A Guide to Numerical Methods for Transport Equations

Dmitri Kuzmin

Friedrich-Alexander-Universität
Erlangen-Nürnberg

2010

Contents

Chapter 1
Getting Started

In this chapter, we start with a brief introduction to numerical simulation of transport phenomena. We consider mathematical models that express certain conservation principles and consist of convection-diffusion-reaction equations written in integral, differential, or weak form. In particular, we discuss the qualitative properties of exact solutions to model problems of elliptic, hyperbolic, and parabolic type. Next, we review the basic steps involved in the design of numerical approximations and the main criteria that a reliable algorithm should satisfy. The chapter concludes with an outline of the rationale behind the scope and structure of the present book.

1.1 Introduction to Flow Simulation

Fluid dynamics and transport phenomena, such as heat and mass transfer, play a vitally important role in human life. Gases and liquids surround us, flow inside our bodies, and have a profound influence on the environment in which we live. Fluid flows produce winds, rains, floods, and hurricanes. Convection and diffusion are responsible for temperature fluctuations and transport of pollutants in air, water or soil. The ability to understand, predict, and control transport phenomena is essential for many industrial applications, such as aerodynamic shape design, oil recovery from an underground reservoir, or multiphase/multicomponent flows in furnaces, heat exchangers, and chemical reactors. This ability offers substantial economic benefits and contributes to human well-being. Heating, air conditioning, and weather forecast have become an integral part of our everyday life. We take such things for granted and hardly ever think about the physics and mathematics behind them.

The traditional approach to investigation of a physical process is based on observations, experiments, and measurements. The amount of information that can be obtained in this way is usually very limited and subject to measurement errors. Moreover, experiments are only possible when a small-scale model or the actual equipment has already been built. An experimental investigation may be very time-consuming, dangerous, prohibitively expensive, or impossible for another reason.

Alternatively, an analytical or computational study can be performed on the basis of a suitable mathematical model. As a rule, such a model consists of several differential and/or algebraic equations which make it possible to predict how the quantities of interest evolve and interact with one another. A drawback to this approach is the fact that complex physical phenomena give rise to complex mathematical equations that cannot be solved analytically, i.e., using paper and pencil.

The most detailed models of fluid flow are based on 'first principles', such as the conservation of mass, momentum, and energy. Mathematical equations that embody these fundamental principles have been known for a very long time but used to be practically worthless until numerical methods and digital computers were invented. The second half of the twentieth century has witnessed the advent of *Computational Fluid Dynamics* (CFD), a new branch of applied mathematics that deals with numerical simulation of fluid flows. Nowadays, computer codes based on CFD models are used routinely to predict a variety of increasingly complex flow phenomena.

The quality of simulation results depends on the choice of the model and on the accuracy of the numerical method. In spite of the inevitable numerical and modeling errors, approximate solutions may provide a lot of valuable information at a fraction of the cost that a full-scale experimental investigation would require. Moreover, the sampling of relevant data is free of errors due to a flow disturbance caused by probes. A further advantage of the computational approach is the fact that it can be applied to flows in domains with arbitrarily large or small dimensions under realistic operating conditions. High pressures, toxic chemicals or hot temperatures pose no hazard to a CFD practitioner. Last but not least, simultaneous computation of *instantaneous* density, velocity, pressure, temperature, and concentration *fields* is feasible. Clearly, no experimental technique can capture the evolution of all flow variables throughout the domain. However, experiments are still required to determine the values of input parameters for a mathematical model and to validate the computational results.

The choice of a CFD model is dictated by the nature of the physical process to be simulated, by the objectives of the numerical study, and by the available resources. As a rule of thumb, the mathematical model should be as detailed as possible without making the computations too expensive. The use of a universally applicable model makes it difficult to develop and implement an efficient numerical algorithm. In many cases, the desired information can be obtained using a simplified version that exploits some *a priori* knowledge of the flow pattern or incorporates empirical correlations supported by theoretical or experimental studies. Thus, a hierarchy of fundamental, phenomenological, and empirical models is usually available for particularly difficult problems, such as the numerical simulation of turbulence.

Over the past three decades, the market for CFD software has expanded rapidly, and remarkable progress has been made in the development of numerical algorithms. An astonishing variety of finite difference, finite element, finite volume, and spectral schemes were developed for the equations of fluid mechanics and applied to virtually every flow problem of practical importance. Modern CFD codes are equipped with automatic mesh generation/adaptation tools, reliable error control mechanisms, and efficient iterative solvers for sparse linear systems. Unstructured mesh methods are available for flows in complex geometries. Problems with moving

boundaries and free interfaces can be solved in a fixed or moving reference frame. Parallelization and vectorization make it possible to perform large-scale computations with more than a billion of degrees of freedom. The rapid growth of computing power has stimulated implementation of sophisticated models and extended the range of possible applications to problems as complex as turbulent multiphase flows and fluid-structure interaction. Nowadays, 3D simulations of unsteady transport processes can be performed on a laptop or desktop computer, whereas supercomputers were required to simulate steady 2D problems a couple of decades ago.

If something sounds too good to be true, it probably is. In spite of the above-mentioned recent advances, there is still a lot of room of improvement when it comes to *reliable* simulation of transport phenomena. The user of a commercial CFD code might be unaware of the numerous subtleties, trade-offs, compromises, and *ad hoc* tricks involved in the computation of beautiful colorful pictures. Usually, there is no guarantee that these pictures are quantitatively correct. If the same problem is solved using another mesh, another time step, and/or another numerical scheme, then a qualitatively different solution may be obtained. Hence, the results of a CFD simulation should not be taken at their face value even if they look 'nice' and plausible. In other cases, the approximate solution may exhibit spurious oscillations and/or assume nonphysical negative values. This behavior is typical of problems with discontinuities and steep fronts that cannot be resolved properly on a given mesh. Therefore, it might be necessary to refine the mesh and/or adjust the coefficients of the numerical scheme if nonphysical solution behavior is detected. Ideally, the numerical algorithm should do it automatically by adapting itself to the nature of the problem at hand so as to compute accurate solutions in an efficient way. The goal of this book is to present a general approach to the design of such algorithms.

1.2 Mathematics of Transport Phenomena

In Part I, we dwell on the numerical treatment of differential equations that govern the evolution of scalar fluid properties. The derivation of these equations is usually based on certain conservation principles, as applied to an arbitrary *control volume* $V \subset \mathbb{R}^d$, where $d = 1, 2$, or 3 is the number of space dimensions. If the fluid is in motion, it may flow in and out across the *control surface* S which forms the boundary of V, see Fig. 1. Individual molecules may travel across the interface even if the fluid is at rest. Therefore, the physical and chemical properties of the fluid inside V are influenced by those of the surrounding medium. Moreover, some quantities, such as mass, momentum, and energy, are *conserved*. That is, they may move from one place to another but cannot emerge out of nothing or disappear spontaneously. The physical forces that transport, produce or destroy these quantities are well-known, and reliable mathematical models are available. Thus, conservation principles can be expressed in terms of differential equations that describe all relevant transport mechanisms, such as *convection* (also called *advection*), *diffusion*, and *dispersion*.

1.2.1 Conservation Principles

Let $c(\mathbf{x},t) \in \mathbb{R}$ denote the concentration (amount per unit mass) of a scalar conserved quantity at point $\mathbf{x} \in V$ and time $t \geq 0$. The corresponding concentration per unit volume is given by $u = \rho c$, where ρ is the density of the carrier fluid. The total amount of the conserved variable inside V is given by the volume integral

$$\int_V u(\mathbf{x},t)\,\mathrm{d}\mathbf{x} = \int_V \rho(\mathbf{x},t)c(\mathbf{x},t)\,\mathrm{d}\mathbf{x}. \tag{1.1}$$

We will call this integral the *mass* and speak of mass conservation even if c represents the energy, a single velocity component, or another dimensional quantity.

Obviously, the variation of (1.1) depends on the rate at which c enters or leaves V through the boundary S. This rate is called the *flux* and denoted by

$$\mathbf{f}(\mathbf{x},t) = (f^1,\ldots,f^d),$$

where f^k corresponds to the rate of transport in the $k-$th coordinate direction, per unit area and time. If $\mathbf{ds} = \mathbf{n}\,\mathrm{d}s$ is an infinitesimally small patch of S with the unit outward normal $\mathbf{n}$, then the mass crossing this patch per unit time is $\mathbf{f}\cdot\mathbf{n}\,\mathrm{d}s$.

In the simplest case, the flux vector $\mathbf{f}$ is a linear function of u and/or $\rho\nabla c$, where

$$\nabla = \left(\frac{\partial}{\partial x_1},\ldots,\frac{\partial}{\partial x_d}\right)^T$$

is the vector of partial derivatives that defines the gradient and divergence operators.

Chemical reactions, heating, cooling, and similar processes give rise to interior sources or sinks that generate $s(\mathbf{x},t)$ units of mass per unit volume and time. Thus, the temporal variation of (1.1) satisfies an integral conservation law of the form

$$\frac{\partial}{\partial t}\int_V u(\mathbf{x},t)\,\mathrm{d}\mathbf{x} + \int_S \mathbf{f}\cdot\mathbf{n}\,\mathrm{d}s = \int_V s(\mathbf{x},t)\,\mathrm{d}\mathbf{x}. \tag{1.2}$$

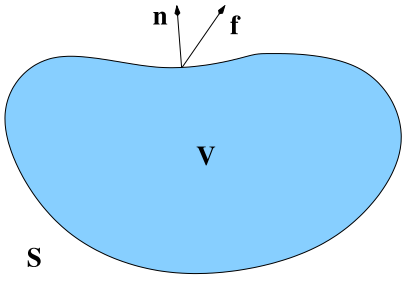


Fig. 1.1 A fixed control volume V bounded by the control surface S.

The surface integral is the mass that leaves V per unit area and time, whereas the right-hand side of (1.2) corresponds to the mass produced inside V per unit time.

If the functions $u(\mathbf{x},t)$ and $\mathbf{f}(\mathbf{x},t)$ are differentiable, then the divergence theorem, as applied to the surface integral in (1.2), yields the identity

$$\int_V \left[\frac{\partial u(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{f}(\mathbf{x},t) - s(\mathbf{x},t) \right] \mathrm{d}\mathbf{x} = 0.$$

Since the choice of V is arbitrary, the expression in the square brackets must vanish, so the evolution of $u(\mathbf{x},t)$ is governed by the partial differential equation (PDE)

$$\frac{\partial u(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{f}(\mathbf{x},t) = s(\mathbf{x},t). \tag{1.3}$$

If the divergence theorem is applicable, this *differential form* of the conservation law is equivalent to the underlying *integral form* (1.2). However, the latter is more fundamental since it does not contain any space derivatives. If the flux $\mathbf{f}(\mathbf{x},t)$ does not depend on the gradients of c, then the generalized solution may exhibit very steep gradients or even discontinuities. Such solutions satisfy (1.2) but not (1.3) since discontinuous functions are not differentiable in the classical sense.

1.2.2 Convective and Diffusive Fluxes

The modeling of the flux function $\mathbf{f}$ should reflect the nature of the involved transport processes. Convective effects arise when fluids flow and transport the quantities of interest downstream. For example, consider a horizontal pipe filled with water which flows from left to right at constant speed. In experimental studies, the flow pattern is commonly visualized by tracking a set of small tracer particles which are convected with the flow as time goes on. Suppose that some particles are white, while others are black. The distribution of particles at time $t = 0$ is displayed in Fig. 1.2a. Since the water is in motion, it carries the suspension of tracer particles towards the outlet. If we keep injecting black particles at the left end of the pipe,

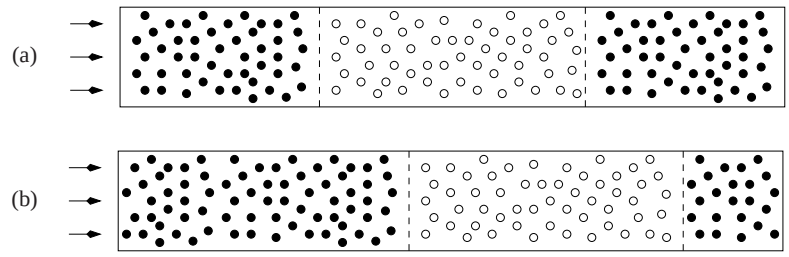


Fig. 1.2 Transport of tracer particles in a pipe filled with moving water.

a snapshot of the particle distribution at a later time $t > 0$ might look as depicted in Fig. 1.2b. Similarly, if we vary the temperature of the water at the inlet, this will affect the temperature distribution inside the pipe and, eventually, at the outlet. This effect is utilized in many heating and cooling devices that we use in everyday life.

Mass and heat may also be transported from one place to another by diffusion or heat conduction, respectively. Random molecular motion induces diffusive fluxes even if the fluid is at rest. To illustrate this process, consider a tank filled with liquid in which two distinct chemical species are dissolved. In Fig. 1.3, the black and white circles represent the molecules of species A and B, respectively. Initially, these species are separated by a diaphragm that divides the tank into two parts (Fig. 1.3a). When the diaphragm is removed, some white molecules may cross the interface and end up in the left half of the tank. Conversely, black ones may travel in the opposite direction and end up in the right half (Fig. 1.3b). After a certain time, the mixture will become homogeneous, and each half of the tank will contain the same number of black and white molecules (Fig. 1.3c). In the context of central heating, convection transports hot water into the radiator but heat transfer inside the room is of a diffusive nature since it is driven primarily by the temperature gradients.

