Omega): \int_{\Omega} \alpha(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} + \int_{\partial\Omega} h v \, dS \quad \forall v \in H_*^1(\Omega) . \quad (2.8.15)$$

→ 2nd-order elliptic mixed Neumann-Dirichlet problem, see Rem. 2.4.16:

$$-\operatorname{div}(\alpha(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \Omega , \quad \begin{aligned} u &= g && \text{on } \Gamma_0 \subset \partial\Omega , \\ (\alpha(\mathbf{x}) \operatorname{grad} u) \cdot \mathbf{n} &= -h && \text{on } \partial\Omega \setminus \Gamma_0 . \end{aligned} \quad (2.9.2)$$

with variational formulation

$$\begin{aligned} u \in H^1(\Omega) : \quad & \int_{\Omega} (\alpha(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x}) v(\mathbf{x}) \, d\mathbf{x} + \int_{\partial\Omega \setminus \Gamma_0} h v \, dS \\ u &= g \quad \text{on } \Gamma_0 \end{aligned} \quad (2.9.3)$$

for all $v \in H^1(\Omega)$ with $v|_{\Gamma_0} = 0$.

In the variational formulations of 2nd-order elliptic BVPs of Sect. 2.8:

Dirichlet boundary conditions are *directly imposed* on trial space and (in homogeneous form) on test space.

Terminology:

essential boundary conditions

Neumann boundary conditions are enforced *only* through the variational equation.

Terminology:

natural boundary conditions

The attribute “natural” has been coined, because Neumann boundary conditions “naturally” emerge when removing constraints on the boundary, as we have seen for the partially free membrane of Rem. 2.4.16.

Remark 2.9.4 (Admissible Dirichlet data).

Requirement for “Dirichlet data” $g : \partial\Omega \mapsto \mathbb{R}$ in (2.4.14):

there is $u \in H^1(\Omega)$ such that $u|_{\partial\Omega} = g$

Analogous to Thm. 2.2.17:

2.9

If $g : \partial\Omega \mapsto \mathbb{R}$ is piecewise continuously differentiable (and bounded with bounded piecewise derivatives), then it can be extended to an $u_0 \in H^1(\Omega)$, if and only if it is continuous on $\partial\Omega$.

Bottom line:

Dirichlet boundary values have to be continuous

This is also stipulated by physical insight, e.g. in the case of the taut membrane model of Sect. 2.1.1: discontinuous displacement on $\partial\Omega$ would entail ripping apart the membrane.



Remark 2.9.5 (Admissible Neumann data).

In the variational problem (2.8.15) Neumann data $h : \partial\Omega \mapsto \mathbb{R}$ enter through the linear form on the right hand side

$$\ell(v) := \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} + \int_{\partial\Omega} h(\mathbf{x})v(\mathbf{x}) \, dS(\mathbf{x}) .$$

Remember the discussion in the beginning of Sect. 2.2, also Rem. 2.3.16: we have to establish that ℓ is continuous on $H_*^1(\Omega)$ defined in (2.8.14). This is sufficient, because the coefficient function κ is uniformly positive and bounded, see (2.5.4). Thus, the energy $\|\cdot\|_a$ associated with the bilinear form

$$a(u, v) = \int_{\Omega} \kappa(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\mathbf{x}$$

can be bounded from above and below by $|\cdot|_{H^1(\Omega)}$, cf. the estimate (2.3.14).

Theorem 2.9.6. Second Poincaré-Friedrichs inequality]

If $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, is bounded, then

$$\exists C = C(\Omega) > 0: \quad \|u\|_0 \leq C \operatorname{diam}(\Omega) \|\operatorname{grad} u\|_0 \quad \forall u \in H_*^1(\Omega) .$$

notation: $C = C(\Omega)$ indicates that the constant C may depend on the shape of the domain Ω .

Proof. (for $d = 1$, $\Omega = [0, 1]$ only, technically difficult in higher dimensions)

As in the proof of Thm. 2.2.16, we employ a density argument and assume that u is sufficiently smooth, $u \in C^1([0, 1])$.

By the fundamental theorem of calculus (2.4.1)

$$u(x) = u(y) + \int_y^x \frac{du}{dx}(\tau) d\tau , \quad 0 \leq x, y \leq 1 .$$

►
$$u(x) = \int_0^1 u(x) dy = \underbrace{\int_0^1 u(y) dy}_{=0} + \int_0^1 \int_y^x \frac{du}{dx}(\tau) d\tau dy .$$

Then use the Cauchy-Schwarz inequality (2.2.15)

$$u(x)^2 \leq \int_0^1 \int_y^x 1 d\tau dy \int_0^1 \int_y^x \left| \frac{du}{dx}(\tau) \right|^2 d\tau dy \leq \int_0^1 \left| \frac{du}{dx}(\tau) \right|^2 d\tau .$$

Integrate over Ω yields the estimate

$$\|u\|_0^2 = \int_0^1 u^2(x) dx \leq \int_0^1 \left| \frac{du}{dx}(\tau) \right|^2 d\tau = \|u\|_{H^1(\Omega)}^2. \quad (\square)$$

By (2.2.15), Thm. 2.9.6 implies the continuity of the first term in ℓ .

Continuity of the boundary contribution to ℓ hinges on a **trace theorem**

Theorem 2.9.7 (Multiplicative trace inequality).

$$\exists C = C(\Omega) > 0: \|u\|_{L^2(\partial\Omega)}^2 \leq C \|u\|_{L^2(\Omega)} \cdot \|u\|_{H^1(\Omega)} \quad \forall u \in H^1(\Omega).$$

Proof. (for $d = 1$, $\Omega = [0, 1]$ only, technically difficult in higher dimensions)

As in the proof of Thms. 2.2.16, 2.9.6, we employ a density argument and assume that u is sufficiently smooth, $u \in C^1([0, 1])$.

By the fundamental theorem of calculus (2.4.1):

$$u(1)^2 = \int_0^1 \frac{dw}{d\xi}(x) dx , \quad \text{with} \quad w(\xi) := \xi u^2(\xi) ,$$

► $u(1)^2 = \int_0^1 u^2(x) + 2u(x) \frac{du}{dx}(x) dx .$

Then use the Cauchy-Schwarz inequality (2.2.15)

$$u(1)^2 \leq \int_0^1 u^2(x) dx + 2 \int_0^1 |u(x)| \left| \frac{du}{dx}(x) \right| dx \leq \|u\|_0^2 + 2 \|u\|_0 \left\| \frac{du}{dx} \right\|_0 .$$

A similar estimate holds for $u(0)^2$. □

Now we can combine

- the Cauchy-Schwarz inequality (2.2.15) on $\partial\Omega$,

- the 2nd Poincaré-Friedrichs inequality of Thm. 2.9.6,
- the multiplicative trace inequality of Thm. 2.9.7:

$$\int_{\partial\Omega} hv \, dS \stackrel{(2.2.15)}{\leq} \|h\|_{L^2(\partial\Omega)} \|v\|_{L^2(\partial\Omega)} \stackrel{\text{Thm. 2.9.7}}{\leq} \|h\|_{L^2(\partial\Omega)} \|v\|_{H^1(\Omega)}$$

Thm. 2.9.6

$$\leq \|h\|_{L^2(\partial\Omega)} |v|_{H^1(\Omega)} \quad \forall v \in H_*^1(\Omega) .$$


$h \in L^2(\partial\Omega)$ provides valid Neumann data for the 2nd order elliptic BVP (2.9.1).

In particular Neumann data h can be *discontinuous*.

3

Finite Element Methods (FEM)

In this chapter:

- Problem : linear scalar second-order elliptic boundary value problem → Ch. 2
- Perspective : **variational** interpretation in Sobolev spaces → Sect. 2.8
- Objective : algorithm for the computation of an **approximate numerical solution**

Preface

Sect. 1.5.1 introduced the fundamental ideas of the **Galerkin discretization** of variational problems, or, equivalently, of minimization problems, posed over function spaces. A key ingredient are suitably

chosen finite-dimensional trial and test spaces, equipped with ordered bases.

In Sect. 1.5.1.2 the abstract approach was discussed for two-point boundary value problems and the concrete case of **piecewise linear** trial and test spaces, built upon a partition (mesh/grid) of the interval (domain). In this context the locally supported tent functions lent themselves as natural basis functions.

This chapter is devoted to extending the linear finite element method in 1D to

- 2nd-order linear variational problems on bounded spatial domains Ω in two and three dimensions,
- piecewise polynomial trial/test functions of higher degree.

The leap from $d = 1$ to $d = 2$ will encounter additional difficulties and many new aspects. This chapter will elaborate on them and present policies how to tackle them.

Throughout, we will restrict ourselves to **linear 2nd-order elliptic variational problems** on spatial domains $\Omega \in \mathbb{R}^d$, $d = 2, 3$, with the properties listed in Rem. 2.1.1.

→ 2nd-order elliptic Dirichlet problem:

$$\begin{aligned} u &\in H^1(\Omega) , \\ u = g &\text{ on } \partial\Omega : \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega) , \end{aligned} \quad (2.3.3)$$

with *continuous* (→ Rem. 2.9.4) Dirichlet data $g \in C^0(\partial\Omega)$.

→ 2nd-order elliptic Neumann problems:

$$u \in H_*^1(\Omega) : \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u) \cdot \operatorname{grad} v \, d\mathbf{x} = \int_{\Omega} fv \, d\mathbf{x} + \int_{\partial\Omega} hv \, dS \quad \forall v \in H_*^1(\Omega) , \quad (2.8.15)$$

with *piecewise continuous* (→ Rem. 2.9.5) Neumann data $h \in C_{\text{pw}}^0(\partial\Omega)$ that satisfy the **compatibility condition** (2.8.12).

A simpler version with homogeneous Neumann data and reaction term:

$$u \in H^1(\Omega) : \int_{\Omega} \boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v + c(\mathbf{x})uv \, d\mathbf{x} = \int_{\Omega} fv \, d\mathbf{x} \quad \forall v \in H^1(\Omega) , \quad (3.0.1)$$

with reaction coefficient $c : \Omega \mapsto \mathbb{R}^+$, $c \in C_{\text{pw}}^0(\Omega)$. Note that no compatibility conditions is required in this case.

Rem. 1.5.3 still applies: all functions (coefficient $\boldsymbol{\alpha}$, source function f , Dirichlet data g) may be given only in procedural form.

3.1 Galerkin discretization

Recall the concept of “discretization”, see Sect. 1.5:

Not a moot point: any computer can only handle a finite amount of information (reals)

Variational boundary value
problem

DISCRETIZATION

System of a finite number of
equations for (real) unknowns

Targetted: linear variational problem (1.4.5)

$$u \in V_0: \quad a(u, v) = l(v) \quad \forall v \in V_0 , \quad (3.1.1)$$

- $V_0 \hat{=} \text{vector space (Hilbert space) (usually a Sobolev space} \rightarrow \text{Sect. 2.2) with norm } \|\cdot\|_V,$
- $a(\cdot, \cdot) \hat{=} \text{bilinear form, continuous in } V_0,$
- $\ell \hat{=} \text{continuous linear form in the sense of, cf. (2.2.1),}$

$$\exists C > 0: |\ell(v)| \leq C \|v\|_V \quad \forall v \in V_0 . \quad (3.1.2)$$

If a is symmetric and positive definite (\rightarrow Def. 2.1.22), we may choose $\|\cdot\|_V := \|\cdot\|_a$, “energy norm”, see Def. 2.1.24.

Recall from Sect. 1.5.1:

Idea of Galerkin discretization

Replace V_0 in (3.1.1) with a **finite dimensional subspace**.
 $(V_{0,N}$ called Galerkin (or discrete) trial space/test space)

Twofold nature of symbol “ N ”:

- N = formal index, tagging “discrete entities” (\rightarrow “finite amount of information”)
- $N = \dim V_{N,0} \hat{=}$ dimension of Galerkin trial/test space



Discrete variational problem, cf. (1.5.7),

$$u_N \in V_{0,N}: \quad a(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N} . \quad (3.1.3)$$

Galerkin solution

Theorem 3.1.4 (Existence and uniqueness of solutions of discrete variational problems).

If the bilinear form $a : V_0 \times V_0 \mapsto \mathbb{R}$ is symmetric and positive definite (\rightarrow Def. 2.1.22) and the linear form $\ell : V_0 \mapsto \mathbb{R}$ is continuous in the sense of

$$\exists C_\ell > 0: |\ell(u)| \leq C_\ell \|u\|_a \quad \forall u \in V_0 , \quad (2.2.1)$$

then the discrete variational problem has a unique **Galerkin solution** $u_N \in V_{0,N}$ that satisfies the stability estimate (\rightarrow Sect. 2.3.2)

$$\|u_N\|_a \leq C_\ell . \quad (3.1.5)$$

Proof. Uniqueness of u_N is clear:

$$\begin{aligned} a(u_N, v_N) &= \ell(v_N) \quad \forall v_N \in V_{0,N} \\ a(w_N, v_N) &= \ell(v_N) \quad \forall v_N \in V_{0,N} \quad \Rightarrow \quad a(u_N - w_N, v_N) = 0 \quad \forall v_N \in V_{N,0} \\ v_N := u_N - w_N &\in V_{0,N} \quad \|u_N - w_N\|_a = 0 \quad \xrightarrow{\text{a s.p.d.}} \quad u_N - w_N = 0 . \end{aligned}$$

The discrete linear variational problem (3.1.3) is set in the *finite-dimensional* space $V_{0,N}$. Thus, uniqueness of solutions is equivalent to existence of solutions (\rightarrow linear algebra).

If you do not like this abstract argument, wait and see the equivalence of (3.1.3) with a linear system

of equations. It will turn out that under the assumptions of the theorem, the resulting system matrix will be symmetric and positive definite in the sense of [14, Def. 2.7.1].

The estimate (3.1.5) is immediate from setting $v_N := u_N$ in (3.1.3)

$$|\mathbf{a}(u_N, u_N)| = |\ell(u_N)| \leq C_\ell (\mathbf{a}(u_N, u_N))^{1/2} . \quad \square$$

Recall from Sect. 1.5.1:

2nd step of Galerkin discretization:

Introduce (ordered) **basis** \mathfrak{B}_N of $V_{0,N}$:

$$\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\} \subset V_N , \quad V_N = \text{Span } \{\mathfrak{B}_N\} , \quad N := \dim(V_N) .$$

3.1

► Unique basis representations:

$$u_N = \mu_1 b_N^1 + \cdots + \mu_N b_N^N, \quad \mu_i \in \mathbb{R} \\ v_N = \nu_1 b_N^1 + \cdots + \nu_N b_N^N, \quad \nu_i \in \mathbb{R}$$

: plug into (3.1.3).

Of course, there are infinitely many ways to choose the basis \mathfrak{B}_N . Below we will study the impact of different choices.

What follows repeats the derivation of (1.5.16) and, in particular, (1.5.40).

$$u_N \in V_{0,N}: \quad \mathbf{a}(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N}. \quad (3.1.3)$$



$$\left[\begin{array}{l} u_N = \mu_1 b_N^1 + \cdots + \mu_N b_N^N, \mu_i \in \mathbb{R} \\ v_N = \nu_1 b_N^1 + \cdots + \nu_N b_N^N, \nu_i \in \mathbb{R} \end{array} \right]$$

$$\sum_{k=1}^N \sum_{j=1}^N \mu_k \nu_j \mathbf{a}(b_N^k, b_N^j) = \sum_{j=1}^N \nu_j \ell(b_N^j) \quad \forall \nu_1, \dots, \nu_N \in \mathbb{R},$$



$$\sum_{j=1}^N \nu_j \left(\sum_{k=1}^N \mu_k \mathbf{a}(b_N^k, b_N^j) - \ell(b_N^j) \right) = 0 \quad \forall \nu_1, \dots, \nu_N \in \mathbb{R} ,$$



$$\sum_{k=1}^N \mu_k \mathbf{a}(b_N^k, b_N^j) = \ell(b_N^j) \quad \text{for } j = 1, \dots, N .$$



$$[\vec{\mu} = (\mu_1, \dots, \mu_N)^\top \in \mathbb{R}^N]$$

$$\mathbf{A} = \left(\mathbf{a}(b_N^k, b_N^j) \right)_{j,k=1}^N \in \mathbb{R}^{N,N} ,$$

$$\vec{\varphi} = \left(\ell(b_N^j) \right)_{j=1}^N .$$

$$\boxed{\mathbf{A}\vec{\mu} = \vec{\varphi}} , \text{ with}$$

A linear system of equations

Linear Discrete variational problem

$$u_N \in V_{0,N}: \quad a(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N}$$

Choosing basis \mathcal{B}_N

Linear system
of equations
 $\mathbf{A}\vec{\mu} = \vec{\varphi}$

Galerkin matrix: $\mathbf{A} = \left(a(b_N^k, b_N^j) \right)_{j,k=1}^N \in \mathbb{R}^{N,N},$

Right hand side vector: $\vec{\varphi} = \left(\ell(b_N^j) \right)_{j=1}^N \in \mathbb{R}^N,$

Coefficient vector: $\vec{\mu} = (\mu_1, \dots, \mu_N)^\top \in \mathbb{R}^N,$

Recovery of solution: $u_N = \sum_{k=1}^N \mu_k b_N^k.$

(Legacy) terminology for FEM:

Galerkin matrix $=$ stiffness matrix

Right hand side vector $=$ load vector

Galerkin matrix for $(u, v) \mapsto \int_{\Omega} uv \, dx =$ mass matrix

Corollary 3.1.6.

(3.1.3) has unique solution $\Leftrightarrow \mathbf{A}$ nonsingular

3.1

p. 269

Remark 3.1.7 (Impact of choice of basis).

Choice of \mathcal{B}_N in theory does **not** affect $u_N \Rightarrow$ No impact on discretization error !

But: Key properties (e.g., conditioning) of matrix \mathbf{A} crucially depend on basis \mathcal{B}_N !

Lemma 3.1.8. Consider (3.1.3) and two bases of $V_{0,N}$,

$$\mathcal{B}_N := \{b_N^1, \dots, b_N^N\} \quad , \quad \underline{\mathcal{B}}_N := \{\underline{b}_N^1, \dots, \underline{b}_N^N\} \quad ,$$

related by

$$\underline{b}_N^j = \sum_{k=1}^N s_{jk} b_N^k \quad \text{with} \quad \mathbf{S} = (s_{jk})_{j,k=1}^N \in \mathbb{K}^{N,N} \quad \text{regular.}$$

► Galerkin matrices $\mathbf{A}, \underline{\mathbf{A}} \in \mathbb{K}^{N,N}$, right hand side vectors $\vec{\varphi}, \underline{\vec{\varphi}} \in \mathbb{K}^N$, and coefficient vectors $\vec{\mu}, \underline{\vec{\mu}} \in \mathbb{R}^N$, respectively, satisfy

$$\underline{\mathbf{A}} = \mathbf{S} \mathbf{A} \mathbf{S}^T \quad , \quad \underline{\vec{\varphi}} = \mathbf{S} \vec{\varphi} \quad , \quad \underline{\vec{\mu}} = \mathbf{S}^{-T} \vec{\mu} . \quad (3.1.9)$$

Proof.

$$\underline{\mathbf{A}}_{lm} = \mathbf{a}(\underline{b}_N^m, \underline{b}_N^l) = \sum_{k=1}^N \sum_{j=1}^N s_{mk} \mathbf{a}(b_N^k, b_N^j) s_{lj} = \sum_{k=1}^N \underbrace{\left(\sum_{j=1}^N s_{lj} \mathbf{A}_{jk} \right)}_{(\mathbf{SA})_{lk}} s_{mk} = (\mathbf{SAS}^T)_{lm},$$

Reminder of linear algebra:

Definition 3.1.10 (Congruent matrices).

*Two matrices $\mathbf{A} \in \mathbb{K}^{N,N}$, $\mathbf{B} \in \mathbb{K}^{N,N}$, $N \in \mathbb{N}$, are called **congruent**, if there is a regular matrix $\mathbf{S} \in \mathbb{K}^{N,N}$ such that $\mathbf{B} = \mathbf{SAS}^H$.*



Equivalence relation on square matrices

Lemma 3.1.11.

Matrix property invariant under congruence



Property of Galerkin matrix invariant under change of basis \mathfrak{B}_N

3.1

p. 271

- regularity → [14, Def. 2.0.1]
- symmetry
- positive definiteness → [14, Def. 2.7.1]

Matrix properties invariant under congruence:



3.2 Case study: Triangular linear FEM in two dimensions

This section elaborates how to extend the linear finite element Galerkin discretization of Sect. 1.5.1.2 to two dimensions. Familiarity with the 1D setting is essential for understanding the current section.

3.2

Initial focus: well-posed 2nd-order linear variational problem posed on $H^1(\Omega)$ (→ Def. 2.2.12)

p. 272

Example: Neuman problem with homogeneous Neumann data and reaction term

$$u \in H^1(\Omega): \quad \int_{\Omega} \boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v + c(\mathbf{x}) u v \, d\mathbf{x} = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in H^1(\Omega) , \quad (3.0.1)$$

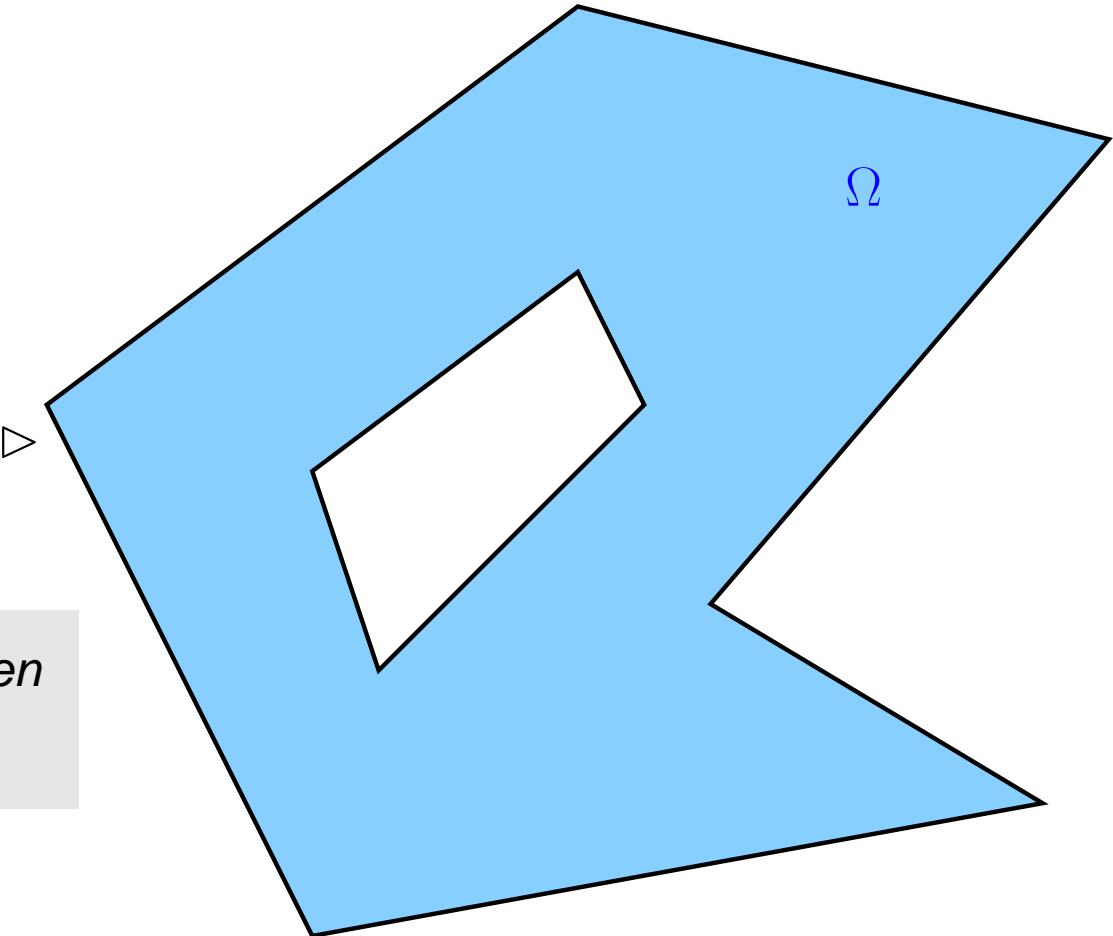
$\Updownarrow \leftarrow$ see Sect. 2.4

BVP:
$$\begin{aligned} -\operatorname{div}(\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u) + c(\mathbf{x}) u &= f && \text{in } \Omega , \\ \operatorname{grad} u \cdot \mathbf{n} &= 0 && \text{on } \partial\Omega . \end{aligned}$$

Assumptions on domain $\Omega \subset \mathbb{R}^2$,
see Rem. 2.1.1:

Ω is a **polygon**

polygon with 10 corners



By default, the domain Ω is assumed to be an *open* set, that is, $x \in \Omega$ implies $x \notin \partial\Omega$!

3.2.1 Triangulations

What is the 2D counterpart of mesh/grid \mathcal{M} from Sect. (1.5.1.2) ?

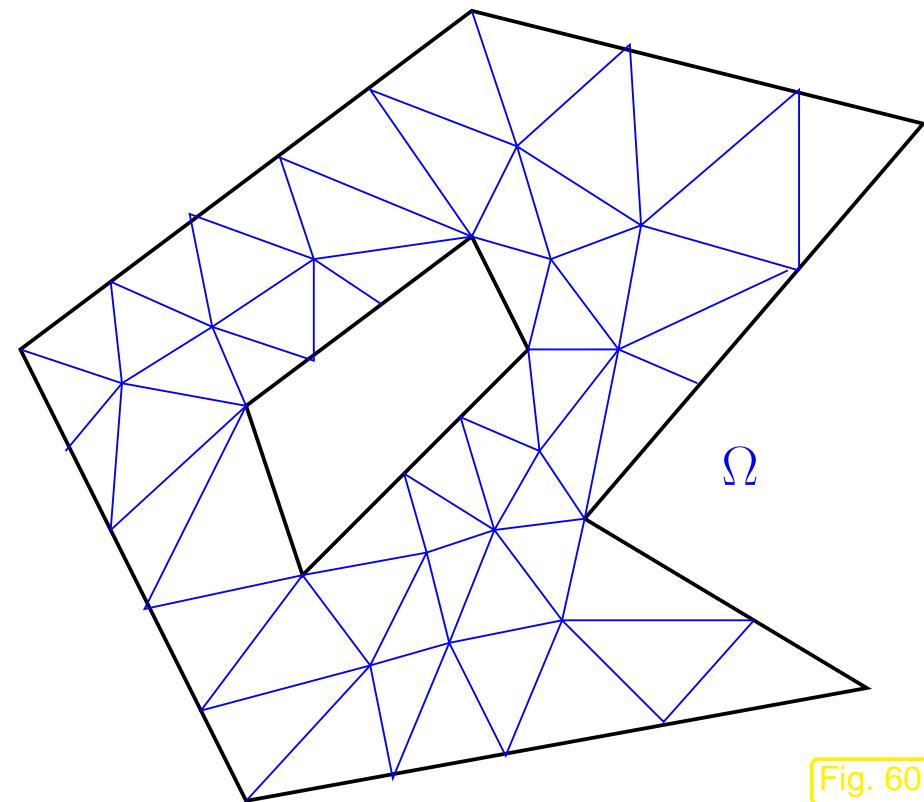


Fig. 60

Triangulation \mathcal{M} of Ω :

- (i) $\mathcal{M} = \{K_i\}_{i=1}^M, M \in \mathbb{N}, K_i \hat{=} \text{open triangle}$
- (ii) disjoint interiors: $i \neq j \Rightarrow K_i \cap K_j = \emptyset$
- (iii) tiling property: $\bigcup_{i=1}^M \overline{K}_i = \overline{\Omega}$
- (iv) intersection $\overline{K}_i \cap \overline{K}_j, i \neq j$,
is
 - either \emptyset
 - or an edge of both triangles
 - or a vertex of both triangles

notation: $\hat{=}$ a subset of \mathbb{R}^d together with its boundary (“closure”)

Parlance: vertices of triangles = nodes of mesh (= set $\mathcal{V}(\mathcal{M})$)

A mesh that does not comply with the property (iv)
from above.

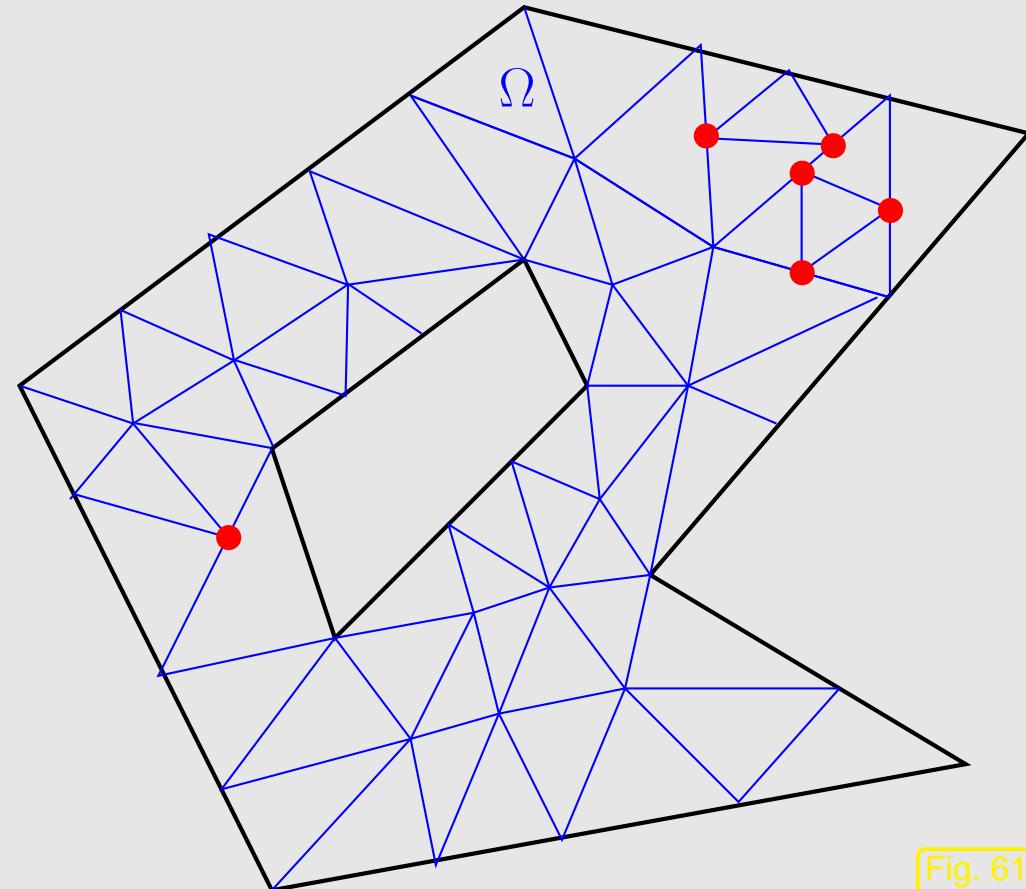


Fig. 61

3.2.2 Linear finite element space

Next goal: generalize the spline space $\mathcal{S}_1^0(\mathcal{M}) \subset H^1([a, b])$ of piecewise linear functions on a 1D grid \mathcal{M} , see Fig. 23, that was used as Galerkin trial/test space in 1D:

$$V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M}) := \left\{ v \in C^0([0, 1]): v|_{[x_{i-1}, x_i]} \text{ linear, } i = 1, \dots, M, v(a) = v(b) = 0 \right\} .$$

$d = 1$

$d = 2$

Grid/mesh **cells**: intervals $]x_{i-1}, x_i[, i = 1, \dots, M$

triangles $K_i, i = 1, \dots, M$

Linear functions: $x \in \mathbb{R} \mapsto \alpha + \beta \cdot x, \alpha, \beta \in \mathbb{R}$

$\boldsymbol{x} \in \mathbb{R}^2 \mapsto \alpha + \boldsymbol{\beta} \cdot \boldsymbol{x}, \alpha \in \mathbb{R}, \boldsymbol{\beta} \in \mathbb{R}^2$

$$V_{0,N} = \mathcal{S}_1^0(\mathcal{M}) := \left\{ v \in C^0(\bar{\Omega}): \forall K \in \mathcal{M}: \begin{array}{l} v|_K(\boldsymbol{x}) = \alpha_K + \boldsymbol{\beta}_K \cdot \boldsymbol{x}, \\ \alpha_K \in \mathbb{R}, \boldsymbol{\beta}_K \in \mathbb{R}^2, \boldsymbol{x} \in K \end{array} \right\} \subset H^1(\Omega)$$

see Thm. 2.2.17

Functions of the form $\mathbf{x} \mapsto \alpha_K + \beta_K \cdot \mathbf{x}$, $\alpha_K \in \mathbb{R}$, $\beta_K \in \mathbb{R}^2$ are called (affine) linear.

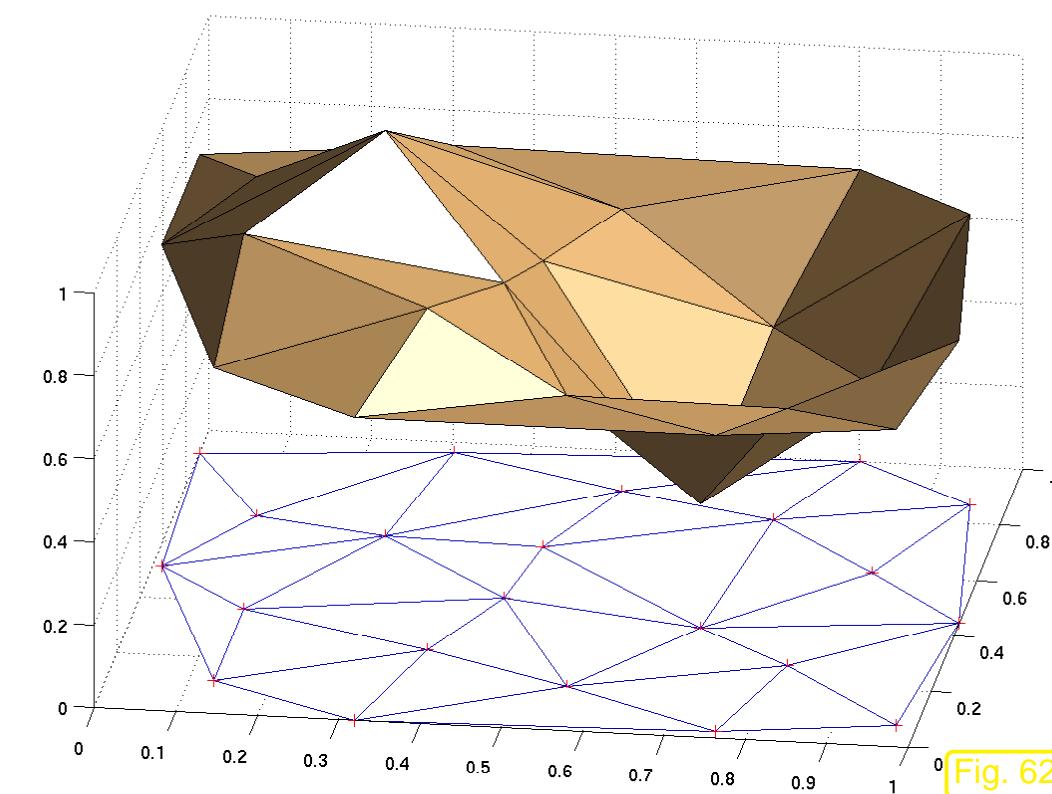
notation:

$$\mathcal{S}_1^0(\mathcal{M})$$

continuous functions, cf. $C^0(\Omega)$

locally 1st degree polynomials

scalar functions



continuous piecewise affine linear function $\in \mathcal{S}_1^0(\mathcal{M})$ on a triangular mesh \mathcal{M}

3.2.3 Nodal basis functions

Next goal: generalization of “tent functions”, see (1.5.49).

Recall condition (1.5.50), which *defines* a tent function in the space $\mathcal{S}_1^0(\mathcal{M})$. This approach carries over to 2D.

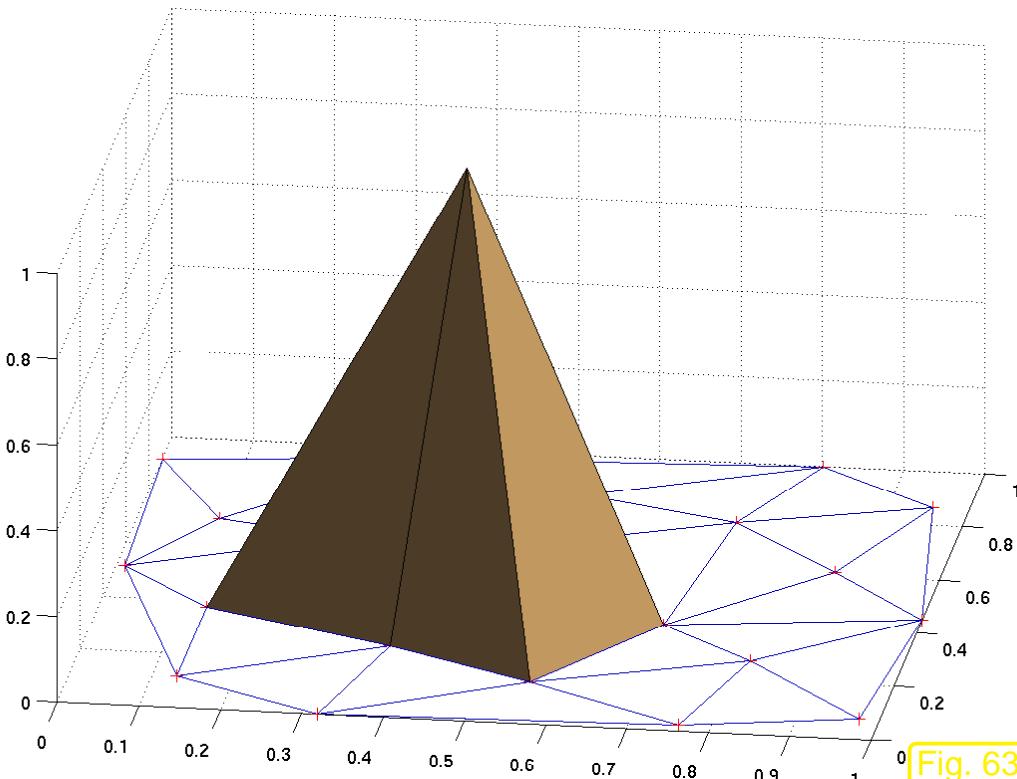
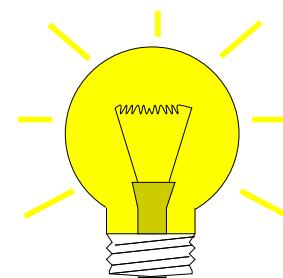


Fig. 63



Idea: define (?) basis function

$$b_N^{\mathbf{x}}, \mathbf{x} \in \mathcal{V}(\mathcal{M}), \text{ by}$$

$$b_N^{\mathbf{x}}(\mathbf{y}) = \begin{cases} 1 & , \text{ if } \mathbf{y} = \mathbf{x} , \\ 0 & , \text{ if } \mathbf{y} \in \mathcal{V}(\mathcal{M}) \setminus \{\mathbf{x}\} \end{cases} \quad (3.2.1)$$

Is this possible ?

Reasoning: there is exactly one plane through three non-collinear points in \mathbb{R}^3 . The graph of a linear function $\mathbb{R}^2 \mapsto \mathbb{R}$ is a plane.

➤ On a triangle K with vertices $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$: (affine) linear $q : K \mapsto \mathbb{R}$ uniquely determined by values $q(\mathbf{a}^i)$.

 $v_N \in \mathcal{S}_1^0(\mathcal{M})$ uniquely determined by $\{v_N(\mathbf{x}), \mathbf{x} \text{ node of } \mathcal{M}\}$!

$$\dim \mathcal{S}_1^0(\mathcal{M}) = \#\mathcal{V}(\mathcal{M})$$

$(\mathcal{V}(\mathcal{M})$) = set of nodes (= vertices of triangles) of \mathcal{M})

Writing $\mathcal{V}(\mathcal{M}) = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$, the **nodal basis** $\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\}$ of $\mathcal{S}_1^0(\mathcal{M})$ is defined by the conditions

$$b_N^i(\mathbf{x}^j) = \begin{cases} 1 & , \text{if } i = j \\ 0 & \text{else,} \end{cases} \quad i, j \in \{1, \dots, N\} .$$

(3.2.2)

Ordering (\leftrightarrow numbering) of nodes assumed !

Piecewise linear nodal basis function ("hat function")

$$u_N = \sum_{i=1}^N \mu_i b_N^i \in \mathcal{S}_1^0(\mathcal{M})$$

- ▶ coefficient μ_j = "nodal value" of u_N at j -th node of \mathcal{M}

$$u_N(\mathbf{x}^j) = \mu_j$$

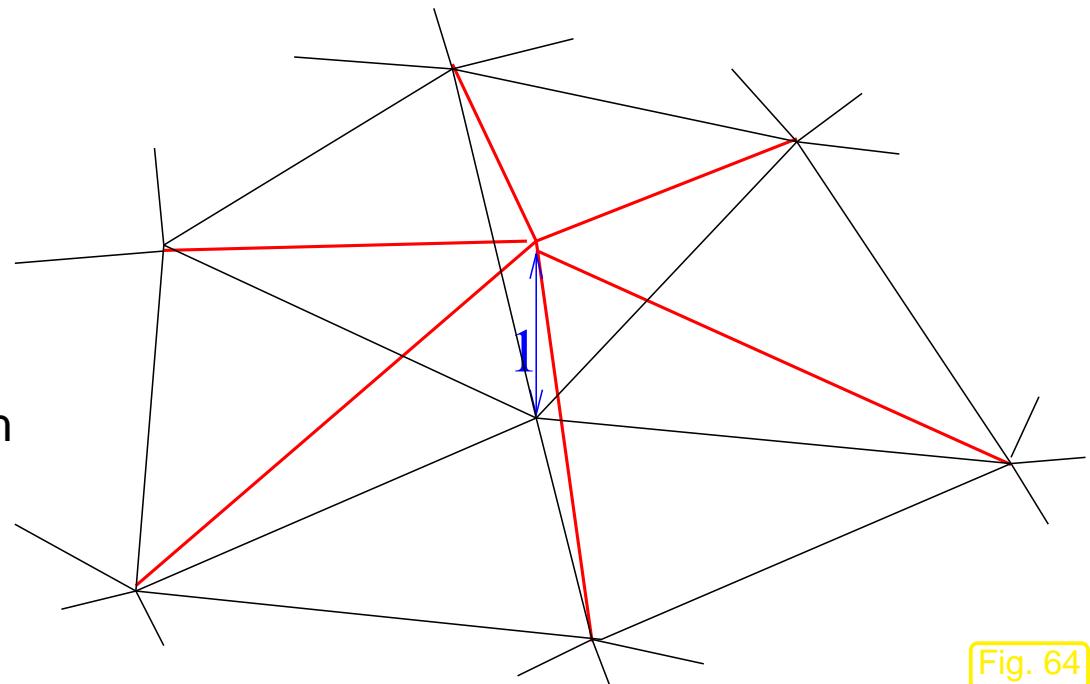


Fig. 64

Remark 3.2.3 (Linear finite element space for homogeneous Dirichlet problem).

Recall that the Dirichlet problem with homogeneous boundary conditions $u|_{\partial\Omega} = 0$ is posed on the Sobolev space $H_0^1(\Omega)$ (\rightarrow Def. 2.2.10), see (2.3.3), Ex. 2.8.1.

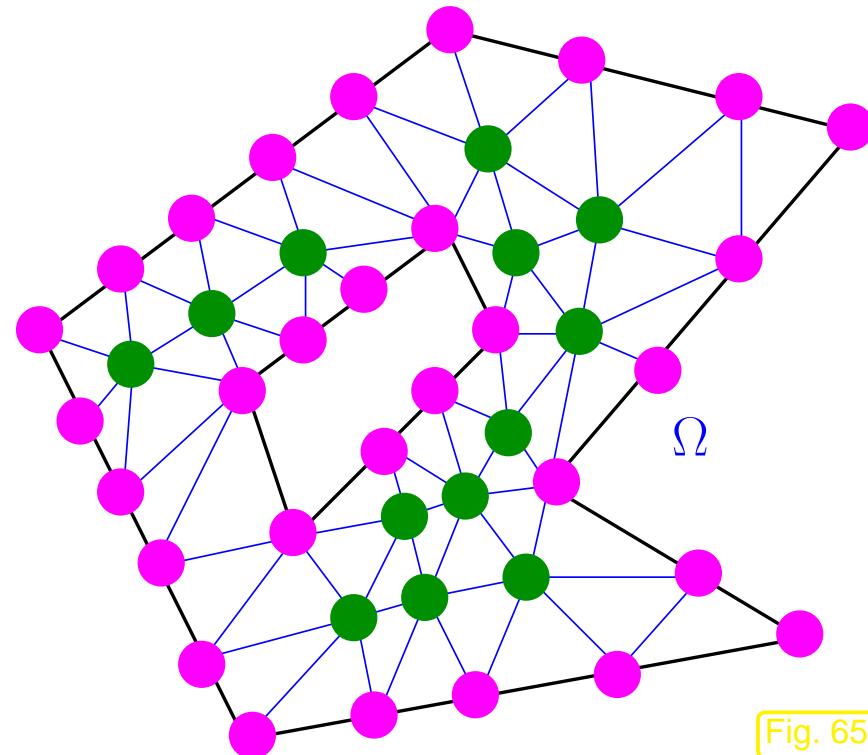
Galerkin space for homogeneous Dirichlet b.c.: $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M}) := \mathcal{S}_1^0(\mathcal{M}) \cap H_0^1(\Omega)$

Notation:

$$\mathcal{S}_{1,0}^0(\mathcal{M}) \xrightarrow{\text{zero on } \partial\Omega, \text{ cf. } H_0^1(\Omega)}$$

► $\mathcal{S}_{1,0}^0(\mathcal{M}) = \text{Span} \left\{ b_N^j : \mathbf{x}^j \in \Omega \text{ (interior node !)} \right\}$

► $\dim \mathcal{S}_{1,0}^0(\mathcal{M}) = \#\{\mathbf{x} \in \mathcal{V}(\mathcal{M}) : \mathbf{x} \notin \partial\Omega\}$



▷ “Location” of nodal basis functions:
(mesh \mathcal{M} → Fig. 146)

- , ● → nodal basis functions of $\mathcal{S}_1^0(\mathcal{M})$
- → nodal basis functions of $\mathcal{S}_{1,0}^0(\mathcal{M})$

Bottom line: the Galerkin trial/test space contained in $H_0^1(\Omega)$ is obtained by dropping all “tent functions” that do not vanish on $\partial\Omega$ from the basis.



3.2.4 Sparse Galerkin matrix

Now: $\mathbf{a} \triangleq$ any (symmetric) bilinear form occurring in a linear 2nd-order variational problem, most general form

$$\mathbf{a}(u, v) := \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \mathbf{grad} u) \cdot \mathbf{grad} v + c(\mathbf{x}) u v \, d\mathbf{x} + \int_{\partial\Omega} h v \, dS , \quad u, v \in H^1(\Omega) . \quad (3.2.4)$$

$b_N^j \triangleq$ nodal basis function associated with vertex \mathbf{x}^j of triangulation \mathcal{M} of Ω , see Sect. 3.2.3.

3.2

Note:

\mathbf{a} symmetric \Rightarrow symmetric Galerkin matrix

Now we study the **sparsity** (\rightarrow [14, Sect. 2.6]) of the Galerkin matrix $\mathbf{A} := \left(\mathbf{a}(b_N^j, b_N^i) \right)_{i,j=1}^N \in \mathbb{R}^{N,N}$, $N := \dim \mathcal{S}_1^0(\mathcal{M}) = \#\mathcal{V}(\mathcal{M})$, see Sect. 3.1.

The consideration are fairly parallel to those that made us understand that the Galerkin matrix for the 1D case was tridiagonal, see (1.5.54).

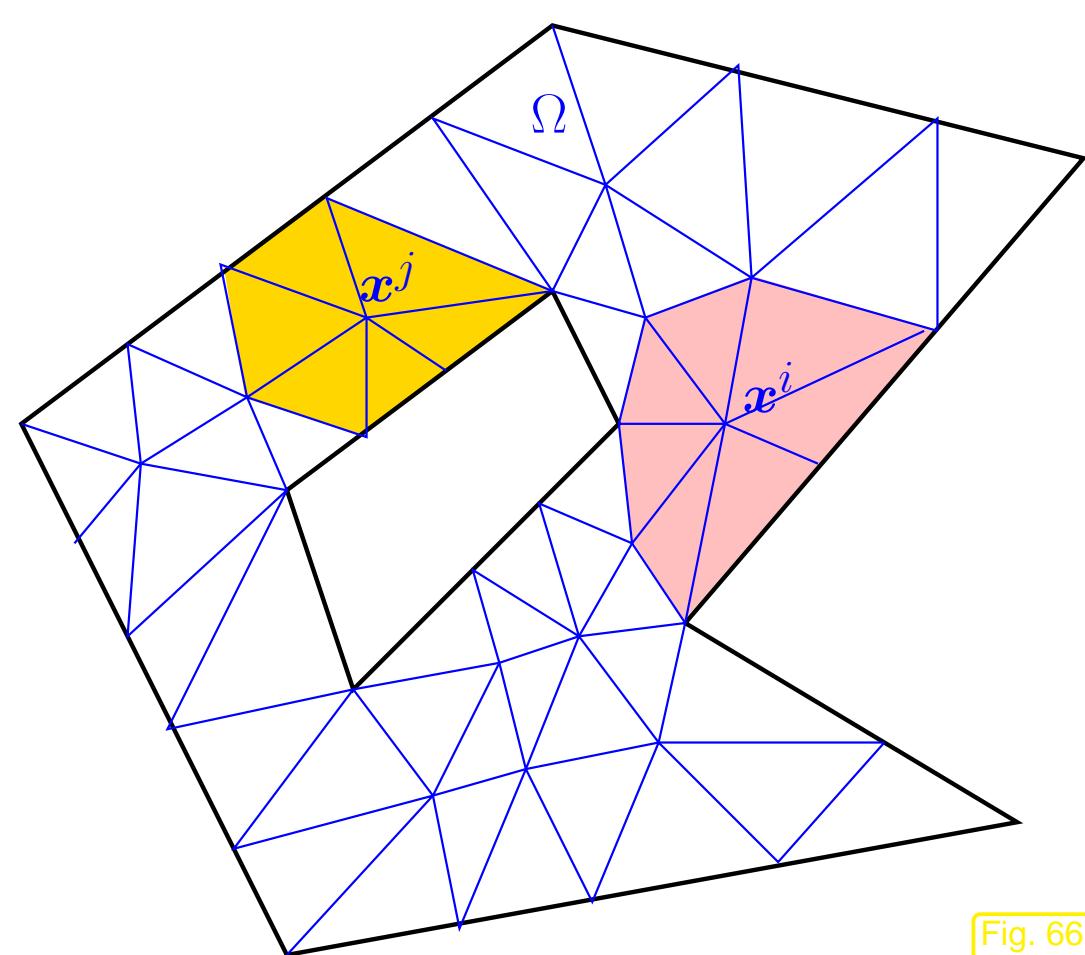


Fig. 66

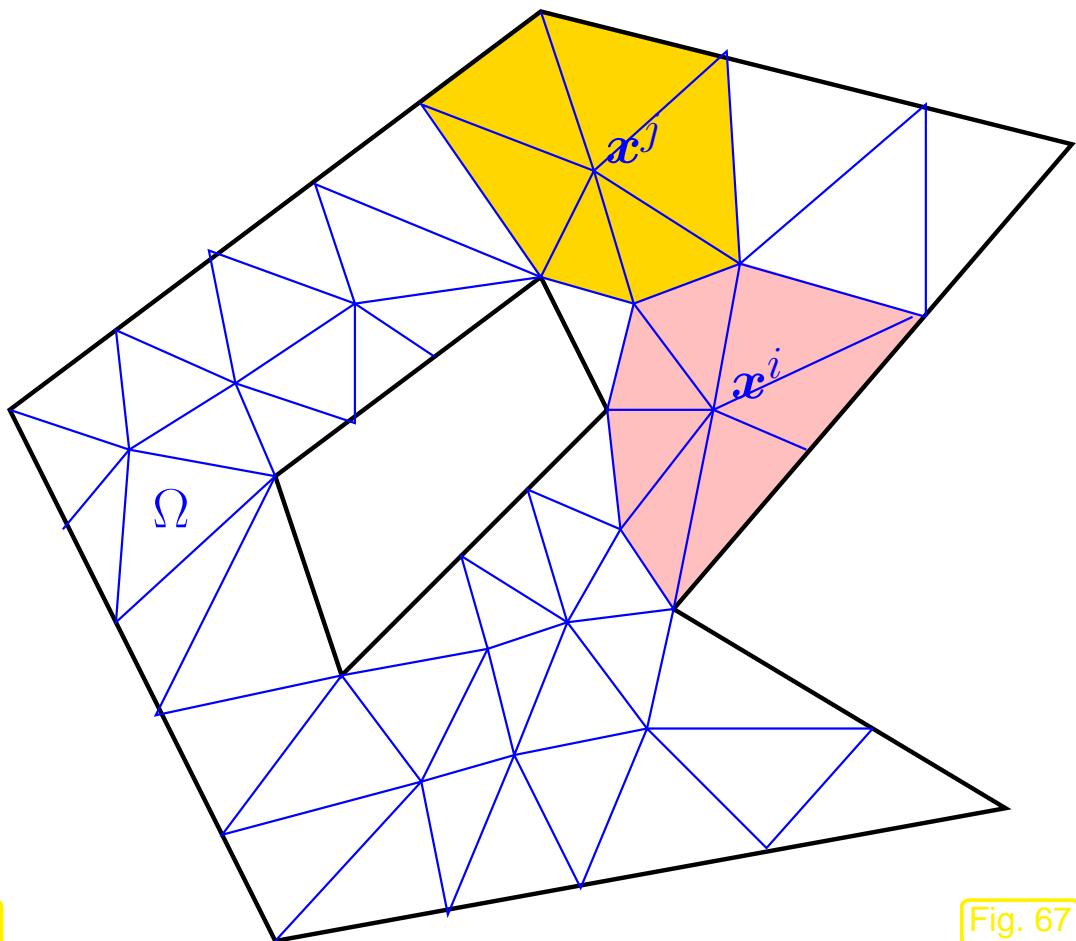


Fig. 67

Nodes $\mathbf{x}^i, \mathbf{x}^j \in \mathcal{V}(\mathcal{M})$
not connected by an edge $\Leftrightarrow \text{Vol}(\text{supp}(b_N^i) \cap \text{supp}(b_N^j)) = 0 \Rightarrow (\mathbf{A})_{ij} = 0$.

Lemma 3.2.5 (Sparsity of Galerkin matrix).

$$\exists C = C(\text{topology of } \Omega): \quad \#\{(i, j) \in \{1, \dots, N\}^2 : (\mathbf{A})_{ij} \neq 0\} \leq 7 \cdot N + C.$$

Proof. Euler's formula (http://en.wikipedia.org/wiki/Euler_characteristic)

$$\#\mathcal{M} - \#\mathcal{E}(\mathcal{M}) + \#\mathcal{V}(\mathcal{M}) = \chi_\Omega, \quad \chi_\Omega = \text{Euler characteristic of } \Omega.$$

Note that χ_Ω is a topological invariant (alternating sum of Betti numbers).

By combinatorial considerations (traverse edges and count triangles):

$$2 \cdot \#\mathcal{E}_I(\mathcal{M}) + \#\mathcal{E}_B(\mathcal{M}) = 3 \cdot \#\mathcal{M},$$

where $\mathcal{E}_I(\mathcal{M}), \mathcal{E}_B(\mathcal{M})$ stand for the sets of interior and boundary edges of \mathcal{M} , respectively.



$$\#\mathcal{E}_I(\mathcal{M}) + 2\#\mathcal{E}_B(\mathcal{M}) = 3(\#\mathcal{V}(\mathcal{M}) - \chi_\Omega).$$

Then use

$$N = \#\mathcal{V}(\mathcal{M}), \quad \text{nnz}(\mathbf{A}) \leq N + 2 \cdot \#\mathcal{E}(\mathcal{M}) \leq 7 \cdot \#\mathcal{V}(\mathcal{M}) - 6\chi_\Omega.$$

□

3.2

p. 287

Recall from [14, Def. 2.6.1]:

Notion 3.2.6 (Sparse matrix). $\mathbf{A} \in \mathbb{K}^{m,n}$, $m, n \in \mathbb{N}$, is *sparse*, if

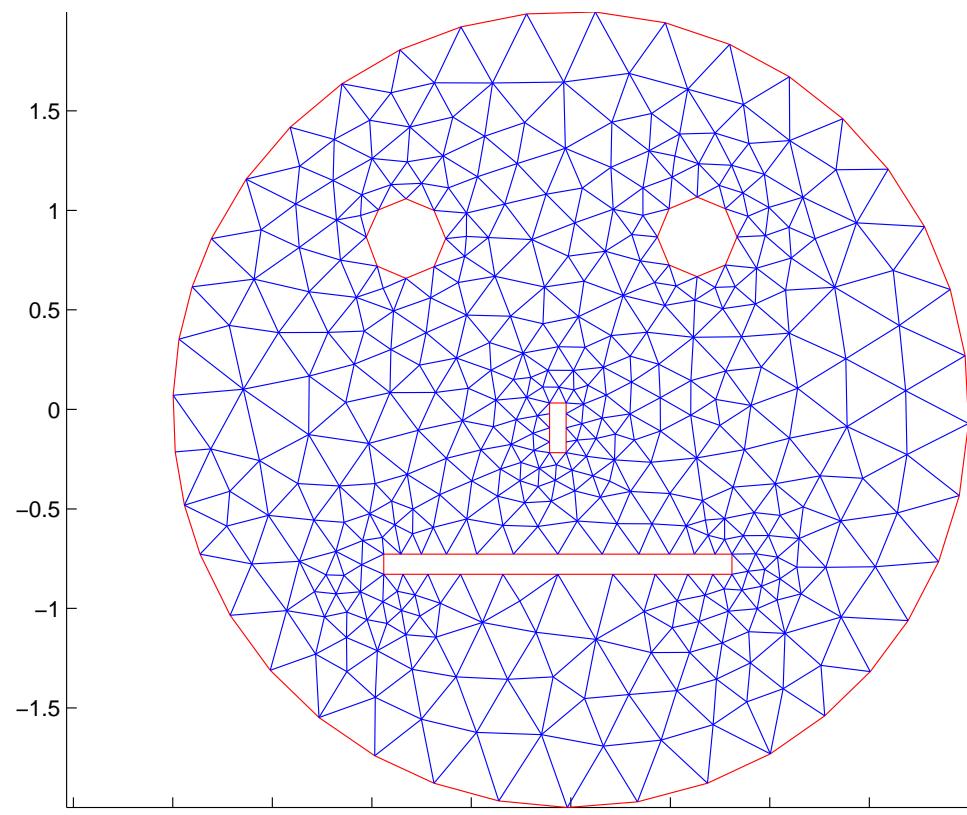
$$\text{nnz}(\mathbf{A}) := \#\{(i, j) \in \{1, \dots, m\} \times \{1, \dots, n\} : a_{ij} \neq 0\} \ll mn .$$

Sloppy parlance: matrix **sparse** : \Leftrightarrow “almost all” entries $= 0$ /“only a few percent of” entries $\neq 0$

Galerkin discretization of a 2nd-order linear variational problems
utilizing the *nodal basis* of $\mathcal{S}_1^0(\mathcal{M})/\mathcal{S}_{1,0}^0(\mathcal{M})$
leads to sparse linear systems of equations.

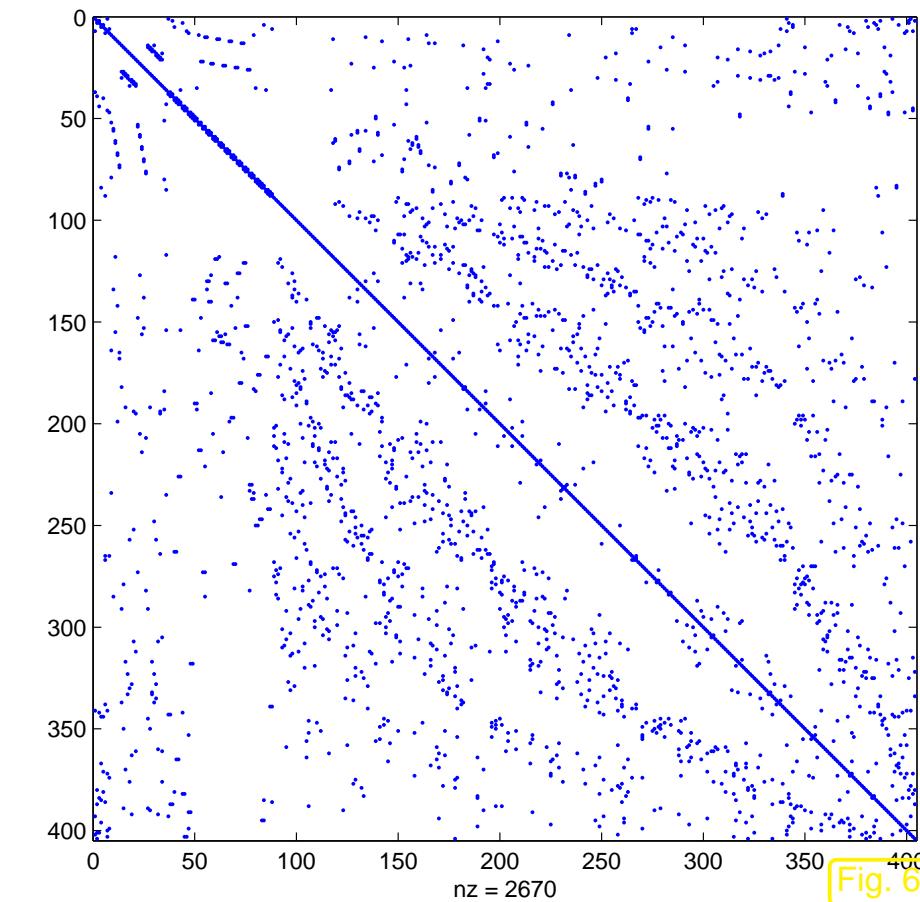
Example 3.2.7 (Sparse Galerkin matrices).

\mathcal{M} = triangular mesh, $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M})$, homogeneous Dirichlet boundary conditions, linear 2nd-order scalar elliptic differential operator.



Triangular mesh \mathcal{M}

Fig. 68



Resulting **sparsity pattern** of Galerkin matrix

Fig. 69

Recall: visualization of sparsity pattern by means of MATLAB spy-command.



3.2.5 Computation of Galerkin matrix

For sake of simplicity consider

$$a(u, v) := \int_{\Omega} \mathbf{grad} u \cdot \mathbf{grad} v \, dx , \quad u, v \in H_0^1(\Omega) .$$

and Galerkin discretization based on

- triangular mesh, see Sect. 3.2.1,

- discrete trial/test space $\mathcal{S}_{1,0}^0(\mathcal{M}) \subset H_0^1(\Omega)$,

- *nodal basis* $\mathfrak{B}_N = \{b_N^j\}$ according to (3.2.1).

► $(\mathbf{A})_{i,j} = \mathbf{a}(b_N^j, b_N^i) = \int_{\Omega} \mathbf{grad} \, b_N^j \cdot \mathbf{grad} \, b_N^i \, dx$

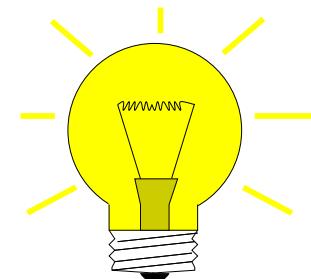
Sect. 3.2.4: we need only study the cases, where $\mathbf{x}^i, \mathbf{x}^j \in \mathcal{V}(\mathcal{M})$

1. are connected by an edge of the triangulation,
2. coincide.

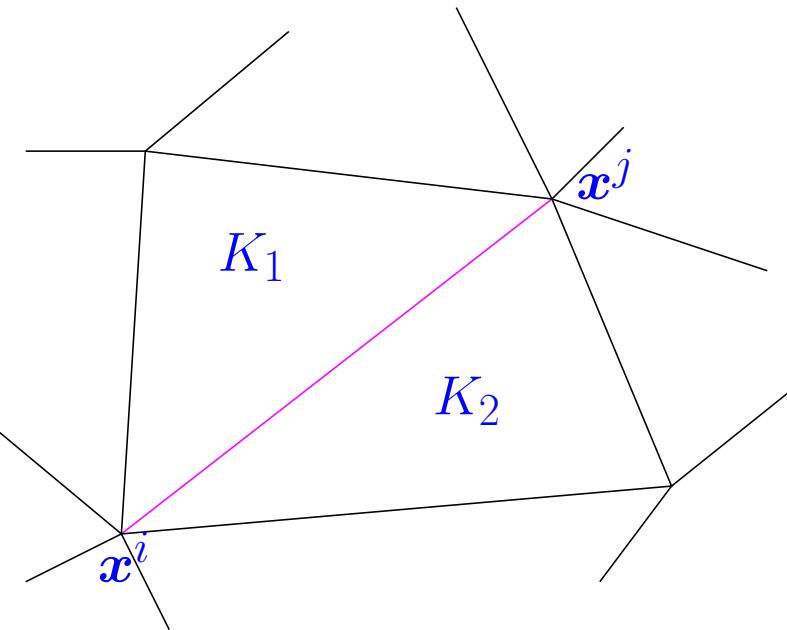
Idea:

“Assembly”

(add up *cell contributions*)



$$(\mathbf{A})_{ij} = \int_{K_1} \mathbf{grad} b_N^j|_{K_1} \cdot \mathbf{grad} b_N^i|_{K_1} dx + \int_{K_2} \mathbf{grad} b_N^j|_{K_2} \cdot \mathbf{grad} b_N^i|_{K_2} dx$$



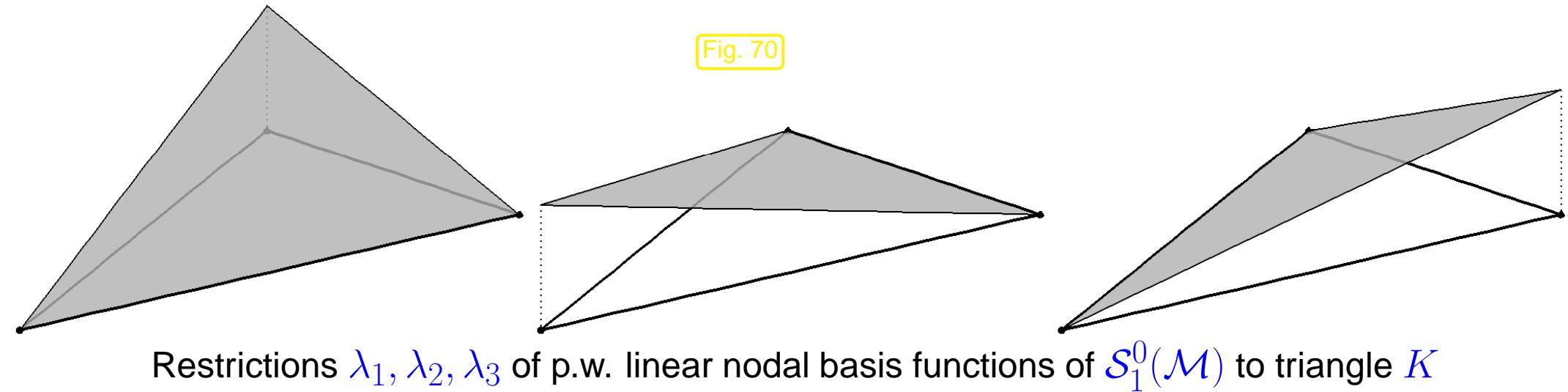
► Zero in on single triangle $K \in \mathcal{M}$:

$$a_K(b_N^j, b_N^i) := \int_K \mathbf{grad} b_N^j|_K \cdot \mathbf{grad} b_N^i|_K dx , \quad x^i, x^j \text{ vertices of } K . \quad (3.2.8)$$

Use analytic representation for $b_N^i|_K$:

if $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$ vertices of K , $\lambda_i := b_N^i|_K$, $\mathbf{a}^i = \mathbf{x}^i$
($i \leftrightarrow$ local vertex number, $j \leftrightarrow$ global node number)

Fig. 70



The functions $\lambda_1, \lambda_2, \lambda_3$ on the triangle K are also known as **barycentric coordinate functions**.

$$\mathbf{a}^3 = (a_1^3, a_2^3)^T$$

$$\lambda_1(\mathbf{x}) = \frac{1}{2|K|} (\mathbf{x} - \mathbf{a}^2) \cdot \begin{pmatrix} a_2^2 - a_2^3 \\ a_1^3 - a_1^2 \end{pmatrix} = -\frac{|e_1|}{2|K|} (\mathbf{x} - \mathbf{a}^2) \cdot \mathbf{n}^1,$$

$$\lambda_2(\mathbf{x}) = \frac{1}{2|K|} (\mathbf{x} - \mathbf{a}^3) \cdot \begin{pmatrix} a_2^3 - a_2^1 \\ a_1^1 - a_1^3 \end{pmatrix} = -\frac{|e_2|}{2|K|} (\mathbf{x} - \mathbf{a}^3) \cdot \mathbf{n}^2,$$

$$\lambda_3(\mathbf{x}) = \frac{1}{2|K|} (\mathbf{x} - \mathbf{a}^1) \cdot \begin{pmatrix} a_2^1 - a_2^2 \\ a_1^2 - a_1^1 \end{pmatrix} = -\frac{|e_3|}{2|K|} (\mathbf{x} - \mathbf{a}^1) \cdot \mathbf{n}^3.$$

(e_i = edge opposite vertex \mathbf{a}^i , see Figure for numbering scheme \triangleright)

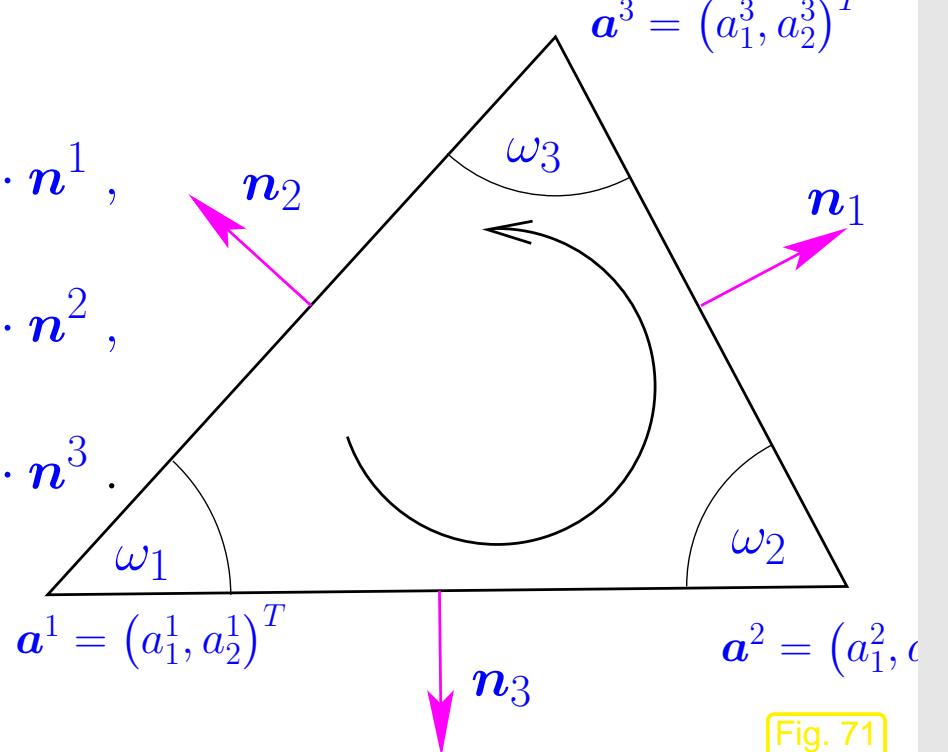


Fig. 71

From the distance formula for a point w.r.t to a line given in **Hesse normal form**:

$$(\mathbf{a}^i - \mathbf{a}^j) \cdot \mathbf{n}_i = \text{dist}(\mathbf{a}^i; e_i) = h_i \quad (h_i \hat{=} \text{height}) \text{ and } 2|K| = |e_i| h_i \Rightarrow \lambda_i(\mathbf{a}^i) = 1.$$

This shows that the λ_i really provide the restrictions of p.w. linear nodal basis functions of $\mathcal{S}_1^0(\mathcal{M})$ to triangle K , because they are clearly (affine) linear as comply with (3.2.1).



$$\mathbf{grad} \lambda_1 = \frac{1}{2|K|} \begin{pmatrix} a_2^2 - a_2^3 \\ a_2^2 - a_1^3 \end{pmatrix}, \quad \mathbf{grad} \lambda_2 = \frac{1}{2|K|} \begin{pmatrix} a_2^3 - a_2^1 \\ a_1^3 - a_1^1 \end{pmatrix}, \quad \mathbf{grad} \lambda_3 = \frac{1}{2|K|} \begin{pmatrix} a_2^1 - a_2^2 \\ a_1^1 - a_1^2 \end{pmatrix}.$$



$$\begin{aligned} \left(\int_K \mathbf{grad} \lambda_i \cdot \mathbf{grad} \lambda_j \, dx \right)_{i,j=1}^3 &= \text{element (stiffness) matrix } \mathbf{A}_K \\ &= \frac{1}{2} \begin{pmatrix} \cot \omega_3 + \cot \omega_2 & -\cot \omega_3 & -\cot \omega_2 \\ -\cot \omega_3 & \cot \omega_3 + \cot \omega_1 & -\cot \omega_1 \\ -\cot \omega_2 & -\cot \omega_1 & \cot \omega_2 + \cot \omega_1 \end{pmatrix}. \quad (3.2.9) \end{aligned}$$

The local numbering and naming conventions are displayed in Fig. 108.

Derivation of (3.2.9), see also [15, Lemma 3.47]: obviously, because the gradients $\mathbf{grad} \lambda_i$ are constant on K ,

$$a(\lambda_i, \lambda_j) = \int_K \mathbf{grad} \lambda_i \cdot \mathbf{grad} \lambda_j \, dx = \frac{1}{4|K|} |e_i| |e_j| \mathbf{n}_i \cdot \mathbf{n}_j.$$

Then use:

- $\mathbf{n}_i \cdot \mathbf{n}_j = \cos(\pi - \omega_k) = -\cos \omega_k, \quad (i \neq j)$
- $|K| = \frac{1}{2} |e_i| |e_j| \sin \omega_k, \quad (i \neq j).$

Case $i = j$ employs a trick: $\sum_{i=1}^3 \lambda_i = 1 \Rightarrow \sum_{i=1}^3 \mathbf{a}(\lambda_i, \lambda_j) = 0.$

Remark 3.2.10 (Scaling of entries of element matrix for $-\Delta$).

(3.2.9): \mathbf{A}_K does not depend on the “size” of triangle K !
 (more precisely, element matrices are equal for *similar* triangles)

This can be seen by the following reasoning:

- Obviously translation and rotation of K does not change. \mathbf{A}_K
- *Scaling* of K by a factor $\rho > 0$ has the following effect that
 - the area $|K|$ is scaled by ρ^2 ,

- the gradients $\text{grad } \lambda_i$ are scaled by ρ^{-1} (the barycentric coordinate functions λ_i become steeper when the triangle shrinks in size.).

Both effects just offset in a_K from (3.2.8) such that A_K remains invariant under scaling.



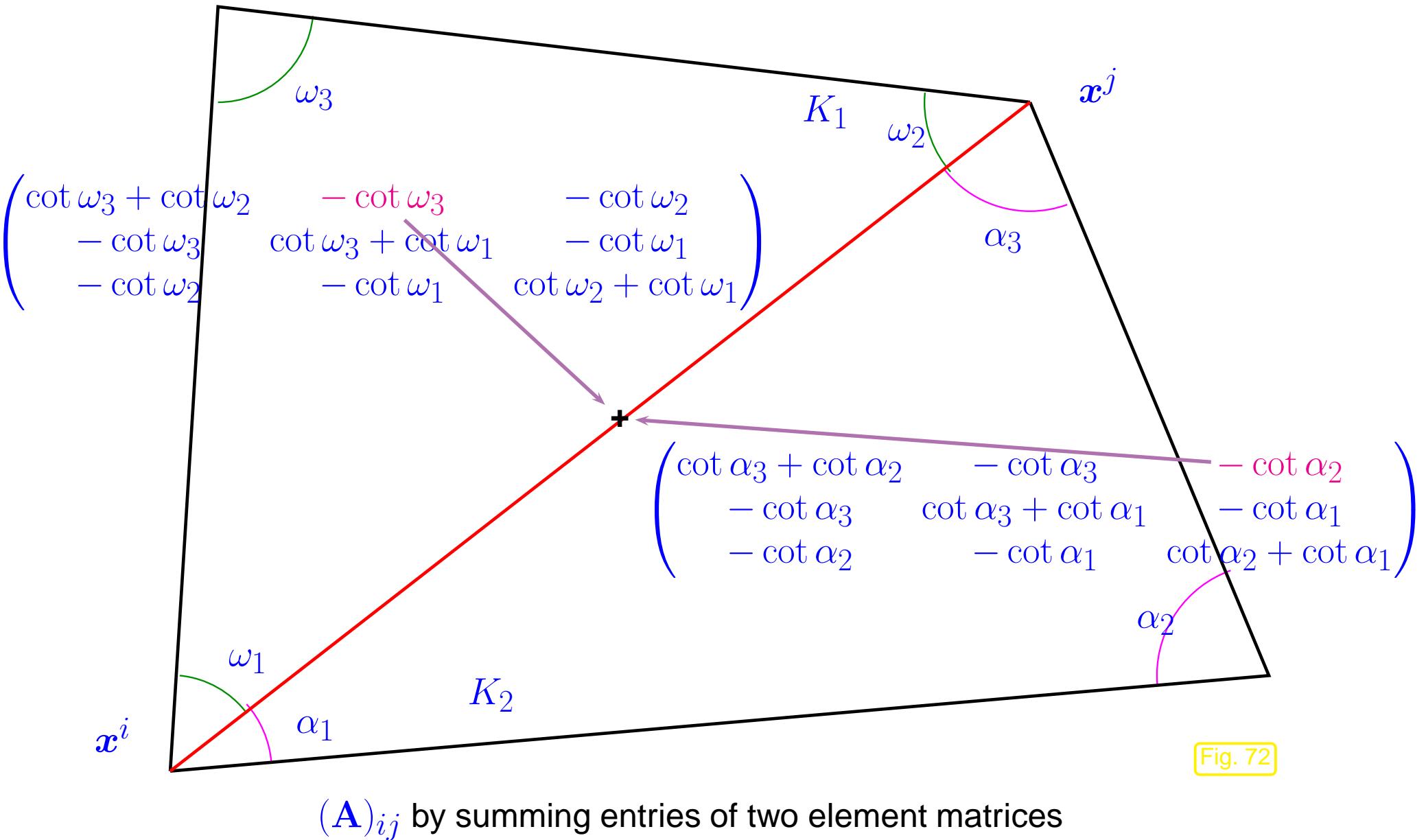
“Assembly” of $(A)_{ij}$ starts from the sum

$$(A)_{ij} = \int_{K_1} \text{grad } b_{N|K_1}^j \cdot \text{grad } b_{N|K_1}^i dx + \int_{K_2} \text{grad } b_{N|K_2}^j \cdot \text{grad } b_{N|K_2}^i dx .$$

- $(A)_{ij}$ can be obtained by summing respective^(*) entries of the elements matrices of the elements adjacent to the edge connecting x^i and x^j

(*): watch correspondence of local and global vertex numbers !

3.2



“Assembly” of diagonal entry $(\mathbf{A})_{ii}$: summing corresponding diagonal entries of element matrices belonging to triangles adjacent to node x^i .

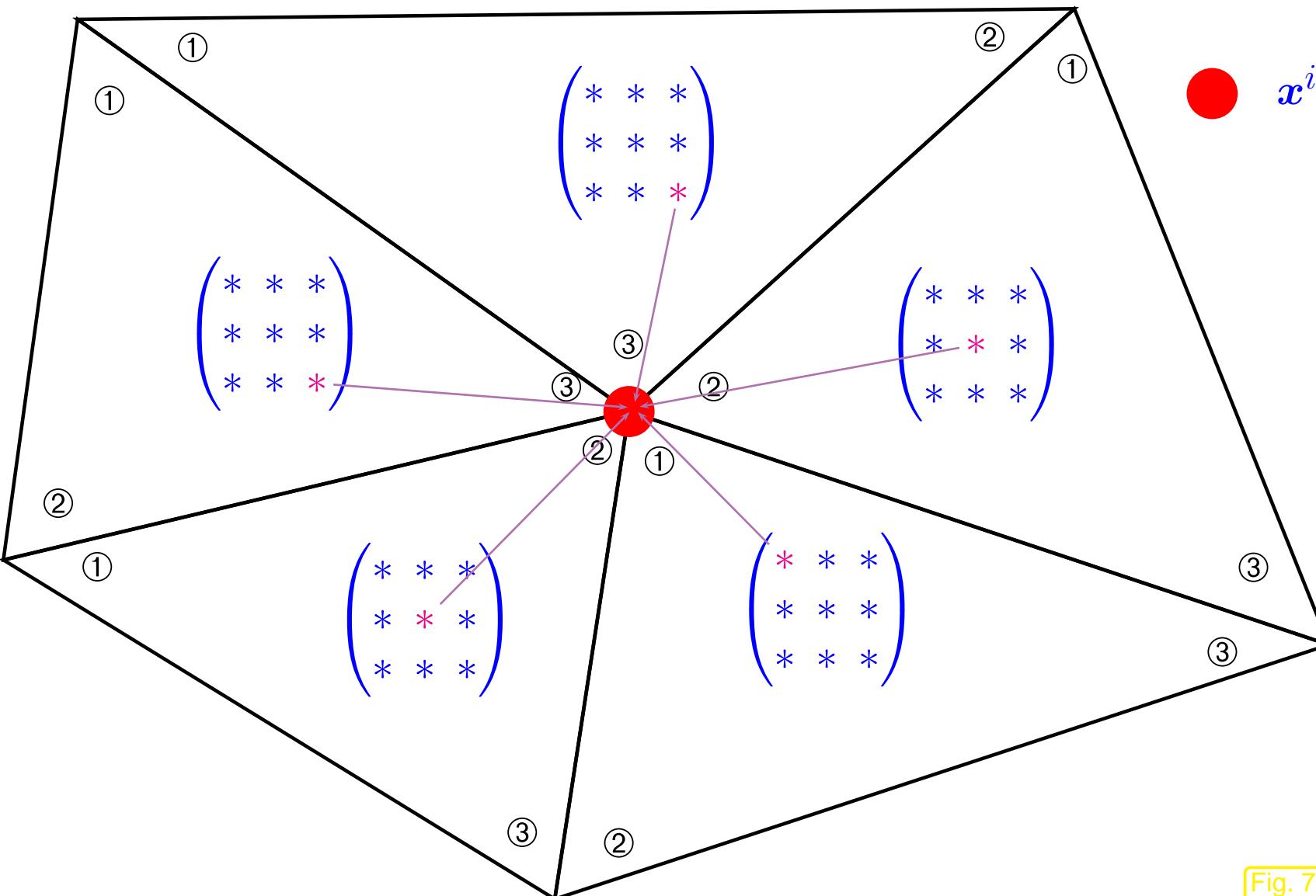


Fig. 73

$(A)_{ii}$ by summing diagonal entries of element matrices of adjacent triangles

3.2.6 Computation of right hand side vector

We consider the linear form (right hand side of linear variational problem), see (2.3.3), (3.0.1):

$$\ell(v) := \int_{\Omega} f(\boldsymbol{x}) v(\boldsymbol{x}) \, d\boldsymbol{x} , \quad v \in H^1(\Omega) , \quad f \in L^2(\Omega) .$$

Recall formula for right hand side vector

$$(\vec{\varphi})_j = \ell(b_N^j) = \int_{\Omega} f(\boldsymbol{x}) b_N^j(\boldsymbol{x}) \, d\boldsymbol{x} , \quad j = 1, \dots, N . \quad (3.2.11)$$

Idea: “Assembly”

$$(\vec{\varphi})_j = \sum_{l=1}^{N_j} \int_{K_l} f(\mathbf{x}) b_N^j |_{K_l} (\mathbf{x}) d\mathbf{x},$$

where K_1, \dots, K_{N_j} $\hat{=}$ triangles adjacent to node \mathbf{x}^j .

(Integration confined to $\text{supp}(b_N^j)$!)

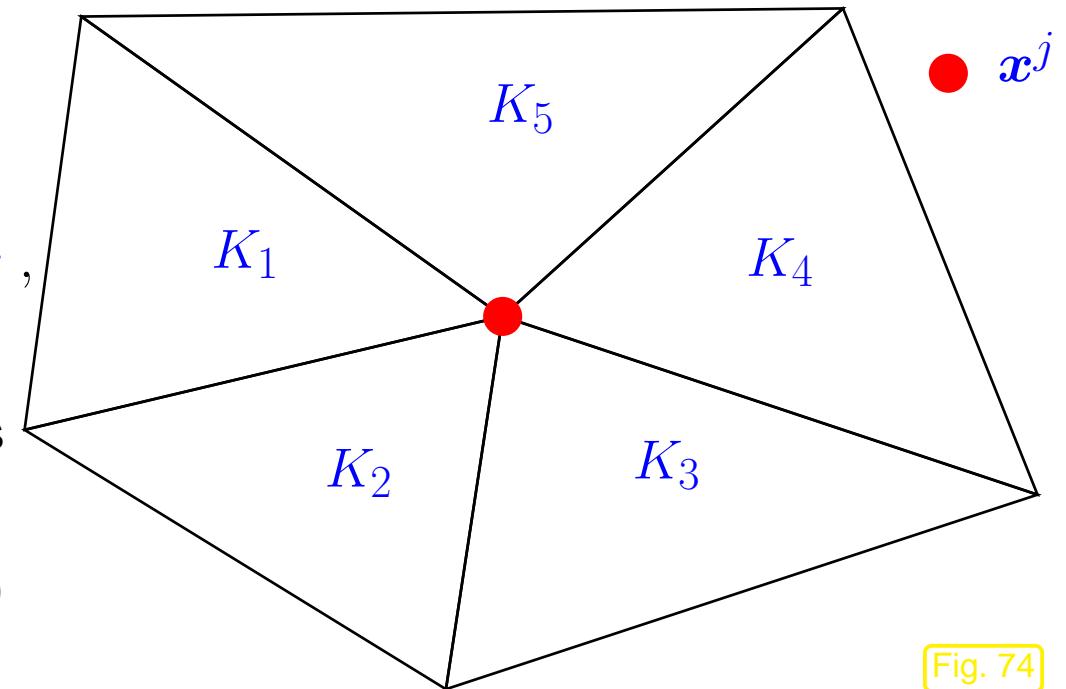


Fig. 74

► Zero in on single triangle $K \in \mathcal{M}$:

$$\ell_K(b_N^j) := \int_K f(\mathbf{x}) b_N^j |_K (\mathbf{x}) d\mathbf{x}, \quad \mathbf{x}^j \text{ vertex of } K. \quad (3.2.12)$$

Rem. 1.5.3: $f : \Omega \mapsto \mathbb{R}$ given in **procedural form**

```
function y = f(x)
```

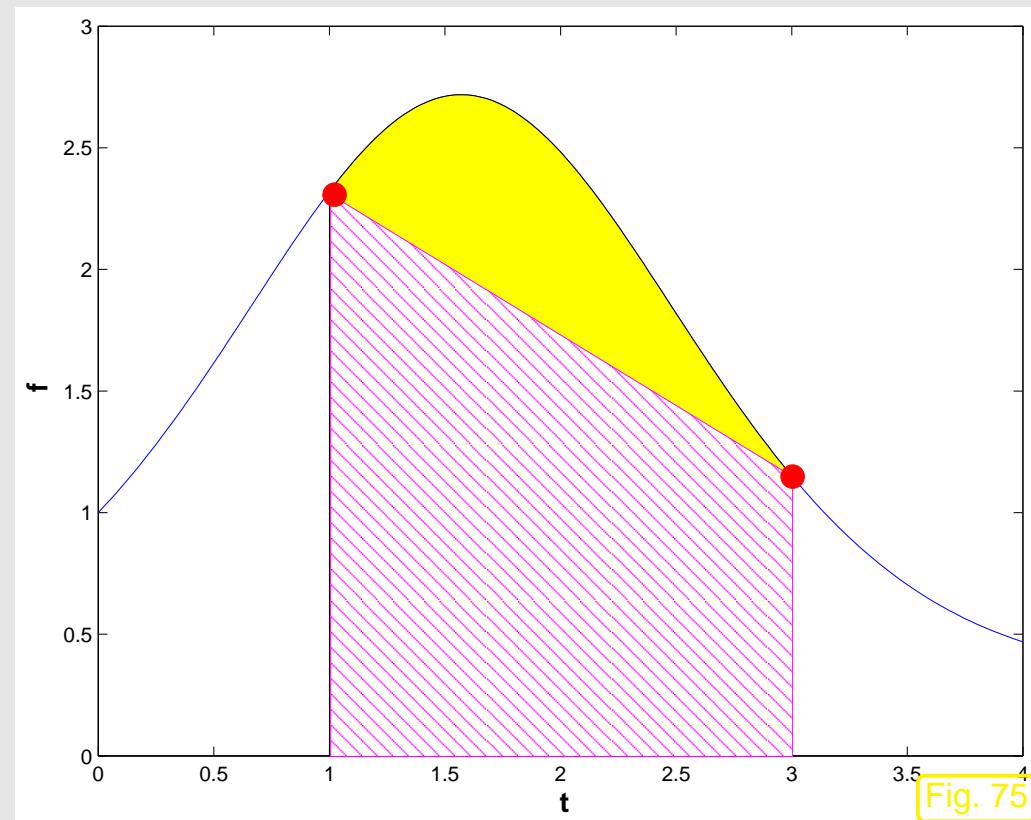
► Mandatory: use of **numerical quadrature** for approximate evaluation of $\ell_K(b_N^j)$, cf. (1.5.55).

1D setting of Sect. 1.5.1.2: use of composite quadrature rules based on low Gauss/Newton-Cotes quadrature formulas on the cells $[x_{j-1}, x_j]$ of the grid, e.g. composite trapezoidal rule (1.5.55).

What is the 2D counterpart of the composite trapezoidal rule ?

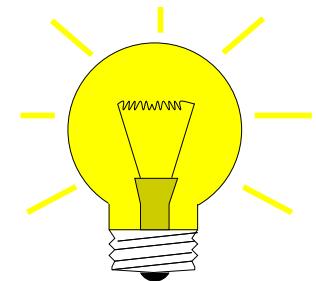
Recall:

trapezoidal rule [14, Eq. 11.4.2] integrates linear interpolant of integrand based on endpoint values



Idea:

2D trapezoidal rule



for triangle K with vertices $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$

$$\int_K f(\mathbf{x}) d\mathbf{x} \approx \frac{|K|}{3} (f(\mathbf{a}^1) + f(\mathbf{a}^2) + f(\mathbf{a}^3)) . \quad (3.2.13)$$

$\hat{=}$ integration of linear interpolant $\sum_{i=1}^3 f(\mathbf{a}^i) \lambda_i$ of f .

► element (load) vector: $\vec{\varphi}_K := \left(\ell_K(b_N^{j(i)}) \right)_{i=1}^3 = \frac{|K|}{3} \begin{pmatrix} f(\mathbf{a}^1) \\ f(\mathbf{a}^2) \\ f(\mathbf{a}^3) \end{pmatrix} ,$

where $\mathbf{x}^{j(i)} = \mathbf{a}^i, i = 1, 2, 3$ (global node number \leftrightarrow local vertex number).

As above in Fig. 73: “Assembly” of $(\vec{\varphi})_j$ by summing up contributions from element vectors of triangles adjacent to \mathbf{x}^j .

$$(\vec{\varphi})_j = \sum_{l=1}^{N_j} \ell_{K_l}(b_N^j|_{K_l}) = \sum_{l=1}^{N_j} (\vec{\varphi}_K)_{i(l,j)} = f(\mathbf{x}^j) \cdot \frac{1}{3} \sum_{l=1}^{N_j} |K_l| , \quad (3.2.14)$$

where $i(l, j)$ is the local vertex index of the node \mathbf{x}^j (global index j) in the triangle K_l .

3.2

p. 303

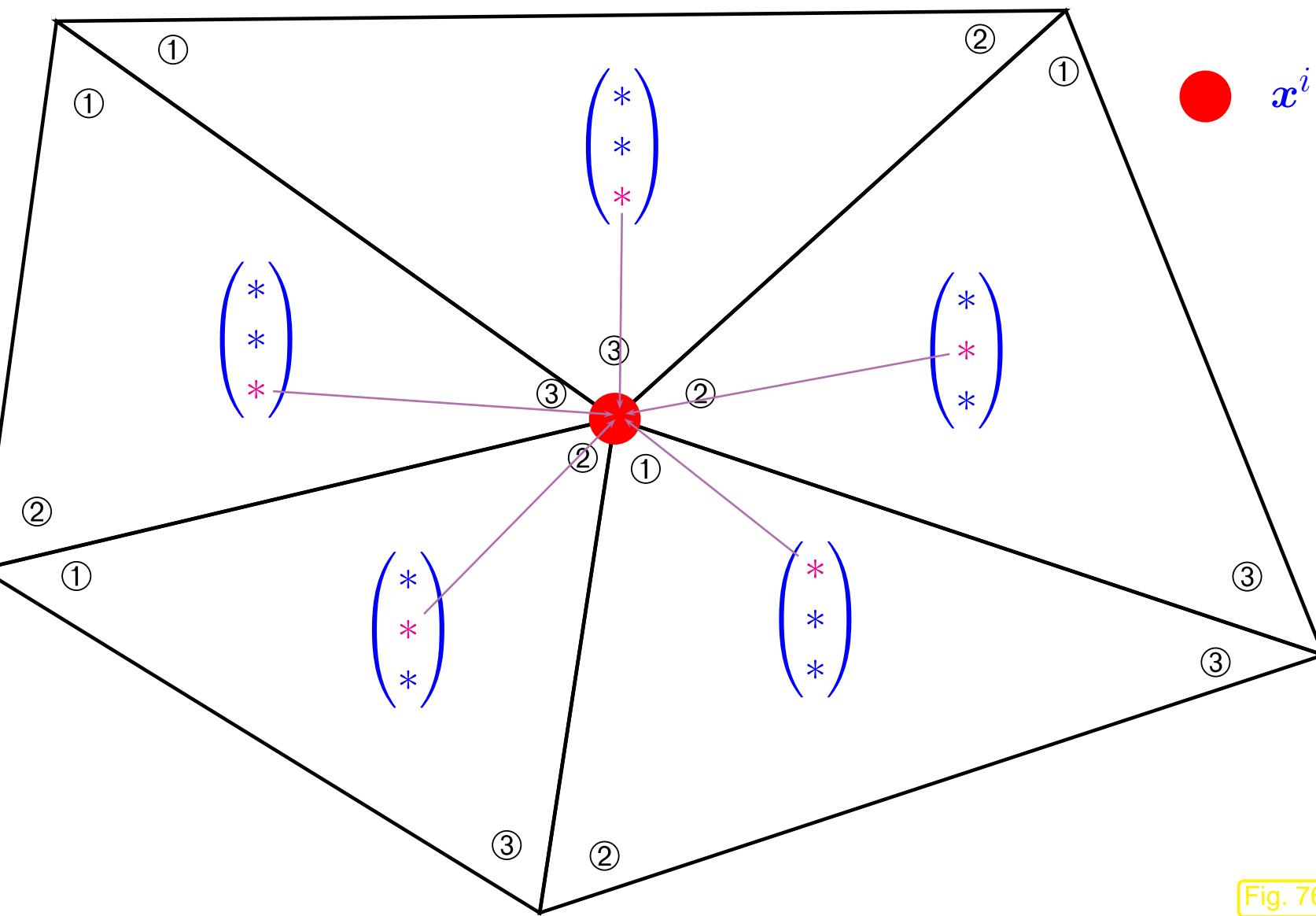


Fig. 76

3.3 Building blocks of general FEM

The previous section explored the details of a simple finite element discretization of 2nd-order elliptic variational problems. Yet, it already introduced *key features and components* that distinguish the finite element approach to the discretization of linear boundary value problems for partial differential equations:

- a focus on the **variational formulation** of a boundary value problem → Sect. 2.8,
- a partitioning of the computational domain Ω by means of a **mesh** \mathcal{M} (→ Sect. 3.2.1)
- the use of Galerkin trial and test spaces based on **piecewise polynomials** w.r.t. \mathcal{M} (→ Sect. 3.2.2),
- the use of **locally supported** basis functions for the assembly of the resulting linear system of equations (→ Sect. 3.2.3).

In this section a more abstract point of view is adopted and the components of a finite element method for scalar 2nd-order elliptic boundary value problems will be discussed in greater generality. However, prior perusal of Sect. 3.2 is strongly recommended.

3.3.1 Meshes

First main ingredient of FEM: triangulation/mesh of $\Omega \rightarrow$ Sect. 3.2.1

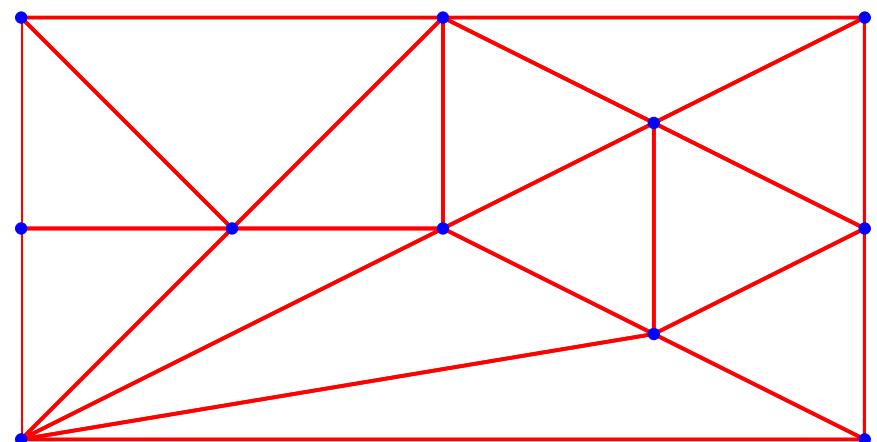
Definition 3.3.1. A *mesh* (or *triangulation*) of $\Omega \subset \mathbb{R}^d$ is a finite collection $\{K_i\}_{i=1}^M$, $M \in \mathbb{N}$, of open non-degenerate (curvilinear) polygons ($d = 2$)/polyhedra ($d = 3$) such that

- (A) $\overline{\Omega} = \bigcup \{\overline{K}_i, i = 1, \dots, M\},$
- (B) $K_i \cap K_j = \emptyset \Leftrightarrow i \neq j,$
- (C) for all $i, j \in \{1, \dots, M\}$, $i \neq j$, the intersection $\overline{K}_i \cap \overline{K}_j$ is either empty or a vertex, edge, or face of both K_i and K_j .

► “vertex”, “edge”, “face” of polygon/polyhedron: \rightarrow geometric intuition

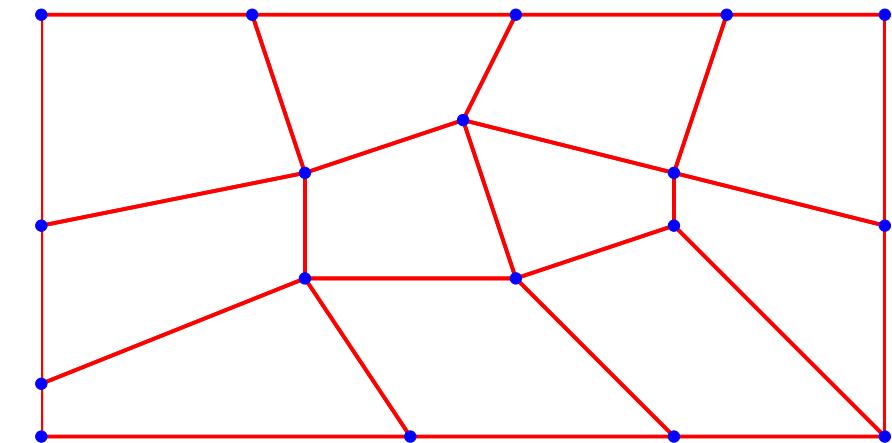
Terminology: Given mesh $\mathcal{M} := \{K_i\}_{i=1}^M$: K_i called **cell** or **element**.
Vertices of a mesh \rightarrow **nodes** (set $\mathcal{V}(\mathcal{M})$)

Types of meshes:



Triangular mesh in 2D

Fig. 77



Quadrilateral mesh in 2D

Fig. 78

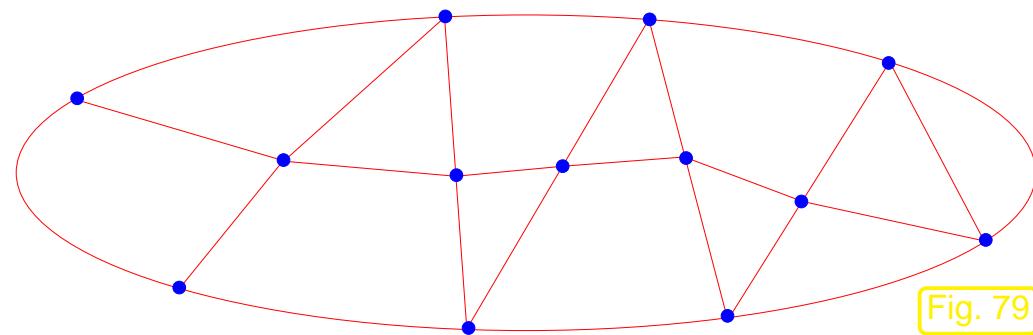
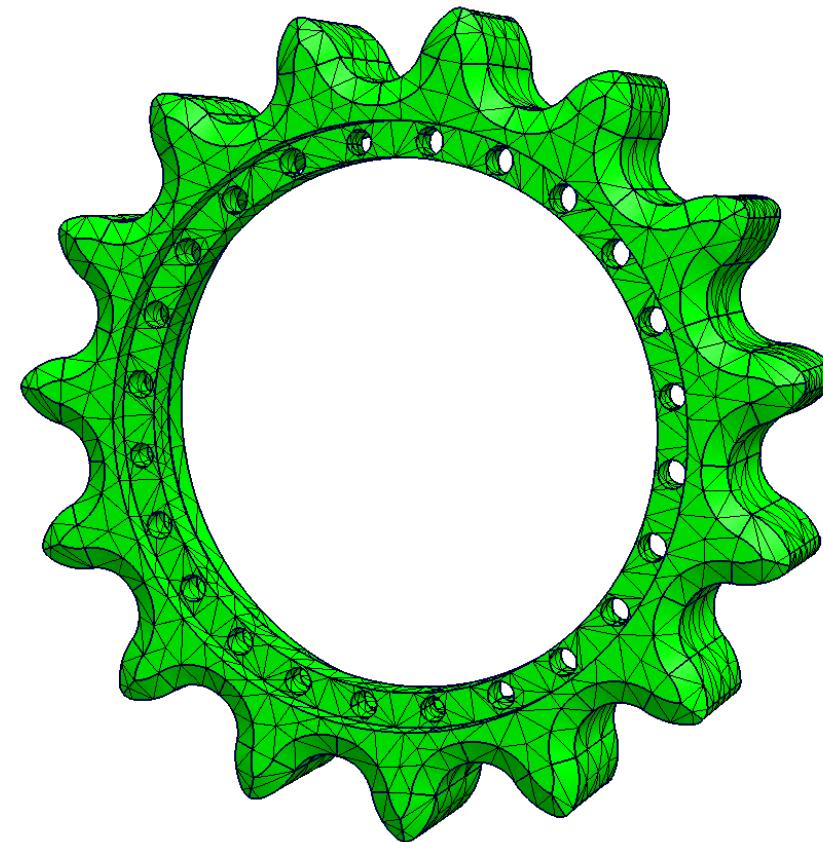
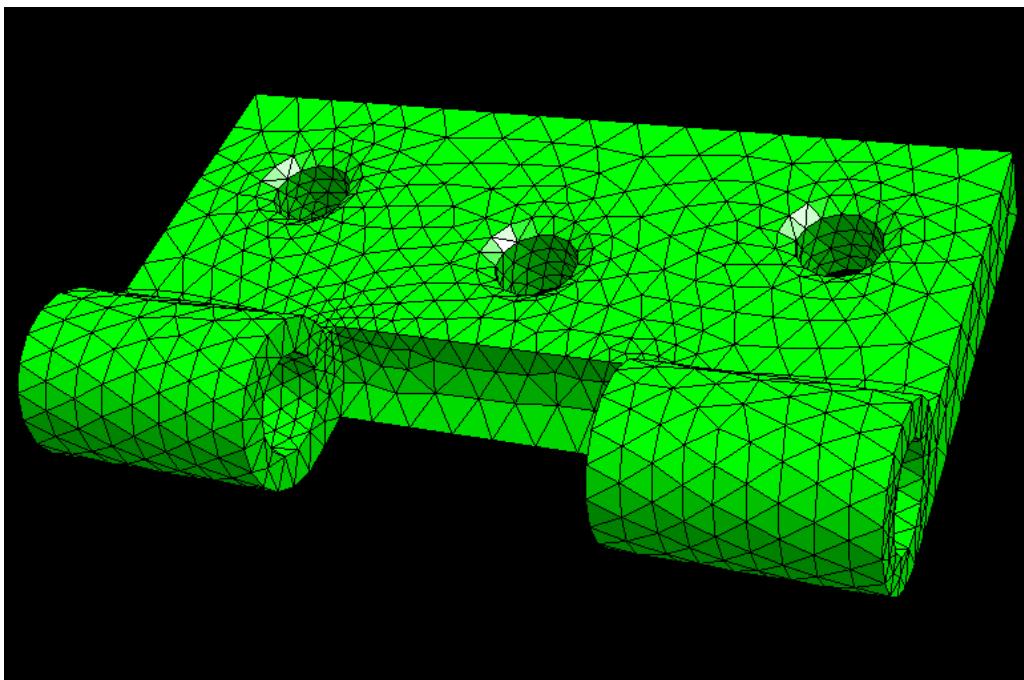


Fig. 79

- ▷ 2D **hybrid** mesh comprising
 - triangles
 - quadrilaterals
 - curvilinear cells (at $\partial\Omega$)

Tetrahedral meshes in 3D (created with NETGEN):

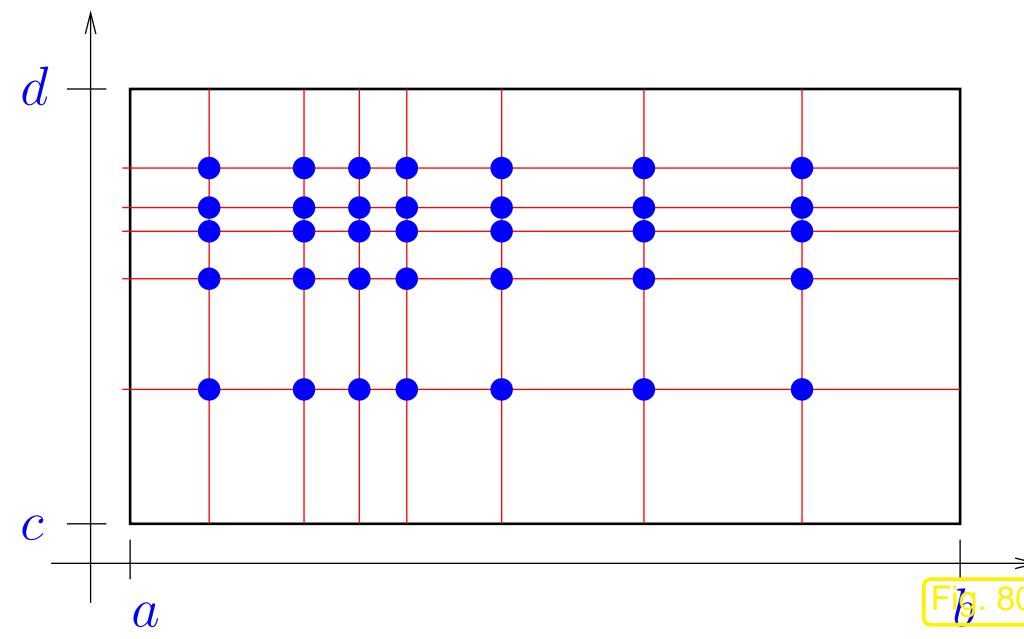


Tensor product mesh = grid



in 2D: $a = x_0 < x_1 < \dots < x_n = b$,
 $c = y_0 < y_1 < \dots < y_m = d$.

► $\mathcal{M} = \{]x_{i-1}, x_i[\times]y_{j-1}, y_j[: \quad (3.3.2)$
 $1 \leq i \leq n, 1 \leq j \leq m\}$.



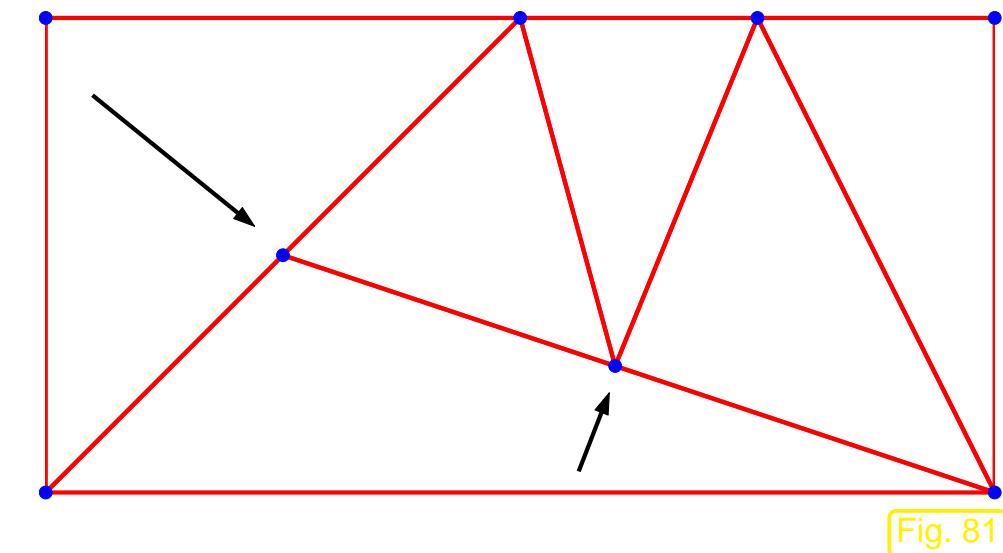
☞ Restricted to tensor product domains

If (C) does not hold

- Triangular non-conforming mesh
(with hanging nodes)

$\overline{K}_i \cap \overline{K}_j$ is only part of an edge/face for at most one of the adjacent cells.

(However, conforming if degenerate quadrilaterals admitted)



Terminology:

Simplicial mesh = triangular mesh in 2D
tetrahedral mesh in 3D

3.3.2 Polynomials

Second main ingredient of FEM:

In FEM: Galerkin trial/test space comprise *locally polynomial* functions on Ω

Clear: polynomials of **degree** $\leq p$, $p \in \mathbb{N}_0$, in 1D (**univariate** polyomials), see (1.5.17)

$$\mathcal{P}_p(\mathbb{R}) := \{x \mapsto c_0 + c_1 x + c_2 x^2 + \dots + c_p x^p\}.$$

In higher dimensions this concept allows various generalizations, one given in the following definition, one given in Def. 3.3.7.

Definition 3.3.3 (Multivariate polynomials).

Space of **multivariate (d-variate) polynomials** of (total) **degree** $p \in \mathbb{N}_0$:

$$\mathcal{P}_p(\mathbb{R}^d) := \{x \in \mathbb{R}^d \mapsto \sum_{\alpha \in \mathbb{N}_0^d, |\alpha| \leq p} c_\alpha x^\alpha, c_\alpha \in \mathbb{R}\}.$$

Def. 3.3.3 relies on **multi-index notation**:

$$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d): \quad \mathbf{x}^{\boldsymbol{\alpha}} := x_1^{\alpha_1} \cdots x_d^{\alpha_d}, \quad (3.3.4)$$

$$|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + \cdots + \alpha_d. \quad (3.3.5)$$

Special case:

$$d = 2: \quad \mathcal{P}_p(\mathbb{R}^2) = \left\{ \sum_{\substack{\alpha_1, \alpha_2 \geq 0 \\ \alpha_1 + \alpha_2 \leq p}} c_{\alpha_1, \alpha_2} x_1^{\alpha_1} x_2^{\alpha_2}, c_{\alpha_1, \alpha_2} \in \mathbb{R} \right\}.$$

Example:

$$\mathcal{P}_2(\mathbb{R}^2) = \text{Span} \left\{ 1, x_1, x_2, x_1^2, x_2^2, x_1 x_2 \right\}$$

Lemma 3.3.6 (Dimension of spaces of polynomials).

$$\dim \mathcal{P}_p(\mathbb{R}^d) = \binom{d+p}{p} \quad \text{for all } p \in \mathbb{N}_0, d \in \mathbb{N}$$

Proof. Distribute p “powers” to the d independent variables or discard them $\triangleright d + 1$ bins.

Combinatorial model: number of different linear arrangements of p identical items and d separators
 $= \binom{d+p}{p}$. □

Leading order

$$\dim \mathcal{P}_p(\mathbb{R}^d) = O(p^d)$$

Definition 3.3.7 (Tensor product polynomials).

Space of *tensor product polynomials* of degree $p \in \mathbb{N}$ in each coordinate direction

$$\mathcal{Q}_p(\mathbb{R}^d) := \{\mathbf{x} \mapsto p_1(x_1) \cdots \cdots p_d(x_d), p_i \in \mathcal{P}_p(\mathbb{R}), i = 1, \dots, d\} .$$

Example:

$$\mathcal{Q}_2(\mathbb{R}^2) = \text{Span} \left\{ 1, x_1, x_2, x_1 x_2, x_1^2, x_1^2 x_2, x_1^2 x_2^2, x_1 x_2^2, x_2^2 \right\}$$

3.3

p. 312

Lemma 3.3.8 (Dimension of spaces of tensor product polynomials).

$$\dim \mathcal{Q}_p(\mathbb{R}^d) = (p+1)^d \quad \text{for all } p \in \mathbb{N}_0, d \in \mathbb{N}$$

Terminology: $\mathcal{P}_p(\mathbb{R}^d)/\mathcal{Q}_p(\mathbb{R}^d)$ = complete spaces of polynomials/tensor product polynomials

3.3.3 Basis functions

Third main ingredient of FEM:

locally supported basis functions

(see Sect. 3.1 for role of bases in Galerkin discretization)

Basis functions b_N^1, \dots, b_N^N for a finite element trial/test space $V_{0,N}$ built on a mesh \mathcal{M} satisfy:

- (a) $\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\}$ is basis of $V_{0,N} \quad \Rightarrow \quad N = \dim V_{0,N}$,
- (b) each b_N^i is associated with a single cell/edge/face/vertex of \mathcal{M} ,
- (c) $\text{supp}(b_N^i) = \bigcup \{\bar{K} : K \in \mathcal{M}, p \subset \bar{K}\}$, if b_N^i associated with cell/edge/face/vertex p .

Finite element terminology: b_N^i = global shape functions/global basis functions

Mesh \mathcal{M} + global shape functions \rightarrow complete description of finite element space

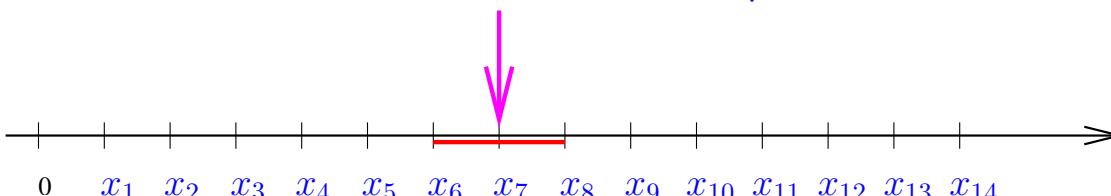
Example 3.3.9 (Supports of global shape functions in 1D). \rightarrow Sect. 1.5.1.2

- $\Omega =]a, b[\hat{=} \text{interval}$
- Equidistant mesh

$$\mathcal{M} := \{]x_{j-1}, x_j[, j = 1, \dots, M\},$$

$$x_j := a + h j, h := (b - a)/M, M \in \mathbb{N}.$$

Support (\rightarrow Def. 1.5.53) of global shape function
associated with x_7



◇

3.3

Example 3.3.10 (Supports of global shape functions on triangular mesh).

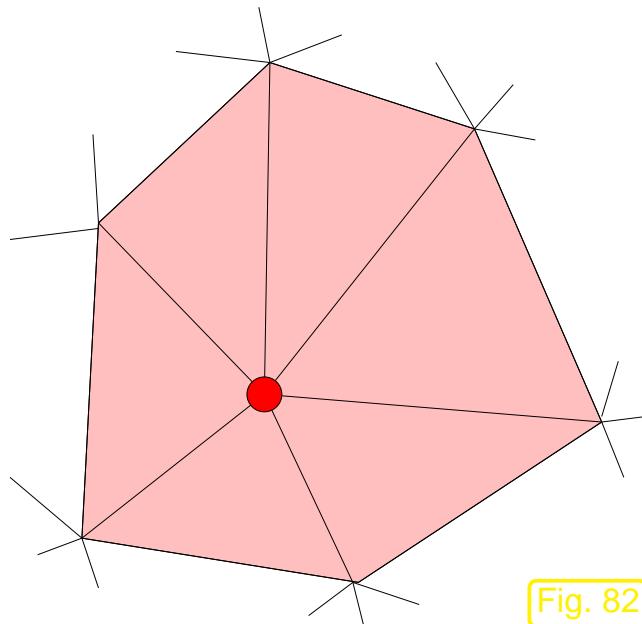


Fig. 82

Support of node-associated
basis function

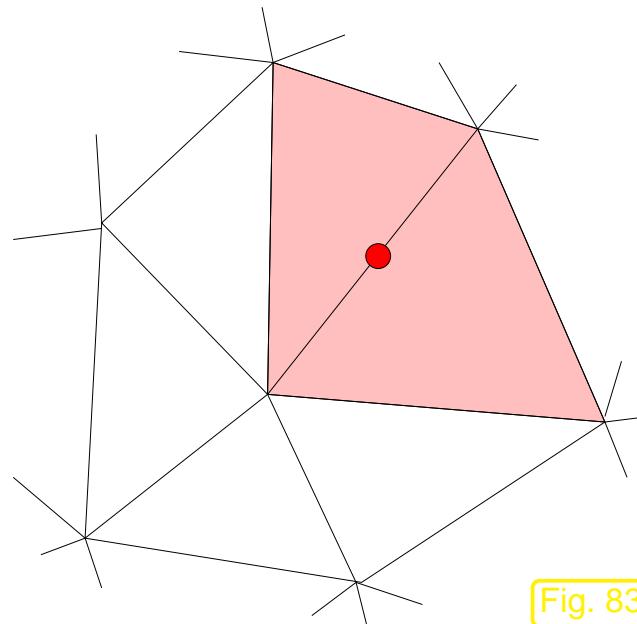


Fig. 83

Support of edge-associated
basis function

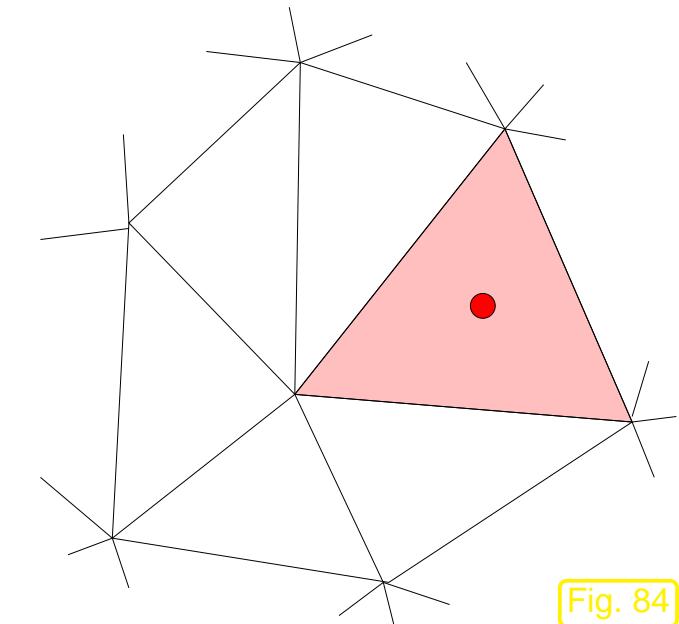


Fig. 84

Support of cell-associated basis
function

◇

Requirement (c) implies that

global finite element basis functions are **locally supported**.

What is the rationale for this requirement ?

Consider a generic bilinear form a arising from a linear scalar 2nd-order elliptic BVP, see (3.2.4): it involves integration over $\Omega/\partial\Omega$ of products of (derivatives of) basis functions. Thus the integrand for $a(b_N^j, b_N^i)$ vanishes outside the overlap of the supports of b_N^j and b_N^i .

- Galerkin matrix $\mathbf{A} \in \mathbb{R}^{N,N}$ with $(\mathbf{A})_{ij} := a(b_N^j, b_N^i)$, $i, j = 1, \dots, N$ satisfies

$$a_{ij} \neq 0 \quad \text{only if}$$

b_N^i and b_N^j associated with
vertices/faces/edges(cells) adjacent to common
cell



Finite element stiffness matrices are **sparse** (\rightarrow Notion 3.2.6)

Global shape functions

Restriction to element

local shape functions

(3.3.11)

3.3

Definition 3.3.12 (Local shape functions).

Given finite element function space on mesh \mathcal{M} with global shape functions b_N^i , $i = 1, \dots, N$:

$$\{b_{N|K}^j, K \subset \text{supp}(b_N^j)\} = \text{set of local shape functions on } K \in \mathcal{M}.$$

- Local shape functions b_K^1, \dots, b_K^Q , $Q = Q(K) \in \mathbb{N}$ also associated with vertices/edges-/faces/interior of K

Example 3.3.13 (Local shape functions for $\mathcal{S}_1^0(\mathcal{M})$ in 2D). → Sect. 3.2.3

Global basis function for $\mathcal{S}_1^0(\mathcal{M})$



On “unit triangle” K with vertices

$$\mathbf{a}^1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{a}^2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{a}^3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$b_K^1(\mathbf{x}) = 1 - x_1 - x_2 ,$$

$$b_K^2(\mathbf{x}) = x_1 ,$$

$$b_K^3(\mathbf{x}) = x_2 .$$

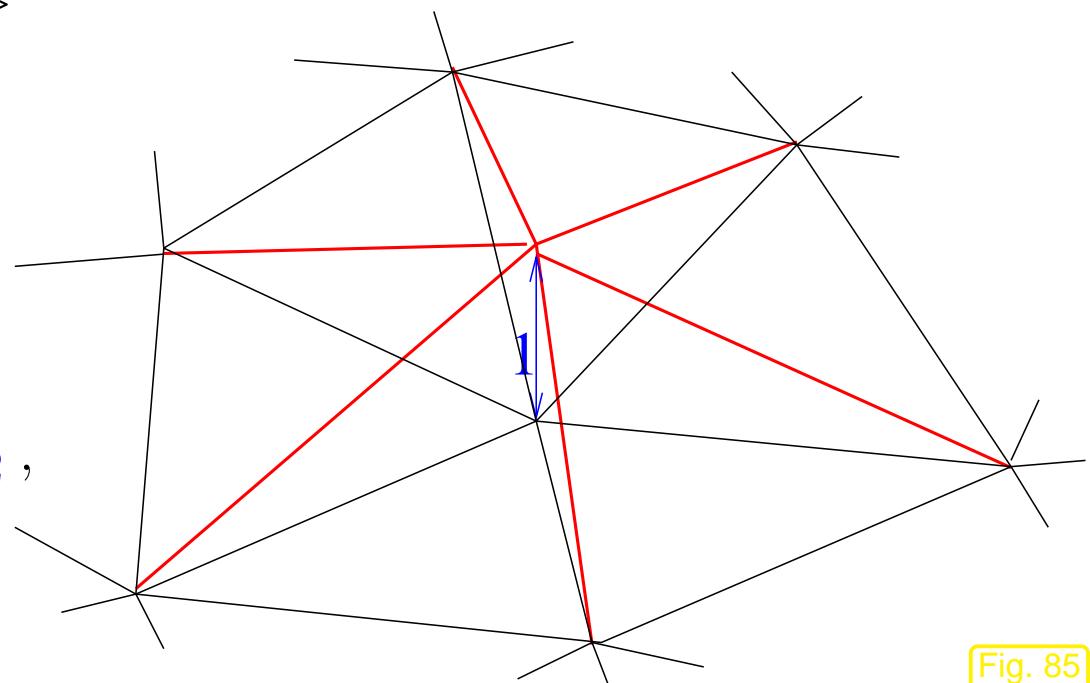


Fig. 85

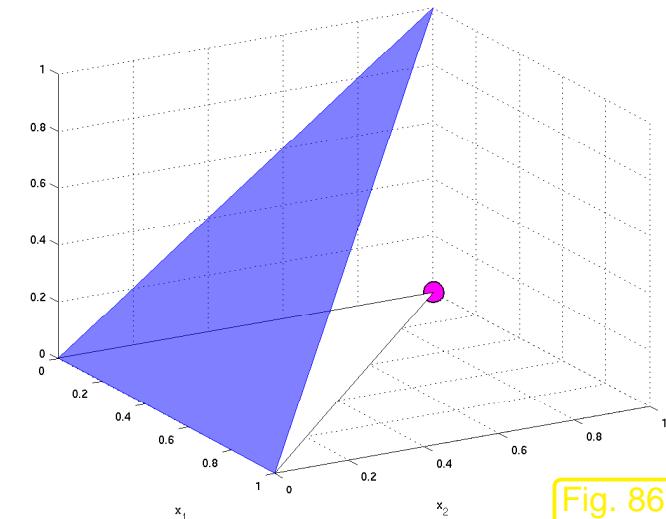
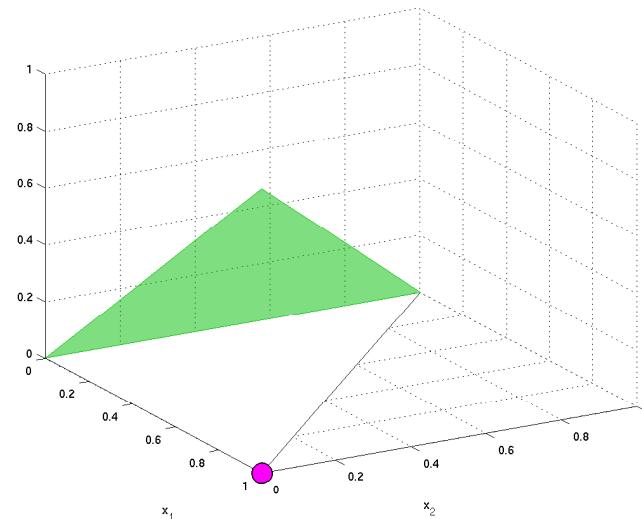
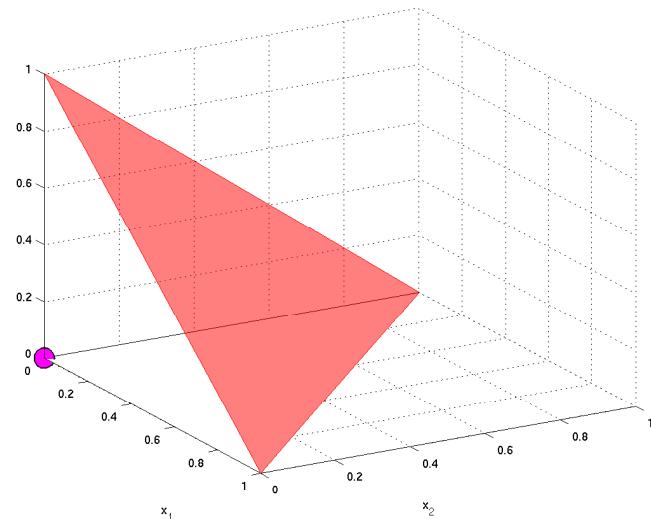


Fig. 86

These are the barycentric coordinate functions $\lambda_1, \lambda_2, \lambda_3$ introduced in Sect. 3.2.5



3.4 Lagrangian FEM

Taken for granted: finite element mesh \mathcal{M} according to Def. 3.3.1.

Goal: construction of finite element spaces and global shape functions of higher polynomials degrees.

Lagrangian finite element spaces provide spaces $V_{0,N}$ of \mathcal{M} -piecewise polynomials that fulfill

$$V_{N,0} \subset C^0(\Omega) \quad \text{Thm. 2.2.17} \implies V_{N,0} \subset H^1(\Omega).$$

Notation:
(Lagrangian FE spaces)

$S_p^0(\mathcal{M})$

continuous functions, cf. $C^0(\Omega)$

locally polynomials of degree p , e.g. $\mathcal{P}_p(\mathbb{R}^d)$

3.4.1 Simplicial Lagrangian FEM

\mathcal{M} : Simplicial mesh, consisting of triangles in 2D, tetrahedra in 3D.

Now we generalize $\mathcal{S}_1^0(\mathcal{M})/\mathcal{S}_{1,0}^0(\mathcal{M})$ from Sect. 3.2 to higher polynomial degree $p \in \mathbb{N}_0$.

Definition 3.4.1 (Simplicial Lagrangian finite element spaces).

Space of *p-th degree Lagrangian finite element* functions on simplicial mesh \mathcal{M}

$$\mathcal{S}_p^0(\mathcal{M}) := \{v \in C^0(\bar{\Omega}): v|_K \in \mathcal{P}_p(K) \quad \forall K \in \mathcal{M}\} .$$

Def. 3.4.1 merely describes the space of trial/test functions used in a Lagrangian finite element method on a Simplicial mesh. A crucial ingredient is still missing (\rightarrow Sect. 3.3.3): the global shape functions still need to be specified. This is done by generalizing (3.2.1) based on sets of special *interpolation nodes*.

Example 3.4.2 (Triangular quadratic Lagrangian finite elements).

interpolation nodes

$$\mathcal{N} := \mathcal{V}(\mathcal{M}) \cup \{\text{midpoints of edges}\} ,$$

$$\mathcal{N} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\} .$$

Nodal basis functions b_N^j , $j = 1, \dots, N$ defined
by, cf. (3.2.1)

$$b_N^j(\mathbf{p}_i) = \begin{cases} 1 & \text{if } i = j , \\ 0 & \text{else.} \end{cases}$$

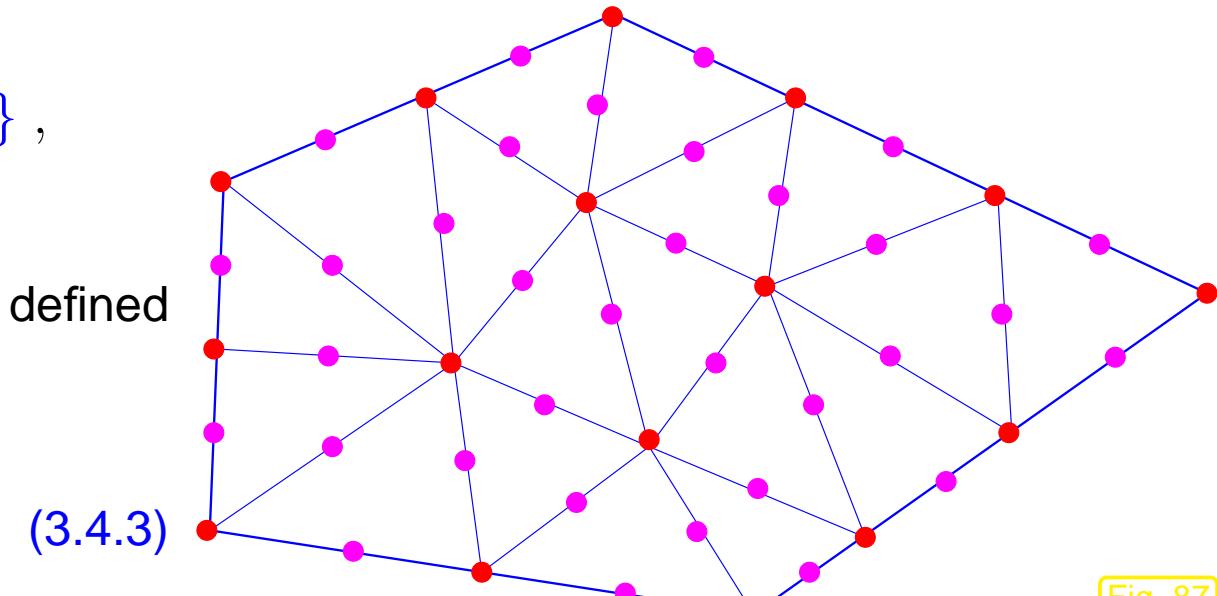


Fig. 87

A “definition” like (3.4.3) is cheap, but it may be pointless, in case no such functions b_N^j exist. To establish their existence, we first study the case of a single triangle K .

We have to show that there is a basis of $\mathcal{P}_2(\mathbb{R}^2)$ that satisfies (3.4.3) in the case of a mesh consisting of a single triangle $\mathcal{M} = \{K\}$.

A first simple consistency check: does the number of interpolation nodes $\#\mathcal{N}$ for $\mathcal{M} = \{K\}$ agree with $\dim \mathcal{P}_2(\mathbb{R}^2) = 6$? Yes, it does!

Local shape functions (barycentric coordinate representation)

$$b_K^1 = (2\lambda_1 - 1)\lambda_1 ,$$

$$b_K^2 = (2\lambda_2 - 1)\lambda_2 ,$$

$$b_K^3 = (2\lambda_3 - 1)\lambda_3 ,$$

$$b_K^4 = 4\lambda_1\lambda_2 ,$$

$$b_K^5 = 4\lambda_2\lambda_3 ,$$

$$b_K^6 = 4\lambda_1\lambda_3 .$$

(3.4.4)

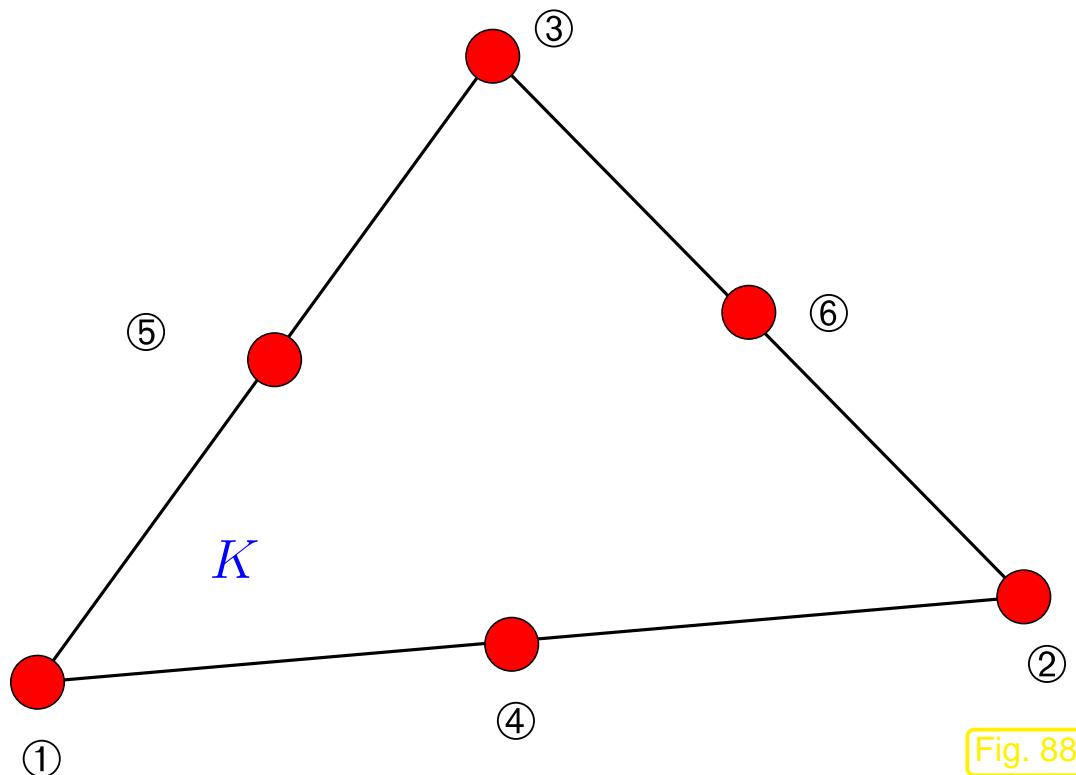
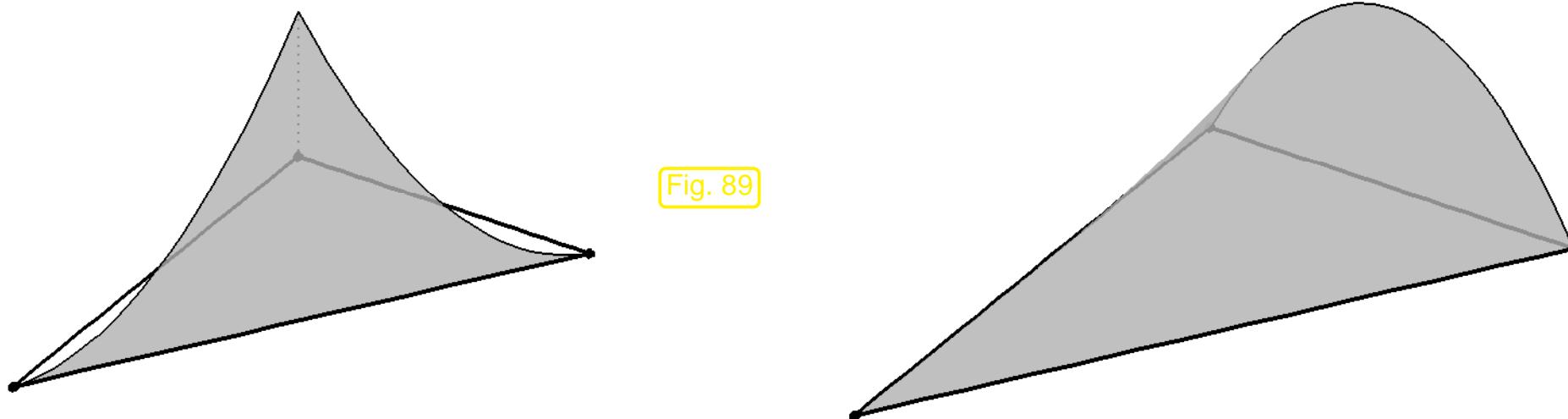


Fig. 88

To see the validity of the formulas (3.4.4), note that

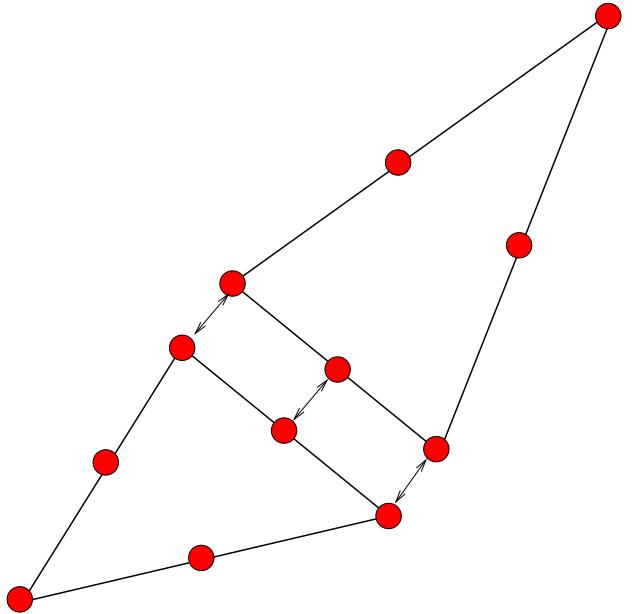
- $\lambda_i(\mathbf{a}^i) = 1$ and $\lambda_i(\mathbf{a}^j) = 0$, if $i \neq j$, where $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$ are the vertices of the triangle K ,
- $\lambda_1(\mathbf{m}^{12}) = \lambda_1(\mathbf{m}^{13}) = \frac{1}{2}$, where $\mathbf{m}^{ij} = \frac{1}{2}(\mathbf{a}^i + \mathbf{a}^j)$ denotes the midpoint of the edge connecting \mathbf{a}^i and \mathbf{a}^j ,
- each barycentric coordinate function λ_i is affine linear such that $\lambda_i\lambda_j \in \mathcal{P}_2(\mathbb{R}^2)$.

Selected local shape functions:



So far we have seen that *local shape functions* can be found that satisfy (3.4.3).

Issue: can the local shape functions from (3.4.4) be “stitched together” across interelement edges such that they yield a *continuous* global basis function? (Remember that Thm. 2.2.17 demands global continuity in order to obtain a subspace of $H^1(\Omega)$.)



The restriction of a quadratic polynomial to an edge is an *univariate* quadratic polynomial.

Fixing its value in three points, the midpoint of the edge and the endpoints, *uniquely* fixes this polynomial.

The local shape functions associated with the same interpolation node “from left and right” agree on the edge.

➤ continuity !

Fig. 90

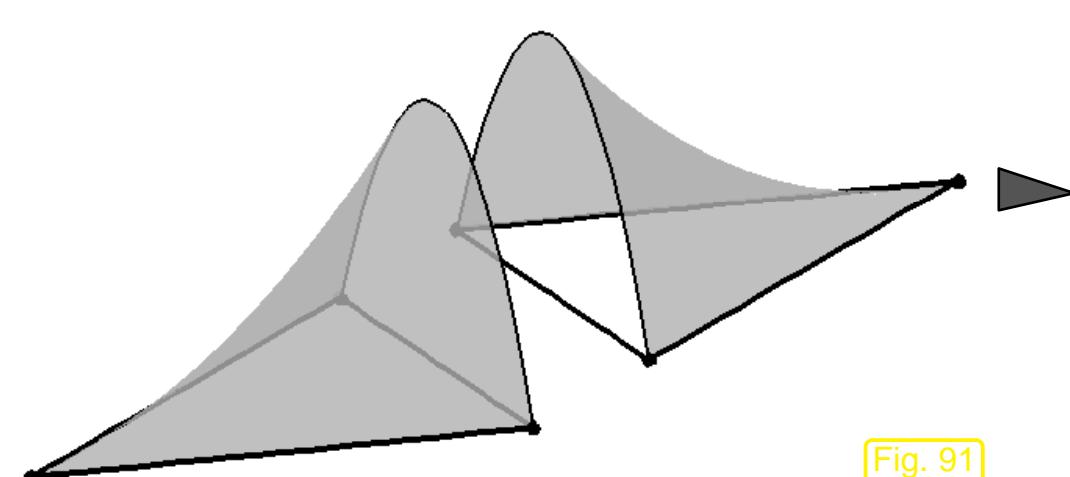


Fig. 91

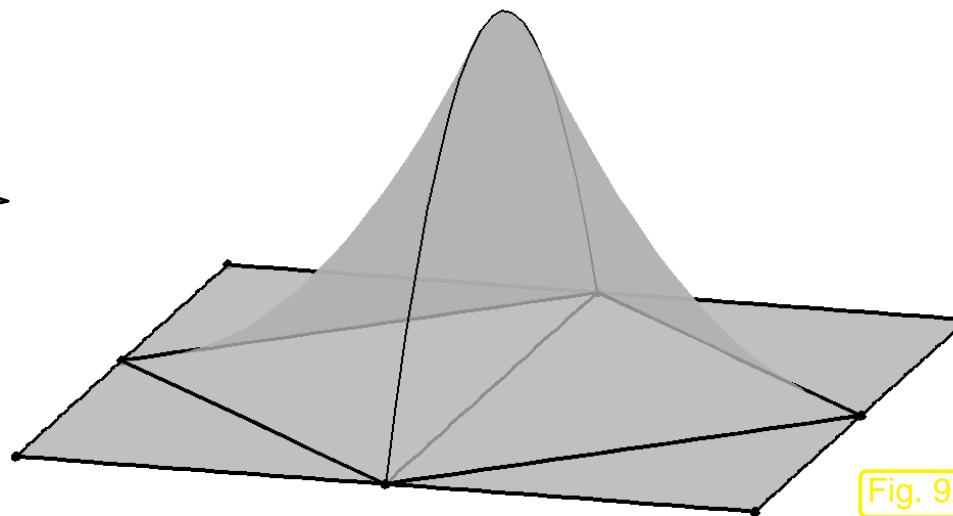
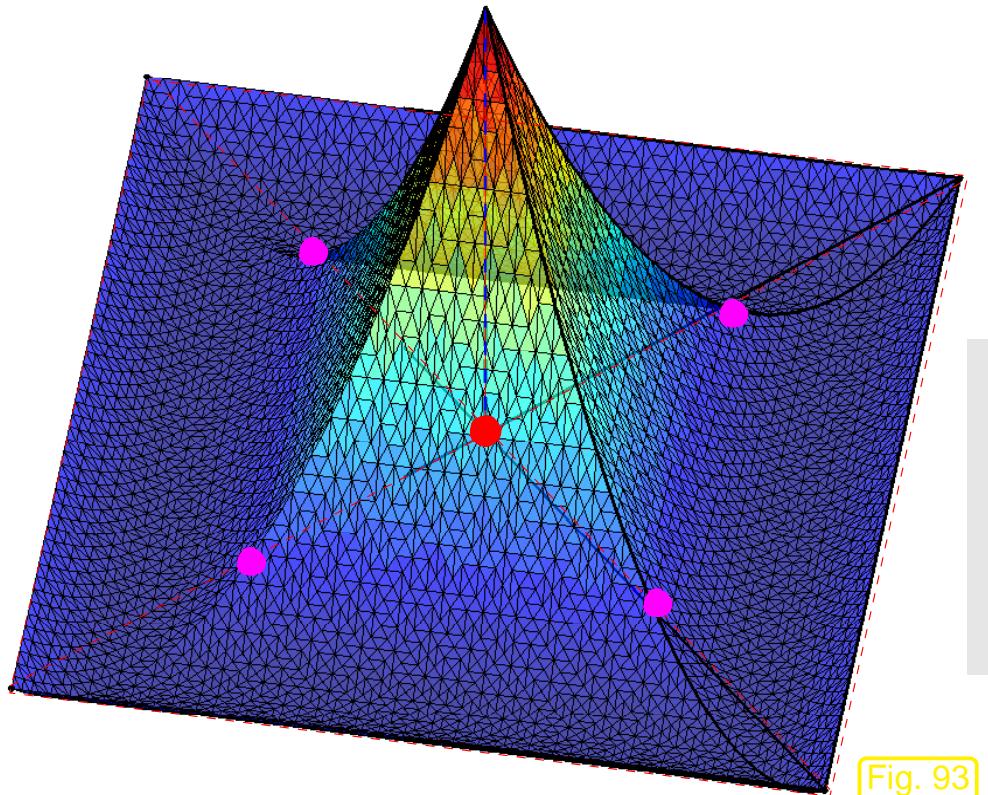


Fig. 92



◀ Global basis function for $\mathcal{S}_2^0(\mathcal{M})$ associated with a vertex

(3.4.3): this function attains value = 1 at a vertex (●) and vanishes at the midpoints (●) of the edges of adjacent triangles, as well as at any other vertex.

Fig. 93



Example 3.4.5 (Interpolation nodes for cubic and quartic Lagrangian FE in 2D).

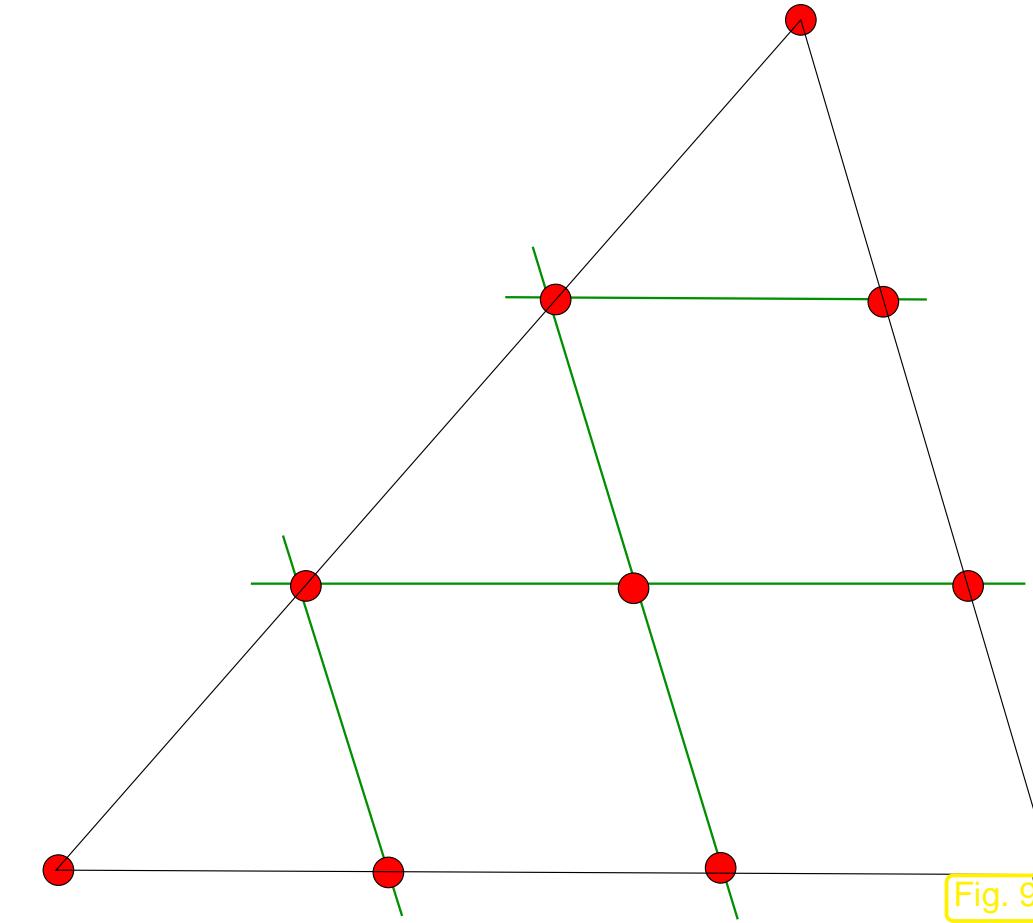


Fig. 94

(local) interpolation nodes for $\mathcal{S}_3^0(\mathcal{M})$

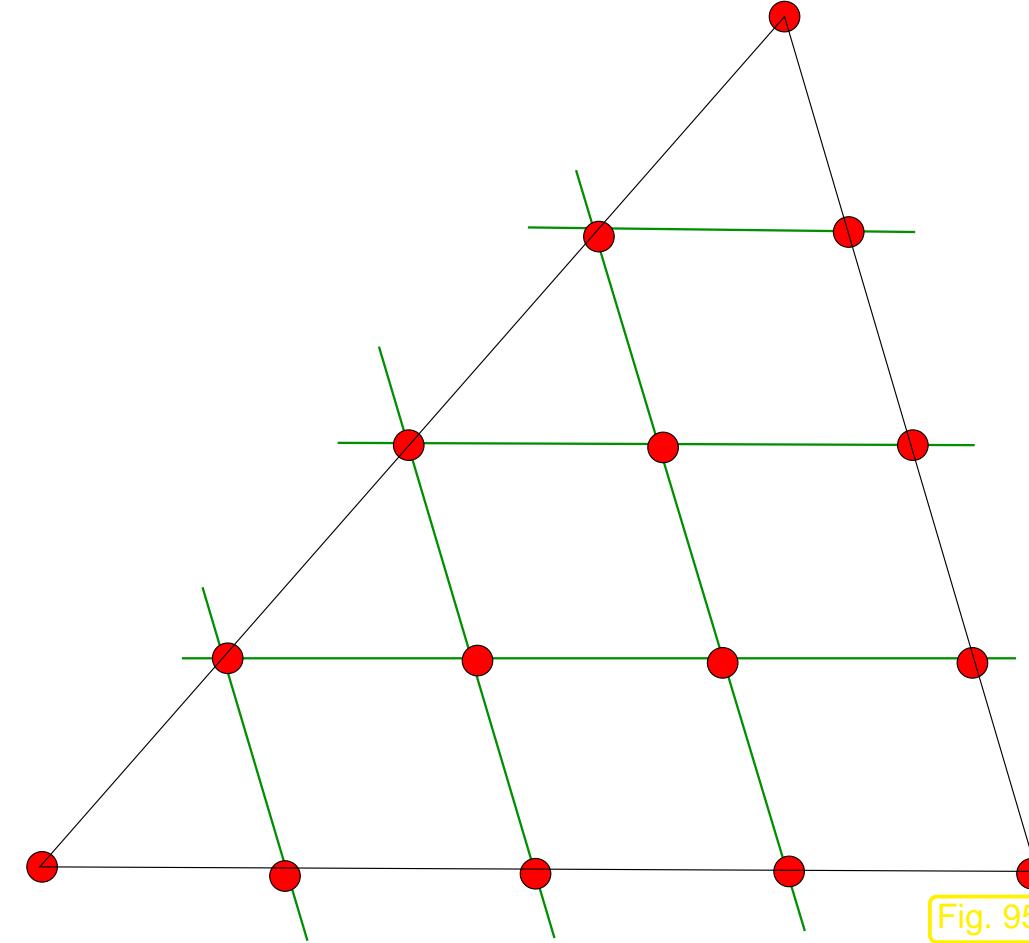


Fig. 95

(local) interpolation nodes for $\mathcal{S}_4^0(\mathcal{M})$



3.4.2 Tensor-product Lagrangian FEM

Now we consider tensor product meshes (grids), see (3.3.2), Fig. 80, for a 2D example.

Example 3.4.6 (Bilinear Lagrangian finite elements).

Sought: generalization of 1D piecewise linear finite element functions from Sect. 1.5.1.2, see Fig. 23, to 2D tensor product grid \mathcal{M} .

Tensor product structure of \mathcal{M} ➤ **tensor product construction** of FE space

This is best elucidated by a tensor product construction of basis functions:

$b_{N,x}^j(x)$: 1D tent function on $\mathcal{M}_x = \{[x_{j-1}, x_j], j = 1, \dots, n\}$
 $b_{N,y}^l(y)$: 1D tent function on $\mathcal{M}_y = \{[y_{j-1}, y_j], j = 1, \dots, n\}$

2D tensor product “tent function” associated with node \mathbf{p} :

$$b_N^{\mathbf{p}}(\mathbf{x}) = b_{N,x}^j(x_1) \cdot b_{N,y}^l(x_2), \quad \text{where } \mathbf{p} = (x_j, y_l)^T. \quad (3.4.7)$$

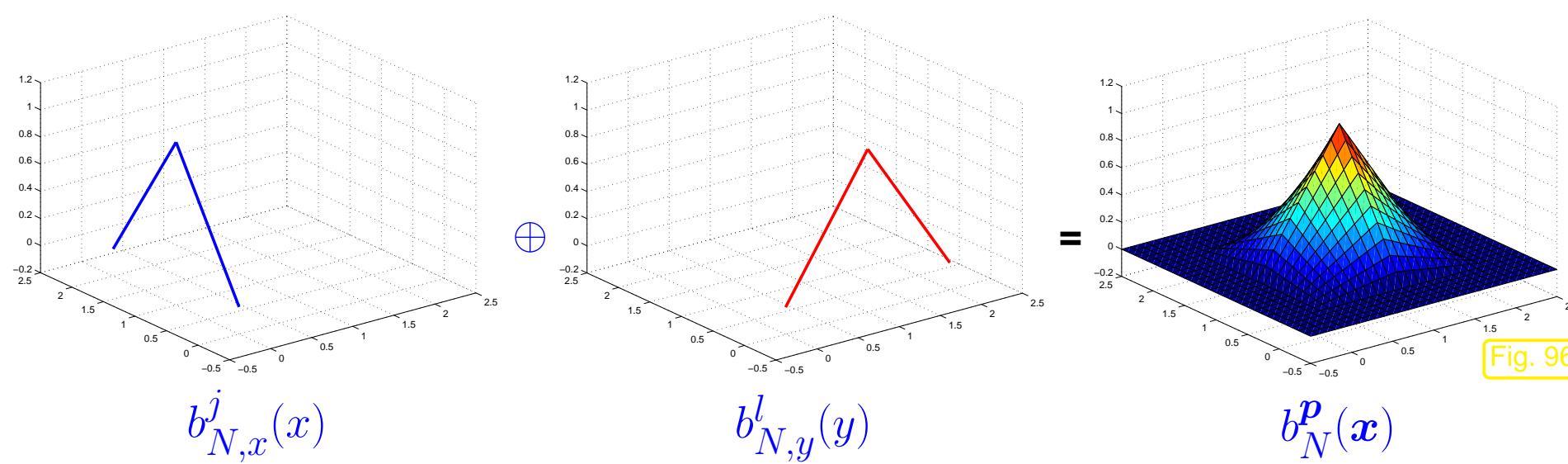
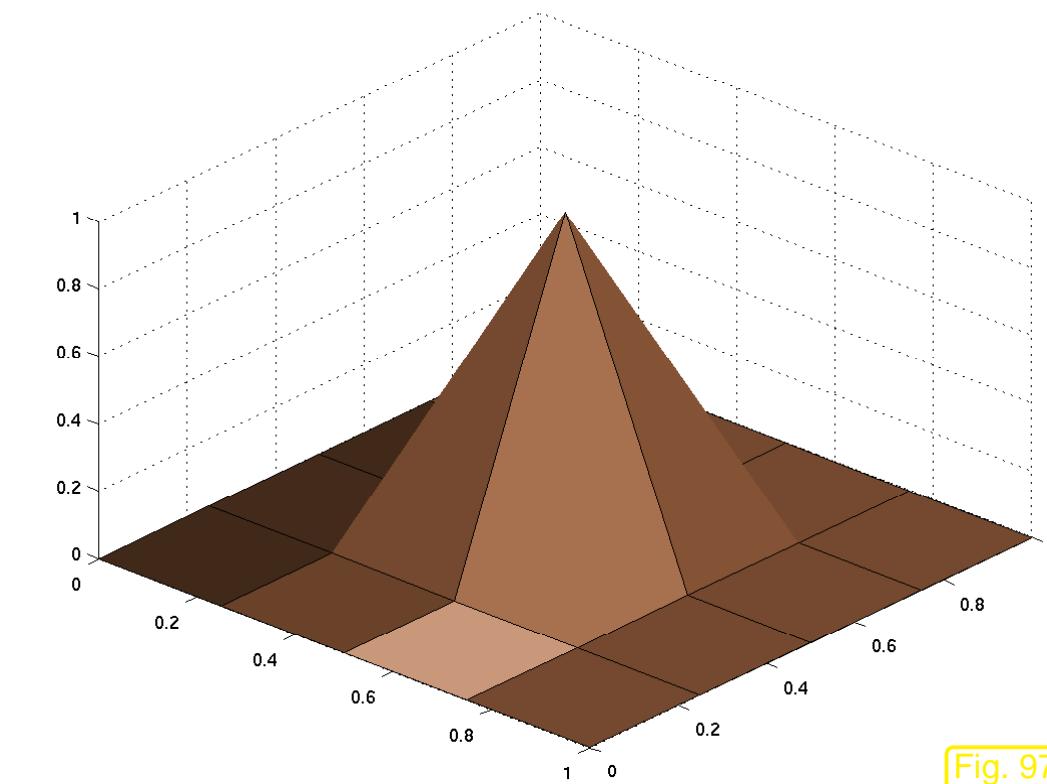


Fig. 96



△ 2D tensor product tent function

No pyramid !

Basis functions associated (\rightarrow Sect. 3.3.3, condition (c)) with nodes of \mathcal{M} ,

Tensor product construction ➤ bilinear local shape functions, e.g. on $K =]0, 1[^2$

$$\begin{aligned} b_K^1(\boldsymbol{x}) &= (1 - x_1)(1 - x_2) , \\ b_K^2(\boldsymbol{x}) &= x_1(1 - x_2) , \\ b_K^3(\boldsymbol{x}) &= x_1x_2 , \\ b_K^4(\boldsymbol{x}) &= (1 - x_1)x_2 . \end{aligned} \quad (3.4.8)$$

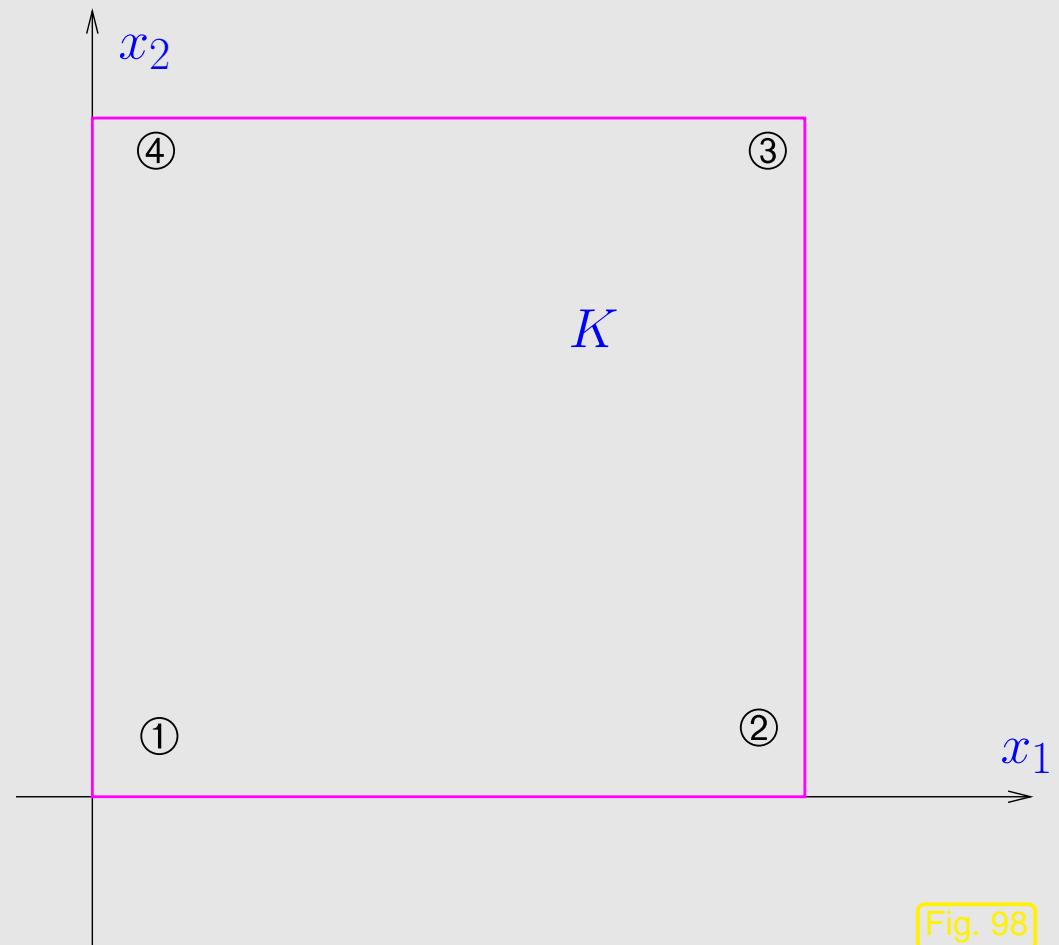
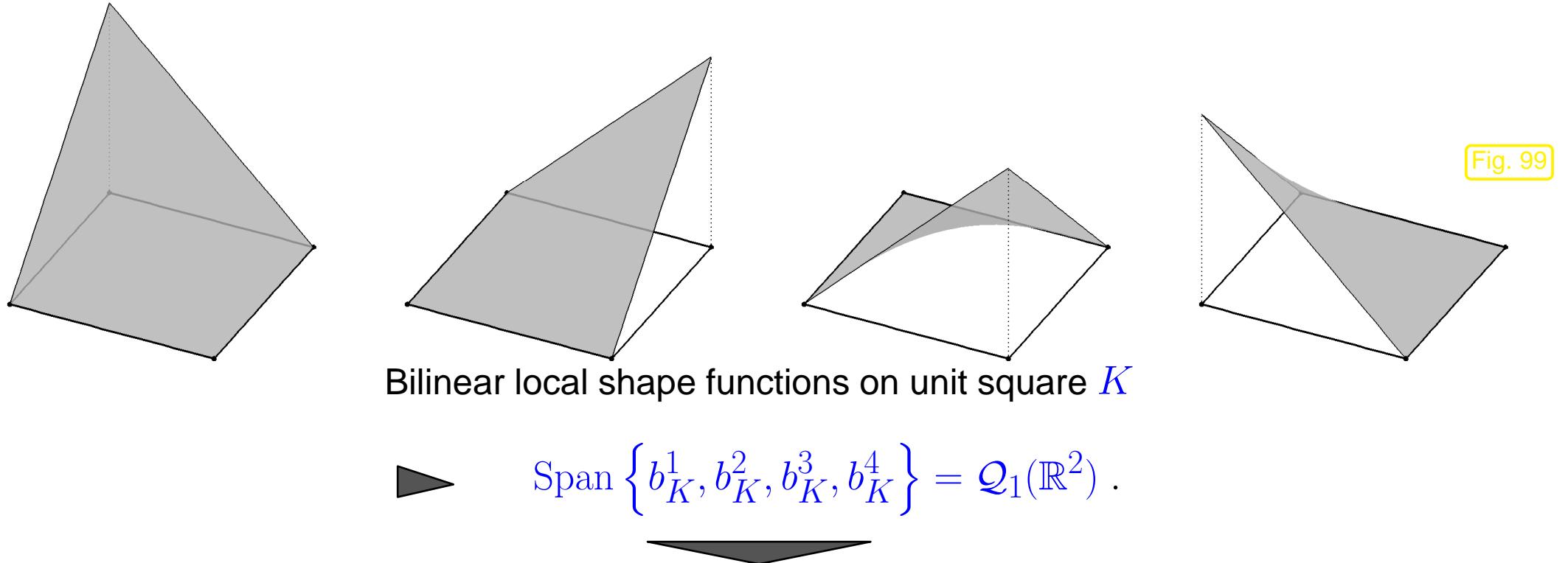


Fig. 98



Bilinear Lagrangian finite element space on 2D tensor product mesh \mathcal{M} :

$$\mathcal{S}_1^0(\mathcal{M}) := \{v \in C^0(\Omega) : v|_K \in \mathcal{Q}_1(\mathbb{R}^2) \ \forall K \in \mathcal{M}\} . \quad (3.4.9)$$

◇

The following is a natural generalization of (3.4.9) to higher degree local tensor product polynomials, see Def. 3.3.7:

Definition 3.4.10 (Tensor product Langrangian finite element spaces).

Space of p -th degree Lagrangian finite element functions on tensor product mesh \mathcal{M}

$$\mathcal{S}_p^0(\mathcal{M}) := \{v \in C^0(\bar{\Omega}): v|_K \in \mathcal{Q}_p(K) \ \forall K \in \mathcal{M}\}.$$

Terminology: $\mathcal{S}_1^0(\mathcal{M})$ = multilinear finite elements ($p = 1, d = 2$ = bilinear finite elements)

Remaining issue: definition of global basis functions (global shape functions)

Policy: use of **interpolation nodes** as in Sect. 3.4.1, see Ex. 3.4.2.

Example 3.4.11 (Quadratic tensor product Lagrangian finite elements).

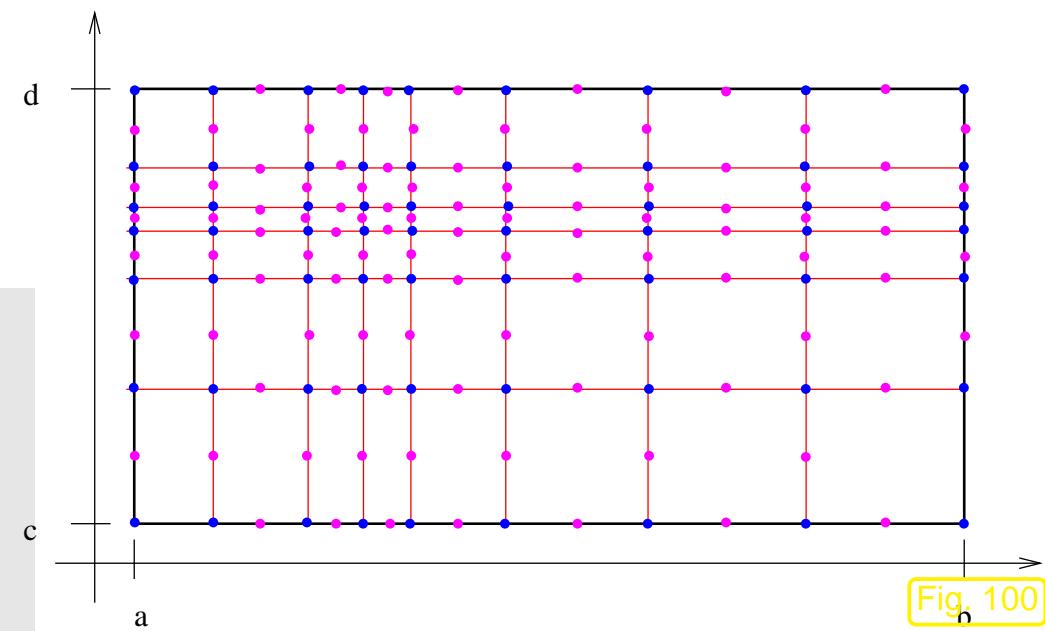
Consider case $p = 2, d = 2$ of Def. 3.4.10:

Interpolation nodes for $\mathcal{S}_2^0(\mathcal{M})$

$$\mathcal{N} = \mathcal{V}(\mathcal{M}) \cup \{\text{midpoints of edges}\} .$$

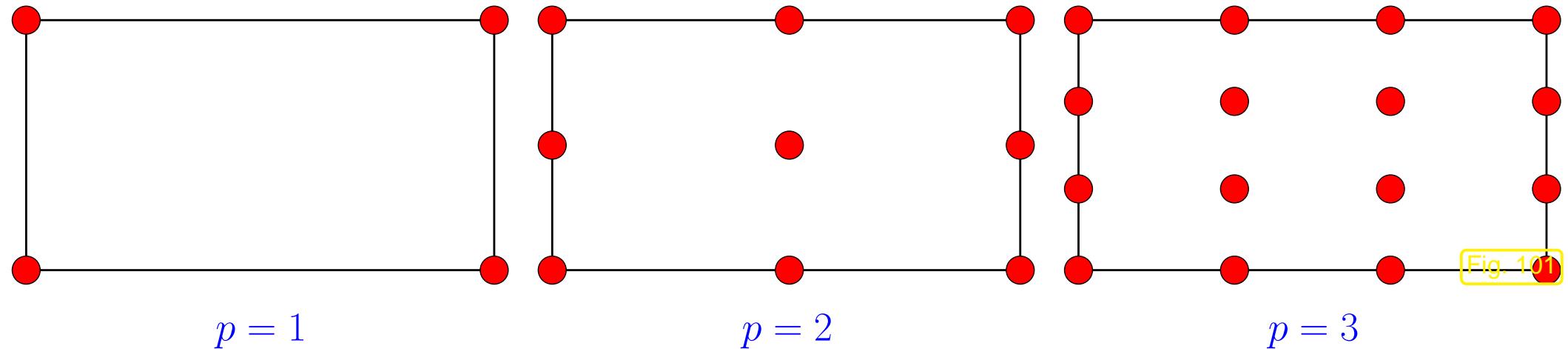
Note: number of interpolation nodes belonging to one cell is

$$9 = \dim \mathcal{Q}_2(\mathbb{R}^3) .$$



► Global basis functions defined analogously to (3.4.3).





Remark 3.4.12 (Imposing homogeneous Dirichlet boundary conditions).

What is a global basis for $\mathcal{S}_p^0(\mathcal{M}) \cap H_0^1(\Omega)$, where \mathcal{M} is either a simplicial mesh or a tensor product mesh?

We proceed analogous to Rem. 3.2.3: recall that global basis functions are defined via interpolation nodes \mathbf{p}^j , $j = 1, \dots, N$, see (3.4.3).

$$\mathcal{S}_{p,0}^0(\mathcal{M}) := \mathcal{S}_p^0(\mathcal{M}) \cap H_0^1(\Omega) = \text{Span} \left\{ b_N^j : \mathbf{p}^j \in \Omega \text{ (interior node)} \right\}. \quad (3.4.13)$$



Remark 3.4.14 ((Bi)-linear Lagrangian finite elements on hybrid meshes).

\mathcal{M} : 2D hybrid mesh comprising triangles & rectangles



Idea: use

- linear functions (\rightarrow Def. 3.3.3, $p = 1$) on triangular cells,
- bi-linear functions (\rightarrow Def. 3.4.10, $p = 1$) on rectangles.

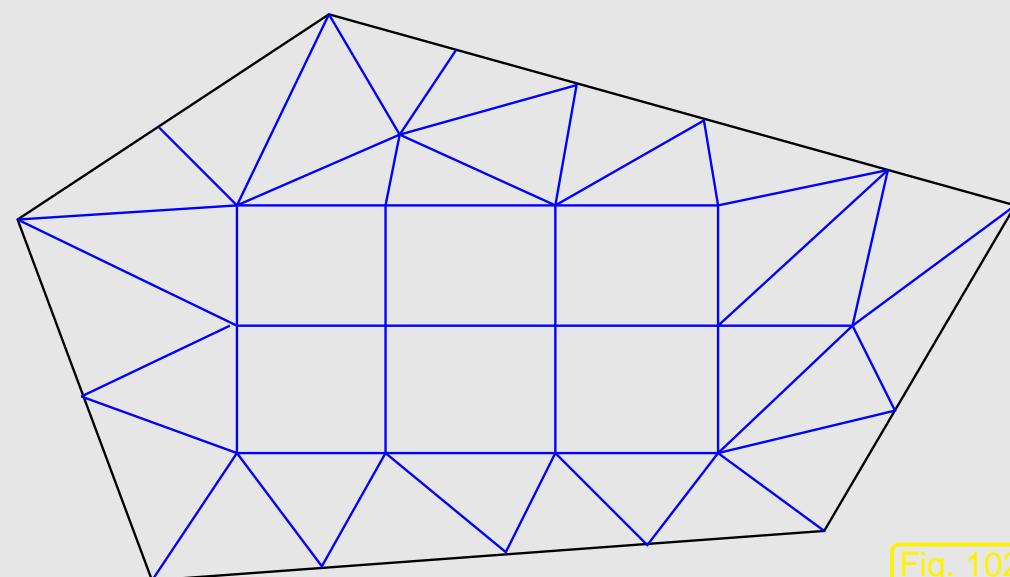
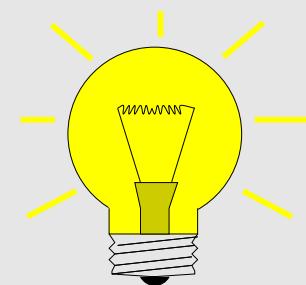


Fig. 102

$$\mathcal{S}_1^0(\mathcal{M}) = \left\{ v \in H^1(\Omega): v|_K \in \begin{cases} \mathcal{P}_1(\mathbb{R}^2) & , \text{ if } K \in \mathcal{M} \text{ is triangle,} \\ \mathcal{Q}_1(\mathbb{R}^2) & , \text{ if } K \in \mathcal{M} \text{ is rectangle} \end{cases} \right\}. \quad (3.4.15)$$

Two issues arise:

1. Does the prescription (3.4.15) yield a large enough space? (Note that $v \in H^1(\Omega) \Rightarrow \mathcal{S}_1^0(\mathcal{M}) \subset C^0(\Omega)$, but continuity might enforce too many constraints.)
2. Does the space from (3.4.15) allow for locally supported basis functions associated with nodes of the mesh?

We will give a positive answer to both questions by constructing the basis functions:

Define global shape functions b_N^j according to (3.2.2)

This makes sense, because

- linear/bi-linear functions on K are uniquely determined by their values in the vertices,
 - the restrictions to an edge of K of the local linear and bi-linear shape functions are both *linear* univariate functions, see Figs. 70, 99.
- Fixing vertex values for $v_N \in \mathcal{S}_1^0(\mathcal{M})$ uniquely determines v on all edges of \mathcal{M} already, thus, *ensuring global continuity*, which is necessary due to Thm. 2.2.17.

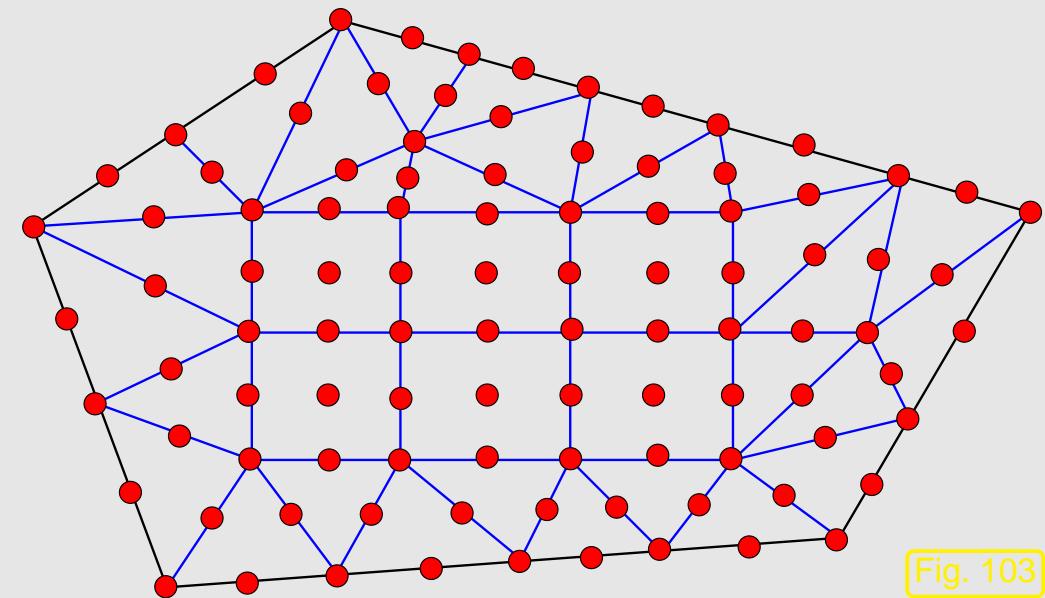
Remark 3.4.16 (Lagrangian finite elements on hybrid meshes).

\mathcal{M} : 2D hybrid mesh comprising triangles & rectangles

→ Matching interpolation nodes on edges of triangles and rectangles

► Glueing of local shape functions on triangles and rectangles possible

global interpolation nodes for $p = 2$ ▶



3.5 Implementation of FEM

This section discusses algorithmic details of Galerkin finite element discretization of 2nd-order elliptic variational problems for spatial dimension $d = 2, 3$ on bounded polygonal/polyhedral domains $\Omega \subset \mathbb{R}^d$.

The presentation matches the [LehrFEM finite element MATLAB library](#), parts of which will be made available for participants of the course. A detailed documentation is available from [1].

The guiding principle behind the implementation of finite element codes is

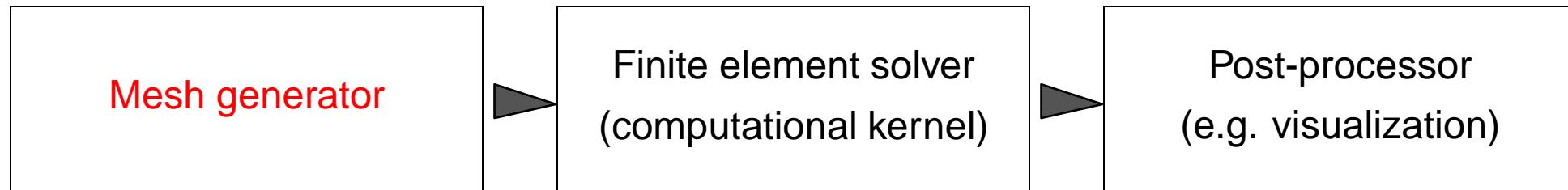
to rely on *local* computations as much as possible!

This is made possible by the *local supports* of the global basis functions, see Sect. 3.3.3, Ex. 3.3.10.

3.5.1 Mesh file format

CAD data

Parameters



Here “” designates passing of information, which is usually done by writing and reading files to and from hard disk. This requires particular file formats.

Example 3.5.1 (Triangular mesh: file format).

File format for storing triangular mesh (of polygonal domain):

```
# Two-dimensional simplicial mesh  
1  $\xi_1$   $\eta_1$  # Coordinates of first node  
2  $\xi_2$   $\eta_2$  # Coordinates of second node  
:  
 $N$   $\xi_N$   $\eta_N$  # Coordinates of  $N$ -th node (3.5.2)  
1  $n_1^1$   $n_2^1$   $n_3^1$   $X_1$  # Indices of nodes of first triangle  
2  $n_1^2$   $n_2^2$   $n_3^2$   $X_2$  # Indices of nodes of second triangle  
:  
 $M$   $n_1^M$   $n_2^M$   $n_3^M$   $X_M$  # Indices of nodes of  $M$ -th triangle
```

$X_i, i = 1, \dots, M \rightarrow$ extra information (e.g. material properties in triangle $\#i$).

Optional: additional information about edges (on $\partial\Omega$):

$K \in \mathbb{N}$ # Number of edges on $\partial\Omega$
 $n_1^1 n_2^1 Y_1$ # Indices of endpoints of first edge
 $n_1^2 n_2^2 Y_2$ # Indices of endpoints of second edge

(3.5.3)

:

$n_1^K n_2^K Y_K$ # Indices of endpoints of K -th edge

$Y_k, k = 1, \dots, K \rightarrow$ extra information



Example 3.5.4 (Mesh file format for MATLAB code “LehrFEM”).

Vertex coordinate file:

```
% List of vertices
1 +0.000000e+00 -1.000000e+00
2 +1.000000e+00 +0.000000e+00
3 +0.000000e+00 +1.000000e+00
4 -1.000000e+00 +0.000000e+00
5 +0.000000e+00 +0.000000e+00
```

Cell information file:

```
% List of elements
1 1 2 5
2 2 3 5
3 3 4 5
4 4 1 5
```

Loading a mesh

```
m = load_Mesh('Coord_Circ.dat', ...  
              'Elem_Circ.dat');  
plot_Mesh(m, 'apts');
```

Option flags:

'a': with axes

'p': vertex labels on

't': cell labels on

's': caption/title on

For details see [1, Sect. 1.3.1], [1, Sect. 1.3.2].

2D triangular mesh

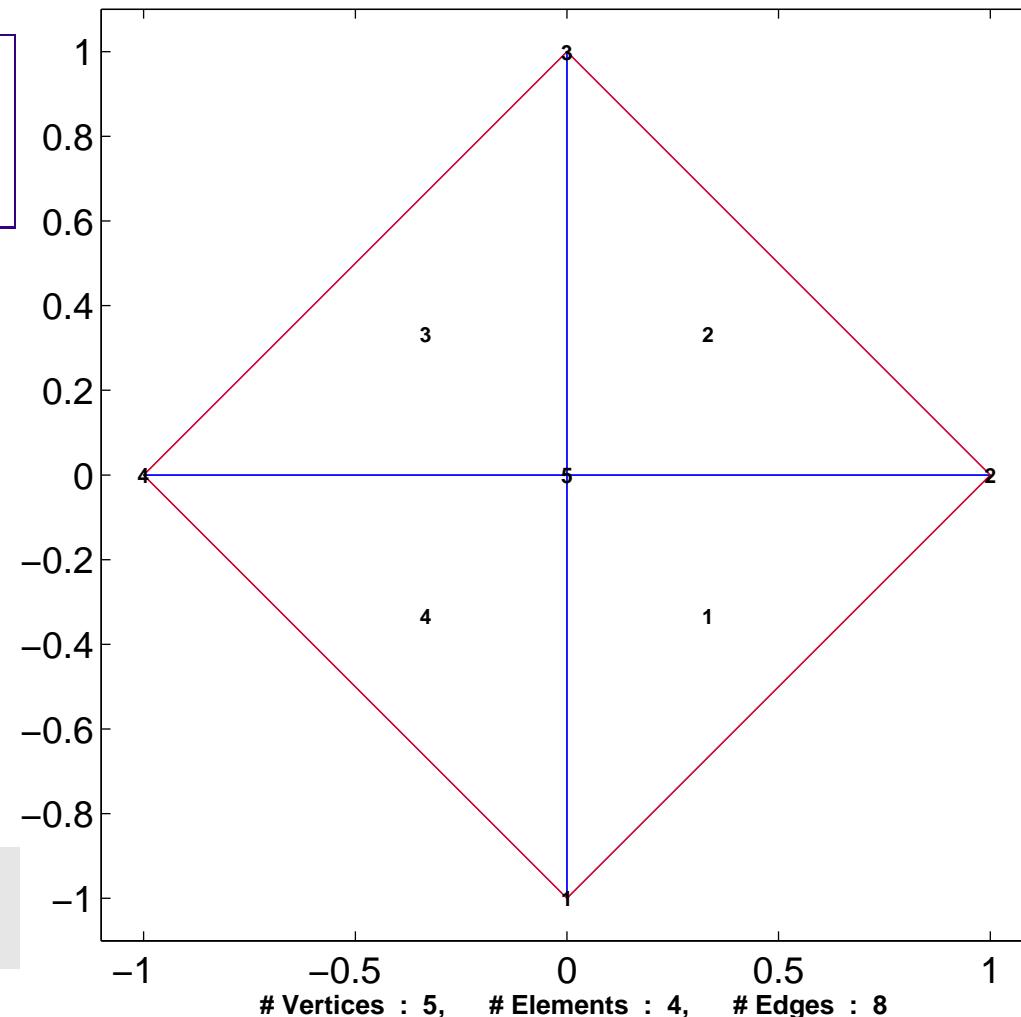


Fig. 104

How to create a mesh ?

→

Mesh generation (beyond scope of this course)

→ <http://www.andrew.cmu.edu/user/sowen/mesh.html>

- Free software:
- DistMesh (MATLAB, used in “LehrFEM”, see [1, Sect. 1.2])
 - NETGEN (industrial strength open source mesh generator)
 - Triangle (easy to use 2D mesh generator)
 - TETGEN (Tetrahedral mesh generation)

Example 3.5.5 (Mesh generation in LehrFEM).

Algorithm & details → [17], more explanations in [1, Sect. 1.2].

```
BBOX = [-1 -1; 1 1];  
H0 = 0.1;  
DHD = @(x) sqrt(x(:,1).^2+x(:,2).^2)-1;  
HHANDLE = @(x) ones(size(x,1),1);  
Mesh = init_Mesh(BBOX,H0,DHD,...  
                  HHANDLE,[],1);  
save_Mesh(Mesh,'Coordinates.dat',...  
          'Elements.dat');
```

Bounding box
Largest reasonable edge length
Signed distance function $\varphi(\mathbf{x})$:
(distance from $\partial\Omega$, $\varphi(\mathbf{x}) < 0 \Leftrightarrow \mathbf{x} \in \Omega$)
Element size function
(determines local edge length)



3.5.2 Mesh data structures [1, Sect. 1.1]

mesh data structure must provide:

1. offer unique identification of cells/(faces)/(edges)/vertices
2. represent **mesh topology** (= incidence relationships of cells/faces/edges/vertices)
3. describe **mesh geometry** (= location/shape of cells/faces/edges/vertices)
4. allow sequential access to edges/faces of a cell
(→ traversal of local shape functions/degrees of freedom)
5. make possible traversal of cells of the mesh (→ **global numbering**)

Focus: **array oriented data layout** (→ MATLAB, FORTRAN)

Notation:

\mathcal{M} = mesh (set of elements), $\mathcal{V}(\mathcal{M})$ = set of nodes (vertices) in \mathcal{M} , $\mathcal{E}(\mathcal{M})$ = set of edges in \mathcal{M}

Case: d -dimensional simplicial triangulation \mathcal{M} , *minimal data structure* (cf. Sect. 3.5.1)

→ Coordinates of vertices $\mathcal{V}(\mathcal{M}) : \#\mathcal{V}(\mathcal{M}) \times d$ -array Coordinates of reals

→ Vertex indices for cells: $\#\mathcal{M} \times (d + 1)$ -array Elements of integers.

► Already offers complete description of the mesh topology and geometry !

Optional extra information:

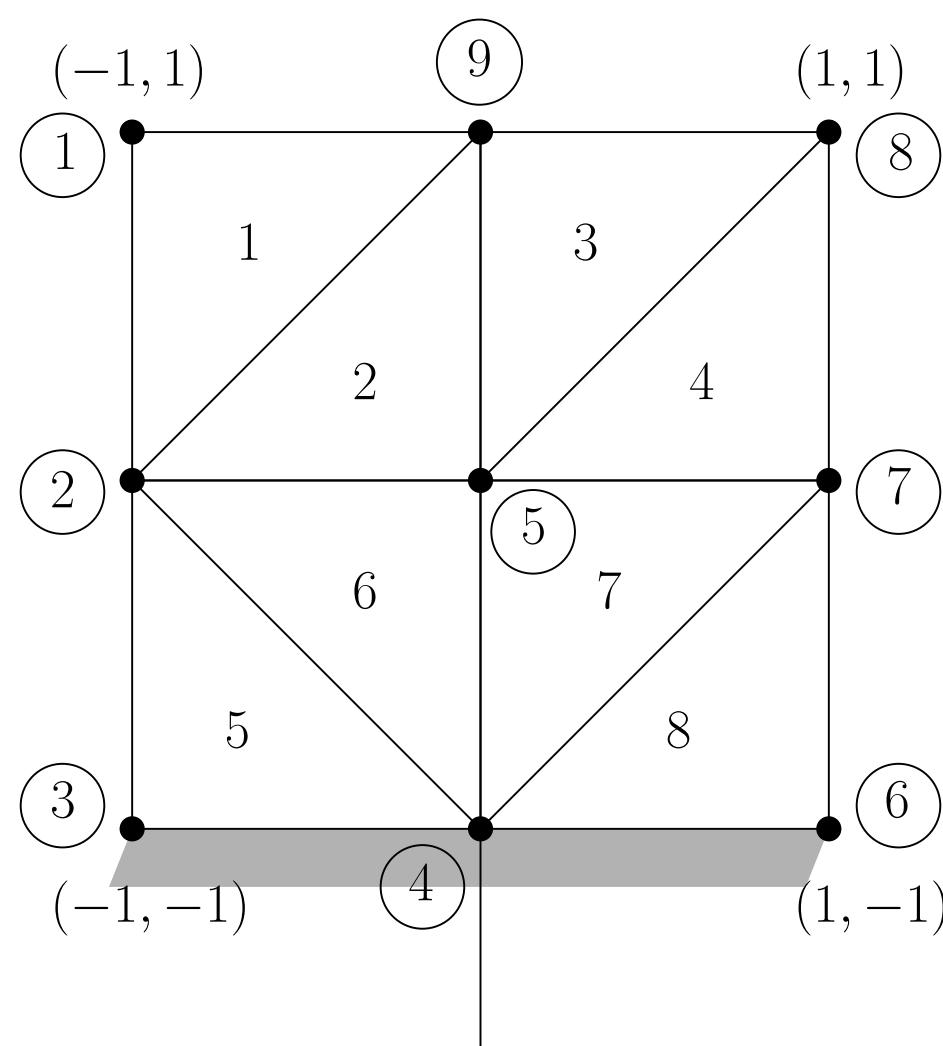
→ Edge connecting vertices: $\#\mathcal{V}(\mathcal{M}) \times \#\mathcal{V}(\mathcal{M})$ symmetric sparse integer matrix $I_{\mathcal{E}}$

$$(\mathbf{I}_{\mathcal{E}})_{ij} := \begin{cases} 0 & , \text{ if vertex } \#i \text{ not linked to } \#j \\ e_{ij} & , \text{ if edge connecting } \#i \text{ and } \#j \end{cases}$$

here e_{ij} is the unique edge number $\in \{1, 2, \dots, \#\mathcal{E}(\mathcal{M})\}$

→ End points of the edges: $\#\mathcal{E}(\mathcal{M}) \times 2$ array of integer (= vertex indices of end points).

→ Cell adjacent to edges: $\#\mathcal{E}(\mathcal{M}) \times 2$ array of integers (=cell indices)
(one cell index = 0 if edge is on $\partial\Omega$)



Example 3.5.6 (Arrays storing 2D triangular mesh).

i	Coordinates		K_j	Vertex indices		
1	-1	1	1	1	2	9
2	-1	0	2	2	5	9
3	-1	-1	3	5	8	9
4	0	-1	4	5	7	8
5	0	0	5	3	4	2
6	1	-1	6	4	5	2
7	1	0	7	4	7	5
8	1	1	8	4	6	7
9	0	1				

Array Coordinates Array Elements

Fig. 105

Note: Global shape functions associated with edges/faces \Rightarrow extra information required !

Example 3.5.7 (Extended MATLAB mesh data structure). $\rightarrow [1, \text{Sect. ??}]$

`mesh = add_Edge2Elem(add_Edges(init_Mesh(BBOX,H0,DHD,HHANDLE,[],1)))`

(init_Mesh \rightarrow Ex. 3.5.5)

<code>mesh =</code>	vertex coordinates, see Ex. 3.5.4
<code>Coordinates: [5x2 double]</code>	vertex indices of triangles, see Ex. 3.5.4
<code>Elements: [4x3 double]</code>	indices of endpoints in Coordinates array
<code>Edges: [8x2 double]</code>	$\#\mathcal{V}(\mathcal{M}) \times \#\mathcal{V}(\mathcal{M})$ sparse integer matrix: entry (i, j) = edge index, if $\neq 0$
<code>Vert2Edge: [5x5 double]</code>	$\#\mathcal{E}(\mathcal{M}) \times 2$ integer array:
<code>Edge2Elem: [8x2 double]</code>	indices of adjacent cells in Elements array
<code>EdgeLoc: [8x2 double]</code>	$\#\mathcal{E}(\mathcal{M}) \times 2$ integer array: local indices of edges w.r.t. adjacent cells

Notation: $\mathcal{E}(\mathcal{M}) \triangleq$ edges of 2D mesh

How to number ⇔ order

local shape functions
global shape functions

?

Elements, Edges arrays ➤ ordering of vertices of cells/endpoints of edges

Arrays (of vertices,cells,edges) \gg array indices \gg numbering of global shape functions

Remark 3.5.8. Second option: C++/JAVA-style object oriented data layout

Nodes, cells of \mathcal{M} \longleftrightarrow dynamically allocated objects (instances of classes Node, Cell)

```
class Node {  
private:  
    double x,y;  
    ID id;  
public:  
    Node(double x,double y,ID id=0);  
    Point getCoords(void) const;  
    ID getId(void) const;  
};
```

```
class Cell {  
private:  
    const vector<Node*> vertices;  
    ID id;  
public:  
    Cell(const vector<Node*> &vertices,ID id=0);  
    int NoNodes(void) const;  
    const Node &getNode(int) const;  
    ID getId(void) const;  
};
```

```
class BdFace {  
private:  
    const vector<Node*> vertices;  
    BdCond bdcond;  
public:  
    BdFace(const vector<Node*> &vertices);  
    int NoNodes(void) const;  
    const Node &getNode(int) const;  
    BdCond getBdCond(void) const;  
};
```

```
class Mesh {  
private:  
    list<Node> nodes;  
    list<Cell> cells;  
    list<BdFace> bdfaces;  
public:  
    Mesh(istream &file);  
    virtual Mesh(void);  
    const list<Node> &Nodes(void) const;  
    const list<Cell> &Cells(void) const;  
    const list<BdFace> &BdFaces(void) const;  
};
```

ID getId() → provides **unique identifier** for each node/cell.

Distinguish:

- **local objects** (→ classes Node, Cell, BdFace)
- **global objects** (“mesh management” class Mesh, see below)



3.5.3 Assembly [1, Sect. 5]

“Assembly” = term used for computing entries of stiffness matrix/right hand side vector (load vector) in a finite element context.

From the dictionary: “Assemble” = to fit together all the separate parts of sth.

Aspects of assembly for linear Lagrangian finite elements ($V_{0,N} = S_{1,0}^0(\mathcal{M})$) were discussed in Sects. 3.2.5, 3.2.6. (Refresh yourself on these sections in case you cannot remember the main ideas behind building the Galerkin matrix and right hand side vector.)

We consider a discrete variational problem ($V_{0,N}$ = FE space, $\dim V_{0,N} = N \in \mathbb{N}$, see (3.1.3))

$$u_N \in V_{0,N}: \quad a(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N}. \quad (3.1.3)$$

To be computed (see also Sect. 3.2.5, Sect. 3.2.6):

- Galerkin matrix (stiffness matrix): $\mathbf{A} = \left(a(b_N^j, b_N^i) \right)_{i,j=1}^N \in \mathbb{R}^{N,N}$
- r.h.s. vector (load vector): $\vec{\varphi} := \left(\ell(b_N^i) \right)_{i=1}^N \in \mathbb{R}^N$

both can be written in terms of **local cell contributions**, since usually

$$a(u, v) = \sum_{K \in \mathcal{M}} a_K(u|_K, v|_K) , \quad \ell(v) = \sum_{K \in \mathcal{M}} \ell_K(v|_K). \quad (3.5.9)$$

Example: bilinear forms/linear forms arising from 2nd-order elliptic BVPs, e.g, (2.9.1), (2.9.2), (2.9.3), can be localized in straightforward fashion by restricting integration to mesh cells:

$$\mathbf{a}(u, v) := \int_{\Omega} \alpha(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\mathbf{x} = \sum_{K \in \mathcal{M}} \underbrace{\int_K \alpha(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\mathbf{x}}_{=: \mathbf{a}_K(u|_K, v|_K)} , \quad (3.5.10)$$

$$\ell(v) := \int_{\Omega} f v \, d\mathbf{x} = \sum_{K \in \mathcal{M}} \underbrace{\int_K f v \, d\mathbf{x}}_{=: \ell_K(v|_K)} . \quad (3.5.11)$$



Recall (3.3.11): Restrictions of global shape functions to cells = local shape functions

Definition 3.5.12 (Element (stiffness) matrix and element (load) vector).

Given local shape functions $\{b_K^1, \dots, b_K^Q\}$, $Q \in \mathbb{N}$, we call

$$\text{element (stiffness) matrix} \quad \mathbf{A}_K := \left(a_K(b_K^j, b_K^i) \right)_{i,j=1}^Q \in \mathbb{R}^{Q,Q},$$

$$\text{element (load) vector} \quad \vec{\varphi}_K := \left(\ell_K(b_K^i) \right)_{i=1}^Q \in \mathbb{R}^Q.$$

Note: Here Q , the number of local shape functions on element $K \in \mathcal{M}$, is independent of K . In general, we could also have $Q = Q_K$ when we blend several element types in one mesh, see Rem. 3.4.14.

Type of FE space	Q
degree p Lagrangian FE on <i>triangular</i> mesh	$\dim \mathcal{P}_p(\mathbb{R}^2) = \frac{1}{2}(p+1)(p+2)$
degree p Lagrangian FE on <i>tetrahedral</i> mesh	$\dim \mathcal{P}_p(\mathbb{R}^3) = \frac{1}{6}(p+1)(p+2)(p+3)$
degree p Lagrangian FE on <i>tensor product</i> mesh in 2D	$\dim \mathcal{Q}_p(\mathbb{R}^2) = (p+1)^2$

Again scrutinize Figs. 72, 73 and the accompanying remarks in Sect. 3.2.5. We learn that in the special setting of this section

- the entries of the finite element Galerkin matrix can be obtained by summing *corresponding* entries of *some* element matrices,
- this corresponding entry of an element matrices is determined by the unique association of a local basis function to a global basis function.

These insights are formalized in the next theorem.

Theorem 3.5.13. *The stiffness matrix and load vector can be obtained from their cell counterparts by*

$$\mathbf{A} = \sum_K \mathbf{T}_K^\top \mathbf{A}_K \mathbf{T}_K , \quad \vec{\varphi} = \sum_K \mathbf{T}_K^\top \vec{\varphi}_K , \quad (3.5.14)$$

with the **index mapping matrices** (“T-matrices”) $\mathbf{T}_K \in \mathbb{R}^{Q,N}$, defined by

$$(\mathbf{T}_K)_{ij} := \begin{cases} 1 & , \text{if } (b_N^j)|_K = b_K^i , \\ 0 & , \text{otherwise.} \end{cases} \quad 1 \leq i \leq Q, 1 \leq j \leq N . \quad (3.5.15)$$

Note: Every T-matrix has exactly one non-vanishing entry per row.

Proof. (of Thm. 3.5.13)

$$(\mathbf{A})_{ij} = \mathbf{a}(b_N^j, b_N^i) = \sum_{K \in \mathcal{M}} \mathbf{a}_K(b_N^j|_K, b_N^i|_K) = \sum_{\substack{K \in \mathcal{M}, \text{supp}(b_N^j) \cap K \neq \emptyset, \\ \text{supp}(b_N^i) \cap K \neq \emptyset}} \mathbf{a}_K(b_K^{l(j)}, b_K^{l(i)}) = \sum_{\substack{K \in \mathcal{M}, \text{supp}(b_N^j) \cap K \neq \emptyset, \\ \text{supp}(b_N^i) \cap K \neq \emptyset}} (\mathbf{A}_K)_{l(i), l(j)}$$

$l(i) \in \{1, \dots, Q\}$, $1 \leq i \leq N \hat{=} \text{index of the local shape function corresponding to the global shape function } b_N^i \text{ on } K$.

➤ By (3.5.15), the indices $l(i)$ encode the T-matrix according to

$$(\mathbf{T}_K)_{l(i),i} = 1 , \quad i = 1, \dots, N ,$$

where all other entries of \mathbf{T}_K are understood to vanish.

$$\Rightarrow (\mathbf{A})_{ij} = \sum_{\substack{K \in \mathcal{M}, \text{supp}(b_N^j) \cap K \neq \emptyset, \\ \text{supp}(b_N^i) \cap K \neq \emptyset}} \sum_{l=1}^Q \sum_{n=1}^Q (\mathbf{T}_K)_{li} (\mathbf{A}_K)_{ln} (\mathbf{T}_K)_{nj} . \quad \square$$

Example 3.5.16 (Assembly for linear Lagrangian finite elements on triangular mesh).

Using the local/global numbering indicated beside

$$\rightarrow \mathbf{T}_{K^*} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

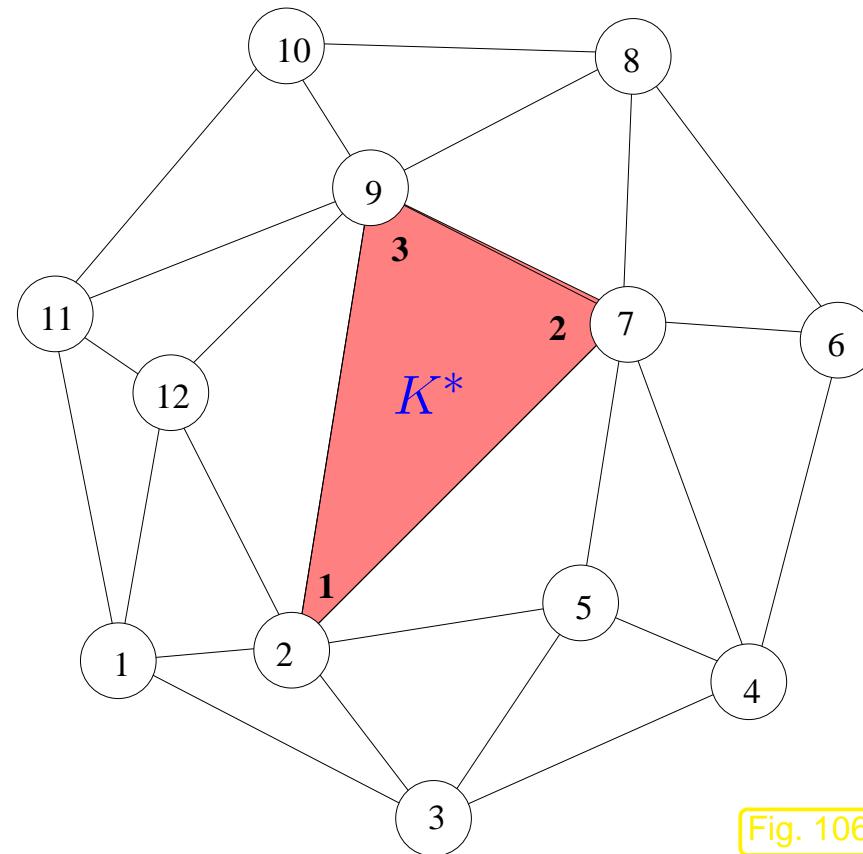


Fig. 106



Cell oriented assembly \leftrightarrow (3.5.14) $\leftrightarrow \mathbf{A} = \sum_K \mathbf{T}_K^\top \mathbf{A}_K \mathbf{T}_K$

$$\mathbf{A} = \sum_K \mathbf{T}_K^\top \mathbf{A}_K \mathbf{T}_K := \left\{ \begin{array}{l} \text{foreach } K \in \mathcal{M} \text{ do} \\ \quad \text{\textcolor{red}{local}} \text{ operations on } K (\rightarrow \mathbf{A}_K) \text{ and } \mathbf{A} = \mathbf{A} + \mathbf{T}_K^\top \mathbf{A}_K \mathbf{T}_K \\ \text{enddo} \end{array} \right\}$$

Notion: local operations $\hat{=}$

- required only data from fixed “neighbourhood” of K
- computational effort “ $O(1)$ ”: independent of $\#\mathcal{M}$

► Computational cost(Assembly of Galerkin matrix \mathbf{A}) = $O(\#\mathcal{M})$

Cell oriented assembly in LehrFEM [1, Sect. ??]

```
function A = assemble(Mesh)
for k = Mesh.Elements'
    idx = ①
    Aloc = ②
    A(idx, idx) = A(idx, idx)+Aloc;
end
```

① row vector of index numbers of global shape functions $b_N^{i_1}, \dots, b_N^{i_Q} \in V_N$ corresponding to local shape functions b_K^1, \dots, b_K^Q :

► $\text{idx} = (i_1, \dots, i_Q)$
 (encodes index mapping matrix \mathbf{T}_K)

② $Q \times Q$ element stiffness matrix

For Lagrangian FEM of fixed degree p (\rightarrow Sect. 3.4):

the total computational effort is of the order $O(\#\mathcal{M}) = O(N)$, $N := \dim \mathcal{S}_p^0(\mathcal{M})$.

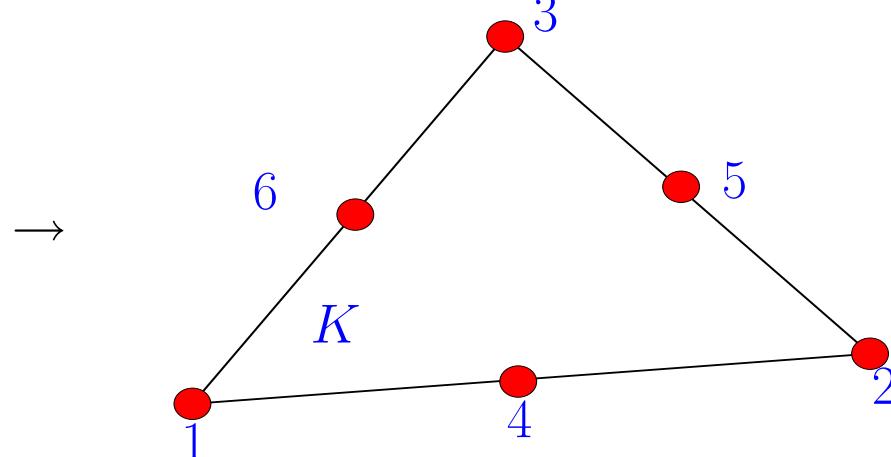
Example 3.5.17 (Assembly for quadratic Lagrangian FE in MATLAB code).

Setting: FE space $\mathcal{S}_2^0(\mathcal{M})$ on triangular mesh \mathcal{M} of polygon $\Omega \subset \mathbb{R}^2$, see Ex. 3.4.2

Recall: 6 local shape functions: 3 vertex-associated, 3 edge-associated \rightarrow (3.4.4)

Convention: vertex-associated global shape functions $\rightarrow b_N^1, \dots, b_N^{\#\mathcal{V}(\mathcal{M})}$
 edge-associated global shape functions $\rightarrow b_N^{\#\mathcal{V}(\mathcal{M})+1}, \dots, b_N^{\#\mathcal{V}(\mathcal{M})+\#\mathcal{E}(\mathcal{M})}$

Local numbering



```

function A = assemMat_QFE(Mesh,EHandle,varargin)
    % ①
    nV = size(Mesh.Coordinates,1);
    nE = size(Mesh.Elements,1)

    % ②
    I = zeros(36*nE,1); J = I; a = I; offset = 0;
    for k = 1:nE
        vidx = Mesh.Elements(k,:);
        idx = [vidx, ...
            Mesh.Vert2Edge(vidx(1),vidx(2))+nV, ...
            Mesh.Vert2Edge(vidx(2),vidx(3))+nV, ...
            Mesh.Vert2Edge(vidx(3),vidx(1))+nV];
        Aloc = transpose(EHandle(Mesh.Coordinates(vidx,:), ...
            % ⑤
            Mesh.ElemFlag(k),varargin{:}))';

        % ④
        Qsq = prod(size(Aloc)); range = offset + 1:Qsq;
        t = idx(ones(length(idx),1),:)'; I(range) = t(:);
        t = idx(ones(1,length(idx)),:); J(range) = t(:);
        a(range) = Aloc(:);
        offset = offset + Qsq;
    end
    % ⑥
    A = sparse(I,J,a);

```

- ①: EHandle (function handle) → provides element stiffness matrix $\mathbf{A}_K \in \mathbb{R}^{6,6}$
- ②: $I, J, a \hat{=} \text{linear arrays storing } (i, j, (\mathbf{A})_{ij})$ for stiffness matrix \mathbf{A} .
Initialized with 0 for the sake of efficiency → Ex. 3.5.18
- ③: idx $\hat{=} \text{ index mapping vector, see ① above}$
- ④: Aloc = $\mathbf{A}_K \in \mathbb{R}^{6,6}$ (element stiffness matrix → Def. 3.5.12)
- ⑤: Mesh.ElemFlag(k) marks groups of elements (e.g. to select local coefficient function $\alpha(\mathbf{x})$ in (2.8.4))
- ⑥: Build sparse MATLAB-matrix (→ Def. 3.2.6) from index-entry arrays, see manual entry for MATLAB function `sparse`.



Remark 3.5.18 (Efficient implementation of assembly). → [14, Sect. 2.6.2]

tic-toe-timing (min of 4v runs), MATLAB V7, Intel Pentium 4 Mobile CPU 1.80GHz, Linux
Computation of element stiffness matrices skipped !

- *Sparse assembly:*

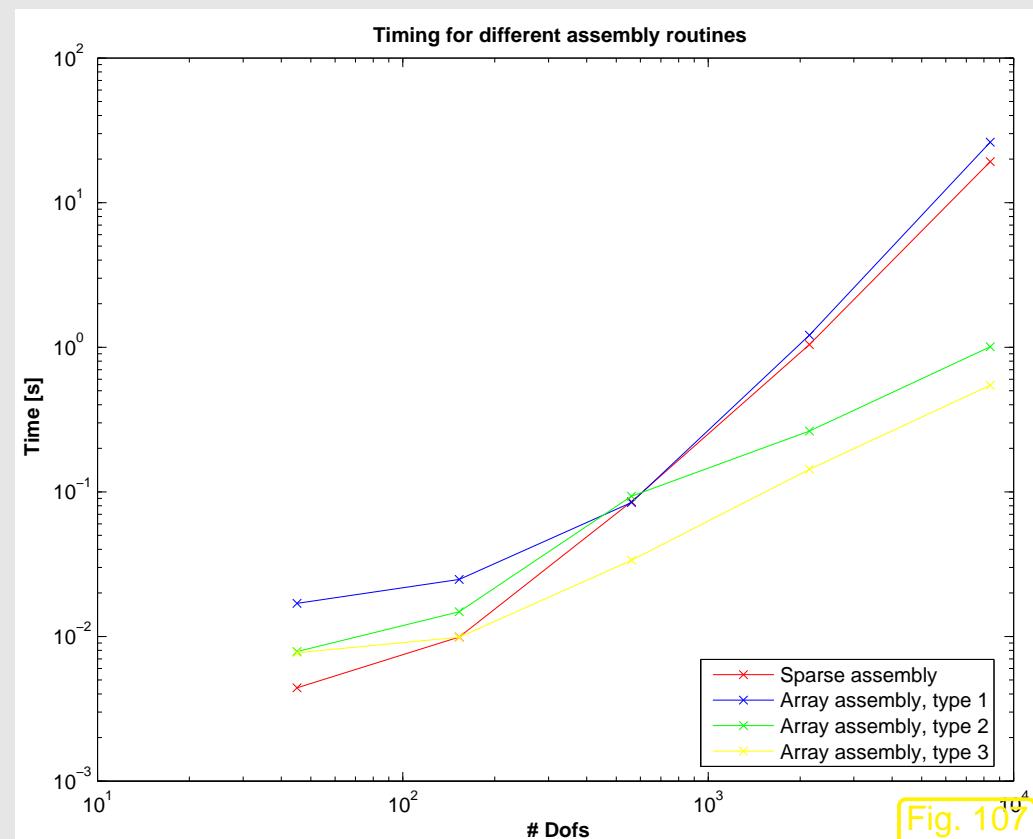
```
A(idx, idx) = A(idx, idx) + Aloc;
```

- *Array assembly I: “growing arrays”*

```
I = [ ]; J = [ ]; a = [ ];  
...  
t = idx(:, ones(length(idx), 1))';  
I = [I; t(:)];  
t = idx(:, ones(1, length(idx)));  
J = [J; t(:)];  
a = [a; Aloc(:)];
```

- *Array assembly III*

→ see code fragment above



More detailed discussion → [20] and [14, Sect. 2.6.2].

3.5.4 Local computations and quadrature

We have seen that the (global) Galerkin matrix and right hand side vector are conveniently generated by “assembling” entries of element (stiffness) matrices and element (load) vectors.

Now we study the computation of these local quantities, see also Sect. 3.2.5, 3.2.6.

First option:

analytic evaluations

We discuss bilinear form related to $-\Delta$, triangular Lagrangian finite elements of degree p , Sect. 3.4.1, Def. 3.4.1:

$$K \text{ triangle: } a_K(u, v) := \int_K \mathbf{grad} u \cdot \mathbf{grad} v \, dx \quad \blacktriangleright \quad \text{element stiffness matrix .}$$

Use **barycentric coordinate representations** of local shape functions, in 2D

$$b_K^i = \sum_{\alpha \in \mathbb{N}_0^3, |\alpha| \leq p} \kappa_\alpha \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} \lambda_3^{\alpha_3}, \quad \kappa_\alpha \in \mathbb{R}, \quad (3.5.19)$$

where λ_i are the affine linear barycentric coordinate functions (linear shape functions), see Fig. 70.

For the barycentric coordinate representation of the quadratic local shape functions see (3.4.4), for a justification of (3.5.19) consult Rem. 3.6.9.

$$\Rightarrow \text{grad } b_K^i = \sum_{\alpha \in \mathbb{N}_0^3, |\alpha| \leq p} \kappa_\alpha \left(\alpha_1 \lambda_1^{\alpha_1-1} \lambda_2^{\alpha_2} \lambda_3^{\alpha_3} \text{grad } \lambda_1 + \alpha_2 \lambda_1^{\alpha_1} \lambda_2^{\alpha_2-1} \lambda_3^{\alpha_3} \text{grad } \lambda_2 + \alpha_3 \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} \lambda_3^{\alpha_3-1} \text{grad } \lambda_3 \right). \quad (3.5.20)$$



To evaluate $\int_K \lambda_1^{\beta_1} \lambda_2^{\beta_2} \lambda_3^{\beta_3} \text{grad } \lambda_i \cdot \text{grad } \lambda_j \, d\mathbf{x}, \quad i, j \in \{1, 2, 3\}, \beta_k \in \mathbb{N}. \quad (3.5.21)$

$$\mathbf{a}^3 = (a_1^3, a_2^3)^T$$

If $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$ vertices of K (counterclockwise ordering):

$$\lambda_1(\mathbf{x}) = \frac{1}{2|K|} \left(\mathbf{x} - \begin{pmatrix} a_1^2 \\ a_2^2 \end{pmatrix} \right) \cdot \begin{pmatrix} a_2^2 - a_2^3 \\ a_1^3 - a_1^2 \end{pmatrix},$$

$$\lambda_2(\mathbf{x}) = \frac{1}{2|K|} \left(\mathbf{x} - \begin{pmatrix} a_1^3 \\ a_2^3 \end{pmatrix} \right) \cdot \begin{pmatrix} a_2^3 - a_2^1 \\ a_1^1 - a_1^3 \end{pmatrix},$$

$$\lambda_3(\mathbf{x}) = \frac{1}{2|K|} \left(\mathbf{x} - \begin{pmatrix} a_1^1 \\ a_2^1 \end{pmatrix} \right) \cdot \begin{pmatrix} a_2^1 - a_2^2 \\ a_1^2 - a_1^1 \end{pmatrix}.$$

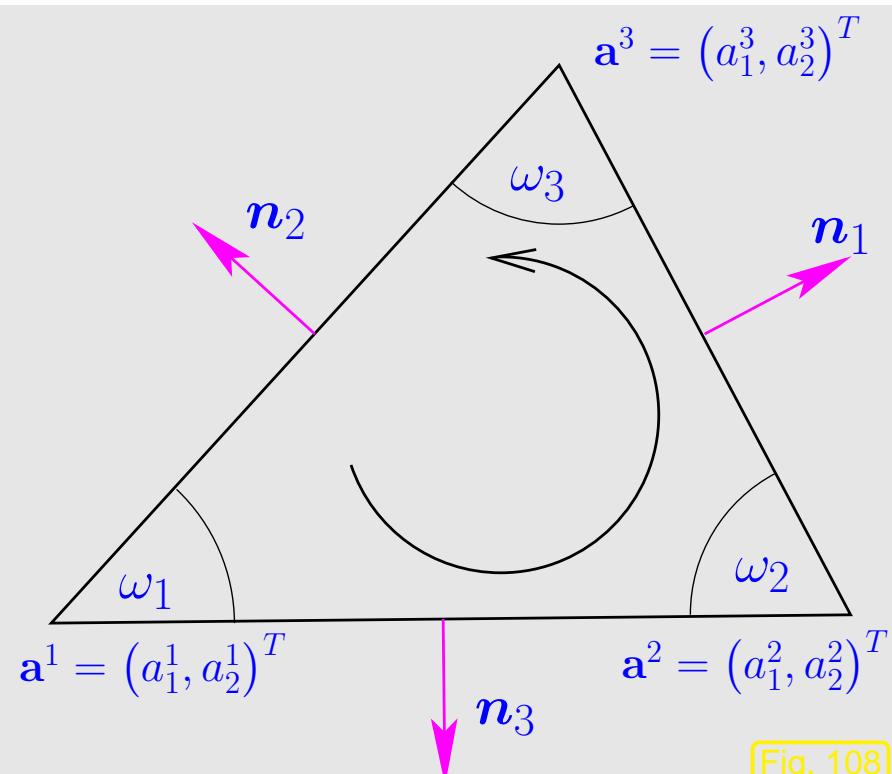


Fig. 108

$$\text{grad } \lambda_1 = \frac{1}{2|K|} \begin{pmatrix} a_2^2 - a_2^3 \\ a_1^3 - a_1^2 \end{pmatrix}, \quad \text{grad } \lambda_2 = \frac{1}{2|K|} \begin{pmatrix} a_2^3 - a_2^1 \\ a_1^1 - a_1^3 \end{pmatrix}, \quad \text{grad } \lambda_3 = \frac{1}{2|K|} \begin{pmatrix} a_2^1 - a_2^2 \\ a_1^2 - a_1^1 \end{pmatrix}. \quad (3.5.22)$$

Lemma 3.5.23 (Integration of powers of barycentric coordinate functions).

For any non-degenerate d -simplex K and $\alpha_j \in \mathbb{N}$, $j = 1, \dots, d+1$,

$$\int_K \lambda_1^{\alpha_1} \cdots \lambda_{d+1}^{\alpha_{d+1}} dx = d!|K| \frac{\alpha_1! \alpha_2! \cdots \alpha_{d+1}!}{(\alpha_1 + \alpha_2 + \cdots + \alpha_{d+1} + d)!} \quad \forall \alpha \in \mathbb{N}_0^{d+1}. \quad (3.5.24)$$

Proof for $d = 2$

Step #1: transformation $K \rightarrow$ “unit triangle” $\widehat{K} := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$,

$$\begin{aligned} \Rightarrow \int_K \lambda_1^{\beta_1} \lambda_2^{\beta_2} \lambda_3^{\beta_3} dx &= 2|K| \int_0^1 \int_0^{1-\xi_1} \xi_1^{\beta_1} \xi_2^{\beta_2} (1 - \xi_1 - \xi_2)^{\beta_3} d\xi_2 d\xi_1 \\ &\stackrel{(*)}{=} 2|K| \int_0^1 \xi_1^{\beta_1} \int_0^1 (1 - \xi_1)^{\beta_2 + \beta_3 + 1} s^{\beta_2} (1 - s)^{\beta_3} ds d\xi_1 \\ &= 2|K| \int_0^1 \xi_1^{\beta_1} (1 - \xi_1)^{\beta_2 + \beta_3 + 1} d\xi_1 \cdot B(\beta_2 + 1, \beta_3 + 1) \\ &= 2|K| B(\beta_1 + 1, \beta_2 + \beta_3 + 2) \cdot B(\beta_2 + 1, \beta_3 + 1), \end{aligned}$$

$(*) \hat{=} \text{substitution } s(1 - \xi_1) = \xi_2, \quad B(\cdot, \cdot) \hat{=} \text{Euler's beta function}$

$$B(\alpha, \beta) := \int_0^1 t^{\alpha-1} (1-t)^{\beta-1} dt, \quad 0 < \alpha, \beta < \infty.$$

Using $\Gamma(\alpha + \beta) B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)$, $\Gamma \hat{=} \text{Gamma function}$, $\Gamma(n) = (n-1)!$,

$$\Rightarrow \int_K \lambda_1^{\beta_1} \lambda_2^{\beta_2} \lambda_3^{\beta_3} d\mathbf{x} = 2|K| \cdot \frac{\Gamma(\beta_1 + 1)\Gamma(\beta_2 + 1)\Gamma(\beta_3 + 1)}{\Gamma(\beta_1 + \beta_2 + \beta_3 + 3)} \quad \square.$$

Remark. Alternative: **symbolic computing** (MAPLE, Mathematica) for local computations

Second option: **cell-based quadrature**

At this point turn the pages back to (1.5.57) and remember the use of numerical quadrature for computing the Galerkin matrix for the linear finite element method in 1D.

Local quadrature formula, cf. (3.2.13)

$$\int_{\Omega} f(\mathbf{x}) d\mathbf{x} \approx |K| \sum_{K \in \mathcal{M}} \sum_{l=1}^P \omega_l^K f(\zeta_l^K), \quad \zeta_l^K \in K, \omega_l^K \in \mathbb{R}, \quad P \in \mathbb{N}. \quad (3.5.25)$$

Terminology:

$$\omega_l^K \rightarrow \text{weights} , \quad \zeta_l^K \rightarrow \text{quadrature nodes}$$

(3.5.25) = P -point local quadrature rule

- Mandatory
- for computation of load vector (f complicated/only available in procedural form, Rem. 1.5.3,)
 - for computation of stiffness matrix, if $\alpha = \alpha(x)$ does not permit analytic integration.

Example for local quadrature rule: 2D trapezoidal rule from (3.2.13)

Guideline [14, Sect. 10.2]: only quadrature rules with positive weights are numerically stable.

How to gauge the quality of parametric local quadrature rules ? → [14, Sect. 10.3]

Quality of a parametric local quadrature rule on $K \sim \text{maximal degree of polynomials}$ (multivariate → Def. 3.3.3, or tensor product → Def. 3.3.7) on K integrated exactly by the corresponding quadrature rule on K .

Parlance: Quadrature rule exact for $\mathcal{P}_p(\mathbb{R}^d)$ \Rightarrow quadrature rule of order $p+1$
degree of exactness p

How are quadrature rules specified for the many different cells of a finite element mesh ?

Remark 3.5.26 (Affine transformation of triangles).

Definition 3.5.27 (Affine (linear) transformation).

Mapping $\Phi : \mathbb{R}^d \mapsto \mathbb{R}^d$ is **affine (linear)**, if $\Phi(\mathbf{x}) = \mathbf{F}\mathbf{x} + \boldsymbol{\tau}$ with some $\mathbf{F} \in \mathbb{R}^{d,d}$, $\boldsymbol{\tau} \in \mathbb{R}^d$.



notation: ‘unit triangle’ $\widehat{K} := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$

Lemma 3.5.28 (Affine transformation of triangles).

For any non-degenerate triangle $K \subset \mathbb{R}^2$ ($|K| > 0$) there is a unique affine transformation Φ_K , $\Phi_K(\widehat{x}) = F_K \widehat{x} + \tau_K$ (\rightarrow Def. 3.5.27), with $K = \Phi(\widehat{K})$.

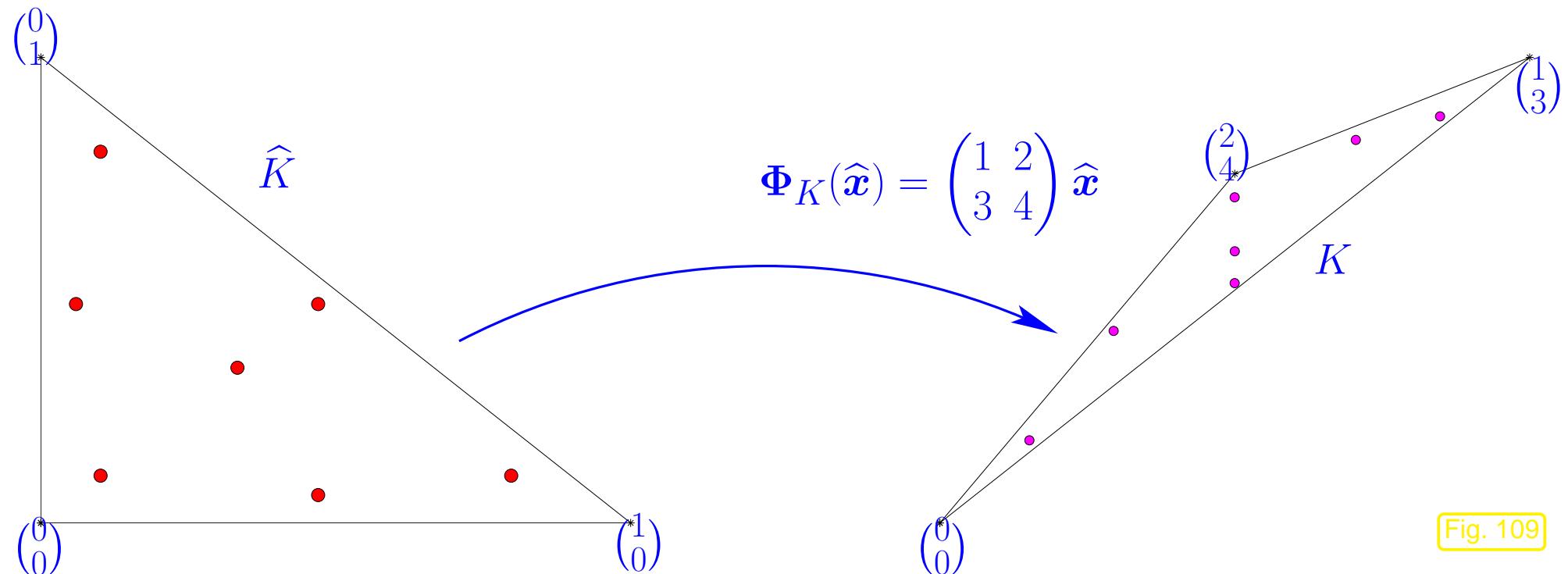


Fig. 109

Formula:

$$K = \text{convex} \left\{ \begin{pmatrix} a_1^1 \\ a_2^1 \end{pmatrix}, \begin{pmatrix} a_1^2 \\ a_2^2 \end{pmatrix}, \begin{pmatrix} a_1^3 \\ a_2^3 \end{pmatrix} \right\} \Rightarrow \Phi_K(\hat{\mathbf{x}}) = \begin{pmatrix} a_1^2 - a_1^1 & a_1^3 - a_1^1 \\ a_2^2 - a_2^1 & a_2^3 - a_2^1 \end{pmatrix} \hat{\mathbf{x}} + \begin{pmatrix} a_1^1 \\ a_2^1 \end{pmatrix}. \quad (3.5.29)$$

Note that

$$|K| = \frac{1}{2} |\det \mathbf{F}_K|.$$



Remark 3.5.30 (Transformation of local quadrature rules on triangles).

$\Phi_K(\hat{\mathbf{x}}) := \mathbf{F}_K \hat{\mathbf{x}} + \boldsymbol{\tau}_K \doteq \text{affine transformation } (\rightarrow \text{Def. 3.5.27}) \text{ mapping } \hat{K} \text{ to triangle } K, \text{ see Lemma 3.5.28.}$

By transformation formula for integrals [19, Satz 8.5.2]

$$\int_K f(\mathbf{x}) d\mathbf{x} = \int_{\hat{K}} f(\Phi_K(\hat{\mathbf{x}})) |\det \mathbf{F}_K| d\hat{\mathbf{x}}. \quad (3.5.31)$$

P -point quadrature formula on \hat{K}  P -point quadrature formula on K

$$\int_{\hat{K}} f(\hat{\mathbf{x}}) d\hat{\mathbf{x}} \approx |\hat{K}| \sum_{l=1}^P \hat{\omega}_l f(\hat{\boldsymbol{\zeta}}_l) \quad \Rightarrow \quad \int_{\Omega} f(\mathbf{x}) d\mathbf{x} \approx \sum_{K \in \mathcal{M}} |K| \sum_{l=1}^P \omega_l^K f(\boldsymbol{\zeta}_l^K) \quad (3.5.32)$$

with $\omega_l^K = \hat{\omega}_l$, $\boldsymbol{\zeta}_l^K = \Phi_K(\hat{\boldsymbol{\zeta}}_l)$.

- Only quadrature formula (3.5.25) on unit triangle \hat{K} needs to be specified!
(The same applies to tetrahedra, where affine mappings for $d = 3$ are used.)

Since the space $\mathcal{P}_p(\mathbb{R}^d)$ is *invariant* under affine mappings,

$$q \in \mathcal{P}_p(\mathbb{R}^d) \Rightarrow \hat{\mathbf{x}} \mapsto q(\Phi(\hat{\mathbf{x}})) \in \mathcal{P}_p(\mathbb{R}^d) \quad \text{for any affine transformation } \Phi, \quad (3.5.33)$$

the orders of the quadrature rules on the left and right hand side of (3.5.31) agree.



Example 3.5.34 (Useful quadrature rules on triangles). → [1, Sect. ??]

Specification of quadrature rule for “unit triangle” $\widehat{K} := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$.

Quadrature rules described by pairs $(\widehat{\omega}_1, \widehat{\zeta}_1), \dots, (\widehat{\omega}_P, \widehat{\zeta}_P)$, $P \in \mathbb{N}$.

- Quadrature rule of order 2 (exact for $\mathcal{P}_1(\widehat{K})$)

$$\left\{ \left(\frac{1}{3}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right), \left(\frac{1}{3}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right), \left(\frac{1}{3}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) \right\}. \quad (3.5.35)$$

- Quadrature rule of order 3 (exact for $\mathcal{P}_2(\widehat{K})$)

$$\left\{ \left(\frac{1}{3}, \begin{pmatrix} 1/2 \\ 0 \end{pmatrix} \right), \left(\frac{1}{3}, \begin{pmatrix} 0 \\ 1/2 \end{pmatrix} \right), \left(\frac{1}{3}, \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \right) \right\}. \quad (3.5.36)$$

- One-point quadrature rule of order 2 (exact for $\mathcal{P}_1(\widehat{K})$)

$$\left\{ \left(1, \begin{pmatrix} 1/3 \\ 1/3 \end{pmatrix} \right) \right\}. \quad (3.5.37)$$

- Quadrature rule of order 6 (exact for $\mathcal{P}_5(\widehat{K})$)

$$\left\{ \left(\frac{9}{40}, \begin{pmatrix} 1/3 \\ 1/3 \end{pmatrix} \right), \left(\frac{155 + \sqrt{15}}{1200}, \begin{pmatrix} 6+\sqrt{15}/21 \\ 6+\sqrt{15}/21 \end{pmatrix} \right), \left(\frac{155 + \sqrt{15}}{1200}, \begin{pmatrix} 9-2\sqrt{15}/21 \\ 6+\sqrt{15}/21 \end{pmatrix} \right), \right. \\ \left(\frac{155 + \sqrt{15}}{1200}, \begin{pmatrix} 6+\sqrt{15}/21 \\ 9-2\sqrt{15}/21 \end{pmatrix} \right), \left(\frac{155 - \sqrt{15}}{1200}, \begin{pmatrix} 6-\sqrt{15}/21 \\ 9+2\sqrt{15}/21 \end{pmatrix} \right), \\ \left. \left(\frac{155 - \sqrt{15}}{1200}, \begin{pmatrix} 9+2\sqrt{15}/21 \\ 6-\sqrt{15}/21 \end{pmatrix} \right), \left(\frac{155 - \sqrt{15}}{1200}, \begin{pmatrix} 6-\sqrt{15}/21 \\ 6-\sqrt{15}/21 \end{pmatrix} \right) \right\} \quad (3.5.38)$$

In [9]: quadrature rules up to order $p = 21$ with $P \leq 1/6p(p + 1) + 5$

Remark 3.5.39 (Numerical quadrature in LehrFEM). → [1, Sect. 3]

Routines return P -point quadrature formulas for

$$\hat{K} = \begin{cases} \text{unit triangle} & \text{convex } \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\} \\ \text{unit square} & \text{convex } \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\} \end{cases}$$

for triangular cell,
for rectangular cell,

in MATLAB *structure* `QuadRule` with fields

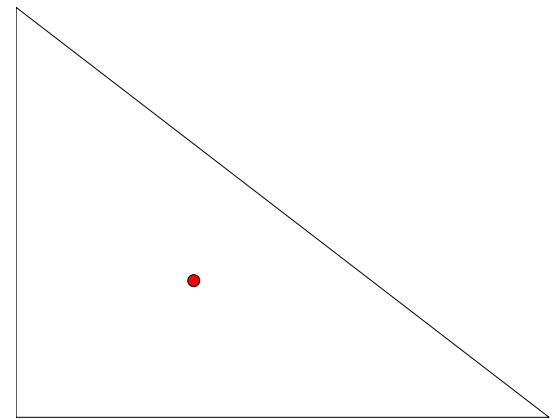
`QuadRule.w`: weights $\hat{\omega}_l$ of quadrature rule on \hat{K} ,

`QuadRule.x`: coordinates of nodes $\hat{\zeta}_l \in \hat{K}$ of quadrature rule on \hat{K}

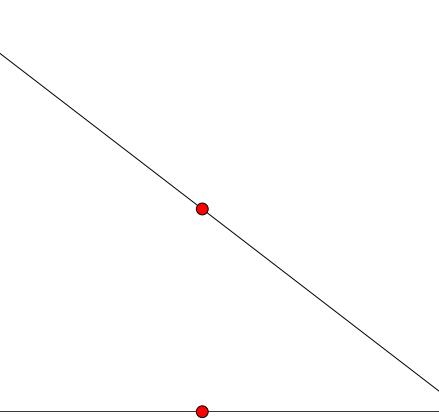
For triangles: `QuadRule = PnOq()`, $\hat{=}$ n -point quadrature of order q

Location of quadrature nodes $\hat{\zeta}_l$ in unit triangle \hat{K} :

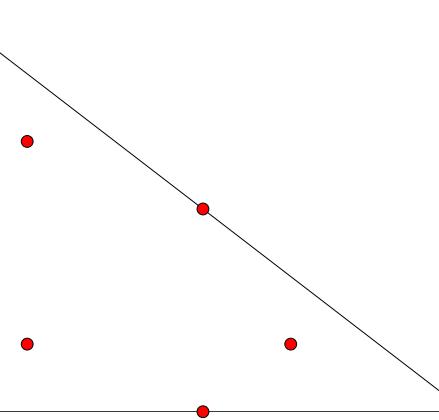
Quadrature rule P1O2



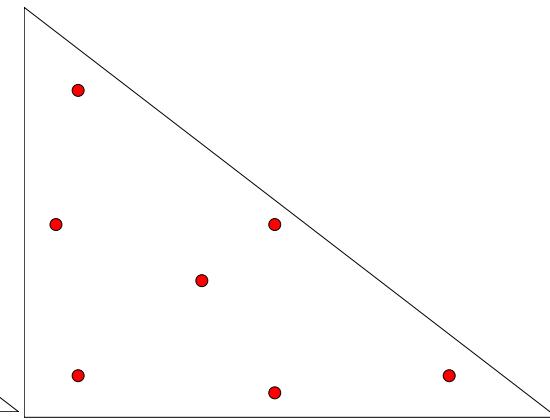
Quadrature rule P3O3



Quadrature rule P6O4



Quadrature rule P7O6





Example 3.5.40 (Local quadrature rules on quadrilaterals).

If K quadrilateral $\Rightarrow \widehat{K} := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$ (unit square).

On \widehat{K} :

tensor product construction:

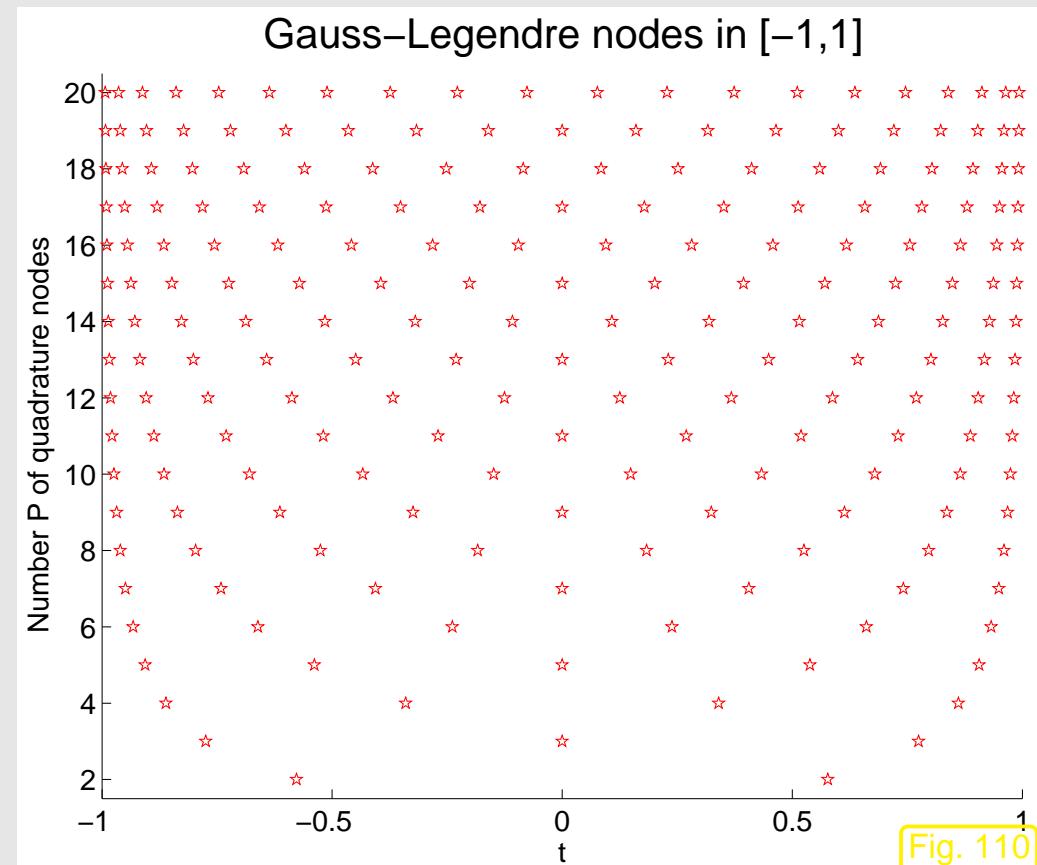
If $\{(\omega_1, \zeta_1), \dots, (\omega_P, \zeta_P)\}$, $P \in \mathbb{N}$, quadrature rule on the interval $]0, 1[$, exact for \mathcal{P}_p $]0, 1[$, then

$$\left\{ \begin{array}{ccc} (\omega_1^2, \binom{\zeta_1}{\zeta_1}) & \cdots & (\omega_1 \omega_P, \binom{\zeta_1}{\zeta_P}) \\ \vdots & & \vdots \\ (\omega_1 \omega_P, \binom{\zeta_P}{\zeta_1}) & \cdots & (\omega_P^2, \binom{\zeta_P}{\zeta_P}) \end{array} \right\}$$

provides a quadrature rule on the unit square \widehat{K} , exact for $\mathcal{Q}_p(\widehat{K})$.

Quadrature rules on $]0, 1[$ (\rightarrow [14, Ch. 10]):

- classical **Newton-Cotes formulas** (equidistant quadrature nodes).
- **Gauss-Legendre quadrature rules**, exact for $\mathcal{P}_{2P}(]0, 1[)$ using only P nodes.
- **Gauss-Lobatto quadrature rules**: P nodes including $\{0, 1\}$, exact for $\mathcal{P}_{2P-1}(]0, 1[)$.



3.5.5 Incorporation of essential boundary conditions

Recall variational formulation of *non-homogeneous* Dirichlet boundary value problem from Ex. 2.8.1:

$$\begin{aligned} u \in H^1(\Omega) : \quad & \int_{\Omega} \kappa(\boldsymbol{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\boldsymbol{x} = \int_{\Omega} f v \, d\boldsymbol{x} \quad \forall v \in H_0^1(\Omega) . \\ u = g \text{ on } \partial\Omega : \quad & \end{aligned} \quad (2.8.4)$$



$$-\operatorname{div}(\kappa(\boldsymbol{x}) \operatorname{grad} u) = f \quad \text{in } \Omega , \quad u = g \quad \text{on } \partial\Omega ,$$

with (admissible → Rem. 2.9.4) Dirichlet data $g \in C^0(\partial\Omega)$.

Recall from Sect. 2.9:

Dirichlet b.c. = essential boundary conditions
(built into trial space)

Remember offset function technique, see (1.3.14) and Sect. 2.1.3:

$$\begin{aligned} w \in H_0^1(\Omega) : \quad & \int_{\Omega} \kappa(\boldsymbol{x}) \operatorname{grad} w \cdot \operatorname{grad} v \, d\boldsymbol{x} \\ (2.8.4) \Leftrightarrow u = u_0 + w , \quad & = \int_{\Omega} -\kappa(\boldsymbol{x}) \operatorname{grad} u_0 \cdot \operatorname{grad} v - f v \, d\boldsymbol{x} \quad \forall v \in H_0^1(\Omega) , \end{aligned} \quad (3.5.41)$$

where

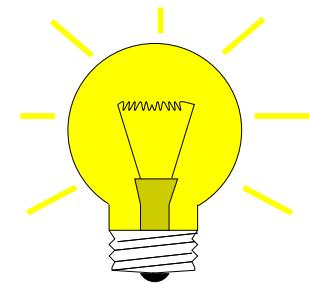
$u_0 = g \text{ on } \partial\Omega$

Adapt this to finite element Galerkin discretization by generalizing the 1D example Rem. 1.5.60 to $d = 2, 3$:

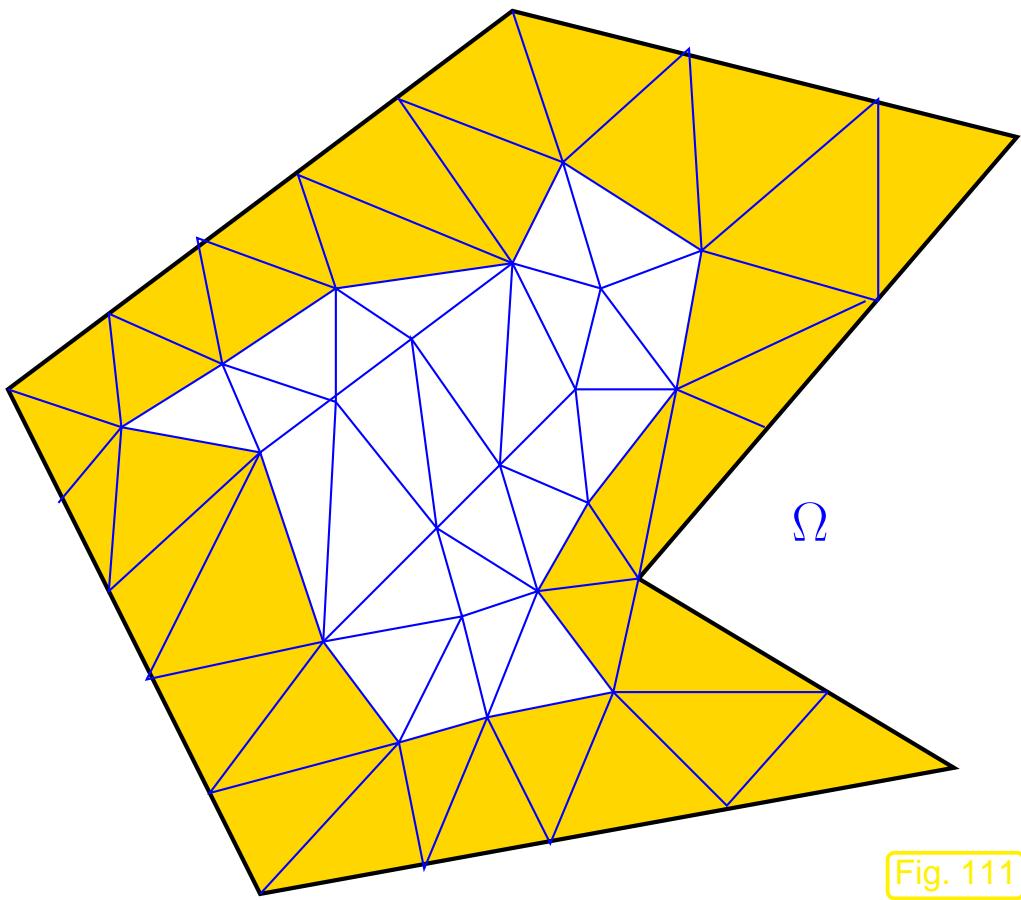
Remember: we already know finite element subspaces $V_{0,N} := \mathcal{S}_{p,0}^0(\mathcal{M}) \subset H_0^1(\Omega)$, see Rem. 3.4.12.

Idea from Rem. 1.5.60:

use offset function $u_0 \in V_N := \mathcal{S}_p^0(\mathcal{M})$
locally supported near the boundary:



$$\text{supp}(u_0) \subset \bigcup\{K \in \mathcal{M}: \overline{K} \cap \partial\Omega \neq \emptyset\} . \quad (3.5.42)$$



◁ Maximal support of u_0 on triangular mesh.

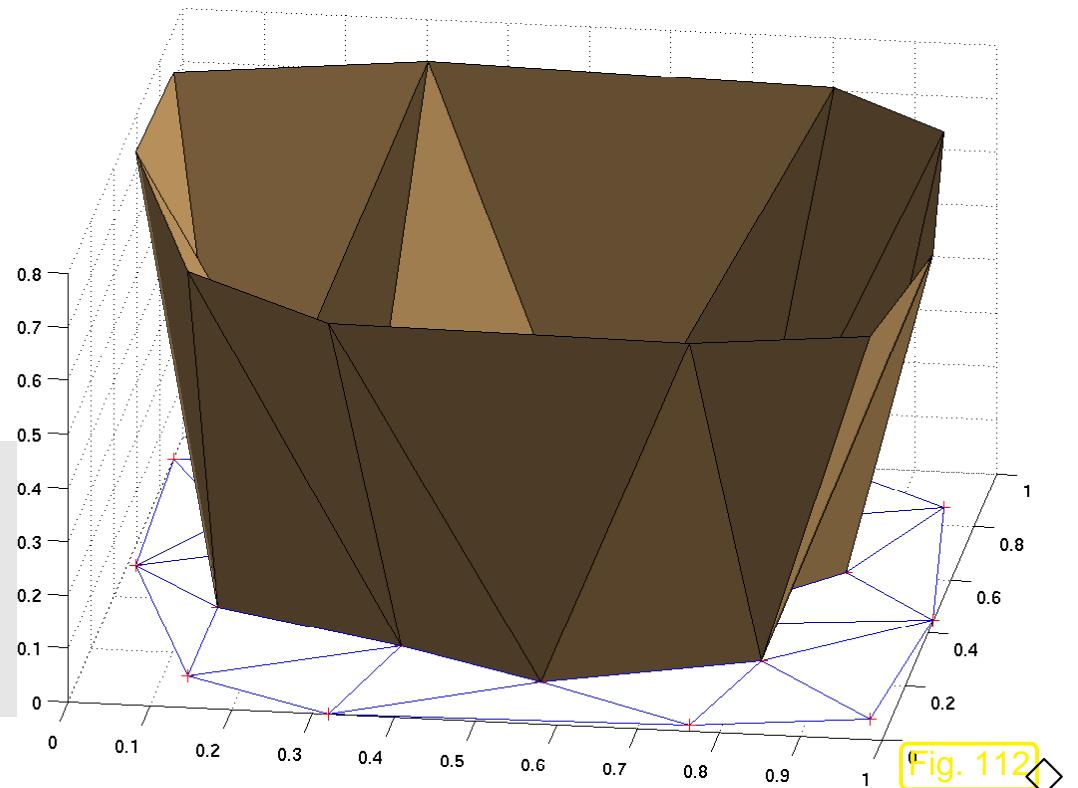
Fig. 111

Example 3.5.43 (offset functions for linear Lagrangian FE).

For Dirichlet data $g \in C^0(\partial\Omega)$

$$u_0 = \sum_{x \in \mathcal{V}(\mathcal{M}) \cap \partial\Omega} g(x) b_N^x \quad (3.5.44)$$

b_N^x $\hat{=}$ tent function associated with node $x \in \mathcal{V}(\mathcal{M})$, cf. Sect. 3.2.3. (3.5.44) generalizes (1.5.61) to 2D.



Remark 3.5.45 (Approximate Dirichlet boundary conditions).

Be aware that for the choice (3.5.44)

$$u_0 \neq g \quad \text{on } \partial\Omega .$$

Rather, \underline{u}_0 is a *piecewise linear interpolant* of the Dirichlet data $\underline{g} \in C^0(\partial\Omega)$. Therefore, another *approximation* comes into play when enforcing Dirichlet boundary conditions by means of piecewise polynomial offset functions.

Example 3.5.46 (Implementation of non-homogeneous Dirichlet b.c. for linear FE).

Consider (2.8.4) and assume the following ordering of the nodal basis functions, see Fig. 65

$$\begin{aligned}\mathfrak{B}_0 &:= \{b_N^1, \dots, b_N^N\} &\hat{=}& \text{ nodal basis of } \mathcal{S}_{1,0}^0(\mathcal{M}), \\ &&& \text{(tent functions associated with interior nodes)} \\ \mathfrak{B} &:= \mathfrak{B}_0 \cup \{b_N^{N+1}, \dots, b_N^M\} &\hat{=}& \text{ nodal basis of } \mathcal{S}_1^0(\mathcal{M}) \\ &&& \text{(extra basis functions associated with nodes } \in \partial\Omega\text{).}\end{aligned}$$

Note: $M = \#\mathcal{V}(\mathcal{M})$, $N = \#\{\mathbf{x} \in \mathcal{V}(\mathcal{M}), \mathbf{x} \notin \partial\Omega\}$ (no. of interior nodes)

$$\begin{aligned}\mathbf{A}_0 &\in \mathbb{R}^{N,N} &\hat{=}& \text{ Galerkin matrix for discrete trial/test space } \mathcal{S}_{1,0}^0(\mathcal{M}), \\ \mathbf{A} &\in \mathbb{R}^{M,M} &\hat{=}& \text{ Galerkin matrix for discrete trial/test space } \mathcal{S}_1^0(\mathcal{M}).\end{aligned}$$

► $\mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_{0\partial} \\ \mathbf{A}_{0\partial}^T & \mathbf{A}_{\partial\partial} \end{pmatrix}, \quad \mathbf{A}_{0\partial} \in \mathbb{R}^{N,M-N}, \quad \mathbf{A}_{\partial\partial} \in \mathbb{R}^{M-N,M-N}. \quad (3.5.47)$

If $u_0 \in \mathcal{S}_1^0(\mathcal{M})$ is chosen according to (3.5.44), then

$$u_0 \in \text{Span} \left\{ b_N^{N+1}, \dots, b_N^M \right\} \Leftrightarrow u_0 = \sum_{j=N+1}^M \gamma_{j-N} b_N^j,$$

which means that the coefficient vector $\vec{\nu}$ of the finite element approximation $w_N \in \mathcal{S}_{1,0}^0(\mathcal{M})$ of $w \in H_0^1(\Omega)$ from (3.5.41) solves the linear system of equations

$$\boxed{\mathbf{A}_0 \vec{\nu} = \vec{\varphi} - \mathbf{A}_{0\partial} \vec{\gamma}}. \quad (3.5.48)$$

➤ Non-homogeneous Dirichlet boundary data are taken into account through a **modified right hand side vector**.

Alternative consideration leading to (3.5.48):

- ① First ignore essential boundary conditions and assemble the linear system of equations arising from the discretization of a on the (larger) FE space $\mathcal{S}_1^0(\mathcal{M})$:

$$\begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_{0\partial} \\ \mathbf{A}_{0\partial}^T & \mathbf{A}_{\partial\partial} \end{pmatrix} \begin{pmatrix} \vec{\mu}_0 \\ \vec{\mu}_\partial \end{pmatrix} = \begin{pmatrix} \vec{\varphi} \\ \vec{\varphi}_\partial \end{pmatrix}. \quad (3.5.49)$$

Here, $\vec{\mu}_0 \hat{=} \text{coefficients for } \textit{interior} \text{ basis functions } b_N^1, \dots, b_N^N$

$\vec{\mu}_\partial \hat{=} \text{coefficient for basis functions } b_N^{N+1}, \dots, b_N^M$ for basis functions associated with nodes $\in \partial\Omega$.

- ② We realize that the coefficient vector of (3.5.49) is that of a FE approximation of u

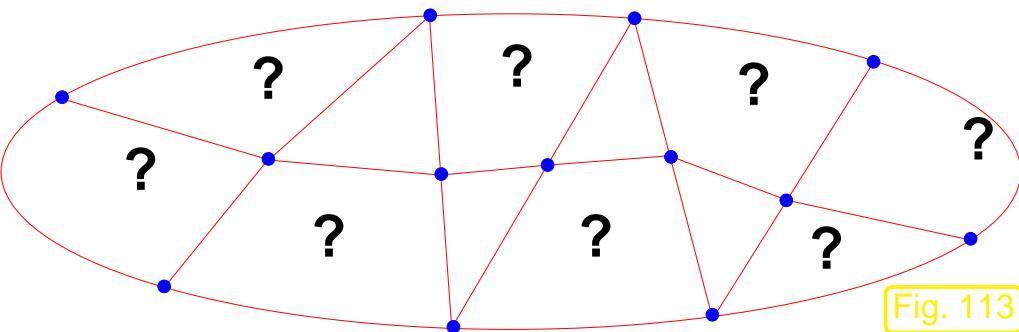


$\vec{\mu}_\partial$ known = values of g at boundary nodes: $\boxed{\vec{\mu}_\partial = \vec{\gamma}}$

- ③ Moving known quantities in (3.5.49) to the right hand side yields (3.5.48).



3.6 Parametric finite elements



▷ 2D hybrid mesh \mathcal{M} with (curvilinear) triangles and quadrilaterals

How to build $\mathcal{S}_1^0(\mathcal{M})$?

3.6.1 Affine equivalence

Recall Lemma 3.5.28: affine transformation of triangles (3.5.29)

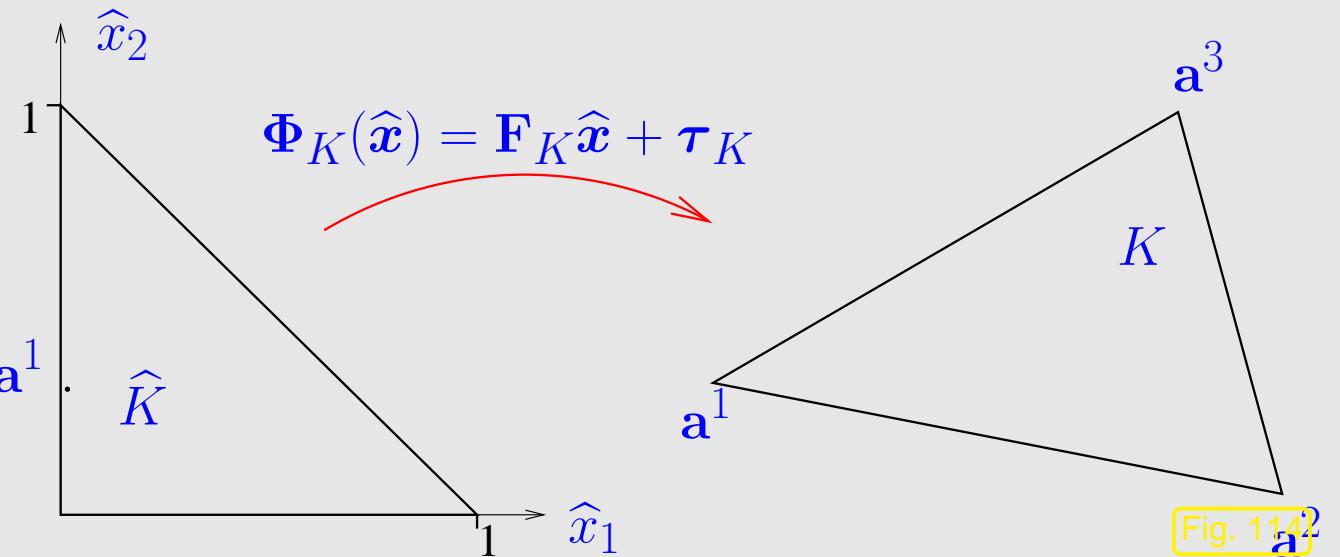


All cells of a triangular mesh are affine images of “unit triangle” \hat{K}

“Unit triangle”: $\hat{K} = \left\langle \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\rangle$

For $K = \text{convex } \{\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3\}$:

$$\mathbf{F}_K = \begin{pmatrix} a_1^2 - a_1^1 & a_1^3 - a_1^1 \\ a_2^2 - a_2^1 & a_2^3 - a_2^1 \end{pmatrix}, \quad \boldsymbol{\tau}_K = \mathbf{a}^1$$



Remark 3.6.1 (Pullback of functions).

In a natural way, a transformation of domains induces a transformation of the functions defined on them:

Definition 3.6.2 (Pullback).

Given domains $\Omega, \hat{\Omega} \subset \mathbb{R}^d$ and a bijective mapping $\Phi : \hat{\Omega} \mapsto \Omega$, the **pullback** $\Phi^* u : \hat{\Omega} \mapsto \mathbb{R}$ of a function $u : \Omega \mapsto \mathbb{R}$ is a function on $\hat{\Omega}$ defined by

$$(\Phi^* u)(\hat{x}) := u(\Phi(\hat{x})) , \quad \hat{x} \in \hat{\Omega} .$$

- Implicitly, we used the pullback of integrands when defining quadrature rules through transformation, see (3.5.31).
- Obviously, the pullback Φ^* induces a *linear mapping* between spaces of functions on Ω and $\hat{\Omega}$, respectively.

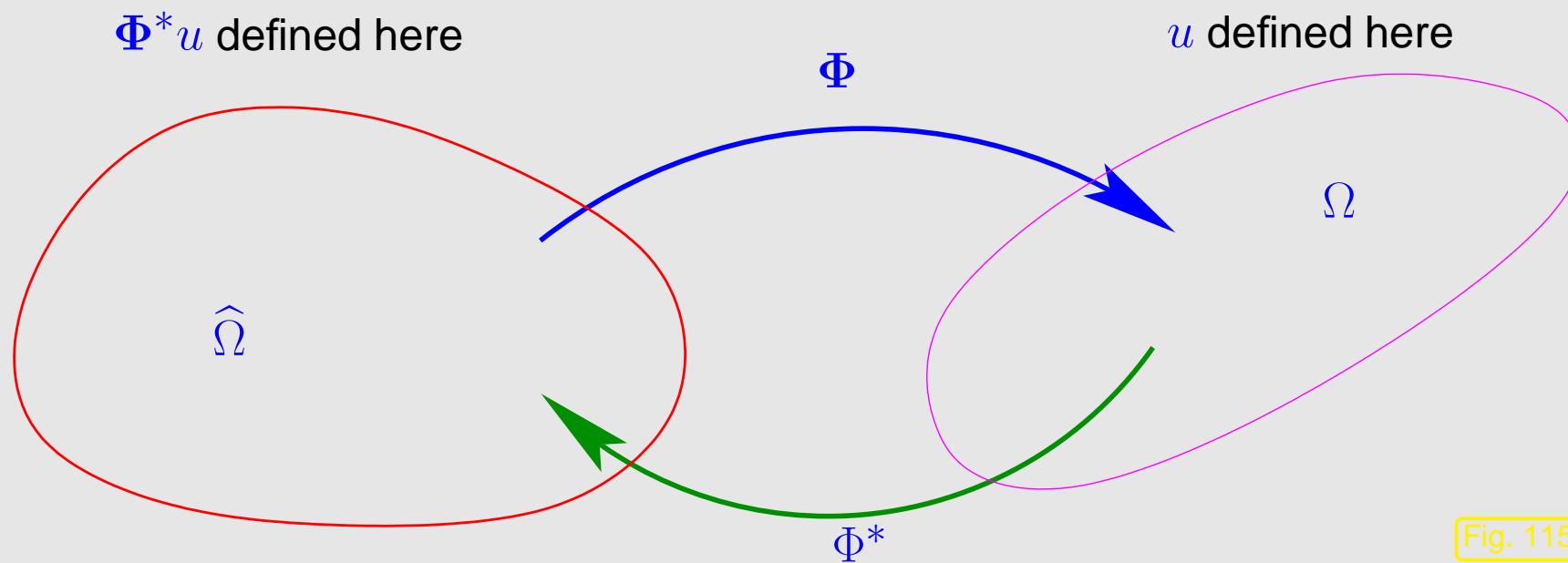


Fig. 115

In the context of numerical quadrature we made the observation, cf. (3.5.33):

Lemma 3.6.3 (Preservation of polynomials under affine pullback).

If $\Phi : \mathbb{R}^d \mapsto \mathbb{R}^d$ is an affine (linear) transformation (\rightarrow Def. 3.5.27), then

$$\Phi^*(\mathcal{P}_p(\mathbb{R}^d)) = \mathcal{P}_p(\mathbb{R}^d) \quad \text{and} \quad \Phi^*(\mathcal{Q}_p(\mathbb{R}^d)) = \mathcal{Q}_p(\mathbb{R}^d) .$$

In fact, Lemma 3.5.28 reveals another reason for the preference for polynomials in building discrete Galerkin spaces.

Proof. (of Lemma 3.5.28)

Since the pullback is linear, we only need to study its action on the (monomial) basis $\mathbf{x} \mapsto \mathbf{x}^\alpha$, $\alpha \in \mathbb{N}_0^d$ of $\mathcal{P}_p(\mathbb{R}^d)$, see Def. 3.3.3 and the explanations on multi-index notation (3.3.4).

Then resort to induction w.r.t. degree p .

$$\Phi_K^*(\mathbf{x}^\alpha) = \Phi_K^*(x_1) \cdot \underbrace{\Phi_K^*(\mathbf{x}^{\alpha'})}_{\in \mathcal{P}_{p-1}(\mathbb{R}^d)} = \underbrace{\left(\sum_{l=1}^d (\mathbf{F})_{1l} \hat{x}_l + \tau_1 \right)}_{\in \mathcal{P}_1(\mathbb{R}^d)} \cdot \underbrace{\Phi_K^*(\mathbf{x}^{\alpha'})}_{\in \mathcal{P}_{p-1}(\mathbb{R}^d)} \in \mathcal{P}_p(\mathbb{R}^d) ,$$

with $\alpha' := (\alpha_1 - 1, \alpha_2, \dots, \alpha_d)$, where we assumed $\alpha_1 > 0$. Here, we have used the induction hypothesis to conclude $\Phi_K^*(x^{\alpha'}) \in \mathcal{P}_{p-1}(\mathbb{R}^d)$. □

A simple observation:

Consider $\mathcal{S}_1^0(\mathcal{M})$, triangle $K \in \mathcal{M}$, unit triangle \hat{K} , affine mapping $\Phi_K : \hat{K} \mapsto K$

- b_K^1, b_K^2, b_K^3 (standard) local shape functions on K , → Ex. 3.3.13
- $\hat{b}^1, \hat{b}^2, \hat{b}^3$ (standard) local shape functions on \hat{K} ,

$$\hat{b}^i = \Phi_K^* b_K^i \Leftrightarrow \hat{b}^i(\hat{x}) = b_K^i(x), \quad x = \Phi_K(\hat{x}) \quad (3.6.4)$$

Of course, we assume that Φ_K respects the local numbering of the vertices of \hat{K} and K .

The proof of (3.6.4) is straightforward: both $\Phi_K^* b_K^i$ (by Lemma 3.6.3) and \hat{b}^i are (affine) linear functions that attain the same values at the vertices of \hat{K} . Hence, they have to agree.

Proof. (of (3.6.4)) Recall the definition of global shape functions and also local shape functions for $\mathcal{S}_p^0(\mathcal{M})$, $p \in \mathbb{N}$, by means of the conditions (3.4.3) at , see Ex. 3.4.2 for $p = 2$. \square

Note: we already used the definition of basis functions through basis functions on the “reference cell” $[0, 1]$ and affine pullback in 1D, see Rem. 1.5.30

Now write $\mathbf{p}_K^i \hat{=} (\text{local}) \text{ interpolation nodes on triangle } K$,
 $\widehat{\mathbf{p}}^i \hat{=} (\text{local}) \text{ interpolation nodes on unit triangle } \widehat{K}$.

Observe: Assuming a matching numbering $\mathbf{p}_K^i = \Phi_K(\widehat{\mathbf{p}}^i)$. where $\Phi_K : \widehat{K} \mapsto K$ is the unique affine transformation mapping \widehat{K} onto K , see (3.5.29).

This is clear for $p = 2$, because affine transformations take midpoints of edges to midpoints of edges. The same applies to the interpolation nodes for higher degree Lagrangian finite elements defined in Ex. 3.4.5.

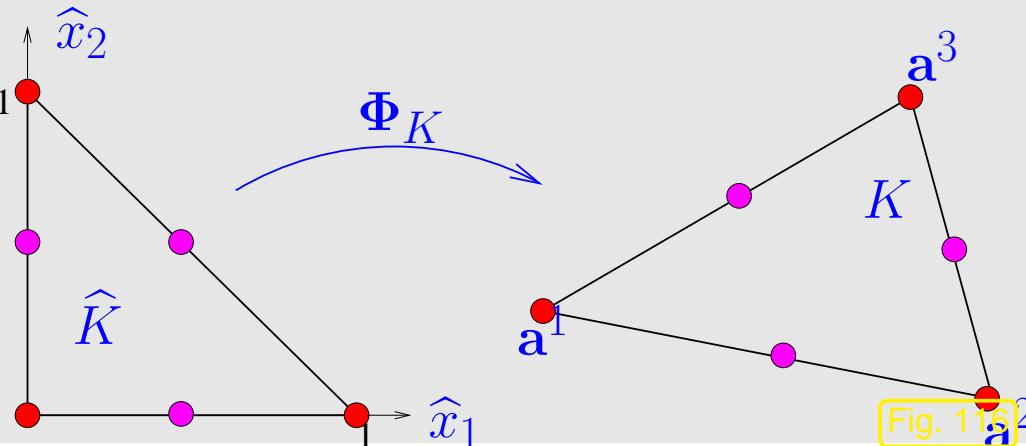


Fig. 11.6.2

The local shape functions $b_K^i \in \mathcal{P}_p(\mathbb{R}^d)$, $\widehat{b}^i \in \mathcal{P}_p(\mathbb{R}^d)$, $i = 1, \dots, Q$, are *uniquely defined* by the interpolation conditions

$$b_K^i(\mathbf{p}_K^j) = \delta_{ij} \quad , \quad \widehat{b}^i(\widehat{\mathbf{p}}^j) = \delta_{ij} . \quad (3.6.5)$$

Together with $\mathbf{p}_K^i = \Phi_K(\widehat{\mathbf{p}}^i)$ this shows that $\Phi_K^* b_K^i$ satisfies the interpolation conditions (3.6.5) on \widehat{K} and, thus, has to agree with \widehat{b}^i . \square

Terminology: finite element spaces satisfying (3.6.4) are called **affine equivalent**

Remark 3.6.6 (Evaluation of local shape functions at quadrature points).

We consider Lagrangian finite element spaces on a simplicial mesh \mathcal{M} .

Recall from Sect. 3.5.4: definition (3.5.32) of local quadrature formulas via “unit simplex”.

In particular:

quadrature nodes on K : $\zeta_l^K = \Phi_K(\hat{\zeta}_l)$

$$b_K^i(\zeta_l^K) \stackrel{\text{Def. 3.6.2}}{=} \Phi_K^*(b_K^i)(\hat{\zeta}^l) \stackrel{(3.6.4)}{=} \hat{b}^i(\hat{\zeta}^l) \quad \text{independent of } K ! . \quad (3.6.7)$$

$$\int_K F(b_K^i, b_K^j) dx \approx |K| \sum_{l=1}^P \omega_l F(\hat{b}^i(\zeta_l), \hat{b}^j(\zeta_l)) , \quad (3.6.8)$$

for any function $F : \mathbb{R}^2 \mapsto \mathbb{R}$.

➤ Precompute $\hat{b}^i(\zeta_l)$, $i = 1, \dots, Q$, $l = 1, \dots, P$ and store the values in a table!



Remark 3.6.9 (Barycentric representation of local shape functions).

We consider Lagrangian finite element spaces on a simplicial mesh \mathcal{M} .

3.6

(3.4.4): formulas for local shape functions for $\mathcal{S}_2^0(\mathcal{M})$ ($d = 2$) in terms of barycentric coordinate functions λ_i , $i = 1, 2, 3$. Is this coincidence? Does (3.5.19) hold for any (simplicial) Lagrangian finite element space?

YES!

$$\begin{aligned} b_K^i(\mathbf{x}) &\stackrel{(3.6.4)}{=} (\Phi_K^{-1})^* \left(\hat{\mathbf{x}} \mapsto \hat{b}^i(\hat{x}_1, \hat{x}_2) \right) \\ &= \hat{b}^i((\Phi_K^{-1})^*(\hat{\lambda}_2)(\hat{\mathbf{x}}), (\Phi_K^{-1})^*(\hat{\lambda}_3)(\hat{\mathbf{x}})) = \hat{b}^i(\lambda_2(\mathbf{x}), \lambda_3(\mathbf{x})) \end{aligned}$$

where $\lambda_2(\hat{\mathbf{x}}) = \hat{x}_1$, $\lambda_3(\hat{\mathbf{x}}) = \hat{x}_2$, $\lambda_1(\hat{\mathbf{x}}) = 1 - \hat{x}_1 - \hat{x}_2 \hat{=}$ barycentric coordinate functions on \hat{K} , see Ex. 3.3.13,

$\lambda_i \hat{=}$ barycentric coordinate functions on triangle K , see Fig. 70,

$\Phi_K \hat{=}$ affine transformation (\rightarrow Def. 3.5.27), $\Phi_K(\hat{K}) = K$, see (3.5.29).

➤ By the chain rule:

$$\mathbf{grad} b_K^i(\mathbf{x}) = \frac{\partial \hat{b}^i}{\partial \hat{x}_1}(\hat{\mathbf{x}}) \mathbf{grad} \lambda_2 + \frac{\partial \hat{b}^i}{\partial \hat{x}_2}(\hat{\mathbf{x}}) \mathbf{grad} \lambda_3, \quad \mathbf{x} = \Phi_K(\hat{\mathbf{x}}).$$

This formula is convenient, because $\mathbf{grad} \lambda_i \equiv \text{const}$, see (3.5.22).

This facilitates the computation of element (stiffness) matrices for 2nd-order elliptic problems in variational form: when using a quadrature formula according to (3.5.32)

$$\int_K (\alpha(x) \operatorname{grad} b_K^i) \cdot \operatorname{grad} b_K^j dx$$

$$\approx |K| \sum_{l=1}^{P_K} \omega_l \left(\begin{pmatrix} \frac{\partial \hat{b}^i}{\partial \hat{x}_1}(\hat{\zeta}_l) \\ \frac{\partial \hat{b}^i}{\partial \hat{x}_2}(\hat{\zeta}_l) \end{pmatrix}^T \begin{pmatrix} \operatorname{grad} \lambda_1 \cdot \operatorname{grad} \lambda_1 & \operatorname{grad} \lambda_1 \cdot \operatorname{grad} \lambda_2 \\ \operatorname{grad} \lambda_1 \cdot \operatorname{grad} \lambda_2 & \operatorname{grad} \lambda_2 \cdot \operatorname{grad} \lambda_2 \end{pmatrix} \begin{pmatrix} \frac{\partial \hat{b}^j}{\partial \hat{x}_1}(\hat{\zeta}_l) \\ \frac{\partial \hat{b}^j}{\partial \hat{x}_2}(\hat{\zeta}_l) \end{pmatrix} \right)$$

This is very interesting, because

- the values $\frac{\partial \hat{b}^i}{\partial \hat{x}_1}(\hat{\zeta}_l)$ can be *precomputed*,
- simple expressions for $\operatorname{grad} \lambda_i \cdot \operatorname{grad} \lambda_j$ are available, see Sect. 3.2.5.

3.6.2 Example: Quadrilateral Lagrangian finite elements

So far, see Sect. 3.3.3 and (3.3.11), we have adopted the perspective

$$\text{global shape functions} \xrightarrow{\text{Restriction to element}} \text{local shape functions}$$

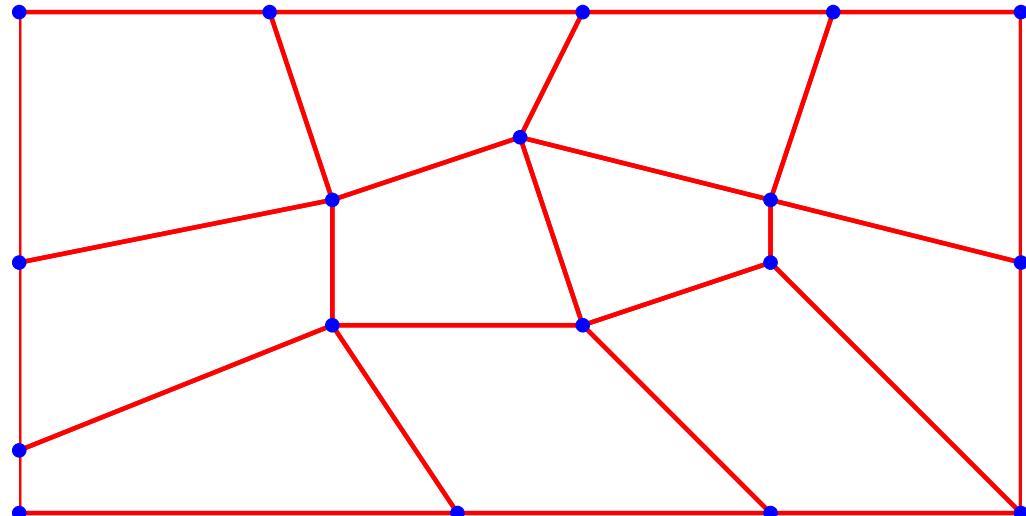
Now we reverse this construction

$$\text{local shape functions} \xrightarrow{\text{"glueing"}} \text{global shape functions} \quad (3.6.10)$$

In fact, when building the global basis functions for quadratic Lagrangian finite elements we already proceeded this way, see Ex. 3.4.2. Fig. 92 lucidly conveys what is meant by “glueing”.

Be aware that the possibility to achieve a continuous global basis function by glueing together local shape function on adjacent cells, entails a judicious choice of the local shape functions.

This section will demonstrate how the policy (3.6.10) together with the formula (3.6.4) will enable us to extend Lagrangian finite element beyond the meshes discussed in Sect. 3.4.



◁ quadrilateral mesh \mathcal{M} in 2D

What is “ $\mathcal{S}_1^0(\mathcal{M})$ ”?

Clear: If K is a rectangle, \hat{K} the unit square, then there is a unique affine transformation Φ_K (\rightarrow Def. 3.5.27) with $K = \Phi_K(\hat{K})$.

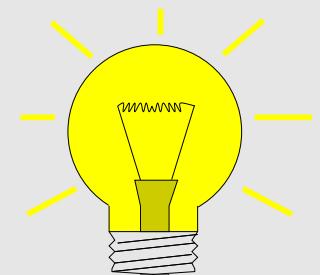
In this case (3.6.4) holds for the local shape functions of bilinear Lagrangian finite elements from Ex. 3.4.6 (and all tensor product Lagrangian finite elements introduced in Sect. 3.4.2)

Idea:

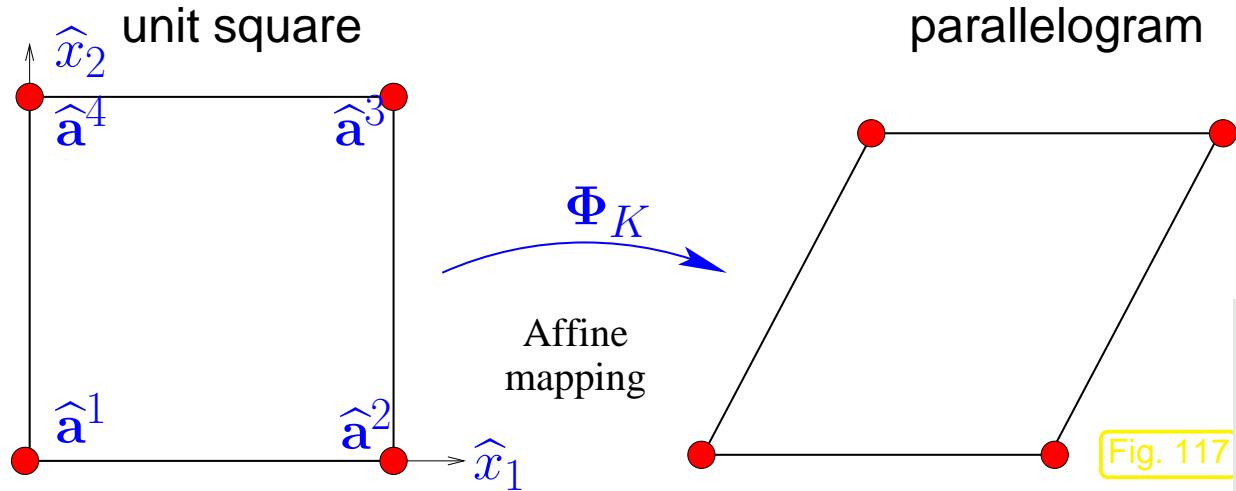
- local shape functions $\xrightarrow{\text{"glueing"}}$ global shape functions
- Build local shape functions by “inverse pullback”

$$b_K^i = (\Phi_K^{-1})^* \hat{b}^i , \quad (3.6.11)$$

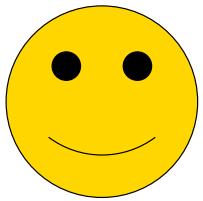
where $\{\hat{b}^i\}_{i=1}^Q \hat{=} \text{set of shape functions on reference element } \hat{K}$.



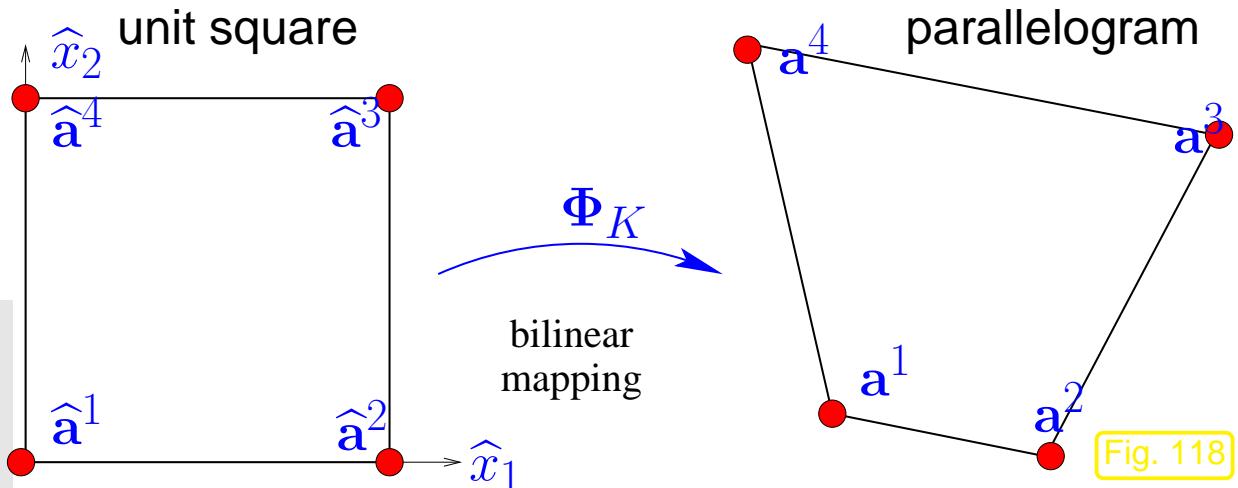
➤ What is Φ_K for a general quadrilateral ?



Affine transformations fail to produce general quadrilaterals from a square. They only give parallelograms.



It takes *bilinear transformations* to obtain a generic quadrilateral from the unit square.



Bilinear transformation of unit square to quadrilateral with vertices $\mathbf{a}^i, i = 1, 2, 3, 4$:

$$\Phi_K(\hat{\mathbf{x}}) = (1 - \hat{x}_1)(1 - \hat{x}_2) \mathbf{a}^1 + \hat{x}_1(1 - \hat{x}_2) \mathbf{a}^2 + \hat{x}_1\hat{x}_2 \mathbf{a}^3 + (1 - \hat{x}_1)\hat{x}_2 \mathbf{a}^4. \quad (3.6.12)$$

↓

$$\Phi_K(\hat{\mathbf{x}}) = \begin{pmatrix} \alpha_1 + \beta_1 \hat{x}_1 + \gamma_1 \hat{x}_2 + \delta_1 \hat{x}_1 \hat{x}_2 \\ \alpha_2 + \beta_2 \hat{x}_1 + \gamma_2 \hat{x}_2 + \delta_2 \hat{x}_1 \hat{x}_2 \end{pmatrix}, \quad \alpha_i, \beta_i, \gamma_i, \delta_i \in \mathbb{R}.$$

The mapping property $\Phi_K(\hat{\mathbf{a}}^i) = \mathbf{a}^i$ is evident. In order to see $\Phi_K(\hat{K}) = K$ (\hat{K} $\hat{=}$ unit square) for (3.6.12), verify that Φ_K maps all parallels to the coordinate axes to straight lines.

Moreover, a simple computation establishes:

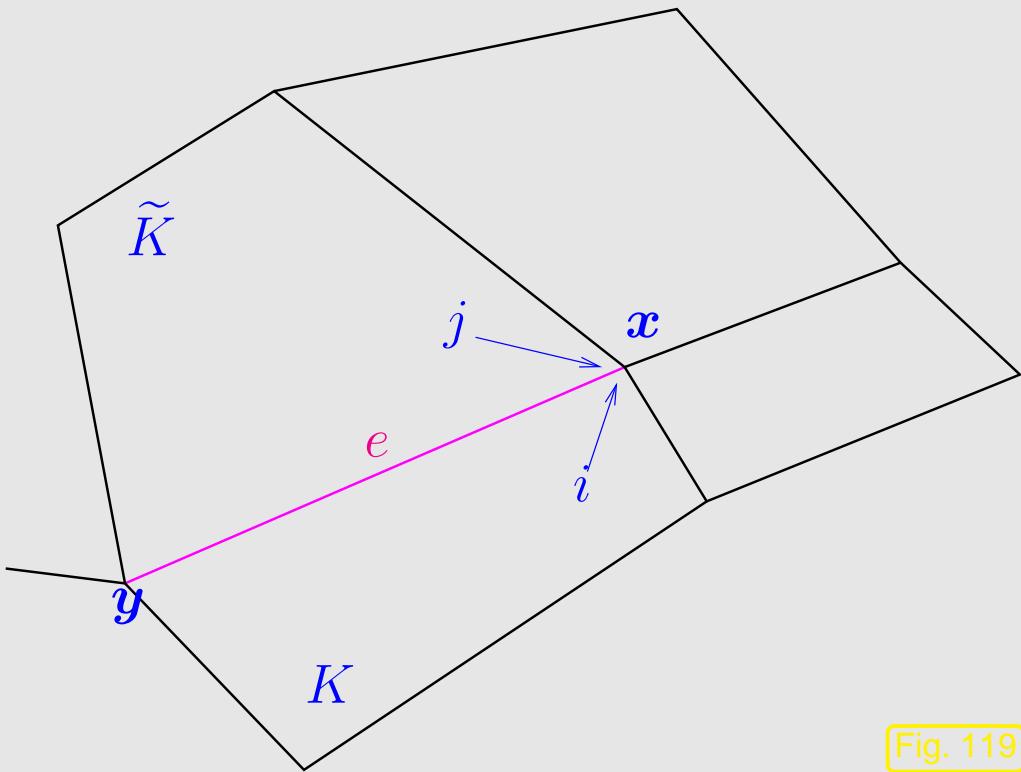
If \hat{K} is the unit square, $\Phi_K : \hat{K} \mapsto K$ a bilinear transformation, and \hat{b}^i the bilinear local shape functions (3.4.8) on \hat{K} ,

then $(\Phi_K^{-1})^* \hat{b}^i$ are linear on the edges of K .

RA

“Glueing” of local shape functions possible

Explanation:



① Pick a vertex $\mathbf{x} \in \mathcal{V}(\mathcal{M})$ and consider an adjacent quadrilateral \tilde{K} , on which there is a local shape function $b_{\tilde{K}}^i$ such that $b_{\tilde{K}}^i(\mathbf{x}) = 1$ and $b_{\tilde{K}}^i$ vanishes on all other vertices of \tilde{K} . This local shape function is obtained by inverse pullback of the \hat{b}^i associated with $\Phi_{\tilde{K}}^{-1}(\mathbf{x})$.

② The same construction can be carried out for another quadrilateral \tilde{K} that shares the vertex \mathbf{x} and an edge e with K . On that quadrilateral we find the local shape function $b_{\tilde{K}}^j$

Fig. 119

③ Both $b_{K|e}^i$ and $b_{\tilde{K}|e}^j$ are linear and attain the same values, that is 0 and 1 at the endpoints \mathbf{x} and \mathbf{y} of e , respectively.



$$b_{K|e}^i = b_{\tilde{K}|e}^j$$



Continuity of global shape function (defined by interpolation conditions at nodes)

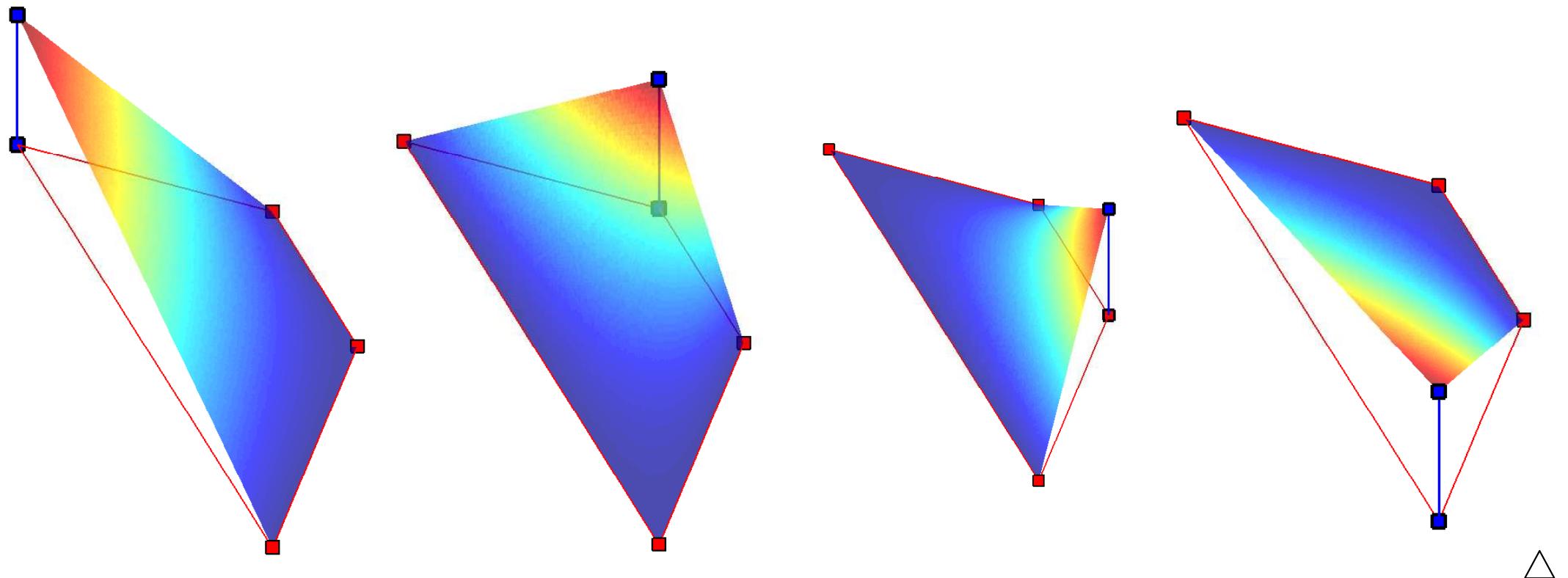
Remark 3.6.13 (Non-polynomial “bilinear” local shape functions).

Note that the components of Φ_K^{-1} are *not polynomial* even if Φ_K is a bilinear transformation (3.6.12).



The local shape functions b_K^i defined by (3.6.11), where Φ_K is a bilinear transformation and \widehat{b}^i are the bilinear local shape functions on the unit square, are **not polynomial** in general.

Visualization of local shape functions on trapezoidal cell $K := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$:



3.6.3 Transformation techniques

“Bilinear” Lagrangian finite elements = a specimen of parametric finite elements

Definition 3.6.14 (Parametric finite elements).

A *finite element space on a mesh \mathcal{M}* is called **parametric**, if there exists a *reference element \widehat{K}* , $Q \in \mathbb{N}$, and functions $\widehat{b}^i \in C^0(\overline{\widehat{K}})$, $i = 1, \dots, Q$, such that

$$\forall K \in \mathcal{M}: \exists \text{ bijection } \Phi_K: \widehat{K} \mapsto K: \widehat{b}^i = \Phi_K^* b_K^i, \quad i = 1, \dots, Q,$$

where $\{b_K^1, \dots, b_K^Q\}$ = set of local shape functions on K .

This definition takes the possibility of “glueing” for granted: the concept of a local shape function, see (3.3.11), implies the existence of a global shape function with the right continuity properties.

How to implement parametric finite elements ?

We consider a generic elliptic 2nd-order variational Dirichlet problem

$$u \in H^1(\Omega) : \int_{\Omega} (\alpha(\mathbf{x}) \operatorname{grad} u(\mathbf{x})) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x}) v(\mathbf{x}) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega). \quad (2.3.3)$$

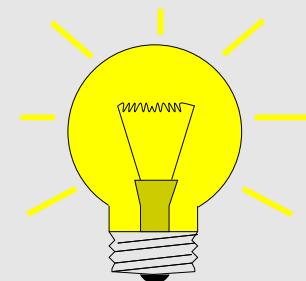
Issue: computation of element (stiffness) matrices and element (load) vectors (\rightarrow Def. 3.5.12).

Challenge: local shape functions $b_K^1, \dots, b_K^Q, K \in \mathcal{M}$, only known implicitly

$$b_K^i = (\Phi_K^{-1})^* \hat{b}^i$$

➤ Known: transformation $\Phi : \hat{K} \mapsto K$, \hat{K} reference element, functions $\hat{b}^1, \dots, \hat{b}^Q$

$$\hat{b}^i = \Phi^* b_K^i, \quad i = 1, \dots, Q \quad (\rightarrow \text{pullback, Def. 3.6.2})$$



Use transformation to \hat{K} to compute element stiffness matrix \mathbf{A}_K , element load vector $\vec{\varphi}_K$:

$$\begin{aligned} (\mathbf{A}_K)_{ij} &= \int_K \boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} b_K^j(\mathbf{x}) \cdot \operatorname{grad} b_K^i(\mathbf{x}) \, d\mathbf{x} \\ &= \int_{\hat{K}} (\Phi^* \boldsymbol{\alpha})(\hat{\mathbf{x}}) \underbrace{(\Phi^* (\operatorname{grad} b_K^j))(\hat{\mathbf{x}})}_{=?} \cdot \underbrace{(\Phi^* (\operatorname{grad} b_K^i))(\hat{\mathbf{x}})}_{=?} |\det D\Phi(\hat{\mathbf{x}})| \, d\hat{\mathbf{x}}, \end{aligned}$$

$$(\vec{\varphi}_K)_i = \int_K f(\mathbf{x}) b_K^i(\mathbf{x}) \, d\mathbf{x} = \int_{\hat{K}} (\Phi_K^* f)(\hat{\mathbf{x}}) \hat{b}^i(\hat{\mathbf{x}}) |\det D\Phi(\hat{\mathbf{x}})| \, d\hat{\mathbf{x}},$$

by **transformation formula** (for multidimensional integrals, see also (3.5.31)):

$$\int_K f(\mathbf{x}) d\mathbf{x} = \int_{\hat{K}} f(\hat{\mathbf{x}}) |\det D\Phi(\hat{\mathbf{x}})| d\hat{\mathbf{x}} \quad \text{for } f : K \mapsto \mathbb{R}, \quad (3.6.15)$$

All integrals have been transformed to the reference element \hat{K} , where we apply a quadrature formula (3.5.32).

Needed: values of determinant of Jacobi matrix $D\Phi$ at quadrature nodes $\hat{\zeta}_l$.

Also needed: gradients $\Phi^*(\operatorname{grad} b_K^i)$ at quadrature nodes $\hat{\zeta}_l$!?

(Seems to be a problem as b_K^i may be elusive, cf. Rem. 3.6.13!)

Lemma 3.6.16 (Transformation formula for gradients).

For differentiable $u : K \mapsto \mathbb{R}$ and any diffeomorphism $\Phi : \hat{K} \mapsto K$ we have

$$(\operatorname{grad}_{\hat{\mathbf{x}}}(\Phi^* u))(\hat{\mathbf{x}}) = (D\Phi(\hat{\mathbf{x}}))^T \underbrace{(\operatorname{grad}_{\mathbf{x}} u)(\Phi(\hat{\mathbf{x}}))}_{=\Phi^*(\operatorname{grad} u)(x)} \quad \forall \hat{\mathbf{x}} \in \hat{K}. \quad (3.6.17)$$

Proof: use **chain rule** for components of the gradient

$$\frac{\partial \Phi^* u}{\partial \hat{x}_i}(\hat{\boldsymbol{x}}) = \frac{\partial}{\partial \hat{x}_i} u(\Phi(\hat{\boldsymbol{x}})) = \sum_{j=1}^d \frac{\partial u}{\partial x_j} \frac{\partial \Phi_j}{\partial \hat{x}_i}(\hat{\boldsymbol{x}}).$$

► $\begin{pmatrix} \frac{\partial \Phi^* u}{\partial \hat{x}_1}(\hat{\boldsymbol{x}}) \\ \vdots \\ \frac{\partial \Phi^* u}{\partial \hat{x}_d}(\hat{\boldsymbol{x}}) \end{pmatrix} = (\text{grad}_{\hat{\boldsymbol{x}}} \Phi^* u)(\hat{\boldsymbol{x}}) = D\Phi(\hat{\boldsymbol{x}})^T \begin{pmatrix} \frac{\partial \Phi_j}{\partial \hat{x}_1}(\hat{\boldsymbol{x}}) \\ \vdots \\ \frac{\partial \Phi_j}{\partial \hat{x}_d}(\hat{\boldsymbol{x}}) \end{pmatrix} = (\text{grad}_{\boldsymbol{x}} u)(\Phi(\hat{\boldsymbol{x}})).$

Here, $D\Phi(\hat{\boldsymbol{x}}) \in \mathbb{R}^{d,d}$ is the Jacobian of Φ at $\hat{\boldsymbol{x}} \in \hat{K}$, see [19, Bem. 7.6.1].

Using Lemma 3.6.16 we arrive at:

$$(\mathbf{A}_K)_{ij} = \int_{\hat{K}} (\boldsymbol{\alpha}(\Phi(\hat{\boldsymbol{x}}))(D\Phi)^{-T} \text{grad } \hat{b}^i) \cdot ((D\Phi)^{-T} \text{grad } \hat{b}^j) |\det D\Phi| d\hat{\boldsymbol{x}}. \quad (3.6.18)$$

Note that the argument $\hat{\boldsymbol{x}}$ is suppressed for some terms in the integrand.

notation: $\mathbf{M}^{-T} := (\mathbf{M}^{-1})^T = (\mathbf{M}^T)^{-1}$

Example 3.6.19 (Transformation techniques for bilinear transformations).

$$\begin{aligned}\Phi(\hat{\mathbf{x}}) &= \begin{pmatrix} \alpha_1 + \beta_1\hat{x}_1 + \gamma_1\hat{x}_2 + \delta_1\hat{x}_1\hat{x}_2 \\ \alpha_2 + \beta_2\hat{x}_1 + \gamma_2\hat{x}_2 + \delta_2\hat{x}_1\hat{x}_2 \end{pmatrix}, \quad \alpha_i, \beta_i, \gamma_i, \delta_i \in \mathbb{R}, \\ \Rightarrow D\Phi(\hat{\mathbf{x}}) &= \begin{pmatrix} \beta_1 + \delta_1\hat{x}_2 & \gamma_1 + \delta_1\hat{x}_1 \\ \beta_2 + \delta_2\hat{x}_2 & \gamma_2 + \delta_2\hat{x}_1 \end{pmatrix}, \\ \Rightarrow \det(D\Phi(\hat{\mathbf{x}})) &= \beta_1\gamma_2 - \beta_2\gamma_1 + (\beta_1\delta_2 - \beta_2\delta_1)\hat{x}_1 + (\delta_1\gamma_2 - \delta_2\gamma_1)\hat{x}_2.\end{aligned}$$

Both $D\Phi(\hat{\mathbf{x}})$ and $\det(D\Phi(\hat{\mathbf{x}}))$ are (componentwise) linear in \mathbf{x} .

If $\Phi = \Phi_K$ for a generic quadrilateral K as in (3.6.12), then the coefficients $\alpha_i, \beta_i, \gamma_i, \delta_i$ depend on the shape of K in a straightforward fashion:

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \mathbf{a}^1, \quad \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \mathbf{a}^2 - \mathbf{a}^1, \quad \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = \mathbf{a}^4 - \mathbf{a}^1, \quad \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} = \mathbf{a}^3 - \mathbf{a}^2 - \mathbf{a}^4 + \mathbf{a}^1.$$



3.6.4 Boundary approximation

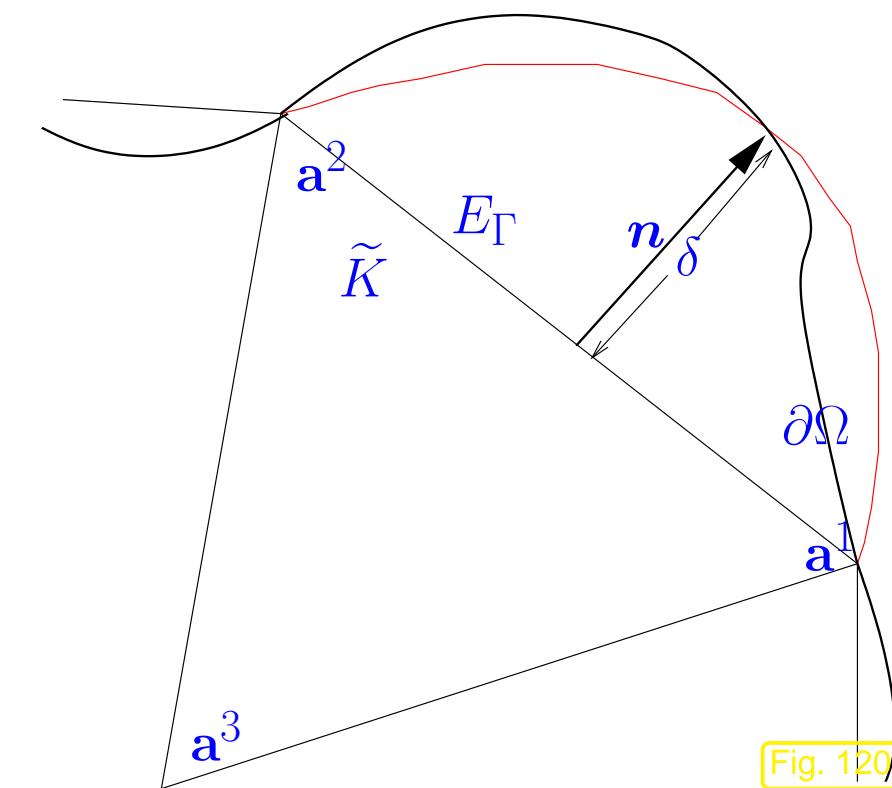
Intuition: Approximating a (smooth) curved boundary $\partial\Omega$ by a polygon/polyhedron will introduce a (sort of) **discretization error**.

Parametric finite element constructions provide a tool for avoiding polygonal/polyhedral approximation of boundaries.

Here we discuss this for a very simple case of triangular meshes in 2D (more details → [4, Sect, 10.2]).

Idea: Piecewise polynomial approximation of boundary (boundary fitting)
 $(\partial\Omega$ locally considered as function over straight edge of an element)

Example: Piecewise quadratic boundary approximation
(Part of $\partial\Omega$ between a^1 and a^2 approximated by parabola)



Mapping $\tilde{K} \rightarrow$ “curved element” K :

$$\tilde{\Phi}_K(\tilde{x}) := \tilde{x} + 4\delta \lambda_1(\tilde{x})\lambda_2(\tilde{x}) \mathbf{n} . \quad (3.6.20)$$

(λ_i barycentric coordinate functions on \tilde{K} , \mathbf{n} normal to E_Γ , see Fig. 120)

Note: Essential: δ sufficiently small $\implies \Phi$ bijective

The complete transformation $\Phi_K : \hat{K} \mapsto K$ is obtained by joining an affine transformation (\rightarrow

Def. 3.5.27) $\Phi_K^a : \hat{K} \mapsto \tilde{K}$, $\Phi_K^a(\hat{x}) := \mathbf{F}_K \hat{x} + \boldsymbol{\tau}_K$, and $\tilde{\Phi}_K$:

$$\Phi_K = \tilde{\Phi}_K \circ \Phi_K^a .$$

For parabolic boundary fitting:

$$D\tilde{\Phi}_K = \mathbf{I} + 4\delta \mathbf{n} \cdot \mathbf{grad}(\lambda_1 \lambda_2)^\top \in \mathbb{R}^{2,2} , \quad \det(D\tilde{\Phi}_K) = 1 + 4\delta \mathbf{n} \cdot \mathbf{grad}(\lambda_1 \lambda_2) .$$

3.7 Linearization

So far we have discussed the finite elements for *linear* second-order variational boundary value problems only.

However, as we have learned in Ex. 1.5.62, in 1D the Galerkin approach based on linear finite elements was perfectly capable of dealing with *on-linear* two-point boundary value problems. Indeed the abstract discussion of the Galerkin approach in Sect. 1.5.1 was aimed at general and possibly non-linear variational problems, see (1.5.7), (1.5.16).

It goes without saying that the abstract (and formal) discussion of Sect. 1.5.1 remains true for *non-linear* second-order boundary value problems in variational form.

Difficult: Characterization of “spaces of functions with finite energy” (\rightarrow Sobolev spaces, Sect. 2.2) for non-linear variational problems.

(Relief!) In this course we do not worry that much about function spaces.

Recall (\rightarrow Rem. 1.3.10): Non-linear variational problem

$$u \in V: \quad \mathbf{a}(u; v) = \ell(v) \quad \forall v \in V_0 , \quad (1.3.12)$$

- $V_0 \hat{=} \text{test space, (real) vector space (usually a function space, "Sobolev-type" space} \rightarrow \text{Sect. 2.2)}$
- $V \hat{=} \text{trial space, affine space: usually } V = u_0 + V_0, \text{ with offset function } u_0 \in V,$
- $f \hat{=} \text{a linear mapping } V_0 \mapsto \mathbb{R}, \text{ a linear form,}$
- $\mathbf{a} \hat{=} \text{a mapping } V \times V_0 \mapsto \mathbb{R}, \text{ linear in the second argument, that is}$

$$\mathbf{a}(u; \alpha v + \beta w) = \alpha \mathbf{a}(u; v) + \beta \mathbf{a}(u; w) \quad \forall u \in V, v, w \in V_0, \alpha, \beta \in \mathbb{R} . \quad (3.7.1)$$

Example 3.7.2 (Heat conduction with radiation boundary conditions).

➤ 2nd-order elliptic boundary value problem, cf. (2.5.6) & (2.6.3)

$$\begin{aligned} -\operatorname{div}(\kappa(\boldsymbol{x}) \operatorname{grad} u) &= f && \text{in } \Omega, \\ \kappa(\boldsymbol{x}) \operatorname{grad} u \cdot \boldsymbol{n}(\boldsymbol{x}) + \Psi(u) &= 0 && \text{on } \partial\Omega. \end{aligned}$$

➤ Variational formulation from Ex. 2.8.5

$$u \in H^1(\Omega): \quad \int_{\Omega} \kappa(\boldsymbol{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\boldsymbol{x} + \int_{\partial\Omega} \Psi(u) v \, dS = \int_{\Omega} fv \, d\boldsymbol{x} \quad \forall v \in H^1(\Omega). \quad (2.8.11)$$

If $\Psi : \mathbb{R} \mapsto \mathbb{R}$ is not an affine linear function, then (2.8.11) represents a non-linear variational problem (1.3.12) with

- trial/test space $V = V_0 = H^1(\Omega)$ (\rightarrow Def. 2.2.12),
- right hand side linear form $\ell(v) := \int_{\Omega} fv \, d\boldsymbol{x}$,
- $\mathbf{a}(u; v) := \int_{\Omega} \kappa(\boldsymbol{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\boldsymbol{x} + \int_{\partial\Omega} \Psi(u) v \, dS$.

Note that the non-linearity enters only through the boundary term.



Pursuing the policy of Galerkin discretization (choice of discrete spaces and corresponding bases, → Sect. 1.5.1) we can convert (1.3.12) into a non-linear system of equations

$$\mathbf{a}(u_0 + \sum_{j=1}^N \mu_j b_N^j; b_N^k) = f(b_N^k) \quad \forall k = 1, \dots, N . \quad (1.5.16)$$

If the left hand side depends smoothly on the unkowns (the corefficients μ_j of $\vec{\mu}$), then the classical Newton method (→ [14, Sect. 3.4]) to solve it iteratively.

Here, we focus on a different approach that reverses the order of the steps:

1. Linearization of problem (“Newton in function space”),
2. Galerkin discretization of linearized problems.

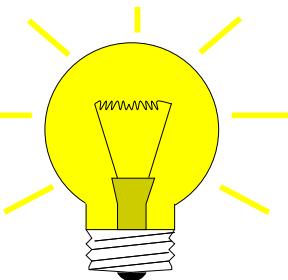
“Newton in function space”:

Idea:

local linearization:

Given $\vec{\xi}^{(k)} \in D \Rightarrow \vec{\xi}^{(k+1)}$ as zero of affine linear model function

$$F(\vec{\xi}) \approx \tilde{F}(\vec{\xi}) := F(\vec{\xi}^{(k)}) + DF(\vec{\xi}^{(k)})(\vec{\xi} - \vec{\xi}^{(k)}) .$$



Newton iteration:

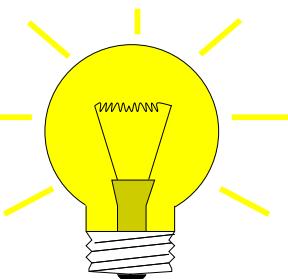
$$\vec{\xi}^{(k+1)} := \vec{\xi}^{(k)} - DF(\vec{\xi}^{(k)})^{-1} F(\vec{\xi}^{(k)}) , \quad [\text{if } DF(\vec{\xi}^{(k)}) \text{ regular}] \quad (3.7.3)$$

← apply idea to (1.3.12)

Idea:

local linearization:

Given $u^{(k)} \in V \Rightarrow u^{(k+1)}$ from



$$w \in V_0: \quad \mathbf{a}(u^{(k)}; v) + D_u \mathbf{a}(u^{(k)}; v) w = \ell(v) \quad \forall v \in V_0 ,$$
$$u^{(k+1)} := u^{(k)} + w .$$

(3.7.4)

The meaning of $DF(\vec{\xi}^{(k)})$ in (3.7.3) is clear: it stands for the **Jacobian** of F evaluated at $\vec{\xi}^{(k)}$

But what is the meaning of $D_u \mathbf{a}(u^{(k)}; v) w$ in (3.7.4)?

3.7

p. 414

Remember the “definition” of the Jacobian (for sufficiently smooth F)

$$DF(\vec{\xi})\vec{\mu} = \lim_{t \rightarrow 0} \frac{F(\vec{\xi} + t\vec{\mu}) - F(\vec{\xi})}{t}, \quad \vec{\xi} \in D, \vec{\mu} \in \mathbb{R}^N. \quad (3.7.5)$$

➤ try the “definition”

$$D_u \mathbf{a}(u^{(k)}; v)w = \lim_{t \rightarrow 0} \frac{\mathbf{a}(u + tw; v) - \mathbf{a}(u; v)}{t}, \quad u^{(k)} \in V, \quad v, w \in V_0. \quad (3.7.6)$$

If $(u, v) \mapsto \mathbf{a}(u; v)$ depends smoothly on u , then

$(v, w) \mapsto D_u \mathbf{a}(u^{(k)}; v)w$ is a **bilinear form** $V_0 \times V_0 \mapsto \mathbb{R}$.

Example 3.7.7 (Derivative of non-linear $u \mapsto \mathbf{a}(u; \cdot)$).

Apply formula (3.7.6) to the non-linear boundary term in (2.8.11), that is, here

$$\mathbf{a}(u; v) := \int_{\partial\Omega} \Psi(u)v \, dS, \quad u, v \in H^1(\Omega).$$

► $\mathbf{a}(u + tw; v) - \mathbf{a}(u; v) = \int_{\partial\Omega} (\Psi(u + tw) - \Psi(u))v \, dS , \quad u, v \in H^1(\Omega) .$

Assume $\Psi : \mathbb{R} \mapsto \mathbb{R}$ is smooth with derivative Ψ' and employ *Taylor expansion* for fixed $w \in H^1(\Omega)$ and $t \rightarrow 0$

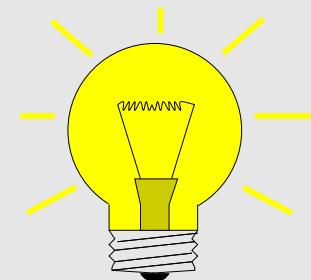
$$\mathbf{a}(u + tw; v) - \mathbf{a}(u; v) = \int_{\partial\Omega} t\Psi'(u)wv \, dS + O(t^2) .$$

► $D_u \mathbf{a}(u^{(k)}; v)w = \lim_{t \rightarrow 0} \frac{\mathbf{a}(u + tw; v) - \mathbf{a}(u; v)}{t} = \int_{\partial\Omega} \Psi'(u)wv \, dS .$

= a **bilinear** form in v, w on $H^1(\Omega) \times H^1(\Omega)$!

This example also demonstrates how to actually compute $D_u \mathbf{a}(u^{(k)}; v)w$!

Idea: Galerkin discretization of the **linear variational** problem from (3.7.4)



$$w \in V_0: \quad \mathbf{c}(w, v) = g(v) \quad \forall v \in V_0 ,$$

$$\mathbf{c}(w, v) = D_u \mathbf{a}(u^{(k)}; v)w , \quad g(v) := \ell(v) - \mathbf{a}(u^{(k)}; v) .$$

► Newton-Galerkin iteration for (1.3.12)

Given $u_N^{(k)} \in V_N^{(k)}$ ➤ $u_N^{(k+1)} \in V_N^{(k+1)}$ from

$$\boxed{w_N \in V_{0,N}^{(k+1)} : D_u \mathbf{a}(u_N^{(k)}; v_N) w_N = \ell(v_N) - \mathbf{a}(u_N^{(k)}; v_N) \quad \forall v_N \in V_{0,N}^{(k+1)}, \\ u_N^{(k+1)} := P_N^{(k+1)} u_N^{(k)} + w.}$$

 **Newton update**

(3.7.8)

Note: different Galerkin trial/test spaces $V_N^{(k)}$, $V_{0,N}^{(k)}$ may be used in different steps of the iteration!

(It may enhance efficiency to use Galerkin trial/test spaces of a rather small dimension in the beginning and switch to larger when the iteration is about to converge.)

Warning! If $V_N^{(k)} \neq V_N^{(k+1)}$ you cannot simply add $u_N^{(k)}$ and w

➤ Linear projection operator $P_N^{(k+1)} : V_N^{(k)} \mapsto V_N^{(k+1)}$ required in (3.7.8)

Any of the Lagrangian finite element spaces introduced in Sect. 3.4 will supply valid $V_N/V_{0,N}$. Offset functions can be chosen according to the recipes from Sect. 3.5.5.

Important aspect: **termination** of iteration, see [14, Thm. 3.4.3].

Option: termination based on relative size of Newton update, with w , $u_N^{(k+1)}$ from (3.7.8)

$$\textbf{STOP, if } \|w\| \leq \|u_N^{(k+1)}\| , \quad (3.7.9)$$

where $\|\cdot\|$ is a relevant norm (e.g., energy norm) on $V_N^{(k+1)}$.

Finite Differences (FD) and Finite Volume Methods (FV)

Now we examine two approaches to the discretization of scalar linear 2nd-order elliptic BVPs that offer an alternative to finite element Galerkin methods discussed in Ch. 3.

What these methods have in common with (low degree) Lagrangian finite element methods is

- that they rely on meshes (\rightarrow Sect. 3.3.1) tiling the computational domain Ω ,
- they lead to *sparse* linear systems of equations.

Remark 4.0.1 (Collocation approach on “complicated” domains).

Sect. 1.5.2.2 taught us **spline collocation methods**. A crucial insight was that collocation methods (see beginning of Sect. 1.5.2 for a presentation of the idea), which target the boundary value problem in ODE/PDE form, have to employ discrete trial spaces comprised of *continuously differentiable* functions, see Rem. 1.5.79.

It is very difficult to construct spaces of piecewise polynomial C^1 -functions on non-tensor product domains for $d = 2, 3$ and find suitable collocation nodes, cf. (1.5.74).

Therefore we skip the discussion of collocation methods for 2nd-order elliptic BVPs on $\Omega \subset \mathbb{R}^d$, $d = 2, 3$.



4.1 Finite differences

A finite difference scheme for a 2-point boundary value problem was presented in Sect. 1.5.3, which you are advised to browse again. Its gist was

to replace the derivatives in the *differential equation* with *difference quotients* connecting approximate values of the solutions *at the nodes of a grid/mesh*.

Recall: Finite differences target the “ODE/PDE-formulation” of the boundary value problem.

Our goal: extension to higher dimensions

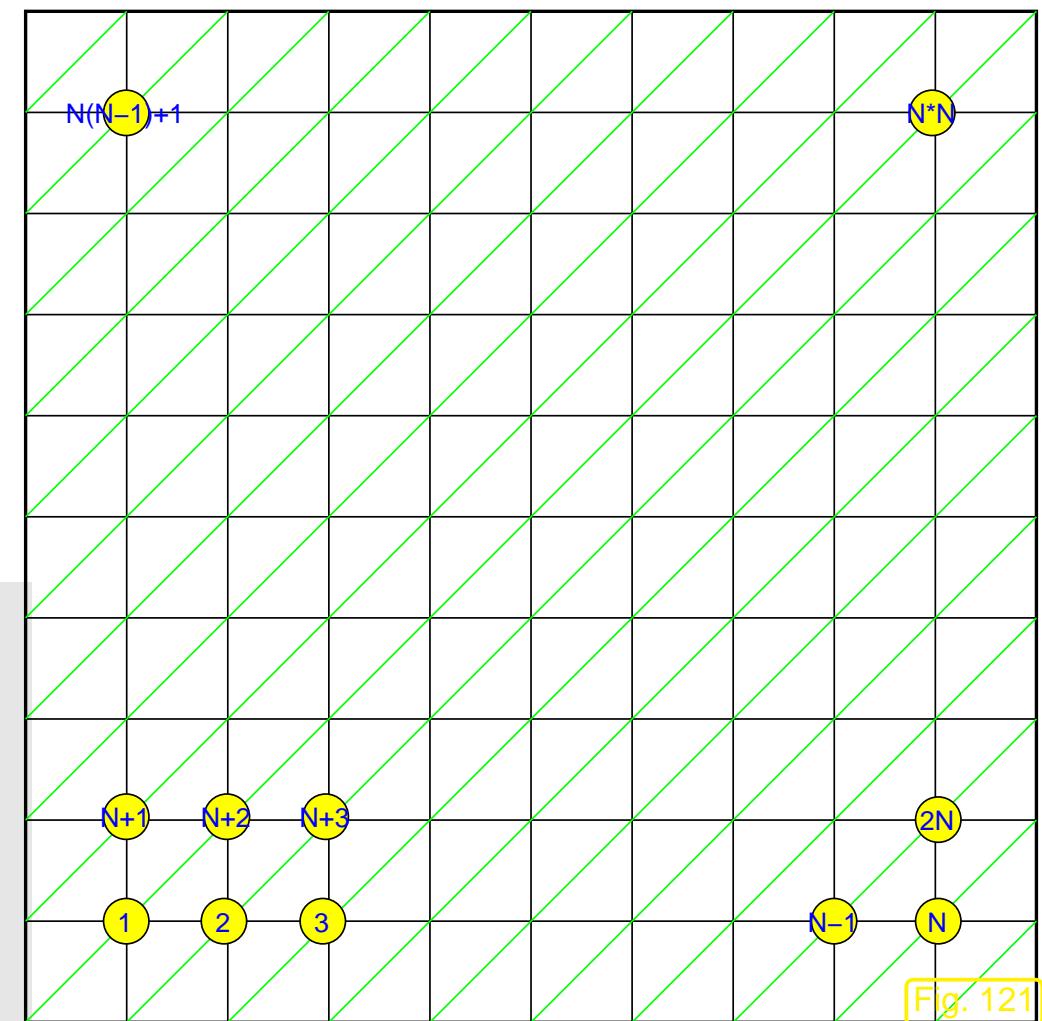
2D model problem:

Homogeneous Dirichlet BVP for Laplacian:

$$\begin{aligned} -\Delta u &= -\frac{\partial^2 u}{\partial x_1^2} - \frac{\partial^2 u}{\partial x_2^2} = f \quad \text{in } \Omega := [0, 1]^2, \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

Discretization based on

\mathcal{M} = (triangular) **tensor-product grid**
(meshwidth $h = (1 + N)^{-1}$, $N \in \mathbb{N}$)
lexikographic (line-by-line) ordering of nodes of \mathcal{M}



- ① finite difference approach to $-\Delta$: approximation of derivatives by symmetric difference quotients

This is nothing new: we did this in (1.5.93).

$$\frac{\partial^2}{\partial x_1^2} u \Big|_{\mathbf{x}=(\xi,\eta)} \approx \frac{u(\xi - h, \eta) - 2u(\xi, \eta) + u(\xi + h, \eta)}{h^2},$$

$$\frac{\partial^2}{\partial x_2^2} u \Big|_{\mathbf{x}=(\xi,\eta)} \approx \frac{u(\xi, \eta - h) - 2u(\xi, \eta) + u(\xi, \eta + h)}{h^2}.$$

► $-\Delta u|_{\mathbf{x}=(\xi,\eta)} \approx \frac{1}{h^2}(4u(\xi, \eta) - u(\xi - h, \eta) - u(\xi + h, \eta) - u(\xi, \eta - h) - u(\xi, \eta + h))$.

Use this approximation at grid point $\mathbf{p} = (ih, jh)$. This will connect the five point values $u(ih, jh)$, $u((i-1)h, jh)$, $u((i+1)h, jh)$, $u(ih, (j-1)h)$, $u(ih, (j+1)h)$.

Approximations $\mu_{i,j}$ to the *point values* $u(ih, jh)$
will be the **unknowns** of the finite difference method.

Centering the above difference quotients at grid points yields linear relationships between the unknowns:

$$\frac{1}{h^2}(4u(ih, jh) - u(ih - h, jh) - u(ih + h, jh) - u(ih, jh - h) - u(ih, jh + h)) = f(ih, jh),$$

$$\frac{1}{h^2} (4\mu_{i,j} - \mu_{i-1,j} - \mu_{i+1,j} - \mu_{i,j-1} - \mu_{i,j+1}) = f(ih, jh) .$$

Also this is familiar from the discussion in 1D. Yet, in 1D the association of the point values and of components of the vector $\vec{\mu}$ of unknowns was straightforward and suggested by the linear ordering of the nodes of the grid. In 2D we have much more freedom.

One option on tensor-product grids is the line-by-line ordering (lexikographic ordering) depicted in Fig. 121. This allows a simple indexing scheme:

$$u(\mathbf{p}) \leftrightarrow \mu_{i,j} \leftrightarrow \mu_{(j-1)N+i}$$



$$\frac{-\mu_{(j-2)N+i} - \mu_{(j-1)N+i-1} + 4\mu_{(j-1)N+i} - \mu_{(j-1)N+i+1} - \mu_{jN+i}}{h^2} = \underbrace{f(ih, jh)}_{=\varphi_{(j-1)N+i}} . \quad (4.1.1)$$

► linear system of N^2 equations $\mathbf{A}\vec{\mu} = \vec{\varphi}$ with $N^2 \times N^2$ block-tridiagonal Poisson matrix

$$\mathbf{A} := \frac{1}{h^2} \begin{pmatrix} \mathbf{T} & -\mathbf{I} & 0 & \cdots & \cdots & 0 \\ -\mathbf{I} & \mathbf{T} & -\mathbf{I} & & & \vdots \\ 0 & -\mathbf{I} & \mathbf{T} & -\mathbf{I} & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & -\mathbf{I} & \mathbf{T} & -\mathbf{I} & \\ 0 & \cdots & \cdots & 0 & -\mathbf{I} & \mathbf{T} \end{pmatrix}, \quad \mathbf{T} := \begin{pmatrix} 4 & -1 & 0 & & & 0 \\ -1 & 4 & -1 & & & \vdots \\ 0 & -1 & 4 & -1 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \\ \vdots & & & -1 & 4 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 4 \end{pmatrix} \in \mathbb{R}^{N,N} \quad (4.1.2)$$

$$\mathbf{A} = \left(\begin{array}{cccccc} \text{[Diagram showing the band structure of the Poisson matrix A. It consists of a 5x5 grid of smaller 3x3 blocks. The main diagonal blocks have magenta hatching. The super-diagonals and sub-diagonals have blue hatching. The corners of the grid are labeled with zeros. Ellipses indicate the continuation of the pattern.]} & & & & & & 0 \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 0 & & & & \end{array} \right)$$

◀ band structure of Poisson matrix

The MATLAB command

`A = gallery('poisson', n)`

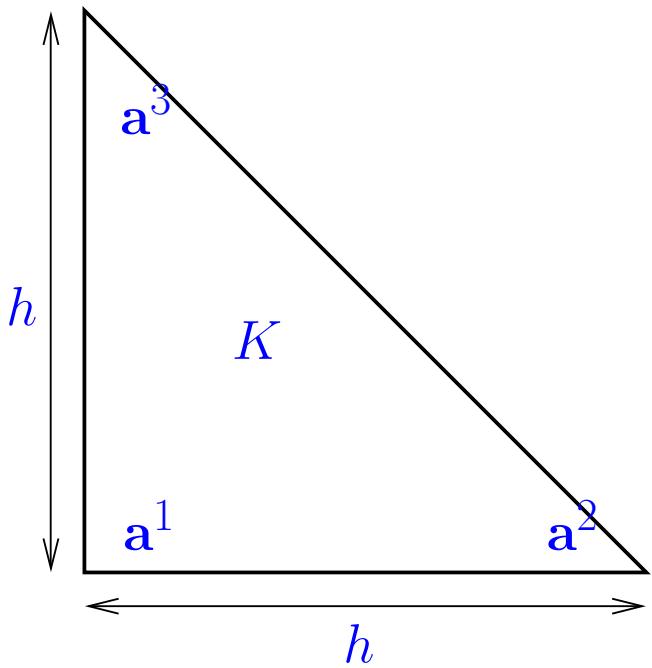
creates a sparse $n^2 \times n^2$ Poisson matrix.

Already in Sect. 1.5.3 we saw that the linear system of equations popping out of the finite difference discretization of the linear two-point BVP (1.5.77) was the same as that obtained via the linear finite Galerkin approach on the same mesh.

In two dimensions we will also come to this conclusion! So, let us derive the Galerkin matrix and right hand side vector for the 2D model problem on the tensor product mesh depicted in Fig. 121. To begin with we convert it into a **triangular mesh \mathcal{M}** by splitting each square into two equal triangles by inserting a diagonal (green lines in Fig. 121). On this mesh we use **linear Lagrangian finite elements** as in Sect. 3.2.

Then we repeat the considerations of Sect. 3.2.