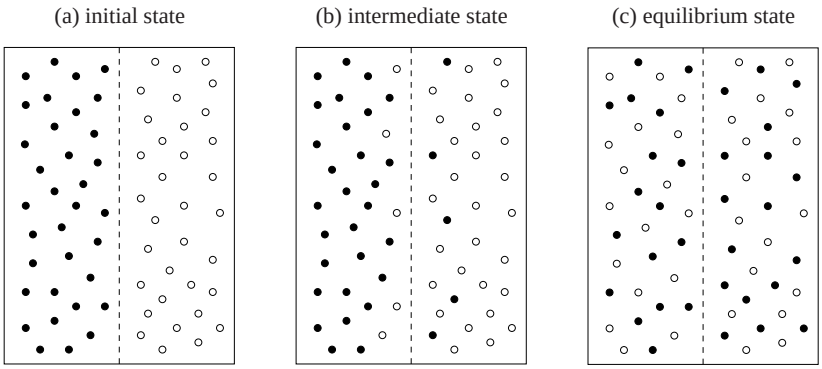


Fig. 1.3 Random motion of molecules across an interface in a stationary liquid.

In general, the transport of conserved quantities from regions of high concentration into regions of low concentration may be caused by random molecular motion or turbulence. Molecular diffusion represents the natural tendency of a physical system towards an equilibrium, whereas turbulent dispersion is due to unresolved eddies that enhance the macroscopic mixing rate. The corresponding mathematical models look the same but the coefficients differ by orders of magnitude. In what follows, both molecular and turbulent mixing will be referred to as 'diffusion.'

Let us now describe the above transport processes in terms of formulas rather than words. Assume that the velocity field $\mathbf{v}(\mathbf{x},t)$ is known. The volume of fluid that crosses an infinitesimally small patch $\mathbf{ds} = \mathbf{n}\,\mathrm{d}s$ during a short time interval $\mathrm{d}t$ is

$$\mathrm{d}V = (\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}s)\mathrm{d}t.$$

Since $u = \rho c$ is the mass of a conserved quantity per unit volume, the amount of mass transported in the normal direction $\mathbf{n}$ per unit area and time is given by

$$\frac{u\,\mathrm{d}V}{\mathrm{d}s\mathrm{d}t} = (\mathbf{v}\cdot\mathbf{n})u.$$

If $\mathbf{n}$ is taken to be the unit vector along the coordinate direction x_d, the above expression yields the $d-$th component of the convective flux

$$\mathbf{f}_C = \mathbf{v}(\mathbf{x},t)u. \tag{1.4}$$

Since diffusion-like processes are driven by the gradients of the concentration field, a typical model for the corresponding flux vector is as follows

$$\mathbf{f}_D = -\mathscr{D}(\mathbf{x},t)\rho\nabla c, \tag{1.5}$$

where $\mathscr{D} = \{d_{ij}\}$ is a symmetric positive definite matrix of diffusion coefficients. If $\mathscr{D} = dI$, where I is the $D\times D$ identity matrix, then the scalar diffusivity $d(\mathbf{x},t) > 0$ is the same for all coordinate directions, and the diffusive flux reduces to

$$\mathbf{f}_D(\mathbf{x},t) = -d(\mathbf{x},t)\rho\nabla c. \tag{1.6}$$

In the realm of mass and heat transfer, this definition of $\mathbf{f}_D$ follows from *Fick's law* of mass diffusion and *Fourier's law* of heat conduction, respectively.

In general, both convective and diffusive effects must be taken into account, so

$$\mathbf{f}(\mathbf{x},t) = \mathbf{v}(\mathbf{x},t)u - \mathscr{D}(\mathbf{x},t)\rho\nabla c. \tag{1.7}$$

However, the rates of convective and diffusive transport may be quite different. For example, the transport of pollutants in a river is dominated by convection, whereas the spreading of pollutants in a lake is dominated by diffusion (dispersion).

The relative strength of $\mathbf{f}_C$ and $\mathbf{f}_D$ can be expressed in terms of the *Peclet number*

$$\mathrm{Pe} = \frac{v_0 L_0}{d_0}, \tag{1.8}$$

where v_0 is a reference velocity, L_0 is a geometric length scale, and d_0 is a diffusion coefficient. The dimensionless Peclet number is infinite in the limit of pure convection ($\mathbf{f} = \mathbf{f}_C,\ \mathscr{D} = 0$) and vanishes in the limit of pure diffusion ($\mathbf{f} = \mathbf{f}_D,\ \mathbf{v} = 0$).

1.2.3 The Generic Transport Equation

Substitution of $u = \rho c$ and (1.7) into (1.3) yields the *generic transport equation*

$$\frac{\partial \rho c}{\partial t} + \nabla\cdot(\mathbf{v}\rho c) - \nabla\cdot(\mathscr{D}\rho\nabla c) = s. \tag{1.9}$$

The terms that appear in this equation admit the following physical interpretation

- the rate-of-change term $\frac{\partial \rho c}{\partial t}$ is the net gain/loss of mass per unit volume and time;
- the convective term $\nabla \cdot (\mathbf{v}\rho c)$ is due to the downstream transport with velocity $\mathbf{v}$;
- the diffusive term $-\nabla \cdot (\mathscr{D}\rho \nabla c)$ is due to a nonuniform spatial distribution of c;
- the source or sink term s combines all other effects that create or destroy ρc.

For the time being, we assume that the parameters ρ, $\mathbf{v}$, $\mathscr{D}$, and s are known. In real-life applications, they may depend on the concentration c and/or other variables.

In particular, conservation laws of the form (1.9) constitute the *Navier-Stokes equations*, in which the conserved variables are the mass, momentum, and total energy. The simplest component of this PDE system is the *continuity equation*

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{1.10}$$

which is responsible for mass conservation and corresponds to (1.9) with $c \equiv 1$ and $s = 0$. Note that the diffusive term vanishes since the gradient of c is zero. If viscosity and heat conduction are neglected, then the Navier-Stokes equations reduce to the *Euler equations* that describe inviscid gas flows at high speeds. Advanced CFD models based on the Euler and Navier-Stokes equations will be treated in Part II.

The common structure of mathematical models which are based on (systems of) scalar conservation laws of the form (1.9) suggests a systematic approach to analysis, discretization, and coding. This strategy facilitates the development, implementation, and testing of numerical methods for advanced CFD applications. In addition to the conceptual and algorithmic simplicity, it offers a simple way to investigate the solution behavior in important limiting cases (steady state, pure convection, pure diffusion etc.) and design simple test problems that can be solved analytically.

The generic transport equation (1.9) can also be written in terms of $u = \rho c$. If the density ρ is constant or the velocity is redefined as $\mathbf{v} := \mathbf{v} + (\mathscr{D}\nabla\rho)/\rho$, then (1.9) is a *convection-diffusion-reaction* (CDR) equation for the mass variable u

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) - \nabla \cdot (\mathscr{D}\nabla u) = s. \tag{1.11}$$

This equation and some simplifications thereof will serve as basic models in Part I.

If the velocity field $\mathbf{v}$ is incompressible, that is, $\nabla \cdot \mathbf{v} = 0$, then the vector identity

$$\nabla \cdot (\mathbf{v}u) = \mathbf{v} \cdot \nabla u + (\nabla \cdot \mathbf{v})u \tag{1.12}$$

makes it possible to write the left-hand side of (1.11) in the nondivergent form

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u - \nabla \cdot (\mathscr{D}\nabla u) = s. \tag{1.13}$$

Another possibility is to take the average of (1.11) and (1.13), which gives a skew-symmetric form of the convective term. All three formulations are equivalent for divergence-free velocity fields but only (1.11) is conservative for $\nabla \cdot \mathbf{v} \neq 0$.

1.2.4 Initial and Boundary Conditions

The same differential equation may describe an amazing variety of flow patterns, so some additional information is required to complete the problem statement. In practical applications, the processes to be investigated take place in a concrete geometry (e.g., in turbines, chemical reactors, heat exchangers, car engines etc.) during a finite interval of time. The choice of the domain and of the time interval to be considered is dictated by the nature of the problem at hand, by the objectives of the analytical or numerical study, and by the available resources. Another important aspect is the choice of initial and/or boundary conditions that lead to a well-posed problem.

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain and $(0,T)$ be the time interval of interest. In general, the boundary Γ of Ω may consist of an inflow part $\Gamma_- = \{\mathbf{x} \in \Gamma \,|\, \mathbf{v}\cdot\mathbf{n} < 0\}$, an outflow part $\Gamma_+ = \{\mathbf{x} \in \Gamma \,|\, \mathbf{v}\cdot\mathbf{n} > 0\}$, and a solid wall $\Gamma_0 = \{\mathbf{x} \in \Gamma \,|\, \mathbf{v}\cdot\mathbf{n} = 0\}$, where $\mathbf{n}$ denotes the unit outward normal to the boundary at the point $\mathbf{x} \in \Gamma$.

Since the CDR equation contains a time derivative, it must be supplemented by an initial condition that defines the distribution of mass at $t = 0$

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \qquad \forall \mathbf{x} \in \Omega. \tag{1.14}$$

Furthermore, the fluid inside Ω interacts with the surrounding medium, so it is also necessary to prescribe suitable boundary conditions on Γ. If the values of u are known on $\Gamma_D \subset \Gamma$, they can be imposed as *Dirichlet boundary conditions*

$$u(\mathbf{x},t) = u_D(\mathbf{x},t), \qquad \forall \mathbf{x} \in \Gamma_D,\ \forall t \in (0,T). \tag{1.15}$$

As a rule, this boundary condition is used at the inlet Γ_- and/or on the solid wall Γ_0.

Alternatively, a given normal flux may be prescribed on the complementary boundary part $\Gamma_N = \Gamma \backslash \Gamma_D$. The so-defined *Neumann boundary condition* reads

$$\mathbf{f}\cdot\mathbf{n} = g(\mathbf{x},t), \qquad \forall \mathbf{x} \in \Gamma_N,\ \forall t \in (0,T). \tag{1.16}$$

The involved flux $\mathbf{f}$ may consist of a convective and/or a diffusive part, depending on the information available. If $\mathbf{f} = \mathbf{f}_C$ or the diffusive flux $\mathbf{f}_D$ is required to vanish, then the right-hand side of (1.16) is given by $g = (\mathbf{v}\cdot\mathbf{n})u$ on $\Gamma_\pm$ and $g = 0$ on Γ_0.

1.2.5 Weighted Residual Formulation

The classical solution u of the CDR equation must belong to the space of functions which are continuous with continuous partial derivatives of first and second order. In other words, it must be very smooth. In order to broaden the class of admissible functions, it is worthwhile to consider an integral or *weak* form of the conservation law. The corresponding generalized solution is supposed to satisfy the strong form of (1.11) for sufficiently smooth data but exist even if the divergence theorem is not applicable and the underlying conservation law holds only in an integral sense.

A very general approach to the derivation of weak forms for a given PDE is called the *method of weighted residuals*. The residual of equation (1.11) is defined as

$$\mathscr{R}(\bar{u}) = \frac{\partial \bar{u}}{\partial t} + \nabla \cdot (\mathbf{v}\bar{u}) - \nabla \cdot (\mathscr{D}\nabla \bar{u}) - s \tag{1.17}$$

so that $\mathscr{R}(u) = 0$ if u is the exact solution of the CDR equation. Thus, the magnitude of the residual $\mathscr{R}(\bar{u})$ measures the accuracy of an approximate solution $\bar{u} \approx u$.

Obviously, a zero residual remains unchanged if we multiply it by a suitable *weighting* (or *test*) function and integrate over the domain of interest. Hence,

$$\int_\Omega w\mathscr{R}(u)\,\mathrm{d}\mathbf{x} = 0, \qquad \forall w \in \mathscr{W}, \tag{1.18}$$

where $\mathscr{W}$ is a space of weighting functions vanishing on Γ_D. Mathematically speaking, the residual $\mathscr{R}(u)$ must be orthogonal to all $w \in \mathscr{W}$. The weak solution u resides in a space $\mathscr{V}$ of functions satisfying the Dirichlet boundary conditions (1.15).

Since the residual of (1.11) is given by (1.17), the associated weak form reads

$$\int_\Omega w\left(\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) - \nabla \cdot (\mathscr{D}\nabla u) - s\right)\mathrm{d}\mathbf{x} = 0, \qquad \forall w \in \mathscr{W}. \tag{1.19}$$

If the number of test functions is infinite, formulations (1.19) and (1.11) are equivalent. Otherwise, the residual $\mathscr{R}(u)$ may be nonzero even if u satisfies (1.19).

The rationale for the use of a weighted residual formulation is the possibility to shift some derivatives onto the test function w using integration by parts. The Green's formula, as applied to the diffusive term in (1.19), yields

$$\int_\Omega \left(w\frac{\partial u}{\partial t} + w\nabla \cdot (\mathbf{v}u) + \nabla w \cdot (\mathscr{D}\nabla u) - ws\right)\mathrm{d}\mathbf{x} - \int_{\Gamma_N} w(\mathscr{D}\nabla u)\cdot\mathbf{n}\,\mathrm{d}s = 0. \tag{1.20}$$

Since $w = 0$ on Γ_D, the surface integral is taken over the boundary part $\Gamma_N = \Gamma \backslash \Gamma_D$.