- ② Linear Lagrangian finite element Galerkin discretization → Sect. 3.2: $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M})$
(global shape functions $\hat{\triangleq}$ “tent functions”, → Fig. 85)



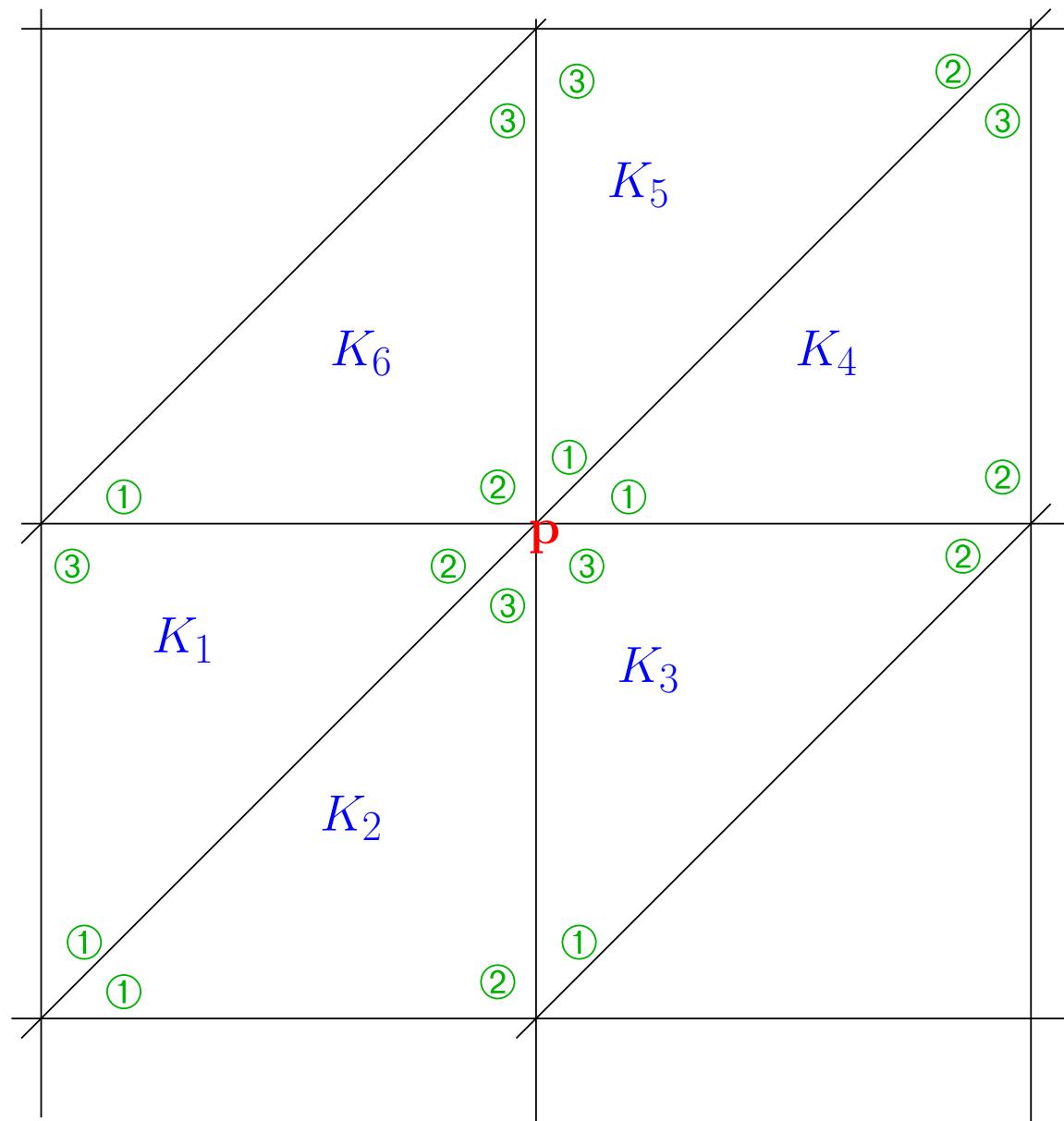
Element stiffness matrix from (3.2.9):

$$\mathbf{A}_K = \frac{1}{2} \begin{pmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} .$$

(← numbering of local shape functions)

Element load vector: use **three-point quadrature formula** (3.5.35)

► $\vec{\varphi}_K = \frac{1}{6}h^2 \begin{pmatrix} f(\mathbf{a}^1) \\ f(\mathbf{a}^2) \\ f(\mathbf{a}^3) \end{pmatrix} .$



Local assembly:

← green: local vertex numbers

Contributions to load vector component associated with node **p**:

From K_1 : $(\vec{\varphi}_{K_1})_2$

From K_2 : $(\vec{\varphi}_{K_2})_3$

From K_3 : $(\vec{\varphi}_{K_3})_3$

From K_4 : $(\vec{\varphi}_{K_4})_1$

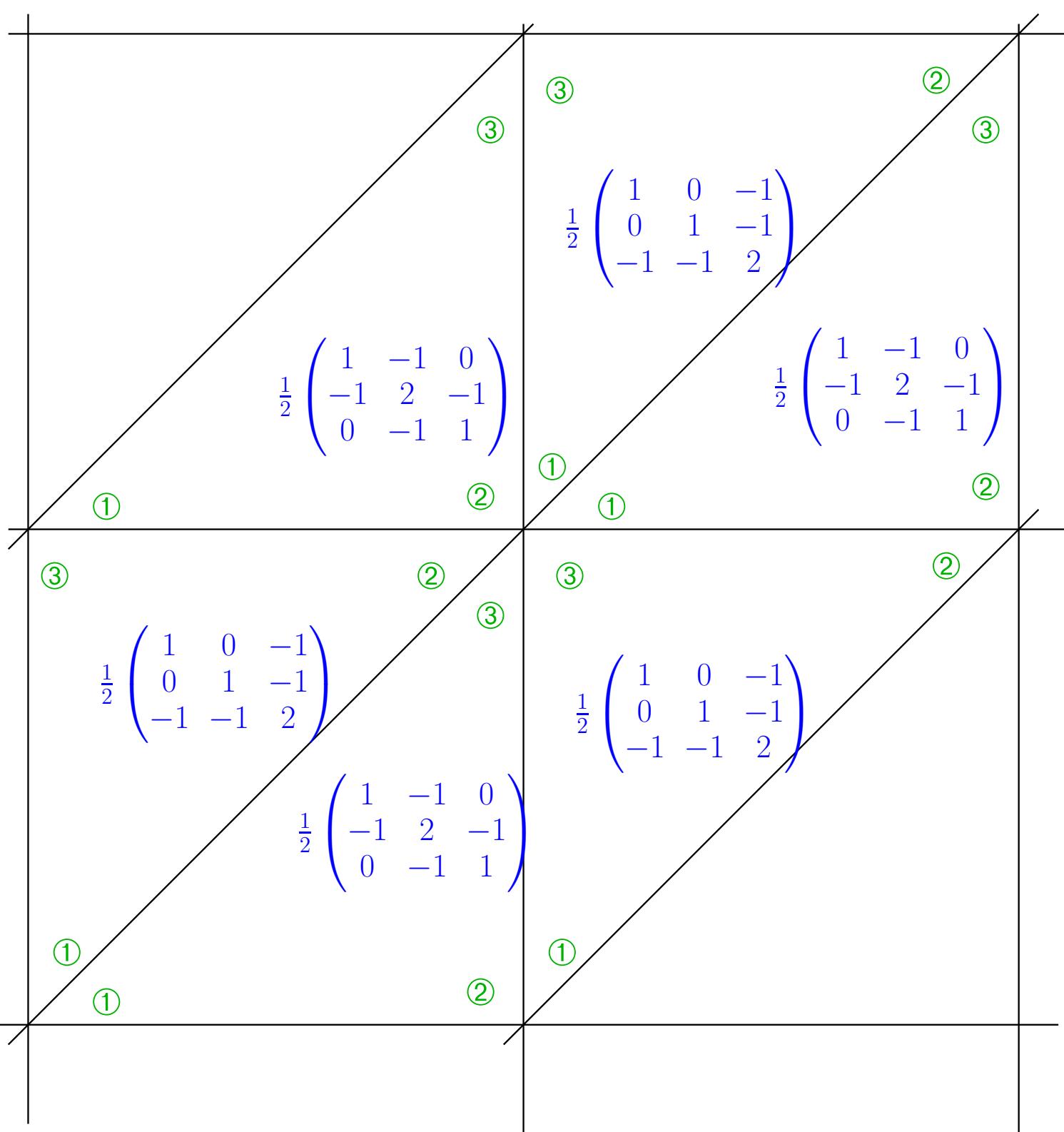
From K_5 : $(\vec{\varphi}_{K_5})_1$

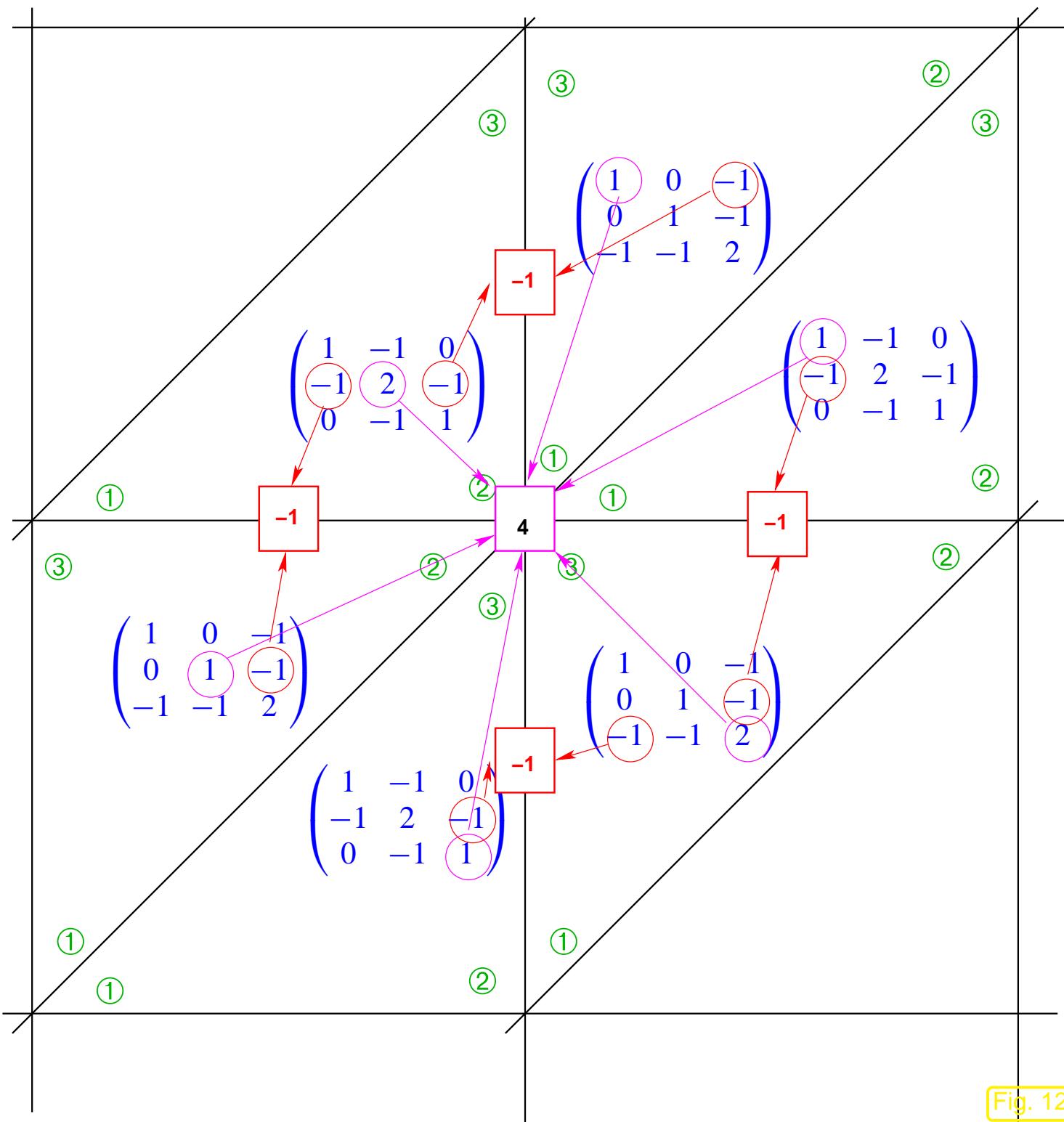
From K_6 : $(\vec{\varphi}_{K_6})_2$

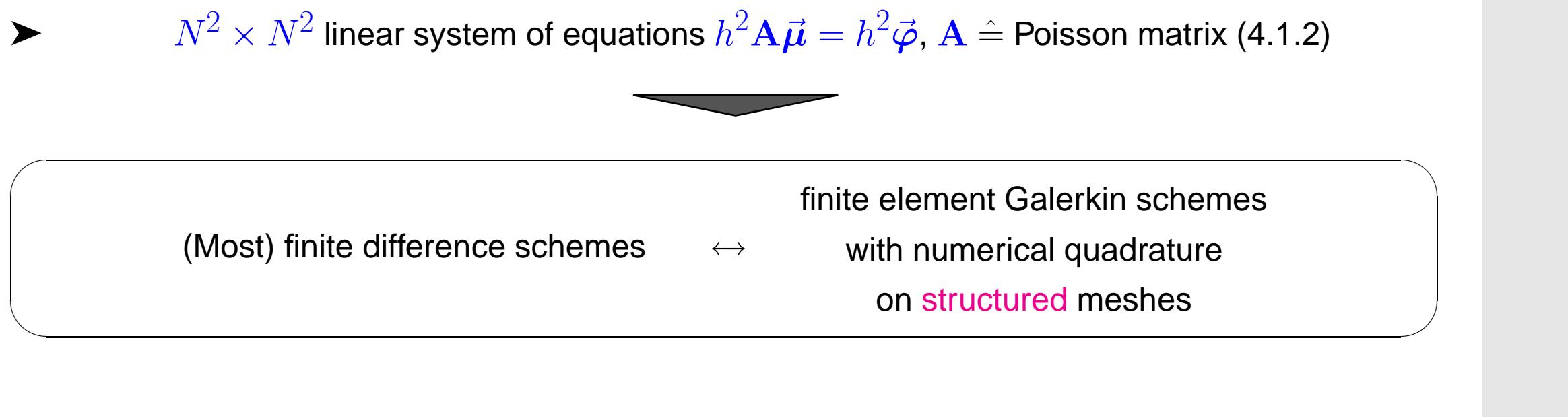


$$\vec{\varphi}_p = h^2 f(p) .$$

Assembly of finite element Galerkin matrix from element (stiffness) matrices (→ Sect. 3.5.3):







- Discussion: finite differences vs. finite element Galerkin methods
(here focused on 2nd-order linear scalar problems)
- Finite element methods can be used on general triangulations and structured (tensor-product) meshes alike, which delivers superior flexibility in terms of geometry resolution (advantage FEM).
 - The correct treatment of all kinds of boundary conditions (→ Sect. 2.6). naturally emerges from the variational formulations in the finite element method (advantage FEM).
 - Finite element methods have built-in “safety rails” because there are clear criteria for choosing viable finite element spaces and once this is done, there is no freedom left to go astray (advantage FEM).
 - Finite element methods are harder to understand (advantage FD, but only with students who have not attended this course!)

4.2 Finite volume methods (FVM)

4.2.1 Gist of FVM

Focus: linear scalar 2nd-order elliptic boundary value problem in 2D (\rightarrow Sect. 2.5), homogeneous Dirichlet boundary conditions (\rightarrow Sect. 2.6), uniformly positive scalar heat conductivity $\kappa = \kappa(\mathbf{x})$

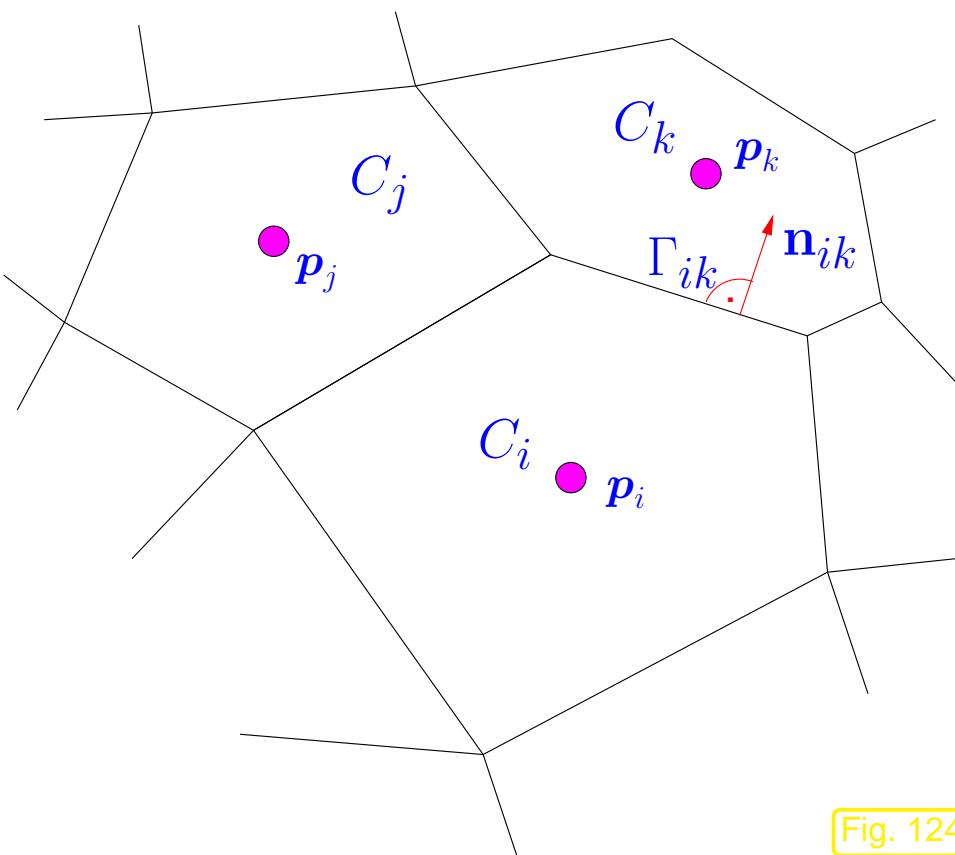
$$-\operatorname{div}(\kappa(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \Omega \quad , \quad u = 0 \quad \text{on } \partial\Omega .$$

Finite volume methods for 2nd-order elliptic BVP are inspired by the *conservation principle* (2.5.2).

$$\int_{\partial V} \mathbf{j} \cdot \mathbf{n} \, dS = \int_V f \, d\mathbf{x} \quad \text{for all “control volumes” } V . \quad (2.5.2)$$

Physics requires that this holds for all (infinitely many) “control volumes” $V \subset \Omega$.

Since discretization has to lead to a finite number of equations, the idea is to demand that (2.5.2) holds for only a *finite number of special control volumes*.



Concrete choice:

Control volumes =

(polygonal) cells of a mesh $\tilde{\mathcal{M}} = \{C_i\}_i$
covering computational domain Ω .

Associate cell $C_i \leftrightarrow$ nodal value μ_i

Meaning: $\mu_i \approx u(\mathbf{p}_i)$, \mathbf{p}_i = “center” of C_i

Fig. 124

The conservation law (2.5.2) had to be linked to the flux law (2.5.3) in order to give rise to a 2nd-order scalar PDE see (2.5.5)–(2.5.6).

Correspondingly, “heat conservation in control volumes” has to be supplemented by a rule that furnishes the heat flux between two adjacent control volumes.



Second ingredient: local **numerical fluxes**

For two adjacent cells C_k, C_i with common edge $\Gamma_{ik} := \overline{C}_i \cap \overline{C}_k$.

$$\text{Numerical flux} \quad J_{ik} = \Psi(\mu_i, \mu_k) \approx \int_{\Gamma_{ik}} \mathbf{j} \cdot \mathbf{n}_{ik} dS$$

(Ψ = numerical flux function, \mathbf{j} = (heat) flux, see (2.5.1), $\mathbf{n}_{ik} \hat{=} \text{edge normal.}$)



Idea: consider balance law on (finitely many !) control volumes C_i

$$\int_{\partial C_i} \mathbf{j} \cdot \mathbf{n}_i dS = \int_{C_i} f d\mathbf{x} \Rightarrow \sum_{k \in \mathcal{U}_i} J_{ik} = \int_{C_i} f d\mathbf{x} .$$

notation: $\mathcal{U}_i := \{j : C_i \text{ and } C_j \text{ share edge, } C_j \in \widetilde{\mathcal{M}}\}, \mathbf{p}_i = \text{node associated with control volume } C_i.$



System of equations ($\widetilde{M} := \#\mathcal{M}$ equations, unknowns μ_i):

$$\sum_{k \in \mathcal{U}_i} \Psi(\mu_i, \mu_k) = \int_{C_i} f \, d\boldsymbol{x} \quad \forall i = 1, \dots, \widetilde{M}. \quad (4.2.1)$$

Further approximation: 1-point quadrature for approximate evaluation of integral over C_i ,

$$\sum_{k \in \mathcal{U}_i} \Psi(\mu_i, \mu_k) = |C_i| f(\boldsymbol{p}_i) \, d\boldsymbol{x} \quad \forall i = 1, \dots, \widetilde{M}. \quad (4.2.2)$$

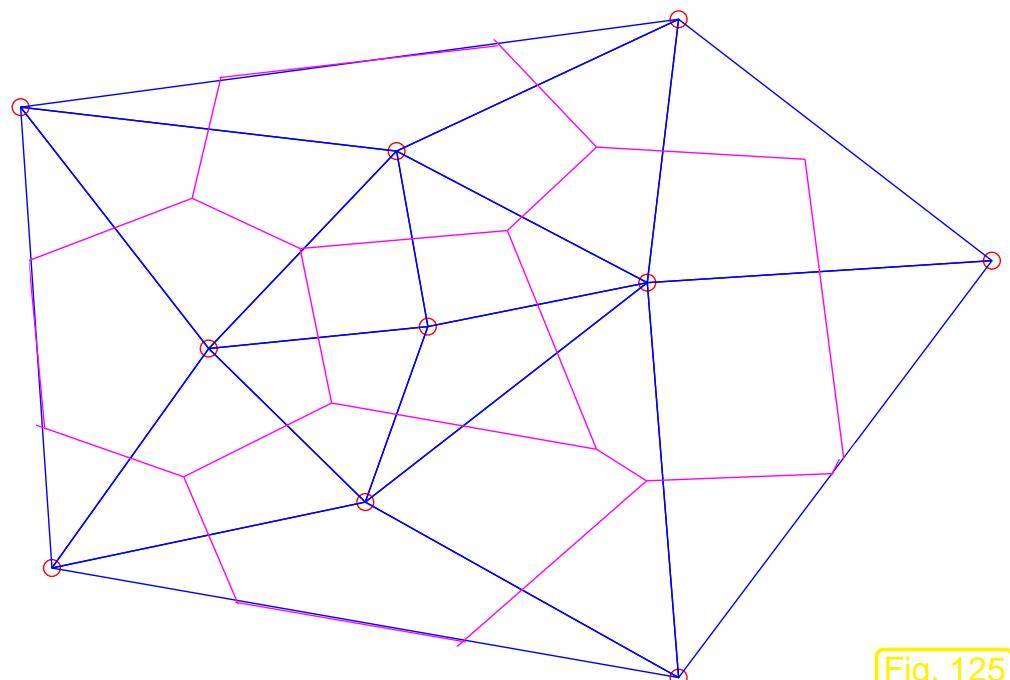
Note: homogeneous Dirichlet problem \gg only “interior” control volumes in (4.2.2)

4.2.2 Dual meshes

Dual meshes are a commonly used technique for the construction of control volumes for FVM, based on conventional FE triangulation \mathcal{M} of Ω (\rightarrow Sect. 3.3.1).

Focus: dual mesh for triangular mesh \mathcal{M} in 2D, Ω polygon

Popular choice: Voronoi dual mesh



$$\mathcal{V}(\mathcal{M}) = \{\mathbf{p}_1, \dots, \mathbf{p}_M\} = \text{nodes of } \mathcal{M}$$

Define Voronoi cells

$$C_i := \{\mathbf{x} \in \Omega: |\mathbf{x} - \mathbf{p}_i| < |\mathbf{x} - \mathbf{p}_j| \forall j \neq i\} . \quad (4.2.3)$$

Voronoi dual mesh $\widetilde{\mathcal{M}} := \{C_i\}_{i=1}^M$

Construction of Voronoi dual cells:
edges → perpendicular bisectors
nodes → circumcenters of triangles

► straightforward generalization to 3D

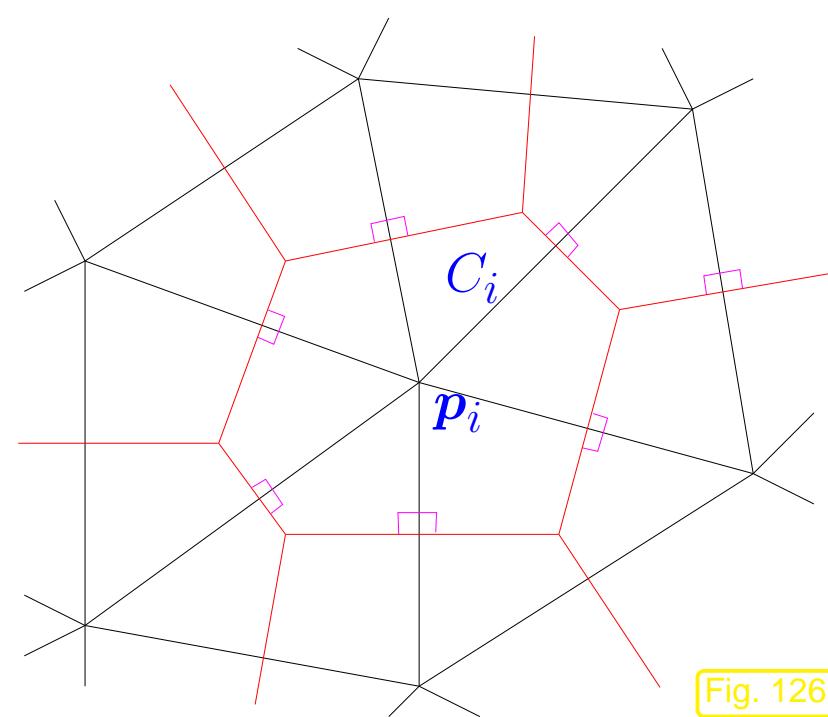


Fig. 126

Remark 4.2.4 (Geometric obstruction to Voronoi dual meshes).

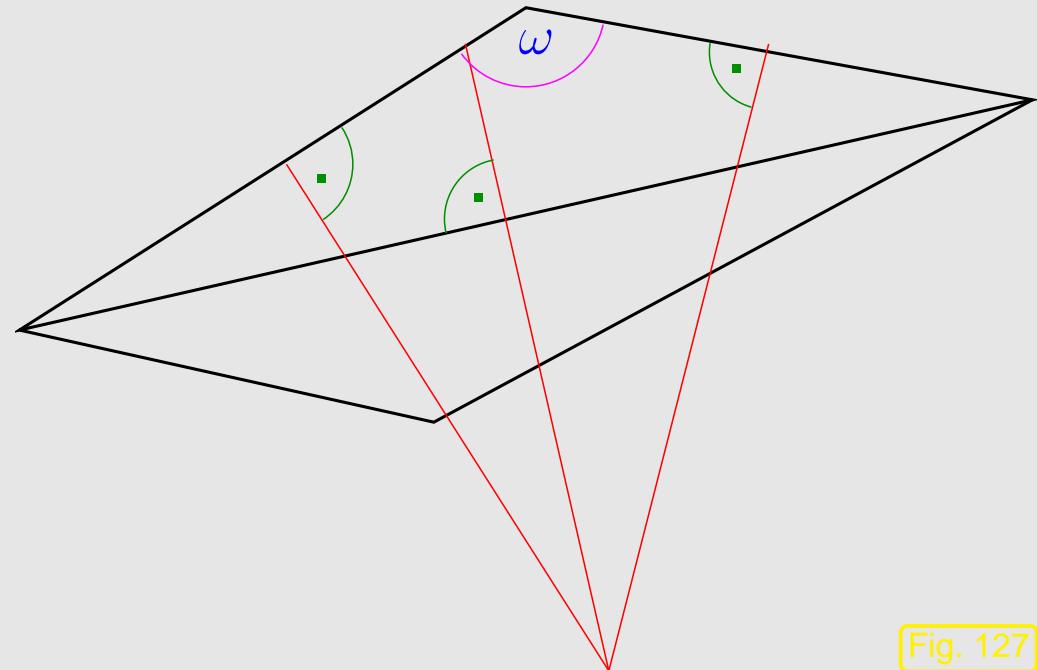


Fig. 127

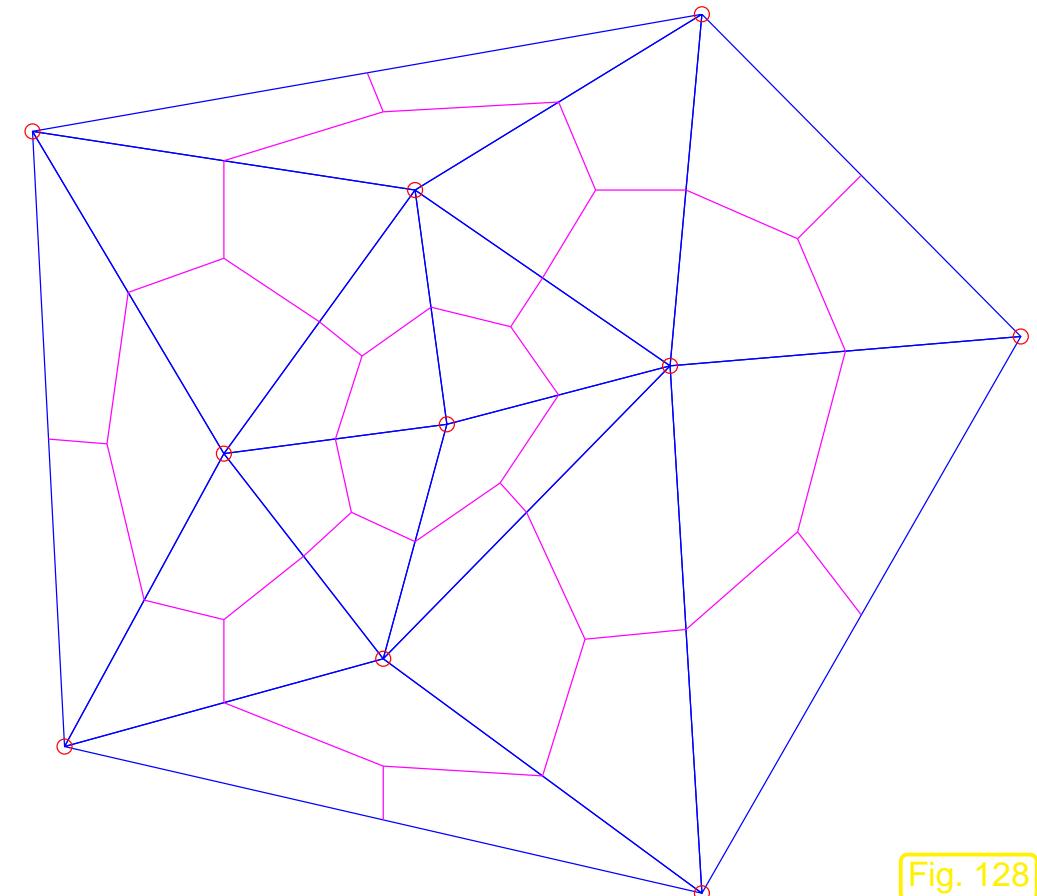
- \Leftarrow Obtuse angle ω :
- circumcenter \notin triangle
 - $\overline{C}_i \cap \overline{C}_j \neq \emptyset \not\Rightarrow$ nodes i, j connected by edge
 - geometric construction breaks down
 - connectivity of unknowns hard to determine

△

Angle condition to ensure $\overline{C}_i \cap \overline{C}_j \neq \emptyset \Leftrightarrow$ nodes i, j connected by edge of \mathcal{M} :

- (i) sum of angles facing interior edge $\leq \pi$,
- (ii) angles facing boundary edges $\leq \pi/2$ (for non-Dirichlet boundary conditions).

► (i), (ii) characterize Delaunay triangulations



Dual cells:

edges → union of lines connecting
barycenters and midpoints of
edges of \mathcal{M}

nodes → barycenters of triangles

► No geometric obstructions

4.2.3 Relationship of finite elements and finite volume methods

Hardly surprising, finite volume methods and finite element Galerkin discretizations are closely related. This will be explored in this section for a model problem.

Setting:

- We consider the homogeneous Dirichlet problem for the Laplacian Δ

$$-\Delta u = f \quad \text{in } \Omega , \quad u = 0 \quad \text{on } \partial\Omega . \quad (4.2.5)$$

- Discretization by finite volume method based on a triangular mesh \mathcal{M} and on Voronoi dual cells
→ Fig. 125:

Assumption: \mathcal{M} = Delaunay triangulation of Ω \Leftrightarrow angle condition

Number of control volumes = number of interior nodes of \mathcal{M}

Still missing: specification of numerical flux function $\Psi : \mathbb{R}^2 \mapsto \mathbb{R}$ for each **dual edge**



Idea: obtain numerical flux from

Fourier's law (2.5.3) applied to a (sufficiently smooth) $u_N : \Omega \mapsto \mathbb{R}$
reconstructed from dual cell values μ_i .

Natural approach, since μ_i is read as approximation of $u(\mathbf{p}_i)$, where the “center” \mathbf{p}^i of the dual cell C_i coincides with a *node* $\mathbf{x}^i \in \mathcal{V}(\mathcal{M})$ of the triangular mesh \mathcal{M} :

$$u_N = \mathbf{l}_1 \vec{\mu} := \sum_{i=1}^N \mu_i b_N^i , \quad (4.2.6)$$

where $N = \#\mathcal{V}(\mathcal{M})$ = number of dual cells, size of vector $\vec{\mu}$,
 $b_N^i \hat{=} \text{nodal basis function (“tent function”) of } \mathcal{S}_{1,0}^0(\mathcal{M}) \text{ belonging to the node inside } C_i$.

$u_N \hat{=} \text{piecewise linear interpolant of vertex values } \mu_i$

Note that u_N is not smooth across inner edges of \mathcal{M} . However, we do not care when computing $\mathbf{j} := \kappa(\mathbf{x}) \operatorname{grad} u_N$, because this flux is *only needed at edges of the dual mesh*, which lie inside triangles of \mathcal{M} (with the exception of single points that are irrelevant for the flux integrals).

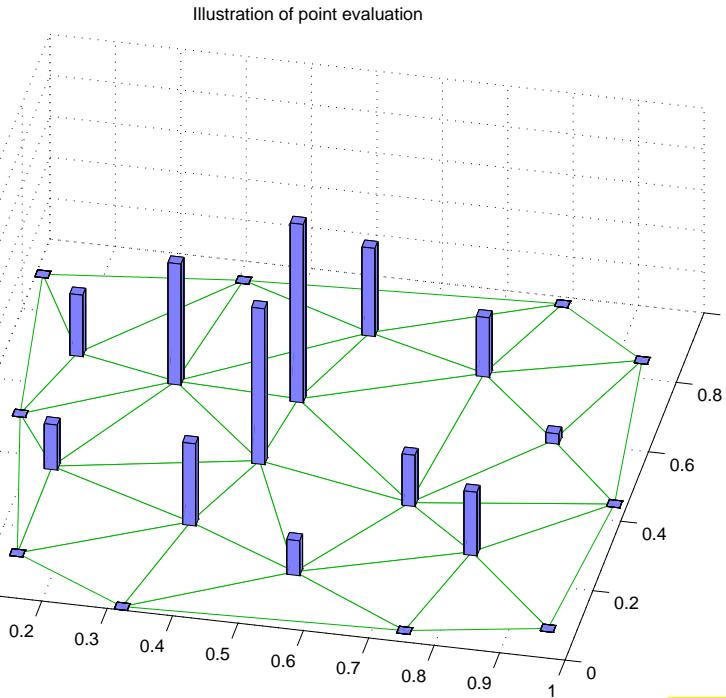


Fig. 129

vertex values μ_i on $\mathcal{V}(\mathcal{M})$

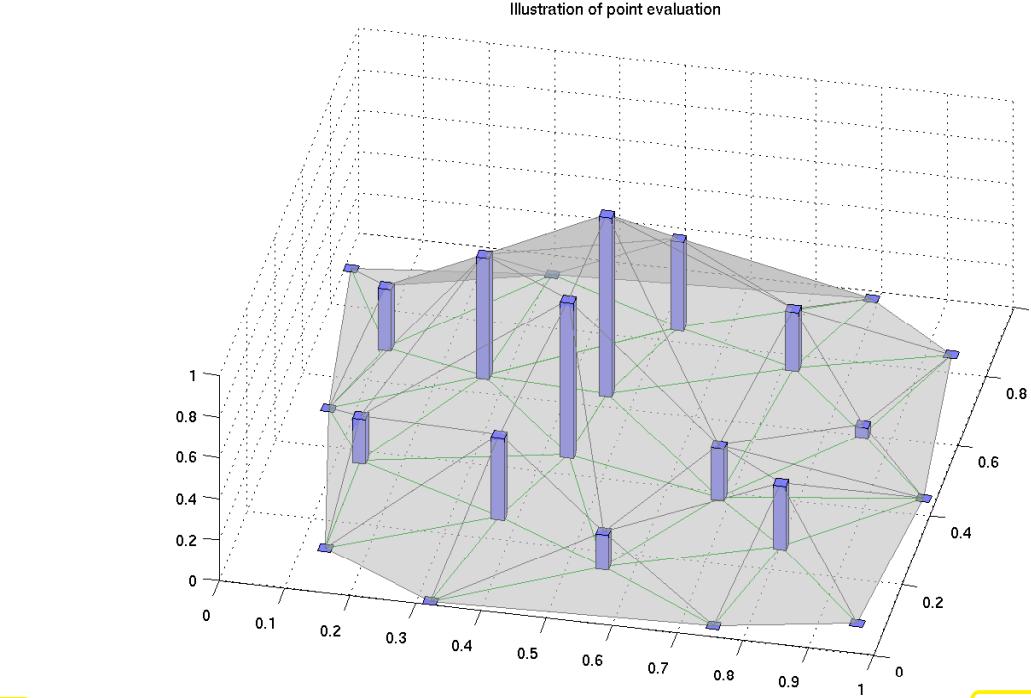


Fig. 130

p.w. linear interpolant $u_N := \mathbf{l}_1 \vec{\mu} \in \mathcal{S}_{1,0}^0(\mathcal{M})$

Choice of numerical flux:

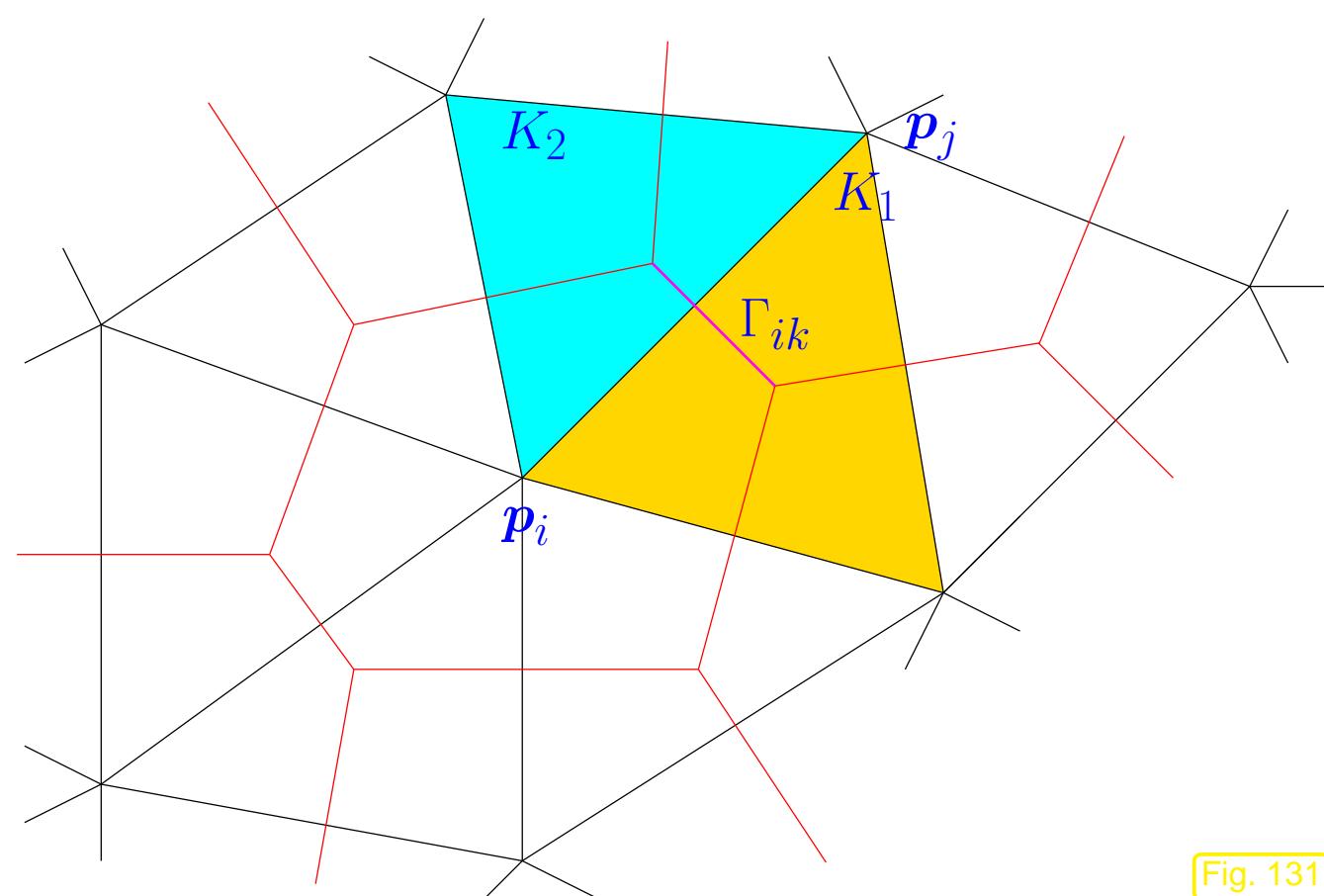
$$J_{ik} := - \int_{\Gamma_{ik}} \operatorname{grad} \mathbf{l}_1 \vec{\mu} \cdot \mathbf{n}_{ik} dS \quad (4.2.7)$$

(4.2.7) \rightarrow (4.2.2) \leftrightarrow one row of finite volume discretization matrix from

$$\sum_{k \in \mathcal{U}_i} \int_{\Gamma_{ik}} \mathbf{grad} \mathfrak{l}_1 \vec{\mu} \cdot \mathbf{n}_{ik} dS = \sum_{j \in \mathcal{U}_i} \mu_j \underbrace{\left(\sum_{k \in \mathcal{U}_i} \int_{\Gamma_{ik}} \mathbf{grad} b_N^j \cdot \mathbf{n}_{ik} dS \right)}_{= \text{matrix entry } (\mathbf{A})_{ij}} = \int_{\mathcal{C}_i} f(\mathbf{x}) d\mathbf{x} .$$

$$\Rightarrow \quad (\mathbf{A})_{ij} = \int_{\partial \mathcal{C}_i} \mathbf{grad} b_N^j \cdot \mathbf{n}_i dS . \quad (4.2.8)$$

$\mathbf{n}_i \hat{=} \text{exterior unit normal vector to } \partial \mathcal{C}_i$.



Part of the boundary of the control volume C_i :

$$\Gamma_i^K := \partial C_i \cap K .$$

Fig. 131

Now, consider $i \neq j \leftrightarrow$ off-diagonal entries of \mathbf{A} :

First, we recall that the intersection of the support of the “tent function” b_N^j with ∂C_i is located inside $K_1 \cup K_2$, see Fig. 131.

► $(\mathbf{A})_{ij} = \int_{\Gamma_i^K} \mathbf{grad} b_N^j \cdot \mathbf{n}_i \, dS + \int_{\Gamma_i^K} \mathbf{grad} b_N^j \cdot \mathbf{n}_i \, dS .$

Next observe that $\mathbf{grad} b_N^j$ is piecewise constant, which implies

$$\operatorname{div} \mathbf{grad} b_N^j = 0 \quad \text{in } K_1 , \quad \operatorname{div} \mathbf{grad} b_N^j = 0 \quad \text{in } K_2 . \quad (4.2.9)$$

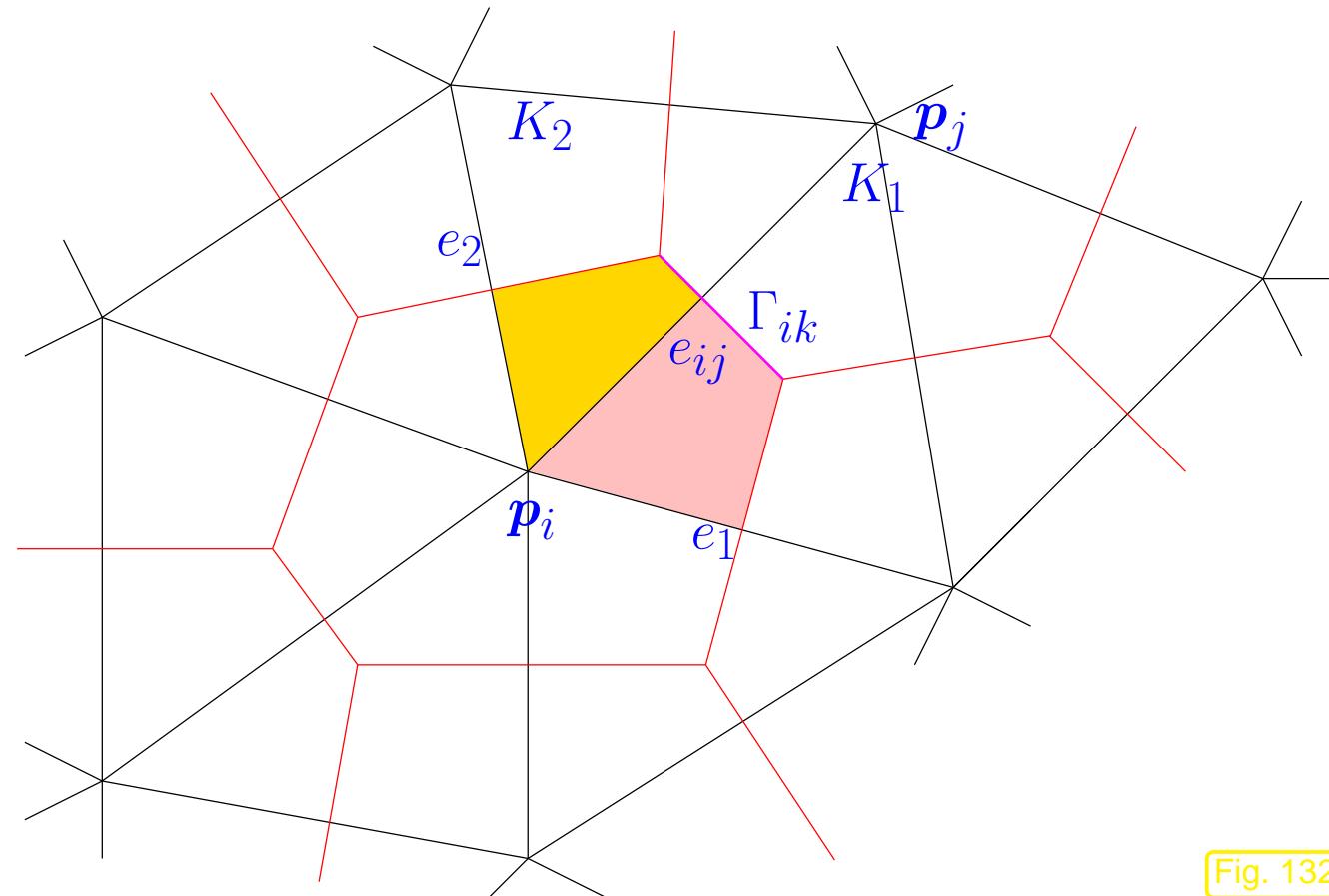


Fig. 132

Now apply Gauss' theorem Thm. 2.4.5 to the domains $C_i \cap K_1$ and $C_i \cap K_2$ (shaded in figure).

Also use again that $\mathbf{grad} b_N^j \equiv \text{const}$ on K_1 and K_2 .

►
$$\begin{aligned} (\mathbf{A})_{ij} = & \frac{1}{2} \int_{e_1} \mathbf{grad} b_N^j|_{K_1} \cdot \mathbf{n}_{e_1} dS + \frac{1}{2} \int_{e_{ij}} \mathbf{grad} b_N^j|_{K_1} \cdot \mathbf{n}_{e_{ij}}^1 dS \\ & + \frac{1}{2} \int_{e_{ij}} \mathbf{grad} b_N^j|_{K_2} \cdot \mathbf{n}_{e_{ij}}^2 dS + \frac{1}{2} \int_{e_2} \mathbf{grad} b_N^j|_{K_1} \cdot \mathbf{n}_{e_2} dS . \quad (4.2.10) \end{aligned}$$

On the other hand, an entry of finite element Galerkin matrix $\tilde{\mathbf{A}}$ based on linear Lagrangian finite element space $\mathcal{S}_1^0(\mathcal{M})$ can be computed as, see Sect. 3.2.5:

$$(\tilde{\mathbf{A}})_{ij} = \int_{K_1} \mathbf{grad} b_N^j \cdot \mathbf{grad} b_N^i dx + \int_{K_2} \mathbf{grad} b_N^j \cdot \mathbf{grad} b_N^i dx .$$

Conduct local integration by parts using Green's first formula from Thm. 2.4.7 and taking into account (4.2.9) and the linearity of the local shape functions

►
$$\begin{aligned} (\tilde{\mathbf{A}})_{ij} &= \int_{\partial K_1} (\mathbf{grad} b_N^j|_{K_1} \cdot \mathbf{n}_1) b_N^i dS + \int_{\partial K_2} (\mathbf{grad} b_N^j|_{K_2} \cdot \mathbf{n}_2) b_N^i dS \\ &= \frac{1}{2}|e_1| \mathbf{grad} b_N^j|_{K_1} \cdot \mathbf{n}_{e_1} + \frac{1}{2}|e_{ij}| \mathbf{grad} b_N^j|_{K_1} \cdot \mathbf{n}_{e_{ij}}^1 + \\ &\quad \frac{1}{2}|e_2| \mathbf{grad} b_N^j|_{K_2} \cdot \mathbf{n}_{e_2} + \frac{1}{2}|e_{ij}| \mathbf{grad} b_N^j|_{K_2} \cdot \mathbf{n}_{e_{ij}}^2 . \end{aligned}$$

This is the same value as for $(\mathbf{A})_{ij}$ from (4.2.10)! Similar considerations apply to the diagonal entries $(\mathbf{A})_{ii}$ and $(\tilde{\mathbf{A}})_{ii}$.



The finite volume discretization and the finite element Galerkin discretization spawn the same system matrix for the model problem (4.2.5).

5

Convergence and Accuracy

In this chapter we resume the discussion of Sect. 1.6 of accuracy of a Galerkin solution u_N of a variational boundary value problem.

More precisely, we are going to study *convergence*, see Rem. 1.6.2

Focus: **finite element Galerkin discretization** of *linear* scalar 2nd-order elliptic boundary value problems in 2D, 3D

Prerequisites (what you should know by now):

- Boundary value problems (from equilibrium models, diffusion models): Sects. 2.4, 2.6,
- Variational formulation: Sect. 2.8, see also (2.3.3), (2.8.15), (3.0.1),
- Some Sobolev spaces and their norms: Sect. 2.2
- Abstract Galerkin discretization: Sect. 3.1,
- Lagrangian finite elements: Sects. 3.4, 3.2.

5.1 Galerkin error estimates

Setting: linear variational problem (1.4.5) in the form

$$u \in V_0: \quad \mathbf{a}(u, v) = \ell(v) \quad \forall v \in V_0 , \quad (3.1.1)$$

- $V_0 \hat{=} (\text{real}) \text{ vector space, a space of functions } \Omega \mapsto \mathbb{R} \text{ for scalar 2nd-order elliptic variational problems,}$
 - $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R} \hat{=} \text{a bilinear form, see Def. 1.3.11,}$
 - $\ell : V_0 \mapsto \mathbb{R} \hat{=} \text{a linear form, see Def. 1.3.11,}$
- ☞ We want (3.1.1) to be related to a quadratic minimization problem (\rightarrow Def. 2.1.18):

Assumption 5.1.1. The bilinear form $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R}$ in (3.1.1) is symmetric and positive definite (\rightarrow Def. 2.1.22).

► \mathbf{a} supplies an inner product on V_0

► \mathbf{a} induces energy norm $\|\cdot\|_{\mathbf{a}}$ on V_0 (\rightarrow Def. 2.1.24)

☞ We want (3.1.1) to be well posed, see Rem. 2.3.11

Assumption 5.1.2. *The right hand side functional $\ell : V_0 \mapsto \mathbb{R}$ from (3.1.1) is continuous w.r.t. to the energy norm (\rightarrow Def. 2.1.24) induced by \mathbf{a} :*

$$\exists C > 0: |\ell(u)| \leq C \|u\|_{\mathbf{a}} \quad \forall u \in V_0 . \quad (2.2.1)$$

☞ An assumption to appease fastidious mathematicians:

Assumption 5.1.3. V_0 equipped with the energy norm $\|\cdot\|_{\mathbf{a}}$ is a *Hilbert space*, that is, complete.

Theorem 5.1.4 (Existence and uniqueness of solution of linear variational problem).

Under Assumptions 5.1.1–5.1.3 the linear variational problem has a unique solution $u \in V_0$.

This theorem is also known as **Riesz representation theorem** for continuous linear functionals.

Remark 5.1.5 (Well-posed 2nd-order linear elliptic variational problems).

For instance, Assumption 5.1.1 is satisfied for the bilinear form

$$\mathbf{a}(u, v) := \int_{\Omega} (\boldsymbol{\alpha}(\mathbf{x}) \operatorname{grad} u) \cdot \operatorname{grad} v \, d\mathbf{x}, \quad u, v \in H_0^1(\Omega), \quad (5.1.6)$$

and uniformly positive definite (\rightarrow Def. 2.1.9) coefficient tensor $\boldsymbol{\alpha} : \Omega \mapsto \mathbb{R}^{d,d}$, see Sect. 2.1.3.

For the right hand side functional

$$\ell(v) := \int_{\Omega} f(\mathbf{x})v(\mathbf{x}) \, d\mathbf{x} + \int_{\partial\Omega} h(\mathbf{x})v(\mathbf{x}) \, dS, \quad v \in H^1(\Omega),$$

we found in Sect. 2.2, see (2.2.15), and Rem. 2.9.5 that $f \in L^2(\Omega)$ and $h \in L^2(\partial\Omega)$ ensures Assumption 5.1.2.

Assumption 5.1.3 for \mathbf{a} from (5.1.6) is a deep result in the theory of Sobolev spaces [10, Sect. 5.2.3, Thm. 2].

Now consider Galerkin discretization of (3.1.1) based on Galerkin trial/test space $V_{0,N} \subset V_0$, $N := \dim V_{0,N} < \infty$:

$$u_N \in V_{0,N}: \quad \mathbf{a}(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N}. \quad (3.1.3)$$

Thm. 3.1.4: existence and uniqueness of Galerkin solution $u_N \in V_{0,N}$

Goal: bound *relevant norm* of **discretization error** $u - u_N$

Here: relevant norm = energy norm $\|\cdot\|_a$

Why is the energy norm a “relevant norm” ?

➤ Bounds of $\|u - u_N\|_a$ provide bounds for the *error in energy*, see Rem. 1.6.7, (1.6.10)

$$\begin{aligned} |J(u) - J(u_N)| &= \frac{1}{2} |\mathbf{a}(u, u) - \mathbf{a}(u_N, u_N)| = \left| \frac{1}{2} \mathbf{a}(u + u_N, u - u_N) \right| \\ &\stackrel{(2.1.25)}{\leq} \|u - u_N\|_a \cdot \|u + u_N\|_a . \end{aligned}$$

(No doubt, energy is a key quantity for the solution of an equilibrium problem, which is defined as the minimizer of a potential energy functional.)

Other “relevant norms” were discussed in Sects. 1.6.1, 2.2:

- the mean square norm or $L^2(\Omega)$ -norm, see Def. 2.2.5,
- the supremum norm or $L^\infty(\Omega)$ -norm, see Def. 1.6.4.

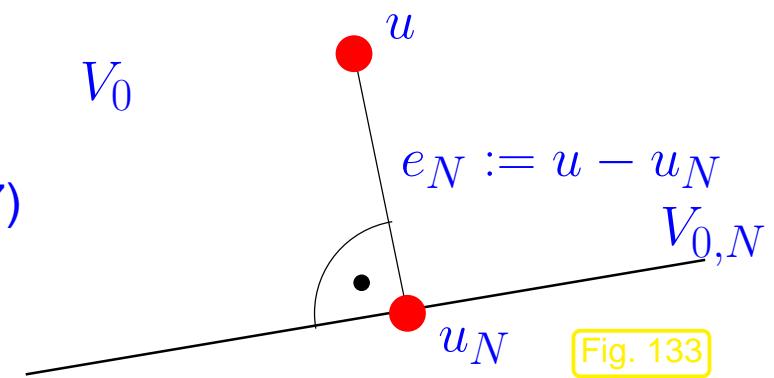
The Galerkin approach allows a remarkably simple bound of the energy norm of the discretization error $u - u_N$:

$$\begin{aligned} \mathbf{a}(u, v) &= \ell(v) \quad \forall v \in V_0, \\ \mathbf{a}(u_N, v_N) &= \ell(v) \quad \forall v_N \in V_{0,N} \end{aligned} \quad \stackrel{V_{0,N} \subset V_0}{\implies} \quad \mathbf{a}(u - u_N, v_N) = 0 \quad \forall v_N \in V_{0,N}.$$

Galerkin orthogonality

$$\mathbf{a}(u - u_N, v_N) = 0 \quad \forall v_N \in V_{0,N}. \quad (5.1.7)$$

[Geometric meaning for inner product $a(\cdot, \cdot) \rightarrow$]



► Discretization error $e_N := u - u_N$ “ $a(\cdot, \cdot)$ -orthogonal” to discrete trial/test space V_N

Remark 5.1.8. If $a(\cdot, \cdot)$ is inner product on V : “Phythagoras’ theorem” \rightarrow Fig. 133

$$\|u - u_N\|_a^2 = \|u\|_a^2 - \|u_N\|_a^2. \quad (5.1.9)$$

(5.1.9) ➤ simple formula for computation of energy norm of Galerkin discretization error in numerical experiments with known \underline{u} .

Theorem 5.1.10 (Cea's lemma).

Under Assumptions 5.1.1–5.1.3 the energy norm of the Galerkin discretization error satisfies

$$\|u - u_N\|_a = \inf_{v_N \in V_{0,N}} \|u - v_N\|_a .$$

Proof. Use bilinearity of a and Galerkin orthogonality (5.1.7): for any $v_N \in V_{0,N}$

$$\|u - u_N\|_a^2 = a(u - u_N, u - u_N) = a(u - v_N, u - u_N) + \underbrace{a(v_N - u_N, u - u_N)}_{=0} .$$

Next, use the Cauchy-Schwarz inequality for the inner product a :

$$\begin{aligned} a(u, v) &\leq \|u\|_a \|v\|_a \quad \forall u, v \in V_0 . \\ \Rightarrow \|u - u_N\|_a^2 &\leq \|u - v_N\|_a \cdot \|u - u_N\|_a , \end{aligned}$$

and cancel one factor $\|u - u_N\|_a$.

5.1

Optimality of Galerkin solutions:

$$\underbrace{\|u - u_N\|_a}_{\text{(norm of) discretization error}} = \underbrace{\inf_{v_N \in V_{0,N}} \|u - v_N\|_a}_{\text{best approximation error}} , \quad (5.1.11)$$

- ☞ To assess accuracy of Galerkin solution: study capability of $V_{0,N}$ to approximate u !

- “Monotonicity” of best approximation: consider different trial/test spaces

$$\begin{array}{c} V_{0,N}, V'_{0,N} \subset V_0 , \\ V_{0,N} \subset V'_{0,N} \end{array} \Rightarrow \inf_{v_N \in V'_{0,N}} \|u - v_N\|_a \leq \inf_{v_N \in V_{0,N}} \|u - v_N\|_a .$$

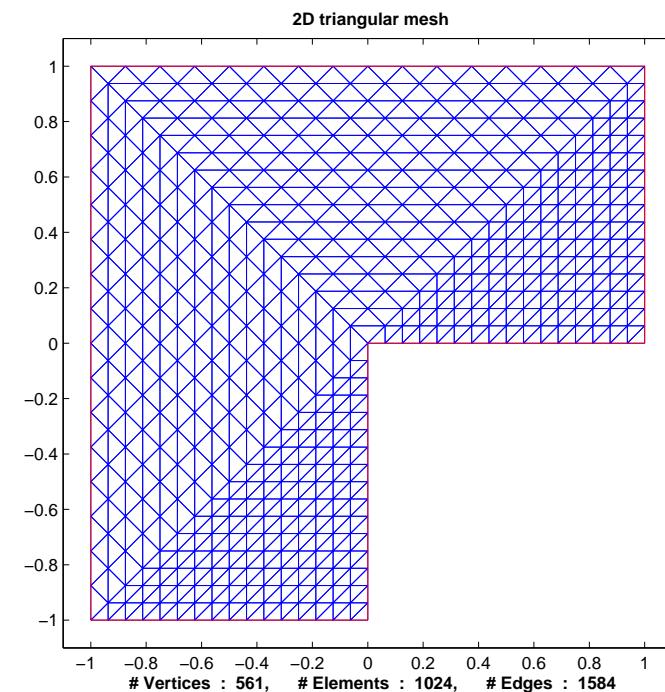
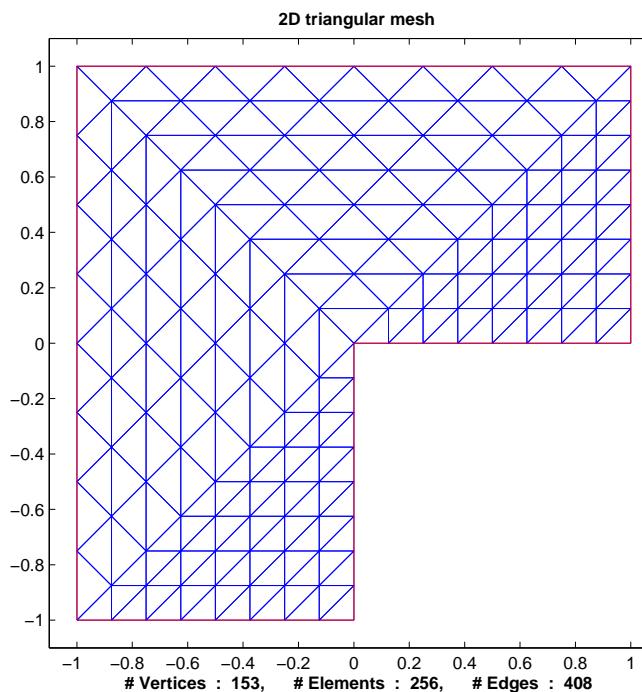
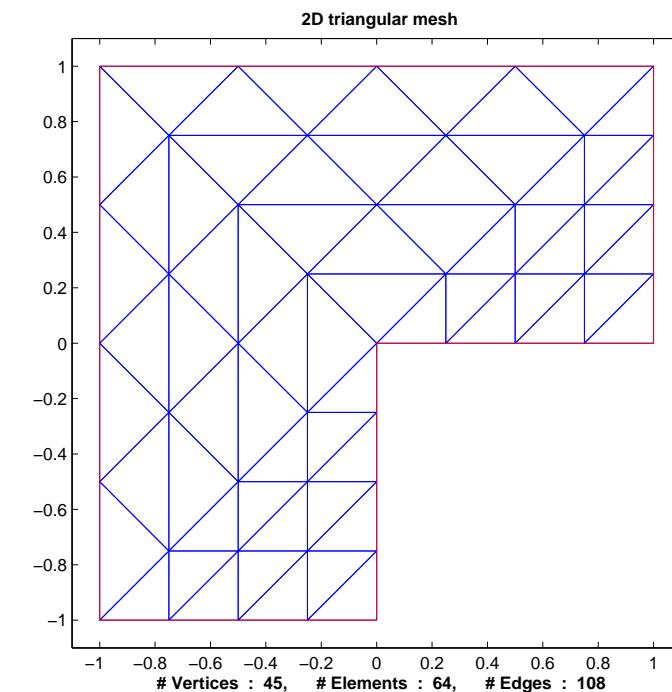
Enhance accuracy by enlarging (“refining”) trial space.

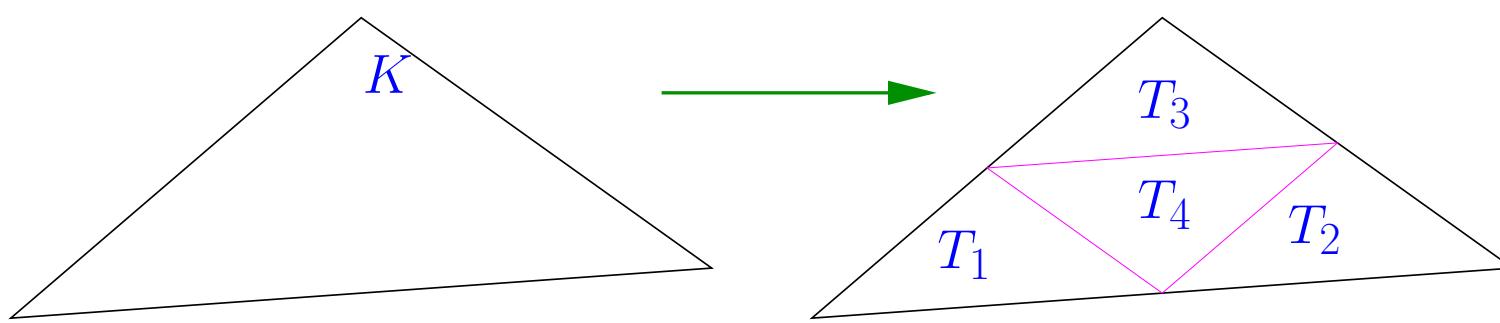
Now return to finite element Galerkin discretization of linear 2nd-order elliptic variational problems.

How to achieve refinement of FE space ?

- **h-refinement:** replace \mathcal{M} (underlying $V_{0,N}$) $\rightarrow \mathcal{M}'$ (underlying larger discrete trial space $V'_{0,N'}$)

Example 5.1.12 (regular refinement of triangular mesh in 2D).





Regular refinement of triangle K into four congruent triangles T_1, T_2, T_3, T_4

◇

- **p-refinement:** replace $V_{0,N} := \mathcal{S}_p^0(\mathcal{M}), p \in \mathbb{N}$ with $V'_{0,N} := \mathcal{S}_{p+1}^0(\mathcal{M}) \Rightarrow V_{0,N} \subset V'_{0,N}$

The extreme case of p -refinement amounts to the use of *global* polynomials on Ω as trial and test functions ➤ (polynomial) **spectral Galerkin method**, see Sect. 1.5.1.1.

Combination of h-refinement and p-refinement ?

OF COURSE **(hp-refinement, [18])**

5.2

5.2 Empirical Convergence of FEM

Recall from Sect. 1.6.2:

Crucial: convergence is an *asymptotic notion* !

sequence of discrete models \Rightarrow sequence of approximate solutions $(u_N^{(i)})_{i \in \mathbb{N}}$
 \Rightarrow study sequence $(\|u_N^{(i)} - u\|)_{i \in \mathbb{N}}$

created by *variation* of a **discretization parameter**:

In this section some numerical experiments will demonstrate

- meaningful notions of “discretization parameters”,
- qualitative behaviors of the sequence $(\|u_N^{(i)} - u\|)_{i \in \mathbb{N}}$ we may expect,

for Lagrangian finite element discretization of linear scalar 2nd-order elliptic variational problems (\rightarrow Sect. 2.8).

Sequences of discrete models will be generated by either h -refinement or p -refinement.

Model problem: Dirichlet problem for Poisson equation:

$$-\Delta u = f \in L^2(\Omega) \quad \text{in } \Omega, \quad u = g \in C^0(\partial\Omega) \quad \text{on } \partial\Omega. \quad (5.2.1)$$

Example 5.2.2 (Convergence of linear and quadratic Lagrangian finite elements in energy norm).

Setting: $\Omega =]0, 1[^2$, $f(x_1, x_2) = 2\pi^2 \sin(\pi x_1) \sin(\pi x_2)$, $\mathbf{x} \in \Omega$, $g = 0$

➤ Smooth solution $u(x, y) = \sin(\pi x) \sin(\pi y)$.

• Galerkin finite element discretization based on triangulas meshes and

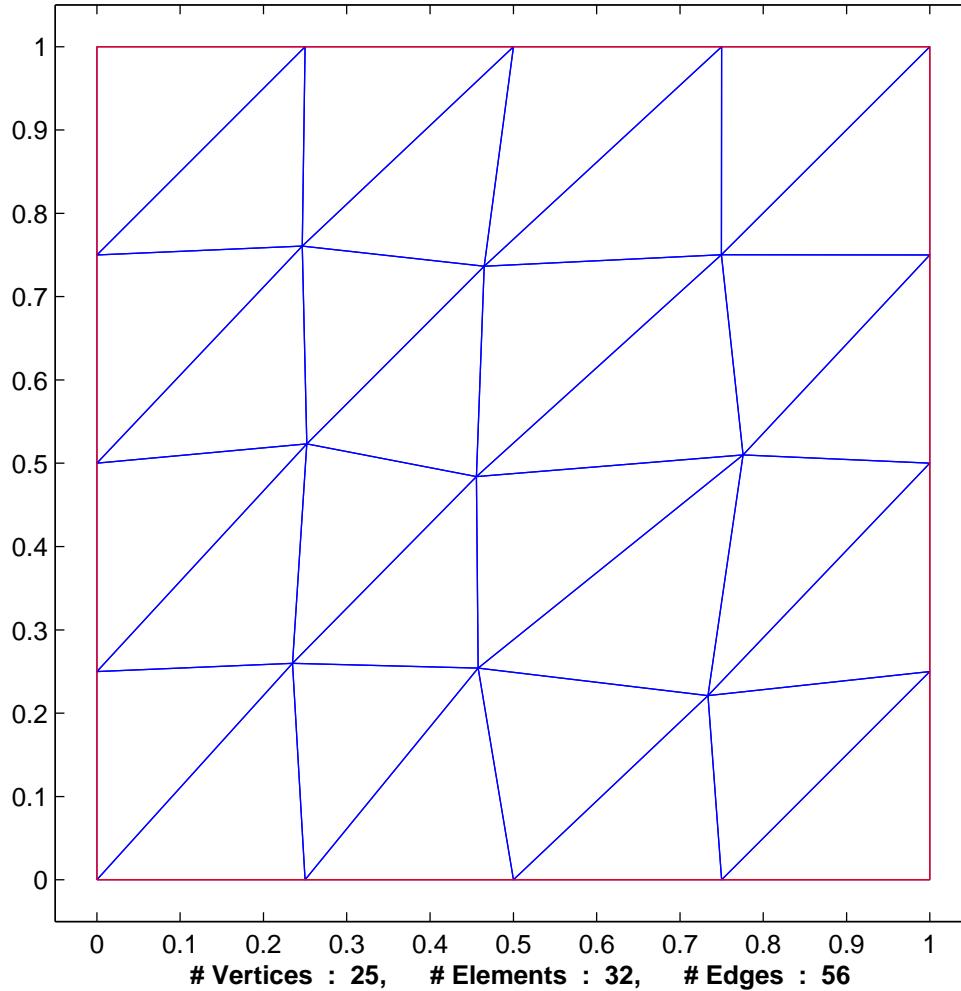
– linear Lagrangian finite elements, $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M}) \subset H_0^1(\Omega)$ (\rightarrow Sect. 3.2),

- quadratic Lagrangian finite elements, $V_{0,N} = \mathcal{S}_{2,0}^0(\mathcal{M}) \subset H_0^1(\Omega)$ (\rightarrow Ex. 3.4.2),
- quadrature rule (3.5.38) for assembly of local load vectors (\rightarrow Sect. 3.5.4),

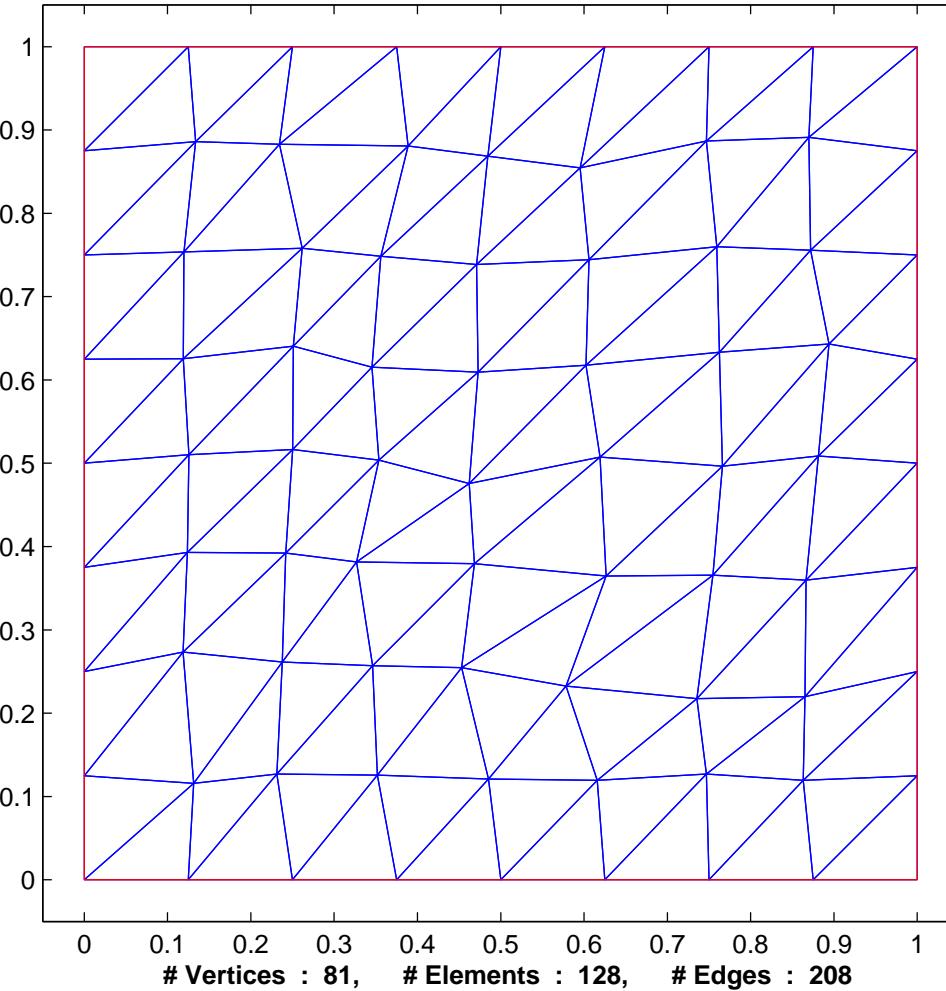
Monitored: $H^1(\Omega)$ -semi-norm (\rightarrow Def. 2.2.10) of the Galerkin discretization error $u - u_N$

- Approximate (*) computation of $|u - u_N|_{H^1(\Omega)}$ on a sequence of meshes (created by successive regular refinement (\rightarrow Ex. 5.1.12) of coarse initial mesh)
- (*): use of local quadrature rule (3.5.38) (on current FE mesh)

2D triangular mesh



2D triangular mesh



Unstructured triangular meshes of $\Omega =]0, 1[^2$ (two coarsest specimens)

Focus on **asymptotics** entails studying a

norm of the discretization error as function of a (real, cardinal) **discretization parameter**.

The discretization parameter must be linked to the **resolution** (“capability to approximate generic solution”) of the Galerkin trial/test space $V_{0,N}$. Possible choices are

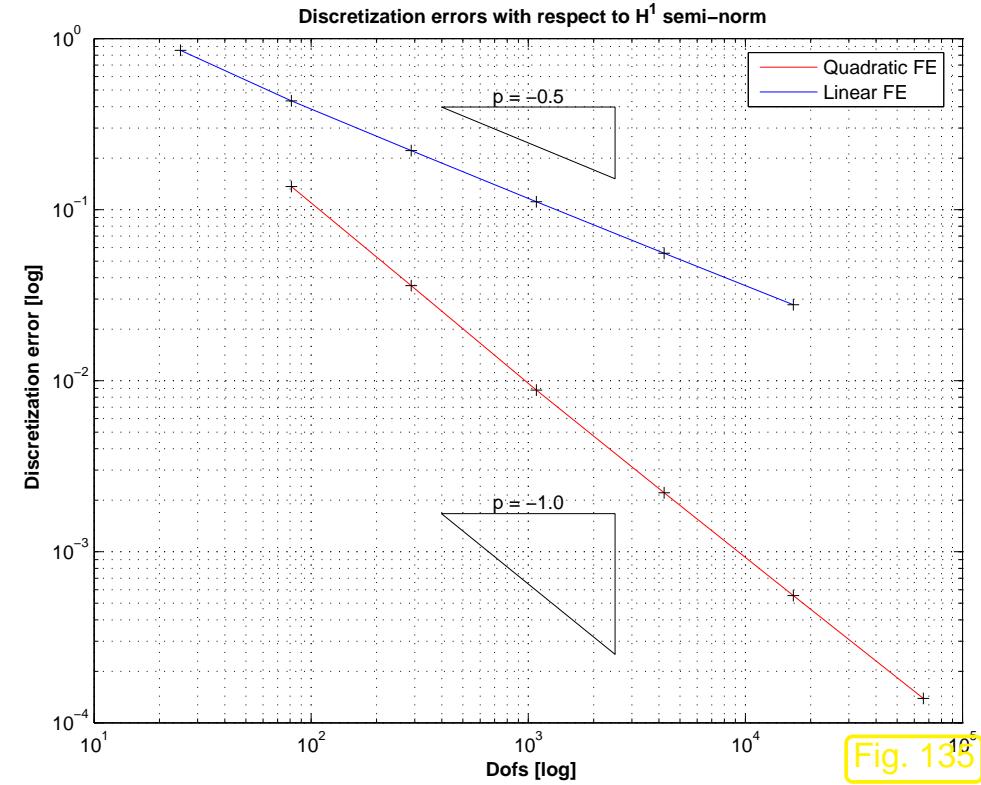
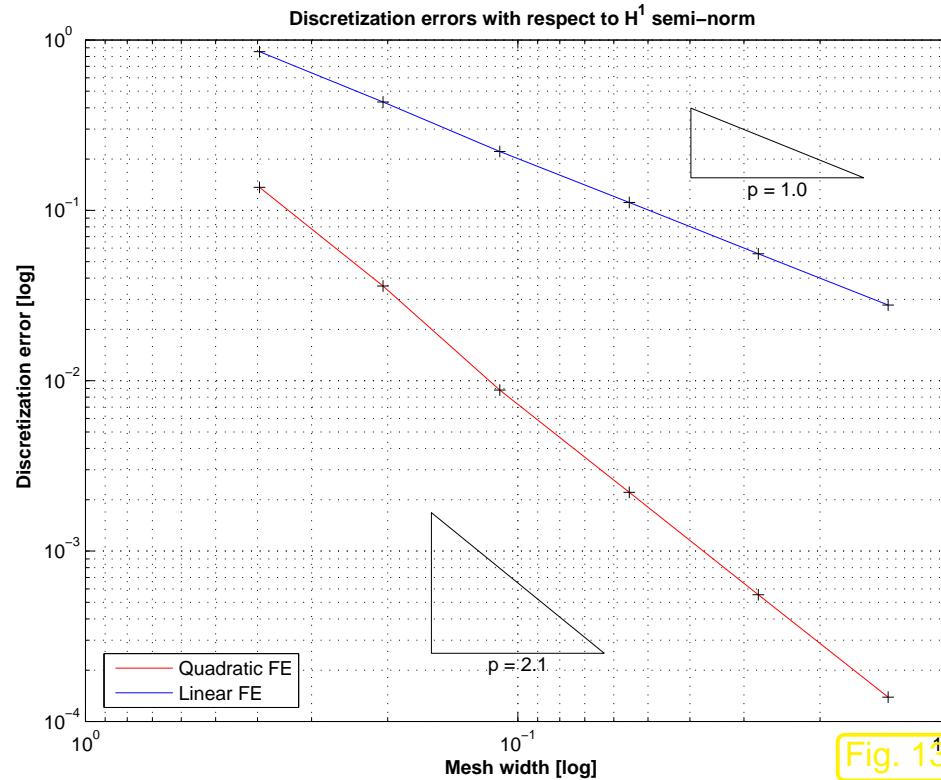
- $N := \dim V_{0,N}$ as a measure of the “cost” of a discretization, see Sect. 1.6.2,
- the maximum “size” of mesh cells, expressed by the mesh width h_M (\rightarrow Def. 5.2.3), see below.

Definition 5.2.3 (Mesh width).

*Given a mesh $M = \{K\}$, its **mesh width** h_M is defined as*

$$h_M := \max\{\text{diam } K : K \in M\} , \quad \text{diam } K := \max\{|\mathbf{p} - \mathbf{q}| : \mathbf{p}, \mathbf{q} \in K\} .$$

This generalizes the concept of “mesh width” introduced in Sect. 1.5.1.2.



$H^1(\Omega)$ -semi-norm of discretization error on unit square ($\text{—} \leftrightarrow p = 1$, $\text{—} \leftrightarrow p = 2$)

Recall type of convergence (algebraic convergence vs. exponential convergence) from Def. 1.6.19 and how to detect them in a numerical experiment by inspecting appropriate graphs, see Rem. 1.6.21.

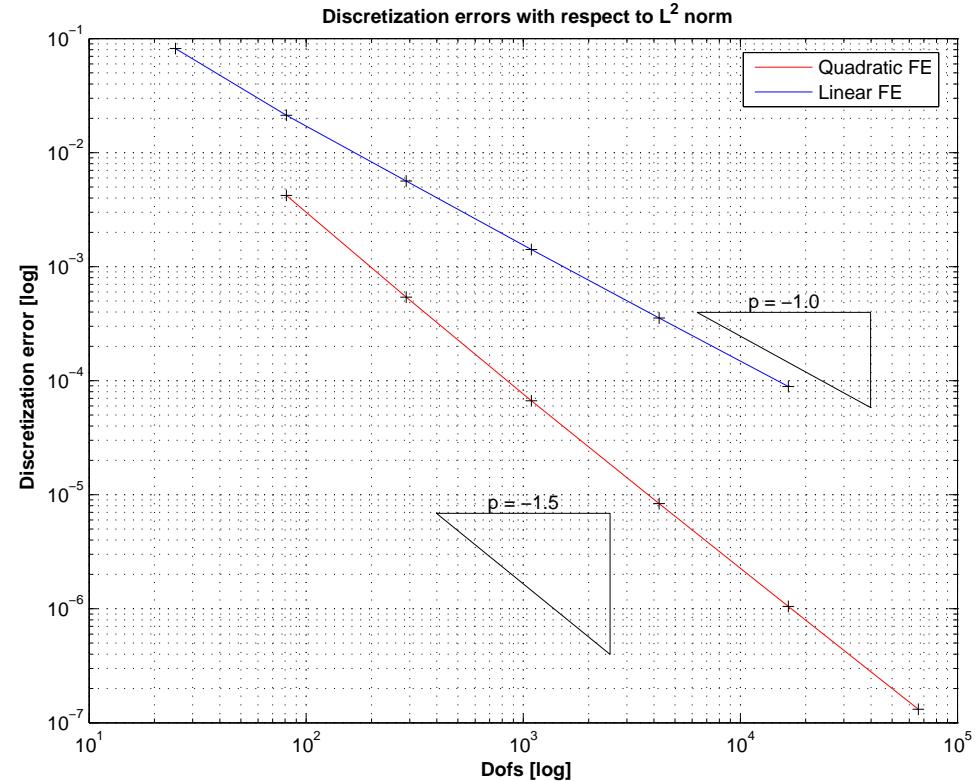
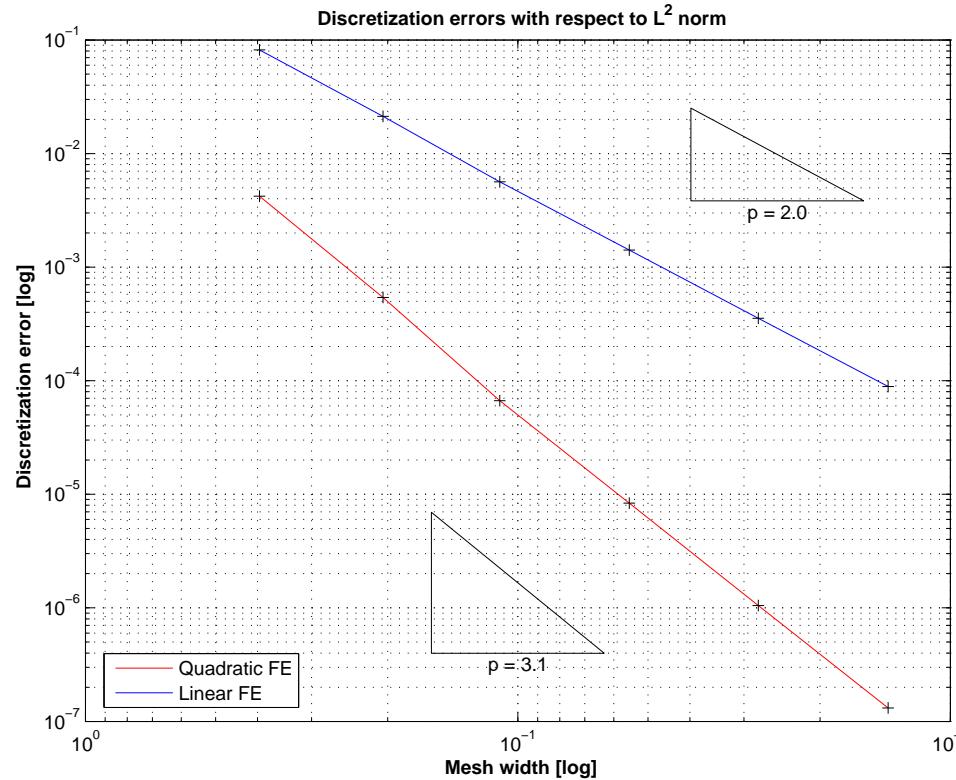
- Observations:
- Algebraic rates of convergence in terms of N and h
 - Quadratic Lagrangian FE converge with double the rate of linear Lagrangian FE



Example 5.2.4 (Convergence of linear and quadratic Lagrangian finite elements in L^2 -norm).

Setting as above in Ex. 5.2.2, $\Omega =]0, 1[^2$.

Monitored: **asymptotics** of the $L^2(\Omega)$ -semi-norm of the Galerkin discretization error (approximate computation of $\|u - u_N\|_{L^2(\Omega)}$ by means of local quadrature rule (3.5.38) on a sequence of meshes created by successive regular refinement (\rightarrow Ex. 5.1.12) of coarse initial mesh).



$L^2(\Omega)$ -norm of discretization error on unit square ($\textcolor{blue}{-} \leftrightarrow p = 1$, $\textcolor{red}{-} \leftrightarrow p = 2$)

- Observations:
- Linear Lagrangian FE ($p = 1$) $\Rightarrow \|u - u_N\|_0 = O(h_{\mathcal{M}}^2) = O(N^{-1})$
 - Quadratic Lagrangian FE ($p = 2$) $\Rightarrow \|u - u_N\|_0 = O(h_{\mathcal{M}}^3) = O(N^{-1.5})$

For the “conversion” of convergence rates with respect to the mesh width $h_{\mathcal{M}}$ and $N := \dim \mathcal{S}_p^0(\mathcal{M})$, note that in 2D for Lagrangian finite element spaces with fixed polynomial degree (\rightarrow Sect. 3.4) and meshes created by global (that is, carried out everywhere) regular refinement

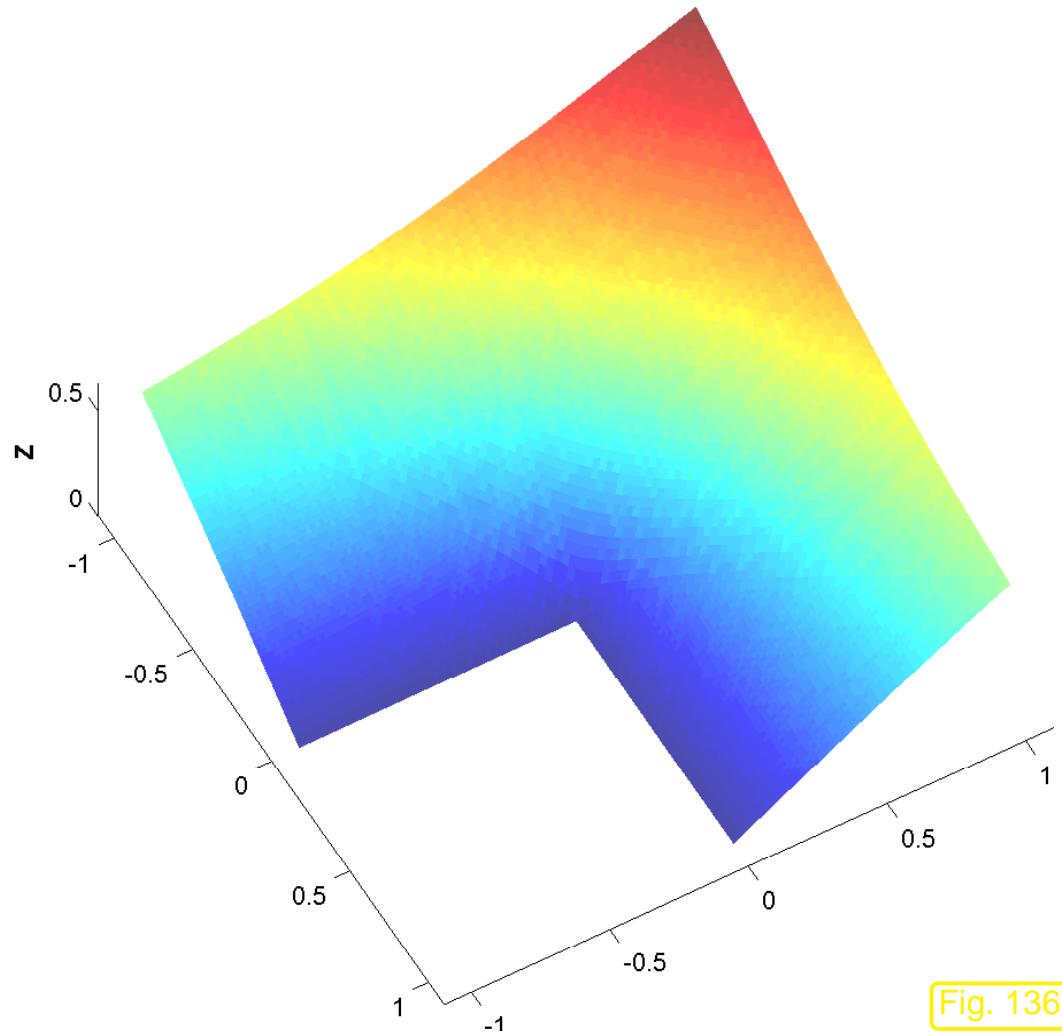
$$N = O(h_{\mathcal{M}}^{-2}). \quad (5.2.5)$$



Example 5.2.6 (h -convergence of Lagrangian FEM on L-shaped domain).

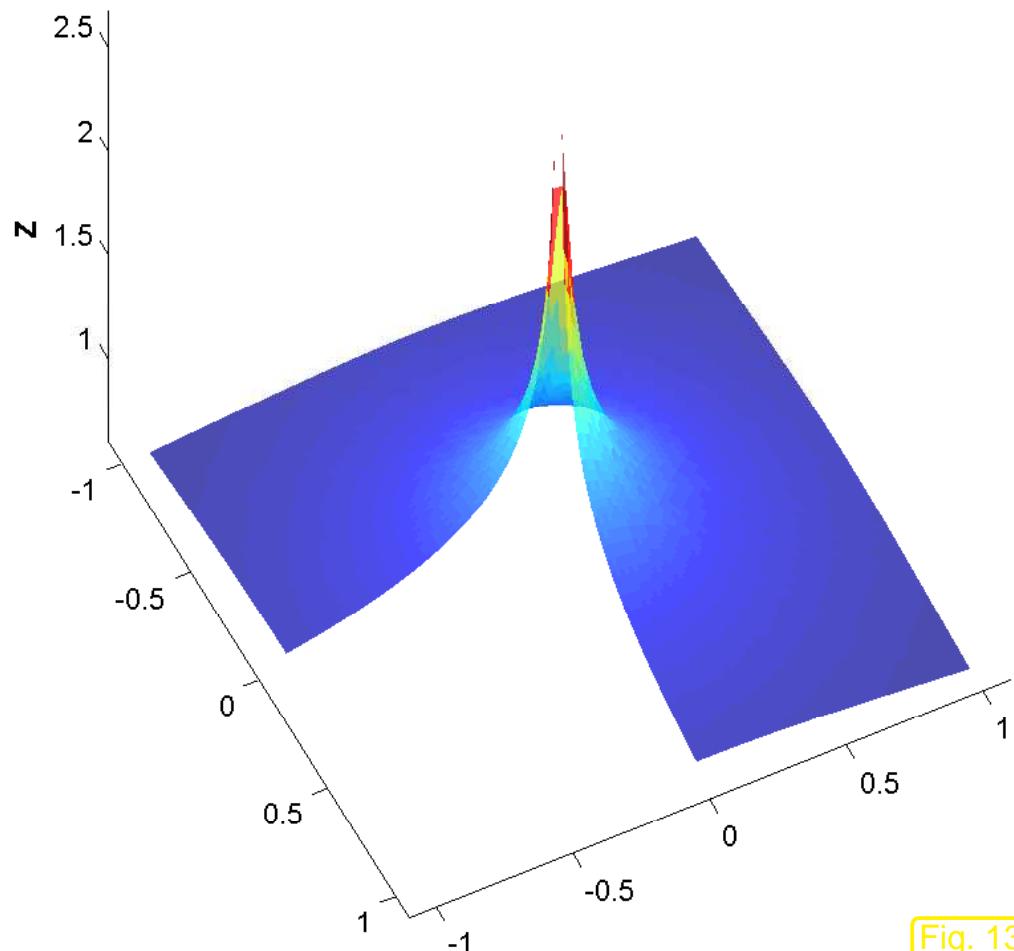
Setting: Model problem (5.2.1) on $\Omega =]-1, 1[^2 \setminus (]0, 1[\times]-1, 0[)$, exact solution (in polar coordinates)

$$u(r, \varphi) = r^{2/3} \sin(2/3\varphi) \quad \Rightarrow \quad f = 0, g = u|_{\partial\Omega}.$$



Exact solution u

Fig. 136



Norm of gradient $\|\mathbf{grad} u\|$

Fig. 137

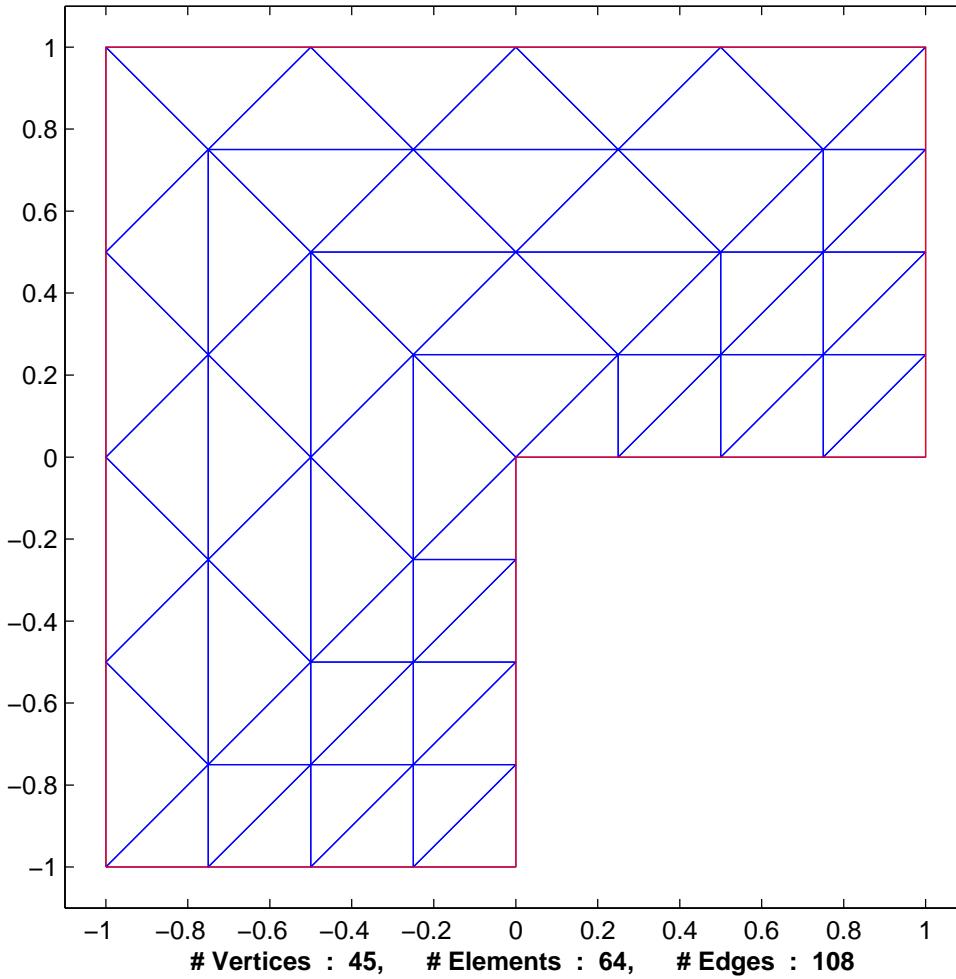
Note: $\mathbf{grad} u$ has a singularity at 0 , that is, “ $\|\mathbf{grad} u(0)\| = \infty$ ”.

- Galerkin finite element discretization based on triangulas meshes and

- linear Lagrangian finite elements, $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M}) \subset H_0^1(\Omega)$ (\rightarrow Sect. 3.2),
- quadratic Lagrangian finite elements, $V_{0,N} = \mathcal{S}_{2,0}^0(\mathcal{M}) \subset H_0^1(\Omega)$ (\rightarrow Ex. 3.4.2),
- linear/quadratic interpolation of Dirichlet data to obtain offset function $u_0 \in \mathcal{S}_{p,0}^0(\mathcal{M})$, $p = 1, 2$,
see Sect. 3.5.5, Ex. 3.5.43.

Sequence of meshes created by successive regular refinement (\rightarrow Ex. 5.1.12) of coarse initial mesh,
see Fig. 138.

2D triangular mesh



2D triangular mesh

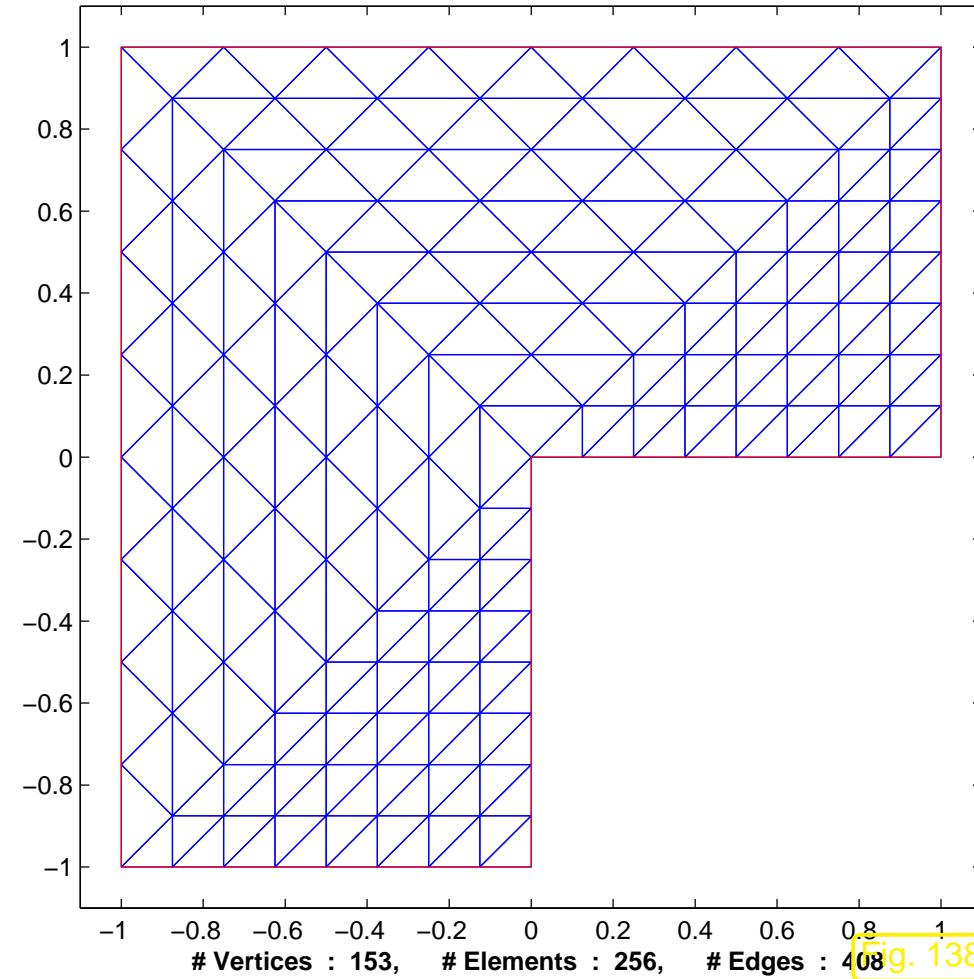
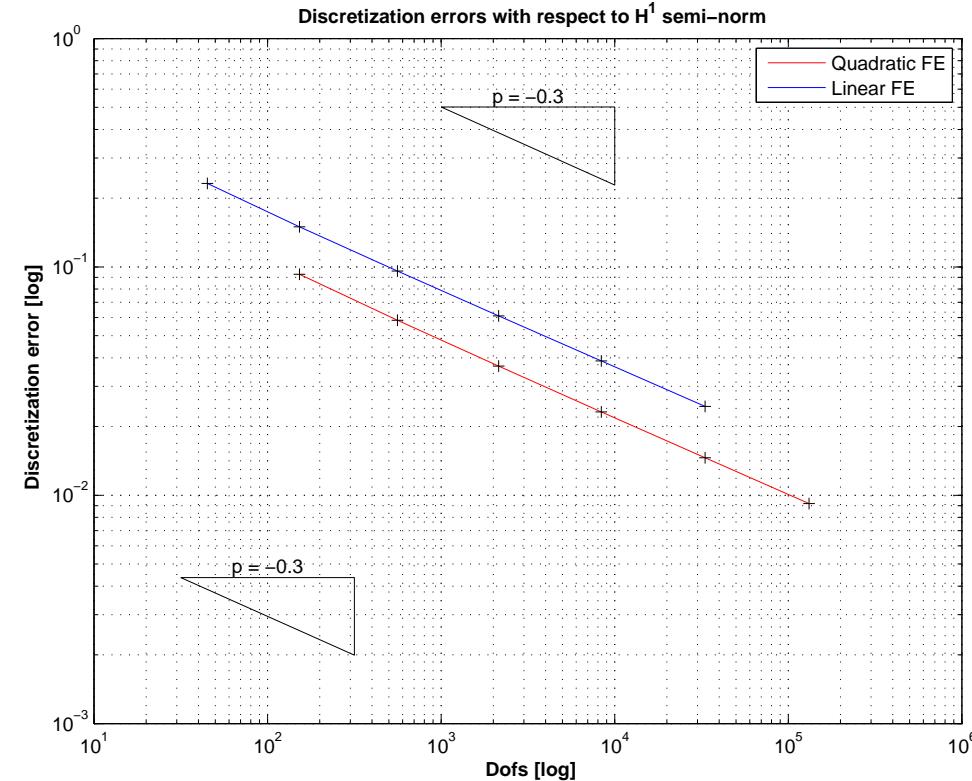
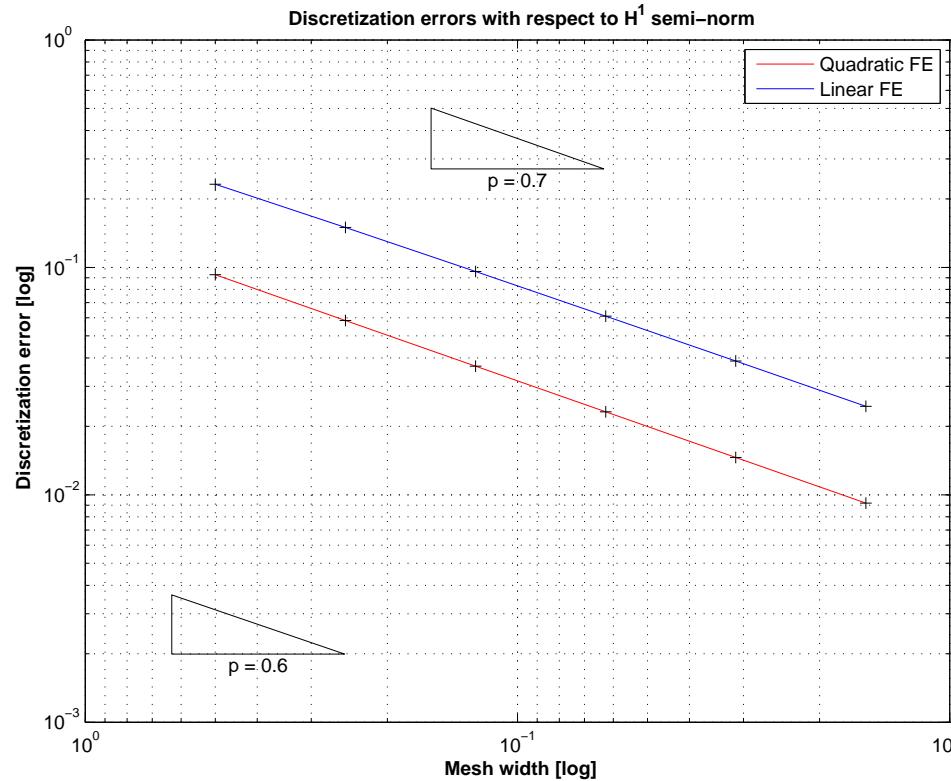


Fig. 138

Unstructured triangular meshes of $\Omega =]-1, 1[^2 \setminus ([0, 1] \times [-1, 0])$ (two coarsest specimens)

Approximate computation of $|u - u_N|_{H^1(\Omega)}$ by using local quadrature formula (3.5.38) on FE meshes.



$H^1(\Omega)$ -semi-norm of discretization error on “L-shaped” domain ($\textcolor{blue}{—} \leftrightarrow p = 1$, $\textcolor{red}{—} \leftrightarrow p = 2$)

Observations:

- For both $p = 1, 2$: $\|u - u_N\|_1 = O(N^{-1/3})$
- No gain from higher polynomial degree

Conjecture: singularity of $\text{grad } u$ at $x = 0$ seems to foil faster algebraic convergence of quadratice Lagrangian finite element solutions!



Example 5.2.7 (Convergence of Lagrangian FEM for p -refinement).

- Model BVP as in Ex. 5.2.2 ($\Omega =]0, 1[^2$) and Ex. 5.2.6 (L-shaped domain $\Omega =]-1, 1[^2 \setminus (]0, 1[\times]-1, 0[)$).
- Galerkin finite element discretization based on $\mathcal{S}_p^0(\mathcal{M})$, $p = 1, 2, 3, 5, 6, 7, 8, 9, 10$, built on a *fixed* coarse triangular mesh of Ω .



p -refinement

Monitored: $H^1(\Omega)$ -semi-norm (energy norm) and $L^2(\Omega)$ -norm of discretization error as functions of polynomial degree p and $N := \dim \mathcal{S}_p^0(\mathcal{M})$.

(Computation of norms by means of local quadrature rule of order 19!. This renders the error in norm computations introduced by numerical quadrature negligible.)

Meaningful discretization parameters for asymptotic study of error norms:

- polynomial degree p for Lagrangian finite element space,
- $N := \dim V_{0,N}$ as a measure of the “cost” of a discretization, see Sect. 1.6.2.

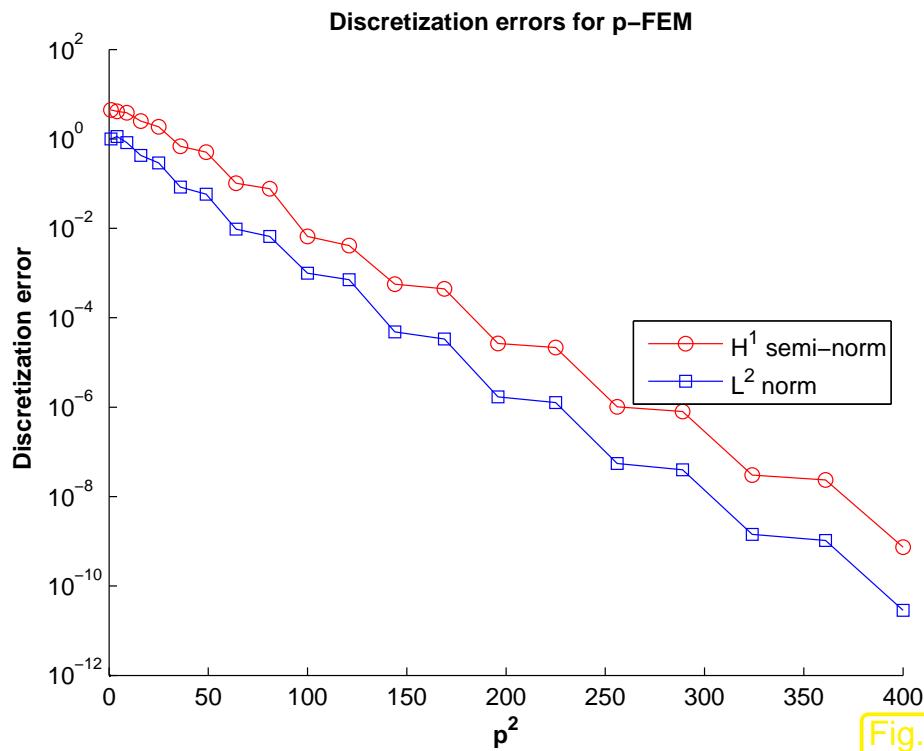


Fig. 139

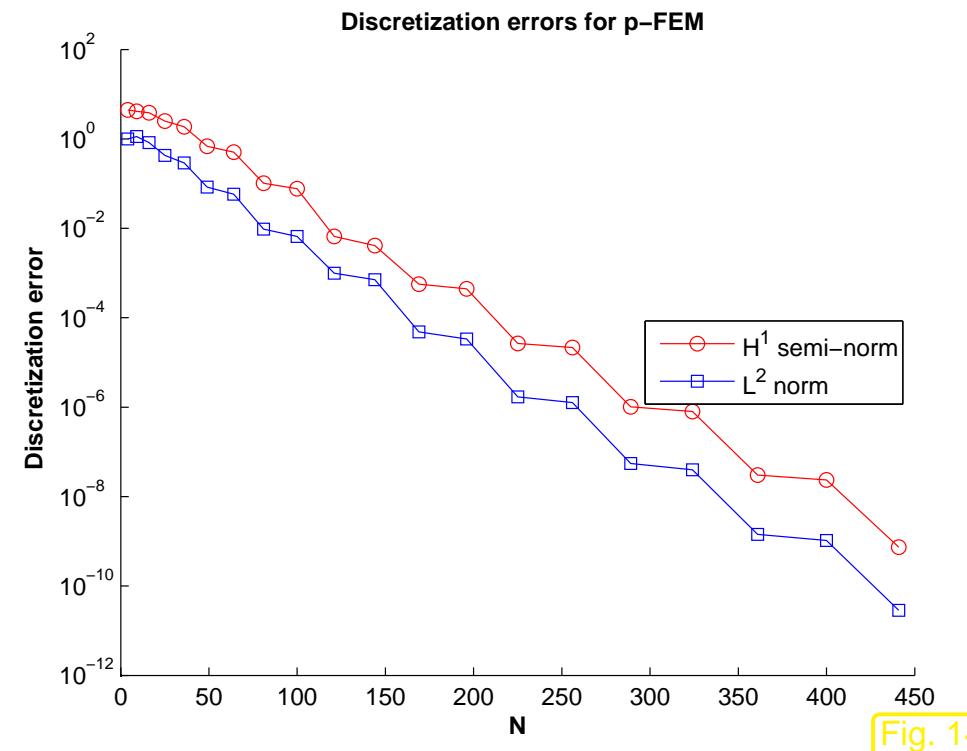
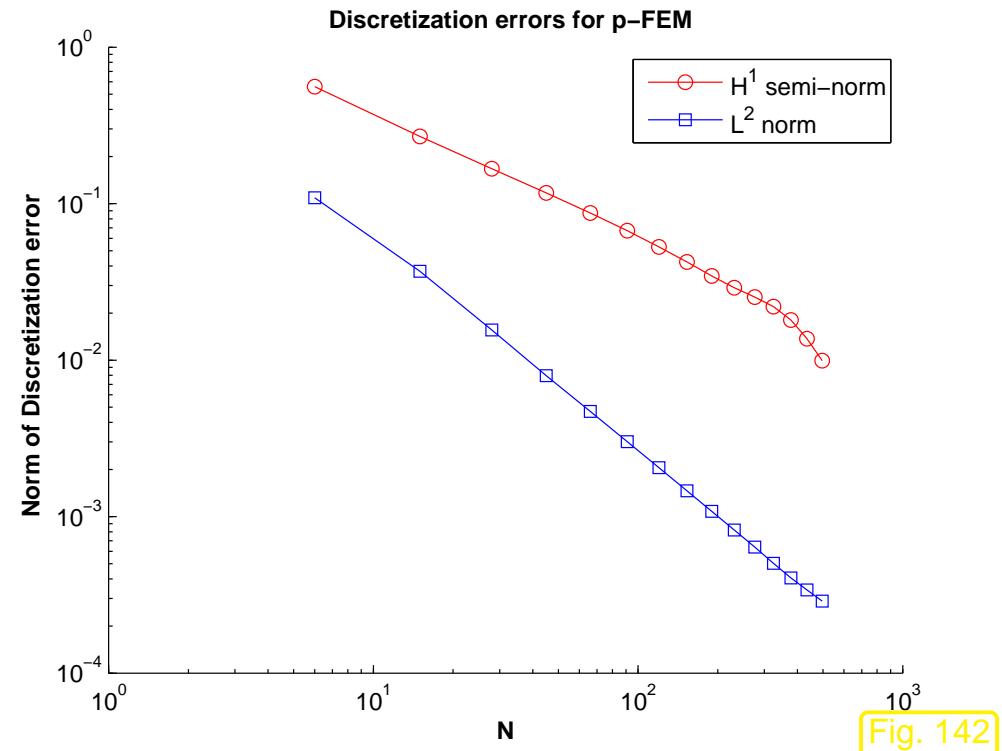
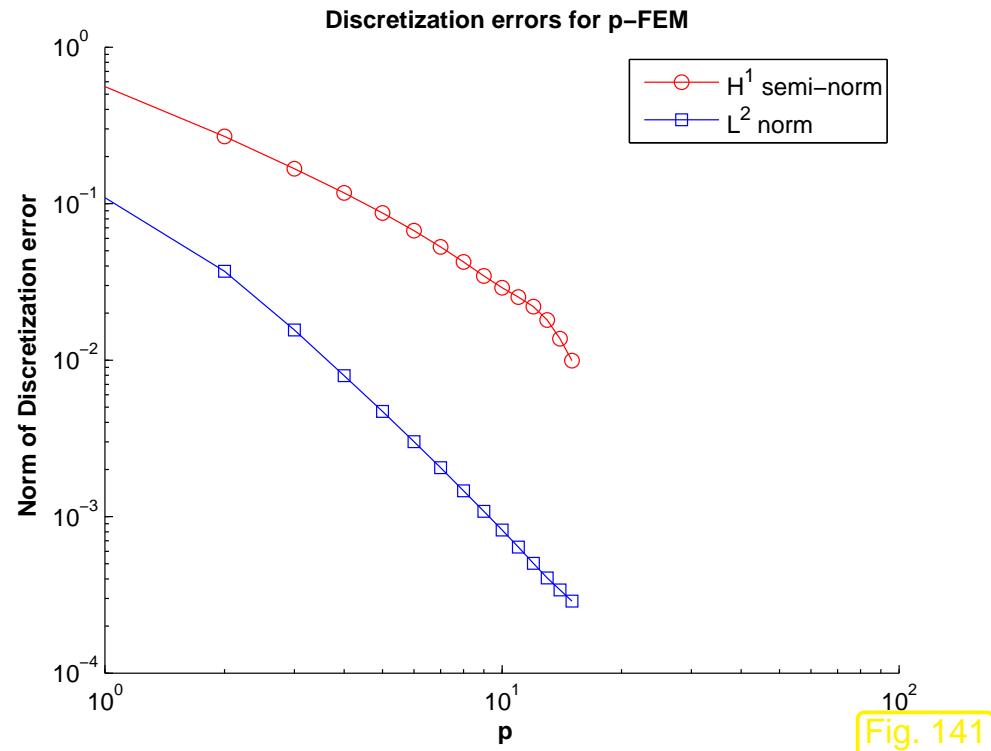


Fig. 140

$\Omega = [0, 1]^2$: behavior of $|u - u_N|_{H^1(\Omega)}$ for different polynomial degrees.
Lagrangian FEM: p -convergence for smooth (analytic) solution

Observation: exponential convergence of FE discretization error, cf. the behavior of the discretization error of spectral collocation and polynomial spectral Galerkin methods in 1D, Ex. 1.6.18.



Lagrangian FEM: p -convergence for solution with singular gradient

Observation: Only algebraic convergence of FE discretization error!

The suspect: “singular behavior” of $\text{grad } u$ at $x = 0$.



5.3 Finite element error estimates

We are interested in **a priori estimates** of norms of the discretization error.

A priori estimate: bounds for error norms available **before** computing approximate solutions.



A posteriori estimate: bounds for error norms based on an approximate solution **already computed**.

Results of Sect. 5.1 provide handle on a priori estimate for Galerkin discretization error:

Optimality (5.1.11) of Galerkin solution ➤ a priori error estimates

Thm. 5.1.10 ➤

Estimate energy norm of Galerkin discretization error $u - u_N$
by bounding best approximation error
for exact solution u in finite element space:

$$\underbrace{\|u - u_N\|_a}_{\text{(norm of) discretization error}} \leq \underbrace{\inf_{v_N \in V_{0,N}} \|u - v_N\|_a}_{\text{best approximation error}} , \quad (5.1.11)$$

How to estimate **best approximation error**

$$\inf_{v_N \in V_{0,N}} \|u - v_N\|_V ?$$

➤ Well, given solution u seek candidate function $w_N \in V_{0,N}$ with

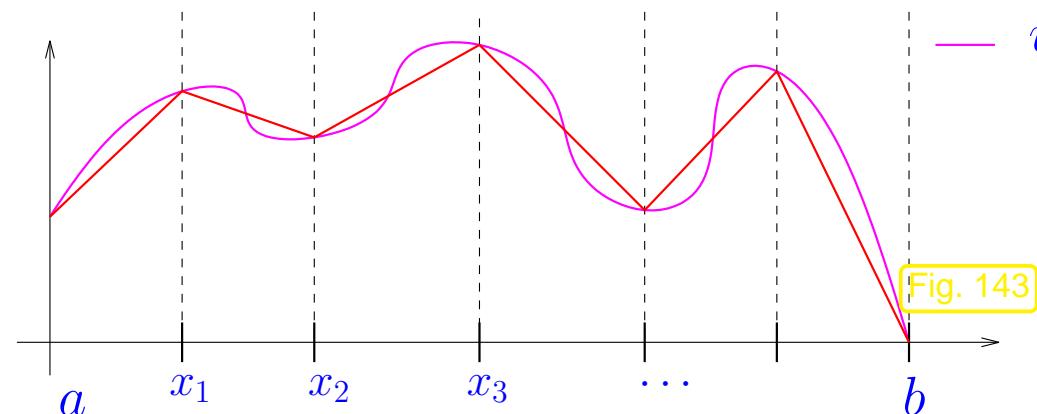
$$\|u - w_N\|_V \approx \inf_{v_N \in V_N} \|u - v_N\|_V .$$

Natural choice: w_N by interpolation/averaging of (*unknown, but existing*) u

5.3.1 Estimates for linear interpolation in 1D

Computational domain (\rightarrow Sect. 1.4): interval $\Omega = [a, b]$

Given: mesh of Ω (\rightarrow Sect. 1.5.1.2): $\mathcal{M} := \{]x_{j-1}, x_j[: j = 1, \dots, M\}, M \in \mathbb{N}$



Piecewise linear interpolant of $u \in C^0([a, b])$

$$I_1 u \in \mathcal{S}_1^0(\mathcal{M}) , \quad (5.3.1)$$

$$(I_1 u)(x_j) = u(x_j) , \quad j = 0, \dots, M . \quad (5.3.2)$$

Fig. 143

➤ [14, Sect. 9.2.1]

Goal:

Bound suitable norm (\rightarrow Sect. 1.6.1) of **interpolation error** $u - \mathbf{I}_1 u$
in terms of geometric quantities (*) characterizing \mathcal{M} .

(*): A typical such quantity is the **mesh width** $h_{\mathcal{M}} := \max_j |x_j - x_{j-1}|$

Now we investigate different norms of the interpolation error.

- $\|u - \mathbf{I}_1 u\|_{L^\infty([a,b])}$, see [14, Sect. 9.2.3]: from [14, Thm. 8.4.1] for $n = 1$: for $u \in C^2([a, b])$

$$\max_{x_{j-1} \leq x \leq x_j} u(t) - (\mathbf{I}u)(x) = \frac{1}{4}u''(\xi_t)(x_j - x_{j-1})^2, \quad \text{for some } \xi_t \in]x_{j-1}, x_j[, \quad (5.3.3)$$

with **local linear interpolant** $(\mathbf{I}u)(x) = \frac{x - x_{j-1}}{x_j - x_{j-1}}u(x_j) - \frac{x_j - x}{x_j - x_{j-1}}u(x_{j-1})$. (5.3.4)

(5.3.3) ► interpolation error estimate in $L^\infty([a, b])$

$$\boxed{\|u - \mathbf{I}_1 u\|_{L^\infty([a,b])} \leq \frac{1}{4}h_{\mathcal{M}}^2 \|u''\|_{L^\infty([a,b])}} . \quad (5.3.5)$$

This is obtained by simply taking the maximum over all *local* norms of the interpolation error.

Recall: supremum norm (maximum norm) from Def. 1.6.4

Now, we also want to study other norms of the interpolation error:

- $\|u - I_1 u\|_{L^2([a,b])}$:

Now all mesh cells contribute to this norm:

$$\|u - I_1 u\|_{L^2([a,b])}^2 = \sum_{j=1}^M \|u - I_1 u\|_{L^2([x_{j-1}, x_j])}^2 = \sum_{j=1}^M \int_{x_{j-1}}^{x_j} |(u - I_1 u)(x)|^2 dx , \quad I_1 u \text{ from (5.3.4)} . \quad (5.3.6)$$

➤ Idea:

localization

(Estimate error on individual mesh cells and sum local bounds)

By integrating by parts (1.3.20) twice, for $u \in C^2([x_{j-1}, x_j])$, $x \in [x_{j-1}, x_j]$,

$$\begin{aligned}
& \int_{x_{j-1}}^x \frac{(x_j - x)(\xi - x_{j-1})}{x_j - x_{j-1}} u''(\xi) d\xi + \int_x^{x_j} \frac{(x - x_{j-1})(x_j - \xi)}{x_j - x_{j-1}} u''(\xi) d\xi \\
&= \underbrace{\frac{x_j - x}{x_j - x_{j-1}} u(x_{j-1}) + \frac{x - x_{j-1}}{x_j - x_{j-1}} u(x_j)}_{=|u(x)|} - u(x) . \quad (5.3.7)
\end{aligned}$$

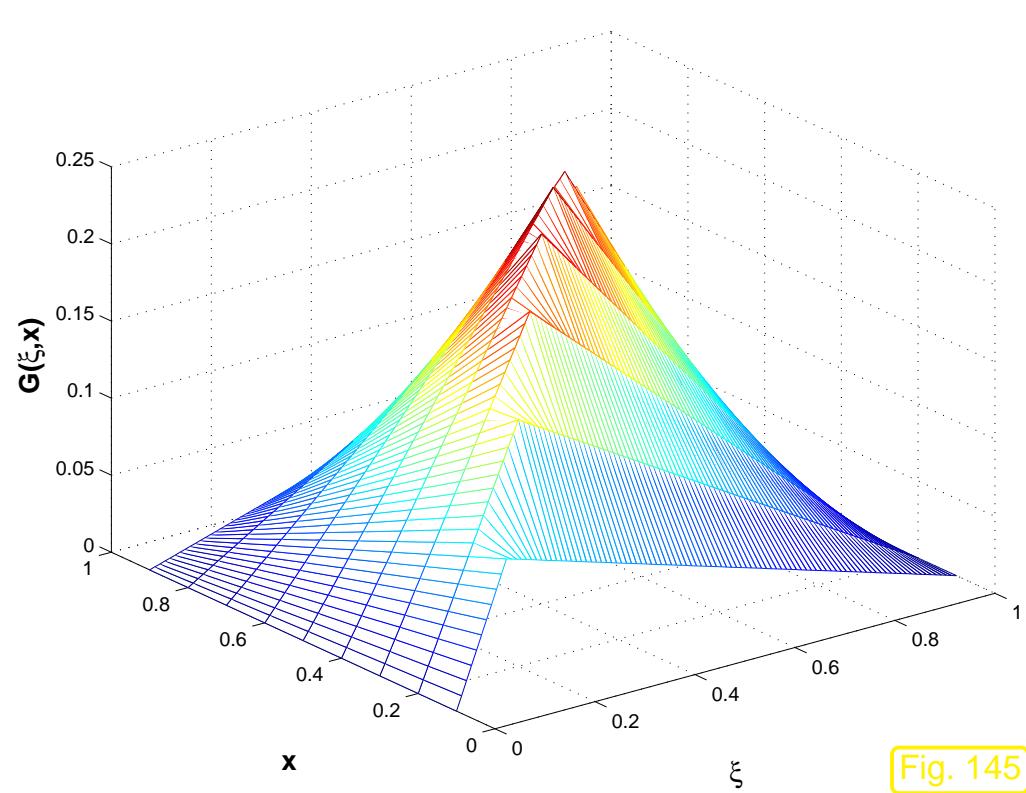
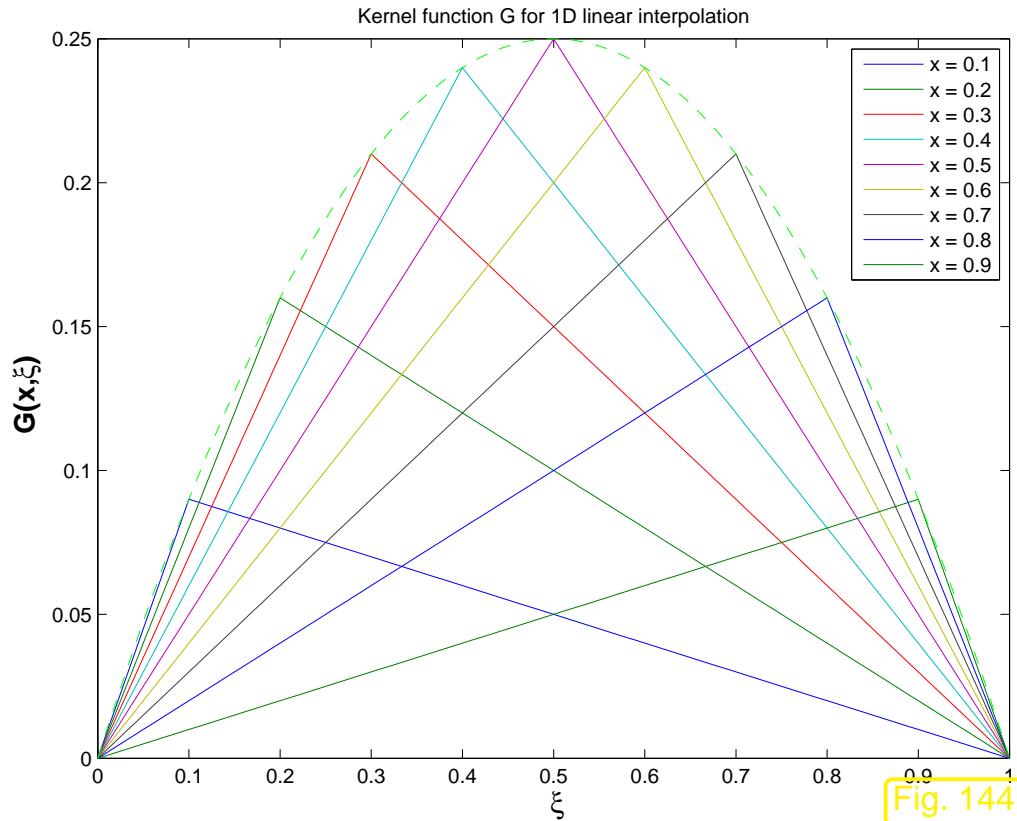
This is a **representation formula** for the local interpolation error $|u - u|$ of the form

$$(|u - u|)(x) = \int_{x_{j-1}}^{x_j} G(x, \xi) u''(\xi) d\xi .$$

with $G(x, \xi) = \begin{cases} \frac{(x_j - x)(\xi - x_{j-1})}{x_j - x_{j-1}} & \text{for } x_{j-1} \leq \xi < x , \\ \frac{(x - x_{j-1})(x_j - \xi)}{x_j - x_{j-1}} & \text{for } x \leq \xi \leq x_j . \end{cases}$, which satisfies

$$|G(x, \xi)| \leq |x_j - x_{j-1}| \Rightarrow \int_{x_{j-1}}^{x_j} G(x, \xi)^2 d\xi \leq |x_j - x_{j-1}|^3 .$$

Kernel functions G for 1D linear interpolation for $x_{j-1} = 0$, $x_j = 1$.



$$\begin{aligned}
 \blacktriangleright \int_{x_{j-1}}^{x_j} |u(x) - u(x)|^2 dx &= \int_{x_{j-1}}^{x_j} \left| \int_{x_{j-1}}^{x_j} G(x, \xi) u''(\xi) d\xi \right|^2 dx \quad (5.3.8) \\
 &\stackrel{(2.2.15)}{\leq} \int_{x_{j-1}}^{x_j} \left\{ \int_{x_{j-1}}^{x_j} G(x, \xi)^2 d\xi \cdot \int_{x_{j-1}}^{x_j} |u''(\xi)|^2 d\xi \right\} dx ,
 \end{aligned}$$

$$(5.3.8) \Rightarrow \|u - |u|\|_{L^2([x_{j-1}, x_j])}^2 = \int_{x_{j-1}}^{x_j} |u(x) - |u(x)||^2 dx \leq |x_j - x_{j-1}|^4 \int_{x_{j-1}}^{x_j} |u''(\xi)|^2 d\xi . \quad (5.3.9)$$

Apply this estimate on $[x_{j-1}, x_j]$, sum over all cells of the mesh \mathcal{M} and take square root.

$$(5.3.9) \Rightarrow \|u - |u|\|_{L^2([a,b])} \leq h_{\mathcal{M}}^2 \|u''\|_{L^2([a,b])} . \quad (5.3.10)$$

- $|u - |u||_{H^1([a,b])}$:

Differentiate representation formula (5.3.7): for $x_{j-1} < x < x_j$

$$\frac{d}{dx}(|u - u|(x)) = \int_{x_{j-1}}^{x_j} -\frac{\xi - x_{j-1}}{x_j - x_{j-1}} u''(\xi) d\xi + \int_{x_{j-1}}^{x_j} \frac{x_j - \xi}{x_j - x_{j-1}} u''(\xi) d\xi . \quad 5.3$$

►
$$\int_{x_{j-1}}^{x_j} \left| \frac{d}{dx} (\mathbf{I}_1 u - u)(x) \right|^2 dx = \int_{x_{j-1}}^{x_j} \left| \frac{\partial G}{\partial x}(x, \xi) u''(\xi) d\xi \right|^2 dx$$

$$\leq \int_{x_{j-1}}^{x_j} \left\{ \int_{x_{j-1}}^{x_j} \underbrace{\left| \frac{\partial G}{\partial x}(x, \xi) \right|^2}_{\leq 1} d\xi \cdot \int_{x_{j-1}}^{x_j} |u''(\xi)|^2 d\xi \right\} .$$

►
$$|\mathbf{I}_1 u - u|_{H^1([x_{j-1}, x_j])}^2 \leq (x_j - x_{j-1})^2 \int_{x_{j-1}}^{x_j} |f''(\xi)|^2 d\xi . \quad (5.3.11)$$

As above, apply this estimate on $[x_{j-1}, x_j]$, sum over all cells of the mesh \mathcal{M} and take square root.

$$(5.3.11) \Rightarrow |\mathbf{I}_1 u - u|_{H^1([a,b])} \leq h_{\mathcal{M}} \|u''\|_{L^2([a,b])} . \quad (5.3.12)$$

What we learn from this example:

1. We have to rely on **smoothness** of the interpoland u to obtain bounds for norms of the interpolation error.
2. The bounds involve norms of derivatives of the interpoland.
3. For smooth u we find **algebraic convergence** (\rightarrow Def. 1.6.19) of norms of the interpolation error *in terms of mesh width $h_{\mathcal{M}} \rightarrow 0$* .

5.3.2 Error estimates for linear interpolation in 2D

Given:

- polygonal domain $\Omega \subset \mathbb{R}^2$
- triangular mesh \mathcal{M} of Ω (\rightarrow Def. 3.3.1)

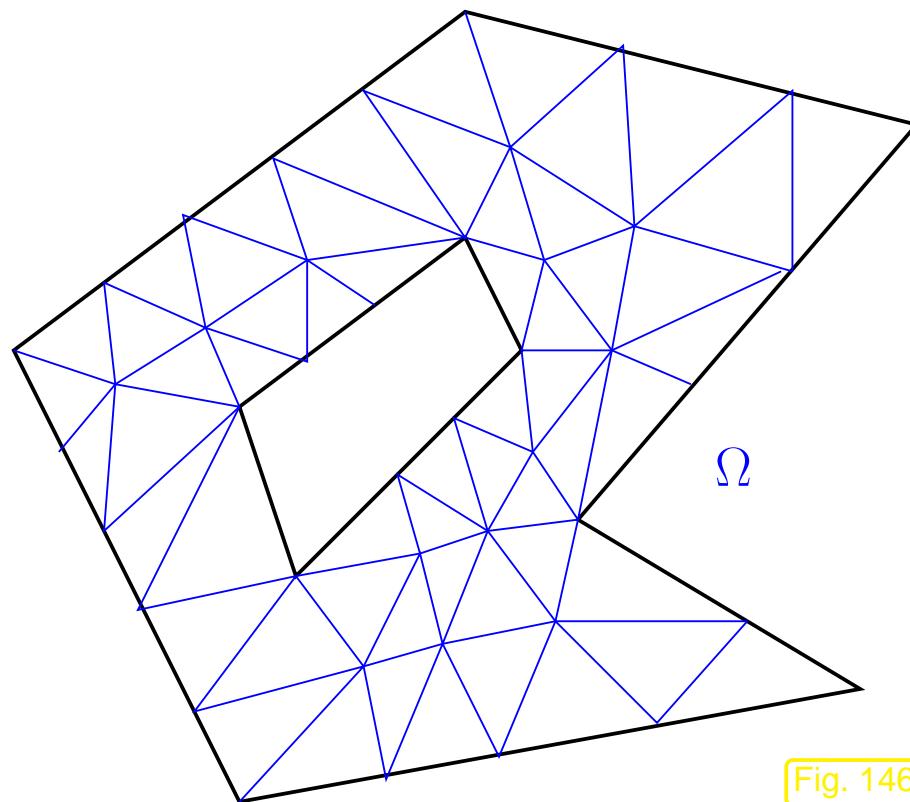


Fig. 146

Sect 5.3.1 introduced piecewise linear interpolation on a mesh/grid in 1D. The next definition gives the natural 2D counterpart on a triangular mesh, which is closely related to the piecewise linear reconstruction (interpolation) operator from (4.2.6), see Figs. 129, 130.

Definition 5.3.13 (Linear interpolation in 2D).

The linear interpolation operator $\mathbf{I}_1 : C^0(\bar{\Omega}) \mapsto \mathcal{S}_1^0(\mathcal{M})$ is defined by

$$\mathbf{I}_1 u \in \mathcal{S}_1^0(\mathcal{M}) \quad , \quad \mathbf{I}_1 u(\mathbf{p}) = u(\mathbf{p}) \quad \forall \mathbf{p} \in \mathcal{V}(\mathcal{M}) .$$

Recalling the definition of the nodal basis $\mathfrak{B} = \{b_N^{\mathbf{p}} : \mathbf{p} \in \mathcal{V}(\mathcal{M})\}$ of $\mathcal{S}_1^0(\mathcal{M})$ from (3.2.2), where $b_N^{\mathbf{p}}$ is the “tent function” associated with node \mathbf{p} , an equivalent definition is, cf. (3.5.44),

$$\mathbf{I}_1 u = \sum_{\mathbf{p} \in \mathcal{V}(\mathcal{M})} u(\mathbf{p}) b_N^{\mathbf{p}} , \quad u \in C^0(\bar{\Omega}) . \quad (5.3.14)$$

Task:  For “sufficiently smooth” $u : \Omega \mapsto \mathbb{R}$ ($\leftrightarrow u \in C^\infty(\bar{\Omega})$ to begin with) estimate

interpolation error norm $\|u - \mathbf{I}_1 u\|_{H^1(\Omega)}$.

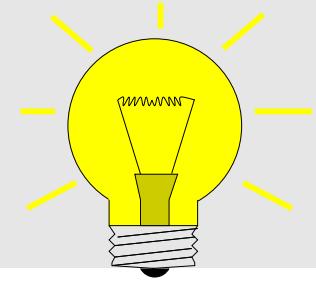
Idea:

Localization

\mathbf{I}_1 local \geq first, estimate $\|u - \mathbf{I}_1 u\|_{H^1(K)}^2, K \in \mathcal{M}$,
then, global estimate via summation as in Sect. 5.3.1.

5.3

\geq Focus on single triangle $K \in \mathcal{M}$

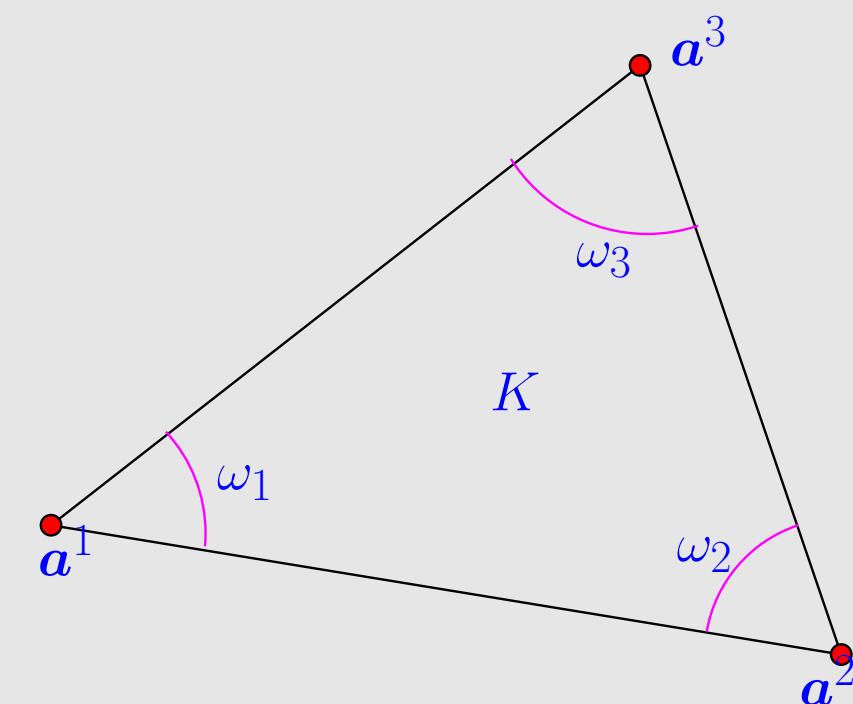


Crucial for localization to work: linear interpolation operator $\mathbf{I}_1 : C^0(\bar{\Omega}) \mapsto \mathcal{S}_1^0(\mathcal{M})$ can be defined **purely locally** by

$$\mathbf{I}_1 u|_K = u(\mathbf{a}^1)\lambda_1 + u(\mathbf{a}^2)\lambda_2 + u(\mathbf{a}^3)\lambda_3 , \quad (5.3.15)$$

for each triangle $K \in \mathcal{M}$ with vertices $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$ ($\lambda_k \hat{=} \text{barycentric coordinate functions} = \text{local shape functions for } \mathcal{S}_1^0(\mathcal{M})$, see Fig. ??).

Next step, cf. (5.3.7): **representation formula** for local interpolation error.



$$u \in C^2(\bar{K}): \text{ by mean value formula } \forall \mathbf{x} \in K,$$

$$u(\mathbf{a}^j) = u(\mathbf{x}) + \mathbf{grad} u(\mathbf{x}) \cdot (\mathbf{a}^j - \mathbf{x}) +$$

$$\int_0^1 (\mathbf{a}^j - \mathbf{x})^\top D^2 u(\mathbf{x} + \xi(\mathbf{a}^j - \mathbf{x})) (\mathbf{a}^j - \mathbf{x})(1 - \xi) d\xi , \quad (5.3.16)$$

$$D^2 u(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 u}{\partial x_1^2}(\mathbf{x}) & \frac{\partial^2 u}{\partial x_1 \partial x_2}(\mathbf{x}) \\ \frac{\partial^2 u}{\partial x_1 \partial x_2}(\mathbf{x}) & \frac{\partial^2 u}{\partial x_2^2}(\mathbf{x}) \end{pmatrix} \hat{=} \text{Hessian.}$$

The formula (5.3.16) is easily verified by applying integration by parts

$$f(b) - f(a) = [\xi f'(\xi)]_a^b - \int_a^b \xi f''(\xi) d\xi = f'(a)(b-a) + \int_a^b (b-\xi) f''(\xi) d\xi .$$

to the function $\phi(t) = u(t\mathbf{a}^j + (1-t)\mathbf{x})$ with $a = 0, b = 1$.

Next, use (5.3.16) to replace $u(\mathbf{a}^j)$ in the formula (5.3.15) for local linear interpolation. Also use the identities for the barycentric coordinate functions

$$\sum_{j=1}^3 \lambda_j(\mathbf{x}) = 1 \quad , \quad \mathbf{x} = \sum_{j=1}^3 \mathbf{a}^j \lambda_j(\mathbf{x}) . \quad (5.3.17)$$

$$\mathsf{I}_1 u(\mathbf{x}) = \sum_{j=1}^3 u(\mathbf{a}^j) \lambda_j(\mathbf{x}) = u(\mathbf{x}) \cdot \underbrace{\sum_{j=1}^3 \lambda_j(\mathbf{x})}_{=1} + \mathbf{grad} u(\mathbf{x}) \cdot \underbrace{\sum_{j=1}^3 (\mathbf{a}^j - \mathbf{x}) \lambda_j(\mathbf{x})}_{=0} + R(\mathbf{x}) ,$$

with $R(\mathbf{x}) := \sum_{j=1}^3 \left(\int_0^1 (\mathbf{a}^j - \mathbf{x})^\top D^2 u(\mathbf{x} + \xi(\mathbf{a}^j - \mathbf{x})) (\mathbf{a}^j - \mathbf{x})(1-\xi) d\xi \right) \lambda_j(\mathbf{x}) . \quad (5.3.18)$

Again, as in the case of (5.3.7) for 1D linear interpolation we have arrived at an **integral representation formula** for the local interpolation error:

$$(u - I_1 u)(\mathbf{x}) = \sum_{j=1}^3 \left(\int_0^1 (\mathbf{a}^j - \mathbf{x})^T D^2 u(\mathbf{x} + \xi(\mathbf{a}^j - \mathbf{x})) (\mathbf{a}^j - \mathbf{x})(1 - \xi) d\xi \right) \lambda_j(\mathbf{x}) . \quad (5.3.19)$$

Together with the triangle inequality, the trivial bound $|\lambda_j| \leq 1$ yields

$$\|u - I_1 u\|_{L^2(K)} \leq \sum_{j=1}^3 \left(\int_K \left(\int_0^1 (\mathbf{a}^j - \mathbf{x})^T D^2 u(\mathbf{x} + \xi(\mathbf{a}^j - \mathbf{x})) (\mathbf{a}^j - \mathbf{x})(1 - \xi) d\xi \right)^2 d\mathbf{x} \right)^{\frac{1}{2}} .$$

To estimate an expression of the form

$$\int_K \left(\int_0^1 (\mathbf{a}^j - \mathbf{x})^T D^2 u(\mathbf{x} + \xi(\mathbf{a}^j - \mathbf{x})) (\mathbf{a}^j - \mathbf{x})(1 - \xi) d\xi \right)^2 d\mathbf{x} , \quad (5.3.20)$$

we may assume, without loss of generality, that $\mathbf{a}^j = 0$.

➤ Task: estimate terms (where 0 is a vertex of K!)

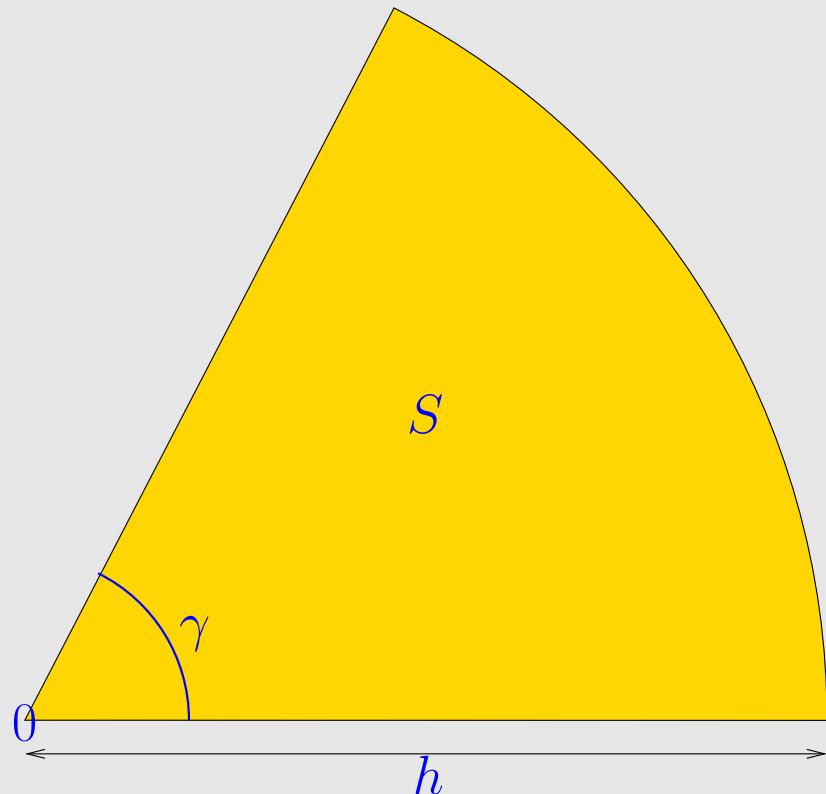
$$\int_K \left(\int_0^1 \mathbf{x}^\top D^2 u((1 - \xi)\mathbf{x}) \mathbf{x} (1 - \xi) d\xi \right)^2 d\mathbf{x} = \int_K \left(\int_0^1 \mathbf{x}^\top D^2 u(\xi \mathbf{x}) \mathbf{x} \xi d\xi \right)^2 d\mathbf{x} . \quad 5.3$$

Denote $\gamma \hat{=} \text{ angle of } K \text{ at vertex } 0,$
 $h \hat{=} \text{ length of longest edge of } K.$

► K is contained in the sector
 $S := \{\mathbf{x} = \begin{pmatrix} r \cos \varphi \\ r \sin \varphi \end{pmatrix} : 0 \leq r < h, 0 \leq \varphi \leq \gamma\}$

Lemma 5.3.21. For any $\psi \in L^2(S)$

$$\int_S \left(\int_0^1 |\mathbf{y}|^2 \psi(\tau \mathbf{y}) \tau d\tau \right)^2 d\mathbf{y} \leq \frac{h^4}{8} \|\psi\|_{L^2(S)}^2.$$



Using polar coordinates (r, φ) , $\hat{\mathbf{s}}_\varphi = \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix}$, see [19, Bsp. 8.5.3], and Cauchy-Schwarz inequality (2.2.15):

$$\int_S \left(\int_0^1 |\mathbf{y}|^2 \psi(\tau \mathbf{y}) \tau d\tau \right)^2 d\mathbf{y} = \int_0^\gamma \int_0^h \left(\int_0^1 r^2 \psi(\tau r \hat{\mathbf{s}}_\varphi) \tau d\tau \right)^2 r dr d\varphi$$

$$\begin{aligned}
&= \int_0^\gamma \int_0^h \left(\int_0^r \psi(\sigma \hat{\mathbf{s}}_\varphi) \sigma d\sigma \right)^2 r dr d\varphi \leq \int_0^\gamma \int_0^h \int_0^r \psi^2(\sigma \hat{\mathbf{s}}_\varphi) \sigma d\sigma \cdot \int_0^r \sigma d\sigma r dr d\varphi \\
&\leq \frac{1}{2} \int_0^\gamma \int_0^h \psi^2(\sigma \hat{\mathbf{s}}_\varphi) \sigma d\sigma d\varphi \cdot \int_0^h r^3 dr .
\end{aligned}$$

Use $|\mathbf{z}^\top \mathbf{A} \mathbf{y}| \leq \|\mathbf{A}\|_F |\mathbf{z}| |\mathbf{y}|$, $\mathbf{A} \in \mathbb{R}^{n,n}$, $\mathbf{z}, \mathbf{y} \in \mathbb{R}^n$, and then apply Lemma 5.3.21 with $\mathbf{y} := \mathbf{x} - \mathbf{a}^j$, $\tau = 1 - \xi$

► $\|u - \mathbf{l}_1 u\|_{L^2(K)}^2 \leq \frac{3}{8} h_K^4 \left\| \left\| D^2 u \right\|_F \right\|_{L^2(K)}^2 , \quad (5.3.22)$

with **Frobenius matrix norm** $\left\| D^2 u(\mathbf{x}) \right\|_F^2 := \sum_{i,j=1}^2 \left| \frac{\partial^2 u}{\partial x_i \partial x_j}(\mathbf{x}) \right|^2$,

size of triangle $h_K := \text{diam } K := \max\{|\mathbf{p} - \mathbf{q}| : \mathbf{p}, \mathbf{q} \in K\}$

Estimate for gradient: from (5.3.16) we infer the local integral representation formula, which can also be obtained by taking the gradient of (5.3.19).

$$\mathbf{grad} \mathbf{l}_1 u(\mathbf{x}) = u(\mathbf{x}) \underbrace{\sum_{j=1}^3 \mathbf{grad} \lambda_j(\mathbf{x})}_{0} + \underbrace{\sum_{j=1}^3 (\mathbf{a}^j - \mathbf{x})^\top \mathbf{grad} \lambda_j(\mathbf{x}) \cdot \mathbf{grad} u(\mathbf{x}) + G(\mathbf{x})}_{\mathbf{I}} , \quad 5.3$$

with $G(\mathbf{x}) := \sum_{j=1}^3 \underbrace{\left(\int_0^1 (\mathbf{a}^j - \mathbf{x})^\top D^2 u(\mathbf{x} + \xi(\mathbf{a}^j - \mathbf{x})) (\mathbf{a}^j - \mathbf{x})(1 - \xi) d\xi \right)}_{cf. (5.3.20)} \mathbf{\text{grad}} \lambda_j(\mathbf{x})$.

Note that $\mathbf{\text{grad}} \sum_{j=1}^3 \lambda_j(\mathbf{x}) = \mathbf{\text{grad}} 1 = 0$ and

$$\sum_{j=1}^3 \mathbf{\text{grad}} \lambda_j(\mathbf{x}) (\mathbf{a}^j - \mathbf{x})^\top = \sum_{j=1}^3 \mathbf{\text{grad}} \lambda_j(\mathbf{x}) (\mathbf{a}^j)^\top = \mathbf{\text{grad}} \left(\sum_{j=1}^3 \lambda_j(\mathbf{x}) \mathbf{a}^j \right) = \mathbf{\text{grad}} \mathbf{x} = \mathbf{I}.$$

$$(3.5.22) \rightarrow \boxed{|\mathbf{\text{grad}} \lambda_j(\mathbf{x})| \leq \frac{h_K}{2|K|}, \quad \mathbf{x} \in K}. \quad (5.3.23)$$

► $\|\mathbf{\text{grad}}(u - I_1 u)\|_{L^2(K)}^2 \leq \frac{h_K^2}{4|K|^2} \|R\|_{L^2(K)}^2 \stackrel{(5.3.22)}{\leq} \frac{3}{8} \frac{h_K^6}{4|K|^2} \left\| \left\| D^2 u \right\|_F \right\|_{L^2(K)}^2. \quad (5.3.24)$

Summary of *local* interpolation error estimates for linear interpolation according to Def. 5.3.13:

Lemma 5.3.25 (Local interpolation error estimates for 2D linear interpolation).

For any triangle K and $u \in C^2(\overline{K})$ the following holds

$$\|u - I_1 u\|_{L^2(K)}^2 \leq \frac{3}{8} h_K^4 \left\| \|D^2 u\|_F \right\|_{L^2(K)}^2, \quad (5.3.22)$$

$$\|\mathbf{grad}(u - I_1 u)\|_{L^2(K)}^2 \leq \frac{3}{24} \frac{h_K^6}{|K|^2} \left\| \|D^2 u\|_F \right\|_{L^2(K)}^2. \quad (5.3.24)$$

New aspect compared to Sect. 5.3.1: *shape* of K enters error bounds of Lemma 5.3.25.

We aim to extract this shape dependence from the bounds.

Definition 5.3.26 (Shape regularity measures).

For a simplex $K \in \mathbb{R}^d$ we define its *shape regularity measure* as the ratio

$$\rho_K := h_K^d : |K| ,$$

and the shape regularity measure of a simplicial mesh $\mathcal{M} = \{K\}$

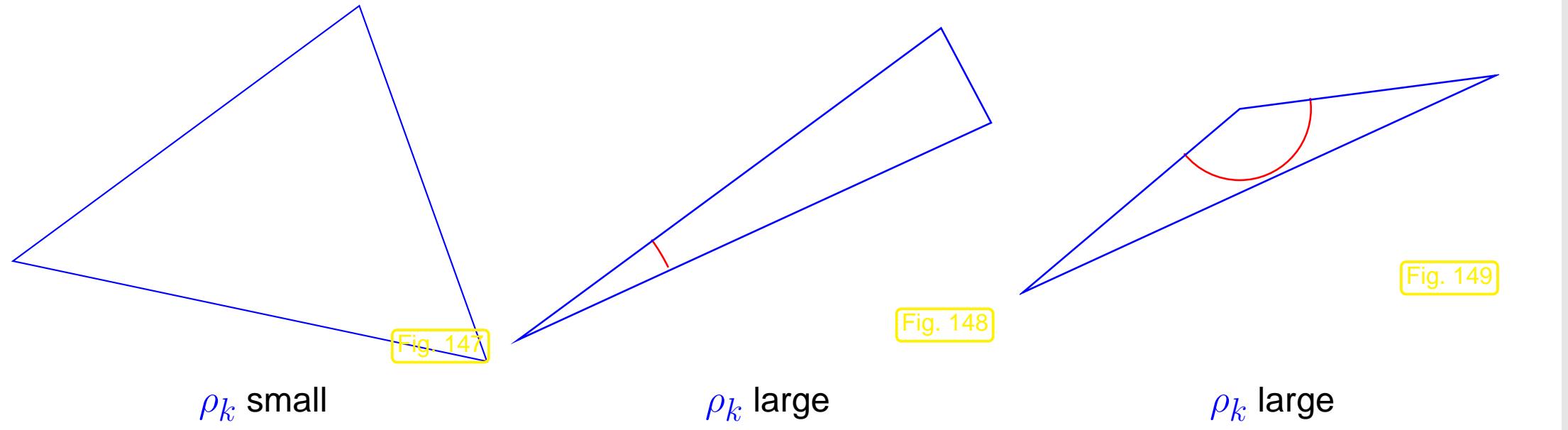
$$\rho_{\mathcal{M}} := \max_{K \in \mathcal{M}} \rho_K .$$

Important: shape regularity measure ρ_K is an invariant of a similarity class of triangles.

(= if a triangle is transformed by scaling, rotation, and translation, the shape regularity measure does not change)

➤ Sloppily speaking, ρ_K depends only on the shape, not the size of K

For triangle K : ρ_K large $\Leftrightarrow K$ “distorted” $\Leftrightarrow K$ has small angles



The shape regularity measure $\rho_{\mathcal{M}}$ is often used to gauge the *quality* of meshes produced by mesh generators.

Final step: we add up the local estimates from Lemma 5.3.25 over all triangles of the mesh and take the square root.

Theorem 5.3.27 (Error estimate for piecewise linear interpolation).

For any $u \in C^2(\bar{\Omega})$

$$\|u - I_1 u\|_{L^2(\Omega)} \leq \sqrt{\frac{3}{8}} h_{\mathcal{M}}^2 \left\| \|D^2 u\|_F \right\|_{L^2(\Omega)},$$

$$\|\mathbf{grad}(u - I_1 u)\|_{L^2(\Omega)} \leq \sqrt{\frac{3}{24}} \rho_{\mathcal{M}} h_{\mathcal{M}} \left\| \|D^2 u\|_F \right\|_{L^2(\Omega)}.$$

Remark 5.3.28 (Local interpolation onto higher degree Lagrangian finite element spaces).

\mathcal{M} : triangular/tetrahedral/quadrilateral/hybrid mesh of domain Ω (\rightarrow Sect. 3.3.1)

Recall (\rightarrow Sect. 3.4): nodal basis functions of p -th degree Lagrangian finite element space $\mathcal{S}_p^0(\mathcal{M})$ defined via **interpolation nodes**, cf. (3.4.3).

Set of interpolation nodes: $\mathcal{N} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\} \subset \bar{\Omega}$, $N = \dim \mathcal{S}_p^0(\mathcal{M})$.

➤ General **nodal Lagrangian interpolation operator**

$$I_p : \begin{cases} C^0(\bar{\Omega}) \mapsto \mathcal{S}_p^0(\mathcal{M}) \\ u \mapsto I_p(u) := \sum_{l=1}^N u(\mathbf{p}_l) b_N^l \end{cases},$$

where b_N^l are the nodal basis functions.

$$(3.4.3) \Rightarrow I_p(u)(\mathbf{p}_l) = u(\mathbf{p}_l), \quad l = 1, \dots, N \quad (\text{Interpolation!}) .$$

By virtue of the location of the interpolation nodes, see Ex. 3.4.2, Ex. 3.4.5, and Fig. 101, the nodal interpolation operators are purely local:

$$\forall K \in \mathcal{M}: \quad |_p u|_K = \sum_{i=1}^Q u(\mathbf{q}_i^K) b_i^K, \quad (5.3.29)$$

$\mathbf{q}_i^K \quad i = 1, \dots, Q$	$K \in \mathcal{M}$	3.4.2	3.4.5	101
$b_i^K \quad i = 1, \dots, Q$	local shape functions	$b_i^K(\mathbf{q}_j^K) = \delta_{ij}$		

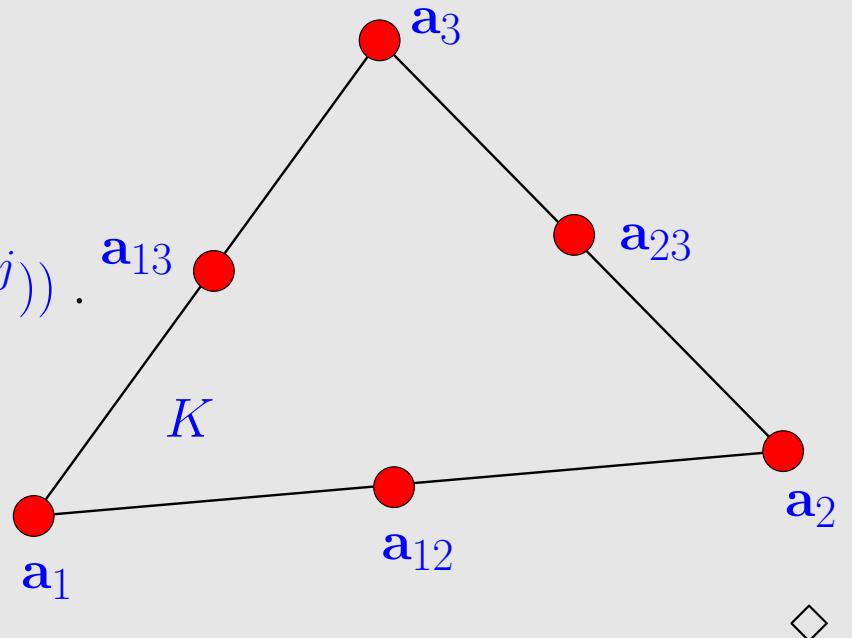
Example 5.3.30 (Piecewise quadratic interpolation). → Ex. 3.4.2

triangle $K = \text{convex}\{\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3\}$, $p = 2$

\Rightarrow local quadratic interpolation:

$$l_2 u|_K = - \sum_{i=1}^3 \lambda_i (1 - 2\lambda_i) u(\mathbf{a}^i) + \sum_{1 \leq i < j \leq 3} 4\lambda_i \lambda_j u\left(\frac{1}{2}(\mathbf{a}^i + \mathbf{a}^j)\right)$$

local shape functions, see (3.4.4)



◇

Remark 5.3.31 (Energy norm and $H^1(\Omega)$ -norm).

Objection! Well, Cea's lemma Thm. 5.1.10 refers to the energy norm, but Thm. 5.3.27 provides estimates in $H^1(\Omega)$ -norm only!

- ☞ For uniformly positive definite (\rightarrow Def. 2.1.9) and bounded coefficient tensor $\alpha : \Omega \mapsto \mathbb{R}^{d,d}$, cf. (2.1.8),

$$\exists 0 < \alpha^- < \alpha^+ : \quad \alpha^- \|z\|^2 \leq z^T \alpha(x) z \leq \alpha^+ \|z\|^2 \quad \forall z \in \mathbb{R}^d, x \in \Omega ,$$

and the energy norm (\rightarrow Def. 2.1.24) induced by

$$a(u, v) := \int_{\Omega} (\alpha(x) \operatorname{grad} u) \cdot \operatorname{grad} v \, dx , \quad u, v \in H_0^1(\Omega) , \quad (5.1.6)$$

we immediately find the **equivalence** (= two-sided uniform estimate)

$$\sqrt{\alpha^-} |v|_{H^1(\Omega)} \leq \|v\|_a \leq \sqrt{\alpha^+} |v|_{H^1(\Omega)} . \quad (5.3.32)$$

Thus, interpolation error estimates in $|\cdot|_{H^1(\Omega)}$ immediately translate into estimates in terms of the energy norm.



5.3.3 The Sobolev scales

Bounds in Thm. 5.3.27 involve $\| \|D^2u\|_F\|_{L^2(\Omega)}$



measures smoothness of u

- Norms of this type are a tool to measure the **smoothness** of functions (that usually are solutions of elliptic BVP):

Definition 5.3.33 (Higher order Sobolev spaces/norms).

The ***m-th order Sobolev norm***, $m \in \mathbb{N}_0$, for $u : \Omega \subset \mathbb{R}^d \mapsto \mathbb{R}$ (sufficiently smooth) is defined by

$$\|u\|_{H^m(\Omega)}^2 := \sum_{k=0}^m \sum_{\alpha \in \mathbb{N}^d, |\alpha|=k} \int_{\Omega} |D^\alpha u|^2 dx , \quad \text{where} \quad D^\alpha u := \frac{\partial^{|\alpha|} u}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}} .$$

Sobolev space $H^m(\Omega) := \{v : \Omega \mapsto \mathbb{R} : \|v\|_{H^m(\Omega)} < \infty\} .$

Recall: multiindex notation (3.3.4), (3.3.5)

Gripe (\rightarrow Sect. 2.2):

Don't bother me with these Sobolev spaces !

Response: Well, these concepts are pervasive in the numerical analysis literature and you have to be familiar with them.

Reassuring:

Again, it is only the norms $\|u\|_{H^m(\Omega)}$ that matter for us !

Now, we have come across an additional purpose of Sobolev spaces and their norms:

provide framework for
variational formulation of
elliptic BVP
 $(\rightarrow$ Sect. 2.2)

Sobolev
spaces

provide norms $\|\cdot\|_{H^m(\Omega)}$ that
measure smoothness of
functions

Sobolev scale:

$$\dots \subset H^3(\Omega) \subset H^2(\Omega) \subset H^1(\Omega) \subset L^2(\Omega)$$

Observation: bounds in Thm. 5.3.27 = “principal parts” of Sobolev norms, that is, the parts containing the highest partial derivatives.

Definition 5.3.34 (Higher order Sobolev semi-norms).

The *m-th order Sobolev semi-norm*, $m \in \mathbb{N}$, for sufficiently smooth $u : \Omega \mapsto \mathbb{R}$ is defined by

$$|u|_{H^m(\Omega)}^2 := \sum_{\alpha \in \mathbb{N}^d, |\alpha|=m} \int_{\Omega} |D^\alpha u|^2 dx .$$

Elementary observation: $|p|_{H^m(\Omega)} = 0 \Leftrightarrow p \in \mathcal{P}_{m-1}(\mathbb{R}^d)$



By density arguments we can rewrite the interpolation error estimates of Thm. 5.3.27 in terms of Sobolev semi-norms:

Corollary 5.3.35 (Error estimate for piecewise linear interpolation in 2D).

Under the assumptions/with notations of Thm. 5.3.27

$$\begin{aligned}\|u - I_1 u\|_{L^2(\Omega)} &\leq \sqrt{\frac{3}{8}} h_M^2 |u|_{H^2(\Omega)}, & \forall u \in H^2(\Omega). \\ |u - I_1 u|_{H^1(\Omega)} &\leq \sqrt{\frac{3}{24}} \rho_M h_M |u|_{H^2(\Omega)},\end{aligned}$$

Remark 5.3.36 (Continuity of interpolation operators).

Apply \triangle -inequality to estimates of Cor. 5.3.35:

$$\|I_1 u\|_{L^2(\Omega)} \leq \|u\|_{L^2(\Omega)} + \sqrt{\frac{3}{8}} h_M^2 |u|_{H^2(\Omega)} \leq 2\|u\|_{H^2(\Omega)}, \quad (5.3.37)$$

if lengths are scaled such that $h_M \leq 1$. Estimate (5.3.37) means that $I_1 : H^2(\Omega) \mapsto L^2(\Omega)$ is a **continuous linear** mapping.

The same conclusion could have been drawn from the following fundamental result:

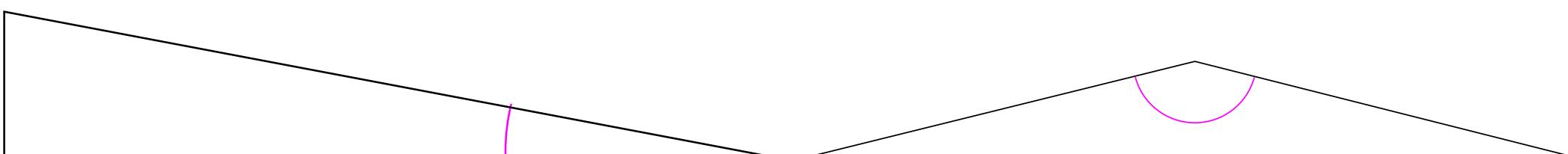
Theorem 5.3.38 (Sobolev embedding theorem).

$$m > \frac{d}{2} \Rightarrow H^m(\Omega) \subset C^0(\bar{\Omega}) \quad \wedge \quad \exists C = C(\Omega) > 0: \|u\|_{\infty} \leq C \|u\|_{H^m(\Omega)} \quad \forall u \in H^m(\Omega).$$

On the other hand $\mathbf{I}_1 : H^1(\Omega) \mapsto L^2(\Omega)$ is **not** continuous, as we learn from Rem. 2.3.16. \triangle

5.3.4 Anisotropic interpolation error estimates

Triangular cells with “bad shape regularity” (ρ_K “large”): very small/large angles:



The estimates of Lemma 5.3.25 might suggest that we face huge local interpolation errors, once ρ_K becomes large.

Issue: are the estimates of Lemma 5.3.25 *sharp*?

We will try to find this out experimentally by computing the best possible constants in the estimates

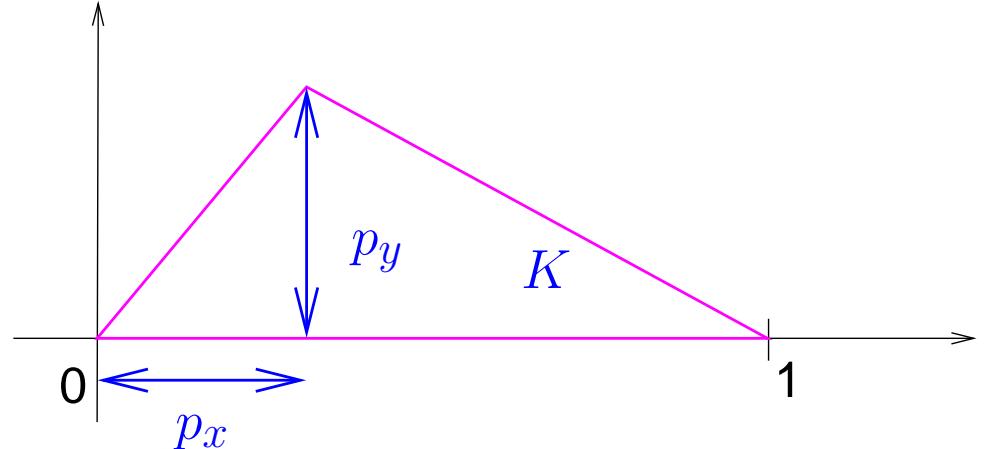
$$\|u - \mathbf{I}_1 u\|_{L^2(K)} \leq C_{K,2} h_k^2 \|u\|_{H^2(K)}, \quad \|u - \mathbf{I}_1 u\|_{H^1(K)} \leq C_K h_K \|u\|_{H^2(K)}.$$

Note: Merely translating, rotating, or scaling K does not affect the constants $C_{K,2}$ and C_K . Therefore, we can restrict ourselves to “canonical triangles”. Every general triangle can be mapped to one of these by translating, rotating, and scaling.

$$C_{K,2} := \sup_{u \in H^2(K) \setminus \{0\}} \frac{\|u - \mathbf{I}_1 u\|_{L^2(K)}}{\|u\|_{H^2(K)}}, \quad C_K := \sup_{u \in H^2(k) \setminus \{0\}} \frac{\|u - \mathbf{I}_1 u\|_{H^1(K)}}{\|u\|_{H^2(K)}},$$

on triangle $K := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} p_x \\ p_y \end{pmatrix} \right\}$.

Sampling the space of “canonical” triangles
(modulo similarity)



$$0 \leq p_x, p_y \leq 1 .$$

+ Numerical computation of $C_K, C_{K,2}$

implementation by A. Inci (spectral polynomial
Galerkin method)

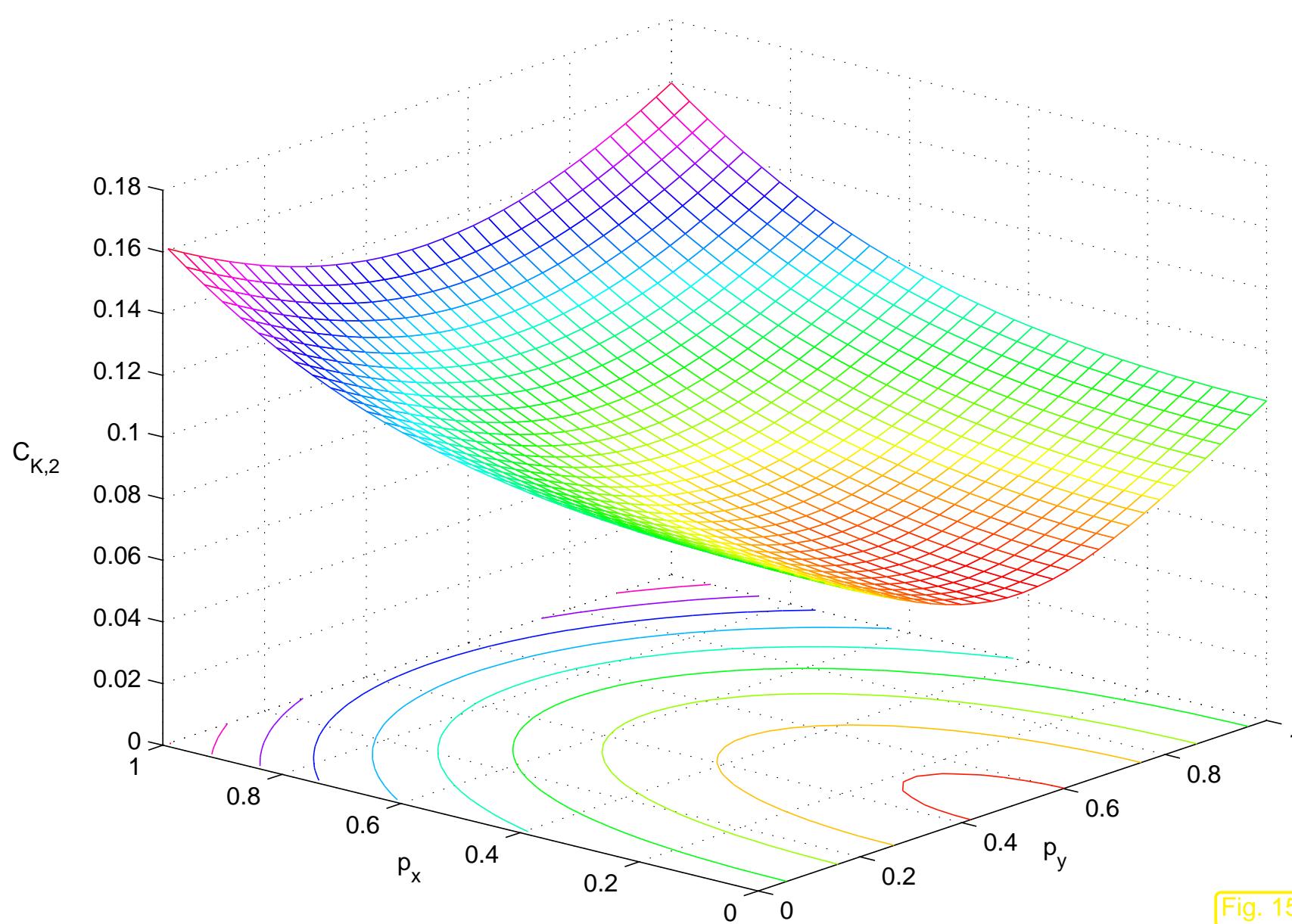


Fig. 150

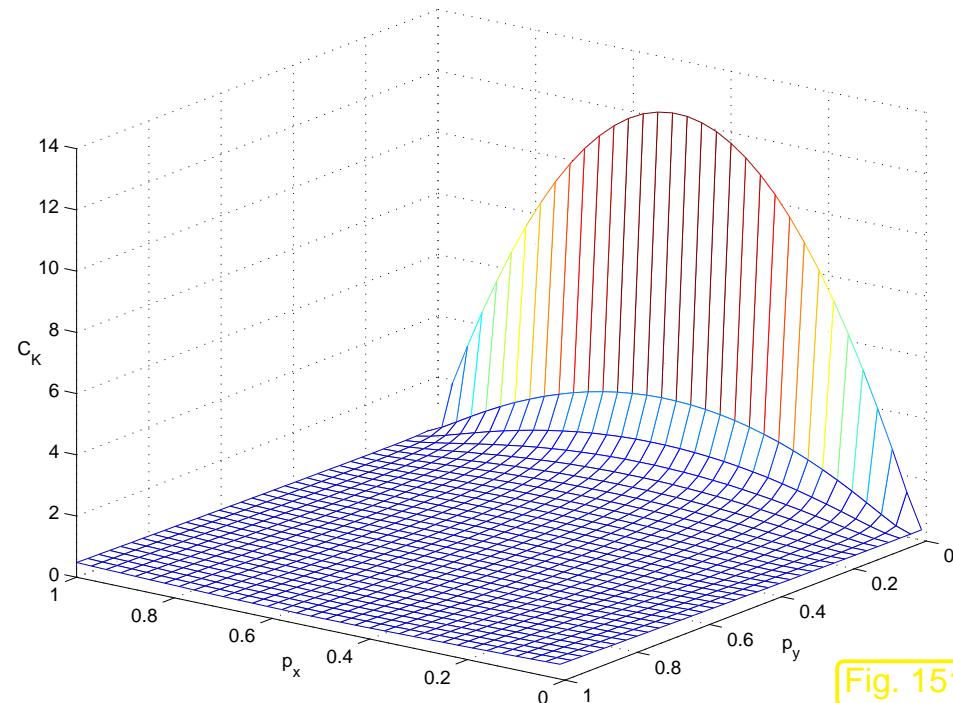


Fig. 151

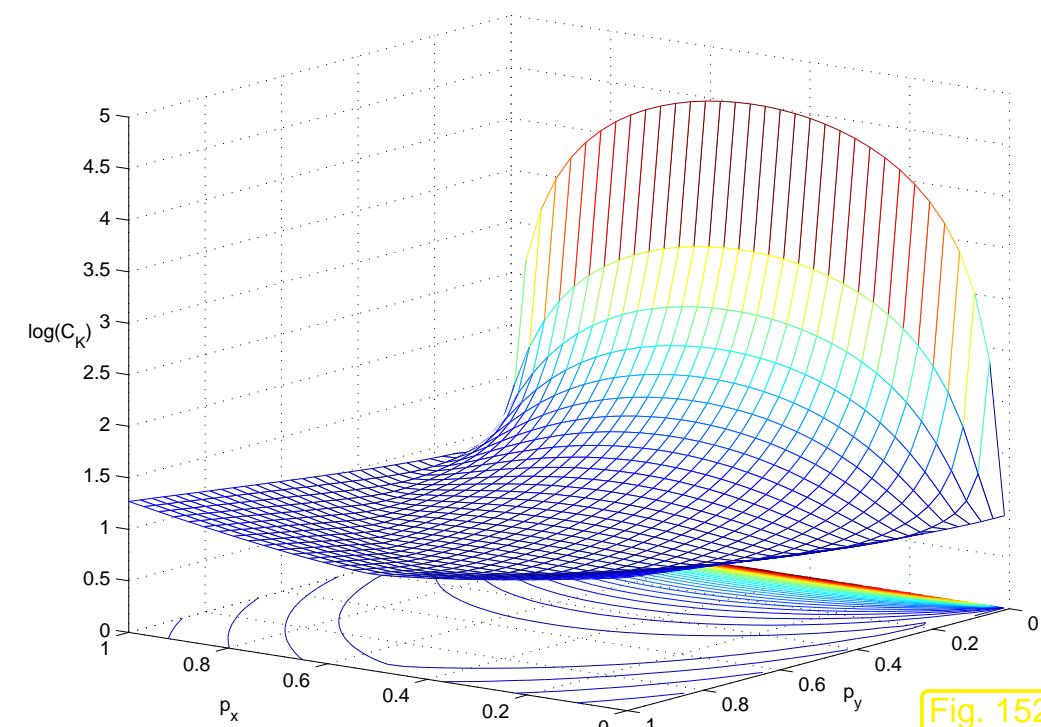


Fig. 152

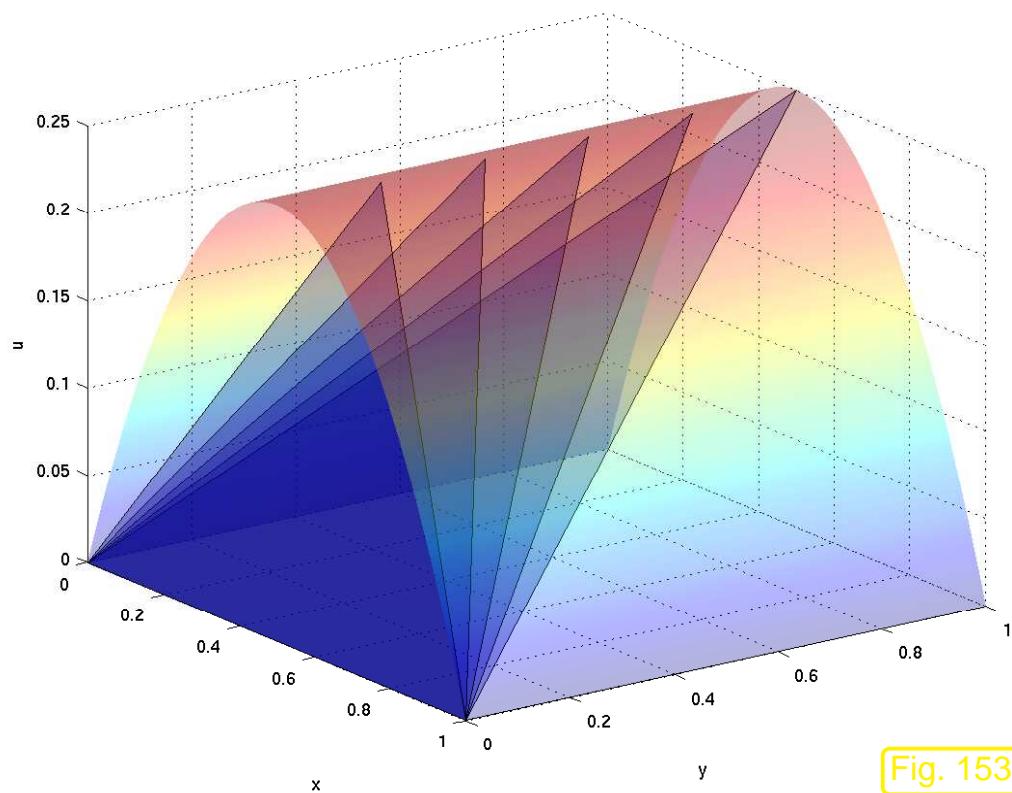
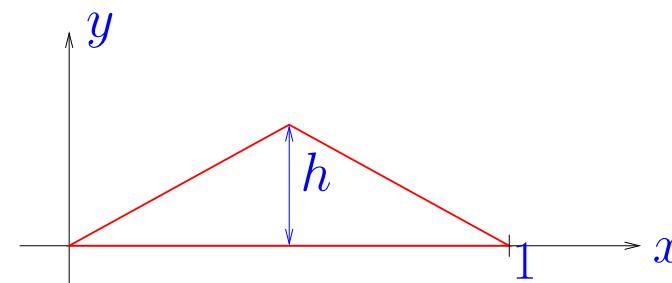


Fig. 153



triangle $K := \text{convex} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1/2 \\ h \end{pmatrix} \right\}$, $h > 0$,
 $u(x, y) = x(1 - x)$, $0 < x < 1$.

◀ linear interpolant of u on K as $h \rightarrow 0$

The interpolant becomes steeper and steeper as $h \rightarrow 0$:

► $\|u\|_{H^2(K)}^2 = \frac{3031}{1440}h$, $\|u - I_1 u\|_{H^1(K)}^2 = \frac{29}{2880}h + \frac{1}{12}h + \frac{1}{32}h^{-1}$, $\|u - I_1 u\|_{L^2(K)}^2 = \frac{29}{2889}h$



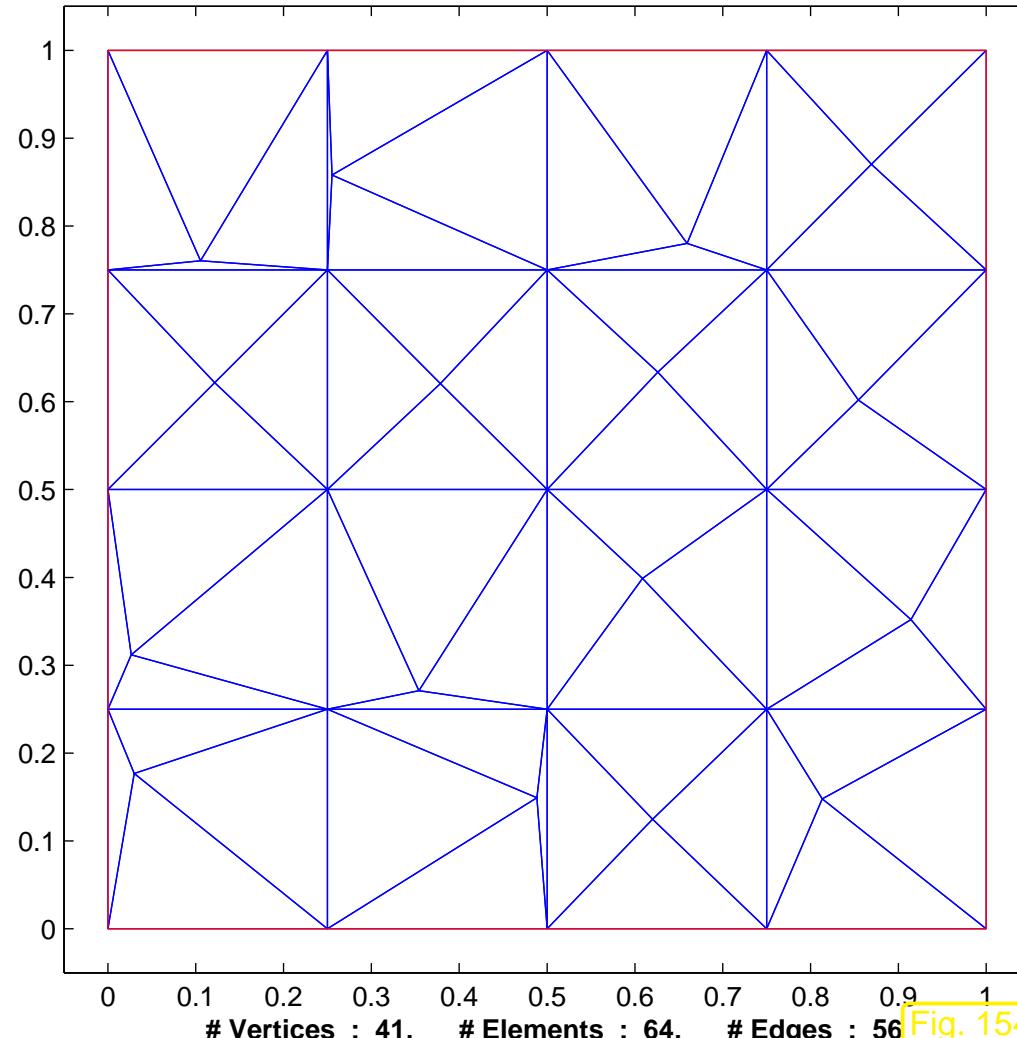
$$\frac{\|u - \mathbb{I}_1 u\|_{H^1(K)}^2}{\|u\|_{H^2(K)}^2} \geq \frac{269}{6062} + \frac{45}{3031} h^{-2} \quad , \quad \frac{\|u - \mathbb{I}_1 u\|_{L^2(K)}^2}{\|u\|_{H^2(K)}^2} = \frac{29}{6062} .$$

Example 5.3.39 (Good accuracy on “bad” meshes).

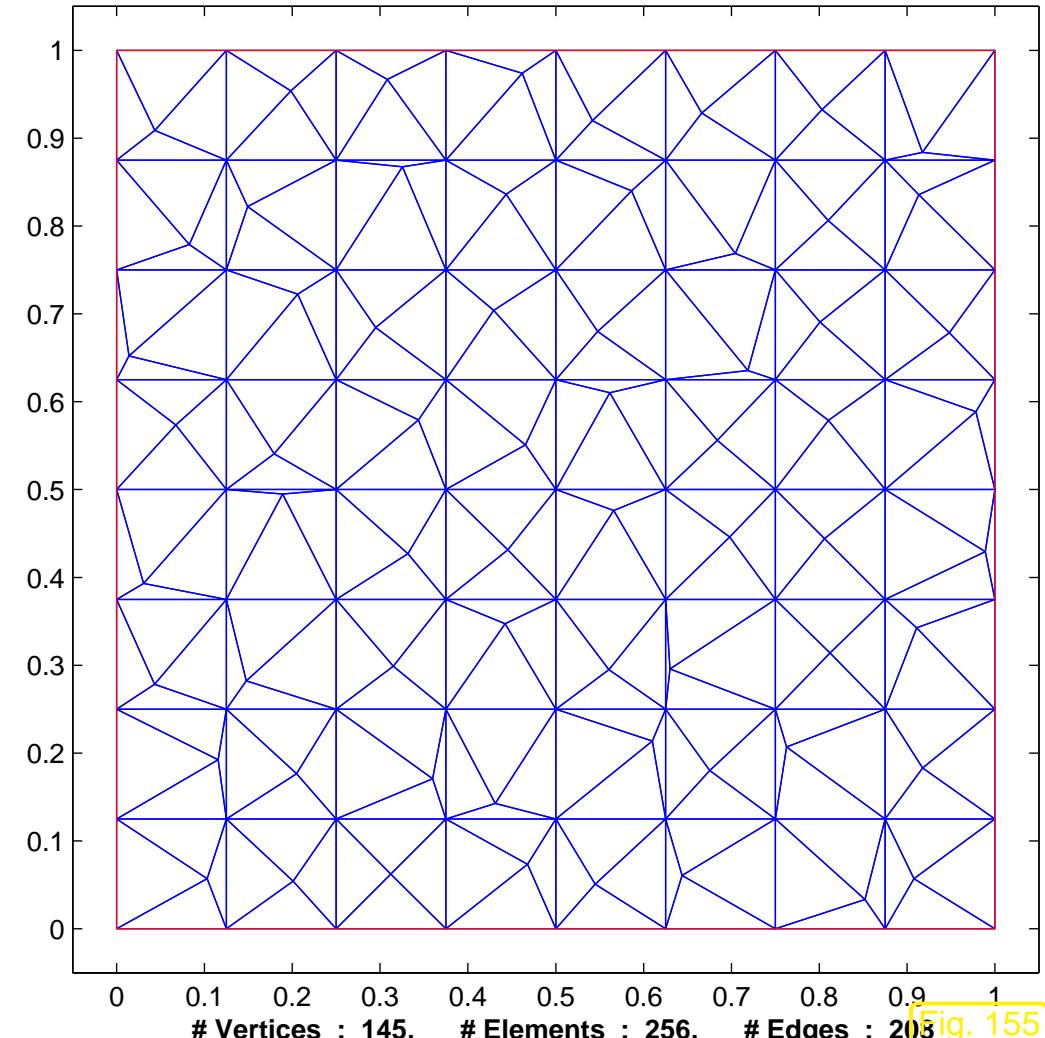
$\Omega =]0, 1[^2$, $u(x_1, x_2) = \sin(\pi x_1) \sin(\pi x_2)$, BVP $-\Delta u = f$, $u|_{\partial\Omega} = 0$, finite element Galerkin discretization on triangular meshes, $V_N = \mathcal{S}_{1,0}^0(\mathcal{M})$.

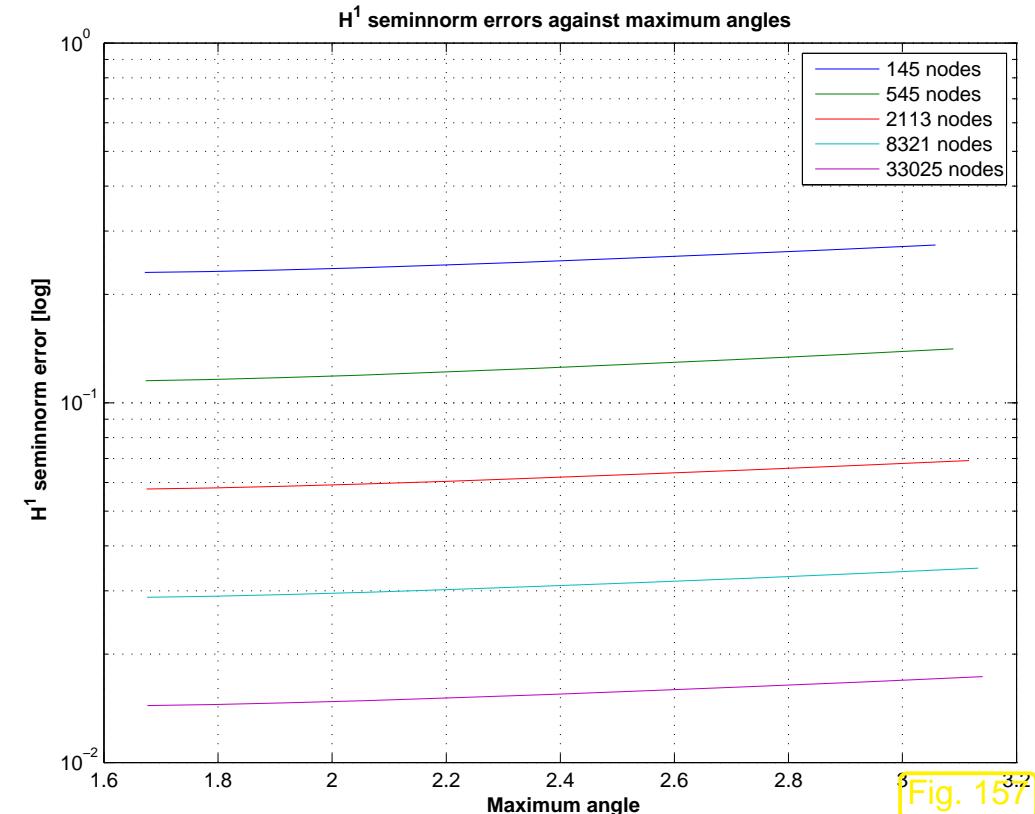
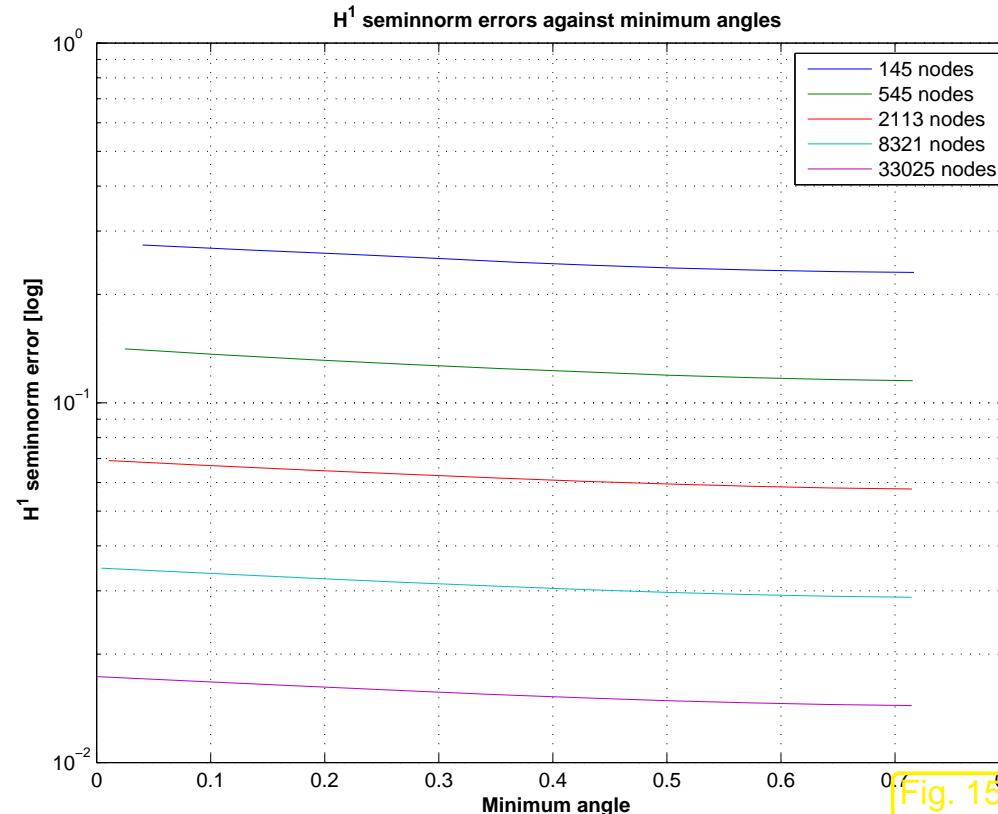
☞ meshes created by random distortion of tensor product grids

2D triangular mesh



2D triangular mesh





Monitored: for different mesh resolutions, $H^1(\Omega)$ -seminorm of discretization error as function of smallest/largest angle in the mesh.

Observation:

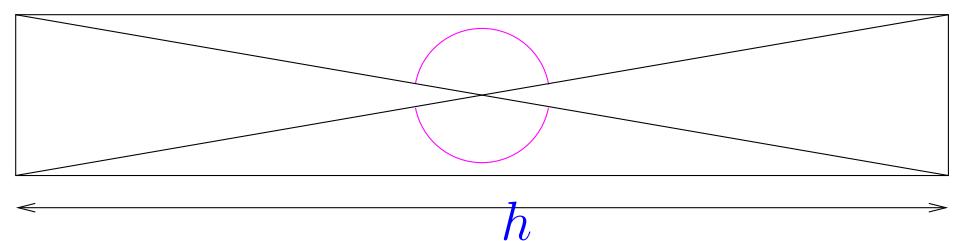
Accuracy does *not* suffer much from distorted elements !

Example 5.3.40 (Gap between interpolation error and best approximation error).

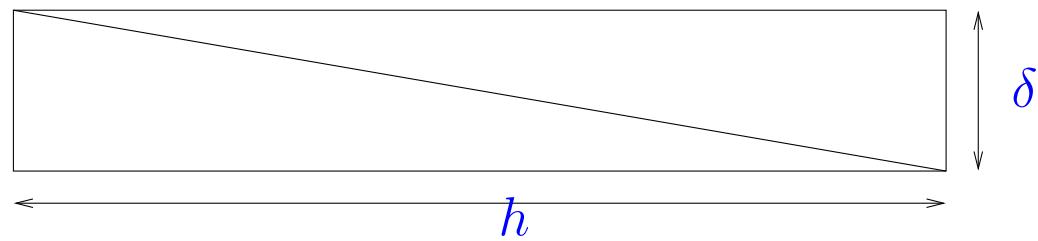
Ex. 5.3.39 raises doubts whether the interpolation error can be trusted to provide good, that is, reasonably sharp bounds for the best approximation error.

In this example we will see that

$$\inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} \|u - v_N\|_1 \ll \|u - I_p u\|_{H^1(\Omega)} \text{ is possible !}$$



Elementary cell of “bad mesh” \mathcal{M}_{bad}



Elementary cell of “good mesh” $\mathcal{M}_{\text{good}}$

On “bad” mesh : $\sup_{u \in H^2(\Omega)} \frac{\|u - I_1 u\|_{H^1(\Omega)}}{\|u\|_{H^2(\Omega)}} \rightarrow \infty$ as $h/\delta \rightarrow \infty$,

On “good” mesh : $\sup_{u \in H^2(\Omega)} \frac{\|u - I_1 u\|_{H^1(\Omega)}}{\|u\|_{H^2(\Omega)}}$ uniformly bounded in h/δ .

Yet, $\inf_{v_N \in \mathcal{S}_1^0(\mathcal{M}_{\text{bad}})} \|u - v_N\|_{H^1(\Omega)} \leq \inf_{v_N \in \mathcal{S}_1^0(\mathcal{M}_{\text{good}})} \|u - v_N\|_{H^1(\Omega)} \quad \forall u \in H^2(\Omega)$.



5.3.5 General approximation error estimates

In Sect. 5.3.2 we only examined the behavior of norms of the interpolation error for piecewise linear interpolation into $\mathcal{S}_1^0(\mathcal{M})$, that is, the case of Lagrangian finite elements of degree $p = 1$.

However, Ex. 5.2.2 sent the clear message that quadratic Lagrangian finite elements achieve fast convergence of the energy norm of the Galerkin discretization error, see Fig. 134, 135.



On the other quadratic finite elements could not deliver faster convergence in Ex. 5.2.6.

In this section we learn about theoretical results that shed light on these observations and extend the results of Sect. 5.3.2.

Remark 5.3.41 (L^∞ interpolation error estimate in 1D).

The faster convergence of quadratic Lagrangian FE in Ex. 5.2.2 does not come as a surprise: recall the esimtate from [14, Eq. 9.2.1]:

$$\|u - \mathbf{I}_p u\|_{L^\infty([a,b])} \leq \frac{h_M^{p+1}}{(p+1)!} \|u^{(p+1)}\|_{L^\infty([a,b])} \quad \forall u \in C^{p+1}([a,b]) ,$$

where $\mathbf{I}_p u$ is the \mathcal{M} -piecewise polynomial interpolant of u of local degree p . It generalizes (5.3.5).

>
$$\|u - \mathbf{I}_p u\|_{L^\infty([a,b])} = O(h_M^{p+1}) !$$



The following theorem summarized best approximation results for **affine equivalent** Lagrangian FE spaces $\mathcal{S}_p^0(\mathcal{M})$ (\rightarrow Sect. 3.4) on mesh \mathcal{M} of a bounded polygonal/polyhedral domain $\Omega \subset \mathbb{R}^d$. It is the result of many years of research in approximation theory, see [18, Sect. 3.3], [2].

Theorem 5.3.42 (Best approximation error estimates for Lagrangian finite elements).

Let $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, be a bounded polygonal/polyhedral domain equipped with a mesh \mathcal{M} consisting of simplices or parallelepipeds. Then, for each $k \in \mathbb{N}$, there is a constant $C > 0$ depending only on k and the shape regularity measure $\rho_{\mathcal{M}}$ such that

$$\inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} \|u - v_N\|_{H^1(\Omega)} \leq C \left(\frac{h_{\mathcal{M}}}{p} \right)^{\min\{p+1,k\}-1} \|u\|_{H^k(\Omega)} \quad \forall u \in H^k(\Omega). \quad (5.3.43)$$

This theorem is a typical example of finite element analysis results that you can find in the literature. It is important to know what kind of information can be gleaned from statements like that of Thm. 5.3.42.

Remark 5.3.44 (“Generic constants”).

A statement like (5.3.43) is typical of a priori error estimates in the numerical analysis literature, which often come in the form

$$\|u - u_N\|_X \leq C \cdot \text{“discretization parameter”} \cdot \|u\|_Y ,$$

where

- $C > 0$ is not specified precisely or only claimed to exist (though, in principle, they could be computed),
- C must neither depend on the exact solution u nor the discrete solution u_N ,
- the possible dependence of C on problem parameters or discretization parameters has to be stated unequivocally.

Such constants $C > 0$ are known as **generic constants**. Customarily, different generic constants are even denoted by the same symbol (“ C ” is most common). △

Remark 5.3.45 (Nature of a priori estimates). → Sect. 1.6.2

Cea's lemma, Thm. 5.1.10 ➤ Thm. 5.3.42 implies a priori estimates of the energy norm of the finite element Galerkin discretization error (see also Rem. 5.3.31) of the form

$$\|u - u_N\|_a \leq C \left(\frac{h_M}{p} \right)^{\min\{p+1,k\}-1} \|u\|_{H^k(\Omega)}, \quad (5.3.46)$$

where u is the exact solution of the discretized 2nd-order elliptic boundary value problem.



(5.3.46) does not give concrete information about $\|u - u_N\|_a$, because

- we do not know the value of the “generic constant” $C > 0$, see Rem. 5.3.44,
- as u is unknown, a bound for $\|u\|_{H^k(\Omega)}$ may not be available.

A priori error estimates like (5.3.46) exhibit only the *trend* of the (norm of) the discretization error as discretization parameters h_M (mesh width), p (polynomial degree) are varied.

Remark 5.3.47. The estimate of Thm. 5.3.42 is *sharp*: the powers of h_M and p cannot be increased.



What do Thm. 5.3.42, (5.3.46), tell us about the *efficiency* of a Lagrangian finite element Galerkin discretization of a 2nd-order elliptic BVP?

Question 5.3.48. *What computational effort buys us what error (measured in energy norm)?*

Bad luck (\rightarrow Rem. 5.3.45): actual error norm remains elusive! Therefore, rephrase the question so that it fits the available information about the effect of changing discretization parameters on the error:

Question 5.3.49. *What increase in computational effort buys us a prescribed decrease of the (energy norm of the) error?*

5.3

The answer to this question offers an a priori gauge of the *asymptotic efficiency* of a discretization method.

Convention: computational effort \approx number of unknowns $N = \dim \mathcal{S}_p^0(\mathcal{M})$ (**problem size**)

Framework: family \mathbb{M} of simplicial meshes of domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, created by *global* regular refinement of a single initial mesh

Global regular refinement of a simplicial mesh (\rightarrow Ex. 5.1.12)

- avoids greater distortion of “child cells” w.r.t. their parents,
- spawns meshes with fairly uniform size h_K of cells.

$$\begin{aligned} \exists C > 0: \quad & \rho_{\mathcal{M}} \leq C, \\ \exists C > 0: \quad & \max\{h_K/h_{K'}, K, K' \in \mathcal{M}\} \leq C, \end{aligned} \quad \forall \mathcal{M} \in \mathbb{M} .$$

Now, for meshes $\in \mathbb{M}$, we investigate “ N -dependence”, $N = \dim \mathcal{S}_p^0(\mathcal{M})$, of energy norm of finite element discretization error:

Counting argument $N = \dim \mathcal{S}_p^0(\mathcal{M}) \approx p^d h_{\mathcal{M}}^{-d} \Rightarrow \boxed{\frac{h_{\mathcal{M}}}{p} \approx N^{-1/d}}.$ (5.3.50)

dimensions of local spaces, Lemma 3.3.6 $\sim \#\mathcal{M} \sim \#\mathcal{V}(\mathcal{M}), \mathcal{E}(\mathcal{M})$ etc.

Notation: $\approx \hat{=}$ equivalence up to constants only depending on γ (in \mathbb{M}_{γ}), Ω

Example 5.3.51 (Dimensions of Lagrangian finite element spaces on triangular meshes).

$d = 2$: for triangular meshes \mathcal{M} , by Lemma 3.3.6

$$\dim \mathcal{S}_p^0(\mathcal{M}) = \#\{\text{nodes}(\mathcal{M})\} + \#\{\text{edges}(\mathcal{M})\} (p - 1) + \#\mathcal{M} \frac{1}{2}(p - 1)(p - 2).$$

1 basis function per vertex

$p - 1$ basis functions per edge

$\frac{1}{2}(p - 1)(p - 2)$ “interior” basis functions

Geometric considerations: the number of triangles sharing a vertex can be bounded in terms of ρ_M , because ρ_M implies a lower bound for the smallest angles of the triangular cells.

$$\exists C = C(\rho_M): \#\{K_j \in \mathcal{M}: \overline{K}_i \cap \overline{K}_j \neq \emptyset\} \leq C \quad (i = 1, 2, \dots, \#\mathcal{M}) .$$

If every vertex belongs only to a small number of triangles, the number $\#\{\text{nodes}(\mathcal{M})\}$ can be bounded by $C \cdot \#\mathcal{M}$, where $C > 0$ will depend on ρ_M only. The same applies to the edges.



$$\#\{\text{nodes}(\mathcal{M})\}, \#\{\text{edges}(\mathcal{M})\} \approx \#\mathcal{M} .$$



$$\dim \mathcal{S}_p^0(\mathcal{M}) \approx (\#\mathcal{M})p^2 , \tag{5.3.52}$$

with constants hidden in \approx depending on ρ_M only.

Now, we merge (5.3.46) and (5.3.50):

$$u \in H^k(\Omega)$$

Thm. 5.3.42

$$\inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} \|u - v_N\|_{H^1(\Omega)} \leq C N^{-\frac{\min\{p,k-1\}}{d}} \|u\|_{H^k(\Omega)}, \quad (5.3.53)$$

with $C > 0$ depending *only* on d , p , k , and $\rho_{\mathcal{M}}$.

(5.3.53) \Rightarrow algebraic convergence (\rightarrow Def. 1.6.19) in problem size

$$(\text{rate } \frac{\min\{p, k-1\}}{d})$$

We observe that

- the rate of convergence is limited by the polynomial degree p of the Lagragian FEM,
- the rate of convergence is limited by the smoothness of the exact solution u , measured by means of the Sobolev index k , see Sect. 5.3.3,
- the rate of convergence will be worse for $d = 3$ than for $d = 2$, the effect being more pronounced for small k or p .

Answer to Question 5.3.49:

Assumption: a priori error estimate (5.3.53) is *sharp*

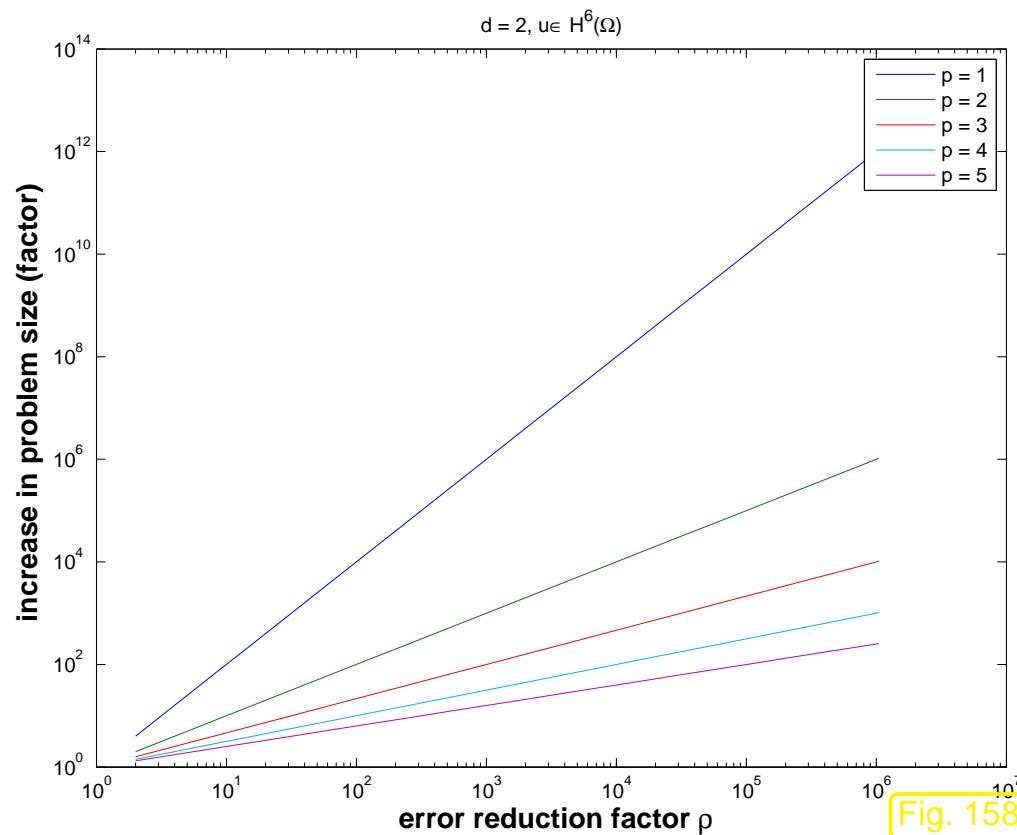
$$\exists C = C(u, \dots) > 0: \text{error norm}(N) \approx CN^{-\frac{\min\{p, k-1\}}{d}} \quad \forall M \in \mathbb{M}.$$

$$\Rightarrow \frac{\text{error norm}(N_1)}{\text{error norm}(N_2)} \approx \left(\frac{N_1}{N_2}\right)^{-\frac{\min\{p, k-1\}}{d}}.$$

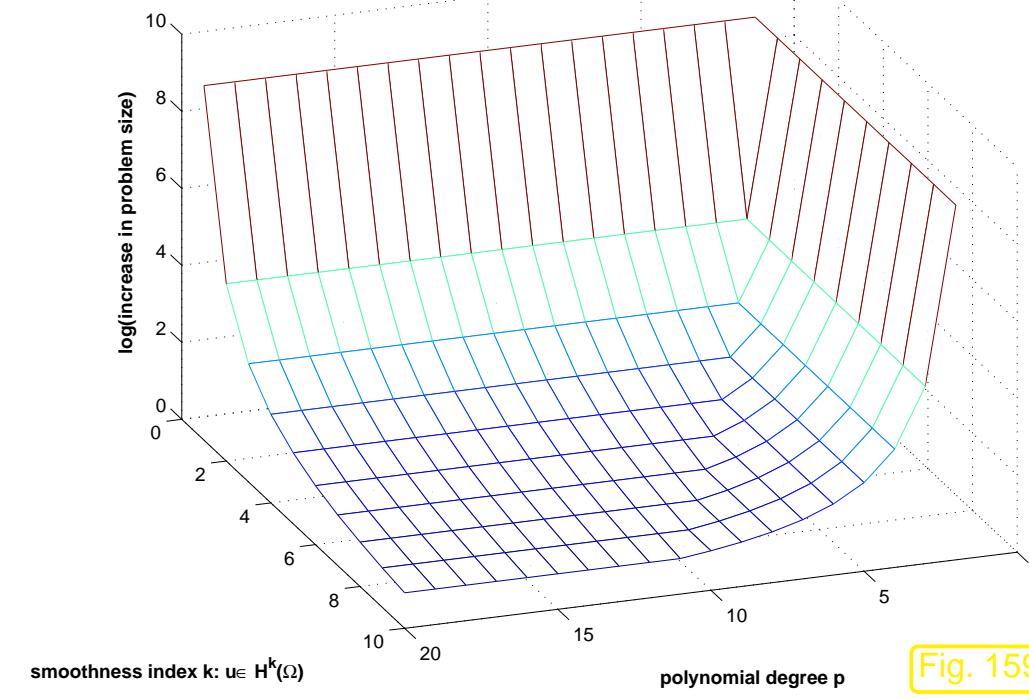
reduction of (the energy norm of
the error by a factor $\rho > 1$

requires

increase of the problem size
by factor $\rho^{\frac{d}{\min\{p, k-1\}}}$



exact solution $u \in H^6(\Omega)$



error reduction by factor $\rho = 100$

Discussion:

Solution $u \in H^k(\Omega) \Rightarrow$ optimal asymptotic efficiency for $p = k - 1$

Remark 5.3.54 (Asymptotic estimates).

Recall (\rightarrow Sect. 1.6.2):

convergence is an asymptotic notion

Now we deduce **asymptotic** estimates for the best approximation errors from Thm. 5.3.42, and (5.3.53), in particular, for the case $N \rightarrow \infty$:

- h-refinement: p fixed, $h_{\mathcal{M}} \rightarrow 0$ for $\mathcal{M} \in \mathbb{M}$:

$$(5.3.53) \Rightarrow \text{algebraic convergence w.r.t. } N$$

☞ $p \leq k - 1$ ►

$$\inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} \|u - v_N\|_1 = O(N^{-p/d}) \quad (5.3.55)$$

☞ $k \leq p + 1$ ►

$$\inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} \|u - v_N\|_1 = O(N^{-(k-1)/d}) \quad (5.3.56)$$

Note: for very smooth solution u , i.e. $k \gg 1$, polynomial degree p limits speed of convergence

- p-refinement: $\mathcal{M} \in \mathbb{M}$ fixed, $p \rightarrow \infty$:

☞ p large ►

$$\inf_{v_N \in \mathcal{S}_p^0(\mathcal{M})} \|u - v_N\|_1 = O(N^{-(k-1)/d}) \quad (5.3.57)$$

Note: arbitrarily fast (**super**-)algebraic convergence for very smooth solutions $u \in C^\infty(\bar{\Omega})$



5.4 Elliptic regularity theory

Crudely speaking, in Sect. 5.3.5 we saw that the asymptotic behavior of the Lagrangian finite element Galerkin discretization error (for 2nd-order elliptic BVPs) can be predicted provided that

- we use families of meshes, whose cells have rather uniform size and whose shape regularity measure is uniformly bounded,
- we have an *idea about the smoothness of the exact solution u* , that is, we know $u \in H^k(\Omega)$ for a (maximal) k , see Thm. 5.3.42.

Knowledge about the mesh can be taken for granted, but

how can we guess the smoothness of the (unknown !) exact solution u ?

A (partial) answer is given in this section.

Focus: Scalar 2nd-order elliptic BVP with homogeneous Dirichlet boundary conditions

$$-\operatorname{div}(\sigma(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \Omega , \quad u = g \quad \text{on } \partial\Omega .$$

To begin with, we summarize the available information:

➤ Known:

u solves BVP

+

Information about coefficient σ , domain Ω , source function f , boundary data g

u will belong to a certain **class of functions** (e.g. subspace $S \subset V$)

Example 5.4.1 (Elliptic lifting result in 1D).

$d = 1$, $\Omega =]0, 1[$, coefficient $\sigma \equiv 1$, homogeneous Dirichlet boundary conditions:

$$u'' = f \quad , \quad u(0) = u(1) = 0 .$$

Obvious:

$$f \in H^k(\Omega) \Rightarrow u \in H^{k+2}(\Omega) \quad (\text{a lifting theorem})$$



Can this be generalized to higher dimensions $d > 1$?

Partly so:

Theorem 5.4.2 (Smooth elliptic lifting theorem).

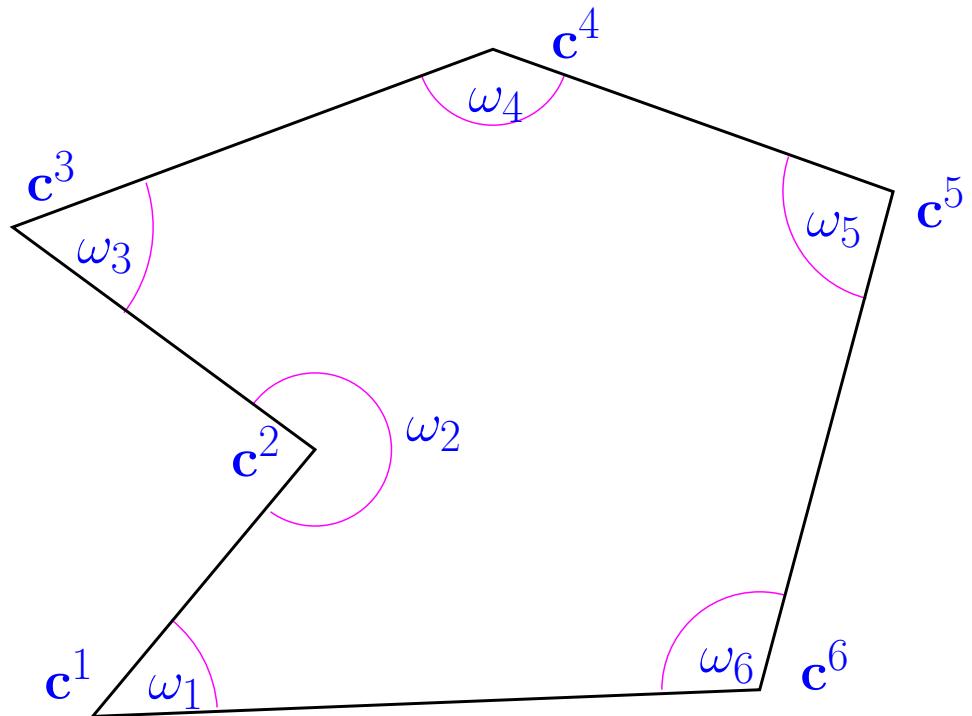
If $\partial\Omega$ is C^∞ -smooth, ie. possesses a local parameterization by C^∞ -functions, and $\sigma \in C^\infty(\overline{\Omega})$, then, for any $k \in \mathbb{N}$,

$$u \in H_0^1(\Omega) \quad \text{and} \quad -\operatorname{div}(\sigma \operatorname{grad} u) \in H^k(\Omega) \quad \Rightarrow \quad u \in H^{k+2}(\Omega) .$$

$$u \in H^1(\Omega), \quad -\operatorname{div}(\sigma \operatorname{grad} u) \in H^k(\Omega) \quad \text{and} \quad \operatorname{grad} u \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega$$

In addition, for such u there is $C = C(k, \Omega, \sigma)$ such that

$$\|u\|_{H^{k+2}(\Omega)} \leq C \|\operatorname{div}(\sigma \operatorname{grad} u)\|_{H^k(\Omega)} .$$



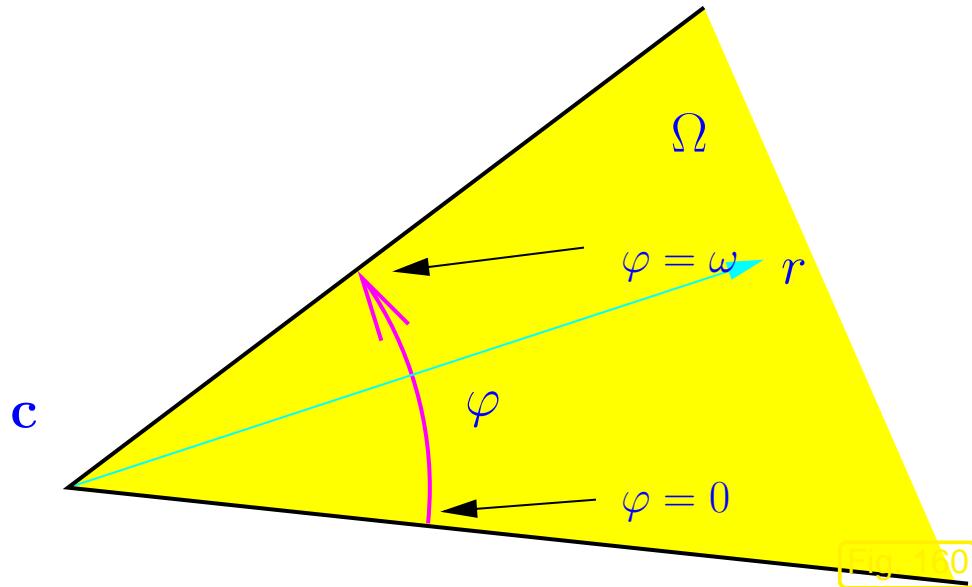
These are very common in engineering applications (“CAD-geometries”).

▷ polygonal domain with corners c^i

How will the corners affect the smoothness of solutions of

$$u \in H_0^1(\Omega): \quad \Delta u = f \in C^\infty(\bar{\Omega})?$$

Example 5.4.3 (Corner singular functions).



corner singular function

$$u_s(r, \varphi) = r^{\frac{\pi}{\omega}} \sin\left(\frac{\pi}{\omega}\varphi\right), \quad (5.4.4)$$

$$r \geq 0, \quad 0 \leq \varphi \leq \omega.$$

(in local polar coordinates)

► $u_s = 0$ on $\partial\Omega$ locally at c !

An in fact:

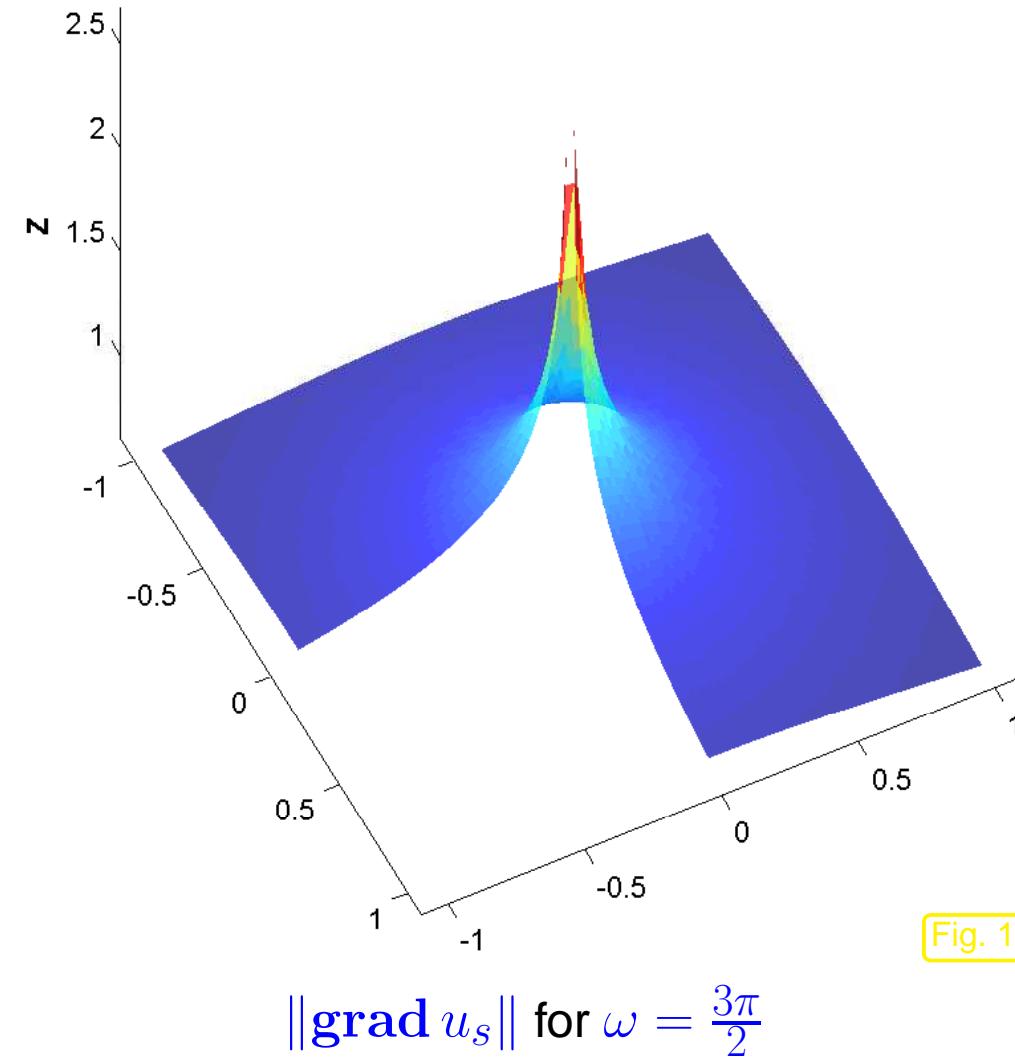
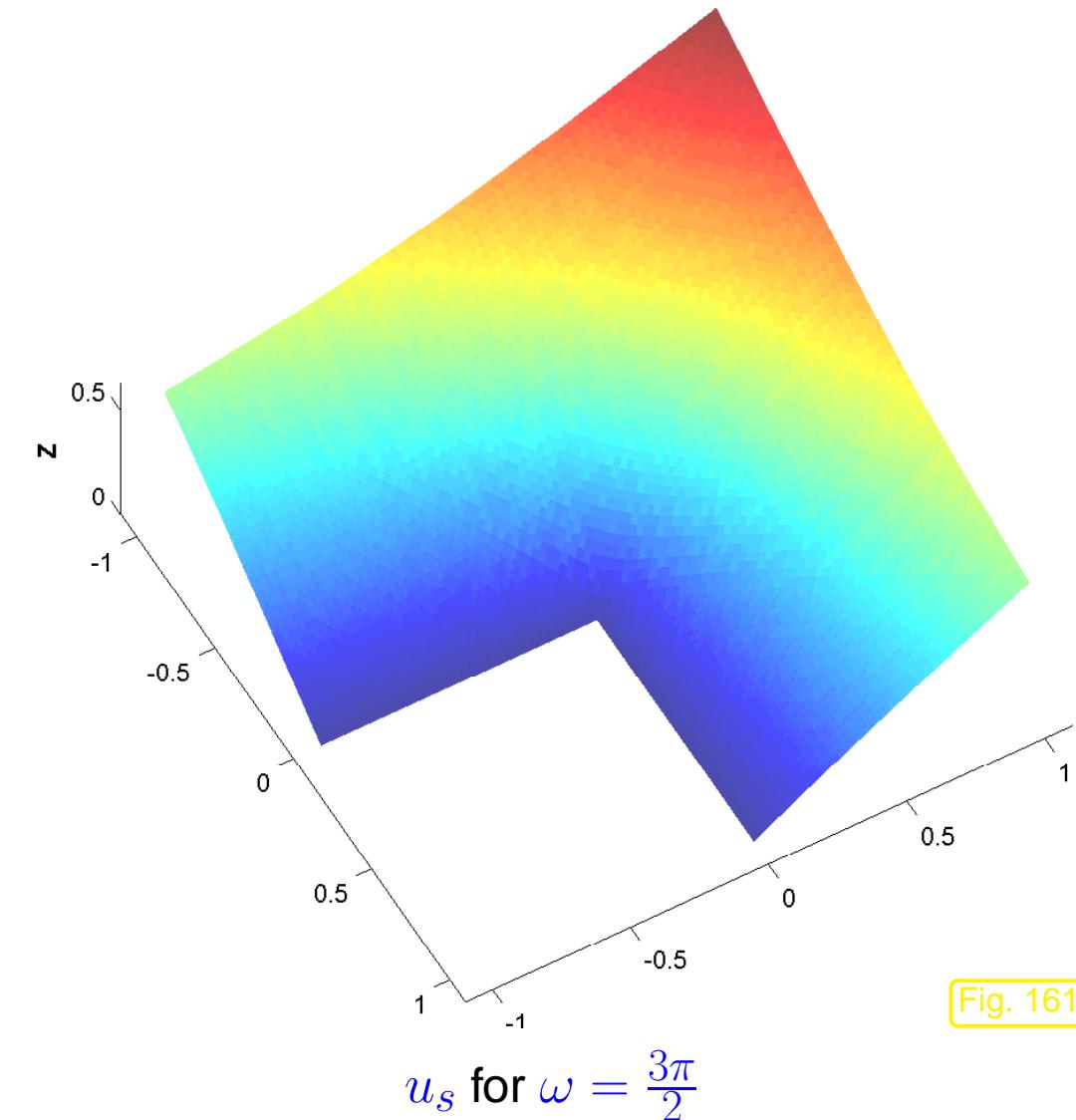
$$\Delta u_s = 0 \quad \text{in } \Omega !$$

Recall: Δ in polar coordinates:

$$\Delta u = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial u}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 u}{\partial \varphi^2}. \quad (5.4.5)$$

$$\begin{aligned} \stackrel{(5.4.4)}{\implies} \Delta u_s(r, \varphi) &= \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\pi}{\omega} r^{\frac{\pi}{\omega}-1} \sin\left(\frac{\pi}{\omega}\varphi\right) \right) + \frac{1}{r^2} r^{\frac{\pi}{\omega}} \frac{\partial}{\partial \varphi} \cos\left(\frac{\pi}{\omega}\varphi\right) \frac{\pi}{\omega} \\ &= \left(\frac{\pi}{\omega}\right)^2 r^{\frac{\pi}{\omega}-2} \sin\left(\frac{\pi}{\omega}\varphi\right) - \left(\frac{\pi}{\omega}\right)^2 r^{\frac{\pi}{\omega}-2} \sin\left(\frac{\pi}{\omega}\varphi\right) = 0. \end{aligned}$$

What is “singular” about these functions? Plot them for $\omega = \frac{3\pi}{2}$, cf. Ex. 5.2.6



Recall gradient (2.3.19) in polar coordinates

$$\mathbf{grad} u = \frac{\partial u}{\partial r} \mathbf{e}_r + \frac{1}{r} \frac{\partial u}{\partial \varphi} \mathbf{e}_\varphi . \quad (2.3.19)$$

$$\stackrel{(5.4.4)}{\Rightarrow} \mathbf{grad} u_s(r, \varphi) = \frac{\pi}{\omega} r^{\frac{\pi}{\omega}-1} \left(\sin\left(\frac{\pi}{\omega}\varphi\right) \mathbf{e}_r + \cos\left(\frac{\pi}{\omega}\varphi\right) \mathbf{e}_\varphi \right) .$$

$$\omega > \pi \text{ ("re-entrant corner")} \implies \text{"} \mathbf{grad} u_s(0) = \infty \text{"}$$

How does this “blow-up” of the gradient affect the **Sobolev regularity** (that is, the smoothness as expressed through “ $u_s \in H^k(\Omega)$ ”) of the corner singular function u_s ?

We try to compute $|u|_{H^2(D)}$, with (in polar coordinates, see Fig. 160)

$$D := \{(r, \varphi) : 0 < r < 1, 0 < \varphi < \omega\} .$$

By tedious computations we find

$$\omega > \pi \implies \int_D \left\| D^2 u_s(r, \varphi) \right\|_F^2 r d(r, \varphi) = \infty .$$

$$\stackrel{\text{Def. 5.3.33}}{\implies} \left\{ \begin{array}{l} \omega > \pi \implies u_s \notin H^2(D) \end{array} \right\} .$$



Bad news: With the exception of concocted examples,
corner singular functions like (5.4.4) will be present in the solution of linear scalar
2nd-order elliptic BVP on polygonal domains!

The meaning of “being present” is elucidated in the following theorem:

Theorem 5.4.6 (Corner singular function decomposition).

Let $\Omega \subset \mathbb{R}^2$ be a polygon with J corners \mathbf{c}^i . Denote the polar coordinates in the corner \mathbf{c}^i by (r_i, φ_i) and the inner angle at the corner \mathbf{c}^i by ω_i . Additionally, let $f \in H^l(\Omega)$ with $l \in \mathbb{N}_0$ and $l \neq \lambda_{ik} - 1$, where the λ_{ik} are given by the **singular exponents**

$$\lambda_{ik} = \frac{k\pi}{\omega_i} \quad \text{for } k \in \mathbb{N}. \quad (5.4.7)$$

Then $u \in H_0^1(\Omega)$ with $-\Delta u = f$ in Ω can be decomposed

$$u = u^0 + \sum_{i=1}^J \psi(r_i) \sum_{\lambda_{ik} < l+1} \kappa_{ik} s_{ik}(r_i, \varphi_i), \quad \kappa_{ik} \in \mathbb{R}, \quad (5.4.8)$$

with **regular part** $u^0 \in H^{l+2}(\Omega)$, **cut-off functions** $\psi \in C^\infty(\mathbb{R}^+)$ ($\psi \equiv 1$ in a neighborhood of 0), and **corner singular functions**

$$\begin{aligned} \lambda_{ik} \notin \mathbb{N}: \quad & s_{ik}(r, \varphi) = r^{\lambda_{ik}} \sin(\lambda_{ik}\varphi), \\ \lambda_{ik} \in \mathbb{N}: \quad & s_{ik}(r, \varphi) = r^{\lambda_{ik}} (\ln r) \sin(\lambda_{ik}\varphi). \end{aligned} \quad (5.4.9)$$

$\Omega \subset \mathbb{R}^2$ has re-entrant corners

\Rightarrow

if u solves $\Delta u = f$ in Ω , $u = 0$ on $\partial\Omega$,
then $u \notin H^2(\Omega)$ in general.

Theorem 5.4.10 (Elliptic lifting theorem on convex domains).

If $\Omega \subset \mathbb{R}^d$ convex, $u \in H_0^1(\Omega)$, $\Delta u \in L^2(\Omega)$ \Rightarrow $u \in H^2(\Omega)$.

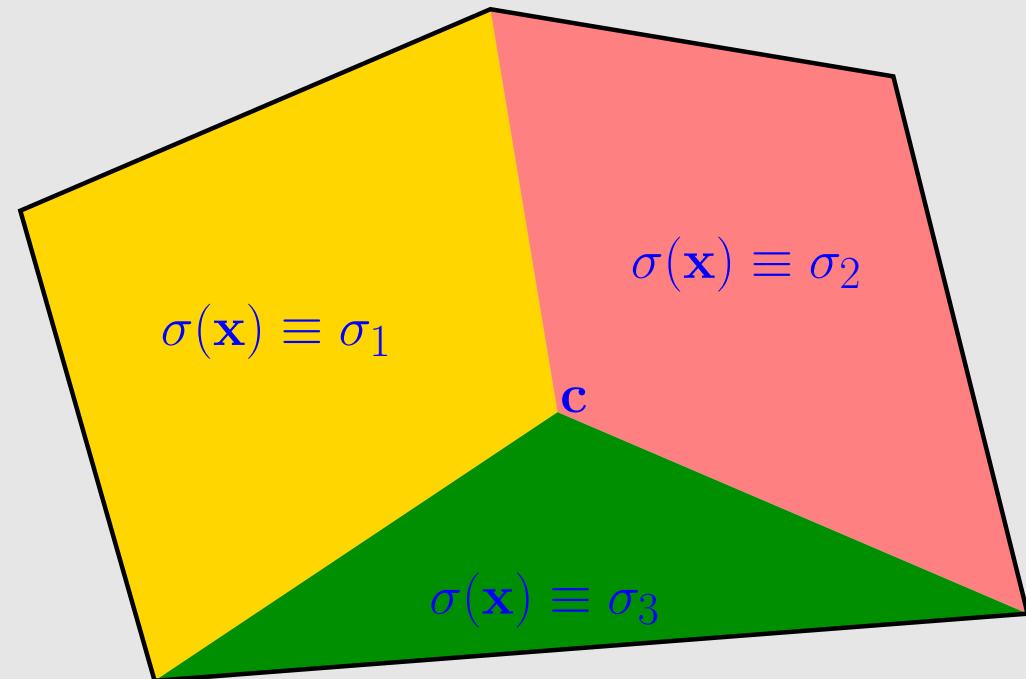
Terminology: if conclusion of Thm. 5.4.10 true \rightarrow Dirichlet problem **2-regular**.

Similar lifting theorems also hold for Neumann BVPs, BVPs with *smooth* coefficients.

Remark 5.4.11 (Causes for non-smoothness of solutions of elliptic BVPs).

Causes for poor Sobolev regularity of solution u of BVPs for $-\operatorname{div}(\sigma(\mathbf{x}) \operatorname{grad} u) = f$:

- Corner of $\partial\Omega$, see above
- Discontinuities of σ
→ singular functions at “material corners”
- Mixed boundary conditions
- Non-smooth source function f



△

5.5 Variational crimes

Variational crime = replacing (exact) discrete (linear) variational problem

$$u_N \in V_{0,N}: \quad \mathbf{a}(u_N, v_N) = f(v_N) \quad \forall v_N \in V_{0,N}, \quad (3.1.3)$$

with perturbed variational problem

$$\tilde{u}_N \in V_{0,N}: \quad \mathbf{a}_N(\tilde{u}_N, v_N) = f_N(v_N) \quad \forall v_N \in V_{0,N}. \quad (5.5.1)$$

► perturbation of Galerkin solution $u_N \rightarrow$ perturbed solution $\tilde{u}_N \in V_{0,N}$

Approximations $\mathbf{a}_N(\cdot, \cdot) \approx \mathbf{a}(\cdot, \cdot)$, $f_N(\cdot) \approx f(\cdot)$ due to

- use of numerical quadrature → Sect. 3.5.4,
- approximation of boundary $\partial\Omega$ → Sect. 3.6.4.

We are all sinners! Variational crimes are *inevitable* in practical FEM, recall Rem. 1.5.3!

Which “variational petty crimes” can be tolerated?

Guideline 5.5.2. Variational crimes must not affect (type and rate) of asymptotic convergence!

5.5.1 Impact of numerical quadrature

Model problem: on polygonal/polyhedral $\Omega \subset \mathbb{R}^d$:

$$u \in H_0^1(\Omega): \quad \mathbf{a}(u, v) := \int_{\Omega} \sigma(\mathbf{x}) \mathbf{grad} u \cdot \mathbf{grad} v \, d\mathbf{x} = f(v) := \int_{\Omega} f v \, d\mathbf{x} . \quad (5.5.3)$$

Assumptions: σ satisfies (2.5.4), $\sigma \in C^0(\overline{\Omega})$, $f \in C^0(\overline{\Omega})$

- Galerkin finite element discretization, $V_N := \mathcal{S}_p^0(\mathcal{M})$ on simplicial mesh \mathcal{M}
- Approximate evaluation of $\mathbf{a}(u_N, v_N)$, $f(v_N)$ by a fixed stable local numerical quadrature rule
(→ Sect. 3.5.4)
► perturbed bilinear form \mathbf{a}_N , right hand side \mathbf{f}_N (see (5.5.1))

Focus: h -refinement (key discretization parameter is the mesh width $h_{\mathcal{M}}$)

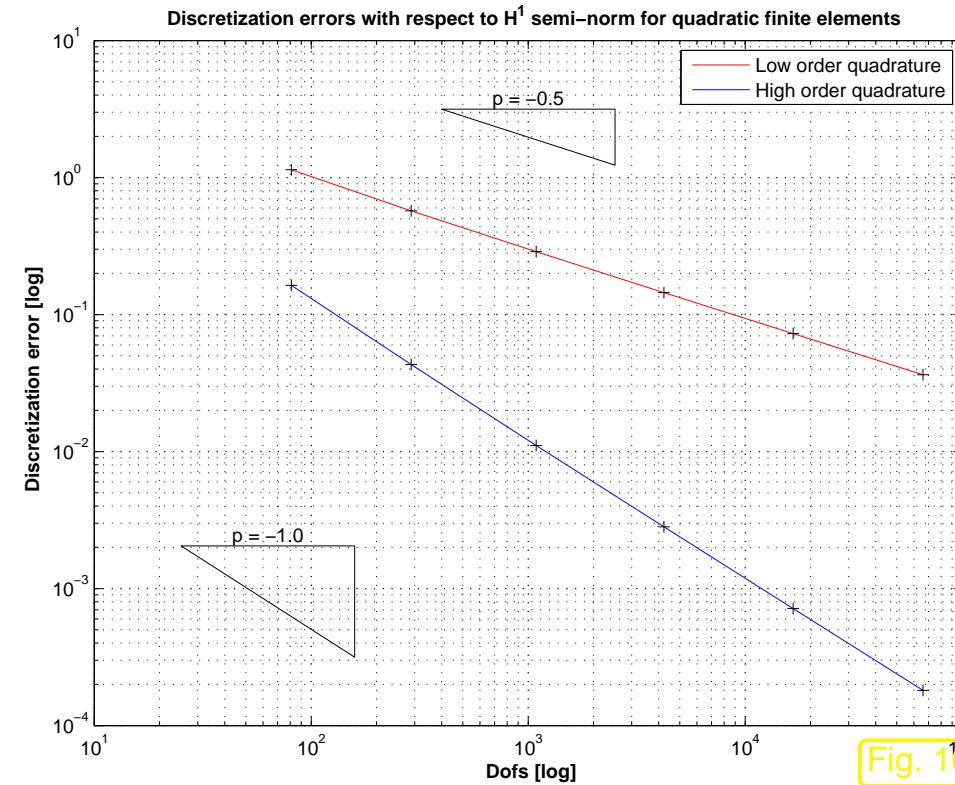
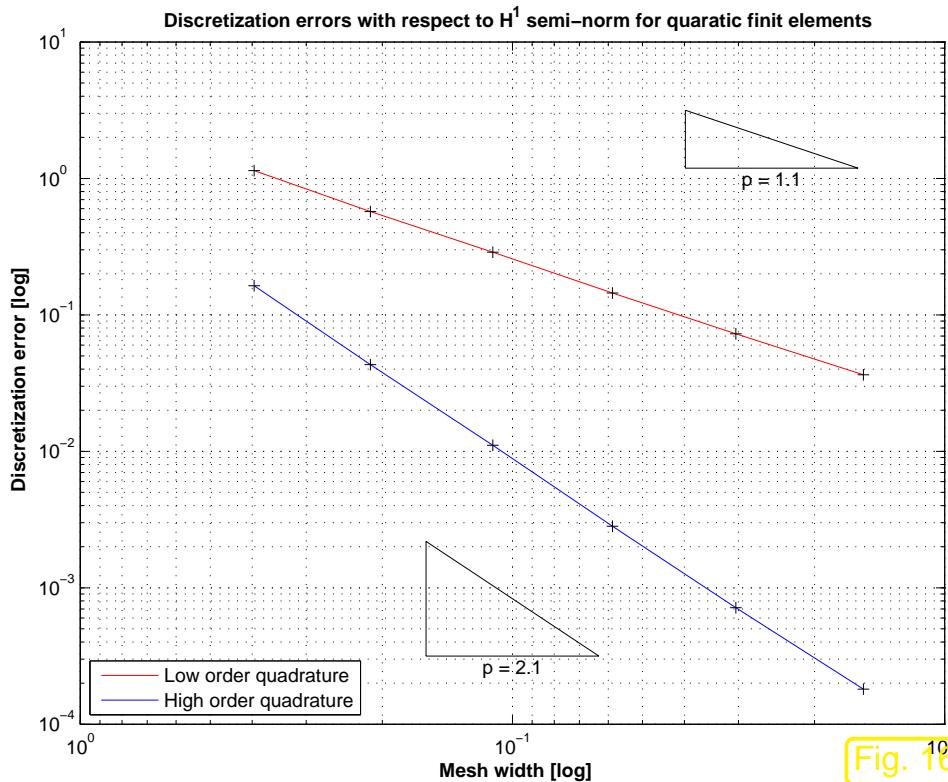
Example 5.5.4 (Impact of numerical quadrature on finite element discretization error).

$$\Omega =]0, 1[^2, \sigma \equiv 1, f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y), (x, y)^T \in \Omega$$

➤ solution $u(x, y) = \sin(\pi x) \sin(\pi y)$, $g = 0$.

Details of numerical experiment:

- Quadratic Lagrangian FE ($V_N = \mathcal{S}_2^0(\mathcal{M})$) on triangular meshes \mathcal{M} , obtained by regular refinement
- “Exact” evaluation of bilinear form by very high order quadrature
- f_N from one point quadrature rule (3.5.37) *of order 2*



$H^1(\Omega)$ -norm of discretization error on unit square ($\text{---} \leftrightarrow$ rule (3.5.37), $\text{—} \leftrightarrow$ rule (3.5.38))

Observation: Use of quadrature rule of order 2

\Rightarrow

Algebraic rate of convergence (w.r.t. N) drops from $\alpha = 1$ to $\alpha = 1/2$!

Finite element theory [6, Ch. 4, §4.1] tells us that the Guideline 5.5.2 can be met, if the local numerical quadrature rule has sufficiently high order. The quantitative results can be condensed into the following rules of thumb:

$$\|u - u_N\|_1 = O(h_{\mathcal{M}}^p) \text{ at best} \quad \blacktriangleright \quad \text{Quadrature rule of order } 2p - 1 \text{ sufficient for } f_N.$$

$$\|u - u_N\|_1 = O(h_{\mathcal{M}}^p) \text{ at best} \quad \blacktriangleright \quad \text{Quadrature rule of order } 2p - 1 \text{ sufficient for } a_N.$$

5.5.2 Approximation of boundary

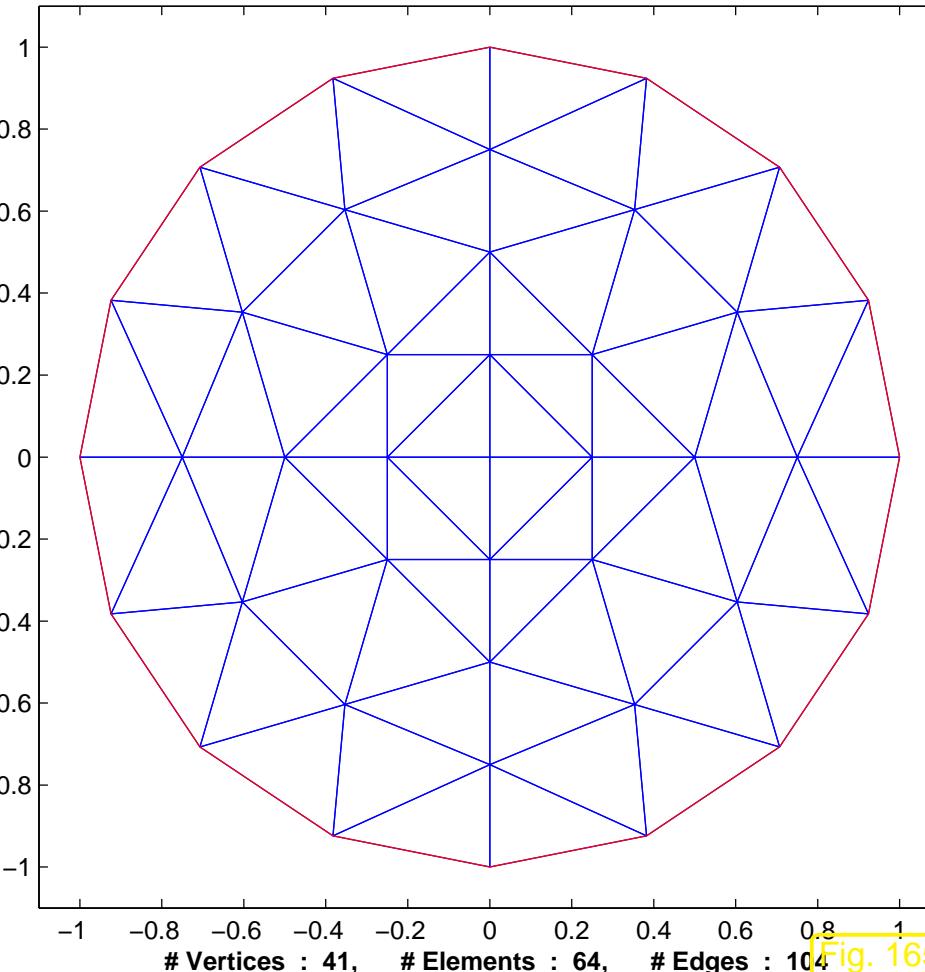
We focus on 2nd-order scalar linear variational problems as in the previous section.

Example 5.5.5 (Impact of linear boundary fitting on FE convergence).

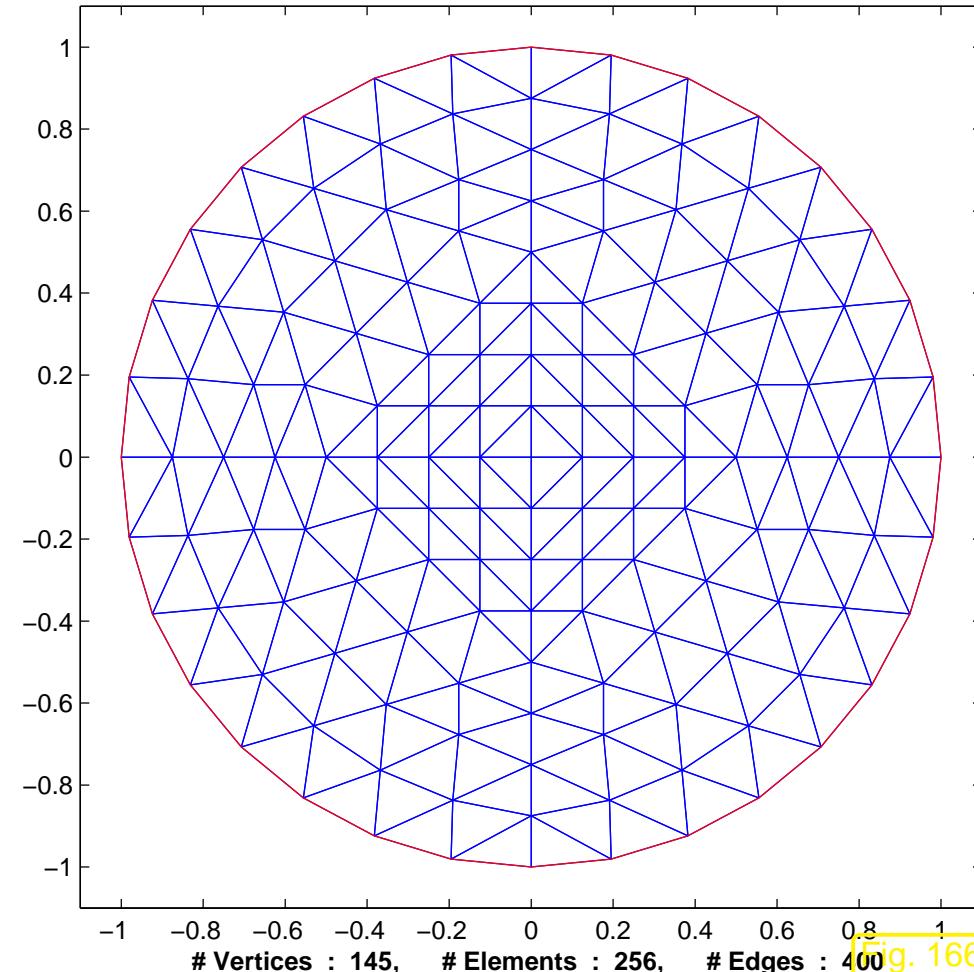
Setting: $\Omega := B_1(0) := \{\mathbf{x} \in \mathbb{R}^2 : |\mathbf{x}| < 1\}$, $u(r, \varphi) = \cos(r\pi/2)$ (polar coordinates)
 $\gg f = \frac{\pi}{2r} \sin(r\pi/2) + \frac{\pi}{2} \cos(r\pi/2)$

- Sequences of unstructured triangular meshes \mathcal{M} obtained by regular refinement (of coarse mesh with 4 triangles) + linear boundary fitting.
- Galerkin FE discretization based on $V_N := \mathcal{S}_{1,0}^0(\mathcal{M})$ or $V_N := \mathcal{S}_{2,0}^0(\mathcal{M})$.
- Recorded: approximate norm $\|\mathbf{u} - \mathbf{u}_N\|_{1,\Omega_h}$, evaluated using numerical quadrature rule (3.5.38).

2D triangular mesh



2D triangular mesh



Linearly boundary fitted unstructured triangular meshes of $\Omega = B_1(0)$.

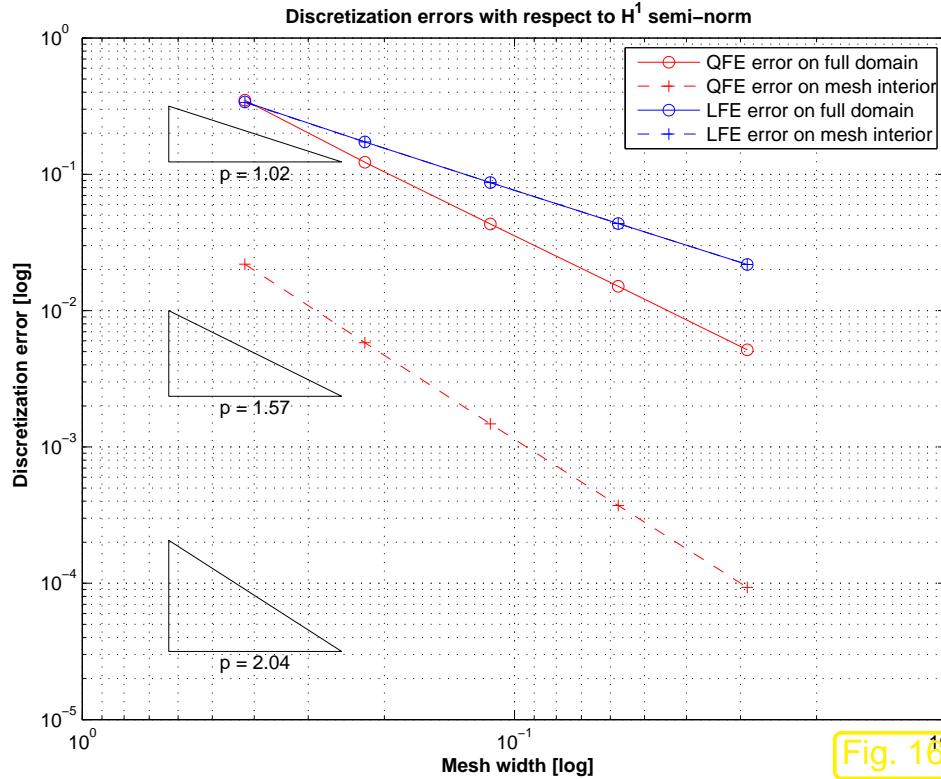


Fig. 167

$H^1(\Omega)$ -norm of discretization error on unit ball ($\text{---} \leftrightarrow p = 1$, $\text{—} \leftrightarrow p = 2$)

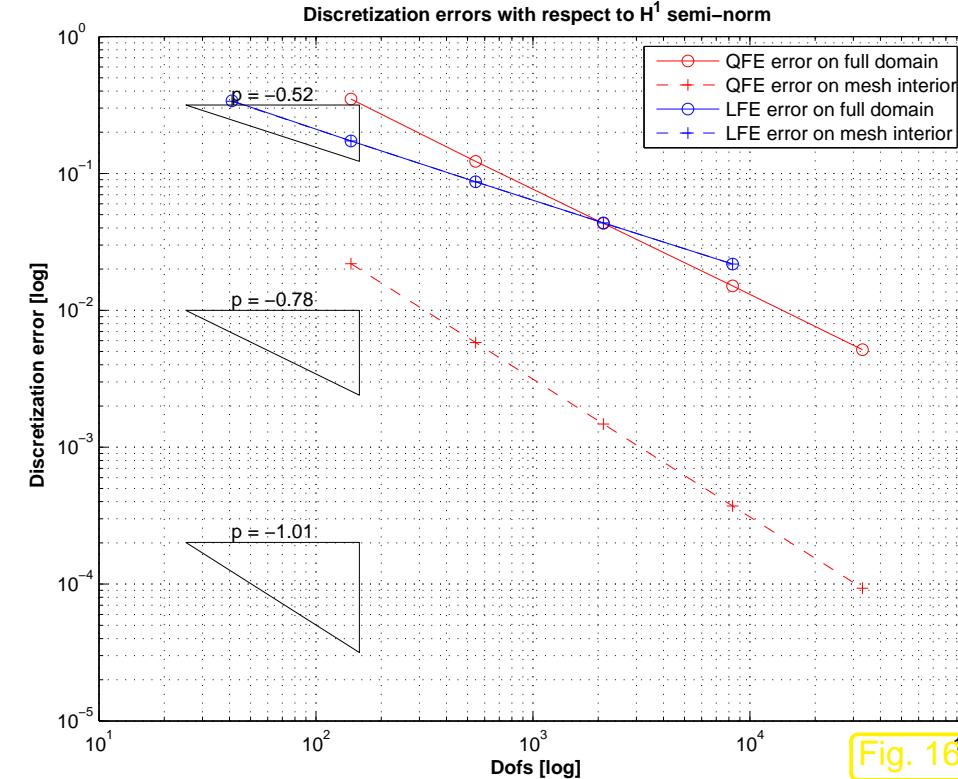


Fig. 168



Theoretical guideline:

If $V_{0,N} = \mathcal{S}_p^0(\mathcal{M})$, use boundary fitting with polynomials of degree p .

5.6 Duality techniques

5.6.1 Linear output functionals

Adopt abstract setting of Sect. 5.1:

linear variational problem (1.4.5) in the form

$$u \in V_0: \quad \mathbf{a}(u, v) = \ell(v) \quad \forall v \in V_0 , \quad (3.1.1)$$

- $V_0 \hat{=} (\text{real})$ vector space, a space of functions $\Omega \mapsto \mathbb{R}$ for scalar 2nd-order elliptic variational problems, usually “energy space” $H^1(\Omega)/H_0^1(\Omega)$, see Sect. 2.2
- $\mathbf{a} : V_0 \times V_0 \mapsto \mathbb{R} \hat{=} \text{a bilinear form}$, see Def. 1.3.11,
- $\ell : V_0 \mapsto \mathbb{R} \hat{=} \text{a linear form}$, see Def. 1.3.11,

- Assumptions 5.1.1, 5.1.2, 5.1.3 are supposed to hold \Rightarrow existence, uniqueness, and stability of solution u by Thm. 5.1.4.

(Examples of 2nd-order linear BVPs discussed in Rem. 5.1.5, Sect. 2.8)

Galerkin discretization using $V_{0,N} \subset V_0$ \Rightarrow discrete variational problem

$$u_N \in V_{0,N}: \quad a(u_N, v_N) = f(v_N) \quad \forall v_N \in V_{0,N}. \quad (3.1.3)$$

New twist: we are interested mainly/only in the *number* $F(u)$, where

$$F : V_0 \mapsto \mathbb{R}$$

is an **output functional**.

Mathematical terminology: **functional** $\hat{=}$ mapping from a function space into \mathbb{R}

Example 5.6.1 (Output functionals).

Some output functionals for solutions of PDEs commonly encountered in applications:

- mean values, see Ex. 5.6.4 below
- total heat flux through a surface (for heat conduction model → Sect. 2.5), see Ex. 5.6.13 below
- total surface charge of a conducting body (for electrostatics → Sect. 2.1.2)
- total heat production (Ohmic losses) by stationary currents
- total force on a charged conductor (for electrostatics → Sect. 2.1.2)
- lift and drag in computational fluid dynamics (aircraft simulation)
- and many more . . .



We consider output functionals with special properties, which are rather common in practice:

Assumption 5.6.2 (Linearity of output functional).

*The output functional F is a **linear** form (→ Def. 1.3.11) on V_0*

To put the next assumption into context, please recall Ass. 5.1.2 and Rem. 2.3.11.

Assumption 5.6.3 (Continuity of output functional).

*The output functional is **continuous** w.r.t. the energy norm in the sense that*

$$\exists C_f > 0: \quad |F(v)| \leq C_f \|v\|_a \quad \forall v \in V_0 .$$

Now consider Galerkin discretization of (3.1.1) based on Galerkin trial/test space $V_{0,N} \subset V_0$, $N := \dim V_{0,N} < \infty \Rightarrow$ discrete variational problem

$$u_N \in V_{0,N}: \quad a(u_N, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N} . \quad (3.1.3)$$

What would you dare to sell as an approximation of $F(u)$? Of course, . . .

Galerkin solution $u_N \in V_{0,N}$ \rightarrow approximate output value $F(u_N)$

How accurate is $F(u_N)$, that is, how big is the output error $|F(u) - F(u_N)|$?

Linearity (\rightarrow Ass. 5.6.2) and continuity Ass. 5.6.3 conspire to furnish a very simple estimate

$$|F(u) - F(u_N)| \leq C_f \|u - u_N\|_a .$$

► A priori estimates for $\|u - u_N\|_a \Rightarrow$ estimates for $|F(u) - F(u_N)|$

Hence, Thm. 5.3.42 immediately tells us the asymptotic convergence of linear and continuous output functionals defined for solutions of 2nd-order scalar elliptic BVPs and Lagrangian finite element discretization.

Example 5.6.4 (Approximation of mean temperature).

Heat conduction model (\rightarrow Sect. 2.5), scaled heat conductivity $\kappa \equiv 1$, on domain $\Omega =]0, 1[^2$, fixed temperature $u = 0$ on $\partial\Omega$:

$$-\Delta u = f \quad \text{in } \Omega , \quad u = 0 \quad \text{on } \partial\Omega .$$

$$f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y) \quad (x, y)^T \in \Omega$$

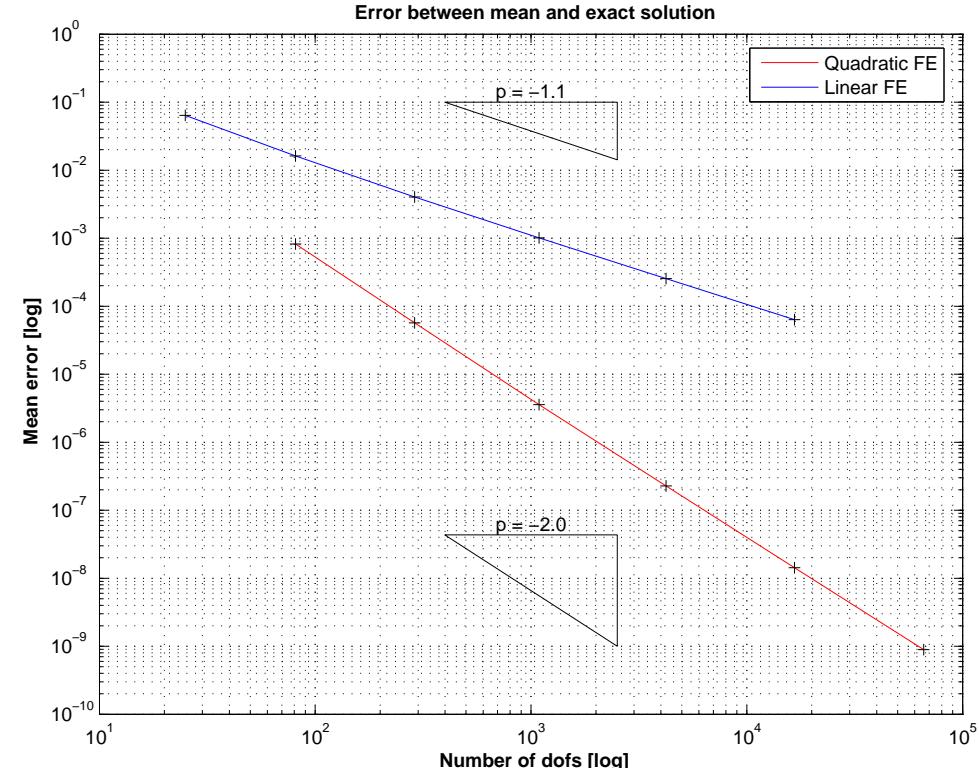
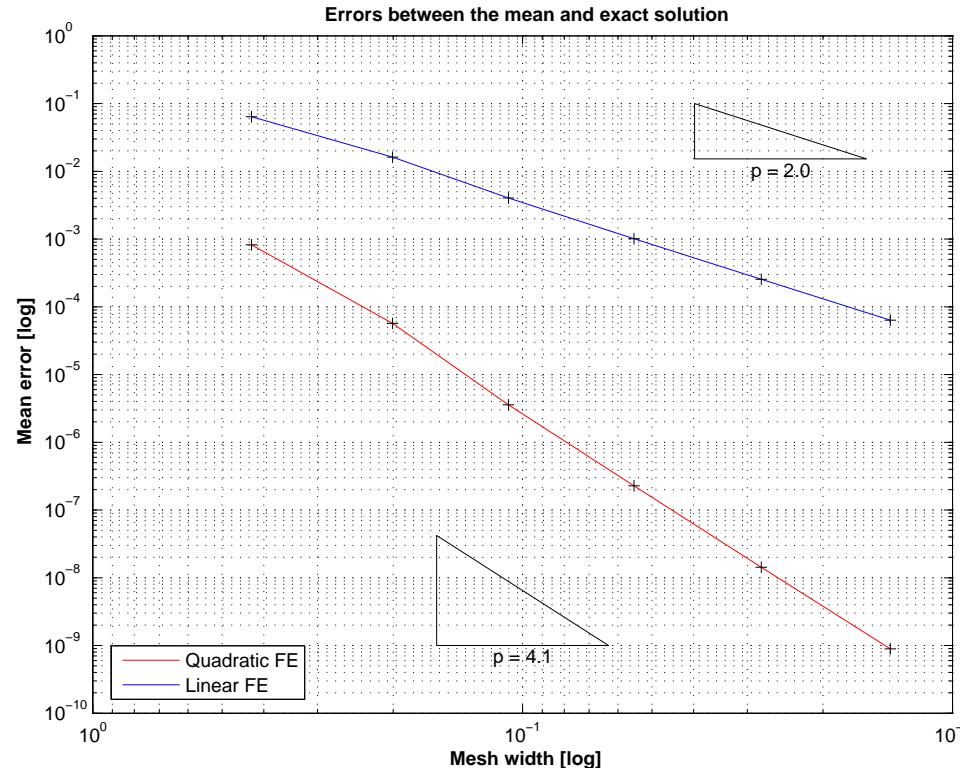
$$u(x, y) = \sin(\pi x) \sin(\pi y)$$

mean temperature $F(u) = \frac{1}{|\Omega|} \int_{\Omega} u \, d\boldsymbol{x}$.

Details of finite element Galerkin discretization:

- Sequence of triangular meshes \mathcal{M} created by regular refinement.
- Galerkin discretization: $V_{0,N} := \mathcal{S}_{1,0}^0(\mathcal{M})$ (linear Lagrangian finite elements → Sect. 3.2).
- Quadrature rule (3.5.38) of order 6 for assembly of right hand side vector
(more than sufficiently accurate → guidelines from Sect. 5.5.1)

Expected: algebraic convergence in $h_{\mathcal{M}}$ with rate 1 of approximate mean temperature



Error in mean value on unit square ($\textcolor{blue}{-} \leftrightarrow p = 1$, $\textcolor{red}{-} \leftrightarrow p = 2$)

Observation: Mean value converges twice as fast as expected: algebraic convergence $O(h_M^2)$!



Theorem 5.6.5 (Duality estimate for linear functional output).

Define the **dual solution** $g_F \in V_0$ to F as solution of

$$g_F \in V_0: \quad \mathbf{a}(v, g_F) = F(v) \quad \forall v \in V_0 .$$

Then

$$|F(u) - F(u_N)| \leq \|u - u_N\|_{\mathbf{a}} \inf_{v_N \in V_{0,N}} \|g_F - v_N\|_{\mathbf{a}} . \quad (5.6.6)$$

Proof. For **any** $v_N \in V_{0,N}$:

$$F(u) - F(u_N) = \mathbf{a}(u - u_N, g_F) \stackrel{(*)}{=} \mathbf{a}(u - u_N, g_F - v_N) \leq \|u - u_N\|_{\mathbf{a}} \|g_F - v_N\|_{\mathbf{a}} .$$

$(*) \leftarrow$ by **Galerkin orthogonality** (5.1.7). □

If g_F can be approximated well in $V_{0,N}$, then the **output error** can converge $\rightarrow 0$ (much) faster than $\|u - u_N\|_{\mathbf{a}}$.

Example 5.6.7 (Approximation of mean temperature cont'd). \rightarrow Ex. 5.6.4

- The mean temperature functional (5.6.6) is obviously linear \rightarrow Ass. 5.6.2
- By the Cauchy-Schwarz inequality (2.2.15) it clearly satisfies Ass. 5.6.3 even with $\|\cdot\|_a = \|\cdot\|_{L^2(\Omega)}$, let alone for $\|\cdot\|_a = |\cdot|_{H^1(\Omega)}$ on $H_0^1(\Omega)$.

What is $g_F \in H_0^1(\Omega)$ in this case? By Thm. 5.6.5 it is the solution of the variational problem

$$\int_{\Omega} \mathbf{grad} g_F \cdot \mathbf{grad} v \, dx = F(v) = \frac{1}{|\Omega|} \int_{\Omega} v \, dx \quad \forall v \in H_0^1(\Omega) .$$

The associated 2nd-order BVP reads

$$-\Delta g_F = \frac{1}{|\Omega|} \quad \text{in } \Omega, \quad g_F = 0 \quad \text{on } \partial\Omega .$$

Now recall the elliptic lifting theory Thm. 5.4.10 for convex domains: since $\Omega =]0, 1[^2$ is convex, we conclude $g_F \in H^2(\Omega)$.

► By interpolation estimate of Thm. 5.3.27 ($I_1 \doteq$ linear interpolation onto $S_1^0(\mathcal{M})$)

$$\inf_{v_N \in S_1^0(\mathcal{M})} |g_F - v_N|_{H^1(\Omega)} \leq |g_F - I_1 g_F|_{H^1(\Omega)} \leq C h_{\mathcal{M}} |g_F|_{H^2(\Omega)} ,$$

5.6

where $C > 0$ may depend on Ω and the shape regularity measure (\rightarrow Def. 5.3.26) of \mathcal{M} .

Plug this into the **duality estimate** (5.6.6) of Thm. 5.6.5 and note that $u \in H^2(\Omega)$ by virtue of Thm. 5.4.10 and $f \in L^2(\Omega)$:

$$\Rightarrow |F(u) - F(u_N)| \leq Ch_{\mathcal{M}} \cdot \underbrace{|u - u_N|_{H^1(\Omega)}}_{\leq Ch_{\mathcal{M}} \text{ if } u \in H^2(\Omega)} \leq Ch_{\mathcal{M}}^2 ,$$

where the “generic constant” $C > 0$ depends only on $\Omega, u, \rho_{\mathcal{M}}$.

Again, by the elliptic lifting theory Thm. 5.4.10 we infer that $u \in H^2(\Omega)$ holds for this example since $f \in L^2(\Omega)$.



5.6.2 Case study: Boundary flux computation

Model problem (process engineering):

Long pipe carrying turbulent flow of coolant (water)

$\Omega \subset \mathbb{R}^2$: cross-section of pipe

κ : (scaled) heat conductivity of pipe material (assumed homogeneous, $\kappa = \text{const}$)

Assumption: Constant temperatures u_o , u_i at outer/inner wall Γ_o , Γ_i of pipe

Task: Compute heat flow pipe \rightarrow water

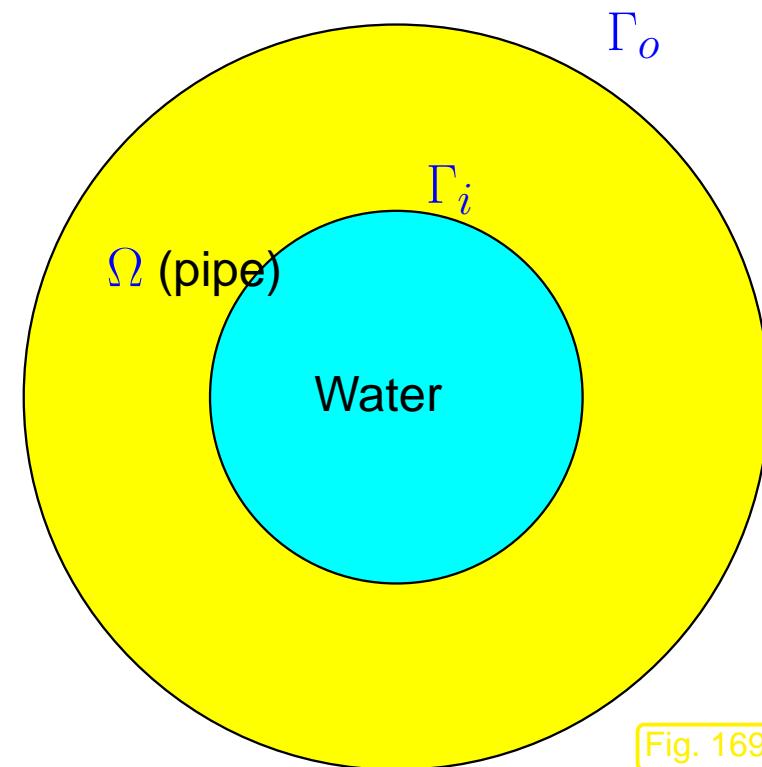


Fig. 169

Mathematical model: elliptic boundary value for stationary heat conduction (\rightarrow Sect. 2.5)

$$-\operatorname{div}(\kappa \operatorname{grad} u) = 0 \quad \text{in } \Omega, \quad u = u_x \quad \text{on } \Gamma_x, x \in \{i, o\}. \quad (5.6.8)$$

Heat flux through Γ_i : $J(u) := \int_{\Gamma_i} \kappa \operatorname{grad} u \cdot \mathbf{n} \, dS.$ (5.6.9)

Relate to abstract framework:

$$(5.6.8) \cong (3.1.1), \quad V_0 \cong H_0^1(\Omega) \quad (\rightarrow \text{Sect. 2.8})$$

(Actually, $u \in H^1(\Omega)$, but by means of offset functions we can switch to the variational space $H_0^1(\Omega)$, see Sects. 2.1.3, 3.5.5.)

Numerical method: finite element computation of heat conduction in pipe
(e.g. linear Lagrangian finite element Galerkin discretization, Sect. 3.2)

Expectation: Algebraic convergence $|J(u) - J(u_N)| = O(h_{\mathcal{M}}^2)$ for regular h -refinement

This expectation is based on the analogy to Ex. 5.6.4 (Approximation of mean temperature), where duality estimates yielded $O(h_{\mathcal{M}}^2)$ convergence of the mean temperature error in the case of Galerkin discretization by means of linear Lagrangian finite elements on a sequence of meshes obtained by regular refinement. Now, it seems, we can follow the same reasoning.

Example 5.6.10 (Computation of heat flux).

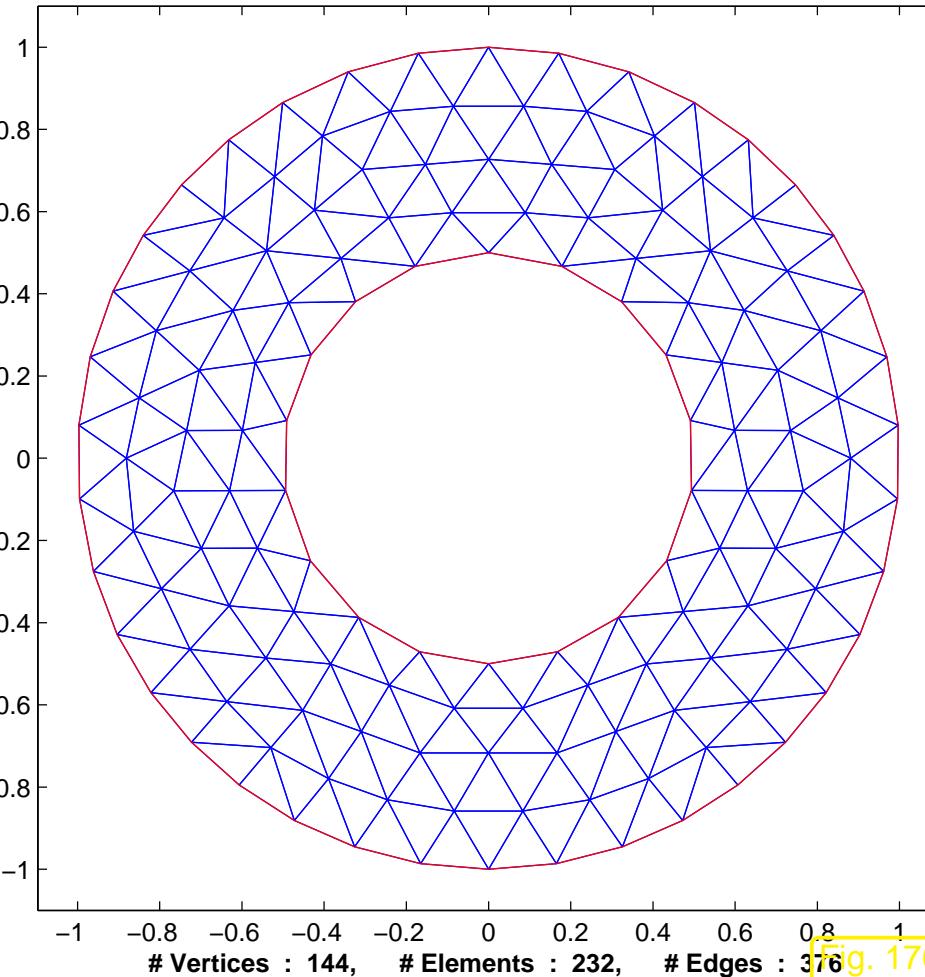
- Setting: model problem “heat flux pipe *to* water”, see (5.6.8) and Fig. 169.
- Linear output functional from (5.6.9)
- Domain $\Omega = B_{R_o}(0) \setminus B_{R_i}(0) := \{\mathbf{x} \in \mathbb{R}^2 : R_i < |\mathbf{x}| < R_o\}$ with $R_o = 1$ and $R_i = 1/2$
- Dirichlet boundary data $u_i = 60^\circ\text{C}$ on Γ_i , $u_o = 10^\circ\text{C}$ on Γ_o , heat source $f \equiv 0$, heat conductivity $\kappa \equiv 1$.
 - Exact solution: $u(r, \varphi) = C_1 \ln(r) + C_2$, with $C_1 := (u_o - u_i)/(\ln R_i - \ln R_o)$,
 - Exact heat flux: $J = 2\pi\kappa C_1$, $C_2 := (\ln R_o u_i - \ln R_i u_o)/(\ln R_i - \ln R_o)$.

Details of linear Lagrangian finite element Galerkin discretization:

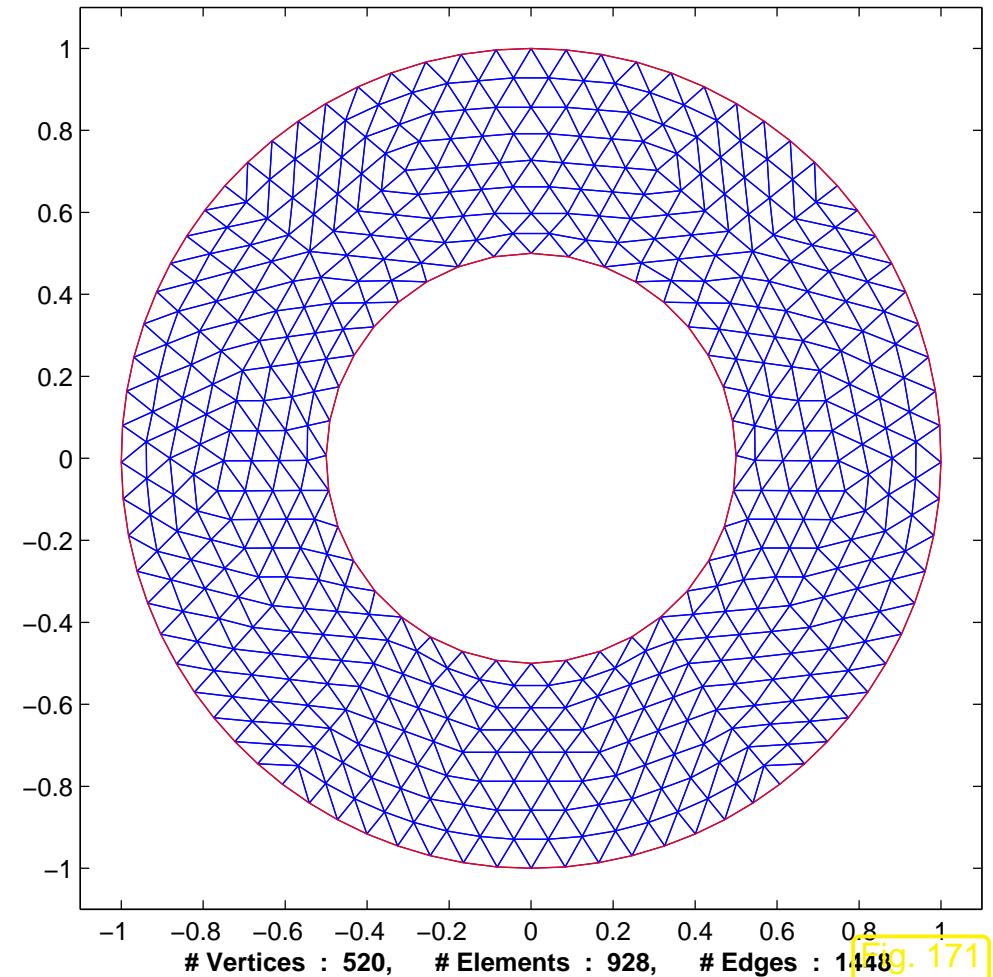
- Sequences of unstructured triangular meshes \mathcal{M} obtained by regular refinement of coarse mesh (from grid generator).
- Galerkin FE discretization based on $V_{0,N} := \mathcal{S}_{1,0}^0(\mathcal{M})$.
- Approximate evaluation of $a(u_N, v_N)$, $f(v_N)$ by six point quadrature rule (3.5.38) (“overkill quadrature”, see Sect. 5.5.1)

- Approximate evaluation of $J(u_N)$ by 4 point Gauss-Legendre quadrature rule on boundary edges of \mathcal{M} .
- Linear boundary approximation (circle replaced by polygon).
- Recorded: errors $|J - J(u_N)|$ on sequence of meshes.

2D triangular mesh



2D triangular mesh



Unstructured triangular meshes for $\Omega = B_1(0) \setminus B_{1/2}(0)$ (two coarsest specimens).

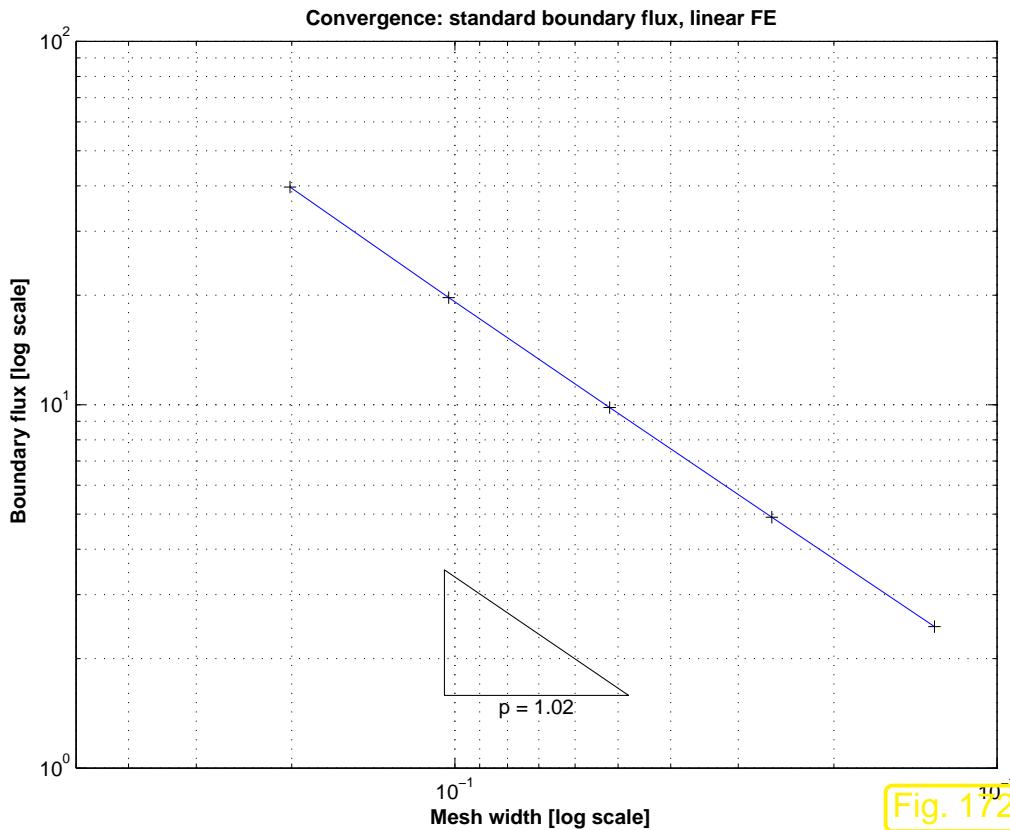


Fig. 172

Observation:

Algebraic convergence of output error for J from (5.6.9) *only with rate 1* (in mesh width h_M)!

(This is not the fault of the piecewise linear boundary approximation, which is sufficient when using piecewise linear Lagrangian finite elements, see Sect. 5.5.2.)



Why was our expectation mistaken ?

Suspicion: the output functional J fails to meet requirements of duality estimates of Thm. 5.6.5:



boundary flux functional J from (5.6.9) is **not** continuous on $H^1(\Omega)$!

Example 5.6.11 (Non-continuity of boundary flux functional).

Idea: find $u \in H^1(\Omega)$, for which “ $J(u) = \infty$ ”, cf. non-continuity of point evaluation functional on $H^1(\Omega) \rightarrow \text{Rem. 2.3.16}$.

On $\Omega =]0, 1[^2$ consider $u(\mathbf{x}) = x_1^\alpha$, $\frac{1}{2} < \alpha < 1$, and boundary flux functional for left side of square

$$J_0(v) = \int_0^1 \frac{\partial v}{\partial x_1}(0, x_2) dx_2 .$$

Straightforward computations of improper integral:

$$\begin{aligned} |u|_{H^1(\Omega)}^2 &= \int_{\Omega} \|\mathbf{grad} u(\mathbf{x})\|^2 d\mathbf{x} = \int_0^1 \int_0^1 \alpha^2 x_1^{2\alpha-2} dx_1 dx_2 = \frac{\alpha^2}{2\alpha - 1} < \infty . \\ &\stackrel{\text{Def. 2.2.12}}{\implies} u \in H^1(\Omega) . \end{aligned}$$

On the other hand

$$\text{“} \frac{\partial u}{\partial x_1}(0, x_2) = \infty \text{”} \Rightarrow J_0(u) = \infty .$$



Ex. 5.6.11 ➤ Thm. 5.6.5 cannot be applied



(Potentially) poor convergence of flux obtained from straightforward evaluation of
 $J(u_N)$ for FE solution $u_N \in \mathcal{S}_{1,0}^0(\mathcal{M})!$

Apparently there is no remedy, because the boundary flux functional (5.6.9) seems to be enforced on us by the problem: we are not allowed to tinker with it, are we?

Trick:

use fixed **cut-off function** $\psi \in C^0(\bar{\Omega}) \cap H^1(\Omega)$, $\psi \equiv 1$ on Γ_i , $\psi|_{\Gamma_o} = 0$

$$\int_{\Gamma_i} \kappa \operatorname{grad} u \cdot \mathbf{n} dS = \int_{\Gamma_i} (\kappa \operatorname{grad} u \cdot \mathbf{n}) \psi dS = \int_{\Omega} \underbrace{\operatorname{div}(\kappa \operatorname{grad} u)}_{=0} \psi + \kappa \operatorname{grad} u \cdot \operatorname{grad} \psi dx$$

► use $J^*(u) := \int_{\Omega} \kappa \operatorname{grad} u \cdot \operatorname{grad} \psi dx$. (5.6.12)

Obviously (*): $J^* : H^1(\Omega) \mapsto \mathbb{R}$ continuous & $J^*(u) = J(u)$ for solution of (5.6.8)

(*): By the Cauchy-Schwarz inequality (2.2.15), since $\kappa = \text{const}$,

$$|J^*(u)| \leq \kappa \|\operatorname{grad} u\|_{L^2(\Omega)} \|\operatorname{grad} \psi\|_{L^2(\Omega)} \leq C|u|_{H^1(\Omega)},$$

with $C := \kappa \|\operatorname{grad} \psi\|_{L^2(\Omega)}$, which is a constant independent of u , as ψ is a fixed function.

Objection: You cannot just tamper with the output functional of a problem just because you do not like it!

Retort: Of course, one can replace the output function J with another one J^* as long as

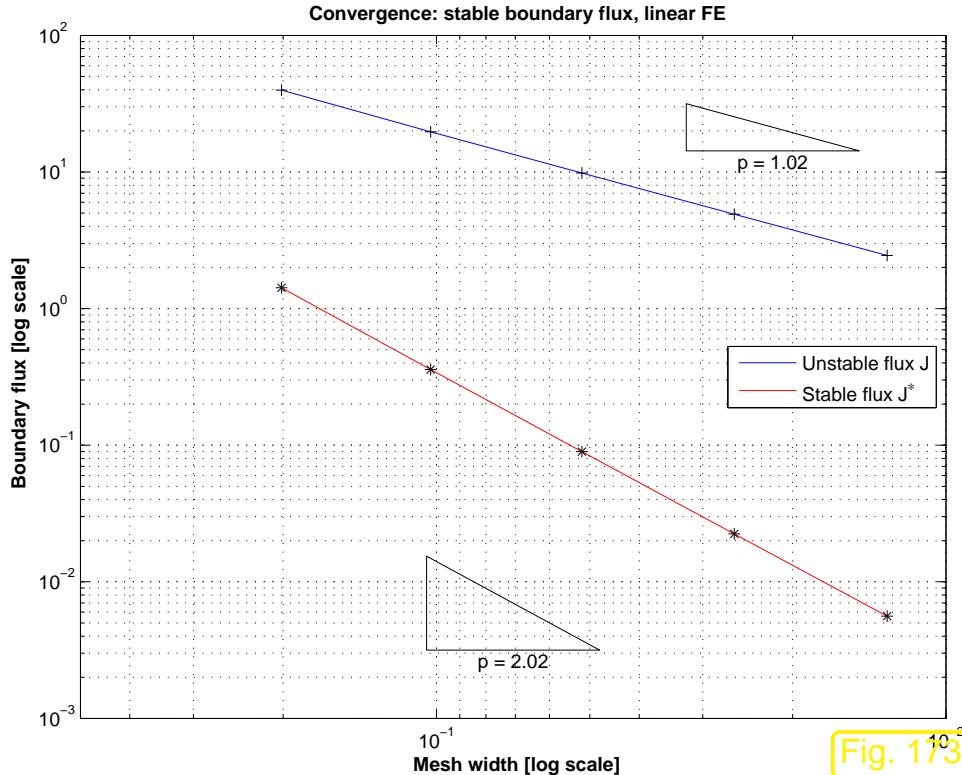
$$J(u) = J^*(u) \quad \text{for the exact solution } u \text{ of the BVP,}$$

because the objective is not to “evaluate J ”, but to obtain an approximation for $J(u)$!

Example 5.6.13 (Computation of heat flux cnt'd). → Ex. 5.6.13

Further details on flux evaluation:

- Galerkin FE discretization based on $V_{0,N} := \mathcal{S}_{1,0}^0(\mathcal{M})$ or $V_{0,N} := \mathcal{S}_{2,0}^0(\mathcal{M})$.
- Approximate evaluation of $J^*(u_N)$ by six point quadrature rule (3.5.38) (“overkill quadrature”, see Sect. 5.5.1)
- Cut-off function with linear decay in radial direction
- Recorded: errors $|J - J(u_N)|$ and $|J - J^*(u_N)|$.



▷ Convergence of $|J(u) - J(u_N)|$ and $|J(u) - J^*(u_N)|$ for linear Lagrangian finite element discretization.

Fig. 173

Additional observations:

- Algebraic convergence $|J(u) - J^*(u_N)| = O(h_{\mathcal{M}}^2)$ (rate 2 !) for alternative output functional J^* from (5.6.12).
- Dramatically reduced output error!



Remark 5.6.14 (Finding continuous replacement functionals).

Now you will ask: How can we find good (continuous) replacement functionals, if we are confronted with an unbounded output functional on the energy space?

Unfortunately, there is *no recipe*, and sometimes it does not seem to be possible to find a suitable J^* at all, for instance in the case of point evaluation, *cf.* Rem. 2.3.16.

Good news: another opportunity to show off how smart you are!



5.6.3 L^2 -estimates

So far we have only studied the energy norm ($\hookrightarrow H^1(\Omega)$ -norm, see Rem. 5.3.31) of the finite element discretization error for 2nd-order elliptic BVP.

The reason was the handy tool of Cea's lemma Thm. 5.1.10.

What about error estimates in other “relevant norms”, e.g.,

- in the mean square norm or $L^2(\Omega)$ -norm, see Def. 2.2.5,
- in the supremum norm or $L^\infty(\Omega)$ -norm, see Def. 1.6.4?

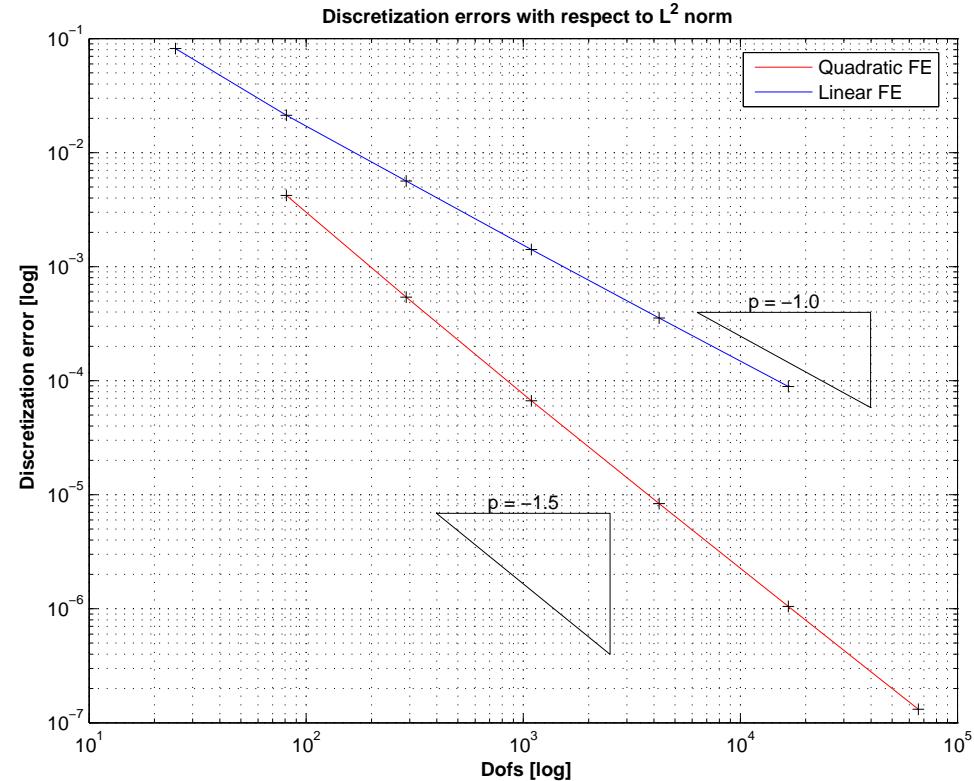
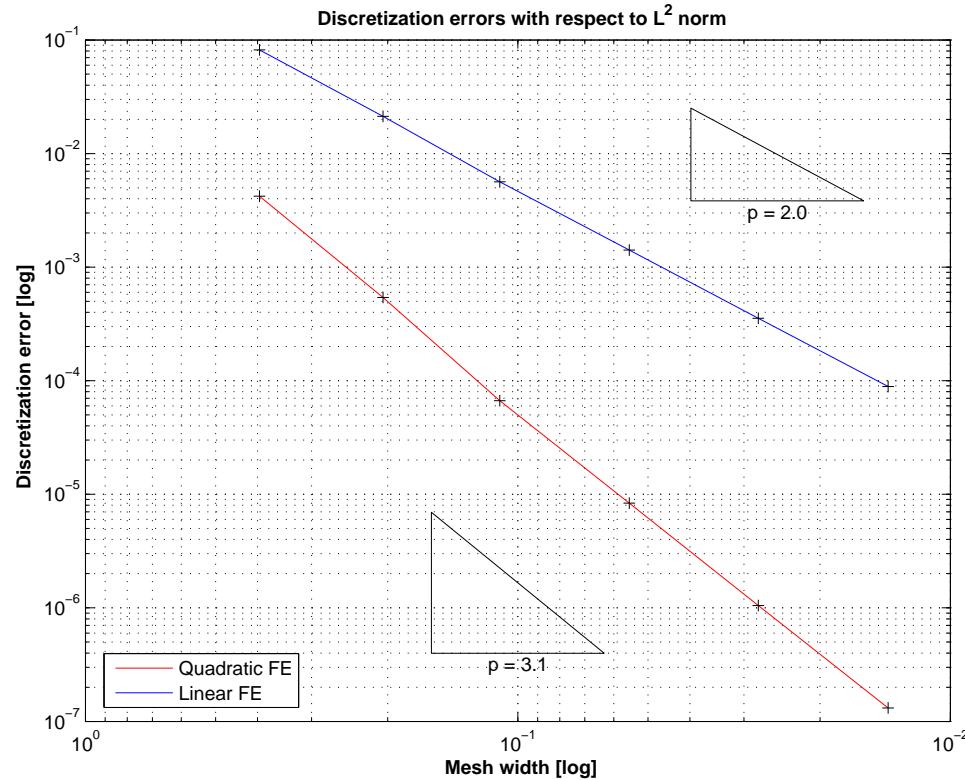
In this section we tackle $\|u - u_N\|_{L^2(\Omega)}$. We largely reuse the abstract framework of Sect. 5.6.1: linear variational problem (3.1.1) with exact solution $u \in V_0$, Galerkin finite element solution $u_N \in V_{0,N}$, see p. 547, and the special framework of linear 2nd-order elliptic BVPs, see Rem. 5.1.5: concretely,

$$\mathbf{a}(u, v) := \int_{\Omega} \kappa(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\mathbf{x}, \quad u, v \in H_0^1(\Omega).$$

Example 5.6.15 (L^2 -convergence of FE solutions). → Ex. 5.2.4

Setting: $\Omega = [0, 1]^2$, $D \equiv 1$, $f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y)$, $(x, y)^\top \in \Omega$
➤ $u(x, y) = \sin(\pi x) \sin(\pi y)$.

- Sequence of triangular meshes \mathcal{M} , created by regular refinement.
- FE Galerkin discretization based on $\mathcal{S}_{1,0}^0(\mathcal{M})$ or $\mathcal{S}_2^0(\mathcal{M})$.
- Quadrature rule (3.5.38) for assembly of local load vectors (\rightarrow Sect. ??).
- Approximate $L^2(\Omega)$ -norm by means of quadrature rule (3.5.38).



$L^2(\Omega)$ -norm of discretization error on unit square ($\textcolor{blue}{-} \leftrightarrow p = 1$, $\textcolor{red}{-} \leftrightarrow p = 2$)

- Observations:
- Linear Lagrangian FE ($p = 1$) $\Rightarrow \|u - u_N\|_0 = O(N^{-1})$
 - Quadratic Lagrangian FE ($p = 2$) $\Rightarrow \|u - u_N\|_0 = O(N^{-1.5})$

Remark 5.6.16 (L^2 interpolation error).

Recall the interpolation error estimate of Thm. 5.3.27

$$\|u - I_1 u\|_{L^2(\Omega)} = O(h_M^2) \quad \text{vs.} \quad |u - I_1 u|_{H^1(\Omega)} = O(h_M) ,$$

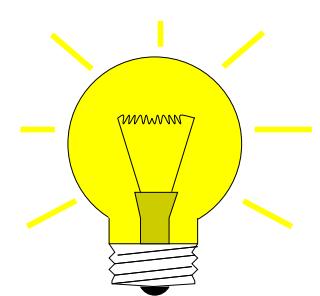
on a family of meshes with uniformly bounded shape regularity measure.

- ☞ Higher rate of algebraic convergence of the interpolation error when measured in the **weaker** $L^2(\Omega)$ -norm compared to the **stronger** $H^1(\Omega)$ -norm.

Therefore a similar observation in the case of the finite element approximation error is not so surprising.



Now we supply a rigorous underpinning and explanation of the behavior of $\|u - u_N\|_{L^2(\Omega)}$ that we have observed and expect.



Idea: Consider special **continuous linear output functional**

$$J(v) := \int_{\Omega} v \cdot (u - u_N) \, d\mathbf{x} \quad !$$

This functional is highly relevant for L^2 -estimates, because

$$F(u) - F(u_N) = \|u - u_N\|_{L^2(\Omega)}^2 \quad !$$

➤ estimates for the output error will provide bounds for $\|u - u_N\|_{L^2(\Omega)}$!

Note: Both u and u_N are *fixed* functions $\in H^1(\Omega)$!

- Linearity of J (\rightarrow Ass. 5.6.2) is obvious.
- Continuity $J : H_0^1(\Omega) \mapsto \mathbb{R}$ (\rightarrow Ass. 5.6.3) is clear, use Cauchy-Schwarz inequality (2.2.15).

Duality estimate of Thm. 5.6.5 can be applied:

5.6

Thm. 5.6.5

$$\blacktriangleright \quad F(u) - F(u_N) = \|u - u_N\|_{L^2(\Omega)}^2 \leq C|u - u_N|_{H^1(\Omega)} \inf_{v_N \in V_{0,N}} |g_F - v_N|_{H^1(\Omega)}, \quad (5.6.17)$$

where $C > 0$ may depend only on κ , and the **dual solution** $g_F \in H_0^1(\Omega)$ satisfies

$$\begin{aligned} \mathbf{a}(g_F, v) = F(v) \quad \forall v \in V_0 \iff & \int_{\Omega} \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} g_F \cdot \mathbf{grad} v \, d\mathbf{x} = \int_{\Omega} v(u - u_N) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega) \\ & \Downarrow \\ -\operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} g_F) = u - u_N \quad \text{in } \Omega \quad , \quad g_F = 0 \quad \text{on } \partial\Omega . \end{aligned} \quad (5.6.18)$$

Assumption 5.6.19 (2-regularity of homogeneous Dirichlet problem).

We assume that the homogeneous Dirichlet problem with coefficient κ is **2-regular** on Ω : There is $C > 0$, which depends on Ω only such that

$$\begin{aligned} u \in H_0^1(\Omega) \\ \operatorname{div}(\kappa(\mathbf{x}) \operatorname{grad} u) \in L^2(\Omega) \end{aligned} \Rightarrow u \in H^2(\Omega) \quad \text{and} \quad |u|_{H^2(\Omega)} \leq C \|\operatorname{div}(\kappa(\mathbf{x}) \operatorname{grad} u)\|_{L^2(\Omega)}.$$

By the elliptic lifting theorem for convex domains Thm. 5.4.10 we know

$$\Omega \text{ convex} \implies \text{Ass. 5.6.19 is satisfied.}$$

Ass. 5.6.19 in conjunction with (5.6.18) yields

$$|g_F|_{H^2(\Omega)} \leq C \|u - u_N\|_{L^2(\Omega)}, \quad (5.6.20)$$

where $C > 0$ depends only on Ω .

Now we can appeal to the general best approximation theorem for Lagrangian finite element spaces Thm. 5.3.42:

$$\inf_{v_N \in S_p^0(\mathcal{M})} |g_F - v_N|_{H^1(\Omega)} \leq C \frac{h_{\mathcal{M}}}{p} |g_F|_{H^2(\Omega)} \stackrel{(5.6.20)}{\leq} C \frac{h_{\mathcal{M}}}{p} \|u - u_N\|_{L^2(\Omega)}, \quad (5.6.21)$$

where the “generic constants” $C > 0$ depend only on Ω and the shape regularity measure $\rho_{\mathcal{M}}$ (\rightarrow Def. 5.3.26).

Combine (5.6.17) and (5.6.21) and cancel one power of $\|u - u_N\|_{L^2(\Omega)}$:

With $C > 0$ depending only on Ω , κ , and the shape regularity measure $\rho_{\mathcal{M}}$ we conclude

$$\text{Ass. 5.6.19} \Rightarrow \|u - u_N\|_{L^2(\Omega)} \leq C \frac{h_{\mathcal{M}}}{p} \|u - u_N\|_{H^1(\Omega)}.$$

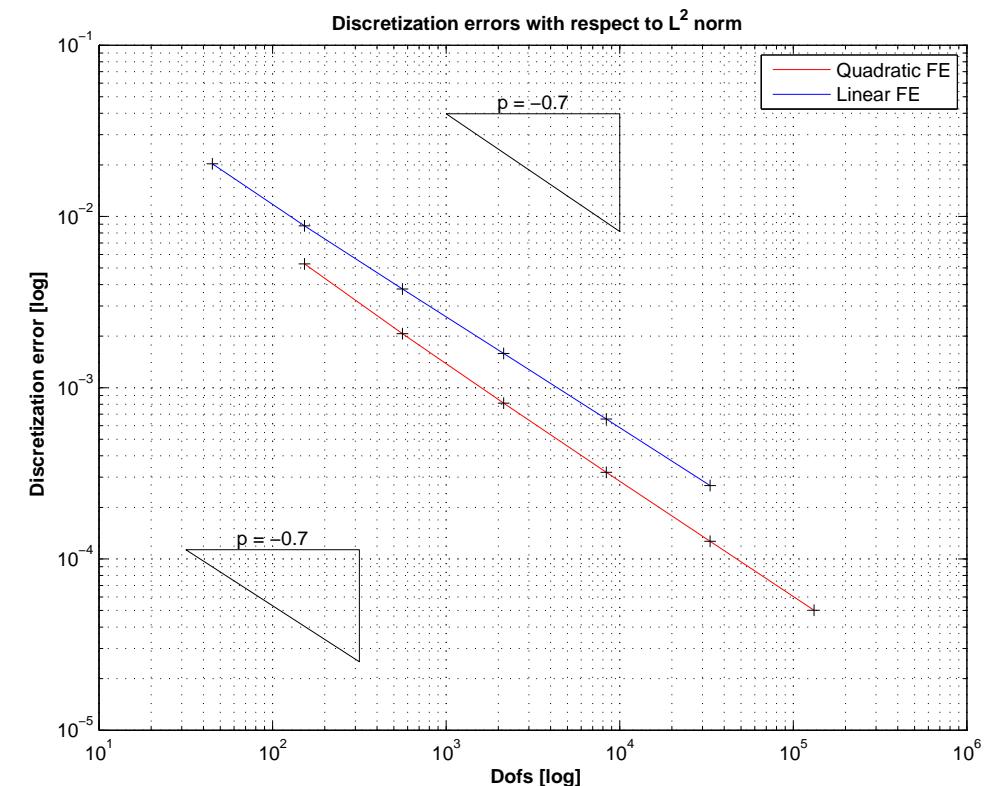
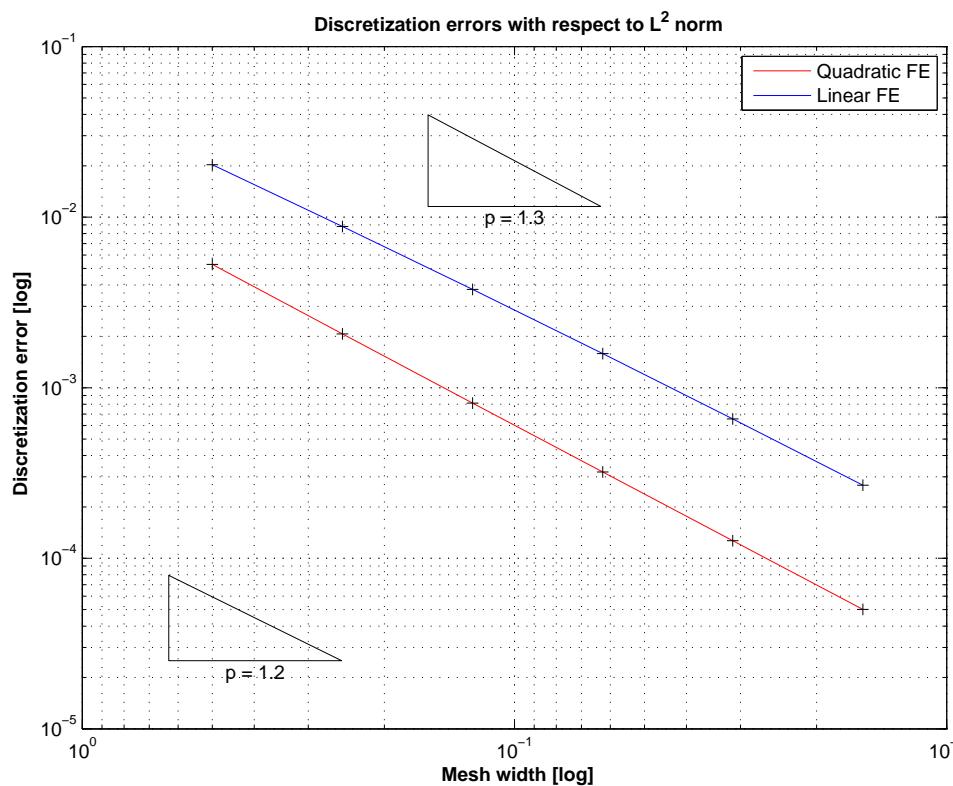
for h -refinement: gain of one factor $O(h_{\mathcal{M}})$ (vs. $H^1(\Omega)$ -estimates)

Is it important to assume 2-regularity, Ass. 5.6.19 or merely a technical requirement of the theoretical approach?

Example 5.6.22 (L^2 -estimates on non-convex domain). cf. Ex. 5.2.6

Setting: $\Omega =]-1, 1[^2 \setminus (]0, 1[\times]-1, 0[)$, $D \equiv 1$, $u(r, \varphi) = r^{2/3} \sin(2/3\varphi)$ (polar coordinates)
 ➤ $f = 0$, Dirichlet data $g = u|_{\partial\Omega}$.

Finite element Galerkin discretization and evaluations as in Ex. 5.6.15.



$L^2(\Omega)$ -norm of discretization error on “L-shaped” domain ($\textcolor{blue}{-} \leftrightarrow p = 1$, $\textcolor{red}{-} \leftrightarrow p = 2$)

Observation: For both ($p = 1, 2$) \Rightarrow algebraic convergence $\|u - u_N\|_0 = O(N^{-2/3})$

Comparison with Ex. 5.2.6: for both linear and quadratic Lagrangian FEM

$$\|u - u_N\|_{L^2(\Omega)} = O(N^{-2/3}) \iff \|u - u_N\|_{H^1(\Omega)} = O(N^{-1/3}),$$

that is, we again observe a doubling of the rate of convergence for the weaker norm.

No gain through the use of quadratic FEM, because of limited smoothness of both u and dual solution g_F . For both the gradient will have a singularity at 0 .



5.7 Discrete maximum principle

So far we have investigated the **accuracy** of finite element Galerkin solutions: we studied relevant norms $\|u - u_N\|$ of the discretization error.

Now new perspective:

structure preservation by FEM

To what extent does the finite element solution u_N inherit key structural properties of the solution u of a 2nd-order scalar elliptic BVP?

This issue will be discussed for a special structural property of the solution of the linear 2nd-order elliptic BVP (inhomogeneous Dirichlet problem) in variational form (\rightarrow Sect. 2.8)

$$u \in \tilde{g} + H_0^1(\Omega): \quad a(u, v) := \int_{\Omega} \kappa \operatorname{grad} u \cdot \operatorname{grad} v \, dx = \int_{\Omega} f v \, dx \quad \forall v \in H_0^1(\Omega). \quad (5.7.1)$$

where $\tilde{g} \hat{=} \text{offset function, extension of Dirichlet data } g \in C^0(\partial\Omega)$, see Sect. 2.3.1, (2.3.5),
 $\kappa \hat{=} \text{bounded and uniformly positive definite diffusion coefficient, see (2.5.4).}$

(5.7.1) \longleftrightarrow BVP

$$-\operatorname{div}(\kappa(x) \operatorname{grad} u) = f \quad \text{in } \Omega \quad , \quad u = g \quad \text{on } \partial\Omega .$$

Recall (\rightarrow Sect. 2.5): (5.7.1) models *stationary* temperature distribution in body, when temperature on its surface is prescribed by g .

- Intuition:
- In the absence of heat sources maximal and minimal temperature attained on surface.
 - In the presence of a heat source ($f \geq 0$) the temperature minimum will be attained on surface $\partial\Omega$.
 - If $f \leq 0$ (heat sink), then the maximal temperature will be attained on the surface.

In fact this is a theorem, *cf.* Sect. 2.7.

Theorem 5.7.2 (Maximum principle for 2nd-order elliptic BVP).

For $u \in C^0(\bar{\Omega}) \cap H^1(\Omega)$ holds the **maximum principle**

$$-\operatorname{div}(\kappa(\mathbf{x}) \operatorname{grad} u) \geq 0 \implies \min_{\mathbf{x} \in \partial\Omega} u(\mathbf{x}) = \min_{\mathbf{x} \in \Omega} u(\mathbf{x}),$$

$$-\operatorname{div}(\kappa(\mathbf{x}) \operatorname{grad} u) \leq 0 \implies \max_{\mathbf{x} \in \partial\Omega} u(\mathbf{x}) = \max_{\mathbf{x} \in \Omega} u(\mathbf{x}).$$

$\Delta u = 0$


 Maximum/minimum on $\partial\Omega$

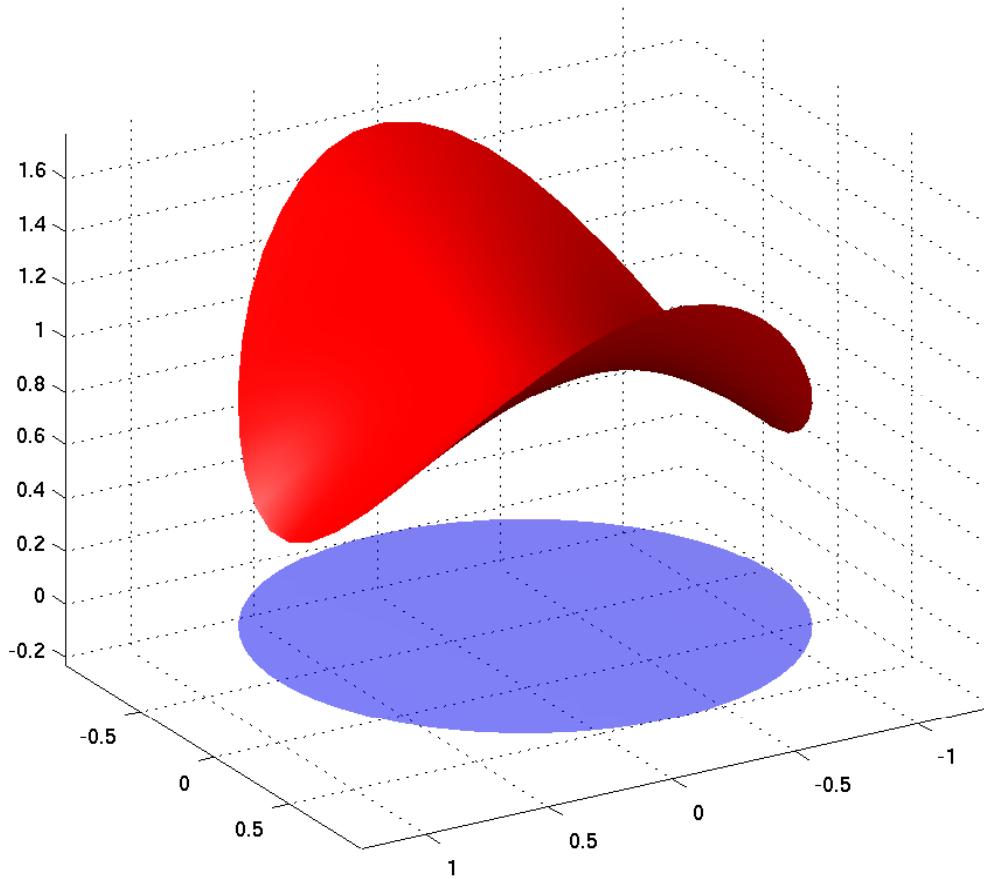


Fig. 174

Proof. (for the case $-\operatorname{div}(\kappa(\mathbf{x}) \operatorname{grad} u) = 0$)

Sect. 2.1.3> u solves quadratic minimization problem

$$u = \underset{\substack{v \in H^1(\Omega) \\ v=g \text{ on } \partial\Omega}}{\operatorname{argmin}} \int_{\Omega} \kappa(\mathbf{x}) \|\operatorname{grad} v(\mathbf{x})\|^2 d\mathbf{x} .$$

If u had a global maximum at \mathbf{x}^* in the interior of Ω , that is

$$\exists \delta > 0: u(\mathbf{x}^*) \geq \max_{\mathbf{x} \in \partial\Omega} u(\mathbf{x}) + \delta.$$

Now “chop off” the maximum and define

$$w(\mathbf{x}) := \min\{u(\mathbf{x}), u(\mathbf{x}^*) - \delta\}, \quad \mathbf{x} \in \Omega.$$

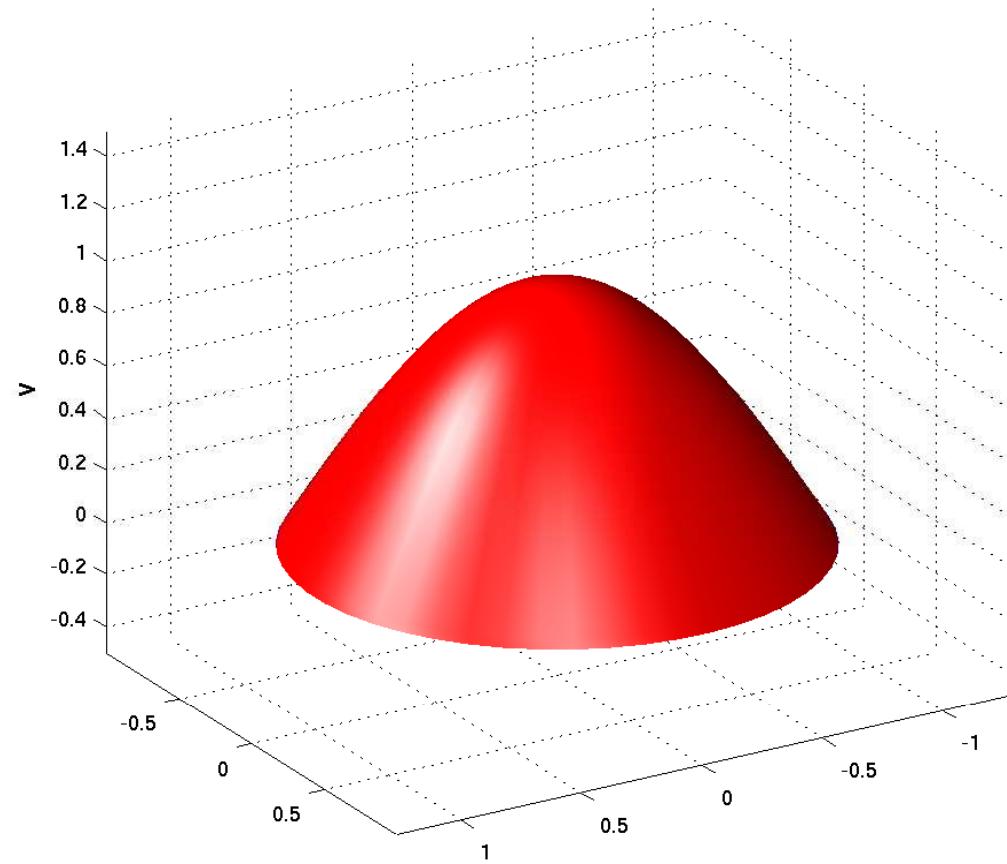


Fig. 175

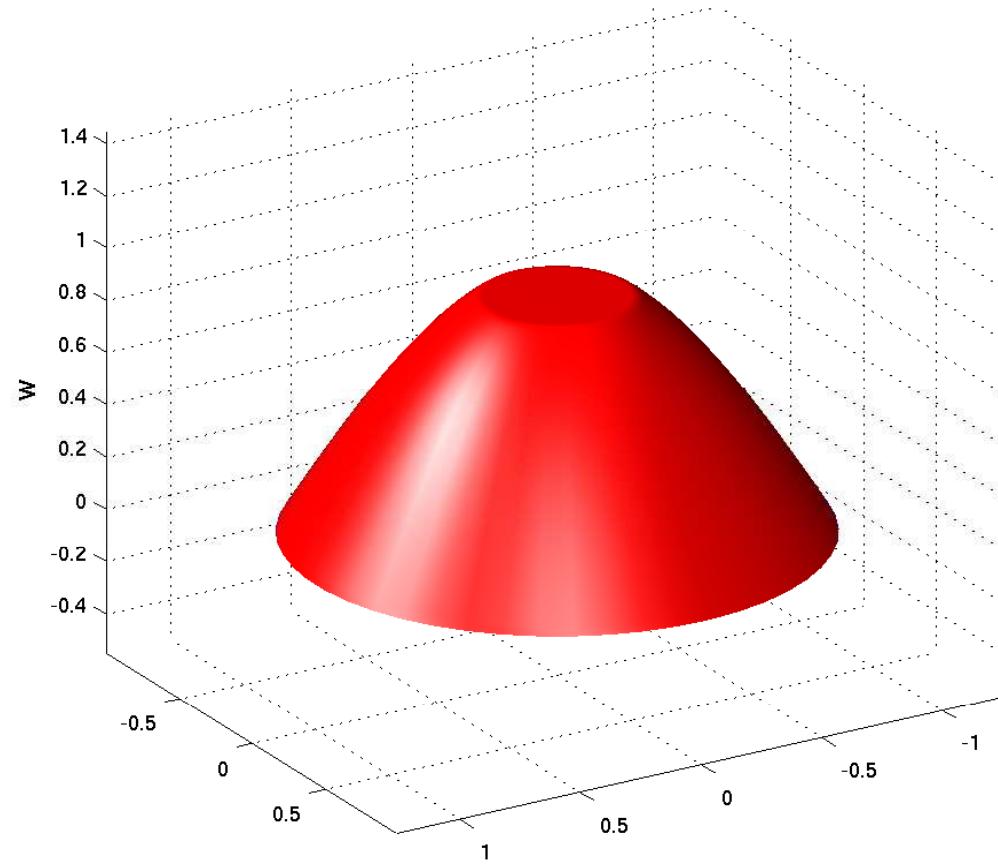


Fig. 176

$$\int_{\Omega} \kappa(\mathbf{x}) \|\operatorname{grad} w(\mathbf{x})\|^2 d\mathbf{x} \geq \int_{\Omega} \kappa(\mathbf{x}) \|\operatorname{grad} v(\mathbf{x})\|^2 d\mathbf{x} .$$

Obviously, $w \in C^0(\bar{\Omega})$, and as a continuous function which is piecewise in H^1 the function w will

also belong to $H^1(\Omega)$. However

$$\int_{\Omega} \kappa(\mathbf{x}) \|\mathbf{grad} w(\mathbf{x})\|^2 d\mathbf{x} < \int_{\Omega} \kappa(\mathbf{x}) \|\mathbf{grad} u(\mathbf{x})\|^2 d\mathbf{x} ,$$

which contradicts the definition of u as the global minimizer of the quadratic energy functional. \square

Now we consider a finite element Galerkin discretization of (5.7.1) by means of linear Lagrangian finite elements (\rightarrow Sect. 3.4), using offset functions supported near $\partial\Omega$ as explained in Sect. 3.5.5.

➤ finite element Galerkin solution $u_N \in \mathcal{S}_1^0(\mathcal{M}) \subset C^0(\overline{\Omega})$

Issue: does u_N satisfy a **maximum principle**, that is, can we conclude

$$\begin{aligned} f \geq 0 &\implies \min_{\mathbf{x} \in \partial\Omega} u_N(\mathbf{x}) = \min_{\mathbf{x} \in \Omega} u_N(\mathbf{x}) , \\ f \leq 0 &\implies \max_{\mathbf{x} \in \partial\Omega} u_N(\mathbf{x}) = \max_{\mathbf{x} \in \Omega} u_N(\mathbf{x}) ? \end{aligned} \tag{5.7.3}$$

Example 5.7.4 (Maximum principle for finite difference discretization).

Recall from Sect. 4.1: finite difference discretization of

$$-\Delta u = 0 \quad \text{in } \Omega := [0, 1]^2 , \quad u = g \quad \text{on } \partial\Omega,$$

on an $M \times M$ tensor product mesh

$$\mathcal{M} := \{[(i-1)h, ih] \times [(j-1)h, jh], 1 \leq i, j \leq M\} , \quad M \in \mathbb{N} .$$

Unknowns in the finite difference method: $\mu_{ij} \approx u((ih, jh)^T)$, $1 \leq i, j \leq M - 1$.

Unknowns are solutions of a linear system of equations, see (4.1.1)

$$\frac{1}{h^2}(4\mu_{i,j} - \mu_{i-1,j} - \mu_{i+1,j} - \mu_{i,j-1} - \mu_{i,j+1}) = 0 , \quad 1 \leq i, j \leq M - 1 , \quad (5.7.5)$$

where values corresponding to points on the boundary are gleaned from g :

$$\mu_{0,j} := g(0, hj) , \quad \mu_{M,j} := g(1, hj) , \quad \mu_{i,0} := g(hi, 0) , \quad \mu_{i,M} := g(hi, 1) , \quad 1 \leq i, j < M .$$

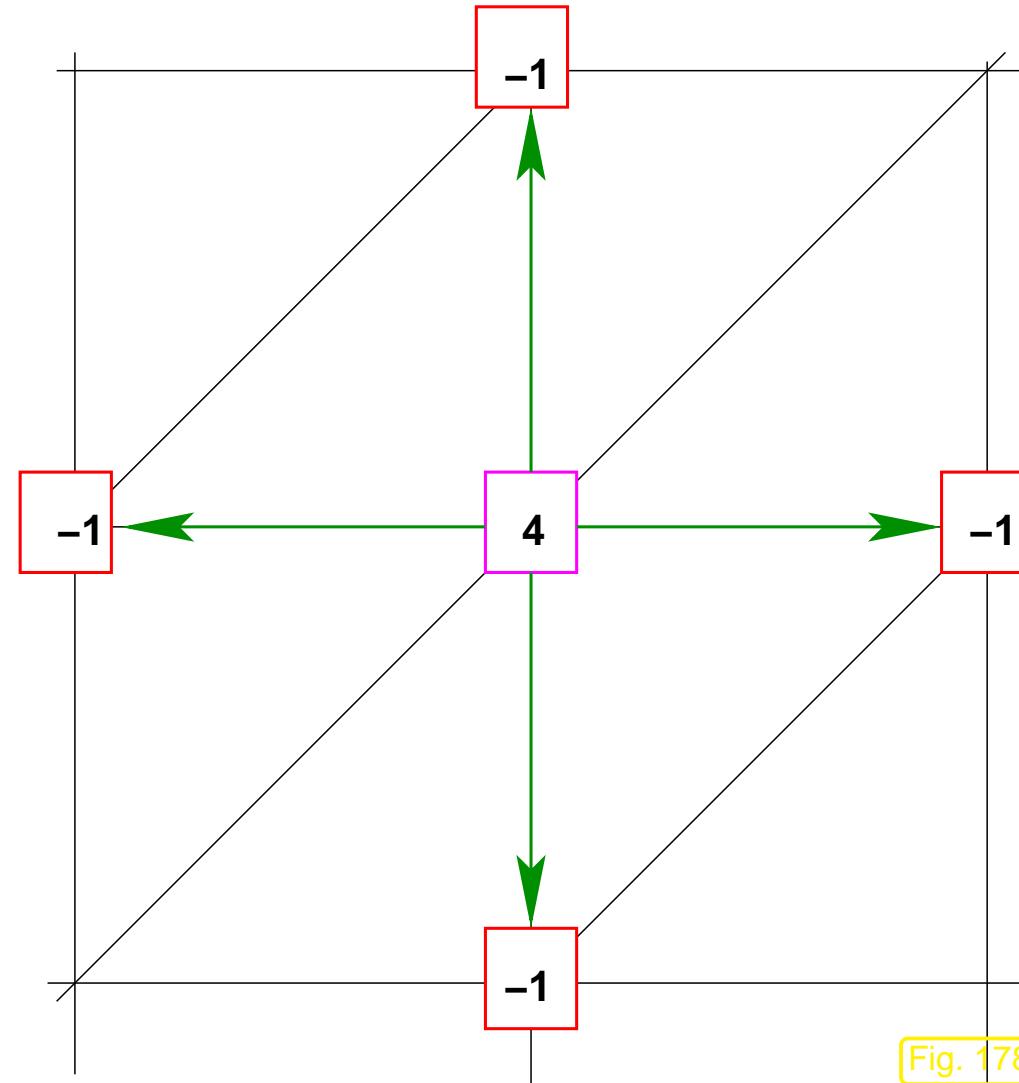
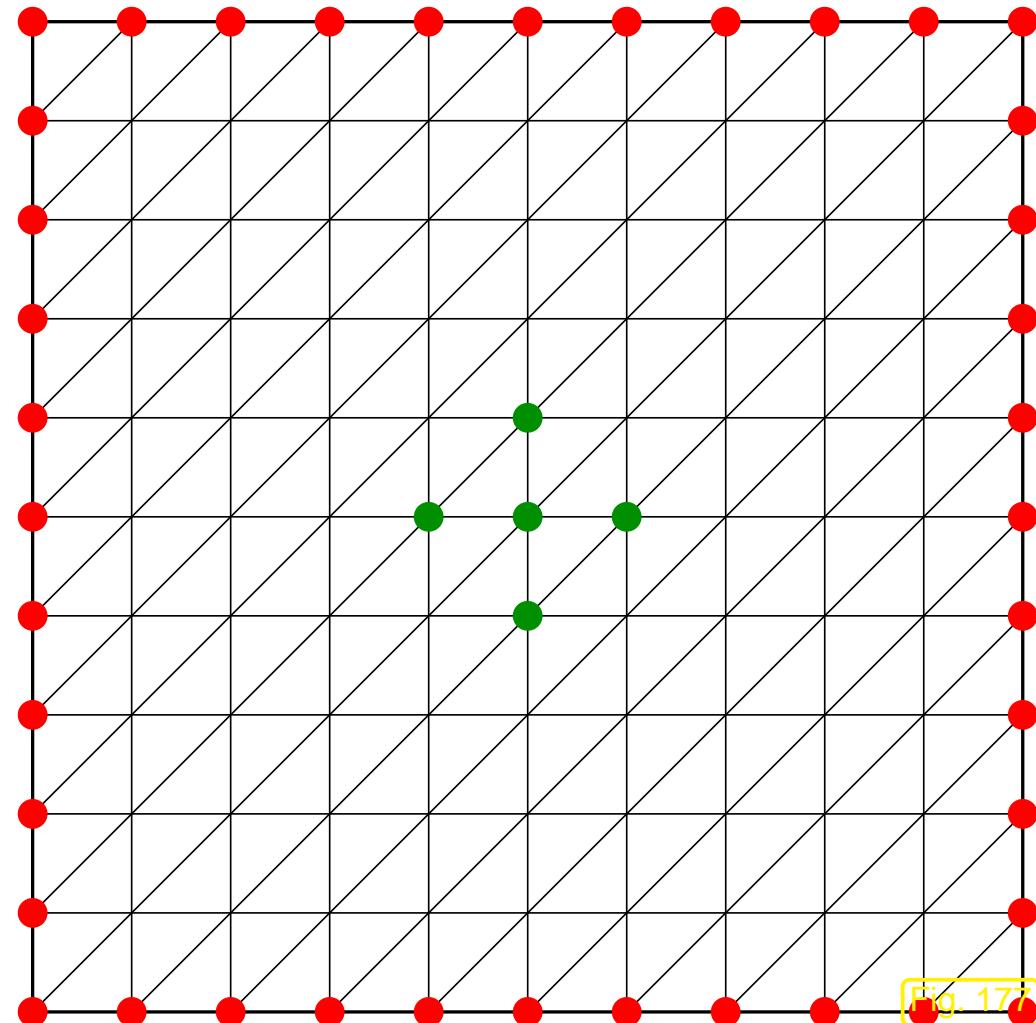


Fig. 178

The finite difference solution $(\mu_{i,j})_{1 \leq i,j \leq M}$ will attain its maximal value somewhere:

$$\exists n, m \in \{1, \dots, M-1\}: \quad \mu_{n,m} = \mu_{\max} := \max_{0 \leq i,j \leq M} \mu_{i,j} .$$

Assume: $(nh, mh)^T$ in the interior $\Leftrightarrow 1 \leq n, m < M$

Be aware of the following two facts:

$$\mu_{n-1,m}, \mu_{n+1,m}, \mu_{n,m-1}, \mu_{n,m+1} \leq \mu_{n,m} , \quad (5.7.6)$$

$$\mu_{n,m} = \frac{1}{4}(\mu_{n-1,m} + \mu_{n+1,m} + \mu_{n,m-1} + \mu_{n,m+1}) \quad (\text{average!}) .$$

$\Downarrow \leftarrow \text{"averaging argument"}$ (5.7.7)

$$\mu_{n-1,m} = \mu_{n+1,m} = \mu_{n,m-1} = \mu_{n,m+1} = \mu_{n,m} ! \quad (5.7.8)$$

The same argument can now target the neighboring grid points $((n - 1)h, mh)^T, ((n + 1)h, mh)^T, (nh, (m - 1)h)^T, (nh, (m + 1)h)^T$. By induction we find:

► $\mu_{i,j} = \mu_{\max} \quad \forall 0 \leq i, j \leq M ,$

that is, the finite difference solution has to be *constant*!