The functions $u \in \mathscr{V}$ and $w \in \mathscr{W}$ are required to possess generalized derivatives of first order. If the convective term is also integrated by parts, one obtains

$$\int_\Omega \left(w\frac{\partial u}{\partial t} - \nabla w \cdot (\mathbf{v}u - \mathscr{D}\nabla u) - ws\right)\mathrm{d}\mathbf{x} + \int_{\Gamma_N} w(\mathbf{v}u - \mathscr{D}\nabla u)\cdot\mathbf{n}\,\mathrm{d}s = 0. \tag{1.21}$$

The surface integrals that pop up in (1.20) and (1.21) contain the normal components of the diffusive and total flux, respectively. Since Neumann boundary conditions of the form (1.16) are prescribed on Γ_N, the corresponding integral is given by

$$\int_{\Gamma_N} w\mathbf{f}\cdot\mathbf{n}\,\mathrm{d}s = \int_{\Gamma_N} wg\,\mathrm{d}s. \tag{1.22}$$

Thus, flux boundary conditions fit naturally into the weak form of the transport equation. Dirichlet boundary conditions (1.15) are imposed in a strong sense, i.e., they are built into the definition of the spaces $\mathscr{V}$ and $\mathscr{W}$ in which u and w reside.

The elegance and generality of the weighted residual method lies in the choice of the test functions w. For example, if we substitute $w \equiv 1$ and $\Gamma_N = \Gamma$ into (1.21), then the integral form (1.2) of the CDR equation is recovered

$$\frac{\partial}{\partial t}\int_\Omega u\,\mathrm{d}\mathbf{x} + \int_\Gamma (\mathbf{v}u - \mathscr{D}\nabla u)\cdot\mathbf{n}\,\mathrm{d}s = \int_\Omega s\,\mathrm{d}\mathbf{x}. \tag{1.23}$$

On the other hand, it is possible to enforce (1.11) in a strong sense by setting the residual to zero at a *collocation point* $\mathbf{x}_0 \in \Omega$. To this end, we substitute the Dirac delta function $w(\mathbf{x}) = \delta(\mathbf{x}-\mathbf{x}_0)$ into (1.19) and obtain the pointwise identity

$$\frac{\partial u}{\partial t} + \nabla\cdot(\mathbf{v}u) - \nabla\cdot(\mathscr{D}\nabla u) = s \qquad \text{at} \quad \mathbf{x} = \mathbf{x}_0.$$

Hence, the weighted residual formulation unites the integral, differential, and weak forms of the conservation law. The existence of several equivalent representations, as presented above, makes it possible to choose the one which is easier to handle for a given choice of functional spaces and boundary conditions. This flexibility turns out to be very useful when it comes to the design of numerical approximations.

1.3 Taxonomy of Reduced Models

The behavior of analytical and numerical solutions to the strong or weak form of (1.11) depends on the interplay of the four terms that appear in this equation. The time derivative may be large for transient transient processes but vanish in the steady-state limit. The relative importance of convective and diffusive effects depends on the Peclet number (1.8). The reactive and diffusive terms are zero in the continuity equation (1.10) but may be dominant in other transport models.

Many useful model problems can be constructed on the basis of the CDR equation by omitting some terms and/or making additional assumptions. These simplifications may affect the type of the partial differential equation, the choice of initial and boundary conditions, the qualitative properties of exact solutions, and the performance of numerical schemes. A good algorithm must be sufficiently robust, accurate, and efficient for all possible manifestations of the transport equation.

1.3.1 Elliptic Transport Equations

If the convective and diffusive fluxes are in equilibrium with the source term, then the time derivative of the transported quantity vanishes, and (1.11) reduces to

$$\nabla\cdot(\mathbf{v}u - \mathscr{D}\nabla u) = s \qquad \text{in } \Omega. \tag{1.24}$$

This second-order PDE is of *elliptic* type provided that the matrix of diffusion coefficients $\mathscr{D}(\mathbf{x})$ is symmetric positive definite for all $\mathbf{x} \in \Omega$. In elliptic problems, information propagates in all directions at infinite speed. The variation of u at any point $\mathbf{x}_1 \in \Omega$ may influence the solution at any other point $\mathbf{x}_2 \in \Omega$ and vice versa. Boundary conditions of Dirichlet or Neumann type are to be prescribed on $\Gamma = \Gamma_D \cup \Gamma_N$, whereas no initial conditions are required for stationary problems like (1.24).

Elliptic CDR equations describe equilibrium transport phenomena and may represent the steady-state limit of a transient process. Indeed, if the velocity field, the diffusion coefficients, and the boundary conditions do not depend on time, then the solution of (1.11) will eventually become stationary and satisfy equation (1.24).

Steady diffusion-reaction processes are described by equation (1.24) with $\mathbf{v} = 0$

$$-\nabla \cdot (\mathscr{D}\nabla u) = s \qquad \text{in } \Omega. \tag{1.25}$$

Elliptic PDEs of this form play an important role, e.g., in mathematical modeling of flows in porous media. In this context, the relationship $\mathbf{v} = -\mathscr{D}\nabla u$ is called the *Darcy law*, in which $\mathbf{v}$ and u represent the velocity and pressure fields, respectively.

If the diffusion tensor is defined as $\mathscr{D} = dI$, where d is a constant diffusion coefficient, then (1.25) divided by d yields the following *Poisson equation*

$$-\Delta u = f, \qquad \text{in } \Omega, \tag{1.26}$$

where $\Delta = \nabla^2$ denotes the Laplacian operator and $f = s/d$. The *Laplace equation*

$$\Delta u = 0, \qquad \text{in } \Omega \tag{1.27}$$

is recovered for $f = 0$. In particular, it can be used to compute the potential of the velocity field $\mathbf{v} = -\nabla u$ for incompressible irrotational flows ($\nabla \cdot \mathbf{v} = 0,\ \nabla \times \mathbf{v} = 0$).

1.3.2 Hyperbolic Transport Equations

The next model problem to be considered is that of purely convective transport. For $\mathscr{D} = 0$, equation (1.24) degenerates into a *hyperbolic* PDE of first order

$$\nabla \cdot (\mathbf{v}u) = s \qquad \text{in } \Omega. \tag{1.28}$$

In this case, information is transported at finite speeds along the streamlines of the stationary velocity field $\mathbf{v}(\mathbf{x})$. The nature of hyperbolic problems requires that boundary conditions be specified only on the inflow part Γ_-, where $\mathbf{v}\cdot\mathbf{n} < 0$. It would be inappropriate and incorrect to prescribe any boundary conditions elsewhere.

The unsteady version of the convection-reaction equation (1.28) is given by

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = s \qquad \text{in } \Omega \times (0,T). \tag{1.29}$$

The most prominent example is the continuity equation (1.10) with $u = \rho$ and $s = 0$.

Like any other PDE of first order, equation (1.29) is of hyperbolic type. The direction and speed of convective transport depend on the velocity field $\mathbf{v}(\mathbf{x},t)$. As before, boundary conditions are to be prescribed only at the inlet Γ_-. The initial condition is given by (1.14), and information travels forward in time. That is, the distribution of u at any time instant $\bar{t}$ depends on the previous evolution history but only the solution at a later time may be influenced by what happens at $t = \bar{t}$.

Analytical solutions to (1.28) and (1.29) can be constructed by the *method of characteristics* to be presented in subsequent chapters. Due to the lack of diffusive effects, hyperbolic conservation laws admit discontinuous and, possibly, nonunique weak solutions. Such problems are particularly difficult to solve numerically, although a lot of information about the properties of exact solutions is available.

1.3.3 Parabolic Transport Equations

If the fluid is at rest, then the contribution of the convective term to the CDR equation (1.11) vanishes, so the evolution of u is driven by diffusion and reaction

$$\frac{\partial u}{\partial t} - \nabla \cdot (\mathscr{D}\nabla u) = s \qquad \text{in } \Omega \times (0,T). \tag{1.30}$$

If diffusion is isotropic then $\mathscr{D}\nabla u = d\nabla u$ and the above equation assumes the form

$$\frac{\partial u}{\partial t} - d\Delta u = s \qquad \text{in } \Omega \times (0,T). \tag{1.31}$$

This model describes unsteady transport processes like mass diffusion or heat conduction. The redistribution of u continues until the time derivative vanishes, and the solution of the elliptic Poisson equation (1.26) is obtained in the steady state limit.

Unsteady partial differential equations of second order are of *parabolic* type if their stationary counterparts are elliptic. Both initial and boundary conditions of the form (1.14)–(1.16) are required for time-dependent transport models based on the parabolic equations (1.11), (1.30), and (1.31). As in the case of hyperbolic PDEs, information propagates forward in time, and there is no backward influence.

Steady CDR equations with $\mathbf{v} \neq 0$ can also be parabolic if there is a predominant flow direction, and the diffusive transport in this direction is neglected. For example, let $\mathbf{v} = (1,0,0)$ and $\mathscr{D} = \mathrm{diag}\{0,d,d\}$. Then equation (1.24) can be written as

$$\frac{\partial u}{\partial x_1} - d\left(\frac{\partial^2 u}{\partial x_2^2} + \frac{\partial^2 u}{\partial x_3^2}\right) = s, \tag{1.32}$$

where x_1 represents a *time-like coordinate* such that information is convected downstream, and no recirculation takes place. This problem is parabolic and exhibits the same structure as the unsteady diffusion-reaction equation (1.30)–(1.27) in 2D.

1.3.4 Summary of Model Problems

As we have seen, the CDR equation (1.11) represents a rich variety of model problems. If the time derivative, convection, or diffusion are neglected, the resulting equation looks simpler than (1.11). Ironically, it may be more difficult to treat numerically. In particular, computation of stationary solutions is hard, unless a good initial guess is available. Therefore, it is common practice to march solutions to the steady state by solving the corresponding unsteady PDE subject to arbitrary initial conditions. Similarly, the lack of diffusive terms in hyperbolic models may adversely affect the performance of a numerical scheme designed to solve (1.11).

Partial differential equations and numerical methods are easier to analyze in one space dimension. In the 1D case, the operator ∇ reduces to the partial derivative with respect to x, the matrix $\mathscr{D}$ degenerates into a scalar diffusion coefficient d, and the unsteady convection-reaction-diffusion equation (1.11) assumes the form

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} - d\frac{\partial^2 u}{\partial x^2} = s.$$

Again, it can be used to generate model problems for the whole range of possible PDE types. The taxonomy of reduced transport models is summarized in Table 1.1. If reaction is not important, we set $s = 0$. This does not change the PDE type. The presented models will help us to develop, evaluate, and compare numerical solution techniques. A detailed analysis of each model will be performed in Chapter 3.

Table 1.1 Summary of models for convection, diffusion, and reaction processes.

PDE type	multidimensional	one-dimensional
elliptic	$\nabla\cdot(\mathbf{v}u - \mathscr{D}\nabla u) = s$	$v\frac{\partial u}{\partial x} - d\frac{\partial^2 u}{\partial x^2} = s$
	$-\nabla\cdot(\mathscr{D}\nabla u) = s$	$-d\frac{\partial^2 u}{\partial x^2} = s$
hyperbolic	$\nabla\cdot(\mathbf{v}u) = s$	$v\frac{\partial u}{\partial x} = s$
	$\frac{\partial u}{\partial t} + \nabla\cdot(\mathbf{v}u) = s$	$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = s$
parabolic	$\frac{\partial u}{\partial t} - \nabla\cdot(\mathscr{D}\nabla u) = s$	$\frac{\partial u}{\partial t} - d\frac{\partial^2 u}{\partial x^2} = s$
	$\frac{\partial u}{\partial t} + \nabla\cdot(\mathbf{v}u - \mathscr{D}\nabla u) = s$	$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} - d\frac{\partial^2 u}{\partial x^2} = s$

1.4 Space Discretization Techniques

Computers are of little help in obtaining closed-form analytical solutions to a PDE model. However, they can be programmed to solve algebraic equations very fast. Replacing calculus by algebra, it is possible to compute approximate solutions to the CDR equation and more advanced mathematical models. To this end, the computational domain, the unknown solution, and its partial derivatives need to be *discretized*, so as to obtain a set of algebraic equations for the function values at a finite number of discrete locations. We will begin with the discretization in space and discuss time-stepping techniques for unsteady PDEs in Section 1.5.1.

1.4.1 Computational Meshes

Recall that the integral conservation law (1.2) which has led us to (1.9) and (1.11) was formulated for a fixed control volume (CV) of finite size. Instead of looking at the whole flow field at once, we have focused our attention on what is happening in a small subdomain. A similar approach is used to discretize differential equations that embody physical conservation principles. The unknowns of the discrete problem are associated with a computational *mesh* or *grid* which represents a subdivision of the domain $\Omega \subset \mathbb{R}^d$ into many small control volumes Ω_k (e.g., intervals in 1D, triangles or quadrilaterals in 2D, tetrahedra or hexahedra in 3D) such that $\bar{\Omega} \approx \bigcup_k \bar{\Omega}_k$.

Many excellent texts are devoted to automatic generation and adaptation of computational meshes, see [55, 115, 118, 226, 318]. Mesh generation is easy for domains of rectangular shape but difficult in the case of curvilinear boundaries, internal obstacles, and small-scale features. Depending on the geometric complexity of Ω, the mesh may be structured, block-structured, or unstructured (see Fig. 1.4).

In the one-dimensional case, the computational domain $\Omega = (a,b)$ is an interval. A subdivision of this interval into N subintervals $\Omega_k = (x_{k-1}, x_k)$ of equal size

$$\Delta x = \frac{b-a}{N}$$

yields the simplest representative of structured meshes. The $N+1$ grid points

$$x_i = i\Delta x, \qquad \forall i = 0,1,\ldots,N \tag{1.33}$$

are numbered from left to right. Each interior grid point x_i has two nearest neighbors whose indices $i \pm 1$ and coordinates $x_{i\pm1}$ are known. The spacing $\Delta x_k = x_k - x_{k-1}$ can also be nonuniform if higher mesh resolution is desired in some regions.

In multidimensions, a structured mesh is a net of grid lines (Fig.1.4, a–c) which can be numbered consecutively. In the simplest case, formula (1.33) is used to discretize each coordinate axis. Again, the search for nearest neighbors is easy, and their number is the same for each interior point. The involved data structures and numerical algorithms are almost as simple as in the one-dimensional case. A nonuni-

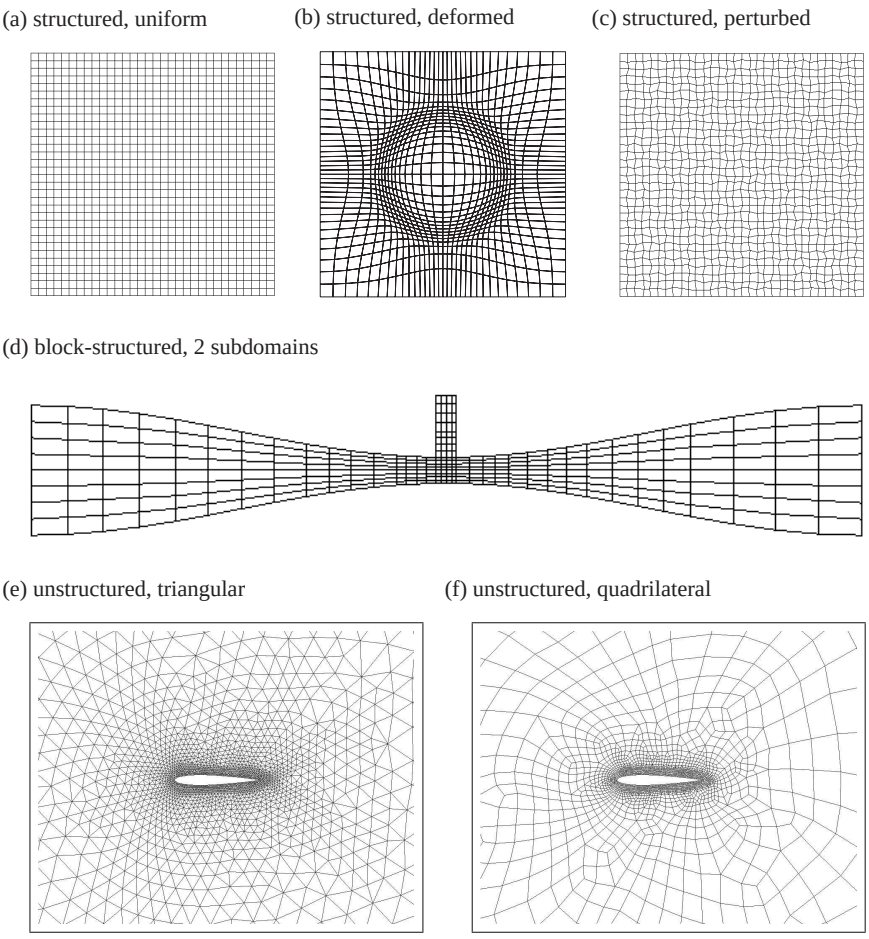


Fig. 1.4 Examples of computational meshes for two-dimensional domains.

form body-fitted mesh can be generated and mapped onto a uniform Cartesian grid [4, 318]. However, since all grid lines must begin and end on the boundary, an attempt to obtain higher resolution in zones of particular interest may entail unintended and, sometimes, harmful mesh refinement in other parts of the domain [104].

The generation of a block-structured mesh is based on a two-level subdivision, whereby a number of overlapping or nonoverlapping subdomains (blocks) are discretized using structured meshes [55, 104]. Figure 1.4 (d) displays such a mesh that consists of two components. The use of multiple blocks makes it easier to deal with nonrectangular domains and moving objects. Domain decomposition methods [55, 143, 209, 210, 277] can be employed to distribute the work between multiple processors in a parallel computing environment. Local mesh refinement and solution algorithms tailored to structured meshes can be applied blockwise but special

care is required to transfer information between the blocks in a conservative manner. It is also possible to use structured grids in some subdomains (e.g., to generate a *Cartesian core* [234] or resolve a boundary layer) and unstructured ones elsewhere.

Domains of particularly complex geometric shape call for the use of a fully unstructured mesh. Recent advances in computational geometry make it possible to generate such meshes automatically for 2D and 3D problems as complex as flow around a car, a submarine, or a space shuttle [226]. Unstructured mesh methods are very flexible and well suited for mesh adaptivity. An arbitrary number of elements are allowed to meet at a single vertex (see Fig.1.4, e–f), so it is easy to insert extra grid points in regions of insufficient mesh resolution. Conversely, it is possible to remove points in regions where a coarser mesh would suffice. Since transport processes move information from one place into another, it is worthwhile to adjust the mesh in the course of simulation, so as to achieve high accuracy at a low cost.

Of course, the flexibility offered by unstructured meshes is not a free lunch since sophisticated data structures are required to handle the irregular connectivity pattern, and data access is rather slow. Moreover, efficient solution methods are more difficult to develop and program than in the case of (block-)structured grids. Nevertheless, most successful general-purpose CFD codes are based on unstructured meshes. We refer to the recent monograph by Löhner [226] for a comprehensive introduction to this approach and a unique collection of state-of-the-art algorithms.

The nodes of the mesh may be fixed or move in a prescribed fashion. The corresponding numerical algorithms can be classified into Eulerian, Lagrangian, and Arbitrary Lagrangian Eulerian (ALE) ones. An Eulerian method is based on a fixed mesh, i.e., the positions of mesh points do not change as time goes on. The nodes of a Lagrangian mesh move with the flow velocity, so that the convective transport is built into the mesh motion and only diffusive fluxes need to be discretized. A major drawback to this approach is the fact that the shape and size of mesh cells cannot be controlled. As a consequence, mesh tangling is possible, unless global or local remeshing is performed on a regular basis. Within the ALE framework, some nodes may remain fixed, while others may move with arbitrary velocities. This is the most general formulation which is often used for simulation of flows with free interfaces and fluid-structure interaction (FSI). Moving meshes are of value for many applications but, for simplicity, only Eulerian methods will be discussed in this book.

1.4.2 Semi-Discrete Problem

Given a suitably designed computational mesh, the continuous function $u(\mathbf{x},t)$ is approximated by a finite number of nodal values $\{u_i\}$ which may be associated with vertices, edges, faces, cells, or control volumes. Depending on the type of approximation, these *degrees of freedom* may represent, e.g., pointwise function values, cell averages, or coefficients of piecewise-polynomial basis functions. If the governing equation models an unsteady process, then the degrees of freedom are time-dependent and should be updated step-by-step, as explained in Section 1.5.1.

The discretization in space is required to obtain a system of equations for the nodal values of the approximate solution. Of course, the number of equations should be the same as the number of unknowns. The only derivative that may still appear in each equation is the one with respect to time. For example, the semi-discretized unsteady CDR equation (1.11) for the nodal value u_i can be written as

$$\sum_j \left(m_{ij} \frac{\mathrm{d}u_j}{\mathrm{d}t} \right) + \sum_j (c_{ij} + d_{ij}) u_j = r_i, \tag{1.34}$$

where the coefficients c_{ij}, d_{ij}, and r_i are due to convection, diffusion, and reaction, respectively. The use of the total derivative notation in the left-hand side of equation (1.34) is appropriate since no space derivatives are present anymore. The weights m_{ij} distribute the gain or loss of mass, if any, between node i and its neighbors.

If $m_{ij} = 0$, $\forall j \neq i$, (1.34) reduces to the ordinary differential equation (ODE)

$$m_{ii} \frac{\mathrm{d}u_i}{\mathrm{d}t} + \sum_j (c_{ij} + d_{ij}) u_j = r_i. \tag{1.35}$$

Since the right-hand sides of (1.34) and (1.35) depend on the solution values at several nodes, the semi-discrete equations must be integrated in time simultaneously.

The matrix form of a space discretization given by (1.34) or (1.35) reads

$$M \frac{\mathrm{d}u}{\mathrm{d}t} + (C + D)u = r, \tag{1.36}$$

where $u = \{u_i\}$ denotes the vector of time-dependent nodal values, $M = \{m_{ij}\}$ is the *mass matrix*, $C = \{c_{ij}\}$ is the discrete transport operator, $D = \{d_{ij}\}$ is the discrete diffusion operator, and $r = \{r_i\}$ is the vector of discretized source or sink terms.

As a rule, the matrices M, C, and D are sparse. That is, most of their entries are equal to zero and do not need to be stored. The sparsity pattern of the discrete operators depends on the type of the underlying mesh (structured or unstructured) and on the numbering of nodes. The mass matrix M is diagonal or symmetric positive definite. Ideally, the discrete diffusion operator D should also be symmetric, as required by the properties of its continuous counterpart [289]. The discrete convection operator C is nonsymmetric since the flow direction must be taken into account. For example, this matrix can be skew-symmetric ($C = -C^T$) or upper/lower triangular.

In the case of a steady governing equation, such as (1.24), the time derivative vanishes, so the semi-discrete problem (1.36) reduces to the algebraic system

$$(C + D)u = r. \tag{1.37}$$

The space discretization of pure convection, pure diffusion, and zero reaction models from Table 1.1 corresponds to $D = 0$, $C = 0$, and $r = 0$ respectively.

Polynomials play an important role in the discretization process since they are easy to differentiate and integrate. The most popular discretization techniques based on polynomial approximations are the finite difference, finite volume, and finite el-

ement method. Spectral and boundary element methods are also worth mentioning but their range of applicability is rather limited, so they will not be discussed in this book. For a general introduction to numerical methods for differential equations, we refer to [76, 104, 149, 276]. In this section, we will introduce some basic discretization concepts and present a concise summary of the approximations involved. An in-depth presentation of numerical schemes tailored to the convection-diffusion-reaction equation and other transport models will follow in subsequent chapters.

1.4.3 Finite Difference Methods

The finite difference method (FDM) is the oldest among the discretization techniques for partial differential equations. Many modern numerical schemes for transport phenomena trace their origins to finite difference approximations developed in the late 1950s through early 1980s. The derivation and implementation of FDM are particularly simple on structured meshes which are topologically equivalent to a uniform Cartesian grid. The nodal value of the approximate solution at node i

$$u_i(t) \approx u(\mathbf{x}_i, t) \tag{1.38}$$

is a pointwise approximation to the true solution of the partial differential equation.

Taylor series expansions or polynomial fitting techniques are used to approximate all space derivatives in terms of u_i and/or solution values at a number of neighboring nodes. For example, if we consider the uniform 1D mesh given by (1.33), then

$$\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x}$$

is a second-order approximation to the first derivative of u at node i, whereas

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2}$$

is a second-order approximation to the second derivative. On a nonuniform mesh, the coefficients are different and must be derived individually for each grid point. Alternatively, a mapping onto a regular Cartesian grid may be employed [4].

After all space derivatives have been approximated by finite differences, the semi-discrete counterpart of the unsteady CDR equation (1.11) can be written in the generic form (1.35) which determines the relationship between u_i and the solution values at a certain number of neighboring nodes. The set of all grid points that make a nonzero contribution to the equation for u_i is called the *stencil*.

The resulting finite difference discretization is of the form (1.36), where M is a diagonal matrix. The coefficients of the matrices C and D depend on the parameters of the model, on the choice of finite difference approximations, and on the mesh size. Global mesh refinement results in higher accuracy but increases the size of the

algebraic system and, consequently, the computational cost. If the mesh is structured, then the stencil of each interior point has the same size, and nearest neighbors are easy to identify provided that the grid lines are numbered consecutively. In the case of an unstructured mesh, the number of neighbors may vary, and the distribution of grid points is nonuniform. Fitting a polynomial to scattered data is feasible but computationally expensive and difficult to implement. For this reason, the use of unstructured meshes is rather uncommon in the realm of finite difference methods.

1.4.4 Finite Volume Methods

Due to the growing demand for numerical simulation of transport processes in 2D and 3D domains of complex shape, the finite difference method has eventually lost its leadership position. Nowadays, general-purpose CFD codes are typically based on the finite volume method (FVM) which yields a finite-difference like approximation on a uniform Cartesian grid but is readily applicable to unstructured meshes.