► The finite difference solution can attain its maximum in the interior only in the case of constant boundary data g !

Maximum principle satisfied for $f = 0$!



Now we try to generalize the considerations of the previous example to the discretization by means of *linear Lagrangian finite elements* on a triangular mesh (of a polygonal domain $\Omega \subset \mathbb{R}^2$) see Sect. 3.2.

$\tilde{\mathbf{A}} \in \mathbb{R}^{M,M} \doteq \mathcal{S}_1^0(\mathcal{M})$ -Galerkin matrix for \mathbf{a} from (5.7.1) ($M := \sharp \mathcal{V}(\mathcal{M})$)

Row of this matrix connects all values $\mu_j = u_N(\mathbf{x}^j)$ of Galerkin solution $u_N \in \mathcal{S}_1^0(\mathcal{M})$ according to

$$(\tilde{\mathbf{A}})_{ii}\mu_i + \sum_{j \neq i} (\tilde{\mathbf{A}})_{ij}\mu_j = (\vec{\varphi})_i , \quad \mathbf{x}^i \text{ interior node} ,$$

where $\mu_j := g(\mathbf{x}^j)$ for $\mathbf{x}^j \in \partial\Omega$.

The above averaging argument from Ex. 5.7.4 carries over, if the entries of $\tilde{\mathbf{A}}$ satisfy the following conditions:

- $(\tilde{\mathbf{A}})_{ii} > 0$ (positive diagonal) , (5.7.9)

- $(\tilde{\mathbf{A}})_{ij} \leq 0$ for $j \neq i$ (non-positive off-diagonal entries) ,(5.7.10)

- $\sum_j (\tilde{\mathbf{A}})_{ij} = 0$, if \mathbf{x}^i is interior node . (5.7.11)

(Recall [14, Def. 2.7.3]: matrix $\tilde{\mathbf{A}}$ satisfying (5.7.9)–(5.7.11) is **diagonally dominant.**)

averaging argument  $u_N(\mathbf{x}^i) = \max_{\mathbf{y} \in \mathcal{V}(\mathcal{M})} u_N(\mathbf{y})$ can only hold for an interior node \mathbf{x}^i ,
if $\mu_N = \text{const.}$

 Since $u_N \in \mathcal{S}_1^0(\mathcal{M})$ attains its extremal values at nodes of the mesh, the maximum principles holds for it in the case $f = 0$ provided that (5.7.9)–(5.7.11) are satisfied.

More general case $f \leq 0$:

 $(\vec{\varphi})_i = \int_{\Omega} f(\mathbf{x}) b_N^i(\mathbf{x}) d\mathbf{x} \leq 0$, since $b_N^i \geq 0$.

Then the averaging argument again rules out the existence of an interior maximum for an non-constant solution. The case $f \geq 0$ follows similarly.

When will (5.7.9)–(5.7.11) hold for $\mathcal{S}_1^0(\mathcal{M})$ -Galerkin matrix?

First consider

$$\kappa \equiv 1, \quad \leftrightarrow \quad -\Delta u = f$$

(The linear finite element discretization of this BVP was scrutinized in Sect. 3.2)

From formula (3.2.9) for element matrix & assembly, see

Fig. 72:

$$(\tilde{\mathbf{A}})_{ij} = -\cot \alpha - \cot \beta = -\frac{\sin(\alpha + \beta)}{\sin \alpha \sin \beta}.$$



$$(\tilde{\mathbf{A}})_{ij} \leq 0 \iff \alpha + \beta < \pi.$$

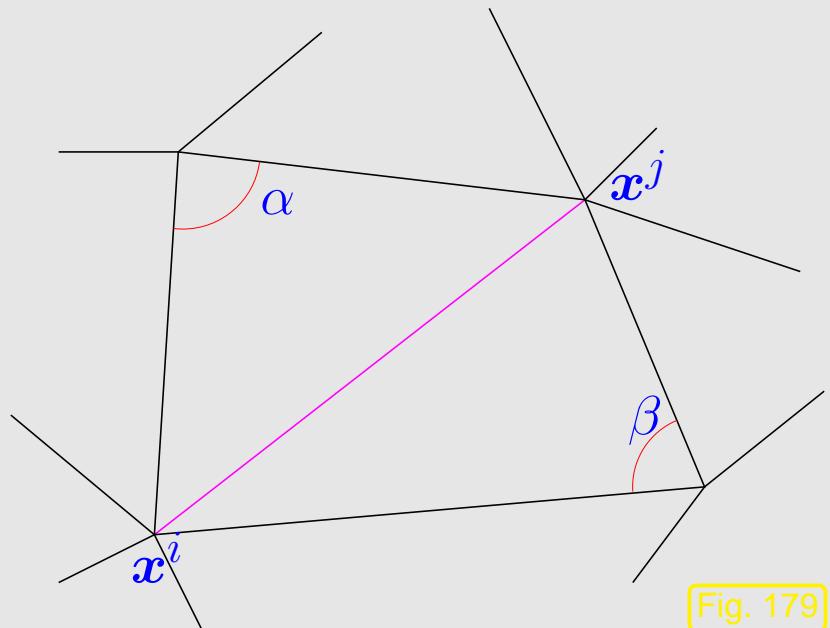


Fig. 179

Moreover

$$\sum_{\mathbf{x} \in \mathcal{V}(\mathcal{M})} b_N^{\mathbf{x}} \equiv 1 \Rightarrow \sum_j (\tilde{\mathbf{A}})_{ij} = 0 \quad (\leftrightarrow (5.7.11)).$$

The condition (5.7.9) $\leftrightarrow (\tilde{\mathbf{A}})_{ii} > 0$ is straightforward.

Theorem 5.7.12 (Maximum principle for linear FE solution of Poisson equation).

The linear finite element solution of

$$-\Delta u = 0 \quad \text{in } \Omega \subset \mathbb{R}^2, \quad u = g \quad \text{on } \partial\Omega,$$

*on a triangular mesh \mathcal{M} satisfies the **maximum principle** (5.7.3), if \mathcal{M} is a Delaunay triangulation.*

Remark 5.7.13 (Maximum principle for linear FE for 2nd-order elliptic BVPs).

For $\mathcal{S}_1^0(\mathcal{M})$ -Galerkin discretization of (5.7.1) on triangular mesh, the conditions (5.7.9)–(5.7.11) are fulfilled,

if all angles of triangles of $\mathcal{M} \leq \frac{\pi}{2}$.



Remark 5.7.14 (Maximum principle for higher order Lagrangian FEM).

Even when using p -degree Lagrangian finite elements with nodal basis functions associated with interpolation nodes, see Sect. 3.4.1, the discrete maximum principle will fail to hold on *any mesh* for $p > 1$.



6

2nd-Order Linear Evolution Problems

Now we study scalar linear partial differential equations for which *one* coordinate direction is special and identified with **time** and denoted by the independent variable t . The other coordinates are regarded as **spatial coordinates** and designated by $\mathbf{x} = (x_1, \dots, x_d)^T$.

➤ solution will be a “function of time and space”: $u = u(\mathbf{x}, t)$

The domain for such PDEs will have ***tensor product structure*** (tensor product of spatial domain and a bounded time interval):

Computational domain:

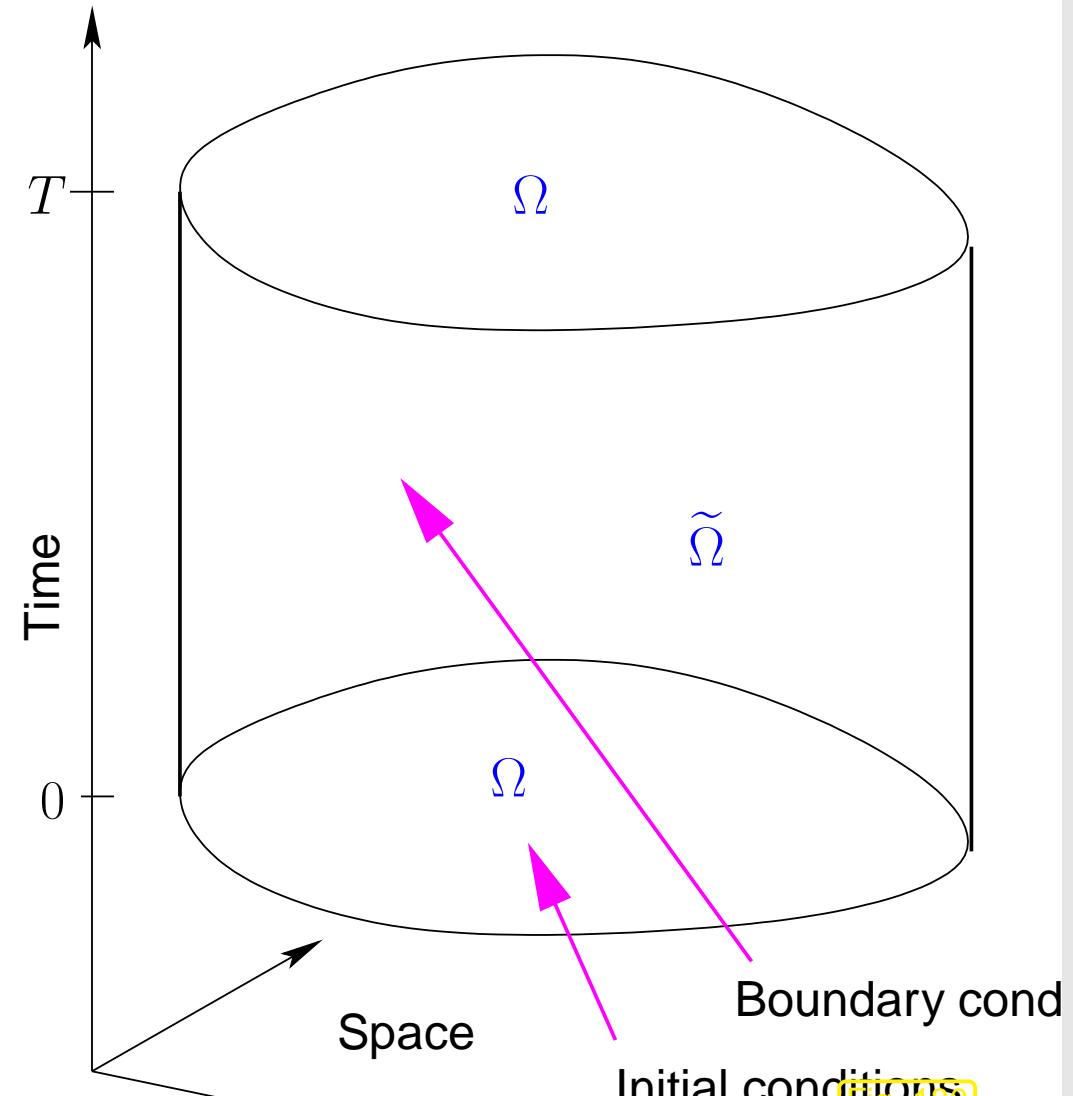
$$\tilde{\Omega} := \Omega \times]0, T[\subset \mathbb{R}^{d+1}.$$

► space-time cylinder

$\Omega \subset \mathbb{R}^d \hat{=} \text{ spatial domain } (\text{satisfying assumptions of Sect. 2.1.1})$

$T > 0 \hat{=} \text{final time}$

On $\Omega \times \{0\} \rightarrow \text{initial conditions},$
on $\partial\Omega \times]0, T[\rightarrow \text{(spatial) boundary conditions.}$



PDE for $u(\mathbf{x}, t)$

+

initial conditions

+

boundary conditions

= evolution problem

Note: No boundary conditions on $\Omega \times \{T\}$ (“final conditions”) are prescribed: time is supposed to have a “direction” that governs the flow of information in the evolution problem.



evolution problems (on bounded spatial domains) are also known as
initial-boundary value problems (IBVP).

Remark 6.0.1 (Initial time).

Why do we always pick initial time $t = 0$?

The modelled physical systems will usually be time-invariant, so that we are free to shift time. Remember the analogous situation with autonomous ODE, see [14, Sect. 11.1].



6.1

p. 595

6.1 Parabolic initial-boundary value problems

6.1.1 Heat equation

Sect. 2.5 treated *stationary* heat conduction: no change of temperature with time (temporal equilibrium)

Now we consider the evolution of a temperature distribution $u = u(\mathbf{x}, t)$.

$\Omega \subset \mathbb{R}^d$: space occupied by solid body (bounded spatial computational domain),
$\kappa = \kappa(\mathbf{x})$: (spatially varying) heat conductivity ($[\kappa] = \frac{\text{W}}{\text{Km}}$),
$T > 0$: final time for “observation period” $[0, T]$,
$u_0 : \Omega \mapsto \mathbb{R}$: initial temperature distribution in Ω ,
$g : \partial\Omega \times [0, T] \mapsto \mathbb{R}$: surface temperature , varying in space and time: $g = g(\mathbf{x}, t)$,
$f : \Omega \times [0, T] \mapsto \mathbb{R}$: time-dependent heat source/sink ($[f] = \frac{\text{W}}{\text{m}^3}$): $f = f(\mathbf{x}, t)$.

Goal: derive PDE governing *transient* heat conduction.

Conservation of energy:

$$\frac{d}{dt} \int_V \rho u \, d\mathbf{x} + \int_{\partial V} \mathbf{j} \cdot \mathbf{n} \, dS = \int_V f \, d\mathbf{x} \quad \text{for all “control volumes” } V \quad (6.1.1)$$

↑ ↑ ↑

energy stored in V power flux through ∂V heat generation in V

$\rho = \rho(\mathbf{x})$: (spatially varying) **heat capacity** ($[\rho] = \text{JK}^{-1}$), uniformly positive, cf. (2.5.4).

As in Sect. 2.5, now apply Gauss' Theorem Thm. 2.4.5 to the power flux integral in (6.1.1). This converts the surface integral to a volume integral over $\text{div } \mathbf{j}$ and we get

$$\frac{d}{dt} \int_V \rho u \, d\mathbf{x} + \int_V \text{div } \mathbf{j} \, d\mathbf{x} = \int_V f \, d\mathbf{x} \quad \text{for all “control volumes” } V$$

Now appeal to another version of the fundamental lemma of the calculus of variations, see Lemma 2.4.10, this time involving piecewise constant test functions.

► Local form of energy balance law (Heat equation)

$$\frac{\partial}{\partial t} (\rho u)(\mathbf{x}, t) + (\text{div}_{\mathbf{x}} \mathbf{j})(\mathbf{x}, t) = f(\mathbf{x}, t) \quad \text{in } \tilde{\Omega}. \quad (6.1.2)$$

The heat flux is linked to temperature variations by Fourier's law:

$$\mathbf{j}(\mathbf{x}) = -\kappa(\mathbf{x}) \mathbf{grad} u(\mathbf{x}), \quad \mathbf{x} \in \Omega. \quad (2.5.3)$$

From here we let all differential operators like **grad** and **div** act on the spatial independent variable **x**. As earlier, the independent variables **x** and **t** will be omitted frequently. Watch out!

Now, plug (2.5.3) into (6.1.2).



$$\frac{\partial}{\partial t}(\rho u) - \operatorname{div}(\kappa(\mathbf{x}) \mathbf{grad} u) = f \quad \text{in} \quad \tilde{\Omega} := \Omega \times]0, T[. \quad (6.1.3)$$

+ **Dirichlet boundary conditions** (fixed surface temperatur) on $\partial\Omega \times]0, T[$:

$$u(\mathbf{x}, t) = g(\mathbf{x}, t) \quad \text{for} \quad (\mathbf{x}, t) \in \partial\Omega \times]0, T[. \quad (6.1.4)$$

+ initial conditions for $t = 0$:

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \text{for all} \quad \mathbf{x} \in \Omega. \quad (6.1.5)$$

Terminology: (6.1.2) & (6.1.4) & (6.1.5) is a specimen of a

2nd-order **parabolic** initial-boundary value problem

Remark 6.1.6 (Compatible boundary and initial data).

Natural regularity requirements for Dirichlet data $\textcolor{blue}{g}$:

$\textcolor{blue}{g}$ continuous in time and space

Natural compatibility requirement at initial time and $\textcolor{blue}{u}_0 \in C^0(\bar{\Omega})$

$$\textcolor{blue}{g}(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \partial\Omega .$$

Remark 6.1.7 (Boundary conditions for 2nd-order parabolic IBVPs).

Physical intuition for transient heat conduction:

On $\partial\Omega[0, T[$ we can impose any of the boundary conditions discussed in Sect. 2.6:

- Dirichlet boundary conditions $u(\mathbf{x}, t) = g(\mathbf{x}, t)$, see (6.1.4) (fixed surface temperature),
- Neumann boundary conditions $\mathbf{j}(\mathbf{x}, t) \cdot \mathbf{n} = -h(\mathbf{x}, t)$ (fixed heat flux through surface),
- radiation boundary conditions $\mathbf{j}(\mathbf{x}, t) \cdot \mathbf{n} = \Psi(u(\mathbf{x}, t))$,

and any combination of these as discussed in Ex. 2.6.7, yet, *only one* of them at any part of $\partial\Omega\times]0, T[$, see Rem. 2.6.6.



6.1.2 Spatial variational formulation

Now we study the linear 2nd-order parabolic initial-boundary value problem with pure Dirichlet boundary conditions, introduced in the preceding section:

$$\frac{d}{dt}(\rho u) - \operatorname{div}(\kappa(\mathbf{x}) \operatorname{grad} u) = f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[, \quad (6.1.3)$$

$$u(\mathbf{x}, t) = g(\mathbf{x}, t) \quad \text{for } (\mathbf{x}, t) \in \partial\Omega \times]0, T[, \quad (6.1.4)$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \Omega . \quad (6.1.5)$$

Assume:

Homogeneous Dirichlet boundary conditions $g = 0$

The general case can be reduced to this by using the offset function trick, see Sect. 3.5.5, and solve the parabolic initial-boundary value problem for $w(\mathbf{x}, t) := u(\mathbf{x}, t) - \tilde{g}(\mathbf{x}, t)$, where $\tilde{g}(\cdot, t)$ is an extension of the Dirichlet data g to $\tilde{\Omega}$. Then w will satisfy homogeneous Dirichlet boundary conditions and solve an evolution equation with a modified source function $\tilde{f}(\mathbf{x}, t)$.

Now we pursue the formal derivation of the *spatial* variational formulation of (6.1.3)–(6.1.4).

The steps completely mirror those discussed in Sect. 2.8

STEP 1:

test PDE with functions $v \in H_0^1(\Omega)$

(do not test, where the solution is known, that is, on the boundary $\partial\Omega$)

Note: test function does *not depend on time*: $v = v(\mathbf{x})$!

STEP 2: *integrate over domain Ω*

STEP 3: *perform integration by parts in space*

(by using Green's first formula, Thm. 2.4.7)

STEP 4: [optional] *incorporate boundary conditions into boundary terms*

For the concrete PDE (6.1.3) and boundary conditions (6.1.4) refer to Ex. 2.8.1, for more general boundary conditions to Ex. 2.8.5.

Spatial variational form of (6.1.3)–(6.1.4): seek $t \in]0, T[$ $\mapsto u(t) \in H_0^1(\Omega)$

$$\int_{\Omega} \rho(\boldsymbol{x}) \dot{u}(t)v \, d\boldsymbol{x} + \int_{\Omega} \kappa(\boldsymbol{x}) \operatorname{grad} u(t) \cdot \operatorname{grad} v \, d\boldsymbol{x} = \int_{\Omega} f(\boldsymbol{x}, t)v(\boldsymbol{x}) \, d\boldsymbol{x} \quad \forall v \in H_0^1(\Omega), \quad (6.1.8)$$

$$u(0) = u_0 \in H_0^1(\Omega). \quad (6.1.9)$$

Be aware: $u(t) \hat{=} \text{function space } (H_0^1(\Omega)\text{-valued function on }]0, T[)$.

Also note that **grad** acts on the spatial independent variables that are suppressed in the notation $u(t)$.

☞ Notation: $\dot{u}(t) = \frac{\partial u}{\partial t}(t) \hat{=} \text{(partial) derivative w.r.t. time.}$

Shorthand notation (with obvious correspondences):

$$t \in]0, T[\mapsto u(t) \in V_0 \quad : \quad \begin{cases} \mathbf{m}(\dot{u}(t), v) + \mathbf{a}(u(t), v) = \ell(t)(v) & \forall v \in V_0 , \\ u(0) = u_0 \in V_0 . \end{cases} \quad (6.1.10)$$

Again, here $\ell(t) \doteq$ linear form valued function on $]0, T[$.

Concretely:

$$\begin{aligned} \mathbf{m}(u, v) &:= \int_{\Omega} \rho(\boldsymbol{x}) \dot{u}(t) v \, d\boldsymbol{x} , \quad u, v \in H_0^1(\Omega) , \\ \mathbf{a}(u, v) &:= \int_{\Omega} \kappa(\boldsymbol{x}) \operatorname{grad} u(t) \cdot \operatorname{grad} v \, d\boldsymbol{x} , \quad u, v \in H_0^1(\Omega) , \\ \ell(t)(v) &:= \int_{\Omega} f(\boldsymbol{x}, t) v(\boldsymbol{x}) \, d\boldsymbol{x} , \quad v \in H_0^1(\Omega) . \end{aligned}$$

Note that both \mathbf{m} and \mathbf{a} are *symmetric, positive definite* bilinear forms (\rightarrow Def. 2.1.22).

Equivalent formulation, since the bilinear form \mathbf{m} does not depend on time:

6.1

$$t \in]0, T[\mapsto u(t) \in V_0 \quad : \quad \begin{cases} \frac{d}{dt} \mathbf{m}(u(t), v) + \mathbf{a}(u(t), v) = \ell(t)(v) & \forall v \in V_0 , \\ u(0) = u_0 \in V_0 . \end{cases} \quad (6.1.11)$$

Now we are concerned with the **stability** of parabolic evolution problems: We investigate whether $\|u\|_{H^1(\Omega)}$ stays bounded for all times in the case $f \equiv 0$.

For the sake of simplicity: consider $\rho \equiv 1$ and $\kappa \equiv 1$

(General case is not more difficult, because both ρ and κ are bounded and uniformly positive, see (2.5.4).)

By the first Poincaré-Friedrichs inequality Thm. 2.2.16

$$\exists \gamma > 0: \|v\|_{H^1(\Omega)}^2 \geq \gamma \|v\|_{L^2(\Omega)}^2 \quad \forall v \in H_0^1(\Omega) . \quad (6.1.12)$$

In fact, Thm. 2.2.16 reveals $\gamma = \text{diam}(\Omega)^{-2}$.

Remark 6.1.13 (Differentiating bilinear forms with time-dependent arguments).

Consider (temporally) smooth $u : [0, T] \mapsto V_0$, $v : [0, T] \mapsto V_0$ and a *symmetric* bilinear form $b : V_0 \times V_0 \mapsto \mathbb{R}$.

What is $\frac{d}{dt}b(u(t), v(t))$?

Formal Taylor expansion:

$$\begin{aligned} b(u(t + \tau), v(t + \tau)) &= b(u(t) + \dot{u}(t)\tau + O(\tau^2), v(t) + \dot{v}(t)\tau + O(\tau^2)) \\ &= b(u(t), v(t)) + \tau(b(\dot{u}(t), v(t)) + b(u(t), \dot{v}(t))) + O(\tau^2). \end{aligned}$$

► $\lim_{\tau \rightarrow 0} \frac{b(u(t + \tau), v(t + \tau)) - b(u(t), v(t))}{\tau} = b(\dot{u}(t), v(t)) + b(u(t), \dot{v}(t))$.

This is a general **product rule**, see [14, Eq. 3.4.3].



Lemma 6.1.14 (Decay of solutions of parabolic evolutions).

For $\rho \equiv 1$, $\kappa \equiv 1$, and $f \equiv 0$ the solution $u(t)$ of (6.1.8) satisfies

$$\|u(t)\|_{L^2(\Omega)} \leq e^{-\gamma t} \|u_0\|_{L^2(\Omega)} , \quad |u(t)|_{H^1(\Omega)} \leq e^{-\gamma t} |u_0|_{H^1(\Omega)} \quad \forall t \in]0, T[.$$

Proof. Multiply the solution of the parabolic IBVP with an exponential weight function:

$$w(t) := \exp(\gamma t) u(t) \in H_0^1(\Omega) \Rightarrow \dot{w} := \frac{dw}{dt}(t) = \gamma w(t) + \exp(\gamma t) \frac{du}{dt}(t) , \quad (6.1.15)$$

solves the parabolic IBVP

$$\begin{aligned} \mathbf{m}(\dot{w}, v) + \tilde{\mathbf{a}}(w, v) &= 0 \quad \forall v \in V , \\ w(0) &= u_0 , \end{aligned} \quad (6.1.16)$$

with $\tilde{\mathbf{a}}(w, v) = \mathbf{a}(w, v) - \gamma \mathbf{m}(w, v)$, γ from (6.1.12). To see this, use that $u(t)$ solves (6.1.11) with $f \equiv 0$ (elementary calculation).

Note:

$$(6.1.12) \Rightarrow \tilde{\mathbf{a}}(v, v) \geq 0 \quad \forall v \in V$$

Exponential decay of $\|\cdot\|_{L^2(\Omega)}$ -norm of solution:

$$\frac{d}{dt} \frac{1}{2} \|w\|_{L^2(\Omega)}^2 = \frac{d}{dt} \frac{1}{2} \mathbf{m}(w, w) = \mathbf{m}(\dot{w}, w) = -\tilde{\mathbf{a}}(w, w) \leq 0 \quad (6.1.17)$$

This confirms that $t \mapsto \|w\|_{L^2(\Omega)}(t)$ is a decreasing function, which involves

$$(6.1.17) \Rightarrow \|w(t)\|_{L^2(\Omega)} \leq \|w(0)\|_{L^2(\Omega)} ,$$

and the first assertion of the Lemma is evident. Next, we verify the exponential decay of $\|\cdot\|_{H^1(\Omega)}$ -norm of solution by a similar trick:

$$\frac{d}{dt} \|w\|_{\tilde{\mathbf{a}}}^2 = \tilde{\mathbf{a}}\left(\frac{d}{dt}w, w\right) = -\mathbf{m}\left(\frac{d}{dt}w, \frac{d}{dt}w\right) \leq 0 \Rightarrow \|w(t)\|_{\tilde{\mathbf{a}}} \leq \|w(0)\|_{\tilde{\mathbf{a}}} ,$$

► $|w(t)|_{H^1(\Omega)}^2 \leq |w(0)|_{H^1(\Omega)}^2 - \underbrace{\gamma(\|w(0)\|_{L^2(\Omega)}^2 - \|w(t)\|_{L^2(\Omega)}^2)}_{\geq 0 \text{ by (6.1.17)}} .$

► Exponential decrease of energy during parabolic evolution without excitation
("Parabolic evolutions dissipate energy")

► MATLAB animation `heatevl()`

6.1.3 Method of lines

Idea: Apply **Galerkin discretization** (\rightarrow Sect. 3.1) to abstract linear parabolic variational problem (6.1.11).

$$t \in]0, T[\mapsto u(t) \in V_0 \quad : \quad \begin{cases} \mathbf{m}(\dot{u}(t), v) + \mathbf{a}(u(t), v) = \ell(t)(v) & \forall v \in V_0 , \\ u(0) = u_0 \in V_0 . \end{cases} \quad (6.1.11)$$

1st step: replace V_0 with a finite dimensional subspace $V_{0,N}$, $N := \dim V_{0,N} < \infty$

► Discrete parabolic evolution problem

$$t \in]0, T[\mapsto u(t) \in V_{0,N} \quad : \quad \begin{cases} \mathbf{m}(\dot{u}_N(t), v_N) + \mathbf{a}(u_N(t), v_N) = \ell(t)(v_N) & \forall v_N \in V_{0,N} , \\ u_N(0) = \text{projection/interpolant of } u_0 \text{ in } V_{0,N} . \end{cases} \quad (6.1.18)$$

2nd step: introduce (ordered) basis $\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\}$ of $V_{0,N}$

$$(6.1.18) \quad \Rightarrow \quad \begin{cases} \mathbf{M} \left\{ \frac{d}{dt} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) = \vec{\varphi}(t) & \text{for } 0 < t < T , \\ \vec{\mu}(0) = \vec{\mu}_0 . \end{cases} \quad (6.1.19)$$

- ▷ s.p.d. stiffness matrix $\mathbf{A} \in \mathbb{R}^{N,N}$, $(\mathbf{A})_{ij} := \mathbf{a}(b_N^j, b_N^i)$ (independent of time),
- ▷ s.p.d. mass matrix $\mathbf{M} \in \mathbb{R}^{N,N}$, $(\mathbf{M})_{ij} := \mathbf{m}(b_N^j, b_N^i)$ (independent of time),
- ▷ source (load) vector $\vec{\varphi}(t) \in \mathbb{R}^N$, $(\vec{\varphi}(t))_i := \ell(t)(b_N^i)$ (time-dependent),
- ▷ $\vec{\mu}_0 \hat{=} \text{coefficient vector of a projection of } u_0 \text{ onto } V_{0,N}$.

For the concrete linear parabolic evolution problem (6.1.8)–(6.1.9) and spatial finite element discretization based on a finite element trial/test space $V_{0,N} \subset H^1(\Omega)$ we can compute

- the mass matrix \mathbf{M} as the Galerkin matrix for the bilinear form $(u, v) \mapsto \int_{\Omega} \rho(\mathbf{x})uv \, d\mathbf{x}$, $u, v \in L^2(\Omega)$,
- the stiffness matrix \mathbf{A} as Galerkin matrix arising from the bilinear form $(u, v) \mapsto \int_{\Omega} \kappa(\mathbf{x}) \mathbf{grad} \, u \cdot \mathbf{grad} \, v \, d\mathbf{x}$, $u, v \in H^1(\Omega)$.

The calculations are explained in Sects. 3.5.3 and 3.5.4 and may involve numerical quadrature.

Note:

(6.1.19) is an ordinary differential equation (ODE) for $t \mapsto \vec{\mu}(t) \in \mathbb{R}^N$

Conversion (6.1.11) \rightarrow (6.1.19) through Galerkin discretization *in space only* is known as **method of lines**.

(6.1.19) $\hat{=}$

A **semi-discrete** evolution problem

Discretized in space \longleftrightarrow but still continuous in time

Remark 6.1.20 (Spatial discretization options).

Beside the Galerkin approach any other method for spatial discretization of 2nd-order elliptic BVPs can be used in the context of the method of lines: the matrices \mathbf{A} , \mathbf{M} may also be generated by finite differences (\rightarrow Sect. 4.1), finite volume methods (\rightarrow Sect. 4.2), or collocation methods (\rightarrow Sect. 1.5.2).



6.1

p. 611

6.1.4 Timestepping

For implementation we need a **fully discrete** evolution problem. This requires additional discretization in time:

semi-discrete evolution problem (6.1.19) + timestepping  **fully discrete** evolution problem

Benefit of method of lines: we can apply already known integrators for initial value problems for ODEs to (6.1.19).

First, refresh central concepts from numerical integration of initial value problems for ODEs, see [14, Ch. 11], [14, Ch. 12]:

- single step methods of order p , see [14, Def. 11.2.1] and [14, Thm. 11.3],
- explicit and implicit Runge-Kutta single step methods, see [14, Sect. 11.4], [14, Sect. ??], encoded by Butcher scheme [14, Eq. 11.4.5], [14, Eq 12.3.3].
- the notion of a **stiff problem** (\rightarrow [14, Notion 12.2.1]),
- the definition of the **stability function** of a single step method, see [14, Thm. 12.3.2],
- the concept of **L-stability** [14, Def 12.3.3] and how to verify it for Runge-Kutta methods.

6.1.4.1 Single step methods

Recall: single step methods (\rightarrow [14, Def. 11.2.1])

- are based on a **temporal mesh** $\{0 = t_0 < t_1 < \dots < t_{M-1} < t_M := T\}$ (with local timestep size $\tau_j = t_j - t_{j-1}$),
- compute sequence $\left(\vec{\mu}^{(j)}\right)_{j=0}^M$ of approximations $\vec{\mu}^{(j)} \approx \mu(t_j)$ to the solution of (6.1.19) at the nodes of the temporal mesh according to

$$\vec{\mu}^{(j)} := \Psi^{t_{j-1}, t_j} \vec{\mu}^{(j-1)} := \Psi(t_{j-1}, t_j, \vec{\mu}^{(j-1)}) , \quad j = 1, \dots, M ,$$

where Ψ is the **discrete evolution** defining the single step method, see [14, Def. 11.2.1].

Example 6.1.21 (Euler timestepping). → [14, Sect. 11.2]

We target the initial value problem

$$\begin{aligned} \mathbf{M} \left\{ \frac{d}{dt} \vec{\boldsymbol{\mu}}(t) \right\} + \mathbf{A} \vec{\boldsymbol{\mu}}(t) &= \vec{\boldsymbol{\varphi}}(t) \quad \text{for } 0 < t < T , \\ \vec{\boldsymbol{\mu}}(0) &= \vec{\boldsymbol{\mu}}_0 . \end{aligned} \tag{6.1.19}$$

Explicit Euler method [14, Eq. 11.2.1]: replace $\frac{d}{dt}$ in (6.1.19) with forward difference quotient, see [14, Rem. 11.2.2]:

$$(6.1.19) \quad \Rightarrow \quad \mathbf{M} \vec{\boldsymbol{\mu}}^{(j)} = \mathbf{M} \vec{\boldsymbol{\mu}}^{(\overrightarrow{j-1})} - \tau_j \mathbf{A} \vec{\boldsymbol{\mu}}^{(j-1)} + \vec{\boldsymbol{\varphi}}(t_{j-1}) , \quad j = 1, \dots, M-1 . \tag{6.1.22}$$

Implicit Euler method [14, Eq. 11.2.4]: replace $\frac{d}{dt}$ in (6.1.19) with backward difference quotient

$$(6.1.19) \quad \Rightarrow \quad \mathbf{M} \vec{\boldsymbol{\mu}}^{(j)} = \mathbf{M} \vec{\boldsymbol{\mu}}^{(\overrightarrow{j-1})} - \tau_j \mathbf{A} \vec{\boldsymbol{\mu}}^{(j)} + \vec{\boldsymbol{\varphi}}(t_j) , \quad j = 1, \dots, M-1 . \tag{6.1.23}$$

Note that both (6.1.22) and (6.1.23) require the solution of a linear system of equations in each step

$$(6.1.22): \quad \vec{\boldsymbol{\mu}}^{(j)} = \vec{\boldsymbol{\mu}}^{(j-1)} + \tau_j \mathbf{M}^{-1} (\vec{\boldsymbol{\varphi}}(t_{j-1}) - \mathbf{A} \vec{\boldsymbol{\mu}}^{(j-1)}) ,$$

$$(6.1.23): \quad \vec{\boldsymbol{\mu}}^{(j)} = (\tau_j \mathbf{A} + \mathbf{M})^{-1} \left(\mathbf{M} \vec{\boldsymbol{\mu}}^{(\overrightarrow{j-1})} + \vec{\boldsymbol{\varphi}}(t_j) \right) .$$

Recall [14, Sect. 11.3]: both Euler methods are of first order.



Example 6.1.24 (Crank-Nicolson timestepping).

Crank-Nicolson method = implicit midpoint rule: replace $\frac{d}{dt}$ in (6.1.19) with symmetric difference quotient and average right hand side:

$$\begin{aligned} \mathbf{M} \left\{ \frac{d}{dt} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) &= \vec{\varphi}(t) \\ \Downarrow \\ \frac{\vec{\mu}^{(j)} - \vec{\mu}^{(j-1)}}{\tau} &= -\frac{1}{2} \mathbf{A} \left(\vec{\mu}^{(j)} + \vec{\mu}^{(j-1)} \right) + \frac{1}{2} (\vec{\varphi}(t_j) + \vec{\varphi}(t_{j-1})) . \end{aligned} \quad (6.1.25)$$

This yields a method that is 2nd-order consistent.



Generalization of Euler methods:

Runge-Kutta single step methods → [14, Sect. 11.4], [14, Sect. 12.3]

Definition 6.1.26 (General Runge-Kutta method). → [14, Def. 12.3.1]

For coefficients $b_i, a_{ij} \in \mathbb{R}$, $c_i := \sum_{j=1}^s a_{ij}$, $i, j = 1, \dots, s$, $s \in \mathbb{N}$, the discrete evolution $\Psi^{s,t}$ of an **s-stage Runge-Kutta single step method** (RK-SSM) for the ODE $\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$, is defined by

$$\mathbf{k}_i := \mathbf{f}\left(t + c_i \tau, \mathbf{y} + \tau \sum_{j=1}^s a_{ij} \mathbf{k}_j\right), \quad i = 1, \dots, s, \quad \Psi^{t,t+\tau} \mathbf{y} := \mathbf{y} + \tau \sum_{i=1}^s b_i \mathbf{k}_i.$$

The $\mathbf{k}_i \in \mathbb{R}^d$ are called **increments**.

Shorthand notation for **s**-stage Runge-Kutta methods: **Butcher scheme** → [14, Eq. 12.3.3]

$$\begin{array}{c|ccccc} \mathbf{c} & \mathfrak{A} \\ \hline \mathbf{b}^T & \hat{=} & \begin{array}{c|cccccc} c_1 & a_{11} & a_{12} & \dots & \dots & a_{1s} \\ c_2 & a_{21} & \ddots & & & a_{2s} \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ c_s & a_{s1} & \vdots & & & a_{ss} \\ \hline b_1 & b_2 & \dots & \dots & \dots & b_s \end{array}, & \mathbf{c}, \mathbf{b} \in \mathbb{R}^s, & \mathfrak{A} \in \mathbb{R}^{s,s}. \end{array} \quad (6.1.27)$$

Concretely for linear parabolic evolution: application of **s**-stage Runge-Kutta method to

$$\mathbf{M} \left\{ \frac{d}{dt} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) = \vec{\varphi}(t) \Leftrightarrow \dot{\vec{\mu}} = \underbrace{\mathbf{M}^{-1} (\vec{\varphi}(t) - \mathbf{A} \vec{\mu}(t))}_{=\mathbf{f}(t, \vec{\mu})}. \quad (6.1.19)$$

Then simply plug this into the formulas of Def. 6.1.26.

- Timestepping scheme for (6.1.19): compute $\vec{\mu}^{(j+1)}$ from $\vec{\mu}^{(j)}$ through

$$\vec{\kappa}_i \in \mathbb{R}^N: \quad \mathbf{M}\vec{\kappa}_i + \sum_{m=1}^s \tau a_{im} \mathbf{A} \vec{\kappa}_m = \vec{\varphi}(t_j + c_i \tau) - \mathbf{A} \vec{\mu}^{(j)}, \quad i = 1, \dots, s, \quad (6.1.28)$$

$$\vec{\mu}^{(j+1)} = \vec{\mu}^{(j)} + \tau \sum_{m=1}^s \vec{\kappa}_m b_m. \quad (6.1.29)$$

Note: For an implicit RK-method (6.1.28) is a linear system of equations of size Ns .

6.1.4.2 Stability

Example 6.1.30 (Convergence of Euler timestepping).

Parabolic evolution problem in one spatial dimension (IBVP):

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad \text{in } [0, 1] \times]0, 1[, \quad (6.1.31)$$

$$u(t, 0) = u(t, 1) = 0 \quad \text{for } 0 \leq t \leq 1, \quad u(0, x) = \sin(\pi x) \quad \text{for } 0 < x < 1. \quad (6.1.32)$$

► exact solution $u(t, x) = \exp(-\pi^2 t) \sin(\pi x).$ (6.1.33)

- Spatial finite element Galerkin discretization by means of linear finite elements ($V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M})$) on equidistant mesh \mathcal{M} with meshwidth $h := \frac{1}{N} \rightarrow$ Sect. 1.5.1.2.
- $u_{N,0} := I_1 u_0$ by linear interpolation on \mathcal{M} , see Sect. 5.3.1.
- Timestepping by explicit and implicit Euler method (6.1.22), (6.1.23) with uniform timestep $\tau := \frac{1}{M}.$

Galerkin matrices, see (1.5.56):

$$\mathbf{A} = \frac{1}{h} \begin{pmatrix} 2 & -1 & 0 & & & & 0 \\ -1 & 2 & -1 & & & & \\ 0 & \ddots & \ddots & \ddots & & & \\ & & & & \ddots & \ddots & 0 \\ & & & & -1 & 2 & -1 \\ 0 & & & & 0 & -1 & 2 \end{pmatrix}, \quad \mathbf{M} = \frac{h}{6} \begin{pmatrix} 4 & 1 & 0 & & & & 0 \\ 1 & 4 & 1 & & & & \\ 0 & \ddots & \ddots & \ddots & & & \\ & & & & \ddots & \ddots & 0 \\ & & & & 1 & 4 & 1 \\ 0 & & & & 0 & 1 & 4 \end{pmatrix}.$$

6.1

Code 6.1.34: Euler timestepping for (6.1.31)

```
1 function [errex,errimp] = sinevl(N,M,u)
2 % Solve fully discrete two-point parabolic evolution problem (6.1.31)
3 % in  $[0,1] \times ]0,1[$ . Use both explicit and implicit Euler method for timestepping
4 % N: number of spatial grid cells
5 % M: number of timesteps
6 % u: handle of type @(t,x) to exact solution
7
8 if ( nargin < 3 ), u = @(t,x) ( exp( - ( pi ^ 2 ) * t ) .* sin( pi * x ) ) ; end %
   Exact solution
9
10 h = 1/N; tau = 1/M; % Spatial and temporal meshwidth
11 x = h:h:1-h; % Spatial grid, interior points
12
13 % Finite element stiffness and mass matrix
14 Amat = gallery('tridiag',N-1,-1,2,-1)/h;
15 Mmat = h/6 * gallery('tridiag',N-1,1,4,1);
16 Xmat = Mmat+tau*Amat;
17
18 mu0 = u(0,x)'; % Discrete initial value
19 mui = mu0; mue = mu0;
20
```

```

21 %Timestepping
22 erre = 0; erri = 0;
23 for k=1:M
24   mue = mue - tau*(Mmat\ (Amat*mue)) ; % explicit Euler step
25   mui = Xmat\ (Mmat*mui); % implicit Euler step
26   utk = u(k*tau,x)';
27   erre = erre + norm(mue-utk)^2; % Computation of error norm
28   erri = erri + norm(mui-utk)^2;
29 end
30
31 errex = sqrt(erre*h*tau);
32 errimp = sqrt(erri*h*tau);

```

Evaluation of approximate space-time L^2 -norm of the discretization error:

$$\text{err}^2 := h\tau \cdot \sum_{j=1}^M \sum_{i=1}^{N-1} |u(t_j, x_i) - \mu_i^{(j)}|^2 . \quad (6.1.35)$$

Error norm for explicit Euler timestepping:

$N \setminus M$	50	100	200	400	800	1600	3200
5	Inf	0.009479	0.006523	0.005080	0.004366	0.004011	0.003834
10	Inf	Inf	Inf	Inf	0.001623	0.001272	0.001097
20	Inf	Inf	Inf	Inf	Inf	Inf	0.000405
40	Inf	Inf	Inf	Inf	Inf	Inf	Inf
80	Inf	Inf	Inf	Inf	Inf	Inf	Inf
160	Inf	Inf	Inf	Inf	Inf	Inf	Inf
320	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Error norm for implicit Euler timestepping:

$N \setminus M$	50	100	200	400	800	1600	3200
5	0.007025	0.001828	0.000876	0.002257	0.002955	0.003306	0.003482
10	0.009641	0.004500	0.001826	0.000461	0.000228	0.000575	0.000749
20	0.010303	0.005175	0.002509	0.001149	0.000461	0.000116	0.000058
40	0.010469	0.005345	0.002681	0.001321	0.000634	0.000289	0.000116
80	0.010511	0.005387	0.002724	0.001364	0.000677	0.000332	0.000159
160	0.010521	0.005398	0.002734	0.001375	0.000688	0.000343	0.000170
320	0.010524	0.005400	0.002737	0.001378	0.000691	0.000346	0.000172

Explicit Euler timestepping: we observe a glaring **instability** (exponential blow-up) in case of *large timestep combined with fine mesh*.

Implicit Euler timestepping: no blow-up at any combination of spatial and temporal mesh width.



Example 6.1.36 (`ode45` for discrete parabolic evolution).

Same IBVP and spatial discretization as in Ex. 6.1.30.

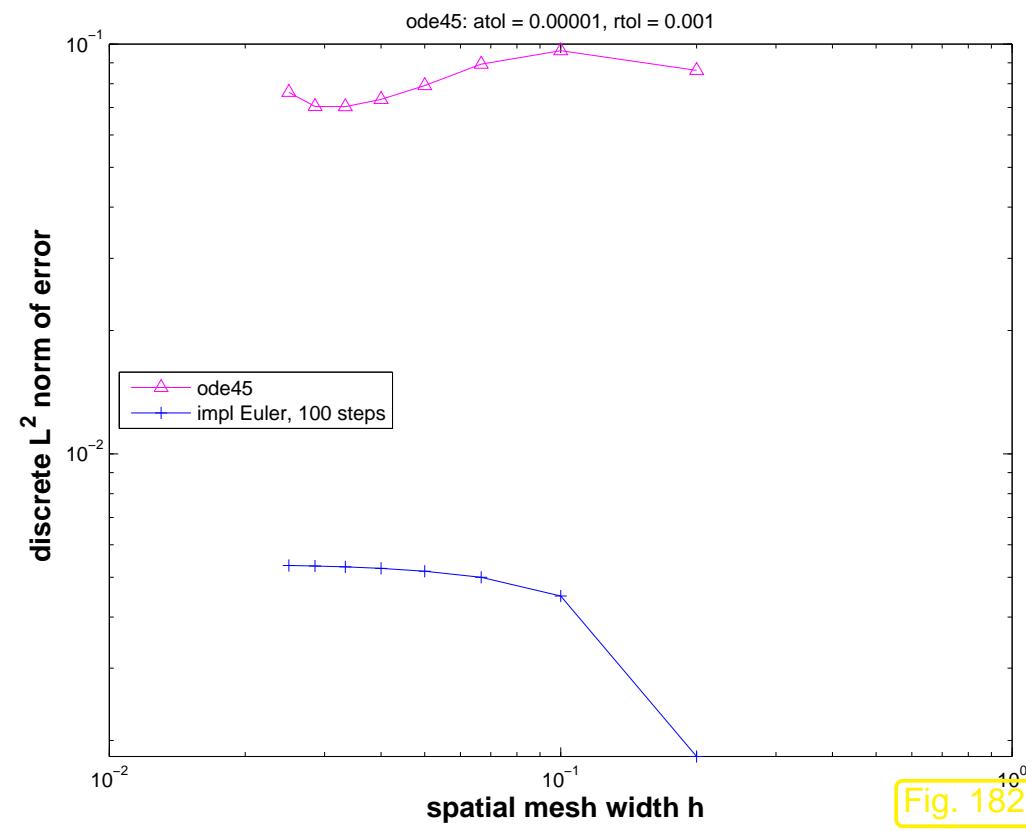
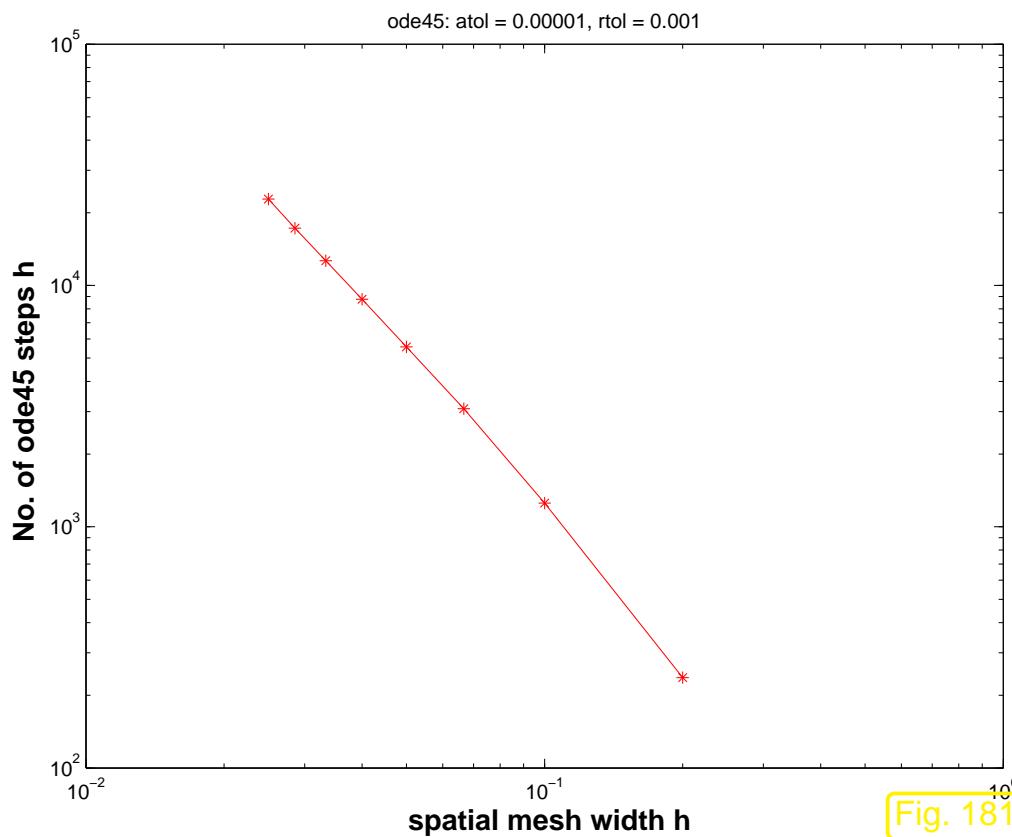
Adaptive Runge-Kutta timestepping by MATLAB standard integrator `ode45`.

Monitored:

- Number of timesteps as a function on spatial meshwidth h ,
- discrete L^2 -error (6.1.35).

Code 6.1.37: ode45 applied semi-discrete (6.1.31)

```
1 function [Nsteps,err] = peode45(N,tol,u)
2 % Solving fully discrete two-point parabolic evolution problem (6.1.31)
3 % in  $[0,1] \times ]0,1[$  by means of adaptiv MATLAB standard Runge-Kutta integrator.
4 if (nargin < 3), u = @(t,x) (exp(-(pi^2)*t).*sin(pi*x)); end %
   Exact solution
5
6 % Finite element stiffness and mass matrix, see Sect. 1.5.1.2
7 h = 1/N; % spatial meshwidth
8 Amat = gallery('tridiag',N-1,-1,2,-1)/h;
9 Mmat = h/6*gallery('tridiag',N-1,1,4,1);
10 x = h:h:1-h; % Spatial grid, interior points
11
12 mu0 = u(0,x)'; % Discrete initial value
13 fun = @(t,muv) -(Mmat\ (Amat*muv)); % right hand side of ODE
14
15 opts = odeset('reltol',tol,'abstol',0.01*tol);
16 [t,mu] = ode45(fun,[0,1],mu0,opts);
17
18 Nsteps = length(t);
19 [T,X] = meshgrid(t,x); err = norm(mu'-u(T,X),'fro');
```



Observations:

- `ode45`: dramatic increase of no. of timesteps for $h_M \rightarrow 0$ without gain in accuracy.
- Implicit Euler achieves better accuracy with only 100 equidistant timesteps!



This reminds us of the **stiff initial value problems** studied in [14, Thm. 12.2]:

Notion 6.1.38 (Stiff IVP). → [14, Notion 12.2.1]

*An initial value problem for an ODE is called **stiff**, if stability imposes much tighter timestep constraints on explicit single step methods than the accuracy requirements.*

Admittedly, this is a fuzzy notion. Yet, it cannot be fleshed out on the abstract level, but has to be discussed for concrete evolution problem, which is done next.

Let us try to understand, why semi-discrete parabolic evolutions (6.1.19) arising from the method of lines lead to stiff initial value problems.

Technique: **Diagonalization**, cf. [14, Eq. 12.2.3]

(Recall the concept of a “square root” $\mathbf{M}^{1/2}$ of an s.p.d. matrix \mathbf{M} , see [14, Sect. 4.3])

\mathbf{A}, \mathbf{M} symmetric positive definite $\Rightarrow \mathbf{M}^{-1/2} \mathbf{A} \mathbf{M}^{-1/2}$ symmetric positive definite .

[14, Cor. 5.1.7] $\Rightarrow \exists$ orthogonal $\mathbf{T} \in \mathbb{R}^{N,N}$: $\mathbf{T}^\top \mathbf{M}^{-1/2} \mathbf{A} \mathbf{M}^{-1/2} \mathbf{T} = \mathbf{D} := \text{diag}(\lambda_1, \dots, \lambda_N)$,

where the $\lambda_i > 0$ are *generalized eigenvalues* for $\mathbf{A}\vec{\xi} = \lambda\mathbf{M}\vec{\xi} \Rightarrow \lambda_i \geq \gamma$ for all i (γ is the constant introduced in (6.1.12)).

► Transformation (“diagonalization”) of (6.1.19) based on substitution $\vec{\eta} := \mathbf{T}^\top \mathbf{M}^{1/2} \vec{\mu}$:

$$(6.1.19) \quad \vec{\eta} := \mathbf{T}^\top \mathbf{M}^{1/2} \vec{\mu} \quad \frac{d}{dt} \vec{\eta}(t) + \mathbf{D} \vec{\eta} = \mathbf{T}^\top \mathbf{M}^{-1/2} \vec{\varphi}(t). \quad (6.1.39)$$

► Since \mathbf{D} is *diagonal*, (6.1.39) amounts to N decoupled scalar ODEs (for eigencomponents η_i of $\vec{\mu}$).

Note: for $\vec{\varphi} \equiv 0, \lambda > 0$: $\eta_i(t) = \exp(-\lambda_i t) \eta_i(0) \rightarrow 0$ for $t \rightarrow \infty$

As in [14, Thm. 12.2.5] this transformation can be applied to the explicit Euler timestepping (6.1.22) (for $\vec{\varphi} \equiv 0$, uniform timestep $\tau > 0$)

$$\vec{\mu}^{(j)} = \vec{\mu}^{(j-1)} - \tau \mathbf{M}^{-1} \mathbf{A} \vec{\mu}^{(j-1)} \quad \vec{\eta} := \mathbf{T}^\top \mathbf{M}^{1/2} \vec{\mu} \quad \vec{\eta}^{(j)} = \vec{\eta}^{(j-1)} - \tau \mathbf{D} \vec{\mu}^{(j-1)},$$

6.1

that is, the decoupling of eigencomponents carries over to the explicit Euler method: for $i = 1, \dots, N$

$$\eta_i^{(j)} = \eta_i^{(j-1)} - \tau \lambda_i \eta_i^{(j-1)} \Rightarrow \boxed{\eta_i^{(j)} = (1 - \tau \lambda_i)^j \eta_i^{(0)}}. \quad (6.1.40)$$


$$|1 - \tau \lambda_i| < 1 \Leftrightarrow \lim_{j \rightarrow \infty} \eta_i^{(j)} = 0. \quad (6.1.41)$$

The condition $|1 - \tau \lambda_i| < 1$ enforces a

timestep size constraint:

$$\tau < \frac{2}{\lambda_i} \quad (6.1.42)$$

in order to achieve the qualitatively correct behavior $\lim_{j \rightarrow \infty} \eta_i^{(j)} = 0$ and to avoid blow-up $\lim_{j \rightarrow \infty} |\eta_i^{(j)}| = \infty$: the timestep size constraint (6.1.42) is necessary *only* for the sake of stability (not in order to guarantee a prescribed accuracy).

This accounts to the observed blow-ups in Ex. 6.1.30. On the other hand, adaptive stepsize control [14, Sect. 11.5] manages to ensure the timestep constraint, but the expense of prohibitively small timesteps that render the method *grossly inefficient, if some of the λ_i are large.*

The next numerical demonstrations and Lemma show that $\lambda_{\max} := \max_i \lambda_i$ will inevitably become huge for finite element discretization on fine meshes.

Example 6.1.43 (Behavior of generalized eigenvalues of $\mathbf{A}\vec{\mu} = \lambda\mathbf{M}\vec{\mu}$).

Bilinear forms associated with parabolic IBVP and homogeneous Dirichlet boundary conditions

$$a(u, v) = \int_{\Omega} \mathbf{grad} u \cdot \mathbf{grad} v \, dx , \quad m(u, v) = \int_{\Omega} u(x)v(x) \, dx , \quad u, v \in H_0^1(\Omega) .$$

Linear finite element Galerkin discretization, see Sect. 1.5.1.2 for 1D, and Sect. 3.2 for 2D.

Numerical experiments in 1D & 2D:

- $\Omega =]0, 1[$, equidistant meshes \rightarrow Ex. 6.1.30
- “disk domain” $\Omega = \{x \in \mathbb{R}^2 : \|x\| < 1\}$, sequence of regularly refined meshes.

Monitored: largest and smallest generalized eigenvalue

Code 6.1.44: Computation of extremal generalized eigenvalues

```
1 % LehrFEM MATLAB script for computing Dirichlet eigenvalues of Laplacian
2 % on a unit disc domain.
3
4 GD_HANDLE = @(x,varargin)zeros(size(x,1),1); % Zero Dirichlet data
5 H0 =[ .25 .2 .1 .05 .02 .01 0.005]'; % target mesh widths
6 NRef = length(H0); % Number of refinement steps
7
8 % Variables for mesh widths and eigenvalues
9 M_W = zeros(NRef,1); lmax = M_W; lmin = M_W;
10
11 % Main refinement loop
12 for iter = 1:NRef
13
14 % Set parameters for mesh
15 C = [0 0]; % Center of circle
16 R = 1; % Radius of circle
17 BBOX = [-1 -1; 1 1]; % Bounding box
18 DHANDLE = @dist_circ; % Signed distance function
19 HHANDLE = @h_uniform; % Element size function
20 FIXEDPOS = [];% Fixed boundary vertices of the mesh
21 DISP = 0; % Display flag
```

```

23 % Mesh generation
24 Mesh =
25     init_Mesh(BBOX,H0(iter),DHANDLE,HHANDLE,FIXEDPOS,DISP,C,R);
26 Mesh = add_Edges(Mesh);    % Provide edge information
27 Loc = get_BdEdges(Mesh);  % Obtain indices of edges on  $\partial\Omega$ 
28 Mesh.BdFlags = zeros(size(Mesh.Edges,1),1);
29 Mesh.BdFlags(Loc) = -1;   % Flag boundary edges
30 Mesh.ElemFlag = zeros(size(Mesh.Elements,1),1);
31 M_W(iter) = get_MeshWidth(Mesh); % Get mesh width
32
33 fprintf('Mesh on level %i: %i elements, h =
34     %f\n',iter,size(Mesh,1),M_W(iter));
35 % Assemble stiffness matrix and mass matrix
36 A = assemMat_LFE(Mesh,@STIMA_Lapl_LFE,P7O6());
37 M = assemMat_LFE(Mesh,@MASS_LFE,P7O6());
38 % Incorporate Dirichlet boundary data (nothing to do here)
39 [U,FreeNodes] = assemDir_LFE(Mesh,-1,GD_HANDLE);
40 A = A(FreeNodes,FreeNodes);
41 M = M(FreeNodes,FreeNodes);
42
43 % Use MATLAB's built-in eigs-function to compute the
44 % extremal eigenvalues, see [14, Sect. 5.4].
45 NEigen = 6;

```

```

44 d = eigs(A,M,NEigen,'sm'); lmin(iter) = min(d);
45 d = eigs(A,M,NEigen,'lm'); lmax(iter) = max(d);
46 end
47
48 figure; plot(M_W,lmin,'b-+',M_W,lmax,'r-*'); grid on;
49 set(gca,'XScale','log','YScale','log','XDir','reverse');
50 title('bf Eigenvalues of Laplacian on unit disc');
51 xlabel('bf mesh width h','fontsize',14);
52 ylabel('bf generalized eigenvalues','fontsize',14);
53 legend('lambda_min','lambda_max','Location','NorthWest')
54 p = polyfit(log(M_W),log(lmax),1);
55 add_Slope(gca,'east',p(1));
56
57 print -depsc2 '../.../Slides/NPDEPics/geneigdisklfe.eps';

```

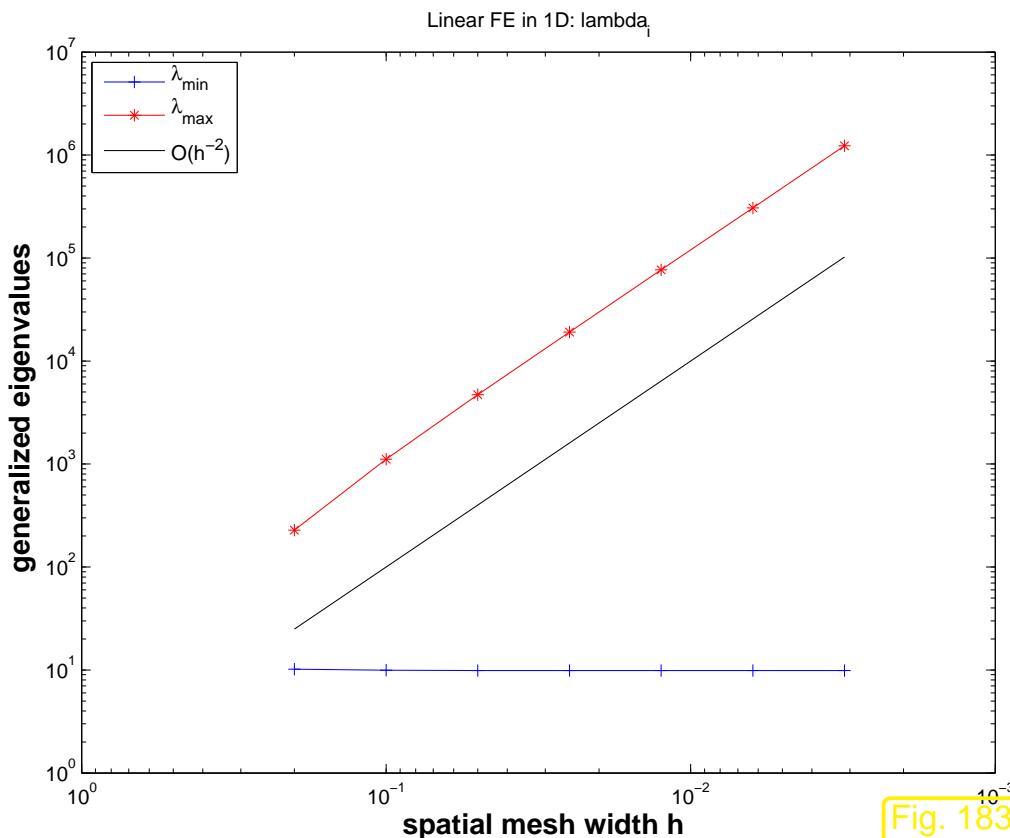


Fig. 183

$$\Omega =]0, 1[$$

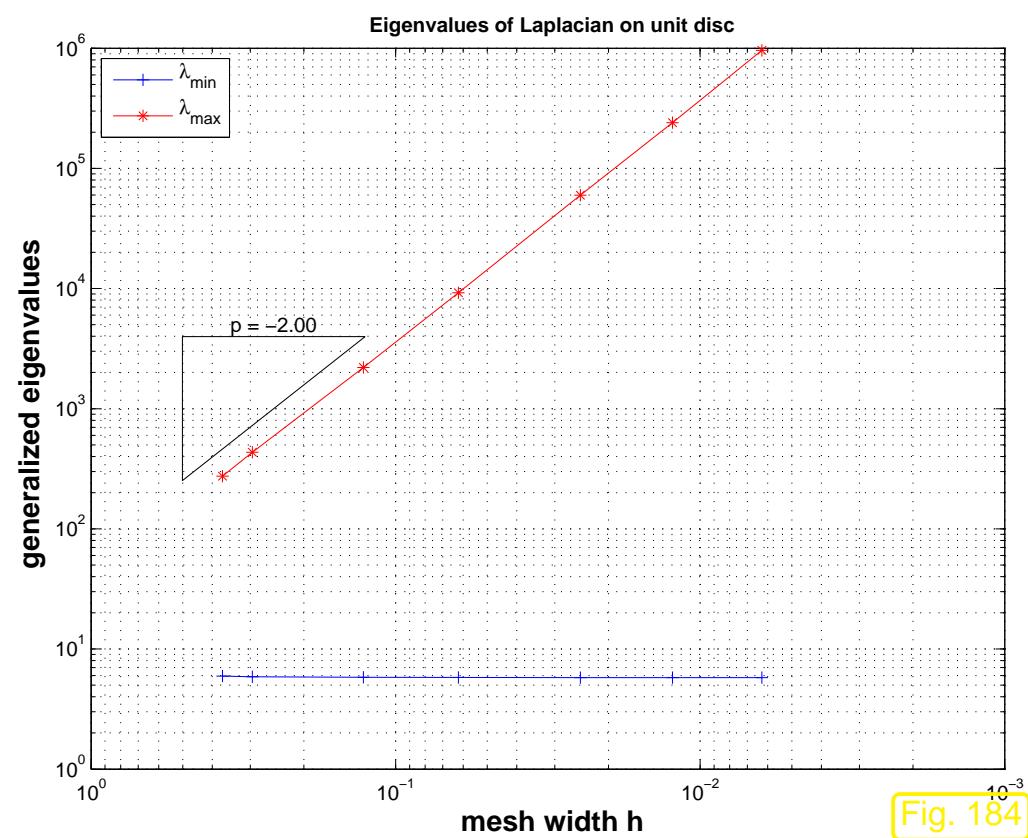


Fig. 184

$$\Omega = \{x \in \mathbb{R}^2 : \|x\| < 1\}$$

Observation:

- $\lambda_{\min} := \min_i \lambda_i$ does hardly depend on the mesh width.
- $\lambda_{\max} := \max_i \lambda_i$ displays a $O(h_M^{-2})$ growth as $h_M \rightarrow 0$



Remark 6.1.45 (Spectrum of elliptic operators).

The observation made in Ex. 6.1.43 is not surprising!

To understand why, let us translate the generalized eigenproblem “back to the ODE/PDE level”:

$$\mathbf{A}\vec{\mu} = \lambda\mathbf{M}\vec{\mu} \quad (6.1.46)$$



$$u_N \in V_{0,N}: \quad \mathbf{a}(u_N, v_N) = \lambda \mathbf{m}(u_N, v_N) \quad \forall v_N \in V_{0,N} .$$

 ← “undo Galerkin discretization”

$$u \in H_0^1(\Omega): \quad \int_{\Omega} \mathbf{grad} u \cdot \mathbf{grad} v \, dx = \lambda \int_{\Omega} u \cdot v \, dx \quad \forall v \in H_0^1(\Omega) .$$



$$-\Delta u = \lambda u \quad \text{in } \Omega , \quad u = 0 \quad \text{on } \partial\Omega , \quad (6.1.47)$$

which is a so-called **elliptic eigenvalue problem**.

It is easily solved in 1D on $\Omega =]0, 1[$:

$$(6.1.47) \hat{=} \quad \frac{d^2u}{dx^2}(x) = \lambda u(x), \quad 0 < x < 1, \quad u(0) = u(1) = 0.$$

$$\Rightarrow \quad u_k(x) = \sin(k\pi x) \quad \leftrightarrow \quad \lambda_k = (\pi k)^2, \quad k \in \mathbb{N}.$$

Note that we find an infinite number of eigenfunctions and eigenvalues, parameterized by $k \in \mathbb{N}$. The eigenvalues tend to ∞ for $k \rightarrow \infty$:

$$\lambda_k = O(k^2) \quad \text{for } k \rightarrow \infty.$$

Of course, (6.1.46) can have a finite number of eigenvectors only. Crudely speaking, they correspond to those eigenfunctions $u_k(x) = \sin(k\pi x)$ that can be resolved by the mesh (if u_k “oscillates too much”, then it cannot be represented on a grid). These are the first N so that we find in 1D for an equidistant mesh

$$\lambda_{\max} = O(N^2) = O(h_{\mathcal{M}}^{-2}).$$

This is heuristics, but the following Lemma will a precise statement.

Lemma 6.1.48 (Behavior of generalized eigenvalues).

Let \mathcal{M} be a simplicial mesh and \mathbf{A} , \mathbf{M} denote the Galerkin matrices for the bilinear forms $a(u, v) = \int_{\Omega} \mathbf{grad} u \cdot \mathbf{grad} v \, dx$ and $m(u, v) = \int_{\Omega} u(x)v(x) \, dx$, respectively, and $V_{0,N} := S_{p,0}^0(\mathcal{M})$. Then the smallest and largest generalized eigenvalues of $\mathbf{A}\vec{\mu} = \lambda\mathbf{M}\vec{\mu}$, denoted by λ_{\min} and λ_{\max} , satisfy

$$\frac{1}{\text{diam}(\Omega)^2} \leq \lambda_{\min} \leq C \quad , \quad \lambda_{\max} \geq Ch_{\mathcal{M}}^{-2} \, ,$$

where the “generic constants” (\rightarrow Rem. 5.3.44) depend only on the polynomial degree p and the shape regularity measure $\rho_{\mathcal{M}}$.

Proof. (partial) We rely on the **Courant-Fischer min-max theorem** [14, Thm. 5.3.5] that, among other consequences, expresses the boundaries of the spectrum of a symmetric matrix through the extrema of its Rayleigh quotient

$$\mathbf{T} = \mathbf{T}^T \in \mathbb{R}^{N,N} \Rightarrow \lambda_{\min}(\mathbf{T}) = \min_{\vec{\xi} \in \mathbb{R}^N \setminus \{0\}} \frac{\vec{\xi}^T \mathbf{T} \vec{\xi}}{\vec{\xi}^T \vec{\xi}} \, , \quad \lambda_{\max}(\mathbf{T}) = \max_{\vec{\xi} \in \mathbb{R}^N \setminus \{0\}} \frac{\vec{\xi}^T \mathbf{T} \vec{\xi}}{\vec{\xi}^T \vec{\xi}} \, .$$

Apply this to the generalized eigenvalue problem

$$\mathbf{A}\vec{\mu} = \lambda\mathbf{M}\vec{\mu} \quad \vec{\zeta} := \mathbf{M}^{1/2}\vec{\mu} \Leftrightarrow \underbrace{\mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}}_{=: \mathbf{T}}\vec{\zeta} = \lambda\vec{\zeta} .$$

► $\lambda_{\min} = \min_{\vec{\mu} \neq 0} \frac{\vec{\mu}^T \mathbf{A} \vec{\mu}}{\vec{\mu}^T \mathbf{M} \vec{\mu}}, \quad \lambda_{\max} = \max_{\vec{\mu} \neq 0} \frac{\vec{\mu}^T \mathbf{A} \vec{\mu}}{\vec{\mu}^T \mathbf{M} \vec{\mu}} .$ (6.1.49)

As a consequence we only have to find bounds for the extrema of a **generalized Rayleigh quotient**, cf. [14, Eq. 5.3.13]. This generalized Rayleigh quotient can be expressed as

$$\frac{\vec{\mu}^T \mathbf{A} \vec{\mu}}{\vec{\mu}^T \mathbf{M} \vec{\mu}} = \frac{\mathbf{a}(u_N, u_N)}{\mathbf{m}(u_N, u_N)}, \quad \vec{\mu} \hat{=} \text{coefficient vector for } u_N . \quad (6.1.50)$$

Now we discuss a lower bound for λ_{\max} , which can be obtained by inserting a suitable *candidate function* into (6.1.50).

Discussion for special setting: $V_{0,N} = \mathcal{S}_1^0(\mathcal{M})$ on triangular mesh \mathcal{M}

Candidate function: “tent function” $u_N = b_N^i$ (\rightarrow Sect. 3.2.3) for some node $\mathbf{x}^i \in \mathcal{V}(\mathcal{M})$ of the mesh!

By elementary computations as in Sect. 3.2.5 we find

$$\mathbf{a}(b_N^i, b_N^i) \approx C, \quad \mathbf{m}(b_N^i, b_N^i) \geq C \max_{K \in \mathcal{U}(\mathbf{x}^i)} h_K^2 , \quad (6.1.51)$$

where the generic constants $C > 0$ depend on the shape regularity measure $\rho_{\mathcal{M}}$ only.

$$(6.1.49) \text{ & } (6.1.51) \Rightarrow \lambda_{\max} \geq Ch_{\mathcal{M}}^{-2}.$$

□

Lemma 6.1.48 ► timestep constraint (6.1.42) unacceptable to semi-discrete parabolic evolutions!

From [14, Sect. 12.3] we already know that some *implicit* single step methods are not affected by stability induced timestep constraints.

Recall [14, Ex. 12.3.1]: apply diagonalization technique, see (6.1.39), to implicit Euler timestepping with uniform timestep $\tau > 0$

$$\vec{\mu}^{(j)} = \vec{\mu}^{(j-1)} - \tau \mathbf{M}^{-1} \mathbf{A} \vec{\mu}^{(j)} \quad \vec{\eta} := \mathbf{T}^{\top} \mathbf{M}^{1/2} \vec{\mu} \quad \vec{\eta}^{(j)} = \vec{\eta}^{(j-1)} - \tau \mathbf{D} \vec{\mu}^{(j)},$$

6.1

that is, the decoupling of eigencomponents carries over to the implicit Euler method: for $i = 1, \dots, N$

$$\eta_i^{(j)} = \eta_i^{(j-1)} - \tau \lambda_i \eta_i^{(j)} \Rightarrow \boxed{\eta_i^{(j)} = \left(\frac{1}{1+\tau\lambda_i}\right)^j \eta_i^{(0)}}. \quad (6.1.52)$$

$$\left[\left| \frac{1}{1+\tau\lambda_i} \right| < 1 \quad \text{and} \quad \lambda_i > 0 \quad \Rightarrow \quad \right] \lim_{j \rightarrow \infty} \eta_i^{(j)} = 0 \quad \forall \tau > 0. \quad (6.1.53)$$

This diagonalization trick can be applied to general Runge-Kutta single step methods (RKSSM, \rightarrow Def. 6.1.26). Loosely speaking, the following diagram commutes

$$\begin{array}{ccc} \mathbf{M} \frac{d}{dt} \vec{\mu} + \mathbf{A} \mu = 0 & \xrightarrow{\text{transformation } \vec{\eta} = \mathbf{T}^T \mathbf{M}^{1/2} \vec{\mu}} & \frac{d}{dt} \eta_i = -\lambda_i \eta_i, i = 1, \dots, N \\ \text{RK-SSM} \downarrow & & \downarrow \text{RK-SSM} \\ \vec{\mu}^{(j)} = \Psi^\tau \vec{\mu}^{(j-1)} & \xrightarrow{\text{transformation } \vec{\eta} = \mathbf{T}^T \mathbf{M}^{1/2} \vec{\mu}} & \vec{\eta}_i^{(j)} = \tilde{\Psi}^\tau \vec{\eta}_i^{(j-1)}, i = 1, \dots, N. \end{array} \quad (6.1.54)$$

The bottom line is

that we have to study the behavior of the RK-SSM *only* for linear scalar ODEs $\dot{y} = -\lambda y$, $\lambda > 0$.

This is the gist of the **model problem analysis** discussed in [14, Sect. 12.3].

There we saw that everything boils down to inspecting the modulus of a rational **L-stability function** on \mathbb{C} , see [14, Thm. 12.3.2]. This gave rise to the concept of **L-stability**, see [14, Def. 12.3.3]. Here, we will not delve into a study of stability functions.

Necessary condition for suitability of a single step method for semi-discrete parabolic evolution problem (6.1.19) (“method of lines”):

The discrete evolution $\Psi_\lambda^\tau : \mathbb{R} \mapsto \mathbb{R}$ of the single step method applied to the scalar ODE $\dot{y} = -\lambda y$ satisfies

$$\lambda > 0 \Rightarrow \lim_{j \rightarrow \infty} (\Psi_\lambda^\tau)^j y_0 = 0 \quad \forall y_0 \in \mathbb{R}, \quad \forall \tau > 0. \quad (6.1.55)$$

Definition 6.1.56 ($L(\pi)$ -stability).

A single step method satisfying (6.1.55) is called **$L(\pi)$ -stable**.

Example 6.1.57 ($L(\pi)$ -stable Runge-Kutta single step methods).

Simplest example: implicit Euler timestepping (6.1.23).

Some commonly used higher order methods, specified through their Butcher schemes, see (6.1.27):

$$\begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ \hline 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array}$$

(6.1.58)

RADAU-3 scheme (order 3)

$$\begin{array}{c|cc} \lambda & \lambda & 0 \\ \hline 1 & 1 - \lambda & \lambda \\ \hline & 1 - \lambda & \lambda \end{array}, \quad \lambda := 1 - \frac{1}{2}\sqrt{2}, \quad (6.1.59)$$

SDIRK-2 scheme (order 2)

More examples → [14, Ex. 12.3.4]



6.1.5 Convergence

Why should one prefer complicated implicit $L(\pi)$ -stable Runge-Kutta single step methods (\rightarrow Ex. 6.1.57) to the simple implicit Euler method?

Silly question! Because these methods deliver “better accuracy”!

However, we need some clearer idea of what is meant by this. To this end, we now study the dependence of (a norm of) the discretization error for a parabolic IBVP on the parameters of the spatial and temporal discretization.

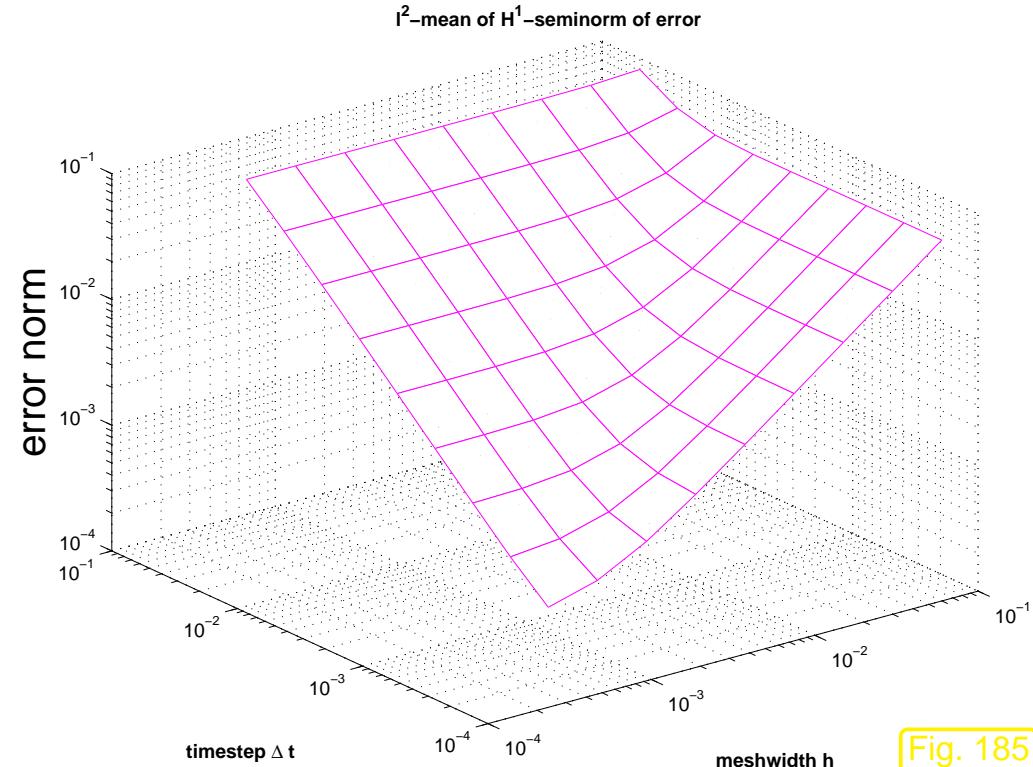
Example 6.1.60 (Convergence of fully discrete timestepping in one spatial dimension).

- $\frac{d}{dt}u - u'' = f(t, x)$ on $]0, 1[\times]0, 1[$
- exact solution $u(x, t) = (1 + t^2)e^{-\pi^2 t} \sin(\pi x)$, source term accordingly
- Linear finite element Galerkin discretization equidistant mesh, see Sect. 1.5.1.2, $V_{0,N} = \mathcal{S}_{1,0}^0(\mathcal{M})$,
- piecewise linear spatial approximation of source term $f(x, t)$
- implicit Euler timestepping (\rightarrow Ex. 6.1.21) with uniform timestep $\tau > 0$

Monitored:

$$\text{error norm } \left(\tau \sum_{j=1}^M |u - u_N(\tau j)|_{H^1(\Omega)}^2 \right)^{\frac{1}{2}}.$$

The norms $|u - u_N(\tau j)|_{H^1(\Omega)}$ were approximated by high order local quadrature rules, whose impact can be neglected.



▷ $h_{\mathcal{M}}$ - and τ -dependence of error norm

Observation:

τ small: error norm $\approx h_{\mathcal{M}}$

$h_{\mathcal{M}}$ small: error norm $\approx \tau$

The error seems to behave like

$$\text{error norm} \approx C_1 h_{\mathcal{M}} + C_2 \tau . \quad (6.1.61)$$

Recall from Sect. 5.3.5, Thm. 5.1.10, Thm. 5.3.42:

energy norm of spatial finite element discretization error $O(h_{\mathcal{M}})$ for $h_{\mathcal{M}} \rightarrow 0$

Since the implicit Euler method is *first order consistent* we expect

temporal timestepping error $O(\tau)$

(6.1.61) ➤ conjecture: total error is **sum** of spatial and temporal discretization error.

From Fig. 185 we draw the compelling conclusion:

- for big mesh width h_M (spatial error dominates) further reduction of timestep size τ is useless,
- if timestep τ is large (temporal error dominates), refinement of the finite element space does not yield a reduction of the total error.



Example 6.1.62 (Higher order timestepping for 1D heat equation).

- same IBVP as in Ex. 6.1.60
- spatial discretization on equidistant grid, *very small meshwidth* $h = 0.5 \cdot 10^{-4}$, $V_N = \mathcal{S}_{1,0}^0(\mathcal{M})$

Various timestepping methods

(\Rightarrow different orders of consistency)

- implicit Euler timestepping (6.1.23), first order
- Crank-Nicolson-method (6.1.25), order 2
- SDIRK-2 timestepping (\rightarrow Ex. 6.1.57), order 2
- Gauss-Radau-Runge-Kutta collocation methods with s stages, order $2s - 1$

Note: all methods $L(\pi)$ -stable (\rightarrow Def. 6.1.56), except for Crank-Nicolson-method.

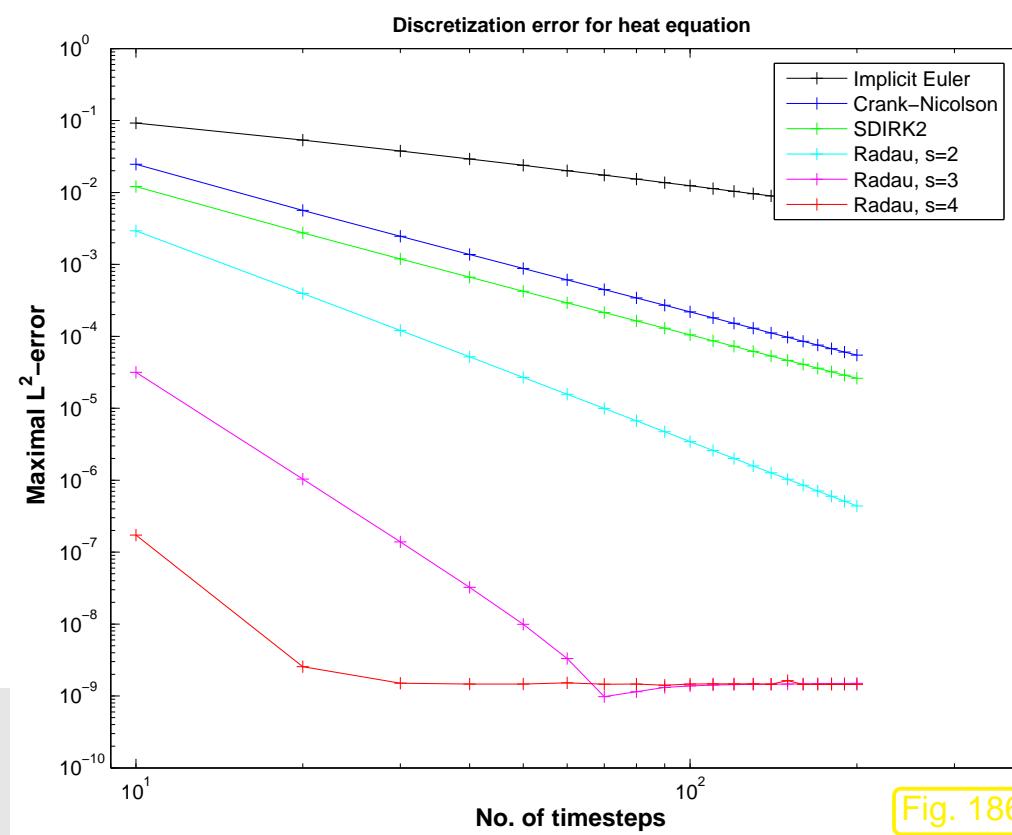


Fig. 186

Monitored: $\max_j \|u(t_j) - u_N^{(j)}\|_{L^2([0,1])}$ (evaluated by high order quadrature)



“Meta-theorem” 6.1.63 (Convergence of solutions of fully discrete parabolic evolution problems).

Assume that

- the solution of the parabolic IBVP (6.1.3)–(6.1.5) is “sufficiently smooth”,
- its spatial Galerkin finite element discretization relies on degree p Lagrangian finite elements (→ Sect. 3.4) on uniformly shape-regular families of meshes,
- timestepping is based on an $L(\pi)$ -stable single step method of order q with uniform timestep $\tau > 0$.

Then we can expect an asymptotic behavior of the total discretization error according to

$$\left(\tau \sum_{j=1}^M |u - u_N(\tau j)|_{H^1(\Omega)}^2 \right)^{\frac{1}{2}} \leq C(h_{\mathcal{M}}^p + \tau^q), \quad (6.1.64)$$

where $C > 0$ must not depend on $h_{\mathcal{M}}, \tau$.

This has been dubbed a “meta-theorem”, because quite a few technical assumptions on the exact solution and the methods have been omitted in its statement. Therefore it is not a mathematically rigorous statement of facts. More details in [15].

A message contained in (6.1.64):

$$\text{total discretization error} = \text{spatial error} + \text{temporal error}$$

Rem. 5.3.45 still applies: (6.1.64) does not give information about actual error, but only about the **trend** of the error, when discretization parameters h_M and τ are varied.

► Nevertheless, as in the case of the a priori error estimates of Sect. 5.3.5, we can draw conclusions about optimal refinement strategies in order to achieve prescribed *error reduction*.

As in Sect. 5.3.5 we make the **assumption** that the estimates (6.1.64) are sharp for all contributions to the total error and that the constants are the same (!)

$$\begin{aligned}\text{contribution of spatial error} &\approx Ch_M^p, \quad h_M \hat{=} \text{mesh width } (\rightarrow \text{Def. 5.2.3}), \\ \text{contribution of temporal error} &\approx C\tau^q, \quad \tau \hat{=} \text{timestep size}.\end{aligned}\tag{6.1.65}$$

This suggests the following change of h_M, τ in order to achieve *error reduction* by a factor of $\rho > 1$:

$$\begin{aligned}\text{reduce mesh width by factor } &\rho^{1/p} \quad (6.1.65) \implies \text{error reduction by } \rho > 1. \\ \text{reduce timestep by factor } &\rho^{1/q}\end{aligned}\tag{6.1.66}$$

Guideline: spatial and temporal resolution have to be adjusted in tandem

Remark 6.1.67 (Potential inefficiency of conditionally stable single step methods).

Terminology: A timestepping scheme is labelled **conditionally stable**, if blow-up can be avoided by using sufficient small timesteps (timestep constraint).

Now we can answer the question, why a stability induced timestep constraint like

$$\tau \leq O(h_{\mathcal{M}}^{-2}) \quad (6.1.68)$$

can render a single step method grossly inefficient for integrating semi-discrete parabolic IBVPs.

(??) ➤ in order to reduce the error by a fixed factor ρ one has to reduce both timestep and mesh-width by some other fixed factors (asymptotically). More concretely, for the timestep τ :

(6.1.66) ➤ *accuracy* requires reduction of τ by a factor $\rho^{1/q}$

(6.1.68) ➤ *stability* entails reduction of τ by a factor $(\rho^{1/p})^2 = \rho^{2/p}$.

$\frac{1}{q} < \frac{2}{p}$ ⇒ stability enforces smaller timestep than required by accuracy
⇒ timestepping is *inefficient!*

Faced with conditional stability (6.1.68), then for the sake of efficiency

use *high-order spatial discretization* combined with *low order timestepping*.

However, this may not be easy to achieve

- because high-order timestepping is much simpler than high-order spatial discretization,
- because limited spatial smoothness of exact solution (→ results of Sect. 5.4 apply!) may impose a limit on q in (6.1.64).

Concretely: 5th-order `ode45` timestepping ($q = 5$) ➤ use degree-10 Lagrangian FEM!

$$\frac{1}{q} = \frac{2}{p}$$

6.1

p. 649



6.2 Wave equations

Lemma 6.1.14 teaches that in the absence of time-dependent sources the rate of change of temperature will decay exponentially in the case of heat conduction.

Now we will encounter a class of evolution problems where temporal and spatial fluctuations will not be damped and will persist for good:

This will be the class of linear conservative wave propagation problems

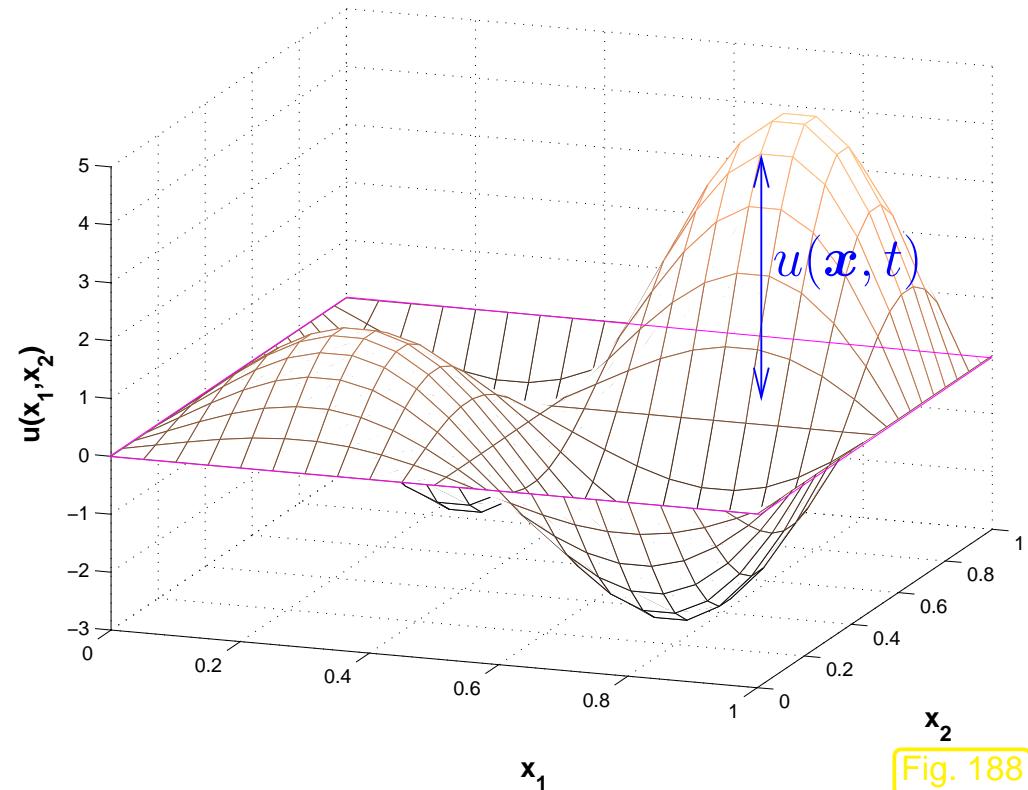
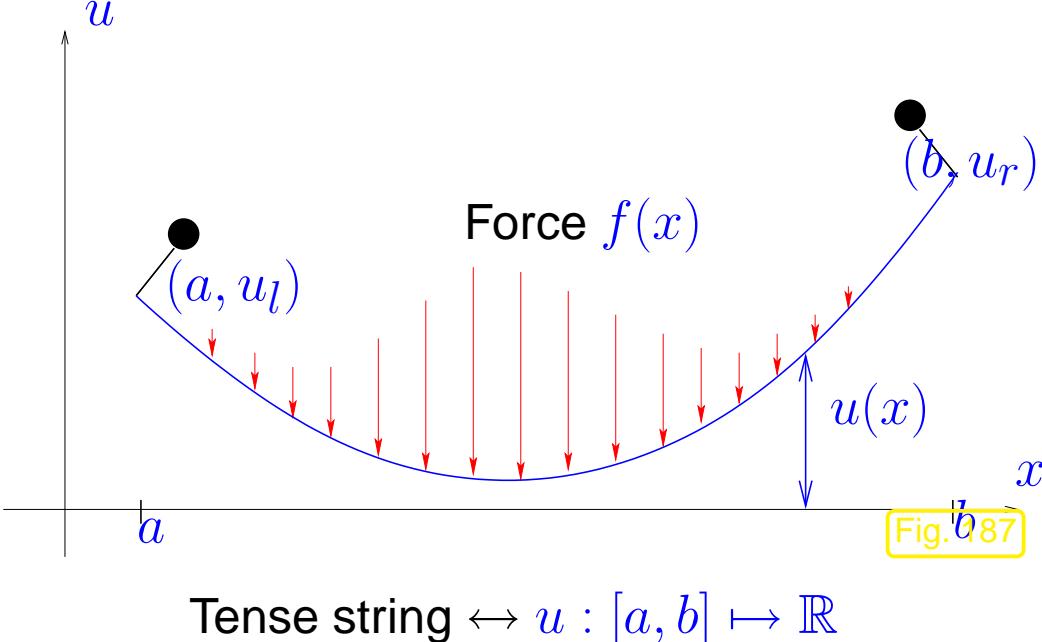
As before these initial-boundary value problems (IBVP) will be posed on a space time cylinder $\tilde{\Omega} := \Omega \times]0, T[\subset \mathbb{R}^{d+1}$ (\rightarrow Fig. 180), where $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, is a bounded spatial domain as introduced in the context of elliptic boundary value problems, see Sect. 2.1.1.

The unknown will be a function $u = (\mathbf{x}, t) : \tilde{\Omega} \mapsto \mathbb{R}$.

6.2.1 Vibrating membrane

Recall:

- Tense string model (\rightarrow Sect. 1.4), shape of string described by continuous displacement function $u : [a, b] \mapsto \mathbb{R}$, $u \in H^1([a, b])$.
- Taut membrane model (\rightarrow Sect. 2.1.1), shape of membrane given by displacement function $u : \Omega \mapsto \mathbb{R}$, $u \in H^1(\Omega)$, over base domain $\Omega \subset \mathbb{R}^2$.



In Sect. 2.1.3 we introduced the general variational formulation: with Dirichlet data (elevation of frame) given by $g \in C^0(\partial\Omega)$,

$$V := \{v \in H^1(\Omega) : v|_{\partial\Omega} = g\}$$

we seek

$$u \in V: \int_{\Omega} \sigma(\mathbf{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\mathbf{x} = \int_{\Omega} f(\mathbf{x}) v(\mathbf{x}) \, d\mathbf{x}, \quad \forall v \in H_0^1(\Omega), \quad (6.2.1)$$

where $f : \Omega \mapsto \mathbb{R} \hat{=} \text{ density of vertical force,}$

$\sigma : \Omega \mapsto \mathbb{R} \hat{=} \text{ uniformly positive stiffness coefficient (characteristic of material of the membrane).}$

Now we switch to a *dynamic setting*: we allow variation of displacement with time, $u = u(\mathbf{x}, t)$, the membrane is allowed to vibrate.

Recall (secondary school): **Newton's second law of motion** (law of inertia)

$$\begin{array}{ccc} \text{force} & \xrightarrow{\quad F = m a \quad} & \\ & = & \text{mass} \cdot \text{acceleration} \end{array} \quad (6.2.2)$$

$$(6.2.3)$$

Apply this in a local version (stated for densities) to membrane

$$\text{force density } f(\mathbf{x}, t) = \rho(\mathbf{x}) \cdot \frac{\partial^2 u}{\partial t^2}(\mathbf{x}, t) , \quad (6.2.4)$$

where $\bullet \rho : \Omega \mapsto \mathbb{R}^+ \hat{=} \text{uniformly positive mass density of membrane, } [\rho] = \text{kg m}^{-2}$,

- $\ddot{u} := \frac{\partial^2 u}{\partial t^2}$ $\hat{=}$ vertical acceleration (second temporal derivative of position).

Now, we assume that the force \underline{f} in (2.3.2) is due to inertia forces only and express these using (6.2.4):

$$(2.3.2) \quad \blacktriangleright \quad \int_{\Omega} \sigma(\mathbf{x}) \operatorname{grad} u(\mathbf{x}, t) \cdot \operatorname{grad} v(\mathbf{x}) \, d\mathbf{x} = - \int_{\Omega} \rho(\mathbf{x}) \cdot \frac{\partial^2 u}{\partial t^2}(\mathbf{x}, t) \, d\mathbf{x} \quad \forall v \in H_0^1(\Omega) .$$

Why the “–”-sign? Because, here the inertia force enters as a *reaction* force.

 Linear wave equation in variational form (Dirichlet boundary conditions):



$$u \in V(t): \quad \mathbf{m}(\ddot{u}, v) + \mathbf{a}(u, v) = 0 \quad \forall v \in V_0 . \quad (6.2.6)$$

where

$V(t) := \{v :]0, T[\mapsto H^1(\Omega) : v(\mathbf{x}, t) = g(\mathbf{x}, t) \text{ for } \mathbf{x} \in \partial\Omega, 0 < t < T\}$
 (with continuous time-dependent Dirichlet data $g : \partial\Omega \times]0, T[\mapsto \mathbb{R}$.)

Undo integration by parts by reverse application of Green's first formula Thm. 2.4.7:

$$(6.2.5) \Rightarrow \int_{\Omega} \left\{ \frac{\partial^2 u}{\partial t^2}(\mathbf{x}, t) - \operatorname{div}_{\mathbf{x}}(\sigma(\mathbf{x})(\operatorname{grad}_{\mathbf{x}} u)(\mathbf{x}, t)) \right\} v(\mathbf{x}) \, d\mathbf{x} = 0 \quad \forall v \in H_0^1(\Omega) . \quad (6.2.7)$$

Here it is indicated that the differential operators grad and div act on the spatial independent variable \mathbf{x} only. This will tacitly be assumed below.

Now appeal to the fundamental lemma of calculus of variations in higher dimensions Lemma 2.4.10.

$$(6.2.7) \quad \xrightarrow{\text{Lemma 2.4.10}} \frac{\partial^2 u}{\partial t^2} - \operatorname{div}(\sigma(\mathbf{x}) \operatorname{grad} u) = 0 \quad \text{in } \tilde{\Omega} . \quad (6.2.8)$$

(6.2.8) is called a (homogeneous) **wave equation**. A general wave equation is obtained, when an addition exciting vertical force density $f = f(\mathbf{x}, t)$ comes into play:

$$\frac{\partial^2 u}{\partial t^2} - \operatorname{div}(\sigma(\mathbf{x}) \operatorname{grad} u) = f(\mathbf{x}, t) \quad \text{in } \tilde{\Omega}. \quad (6.2.9)$$

The wave equations (6.2.8), (6.2.9) have to be supplemented by

- **spatial boundary conditions:** $v(\mathbf{x}, t) = g(\mathbf{x}, t)$ for $\mathbf{x} \in \partial\Omega, 0 < t < T,$
- **two initial conditions**

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad , \quad \frac{\partial u}{\partial t}(\mathbf{x}, 0) = v_0 \quad \text{for } \mathbf{x} \in \Omega ,$$

with initial data $u_0, v_0 \in H^1(\Omega)$, satisfying the compatibility conditions $u_0(\mathbf{x}) = g(\mathbf{x}, 0)$ for $\mathbf{x} \in \partial\Omega$.

(6.2.8) & boundary conditions & initial conditions = **hyperbolic evolution problem**

Hey, why do we need **two** initial conditions in contrast to the heat equation?

Remember that

- (6.2.8) is a *second-order equation* also in time (whereas the heat equation is merely first-order),
- for second order ODEs $\ddot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$ we need **two** initial conditions

$$\mathbf{y}(0) = \mathbf{y}_0 \quad \text{and} \quad \dot{\mathbf{y}}(0) = \mathbf{v}_0 , \quad (6.2.10)$$

in order to get a well-posed initial value problem, see [14, Rem. 11.1.7].

The physical meaning of the initial conditions (6.2.10) in the case of the membrane model is

- $u_0 \hat{=} \text{initial displacement of membrane, } u_0 \in H^1(\Omega) \text{ “continuous”,}$
- $v_0 \hat{=} \text{initial vertical velocity of membrane.}$

Remark 6.2.11 (Boundary conditions for wave equation).

Rem. 6.1.7 also applies to the wave equation (6.2.8):

On $\partial\Omega \times]0, T[$ we can impose any of the boundary conditions discussed in Sect. 2.6:

- Dirichlet boundary conditions $u(\mathbf{x}, t) = g(\mathbf{x}, t)$ (membrane attached to frame),
- Neumann boundary conditions $\mathbf{j}(\mathbf{x}, t) \cdot \mathbf{n} = 0$ (free boundary, Rem. 2.4.16)
- radiation boundary conditions $\mathbf{j}(\mathbf{x}, t) \cdot \mathbf{n} = \Psi(u(\mathbf{x}, t))$,

and any combination of these as discussed in Ex. 2.6.7, yet, *only one* of them at any part of $\partial\Omega \times]0, T[$, see Rem. 2.6.6.



Remark 6.2.12 (Wave equation as first order system in time).

Usual procedure [14, Rem. 11.1.7]: higher-order ODE can be converted into first-order ODEs by introducing derivatives as additional solution components. This approach also works for the second-order (in time) wave equation (6.2.8):

Additional unknown:

velocity

$$v(\boldsymbol{x}, t) = \frac{\partial u}{\partial t}(\boldsymbol{x}, t)$$

$$\frac{\partial^2 u}{\partial t^2} - \operatorname{div}(\sigma(\boldsymbol{x}) \operatorname{grad} u) = 0 \quad \Rightarrow \quad \begin{cases} \dot{u} = v, \\ \dot{v} = \operatorname{div}(\sigma(\boldsymbol{x}) \operatorname{grad} u) \end{cases} \quad \text{in } \tilde{\Omega} \quad (6.2.13)$$

with initial conditions

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}) , \quad v(\boldsymbol{x}, 0) = v_0(\boldsymbol{x}) \quad \text{for } \boldsymbol{x} \in \Omega . \quad (6.2.14)$$



6.2.2 Wave propagation

Constant coefficient wave equation for $d = 1$, $\Omega = \mathbb{R}$ (“Cauchy problem”)

$$c > 0: \quad \frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 , \quad u(x, 0) = u_0(x) , \quad \frac{\partial u}{\partial t}(x, 0) = v_0(x) , \quad x \in \mathbb{R} . \quad (6.2.15)$$

6.2

p. 659

Change of variables: $\xi = x + ct$, $\tau = x - ct$: $\tilde{u}(\xi, \tau) := u\left(\frac{\xi+\tau}{2}, \frac{\xi-\tau}{2c}\right)$. Applying the chain rule we immediately see

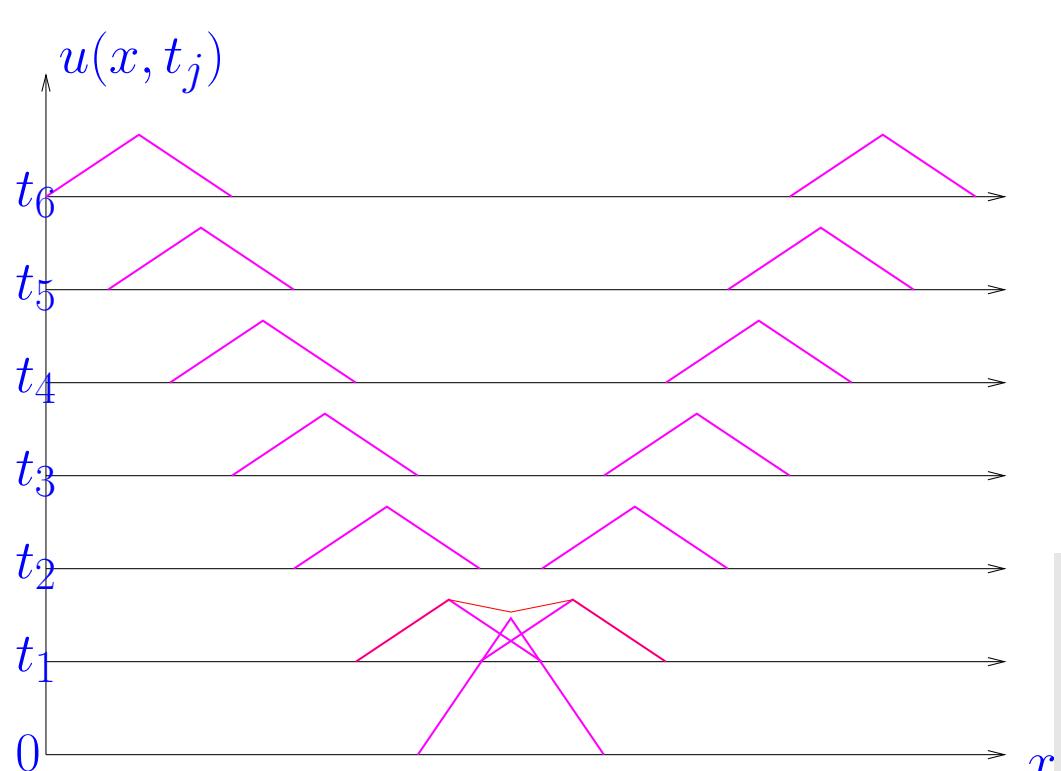
$$\textcolor{blue}{u \text{ satisfies (6.2.15)}} \quad \blacktriangleright \quad \frac{\partial^2 \tilde{u}}{\partial \xi \partial \tau} = 0 \quad \Rightarrow \quad \tilde{u}(\xi, \tau) = F(\xi) + G(\tau) ,$$

for any $F, G \in C^2(\mathbb{R})$!

 ← matching initial data

$$u(x, t) = \frac{1}{2}(u_0(x + ct) + u_0(x - ct)) + \frac{1}{2} \int_{x-ct}^{x+ct} v_0(s) \, ds . \quad (6.2.16)$$

(6.2.16) = d'Alembert solution of Cauchy problem (6.2.15).



$v_0 = 0 \rightarrow$ initial data u_0 travel with speed c in opposite directions

finite speed of propagation is typical feature of solutions of wave equations

Note: (6.2.16) meaningful even for discontinuous u_0, v_0 !
 ➡ “generalized solutions” !

finite speed of propagation ➡

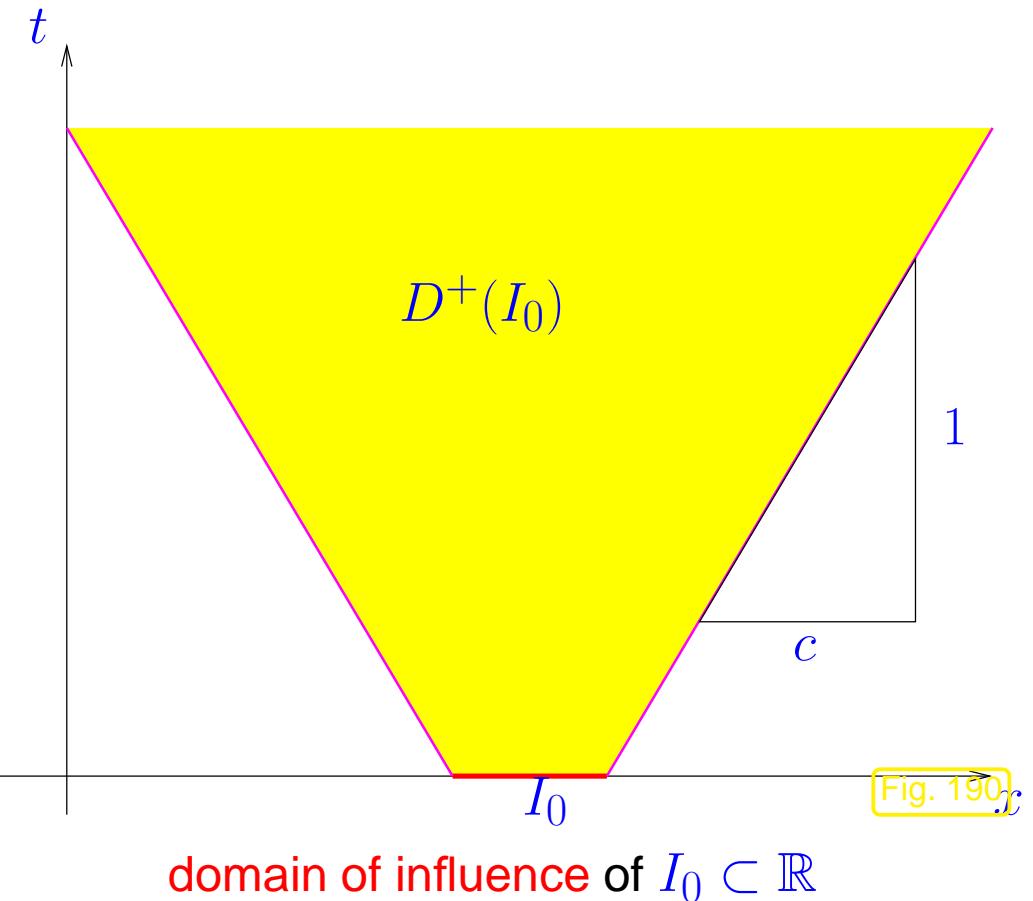
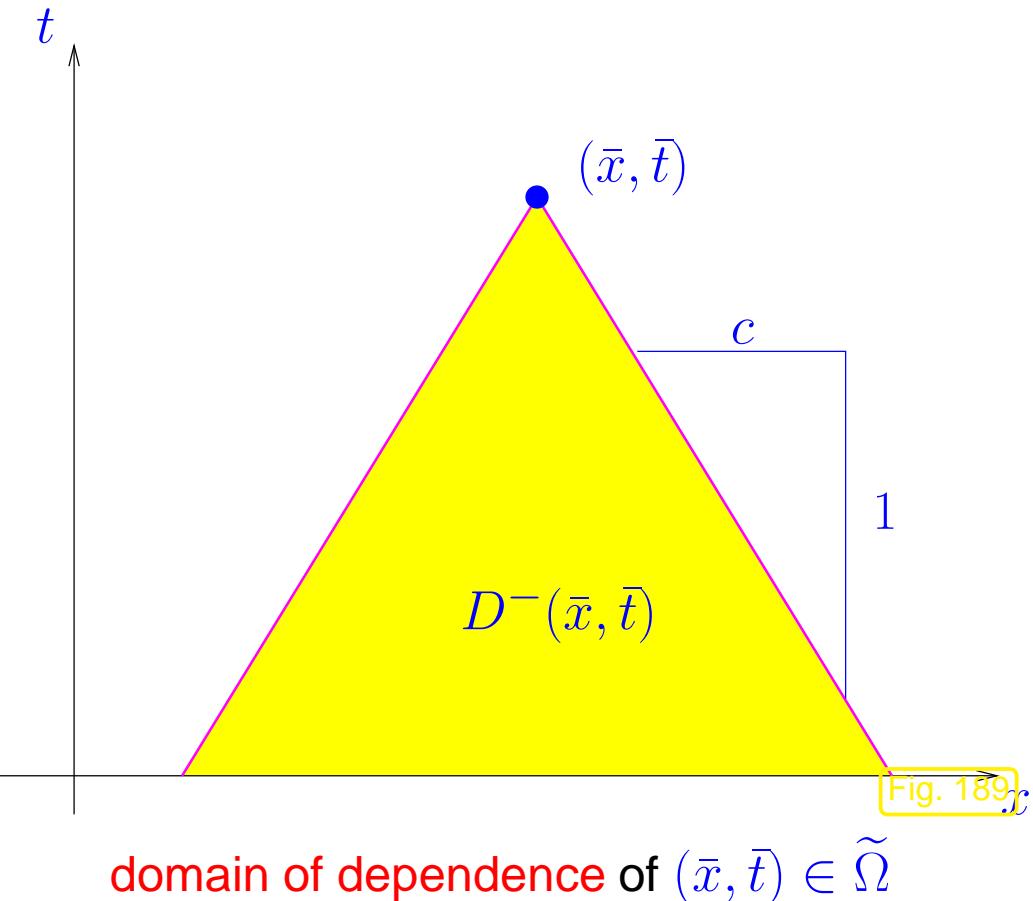
“point value” $u(\bar{x}, \bar{t})$, $(\bar{x}, \bar{t}) \in \tilde{\Omega}$, may not depend on initial values outside proper subdomain of Ω !

Example 6.2.17 (Domain of dependence/influence for 1D wave equation, constant coefficient case).

Consider $d = 1$, initial-boundary value problem (6.2.15) for wave equation:

$$c > 0: \frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 , \quad u(x, 0) = u_0(x) , \quad \frac{\partial u}{\partial t}(x, 0) = v_0(x) , \quad x \in \mathbb{R} . \quad (6.2.15)$$

Intuitive: from D'Alembert formula (6.2.16)



Domain of dependence: the value of the solution in (\bar{x}, \bar{t}) (●) will depend only on data in the yellow triangle in Fig. 189.

Domain of influence: initial data in I_0 will be relevant for the solution only in the yellow triangle in Fig. 190.



Theorem 6.2.18 (Domain of dependence for isotropic wave equation). $\rightarrow [10, 2.5, \text{Thm. 6}]$

Let $u : \tilde{\Omega} \mapsto \mathbb{R}$ be a (classical) solution of $\frac{\partial^2 u}{\partial t^2} - c\Delta u = 0$. Then

$$\left(|x - x_0| \geq R \Rightarrow u(x, 0) = 0 \right) \Rightarrow u(x, t) = 0 \quad , \text{ if } |x - x_0| \geq R + ct .$$

The solution formula (6.2.16) clearly indicates that in 1D and in the absence of boundary conditions the solution of the wave equation will persist undamped for all times.

This absence of damping corresponds to a *conservation of total energy*, which is a distinguishing feature of conservative wave propagation phenomena.

Now, we examine this for the model problem

$$u \in H_0^1(\Omega): \int_{\Omega} \rho(\boldsymbol{x}) \cdot \frac{\partial^2 u}{\partial t^2} v \, d\boldsymbol{x} + \int_{\Omega} \sigma(\boldsymbol{x}) \operatorname{grad} u \cdot \operatorname{grad} v \, d\boldsymbol{x} = 0 \quad \forall v \in H_0^1(\Omega) \quad (6.2.19)$$

$$u \in V_0: \quad m(\ddot{u}, v) + a(u, v) = 0 \quad \forall v \in V_0 \quad (6.2.20)$$

Here we do not include the case of non-homogeneous spatial Dirichlet boundary conditions through an affine trial space. This can always be taken into account by offset functions, see the remark after (6.1.5).

Theorem 6.2.21 (Energy conservation in wave propagation).

If $\mathbf{u} : \tilde{\Omega} \mapsto \mathbb{R}$ solves (6.2.20), then

$$t \mapsto \frac{1}{2}\mathbf{m}\left(\frac{\partial \mathbf{u}}{\partial t}, \frac{\partial \mathbf{u}}{\partial t}\right) + \frac{1}{2}\mathbf{a}(\mathbf{u}, \mathbf{u}) \equiv \text{const.}$$

kinetic energy *elastic (potential) energy, see (2.1.3)*

Proof. A “formal proof” boils down to a straightforward application of the product rule (\rightarrow Rem. 6.1.13) together with the symmetry of the bilinear forms \mathbf{m} and \mathbf{a} .

Introduce the **total energy**

$$E(t) := \frac{1}{2}\mathbf{m}\left(\frac{\partial \mathbf{u}}{\partial t}, \frac{\partial \mathbf{u}}{\partial t}\right) + \frac{1}{2}\mathbf{a}(\mathbf{u}, \mathbf{u}).$$

► $\frac{dE}{dt}(t) = \mathbf{m}(\ddot{\mathbf{u}}, \dot{\mathbf{u}}) + \mathbf{a}(\dot{\mathbf{u}}, \mathbf{u}) = 0 \quad \text{for solution } \mathbf{u} \text{ of (6.2.20)},$

because this is what we conclude from (6.2.20) for the special test function $v(\mathbf{x}) = \dot{\mathbf{u}}(\mathbf{x}, t)$ for any $t \in]0, T[$. □

6.2.3 Method of lines

The method of lines approach to the wave equation (6.2.19), (6.2.20) is exactly the same as for the heat equation, see Sect. 6.1.3.

Idea: Apply **Galerkin discretization** (\rightarrow Sect. 3.1) to abstract linear parabolic variational problem (6.1.11).

$$t \in]0, T[\mapsto u(t) \in V_0 : \quad \begin{cases} m\left(\frac{d^2u}{dt^2}(t), v\right) + a(u(t), v) = 0 & \forall v \in V_0 , \\ u(0) = u_0 \in V_0 , \quad \frac{du}{dt}(0) = v_0 \in V_0 . \end{cases} \quad (6.2.22)$$

1st step: replace V_0 with a finite dimensional subspace $V_{0,N}$, $N := \dim V_{0,N} < \infty$

► Discrete hyperbolic evolution problem

$$t \in]0, T[\mapsto u(t) \in V_{0,N} : \quad \begin{cases} m\left(\frac{d^2u_N}{dt^2}(t), v_N\right) + a(u_N(t), v_N) = 0 & \forall v_N \in V_{0,N} , \\ u_N(0) = \text{projection/interpolant of } u_0 \text{ in } V_{0,N} , \\ \frac{du_N}{dt}(0) = \text{projection/interpolant of } v_0 \text{ in } V_{0,N} . \end{cases} \quad (6.2.23)$$

2nd step: introduce (ordered) basis $\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\}$ of $V_{0,N}$

$$(6.2.23) \quad \Rightarrow \quad \begin{cases} \mathbf{M} \left\{ \frac{d^2}{dt^2} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) = 0 & \text{for } 0 < t < T, \\ \vec{\mu}(0) = \vec{\mu}_0 \quad , \quad \frac{d\vec{\mu}}{dt}(0) = \vec{\nu}_0 . \end{cases} \quad (6.2.24)$$

- ▷ s.p.d. stiffness matrix $\mathbf{A} \in \mathbb{R}^{N,N}$, $(\mathbf{A})_{ij} := \mathbf{a}(b_N^j, b_N^i)$ (independent of time),
- ▷ s.p.d. mass matrix $\mathbf{M} \in \mathbb{R}^{N,N}$, $(\mathbf{M})_{ij} := \mathbf{m}(b_N^j, b_N^i)$ (independent of time),
- ▷ source (load) vector $\vec{\varphi}(t) \in \mathbb{R}^N$, $(\vec{\varphi}(t))_i := \ell(t)(b_N^i)$ (time-dependent),
- ▷ $\vec{\mu}_0 \hat{=} \text{coefficient vector of a projection of } u_0 \text{ onto } V_{0,N}$.
- ▷ $\vec{\nu}_0 \hat{=} \text{coefficient vector of a projection of } v_0 \text{ onto } V_{0,N}$.

Note:

(6.2.24) is a **2nd-order** ordinary differential equation (ODE) for $t \mapsto \vec{\mu}(t) \in \mathbb{R}^N$

Remark 6.2.25 (First-order semidiscrete hyperbolic evolution problem).

Completely analogous to Rem. 6.2.12:

$$\mathbf{M} \left\{ \frac{d^2}{dt^2} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) = 0$$



← auxiliary unknown $\nu = \dot{\mu}$

$$\begin{cases} \frac{d}{dt} \vec{\mu}(t) = \vec{\nu}(t) , \\ \frac{d}{dt} \vec{\nu}(t) = -\mathbf{A} \vec{\mu}(t) , \end{cases}, \quad 0 < t < T . \quad (6.2.26)$$

with intial conditions

$$\vec{\mu}(0) = \vec{\mu}_0 , \quad \vec{\nu}(0) = \vec{\nu}_0 . \quad (6.2.27)$$

6.2.4 Timestepping

The method of lines approach gives us the semi-discrete hyperbolic evolution problem = 2nd-order ODE:

$$\mathbf{M} \left\{ \frac{d^2}{dt^2} \vec{\mu}(t) \right\} + \mathbf{A} \vec{\mu}(t) = 0 , \quad \vec{\mu}(0) = \vec{\mu}_0 , \quad \frac{d\vec{\mu}}{dt}(0) = \vec{\eta}_0 . \quad (6.2.28)$$

Key features of (6.2.28) \rightarrow to be respected “approximately” by timestepping:

- **reversibility:** (6.2.28) invariant under time-reversal $t \leftarrow -t$
- **energy conservation, cf. Thm. 6.2.21:** $E_N(t) := \frac{1}{2} \frac{d\vec{\mu}}{dt} \cdot \mathbf{M} \frac{d\vec{\mu}}{dt} + \frac{1}{2} \vec{\mu} \cdot \mathbf{A} \vec{\mu} = \text{const}$

Example 6.2.29 (Euler timestepping for 1st-order form of semi-discrete wave equation).

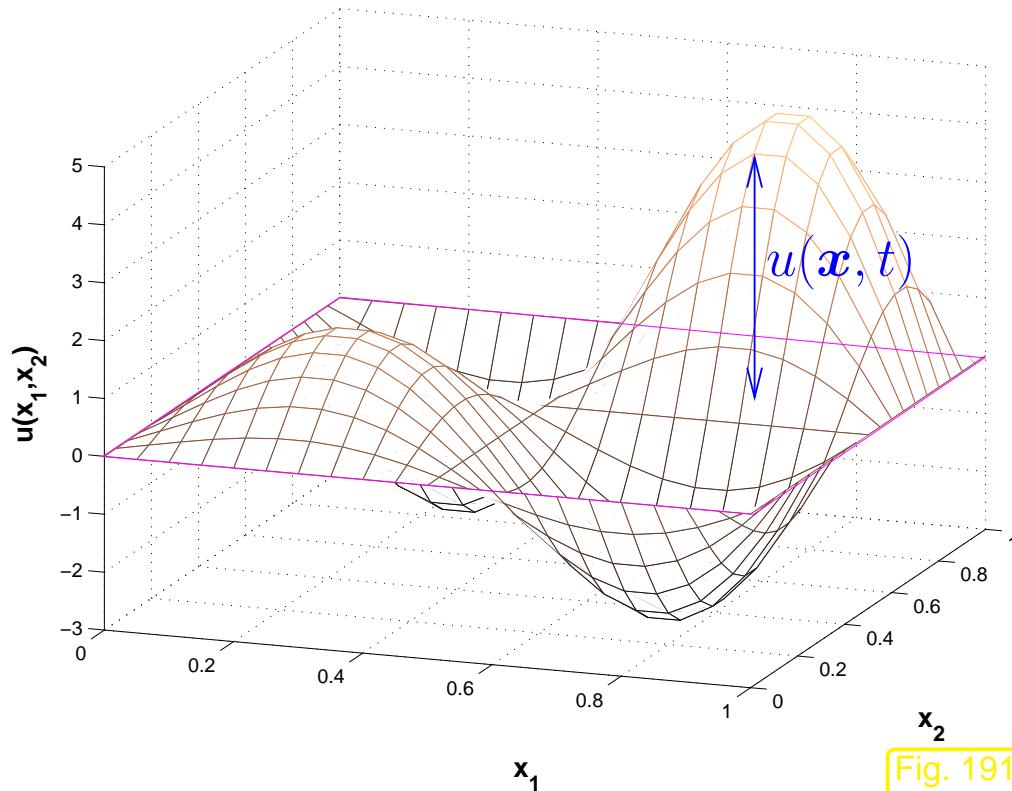


Fig. 191

Model problem: wave propagation on a square membrane

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} - \Delta u &= 0 \quad \text{on }]0, 1[^2 \times]0, 1[, \\ u(\mathbf{x}, t) &= 0 \quad \text{on } \partial\Omega \times]0, T[, \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}) \quad , \quad \frac{\partial u}{\partial t}(\mathbf{x}, 0) = 0 . \end{aligned}$$

- Initial data $u_0(\mathbf{x}) = \max\{0, \frac{1}{5} - \|\mathbf{x}\|\}$, $v_0(\mathbf{x}) = 0$,
- $\mathcal{M} \doteq$ “structured triangular tensor product mesh”, see Fig. 121, n squares in each direction,
- linear finite element space $V_{N,0} = \mathcal{S}_{1,0}^0(\mathcal{M})$, $N := \dim \mathcal{S}_{1,0}^0(\mathcal{M}) = (n-1)^2$,
- All local computations (\rightarrow Sect. 3.5.4) rely on 3-point vertex based local quadrature formula “2D trapezoidal rule” (3.2.13). More explanations will be given in Rem. 6.2.34 below.

- $\mathbf{A} = N \times N$ Poisson matrix, see (4.1.2), scaled with $h := n^{-1}$,
 - mass matrix $\mathbf{M} = h\mathbf{I}$, thanks to quadrature formula, see Rem. 6.2.34.

Timestepping: implicit and explicit Euler method (\rightarrow Ex. 6.1.21, [14, Sect. 11.2]) for 1st-order ODE (6.2.26), timestep $\tau > 0$:

$\vec{\mu}^{(j)} - \vec{\mu}^{(j-1)} = \tau \vec{\nu}^{(j-1)},$ $\mathbf{M}(\vec{\nu}^{(j)} - \vec{\nu}^{(j-1)}) = -\tau \mathbf{A} \vec{\mu}^{(j-1)}.$ <p>explicit Euler</p>	$\vec{\mu}^{(j)} - \vec{\mu}^{(j-1)} = \tau \vec{\nu}^{(j)},$ $\mathbf{M}(\vec{\nu}^{(j)} - \vec{\nu}^{(j-1)}) = -\tau \mathbf{A} \vec{\mu}^{(j)}.$ <p>implicit Euler</p>
--	--

Monitored: behavior of (discrete) kinetic, potential, and total energy

$$E_{\text{kin}}^{(j)} = (\vec{\nu}^{(j)})^T \mathbf{M} \vec{\nu}^{(j)} \quad , \quad E_{\text{pot}}^{(j)} = (\vec{\mu}^{(j)})^T \mathbf{A} \vec{\mu}^{(j)} \quad , \quad j = 0, 1, \dots .$$

Explicit Euler timestepping:

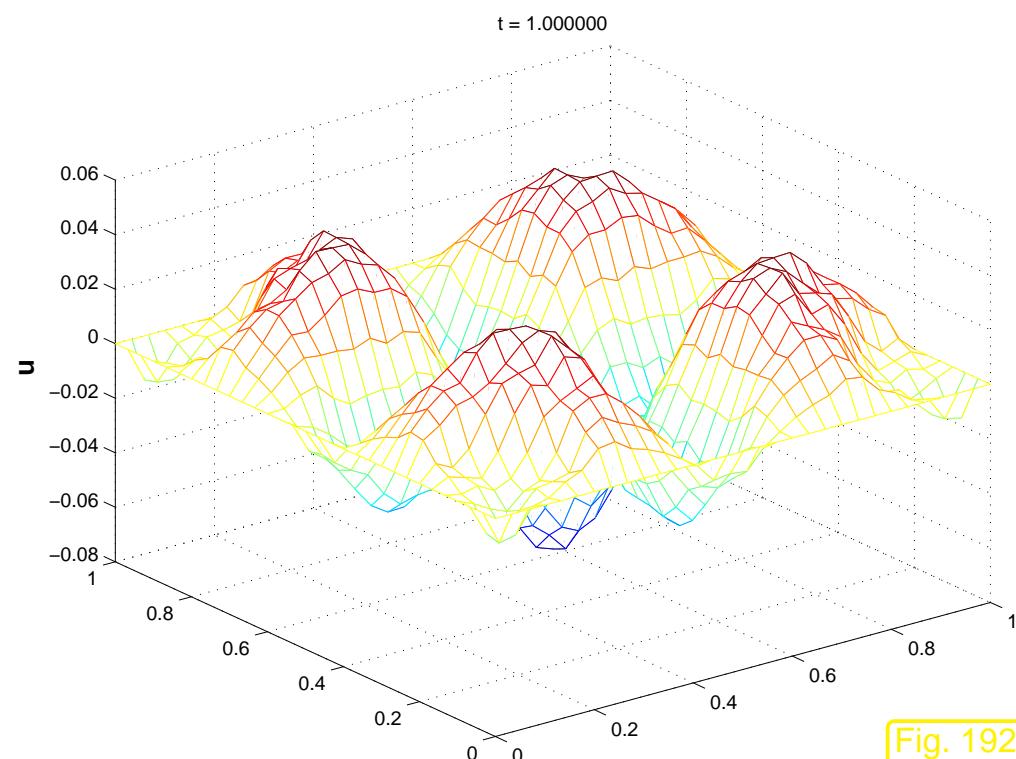


Fig. 192

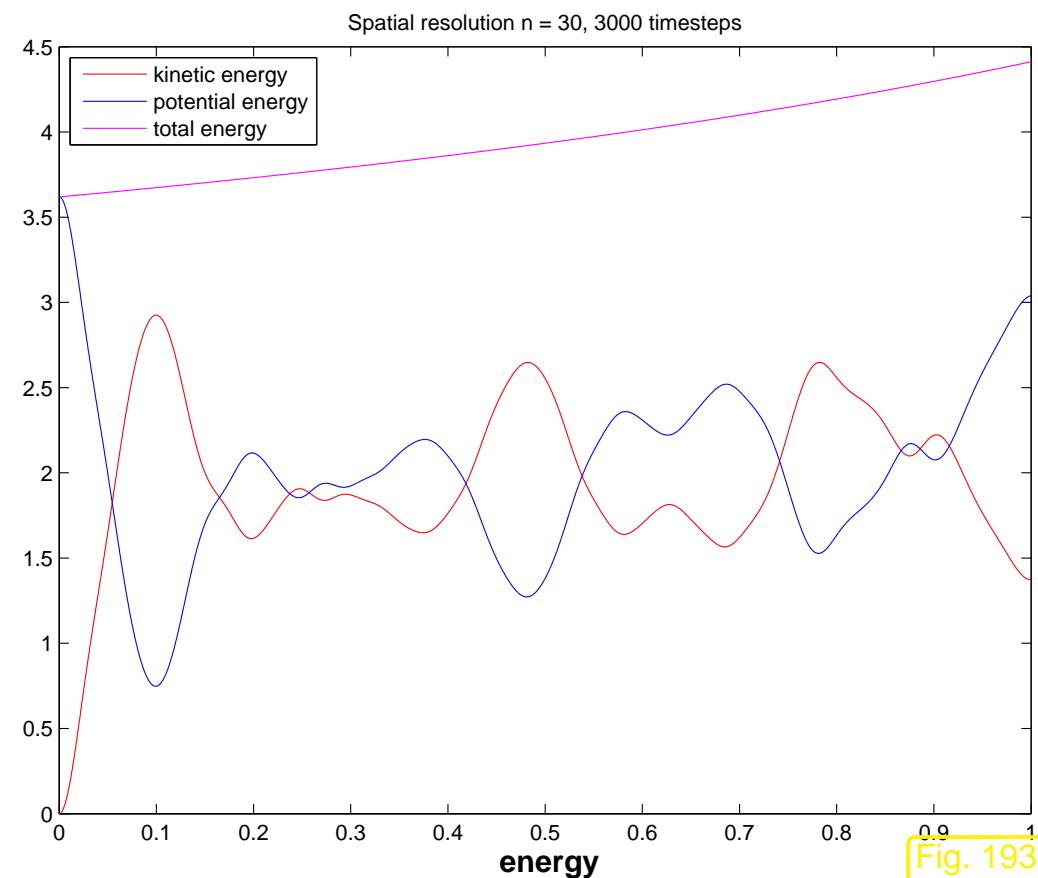


Fig. 193

Implicit Euler timestepping:

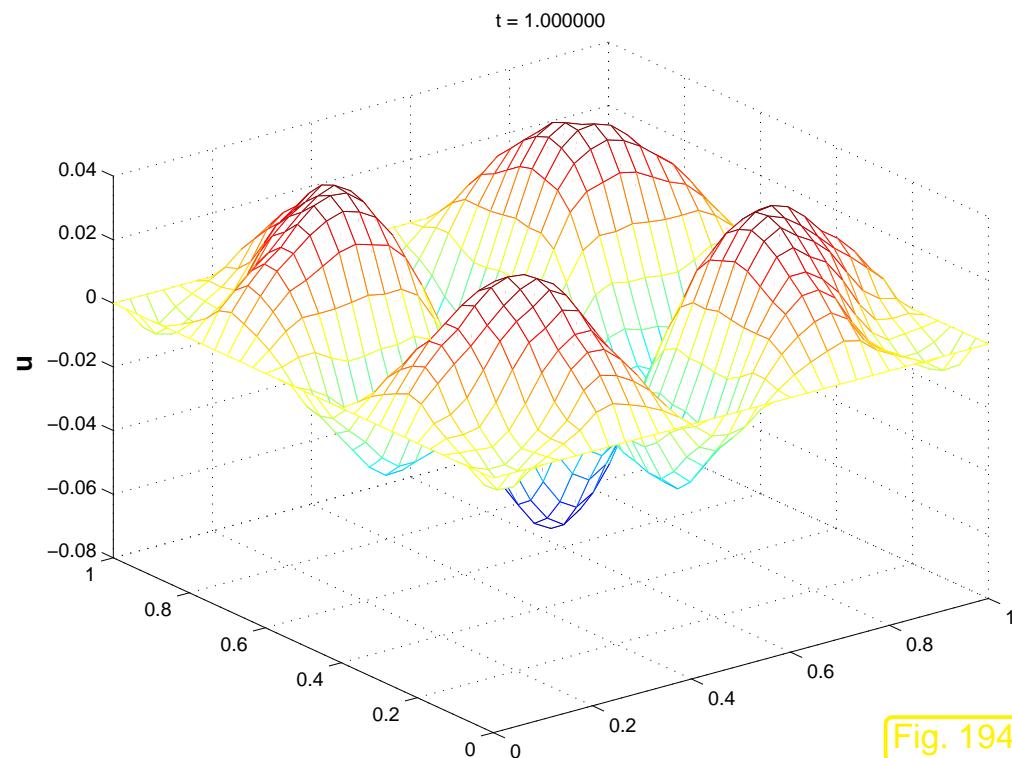


Fig. 194

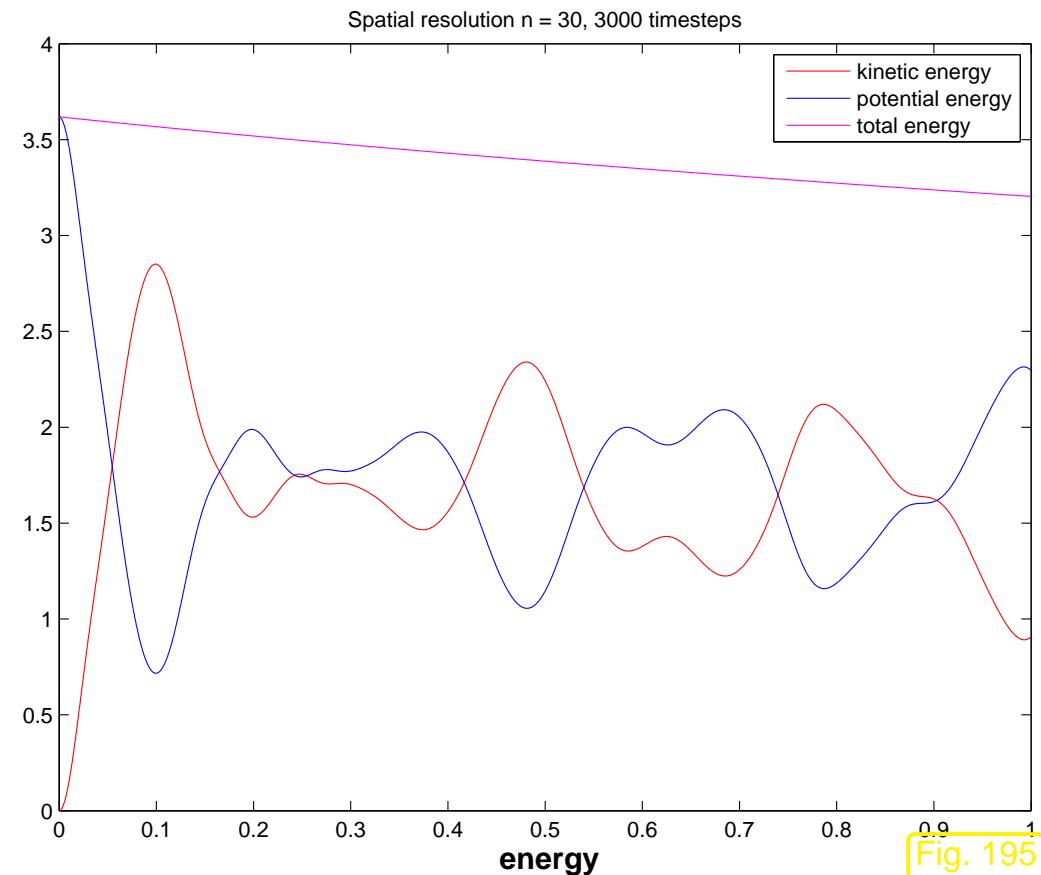


Fig. 195

Observation: neither method conserves energy,

- ☞ explicit Euler timestepping > steady increase of total energy
- ☞ implicit Euler timestepping > steady decrease of total energy

Ex. 6.2.29 ➤ Euler methods violate energy conservation!

(The same is true of all explicit Runge-Kutta methods, which lead to an increase of the total energy over time, and L(π)-stable implicit Runge-Kutta method, which make the total energy decay.)

Let us try another simple idea for the 2nd-order ODE (6.2.24):

Replace $\frac{d^2}{dt^2}\vec{\mu}$ with symmetric difference quotient (1.5.92)

$$\mathbf{M} \left\{ \frac{d^2}{dt^2}\vec{\mu}(t) \right\} + \mathbf{A}\vec{\mu}(t) = 0 \quad (6.2.28)$$

$$\mathbf{M} \frac{\vec{\mu}^{(j+1)} - 2\vec{\mu}^{(j)} + \vec{\mu}^{(j-1)}}{\tau^2} = -\mathbf{A}\vec{\mu}^{(j)}, \quad j = 0, 1, \dots \quad (6.2.30)$$

This is a two-step method, the Störmer scheme/explicit trapezoidal rule

6.2

By Taylor expansion:

Störmer scheme is a 2nd-order method

However, from where do we get $\vec{\mu}^{(-1)}$? Two-step methods need to be kick-started by a *special initial step*: This is constructed by approximating the second initial condition by a symmetric difference quotient:

$$\frac{d}{dt}\vec{\mu}(0) = \vec{\nu}_0 \quad \Rightarrow \quad \frac{\vec{\mu}^{(1)} - \vec{\mu}^{(-1)}}{\tau} = \vec{\nu}_0 . \quad (6.2.31)$$

Example 6.2.32 (Leapfrog timestepping).

For the semi-discrete wave equation we again consider the explicit trapezoidal rule (Störmer scheme):

$$\mathbf{M} \frac{\vec{\mu}^{(j+1)} - 2\vec{\mu}^{(j)} + \vec{\mu}^{(j-1)}}{\tau^2} = -\mathbf{A}\vec{\mu}^{(j)}, \quad j = 1, \dots . \quad (6.2.30)$$

Inspired by Rem. 6.2.25 we introduce the auxiliary variable

$$\vec{\nu}^{(j+1/2)} := \frac{\vec{\mu}^{(j+1)} - \vec{\mu}^{(j)}}{\tau},$$

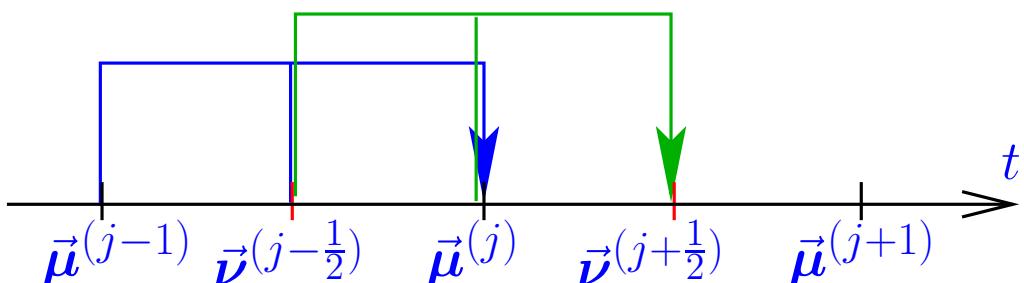
which can be read as an approximation of the velocity $v := \dot{u}$.

This leads to a timestepping scheme, which is *algebraically equivalent* to the explicit trapezoidal rule:

leapfrog timestepping (with uniform timestep $\tau > 0$):

$$\boxed{\begin{aligned} M \frac{\vec{\nu}^{(j+\frac{1}{2})} - \vec{\nu}^{(j-\frac{1}{2})}}{\tau} &= -A \vec{\mu}^{(j)}, \\ \frac{\vec{\mu}^{(j+1)} - \vec{\mu}^{(j)}}{\tau} &= \vec{\nu}^{(j+\frac{1}{2})}, \end{aligned} \quad j = 0, 1, \dots}, \quad (6.2.33)$$

+ initial step $\vec{\nu}^{(-\frac{1}{2})} + \vec{\nu}^{(\frac{1}{2})} = 2\vec{\nu}_0$.



work per step:

$1 \times$ evaluation $A \times$ vector,
 $1 \times$ solution of linear system for M

◇

Remark 6.2.34 (Mass lumping).

Required in each step of leapfrog timestepping: solution of linear system of equations with (large sparse) system matrix $\mathbf{M} \in \mathbb{R}^{N,N}$ ➤ **expensive!**

Trick for (bi-)linear finite element Galerkin discretization: $V_{0,N} \subset \mathcal{S}_1^0(\mathcal{M})$:

use *vertex based local quadrature rule*

(e.g. “2D trapezoidal rule” (3.2.13) on triangular mesh)

$$\int_K f(\mathbf{x}) d\mathbf{x} \approx \frac{|K|}{\#\mathcal{V}(K)} \sum_{\mathbf{p} \in \mathcal{V}(K)} f(\mathbf{p}), \quad \mathcal{V}(K) := \text{set of vertices of } K.$$

(For a comprehensive discussion of local quadrature rules see Sect. 3.5.4)

- Mass matrix \mathbf{M} will become a *diagonal* matrix (due to defining equation (3.2.2) for nodal basis functions, which are associated with nodes of the mesh).

This so-called mass lumping trick was used in the finite element discretization of Ex. 6.2.29.

6.2



p. 677

Example 6.2.35 (Energy conservation for leapfrog).

Model problem and discretization as in Ex. 6.2.29.

Leapfrog timestepping with constant timestep size $\tau = 0.01$

Code 6.2.36: Computing behavior of energies for Störmer timestepping

```
1 function lfen(n,m)
2 % leapfrom timestepping for 2D wave equation, computation of energies
3 % n: spatial resolution (no. of cells in one direction)
4 % m: number of timesteps
5
6 % Assemble stiffness matrix, see Sect. 4.1, (4.1.2)
7 N = (n-1)^2; h = 1/n; A = gallery('poisson',n-1)/(h*h);
8
9 % initial displacement  $u_0(\mathbf{x}) = \max\{0, \frac{1}{5} - \|\mathbf{x}\|\}$ 
10 [X,Y] = meshgrid(0:h:1,0:h:1);
```

```

11 U0 = 0.2-sqrt((X-0.5).^2+(Y-0.5).^2);
12 U0(find(U0 < 0)) = 0.0;
13 u0 = reshape(U0(2:end-1,2:end-1),N,1);
14 v0 = zeros(N,1); % initial velocity
15
16 % loop for Störmer timestepping, see (6.2.30)
17 tau = 1/m; % uniform timestep size
18 u = u0+tau*v0-0.5*tau^2*A*u0; % special initial step
19 u_old = u0;
20 [pen,ken] = geten(A,tau,u0,u); % compute potential and kinetic energy
21 E = [0.5*tau,pen,ken,pen+ken];
22 for k=1:m-1
23     u_new = -(tau^2)*(A*u) + 2*u - u_old;
24     [pen,ken] = geten(A,tau,u,u_new);
25     E = [E;(k+0.5)*tau,pen,ken,pen+ken];
26     u_old = u; u = u_new;
27 end
28
29 figure('name','Leapfrog energies');
30 plot(E(:,1),E(:,3),'r-',E(:,1),E(:,2),'b-',E(:,1),E(:,4),'m-');
31 xlabel('{\bf time } t', 'fontsize',14);
32 ylabel('{\bf energies}', 'fontsize',14);
33 legend('kinetic energy', 'potential energy', 'total'

```

```

energy' 'location' 'north'
34 title sprintf 'Spatial resolution n = %i, %i timesteps'
35
36 print '-depsc' sprintf '../.../.../.../rw/Slides/NPDEPics/leapfrogend.

```

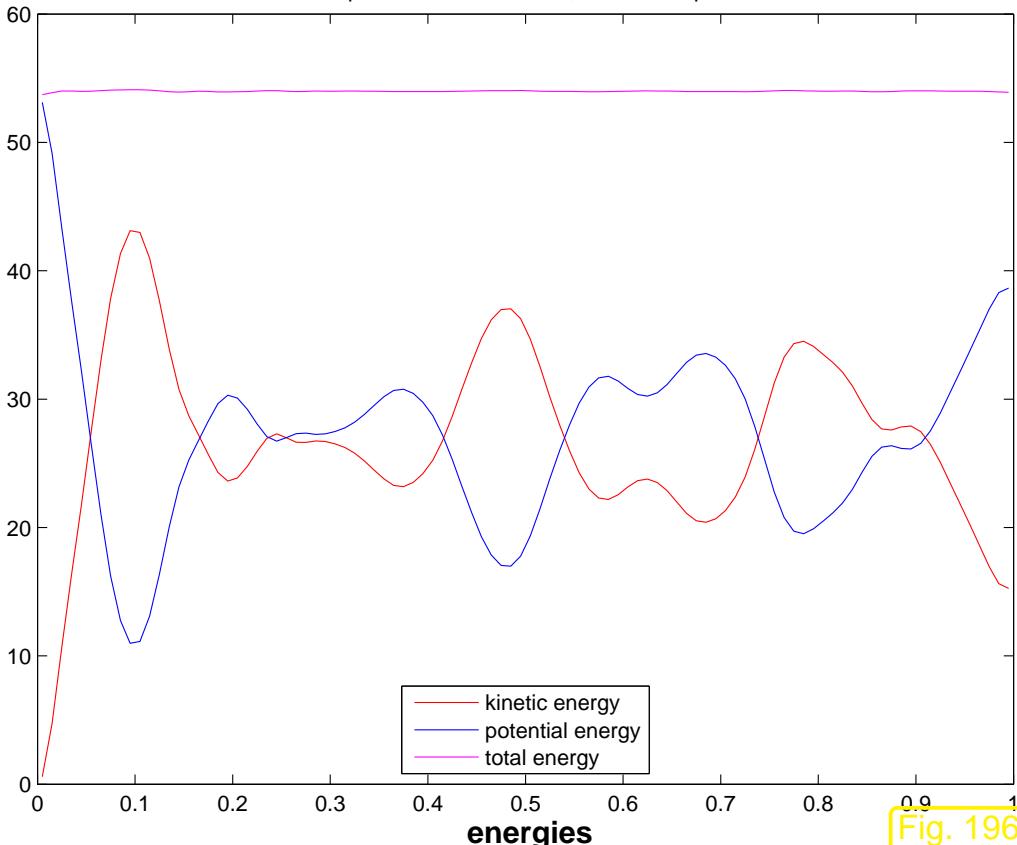
Code 6.2.37: Computing potential and kinetic energiy for Störmer timestepping

```

1 function [pen,ken] = geten(A,ts,u_old,u_new)
2 % Compute the current approximate potential and kinetic energies for u_old
3 % and u_new from Sörmer timestepping
4 %  $E_{\text{kin}}^{(j)} = (\vec{\mu}^{(j)} - \vec{\mu}^{(j-1)})^T \mathbf{M} (\vec{\mu}^{(j)} - \vec{\mu}^{(j-1)})$  ,  $E_{\text{pot}}^{(j)} = \frac{1}{4} (\vec{\mu}^{(j)} + \vec{\mu}^{(j-1)})^T \mathbf{A} (\vec{\mu}^{(j)} + \vec{\mu}^{(j-1)})$  ,  $j = 0, 1, \dots$ .
5 meanv = 0.5*(u_old+u_new); pen = 0.5*dot(meanv,A*meanv); % potential
energy
6 dtemp = (u_new-u_old)/ts; ken = 0.5*dot(dtemp,dtemp); % kinetic
energy

```

Spatial resolution $n = 30$, 100 timesteps



Leapfrog is (nearly) energy conserving
(no energy drift, only small oscillations)

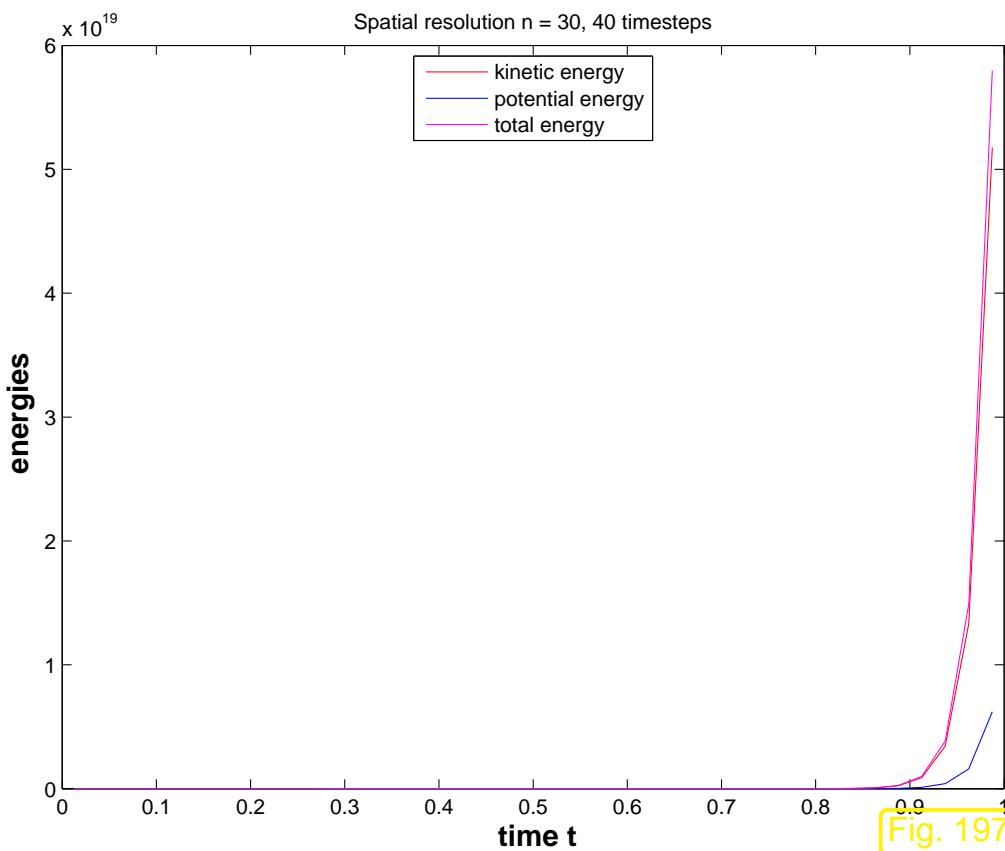
This behavior is explained by the deep mathematical theories of **symplectic integrators**, see [13].

Fig. 196



6.2.5 CFL-condition

Example 6.2.38 (Blow-up for leapfrog timestepping).



▷ Ex. 6.2.35 repeated with $\tau = 0.04$

Observation:

Leapfrog suffers a **blow-up**: exponential increase of energies!

A similar behavior is observed with the explicit Euler scheme for the semi-discrete heat equation, in case the timestep constraint is violated, see Sect. 6.1.4.2.

Fig. 197



➤ (as in Sect. 6.1.4.2) Stability analysis of leapfrog timestepping based on **diagonalization**:

$$\exists \text{ orthogonal } \mathbf{T} \in \mathbb{R}^{N,N}: \quad \mathbf{T}^\top \mathbf{M}^{-1/2} \mathbf{A} \mathbf{M}^{-1/2} \mathbf{T} = \mathbf{D} := \text{diag}(\lambda_1, \dots, \lambda_N).$$

where the $\lambda_i > 0$ are *generalized eigenvalues* for $\mathbf{A}\vec{\xi} = \lambda\mathbf{M}\vec{\xi}$ ➤ $\lambda_i \geq \gamma$ for all i (γ is the constant introduced in (6.1.12)).

Next, apply transformation $\vec{\eta} := \mathbf{T}^T \mathbf{M}^{1/2} \vec{\mu}$ to the 2-step formulation (6.2.30)

$$(6.2.30) \quad \vec{\eta} := \mathbf{T}^T \mathbf{M}^{1/2} \vec{\mu} \quad \vec{\eta}^{(j+1)} - 2\vec{\eta}^{(j)} + \vec{\eta}^{(j-1)} = -\tau^2 \mathbf{D} \vec{\eta}^{(j)} .$$

Again, we have achieved a complete decoupling of the timestepping for the eigencomponents.

$$\eta_i^{(j+1)} - 2\eta_i^{(j)} + \eta_i^{(j-1)} = -\tau^2 \lambda_i \eta_i^{(j)}, \quad i = 1, \dots, N, \quad j = 1, 2, \dots . \quad (6.2.39)$$

In fact, (6.2.39) is what we end up with then applying Störmers scheme to the *scalar* linear 2nd-order ODE $\ddot{\eta}_i = -\lambda_i \eta_i$. In a sense, the commuting diagram (6.1.54) remains true for 2-step methods and second-order ODEs.

(6.2.39) is a **linear two-step recurrence** formula for the sequences $(\eta_i^{(j)})_j$.

Try:

$$\eta_i^{(j)} = \xi^j \quad \text{for some } \xi \in \mathbb{C} \setminus \{0\}$$

Plug this into (6.2.39)

► $\xi^2 - 2\xi + 1 = -\tau^2 \lambda_i \xi \Leftrightarrow \xi^2 - (2 - \tau^2 \lambda_i) \xi + 1 = 0 .$

\Rightarrow two solutions $\xi_{\pm} = \frac{1}{2} \left(2 - \tau^2 \lambda_i \pm \sqrt{(2 - \tau^2 \lambda_i)^2 - 4} \right) .$

We can get a blow-up of some solutions of (6.2.39), if $|\xi_+| > 1$ or $|\xi_-| > 1$. From secondary school we know Vieta's formula

$$\xi_+ \cdot \xi_- = 1 \Rightarrow \begin{cases} \xi_{\pm} \in \mathbb{R} \text{ and } \xi_+ \neq \xi_- \Rightarrow |\xi_+| > 1 \text{ or } |\xi_-| > 1 \\ \xi_- = \xi_+^* \Rightarrow |\xi_-| = |\xi_+| = 1 \end{cases},$$

where ξ_+^* designates complex conjugation. So the recurrence (6.2.39) has only bounded solution, if and only if

$$\text{discriminant } D := (2 - \tau^2 \lambda_i)^2 - 4 \leq 0 \Leftrightarrow \boxed{\tau < \frac{2}{\sqrt{\lambda_i}}}. \quad (6.2.40)$$

←→

stability induced timestep constraint for leapfrog timestepping

Special setting: spatial finite element Galerkin discretization based on fixed degree Lagrangian finite element spaces (→ Sect. 3.4), meshes created by uniform regular refinement.

Under these conditions a generalization of Lemma 6.1.48 shows

Stability of leapfrog timestepping entails $\tau \leq O(h_M)$ for $h_M \rightarrow 0$

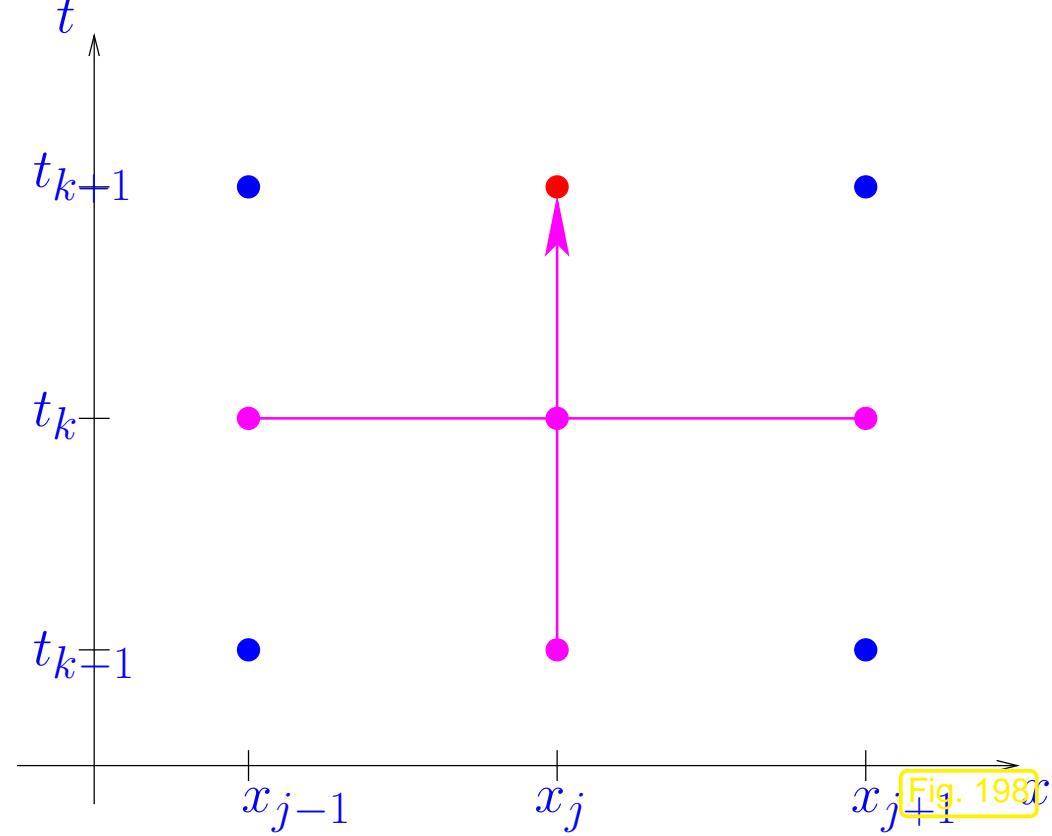
Remark 6.2.41 (Geometric interpretation of CFL condition in 1D).

Setting:

- 1D wave equation, (spatial) boundary conditions ignored (“Cauchy problem”),

$$c > 0: \frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \quad , \quad u(x, 0) = u_0(x) , \quad \frac{\partial u}{\partial t}(x, 0) = v_0(x) , \quad x \in \mathbb{R} . \quad (6.2.15)$$

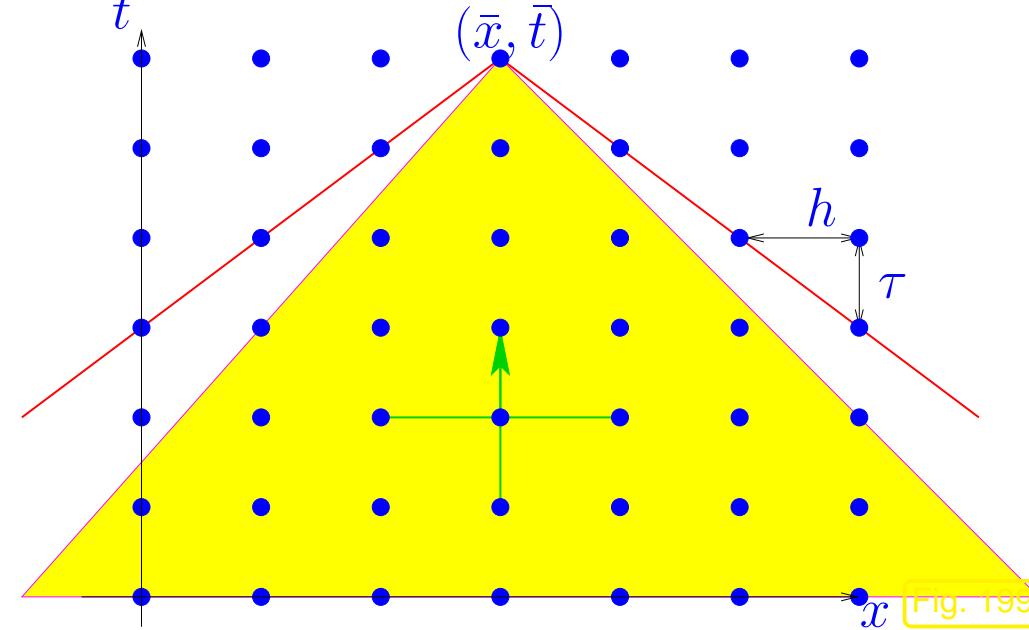
- Linear finite element Galerkin discretization on equidistant spatial mesh $\mathcal{M} := \{[x_{j-1}, x_j] : j \in \mathbb{Z}\}$, $x_j := h j$ (meshwidth h), see Sect. 1.5.1.2.
- Mass lumping for computation of mass matrix, which will become $h \cdot \mathbf{I}$, see Rem 6.2.34.
- Timestepping by Sörmer scheme (6.2.30) with constant timestep $\tau > 0$.