Finite volume methods for the CDR equation (1.11) are based on the underlying integral conservation law. Inserting the flux $\mathbf{f} = \mathbf{v}u - \mathscr{D}\nabla u$ into (1.2), one obtains

$$\frac{\partial}{\partial t}\int_{V_i} u(\mathbf{x},t)\,\mathrm{d}\mathbf{x} + \int_{S_i} (\mathbf{v}u - \mathscr{D}\nabla u)\cdot\mathbf{n}\,\mathrm{d}s = \int_{V_i} s(\mathbf{x},t)\,\mathrm{d}\mathbf{x}, \tag{1.39}$$

where V_i is a control volume (CV) bounded by the control surface S_i, and $\mathbf{n}$ is the unit outward normal. In *cell-centered* finite volume methods, $V_i = \Omega_i$ is a single cell of the computational mesh, see Fig. 1.5. Alternatively, a dual tessellation can be used to define V_i for a *vertex-centered* finite volume method. In the two-dimensional case, the dual cell V_i around the vertex $\mathbf{x}_i$ can be constructed by joining the midpoints of mesh edges and the centroids of the neighboring cells, as shown in Fig. 1.6.

The outward normals to the interface $S_{ij} = S_i \cap S_j$ between any pair of adjacent control volumes V_i and V_j have opposite signs, so the integrals over internal boundaries cancel out if equations (1.39) are summed over i. Hence, the integral conservation law (1.2) holds not only for all V_i but also for the whole domain $V = \Omega$

$$\frac{\partial}{\partial t}\int_{\Omega} u(\mathbf{x},t)\,\mathrm{d}\mathbf{x} + \int_{\Gamma} (\mathbf{v}u - \mathscr{D}\nabla u)\cdot\mathbf{n}\,\mathrm{d}s = \int_{\Omega} s(\mathbf{x},t)\,\mathrm{d}\mathbf{x}. \tag{1.40}$$

Equations (1.39) and (1.40) express the local and global conservation principles, respectively. The former implies the latter but the reverse is generally not true.

The degrees of freedom for cell-centered or vertex-centered FVM can be defined as mean values over the CVs associated with cells and vertices, respectively. Let

$$u_i(t) = \frac{1}{|V_i|}\int_{V_i} u(\mathbf{x},t)\,\mathrm{d}\mathbf{x}, \tag{1.41}$$

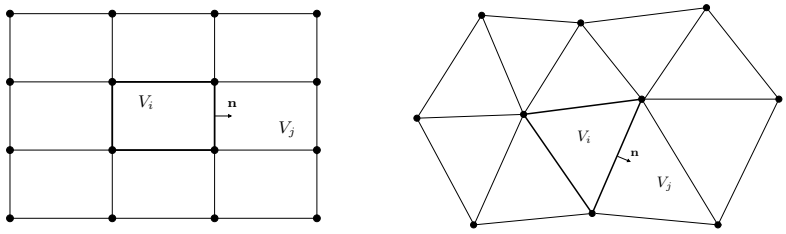


Fig. 1.5 Control volumes for a cell-centered FVM in two dimensions.

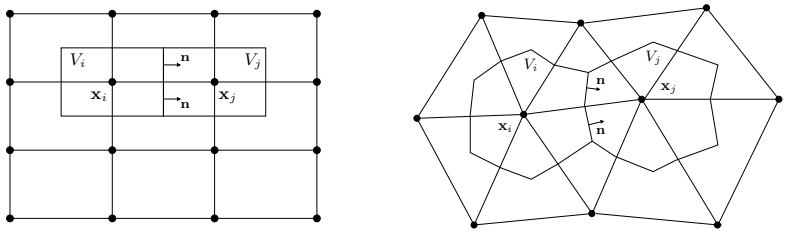


Fig. 1.6 Control volumes for a vertex-centered FVM in two dimensions.

where $|V_i|$ denotes the volume of V_i. According to (1.39), the evolution of the so-defined mean value u_i is governed by the integro-differential equation

$$|V_i|\frac{\mathrm{d}u_i}{\mathrm{d}t} + \int_{S_i} (\mathbf{v}u - \mathscr{D}\nabla u)\cdot\mathbf{n}\,\mathrm{d}s = |V_i|s_i \tag{1.42}$$

in which s_i denotes the average rate of production inside V_i, that is,

$$s_i(t) = \frac{1}{|V_i|}\int_{V_i} s(\mathbf{x},t)\,\mathrm{d}\mathbf{x}. \tag{1.43}$$

By definition, the boundary S_i of the control volume V_i consists of several patches

$$S_i = \bigcup_j S_{ij}. \tag{1.44}$$

For any index $j \neq i$, the patch $S_{ij} = S_i \cap S_j$ is the interface between V_i and one of its neighbors V_j. The index $j = i$ is reserved for boundary patches $S_{ii} = S_i \cap \Gamma$.

Splitting the surface integral into a sum over patches, one can write (1.42) as

$$|V_i|\frac{\mathrm{d}u_i}{\mathrm{d}t} + \sum_j \int_{S_{ij}} (\mathbf{v}u - \mathscr{D}\nabla u)\cdot\mathbf{n}\,\mathrm{d}s = |V_i|s_i. \tag{1.45}$$

If the control volumes for a cell-centered or vertex-centered FVM are defined as shown in Figs. 1.5–1.6, then the normal to S_{ij} is constant or piecewise-constant.

The total flux across the patch S_{ij} of the control surface S_i is given by

$$f_{ij} = \int_{S_{ij}} (\mathbf{v}u - \mathscr{D}\nabla u) \cdot \mathbf{n}\,\mathrm{d}s. \tag{1.46}$$

If the patch S_{ij} is the common boundary of V_i and V_j, then $j \neq i$ and $f_{ji} = -f_{ij}$, so that the mass leaving V_i is equal to the mass entering V_j and vice versa.

Equation (1.45) can be expressed in terms of the integrated fluxes f_{ij} as follows

$$|V_i|\frac{\mathrm{d}u_i}{\mathrm{d}t} + \sum_j f_{ij} = |V_i|s_i \tag{1.47}$$

and transformed into an equation of the form (1.35) with $m_{ii} = |V_i|$ and $r_i = |V_i|s_i$

$$|V_i|\frac{\mathrm{d}u_i}{\mathrm{d}t} + \sum_j (c_{ij} + d_{ij})u_j = |V_i|s_i. \tag{1.48}$$

To this end, it is necessary to approximate the integrated fluxes f_{ij} in terms of the mean values u_i. The derivation of (1.48) involves two levels of approximation [104]

- Surface and volume integrals are approximated using numerical quadrature (cubature) which requires evaluation of the integrand at one or more locations.
- Interpolation techniques are employed to approximate the function values and derivatives at the quadrature points in terms of the primary unknowns u_i.

By the midpoint rule, the mean values $u_i(t)$ and $s_i(t)$ represent second-order accurate approximations to $u(\bar{\mathbf{x}}_i, t)$ and $s(\bar{\mathbf{x}}_i, t)$ evaluated at the center of mass

$$\bar{\mathbf{x}}_i = \frac{1}{|V_i|}\int_{V_i} \mathbf{x}\,\mathrm{d}\mathbf{x}.$$

The surface integral in the right-hand side of (1.46) can also be evaluated by the midpoint rule. If the normal to the interface S_{ij} is not constant, numerical integration is performed patchwise. Interpolation is required to approximate the flux function at the quadrature points, because only $u_i(t) \approx u(\bar{\mathbf{x}}_i, t)$ are available. If the same interpolation formula is used on both sides of the interface, then the finite volume discretization (1.48) is conservative, both locally and globally. Summation over i yields a discrete counterpart of the integral conservation law (1.40) for $V = \Omega$.

The finite volume method is promoted in the majority of introductory courses and textbooks on numerical methods for CFD. It is relatively easy to understand and implement, especially in the case of first- and second-order approximations on structured meshes. Higher-order schemes are difficult to derive within the framework of the above-mentioned two-level approximation strategy that involves interpolation and integration. Finite volumes lend themselves to the discretization of hyperbolic equations but the approximation of diffusive fluxes requires numerical differentiation, which makes it rather difficult to achieve high accuracy and preserve the symmetry of the continuous diffusion operator at the discrete level.

1.4.5 Finite Element Methods

The finite element method (FEM) is a relative newcomer to CFD and a very promising alternative to finite differences and finite volumes. It is usually used in conjunction with unstructured meshes and provides the *best approximation property* when applied to elliptic and parabolic problems at relatively low Peclet numbers. The development of high-resolution finite element schemes for hyperbolic and convection-dominated transport equations is a topic of active research. A summary of the author's contributions to this field will be the main highlight of the present book.

Finite element approximations to the CDR equation are based on the weighted residual method. The weak formulations (1.20) and (1.21) can be written as

$$\left(w, \frac{\partial u}{\partial t}\right) + c(w,u) + d(w,u) = r(w), \qquad \forall w \in \mathscr{W}. \tag{1.49}$$

The operators $c(\cdot,\cdot)$ and $d(\cdot,\cdot)$ are associated with the weak form of the convective and diffusive terms, respectively. The reactive part $r(\cdot)$ combines the contributions of the source/sink term s and of the surface integral (1.22), if any. The scalar product $(\cdot,\cdot)$ is defined in the space $L_2(\Omega)$ of functions that are square integrable in Ω

$$(w,v) = \int_\Omega wv \,\mathrm{d}\mathbf{x}, \qquad \forall v, w \in L_2(\Omega).$$

The approximate solution $u_h \in \mathscr{V}_h$ to problem (1.49) is defined as follows

$$u_h(\mathbf{x},t) = \sum_j u_j(t)\varphi_j(\mathbf{x}), \tag{1.50}$$

where $\{\varphi_i\}$ is a set of basis functions spanning the finite-dimensional space $\mathscr{V}_h$. As a rule, these basis functions are required to possess the following properties [76]

- there exists a set of nodes $\mathbf{x}_i \in \Omega$ such that $\varphi_i(\mathbf{x}_i) = 1$ and $\varphi_i(\mathbf{x}_j) = 0$, $\forall j \neq i$;
- the restriction of φ_i to each cell is a polynomial function of local coordinates.

Due to the first property and (1.50), the nodal values of the approximate solution are given by $u_h(\mathbf{x}_i,t) = u_i(t)$. The points $\mathbf{x}_i$ are usually located at the vertices of the mesh. Other possible locations are the midpoints/barycenters of edges, faces, and cells [76]. A typical piecewise-linear basis function φ_i is depicted in Fig. 1.7.

Let $\mathscr{W}_h$ be the space of test functions spanned by $\{\psi_i\}$. Using $u = u_h$ and $w = \psi_i$ in (1.49), one obtains a semi-discrete equation of the form (1.34), where

$$\begin{aligned} m_{ij} &= (\psi_i, \varphi_j), \quad c_{ij} = c(\psi_i, \varphi_j), \\ d_{ij} &= d(\psi_i, \varphi_j), \quad r_i = r(\psi_i). \end{aligned}$$

The conventional *Galerkin method* takes the trial functions φ_j and test functions ψ_i from the same space $\mathscr{W}_h = \mathscr{V}_h$. That is, $\psi_i = \varphi_i$, so the number of equations equals the number of unknowns. Sometimes it is worthwhile to consider test functions from $\mathscr{W}_h \neq \mathscr{V}_h$ spanned by the same number of basis functions $\psi_i \neq \varphi_i$. Such finite

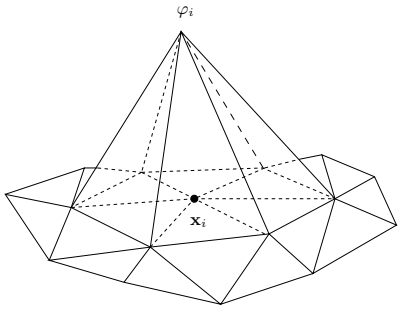


Fig. 1.7 A piecewise-linear finite element basis function on a triangular mesh.

element approximations are known as *Petrov-Galerkin methods* and offer certain advantages, e.g., in the case of convection-dominated transport problems.

The finite element method is supported by a large body of mathematical theory that makes it possible to obtain rigorous error estimates and proofs of convergence. Moreover, it can be combined with $h-p$ *adaptivity*, whereby the local mesh size and the order of polynomials are chosen so as to obtain the best possible resolution. The finite element mesh (also called *triangulation*) is usually unstructured, and the shape of mesh cells can be fitted to the shape of a curvilinear boundary. Matrix assembly is performed element-by-element in a fully automatic way. The remarkable generality and flexibility of the FEM makes it very powerful. Almost all codes for structural mechanics problems are based on finite element approximations, and a lot of current research is aimed at the development of adaptive FEM for fluid dynamics.

Finite elements and finite volumes have a lot in common and are largely equivalent in the case of low-order polynomials [8, 166, 226, 300]. The traditional strengths of FVM and FEM, as applied to the CDR equation, are complementary. Convective terms call for the use of an upwind-biased discretization which is easier to construct within the finite volume framework. On the other hand, the finite element approach takes the lead when it comes to the discretization of diffusive terms. Therefore, many hybrid FVM-FEM schemes have been proposed. For example, finite element shape functions are used to interpolate the fluxes for a vertex-centered finite volume method [104]. Conversely, FVM-like approximations of convective terms are frequently employed to achieve the upwinding effect in finite element codes [11, 322]. *Fluctuation splitting* (alias *residual distribution*) methods [54, 79] represent another attempt to bridge the gap between the FVM and FEM worlds.

A current trend in CFD is towards the use of *discontinuous Galerkin* (DG) methods [66] which represent a generalization of FVM and incorporate some of their most attractive features, such as local conservation. At the same time, the treatment of diffusive fluxes is not straightforward and requires special care, as in the case of classical FVM. We welcome the advent of DG methods but feel that the potential of continuous finite elements has not yet been exploited to the full extent in the context of transport equations. This is why the high-resolution finite element schemes to be presented in this book will be based on conventional Galerkin discretizations.

1.5 Systems of Algebraic Equations

Space discretization of a stationary problem leads to a set of coupled algebraic equations which do not contain any derivatives and, therefore, do not require any further discretization. An algebraic system like (1.37) needs to be solved just once since the nodal values of the approximate solution are assumed to be independent of time.

If the problem at hand is nonstationary, then semi-discrete equations of the form (1.34) must be integrated in time using a suitable numerical method. To this end, the time interval $(0,T)$ is discretized in much the same way as the spatial domain for one-dimensional problems. Consider a sequence of discrete time levels

$$0 = t^0 < t^1 < \ldots < t^K = T.$$

The time step $\Delta t^n = t^{n+1} - t^n$ may be constant or variable. In the former case

$$t^n = n\Delta t, \qquad \forall n = 0, 1, \ldots, K.$$

The value of the approximate solution at node i and time level t^n is denoted by

$$u_i^n \approx u_i(t^n).$$

Due to the initial condition (1.14), the value of $u_i^0 = u_i(0)$ is assumed to be known.

In principle, the time can be treated just like an extra space dimension. Finite difference, finite volume, and finite element methods are readily applicable to functions of $\mathbf{x} = (x_1, \ldots, x_d, t) \in \mathbb{R}^{d+1}$ defined in the space-time domain $\Omega \times (0,T)$. However, simultaneous computation of u_i^n for all nodes and time levels is often too expensive. Since information propagates forward in time, it is worthwhile to take advantage of this fact and advance the numerical solution in time step-by-step.

Time-stepping (or *marching*) methods initialize the vector of discrete nodal values by $u^0 = \{u_i^0\}$ and use $u^n = \{u_i^n\}$ as initial data for the computation of $u^{n+1} = \{u_i^{n+1}\}$. This solution strategy is faster and requires less memory than a coupled space-time discretization. First, the size of the algebraic systems to be solved at each time step depends only on the number of spatial degrees of freedom and not on the number of time levels. In other words, a huge system is replaced by a sequence of smaller ones, which results in considerable savings of computer time. Second, intermediate data are overwritten as soon as they are not needed anymore. Last but not least, a nonphysical dependence of u^n on u^{n+1} is ruled out by construction.

Regardless of the methods chosen to discretize the continuous problem in space and time, the result is an algebraic system that can be written in the generic form

$$Au = b, \tag{1.51}$$

where $A = \{a_{ij}\}$ is a sparse matrix, $u = u^{n+1}$ is the vector of unknowns, and b is evaluated using the previously computed data from one or more time levels. The discretization is said to be *explicit* if A is a diagonal matrix and *implicit* otherwise.

1.5.1 Time-Stepping Techniques

A semi-discrete system like (1.36) can be discretized in time using a wealth of methods developed for numerical integration of ODEs and differential algebraic equations (DAEs). This approach is called the *method of lines* (MOL) since the problem at hand consists of many one-dimensional subproblems for the nodal values $u_i(t)$ to be integrated over the time interval (t^n, t^{n+1}) subject to the initial condition $u_i(t^n) = u_i^n$. The decoupling of space and time coordinates makes it possible to use any discretization technique applicable to an initial value problem of the form

$$\frac{\mathrm{d}u}{\mathrm{d}t} = F(u,t), \qquad t^n < t \le t^{n+1}, \qquad u(t^n) = u^n. \tag{1.52}$$

Within the MOL framework, the solution u is the vector of time-dependent nodal values, whereas the vector $F(u,t)$ contains the discretized space derivatives, sources, sinks, and boundary conditions. In the case of our DAE system (1.36)

$$MF(u,t) = r(t) - (C+D)u(t). \tag{1.53}$$

If the underlying velocity field and/or the diffusion tensor depend on t, then so do the coefficients of the discrete operators $C = \{c_{ij}\}$ and $D = \{d_{ij}\}$, respectively.

The simplest time-stepping methods are based on a finite difference discretization of the time derivative that appears in (1.36) and (1.52). Let $t^{n+1} = t^n + \Delta t$ and

$$\frac{u^{n+1} - u^n}{\Delta t} = \theta F(u^{n+1}) + (1-\theta)F(u^n), \qquad 0 \le \theta \le 1. \tag{1.54}$$

This generic formula unites the first-order accurate forward Euler method ($\theta = 0$)

$$u^{n+1} = u^n + \Delta t F(u^n), \tag{1.55}$$

the second-order accurate Crank-Nicolson scheme which corresponds to $\theta = \frac{1}{2}$

$$u^{n+1} = u^n + \frac{\Delta t}{2}[F(u^{n+1}) + F(u^n)], \tag{1.56}$$

and the first-order accurate backward Euler method ($\theta = 1$) which yields

$$u^{n+1} = u^n + \Delta t F(u^{n+1}). \tag{1.57}$$

In general, the right-hand side of (1.54) is a weighted average of $F(u,t)$ evaluated at the old and new time level. Depending on the value of θ, the resulting time discretization can be explicit or implicit. In an update like (1.55), each nodal value u_i^{n+1} can be calculated explicitly using the data from the previous time level. In the case of implicit methods ($\theta > 0$), each algebraic equation contains several unknowns, whence a simultaneous update of all nodal values is required. Explicit methods are easier to implement but implicit methods are more stable, as explained below.

To obtain a fully discrete counterpart of system (1.36), all we have to do is to multiply (1.54) by the mass matrix M and invoke (1.53). The result is an algebraic system of the form (1.51), where the left-hand side matrix A is given by

$$A = M + \theta \Delta t (C + D),$$

and the right-hand side b is the sum of all terms that do not depend on u^{n+1}

$$b = [M - (1-\theta)\Delta t(C+D)]u^n + \Delta t[\theta r^{n+1} + (1-\theta)r^n].$$

This discretization is fully explicit if $\theta = 0$ and the mass matrix M is diagonal. The latter condition holds for any finite difference or finite volume scheme. However, many finite element approximations produce nondiagonal mass matrices, so that a linear system needs to be solved even if the forward Euler method is employed.

The generic $\theta-$scheme (1.54) belongs to the family of *two-level* methods since only u^n is involved in the computation of u^{n+1}. Such time-stepping methods can be at most second-order accurate. Higher-order approximations must use information from additional time levels. In Runge-Kutta methods, all time levels $t^{n+\alpha}$, where $0 \leq \alpha \leq 1$, belong to the interval $[t^n, t^{n+1}]$, and a predictor-corrector strategy is employed to compute u^{n+1}. Another possibility is to integrate a polynomial fitted to the values of $F(u,t)$ at $t^{n+\alpha}, \ldots, t^{n-\beta}$, where α and β are nonnegative integers. Multipoint methods of Adams-Bashforth (explicit) and Adams-Moulton (implicit) type correspond to $\alpha = 0$ and $\alpha = 1$, respectively. Their pros and cons, as compared to Runge-Kutta time-stepping schemes of the same order, are explained in [104].

1.5.2 Direct vs. Iterative Solvers

The last step in the development of a numerical algorithm is the solution of the algebraic system (1.51) that results from the discretization of a continuous problem. In the case of an explicit scheme, the computation of $u = A^{-1}b$ is trivial

$$u_i = \frac{b_i}{a_{ii}}, \qquad \forall i.$$

Otherwise, the tools of numerical linear algebra are required to solve (1.51). The choice of the solution method is largely independent of the underlying discretization techniques but the size and structure of the matrix A need to be taken into account.

1.5.2.1 Direct Methods

Direct methods for solving linear systems of the form (1.51) accomplish this task in one step. The input parameters are the matrix A and the right-hand side b. The result is the solution vector u. In the absence of roundoff errors, this solution is exact.

Direct methods perform well for linear systems of moderate size but the memory requirements and CPU time increase nonlinearly with the number of unknowns. As a rule of thumb, direct solvers are not to be recommended for very large systems.

If the sparsity pattern of the matrix A exhibits some regular structure, then this structure can sometimes be exploited to design a fast direct solver, such as the Thomas algorithm for tridiagonal matrices. This algorithm can also be embedded in Alternating-Direction-Implicit (ADI) solvers for banded matrices that result from 2D and 3D discretizations on structured meshes. The multifrontal method implemented in the open-source software package UMFPACK [330] is one of the fastest direct solvers for nonsymmetric sparse linear systems. The rapid increase in computer memory and the possibility of efficient implementation on parallel supercomputers have revived the interest in direct solvers for large-scale applications [226].

1.5.2.2 Iterative Methods

Iterative methods solve linear systems of the form (1.51) using a sequence of explicit updates starting with an initial guess which must be supplied as another input parameter. Such an algorithm is said to be convergent if each update brings the solution closer to that of (1.51) which is recovered after sufficiently many iterations. In practice, it is neither necessary nor wise to iterate until convergence. The iterative process is terminated when certain stopping criteria are satisfied. As a rule, these criteria amount to monitoring the differences between two successive iterates and/or the residuals measured in a suitably chosen norm. Stopping too early gives rise to large iteration errors; stopping too late results in a waste of CPU time.

Iterative solvers use a rather small amount of computer memory. Many of them do not even require that the matrix A be available. All they need is a subroutine that evaluates the residual of the linear system for a given tentative solution. The most efficient iterative algorithms are based on multigrid methods [131, 345]. If properly configured, they can solve a system of N equations using as few as $\mathcal{O}(N)$ arithmetic operations, as compared to $\mathcal{O}(N^3)$ for direct solvers based on Gaussian elimination and $\mathcal{O}(N^2)$ for forward/backward substitution, given a precomputed LU factorization. Clearly, this makes a big difference, especially if N is as large as 10^6 and more. Thus, an iterative solution strategy pays off for large sparse linear systems.

If the system to be solved corresponds to the discretization of a stationary problem, a good initial guess is rarely available. The default is zero, and many iterations are usually required to achieve the prescribed tolerance. The computational effort associated with advancing the solution of an unsteady problem from one time level to the next is much smaller. Since u^n provides a relatively good guess for u^{n+1}, just a few iterations per time step are normally required to obtain a sufficiently accurate solution. Therefore, each update is typically much cheaper than that performed with a direct solver. If the time step is very small, then a single iteration may suffice, and the cost of solving the linear system is comparable to that of a fully explicit update. As the time step increases, so does the number of iterations for the linear solver.

In light of the above, an iterative solver for the discretization of a stationary problem may be more expensive and/or difficult to implement than a time-stepping method for the corresponding unsteady problem. It might be easier to march the solution to the steady state. If we discretize system (1.36) in time by an explicit or implicit method, prescribe arbitrary initial conditions, and run the code for a sufficiently long time, then we will end up with the solution of (1.37). This popular approach to steady-state computations is called *pseudo time-stepping* since it represents an iterative solver in which the time step serves as a relaxation parameter. In this case, the accuracy of the time discretization is not important, and the artificial time step should be chosen so as to reach the steady state limit as fast as possible. Note that the use of excessively large time steps or inappropriate parameter settings may cause divergence, which makes iterative methods less robust than direct ones.

1.5.2.3 Nonlinear Systems

If the coefficients of the discrete problem depend on the unknown solution, then the algebraic system (1.51) is nonlinear. In this case, an iterative solution strategy is a must. Starting with a suitably chosen initial guess, the currently available solution values are used to update the matrix coefficients and obtain the next guess by solving a nominally linear system. This process is repeated until the changes and/or the residual of the nonlinear system become small enough. In principle, both direct and iterative methods can be used to solve the involved linear systems. However, the use of direct methods is usually impractical since they spend an excessive amount of CPU time on solving linear systems with only tentative coefficients.

Within a fully iterative approach, the coefficients are calculated using the solution values from the previous *outer iteration*, and a few *inner iterations* are performed to obtain an improved solution. Since the coefficients are tentative, the amount of work spent on the solution of each linear system should not be inordinately large [268]. As soon as the residuals have decreased by a factor of 10 or so, one may stop the inner iteration process and proceed to the calculation of the coefficients for the next outer iteration. Well-balanced stopping criteria make it possible to minimize the computational cost that depends on the total number of inner and outer iterations.

Of course, an iterative method is of little value if it does not converge. If the nonlinearity is too strong, the solution may exhibit oscillatory behavior that inhibits convergence. A possible remedy to this problem is the use of underrelaxation techniques. The basic idea is to take a weighted average of the old and new data, so as to slow down the changes of solution values and matrix coefficients [268].

1.5.3 Explicit vs. Implicit Schemes

In the case of an fully explicit scheme, no linear or nonlinear systems need to be solved. Explicit algorithms are easy to code/parallelize and require a modest amount

of computer memory. However, the time step may not exceed a certain threshold that depends on the Peclet number and on the smallest mesh size. Otherwise, the scheme may become unstable and produce meaningless numbers going to infinity. The lack of stability is the price to be paid for algorithmic simplicity of explicit schemes. The cost of a single solution update is minimal but an inordinately large number of time steps may be required to perform simulation over a given interval of time.