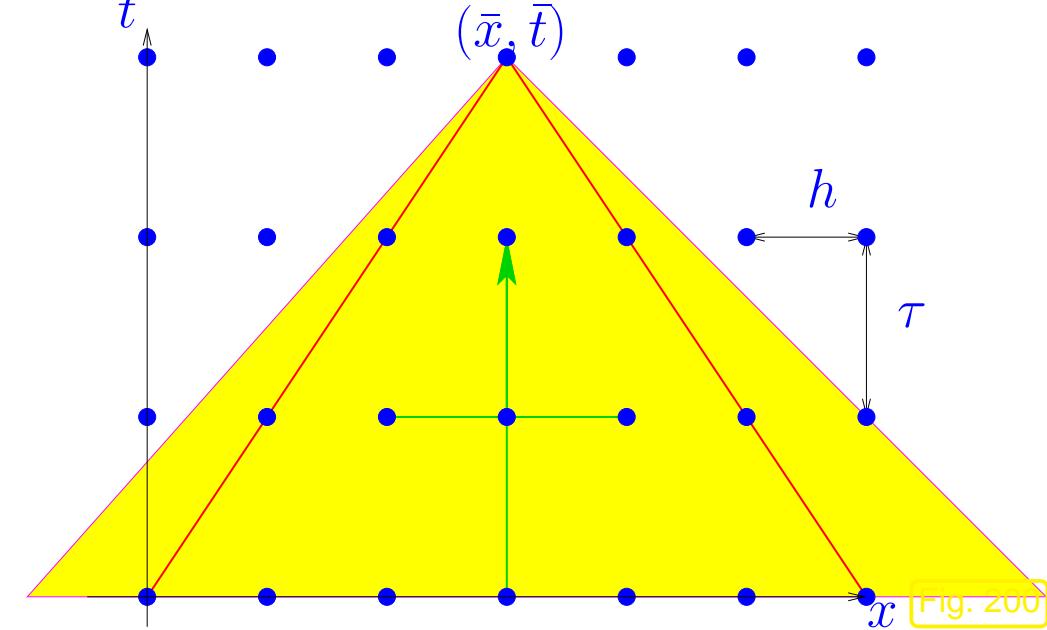
▷ flow of information in one step of Störmer scheme

Since the method is a two-step method, information from time-slices t_k and t_{k-1} is needed.

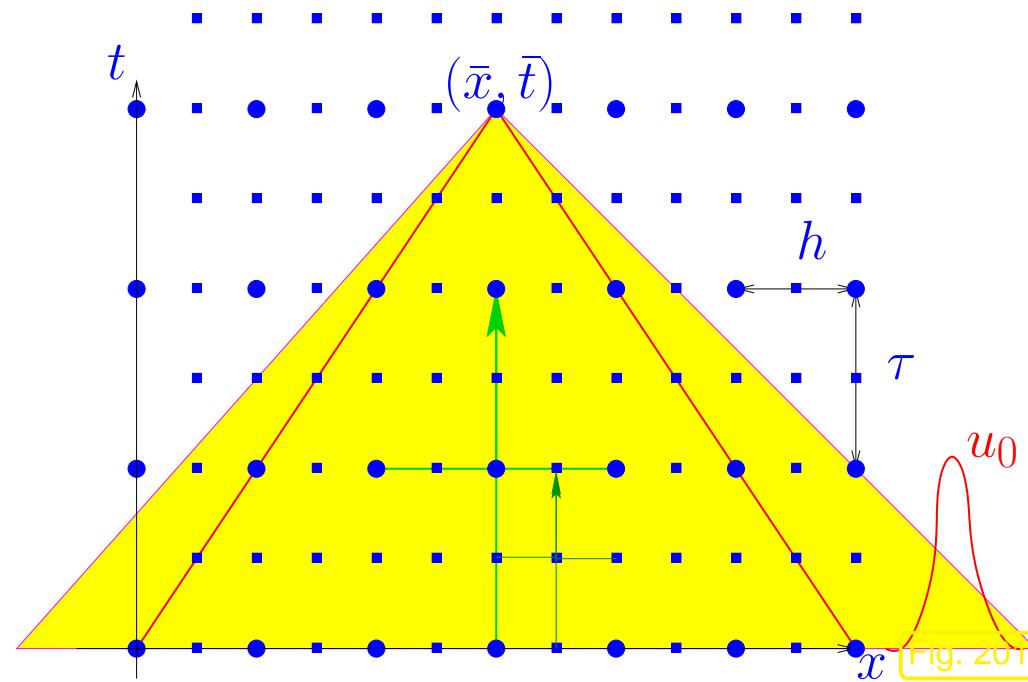
Below: yellow region $\hat{=}$ domain of dependence (d.o.d.) of (\bar{x}, \bar{t})



$c\tau < h$: numerical domain of dependence
 (marked —) contained in d.o.d.
 \Rightarrow CFL-condition met



$c\tau > h$: numerical domain of dependence
 (marked —) **not** contained in d.o.d.
 \Rightarrow CFL-condition violated



($\bullet \hat{=} \text{coarse grid}$, $\blacksquare \hat{=} \text{fine grid}$, $\square \hat{=} \text{d.o.d.}$)
 \triangleleft 1D consideration:
 sequence of equidistant space-time grids of $\tilde{\Omega}$ with
 $\tau = \gamma h$ (τ/h = meshwidth in time/space)
 If $\gamma >$ CFL-constraint (here $\gamma > c^{-1}$), then
 analytical domain of dependence $\not\subset$ numerical domain of dependence

- ▲ initial data u_0 outside numerical domain of dependence cannot influence approximation at grid point (\bar{x}, \bar{t}) on *any* mesh  no convergence !

CFL-condition \Leftrightarrow analytical domain of dependence \subset numerical domain of dependence



Will the CFL-condition thwart the efficient use of leapfrog, see Rem. 6.1.67 ?

To this end we need an idea about the convergence of the solutions of the fully discrete method:

“Meta-theorem” 6.2.42 (Convergence of fully discrete solutions of the wave equation).

Assume that

- the solution of the IBVP for the wave equation (6.2.19) is “sufficiently smooth”,
- its spatial Galerkin finite element discretization relies on degree p Lagrangian finite elements (\rightarrow Sect. 3.4) on uniformly shape-regular families of meshes,
- timestepping is based on the leapfrog method (6.2.33) with uniform timestep $\tau > 0$.

Then we can expect an asymptotic behavior of the total discretization error according to

$$\left(\tau \sum_{j=1}^M \|u - u_N(\tau j)\|_{H^1(\Omega)}^2 \right)^{\frac{1}{2}} \leq C(h_M^p + \tau^2), \quad (6.2.43)$$

where $C > 0$ must not depend on h_M , τ .

L.F. is 2nd-order!

As in the case of Metatheorem 6.1.63 (\Rightarrow nothing new!) we find:

$$\text{total discretization error} = \text{spatial error} + \text{temporal error}$$

Rem. 5.3.45 still applies: (6.2.43) does not give information about actual error, but only about the **trend** of the error, when discretization parameters $h_{\mathcal{M}}$ and τ are varied.

► Nevertheless, as in the case of the a priori error estimates of Sect. 5.3.5, we can draw conclusions about optimal refinement strategies in order to achieve prescribed *error reduction*.

As in Sect. 5.3.5 we make the **assumption** that the estimates (6.2.43) are sharp for all contributions to the total error and that the constants are the same (!)

$$\begin{aligned} \text{contribution of spatial error} &\approx Ch_{\mathcal{M}}^p, h_{\mathcal{M}} \hat{=} \text{mesh width } (\rightarrow \text{Def. 5.2.3}), \\ \text{contribution of temporal error} &\approx C\tau^2, \tau \hat{=} \text{timestep size}. \end{aligned} \quad (6.2.44)$$

This suggests the following change of $h_{\mathcal{M}}, \tau$ in order to achieve *error reduction* by a factor of $\rho > 1$:

$$\begin{aligned} \text{reduce mesh width by factor } &\rho^{1/p} && \xrightarrow{(6.1.65)} && \text{error reduction by } \rho > 1. \\ \text{reduce timestep by factor } &\rho^{1/2} \end{aligned} \quad (6.2.45)$$

Guideline: spatial and temporal resolution have to be adjusted in tandem

Parallel zu Rem. 6.1.67 we may wonder whether the timestep constraint $\tau < O(h_{\mathcal{M}})$ (asymptotically) enforces small timesteps not required for accuracy:

Only for $p = 1$ (linear Lagrangian finite elements) the requirement $\tau < O(h_M)$ stipulates the use of a smaller timestep than accuracy balancing according to (6.2.45).

The leapfrog timestep constraint $\tau \leq O(h_M)$ does not compromise (asymptotic) efficiency, if $p \geq 2$ ($p \hat{=} \text{degree of spatial Lagrangian finite elements}$).

Convection-Diffusion Problems

7.1 Heat conduction in a fluid

$\Omega \subset \mathbb{R}^d \hat{=} \text{bounded computational domain, } d = 1, 2, 3$

To begin with we want to develop a mathematical model for stationary fluid flow, for instance, the steady streaming of water.

7.1.1 Modelling fluid flow

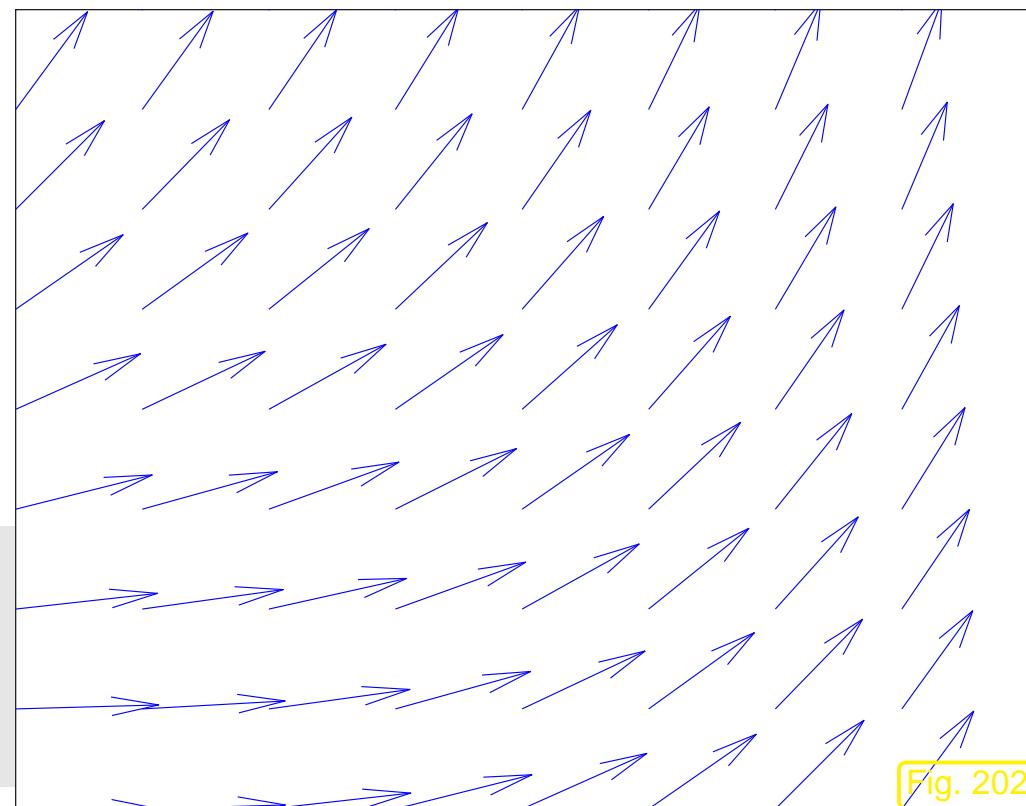
Flow field:

$$\mathbf{v} : \Omega \mapsto \mathbb{R}^d$$

Assumption:

\mathbf{v} is *continuous*, $\mathbf{v} \in (C^0(\bar{\Omega}))^d$

In fact, we will require that \mathbf{v} is uniformly Lipschitz continuous, but this is a mere technical assumption.



Clearly:

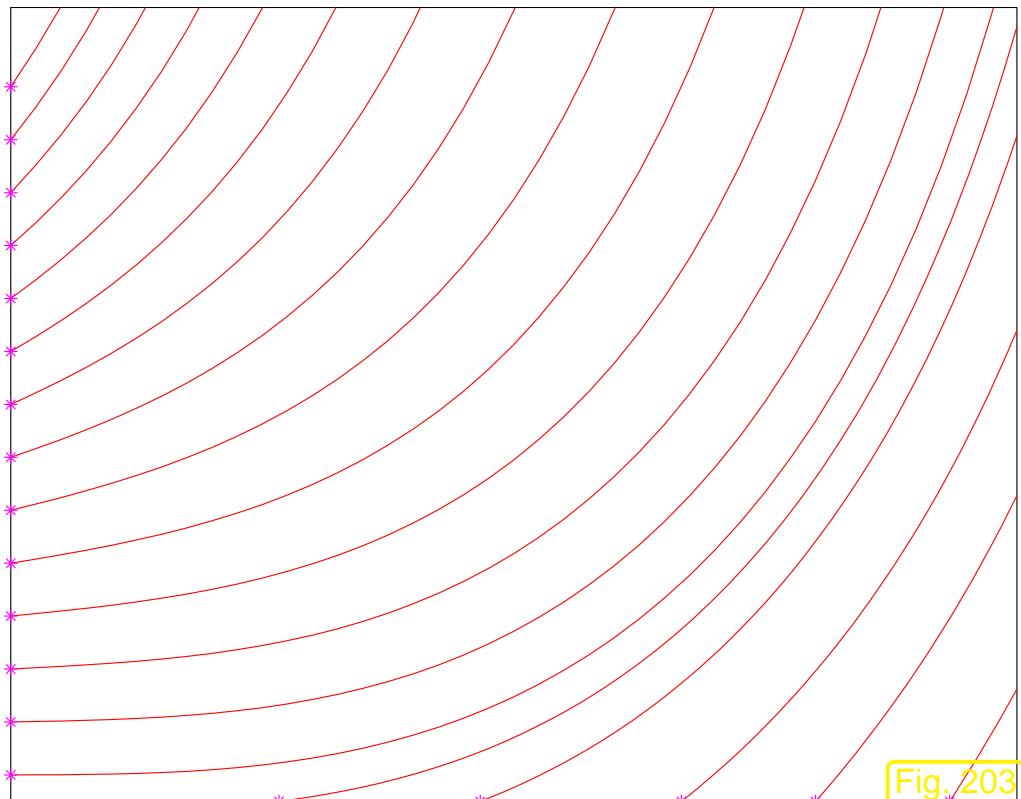
$\mathbf{v}(\mathbf{x}) \hat{=} \text{fluid velocity at point } \mathbf{x} \in \Omega$

➤ \mathbf{v} corresponds to a **velocity field**!

Given a flow field $\mathbf{v} \in (C^0(\bar{\Omega}))^d$ we can consider the autonomous initial value problems

$$\frac{d}{dt}\mathbf{y} = \mathbf{v}(\mathbf{y}) \quad , \quad \mathbf{y}(0) = \mathbf{x}_0 . \quad (7.1.1)$$

Its solution $t \mapsto \mathbf{y}(t)$ defines the path travelled by a particle carried along by the fluid, a **particle trajectory**, also called a **streamline**.



▷ particle trajectories (streamlines) in flow field of Fig. 202.

(* $\hat{=}$ initial particle positions)

A flow field induces a transformation (mapping) of space! to explain this, let us temporarily make the assumption that

the flow does neither enter nor leave Ω ,
(this applies to fluid flow in a close container)

which can be modelled by

$$\mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \partial\Omega , \quad (7.1.2)$$

that is, the flow is always parallel to the boundary of Ω : all particle trajectories stay inside Ω .

Now we fix some “time of interest” $t > 0$.

$$\Rightarrow \text{ mapping } \Phi^t : \begin{cases} \Omega \mapsto \Omega \\ \mathbf{x}_0 \mapsto \mathbf{y}(t) \end{cases}, \quad t \mapsto \mathbf{y}(t) \text{ solution of IVP (7.1.1)} , \quad (7.1.3)$$

is well-defined mapping of Ω to itself, the **flow map**. Obviously, it satisfies

$$\Phi^0 \mathbf{x}_0 = \mathbf{x}_0 \quad \forall \mathbf{x}_0 \in \Omega . \quad (7.1.4)$$

In [14, Def. 11.1.6] the more general concept of an **evolution operator** was introduced, which agrees with the flow map in the current setting.

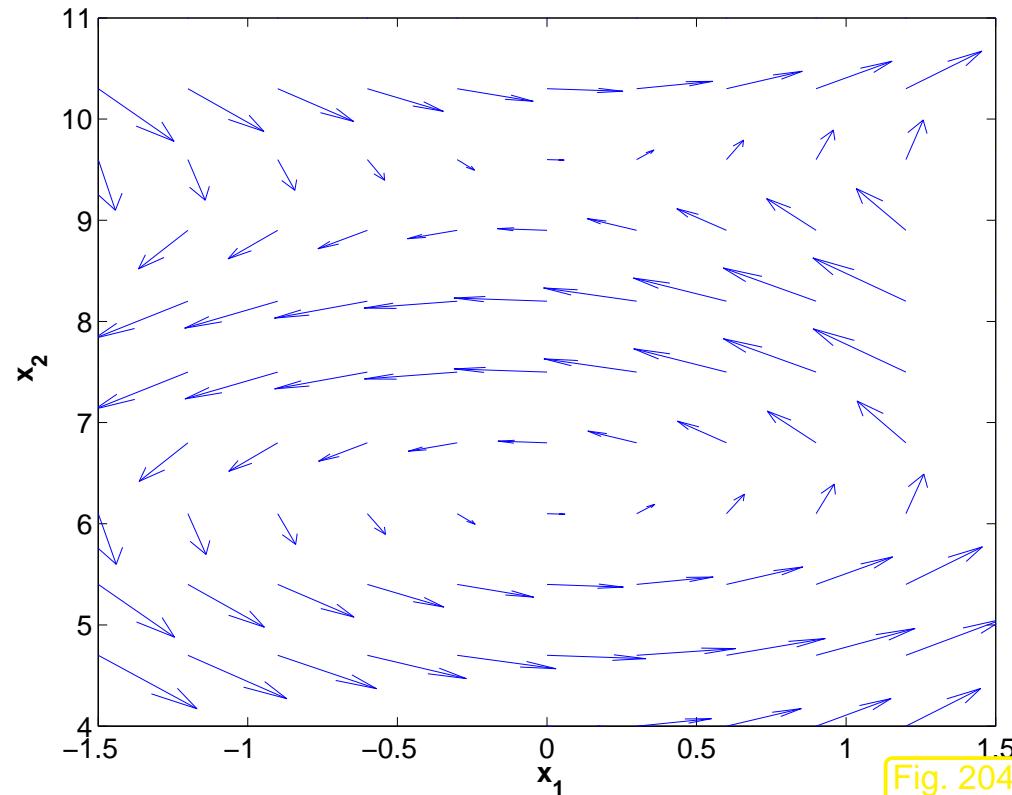


Fig. 204

flow field $\mathbf{v} : \Omega \mapsto \mathbb{R}^2$

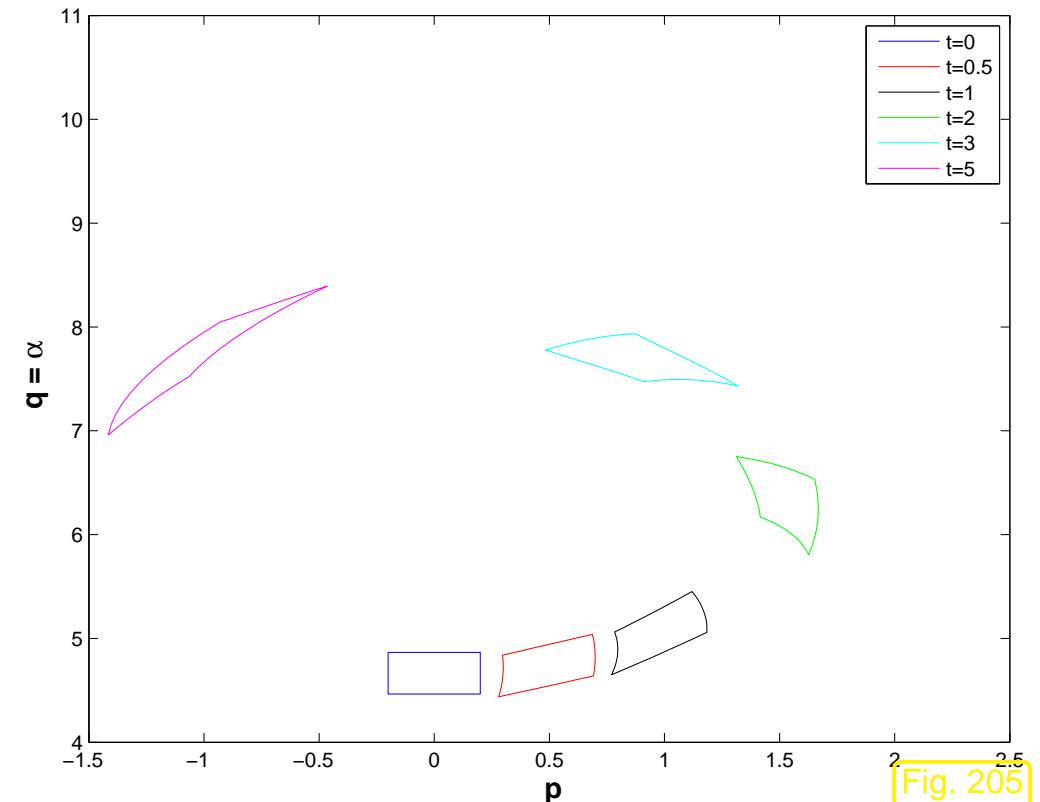


Fig. 205

snapshots of $\Phi^t(V)$ for control volume V

$\Phi^\tau(V) \hat{=} \text{volume occupied at time } t = \tau \text{ by particles that occupied } V \subset \Omega \text{ at time } t = 0.$

7.1.2 Heat convection and diffusion

$u : \Omega \mapsto \mathbb{R} \hat{=} \text{stationary temperature distribution in fluid } \textcolor{red}{\text{moving}}$ according to a stationary flow field
 $\mathbf{v} : \Omega \mapsto \mathbb{R}^d$

We adapt the considerations of Sect. 2.5 that led to the stationary heat equation. Recall

Conservation of energy

$$\int_{\partial V} \mathbf{j} \cdot \mathbf{n} dS = \int_V f dx \quad \text{for all "control volumes" } V . \quad (2.5.2)$$

power flux through surface of V heat production inside V

From 2.5.2 by Gauss' theorem Thm. 2.4.5

$$\int_V \operatorname{div} \mathbf{j}(\mathbf{x}) dx = \int_V f(\mathbf{x}) dx \quad \text{for all "control volumes" } V \subset \Omega .$$

Now appeal to another version of the fundamental lemma of the calculus of variations, see Lemma 2.4.10, this time sporting piecewise constant test functions.

► local form of energy conservation:

$$\operatorname{div} \mathbf{j} = f \quad \text{in } \Omega . \quad (2.5.5)$$

However, in a moving fluid a power flux through a fixed surface is already caused by the sheer fluid flow carrying along thermal energy. This is reflected in a modified Fourier's law (2.5.3):

Fourier's law in moving fluid

$$\mathbf{j}(\mathbf{x}) = -\kappa \operatorname{grad} u(\mathbf{x}) + \mathbf{v}(\mathbf{x})\rho u(\mathbf{x}) , \quad \mathbf{x} \in \Omega . \quad (7.1.5)$$

diffusive heat flux

(due to spatial variation of temperature)

convective heat flux

(due to fluid flow)

$\kappa > 0 \hat{=} \text{heat conductivity } ([\kappa] = 1 \frac{\text{W}}{\text{Km}})$, $\rho > 0 \hat{=} \text{heat capacity } ([\rho] = \frac{\text{J}}{\text{Km}^3})$, both assumed to be constant (in contrast to the models of Sect. 2.5 and Sect. 6.1.1).

Combine equations (2.5.5) & (7.1.5):

$$\begin{aligned} \operatorname{div} \mathbf{j} = f &+ \mathbf{j}(\mathbf{x}) = -\kappa \operatorname{grad} u(\mathbf{x}) + \mathbf{v}(\mathbf{x})\rho u(\mathbf{x}) \\ &\quad \downarrow \\ -\operatorname{div}(\kappa \operatorname{grad} u) + \operatorname{div}(\rho \mathbf{v}(\mathbf{x})u) &= f \quad \text{in } \Omega . \end{aligned} \tag{7.1.6}$$



Linear scalar **convection-diffusion equation** (for unknown temperature u)

Terminology :

$$-\operatorname{div}(\kappa \operatorname{grad} u) + \operatorname{div}(\rho \mathbf{v}(\mathbf{x})u) = f .$$

\downarrow \downarrow
diffusive term **convective term**

The 2nd-order elliptic PDE (7.1.6) has to be supplemented with exactly one *boundary condition* on any part of $\partial\Omega$, see Sect. 2.6, Ex. 2.6.7. This can be any of the boundary conditions introduced in Sect. 2.6:

- Dirichlet boundary conditions: $u = g \in C^0(\partial\Omega)$ on $\partial\Omega$ (fixed surface temperature),
- Neumann boundary conditions: $\mathbf{j} \cdot \mathbf{n} = -h$ on $\partial\Omega$ (fixed heat flux),
- (non-linear) radiation boundary conditions: $\mathbf{j} \cdot \mathbf{n} = \Psi(u)$ on $\partial\Omega$ (temperature dependent heat flux).

7.1.3 Incompressible fluids

For the sake of simplicity we will mainly consider **incompressible fluids**.

Definition 7.1.7 (Incompressible flow field).

A fluid flow is called **incompressible**, if the associated flow map Φ^t is **volume preserving**,

$$|\Phi^t(V)| = |\Phi^0(V)| \quad \text{for all sufficiently small } t > 0, \text{ for all control volumes } V.$$

Can incompressibility be read off the velocity field \mathbf{v} of the flow?

To investigate this issue, again assume the “no flow through the boundary condition” (7.1.2) and recall that the flowmap Φ^t from (7.1.3) satisfies

$$\frac{\partial}{\partial t} \Phi(t, \mathbf{x}) = \mathbf{v}(\Phi(t, \mathbf{x})) , \quad \mathbf{x} \in \Omega, t > 0 . \quad (7.1.8)$$

Here, in order to make clear the dependence on independent variables, time occurs as an argument of Φ in brackets, on par with \mathbf{x} .

Next, formal differentiation w.r.t. \mathbf{x} and change of order of differentiation yields a differential equation for the Jacobian $D_{\mathbf{x}} \Phi^t$,

$$(7.1.8) \Rightarrow \frac{\partial}{\partial t} (D_{\mathbf{x}} \Phi)(t, \mathbf{x}) = D\mathbf{v}(\Phi(t, \mathbf{x})) (D_{\mathbf{x}} \Phi)(t, \mathbf{x}) . \quad (7.1.9)$$

Jacobian $\in \mathbb{R}^{d,d}$ Jacobian $\in \mathbb{R}^{d,d}$

Second strand of thought: apply transformation formula for integrals (3.5.31), [19, Satz 8.5.2]: for fixed $t > 0$

$$|\Phi(t, V)| = \int_{\Phi(t, V)} 1 d\mathbf{x} = \int_V |\det(D_{\mathbf{x}} \Phi)(t, \hat{\mathbf{x}})| d\hat{\mathbf{x}} . \quad (7.1.10)$$

Volume preservation by the flow map is equivalent to

$$t \mapsto |\Phi(t, V)| = \text{const.} \iff \frac{d}{dt} |\Phi(t, V)| = 0 ,$$

for any control volume $V \subset \Omega$.

$$(7.1.10) \Rightarrow \frac{d}{dt} |\Phi(t, V)| = \int_V \frac{\partial}{\partial t} |\det(D_{\mathbf{x}} \Phi)(t, \hat{\mathbf{x}})| d\hat{\mathbf{x}} .$$

Theorem 7.1.11 (Differentiation formula for determinants).

Let $\mathbf{S} : I \subset \mathbb{R} \mapsto \mathbb{R}^{n,n}$ be a smooth matrix-valued function. If $\mathbf{S}(t_0)$ is regular for some $t_0 \in I$, then

$$\frac{d}{dt} (\det \circ \mathbf{S})(t_0) = \det(\mathbf{S}(t_0)) \operatorname{tr}\left(\frac{d\mathbf{S}}{dt}(t_0) \mathbf{S}^{-1}(t_0)\right) .$$

►
$$\frac{\partial}{\partial t} \det(D_{\mathbf{x}}\Phi)(t, \hat{\mathbf{x}}) \stackrel{(7.1.9)}{=} \det(D_{\mathbf{x}}\Phi)(t, \hat{\mathbf{x}}) \operatorname{tr}(D\mathbf{v}(\Phi(t, \hat{\mathbf{x}}))) \underbrace{(D_{\mathbf{x}}\Phi)(t, \hat{\mathbf{x}})(D_{\mathbf{x}}\Phi)^{-1}(t, \hat{\mathbf{x}})}_{=\mathbf{I}}$$

$$= \det(D_{\mathbf{x}}\Phi)(t, \hat{\mathbf{x}}) \operatorname{div} \mathbf{v}(\Phi(t, \hat{\mathbf{x}})) ,$$

because the divergence of a vector field \mathbf{v} is just the trace of its Jacobian $D\mathbf{v}$! From (7.1.4) we know that for small $t > 0$ the Jacobian $D_{\mathbf{x}}\Phi(t, \hat{\mathbf{x}})$ will be close to \mathbf{I} and, therefore, $\det(D_{\mathbf{x}}\Phi)(t, \hat{\mathbf{x}}) \neq 0$ for $t \approx 0$. Thus, for small $t > 0$ we conclude

$$\frac{d}{dt} |\Phi(t, V)| = 0 \Leftrightarrow \operatorname{div} \mathbf{v}(\Phi(t, \hat{\mathbf{x}})) = 0 \quad \forall \hat{\mathbf{x}} \in V .$$

Since this is to hold for **any** control volume V , the final equivalence is

$$\frac{d}{dt} |\Phi(t, V)| = 0 \quad \forall \text{control volumes } V \Leftrightarrow \operatorname{div} \mathbf{v} = 0 \quad \text{in } \Omega .$$

Theorem 7.1.12 (Divergence-free velocity fields for incompressible flows).

A stationary fluid flow in Ω is incompressible (\rightarrow Def. 7.1.7), if and only if its associated velocity field \mathbf{v} satisfies $\operatorname{div} \mathbf{v} = 0$ everywhere in Ω .

In the sequel we make the **assumption**:

$$\operatorname{div} \mathbf{v} = \sum_{j=1}^d \frac{\partial v_j}{\partial x_j} = 0$$

(Note: for $d = 1$ this boils down to $\frac{dv}{dx} = 0$ and implies $v = \text{const.}$)

Then we can use the product rule in higher dimensions of Lemma 2.4.3:

$$\operatorname{div}(\rho \mathbf{v} u) \stackrel{\text{Lemma 2.4.3}}{=} \rho(u \operatorname{div} \mathbf{v} + \mathbf{v} \cdot \operatorname{grad} u) \stackrel{\operatorname{div} \mathbf{v} = 0}{=} \rho \mathbf{v} \cdot \operatorname{grad} u .$$

Thus, we can rewrite the scalar convection-diffusion equation (7.1.6) for an incompressible flow field

$$-\operatorname{div}(\kappa \operatorname{grad} u) + \operatorname{div}(\rho \mathbf{v}(\mathbf{x}) u) = f \quad \text{in } \Omega$$

$$\leftarrow \operatorname{div} \mathbf{v} = 0$$

$$-\kappa \Delta u + \rho \mathbf{v} \cdot \operatorname{grad} u = f \quad \text{in } \Omega . \quad (7.1.13)$$

When carried along by the flow of an incompressible fluid, the temperature cannot be increased by local compression, the effect that you can witness when pumping air. Hence, only sources/sinks can lead to local extrema of the temperature.

Now recall the discussion of the physical intuition behind the **maximum principle** of Thm. 5.7.2. These considerations still apply to stationary heat flow in a moving incompressible fluid.

Theorem 7.1.14 (Maximum principle for scalar 2nd-order convection diffusion equations).

Let $\mathbf{v} : \Omega \mapsto \mathbb{R}^d$ be a continuously differentiable vector field. Then there holds the **maximum principle**

$$-\Delta u + \mathbf{v} \cdot \operatorname{grad} u \geq 0 \implies \min_{\mathbf{x} \in \partial\Omega} u(\mathbf{x}) = \min_{\mathbf{x} \in \Omega} u(\mathbf{x}) ,$$

$$-\Delta u + \mathbf{v} \cdot \operatorname{grad} u \leq 0 \implies \max_{\mathbf{x} \in \partial\Omega} u(\mathbf{x}) = \max_{\mathbf{x} \in \Omega} u(\mathbf{x}) .$$

7.1.4 Transient heat conduction

In Sect. 6.1.1 we generalized the laws of stationary heat conduction derived in Sect. 2.5 to time-dependent temperature distributions $u = u(\mathbf{x}, t)$ sought on a space-time cylinder $\tilde{\Omega} := \Omega \times]0, T[$. The same ideas apply to heat conduction in a fluid:

- Start from energy balance law (6.1.1) and convert it into local form (6.1.2).
- Combine it with the extended Fourier's law (7.1.5).



$$\frac{\partial}{\partial t}(\rho u) - \operatorname{div}(\kappa \operatorname{grad} u) + \operatorname{div}(\rho \mathbf{v}(\mathbf{x}, t)u) = f(\mathbf{x}, t) \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[. \quad (7.1.15)$$

For details and notations refer to Sect. 6.1.1.

This PDE has to be supplemented with

- boundary conditions (as in the stationary case, see Sect. 2.6),

- initial conditions (same as for pure diffusion, see Sect. 6.1.1).

Under the assumption $\operatorname{div}_{\mathbf{x}} \mathbf{v}(\mathbf{x}, t) = 0$ of incompressibility (\rightarrow Def. 7.1.7 and Thm. 7.1.12) (7.1.15) is equivalent to

$$\frac{\partial}{\partial t}(\rho u) - \kappa \Delta u + \rho \mathbf{v}(\mathbf{x}, t) \cdot \operatorname{grad} u = f(\mathbf{x}, t) \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[. \quad (7.1.16)$$

7.2 Stationary convection-diffusion problems

Model problem, cf. (7.1.13), modelling stationary heat flow in an incompressible fluid with prescribed temperature at “walls of the container” (\leftrightarrow Dirichlet boundary conditions).

$$-\kappa \Delta u + \rho \mathbf{v}(\mathbf{x}) \cdot \operatorname{grad} u = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega.$$

Perform **scaling** $\hat{=}$ choice of physical units: makes equation non-dimensional by fixing “reference length”, “reference time interval”.

A suitable choice of physical units leads to rescaled physical constants $\kappa \rightarrow \epsilon$, $\rho \rightarrow 1$, $\|\mathbf{v}\|_{L^\infty(\Omega)} \rightarrow 1$.

After scaling we deal with the non-dimensional boundary value problem

$$-\epsilon \Delta u + \mathbf{v}(\mathbf{x}) \cdot \operatorname{grad} u = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega, \tag{7.2.1}$$

diffusion term
 (2nd-order term) convection term
 (1st-order term)

with $\epsilon > 0$, $\|\mathbf{v}\|_{L^\infty(\Omega)} = 1$, $\operatorname{div} \mathbf{v} = 0 \rightarrow$ incompressible fluid, see Def. 7.1.7.

Remark 7.2.2 (Variational formulation for convection-diffusion BVP).

Standard “4-step approach” of Sect. 2.8 can be directly applied to BVP (7.2.1) with one new twist:

Do not use integration by parts (Green’s formula, Thm. 2.4.7) on convective terms!

► variational formulation for BVP (7.2.1):

$$u \in H_0^1(\Omega): \underbrace{\epsilon \int_{\Omega} \mathbf{grad} u \cdot \mathbf{grad} v \, dx + \int_{\Omega} (\mathbf{v} \cdot \mathbf{grad} u) v \, dx}_{\text{bilinear form } \mathbf{a}(u,v)} = \underbrace{\int_{\Omega} f(x) \, dx}_{\text{linear form } \ell(v)} \quad \forall v \in H_0^1(\Omega) .$$

$\hat{=}$ a linear variational problem, see Sect. 2.3.1.

Surprise: \mathbf{a} is *positive definite* (\rightarrow Def. 2.1.22), because

$$\int_{\Omega} (\mathbf{v} \cdot \mathbf{grad} u) u \, dx = \int_{\Omega} (\mathbf{v} u) \cdot \mathbf{grad} u \, dx$$

Green’s formula $\underset{=} - \int_{\Omega} \operatorname{div}(\mathbf{v} u) \cdot \mathbf{grad} u \, dx + \int_{\partial\Omega} \underbrace{u^2}_{=0} \mathbf{v} \cdot \mathbf{n} \, dS$

$$(2.4.4) \& \stackrel{\text{div } \mathbf{v}=0}{=} - \int_{\Omega} (\mathbf{v} \cdot \mathbf{grad} u) u \, d\mathbf{x} .$$

► $\mathbf{a}(u, u) = \epsilon \int_{\Omega} \|\mathbf{grad} u\|^2 \, d\mathbf{x} > 0 \quad \forall u \in H_0^1(\Omega) \setminus \{0\} . \quad (7.2.3)$

From this we conclude existence and uniqueness of solutions of the BVP (7.2.1) in the Sobolev space $H_0^1(\Omega)$.

7.2.1 Singular perturbation

Setting: fast-moving fluid \leftrightarrow convection dominates diffusion $\leftrightarrow \epsilon \ll 1$ in (7.2.1)

Example 7.2.4 (1D convection-diffusion boundary value problem).

$$-\epsilon \frac{d^2u}{dx^2} + \frac{du}{dx} = 1 \quad \text{in } \Omega ,$$

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

► $u(x) = x + \frac{\exp(-x/\epsilon) - 1}{1 - \exp(-1/\epsilon)} .$

For $\epsilon \ll 1$:

boundary layer at $x = 1$

For $\epsilon \rightarrow 0$:

$$u(x) \rightarrow x .$$

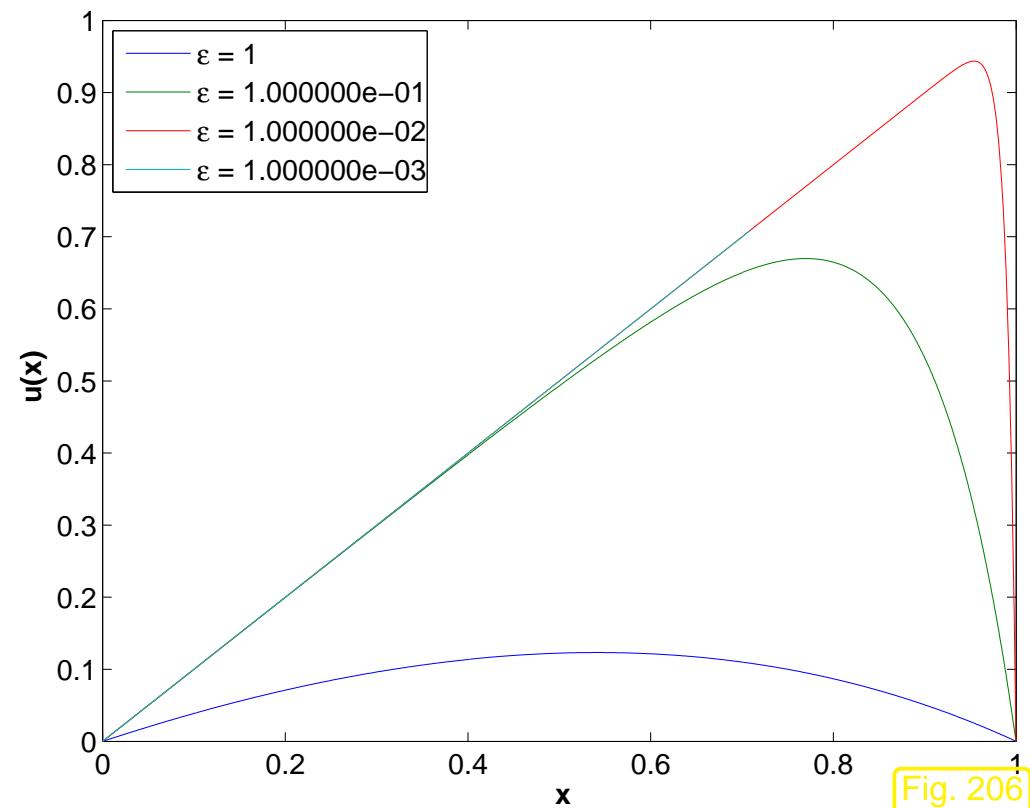


Fig. 206



“Limit problem”: ignore diffusion \Rightarrow set $\epsilon = 0$

$$(7.2.1) \quad \xrightarrow{\epsilon=0} \quad \mathbf{v}(\mathbf{x}) \cdot \nabla u = f(\mathbf{x}) \quad \text{in } \Omega . \quad (7.2.5)$$

Case $d = 1$ ($\Omega =]0, 1[$, $v = \pm 1$)

$$(7.2.5) \quad \xrightarrow{d=1} \quad \pm \frac{du}{dx}(x) = f(x) \quad \Rightarrow \quad u(x) = \int f \, dx + C . \quad (7.2.6)$$

What about this constant C ?

If $v = 1 \leftrightarrow$ fluid flows “from left to right”, so we should integrate the source from 0 to x :

$$u(x) = u(0) + \int_0^x f(s) \, ds = \int_0^x f(s) \, ds , \quad (7.2.7)$$

because $u(0) = 0$ by the boundary condition $u = 0$ on $\partial\Omega$. If $v = -1$ we start the integration at $x = 1$. Note that this makes the maximum principle of Thm. 7.1.14 hold.

For $d > 1$ we can solve (7.2.5) by **the method of characteristics**:

To motivate it, be aware that (7.2.5) describes **pure transport** of a temperature distribution in the velocity field \mathbf{v} , that is, the heat/temperature is just carried along particle trajectories and changes only under the influence of heat sources/sinks along that trajectory.

Denote by \mathbf{u} the solution of (7.2.5) and recall the differential equation (7.1.1) for a particle trajectory

$$\frac{d\mathbf{y}}{dt}(t) = \mathbf{v}(\mathbf{y}(t)) \quad , \quad \mathbf{y}(0) = \mathbf{x}_0 . \quad (7.1.1)$$

► $\frac{d}{dt}u(\mathbf{y}(t)) = \mathbf{grad} u(\mathbf{y}(t)) \cdot \frac{d}{dt}\mathbf{y}(t) = \mathbf{grad} u \cdot \mathbf{v}(\mathbf{y}(t)) \stackrel{(7.2.5)}{=} f(\mathbf{y}(t)) .$

➤ Compute $\mathbf{u}(\mathbf{y}(t))$ by integrating source f along particle trajectory!

$$u(\mathbf{y}(t)) = u(\mathbf{x}_0) + \int_0^t f(\mathbf{y}(s)) \, ds \quad (7.2.8)$$

Taking the cue from $d = 1$ we choose \mathbf{x}_0 as “the point on the boundary where the particle enters Ω ”. These points form the part of the boundary through which the flow enters Ω , the **inflow boundary**

$$\Gamma_{\text{in}} := \{\mathbf{x} \in \partial\Omega: \mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0\} . \quad (7.2.9)$$

Its complement in $\partial\Omega$ contains the **outflow boundary**

$$\Gamma_{\text{out}} := \{\mathbf{x} \in \partial\Omega: \mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) > 0\} . \quad (7.2.10)$$

Remark 7.2.11 (Recirculating flow).

- velocity field
- : Streamline connecting Γ_{in} and Γ_{out}
- : Closed streamline

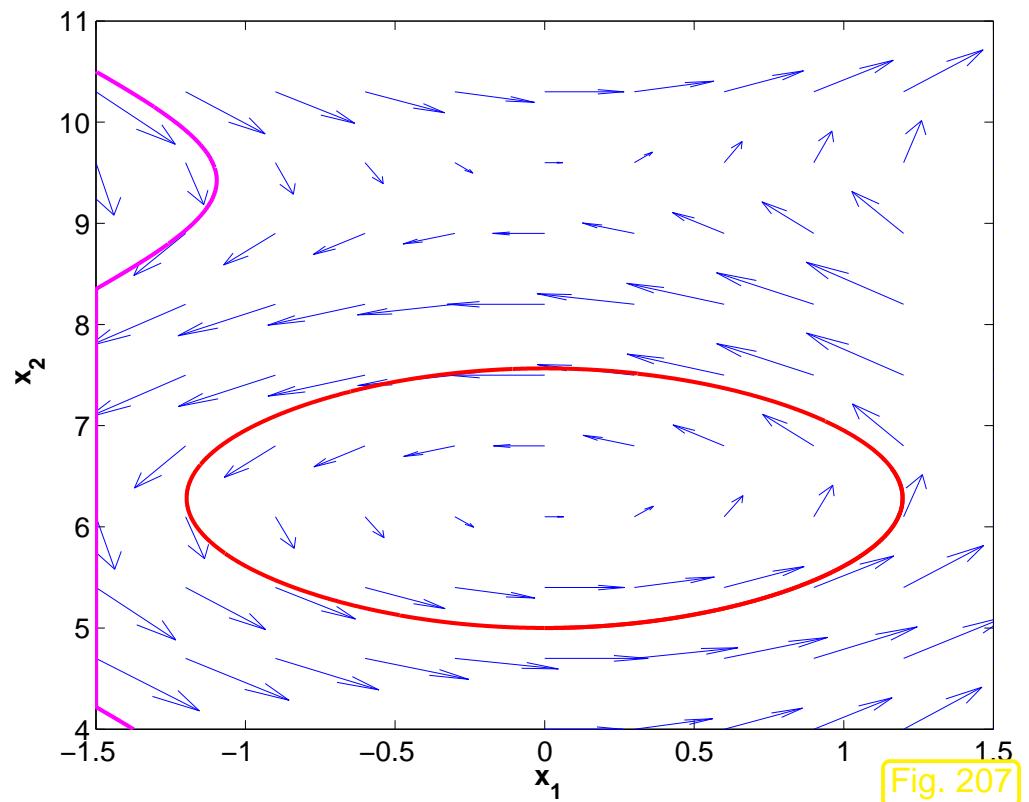


Fig. 207

In the case of closed streamlines the stationary pure transport problem fails to have a unique solution: on a closed streamline u can attain “any” value, because there is no boundary value to fix u .



Return to case $d = 1$. In general solution $u(x)$ from (7.2.6) will **not** satisfy the boundary condition $u(1) = 0$! Also for $u(x)$ from (7.2.8) the homogeneous boundary conditions may be violated where the particle trajectory leaves Ω !

In the limit case $\epsilon = 0$ not all boundary conditions of (7.2.1) can be satisfied.

Definition 7.2.12 (Singularly perturbed problem).

A *boundary value problem depending on parameter $\epsilon \approx \epsilon_0$ is called **singularly perturbed**, if the limit problem for $\epsilon \rightarrow \epsilon_0$ is not compatible with the boundary conditions.*

Especially in the case of 2nd-order elliptic boundary value problems:

Singular perturbation = 1st-order terms become dominant for $\epsilon \rightarrow \epsilon_0$

7.2.2 Upwinding

Focus: linear finite element Galerkin discretization for 1D model problem, cf. Ex. 7.2.4

$$-\epsilon \frac{d^2u}{dx^2} + \frac{du}{dx} = f(x) \quad \text{in } \Omega , \quad u(0) = 0 , \quad u(1) = 0 . \quad (7.2.13)$$

Variational formulation, see Rem. 7.2.2:

$$u \in H_0^1([0, 1]): \quad \underbrace{\int_0^1 \frac{du}{dx}(x) \frac{du}{dx}(x) dx}_{=:a(u, v)} + \underbrace{\int_0^1 \frac{du}{dx}(x) v(x) dx}_{=:l(v)} = \int_0^1 f(x)v(x) dx \quad \forall v \in H_0^1([0, 1]) .$$

As in Sect. 1.5.1.2: use equidistant mesh \mathcal{M} (mesh width $h > 0$), composite trapezoidal rule (1.5.55) for right hand side linear form, standard “tent function basis”, see (1.5.49).

► linear system of equations for coefficients μ_i , $i = 1, \dots, M - 1$, providing approximations for point values $u(ih)$ of exact solution u .

$$\left(-\frac{\epsilon}{h} - \frac{1}{2} \right) \mu_{i-1} + \frac{2\epsilon}{h} \mu_i + \left(-\frac{\epsilon}{h} + \frac{1}{2} \right) \mu_{i+1} = h f(ih) , \quad i = 1, \dots, M - 1 , \quad (7.2.14)$$

where the homogeneous Dirichlet boundary conditions are taken into account by setting $\mu_0 = \mu_M = 0$.

Remark 7.2.15 (Finite differences for convection-diffusion equation in 1D).

As in Sect. 1.5.3 on the finite difference in 1D, we can also obtain (7.2.14) by replacing the derivatives by suitable difference quotients:

$$\begin{aligned}
 -\epsilon \frac{d^2u}{dx^2} + \frac{du}{dx} &= f(x) \\
 \uparrow &\quad \uparrow &\quad \uparrow \\
 \epsilon \underbrace{\frac{-\mu_{i+1} + 2\mu_i - \mu_{i-1}}{h^2}}_{\text{difference quotient for } \frac{d^2u}{dx^2}} + \underbrace{\frac{\mu_{i+1} - \mu_{i-1}}{2h}}_{\text{symmetric d.q. for } \frac{du}{dx}} &= f(ih) .
 \end{aligned} \tag{7.2.14}$$



- Model boundary value problem (7.2.13)
- linear finite element Galerkin discretization as described above
- As in Ex. 7.2.4: $f \equiv 1$

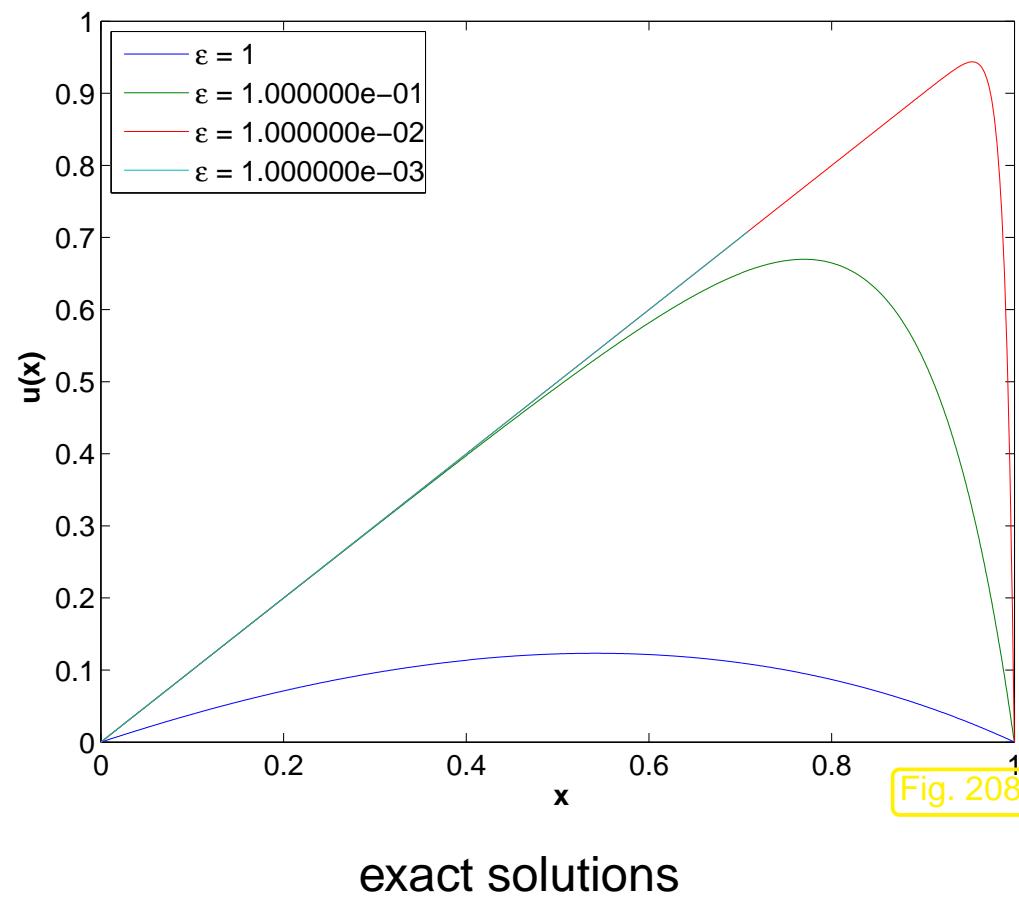


Fig. 208

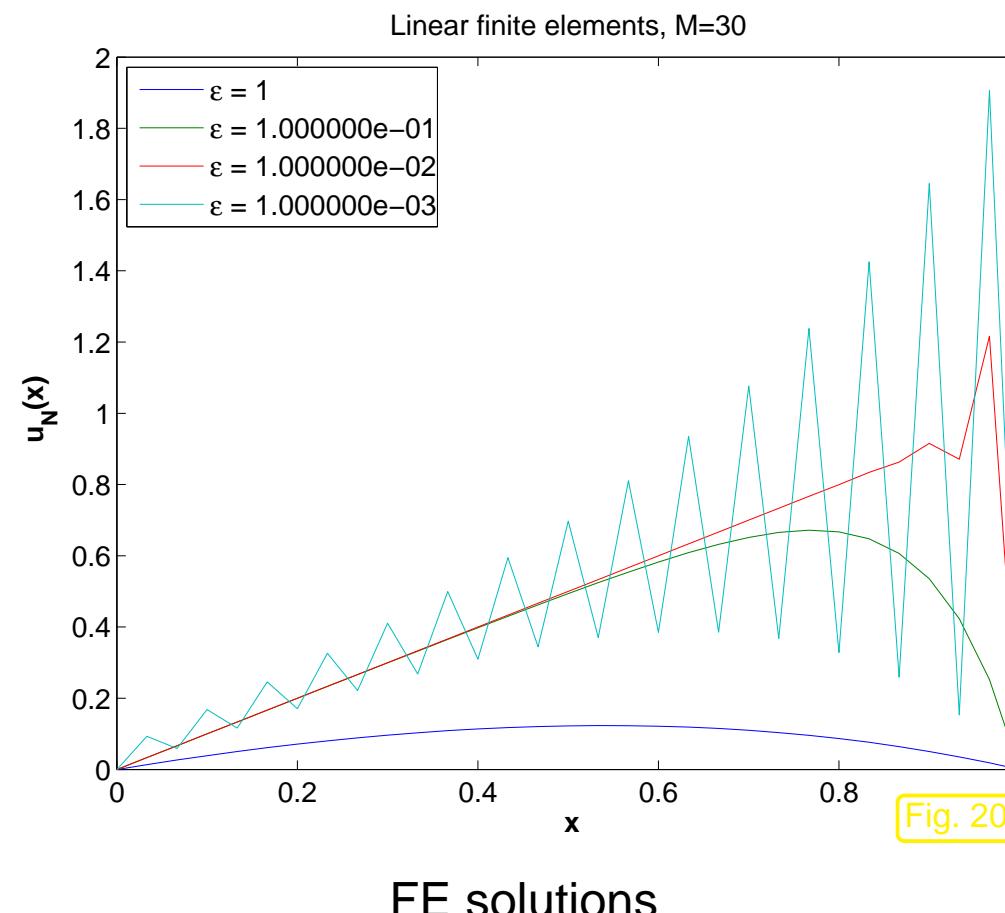


Fig. 209

For very small ϵ : spurious *oscillations* of linear FE Galerkin solution.



In order to understand this observation, study the linear finite element Galerkin discretization in the limit case $\epsilon = 0$

$$(7.2.14) \quad \blacktriangleright \quad \stackrel{\epsilon=0}{\mu_{i+1} - \mu_{i-1} = 2hf(ih)}, \quad i = 1, \dots, M. \quad (7.2.17)$$

\blacktriangleright (7.2.17) $\hat{=}$ Linear system of equations with *singular* system matrix!

For $\epsilon > 0$ the Galerkin matrix will always be regular due to (7.2.3), but the linear relationship (7.2.17) will become more and more dominant as $\epsilon > 0$ becomes smaller and smaller. In particular, (7.2.17) sends the message that values at even and odd numbered nodes will become decoupled, which accounts for the oscillations.

Desired:

robust discretization of (7.2.13)

= discretization that produces qualitatively correct solutions for **any** $\epsilon > 0$

Guideline:

Numerical methods for singularly perturbed problems must “work” for the limit problem

What is a meaningful scheme for limit problem $u' = f$ on an equidistant mesh of $\Omega :=]0, 1[$?

Explicit Euler method: $\mu_{i+1} - \mu_i = hf(\xi_i) \quad i = 0, \dots, N ,$

Implicit Euler method: $\mu_{i+1} - \mu_i = hf(\xi_{i+1}) \quad i = 0, \dots, N .$

► Use **one-sided difference quotients** for discretization of convective term !

Which type ? (Explicit or implicit Euler ?)

Linear system arising from *use of backward difference quotient* $\frac{du}{dx}|_{x=x_i} = \frac{\mu_i - \mu_{i-1}}{h} :$

$$\left(-\frac{\epsilon}{h} - 1\right) \mu_{i-1} + \left(\frac{2\epsilon}{h} + 1\right) \mu_i + -\frac{\epsilon}{h} \mu_{i+1} = hf(ih), \quad i = 1, \dots, M-1, \quad (7.2.18)$$

Linear system arising from *use of forward difference quotient* $\frac{du}{dx}|_{x=x_i} = \frac{\mu_{i+1} - \mu_i}{h} :$

$$-\frac{\epsilon}{h} \mu_{i-1} + \left(\frac{2\epsilon}{h} - 1\right) \mu_i + \left(-\frac{\epsilon}{h} + 1\right) \mu_{i+1} = hf(ih), \quad i = 1, \dots, M-1, \quad (7.2.19)$$

Example 7.2.20 (One-sided difference approximation of convective terms).

Model problem of Ex. 7.2.16, discretizations (7.2.18) and (7.2.19).

Upwind discretization, M=30

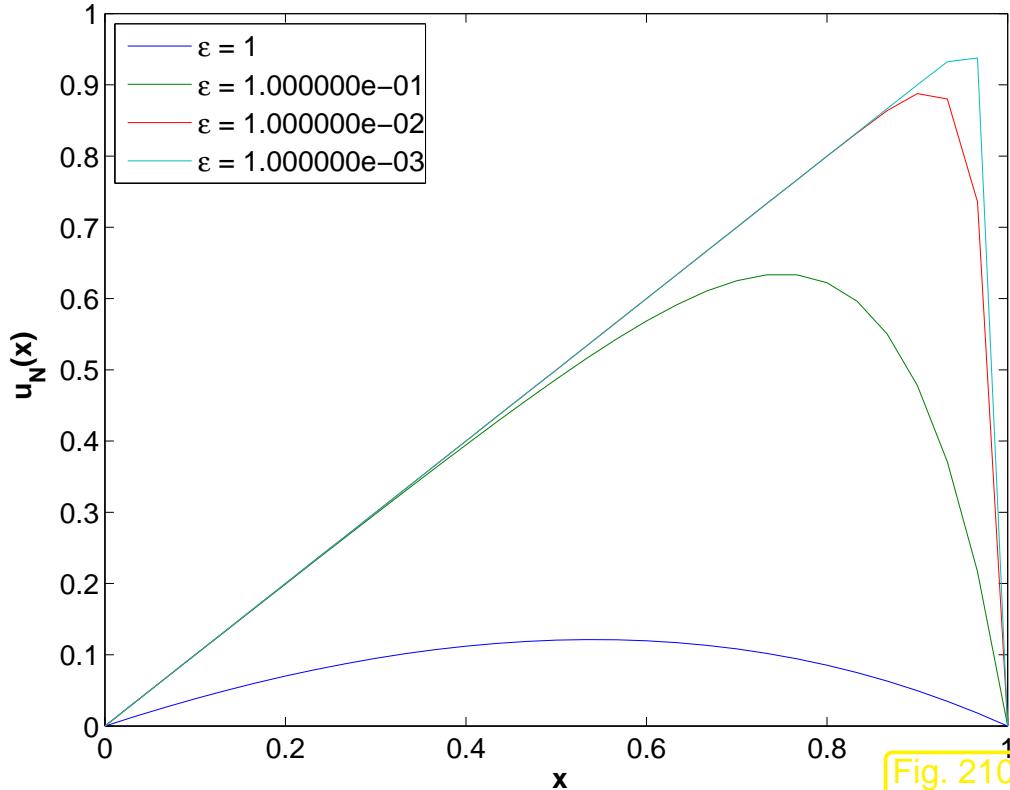


Fig. 210

backward difference quotient

Downwind discretization, M=30

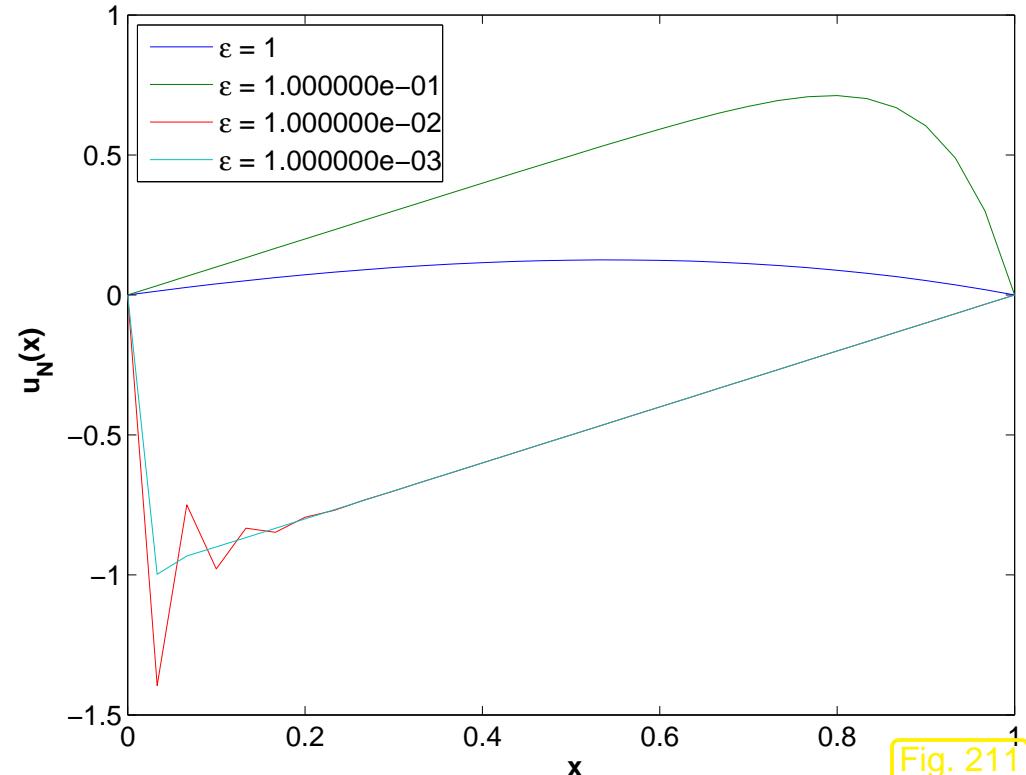


Fig. 211

forward difference quotient

Only the discretization of $\frac{du}{dx}$ based on the backward difference quotient generates qualitatively correct (piecewise linear) discrete solutions (a “good method”).

If the forward difference quotient is used, the discrete solutions may violate the maximum principle of Thm. 7.1.14 (a “bad method”).

How can we tell a good method from a bad method by merely examining the system matrix?



- ▶ Heuristic criterion for $\epsilon \rightarrow 0$ -robust stability of nodal finite element Galerkin discretization/finite difference discretization of *singularly perturbed* scalar linear convection-diffusion BVP (7.2.1) (with Dirichlet b.c.):

(Linearly interpolated) discrete solution satisfies **maximum principle** (5.7.3).



System matrix complies with sign-conditions (5.7.9)–(5.7.11).

Nodal finite element Galerkin discretization $\hat{=}$ basis expansion coefficients μ_i of Galerkin solution $u_N \in V_N$ double as point values of u_N at interpolation nodes. This is satisfied for Lagrangian finite element methods (\rightarrow Sect. 3.4) when standard nodal basis functions according to (3.4.3) are used.

Recall the sign-conditions (5.7.9)–(5.7.11) for the system matrix \mathbf{A} arising from nodal finite element Galerkin discretization or finite difference discretization:

- (5.7.9): positive diagonal entries,

$$(\mathbf{A})_{ii} > 0 ,$$

- (5.7.10): non-positive off-diagonal entries,

$$(\mathbf{A})_{ij} \leq 0, \text{ if } i \neq j ,$$

- “(5.7.11)": diagonal dominance,

$$\sum_j (\mathbf{A})_{ij} \geq 0 .$$

These conditions are met for *equidistant meshes in 1D*

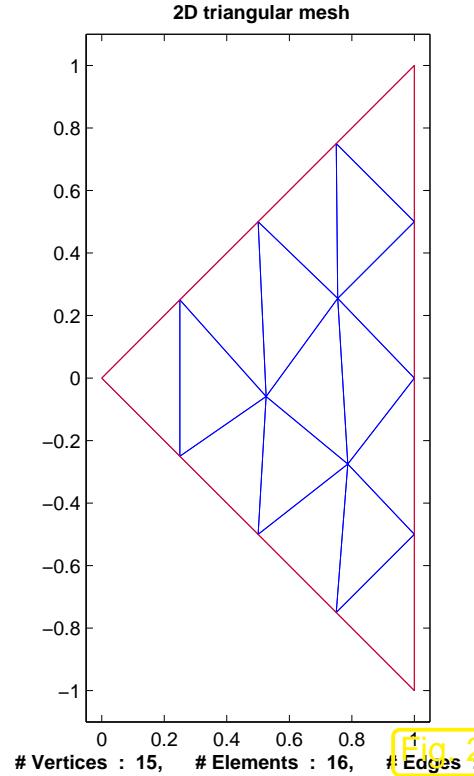
- for the standard $S_1^0(\mathcal{M})$ -Galerkin discretization (7.2.14), **provided that** $|\epsilon h^{-1}| \geq \frac{1}{2}$,
 - when using *backward* difference quotients for the convective term (7.2.18) for **any** choice of $\epsilon \geq 0$, $h > 0$,
 - when using *forward* difference quotients for the convective term (7.2.19), **provided that** $|\epsilon h^{-1}| \geq 1$.
- Only the use of a *backward* difference quotient for the convective term guarantees the (discrete) maximum principle in an $\epsilon \rightarrow 0$ -robust fashion!

Terminology: Approximation of $\frac{du}{dx}$ by *backward* difference quotients $\hat{=}$ **upwinding**

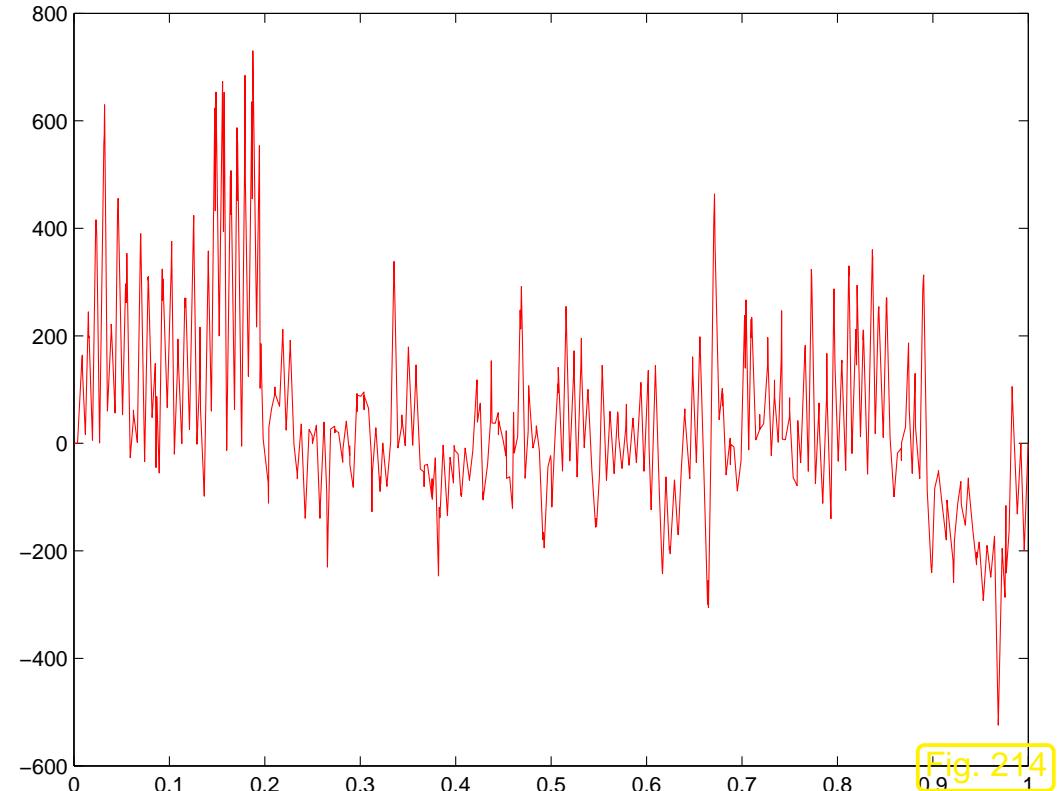
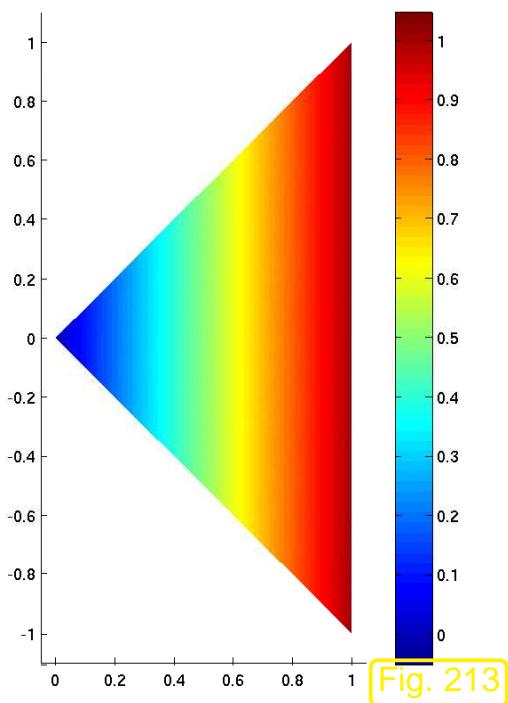
7.2

p. 724

- Triangle domain $\Omega = \{(x, y) : 0 \leq x \leq 1, -x \leq y \leq x\}$.
- Velocity $\mathbf{v}(\mathbf{x}) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ \Rightarrow (7.2.1) becomes $-\epsilon\Delta u + u_x = 1$.
- Exact solution: $u_\epsilon(x_1, x_2) = x - \frac{1}{1-e^{-1/\epsilon}}(e^{-(1-x_1)/\epsilon} - e^{-1/\epsilon})$, Dirichlet boundary conditions set accordingly
- Standard Galerkin discretization by means of linear finite elements on sequence of triangular mesh created by regular refinement.



Coarse initial mesh



As expected:

spurious oscillations mar Galerkin solution

- Difficulty observed in 1D also haunts discretization in higher dimensions.

Issue: extension of upwinding idea to $d > 1$

7.2.2.1 Upwind quadrature

Revisit 1D model problem

$$-\epsilon \frac{d^2u}{dx^2} + \frac{du}{dx} = f(x) \quad \text{in } \Omega, \quad u(0) = 0, \quad u(1) = 0, \quad (7.2.13)$$

with variational formulation, see Rem. 7.2.2:

$$u \in H_0^1([0, 1]): \quad \underbrace{\epsilon \int_0^1 \frac{du}{dx}(x) \frac{du}{dx}(x) dx}_{=: \mathbf{a}(u, v)} + \underbrace{\int_0^1 \frac{du}{dx}(x) v(x) dx}_{=: \ell(v)} = \int_0^1 f(x) v(x) dx \quad \forall v \in H_0^1([0, 1]).$$

convective term

Linear finite element Galerkin discretization on equidistant mesh \mathcal{M} with M cells, meshwidth $h = \frac{1}{M}$, cf. Sect. 1.5.1.2.

We opt for the composite trapezoidal rule

$$\int_0^1 \psi(x) dx \approx h \sum_{j=1}^{M-1} \psi(jh) , \quad \text{for } \psi \in C^0([0, 1]), \psi(0) = \psi(1) = 0 .$$

for evaluation of convective term in bilinear form a :

$$\int_0^1 \frac{du_N}{dx}(x) v_N(x) dx \approx h \sum_{j=1}^{M-1} \frac{du_N}{dx}(jh) v(hj) , \quad v_N \in \mathcal{S}_{1,0}^0(\mathcal{M}) . \quad (7.2.22)$$

Note: this is not a valid formula, because $\frac{du_N}{dx}(jh)$ is *ambiguous*, since $\frac{du_N}{dx}$ is discontinuous at nodes of the mesh for $u_N \in \mathcal{S}_{1,0}^0(\mathcal{M})$!

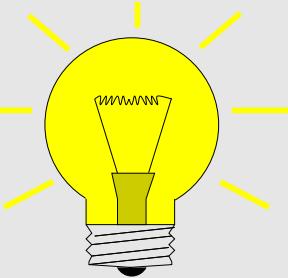
Up to now we resolved this ambiguity by the policy of *local* quadrature, see Sect. 3.5.4: quadrature rule applied locally on each cell with all information taken from that cell.

However:

Convection transports information in the direction of v !

7.2

Idea:



Use upstream/upwind information to evaluate $\frac{du_N}{dx}(jh)$ in (7.2.22)

$$\frac{du_N}{dx}(jh) := \lim_{\delta \rightarrow 0} \frac{du_N}{dx}(jh - \delta) = \frac{du_N}{dx} |_{]x_{j-1}, x_j[} .$$

$\hat{=}$ upwind quadrature

Upwind quadrature yields the following contribution of the discretized convective term to the linear system using the basis expansion $u_N = \sum_{l=1}^{M-1} \mu_l b_N^l$ into *locally supported* nodal basis functions (“tent functions”)

$$\int_0^1 \sum_{l=1}^{M-1} \mu_l \frac{db_N^l}{dx}(x) b_N^i(x) dx \stackrel{(7.2.22)}{\approx} h \frac{\mu_i - \mu_{i-1}}{h} ,$$

where we used

- $b_N^i(jh) = \delta_{ij}$, see (1.5.50),
- $\frac{du_N}{dx} |_{]x_{j-1}, x_j[} = \frac{\mu_i - \mu_{i-1}}{h}$ from (1.5.51).

► Linear system from upwind quadrature:

$$\left(-\frac{\epsilon}{h} - 1\right) \mu_{i-1} + \left(\frac{2\epsilon}{h} + 1\right) \mu_i + -\frac{\epsilon}{h} \mu_{i+1} = h f(ih), \quad i = 1, \dots, M-1, \quad (7.2.18)$$

which is the **same** as that obtained from a backward finite difference discretization of $\frac{du}{dx}$!

The idea of upwind quadrature can be generalized to $d > 1$: we consider $d = 2$ and linear Lagrangian finite element Galerkin discretization on triangular meshes, see Sect. 3.2.

- ① Approximation of contribution of convective terms to bilinear form by means of *global trapezoidal rule*:

$$\int_{\Omega} (\mathbf{v} \cdot \mathbf{grad} u) v \, dx \approx \sum_{\mathbf{p} \in \mathcal{N}(\mathcal{M})} \left(\frac{1}{3} \sum_{K \in \mathcal{U}_{\mathbf{p}}} |K| \right) \cdot \mathbf{v}(\mathbf{p}) \cdot \mathbf{grad} u(\mathbf{p}) v(\mathbf{p}). \quad (7.2.23)$$

7.2
p. 730

ambiguous for $u \in \mathcal{S}_1^0(\mathcal{M})$!

notation: $\mathcal{U}_p := \{K \in \mathcal{M} : p \in \overline{K}\}$

- ② Fix the ambiguous value of $v(p) \cdot \text{grad } u_N(p)$, $u_N \in S_1^0(\mathcal{M})$, by taking the gradient from the triangle upstream to the node p :

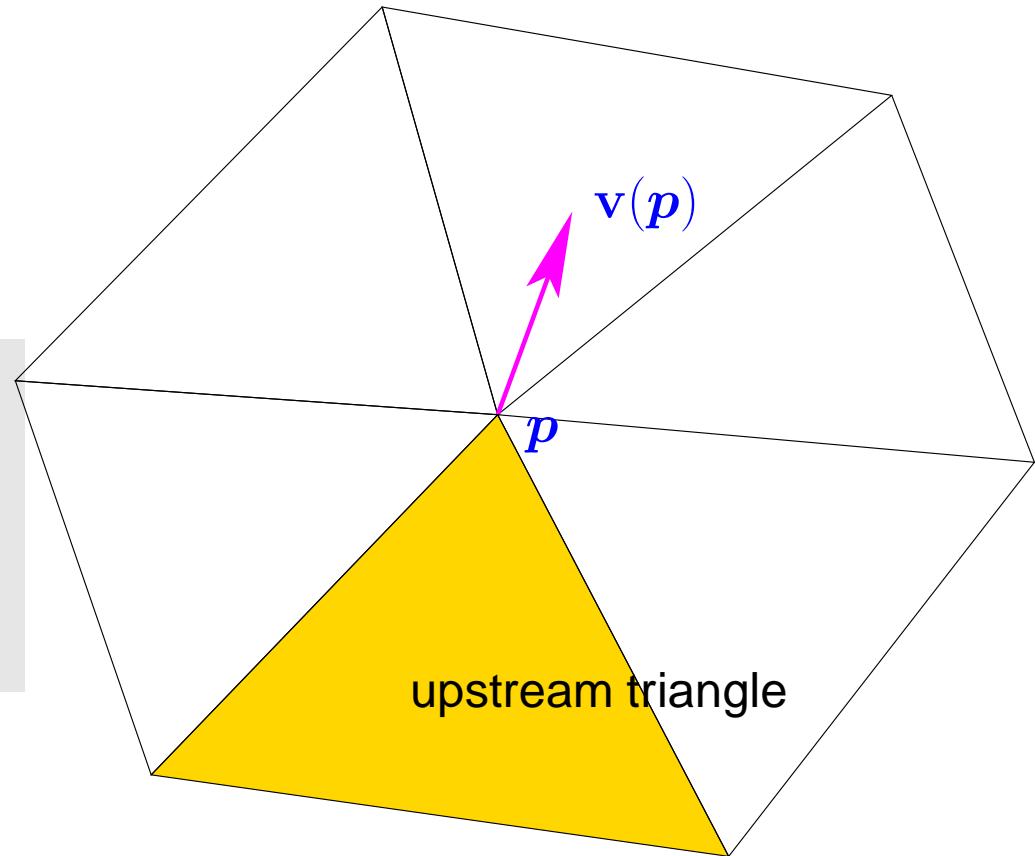
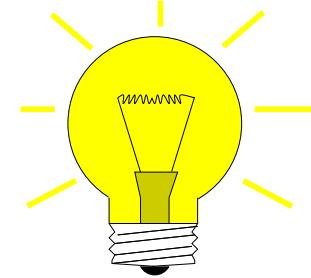


Fig. 215



Idea: Use upstream/upwind information to evaluate $\text{grad } u_N(p)$ in (7.2.23)

$$v(p) \cdot \text{grad } u_N(p) := \lim_{\delta \rightarrow 0} v(p) \cdot \text{grad } u_N(p - \delta v(p)). \quad (7.2.24)$$

$\hat{=}$ general upwind quadrature

Note: By (7.1.1) the vector $\mathbf{v}(\mathbf{p})$ supplies the direction of the streamline through \mathbf{p} . Hence, $-\mathbf{v}(\mathbf{p})$ is the direction from which information is “carried into \mathbf{p} ” by the flow.

Contribution of convective term to the i -th linear of the final linear system of equations (test function = tent function b_N^i)

$$\underbrace{\left(\frac{1}{3} \sum_{K \in \mathcal{U}_i} |K| \right)}_{=: U_i} \mathbf{v}(\mathbf{x}^i) \cdot \mathbf{grad} u_N|_{K_u},$$

where K_u is the upstream triangle of \mathbf{p} .

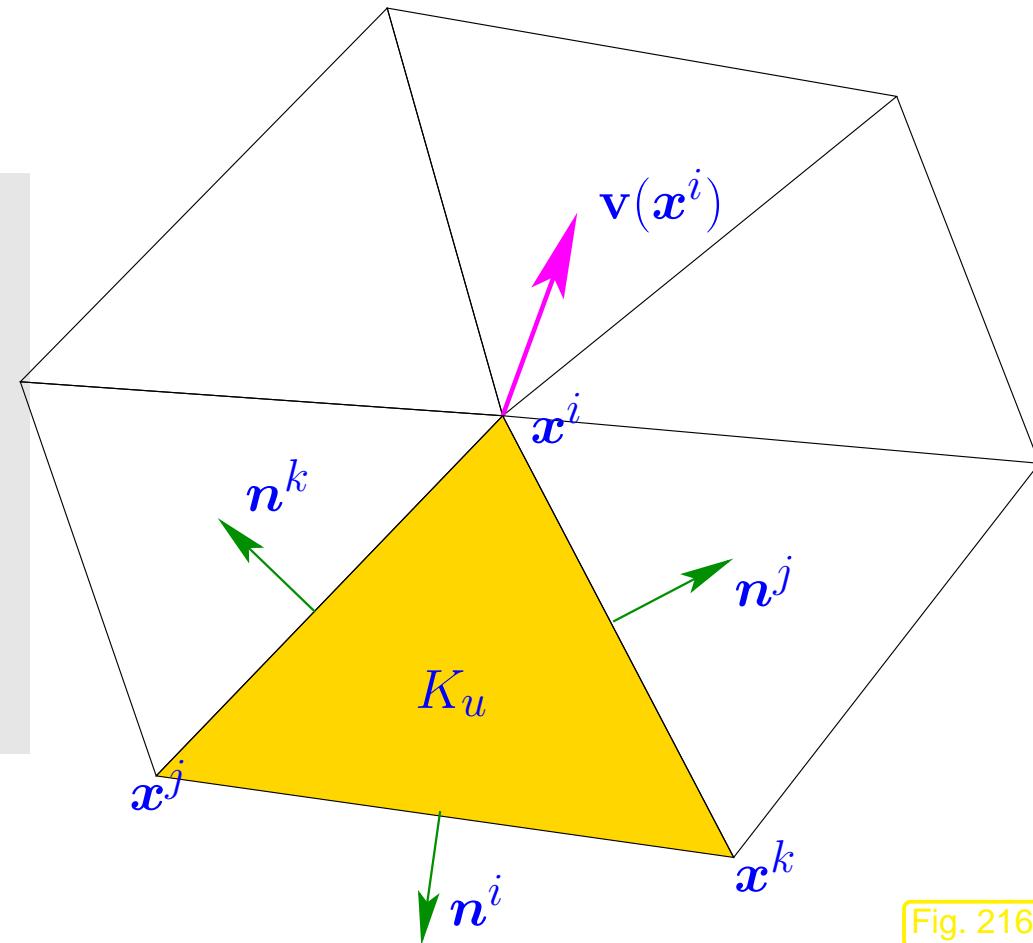


Fig. 216

Using the expressions for the gradients of barycentric coordinate functions from Sect. 3.2.5 and the nodal basis expansion of u_N , we obtain for the convective contribution to the i -th line of the final

linear system

$$\frac{U_i}{2|K_u|} \left(\underbrace{-\|\mathbf{x}^j - \mathbf{x}^k\| \mathbf{n}^i \cdot \mathbf{v}(\mathbf{x}^i) \mu_i - \|\mathbf{x}^i - \mathbf{x}^j\| \mathbf{n}^k \cdot \mathbf{v}(\mathbf{x}^i) \mu_k}_{\leftrightarrow \text{diagonal entry}} - \|\mathbf{x}^i - \mathbf{x}^k\| \mathbf{n}^j \cdot \mathbf{v}(\mathbf{x}^i) \mu_j \right)$$

By the very definition of the upstream triangle K_u we find

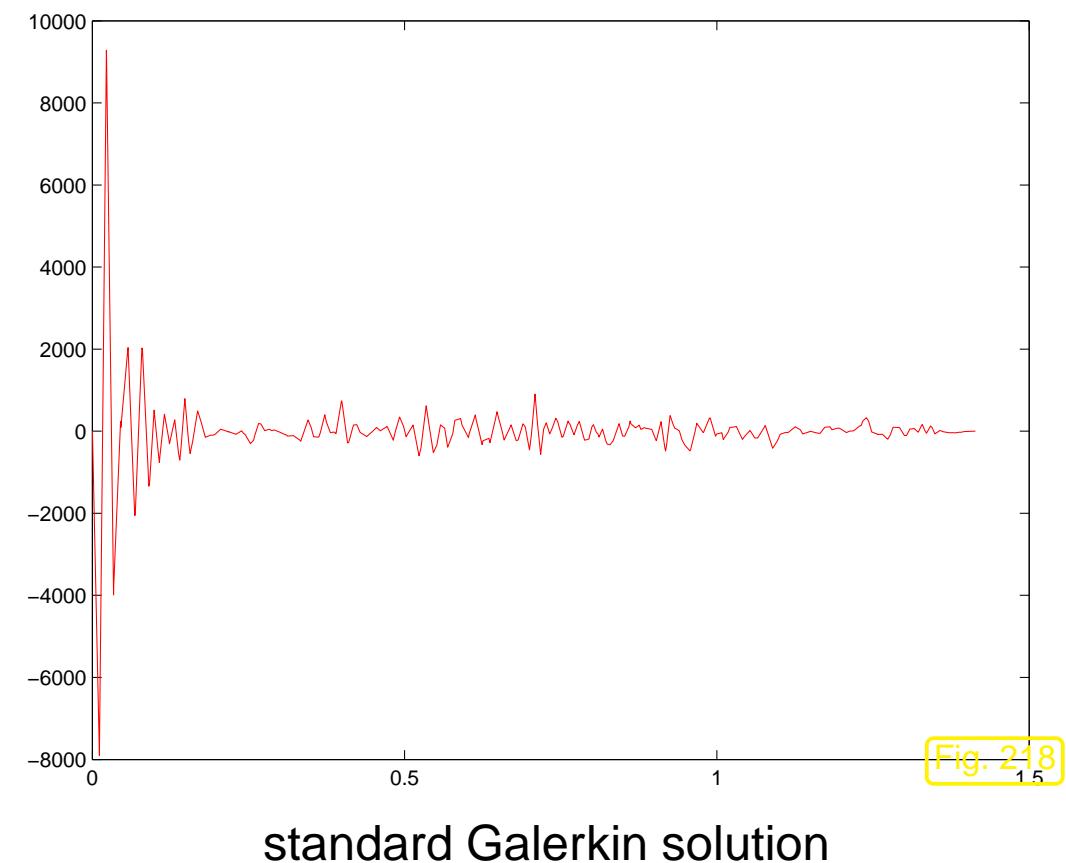
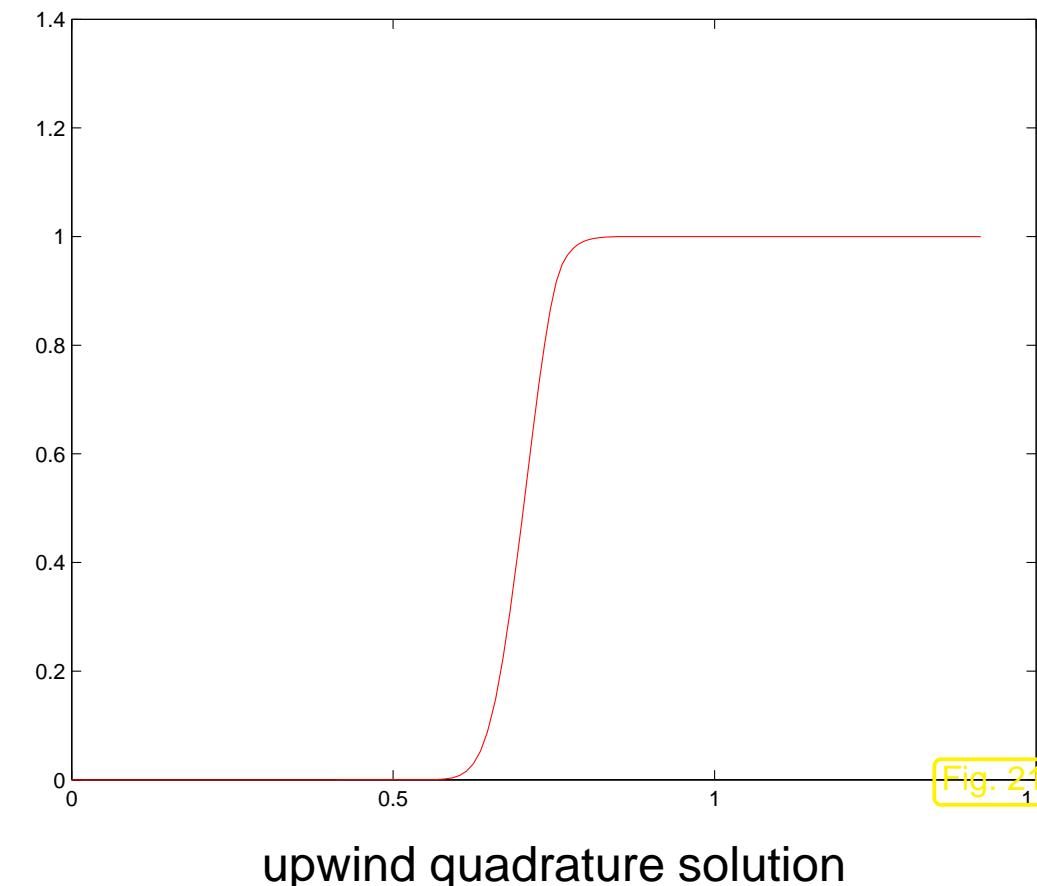
$$\mathbf{n}^i \cdot \mathbf{v}(\mathbf{x}^i) \leq 0 \quad , \quad \mathbf{n}^k \cdot \mathbf{v}(\mathbf{x}^i) \geq 0 \quad , \quad \mathbf{n}^j \cdot \mathbf{v}(\mathbf{x}^i) \geq 0 .$$

➤ sign conditions (5.7.9), (5.7.10) are satisfied, (5.7.11) is obvious.

Example 7.2.25 (Upwind quadrature discretization).

- $\Omega = [0, 1]^2$
- $-\epsilon \Delta u + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \cdot \mathbf{grad} u = 0$
- Dirichlet boundary conditions: $u(x, y) = 1$ for $x > y$ and $u(x, y) = 0$ for $x \leq y$
- Limiting case ($\epsilon \rightarrow 0$): $u(x, y) = 1$ for $x > y$ and $u(x, y) = 0$ for $x \leq y$
- layer along the diagonal from $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ to $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ in the limit $\epsilon \rightarrow 0$
- linear finite element upwind quadrature discretization on triangular mesh.

► Monitored: discrete solutions along diagonal $\binom{0}{1} - \binom{1}{0}$ for $\epsilon = 10^{-10}$.



Upwind quadrature scheme respects maximum principle, whereas the standard Galerkin solution is rendered useless by spurious oscillations.



7.2.2.2 Streamline diffusion

We take another look at the 1D upwind discretization of (7.2.13) and view it from a different perspective.

1D **upwind** (finite difference) discretization of (7.2.13):

$$\left(-\frac{\epsilon}{h} - 1\right)\mu_{i-1} + \left(\frac{2\epsilon}{h} + 1\right)\mu_i + -\frac{\epsilon}{h}\mu_{i+1} = hf(ih) . i = 1, \dots, M - 1 . \quad (7.2.18)$$

$$(\epsilon+h/2) \underbrace{\frac{-\mu_{i-1} + 2\mu_i - \mu_{i+1}}{h^2}}_{\hat{=} \text{ difference quotient for } \frac{d^2u}{dx^2}} + \underbrace{\frac{-\mu_{i-1} + \mu_{i+1}}{2h}}_{\hat{=} \text{ difference quotient for } \frac{du}{dx}} = f(ih) ,$$

for $i = 1, \dots, M - 1$.

Upwinding = h -dependent enhancement of diffusive term

artificial diffusion/viscosity

We also observe that the upwinding strategy just adds the *minimal amount of diffusion* to make the resulting system matrix comply with the conditions (5.7.9)–(5.7.11), which ensure that the discrete solution satisfies the maximum principle.

Issue: How to extend the trick of adding artificial diffusion to $d > 1$?

Well, just add an extra h -dependent multiple of $-\Delta$! Let's try.

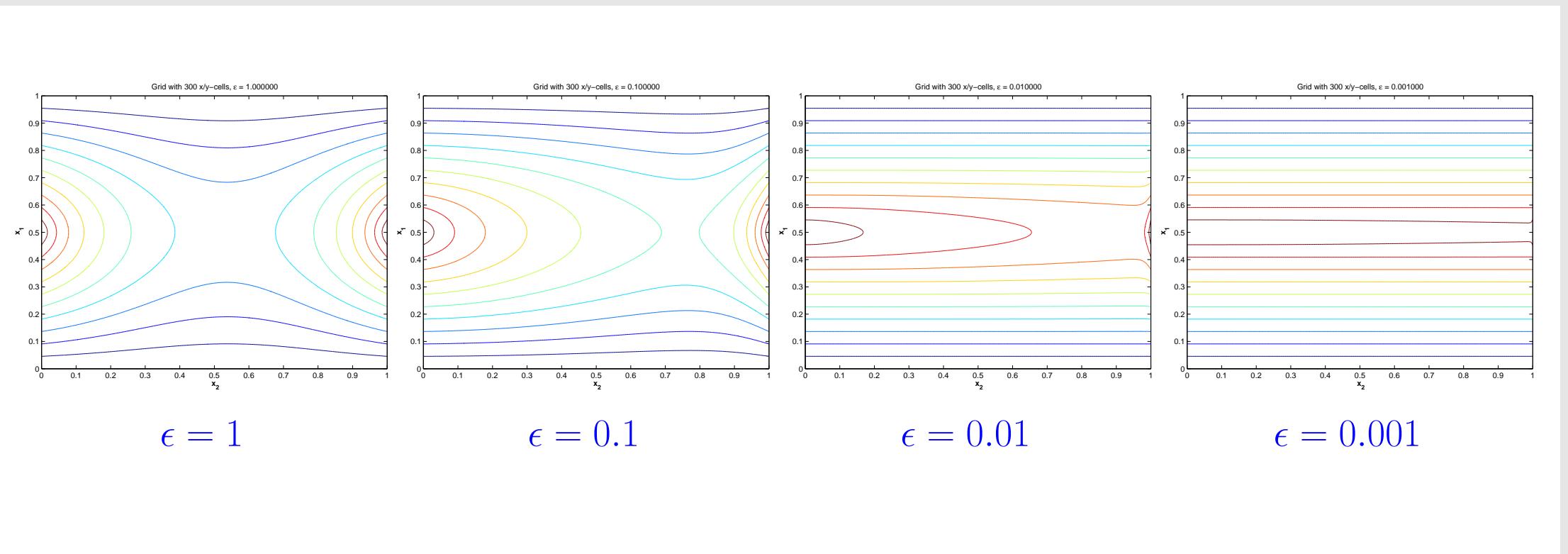
Example 7.2.26 (Effect of added diffusion).

Convection-diffusion boundary value problem ((7.2.1) with $\mathbf{v} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$)

$$-\epsilon\Delta u + \frac{\partial u}{\partial x_1} = 0 \quad \text{in } \Omega =]0, 1[^2, \quad u = g \quad \text{on } \partial\Omega .$$

Here, Dirichlet data $g(\mathbf{x}) = 1 - 2|x_2 - \frac{1}{2}|$.

Thus, for $\epsilon \approx 0$ we expect $u \approx g$, because the Dirichlet data are just transported in x_1 -direction and there are no boundary layers.



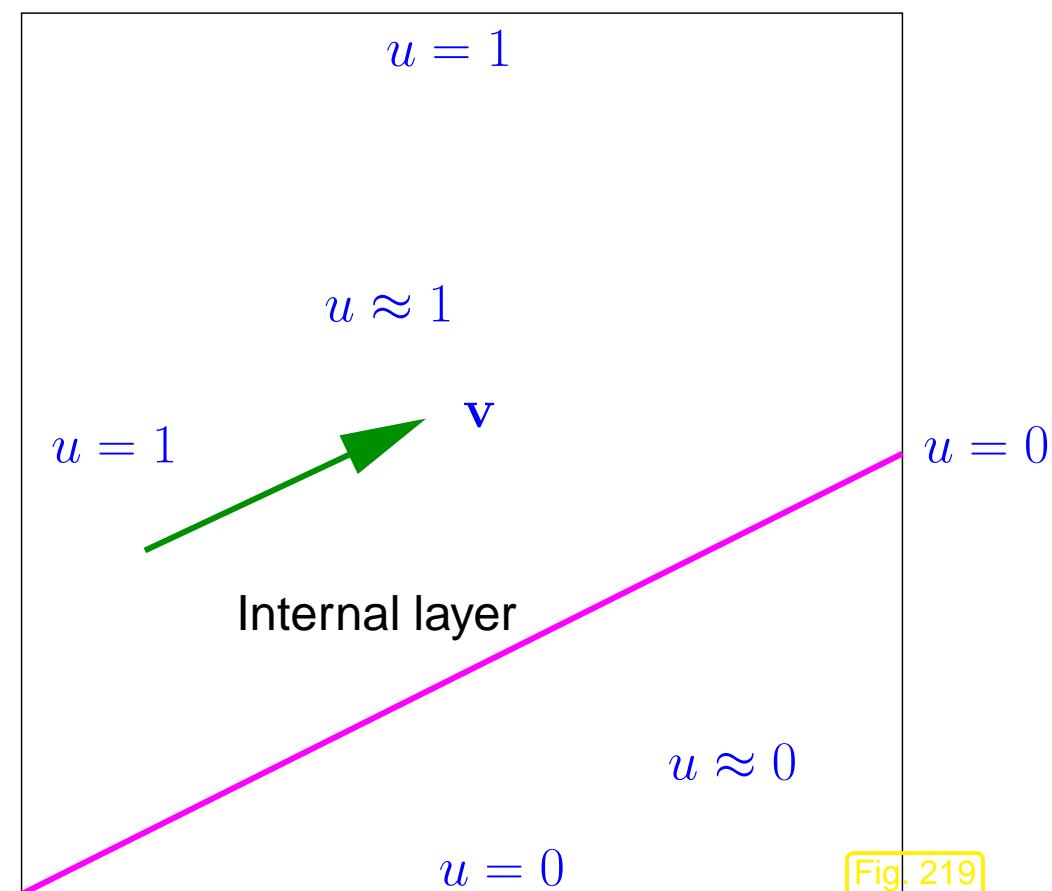
Stronger diffusion leads to “smearing” of features that the flow field transports into the interior of the domain.



(Too much) artificial diffusion \Rightarrow smearing of internal layers

(We are no longer solving the right problem!)

Remark 7.2.27 (Internal layers).



Pure transport problem:

$$\mathbf{v} \cdot \nabla u = 0 \quad \text{in } \Omega,$$

where $\Omega = [0, 1]^2$, $\mathbf{v} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$, $\epsilon = 10^{-4}$,

Dirichlet b.c. that can only partly be fulfilled: $u = 1$ on $\{x_1 = 0\} \cup \{x_2 = 1\}$, $u = 0$ on $\{x_1 = 1\} \cup \{x_2 = 0\}$

Solution of pure transport problem with discontinuous boundary data

- displays a discontinuity across the streamline emanating from the point of discontinuity on $\partial\Omega$,
- is *smooth along streamlines*.

Qualitative solution of

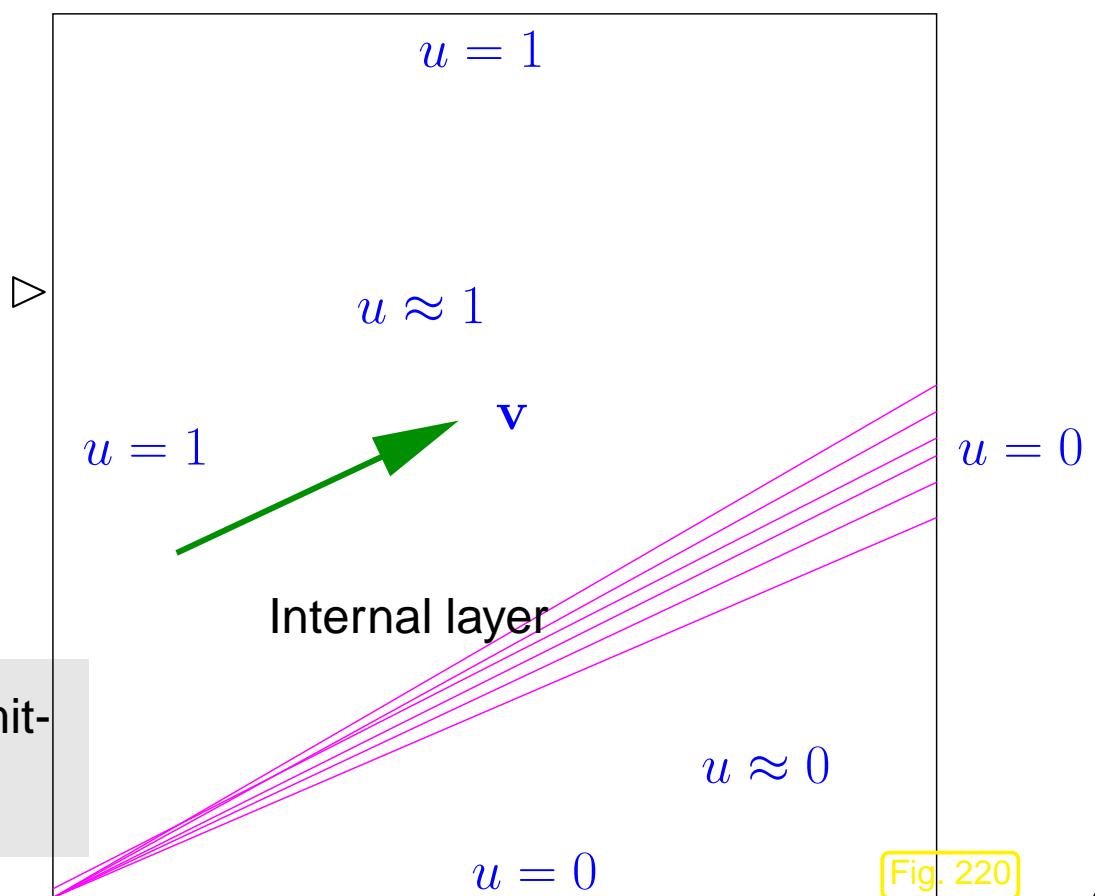
$$-\delta\Delta + \mathbf{v} \cdot \nabla u = 0 \quad \text{in } \Omega,$$

with $\delta > 0$, the same boundary data

➤

Smearing of internal layer !

We would also find a boundary layer which is omitted in the figure.



Heuristics: If the solution is smooth along streamlines, then adding diffusion in the direction of streamlines cannot do much harm.

What does “diffusion in a direction” mean?

☞ Think of a generalized Fourier’s law (2.5.3) for $d = 2$, e.g.,

$$\mathbf{j}(\mathbf{x}) = - \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{\text{grad}} u(\mathbf{x}) .$$

This means, only a temperature variation in \mathbf{x}_1 -direction triggers a heat flow.

→ diffusion in a direction $\mathbf{v} \in \mathbb{R}^2$

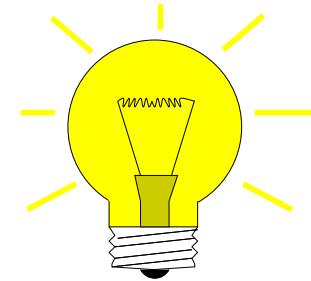
$$\mathbf{j}(\mathbf{x}) = -\mathbf{v}\mathbf{v}^T \mathbf{\text{grad}} u(\mathbf{x}) \quad (7.2.28)$$

Such an extended Fourier’s law is an example of **anisotropic diffusion**.

Anisotropic diffusion can simply be taken into account in variational formulations and Galerkin discretization by replacing the heat conductivity κ /stiffness σ with a symmetric, positive (semi-)definite matrix, the **diffusion tensor**.

Idea:

Anisotropic artificial diffusion in streamline direction



On cell K replace: $\epsilon \leftarrow \underbrace{\epsilon \mathbf{I} + \delta_K \mathbf{v}_K \mathbf{v}_K^T}_{\text{new diffusion tensor}} \in \mathbb{R}^{2,2}$.

$\mathbf{v}_K \hat{=} \text{local velocity (e.g., obtained by averaging)}$

$\delta_K > 0 \hat{=} \text{method parameter controlling the strength of anisotropic diffusion}$

This idea underlies the so-called

streamline-diffusion method

When combined with linear finite element Galerkin discretization ($V_{N,0} \subset \mathcal{S}_1^0(\mathcal{M})$), then it leads to a linear variational problem

$$u_N \in V_{0,N}: \quad \mathbf{a}_{\text{SD}}(u_N, v_N) = \ell_{\text{SG}}(v_N) \quad \forall v_N \in V_{0,N},$$

with the streamline diffusion bilinear form

$$\begin{aligned} \mathbf{a}_{\text{SD}}(u, v) = & \int \epsilon \operatorname{grad} u \cdot \operatorname{grad} v + \operatorname{grad} u \cdot \mathbf{v} v \, dx \\ & + \underbrace{\sum_{K \in \mathcal{M}} \int_K (-\epsilon \Delta u + \delta_K \mathbf{v}_K \cdot \operatorname{grad} u)(\mathbf{v}_K \cdot \operatorname{grad} v) \, dx}_{\text{"stabilizing term"}}, \quad u, v \in H^1(\Omega). \end{aligned}$$

(7.2.29)

7.2

p. 741

Important:

Maintain *consistency* of variational problem!

This means

$$\mathbf{a}_{\text{SD}}(u, v_N) = \ell(v_N) \quad \forall v_N \in V_{0,N} \quad \text{for the exact solution } u \text{ of the BVP.} \quad (7.2.30)$$

To guarantee (7.2.30), we have to modify the standard right-hand-side functional and replace it with the streamline diffusion source functional

$$\ell(v_N) := \int_{\Omega} f(\mathbf{x})v_N(\mathbf{x}) \, d\mathbf{x} + \sum_{K \in \mathcal{M}} \int_K \delta_K f(\mathbf{x})(\mathbf{v}_K \cdot \mathbf{grad} v_N(\mathbf{x})) \, d\mathbf{x}, \quad \mathbf{v}_N \in V_{0,N}. \quad (7.2.31)$$

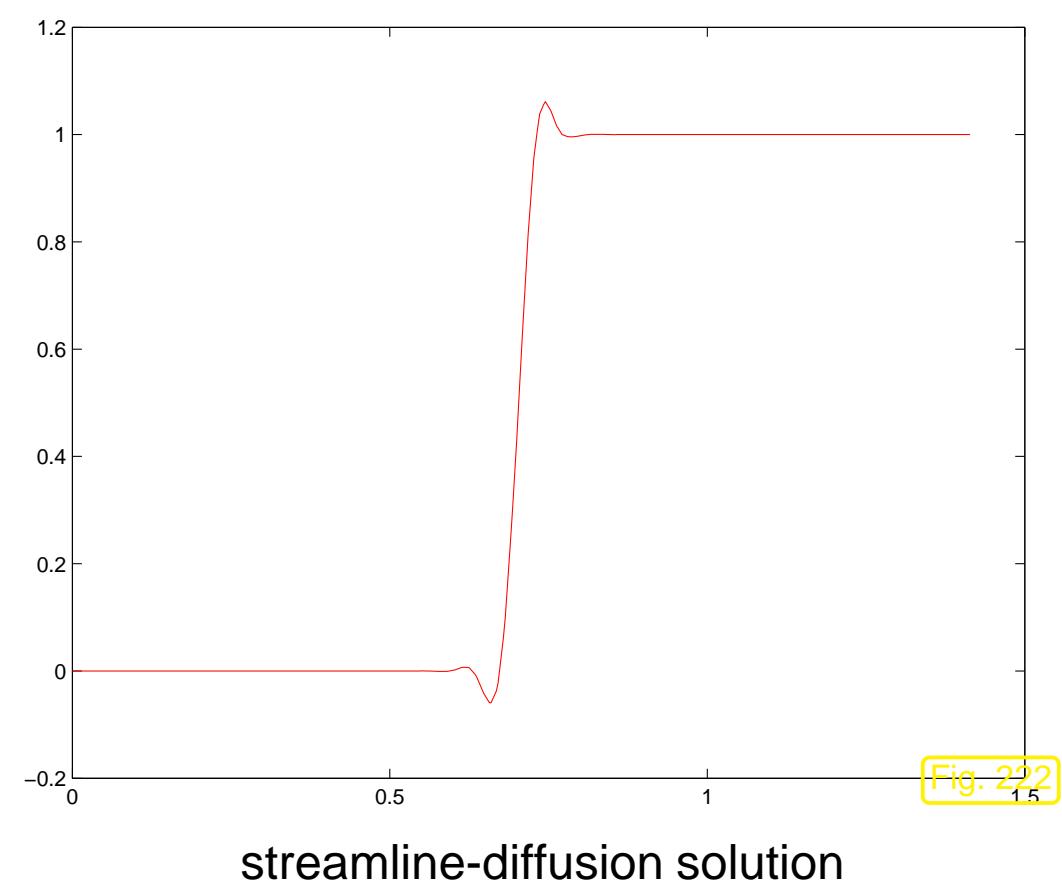
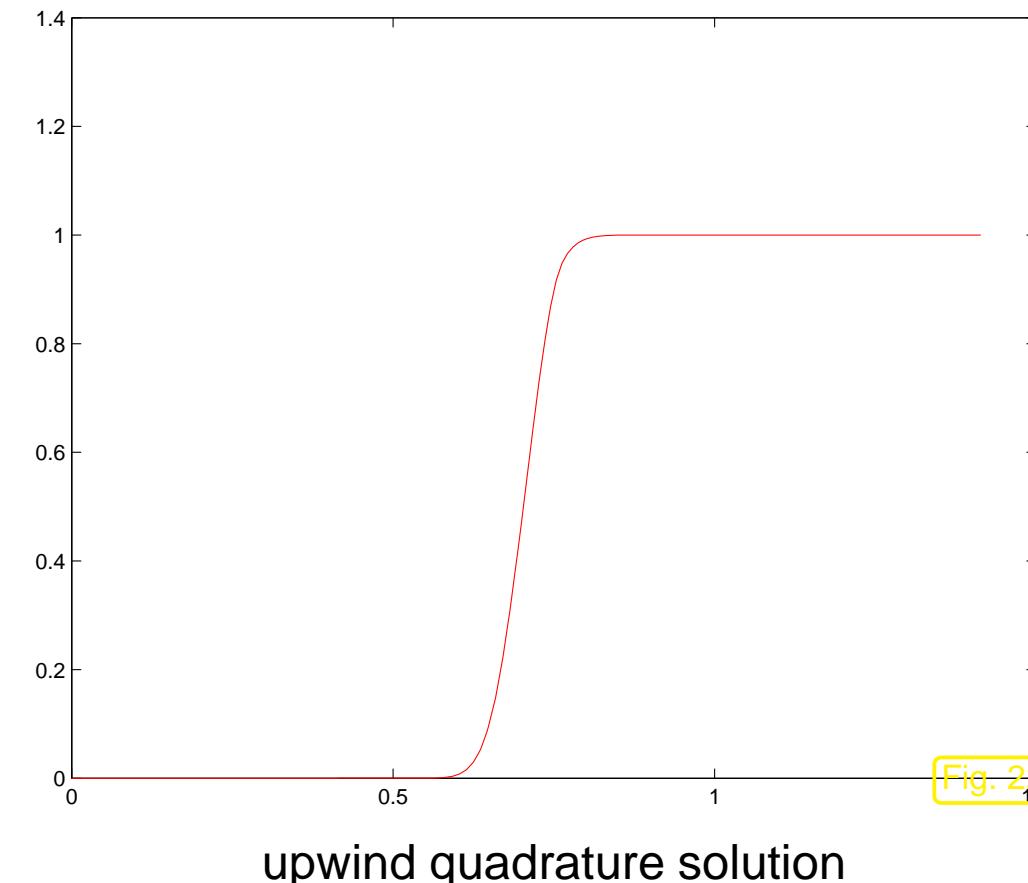
The control parameter is usually chosen according to the rule

$$\delta_K := \begin{cases} \epsilon^{-1} h_K^2 & , \text{ if } \frac{\|\mathbf{v}\|_{K,\infty} h_K}{2\epsilon} \leq 1 , \\ h_K & , \text{ if } \frac{\|\mathbf{v}\|_{K,\infty} h_K}{2\epsilon} > 1 . \end{cases}$$

which is suggested by theoretical investigations and practical experience.

Example 7.2.32 (Streamline-diffusion discretization).

Exactly the same setting as in Ex. 7.2.25 with the upwind quadrature approach replaced with the streamline diffusion method.



Observations:

7.2

p. 743

- The streamline upwind method does not exactly respect the maximum principle, but offers a better resolution of the internal layer compared with upwind quadrature (Parlance: streamline diffusion method is “less diffusive”).



7.3 Transient convection-diffusion BVP

Sect. 7.1.4 introduced the transient heat conduction model in a fluid, whose motion is described by a non-stationary velocity field (\rightarrow Sect. 7.1.1) $\mathbf{v} : \Omega \times]0, T[\mapsto \mathbb{R}^d$

$$\frac{\partial}{\partial t}(\rho u) - \operatorname{div}(\kappa \operatorname{grad} u) + \operatorname{div}(\rho \mathbf{v}(\mathbf{x}, t) u) = f(\mathbf{x}, t) \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[, \quad (7.1.15)$$

7.3

where $u = u(\mathbf{x}, t) : \tilde{\Omega} \mapsto \mathbb{R}$ is the unknown temperature.

Assuming $\operatorname{div} \mathbf{v}(\mathbf{x}, t) = 0$, as in Sect. 7.2, by scaling we arrive at the model equation for transient convection-diffusion

$$\frac{\partial u}{\partial t} - \epsilon \Delta u + \mathbf{v}(\mathbf{x}, t) \cdot \operatorname{grad} u = f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[, \quad (7.3.1)$$

supplemented with

- Dirichlet boundary conditions: $u(\mathbf{x}, t) = g(\mathbf{x}, t) \quad \forall \mathbf{x} \in \partial\Omega, \quad 0 < t < T,$
- initial conditions: $u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega.$

7.3.1 Method of lines

For the solution of IBVP (7.3.1) follow the general policy introduced in Sect. 6.1.3:

- ① Discretization in space on a *fixed* mesh ➤ initial value problem for ODE
- ② Discretization in time (by suitable numerical integrator = timestepping)

For instance, in the case of Dirichlet boundary conditions,

$$\begin{cases} \frac{\partial u}{\partial t} - \epsilon \Delta u + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = f & \text{in } \tilde{\Omega} := \Omega \times]0, T[, \\ u(\mathbf{x}, t) = g(\mathbf{x}, t) & \forall \mathbf{x} \in \partial\Omega, 0 < t < T , \quad u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega . \end{cases} \quad (7.3.2)$$

 ← spatial discretization

$$\mathbf{M} \frac{d\vec{\mu}}{dt}(t) + \epsilon \mathbf{A} \vec{\mu}(t) + \mathbf{B} \vec{\mu}(t) = \vec{\varphi}(t) , \quad (7.3.3)$$

- where
- $\vec{\mu} = \vec{\mu}(t) :]0, T[\mapsto \mathbb{R}^N \hat{=} \text{coefficient vector describing approximation } u_N(t) \text{ of } u(\cdot, t),$
 - $\mathbf{A} \in \mathbb{R}^{N,N} \hat{=} \text{s.p.d. matrix of discretized } -\Delta, \text{ e.g., (finite element) Galerkin matrix,}$
 - $\mathbf{M} \in \mathbb{R}^{N,N} \hat{=} (\text{lumped} \rightarrow \text{Rem. 6.2.34}) \text{ mass matrix}$
 - $\mathbf{B} \in \mathbb{R}^{N,N} \hat{=} \text{matrix for discretized convective term, e.g., Galerkin matrix, upwind quadrature matrix} (\rightarrow \text{Sect. 7.2.2.1}), \text{ streamline diffusion matrix} (\rightarrow \text{Sect. 7.2.2.2}).$

Example 7.3.4 (Implicit Euler method of lines for transient convection-diffusion).

1D convection-diffusion IBVP:

$$\frac{\partial u}{\partial t} - \epsilon \frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial x} = 0, \quad u(x, 0) = \max(1 - 3|x - \frac{1}{3}|, 0), \quad u(0) = u(1) = 0. \quad (7.3.5)$$

- Spatial discretization on equidistant mesh with meshwidth $h = 1/N$:

1. central finite difference scheme, see (7.2.14),
2. upwind finite difference discretization, see (7.2.18),

- $\mathbf{M} = h\mathbf{I}$ (“lumped” mass matrix, see Rem. 6.2.34),
- Temporal discretization with uniform timestep $\tau > 0$:
 1. implicit Euler method, see (6.1.23),
 2. explicit Euler method, see (6.1.22),

Computations with $\epsilon = 10^{-5}$, implicit Euler discretization, $h = 0.01$, $\tau = 0.00125$:

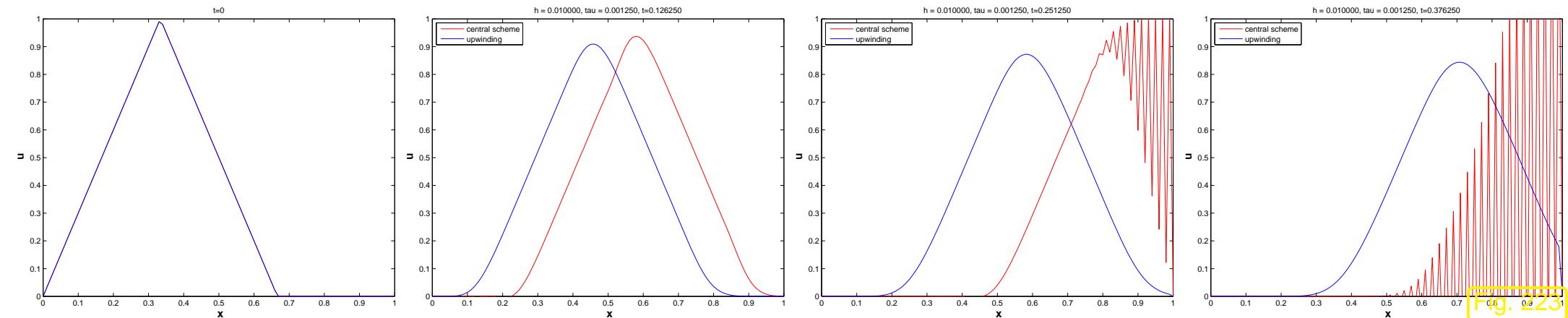
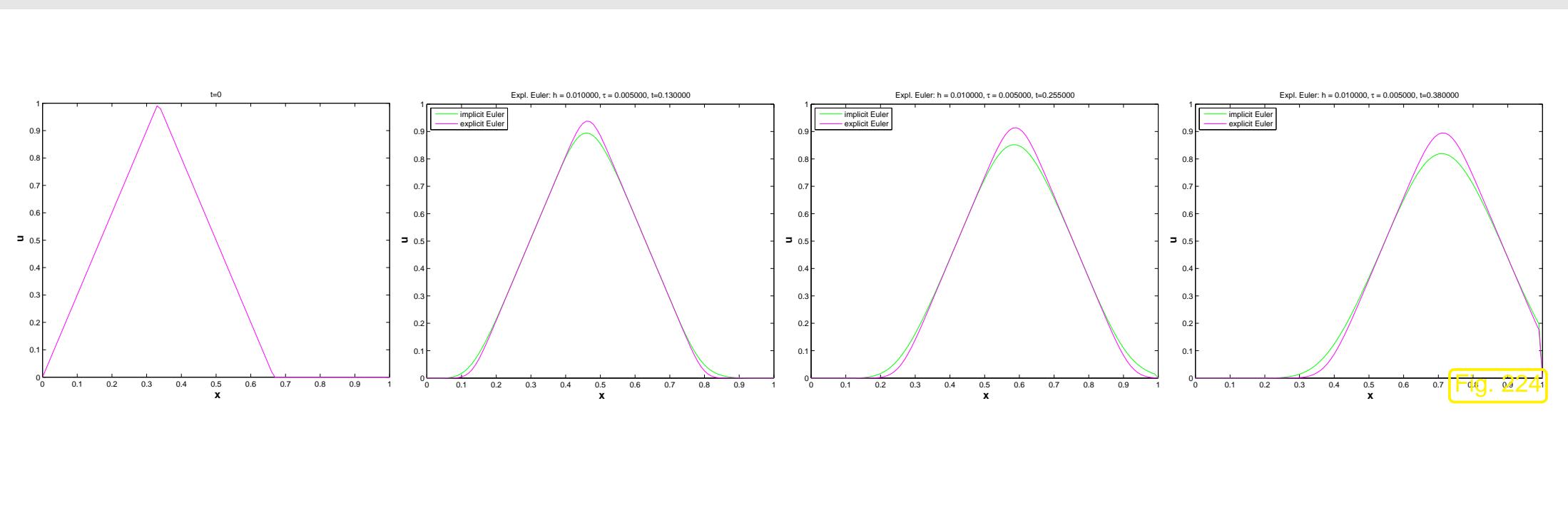


Fig. 7.23

Observation:

- Central finite differences display spurious oscillations as in Ex. 7.2.16.
- Upwinding suppresses spurious oscillations, but introduces *spurious damping*.

Computations with $\epsilon = 10^{-5}$, spatial upwind discretization, $h = 0.01$, $\tau = 0.005$:



Observation: implicit Euler timestepping causes stronger spurious damping than explicit Euler timestepping.

However, explicit Euler subject to tight stability induced timestep constraint for larger values of ϵ , see Sect. 6.1.4.2.



- ▶ Advice for spatial discretization for method of lines approach

Use ϵ -robustly stable spatial discretization of convective term.

Remark 7.3.6 (Choice of timestepping for m.o.l. for transient convection-diffusion).

If ϵ -robustness *for all* $\epsilon > 0$ (including $\epsilon > 1$) desired \geqslant Arguments of Sect. 6.1.4.2 stipulate use of $L(\pi)$ -stable (\rightarrow Def. 6.1.56) timestepping methods (implicit Euler (6.1.23), RADAU-3 (6.1.58), SDIRK-2 (6.1.59))

In the *singularly perturbed case* $0 < \epsilon \ll 1$ conditionally stable explicit timestepping is an option, due to a timestep constraint of the form " $\tau < O(h_M)$ ", which does not interfere with efficiency, *cf.* the discussion in Sect. 6.1.5.



7.3.2 Transport equation

Focus on the situation of **singular perturbation** (\rightarrow Def. 7.2.12): $0 < \epsilon \ll 1$

➤ study limit problem (as in Sect. 7.2.1)

$$\frac{\partial u}{\partial t} - \epsilon \Delta u + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[,$$

 $\leftarrow \epsilon = 0$

$$\frac{\partial u}{\partial t} + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[. \quad (7.3.7)$$

=
transport equation

Now:

focus on case $f \equiv 0$ (no sources)

Let $u = u(\mathbf{x}, t)$ be a C^1 -solution of

$$\frac{\partial u}{\partial t} + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = 0 \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[. \quad (7.3.8)$$

7.3

p. 752

Recall: for the stationary pure transport problem (7.2.5) we found solutions by integrating the source term along streamlines (following the flow direction).

➤ study the behavior of \underline{u} “as seen from a moving fluid particle”

$$t \mapsto u(\mathbf{y}(t), t) , \quad \text{where} \quad \mathbf{y}(t) \quad \text{solves} \quad \frac{d\mathbf{y}}{dt}(t) = \mathbf{v}(\mathbf{y}(t), t) , \quad \text{see (7.1.1)} .$$

By the chain rule

$$\blacktriangleright \quad \frac{d}{dt}u(\mathbf{y}(t), t) = \mathbf{grad} u(\mathbf{y}(t), t) \cdot \frac{d\mathbf{y}}{dt}(t) + \frac{\partial u}{\partial t}(\mathbf{y}(t), t) \quad (7.3.9)$$

$$= \mathbf{grad} u(\mathbf{y}(t), t) \cdot \mathbf{v}(\mathbf{y}(t), t) + \frac{\partial u}{\partial t}(\mathbf{y}(t), t) \stackrel{(7.3.8)}{=} 0 . \quad (7.3.10)$$

A fluid particle “sees” a constant temperature!

Remark 7.3.11 (Solution formula for sourceless transport).

Situation: no inflow/outflow (e.g., fluid in a container)

$$\mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \partial\Omega , \quad 0 < t < T . \quad (7.1.2)$$

- all streamlines will “stay inside Ω ”, flow map Φ^t (7.1.3) defined for all times $t \in \mathbb{R}$.

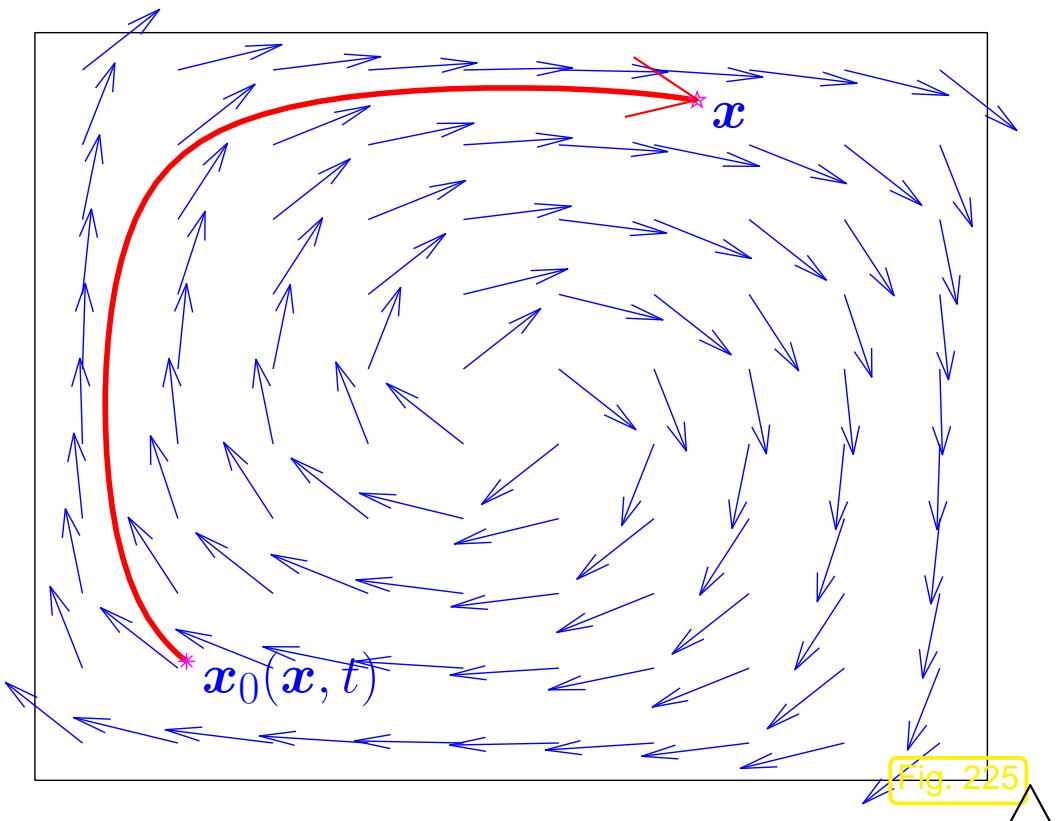
Initial value problem:

$$\mathbf{v}(\mathbf{x}, t) \cdot \operatorname{grad} u = 0 \quad \text{in } \tilde{\Omega} \quad , \quad u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega .$$

Exact solution

$$u(\mathbf{x}, t) = u_0(\mathbf{x}_0(\mathbf{x}, t)) , \quad (7.3.12)$$

where $\mathbf{x}_0(\mathbf{x}, t)$ is the position of the fluid particle that is located in \mathbf{x} at time t at initial time $t = 0$.



This solution formula can be generalized to any divergence free velocity field $\mathbf{v} : \Omega \mapsto \mathbb{R}^d$ and $f \neq 0$. The new aspect is that streamlines can *enter* and *leave* the domain Ω . In the former case the solution value is given by a “transported boundary value”:

$$\frac{d}{dt}u(\mathbf{y}(t)) = f(\mathbf{y}(t), t)$$

$$\blacktriangleright u(\mathbf{x}, t) = \begin{cases} u_0(\mathbf{x}_0) + \int_0^t f(\mathbf{y}(s), s) ds & , \text{ if } \mathbf{y}(s) \in \Omega \quad \forall 0 < s < t , \\ g(\mathbf{y}(s_0), s_0) + \int_{s_0}^t f(\mathbf{y}(s), s) ds & , \text{ if } \mathbf{y}(s_0) \in \partial\Omega, \mathbf{y}(s) \in \Omega \quad \forall s_0 < s < t . \end{cases} \quad (7.3.13)$$

7.3.3 Lagrangian split-step method

Lagrangian discretization schemes for the IBVP (7.3.2) are inspired by insight into the traits of solutions of pure transport problems.

The variant that we are going to study separates the transient convection-diffusion problem into a pure diffusion problem (heat equation → Sect. 6.1.1) and a pure transport problem (7.3.7). This is achieved by means of a particular approach to timestepping.

7.3.3.1 Split-step timestepping

Abstract perspective: consider ODE, whose right hand side is the sum of two (smooth) functions

$$\dot{\mathbf{y}} = \mathbf{g}(t, \mathbf{y}) + \mathbf{h}(t, \mathbf{y}) , \quad \mathbf{g}, \mathbf{h} : \mathbb{R}^m \mapsto \mathbb{R}^m . \quad (7.3.14)$$

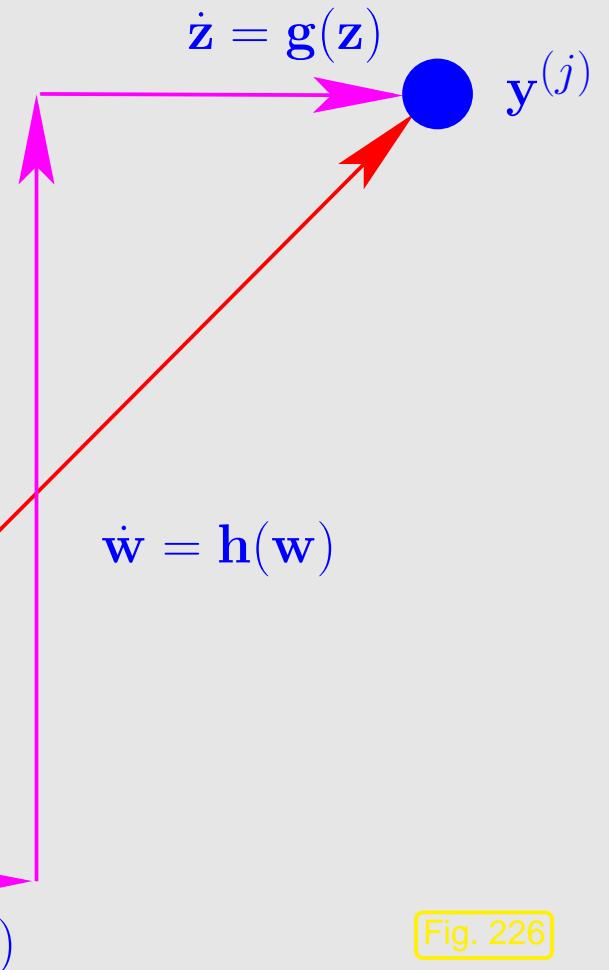
There is an abstract timestepping scheme that offers great benefits if one commands efficient methods to solve initial value problems for both $\dot{\mathbf{z}} = \mathbf{g}(\mathbf{z})$ and $\dot{\mathbf{w}} = \mathbf{h}(\mathbf{w})$.

Strang splitting single step method for (7.3.14), timestep $\tau := t_j - t_{j-1} > 0$: compute $\mathbf{y}^{(j)} \approx \mathbf{y}(t_j)$ from $\mathbf{y}^{(j-1)} \approx \mathbf{y}(t_{j-1})$ according to

$$\tilde{\mathbf{y}} := \mathbf{z}(t_{j-1} + \frac{1}{2}\tau) , \quad \text{where } \mathbf{z}(t) \text{ solves } \dot{\mathbf{z}} = \mathbf{g}(t, \mathbf{z}) , \quad \mathbf{z}(t_{j-1}) = \mathbf{y}^{(j-1)} , \quad (7.3.15)$$

$$\hat{\mathbf{y}} := \mathbf{w}(t_j) \quad \text{where } \mathbf{w}(t) \text{ solves } \dot{\mathbf{w}} = \mathbf{h}(t, \mathbf{w}) , \quad \mathbf{w}(t_{j-1}) = \tilde{\mathbf{y}} , \quad (7.3.16)$$

$$\mathbf{y}^{(j)} := \mathbf{z}(t_j) , \quad \text{where } \mathbf{z}(t) \text{ solves } \dot{\mathbf{z}} = \mathbf{g}(t, \mathbf{z}) , \quad \mathbf{z}(t_{j-1} + \frac{1}{2}\tau) = \hat{\mathbf{y}} . \quad (7.3.17)$$



One timestep involves three sub-steps:

1. Solve $\dot{\mathbf{z}} = \mathbf{g}(t, \mathbf{z})$ over time $[t_{j-1}, t_{j-1} + \frac{1}{2}\tau]$ using the result of the previous timestep as initial value \leftrightarrow (7.3.15).
2. Solve $\dot{\mathbf{w}} = \mathbf{h}(t, \mathbf{w})$ over time τ using the result of 1. as initial value \leftrightarrow (7.3.16).
3. Solve $\dot{\mathbf{z}} = \mathbf{g}(t, \mathbf{z})$ over time $[t_{j-1} + \frac{1}{2}\tau, t_j]$ using the result of 2. as initial value \leftrightarrow (7.3.17).

Fig. 226

Theorem 7.3.18 (Order of Strang splitting single step method).

Assuming exact solution of the initial value problems of the sub-steps, the Strang splitting single step method is of **second order**.

This applies to Strang splitting timestepping for initial value problems for ODEs. Now we boldly regard (7.3.2) as an “*ODE in function space*” for the unknown “function space valued function” $u = u(t) : [0, T] \mapsto H^1(\Omega)$.

$$\frac{du}{dt} = \epsilon \Delta u + f - \mathbf{v} \cdot \mathbf{grad} u$$

$$\overset{\uparrow}{\dot{\mathbf{y}}} = \overset{\uparrow}{\mathbf{g}(\mathbf{y})} + \overset{\uparrow}{\mathbf{h}(\mathbf{y})}$$

Formally, we arrive at the following “timestepping scheme in function space” on a temporal mesh $0 = t_0 < t_1 < \dots < t_M := T$:

Given approximation $u^{(j-1)} \approx u(t_{j-1})$,

① Solve (autonomous) IBVP for *pure diffusion* from t_{j-1} to $t_{j-1} + \frac{1}{2}\tau$

$$\begin{aligned} \frac{\partial w}{\partial t} - \epsilon \Delta w &= 0 \quad \text{in } \Omega \times]t_{j-1}, t_{j-1} + \frac{1}{2}\tau[, \\ (7.3.15) \leftrightarrow w(\mathbf{x}, t) &= g(\mathbf{x}, t_{j-1}) \quad \forall \mathbf{x} \in \partial\Omega, t_{j-1} < t < t_{j-1} + \frac{1}{2}\tau , \\ w(\mathbf{x}, t_{j-1}) &= u^{(j-1)}(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega bdot \end{aligned} \tag{7.3.19}$$

② Solve IBVP for *pure transport* (= advection), see Sect. 7.3.2,

$$(7.3.16) \quad \leftrightarrow \quad \begin{aligned} \frac{\partial z}{\partial t} + \mathbf{v}(\mathbf{x}, t) \cdot \operatorname{grad} z &= f(\mathbf{x}, t) \quad \text{in } \Omega \times]t_{j-1}, t_j[, \\ z(\mathbf{x}, t) &= g(\mathbf{x}, t) \quad \text{on inflow boundary } \Gamma_{\text{in}}, t_{j-1} < t < t_j , \\ z(\mathbf{x}, t_{j-1}) &= w(\mathbf{x}, t_{j-1} + \frac{1}{2}\tau) \quad \forall \mathbf{x} \in \Omega . \end{aligned} \quad (7.3.20)$$

③ Solve IBVP for *pure diffusion* from $t_{j-1} + \frac{1}{2}\tau$ to t_j

$$(7.3.17) \quad \leftrightarrow \quad \begin{aligned} \frac{\partial w}{\partial t} - \epsilon \Delta w &= 0 \quad \text{in } \Omega \times]t_{j-1} + \frac{1}{2}\tau, t_j[, \\ w(\mathbf{x}, t) &= g(\mathbf{x}, t_j) \quad \forall \mathbf{x} \in \partial\Omega, t_{j-1} + \frac{1}{2}\tau < t < t_j , \\ w(\mathbf{x}, t_{j-1} + \frac{1}{2}\tau) &= z(\mathbf{x}, t_j) \quad \forall \mathbf{x} \in \Omega . \end{aligned} \quad (7.3.21)$$

Then set $u^{(j)}(\mathbf{x}) := w(\mathbf{x}, t_j), \mathbf{x} \in \Omega$.

Efficient “implementation” of Strang splitting timestepping, if $\mathbf{g} = \mathbf{g}(\mathbf{y})$:

combine last sub-step with first sub-step of next timestep

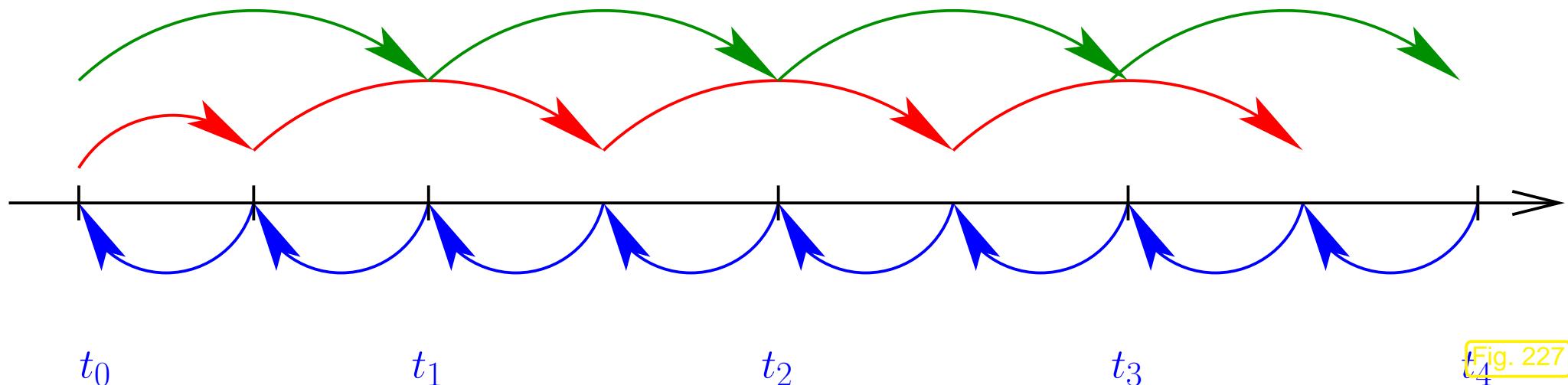


Fig. 227

Remark 7.3.22 (Approximate sub-steps for Strang splitting time).

The solutions of the initial value problems in the sub-steps of Strang splitting timestepping may be computed *only approximately*.

If this is done by one step of a 2nd-order timestepping method in each case, then the resulting approximate Strang splitting timestepping will still be of second order, cf. Thm. 7.3.18.

7.3.3.2 Particle method for advection

Recall the discussion of the IBVP for the pure transport (= advection) equation from Sect. 7.3.2

$$\begin{aligned} \frac{\partial u}{\partial t} + \mathbf{v}(\mathbf{x}, t) \cdot \operatorname{grad} u &= f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[, \\ u(\mathbf{x}, t) &= g(\mathbf{x}, t) \quad \text{on } \Gamma_{\text{in}} \times]0, T[, \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}) \quad \text{in } \Omega , \end{aligned} \tag{7.3.23}$$

with inflow boundary

$$\Gamma_{\text{in}} := \{\mathbf{x} \in \partial\Omega : \mathbf{v}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0\} . \tag{7.2.9}$$

Case $f \equiv 0$: a travelling fluid particle sees a constant solution

$$\blacktriangleright u(\mathbf{x}, t) = \begin{cases} u_0(\mathbf{x}_0) & , \text{if } \mathbf{y}(s) \in \Omega \quad \forall 0 < s < t , \\ g(\mathbf{y}(s_0), s_0) & , \text{if } \mathbf{y}(s_0) \in \partial\Omega, \mathbf{y}(s) \in \Omega \quad \forall s_0 < s < t , \end{cases} \tag{7.3.24}$$

where $s \mapsto \mathbf{y}(s)$ solves the initial value problem $\frac{d\mathbf{y}}{ds}(s) = \mathbf{v}(\mathbf{y}(s), s)$, $\mathbf{y}(t) = \mathbf{x}$ (“backward particle trajectory”).

Case of general f : Since $\frac{d}{dt}u(\mathbf{y}(t)) = f(\mathbf{y}(t), t)$

$$\blacktriangleright u(\mathbf{x}, t) = \begin{cases} u_0(\mathbf{x}_0) + \int_0^t f(\mathbf{y}(s), s) ds & , \text{ if } \mathbf{y}(s) \in \Omega \quad \forall 0 < s < t , \\ g(\mathbf{y}(s_0), s_0) + \int_{s_0}^t f(\mathbf{y}(s), s) ds & , \text{ if } \mathbf{y}(s_0) \in \partial\Omega, \mathbf{y}(s) \in \Omega \quad \forall s_0 < s < t . \end{cases} \quad (7.3.13)$$

The solution formula (7.3.13) suggests an approach for solving (7.3.23) approximately.

We first consider the simple situation of no inflow/outflow (e.g., fluid in a container, see Rem. 7.3.11)

$$\mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \partial\Omega , \quad 0 < t < T . \quad (7.1.2)$$

① Pick suitable interpolation nodes $\{\mathbf{p}_i\}_{i=1}^N \subset \Omega$ (initial ‘particle positions’)

② Solve initial value problems (cf. ODE (7.1.1) for particle trajectories)

$$\dot{\mathbf{y}}(t) = \mathbf{v}(\mathbf{y}(t), t) , \quad \mathbf{y}(0) = \mathbf{p}_i , \quad i = 1, \dots, N ,$$

by means of a suitable single-step method with uniform timestep $\tau := T/M, M \in \mathbb{N}$.

➤ sequences of solution points $\mathbf{p}_i^{(j)}, j = 0, \dots, M, i = 1, \dots, N$

③ Reconstruct approximation $u_N^{(j)} \approx u(\cdot, t_j), t_j := j\tau$, by interpolation:

$$u_N^{(j)}(\mathbf{p}_i^{(j)}) := u_0(\mathbf{p}_i) + \tau \sum_{l=1}^{j-1} f\left(\frac{1}{2}(\mathbf{p}_i^{(l)} + \mathbf{p}_i^{(l-1)}), \frac{1}{2}(t_l + t_{l-1})\right) , \quad i = 1, \dots, N$$

where the composite midpoint quadrature rule was used to approximate the source integral in (7.3.13).

This method falls into the class of

- **particle methods**, because the interpolation nodes can be regarded fluid particles tracked by the method,
- **Lagrangian methods**, which treat the IVP in coordinate systems moving with the flow,
- **characteristic method**, which reconstruct the solution from knowledge about its behavior along streamlines.

For general velocity field $\mathbf{v} : \Omega \mapsto \mathbb{R}^d$:

- Stop tracking i -th trajectory as soon as an interpolation nodes $\mathbf{p}_i^{(j)}$ lies outside spatial domain Ω .
- In each timestep start new trajectories from fixed locations on inflow boundary Γ_{in} (“particle injection”). These interpolation nodes will carry the boundary value.

Example 7.3.25 (Point particle method for pure advection).

- IBVP (7.3.23) on $\Omega =]0, 1[^2$, $T = 2$, with $f \equiv 0$, $g \equiv 0$.
- Initial locally supported bump $u_0(\mathbf{x}) = \max\{0, 1 - 4 \|\mathbf{x} - (1/2, 1/4)\|\}$.
- Two stationary divergence-free velocity fields
 - $\mathbf{v}_1(\mathbf{x}) = \begin{pmatrix} -\sin(\pi x_1) \cos(\pi x_2) \\ \cos(\pi x_1) \sin(\pi x_2) \end{pmatrix}$ satisfying (7.1.2),
 - $\mathbf{v}_2(\mathbf{x}) = \begin{pmatrix} -x_2 \\ x_1 \end{pmatrix}$.

- Initial positions of interpolation points on regular tensor product grid with meshwidth $h = \frac{1}{40}$.
- Approximation of trajectories by means of explicit trapezoidal rule [14, Eq. 11.4.3] (method of Heun).

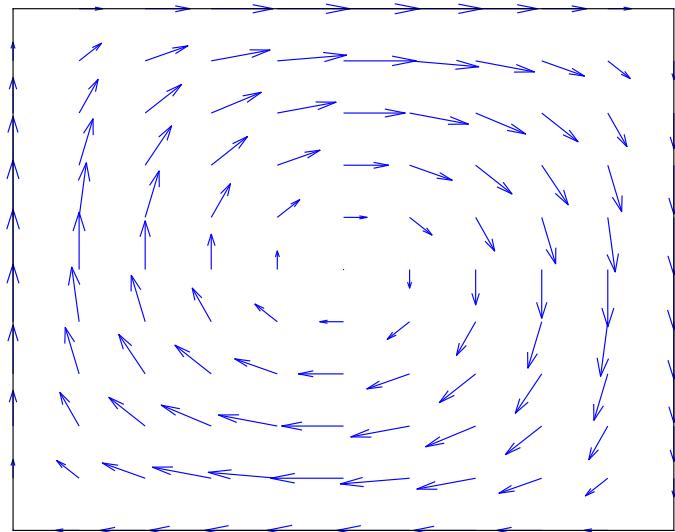


Fig. 228

velocity field \mathbf{v}_1

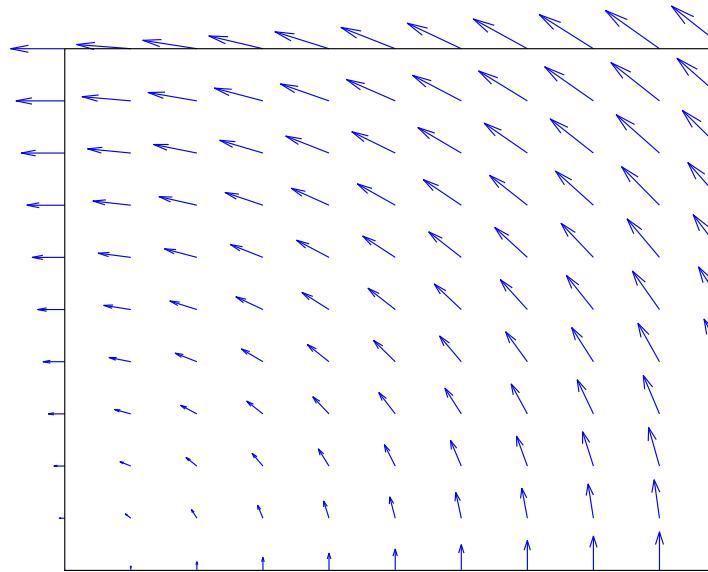


Fig. 229

velocity field \mathbf{v}_2

Code 7.3.26: Point particle method for pure advection

```

1 function partadv(v,u0,g,n,tau,m)
2 % Point particle method for pure advection problem
3 % on the unit square

```

```

4 % v: handle to a function returning the velocity field for (an array) of
5 % points
6 % u0: handle to a function returning the initial value  $u_0$  for (an array)
7 % of points
8 % g: handle to a function  $g = g(\mathbf{x})$  returning the Dirichlet boundary values
9 % n:  $h = 1/n$  is the grid spacing of the initial point distribution
10 % tau: timestep size, m: number of timesteps, that is,  $T = m\tau$ 
11
12 % Initialize points
13 h = 1/n; [Xp,Yp] = meshgrid(0:h:1,0:h:1);
14 P = [reshape(Xp,1,(n+1)^2);reshape(Yp,1,(n+1)^2)];
15
16 % Initialize points on the boundary
17 BP = [[[0:h:1);zeros(1,n+1)], [ones(1,n+1);(0:h:1)],...
18 % ...
19 [ (0:h:1);ones(1,n+1)], [zeros(1,n+1);(0:h:1)]];
20 U = u0(P); % Initial values
21
22 % Plot velocity field
23 hp = 1/10; [Xp,Yp] = meshgrid(0:hp:1,0:hp:1);
24 Up = zeros(size(Xp)); Vp = zeros(size(Xp));
25 for i=0:10, for j=0:10
26     x = v([Xp(i+1,j+1);Yp(i+1,j+1)]);
27     Up(i+1,j+1) = x(1); Vp(i+1,j+1) = x(2);
28 end; end
29 figure('name','velocity field','renderer','painters');
30 quiver(Xp,Yp,Up,Vp,'b-'); set(gca,'fontsize',14); hold on;

```

```

28 plot([0 1 1 0 0],[0 0 1 1 0], 'k-');
29 axis([-0.1 1.1 -0.1 1.1]);
30 xlabel(' {\bf x}_1' ); ylabel(' {\bf x}_2' );
31 axis off;
32
33 fp = figure('name','particles','renderer','painters');
34 fs = figure('name','solution','renderer','painter');
35
36 % Visualize points (interior points in red, boundary points in blue)
37 figure(fp); plot(P(1,:),P(2,:),'r+',BP(1,:),BP(2,:),'b*');
38 title(sprintf('n = %i, t = %f, \\\tau = %f, %i
39 points',n,0,tau,size(P,2)));
40 drawnow; pause;
41
42 % Visualize solution
43 figure(fs); plotpartsol(P,U); drawnow;
44
45 t = 0;
46 for l=1:m
47 % Advect points (explicit trapezoidal rule)
48 P1 = P + tau/2*v(P); P = P + tau*v(P1);
49
50 % Remove points on the boundary or outside the domain
51 Pnew = []; Unew = []; l = 1;

```

```

51 for p=P
52   if ((p(1) > eps) || (p(1) < 1-eps) || (p(2) > eps) || (p(2) <
53     1-eps))
54     Pnew = [Pnew,p]; Unew = [Unew; U(l)];
55   end
56   l = l+1;
57 end

58 % Add points on the boundary (particle injection)
59 P = [Pnew, BP]; U = [Unew; g(BP)];

60 % Visualize points
61 figure(fp); plot(P(1,:),P(2,:),'r+',BP(1,:),BP(2,:),'b*');
62 title(sprintf('n = %i, t = %f, \\\tau = %f, %i
63   points',n,t,tau,size(P,2)));
64 drawnow;
65 % Visualize solution
66 figure(fs); plotpartsol(P,U); drawnow;

67
68 t = t+tau;
69 end

```

partadv(@circvel,@initvals,@bdvals,40,0.025,80);

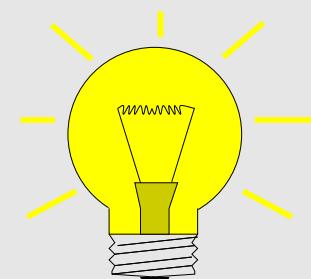


7.3.3.3 Particle mesh method

The method introduced in the previous section, can be used to tackle the pure advection problem (7.3.20) in the 2nd sub-step of the Strang splitting timestepping.

Issue: How to combine Lagrangian advection with a method for the pure diffusion problem (7.3.19) faced in the other sub-steps of the Strang splitting timestepping?

Idea: two views



“particle temperatures” $u(\mathbf{p}_i^{(j)})$



Nodal values of finite element function $u_N^{(j)} \in \mathcal{S}_1^0(\mathcal{M})$

- Outline: algorithm for one step of size $\tau > 0$ of Strang splitting timestepping for transient convection-diffusion problem

$$\begin{cases} \frac{\partial u}{\partial t} - \epsilon \Delta u + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = f & \text{in } \tilde{\Omega} := \Omega \times [0, T], \\ u(\mathbf{x}, t) = 0 \quad \forall \mathbf{x} \in \partial\Omega, 0 < t < T, \quad u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega. \end{cases} \quad (7.3.27)$$

- ① Given
- triangular mesh $\mathcal{M}^{(j-1)}$ of Ω ,
 - $u_N^{(j-1)} \in \mathcal{S}_{1,0}^0(\mathcal{M}^{(j-1)}) \leftrightarrow$ coefficient vector $\vec{\mu}^{(j-1)} \in \mathbb{R}^{N_{j-1}}$,
- approximately solve (7.3.19) by a single step of implicit Euler (??) (size $\frac{1}{2}\tau$)

$$\vec{\nu} = (\mathbf{M} + \frac{1}{2}\tau\epsilon\mathbf{A})^{-1} \vec{\mu}^{(j-1)},$$

where $\mathbf{A} \in \mathbb{R}^{N_{j-1}, N_{j-1}} \hat{=} \mathcal{S}_{1,0}^0(\mathcal{M})$ -Galerkin matrix for $-\Delta$, $\mathbf{M} \hat{=} (\text{possibly lumped}) \mathcal{S}_{1,0}^0(\mathcal{M})$ -mass matrix.

- ② Lagrangian advection step (of size τ) for (7.3.20) with

- initial “particle positions” \mathbf{p}_i given by nodes of $\mathcal{M}^{(j-1)}$, $i = 1, \dots, N_j$,
- initial “particle temperatures” given by corresponding coefficients ν_i .

③ *Remeshing*: advection step has moved nodes to new positions $\tilde{\mathbf{p}}_i$ (and, maybe, introduced new nodes by “particle injection”, deleted nodes by “particle removal”).

➢ Create **new** triangular mesh $\mathcal{M}^{(j)}$ with nodes $\tilde{\mathbf{p}}_i$ (+ boundary nodes), $i = 1, \dots, N_j$

④ Repeat diffusion step ① starting with $w_N \in \mathcal{S}_{1,0}^0(\mathcal{M}^{(j)})$ = linear interpolant (\rightarrow Def. 5.3.13) of “particle temperatures” on $\mathcal{M}^{(j)}$.

➢ new approximate solution $u_N^{(j)}$

Example 7.3.28 (Delaunay-remeshing in 2D).

Delaunay algorithm for creating a 2D triangular mesh *with prescribed nodes*:

- ① Compute Voronoi cells, see (4.2.3) &
<http://www.qhull.org/>.
 - ② Connect two nodes, if their associated Voronoi dual cells have an edge in common.
- MATLAB `TRI = delaunay(x,y)`

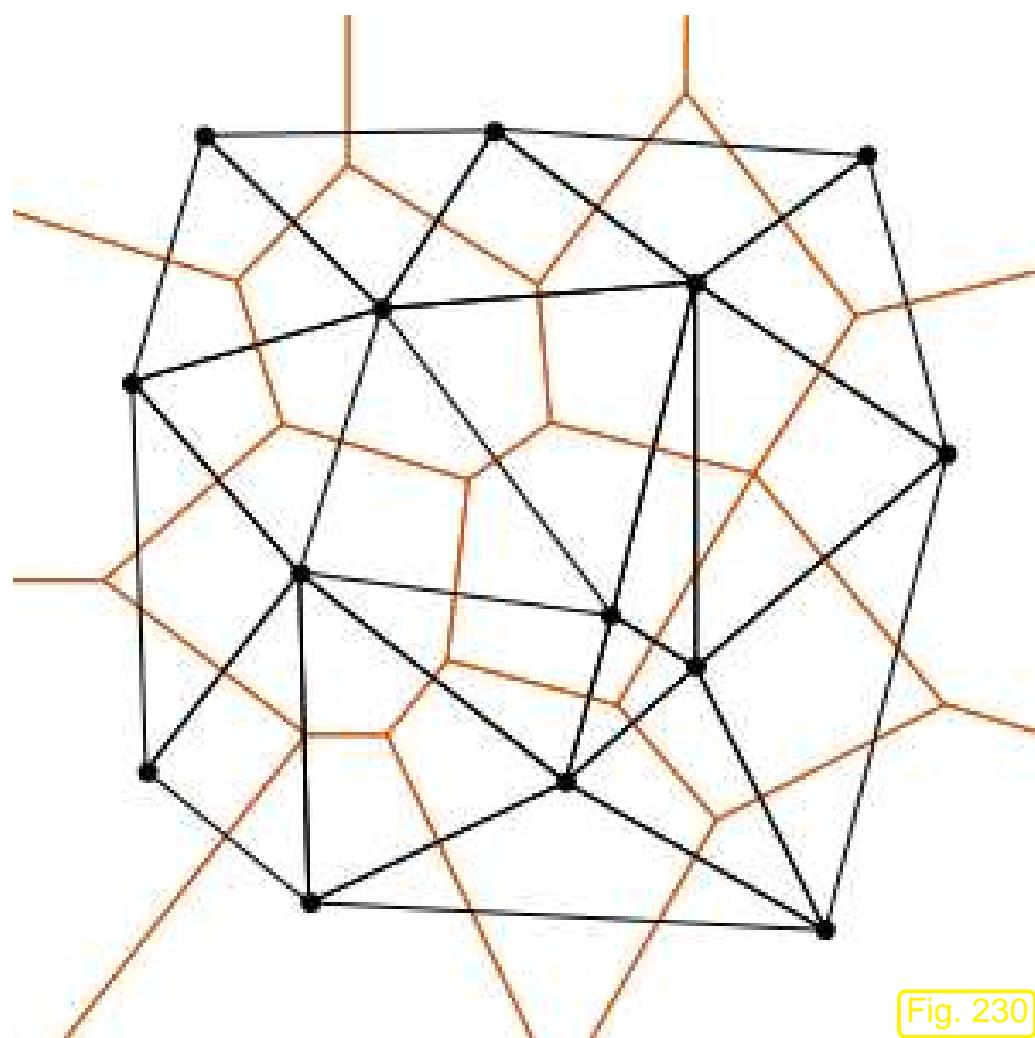


Fig. 230

Code 7.3.29: Demonstration of Delaunay-remeshing

```
1 function meshadv(v,n,tau,m)
2 % Point advection and remeshing for Lagrangian method
3 % v: handle to a function returning the velocity field for (an array) of
4 % points
5 % n: h = 1/n is the grid spacing of the initial point distribution
6 % Initialize points
7 h = 1/n; [Xp,Yp] = meshgrid(0:h:1,0:h:1);
```

```

8 P = [ reshape(Xp,1,(n+1)^2); reshape(Yp,1,(n+1)^2) ];
9 % Initialize points on the boundary
10 BP = [[[0:h:1);zeros(1,n+1)], [ones(1,n+1);(0:h:1)], ...
11 [(0:h:1);ones(1,n+1)], [zeros(1,n+1);(0:h:1)]]];
12
13 % Plot triangulation
14 fp = figure('name','evolving meshes','renderer','painters');
15 TRI = delaunay(P(1,:),P(2,:));
16 plot(P(1,:),P(2,:),'r+'); hold on;
17 triplot(TRI,P(1,:),P(2,:),'blue'); hold off;
17 title(sprintf('n = %i, t = %f, \\\tau = %f, %i
18 points',n,0,tau,size(P,2)));
18 drawnow; pause;
19
20 t = 0;
21 for l=1:m
22 % Advect points (explicit trapezoidal rule)
23 P1 = P + tau/2*v(P); P = P + tau*v(P1);
24
25 % Remove points on the boundary or outside the domain
26 Pnew = []; l = 1;
27 for p=P
28 if ((p(1) > eps) | (p(1) < 1-eps) | (p(2) > eps) | (p(2) <
    1-eps))

```

```

29 Pnew = [Pnew,p];
30
31 l = l+1;
32
33
34 P = [Pnew, BP]; % Add points on the boundary (particle injection)
35
36 % Plot triangulation
37 TRI = delaunay(P(1,:),P(2,:));
38 plot(P(1,:),P(2,:),'r+'); hold on;
39 triplot(TRI,P(1,:),P(2,:),'blue'); hold off;
40 title(sprintf('n = %i, t = %f, \\\tau = %f, %i
41 points',n,t,tau,size(P,2)));
42 drawnow;
43
44 t = t+tau;
45
46 end

```

$\Omega =]0, 1[^2$, velocity fields like in Ex. 7.3.25. Advection of interpolation nodes by means of explicit trapezoidal rule.

```
meshadv(@circvel,20,0.05,40);  
meshadv(@rotvel,20,0.05,40);
```



Example 7.3.30 (Lagrangian method for convection-diffusion in 1D).

Same IVP as in Ex. 7.3.4

- Linear finite element Galerkin discretization with mass lumping in space
- Strang splitting applied to diffusive and convective terms
- Implicit Euler timestepping for diffusive partial timestep

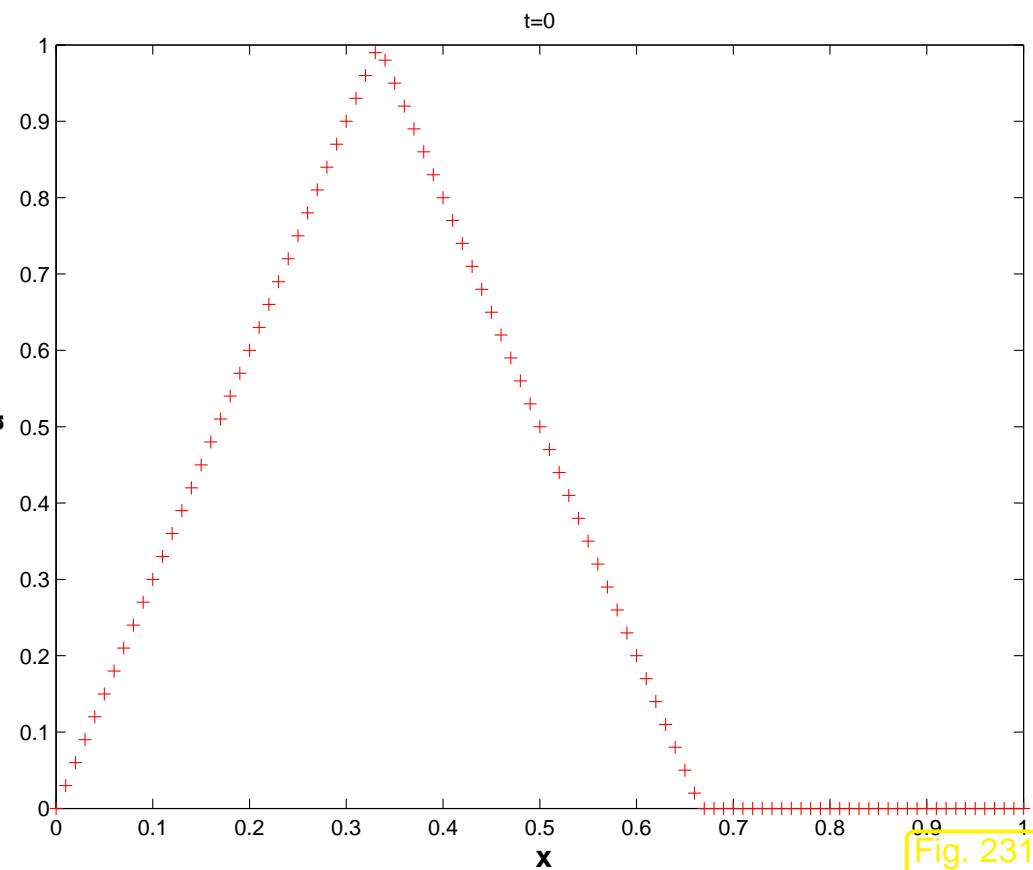


Fig. 231

Code 7.3.31: Lagrangian method for (7.3.5)

7.3

```
1 function lagr(epsilon,N,M)
```

p. 776

```

2 % This function implements a simple Lagrangian advection scheme for the 1D
3 % convection-diffusion
4 % IBVP  $-\epsilon \frac{d^2 u}{dx^2} + \frac{du}{dx} = 0$ ,  $u(x, 0) = \max(1 - 3|x - \frac{1}{3}|, 0)$ ,
5 % and homogeneous Dirichlet boundary conditions  $u(0) = u(1) = 0$ . Timestepping
6 % employs Strang splitting
7 % applied to diffusive and convective spatial operators.
8 % epsilon: strength of diffusion
9 % N: number of cells of spatial mesh
10 % M: number of timesteps
11
12
13 T = 0.5; tau = T/M; % timestep size
14 h = 1/N; x = 0:h:1; u = max(1-3*abs(x(2:end-1)-1/3),0)'; % Initial value
15
16 [Amat,Mmat] = getdeltamat(x); % Obtain stiffness and mass matrix
17 u = (Mmat+0.5*tau*epsilon*Amat)\(Mmat*u); % Implicit Euler timestep
18
19 for j=1:M+1
20 % Advection step: shift meshpoints, drop those travelling out of  $\Omega = ]0, 1[$ , insert
21 % new meshpoints from the left. Solution values are just copied.
22 xm = x(2:end-1)+tau; % Transport of meshpoints (here: explicit Euler)
23 idx = find(xm < 1); % Drop meshpoints beyond  $x = 1$ 
24 x = [0,tau,xm(idx),1]; % Insert new meshpoint at left end of  $\Omega$ 
25 u = [0;u(idx)]; % Copy nodal values and feed 0 from left

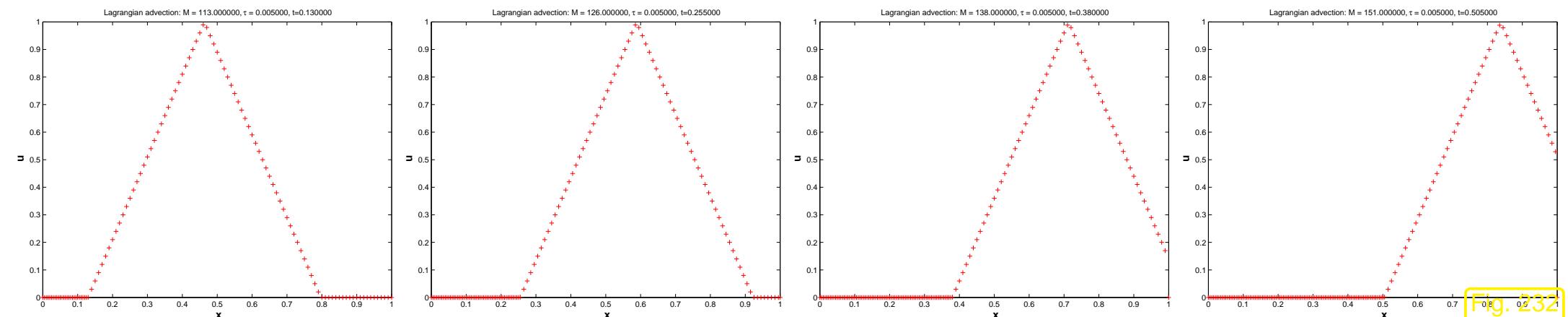
```

```

23
24 % Diffusion partial timestep
25 [Amat,Mmat] = getdeltamat(x); % Obtain stiffness and mass
26 % matrix on new mesh
27 u = (Mmat+tau*epsilon*Amat)\(Mmat*u); % Implicit Euler step
28 end
end

```

$$\epsilon = 10^{-5}$$



$$\epsilon = 0.1$$

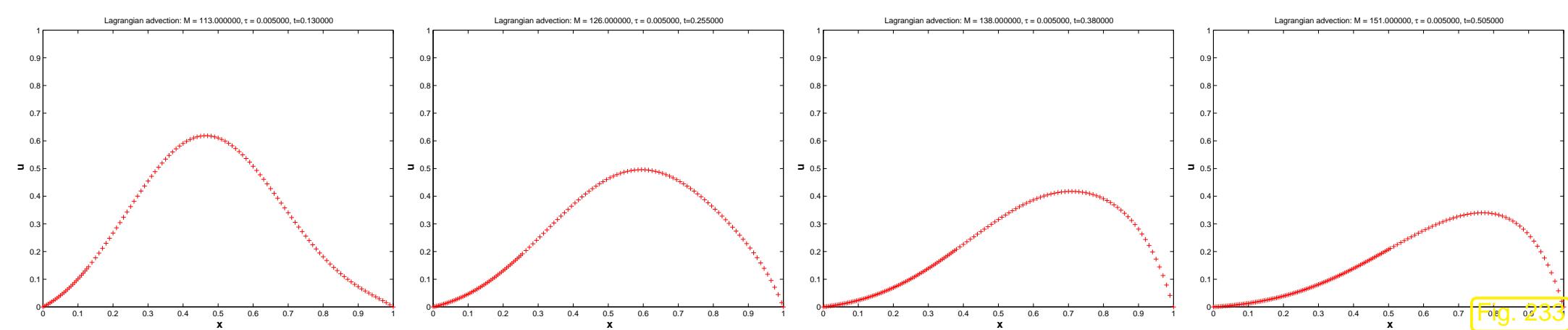


Fig. 7.33

“Reference solution” computed by method of lines, see Ex. 7.3.4, with $h = 10^{-3}$, $\tau = 5 \cdot 10^{-5}$:

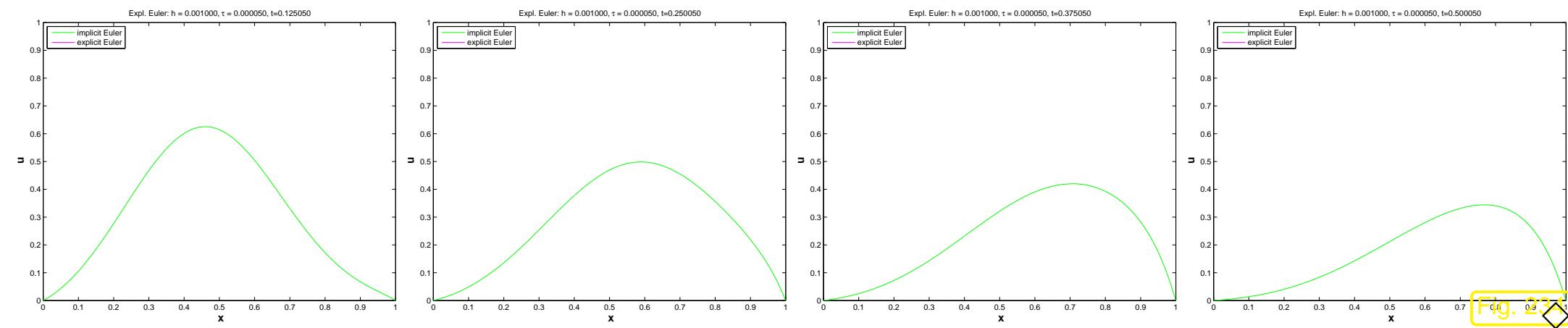
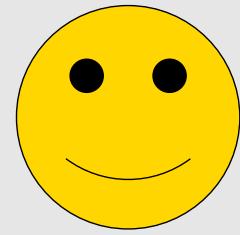


Fig. 7.34



Advantage of Lagrangian (particle) methods for convection diffusion:

No artificial diffusion required (no “smearing”)

No stability induced timestep constraint



Drawback of Lagrangian (particle) methods for convection diffusion:

Remeshing (may be) expensive and difficult.

Point advection may produce “voids” in point set.

7.3.4 Semi-Lagrangian method

Now we study a family of methods for transient convection-diffusion that takes into account transport along streamlines, but, in contrast to genuine Lagrangian methods, relies on a *fixed* mesh.

Definition 7.3.32 (Material derivative).

Given a velocity field $\mathbf{v} : \Omega \times]0, T[\mapsto \mathbb{R}^d$, the **material derivative** of a function $f = f(\mathbf{x}, t)$ at (\mathbf{x}, t) is

$$\frac{Df}{D\mathbf{v}}(\mathbf{x}, t) = \lim_{\delta \rightarrow 0} \frac{f(\mathbf{y}(\delta), t + \delta) - f(\mathbf{x}, t)}{\delta},$$

where $s \mapsto \mathbf{y}(s)$ solves the initial value problem

$$\frac{d\mathbf{y}}{ds}(s) = \mathbf{v}(\mathbf{y}(s), t + s) \quad , \quad \mathbf{y}(0) = \mathbf{x}.$$

By a straightforward application of the chain rule for smooth f

$$\boxed{\frac{Df}{D\mathbf{v}}(\mathbf{x}, t) = \mathbf{grad}_{\mathbf{x}} f(\mathbf{x}, t) \cdot \mathbf{v}(\mathbf{x}, t) + \frac{\partial f}{\partial t}(\mathbf{x}, t)}. \quad (7.3.33)$$

➤ The transient convection-diffusion equation can be rewritten as (7.3.1)

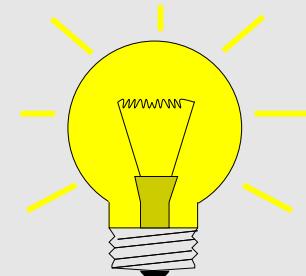
$$\frac{\partial u}{\partial t} - \epsilon \Delta u + \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{grad} u = f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[,$$



← (7.3.33)

$$\frac{Du}{D\mathbf{v}} - \epsilon \Delta u = f \quad \text{in } \tilde{\Omega} := \Omega \times]0, T[. \quad (7.3.34)$$

Idea: *Backward difference* (“implicit Euler”) discretization of material derivative



$$\frac{Du}{D\mathbf{v}}|_{(\mathbf{x},t)=(\bar{\mathbf{x}},\bar{t})} \approx \frac{u(\bar{\mathbf{x}},\bar{t}) - u(\mathbf{y}_{\bar{\mathbf{x}}}(-\tau),\bar{t}-\tau)}{\tau},$$

where $s \mapsto \mathbf{y}(s)$ solves the initial value problem $\frac{d\mathbf{y}_{\bar{\mathbf{x}}}}{ds}(s) = \mathbf{v}(\mathbf{y}_{\bar{\mathbf{x}}}(s), \bar{t}+s)$, $\mathbf{y}_{\bar{\mathbf{x}}}(0) = \bar{\mathbf{x}}$.

Combine this with *Galerkin discretization* (here discussed for linear finite elements, homogeneous Dirichlet boundary conditions $u = 0$ on $\partial\Omega$).

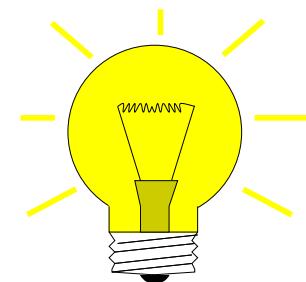
This yields one timestep for the **Semi-Lagrangian method**:

$$u_N^{(j)} \in \mathcal{S}_{1,0}^0(\mathcal{M}): \int_{\Omega} \frac{u_N^{(j)}(\mathbf{x}) - u_N^{(j-1)}(\mathbf{y}_{\mathbf{x}}(-\tau))}{\tau} v_N(\mathbf{x}) d\mathbf{x} + \epsilon \int_{\Omega} \operatorname{grad} u_N^{(j)} \cdot \operatorname{grad} v_N d\mathbf{x}$$

$$= \int_{\Omega} f(\boldsymbol{x}, t_j) v_N(\boldsymbol{x}) \, d\boldsymbol{x} \quad \forall v_N \in \mathcal{S}_{1,0}^0(\mathcal{M}) . \quad (7.3.35)$$

Here, \mathcal{M} is supposed to be a *fixed* triangular mesh of Ω .

However, (7.3.35) cannot be implemented: $\boldsymbol{x} \mapsto u_N^{(j-1)}(\mathbf{y}_{\boldsymbol{x}}(-\tau)) \notin \mathcal{S}_1^0(\mathcal{M})$! (Not even piecewise smooth on \mathcal{M} , which thwarts local quadrature.)



Idea:

- replace $u_N^{(j-1)}(\mathbf{y}_{\boldsymbol{x}}(-\tau))$ with linear interpolant
 $I_1 u_N^{(j-1)}(\mathbf{y}_{\boldsymbol{x}}(-\tau)) \in \mathcal{S}_{1,0}^0(\mathcal{M}),$
- approximate $\mathbf{y}_{\boldsymbol{x}}(-\tau)$ by $\boldsymbol{x} - \tau \mathbf{v}(\boldsymbol{x}, t_j)$ (explicit Euler).

► Implementable version of (7.3.35) (using mass lumping, see Rem. 6.2.34)

$$\begin{aligned} u_N^{(j)} \in \mathcal{S}_{1,0}^0(\mathcal{M}): \quad & \frac{1}{3} |U_{\boldsymbol{p}}| (\mu_{\boldsymbol{p}}^{(j)} - u_N^{(j-1)}(\boldsymbol{p} - \tau \mathbf{v}(\boldsymbol{p}, t_j)) + \tau \int_{\Omega} \mathbf{grad} u_N^{(j)} \cdot \mathbf{grad} b_N^{\boldsymbol{p}} \, d\boldsymbol{x} \\ & = \frac{1}{3} |U_{\boldsymbol{p}}| f(\boldsymbol{p}), \quad \boldsymbol{p} \in \mathcal{N}(\mathcal{M}) \cap \Omega , \quad (7.3.36) \end{aligned}$$

where $\mu_{\mathbf{p}}^{(j)}$ are the nodal values of $u_N^{(j)} \in \mathcal{S}_{1,0}^0(\mathcal{M})$ associated with the interior nodes of the mesh \mathcal{M} , $b_N^{\mathbf{p}}$ is the “tent function” belonging to node \mathbf{p} , $|U_{\mathbf{p}}|$ is the sum of the areas of all triangles adjacent to \mathbf{p} .

Example 7.3.37 (Semi-Lagrangian method for convection-diffusion in 1D).

Same IVP as in Ex. 7.3.30

- Linear finite element Galerkin discretization with mass lumping in space
- Semi-Lagrangian method: 1D version of (7.3.35)
- Explicit Euler streamline backtracking

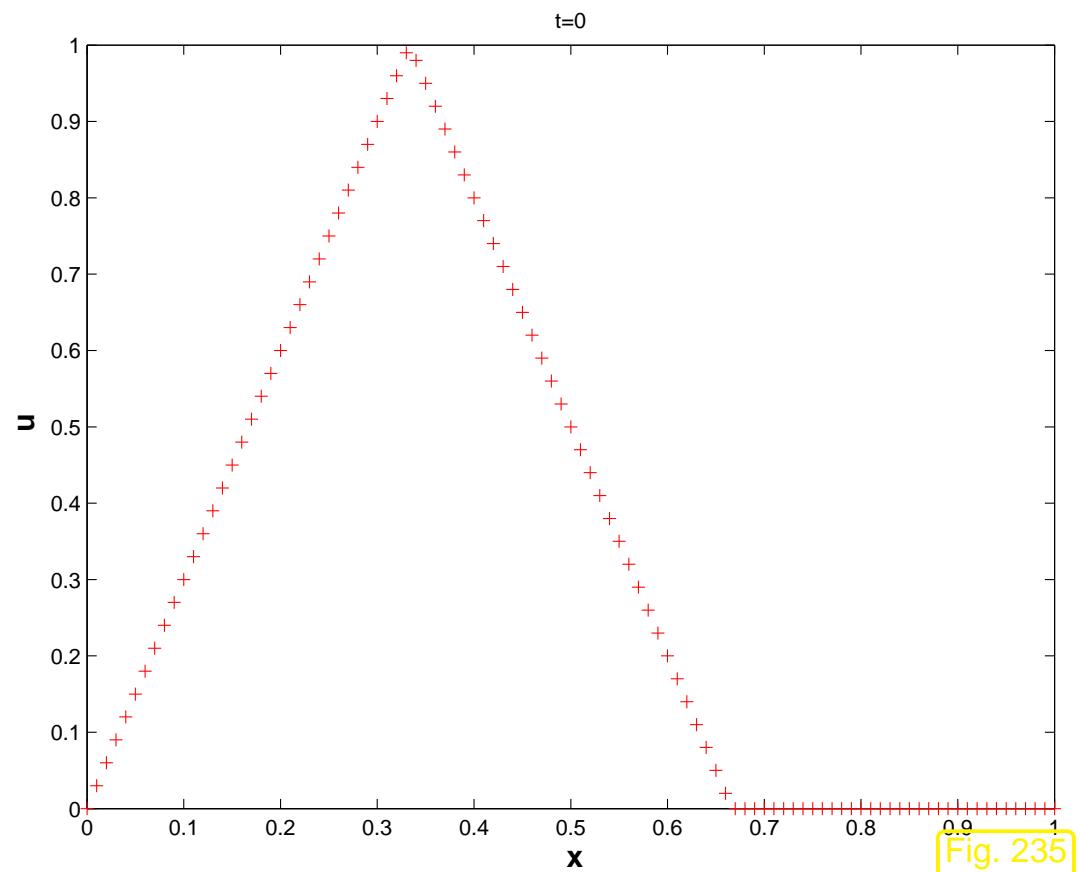


Fig. 235

$$\epsilon = 10^{-5}$$

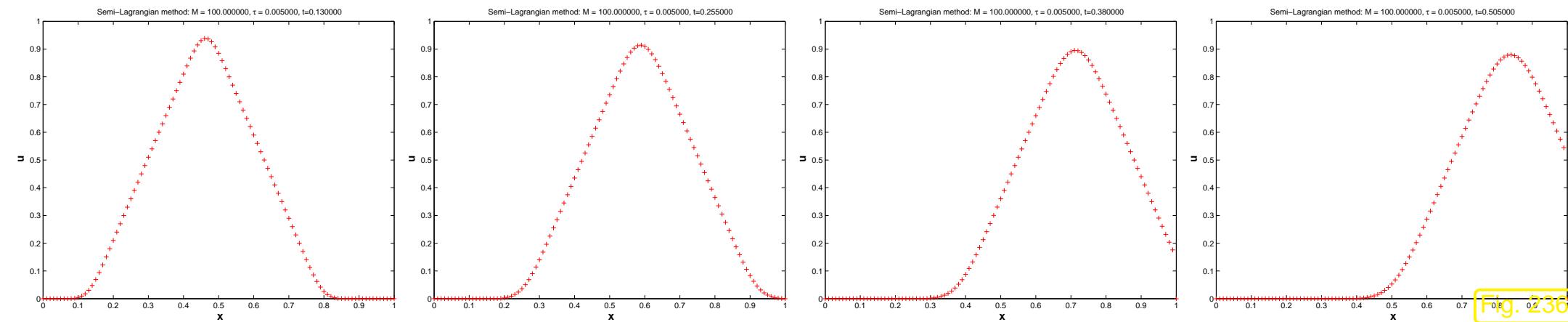


Fig. 236

$$\epsilon = 0.1:$$

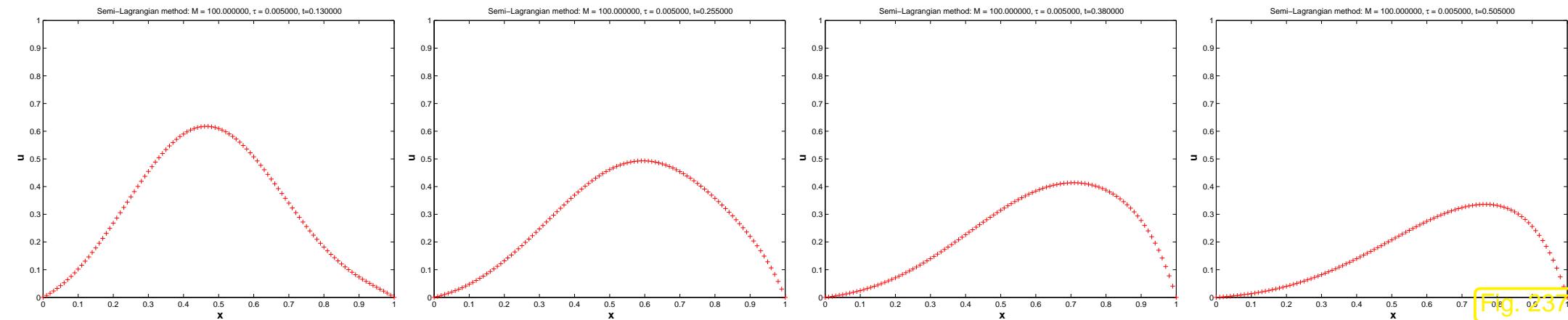
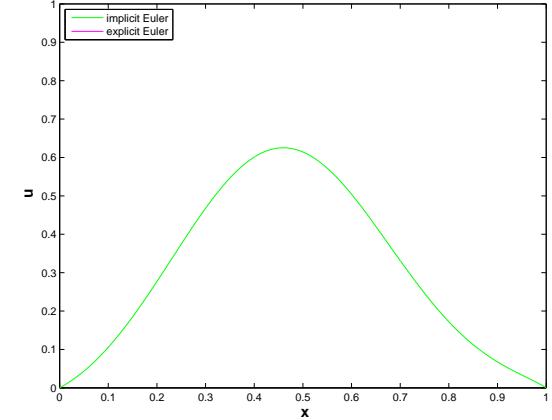


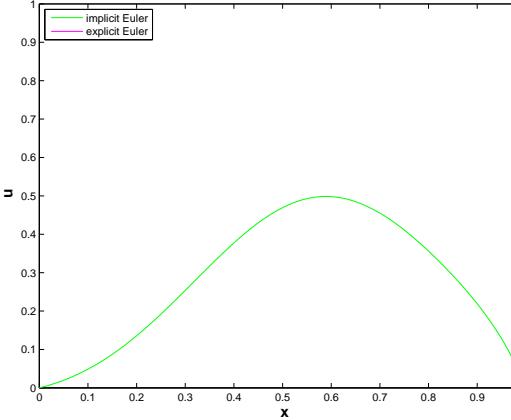
Fig. 237

“Reference solution” computed by method of lines, see Ex. 7.3.4, with $h = 10^{-3}$, $\tau = 5 \cdot 10^{-5}$:

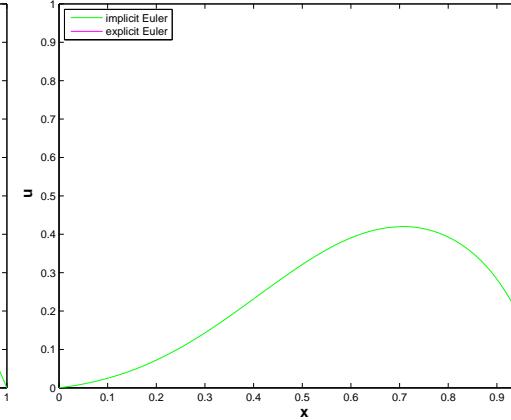
Expl. Euler: $h = 0.001000$, $\tau = 0.000050$, $t=0.125050$



Expl. Euler: $h = 0.001000$, $\tau = 0.000050$, $t=0.250050$



Expl. Euler: $h = 0.001000$, $\tau = 0.000050$, $t=0.375050$



Expl. Euler: $h = 0.001000$, $\tau = 0.000050$, $t=0.500050$

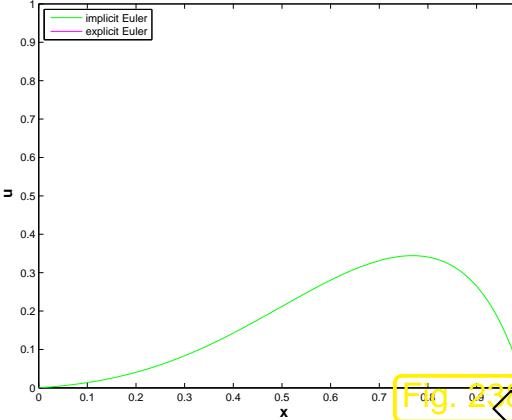


Fig. 7.37

Numerical Methods for Conservation Laws

Conservation laws describe physical phenomena governed by

- *conservation* laws for certain physical quantities (e.g., mass momentum, energy, etc.),
- *transport* of conserved physical quantities.

We have already examined problems of this type in connection with transient heat conduction in Sect. 7.1.4. There thermal energy was the conserved quantity and a *prescribed* external velocity field \mathbf{v} determined the transport.

A new aspect emerging for general conservation laws is that the transport velocity itself may depend on the conserved quantities themselves, which gives rise to *non-linear models*.

8.1 Conservation laws: Examples

Focus:

Cauchy problems

Spatial domain $\Omega = \mathbb{R}^d$ (unbounded!)

- Cauchy problems are pure initial value problems (no boundary values).

Rationale: ① *Finite speed of propagation* typical of conservation laws

(Potential spatial boundaries will not affect the solution for some time in the case of compactly supported initial data, *cf.* situation for wave equation, where we also examined the Cauchy problem, see (6.2.15).)

② No spatial boundary ➤ need not worry about (spatial) boundary conditions!

(Issue of spatial boundary conditions can be very intricate for conservation laws)

8.1.1 Linear advection

Cauchy problem for linear transport equation (advection equation) → Sect. 7.1.4, (7.1.15):

$$\frac{\partial}{\partial t}(\rho u) + \operatorname{div}(\mathbf{v}(\mathbf{x}, t)(\rho u)) = f(\mathbf{x}, t) \quad \text{in } \widetilde{\Omega} := \mathbb{R}^d \times]0, T[, \quad (8.1.1)$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathbb{R}^d \quad (\text{initial conditions}) . \quad (8.1.2)$$

$u = u(\mathbf{x}, t)$ ≈ temperature, $\rho > 0$ ≈ heat capacity, $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ ≈ prescribed velocity field.

(8.1.1) = linear scalar conservation law



Conserved quantity: thermal energy (density) ρu

(Recall the derivation of (7.1.15) through conservation of energy, cf. (6.1.1).)

Simplified problem: assume constant heat capacity $\rho \equiv 1$, no sources $f \equiv 0$, stationary velocity field $\mathbf{v} = \mathbf{v}(\mathbf{x}) \Rightarrow$ rescaled initial value problem *written in conserved variables*

$$\begin{aligned} \frac{\partial u}{\partial t} + \operatorname{div}(\mathbf{v}(\mathbf{x})u) &= 0 \quad \text{in } \tilde{\Omega} := \mathbb{R}^d \times]0, T[, \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathbb{R}^d \quad (\text{initial conditions}) . \end{aligned} \tag{8.1.3}$$

Convention: differential operator div acts on spatial independent variable only,

$$(\operatorname{div} \mathbf{f})(\mathbf{x}, t) := \frac{\partial f_1}{\partial x_1} + \cdots + \frac{\partial f_1}{\partial x_d}, \quad \mathbf{f}(\mathbf{x}, t) = \begin{pmatrix} f_1(\mathbf{x}, t) \\ \vdots \\ f_d(\mathbf{x}, t) \end{pmatrix} .$$

Special case:

Constant coefficient linear advection in 1D

- $d = 1 \Rightarrow \Omega = \mathbb{R}$,
- constant velocity $v = \text{const.}$.

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(vu) = 0 \quad \text{in } \tilde{\Omega} = \mathbb{R} \times]0, T[, \quad u(x, 0) = u_0(x) \quad \forall x \in \mathbb{R} . \tag{8.1.4}$$

This is the 1D version of the transport equation (7.3.7) ➤ solution given by (7.3.12)

(7.3.12)

$$\blacktriangleright \quad u(x, t) = u_0(x - vt) , \quad x \in \mathbb{R} , \quad 0 \leq t < T . \quad (8.1.5)$$

Solution $u = u(x, t)$ = initial data “travelling” with velocity v .

Solution formula (8.1.5) makes perfect sense even for *discontinuous* initial data u_0 !

- We should not expect $u = u(x, t)$ to be differentiable in space or time.
A “weaker” concept of solution is required, see Sect. 8.2.3 below.

This consideration should be familiar: for second order elliptic boundary value problems, for which classical solutions are to be twice continuously differentiable, the concept of a variational solution

made it possible to give a meaning to solutions $\in H^1(\Omega)$ that are merely continuous and piecewise differentiable, see Rem. 1.3.23.

Remark 8.1.6 (Boundary conditions for linear advection).

Recall the discussion in Sects. 7.2.1, 7.3.2, cf. solution formula (7.3.13):

For the scalar linear advection initial boundary value problem

$$\frac{\partial u}{\partial t} + \operatorname{div}(\mathbf{v}(\mathbf{x}, t)u) = f(\mathbf{x}, t) \quad \text{in} \quad \tilde{\Omega} := \Omega \times]0, T[, \quad (8.1.7)$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \text{for all} \quad \mathbf{x} \in \Omega , \quad (8.1.8)$$

on a bounded domain $\Omega \subset \mathbb{R}^d$, **boundary conditions** (e.g., prescribed temperature)

$$u(\mathbf{x}, t) = g(\mathbf{x}, t) \quad \text{on} \quad \Gamma_{\text{in}}(t) \times]0, T[,$$

can be imposed on the **inflow boundary**

$$\Gamma_{\text{in}}(t) := \{\mathbf{x} \in \partial\Omega: \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) < 0\} , \quad 0 < t < T . \quad (8.1.9)$$

Note: Γ_{in} can change with time!

Bottom line:

Knowledge of local and current direction of transport
needed to impose meaningful boundary conditions!

8.1.2 Inviscid gas flow

Frictionless gas flow in (infinitely) long pipe



Gas

Fig. 239^c

Terminology: frictionless $\hat{=}$ inviscid

Assumption: variation of gas density negligible (“near incompressibility”) motion of fluid driven by inertia \leftrightarrow *conservation of linear momentum*

We derive a **continuum model** for inviscid, nearly incompressible fluid in a straight infinitely long pipe
 $\leftrightarrow \Omega = \mathbb{R}$ (Cauchy problem).

This simple model will be based on *conservation of linear momentum*, whereas conservation of mass and energy will be neglected (and violated). Hence, the crucial conserved quantity will be the momentum.

Unknown: $u = u(x, t)$ = momentum *density* \leftrightarrow momentum density \sim local velocity $v = v(x, t)$ of fluid

Conserved quantity: (linear) **momentum** of fluid $u = u(x, t)$

\gg flux of linear momentum $f \sim v \cdot u$ (after scaling: $f(u) = \frac{1}{2}u \cdot u$)
 (“momentum u advected by velocity u ”)

Conservation of linear momentum ($\sim u$): for all control volumes $V :=]x_0, x_1[\subset \Omega$:

$$\underbrace{\int_{x_0}^{x_1} u(x, t_1) - u(x, t_0) dx}_{\text{change of momentum in } V} + \underbrace{\int_{t_0}^{t_1} \frac{1}{2}u^2(x_1, t) - \frac{1}{2}u^2(x_0, t) dt}_{\text{outflow of momentum}} = 0 \quad \forall 0 < t_0 < t_1 < T . \quad (8.1.10)$$

Temporarily assume that $u = u(x, t)$ is smooth in both x and t and set $x_1 = x_0 + h$, $t_1 = t_0 + \tau$. First approximate the integrals in (8.1.10).

$$\int_{x_0}^{x_1} u(x, t_1) - u(x, t_0) dx = h(u(x_0, t_1) - u(x_0, t_0)) + O(h^2) \quad \text{for } h \rightarrow 0 ,$$

$$\int_{t_0}^{t_1} \frac{1}{2}u^2(x_1, t) - \frac{1}{2}u^2(x_0, t) dt = \tau(\frac{1}{2}u^2(x_1, t_0) - \frac{1}{2}u^2(x_0, t_0)) + O(\tau^2) \quad \text{for } \tau \rightarrow 0 .$$

Then employ Taylor expansion for the differences:

$$u(x_0, t_1) - u(x_0, t_0) = \frac{\partial u}{\partial t}(x_0, t_0)\tau + O(\tau^2) \quad \text{for } \tau \rightarrow 0 ,$$
$$\frac{1}{2}u^2(x_1, t_0) - \frac{1}{2}u^2(x_0, t_0) = \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right)(x_0, t_0)h + O(h^2) \quad \text{for } h \rightarrow 0 .$$

Finally, divide by h and τ and take the limit $\tau \rightarrow 0, h \rightarrow 0$:


$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0 \quad \text{in } \Omega \times]0, T[. \quad (8.1.11)$$

(8.1.11) = **Burgers equation**: a one-dimensional scalar conservation law (without sources)

Remark 8.1.12 (Euler equations).

The above gas model blatantly ignores the fundamental laws of conservation of mass and of energy. These are taken into account in a famous more elaborate model of inviscid fluid flow:

Euler equations [5], a more refined model for inviscid gas flow in an infinite pipe

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (E + p)u \end{pmatrix} = 0 \quad \text{in } \mathbb{R} \times]0, T[, \quad (8.1.13)$$

$$u(x, 0) = u_0(x) , \quad \rho(x, 0) = \rho_0(x) , \quad E(x, 0) = E_0(x) \quad \text{for } x \in \mathbb{R} ,$$

where

- $\rho = \rho(x, t) \hat{=} \text{fluid density}$, $[\rho] = \text{kg m}^{-1}$,
- $u = u(x, t) \hat{=} \text{fluid velocity}$, $[u] = \text{m s}^{-1}$,
- $p = p(x, t) \hat{=} \text{fluid pressure}$, $[p] = \text{N}$,
- $E = E(x, t) \hat{=} \text{total energy density}$, $[E] = \text{J m}^{-1}$.

+ state equation (material specific constitutive equations), e.g., for ideal gas

$$p = (\gamma - 1)(E - \frac{1}{2}\rho u^2) , \quad \text{with adiabatic index } 0 < \gamma < 1 .$$

Conserved quantities (densities):

$$\rho \leftrightarrow \text{mass density} , \quad \rho u \leftrightarrow \text{momentum density} , \quad E \leftrightarrow \text{energy density}.$$

Underlying physical conservation principles for individual densities:

8.1

p. 797

- First equation $\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0 \leftrightarrow \text{conservation of mass},$
- Second equation $\frac{\partial(\rho u)}{\partial t} + \frac{\partial}{\partial x}(\rho u^2 + p) = 0 \leftrightarrow \text{conservation of momentum},$
- Third equation $\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}((E + p)u) = 0 \leftrightarrow \text{conservation of energy}.$

Euler equations (8.1.13) = non-linear system of conservation laws (in 1D)

As is typical of non-linear systems of conservation laws, the analysis of the Euler equations is intrinsically difficult: hitherto not even existence and uniqueness of solutions for general initial values could be established. Moreover, solutions display a wealth of complicated structures. Therefore, this course is confined to scalar conservation laws, for which there is only one unknown real-valued function of space and time.



8.2 Scalar conservation laws in 1D

8.2.1 Integral and differential form

What we have seen so far (except for Euler's equations in Rem. 8.1.12)

Burgers equation: $\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{1}{2} u^2 \right) = 0 \quad \text{in } \Omega \times]0, T[, \quad (8.1.11)$

linear advection: $\frac{\partial}{\partial t}(\rho u) + \operatorname{div}(\mathbf{v}(\mathbf{x}, t)(\rho u)) = f(\mathbf{x}, t) \quad \text{in } \mathbb{R}^d \times]0, T[. \quad (8.1.1)$

Now, we learn about a class of Cauchy problems to which these two belong. First some notations and terminology:

- $\Omega \subset \mathbb{R}^d \hat{=} \text{fixed (bounded/unbounded) spatial domain}$ ($\Omega = \mathbb{R}^d = \text{Cauchy problem}$)
- computational domain: space-time cylinder $\tilde{\Omega} := \Omega \times]0, T[$, $T > 0$ final time

- $U \subset \mathbb{R}^m$ ($m \in \mathbb{N}$) $\hat{=}$ phase space (state space) for conserved quantities u_i (usually $U = \mathbb{R}^m$)

Our focus below:

scalar case $m = 1$

Conservation law for transient state distribution $u : \tilde{\Omega} \mapsto U$: $u = u(\mathbf{x}, t)$, for $0 \leq t \leq T$

$$\frac{d}{dt} \int_V u \, d\mathbf{x} + \int_{\partial V} \mathbf{f}(u, \mathbf{x}) \cdot \mathbf{n} \, dS(\mathbf{x}) = \int_V s(u, \mathbf{x}, t) \, d\mathbf{x} \quad \forall \text{ "control volumes" } V \subset \Omega . \quad (8.2.1)$$

Terminology:

- ▷ flux function $\mathbf{f} : U \times \Omega \mapsto \mathbb{R}^d$
- ▷ source function $s : U \times \Omega \times]0, T[\mapsto \mathbb{R}$ (here usually $s = 0$)

- For Burgers equation (8.1.11): $f(u, x) = \frac{1}{2}u^2$, $s = 0$,
- For linear advection (8.1.1): $\mathbf{f}(u, \mathbf{x}) = \mathbf{v}(\mathbf{x}, t)u$, $s = f(\mathbf{x}, t)$
(Note: in this case the conserved quantity is actually ρu , which was again denoted by u)

☞ (8.2.1) has the same structure as the “conservation of energy law” (6.1.1) for heat conduction.

Conservation of energy:

$$\frac{d}{dt} \int_V \rho u \, d\mathbf{x} + \int_{\partial V} \mathbf{j} \cdot \mathbf{n} \, dS = \int_V f \, d\mathbf{x} \quad \text{for all “control volumes” } V \quad (6.1.1)$$

energy stored in V

power flux through ∂V

heat generation in V

In this case the heat flux was given by

$$\text{Fourier's law} \quad \mathbf{j}(\mathbf{x}) = -\kappa(\mathbf{x}) \operatorname{grad} u(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (2.5.3)$$

or its extended version (7.1.5). In Fourier's law the flux is a *linear* function of *derivatives* of u .

Conversely, for the **flux function** $\mathbf{f} : U \times \Omega \mapsto \mathbb{R}^d$ in (8.2.1) we assume

f only depends on local state u , not on derivatives of u !

On the other hand we go far beyond Fourier's law, since

f will in general be a **non-linear** function of **u**!

Remark 8.2.2 (Diffusive flux).

Taking into account the relationship with heat “diffusion”, a flux function of the form of Fourier’s law
(2.5.3)

$$\mathbf{f}(u) = -\kappa(\mathbf{x}) \operatorname{grad} u ,$$

is called a **diffusive flux**.



Now, integrate (8.2.1) over time period $[t_0, t_1] \subset [0, T]$ and use fundamental theorem of calculus:

- Space-time integral form of (8.2.1), cf. (8.1.10),

$$\int_V u(\mathbf{x}, t_1) d\mathbf{x} - \int_V u(\mathbf{x}, t_0) d\mathbf{x} + \int_{t_0}^{t_1} \int_{\partial V} \mathbf{f}(u, \mathbf{x}) \cdot \mathbf{n} dS(\mathbf{x}) dt = \int_{t_0}^{t_1} \int_V s(u, \mathbf{x}, t) d\mathbf{x} dt \quad (8.2.3)$$

for all $V \subset \Omega$, $0 < t_0 < t_1 < T$, $\mathbf{n} \hat{=} \text{exterior unit normal at } \partial V$

- [Gauss theorem Thm. 2.4.5] (local) differential form of (8.2.1):

$$\boxed{\frac{\partial}{\partial t} u + \operatorname{div}_{\mathbf{x}} \mathbf{f}(u, \mathbf{x}) = s(u, \mathbf{x}, t) \quad \text{in } \tilde{\Omega}} \quad (8.2.4)$$

div acting on spatial variable \mathbf{x} only

- + initial condition

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega$$

Special case $d = 1 \leftrightarrow$ (8.2.4) = one-dimensional scalar conservation law for “density” $u : \tilde{\Omega} \mapsto \mathbb{R}$

$$\frac{\partial u}{\partial t}(x, t) + \frac{\partial}{\partial x}(f(u(x, t), x)) = s(u(x, t), x, t) \quad \text{in }]\alpha, \beta[\times]0, T[, \alpha, \beta \in \mathbb{R} \cup \{\pm\infty\}. \quad (8.2.5)$$

Remark 8.2.6 (Boundary values for conservation laws).

Suitable boundary values on $\partial\Omega \times]0, T[$? → usually tricky question (highly f -dependent)

Reason: remember discussion in Rem. 8.1.6, meaningful boundary conditions hinge on knowledge of local (in space and time) transport direction, which, in a *non-linear* conservation law, will usually depend on the unknown solution $u = u(x, t)$.



8.2.2 Characteristics

We consider Cauchy problem ($\Omega = \mathbb{R}$) for one-dimensional scalar conservation law (8.2.5):

$$\begin{aligned} & \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \text{in } \mathbb{R} \times]0, T[, \\ & u(x, 0) = u_0(x) \quad \text{in } \mathbb{R} . \end{aligned} \tag{8.2.7}$$

Assumption: flux function $f : \mathbb{R} \mapsto \mathbb{R}$ smooth ($f \in C^2$) and convex [19, Def. 5.5.2]

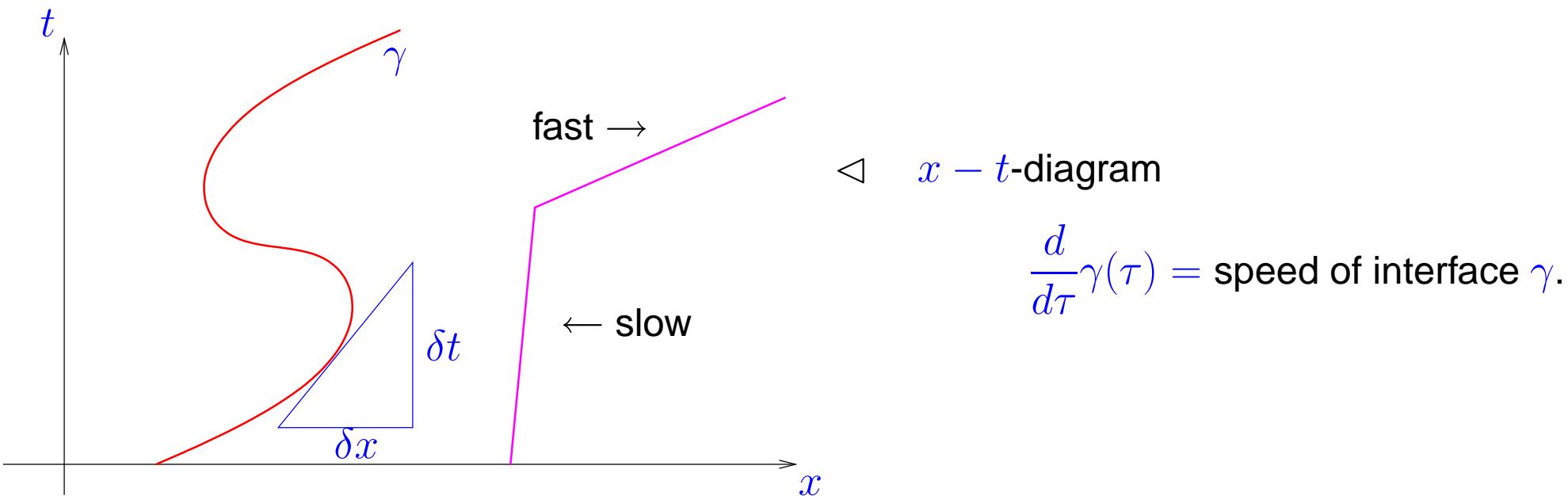
Recall [19, Thm. 5.5.2]: f convex \Rightarrow derivative f' increasing

Definition 8.2.8 (Characteristic curve for one-dimensional scalar conservation law).

A curve $\Gamma := (\gamma(\tau), \tau) : [0, T] \mapsto \mathbb{R} \times]0, T[$ in the (x, t) -plane is a **characteristic curve** of (8.2.7), if

$$\frac{d}{d\tau} \gamma(\tau) = f'(u(\gamma(\tau), \tau)) , \quad 0 \leq \tau \leq T , \quad (8.2.9)$$

where u is a continuously differentiable solution of (8.2.7).



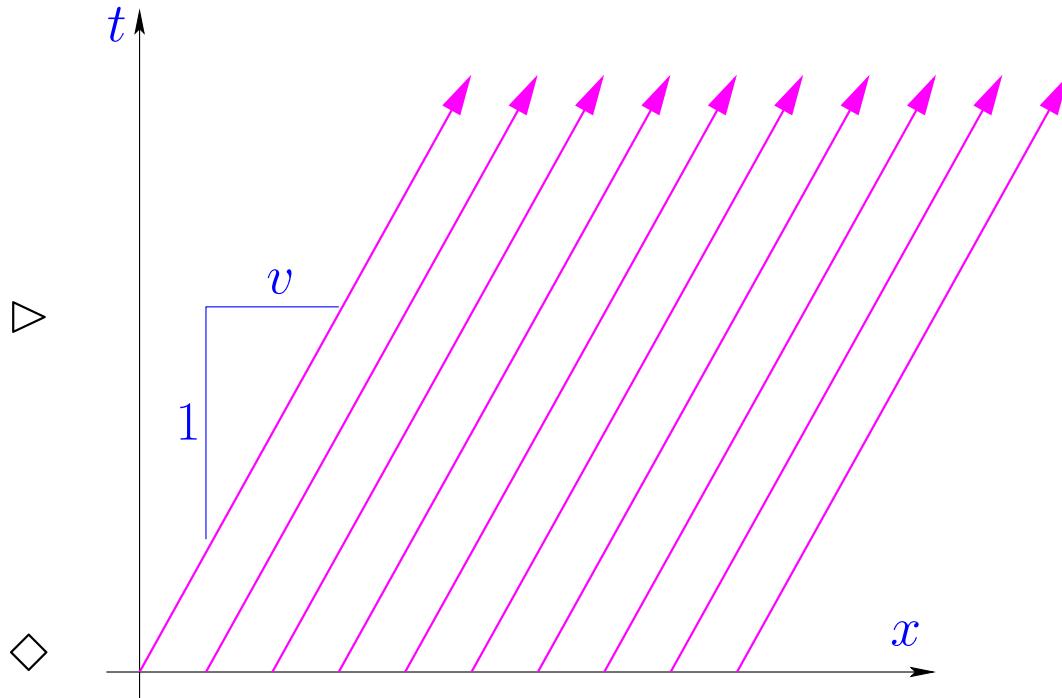
Example 8.2.10 (Characteristics for advection).

Constant linear advection (8.1.4): $f(u) = vu$

→ characteristics $\gamma(\tau) = v\tau + c, c \in \mathbb{R}$.

solution (8.1.5) $u(x, t) = u_0(x - vt)$

meaningful for any u_0 ! (cf. Sect. 7.3.2)



This example reveals a close relationship between streamlines (→ Sect. 7.1.1) and characteristic curves. That the latter are a true generalization is also reflected by the following simple observation, which generalizes the considerations in Sect. 7.3.2, (7.3.9).

Lemma 8.2.11 (Classical solutions and characteristic curves).

Smooth solutions of (8.2.7) are constant along characteristic curves.

Proof. Apply chain rule twice, cf. (7.3.9), and use the defining equation (8.2.9) for a characteristic curve:

$$\begin{aligned}
 \frac{d}{d\tau} u(\gamma(\tau), \tau) &\stackrel{\text{chain rule}}{=} \frac{\partial u}{\partial x}(\gamma(\tau), \tau) \frac{d}{d\tau} \gamma(\tau) + \frac{\partial u}{\partial t}(\gamma(\tau), \tau) \\
 &\stackrel{(8.2.9)}{=} \frac{\partial u}{\partial x}(\gamma(\tau), \tau) \cdot f'(u(\gamma(\tau), \tau)) + \frac{\partial u}{\partial t}(\gamma(\tau), \tau) \\
 &\stackrel{\text{chain rule}}{=} \left(\frac{\partial}{\partial x} f(u) \right)(\gamma(\tau), \tau) + \frac{\partial u}{\partial t}(\gamma(\tau), \tau) = 0 .
 \end{aligned}$$

☞ notation: $f' \hat{=} \text{derivative of flux function } f : U \subset \mathbb{R} \mapsto \mathbb{R}$

So, u is constant on a characteristic curve.

➤ $f'(u)$ is constant on a characteristic curve.

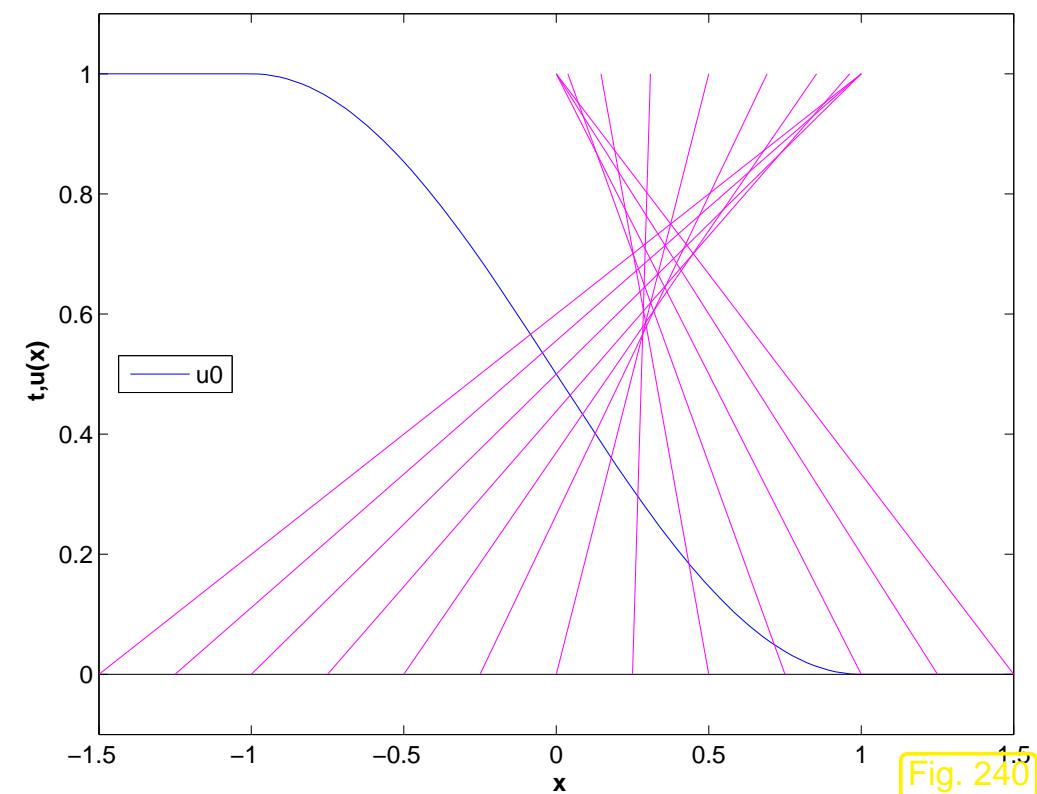
(8.2.9) \Rightarrow slope of characteristic curve is constant!

► Characteristic curve through $(x_0, 0)$ = straight line $(x_0 + f'(u_0(x_0))\tau, \tau)$, $0 \leq \tau \leq T$!

!? implicit solution formula for (8.2.7) (f' monotone !):

$$u(x, t) = u_0(x - f'(u(x, t))t) . \quad (8.2.12)$$

Example 8.2.13 (Breakdown of characteristic solution formula).



for Burger's equation (8.1.11):

($f(u) = \frac{1}{2}u^2$ smooth and strictly convex)

▷ $f'(u) = u$ (increasing)

◁ if u_0 smooth and decreasing

► characteristic curves intersect !

► solution formula (8.2.12) becomes invalid

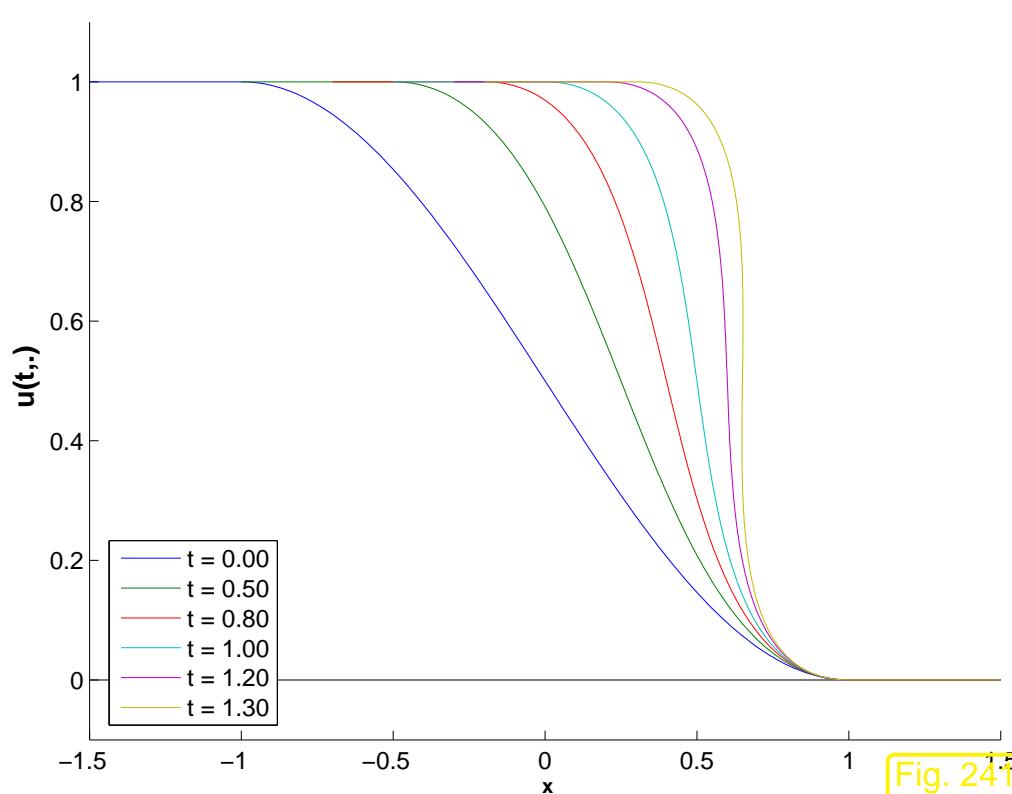


Fig. 241

$t < 1.3$: solution by (8.2.12)

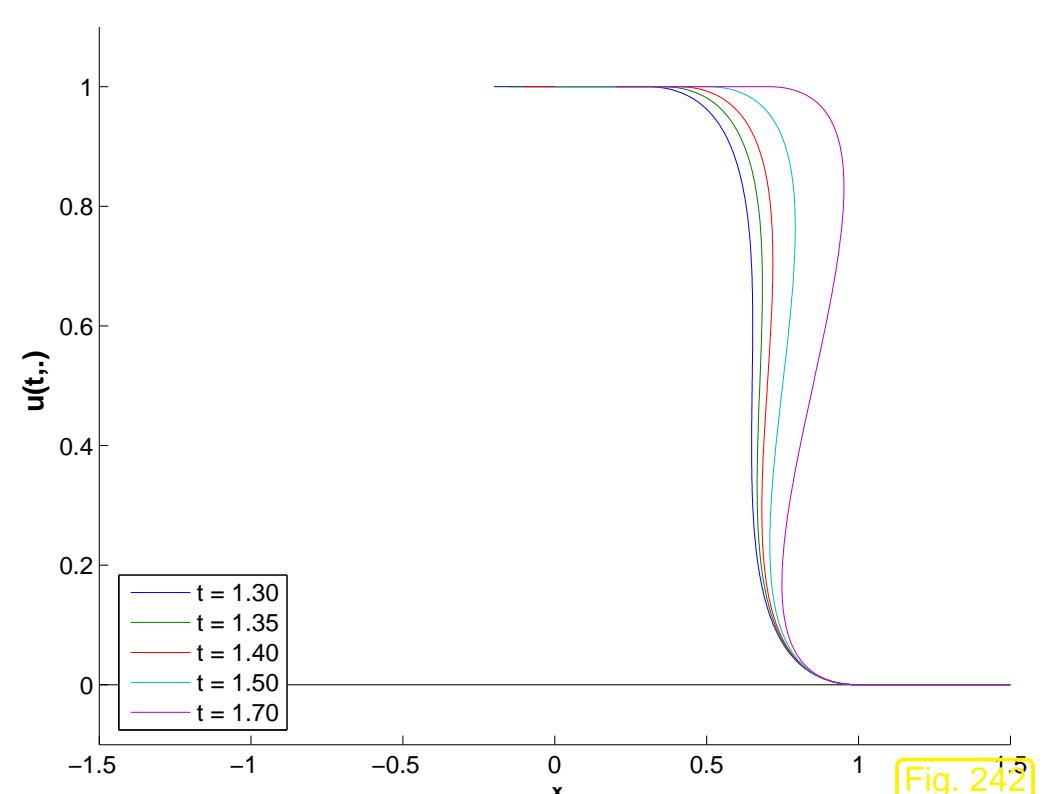


Fig. 242

the wave breaks: “multivalued solution”



breakdown of classical solutions & Ex. 8.2.10 → new concept of solution of (8.2.7)

8.2.3 Weak solutions

“Space-time Gaussian theorem”

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad (8.2.14)$$

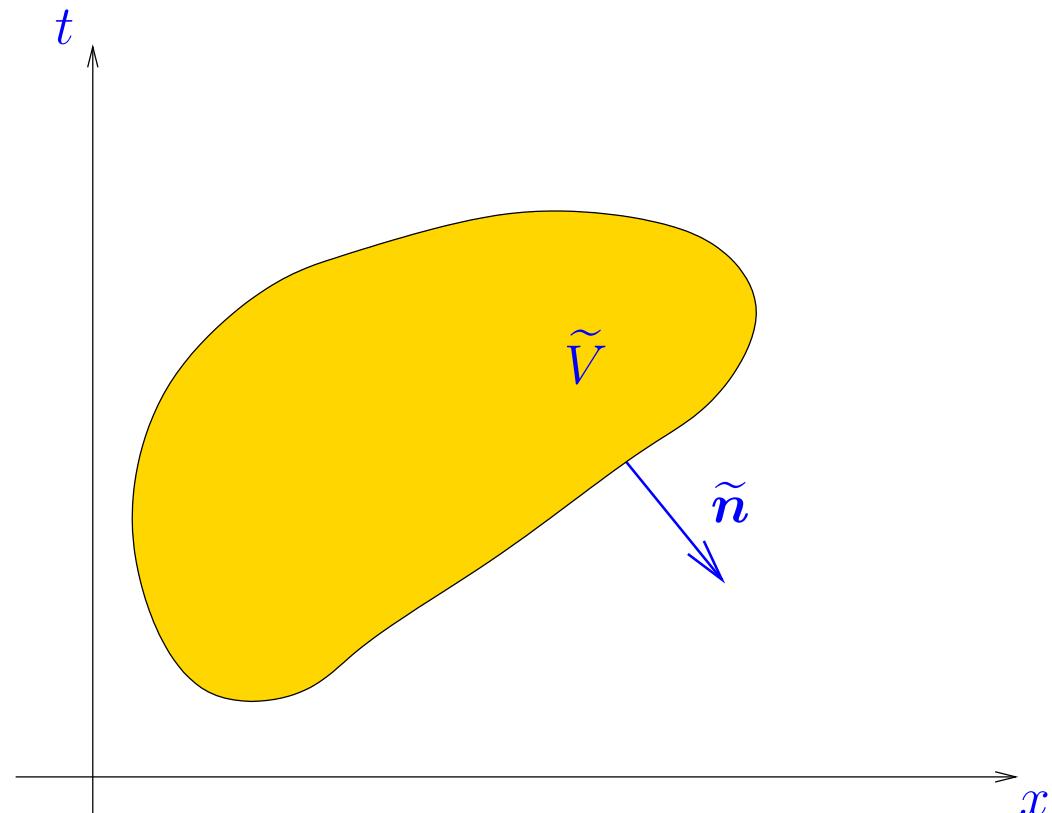


$$\operatorname{div}_{(x,t)} \begin{pmatrix} f(u) \\ u \end{pmatrix} = 0 \quad \text{in } \tilde{\Omega} . \quad (8.2.15)$$

► ∀ “space-time control volumes” $\tilde{V} \subset \tilde{\Omega}$:

$$\int_{\partial \tilde{V}} \begin{pmatrix} f(u(\tilde{x})) \\ u(\tilde{x}) \end{pmatrix} \cdot \begin{pmatrix} n_x(\tilde{x}) \\ n_t(\tilde{x}) \end{pmatrix} dS(\tilde{x}) = 0 ,$$

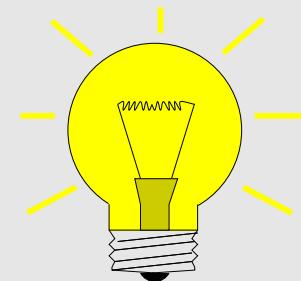
$\tilde{n} := (n_x, n_t)^T \doteq$ space-time unit normal



(8.2.15) for space-time rectangle $\tilde{V} =]x_0, x_1[\times]t_0, t_1[$ ► integral form of (8.2.14), cf. (8.2.3):

$$\int_{x_0}^{x_1} u(x, t_1) dx - \int_{x_0}^{x_1} u(x, t_0) dx = \int_{t_0}^{t_1} f(u(x_0, t)) dt - \int_{t_0}^{t_1} f(u(x_1, t)) dt . \quad (8.2.16)$$

Still, (8.2.16) encounters problems, if a discontinuity of u coincides with an edge of the space-time rectangle.



Idea: Obtain weak form of (8.2.14) from (8.2.15) by integration by parts, that is, application of Green's first formula Thm. 2.4.7 in space-time!

STEP I: Test (8.2.15) with **compactly supported smooth** function $\Phi : \tilde{\Omega} \mapsto \mathbb{R}$, $\Phi(\cdot, T) = 0$, and integrate over space-time cylinder $\tilde{\Omega} = \mathbb{R} \times [0, T]$:

$$(8.2.15) \quad \Rightarrow \quad \int_{\tilde{\Omega}} \operatorname{div}_{(x,t)} \left(\begin{pmatrix} f(u) \\ u \end{pmatrix} \right) \Phi(\mathbf{x}, t) \, d\mathbf{x} \, dt = 0 .$$

STEP II: Perform integration by parts using Green's first formula Thm. 2.4.7 on $\tilde{\Omega}$:

$$\int_{\tilde{\Omega}} \operatorname{div}_{(x,t)} \left(\begin{pmatrix} f(u) \\ u \end{pmatrix} \right) \Phi(\mathbf{x}, t) \, d\mathbf{x} \, dt = 0$$

$$\stackrel{\text{Thm. 2.4.7}}{\Rightarrow} \int_{\tilde{\Omega}} \left(\begin{pmatrix} f(u) \\ u \end{pmatrix} \cdot \operatorname{grad}_{(x,t)} \Phi \right) \, d\mathbf{x} \, dt + \int_{-\infty}^{\infty} u(x, 0) \Phi(x, 0) \, dx = 0 ,$$

because $\partial\tilde{\Omega} = \mathbb{R} \times \{0\} \cup \mathbb{R} \times \{T\}$ with “normals” $\mathbf{n} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ ($t = 0$ boundary) and $\mathbf{n} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ ($t = T$ boundary), which has to be taken into account in the boundary term in Green’s formula. The “ $t = T$ boundary part” does not enter as $\Phi(\cdot, T) = 0$.

Note that $u(x, 0)$ is fixed by the initial condition: $u(x, 0) = u_0(x)$.

Definition 8.2.17 (Weak solution of Cauchy problem for scalar conservation law).

For $u_0 \in L^\infty(\mathbb{R})$, $u : \mathbb{R} \times]0, T[\mapsto \mathbb{R}$ is a **weak solution of the Cauchy problem (8.2.7)**, if

$$u \in L^\infty(\mathbb{R} \times]0, T[) \quad \wedge \quad \int_{-\infty}^{\infty} \int_0^T \left\{ u \frac{\partial \Phi}{\partial t} + f(u) \frac{\partial \Phi}{\partial x} \right\} dt dx + \int_{-\infty}^{\infty} u_0(x) \Phi(x, 0) dx = 0 ,$$

for all $\Phi \in C_0^\infty(\mathbb{R} \times [0, T[)$, $\Phi(\cdot, T) = 0$.

Remark 8.2.18 (Properties of weak solutions).

By reversing integration by parts, it is easy to see that

$$\textcolor{blue}{u} \text{ weak solution of (8.2.7) \&} \textcolor{blue}{u} \in C^1 \iff \textcolor{blue}{u} \text{ classical solution of (8.2.7).}$$

Arguments from mathematical integration theory confirm

$$\textcolor{blue}{u} \in L_{\text{loc}}^\infty(\mathbb{R} \times]0, T[) \text{ weak solution of (8.2.7)} \Rightarrow \begin{aligned} &\textcolor{blue}{u} \text{ satisfies integral form (8.2.16)} \\ &\text{for "almost all" } \textcolor{blue}{x}_0 < x_1, 0 < t_0 < t_1 < T. \end{aligned}$$



8.2.4 Jump conditions

For piecewise smooth vectorfield $\mathbf{j} : \Omega \subset \mathbb{R}^2$:

$$\text{"div } \mathbf{j} = 0\text{"}$$

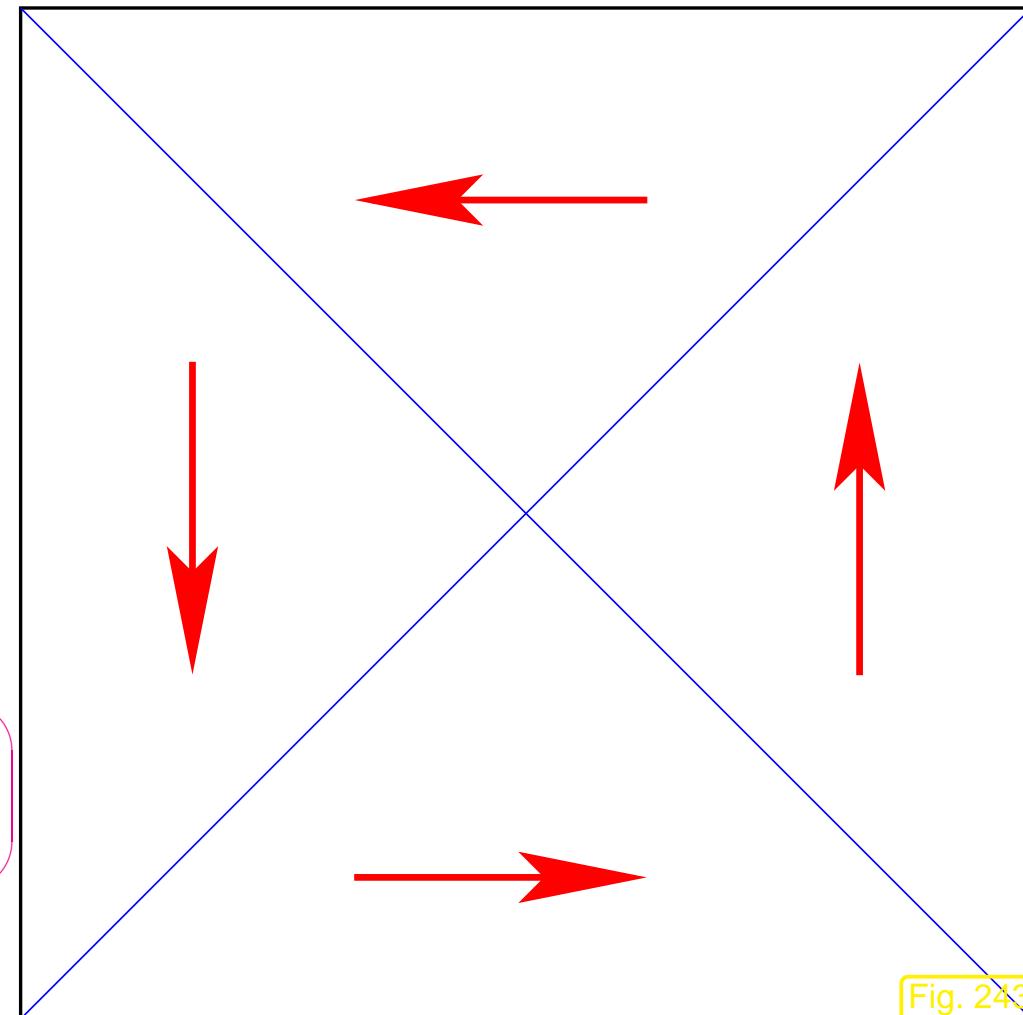
\Updownarrow

$$\int_V \mathbf{j} \cdot \mathbf{n} dS = 0 \quad \forall \text{ control volumes } V \subset \Omega$$

Necessary condition:

Continuity of **normal components**
across discontinuities

discontinuous divergence-free vectorfield



Apply this to vectorfield on space-time domain $\tilde{\Omega} = \mathbb{R} \times]0, T[$:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \iff \text{div}_{(x,t)} \underbrace{\begin{pmatrix} f(u) \\ u \end{pmatrix}}_{=: \mathbf{j}} = 0 \quad \text{in } \tilde{\Omega}. \quad (8.2.15)$$

Normal at C^1 -curve $\Gamma := \tau \mapsto (\gamma(\tau), \tau)$ in $(\gamma(\tau), \tau)$

$$\tilde{\mathbf{n}} = \frac{1}{\sqrt{1 + |\dot{s}|^2}} \begin{pmatrix} 1 \\ -\dot{s} \end{pmatrix}, \quad \dot{s} := \frac{d\gamma}{d\tau}(\tau) \quad \text{"speed of curve" }.$$

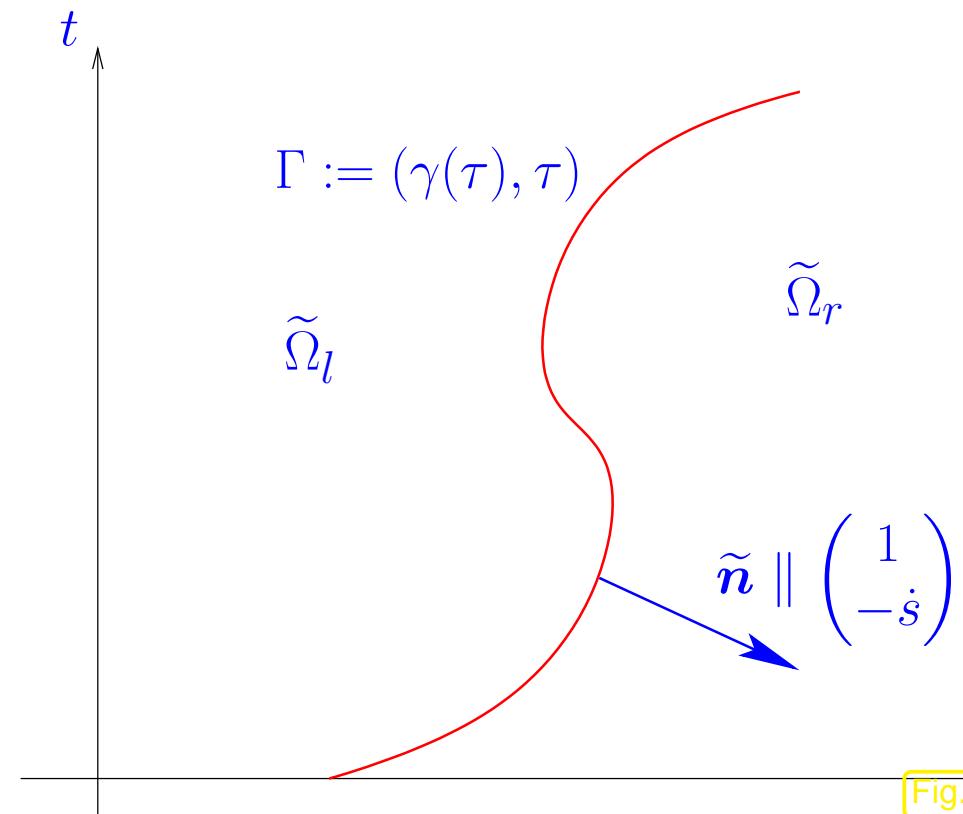
To see this, recall that the normal is orthogonal to the tangent vector $\begin{pmatrix} \dot{s} \\ 1 \end{pmatrix}$ and that in 2D the direction orthogonal to $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ is given by $\begin{pmatrix} -x_2 \\ x_1 \end{pmatrix}$.

“normal continuity” of
piecewise smooth vectorfield $(f(u), u)^T$



$$\begin{pmatrix} 1 \\ -\frac{d\gamma}{d\tau} \end{pmatrix} \cdot \begin{pmatrix} [f(u)] \\ [u] \end{pmatrix} = 0, \quad (8.2.19)$$

where $[\cdot] \hat{=} \text{jump across } \Gamma$.



Terminology: (8.2.19) = Rankine-Hugoniot (jump) condition, shorthand notation:

$$\boxed{\dot{s}(u_l - u_r) = f_l - f_r} \quad , \quad \dot{s} := \frac{d\gamma}{d\tau} \quad \text{"propagation speed of discontinuity"} \quad (8.2.20)$$

Remark 8.2.21 (Discontinuity connecting constant states).

The simplest situation compliant with Rankine-Hugoniot jump condition: **constant states** to the left and right of the curve of discontinuity (8.2.19):

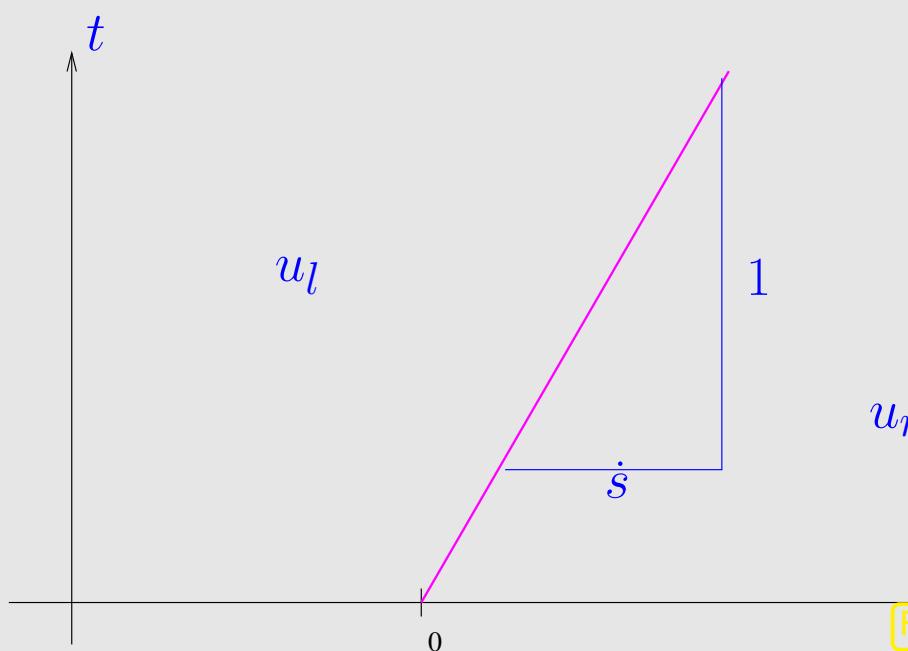


Fig. 245c

$$u(x, t) = \begin{cases} u_l \in \mathbb{R} & , \text{for } x < \dot{s}t, \\ u_r \in \mathbb{R} & , \text{for } x > \dot{s}t, \end{cases} \quad (8.2.22)$$

with **constant** speed \dot{s} of discontinuity, according to (8.2.20) given by (for $u_l \neq u_r$)

$$\dot{s} = \frac{f(u_l) - f(u_r)}{u_l - u_r}.$$



8.2.5 Riemann problem

Rem. 8.2.21: situation of locally constant states in particularly easy.

- Consider: Cauchy-problem (8.2.7) for piecewise constant initial data u_0 .

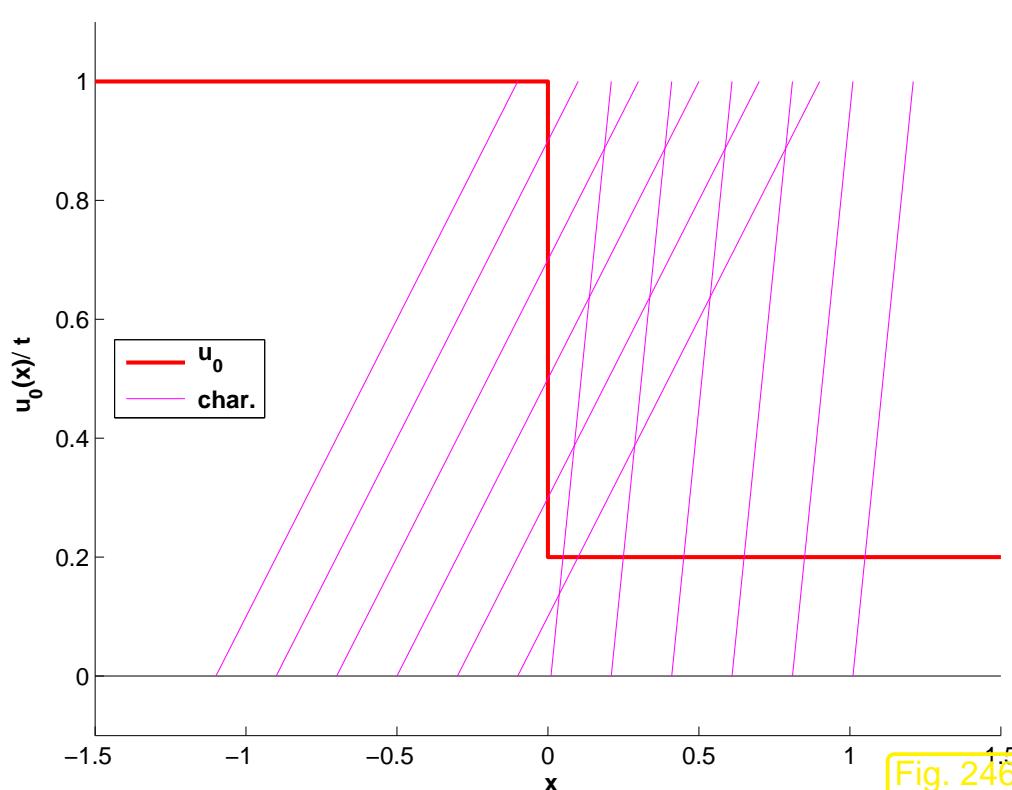
Definition 8.2.23 (Riemann problem).

$$u_0(x) = \begin{cases} u_l \in \mathbb{R} & , \text{ if } x < 0 \\ u_r \in \mathbb{R} & , \text{ if } x > 0 \end{cases} \quad \hat{=} \quad \textcolor{red}{\text{Riemann problem for (8.2.7)}}$$

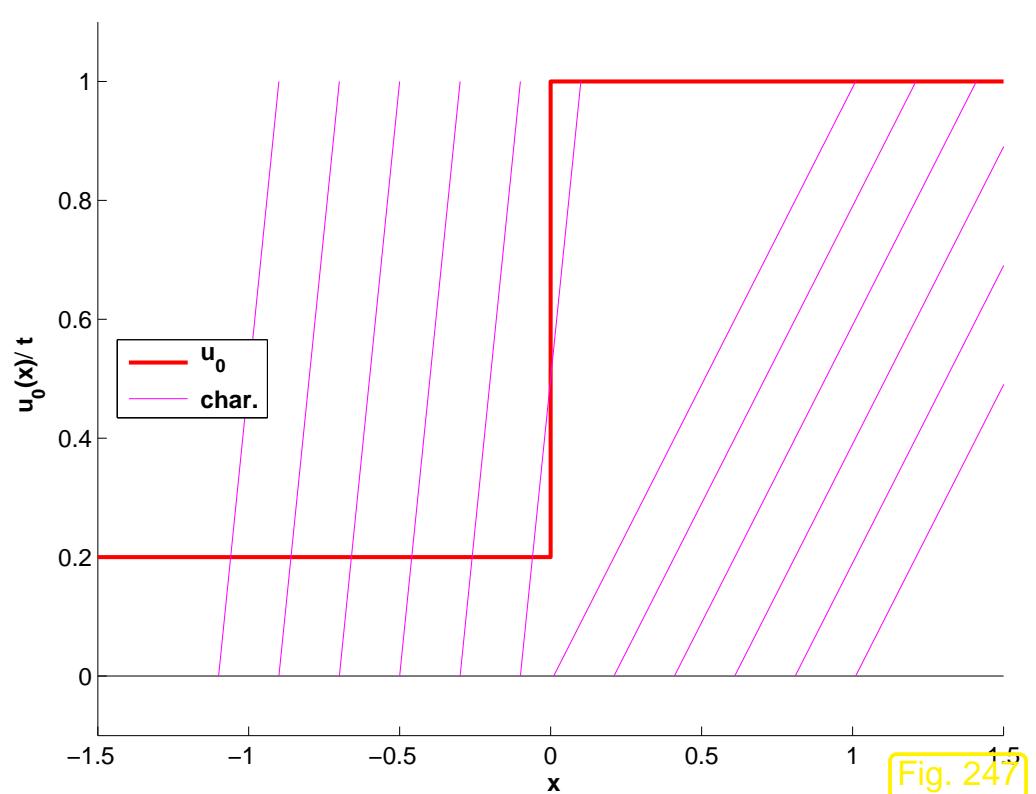
Assumption, cf. Sect. 8.2.2:

flux function $f : \mathbb{R} \mapsto \mathbb{R}$ smooth & **convex**

- f' non-decreasing ► pattern of characteristic curves for Riemann problem:



intersecting characteristics



diverging characteristics

Definition 8.2.24 (Shock).

If Γ is a smooth curve in the (x, t) -plane and u a weak solution of (8.2.7), a discontinuity of u across Γ is called a **shock**.

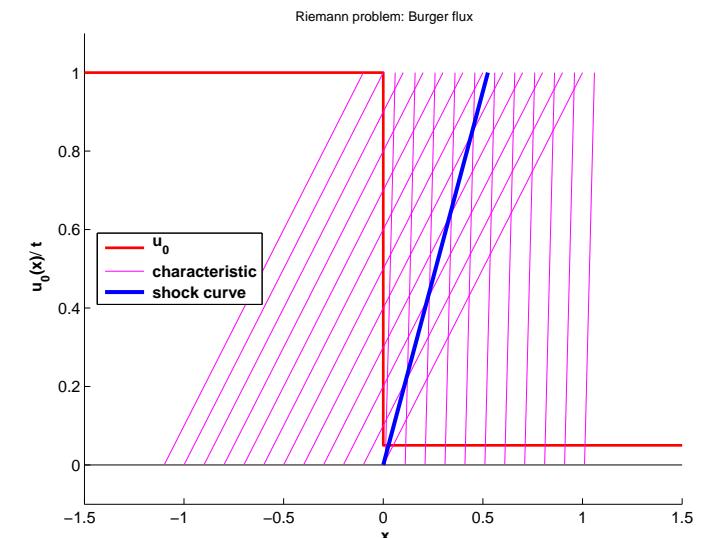
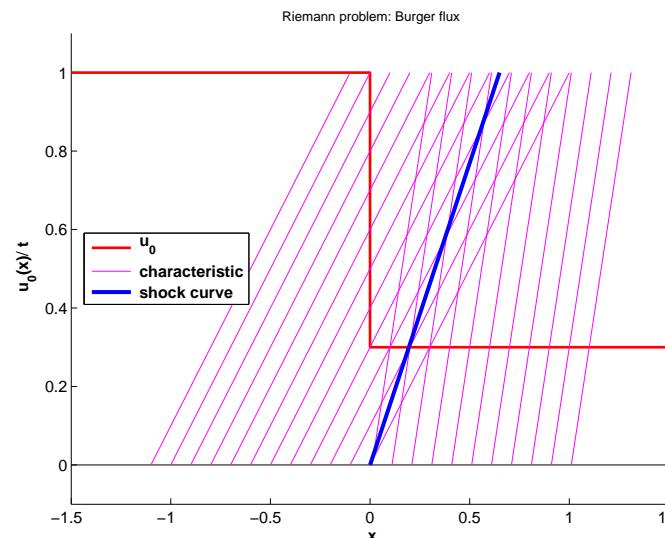
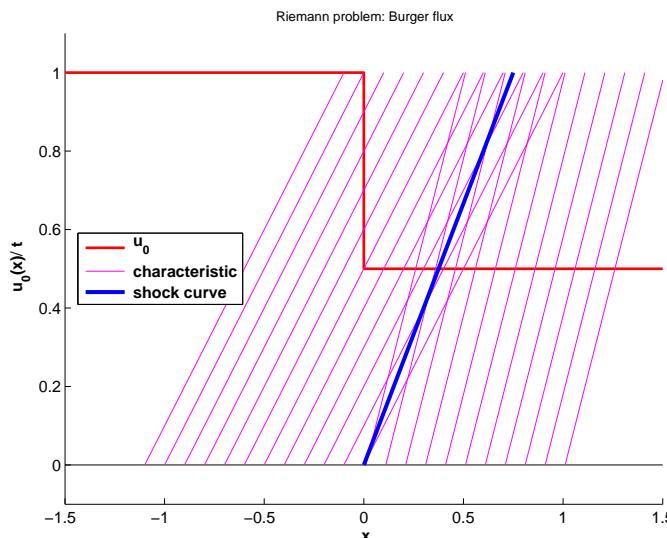
Rem. 8.2.21 ➤ **shock speed s** \leftrightarrow Rankine-Hugoniot jump conditions:

$$(x_0, t_0) \in \Gamma: \quad \dot{s} = \frac{f(u_l) - f(u_r)}{u_l - u_r}, \quad u_l := \lim_{\epsilon \rightarrow 0} u(x_0 - \epsilon, t_0), \quad u_r := \lim_{\epsilon \rightarrow 0} u(x_0 + \epsilon, t_0). \quad (8.2.25)$$

Lemma 8.2.26 (Shock solution of Riemann problem).

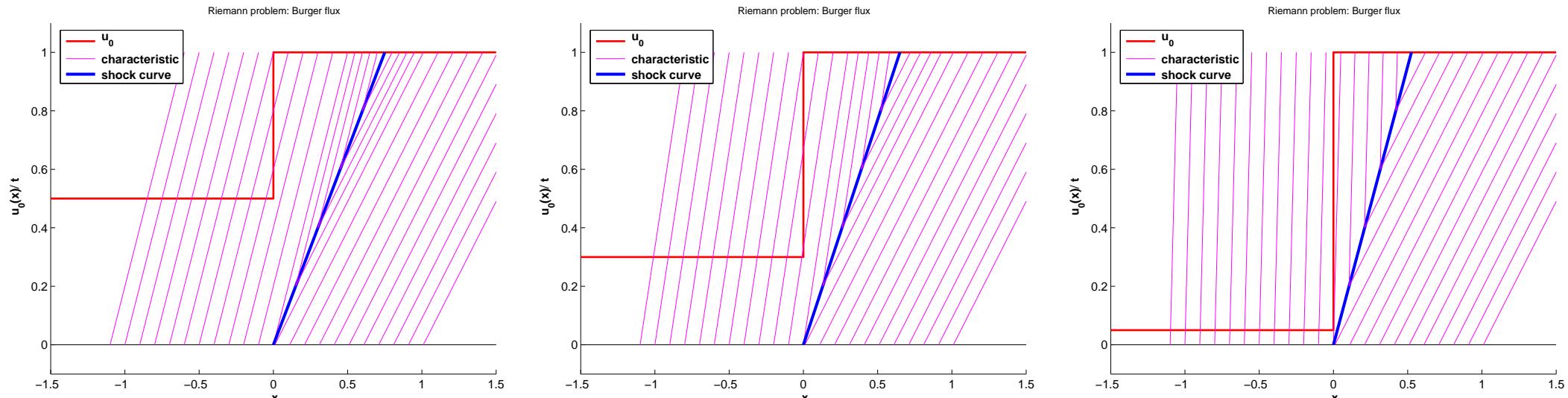
$$u(x, t) = \begin{cases} u_l & \text{for } x < st, \\ u_r & \text{for } x > st, \end{cases} \quad \dot{s} := \frac{f(u_l) - f(u_r)}{u_l - u_r}, \quad x \in \mathbb{R}, 0 < t < T,$$

is weak solution of Riemann problem (\rightarrow Def. 8.2.23) for (8.2.7).



Burgers flux $f(u) = \frac{1}{2}u^2$, $u_l > u_r$: characteristic curves impinge on shock

Fig. 248



Burgers flux $f(u) = \frac{1}{2}u^2$, $u_l < u_r$: characteristic curves emanate from shock
(expansion shock)

Fig. 249

Example 8.2.27 (Vanishing viscosity for Burgers equation).

There is no such material as an “inviscid” fluid in nature, because in any physical system there will be a tiny amount of friction. This leads us to the very general understanding that conservation laws can usually be regarded as limit problems $\epsilon = 0$ for singularly perturbed transport-diffusion problems with an “ ϵ -amount” of diffusion.

In 1D, for any $\epsilon > 0$ these transport-diffusion problems will possess a unique smooth solution. Studying its behavior for $\epsilon \rightarrow 0$ will tell us, what are “physically meaningful” solutions for the conservation

law. This consideration is called the **vanishing viscosity** method to define solutions for conservation laws.

Here we pursue this idea for Burgers equation, see Sect. 8.1.2.

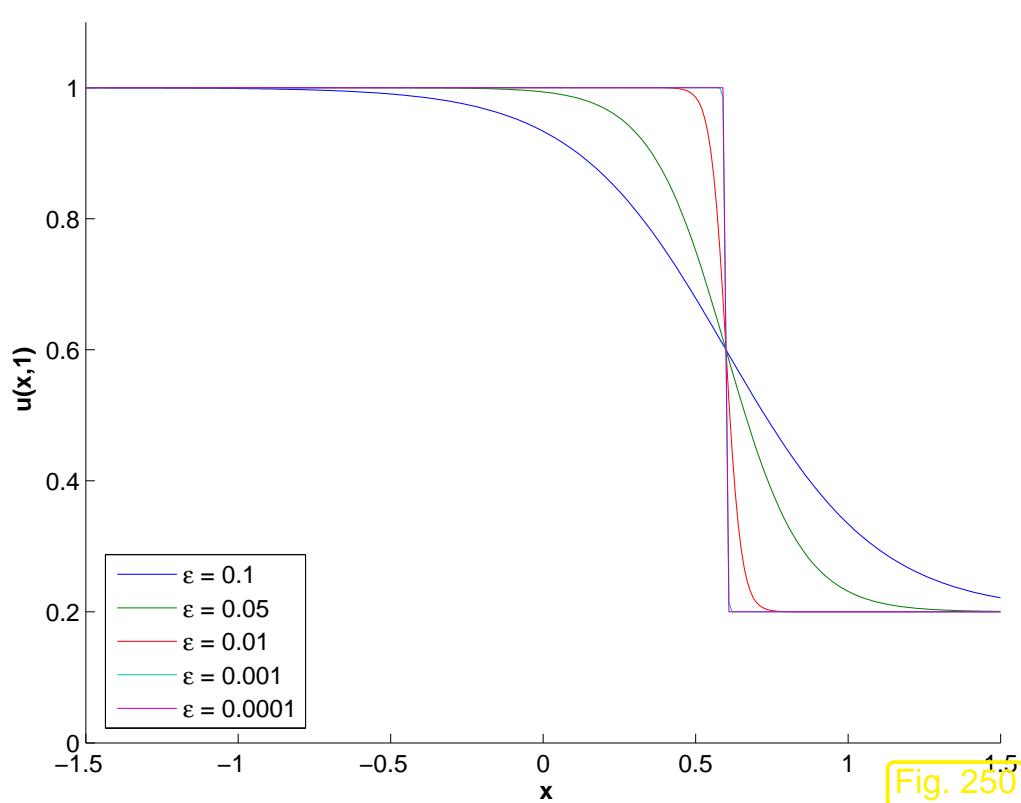
Viscous Burgers equation:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{1}{2} u^2 \right) = \epsilon \frac{\partial^2 u}{\partial x^2}. \quad (8.2.28)$$


dissipative term

Travelling wave solution of Riemann problem for (8.2.28) via Cole-Hopf transform → [10, Sect. 4.4.1]

$$u_\epsilon(x, t) = w(x - \dot{s}t) \quad , \quad w(\xi) = u_r + \frac{1}{2}(u_l - u_r)(1 - \tanh\left(\frac{\xi(u_l - u_r)}{4\epsilon}\right)) \quad , \quad \dot{s} = \frac{1}{2}(u_l + u_r).$$



$u_\epsilon(x, t)$ = classical solution of (8.2.28) for all $t > 0$,
 $x \in \mathbb{R}$ (only for $u_l > u_r$!).

▷

$$u_l > u_r, \quad t = 0.5$$

emerging shock for $\epsilon \rightarrow 0$

$u_\epsilon \rightarrow u$ from Lemma 8.2.26 in $L^\infty(\mathbb{R})$.

Fig. 250

Highly accurate numerical solution of

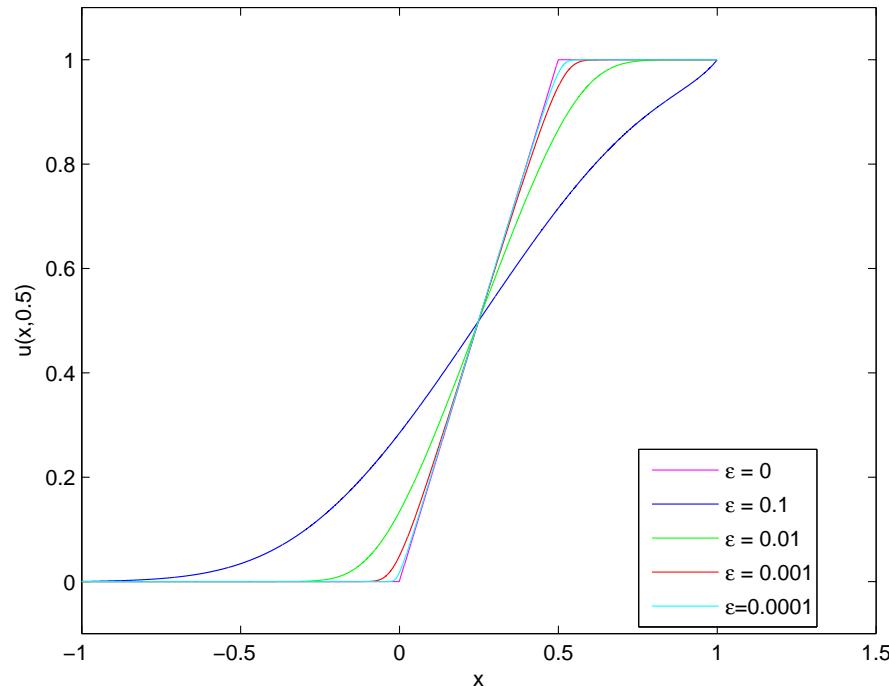
Riemann problem for (8.2.28)

$$u_l < u_r$$

$$u_\epsilon(x, 0.5) \triangleright$$

no shock as $\epsilon \rightarrow 0$!

$u_\epsilon \rightarrow$ a piecewise linear function!



Let us try to derive a (weak) solution of the homogeneous scalar conservation law (8.2.14) with the structure observed in Ex. 8.2.27.

Idea: conservation law (8.2.14) homogeneous in spatial/temporal derivatives:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \text{in } \mathbb{R} \times \mathbb{R}^+ \Rightarrow \frac{\partial u_\lambda}{\partial t} + \frac{\partial}{\partial x} f(u_\lambda) = 0 \quad \text{in } \mathbb{R} \times \mathbb{R}^+,$$

8.2

p. 824

$u_\lambda(x, t) := u(\lambda x, \lambda t)$, $\lambda > 0$. This suggests that we look for solutions of the Riemann problem that are constant on all straight lines in the $x - t$ -plane that cross $(0, 0)^T$.

► try similarity solution:

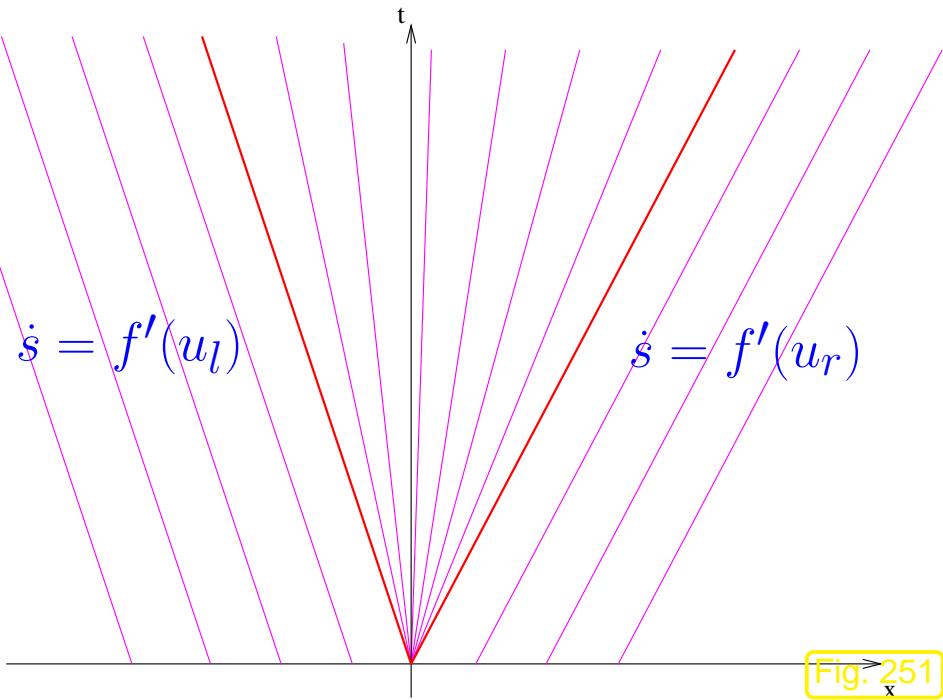
$$u(x, t) = \psi(x/t)$$

→ insert in $\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0$

$$f'(\psi(x/t))\psi'(x/t) = (x/t)\psi'(x/t) \quad \forall x \in \mathbb{R}, 0 < t < T .$$

► $\psi' \equiv 0 \quad \vee \quad f'(\psi(w)) = w \iff \psi(w) = (f')^{-1}(w) .$

f' strictly monotone !



Lemma 8.2.29. (Rarefaction solution of Riemann problem)

If $f \in C^2(\mathbb{R})$ strictly convex, $u_l < u_r$, then

$$u(x, t) := \begin{cases} u_l & \text{for } x < f'(u_l)t, \\ g\left(\frac{x}{t}\right) & \text{for } f'(u_l) < \frac{x}{t} < f'(u_r), \\ u_r & \text{for } x > f'(u_r)t, \end{cases}$$

$g := (f')^{-1}$, is a weak solution of the Riemann problem (\rightarrow Def. 8.2.23).

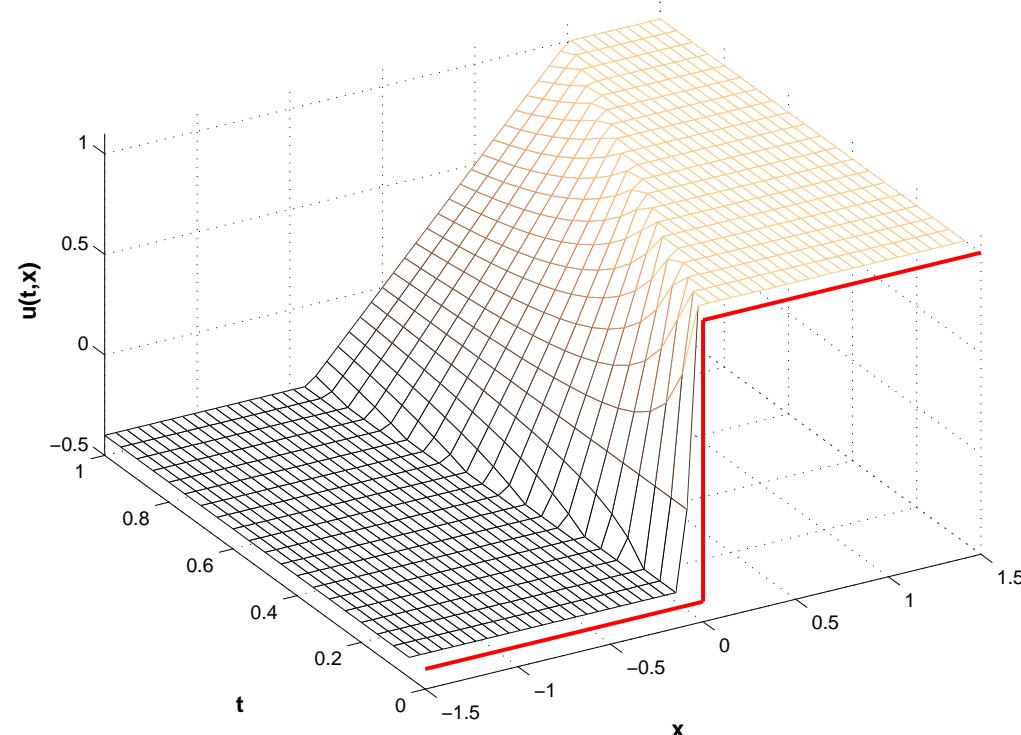
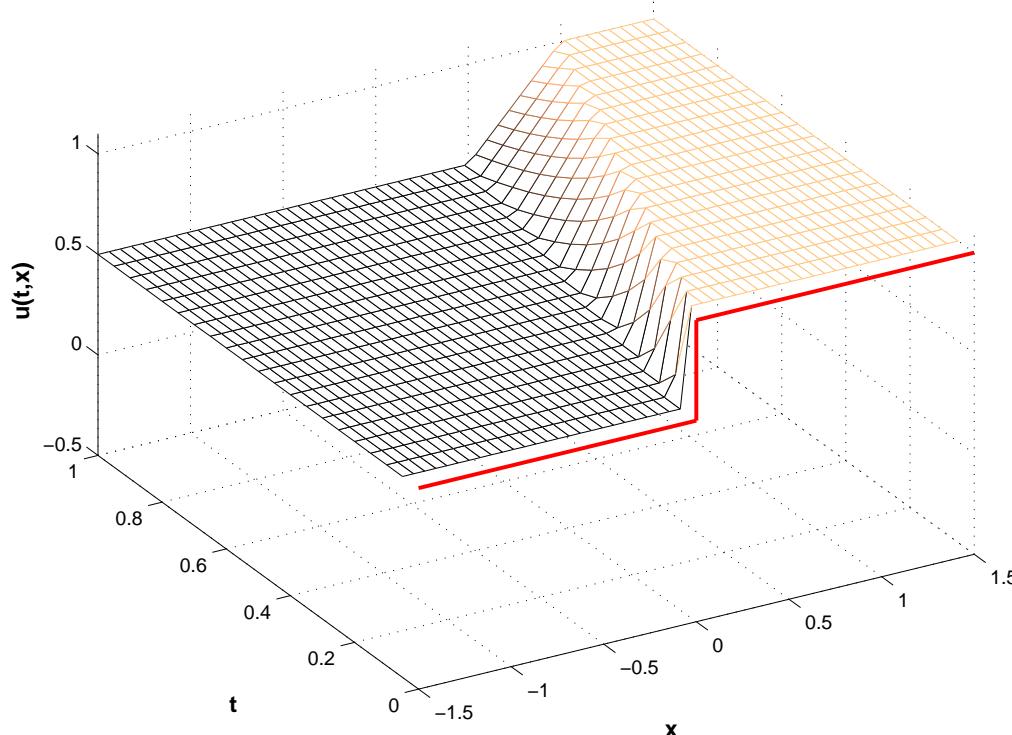
Proof. We show that the rarefaction solution is a weak solution according to Def. 8.2.17 \Rightarrow for $\Phi \in C_0^\infty(\mathbb{R} \times]0, T[)$

$$\int_0^T \left\{ \int_{-\infty}^{f'(u_l)t} u_l \frac{\partial \Phi}{\partial t} + f(u_l) \frac{\partial \Phi}{\partial x} dx + \int_{f'(u_l)t}^{f'(u_r)t} g\left(\frac{x}{t}\right) \frac{\partial \Phi}{\partial t} + f(g\left(\frac{x}{t}\right)) \frac{\partial \Phi}{\partial x} dx + \int_{f(u_r)t}^\infty u_r \frac{\partial \Phi}{\partial t} + F(u_r) \frac{\partial \Phi}{\partial x} dx \right\} dt$$

$$= \int_0^T \int_{f'(u_l)t}^{f'(u_r)t} g'\left(\frac{x}{t}\right) \frac{x}{t^2} \Phi - f'(g\left(\frac{x}{t}\right)) \frac{1}{t} g'\left(\frac{x}{t}\right) \Phi dx dt = 0,$$

because $f' \circ g = Id$ and by fundamental theorem of calculus. □

Terminology: solution of Lemma 8.2.29 = rarefaction wave: continuous solution !

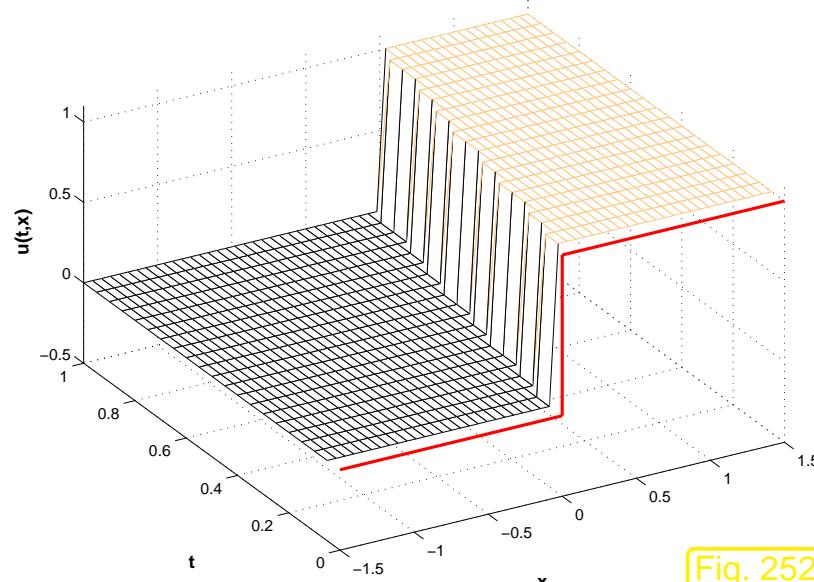


Burger flux function $f(u) = \frac{1}{2}u^2$, $u_l < u_r$: rarefaction wave solutions

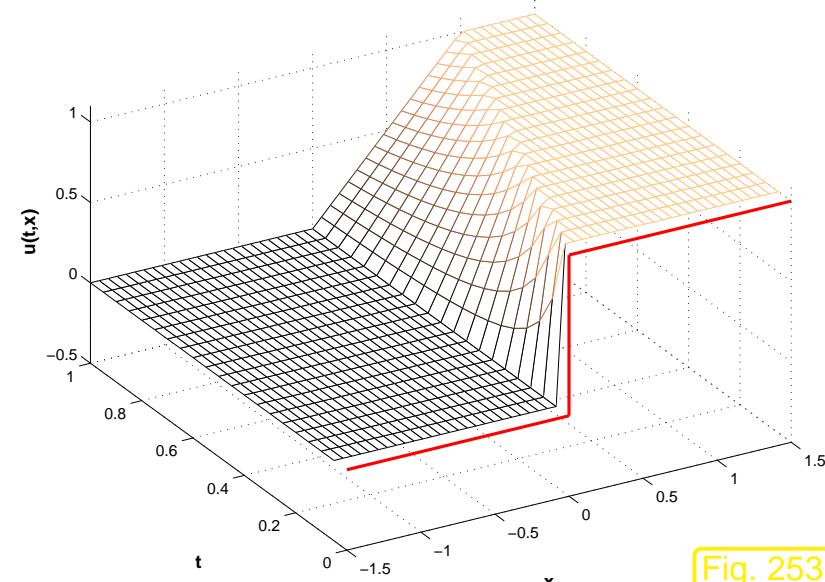
8.2.6 Entropy condition

Sect 8.2.5 ➤ Non-uniqueness of weak solutions:

if f' is increasing and $u_l < u_r$ both a shock and a rarefaction wave provide valid weak solutions.



Riemann solution (Burgers equation):
shock



Riemann solution (Burgers equation):
rarefaction wave

How to select “physically meaningful” = admissible solution ?

Vanishing viscosity technique (\rightarrow Ex. 8.2.27): add an “ ϵ -amount” of diffusion (“friction”) and study solution for $\epsilon \rightarrow 0$.

8.2

p. 828

However, desirable: simple selection criteria (entropy conditions)

Definition 8.2.30 (Lax entropy condition).

u $\hat{=}$ weak solution of (8.2.7), piecewise classical solution in a neighborhood of C^2 -curve $\Gamma := (\gamma(\tau), \tau), 0 \leq \tau \leq T$, discontinuous across Γ .

u satisfies the Lax entropy condition in $(x_0, t_0) \in \Gamma$ $\Leftrightarrow f'(u_l) > \dot{s} := \frac{f(u_l) - f(u_r)}{u_l - u_r} > f'(u_r)$.

\Updownarrow

Characteristic curves must not emanate from shock \leftrightarrow no “generation of information”

Parlance: shock satisfying Lax entropy condition = physical shock

Note: f' increasing \rightarrow Def 8.2.30: necessary for physical shock

$$u_l > u_r$$

Physically meaningful weak solution of conservation law = **entropy solution**

For *scalar* conservation laws with locally Lipschitz-continuous flux function f :

Existence & uniqueness of entropy solutions

Remark 8.2.31 (General entropy solution for 1D scalar Riemann problem). → [16]

Entropy solution of Riemann problem (→ Def. 8.2.23) for (8.2.7) with arbitrary $f \in C^1(\mathbb{R})$:

$$u(x, t) = \psi(x/t) \quad , \quad \psi(\xi) := \begin{cases} \underset{u_l \leq u \leq u_r}{\operatorname{argmin}} (f(u) - \xi u) & , \text{ if } u_l < u_r , \\ \underset{u_r \leq u \leq u_l}{\operatorname{argmax}} (f(u) - \xi u) & , \text{ if } u_l \geq u_r . \end{cases} \quad (8.2.32)$$



Example 8.2.33 (Entropy solution of Burgers equation).

Analytical solution available for Burgers equation (8.1.11) with initial data, see [10, Sect. 3.4, Ex. 3]

$$u_0(x) = \begin{cases} 0 & , \text{ if } x < 0 \text{ or } x > 1 , \\ 1 & , \text{ if } 0 \leq x \leq 1 . \end{cases}$$

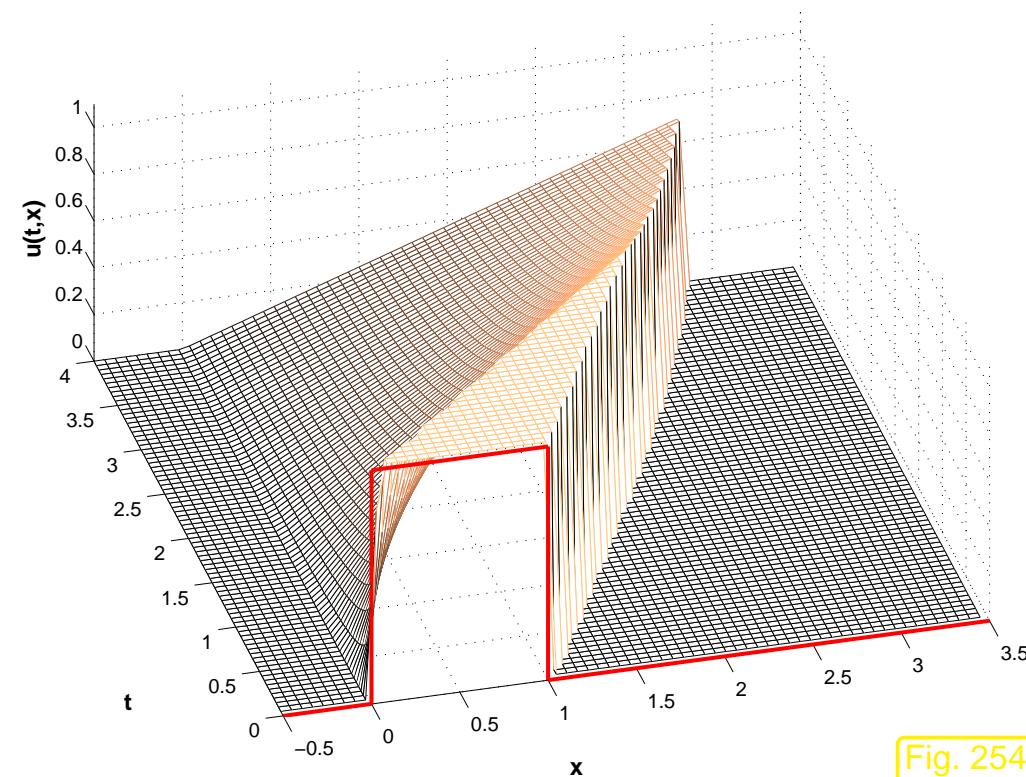


Fig. 254

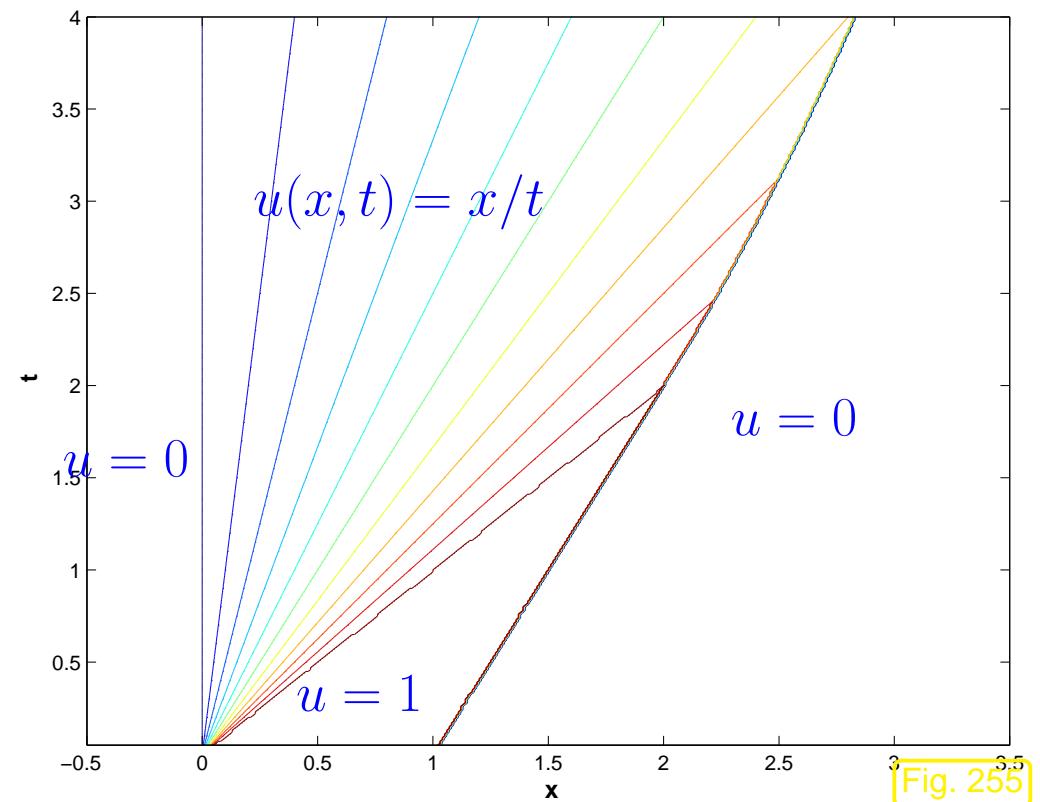


Fig. 255

Vector field in $x - t$ -plane

$$\begin{pmatrix} f(u(x, t)) \\ u(x, t) \end{pmatrix}$$

for entropy solution $u = u(x, t)$ ▷.

Observe the normal continuity across the shock:
the vector field is tangential to the shock curve.

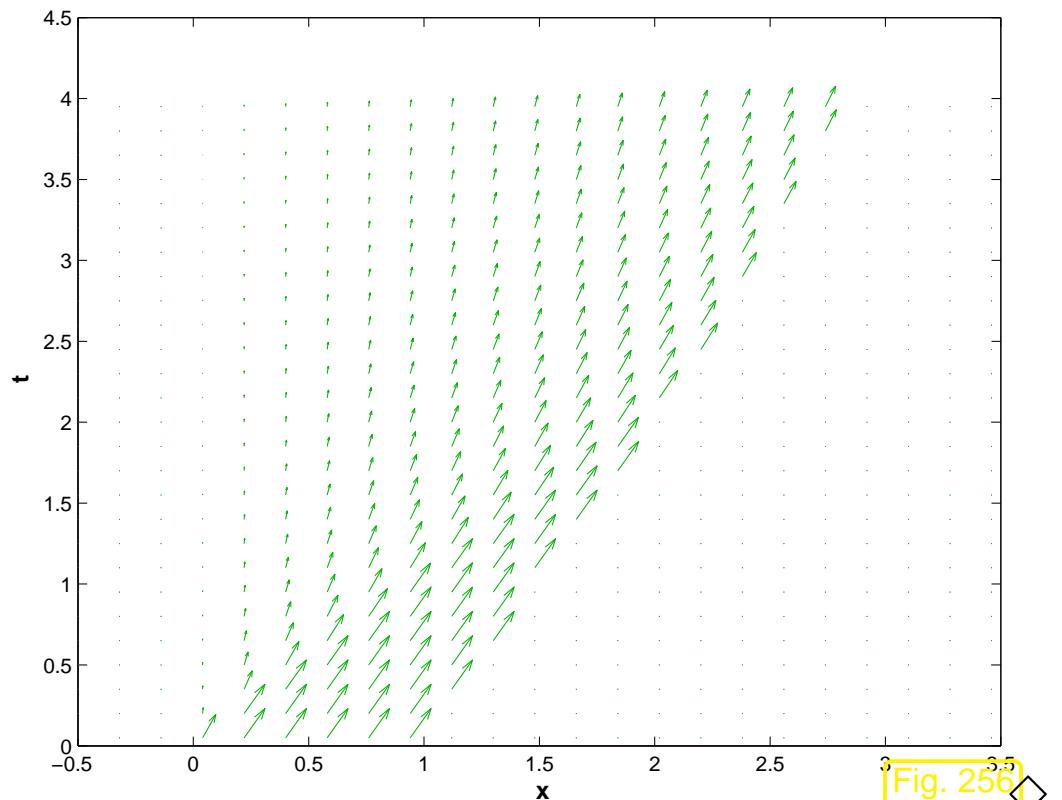


Fig. 256 ◇

8.2.7 Properties of entropy solutions

Setting: $u \in L^\infty(\mathbb{R} \times]0, T[)$ weak (\rightarrow Def. 8.2.17) entropy solution of Cauchy problem

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \text{in } \mathbb{R} \times]0, T[, \quad u(x, 0) = u_0(x) , \quad x \in \mathbb{R} . \quad (8.2.7)$$

with flux function $f \in C^1(\mathbb{R})$ (not necessarily convex/concave).

Notation: $\bar{u} \in L^\infty(\mathbb{R} \times]0, T[) \hat{=} \text{entropy solution w.r.t. initial data } \bar{u}_0 \in L^\infty(\mathbb{R})$.

Theorem 8.2.34 (Comparison principle for scalar conservation laws).

$$\text{If } u_0 \leq \bar{u}_0 \text{ a.e. on } \mathbb{R} \Rightarrow u \leq \bar{u} \text{ a.e. on } \mathbb{R} \times]0, T[$$

With obvious consequences:

$$\blacktriangleright \quad u_0(x) \in [\alpha, \beta] \text{ on } \mathbb{R} \Rightarrow u(x, t) \in [\alpha, \beta] \text{ on } \mathbb{R} \times]0, T[$$

► L^∞ -stability (\Rightarrow no blow-up can occur!)

$$\forall 0 \leq t \leq T: \|u(\cdot, t)\|_{L^\infty(\mathbb{R})} \leq \|u_0\|_{L^\infty(\mathbb{R})} . \quad (8.2.35)$$

Theorem 8.2.36 (L^1 -contractivity of evolution for scalar conservation law).

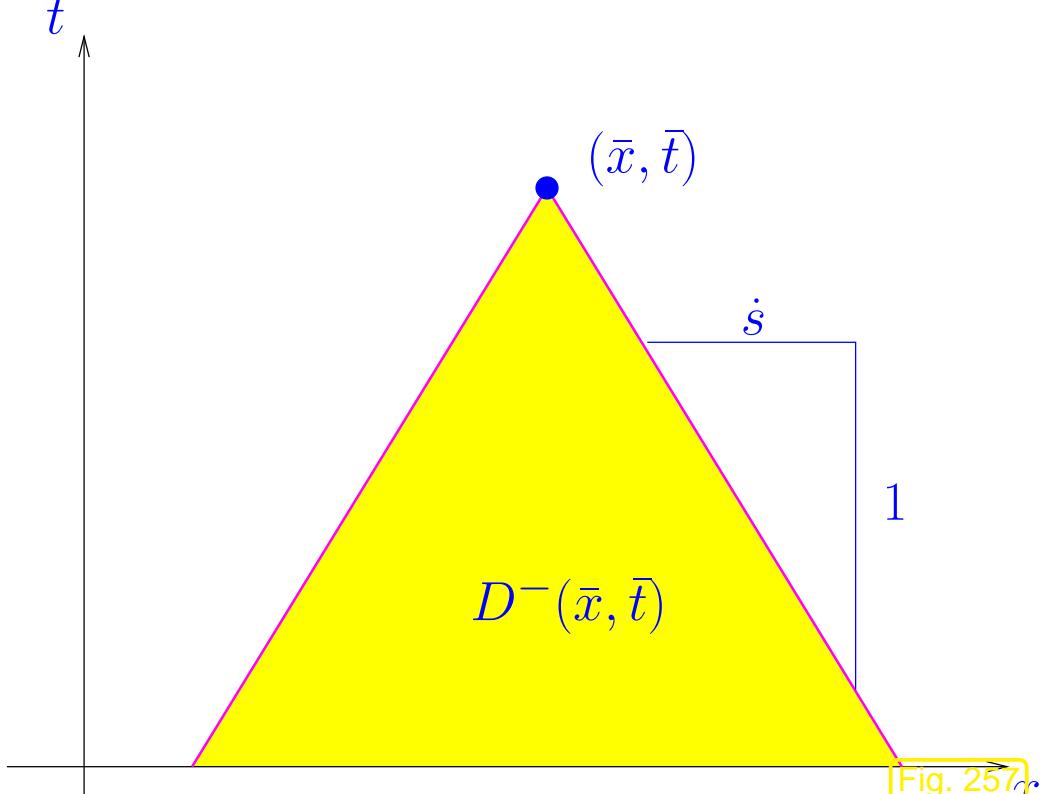
$$\forall t \in]0, T[, R > 0: \int_{|x| < R} |u(x, t)| dx \leq \int_{|x| < R + \dot{s}t} |u_0(x)| dx ,$$

with *maximal speed of propagation*

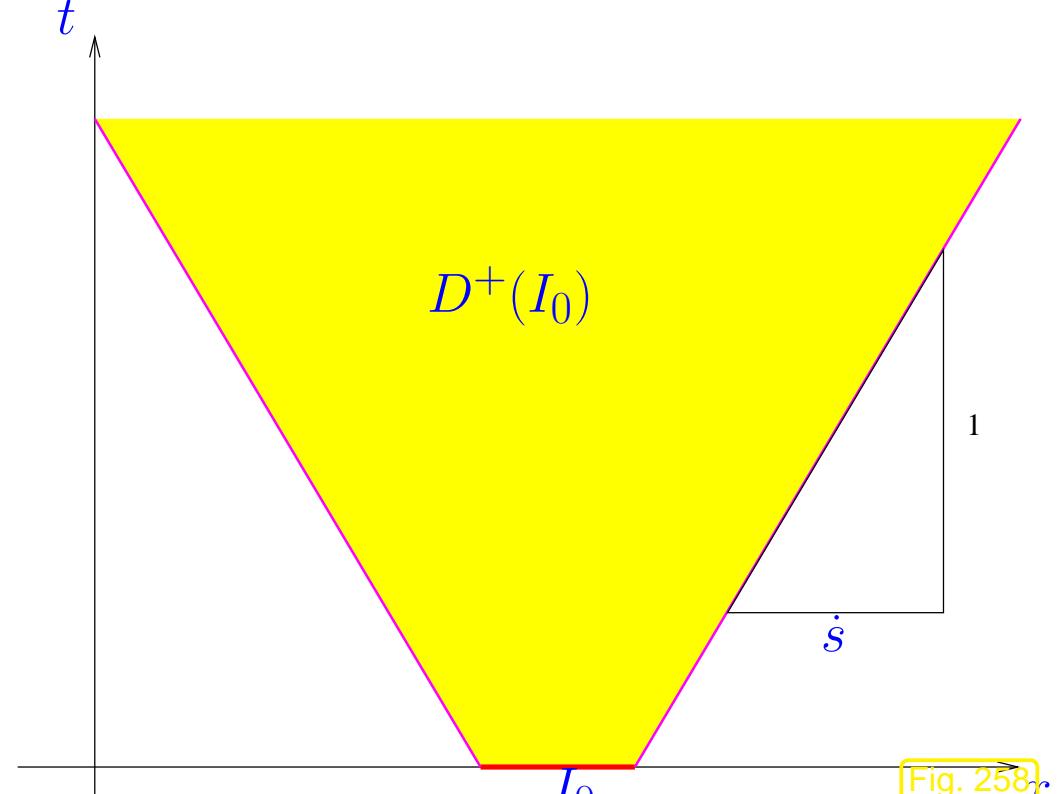
$$\dot{s} := \max\{|f'(\xi)| : \inf_{x \in \mathbb{R}} u_0(x) \leq \xi \leq \sup_{x \in \mathbb{R}} u_0(x)\} . \quad (8.2.37)$$

Thm. 8.2.36 ► finite speed of propagation in conservation law, bounded by \dot{s} from (8.2.37):

► As in the case of the wave equation → Sect. 6.2.2:



maximal domain of dependence of $(\bar{x}, \bar{t}) \in \tilde{\Omega}$



maximal domain of influence of $I_0 \subset \mathbb{R}$

Analogous to Thm. 6.2.18:

Corollary 8.2.38 (Domain of dependence for scalar conservation law). $\rightarrow [8, \text{Cor. 6.2.2}]$

The value of the entropy solution at $(\bar{x}, \bar{t}) \in \tilde{\Omega}$ depends only on the restriction of the initial data to $\{x \in \mathbb{R}: |x - \bar{x}| < \dot{s}\bar{t}\}$.

Another strand of theoretical results asserts that the solution of a 1D scalar conservation law cannot develop oscillations:

u solves (8.2.7) ➤ No. of local extrema (in space) of $u(\cdot, t)$ decreasing with time

8.3 Conservative finite volume discretization

Example 8.3.1 (Naive finite difference scheme).

Cauchy problem for Burgers equation (8.1.11) rewritten using product rule:

$$\frac{\partial u}{\partial t}(x, t) + u(x, t) \frac{\partial u}{\partial x}(x, t) = 0 \quad \text{in } \mathbb{R} \times]0, T[.$$

\leftrightarrow related to advection with velocity $v(x, t) = u(x, t)$:

$$\frac{\partial u}{\partial t}(x, t) + u(x, t) \frac{\partial u}{\partial x}(x, t) = 0 \quad \text{in } \mathbb{R} \times]0, T[.$$



$$\frac{\partial u}{\partial t}(x, t) + v(x, t) \frac{\partial u}{\partial x}(x, t) = 0 \quad \text{in } \mathbb{R} \times]0, T[.$$

If $u_0(x) \geq 0$, then, by Thm. 8.2.34, $u(x, t) \geq 0$ for all $0 < t < T$, that is, positive direction of transport throughout.

Heeding guideline from Sect. 7.3.1: use **upwind discretization** (backward differences) in space!

- on (infinite) equidistant grid, meshwidth $h > 0$, $x_j = hj$, $j \in \mathbb{Z}$, obtain semi-discrete problem for nodal values $\mu_j = \mu_j(t) \approx u(x_j, t)$

$$\frac{\partial u}{\partial t}(x, t) + u(x, t) \frac{\partial u}{\partial x}(x, t) = 0 \quad \text{in } \mathbb{R} \times]0, T[.$$

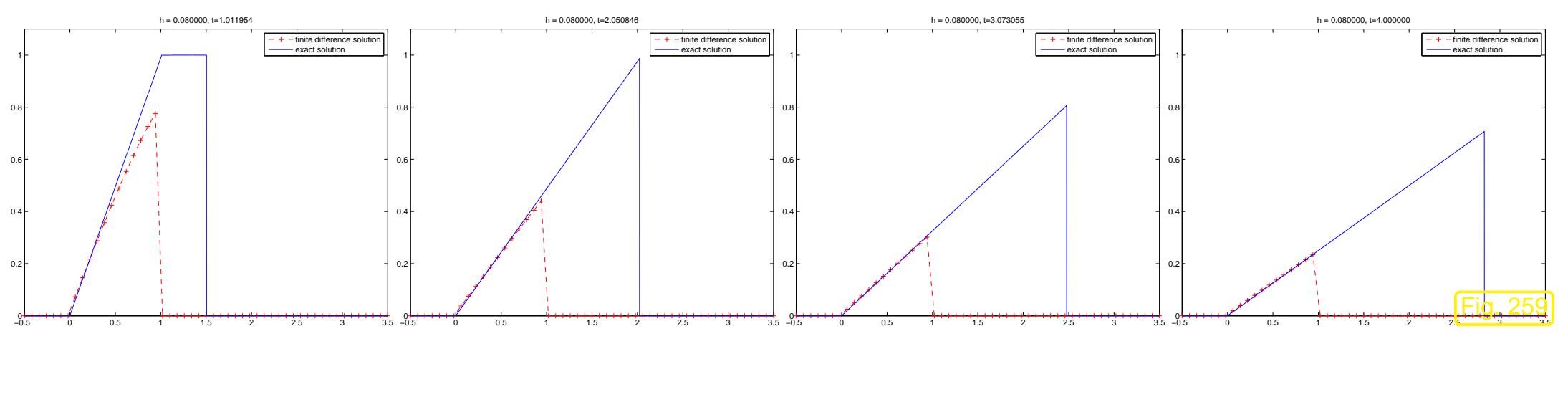


(8.3.2)

$$\dot{\mu}_j(t) + \mu_j \frac{\mu_j - \mu_{j-1}}{h} = 0, \quad j \in \mathbb{Z}, \quad 0 < t < T .$$

Numerical experiment with Cauchy problem from Ex. 8.2.33, $h = 0.08$, integration of (8.3.2) with MATLAB `ode45`.

8.3



Observation from numerical experiment: OK for rarefaction wave, but *scheme cannot capture speed of shock correctly!*

Analysis: consider $\mu_j(0) = \begin{cases} 1 & , \text{if } j < 0 , \\ 0 & , \text{if } j \geq 0 . \end{cases}$

\leftrightarrow Riemann problem with $u_0(x) = 1$ for $x < 0 - \epsilon$, $u_0(x) = 0$ for $x > 0 - \epsilon$, $\epsilon \ll 1$.

Entropy solution (for this u_0) = travelling
shock (\rightarrow Lemma 8.2.26), speed
 $\dot{s} = \frac{1}{2} > 0$



Numerical solution:

$$\vec{\mu}(t) = \vec{\mu}_0 \text{ for all } t > 0 !$$

- 3-point FDM (??) “converges” to wrong solution !

◇

8.3.1 Semi-discrete conservation form

Objective: spatial semi-discretization of Cauchy problem

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \text{in } \mathbb{R} \times]0, T[, \quad u(x, 0) = u_0(x) , \quad x \in \mathbb{R} . \quad (8.2.7)$$

on (infinite) equidistant spatial mesh with mesh width $h > 0$

$$\mathcal{M} := \{]x_{j-1}, x_j[: x_j := jh, j \in \mathbb{Z}\} . \quad (8.3.3)$$

mesh cells and dual cells

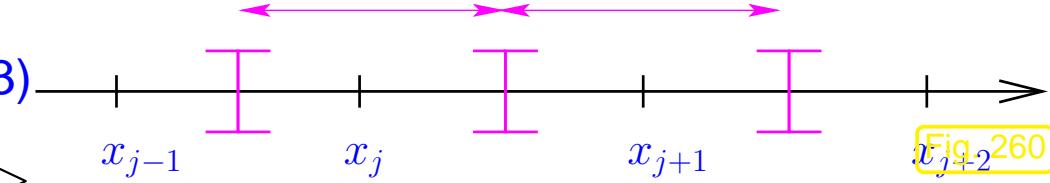


Fig. 260

8.3

p. 839

Finite volume interpretation of nodal unknowns

= conserved quantities in **dual cells** $]x_{j-1/2}, x_{j+1/2}[$, midpoints $x_{j-1/2} := \frac{1}{2}(x_j + x_{j-1})$:

$$\mu_j(t) \approx \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t) dx . \quad (8.3.4)$$

$$\vec{\mu}(t) := (\mu_j(t))_{j \in \mathbb{Z}} \in \mathbb{R}^{\mathbb{Z}} \longleftrightarrow u_N(x, t) = \sum_{j \in \mathbb{Z}} \mu_j \chi_{]x_{j-1/2}, x_{j+1/2}[}(x) . \quad (8.3.5)$$

☞ notation: **characteristic function** $\chi_{]x_{j-1/2}, x_{j+1/2}[}(x) = \begin{cases} 1 & , \text{ if } x_{j-1/2} < x \leq x_{j+1/2} , \\ 0 & \text{elsewhere.} \end{cases}$

☞ $(\mu_j(t))_{j \in \mathbb{Z}}$ \longleftrightarrow ***piecewise constant*** approximation $u_N(t) \approx u(\cdot, t)$

By spatial integration over dual cells, which now play the role of the control volumes in (8.2.1):

$$\frac{d}{dt} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t) dx + f(u(x_{j+1/2}, t)) - f(u(x_{j-1/2}, t)) = 0 , \quad j \in \mathbb{Z} , \quad (8.3.6)$$

► $\frac{d\mu_j}{dt}(t) + \frac{1}{h} \left(\underbrace{f(u_N(x_{j+1/2}, t))}_{?} - \underbrace{f(u_N(x_{j-1/2}, t))}_{?} \right) = 0 , \quad j \in \mathbb{Z} . \quad (8.3.7)$

Problem: ambiguity of values $u_N(x_{j+1/2}, t)$, $u_N(x_{j-1/2}, t)$, as we encountered it in the context of upwind quadrature in Sect. 7.2.2.1.

Abstract “solution”:

Approximation $f(u_N(x_{j+1/2}, t)) \approx f_{j+1/2}(t) := F(\mu_{j-m_l+1}(t), \dots, \mu_{j+m_r}(t)) , \quad j \in \mathbb{Z} ,$

with **numerical flux function** $F : \mathbb{R}^{m_l+m_r} \mapsto \mathbb{R}$, $m_l, m_r \in \mathbb{N}_0$.

Note: the **same** numerical flux function is used for all dual cells!



Finite volume semi-discrete evolution for (8.2.7) in **conservation form**:

$$\frac{d\mu_j}{dt}(t) = \frac{1}{h} (F(\mu_{j-m_l+1}(t), \dots, \mu_{j+m_r}(t)) - F(\mu_{j-m_l}(t), \dots, \mu_{j+m_r-1}(t))) , \quad j \in \mathbb{Z} . \quad (8.3.8)$$

numerical flux (function) $F : \mathbb{R}^{m_l+m_r} \mapsto \mathbb{R}$

Special case: **2-point numerical flux** ($m_l = m_r = 1$): $F = F(v, w)$
($v \hat{=} \text{left state}$, $w \hat{=} \text{right state}$)

$$(8.3.8) \quad \blacktriangleright \quad \frac{d\mu_j}{dt}(t) = -\frac{1}{h} (F(\mu_j(t), \mu_{j+1}(t)) - F(\mu_{j-1}(t), \mu_j(t))) , \quad j \in \mathbb{Z} . \quad (8.3.9)$$

Assumption on numerical flux functions: F Lipschitz-continuous in each argument.

8.3.2 Discrete conservation property

An evident first property of finite volume methods in conservation form:

$$\mu_j(0) = \mu_0 \in \mathbb{R} \quad \forall j \in \mathbb{Z} \quad \Rightarrow \quad \mu_j(t) = \mu_0 \quad \forall j \in \mathbb{Z}, \quad \forall t > 0 . \quad (8.3.10)$$

that is, constant solutions are preserved by the method.

A “telescopic sum argument” combined with the interpretation (8.3.5) shows that the conservation form (8.3.8) of the semi-discrete conservation law involves

$$\frac{d}{dt} \int_{x_{k-1/2}}^{x_{m+1/2}} u_N(x, t) dx = h \sum_{l=k}^m \frac{d\mu_j}{dt}(t) = - (f_{m+1/2}(t) - f_{k-1/2}(t)) \quad \forall k, m \in \mathbb{Z} .$$

↔

$$\frac{d}{dt} \int_{x_{k-1/2}}^{x_{m+1/2}} u(x, t) dx = - (f(u(x_{j+1/2}, t)) - f(u(x_{k-1/2}, t))) ,$$

- With respect to unions of dual cells and numerical fluxes, the semidiscrete solution $u_N(t)$ satisfies a balance law of the same structure as a (weak) solution of (8.2.7).

Of course, the numerical flux function F has to fit the flux function f of the conservation law:

Definition 8.3.11 (Consistent numerical flux function).

A numerical flux function $F : \mathbb{R}^{m_l+m_r} \mapsto \mathbb{R}$ is **consistent** with the flux function $f : \mathbb{R} \mapsto \mathbb{R}$, if

$$F(u, \dots, u) = f(u) \quad \forall u \in \mathbb{R}.$$

Focus: solution of Riemann problem (\rightarrow Def. 8.2.23) by finite volume method in conservation form (8.3.8):

Initial data “constant at $\pm\infty$ ”: $\mu_{-j}(0) = u_l$, $\mu_j(0) = u_r$ for large j .

8.3

Consistency of the numerical flux function implies for large $m \gg 1$

$$\frac{d}{dt} \int_{-x_{-m-1/2}}^{x_{m+1/2}} u_N(x, t) dt = -\left(F(u_r, \dots, u_r) - F(u_l, \dots, u_l) \right) = -(f(u_r) - f(u_l)). \quad (8.3.12)$$

Exactly the same balance law holds for any weak solutions of the Riemann problem!

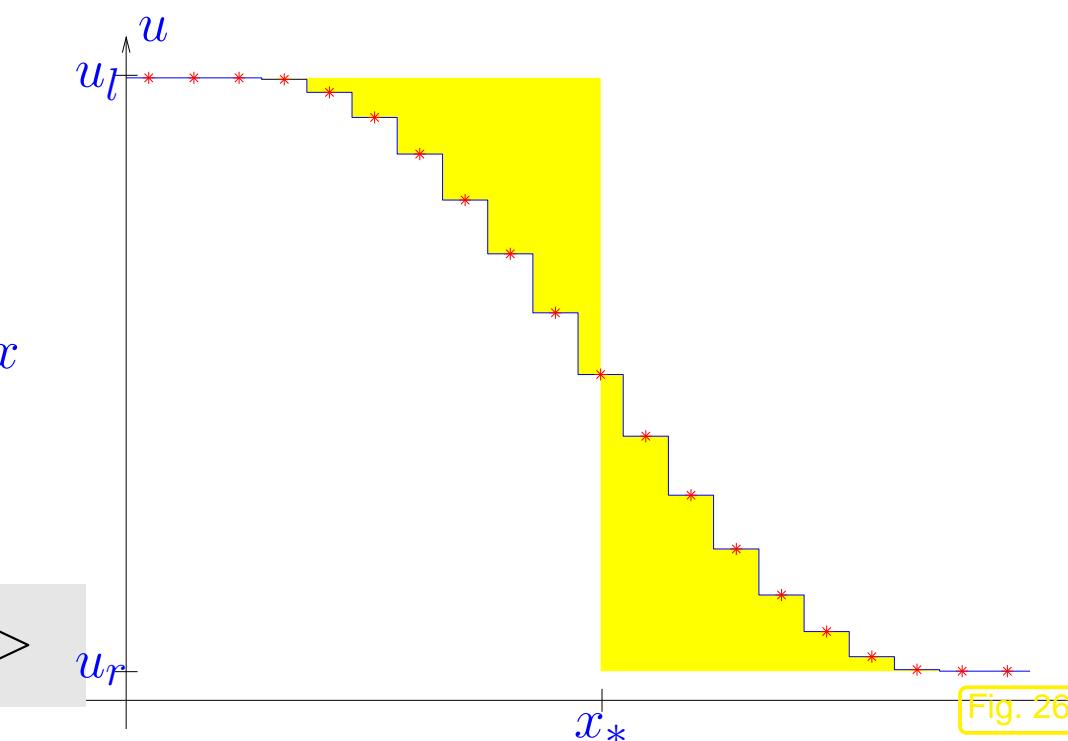
Situation: discrete solution $u_N(t)$ decreasing & supposed to approximate a shock

approximate location of shock at time t):

$$x_*(t) \in \mathbb{R}:$$

$$\int_{-\infty}^{x_*(t)} u_l - u_N(x, t) dx = \int_{x_*(t)}^{\infty} u_N(x, t) - u_r dx$$

equality of yellow areas



►
$$\int_{x_{-m-1/2}}^{x_{m+1/2}} u_N(x, t) dx = (x_*(t) + x_{m+1/2})u_l + (x_{m+1/2} - x_*(t))u_r .$$

$$(8.3.12) \quad \frac{dx_*}{dt}(t) = \frac{1}{u_l - u_r} \sum_{j \in \mathbb{Z}} \frac{d\mu_j}{dt}(t) = \frac{f(u_l) - f(u_r)}{u_l - u_r} \stackrel{(8.2.20)}{=} \dot{s} .$$

Conservation form with consistent numerical flux yields correct “discrete shock speed”
 (not liable to effect of Ex. 8.3.1)

8.3.3 Numerical flux functions

8.3.3.1 Central flux

Example 8.3.13 (Central flux for Burgers equation).

- Cauchy problem for Burgers equation (8.1.11) (flux function $f(u) = \frac{1}{2}u^2$) from Ex. 8.2.33 (“box” initial data)
- Spatial finite volume discretization in conservation form (8.3.8) with **central numerical fluxes**

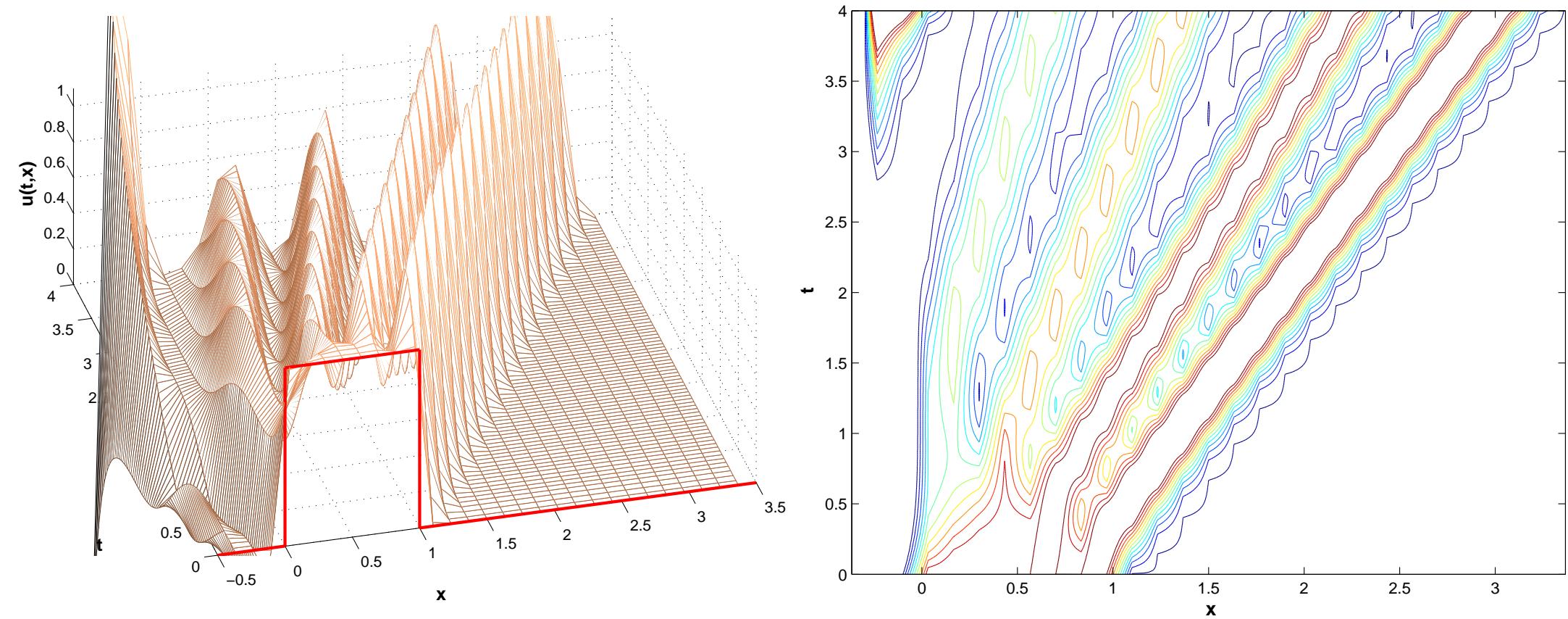
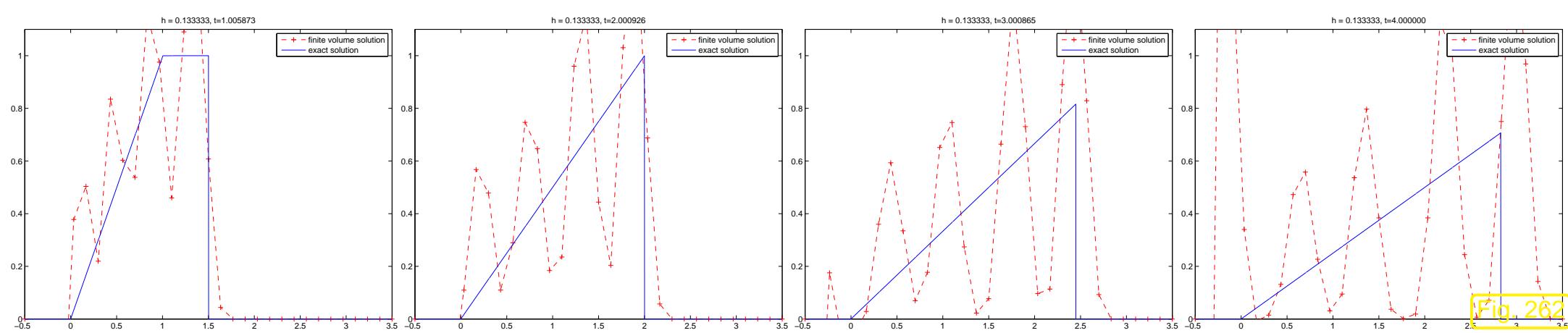
$$F_1(v, w) := \frac{1}{2}(f(v) + f(w)) , \quad F_2(v, w) := f\left(\frac{1}{2}(v + w)\right) . \quad (8.3.14)$$

Obviously the 2-point numerical fluxes F_1 and F_2 are consistent according to Def. 8.3.11. The resulting spatially semi-discrete scheme is given by, see (8.3.9)

$$\begin{aligned} F_1: \quad & \frac{d\mu_j}{dt}(t) = -\frac{1}{2h}(f(\mu_{j+1}(t)) - f(\mu_{j-1}(t))) , \\ F_2: \quad & \frac{d\mu_j}{dt}(t) = -\frac{1}{h}(f\left(\frac{1}{2}(\mu_j(t) + \mu_{j+1}(t))\right) - f\left(\frac{1}{2}(\mu_j(t) + \mu_{j-1}(t))\right)) . \end{aligned}$$

- timestepping based on adaptive Runge-Kutta method `ode45` of MATLAB
`(opts = odeset('abstol', 1E-7, 'reltol', 1E-6);).`

Fully discrete evolution for central numerical flux F_1 :



Fully discrete evolution for central numerical flux F_1 :

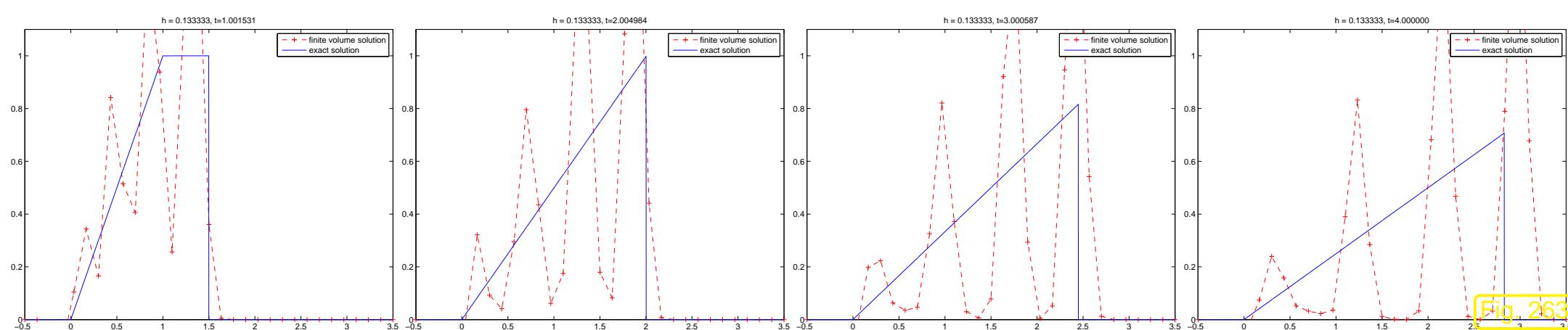
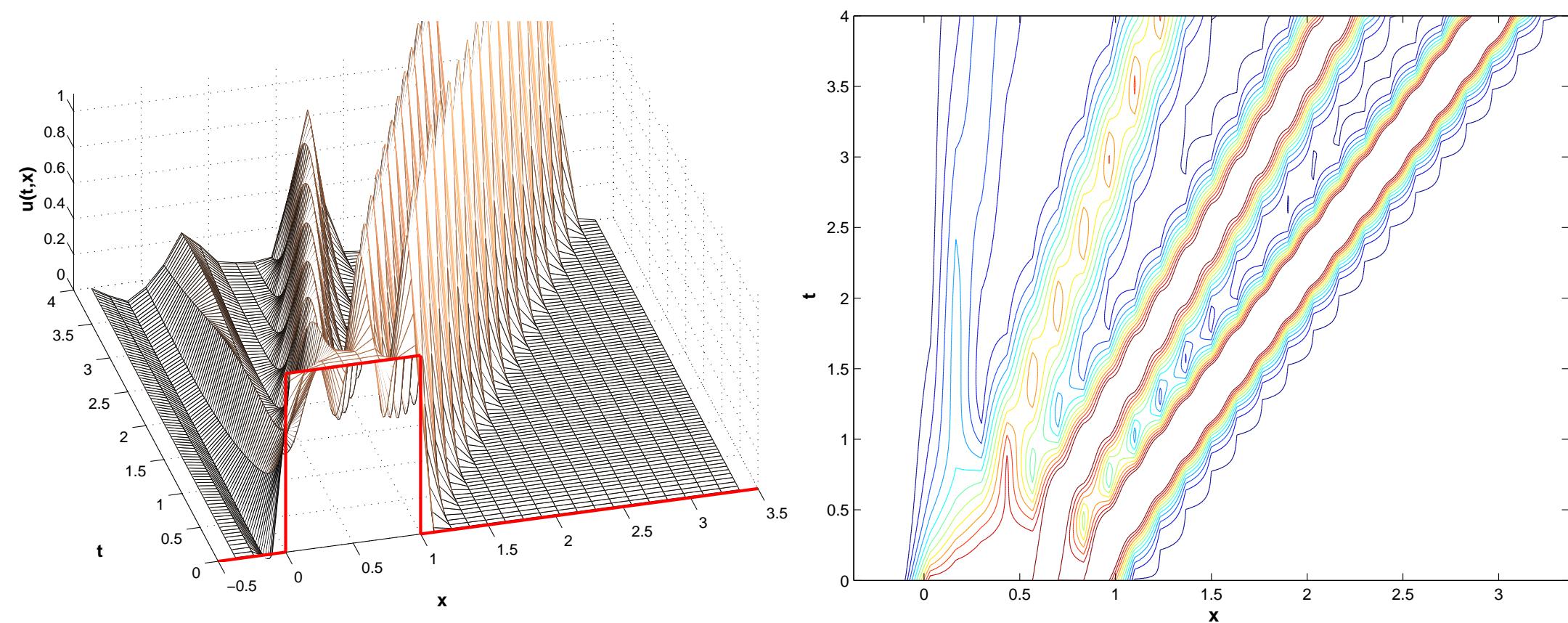


Fig. 263



Observation: massive spurious oscillations utterly pollute numerical solution



Example 8.3.15 (Central flux for linear advection).

Cauchy problem (8.1.4): constant valocity scalar linear advection, $v = 1$, flux function $f(u) = vu$

$$\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} = 0 \quad \text{in } \tilde{\Omega} = \mathbb{R} \times]0, T[, \quad u(x, 0) = u_0(x) \quad \forall x \in \mathbb{R} . \quad (8.1.4)$$

= Cauchy problem for 1D transport equation (7.3.7)!

Finite volume spatial discretization in conservation form (8.3.8) with central numerical fluxes from (8.3.14):

$$\begin{aligned} F_1(v, w) &:= \frac{1}{2}(f(v) + f(w)) & \Rightarrow \quad \frac{d\mu_j}{dt}(t) &= -\frac{v}{2h}(\mu_{j+1}(t) - \mu_{j-1}(t)) , \quad j \in \mathbb{Z} . \\ F_2(v, w) &:= f\left(\frac{1}{2}(v + w)\right) \end{aligned}$$

8.3

- = spatial semi-discretization using linear finite element Galerkin discretization of convective term, see (7.2.14).

Sect. 7.3.1: this method is *prone to spurious oscillations*, see Ex. 7.3.4.

This offers an explanation also for its failure for Burgers equation, see Ex. 8.3.13



8.3.3.2 Lax-Friedrichs flux

Sect. 7.2.2.2: **artificial diffusion** cures instability of central difference quotient

$$\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} = 0 \quad \text{in } \mathbb{R} \times]0, T[,$$

↑

$$\frac{\partial u}{\partial t} + (vh/2) \underbrace{\frac{-\mu_{j-1} + 2\mu_j - \mu_{j+1}}{h^2}}_{\hat{\triangleq} \text{ difference quotient for } \frac{d^2u}{dx^2}} + v \underbrace{\frac{\mu_{j+1} + \mu_{j-1}}{2h}}_{\hat{\triangleq} \text{ difference quotient for } \frac{du}{dx}} = 0, \quad j \in \mathbb{Z} .$$

8.3

p. 851

Can this be rewritten in conservation form (8.3.8)? YES!

$$(vh/2) \frac{-\mu_{j-1} + 2\mu_j - \mu_{j+1}}{h^2} + v \frac{\mu_{j+1} + \mu_{j-1}}{2h} = \frac{1}{h} (F(\mu_j, \mu_{j+1}) - F(\mu_{j-1}, \mu_j)) ,$$

with $F(v, w) := \frac{v}{2}(v + w) - \frac{v}{2}(w - v)$. (8.3.16)

central numerical flux diffusive/viscous numerical flux

Recall from Rem. 8.2.2: the flux function $f(u) = -\frac{\partial u}{\partial x}$ models diffusion. Hence, the diffusive numerical flux amounts to a central finite difference discretization of the partial derivative in space:

$$-\frac{\partial u}{\partial x}(x, t) \Big|_{x=x_{j+1/2}} \approx -\frac{1}{h} (u(x_{j+1}, t) - u(x_j, t)) .$$

How to adapt this to general scalar conservation laws?

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = \frac{\partial u}{\partial t} + f'(u) \frac{\partial u}{\partial x} = 0 (8.3.17)$$

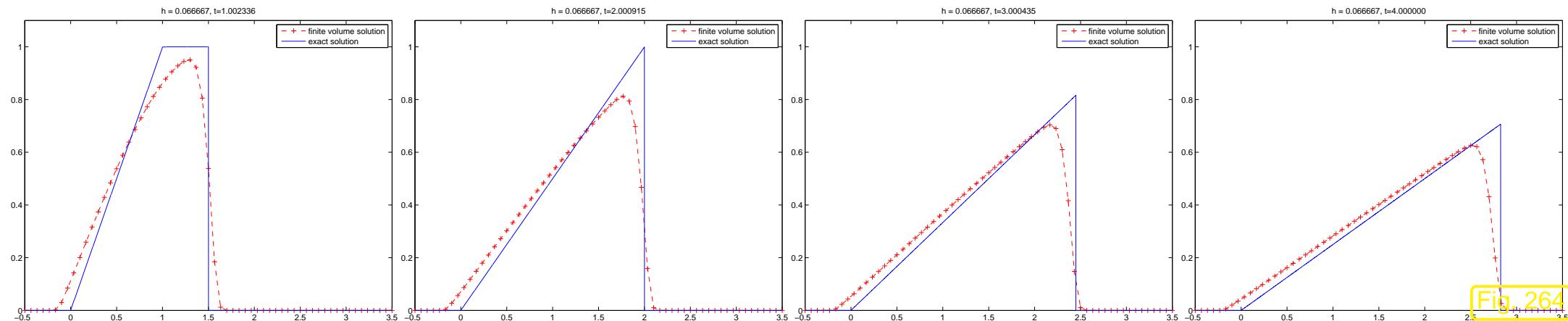
local speed of transport

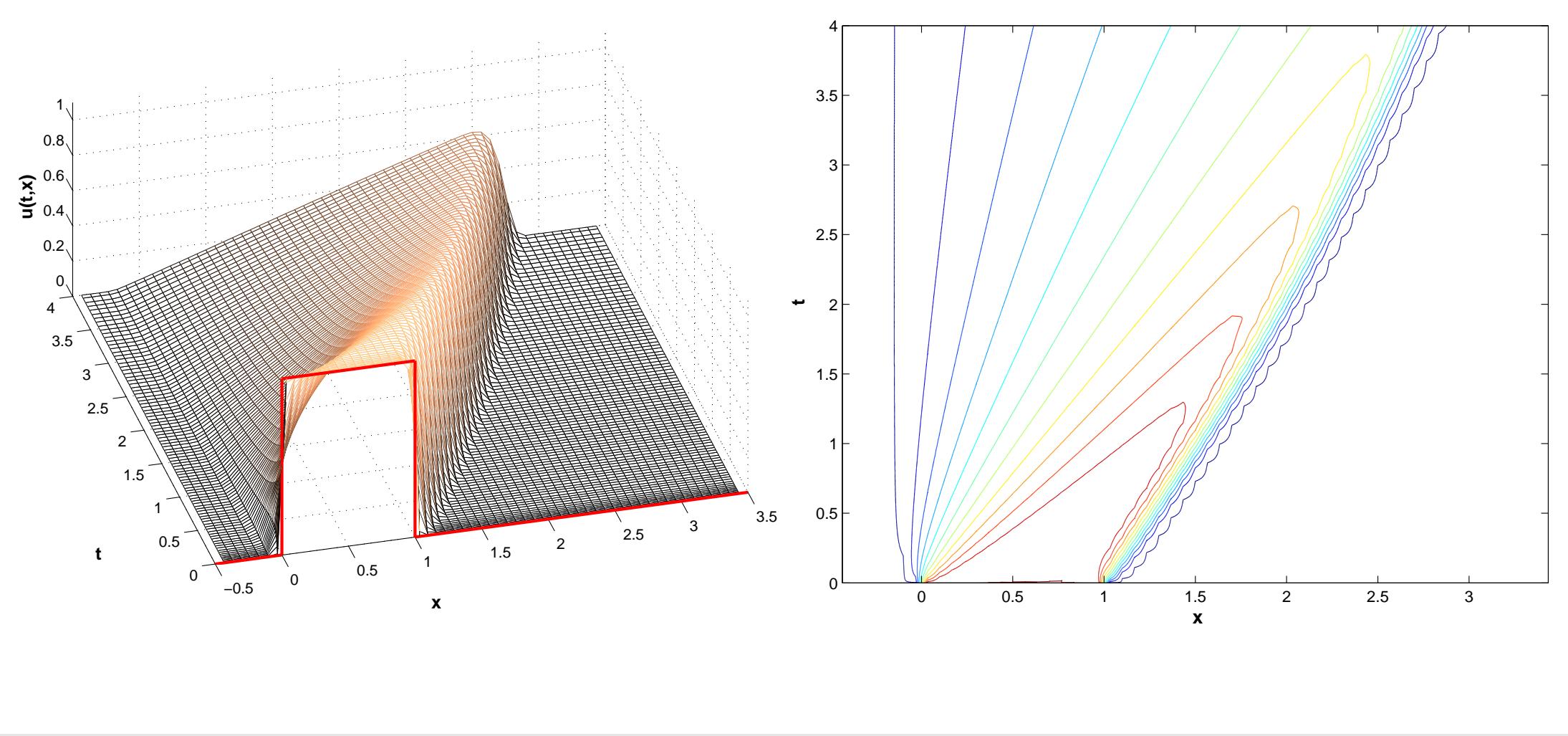
(local) Lax-Friedrichs flux

$$F_{\text{LF}}(v, w) = \frac{1}{2}(f(v) + f(w)) - \frac{1}{2} \max_{\min\{v,w\} \leq u \leq \max\{v,w\}} |f'(u)|(w - v). \quad (8.3.18)$$

Example 8.3.19 (Lax-Friedrichs flux for Burgers equation).

- same setting and conservative discretization as in Ex. 8.3.13
- Numerical flux function: Lax-Friedrichs flux (8.3.18)





Observation: spurious completely suppressed, qualitatively good resolution of both shock and rarefaction.

Effect of artificial diffusion: smearing of shock, cf. discussion in Ex. 7.2.26.

8.3.3.3 Upwind flux

Another idea for stable spatial discretization of stationary transport in Sect. 7.2.2.1:

“**upwinding**” = obtain information from where transport brings it

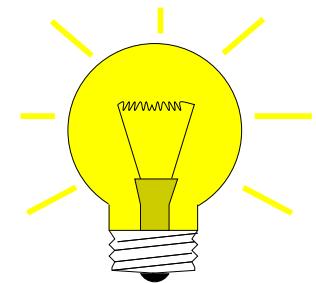
☞ remedy for ambiguity of evaluation of discontinuous gradient in upwind quadrature

Ambiguity also faced in the evaluation of $f(u_N(x_{j+1/2}), t), f(u_N(x_{j-1/2}), t)$, see (8.3.7), which forced us to introduce numerical flux functions in (8.3.8).

(8.3.17): local velocity of transport at $(x, t) \in \tilde{\Omega}$ is given by $f'(u(x, t))$

➤ ambiguous local velocity of transport at discontinuity of u_N !

Idea: deduce local velocity of transport from Rankine-Hugoniot jump condition
(8.2.20)



local velocity of transport = $\begin{cases} f'(u) & \text{for unique state, } u = u_l = u_r \\ \frac{f(u_r) - f(u_l)}{u_r - u_l} & \text{at discontinuity.} \end{cases}$

$(u_l, u_r \hat{=} \text{states to left and right of discontinuity})$

► upwind flux for scalar conservation law with flux function f :

$$F_{uw}(v, w) = \begin{cases} f(v) & , \text{if } \dot{s} > 0 , \\ f(w) & , \text{if } \dot{s} < 0 , \end{cases} \quad \dot{s} := \frac{f(w) - f(v)}{w - v} .$$

Example 8.3.20 (Upwind flux for Burgers equation).