Implicit methods produce nondiagonal matrices, whence each algebraic equation contains several unknowns and cannot be solved in a stand-alone fashion. The design of an efficient implicit algorithm is particularly difficult if the underlying PDE and/or the discretization procedure are nonlinear. The cost per time step is large as compared to that of an explicit solution update. Also, the programming of iterative solvers for (1.51) is time-consuming, and their efficiency depends on the parameter settings, stopping criteria etc. On the other hand, most implicit schemes are unconditionally stable, and the use of large time steps makes it possible to reach the final time faster than with an explicit scheme subject to a restrictive stability limit. Of course, it should be borne in mind that the accuracy of the time discretization and the convergence behavior of iterative solvers also depend on the time step.

It is essential to distinguish between truly transient problems and the ones in which the solution varies slowly and/or becomes stationary in the long run. The optimal choice of the time-stepping scheme depends on whether the goal is

- to perform a transient computation in which evolution details are important, or
- to predict the long-term flow behavior or to compute a steady-state solution.

Depending on the objectives of the simulation to be performed, an explicit or implicit solution strategy may be preferable. Explicit schemes lend themselves to the treatment of problems in which the use of small time steps is dictated by accuracy considerations. If only the steady-state solution is of interest, then it is worthwhile to use *local time-stepping* (different time steps for different nodes) and/or an unconditionally stable implicit scheme, such as the backward Euler method (1.57). If it is not known in advance, whether the transport process to be simulated is steady or unsteady, it is possible to start with an explicit scheme and switch to an implicit one if the ratio $(u^{n+1} - u^n)/\Delta t$ becomes small as compared to other terms. Therefore, both explicit and implicit solvers belong into a general-purpose CFD toolbox.

1.6 Fundamental Design Principles

No numerical method is perfect, and many compromises are involved in the design process. The quality of numerical approximations to convection-diffusion equations depends on the underlying mesh, on the properties of the employed discretization techniques, and on the Peclet number. The mesh size and time step should also be chosen carefully, especially in the case of conditionally stable explicit schemes. All of the above-mentioned factors influence the coefficients of the algebraic system (1.51) for the nodal values of the approximate solution. The properties of the matrix

A and of the discrete operators involved in the assembly of the right-hand side b may make or break the entire numerical model. The type of the governing equation and the smoothness of the solution may also play an important role. Some algorithms are tailored to equations of a certain type and/or perform poorly if the solution exhibits steep gradients. Other methods do not work at all or produce nice-looking results which have little in common with the true solution of the mathematical model.

A good numerical method must fulfil a number of prerequisites dictated by the physics, mathematical theory, and numerical analysis of the problem at hand. These criteria lead to a set of rules that guarantee a certain level of accuracy and robustness for a sufficiently broad range of applications. Some of the fundamental design principles are summarized in this section, and their implications are explained.

1.6.1 Numerical Analysis

The difference between the exact solution u of the continuous problem and an approximate solution $u_h^{\Delta t}$ produced by a computer code is the sum of numerical errors. Aside from programming bugs, we can distinguish between discretization errors, roundoff errors, and iteration errors. The discretization error $\varepsilon_h^{\Delta t}$ depends on the mesh size h and time step Δt. It can be estimated using Taylor series expansions or, in the case of finite element methods, sophisticated tools of functional analysis.

Roundoff errors due to the finite precision of computer arithmetics are usually much smaller than $\varepsilon_h^{\Delta t}$, whereas iteration errors depend on the prescribed tolerances and stopping criteria for linear solvers. A properly designed numerical scheme must be sufficiently accurate and converge to the exact solution of the differential equation as the mesh size h and time step Δt are refined. Therefore, it must contain inherent mechanisms to control the magnitude of the total error in the course of simulation. A rigorous analysis of *consistency, stability, and convergence* is required to evaluate new discretization techniques and identify the range of their applicability.

1.6.1.1 Consistency

A numerical method is said to be consistent if the discretization error $\varepsilon_h^{\Delta t}$ goes to zero as $h \to 0$ and $\Delta t \to 0$. Consistency refers to the relationship between the exact solutions of the continuous and discrete problems. In essence, it guarantees that the discretization is asymptotically correct. Of course, finite values of h and Δt are used in practice to keep simulations affordable. Since the computational cost increases rapidly with the number of unknowns, it is natural to require that the discretization error $\varepsilon_h^{\Delta t}$ become smaller if we take a finer mesh and/or time step. Moreover, it is desirable to have some idea of how much accuracy we can gain by doing so. This information can be inferred from an *a priori* error estimate of the form

$$\varepsilon_h^{\Delta t} = \mathcal{O}(h^p, \Delta t^q). \tag{1.58}$$

The corresponding discretization is consistent if $p > 0$ and $q > 0$. Its spatial and temporal accuracy is of order p and q, respectively. This gives an asymptotic estimate of the rate at which $\varepsilon_h^{\Delta t}$ shrinks as the mesh and/or time step are refined.

It is worth mentioning that the formal order of approximation is not the sole indicator of accuracy and, in many cases, not even a particularly good one. The absolute values of the errors produced by two schemes of the same order may differ significantly, and a low-order scheme might perform better than a high-order one on a coarse mesh. Strictly speaking, *a priori* estimates based on Taylor series expansions are not applicable to discontinuous solutions, Also, the contribution of higher-order terms may become nonnegligible if the corresponding derivatives are too large. As a consequence, the error $\varepsilon_h^{\Delta t}$ might decrease much slower than expected. Last but not least, even a consistent scheme may fail to converge if it turns out to be unstable.

1.6.1.2 Stability

A numerical method is said to be stable if numerical errors, e.g., due to roundoff, are not amplified, and the approximate solution remains bounded. This criterion applies to time-stepping schemes and iterative solvers alike. Stability refers to the relationship between the exact solution of the discrete problem and the actually computed solution that includes roundoff and iteration errors. Mathematical tools of stability analysis are available for linear problems with constant coefficients. The most popular technique is the von Neumann method. Nonlinear problems are more difficult to analyze and may require a stronger form of stability, see Chapter 3. Some approximations enjoy unconditional stability, others are stable under certain conditions for the choice of input parameters. Unstable schemes may be fixed by adding extra terms provided that the discretization remains consistent. This idea leads to a rich variety of stabilized schemes to be presented in what follows.

1.6.1.3 Convergence

A numerical method is said to be convergent if the numerical solution of the discrete problem approaches the exact solution of the differential equation as the mesh size and time step go to zero. High-order methods converge to smooth solutions faster than low-order ones. Consistency and stability are the necessary and sufficient conditions of convergence for finite difference approximations to well-posed linear initial value problems. This statement is known as the *Lax equivalence theorem*. In the nonlinear case, compactness is the main ingredient of convergence proofs [216].

In practical computations, convergence must be verified numerically by running the same simulation on a series of successively refined meshes and varying the time step size. The results of this *grid convergence* study can also be used to estimate the genuine order of accuracy [104, 218, 283]. If the discrepancy between the solutions obtained with several different meshes and time steps is insignificant, this indicates that the numerical errors are small, and the results are close enough to the exact

solution of the differential equation. Otherwise, the refinement process should be continued to make sure that the method converges to a grid-independent solution.

1.6.2 Physical Constraints

Consistency, stability, and convergence are the three cornerstones of numerical approximation. A discretization that meets all of these requirements is guaranteed to produce an accurate solution provided that h and Δt are sufficiently small. However, the definition of "sufficiently small" is highly problem-dependent. If a numerical scheme fails to resolve a small-scale feature properly on a given mesh, it typically reacts by generating large numerical errors and/or nonphysical side effects, such as a spontaneous loss/gain of mass or spurious oscillations, also known as 'wiggles'.

The strongest violation of physical realism is likely to occur in the vicinity of discontinuities, moving fronts, interior/boundary layers, and other regions in which the solution gradients are steep. In many cases, small imperfections may be tolerated but there are situations in which nonphysical solution behavior is totally unacceptable. Discretization techniques that may give rise to artificial sources/sinks or negative concentrations should be avoided. Therefore, certain physical constraints, such as *conservation* and *boundedness*, may need to be enforced at the discrete level [104].

1.6.2.1 Conservation

Since mathematical models of transport phenomena are based on conservation principles, similar principles should apply to the approximate solution. If the quantity of interest is conserved, then numerical errors can only distribute it improperly. Discrete conservation is a constraint that forces the numerical algorithm to reproduce an important qualitative property of the physical system correctly. The *Lax-Wendroff theorem* states that if a consistent and conservative scheme converges, then it converges to a weak solution of the conservation law. Convergence to nonphysical weak solutions, such as shocks moving at wrong speeds, is ruled out by this theorem.

In the asymptotic limit, even a nonconservative scheme will produce correct results if it is consistent and stable. Unfortunately, it is difficult to tell how fine the mesh and time step must be chosen to keep conservation errors small enough. Thus, even solutions to rather simple problems may behave in an unpredictable manner.

A finite difference scheme proves conservative if it can be written in terms of numerical fluxes from one grid point into another. Finite volume and discontinuous Galerkin methods are conservative by construction, both globally and locally. The continuous Galerkin FEM provides global conservation [134] and is claimed to be locally conservative by some authors [158, 162]. Conservation may be lost if inaccurate quadrature rules or nonstandard approximations are employed. Any modification of a conservative scheme is dangerous and must be examined critically.

1.6.2.2 Boundedness

In many applications, the transported quantity must stay within certain bounds for physical reasons. For example, densities and temperatures must be nonnegative; volume and mass fractions must be bounded by 0 and 1. Under certain assumptions, analytical solutions to scalar transport equations are known to attain their maxima and/or minima on the boundary of the domain. In unsteady problems, the local extrema of initial data may also serve as upper or lower bounds. Maximum and minimum principles for PDEs of different types are presented in Chapter 3. Similar constraints can be formulated for the nodal values of the discrete solution.

If a numerical approximation fails to satisfy an *a priori* bound based on the known properties of the exact solution, it can easily be repaired by clipping all undershoots and overshoots. However, pointwise correction of nodal values is a dangerous practice since the conservation property may be lost. Some low-order approximations are conservative and bounded but their accuracy leaves a lot to be desired. High-order schemes perform well for smooth data but may produce unbounded solutions in the neighborhood of steep fronts. A conservative and bounded convergent scheme of high order is rather difficult to design and expensive to run but the results are usually rewarding. Indeed, if the total amount of the conserved quantity is correct, its distribution is accurate, and all relevant upper/lower bounds are satisfied, then the numerical solution must be very close to the exact one. Boundedness implies strong stability, so a consistent and bounded scheme is convergent.

1.6.2.3 Causality

In some situations, it is important to make sure that information travels in the right direction (downstream and forward in time) and at the right speed. This principle is known as *casuality* [265]. Some algortihms transmit information too far or fast; others fail to reflect the one-way pattern of wave propagation. The principle of casuality requires that large differences between the analytical and numerical domains of dependence be avoided, as far as possible. A good numerical method should be faithful to the nature of the physical and mathematical problem to be solved.

1.6.3 The Basic Rules

Since the end product of the discretization process is an algebraic system, the above design principles impose certain restrictions on the coefficients of the numerical scheme. Four basic rules that ensure conservation, boundedness, and physical realism were formulated by Patankar [268] three decades ago. In this section, we restate these guidelines in a form suitable for our purposes. Later, we will put them on a firm mathematical basis, explain their far-reaching implications, and use them as a

design tool. As we will see, Patankar was far ahead of his time in recognizing the importance of algebraic constraints for the development of numerical methods.

1.6.3.1 Mass Balance

The first basic rule is: no mass should be created or destroyed inside the domain by the discretized convective and diffusive terms. The global mass balance may only change due to sources, sinks, and nonzero fluxes across the boundary.

In our systems (1.36) and (1.37), the discrete mass that belongs to node i is given by $m|_i = \sum_j m_{ij} u_j$, where m_{ij} is a coefficient of the mass matrix M. The sum of its coefficients should be equal to the area/volume of the computational domain.

The evolution of masses $m|_i$ is governed by (1.34) or, in the case of a diagonal mass matrix, (1.35). Summing over i, one obtains the global mass balance

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \sum_i \left[r_i - \sum_j (c_{ij} + d_{ij}) u_j \right], \tag{1.59}$$

where $m = \sum_i m|_i$ is the total mass that must satisfy a discrete conservation principle.

Depending on the choice of discretization techniques, the correctness of the mass balance (1.59) can be checked and maintained in (at least) two different ways:

- Finite volume methods and some of their finite difference counterparts can be expressed in terms of numerical fluxes f_{ij} defined as suitable approximations to (1.46). For the scheme to be conservative, each pair of fluxes must satisfy

  $$f_{ji} = -f_{ij}, \qquad \forall j \neq i. \tag{1.60}$$

  The flux f_{ij} corresponds to the amount of mass transported by convection and/or diffusion from node i into node j. Since the flux f_{ji} is the negative of f_{ij}, what is subtracted from one node is added to another. Hence, mass is conserved.

- Finite element methods can also be written in terms of numerical fluxes that satisfy (1.60), see Chapter 3. Another useful criterion is based on the properties of the discrete convection and diffusion operators $C = \{c_{ij}\}$ and $D = \{d_{ij}\}$. Note that the contribution of u_j to the right-hand side of (1.59) vanishes if

  $$\sum_i c_{ij} = 0, \qquad \sum_i d_{ij} = 0. \tag{1.61}$$

  Therefore, the scheme is conservative if the matrices C and D have zero column sums, except for a small set of nodes located on the boundary or next to it. This property was mentioned in [43] in the context of finite difference discretizations.