- same setting and conservative discretization as in Ex. 8.3.13
- Numerical flux function: upwind flux (8.3.3.3)

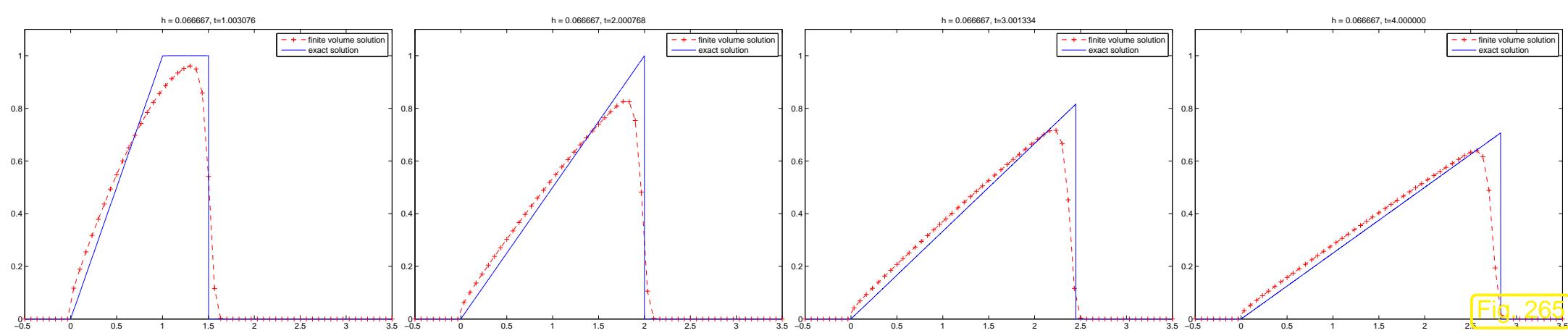
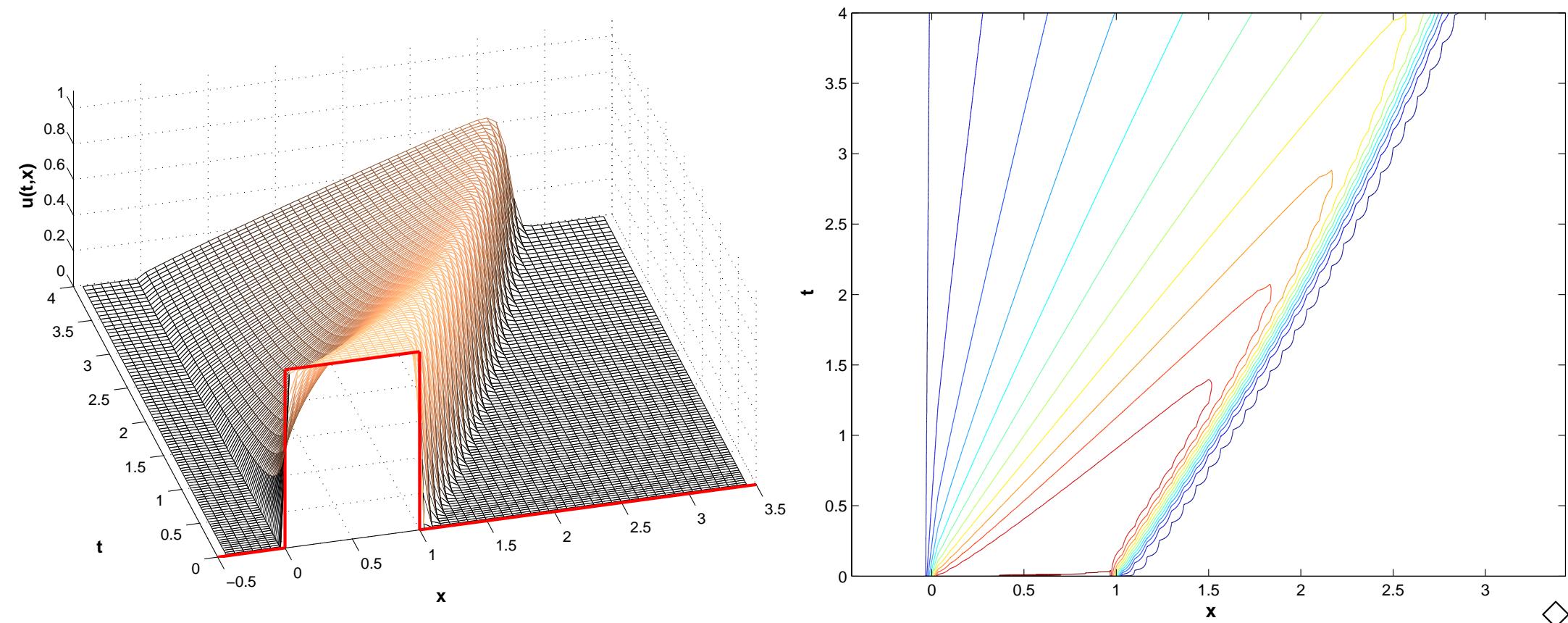


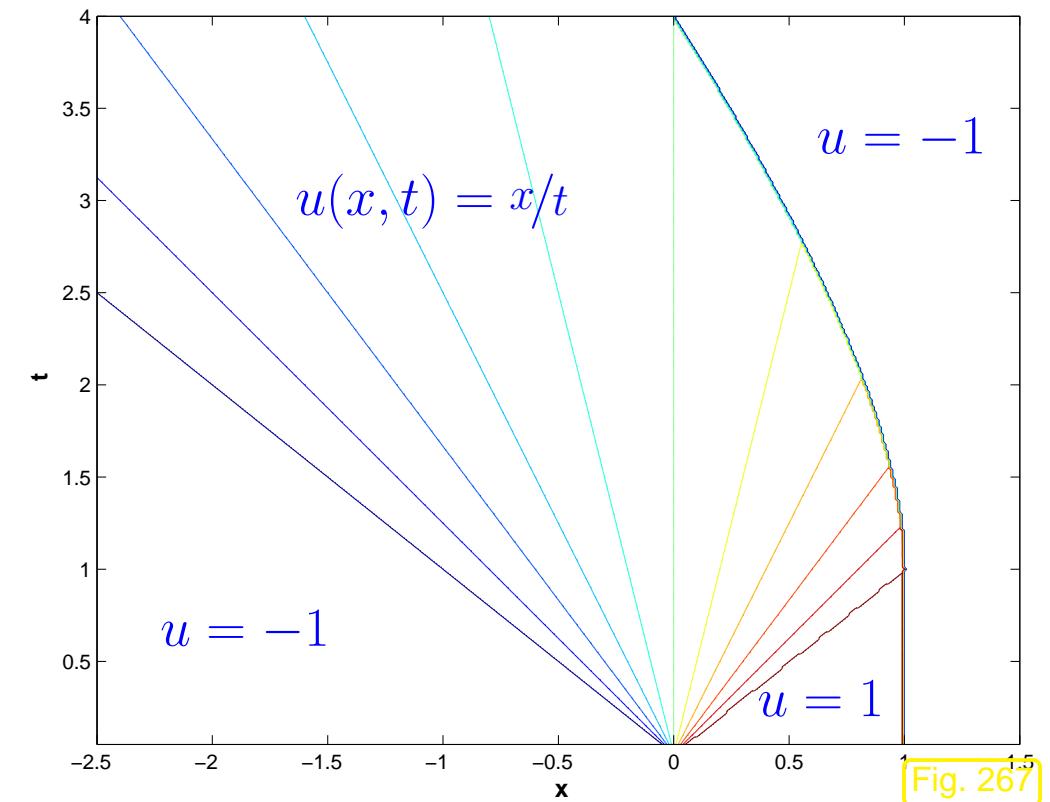
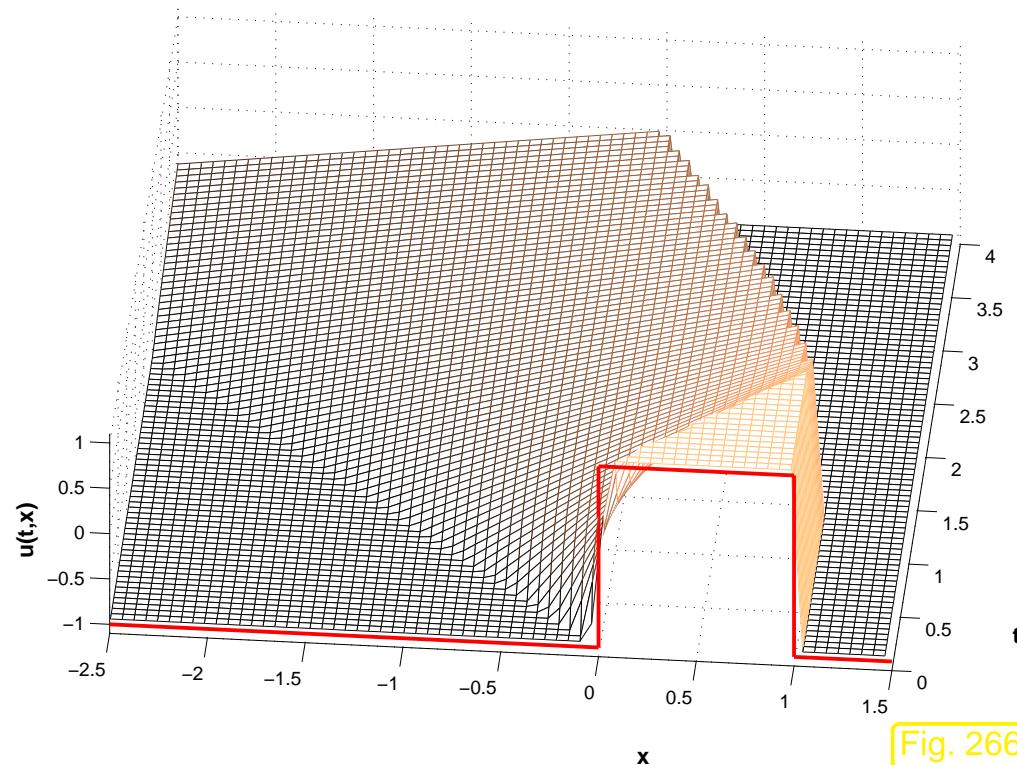
Fig. 265



Example 8.3.21 (Upwind flux and transsonic rarefaction).

Cauchy problem (8.2.7) for Burgers equation (8.1.11), i.e., $f(u) = \frac{1}{2}u^2$ and initial data

$$u_0(x) = \begin{cases} -1 & \text{for } x < 0 \text{ or } x > 1, \\ 1 & \text{for } 0 < x < 1. \end{cases}$$



The *entropy solution* (\rightarrow Sect. 8.2.6) of this Cauchy problem features a **transsonic rarefaction fan** at $x = 0$: this is a rarefaction solution (\rightarrow Lemma 8.2.29) whose “edges” move in opposite directions.

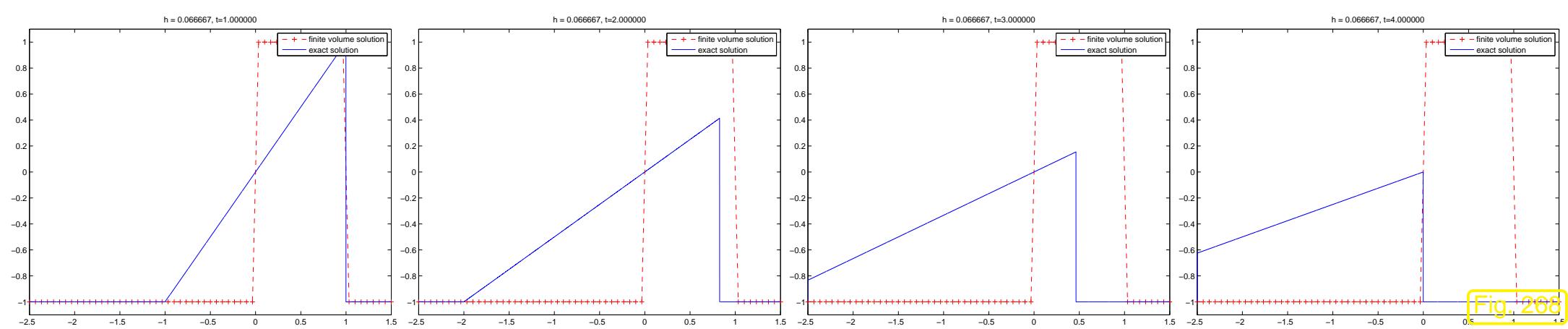
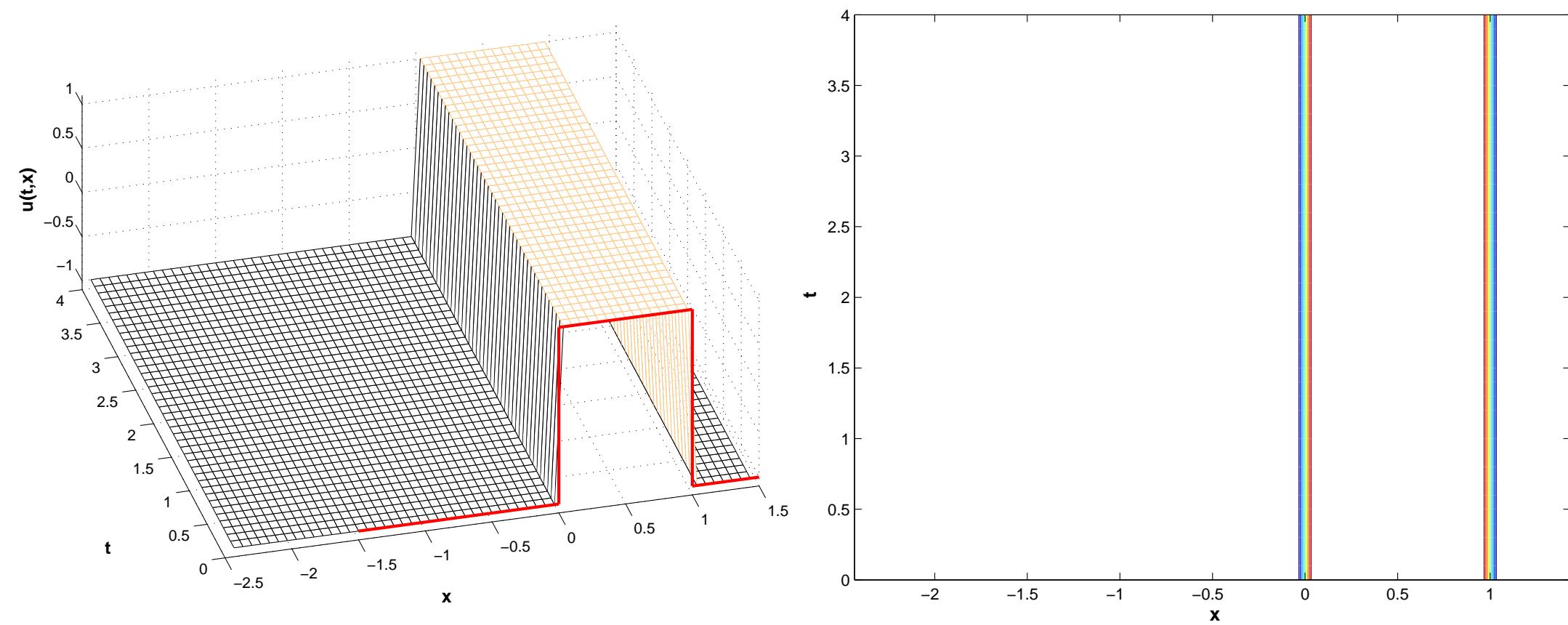


Fig. 268



Conservative finite volume discretization with upwind flux produces (stationary) *expansion shock* instead of transonic rarefaction!

Sect. 8.2.6: this is a weak solution, but it violates the entropy condition, “non-physical shock”.



Example 8.3.22 (Upwind flux: Convergence to expansion shock).

- Cauchy problem (8.2.7) for Burgers equation (8.1.11), i.e., $f(u) = \frac{1}{2}u^2$
- $u_0(x) = 1$ for $x > 0$, $u_0(x) = -1$ for $x < 0$
 - entropy solution = rarefaction wave (\rightarrow Lemma 8.2.29)
- FV in conservation form, upwind flux (8.3.3.3), on equidistant grid, $x_j = (j + \frac{1}{2})h$, meshwidth $h > 0$

- initial nodal values $\mu_j(0) = \begin{cases} -1 & \text{for } j < 0 , \\ 1 & \text{for } j \geq 0 . \end{cases}$
- Semi-discrete evolution equation:
$$\frac{d\mu_j}{dt}(t) = -\frac{1}{h} \cdot \begin{cases} \mu_{j+1}^2(t) - \mu_j^2(t) & \text{for } j \geq 0 , \\ \mu_j^2(t) - \mu_{j-1}^2(t) & \text{for } j < 0 . \end{cases}$$
- $\mu_j(t) = \mu_j(0)$ for all t ➤ for $h \rightarrow 0$, convergence to entropy violating expansion shock !
- finite volume method may converge to non-physical weak solutions !

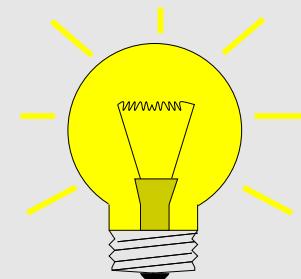
◇

8.3.3.4 Godunov flux

The upwind flux (8.3.3.3) is a numerical flux of the form

$$F(v, w) = f(u^\downarrow(v, w)) \quad \text{with an intermediate state } u^\downarrow(v, w) \in \mathbb{R}.$$

For the upwind flux the intermediate state is not really “intermediate”, but coincides with one of the states v, w depending on the sign of the “local shock speed” $\dot{s} := \frac{f(w) - f(v)}{w - v}$.



Idea: obtain suitable intermediate state as

$$u^\downarrow(v, w) = \psi(0), \quad (8.3.23)$$

where $u(x, t) = \psi(x/t)$ solves the Riemann problem (\rightarrow Def. 8.2.23)

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0, \quad u(x, 0) = \begin{cases} v & , \text{for } x < 0, \\ w & , \text{for } x \geq 0. \end{cases} \quad (8.3.24)$$

We focus on $f : \mathbb{R} \mapsto \mathbb{R}$ strictly convex & smooth (e.g. Burgers equations (8.1.11))

- Riemann problem (8.3.24) (\rightarrow Def. 8.2.23) has the *entropy solution* (\rightarrow Sect. 8.2.6):
- ① If $v > w$ ➤ discontinuous solution, **shock** (\rightarrow Lemma 8.2.26)

$$u(t, x) = \begin{cases} v & \text{if } x < \dot{s}t, \\ w & \text{if } x > \dot{s}t, \end{cases} \quad \dot{s} = \frac{f(v) - f(w)}{v - w}. \quad (8.3.25)$$

② If $v \leq w$ ➤ continuous solution, rarefaction wave (\rightarrow Lemma 8.2.29)

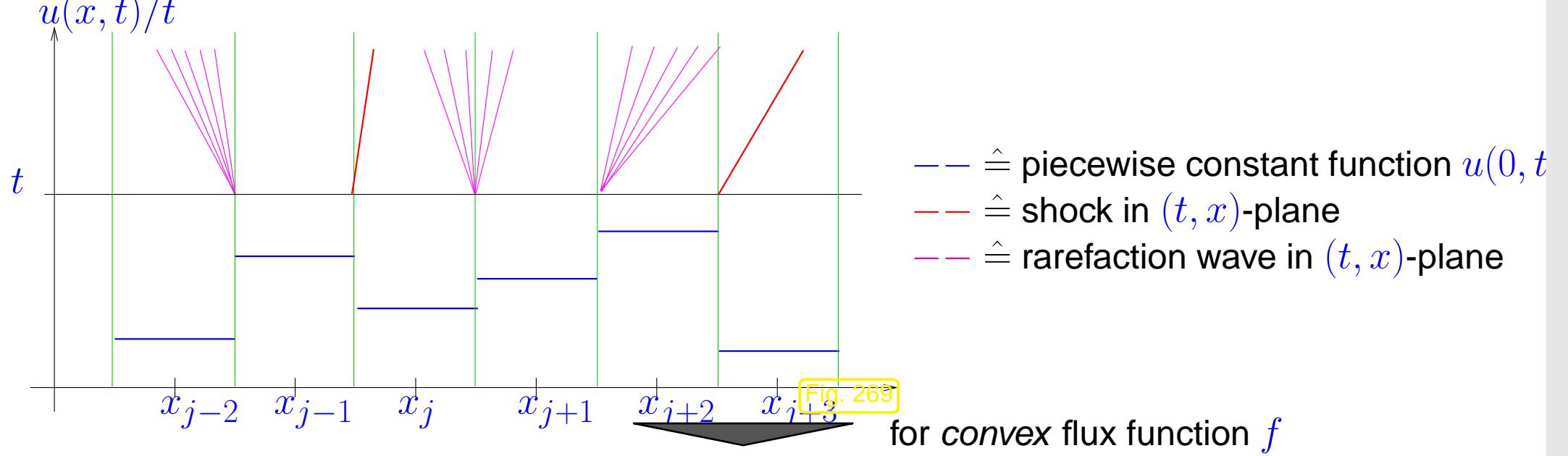
$$u(t, x) = \begin{cases} v & \text{if } x < f'(v)t, \\ g(x/t) & \text{if } f'(v) \leq x/t \leq f'(w), \\ w & \text{if } x > f'(w)t, \end{cases} \quad g := (f')^{-1}. \quad (8.3.26)$$

➤ All weak solutions of a Riemann problem are of the form $u(x, t) = \psi(x/t)$ with a suitable function ψ , which is

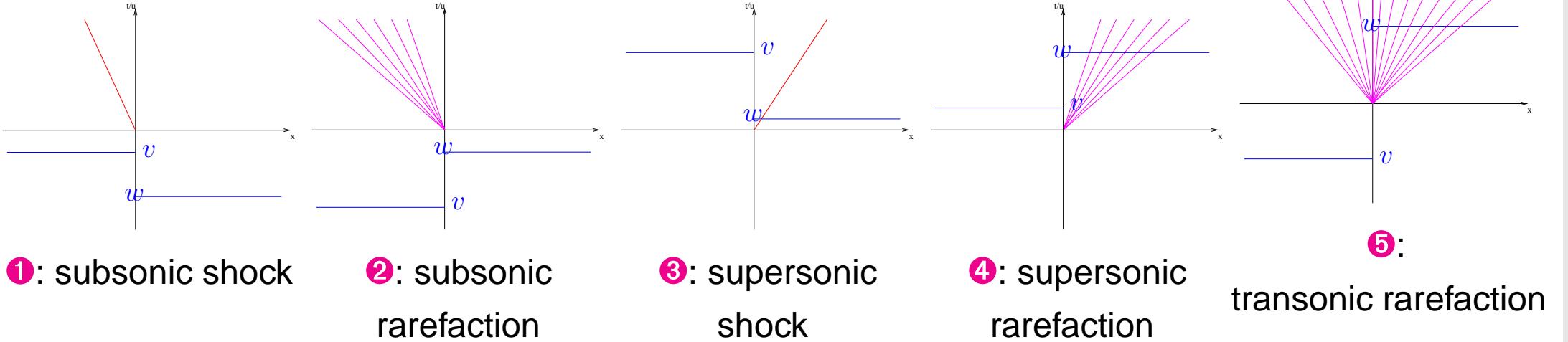
- piecewise constant with a jump at $\dot{s} := \frac{f(w)-f(v)}{w-v}$ for a shock solution (8.3.25),
- the continuous function (in the case of strictly convex flux function f)

$$\psi(\xi) := \begin{cases} v & , \text{ if } \xi < f'(v) , \\ (f')^{-1}(\xi) & , \text{ if } f'(v) < \xi < f'(w) , \\ w & , \text{ if } \xi > f'(w) , \end{cases}$$

provided that $w > v$ = situation of a rarefaction solution (8.3.26), see Lemma 8.2.29.



$$u^\downarrow(v, w) = \begin{cases} w & , \text{if } v > w \wedge \dot{s} < 0 \text{ (shock ①)} , \\ v & , \text{if } v < w \wedge f'(w) < 0 \text{ (rarefaction ②)} , \\ (f')^{-1}(0) & , \text{if } v < w \wedge \dot{s} > 0 \text{ (shock ③)}, \\ & v < w \wedge f'(v) > 0 \text{ (rarefaction ④)} , \\ & v < w \wedge f'(v) \leq 0 \leq f'(w) \text{ (rarefaction ⑤)}. \end{cases} \quad (8.3.27)$$



Detailed analysis of (8.3.27):

$$v > w \quad (\text{shock case}): \quad f(u^\downarrow(v, w)) = \begin{cases} f(v) & , \text{ if } \frac{f(w) - f(v)}{w - v} > 0 \Leftrightarrow f(w) < f(v) , \\ f(w) & , \text{ if } \frac{f(w) - f(v)}{w - v} \leq 0 \Leftrightarrow f(w) \geq f(v) . \end{cases}$$

► $f(u^\downarrow(v, w)) = \max\{f(v), f(w)\} .$

For a convex flux function f :

$$v < w \Rightarrow f'(v) \leq \frac{f(w) - f(v)}{w - v} \leq f'(w).$$

► For $v < w$ (rarefaction case)

$$f(u^\downarrow(v, w)) = \begin{cases} f(v) & , \text{if } f'(v) > 0 , \\ f(z) & , \text{if } f'(v) < 0 < f'(w) , \\ f(w) & , \text{if } f'(w) < 0 , \end{cases}$$

where $f'(z) = 0 \Leftrightarrow f$ has a global minimum in z .

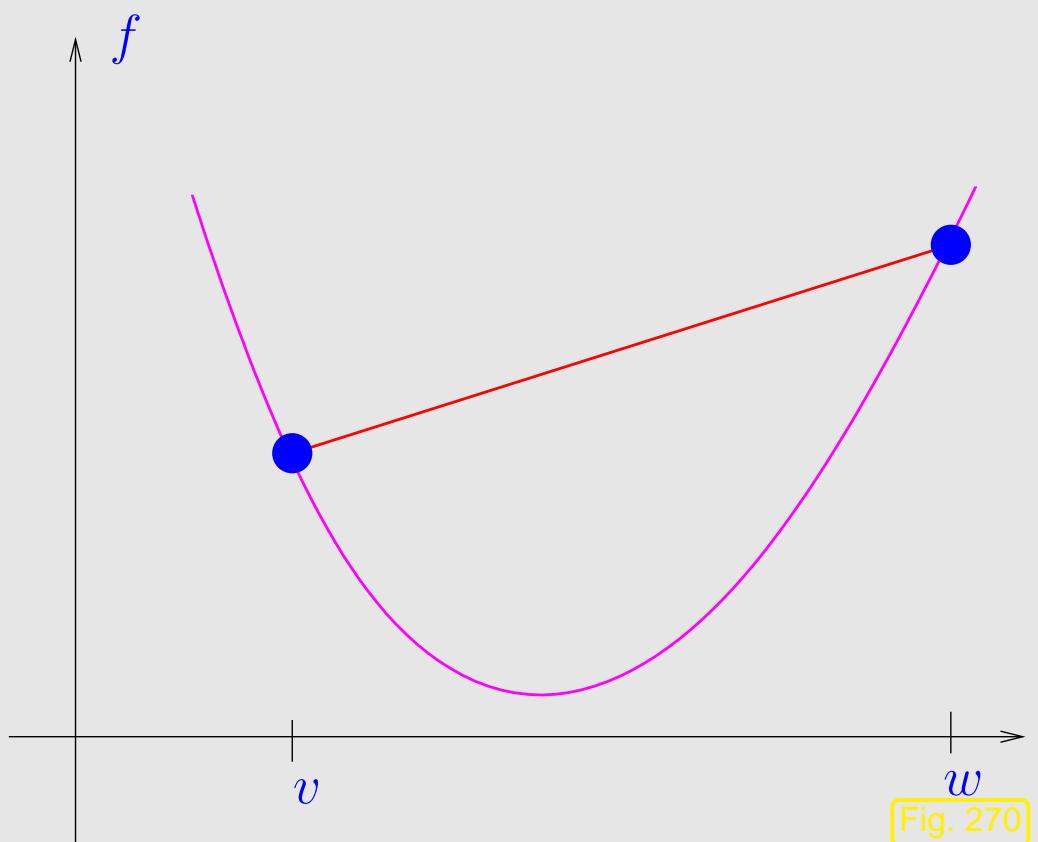


Fig. 270

2-point numerical flux function according to (8.3.23) and (8.3.24): **Godunov numerical flux**

Using general Riemann solution (8.2.32): for any flux function

► Godunov numerical flux function

$$F_{\text{GD}}(v, w) = \begin{cases} \min_{v \leq u \leq w} f(u) & , \text{ if } v < w , \\ \max_{w \leq u \leq v} f(u) & , \text{ if } w \leq v . \end{cases} \quad (8.3.28)$$

for Burgers equation (8.1.11)

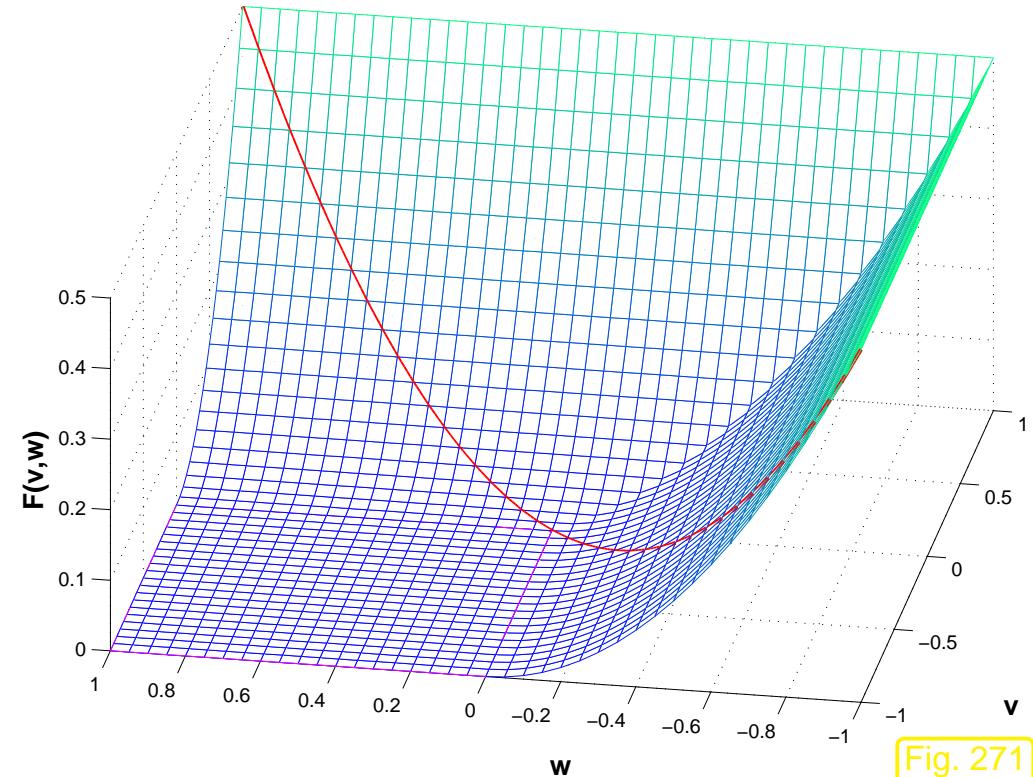


Fig. 271

Remark 8.3.29 (Upwind flux and expansion shocks).

$F_{uw}(v, w) = F_{\text{GD}}(v, w)$, except for the case of *transsonic rarefaction*!

(transsonic rarefaction = rarefaction fan with edges moving in opposite direction, see Ex. 8.3.21)

What does the upwind flux $F_{\text{uw}}(v, w)$ from (8.3.3.3) yield in the case of transsonic rarefaction?

If f convex, $v < w$, $f'(v) < 0 < f'(w)$,

$$\blacktriangleright \quad F_{\text{uw}}(v, w) = f(\psi(0)) ,$$

where $u(x, t) = \psi(x/t)$ is a non-physical *entropy-condition violating* (\rightarrow Def. 8.2.30) expansion shock weak solution of (8.3.24).

Upwind flux treats transsonic rarefaction as expansion shock!

➤ Explanation for observation made in Ex. 8.3.21.



Example 8.3.30 (Godunov flux for Burgers equation).

- ☞ same setting and conservative discretization as in Ex. 8.3.21
- ☞ Numerical flux function: Godunov numerical flux (8.3.28)

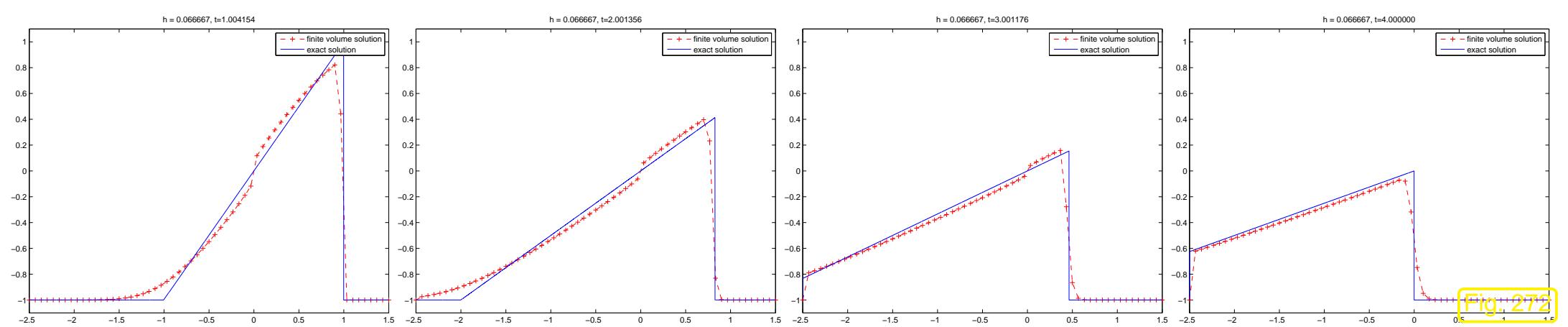
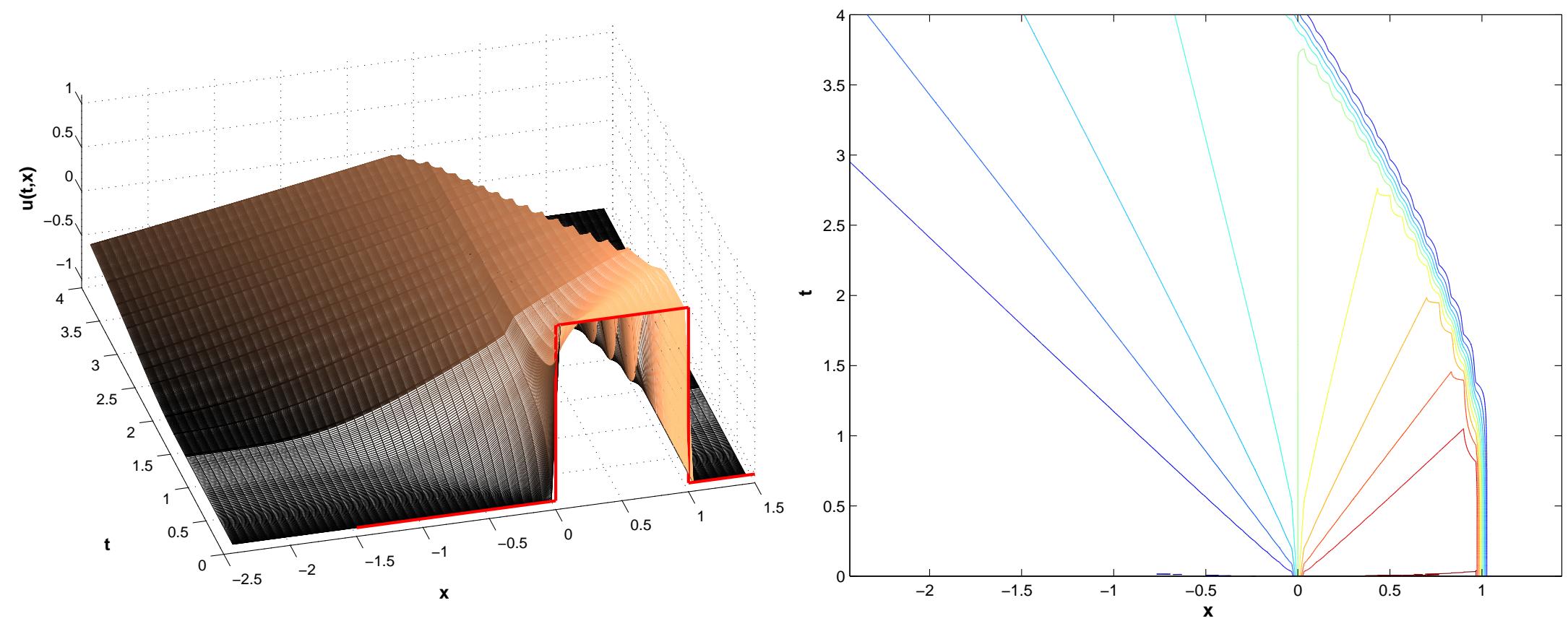


Fig. 272



Observation: Transonic rarefaction captured by discretization, but small remnants of an expansion shock still observed.



8.3.4 Montone schemes

Observations made for some piecewise constant solutions $u_N(t)$ of semi-discrete evolutions arising from spatial finite volume discretization in conservation form (8.3.9):

Ex. 8.3.19 (Lax-Friedrichs numerical flux (8.3.18))

$$\min_{x \in \mathbb{R}} u_0(x) \leq u_N(x, t) \leq \max_{x \in \mathbb{R}} u_0(x)$$

Ex. 8.3.30 (Godunov numerical flux (8.3.28))

: **no new** local extrema in numerical solution

In these respects the conservative finite volume discretizations based on either the Lax-Friedrichs numerical flux or the Godunov numerical flux inherit crucial structural properties of the exact solution,

see Sect. 8.2.7, in particular, Thm. 8.2.34 and the final remark: they display **structure preservation**, cf. (5.7).

Is this coincidence for the special settings examined in Ex. 8.3.19 and Ex. 8.3.30?

Focus: semi-discrete evolution (8.3.9) resulting from finite volume discretization in conservation form on an equidistant infinite mesh

$$(8.3.8) \quad \blacktriangleright \quad \frac{d\mu_j}{dt}(t) = -\frac{1}{h} (F(\mu_j(t), \mu_{j+1}(t)) - F(\mu_{j-1}(t), \mu_j(t))) , \quad j \in \mathbb{Z} , \quad (8.3.9)$$

for Cauchy problem

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \text{in } \mathbb{R} \times]0, T[, \quad u(x, 0) = u_0(x) , \quad x \in \mathbb{R} , \quad (8.2.7)$$

induced by Lax-Friedrichs numerical flux (8.3.18)

$$F_{\text{LF}}(v, w) = \frac{1}{2}(f(v) + f(w)) - \frac{1}{2} \max_{\min\{v, w\} \leq u \leq \max\{v, w\}} |f'(u)|(w - v) . \quad (8.3.18)$$

►
$$\frac{d\mu_j}{dt} = -\frac{1}{2h} \left(f(\mu_{j+1}) - f(\mu_{j-1}) - \max_{u \in [\mu_j, \mu_{j+1}]} |f'(u)|(\mu_{j+1} - \mu_j) + \max_{u \in [\mu_{j-1}, \mu_j]} |f'(u)|(\mu_j - \mu_{j-1}) \right). \quad (8.3.31)$$

Goal: show that $\vec{u}_N(t)$ linked to $\vec{\mu}(t)$ from (8.3.31) through piecewise constant reconstruction (8.3.5) satisfies

$$\min_{x \in \mathbb{R}} u_N(x, 0) \leq u_N(x, t) \leq \max_{x \in \mathbb{R}} u_N(x, 0) \quad \forall x \in \mathbb{R}, \quad \forall t \in [0, T]. \quad (8.3.32)$$

Recall from Sect. 8.2.7: estimate (8.3.32) for the exact solution $u(x, t)$ of (8.2.7) is a consequence of the comparison principle of Thm. 8.2.34 and the fact that constant initial data are preserved during the evolution. The latter property is straightforward for conservative finite volume spatial semi-discretization, see (8.3.10).

➤ Goal: Establish comparison principle for finite volume semi-discrete solutions based on Lax-Friedrichs numerical flux:

$$\left\{ \begin{array}{l} \vec{\mu}(t), \vec{\eta}(t) \text{ solve (8.3.31)} , \\ \eta_j(0) \leq \mu_j(0) \quad \forall j \in \mathbb{Z} \end{array} \right\} \Rightarrow \eta_j(t) \leq \mu_j(t) \quad \forall j \in \mathbb{Z}, \quad \forall 0 \leq t \leq T .$$

Assumption: $\vec{\mu} = \vec{\mu}(t)$ and $\vec{\eta} = \vec{\eta}(t)$ solve (8.3.31) and satisfy for some $t \in [0, T]$

$$\eta_k(t) \leq \mu_k(t) \quad \forall k \in \mathbb{Z} , \quad \xi := \eta_j(t) = \mu_j(t) \quad \text{for some } j \in \mathbb{Z} .$$

Can η_j raise above μ_j ?

$$\frac{d}{dt}(\mu_j - \eta_j) = -\frac{1}{h} \left(F_{\text{LF}}(\xi, \mu_{j+1}) - F_{\text{LF}}(\xi, \eta_{j+1}) + F_{\text{LF}}(\eta_{j-1}, \xi) - F_{\text{LF}}(\mu_{j-1}, \xi) \right) .$$

To show: $\frac{d}{dt}(\mu_j - \eta_j) \geq 0 \Rightarrow \mu_j(t) \text{ will stay above } \eta_j(t)$.

This can be concluded, if

$$F_{\text{LF}}(\xi, \mu_{j+1}) - F_{\text{LF}}(\xi, \eta_{j+1}) \leq 0 \quad \text{and} \quad F_{\text{LF}}(\eta_{j-1}, \xi) - F_{\text{LF}}(\mu_{j-1}, \xi) \leq 0 . \quad (8.3.33)$$

The only piece of information we are allowed to use is

$$\mu_{j+1} \geq \eta_{j+1} \quad \text{and} \quad \mu_{j-1} \geq \eta_{j-1} .$$

This would imply (8.3.33), if F_{LF} was increasing in the first argument and decreasing in the second argument.

Definition 8.3.34 (Monotone numerical flux function).

A 2-point numerical flux function $F = F(v, w)$ is called **monotone**, if

F is an **increasing** function of its **first** argument

and

F is a **decreasing** function of its **second** argument.

Simple criterion: A continuously differentiable 2-point numerical flux function $F = F(v, w)$ is monotone, if and only if

$$\frac{\partial F}{\partial v}(v, w) \geq 0 \quad \text{and} \quad \frac{\partial F}{\partial w}(v, w) \leq 0 \quad \forall (v, w). \quad (8.3.35)$$

Lemma 8.3.36 (Monotonicity of Lax-Friedrichs numerical flux and Godunov flux).

For any continuously differentiable flux function f the associated Lax-Friedrichs flux (8.3.18) and Godunov flux (8.3.28) are monotone.

Proof.

① Lax-Friedrichs numerical flux:

$$F_{\text{LF}}(v, w) = \frac{1}{2}(f(v) + f(w)) - \frac{1}{2} \max_{\min\{v,w\} \leq u \leq \max\{v,w\}} |f'(u)|(w - v) . \quad (8.3.18)$$

Application of the criterion (8.3.35) is straightforward:

$$\begin{aligned} \frac{\partial F_{\text{LF}}}{\partial v}(v, w) &= f'(v) + \max_{\min\{v,w\} \leq u \leq \max\{v,w\}} |f'(u)| \geq 0 , \\ \frac{\partial F_{\text{LF}}}{\partial w}(v, w) &= f'(w) - \max_{\min\{v,w\} \leq u \leq \max\{v,w\}} |f'(u)| \leq 0 . \end{aligned}$$

② Godunov numerical flux

$$F_{\text{GD}}(v, w) = \begin{cases} \min_{v \leq u \leq w} f(u) & , \text{if } v < w , \\ \max_{w \leq u \leq v} f(u) & , \text{if } w \leq v . \end{cases} \quad (8.3.28)$$

$v < w$: If v increases, then the range of values over which the minimum is taken will shrink, which makes $F_{\text{GD}}(v, w)$ increase.

If w is raised, then the minimum is taken over a larger interval, which causes $F_{\text{GD}}(v, w)$ to become smaller.

$v \geq w$: If v increases, then the range of values over which the maximum is taken will grow, which makes $F_{\text{GD}}(v, w)$ increase.

If w is raised, then the maximum is taken over a smaller interval, which causes $F_{\text{GD}}(v, w)$ to decrease. \square

Lemma 8.3.37 (Comparison principle for monotone semi-discrete conservative evolutions).

Let the 2-point numerical flux function $F = F(v, w)$ be monotone (\rightarrow Def. 8.3.34) and $\vec{\mu} = \vec{\mu}(t)$, $\vec{\eta} = \vec{\eta}(t)$ solve (8.3.9). Then

$$\eta_k(0) \leq \mu_k(0) \quad \forall k \in \mathbb{Z} \quad \Rightarrow \quad \eta_k(t) \leq \mu_k(t) \quad \forall k \in \mathbb{Z}, \quad \forall 0 \leq t \leq T.$$

The assertion of Lemma 8.3.37 means that for monotone numerical flux, the semi-discrete evolution satisfies the **comparison principle** of Thm. 8.2.34.

Proof (of Lemma 8.3.37, following the above considerations for the Lax-Friedrichs flux).

The two sequences of nodal values satisfy (8.3.9)

$$\frac{d\mu_j}{dt}(t) = -\frac{1}{h} (F(\mu_j(t), \mu_{j+1}(t)) - F(\mu_{j-1}(t), \mu_j(t))) , \quad j \in \mathbb{Z} , \quad (8.3.38)$$

$$\frac{d\eta_j}{dt}(t) = -\frac{1}{h} (F(\eta_j(t), \eta_{j+1}(t)) - F(\eta_{j-1}(t), \eta_j(t))) , \quad j \in \mathbb{Z} . \quad (8.3.39)$$

Let t_0 be the *earliest* time, at which $\vec{\eta}$ “catches up” with $\vec{\mu}$ in at least one node x_j , $j \in \mathbb{Z}$, of the mesh, that is

$$\eta_k(t_0) \leq \mu_k(t_0) \quad \forall k \in \mathbb{Z} , \quad \xi := \eta_j(t_0) = \mu_j(t_0) .$$

By subtracting (8.3.38) and (8.3.39) we get

$$\frac{d}{dt}(\mu_j - \eta_j)(t_0) = -\frac{1}{h} (F(\xi, \mu_{j+1}(t_0)) - F(\xi, \eta_{j+1}(t_0)) + F(\eta_{j-1}(t_0), \xi) - F(\mu_{j-1}(t_0), \xi)) \geq 0 ,$$

because for a *monotone* numerical flux function (\rightarrow Def. 8.3.34)

$$\begin{array}{lll} \eta_{j-1}(t_0) \leq \mu_{j-1}(t_0) & \text{increasing in first argument} & F(\eta_{j-1}(t_0), \xi) - F(\mu_{j-1}(t_0), \xi) \leq 0 , \\ \eta_{j+1}(t_0) \leq \mu_{j+1}(t_0) & \text{decreasing in second argument} & F(\xi, \mu_{j+1}(t_0)) - F(\xi, \eta_{j+1}(t_0)) \leq 0 . \end{array}$$

This means that “ η_j cannot overtake μ_j ”: no value η_j can ever raise above μ_j . □

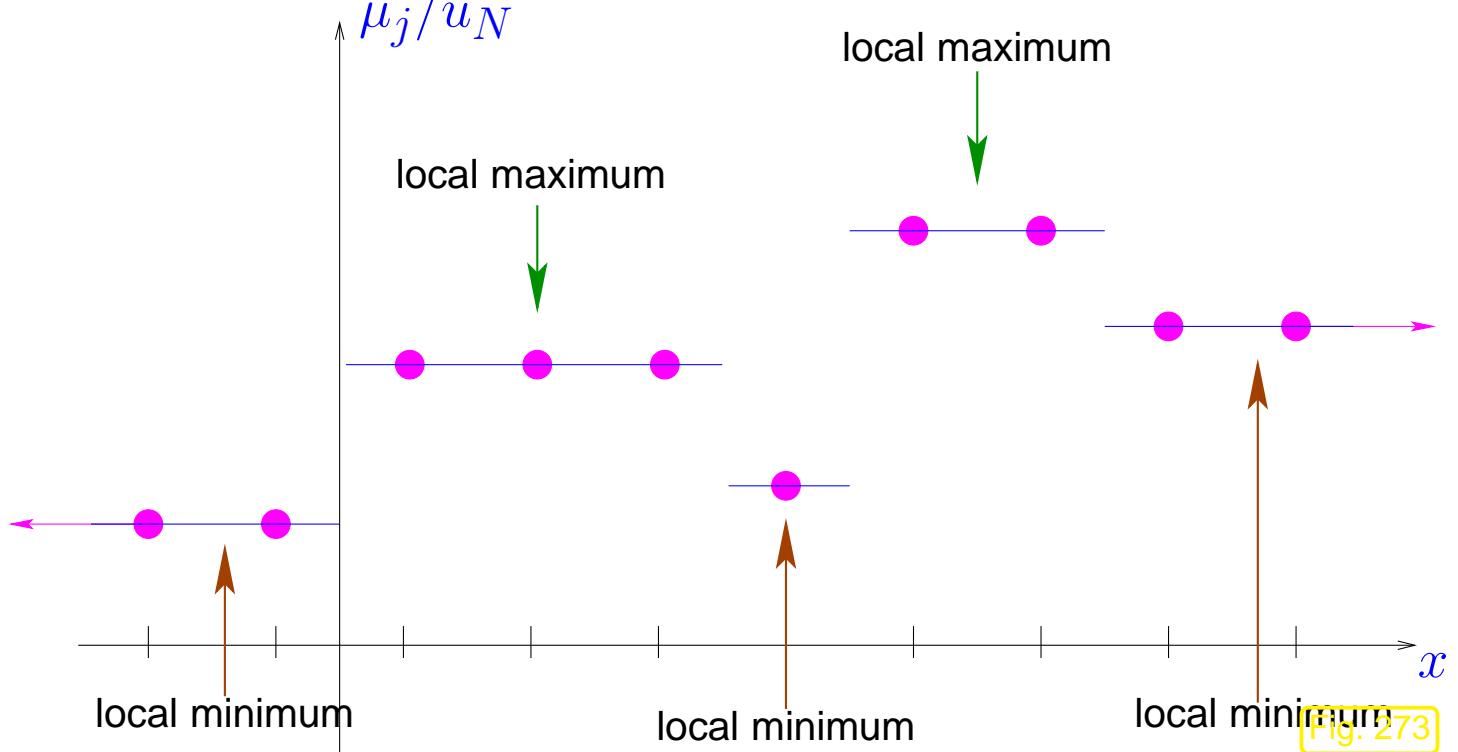
Now we want to study the “preservation of the number of local extrema” during a semi-discrete evolution, another *structural property* of exact solutions of conservation laws, see Sect. 8.2.7.

Intuitive terminology: $\vec{\mu}$ has a **local maximum** $u_m \in \mathbb{R}$, if

$$\exists j \in \mathbb{Z}: \mu_j = u_m \text{ and } \exists k_l < j < k_r \in \mathbb{N}: \max_{k_l < l < k_r} \mu_l = u_m \text{ and } \mu_{k_l} < u_m, \mu_{k_r} < u_m.$$

In analogous fashion, we define a local minimum. If $\vec{\mu}$ is constant for large indices, these values are also regarded as local extrema.

Counting local extrema of $\vec{\mu}$
and the associated piecewise
linear reconstruction.



Lemma 8.3.40 (Non-oscillatory monotone semi-discrete evolutions).

If $\vec{\mu} = \vec{\mu}(t)$ solves (8.3.9) with a monotone numerical flux function $F = F(v, w)$ and $\vec{\mu}(0)$ has finitely many local extrema, then the number of local extrema of $\vec{\mu}(t)$ cannot be larger than that of $\vec{\mu}(0)$.

Proof. $i \hat{=} \text{index of local maximum of } \vec{\mu}(t), t \text{ fixed}$

$$\begin{aligned} \mu_{i-1}(t) &\leq \mu_i(t) \text{ , monotone flux} \implies F(\mu_i, \mu_{i+1}) \geq F(\mu_i, \mu_i) \geq F(\mu_{i-1}, \mu_i) , \\ \mu_{i+1}(t) &\leq \mu_i(t) \\ \Rightarrow \quad \frac{d}{dt} \mu_i(t) &= -\frac{1}{h} (F(\mu_i, \mu_{i+1}) - F(\mu_{i-1}, \mu_i)) \leq 0 . \end{aligned}$$

➤ maxima of $\vec{\mu}$ subside, (minima of $\vec{\mu}$ rise !)

Idea of proof:

No new (local) extrema can arise !

Adjacent values cannot “overtake”:

local maximum: cannot move up

local minimum: cannot move down

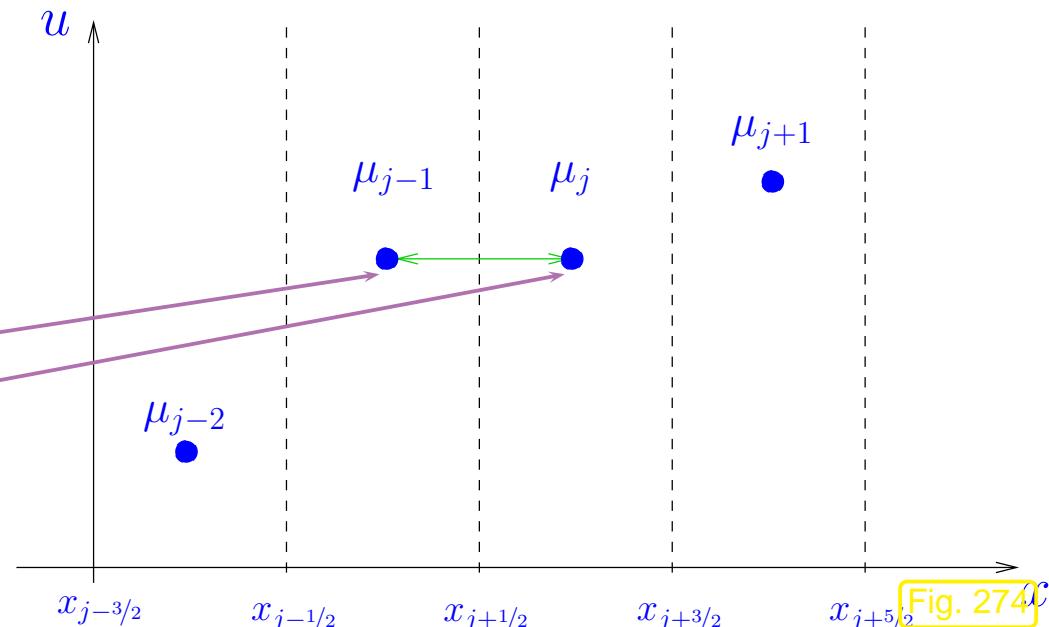


Fig. 274c

8.4 Timestepping

Focus:

Explicit Runge-Kutta timestepping methods (\rightarrow Def. 6.1.26)

Recall [14, Def. 11.4.1]: for explicit s -stage Runge-Kutta single step methods the coefficients a_{ij} vanish for $j \geq i, 1 \leq i, j \leq s$ \Rightarrow the increments \mathbf{K}_i can be computed in turns (without solving a non-linear system of equations).

Initial value problem for abstract semi-discrete evolution in $\mathbb{R}^{\mathbb{Z}}$:

$$\frac{d\vec{\mu}}{dt}(t) = \mathcal{L}_h(\vec{\mu}(t)) , \quad 0 \leq t \leq T , \quad \vec{\mu}(0) = \vec{\mu}_0 \in \mathbb{R}^{\mathbb{Z}} . \quad (8.4.1)$$

Here: $\mathcal{L}_h : \mathbb{R}^{\mathbb{Z}} \mapsto \mathbb{R}^{\mathbb{Z}} \doteq$ (non-linear) **finite difference operator**, e.g. for finite volume semi-discretization in conservation form with 2-point numerical flux:

$$(8.3.9) \Rightarrow (\mathcal{L}_h \vec{\mu})_j := -\frac{1}{h} (F(\mu_j, \mu_{j+1}) - F(\mu_{j-1}, \mu_j)) . \quad (8.4.2)$$

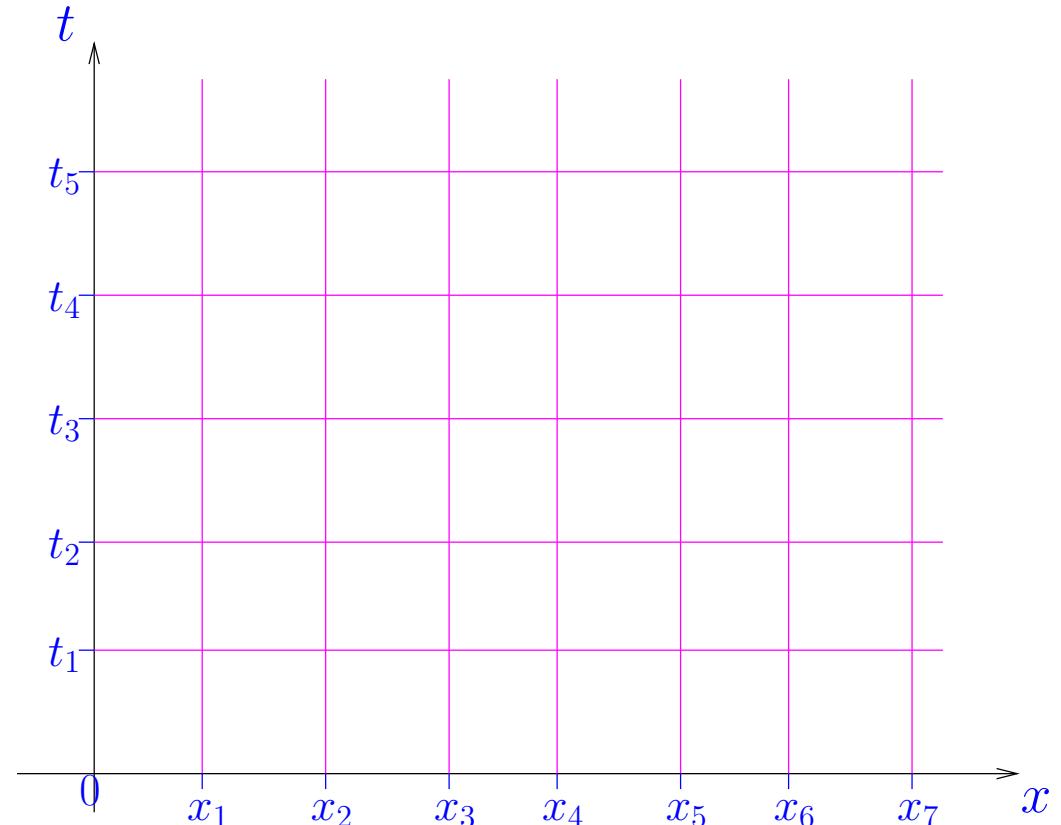
\mathcal{L}_h is local: $(\mathcal{L}_h(\vec{\mu}))_j$ depends only on “neighboring values” $\mu_{j-n_l}, \dots, \mu_{j+n_r}$.

► Explicit s -stage Runge-Kutta single step method for (8.4.1), timestep $\tau > 0$:

$$\begin{aligned} \vec{\kappa}_1 &= \mathcal{L}_h(\vec{\mu}^{(k)}) , \\ \vec{\kappa}_2 &= \mathcal{L}_h(\vec{\mu}^{(k)} + \tau a_{21} \vec{\kappa}_1) , \\ \vec{\kappa}_3 &= \mathcal{L}_h(\vec{\mu}^{(k)} + \tau a_{31} \vec{\kappa}_1 + \tau a_{32} \vec{\kappa}_2) , \\ &\vdots \\ \vec{\kappa}_s &= \mathcal{L}_h(\vec{\mu}^{(k)} + \tau \sum_{j=1}^{s-1} a_{sj} \vec{\kappa}_j) , \end{aligned} \quad \vec{\mu}^{(k+1)} = \vec{\mu}^{(k)} + \tau \sum_{l=1}^s b_l \vec{\kappa}_l . \quad (8.4.3)$$

Here, $a_{ij} \in \mathbb{R}$ and $b_l \in \mathbb{R}$ are the coefficients from the Butcher scheme (6.1.27). For explicit RK-methods the coefficient matrix \mathfrak{A} is strictly lower triangular.

Setting: equidistant spatial mesh \mathcal{M} , meshwidth $h > 0$, nodes $x_j := h j$, $j \in \mathbb{Z}$,
uniform timestep $\tau > 0$, $t_k := \tau k$, $k \in \mathbb{N}_0$.



Single step timestepping for (8.4.1) produces a sequence $(\vec{\mu}^{(k)})_{k \in \mathbb{N}_0}$

$$\mu_j^{(k)} \approx u(x_j, t_k), \quad j \in \mathbb{Z}, k \in \mathbb{N}_0.$$

► Fully discrete evolution

$$\vec{\mu}^{(k+1)} = \mathcal{H}_h(\vec{\mu}^{(k-1)}), \quad k \in \mathbb{N}_0.$$

$\mathcal{H}_h : \mathbb{R}^{\mathbb{Z}} \mapsto \mathbb{R}^{\mathbb{Z}}$: fully discrete evolution operator, arising from applying single step timestepping (8.4.3) to (8.4.1).

Example 8.4.4 (Fully discrete evolutions).

Fully discrete evolution arising from finite volume semi-discretization in conservation form with 2-point numerical flux $F = F(v, w)$

$$(8.3.9) \Rightarrow (\mathcal{L}_h \vec{\mu})_j := -\frac{1}{h} (F(\mu_j, \mu_{j+1}) - F(\mu_{j-1}, \mu_j)) . \quad (8.4.2)$$

in combination with *explicit Euler* timestepping ($\hat{=}$ 1-stage explicit RK-method)

$$\vec{\mu}^{(k+1)} = \vec{\mu}^{(k)} + \tau \mathcal{L}_h(\vec{\mu}^{(k)}) .$$

► $(\mathcal{H}_h(\vec{\mu}))_j = \mu_j^{(k)} - \frac{\tau}{h} (F(\mu_j^{(k)}, \mu_{j+1}^{(k)}) - F(\mu_{j-1}^{(k)}, \mu_j^{(k)})) .$

►
(8.4.5)

In the case of *explicit trapezoidal rule* timestepping [14, Eq. 11.4.3] (method of Heun)

$$\vec{\kappa} = \vec{\mu}^{(k)} + \frac{\tau}{2} \mathcal{L}_h(\vec{\mu}^{(k)}) , \quad \vec{\mu}^{(k+1)} = \vec{\mu}^{(k)} + \tau \mathcal{L}_h(\vec{\kappa}) .$$

► $\kappa_j := (\vec{\kappa})_j = \mu_j^{(k)} - \frac{\tau}{h} (F(\mu_j^{(k)}, \mu_{j+1}^{(k)}) - F(\mu_{j-1}^{(k)}, \mu_j^{(k)})) ,$
 $(\mathcal{H}_h(\vec{\mu}))_j = \mu_j^{(k)} - \frac{\tau}{h} (F(\kappa_j, \kappa_{j+1}) - F(\kappa_{j-1}, \kappa_j)) .$

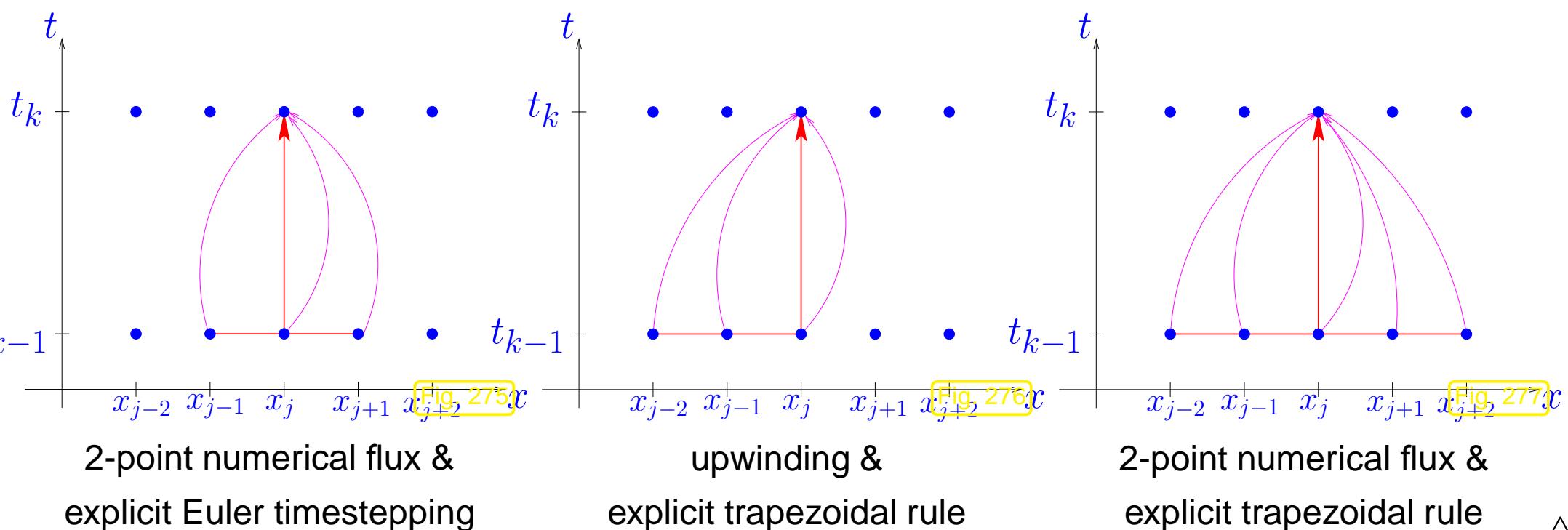
►
(8.4.6)



8.4.1 CFL-condition

Remark 8.4.7 (Difference stencils).

Stencil notation: Visualization of flow of information in fully discrete *explicit* evolution (action of \mathcal{H}_h), cf. Fig. 198.



A consequence of *explicit* timestepping: *locality* of fully discrete evolution operator:

$$\exists m_l, m_r \in \mathbb{N}_0: (\mathcal{H}(\vec{\mu}))_j = \mathcal{H}_j(\mu_{j-m_l}, \dots, \mu_{j+m_r}) . \quad (8.4.8)$$

If flux function f does not depend on x , $f = f(u)$ as in (8.2.7), we can expect

$$\mathcal{H}_h \text{ is translation-invariant: } \mathcal{H}_j = \mathcal{H} \quad \forall j \in \mathbb{Z} .$$

This is the case for (8.4.5) and (8.4.6).

By inspection of (8.4.3): if \mathcal{L}_h is translation-invariant

$$(\mathcal{L}_h(\vec{\mu}))_j = \mathcal{L}(\mu_{j-n_l}, \dots, \mu_{j+n_r}) , \quad j \in \mathbb{Z} ,$$

and timestepping relies on an s -stage explicit Runge-Kutta method, then we conclude for m_l, m_r in (8.4.8)

$$m_l \leq s \cdot n_l , \quad m_r \leq s \cdot n_r .$$

Now we revisit a concept from Sect. 6.2.5, see, in particular, Rem. 6.2.41:

Definition 8.4.9 (Numerical domain of dependence).

Consider explicit translation-invariant fully discrete evolution $\vec{\mu}^{(k+1)} := \mathcal{H}(\vec{\mu}^{(k)})$ on uniform spatio-temporal mesh ($x_j = hj, j \in \mathbb{Z}, t_k = k\tau, k \in \mathbb{N}_0$) with

$$\exists m \in \mathbb{N}_0: (\mathcal{H}(\vec{\mu}))_j = \mathcal{H}(\mu_{j-m}, \dots, \mu_{j+m}), \quad j \in \mathbb{Z}. \quad (8.4.10)$$

Then the **numerical domain of dependence** is given by

$$D_h^-(x_j, t_k) := \{(x_m, t_l) \in \mathbb{R} \times [0, t_k]: j - m(k-l) \leq m \leq j + m(k-l)\}.$$

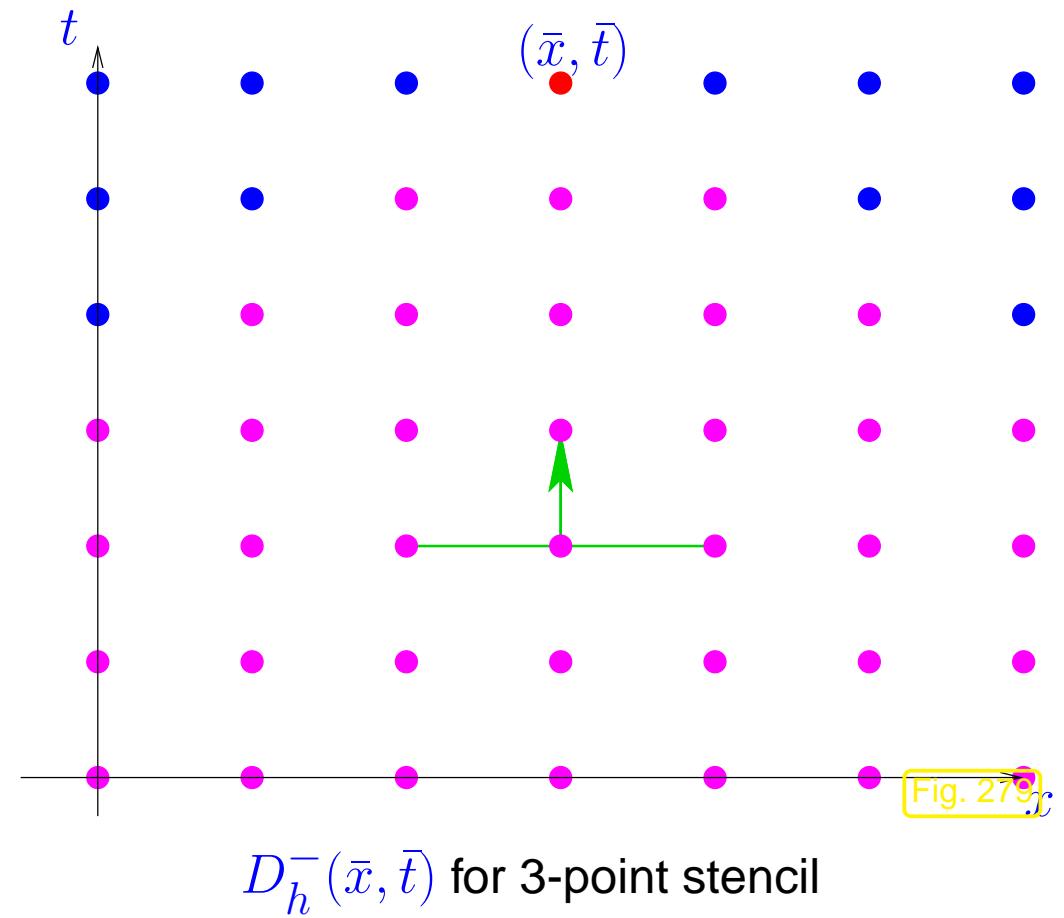
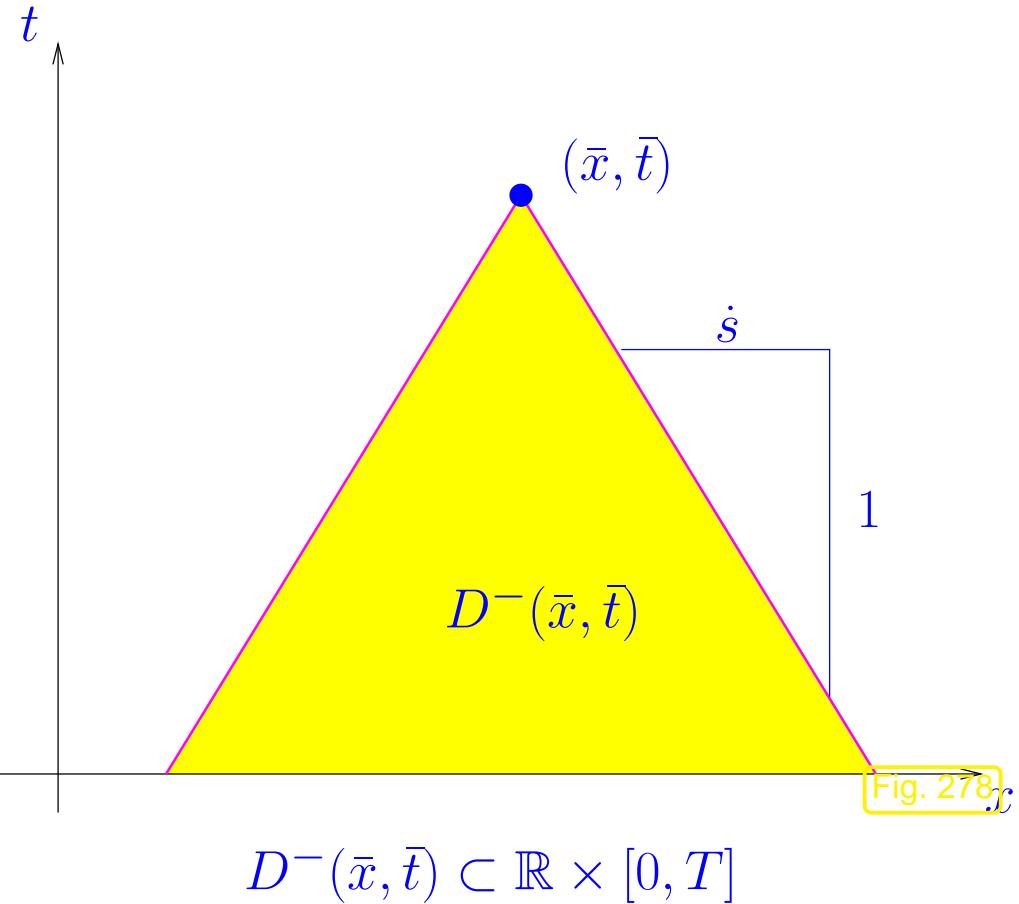
From Thm. 8.2.36 recall the **maximal analytical domain of dependence** for a solution of (8.2.7)

$$D^-(\bar{x}, \bar{t}) := \{(x, t) \in \mathbb{R} \times [0, \bar{t}]: \dot{s}_{\min}(\bar{t}-t) \leq x - \bar{x} \leq \dot{s}_{\max}(\bar{t}-t)\}.$$

with **maximal speeds of propagation**

$$\dot{s}_{\min} := \min\{f'(\xi): \inf_{x \in \mathbb{R}} u_0(x) \leq \xi \leq \sup_{x \in \mathbb{R}} u_0(x)\}, \quad (8.4.11)$$

$$\dot{s}_{\max} := \max\{f'(\xi): \inf_{x \in \mathbb{R}} u_0(x) \leq \xi \leq \sup_{x \in \mathbb{R}} u_0(x)\}. \quad (8.4.12)$$



Definition 8.4.13 (Courant-Friedrichs-Lowy (CFL-)condition). → Rem. 6.2.41

An explicit translation-invariant local fully discrete evolution $\vec{\mu}^{(k+1)} := \mathcal{H}(\vec{\mu}^{(k)})$ on uniform spatio-temporal mesh ($x_j = hj$, $j \in \mathbb{Z}$, $t_k = k\tau$, $k \in \mathbb{N}_0$) as in Def. 8.4.9 satisfies the **Courant-Friedrichs-Lowy (CFL-)condition**, if the convex hull of its numerical domain of dependence contains the maximal analytical domain of dependence:

$$D^-(x_j, t_k) \subset \text{convex}(D_h^-(x_j, t_k))$$

By definition of $D^-(\bar{x}, \bar{t})$ and $D_h^-(x_j, t_k)$ sufficient for the CFL-condition is

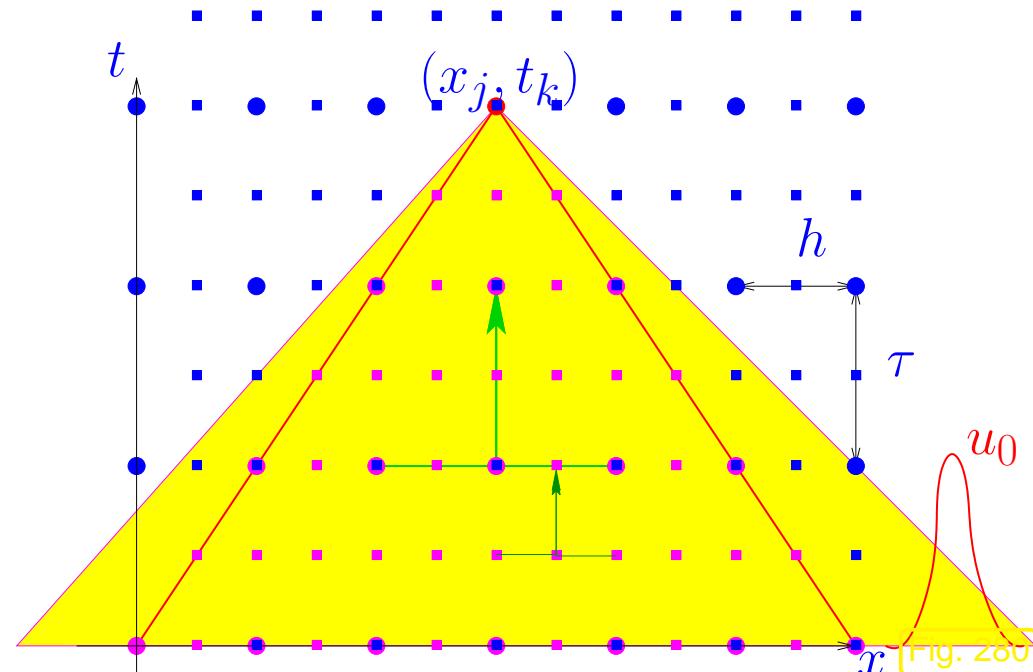
$$\boxed{\frac{\tau}{h} \leq \frac{m}{\dot{s}}} \quad \longleftrightarrow \quad \text{timestep constraint! .} \quad (8.4.14)$$

This is a timestep constraint similar to the one encountered in Sect. 6.2.5 in the context of leapfrog timestepping for the semi-discrete wave equation.

As discussed in Rem. 6.2.41,

We cannot expect convergence for *fixed ratio* $\tau : h$, for $h \rightarrow 0$ in case the CFL-condition is violated.

Refer to Fig. 201 for a “graphical argument”:



($\bullet \hat{=} \text{coarse grid}$, $\blacksquare \hat{=} \text{fine grid}$, $\blacksquare \hat{=} \text{d.o.d}$)

◀ Sequence of equidistant space-time grids of $\mathbb{R} \times [0, T]$ with $\tau = \gamma h$ (τ/h = meshwidth in time/space)

If $\gamma >$ CFL-constraint (8.4.14) then
analytical domain
of dependence

numerical domain
of dependence

8.4.2 Linear stability

In Sect. 6.1.4.2 and Sect. 6.2.5 we found that for explicit timestepping

timestep constraints $\tau \leq O(h^r)$, $r \in \{1, 2\}$, *necessary* to avoid exponential blow-up
(instability)

Is the timestep constraint (8.4.14) suggested by the CFL-condition also stipulated by stability requirements?

We are going to investigate the question only for the Cauchy problem for scalar *linear* advection in 1D with constant velocity $v > 0$:

$$\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} = 0 \quad \text{in } \mathbb{R} \times]0, T[, \quad u(x, 0) = u_0(x) \quad \forall x \in \mathbb{R} . \quad (8.1.4)$$

Semi-discretization in space on equidistant mesh with meshwidth $h > 0$

➤ *linear, local, and translation-invariant* semi-discrete evolution

$$\frac{d\vec{\mu}}{dt}(t) = \mathcal{L}_h(\vec{\mu}(t)) , \quad \text{with} \quad (\mathcal{L}_h(\vec{\mu}))_j = \sum_{l=-m}^m c_l \mu_{j+l} , \quad j \in \mathbb{Z} , \quad (8.4.15)$$

for suitable weights $c_l \in \mathbb{R}$.

Example 8.4.16 (Upwind difference operator for linear advection).

Finite volume semi-discretization of (8.1.4) in conservation form with Godunov numerical flux (8.3.28)
(= upwind flux (8.3.3.3))

$$(\mathcal{L}_h(\vec{\mu}))_j = -\frac{v}{h}(\mu_j - \mu_{j-1}) . \quad (8.4.17)$$

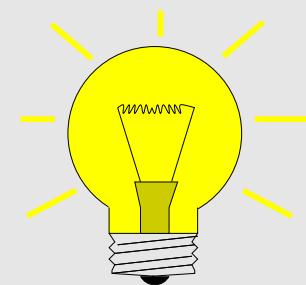
► In (8.4.15): $c_0 = -\frac{v}{h}$, $c_{-1} = \frac{v}{h}$.

Note: Lax-Friedrichs numerical flux (8.3.18) yields the same \mathcal{L}_h .



As in Sect. 6.1.4.2 and Sect. 6.2.5: **diagonalization technique** (with a new twist)

The new twist is that \mathcal{L}_h acts on the sequence space $\mathbb{R}^{\mathbb{Z}}$!



Idea: trial expression for “eigenvectors”

$$\left(\zeta^\xi\right)_j := \exp(i\xi j), \quad j \in \mathbb{Z}, \quad -\pi < \xi \leq \pi. \quad (8.4.18)$$

By straightforward computations:

$$(\mathcal{L}_h(\vec{\mu}))_j = \sum_{l=-m}^m c_l \mu_{j+l} \Rightarrow \mathcal{L}_h \zeta^\xi = \underbrace{\left(\sum_{l=-m}^m c_l \exp(i\xi l) \right)}_{\text{“eigenvalue” } \hat{c}_h(\xi)} \zeta^\xi.$$

► spectrum of \mathcal{L}_h : $\sigma(\mathcal{L}_h) = \{\hat{c}_h(\xi) := \sum_{l=-m}^m c_l \exp(i\xi l): -\pi < \xi \leq \pi\}. \quad (8.4.19)$

Terminology: The function $\hat{c}_h(\xi)$ is known as the **symbol** of the difference operator \mathcal{L}_h , cf. the concept of symbol of a differential operator.

Example 8.4.20 (Spectrum of upwind difference operator).

Apply formula (8.4.19) with $c_0 = -\frac{v}{h}$, $c_{-1} = \frac{v}{h}$ (from (8.4.17)):

For \mathcal{L}_h from (8.4.17):
$$\sigma(\mathcal{L}_h) = \left\{ \frac{v}{h} (\exp(-i\xi) - 1) : -\pi < \xi \leq \pi \right\}$$

Spectrum of upwind finite difference operator for linear advection with velocity $v > 0$ (meshwidth $h > 0$)

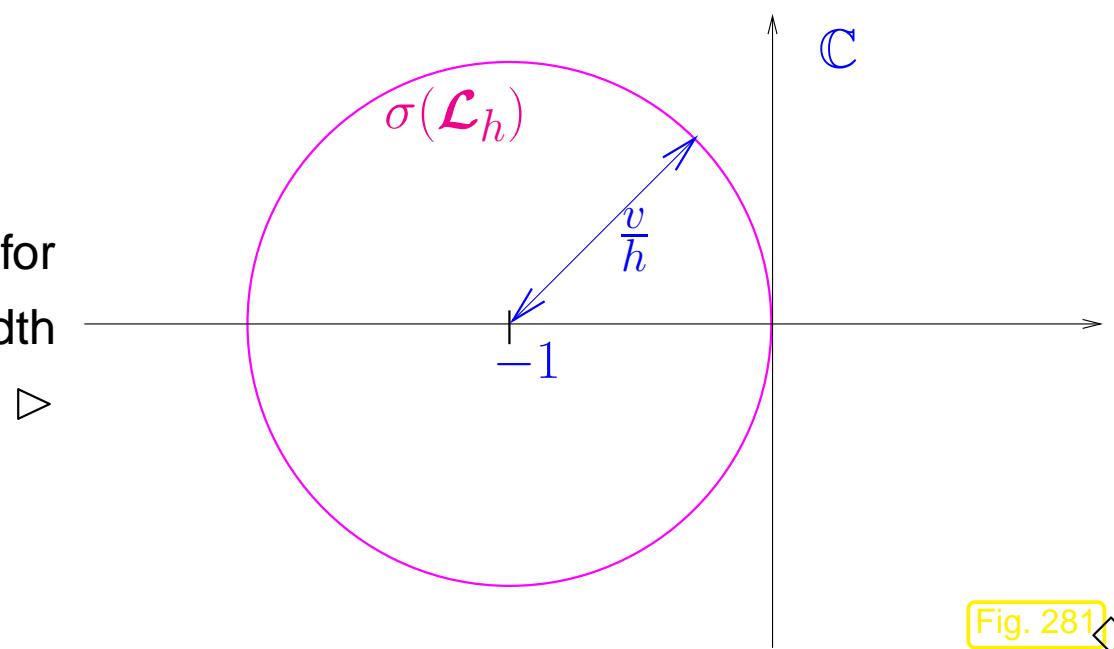


Fig. 281 ◇

Also here: diagonalization of semi-discrete evolution leads to decoupled scalar linear ODEs. However, now we have uncountably many “eigenvectors” $\vec{\zeta}^\xi$, so that linear combination becomes integration:

$$\vec{\mu}(t) = \int_{-\pi}^{\pi} \hat{\mu}(t, \xi) \vec{\zeta}^\xi d\xi \Leftrightarrow \mu_j(t) = \int_{-\pi}^{\pi} \hat{\mu}(t, \xi) \exp(i\xi j) d\xi . \quad (8.4.21)$$

$$\blacktriangleright \frac{d\vec{\mu}}{dt}(t) = \mathcal{L}_h(\vec{\mu}(t)) \Rightarrow \boxed{\frac{\partial \hat{\mu}}{\partial t}(t, \xi) = \hat{c}_h(\xi) \hat{\mu}(t, \xi)} . \quad (8.4.22)$$

This is a family of scalar, linear ODEs parameterized by $\xi \in] -\pi, \pi]$.

Remark 8.4.23 (Fourier series).

Up to normalization the relationship

$$\vec{\mu}^{(0)} \in \mathbb{R}^{\mathbb{Z}} \leftrightarrow \hat{\mu}^{(0)} :] -\pi, \pi] \mapsto \mathbb{C}$$

from (8.4.24) is the **Fourier series transform**, which maps a sequence to a 2π -periodic function. It has the important isometry property

$$\sum_{j=-\infty}^{\infty} |\mu_j|^2 = 2\pi \int_{-\pi}^{\pi} |\hat{\mu}(\xi)|^2 d\xi .$$

- The symbol \hat{c}_h can be viewed as the *representation of a difference operator in Fourier domain.* △

The decoupling manifest in (8.4.22) carries over to Runge-Kutta timestepping in the sense of the commuting diagram (6.1.54).

We introduce the Fourier transforms of the members of the sequence $(\vec{\mu}^{(k)})_k$ created by timestepping

$$\vec{\mu}^{(k)} = \int_{-\pi}^{\pi} \hat{\mu}^{(k)}(\xi) \vec{\zeta}^\xi d\xi \Leftrightarrow \mu_j^{(k)} = \int_{-\pi}^{\pi} \hat{\mu}^{(k)}(\xi) \exp(i\xi j) d\xi . \quad (8.4.24)$$

Example 8.4.25 (Explicit Euler in Fourier domain).

Explicit Euler timestepping [14, Eq. 11.2.1] for semi-discrete evolution (8.4.15), see also (8.4.5),

$$\vec{\mu}^{(k+1)} = \vec{\mu}^{(k)} + \tau \mathcal{L}_h \vec{\mu}^{(k)} .$$

$$\begin{aligned} \blacktriangleright \quad \int_{-\pi}^{\pi} \hat{\mu}^{(k+1)}(\xi) \vec{\zeta}^\xi d\xi &= (\text{Id} + \tau \mathcal{L}_h) \int_{-\pi}^{\pi} \hat{\mu}^{(k)}(\xi) \vec{\zeta}^\xi d\xi = \int_{-\pi}^{\pi} \hat{\mu}^{(k)}(\xi) (1 + \tau \hat{c}_h(\xi)) d\xi . \\ \blacktriangleright \quad \hat{\mu}^{(k+1)}(\xi) &= \hat{\mu}^{(k)}(\xi) (1 + \tau \hat{c}_h(\xi)) . \end{aligned}$$

In Fourier domain a single explicit Euler timestep corresponds to a multiplication of $\hat{\mu} :] -\pi, \pi] \mapsto \mathbb{C}$ with the function $(1 + \tau \hat{c}_h) :] -\pi, \pi] \mapsto \mathbb{C}$.

Relate this to an explicit Euler step for the ODE $\frac{\partial \hat{\mu}}{\partial t}(t, \xi) = \hat{c}_h(\xi) \hat{\mu}(t, \xi)$ from (8.4.22) with parameter ξ :

$$\hat{\mu}^{(k+1)}(\xi) = (1 + \tau \hat{c}_h(\xi)) \hat{\mu}^{(k)}(\xi) .$$

Generalize the observation made in the previous example:

$$\vec{\mu}^{(k)} = \int_{-\pi}^{\pi} \hat{\mu}^{(k)}(\xi) \vec{\zeta}^\xi d\xi ,$$

where $\left(\eta^{(k)}(\xi)\right)_{k \in \mathbb{N}_0}$ is the sequence of approximations created by the Runge-Kutta method when applied to the scalar linear initial value problem

$$\dot{y} = \hat{c}(\xi) y \quad , \quad y(0) = \hat{\mu}^{(0)}(\xi) .$$

Clearly, timestepping can only be stable, if blowup $|\hat{\mu}^{(k)}(\xi)| \rightarrow \infty$ for $k \rightarrow \infty$ can be avoided **for all** $-\pi < \xi \leq \pi$.

From [14, Thm. 12.1.1] we know:

Theorem 8.4.26 (Stability function of explicit Runge-Kutta methods).

The execution of one step of size $\tau > 0$ of an explicit s -stage Runge-Kutta single step method (→ Def. 6.1.26) with Butcher scheme $\begin{array}{c|cc} \mathbf{c} & \mathfrak{A} \\ \hline & \mathbf{b}^T \end{array}$ (see (6.1.27)) for the scalar linear ODE $\dot{y} = \lambda y$, $\lambda \in \mathbb{C}$, amounts to a multiplication with the number

$$\Psi_\lambda^\tau = \underbrace{1 + z\mathbf{b}^T (\mathbf{I} - z\mathfrak{A})^{-1} \mathbf{1}}_{\text{stability function } S(z)} = \det(\mathbf{I} - z\mathfrak{A} + z\mathbf{1}\mathbf{b}^T), \quad z := \lambda\tau, \quad \mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^s.$$

Example 8.4.27 (Stability functions of explicit RK-methods).

- Explicit Euler method (8.4.5) : $\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \Rightarrow S(z) = 1 + z.$

- Explicit trapezoidal rule (8.4.6) : $\begin{array}{c|ccc} 0 & 0 & 0 \\ \hline 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \Rightarrow S(z) = 1 + z + \frac{1}{2}z^2.$

- Classical RK4-method
- [Ex. 11.4.3]

[14,

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
<hr/>				
$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$	

$$\Rightarrow S(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4 .$$

◇

Thm 8.4.26 together with the combinatorial formula for the determinant means that $\Psi_\lambda^\tau(z)$ is a polynomial of degree $\leq s$ in $z \in \mathbb{C}$.

So we conclude for the evolution of “Fourier transforms” $\hat{\mu}^{(k)}(\xi)$:

$$\hat{\mu}^{(k+1)}(\xi) = S(\tau \hat{c}(\xi)) \cdot \hat{\mu}^{(k)}(\xi) , \quad k \in \mathbb{N}_0 , \quad -\pi < \xi \leq \pi ,$$

where $z \mapsto S(z)$ is the **stability function** of the Runge-Kutta timestepping method, see Thm. 8.4.26.

For the explicit Euler method we recover the formula of Ex. 8.4.25.

8.4

$$\text{Stability of RK-timestepping of linear semi-discrete evolution} \iff \max_{-\pi < \xi \leq \pi} |S(\tau \hat{c}(\xi))| \leq 1$$

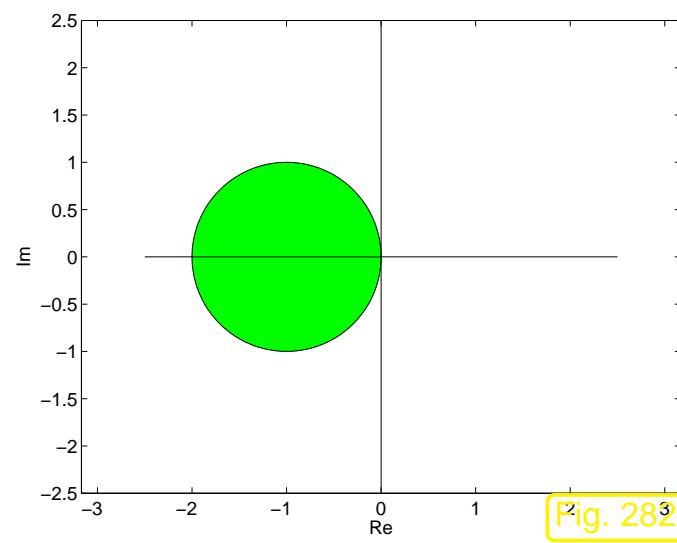
von Neumann stability analysis

Remark 8.4.28 (Stability domains).

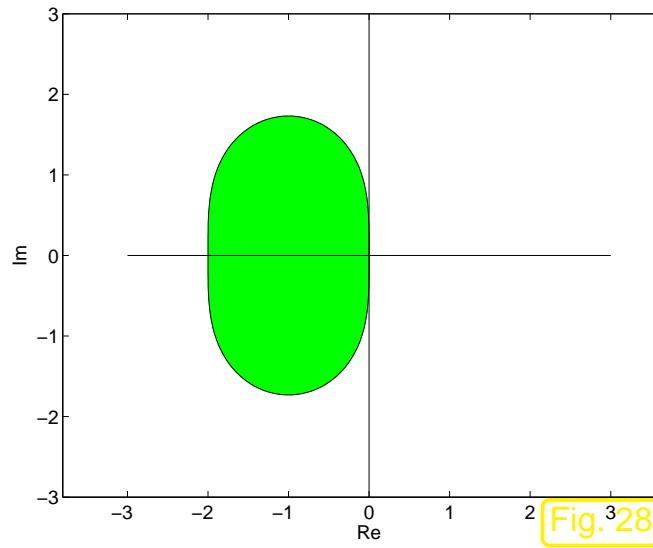
Terminology in the theory of Runge-Kutta single step methods

Stability domain: $\{z \in \mathbb{C}: |S(z)| \leq 1\}$.

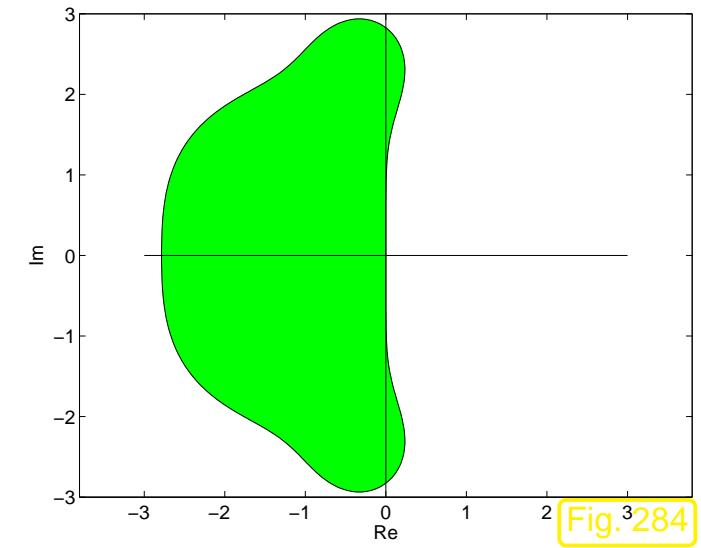
Stability domains:



explicit Euler method



explicit trapezoidal rule



Classical RK4-method

[Fig. 282]

[Fig. 283]

[Fig. 284]

► Necessary stability condition:

$$\{\tau \hat{c}(\xi), -\pi < \xi \leq \pi\} \subset \text{stability domain of RK-method}$$



Example 8.4.29 (Stability and CFL condition).

Consider: upwind spatial discretization (8.4.17) & explicit Euler timestepping

➤ symbol of difference operator (\rightarrow Ex. 8.4.20): $\hat{c}_h(\xi) = \frac{v}{h}(\exp(-i\xi) - 1)$,
 stability function: $S(z) = 1 + z$.

Locus of

$$\Sigma := S(\tau\hat{c}(\xi)) , \quad -\pi < \xi \leq \pi ,$$

in the complex plane

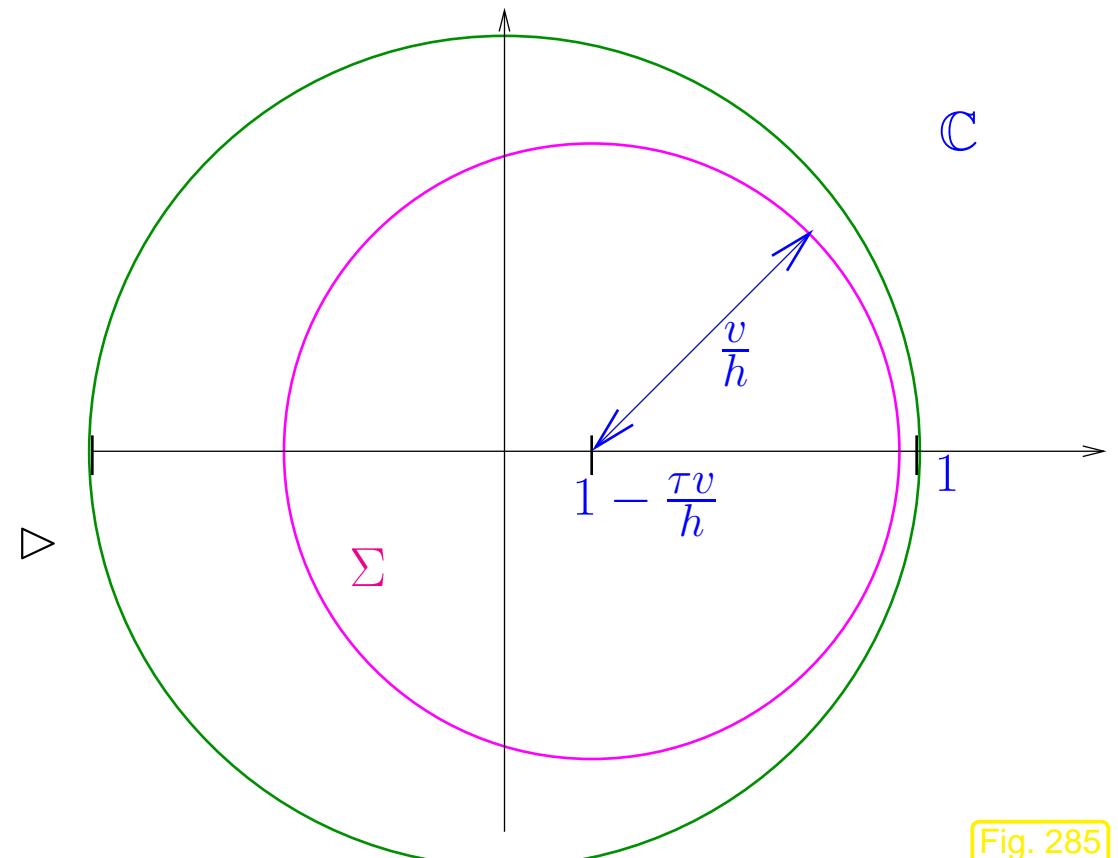


Fig. 285

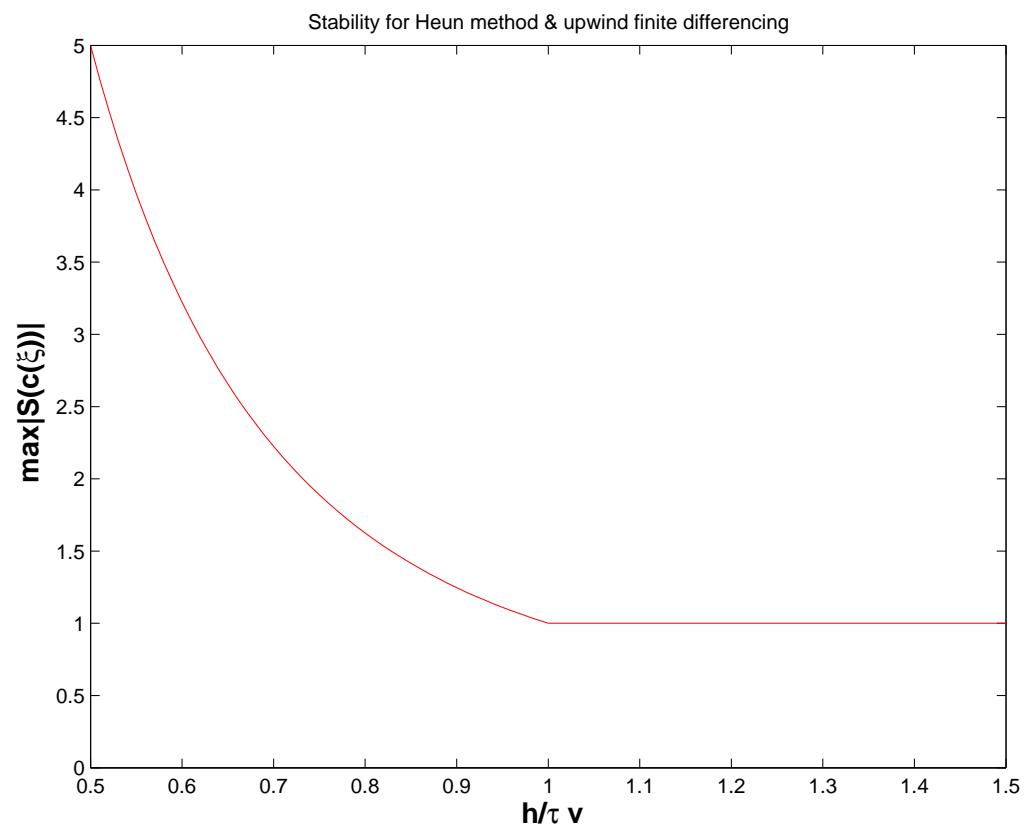
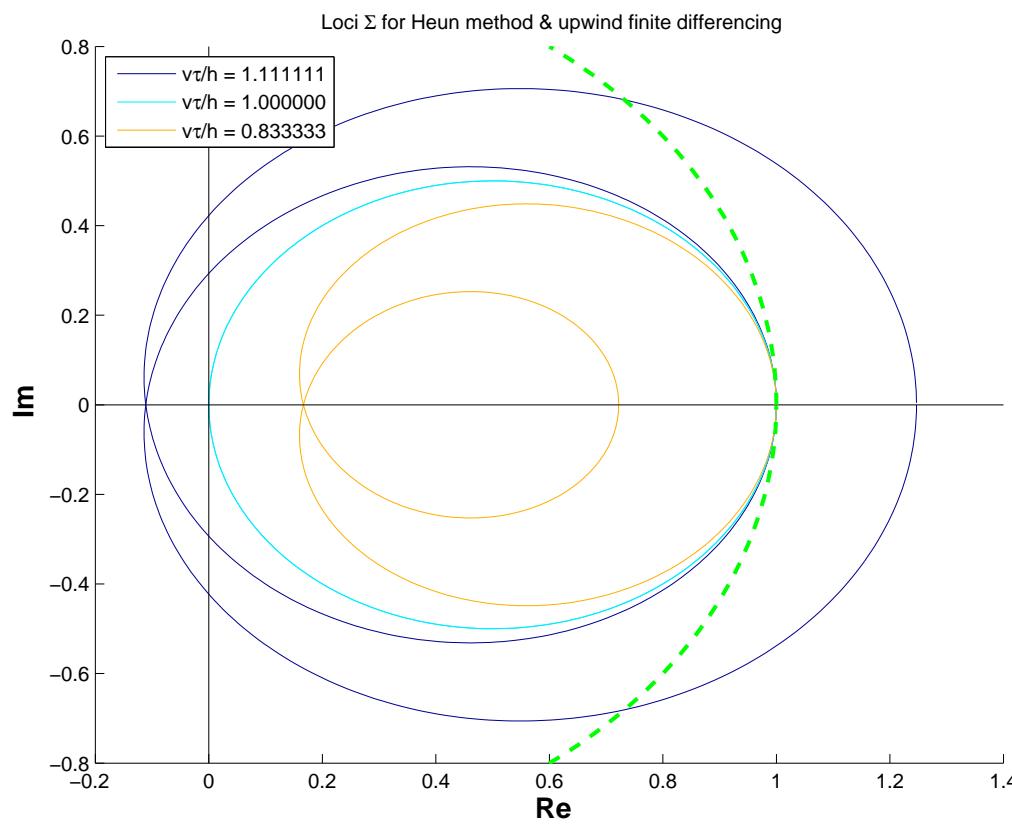
► $|S(\tau\hat{c}(\xi))| \leq 1 \quad \forall -\pi < \xi \leq \pi \iff v\frac{\tau}{h} \leq 1 .$

= CFL-condition of Def. 8.4.13!

Note that the maximal analytic region of dependence for constant velocity v linear advection is merely a line with slope v in the $x - t$ -plane, see Ex. 8.2.10.

Consider: upwind spatial discretization (8.4.17) & explicit trapezoidal: stability function $S(z) = 1 + z + \frac{1}{2}z^2$

Plots for $v = 1, \tau = 1$



$$|S(\tau \hat{c}(\xi))| \leq 1 \quad \forall -\pi < \xi \leq \pi \iff v \frac{\tau}{h} \leq 1 .$$

= *tighter timestep constraint* than stipulated by mere CFL-condition (8.4.14).

To see this note that the explicit trapezoidal rule is a 2-stage Runge-Kutta method. Hence, the spatial stencil has width 2 in upwind direction, see Fig. 276.



8.4.3 Convergence

Example 8.4.30 (Convergence of fully discrete finite volume methods for Burgers equation).

- Cauchy problem for Burgers equation (8.1.11)

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{1}{2} u^2 \right) = 0 \quad \text{in } \mathbb{R} \times]0, T[, \quad u(x, 0) = u_0(x) , \quad x \in \mathbb{R} .$$

- smooth, non-smooth and discontinuous initial data, supported in $[0, 1]$:

$$u_0(x) = 1 - \cos^2(\pi x), \quad 0 \leq x \leq 1, \quad 0 \text{ elsewhere}, \quad (8.4.31)$$

$$u_0(x) = 1 - 2 * |x - \frac{1}{2}|, \quad 0 \leq x \leq 1, \quad 0 \text{ elsewhere}, \quad (8.4.32)$$

$$u_0(x) = 1, \quad 0 \leq x \leq 1, \quad 0 \text{ elsewhere}. \quad (8.4.33)$$

➤ maximum speed of propagation $\dot{s} = 1$.

- Spatial discretization on equidistant mesh with meshwidth $h > 0$ based on finite volume method in conservation form with ① (local) Lax-Friedrichs numerical flux (8.3.18), ② Godunov numerical flux (8.3.28).
- Initial values $\vec{\mu}^{(0)}$ obtained from dual cell averages.
- Explicit Runge-Kutta (order 4) timestepping with uniform timestep $\tau > 0$.
- Fixed ratio: $\tau : h = 1$ (➤ CFL-condition satisfied)
- Monitored: error norms (log-log plots)

$$\text{err}_1(h) := \max_{k>0} h \sum_j |\mu_j^{(k)} - u(x_j, t_k)| \approx \max_{k>0} \|u_N^{(k)} - u(\cdot, t_k)\|_{L^1(\mathbb{R})}, \quad (8.4.34)$$

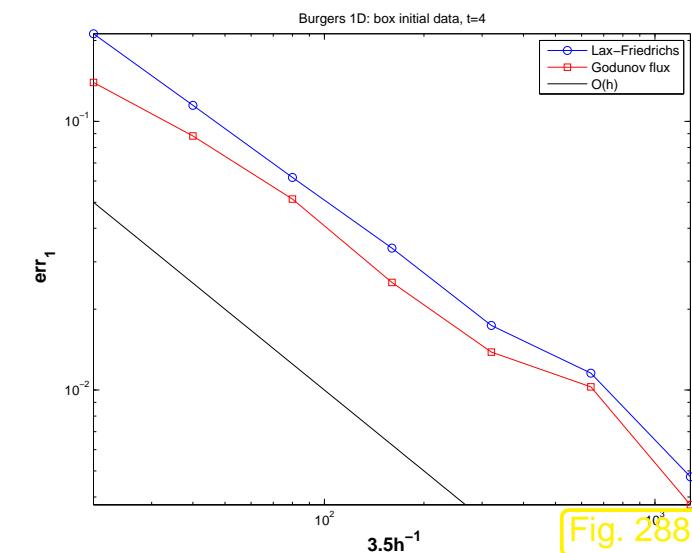
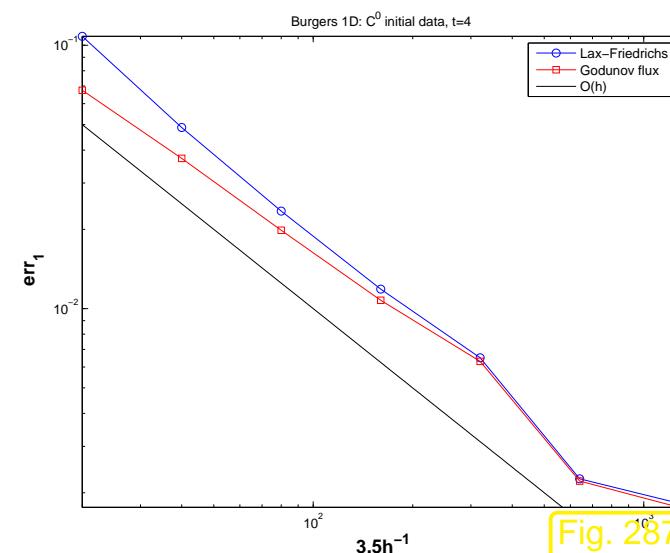
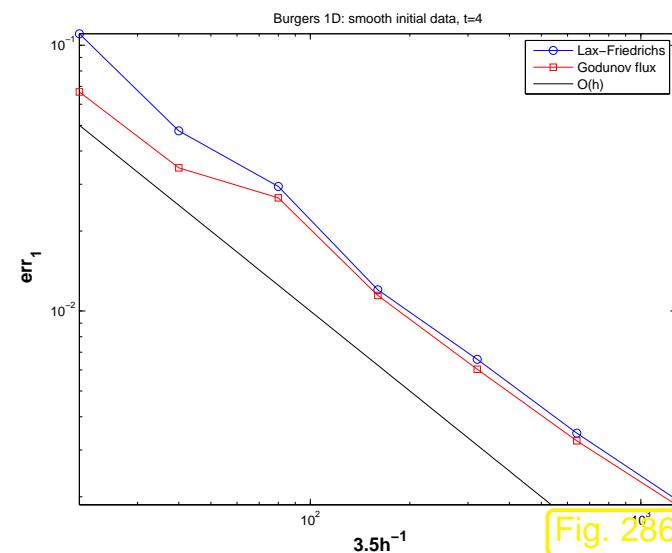
$$\text{err}_\infty(h) := \max_{k>0} \max_{j \in \mathbb{Z}} |\mu_j^{(k)} - u(x_j, t_k)| \approx \max_{k>0} \|u_N^{(k)} - u(\cdot, t_k)\|_{L^\infty(\mathbb{R})}. \quad (8.4.35)$$

for different final times $T = 0.3, 4$, $h \in \{\frac{1}{20}, \frac{1}{40}, \frac{1}{80}, \frac{1}{160}, \frac{1}{320}, \frac{1}{640}, \frac{1}{1280}\}$.

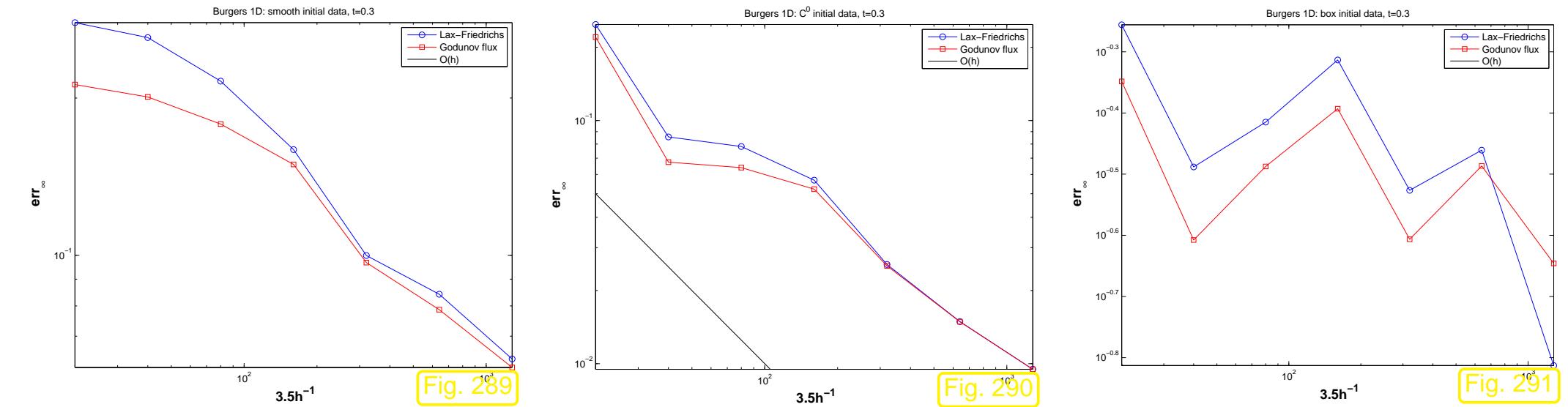
Why do we study the particular error norms (8.4.34) and (8.4.35)?

From Thm. 8.2.34 and Thm. 8.2.36 we know that the evolution for a scalar conservation law in 1D enjoys stability on the norms $\|\cdot\|_{L^1(\mathbb{R})}$ and $\|\cdot\|_{L^\infty(\mathbb{R})}$. Hence, these norms are the natural norms for measuring discretization errors, cf. the use of the energy norm for measuring the finite element discretization error for 2nd order elliptic BVP.

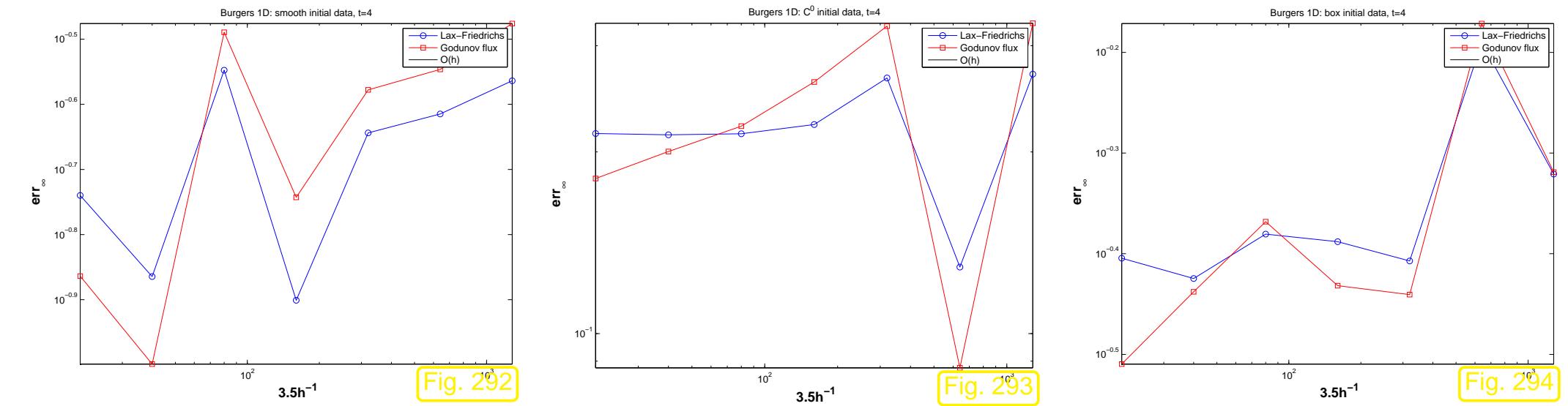
$T = 4$, error err_1



$T = 0.3$: error err_∞



$T = 4$: error err_{∞}



Error obtained by comparison with numerical “reference solution” obtained on a very fine spatio-temporal grid.

Oberservations: for either numerical flux function

- (near) first order algebraic convergence (\rightarrow Def. 1.6.19) w.r.t. mesh width h in err_1 ,
- algebraic convergence w.r.t. mesh width h in err_∞ *before* the solution develops discontinuities (shocks),
- no covergence in norm err_∞ after shock formation.



Best we get: merely first order algebraic convergence $O(h)$

Heuristic explanation for limited order:

$u = u(x, t) \hat{=} \text{smooth}$ entropy solution of Cauchy problem

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \quad \text{in } \mathbb{R} \times]0, T[, \quad u(x, 0) = u_0(x) , \quad x \in \mathbb{R} . \quad (8.2.7)$$

We study the so-called **consistency error** of the numerical flux $F = F(v, w)$

$$(\vec{\tau}_F(t))_j = F(u(x_j), u(x_{j+1}, t)) - f(u(x_{j+1/2}, t)) , j \in \mathbb{Z} ,$$

which measures the deviation of the approximate flux and the true flux, when the approximate solution agreed with the exact solution at the nodes of the mesh.

What we are interested in

behavior of $(\vec{\tau}_F(t))_j$ as mesh width $h \rightarrow 0$,

where an equidistant spatial mesh is assumed.

Terminology:

$$\max_{j \in \mathbb{Z}} (\vec{\tau}_F(t))_j = O(h^q) \quad \text{for } h \rightarrow 0 \quad \leftrightarrow \quad \text{numerical flux consistent of order } q \in \mathbb{N} .$$

Rule of thumb: Order of consistency of numerical flux function limits (algebraic) order of convergence of (semi-discrete and fully discrete) finite volume schemes.

Example 8.4.36 (Consistency error of upwind numerical flux).

Assumption: f continuously differentiable $u_0 \geq 0$ and $f'(u) \geq 0$ for $u \geq 0$ \Rightarrow no transsonic rarefactions!

In this case the upwind numerical flux (8.3.3.3) agrees with the Godunov flux (8.3.28), see Rem. 8.3.29 and

$$F_{\text{uw}}(u(x_j, t), u(x_{j+1}, t)) = f(u(x_j, t)) , \quad j \in \mathbb{Z} .$$

► $(\vec{\tau}_{F_{\text{uw}}}(t))_j = f(u(x_j, t)) - f(u(x_{j+1/2}, t))$
 $= f'(u(x_{j+1/2}, t))(u(x_j, t) - u(x_{j+1/2}, t)) + O(|u(x_j, t) - u(x_{j+1/2}, t)|^2)$
 $= -f'(u(x_{j+1/2}, t))\frac{\partial u}{\partial x}(x_{j+1/2}, t)\frac{1}{2}h + O(h^2) \quad \text{for } h \rightarrow 0 ,$

by *Taylor expansion* of f and u .

This means that the upwind/Godunov numerical flux is (only) *first order consistent*.

Example 8.4.37 (Consistency error of Lax-Friedrichs numerical flux).

Assumption: smooth flux function

Recall: The (local) Lax-Friedrichs numerical flux

$$F_{\text{LF}}(v, w) = \frac{1}{2}(f(v) + f(w)) - \frac{1}{2} \max_{\min\{v,w\} \leq u \leq \max\{v,w\}} |f'(u)|(w - v), \quad (8.3.18)$$

is composed of the central flux and a diffusive flux.

We examine the consistency error for both parts separately, using Taylor expansion

① central flux:

$$\begin{aligned} & \frac{1}{2}(f(u(x_j, t)) + f(u(x_{j+1}, t))) - f(u(x_{j+1/2}, t)) \\ &= \frac{1}{2}f'(u(x_{j+1/2}, t))(u(x_j, t) - u(x_{j+1/2}, t) + u(x_{j+1}, t) - u(x_{j+1/2}, t)) + O(h^2) \quad (8.4.38) \\ &= \frac{1}{2}f'(u(x_{j+1/2}, t))\left(\frac{\partial u}{\partial x}(x_{j+1/2}, t)(-\frac{1}{2}h + \frac{1}{2}h) + O(h^2)\right) + O(h^2) \\ &= O(h^2) \quad \text{for } h \rightarrow 0. \end{aligned}$$

>

The central flux is *second order consistent*.

However, due to instability the central flux is useless, see Sect. 8.3.3.1.

② diffusive flux part:

$$u(x_{j+1}, t) - u(x_j, t) = \frac{\partial u}{\partial x}(x_{j+1/2}, t)h + O(h^2) \quad \text{for } h \rightarrow 0 .$$


$$F_{\text{LF}}(u(x_j, t), u(x_{j+1}, t)) - f(u(x_{j+1/2}, t)) = O(h) \quad \text{for } h \rightarrow 0 ,$$

because the consistency error is dominated by the diffusive flux.

◇

The observations made in the above examples are linked to a general fact:

Monotone numerical fluxes (\rightarrow Def. 8.3.34) are at most first order consistent.

8.5 Higher order conservative schemes

In standard semi-discrete finite volume schemes in conservation form for 2-point numerical flux function, `textcolor{blue}`

$$\frac{d\mu_j}{dt}(t) = -\frac{1}{h} \left(F(\mu_j(t), \mu_{j+1}(t)) - F(\mu_{j-1}(t), \mu_j(t)) \right), \quad j \in \mathbb{Z}, \quad (8.3.9)$$

the numerical flux function is evaluated for the cell averages μ_j , which can be read as approximate values of a projection of the exact solution onto piecewise constant functions (on dual cells)

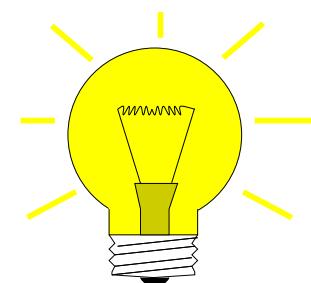
$$\mu_j(t) \approx \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t) dx. \quad (8.3.4)$$

By Taylor expansion we find

$$u(x_{j+1/2}, t) - \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t) dx = O(h) \quad \text{for } h \rightarrow 0 ,$$

and, unless some lucky cancellation occurs as in the case of the central flux, see Ex. 8.4.37, this does not allow more than first order consistency.

Idea: Plug “better” approximations of $u(x_{j\pm 1/2}, t)$ into numerical flux function in (8.3.9)



$$\frac{d\mu_j}{dt}(t) = -\frac{1}{h} (F(\nu_j^+(t), \nu_{j+1}^-(t)) - F(\nu_{j-1}^+(t), \nu_j^-(t))) , \quad j \in \mathbb{Z} ,$$

where ν_j^\pm are obtained by piecewise linear reconstruction from the (dual) cell values μ_j .

$$\begin{aligned}\nu_j^-(t) &:= \mu_j(t) - \frac{1}{2}h\sigma_j(t), \\ \nu_j^+(t) &:= \mu_j(t) + \frac{1}{2}h\sigma_j(t),\end{aligned}\quad j \in \mathbb{Z}, \quad (8.5.1)$$

with suitable **slopes** $\sigma_j(t) = \sigma(\vec{\mu}(t))$.

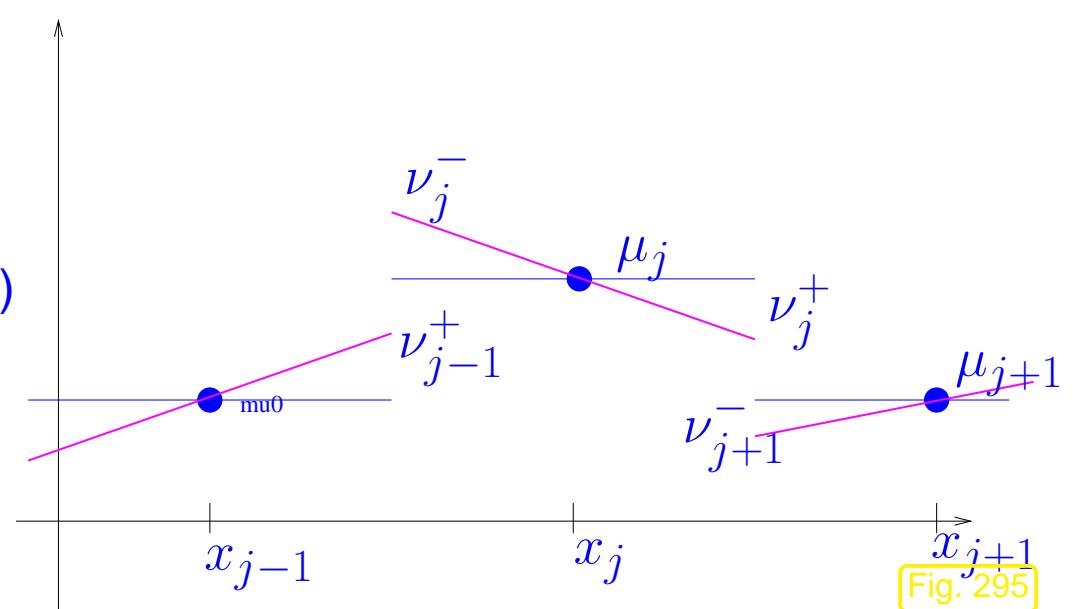


Fig. 295

Index

H^1 -semi-norm, 137

L^2 -norm, 134

2-regularity
of Dirichlet problem, 575

a priori estimates, 475

a-orthogonal, 454

affine linear function
in 2D, 278

affine transformation, 368

algorithm
numerical, 66

analytic solution, 65

angle condition
for Delaunay mesh, 438

anisotropic diffusion, 740

artificial diffusion, 736

artificial viscosity, 736

assembly, 292

cell oriented, 356

in FEM, 349

barycentric coordinate representation
of local shape functions, 363

barycentric coordinates, 293

basis

change of, 270

best approximation error, 456

beta function, 366

bilinear form, 46, 60

positive definite, 174

bilinear transformation, 397

boundary conditions, 24, 63, 223

Dirichlet, 223, 234

homogeneous, 236

Neumann, 226, 235

radiation, 235

boundary fitting, 408

parabolic, 408

boundary flux

computation of, 557

boundary layer, 711

boundary value, 156

boundary value problem
 elliptic, 237
boundary value problem (BVP), 156
bounding box, 342
Burger's equation, 821
Burgers equation, 796
calculus of variations, 41
Cauchy problem, 788, 799, 813
 for one-dimensional conservation law, 805
 for wave equation, 659
cell, 306
cell contributions, 350
central flux, 846
CFL-condition, 685, 889
characteristic curve, 806
characteristic method, 764
Chebychev nodes, 120
circumcenter, 437
classical solution, 56, 246
coefficient vector, 269
collocation, 114, 115
 spline, 123
compatibility condition, 248
compatibility conditions
 for $H^1(\Omega)$, 196
complete space, 450
composite midpoint rule, 104
composite trapezoidal rule, 104
computational domain, 228
computational effort, 520
conditionally stable, 648
configuration space, 24
 for taut membrane, 162
congruent matrices, 271
conservation
 of energy, 669
conservation form, 841
conservation law, 800
 differential form, 803
 integral form, 803
 one-dimensional, 804
 scalar, 804
conservation of energy, 229
consistency error, 910
continuity
 of a linear form, 193
 of linear functional, 550
control volume, 433, 800
convection-diffusion equation, 699
convective cooling, 235
convective terms, 699
convergence, 130, 459
convex function, 177
corner singular function, 532
Courant-Friedrichs-Levy condition (CFL), 685
Crank-Nicolson method, 615
curve, 24
 length, 25
 parametrization, 24
cut-off function, 565
d'Alembert solution, 660
Delaunay mesh

angle condition, 438
delta distribution, 211
dielectric tensor, 167
difference quotient, 127
differential operator, 118, 156
diffusion tensor, 740
diffusive flux, 802
diffusive terms, 699
Dirac delta function, 211
Dirichlet boundary conditions, 223, 234
Dirichlet data, 203, 252
Dirichlet problem, 223

- variational formulation, 202

discrete model, 30
discrete variational problem, 264, 269
discretization, 66, 262
discretization error, 130, 456
discretization parameter, 459
displacement, 63
displacement function, 63
dissipation, 608
DistMesh, 342
divergence

- of a vectorfield, 217

domain, 156

- computational, 228
- spatial, 64, 160

domain of dependence, 662
domain of influence, 662
dual mesh, 436
dual problem, 554

duality estimate, 554
edge, 306
elastic energy, 29

- mass-spring model, 30

elastic string, 22
electric field, 165
electric scalar potential, 166
electromagnetic field energy, 166
electrostatic field energy, 167
electrostatics, 165
element, 306
element load vector, 352
element stiffness matrix, 352
elliptic

- linear scalar second order PDE, 233

elliptic boundary value problem, 237
energy

- conservation, 229
- of electrostatic field, 167

energy conservation

- for wave equation, 665

energy norm, 175
equidistant mesh, 97, 127, 314
equilibrium length

- of spring, 29

equilibrium principle, 31
equivalence

- of norms, 498

Euler equations, 796
Euler method, 614
evolution operator

- fully discrete, 883
semi-discrete, 882
evolution problem, 595
expansion shock, 821
explicit Euler method, 614

face, 306
field energy
 electromagnetic, 166
finite difference methods, 421
finite differences
 1D, 126
 in 2D, 421
finite elements
 parametric, 401
finite volume methods, 432
flow field, 693
flux function, 800, 805
force density, 163
Fourier's law, 230
 if fluid, 698
Frobenius norm, 491
functional, 548
 linear, 550
fundamental lemma of calculus of variations, 221

Galerkin discretization, 69, 263
Galerkin matrix, 350
Galerkin orthogonality, 454
Galerkin solution, 71
 quasi-optimality, 456
Galerkin test space, 71

Galerkin trial space, 71
Gamma function, 366
Gauss' theorem, 218, 233, 698, 811
Gauss-Legendre quadrature, 376
Gauss-Lobatto quadrature, 376
General entropy solution for 1D scalar Riemann problem, 830
generic constants, 517
global shape functions, 314
Godunov numerical flux, 866
gradient
 of a function, 166
Green's first formula, 218
grid
 1D, 97, 127
grid function, 131

h-refinement, 526
hanging node, 309
hat function, 199, 282
heat capacity, 597
heat conductivity, 231
heat equation, 597
heat flux, 228, 231
 computation of, 557
 convective, 698
 diffusive, 698
heat source, 230
Hessian, 487
Heun method, 884
Hilbert space, 450
homogeneous boundary conditions, 236

Hooke's law, 29
hyperbolic evolution problem, 656
discrete case, 666

implicit Euler method, 614
implicit midpoint rule, 615
increments
 Runge-Kutta, 616
index mapping matrix, 354
inflow, 800
inflow boundary, 713
initial conditions, 594
initial value problem
 stiff, 625
initial-boundary value problems (IBVP), 595
 parabolic, 599
integrated Legendre polynomials, 81
integration by parts
 in 1D, 53
 multidimensional, 218
intermediate state, 862
interpolant
 piecewise linear, 441
interpolation error, 486
interpolation error estimates
 anisotropic, 504
 in 1D, 477
interpolation nodes, 321
inviscid, 793
kinetic energy, 665
 $L(\pi)$ -stability, 639

L-shaped domain, 467
L-stability, 639
Lagrangian finite elements, 320, 358
 on quadrilaterals, 393
Lagrangian method, 764
 for advection, 755
Laplace equation, 222
Lax entropy condition, 829
Lax-Friedrichs flux, 852
leapfrog, 675
Legendre polynomials, 82
 integrated, 81
LehrFEM, 336
lifting theorem, 530, 537
linear boundary fitting, 544
linear form, 46
 continuity, 193
linear function
 in 2D, 278
linear interpolation
 in 1D, 477
 in 2D, 485, 486
linear variational problem, 60
Linearity, 238
linearization
 of variational problems, 409
load vector, 269, 350
local linearization, 414
local operations, 357
local shape function, 316
 barycentric representation, 391

local shape functions
quadratic, 322

mass lumping, 677

mass-spring model, 28
elastic energy, 30

material coordinate, 26

material tensor, 203

mathematical modelling, 20

maximum principle, 580, 705

Maxwell's equations
static case, 166

mean value formula, 487

membrane, 158
potential energy, 162, 163

membrane problem
variational formulation, 202

mesh, 306
1D, 97, 127
data structures, 343
equidistant, 97, 127
node, 275
non-conforming, 309
quadrilateral, 307
simplicial, 309
triangular, 307

mesh data structure, 343

mesh file format, 337
triangular mesh, 338

mesh generation, 342

mesh generator, 337

mesh width, 463

method of characteristics, 712

method of lines, 611

midpoint rule
composite, 104

mixed boundary conditions, 237

Mixed Neumann–Dirichlet problem, 227

model
continuous, 66
discrete, 30, 66

monomial basis, 80

multi-index notation, 311

multiplicative trace inequality, 256

NETGEN, 342

Neumann boundary conditions, 226, 235

Neumann data
admissibilty conditions, 253

Neumann problem, 247
compatibility condition, 248
variational form, 247

Newton update, 417

Newton's method, 413
in function space, 413

Newton's second law of motion, 653

Newton-Cotes formula, 376

Newton-Galerkin iteration, 417

nodal basis, 281

nodal interpolation operators, 496

nodal value, 282

node, 275
1D, 97, 127

quadrature, 86

norm, 132
on function space, 133
numerical domain of dependence, 887
numerical flux, 434, 841
numerical flux function, 434
numerical quadrature, 86
 nodex, 367
 weights, 367
offset function, 47
order of quadrature rule, 368
outflow, 800
outflow boundary, 713
output functional, 548

p-refinement, 526
parametric finite elements, 401, 402
parametric quadrature rule, 367
parametrization
 of curve, 24
particle method, 764
PDE
 ILinear scalar second order elliptic, 233
perpendicular bisector, 437
phase space, 800
Phythagoras' theorem, 454
piecewise linear interpolant, 441
piecewise linear reconstruction, 915
piecewise quadratic interpolation, 497
Poincaré-Friedrichs inequality, 192, 254
point force, 52
Poisson equation, 222, 460

Poisson matrix, 425
polar coordinates, 212
polynomials
 degree, 310
 multivariate, 310
 univariate, 78
positive definite
 bilinear form, 174
 uniformly, 168
postprocessing, 131
potential energy, 665
 of taut membrane, 162
problem parameters
 for elastic string, 28
problem size, 520
procedural form
 of functions, 261
product rule, 606
 in higher dimensions, 216
production term, 800
pullback, 385

quadratic functional, 171
quadratic local shape functions, 322
quadratic minimization problems, 170
quadrature formula, 86
quadrature node, 86
quadrature nodes, 367
quadrature rule, 367
 on triangle, 372
 order, 368
 parametric, 367

- quadrature rules
 Gauss-Legendre, 376
 Gauss-Lobatto, 376
quadrature weight, 86
quadrature weights, 367
quadrilateral mesh, 307
quasi-optimality, 456
- Radau timestepping, 639
radiation boundary conditions, 235, 245
rarefaction
 subsonic, 865
 supersonic, 865
 transonic, 865
rarefaction wave/fan, 827
reaction term
 in 2nd-order BVP, 273
reference elements, 402
regular refinement, 457
reversibility, 669
Riemann problem, 818
Riesz representation theorem, 450
right hand side vector, 350
Robin boundary conditions, 235
rubber band, 22
Runge-Kutta
 increments, 616
Runge-Kutta method, 616
Runge-Kutta methods
 stability function, 899
- SDIRK timestepping, 639
- semi-discrete, 611
semi-norm, 137
sensitivity
 of a problem, 207
shape functions
 global, 314
shape regularity measure, 494
shock, 819
 physical, 829
 subsonic, 865
 supersonic, 865
shock speed, 819
similarity solution, 825
simplicial mesh, 309
singular perturbation, 715
Sobolev norms, 500
Sobolev semi-norms, 502
Sobolev space $H^1(\Omega)$, 190
Sobolev space $H_0^1(\Omega)$, 187
Sobolev spaces, 181, 500
solution
 analytic, 65
 approximate, 66
source term, 156
space-time-cylinder, 594
sparsity pattern, 289
spatial domain, 160
spectrum, 893
spline
 cubic, 124
spline collocation, 123
- 8.5
p. 924

- spring constant, 29
Störmer scheme, 674
stability
 of linear variational problem, 206
stability domain, 901
stability function
 of explicit Runge-Kutta methods, 899
 of RK-SSM, 639
stiff IVP, 625
stiffness
 of spring, 29
stiffness matrix, 269, 350
 sparsity, 316
Strang splitting, 756
streamline, 694
streamline diffusion, 735
strong form, 56
subsonic rarefaction, 865
subsonic shock, 865
supersonic rarefaction, 865
supersonic shock, 865
supremum norm, 133
symbol
 of a difference operator, 893
- T-matrix, 354
Taylor expansion, 43
tensor product polynomials, 312
tensor-product grid, 422
tent function, 98
test function, 47
test space, 47
- TETGEN, 342
trace theorem, 256
trajectory, 694
transformation of functions, 385
transformation techniques, 401
translation-invariant, 886
transonic rarefaction, 865
transport equation, 752
transsonic rarefaction fan, 858
trapezoidal rule
 composite, 103, 104
trial space, 47, 115
triangle inequality, 132
triangular mesh, 307
triangular mesh: file format, 338
triangulation, 306
two-point boundary value problem, 56
two-step method, 674
- uniformly positive, 232
upwind quadrature, 729, 731
upwinding, 724
- variational crime, 539
variational equation
 linear, 60
 non-linear, 44
variational problem
 discrete, 264, 269
 linear, 60
 non-linear, 410
 perturbed, 539

vertex, 306
virtual work principle, 45
von Neumann stability analysis, 901
Voronoi cell, 436
Voronoi dual mesh, 436

wave equation, 656
weak form, 56
weak solution, 813
weight
 quadrature, 86
width
 of a mesh, 463

Examples and Remarks

- L^2 interpolation error, 572
 L^2 -convergence of FE solutions, 569
 L^2 -estimates on non-convex domain, 577
 L^∞ interpolation error estimate in 1D, 515
 $|\cdot|_{H^1(\Omega)}$ -semi-norm, 190
(Bi)-linear Lagrangian finite elements on hybrid meshes, 334
[Fully discrete evolutions, 883
[Membrane with free boundary values, 223
ode45 for discrete parabolic evolution, 622
“PDEs” for univariate functions, 21
“Physics based” discretization, 67
1D convection-diffusion boundary value problem, 711
Admissible Dirichlet data, 252
Admissible Neumann data, 253
Affine transformation of triangles, 368
Approximate Dirichlet boundary conditions, 380
Approximate sub-steps for Strang splitting time, 761
Approximation of mean temperature, 551, 554
Arrays storing 2D triangular mesh, 346
Assembly for linear Lagrangian finite elements on triangular mes, 355
Assembly for quadratic Lagrangian FEM, 358
Asymptotic estimates, 525
Asymptotic nature of a priori estimates, 517
Barycentric representation of local shape functions, 391
Bases for polynomial spectral collocation, 120
Behavior of generalized eigenvalues of $\mathbf{A}\vec{\mu} = \lambda\mathbf{M}\vec{\mu}$, 628
Benefit of variational formulation of BVPs, 99
Bilinear Lagrangian finite elements, 327
Blow-up for leapfrog timestepping, 681
Boundary conditions and $L^2(\Omega)$, 185
Boundary conditions for 2nd-order parabolic IBVPs, 600
Boundary conditions for linear advection, 792
Boundary conditions for wave equation, 658
Boundary conditions in $H_0^1(\Omega)$, 188
Boundary value problems, 155
Boundary values for conservation laws, 804
Breakdown of characteristic solution formula, 809

- Causes for non-smoothness of solutions of elliptic BVPs, 538
- Central flux for Burgers equation, 846
- Central flux for linear advection, 850
- Characteristics for advection, 807
- Choice of basis for polynomial spectral Galerkin methods, 80
- Choice of timestepping for m.o.l. for transient convection-diffusion, 750
- Coefficients/data in procedural form, 68
- Collocation approach on “complicated” domains, 420
- Collocation points for polynomial spectral collocation, 120
- Compatible boundary and initial data, 599
- Computation of heat flux, 558, 566
- Conditioning of linear variational problems, 206
- Conrner singular functions, 531
- Consistency error of Lax-Friedrichs numerical flux, 912
- Consistency error of upwind numerical flux, 911
- Continuity of interpolation operators, 503
- Convective cooling, 235
- Convergence of Euler timestepping, 617
- Convergence of fully discrete finite volume methods for Burgers equation, 905
- Convergence of fully discrete timestepping in one spatial dimension, 641
- Convergence of Lagrangian FEM for p -refinement, 472
- Convergence of linear and quadratic Lagrangian finite elements in L^2 -norm, 465
- Convergence of linear and quadratic Lagrangian finite elements in energy norm, 460
- Crank-Nicolson timestepping, 615
- Delaunay-remeshing in 2D, 772
- Derivative of non-linear $u \mapsto a(u; \cdot)$, 415
- Difference stencils, 885
- Differentiating a functional on a space of curves, 44
- Differentiating bilinear forms with time-dependent arguments 606
- Diffusive flux, 802
- Dimensionless equations, 27
- Dimensions of Lagrangian finite element spaces on triangular meshes, 521
- Discontinuity connecting constant states, 817
- Domain of dependence/influence for 1D wave equation, constant coefficient case, 661
- Effect of added diffusion, 736
- Efficient implementation of assembly, 361
- Elastic string shape by finite element discretization, 112
- Elliptic lifting result in 1D, 529
- Energy conservation for leapfrog, 678
- Energy norm, 134
- Energy norm and $H^1(\Omega)$ -norm, 498
- Entropy solution of Burgers equation, 830
- Euler equations, 796
- Euler timestepping, 614
- Euler timestepping for 1st-order form of semi-discrete wave equation, 669
- Evaluation of local shape functions at quadrature points, 390
- Explicit Euler in Fourier domain, 897
- Exploring convergence experimentally, 149
- Extended MATLAB mesh data structure, 346
- Extra regularity requirements, 57

- Extra smoothness requirement for PDE formulation, 223
Finding continuous replacement functionals, 568
Finite differences for convection-diffusion equation in 1D, 717
First-order semidiscrete hyperbolic evolution problem, 668
Formulation in physical space coordinate, 62
Fourier series, 895
Gap between interpolation error and best approximation error, 513
Generic constants, 517
Geometric interpretation of CFL condition in 1D, 685
Geometric obstruction to Voronoi dual meshes, 437
Godunov flux for Burgers equation, 868
Good accuracy on “bad” meshes, 510
Grid functions, 131
heat conduction, 237
Heat conduction with radiative cooling, 411
Higher order timestepping for 1D heat equation, 644
Impact of choice of basis, 270
Impact of linear boundary fitting on FE convergence, 544
Impact of numerical quadrature on finite element discretization error, 541
Implementation of non-homogeneous Dirichlet b.c. for linear FE, 381
Implementation of spectral Galerkin discretization for elastic string problem, 91
Implementation of spectral Galerkin discretization for linear 2nd-order two-point BVP, 87
Implicit Euler method of lines for transient convection-diffusion, 747
Imposing homogeneous Dirichlet boundary conditions, 333
Inefficiency of conditionally stable single step methods, 648
Initial time, 595
Internal layers, 738
Interpolation nodes for cubic and quartic Lagrangian FE in 2D, 325
 $L(\pi)$ -stable Runge-Kutta single step methods, 639
Lagrangian finite elements on hybrid meshes, 336
Lagrangian method for convection-diffusion in 1D, 776
Laplace operator, 222
Lax-Friedrichs flux for Burgers equation, 853
Leapfrog timestepping, 675
Linear FE discretization of 1D convection-diffusion problem, 717
Linear finite element Galerkin discretization for elastic string model, 106
Linear finite element space for homogeneous Dirichlet problem, 282
Linear variational problems, 60
Local interpolation onto higher degree Lagrangian finite element spaces, 496
Local quadrature rules on quadrilaterals, 375
Local quadrature rules on triangles, 372
Mass lumping, 677
Material coordinate, 26
Mathematical modelling, 20
Maximum principle for finite difference discretization, 585

- Maximum principle for higher order Lagrangian FEM, 592
 Maximum principle for linear FE for 2nd-order elliptic BVPs, 591
 Mesh file format for MATLAB code, 340
 Minimal regularity of membrane displacement, 164
 Mixed boundary conditions, 237
 Naive finite difference scheme, 836
 Non-continuity of boundary flux functional, 563
 Non-differentiable function in $H_0^1([0, 1])$, 199
 Non-existence of solutions of positive definite quadratic minimization problem, 179
 Non-linear variational equation, 46
 Non-polynomial “bilinear” local shape functions, 399
 Non-smooth external forcing, 51
 Norms on grid function spaces, 138
 Numerical quadrature in LehrFEM, 373
 Offset function for finite element Galerkin discretization, 106
 Offset functions and Galerkin discretization, 73
 offset functions for linear Lagrangian FE, 379
 One-sided difference approximation of convective terms, 721
 Ordered basis of test space, 75
 Output functionals, 548
 Parametrization of a curve, 24
 Piecewise linear functions (not) in H^1 , 196
 Piecewise quadratic interpolation, 497
 Point charge, 210
 Point particle method for pure advection, 765
 Properties of weak solutions, 814
 Pullback of functions, 385
 Quadratic functionals with positive definite bilinear form in 2D, 176
 Quadratic minimization problems on Hilbert spaces, 186
 Quadratic tensor product Lagrangian finite elements, 331
 Quasi-locality of solution of scalar elliptic boundary value problem, 241
 Radiative cooling, 236
 Recirculating flow, 713
 Relationship between discrete minimization problem and discrete variational problem, 71
 Scalar elliptic boundary value problem in one space dimension, 239
 Scaling of entries of element matrix for $-\Delta$, 296
 Semi-Lagrangian method for convection-diffusion in 1D, 784
 Smoothness of solution of scalar elliptic boundary value problem, 240
 Smoothness requirements for collocation trial space, 117
 Solution formula for sourceless transport, 753
 Space of square integrable functions, 183
 Sparse stiffness matrices, 288
 Spatial difference operators for linear advection, 892
 Spatial discretization options, 611
 Spatial domains, 160
 Specification of local quadrature rules, 370
 Spectral Galerkin discretization of non-linear variational problem, 95
 Spectral Galerkin discretization with quadrature, 86
 Spectrum of elliptic operators, 633

Spectrum of upwind difference operator, 893	Wave equation as first order system in time, 658
Spurious Galerkin solution for 2D convection-diffusion BVP, 725	Well-posed 2nd-order linear elliptic variational problems, 451
Stability and CFL condition, 902	
Stability domains, 901	
Stability functions of explicit RK-methods, 899	
Streamline-diffusion discretization, 742	
Supports of global shape functions in 1D, 314	
Supports of global shape functions on triangular mesh, 315	
Tense string without external forcing, 38	
Timestepping for ODEs, 68	
Transformation of basis functions, 85	
Transformation techniques for bilinear transformations, 406	
Triangular mesh: file format, 338	
Triangular quadratic Lagrangian finite elements, 320	
Uniqueness of solutions of Neumann problem, 248	
Upwind flux and expansion shocks, 867	
Upwind flux and transsonic rarefaction, 857	
Upwind flux for Burgers equation, 856	
Upwind quadrature discretization, 733	
Vanishing viscosity for Burgers equation, 821	
Variational formulation for convection-diffusion BVP, 709	
Variational formulation for heat conduction with Dirichlet boundary conditions, 243	
Variational formulation for pure Neumann problem, 247	
Variational formulation: heat conduction with general radiation boundary conditions, 245	
Virtual work principle, 45	
	8.5
	p. 931

Definitions

H^1 -semi-norm, 137

Affine transformation, 368

Characteristic curve for one-dimensional scalar conservation law, 806

congruent matrices, 271

Consistent numerical flux function, 844

Courant-Friedrichs-Levy (CFL-)condition, 889

Cubic spline, 124

element load vector, 352

element stiffness matrix, 352

Energy norm, 175

Higher order Lagrangian finite element spaces, 320

Higher order Sobolev norms, 500

Higher order Sobolev semi-norms, 502

Higher order Sobolev spaces, 500

Imcompressible flow field, 701

$L(\pi)$ -stability, 639

Lax entropy condition, 829

Legendre polynomials, 82

Linear interpolation in 2D, 486

Local shape functions, 317

Material derivative, 781

Mean square norm/ L^2 -norm, 134

mesh, 306

mesh width, 463

Monotone numerical flux function, 874

multivariate polynomials, 310

norm, 132

Numerical domain of dependence, 887

parametric finite elements, 402

Positive definite bilinear form, 174

pullback, 385

Quadratic functional, 171

Quadratic minimization problem, 172

Riemann problem, 818

Runge-Kutta method, 616

Shape regularity measure for simplex, 494

shock, 819

Singularly perturbed problem, 715

Sobolev space $H^1(\Omega)$, 190

Sobolev space $H_0^1(\Omega)$, 187

Space $L^2(\Omega)$, 184

sparse matrix, 288

Support of a function, 101

Supremum norm, 133

Tensor product Langrangian finite element spaces, 331

tensor product polynomials, 312

Uniformly positive definite tensor field, 168

Weak solution of Cauchy problem for conservation law,

813

MATLAB codes

assemMat_QFE , 359
sparse (MATLAB-function) , 359
add_Edge2Elem , 346
add_Edges , 346
init_Mesh , 346
Loading a mesh from file , 341

List of Symbols

$C_0^2([0, 1]) := \{v \in C^2([0, 1]): v(0) = v(1) = 0\}$, 41	$\mathcal{V}(\mathcal{M}) \hat{=} \text{set of vertices of a mesh}$, 275
$C_0^\infty(\Omega) \hat{=} \text{smooth functions with support inside } \Omega$, 192	$\Delta \hat{=} \text{Laplace operator}$, 222
$C^k([a, b]) \hat{=} k\text{-times continuously differentiable functions}$ on $[a, b] \subset \mathbb{R}$, 25	$\text{div } \mathbf{j} \hat{=} \text{divergence of a vectorfield}$, 217
$C_{\text{pw}}^k([a, b])$, 51	$\mathcal{E}(\mathcal{M})$, 347
$D^-(\bar{x}, \bar{t}) \hat{=} \text{maximal analytical domain of dependence of}$ (\bar{x}, \bar{t}) , 888	I_1 , 486
$D^\alpha u \hat{=} \text{multiple partial derivatives}$, 500	$H^m(\Omega) \hat{=} m\text{-th order Sobolev space}$, 500
$M_i \hat{=} i\text{-th integrated Legendre polynomial}$, 81	$\mathcal{S}_1^0(\mathcal{M})$, 277
$S(z) \hat{=} \text{stability function of Runge-Kutta method}$, 899	$I_1 \hat{=} \text{piecewise linear interpolation on finite element mesh}$, 441
\mathbf{n} , 234	$P_n \hat{=} n\text{-th Legendre polynomial}$, 82
$\mathbf{n} \hat{=} \text{exterior unit normal vectorfield}$, 218	$\mathcal{S}_p^0(\mathcal{M}) \hat{=} H^1(\Omega)\text{-conforming Lagrangian FE space}$, 320
$\mathcal{H}_h \hat{=} \text{fully discrete evolution operator}$, 883	$L^\infty(\Omega) \hat{=} \text{space of (essentially) bounded functions on } \Omega$, 133
$\mathcal{L}_h \hat{=} \text{semi-discrete evolution operator due to 1D conserva-}$ tion law, 882	$L^2(\Omega) \hat{=} \text{space of square-integrable functions on } \Omega$, 184
$\mathcal{P}_p(\mathbb{R}) \hat{=} \text{space of univariate polynomials of degree } \leq p$, 78	$\ \cdot\ _0 \hat{=} \text{norm on } L^2(\Omega)$, 184
$\mathcal{P}_p(\mathbb{R}^d)$, 310	$\ \cdot\ _\infty \hat{=} \text{supremum norm of a function/maximum norm of}$ a vector, 133
$\mathcal{P}_p(\mathbb{R}^d) \hat{=} \text{space of } d\text{-variate polynomials}$, 310	$\ u\ _{H^m(\Omega)} \hat{=} m\text{-th order Sobolev norm}$, 500
$\mathcal{Q}_p(\mathbb{R}^d)$, 312	$\ u\ _{L^\infty(\Omega)} \hat{=} \text{supremum norm of } u : \Omega \mapsto \mathbb{R}^n$, 133
	$\ \cdot\ _{L^2(\Omega)} \hat{=} L^2\text{-norm of a function}$, 134

$\ \cdot\ _{L^2(\Omega)}$	$\hat{=}$ norm on $L^2(\Omega)$, 184	
$\ \cdot\ _0$	$\hat{=}$ L^2 -norm of a function, 134	
$\mathcal{V}(\mathcal{M})$, 306	
Ω	, 156	
Ω	$\hat{=}$ spatial domain or parameter domain, 24	
Φ^*	, 385	
$ u _{H^m(\Omega)}$	m -th order Sobolev semi-norm, 502	
$ \cdot _{H^1(\Omega)}$	$\hat{=}$ H^1 -semi-norm of a function, 137	
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \dots$	(matrices), 269	
\mathbf{M}^{-T}	$\hat{=}$ inverse transposed of matrix, 405	
\mathbf{a}_K	$\hat{=}$ restriction of bilinear form \mathbf{a} to cell K , 292	
\cdot	$\hat{=}$ inner product of vectors in \mathbb{C}^n , 31	
χ_I	$\hat{=}$ characteristic function of an interval $I \subset \mathbb{R}$, 840	
$\ddot{u} := \frac{\partial u}{\partial t^2}$, 654	
$\dot{u}(t)$	$\hat{=}$ (partial) derivative w.r.t. time, 603	
ℓ_K	restriction of linear form ℓ to cell K , 301	
$\frac{Df}{D\mathbf{v}}(t)$	$\hat{=}$ material derivative w.r.t. velocity field \mathbf{v} , 781	
grad	$\hat{=}$ gradient of a scalar valued function, 166	
$\hat{c}(\xi)$	$\hat{=}$ symbol of a finite difference operator, 893	
$\mathbf{1} = (1, \dots, 1)^T$, 899	
dS	$\hat{=}$ integration over a surface, 218	
\mathcal{M}	, 306	
$\nabla F(\mathbf{x}) := \text{grad } F(\mathbf{x})$	$\hat{=}$ nabla nontation for gradient, 167	
$\text{diam}(\Omega)$	$\hat{=}$ diameter of $\Omega \subset \mathbb{R}^d$, 161	
nnz	, 288	
$\partial\Omega$	$\hat{=}$ boundary of domain Ω , 161	
ρ_K	$\hat{=}$ shape regularity measure of cell K , 494	
$\rho_{\mathcal{M}}$	$\hat{=}$ shape regularity measure of a mesh \mathcal{M} , 494	
$\vec{\mu}, \vec{\varphi}, \vec{\xi}, \dots$	(coefficient vectors), 269	
$\mathcal{S}_{p,0}^0(\mathcal{M})$	$\hat{=}$ Degree p Lagrangian finite element space with zero Dirichlet boundary conditions., 333	
$\mathcal{S}_{1,0}^0(\mathcal{M})$	$\hat{=}$ space of p.w. linear C^0 -finite elements, 98	
$H_0^1(\Omega)$	Sobolev space, 187	
$h_{\mathcal{M}}$	$\hat{=}$ mesh width of mesh \mathcal{M} , 463	
$h_{\mathcal{M}}$	$\hat{=}$ meshwidth of a grid, 97	
$x_{j-1/2} := \frac{1}{2}(x_j + x_{j-1})$	$\hat{=}$ midpoint of cell in 1D, 840	

Bibliography

- [1] A. Burtscher adn E. Fonn, P. Meury, and C. Wiesmayr. *LehrFEM - A 2D Finite Element Toolbox*. SAM, ETH Zürich, Zürich, Switzerland, 2010. [http:..](http://)
- [2] I. Babuška and M. Suri. The optimal convergence rate of the p-version of the finite element method. *SIAM J. Numer. Anal.*, 24(4):750–769, 1987.
- [3] D. Braess. *Finite Elements*. Cambridge University Press, 2nd edition, 2001.
- [4] S. Brenner and R. Scott. *Mathematical theory of finite element methods*. Texts in Applied Mathematics. Springer–Verlag, New York, 2nd edition, 2002.
- [5] D. Christodoulou. The Euler equations of compressible fluid flow. *Bull. American Math. Soc.*, 44(4):581–602, 2007.
- [6] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*, volume 4 of *Studies in Mathematics and its Applications*. North-Holland, Amsterdam, 1978.

- [7] B. Cockburn and J. Gopalakrishnan. New hybridization techniques. *GAMM-Mitteilungen*, (2):28, 2005.
- [8] C.M. Dafermos. *Hyperbolic conservation laws in continuum physics*, volume 325 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, 2000.
- [9] D.A. Dunavant. High degree efficient symmetrical Gaussian quadrature rules for the triangle. *Int. J. Numer. Meth. Engr.*, 21:1129–1148, 1985.
- [10] L.C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 1998.
- [11] W. Hackbusch. *Elliptic Differential Equations. Theory and Numerical Treatment*, volume 18 of *Springer Series in Computational Mathematics*. Springer, Berlin, 1992.
- [12] W. Hackbusch. *Integral equations. Theory and numerical treatment.*, volume 120 of *International Series of Numerical Mathematics*. Birkhäuser, Basel, 1995.
- [13] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer, Heidelberg, 2002.
- [14] R. Hiptmair. Numerische mathematik für studiengang rechnergestützte wissenschaften. Lecture Slides, 2005. <http://www.sam.math.ethz.ch/~hiptmair/tmp/NCSE.pdf>.
- [15] P. Knabner and L. Angermann. *Numerical Methods for Elliptic and Parabolic Partial Differential Equations*, volume 44 of *Texts in Applied Mathematics*. Springer, Heidelberg, 2003.
- [16] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM J. Numer. Anal.*, 21(2):217–235, 1984.

- [17] P.-O. Persson and G. Strang. A simple mesh generator in matlab. *SIAM Review*, 46(2):329–345, 2004.
- [18] C. Schwab. *p- and hp-Finite Element Methods. Theory and Applications in Solid and Fluid Mechanics*. Numerical Mathematics and Scientific Computation. Clarendon Press, Oxford, 1998.
- [19] M. Struwe. Analysis für informatiker. Lecture notes, ETH Zürich, 2009. <https://moodle-app1.net.ethz.ch/lms/mod/resource/index.php?id=145>.
- [20] S. Zlotnik and P. Díez. Assembling sparse matrices in MATLAB. *Communications in Numerical Methods in Engineering*, 2008. Published Online: 1 Sep 2008.

Appendix

A

Essential skills

This chapter lists essential skills that you possess after having studied the individual chapters of the course.

A.1 Chapter ??: Prologue: A Two-point Boundary Value Problem

You should know:

- a simple example for a two-point boundary value problem,
- the ideas behind its discretization by means of finite differences, collocation, and the Galerkin method,

- what is meant by discretization error and convergence of a method.
- different qualitative kinds of convergence and how to detect them in numerical experiments,
- how to measure the rate of algebraic convergence in numerical experiments
- what is meant by the “asymptotic nature” of convergence.

A.2 Chapter ??: Second-order scalar elliptic boundary value problems

You should know

- the concept of a second-order scalar elliptic boundary value problem together with appropriate boundary conditions (Dirichlet, Neumann, radiation, mixed).
- the concept of uniform positivity of the conductivity/coefficient of a second-order scalar differential operator and its consequences for the associated bilinear form.
- the definition of weak derivatives and the rationale for introducing them

- to derive the complete variational form (bilinear form, right hand side functional and Hilbert space framework) for any linear second-order scalar elliptic boundary value problem.
- fundamental notions like ellipticity and continuity (of linear/bilinear forms)
- that functions in $H^1(\Omega)$ can be unbounded for $d > 1$.
- the Lax-Milgram lemma and how to prove the ellipticity/continuity of bilinear forms arising from linear second-order scalar elliptic boundary value problems (Poincaré-Friedrichs inequalities)
- the compatibility condition for the pure Neumann problem.
- the trace theorem from $H^1(\Omega)$ and its significance for admissible Dirichlet and Neumann boundary data.
- how to tell invalid and valid source terms and boundary data from each other.
- how to express the variational form of a linear second-order scalar elliptic boundary value problem as an equivalent minimization problem.

A.3 Chapter ??: The Finite Element Method (FEM)

A.3

- the idea of the Galerkin approximation, Galerkin-orthogonality (5.1.7), and Cea's Lemma Thm. 5.1.10 (including the proof)
- how to derive a linear system of equations from linear variational problems
- the terms stiffness matrix and load vector
- the impact of a change of bases on the stiffness matrix and the Galerkin solution
- the concept of a mesh
- that FE functions for $H^1(\Omega)$ have to be continuous
- that FE spaces possess bases of locally supported functions associated with vertices/edges/cells
- the rationale behind the use of locally supported basis functions in FEM
- simplicial and quadrilateral Lagrangian FE (their local polynomial spaces and interpolation points)
- the concept of parametric (particularly affine equivalent) Lagrangian FE
- the concept of local assembly for the efficient computation of the stiffness matrix and load vector
- the use of parametric FE for the approximation of curved boundaries
- how to use numerical quadrature to approximate the coefficients of the stiffness matrix and load vector
- how to deal with non-homogeneous Dirichlet boundary conditions
- the notation of difference stencils

- the discrete maximum principle and its consequences for the discrete solution
- the idea behind finite volume methods and the construction of dual meshes
- different types of refinement and convergence and how to tell them from raw error data
- what the Bramble-Hilbert lemma Thm. ?? does tell
- the concept of shape regularity of simplicial meshes and its role in the transformation estimates (Lemma ??) for norms
- that acute angles do not affect accuracy of FE Galerkin solutions but obtuse angles are harmful
- the notion 2-regularity of a 2nd order elliptic boundary value problem
- what happens in case of reentrant corners to solutions of 2nd order elliptic boundary value problem
- the impact of numerical quadrature on the convergence rate of Lagrangian FE
- the convergence rates of Lagrangian FE in the energy and the L^2 -norm in case of h -refinement
- that you can gain up to twice the convergence rate in the energy norm for the evaluation of $H^1(\Omega)$ -continuous linear functionals

Practical: Implementing Lagrangian FE for 2nd order boundary value problems in 2D using the MATLAB environment of the exercises

A.4 Chapter ??: Special elliptic boundary value problems

You should know

- what a singular perturbed problem is
- what special phenomena are encountered in the case of convection-diffusion problems
- the idea behind upwinding and streamline diffusion
- the result of quasi optimality of Galerkin solutions and the notion dispersion in the case of the Helmholtz equation
- that saddle point problems lead to mixed formulations
- the notions continuous and discrete inf-sup condition and their importance for the discretization of saddle point problems
- the Stokes equation and some stable pairs for its FE discretization
- what consistent iteration does mean
- the principle underlying a fix point iteration
- how to derive Newton's method for non-linear elliptic boundary value problems

A.5 Chapter ??: Solving discrete boundary value problems

You should know

- what the idea behind successive subspace correction is
- the terms iteration matrix, contraction number, and rate of convergence of linear stationary iterative methods
- how the hierarchical basis is defined
- the idea behind multigrid methods
- what the idea behind the cg- and the pcg-method is
- that the condition number of the stiffness matrix to 2nd order elliptic boundary value problems grows like $h_{\mathcal{M}}^{-2}$
- that increasing condition numbers of the iteration matrix generally slow down the convergence of linear stationary iterative methods

A.6 Chapter ??: Parabolic Boundary Value Problems

You should know

- how a 2nd order parabolic initial BVP looks like
- what the method of lines is
- what a stable single step method is, particularly the notion $L(\pi)$ -stability
- why implicit timestepping schemes have to be used
- that spatial and temporal errors enter the a priori estimates in an additive fashion

A.7 Chapter ??: Numerical Methods for Conservation Laws

You should know

A.7

- how (nonlinear) conservation laws look like

p. 948

- the concept of characteristics and their importance
- that classical solutions make no sense in case of shocks
- how to derive physically meaningful weak solutions
- the solution of a Riemann problem
- what a method in conservation form is
- what the idea behind the Godunov scheme is and its properties
- what the Lax-Friedrichs and Lax-Wendroff schemes are
- monotone schemes and the order barrier theorem
- the continuous and numerical region of dependence and their consequences (\rightarrow CFL-condition)

A.8 Chapter ??: Adaptive Finite Element Schemes

You should know

- the idea behind a priori adaptivity

- a few important a posteriori estimators for 2nd order elliptic BVPs
- how to derive goal oriented error estimators
- the properties of reliable and efficient error estimators
- the algorithm for adaptive local mesh refinement controlled by an a posteriori error estimator