At the stage of testing and debugging, it is useful to check if the total mass evolves in the right way. If this is not the case, the code is likely to contain a pernicious bug.

1.6.3.2 Zero Row Sums

The second basic rule is: if a continuous operator produces zero when applied to a constant, so should its discrete counterpart. If we consider (1.13) with $s \equiv 0$, this equation remains valid if the solution and its boundary values are increased by an arbitrary constant. For the solution of the discrete problem to possess the same property, it is sufficient to require that the matrices C and D have zero row sums

$$\sum_j c_{ij} = 0, \qquad \sum_j d_{ij} = 0. \tag{1.62}$$

Then the discrete solution is defined up to an additive constant. Uniqueness follows from the Dirichlet boundary conditions to be prescribed for at least one node.

By definition, the diffusive flux is proportional to the gradient of u, so the rows of D should always satisfy the zero-sum rule. However, the row sums of C may be nonzero if its continuous counterpart is given by (1.12) and the velocity field is not divergence-free. In this case, the governing equation contains a zeroth-order term of the form $(\nabla \cdot \mathbf{v})u$, so the rule is not applicable to the discrete convection operator. If we force the numerical solution to behave in a certain way, then our intention is to mimic some qualitative properties of the exact solution. The basic rules should not be used blindly in situations when the underlying assumptions do not hold.

1.6.3.3 Positive Coefficients

The third basic rule is: if convection and diffusion are the only processes to be simulated, the nodal value u_i^{n+1} should not decrease as result of increasing any other nodal value that appears in the discretized equation for node i. Conversely, it should not increase if another nodal value is decreased, all other things being fixed.

In the absence of a reactive term, the fully discrete problem can be written as

$$Au^{n+1} = Bu^n, \tag{1.63}$$

where $A = \{b_{ij}\}$ and $B = \{b_{ij}\}$ contain the coefficients of the implicit and explicit part, respectively. The nodal value u_i^{n+1} satisfies the following algebraic equation

$$a_{ii}u_i^{n+1} = \sum_j b_{ij}u_j^n - \sum_{j \neq i} a_{ij}u_j^{n+1}. \tag{1.64}$$

Obviously, the requirements of the third rule are satisfied if the coefficients of all nodal values that contribute to this equation have the same sign. Following Patankar [268], we choose them to be positive. Strictly speaking, we require that

$$a_{ii} > 0, \quad b_{ii} \geq 0, \qquad \forall i, \tag{1.65}$$

$$a_{ij} \leq 0, \quad b_{ij} \geq 0, \qquad \forall j \neq i. \tag{1.66}$$

In the context of explicit finite difference schemes, positivity constraints of this form were formulated by Book *et al.* [43] as early as in 1975. If $a_{ij} = 0$ for all $j \neq i$, then it is obvious that a scheme of the form (1.64) is *positivity preserving*, i.e.,

$$u^n \geq 0 \quad \Rightarrow \quad u^{n+1} \geq 0, \qquad \forall n. \tag{1.67}$$

In the implicit case, a proof of this property is based on the *M-matrix* property of A which ensures that all coefficients of A^{-1} are nonnegative, see Chapter 3.

As we will see, conditions (1.65)–(1.66) and similar algebraic constraints can be used to achieve many favorable properties, such as positivity, monotonicity, nonincreasing total variation, and the discrete maximum principle, to name just a few.

1.6.3.4 Negative Slopes

The fourth basic rule is: if the discretization of convective and diffusive terms is positivity-preserving, inclusion of a reactive part should not destroy this property.

Sources and sinks may reverse the sign of analytical and numerical solutions alike. It is not unusual that they trigger numerical instabilities, cause divergence of iterative solvers, or give rise to nonphysical artifacts. Hence, the treatment of zeroth-order terms requires special care. Assume that a nonlinear reactive term r_i can be split into a (positive) source and a (negative) sink proportional to u_i as follows

$$r_i = \beta_i - \alpha_i u_i. \tag{1.68}$$

The linearization parameters α_i and β_i may depend on the unknown solution values.

It is instructive to consider the extreme situation in which r_i is very large as compared to other terms and, therefore, equation (1.34) reduces to $r_i = 0$ or

$$\alpha_i u_i = \beta_i. \tag{1.69}$$

To secure convergence and keep the numerical solution positive, the splitting of the reactive term should be designed so that $\alpha_i \geq 0$ and $\beta_i \geq 0$ for all i, see [268].

Even if the contribution of other terms to the discretized equation cannot be neglected, it is worthwhile to express the reactive term r_i into the form (1.68) with nonnegative coefficients α_i and β_i. This *negative-slope linearization* technique was invented by Patankar [268] to preserve the sign of inherently positive variables. Guidelines for constructing splittings of the form (1.68) can be found in his book.

The aspects of positivity preservation for linearized reactive terms of the form (1.68) with $\beta_i = c_i u_i \geq 0$ were also analyzed by MacKinnon and Carey [246] who mentioned that such a splitting is possible, e.g., for zeroth-order kinetics (radiative decay: $r = -cu \;\Rightarrow\; \alpha = c \geq 0,\ \beta = 0$), a combination of first- and second-order kinetics (population growth: $r = c_1 u - c_2 u^2 \;\Rightarrow\; \alpha = c_2 u \geq 0,\ \beta = c_1 u \geq 0$), and reactions of fractional order (for instance: $r = -cu^{3/2} \;\Rightarrow\; \alpha = cu^{1/2} \geq 0,\ \beta = 0$).

1.7 Scope of This Book

The models, techniques, and concepts presented so far were chosen so as to get started with numerical simulation of transport phenomena without going too much into detail. The range of topics we have covered is certainly eclectic and incomplete. Other texts should be consulted for an in-depth introduction to the field of Computational Fluid Dynamics.

This book is concerned with the numerical treatment of transport equations that cannot be handled using conventional discretization techniques, or the results are unsatisfactory. If the solution of the continuous problem varies on a length scale shorter than the mesh size, small-scale features cannot be captured accurately by any numerical scheme. Thus, insufficient mesh resolution gives rise to large errors that may result in an incorrect qualitative behavior of approximate solutions.

Each scheme reacts in its own way when it encounters an unresolvable subgrid-scale feature. Typical side effects are strong numerical diffusion and/or spurious oscillations (wiggles, ripples). Arguably, an overly diffusive scheme is the lesser of the two evils if it is guaranteed to be bounded. Another arguable viewpoint is [128]: "don't suppress the wiggles — they're telling you something!" We take the liberty to formulate another rule: a good algorithm must contain just as much numerical dissipation as is necessary to avoid nonphysical artifacts. Moreover, the mesh should be refined adaptively in troublesome regions where small-scale effects are present.

In most cases, the bizarre behavior of numerical solutions is due the fact that

- the model is dominated by convection, anisotropic diffusion, or stiff reaction;
- discontinuities, steep gradients, and/or interior/boundary layers are involved;
- the employed numerical scheme violates at least one of the four basic rules.

A possible remedy is to take a suitable high-order discretization and constrain it at the algebraic level so as to enforce desirable properties without losing too much accuracy. This is the approach that we will pursue and promote in this book.

Most of the material presented in this text is not really new. Many excellent books and review articles have been written about numerical methods for convection-diffusion equations and hyperbolic conservation laws. The main reasons that have led the author to retell the story are as follows:

- useful techniques are scattered over a vast body of literature and difficult to find;
- many algorithms are inherently explicit or require the use of structured meshes;
- texts overloaded with complex mathematical theory are unreadable to engineers;
- rigorous convergence proofs may disguise the fact that the method does not really work when applied to problems in which subgrid-scale effects are important.

In the introductory part, we select a number of particularly good and relatively simple methods. We analyze the properties of these methods and put them in a unified framework so as to highlight existing similarities. Also, we present some well-known concepts in a new light and interpret them from the algebraic viewpoint. This preliminary study is intended to pave the way for various generalizations.

The *algebraic flux correction* paradigm [200] to be introduced in Chapter 4 is the methodology developed by the author and his coworkers to enforce boundedness in a conservative manner. It will be explained in the context of scalar transport equations discretized by explicit and implicit finite element methods on arbitrary meshes. Special-purpose algorithms will be presented for the numerical treatment of stationary and time-dependent problems alike. However, our main goal is to show how the underlying design philosophy works and to equip the reader with tools that make it possible to fix a given discretization building on the four basic rules and similar algebraic constraints.

A large share of CFD research is performed by people living in the parallel worlds of 'viscous incompressible' or 'inviscid compressible' flows. The former is populated by practitioners who are interested primarily in elliptic/parabolic problems. People from this group typically favor implicit algorithms, finite element discretizations, and unstructured meshes. Inviscid flows are governed by hyperbolic conservation laws, and the traditional solution strategy relies on explicit finite difference or finite volume discretizations. Of course, there are many notable exceptions to this rule, and some diffusion of ideas takes place at the interface between the two worlds. In our experience, a lot can be gained by looking at what is going on both sides of this interface.

Space and time do not permit a comprehensive review of all promising numerical schemes and discretization concepts. We will not discuss the recent developments in the realm of finite volume and discontinuous Galerkin methods. These two mainstream trends are covered in many texts and research papers. Instead, we focus on continuous Galerkin methods and make them fit for the numerical treatment of transport equations at arbitrary Peclet numbers. Many methods presented in this book were chosen with this objective in mind. Of course, this choice has been influenced by the author's personal preferences, views, and research objectives. However, the results of recent comparative studies [174, 175] indicate that it is a fairly good one.

Chapter 2
Finite Element Approximations

This chapter presents a self-contained review of some promising discretization and stabilization techniques for multidimensional transport equations in two and three space dimensions. The methods to be discussed do not require directional splitting and are readily applicable to unstructured meshes. The group finite element interpolation of the flux function provides a handy link between finite element and finite volume approximations. The existence of a conservative flux decomposition paves the way to various extensions of one-dimensional algorithms. Also, it leads to efficient edge-based data structures that offer a number of significant advantages as compared to the traditional element-based implementation. In this chapter, we consider unstructured grid methods for convection-diffusion equations and discuss relevant algorithmic details, such as matrix assembly. Furthermore, we analyze the properties of discrete operators and describe some popular approaches to the design of stabilized finite element methods for stationary and time-dependent problems.

2.1 Discretization on Unstructured Meshes

Unstructured grid methods are commonly employed if the domain of interest has a complex geometrical shape. The finite element approach provides a particularly convenient framework for the development of general-purpose software packages. Its advantages include but are not limited to the flexible choice of basis functions, simple treatment of natural boundary conditions, fully automatic matrix assembly, and the possibility of mesh adaptation backed by rigorous mathematical theory.

The roots and the traditional strength of finite element methods lie in the field of elliptic problems and applications to structural engineering, whereas finite volumes lend themselves to numerical simulation of transport phenomena. One of the reasons for this state of affairs is that many one-dimensional concepts and geometric design criteria introduced in the context of finite difference schemes are relatively easy to extend to finite volume discretizations on unstructured meshes, while an extension to finite elements is often nontrivial or even impossible. In the case of low-order

approximations, it turns out that finite element and finite volume schemes are largely equivalent, although their derivation is based on entirely different premises.

The existing similarities between the two approaches to discretization of transport equations on unstructured meshes have been recognized, documented, and exploited by many authors [8, 56, 166, 300, 301]. In particular, *edge-based* finite element methods have been formulated in terms of numerical fluxes that move the mass from one node into another without changing the global balance [226, 239, 259]. This formulation is particularly well suited for compressible flows. In spite of its enormous potential, as demonstrated by spectacular simulation results for aerodynamic applications [22, 23, 226], many finite element practitioners are unaware of its existence or reluctant to use an unconventional methodology. It is hoped that the present book will provide additional evidence in favor of edge-based algorithms.

The most general model problem to be dealt with in the present chapter is a scalar transport equation to be solved in a domain Ω with boundary $\Gamma = \Gamma_D \cup \Gamma_N$

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f} = s \qquad \text{in } \Omega, \tag{2.1}$$

where u is a conserved scalar quantity, s is a source term, and $\mathbf{f}$ is a generic flux function that consists of a diffusive and/or convective part. We will also consider the steady-state counterpart of equation (2.1) in which the time derivative is omitted.

A typical set of initial and boundary conditions for problem (2.1) is as follows

$$\begin{array}{ll} u = u_0, & \text{at } t = 0, \\ u = g, & \text{on } \Gamma_D, \\ \mathbf{f} \cdot \mathbf{n} = h, & \text{on } \Gamma_N, \end{array} \tag{2.2}$$

where $\mathbf{n}$ denotes the unit outward normal. The appropriate choice of initial/boundary conditions depends on the problem at hand and on the available information.

In what follows, we consider continuous (linear or multilinear) finite element discretizations and illustrate their relationship to finite volumes. Starting with a weak form of (2.1), we derive the coefficient matrices and examine their properties. We revisit the aspects of conservation, upwinding, and artificial diffusion in the FEM context. Also, we split the contributions of discrete convection and diffusion operators into skew-symmetric numerical fluxes that represent the rate of bilateral mass transfer between two neighboring nodes. The use of edge-based data structures is feasible but not mandatory. Furthermore, it is possible to assemble matrices element-by-element but define extra stabilization terms in terms of numerical fluxes associated with edges of the sparsity graph. This approach simplifies the implementation of edge-based artificial diffusion methods in an existing finite element code.

The presentation is focused on simple stabilization mechanisms suitable for multidimensional transport problems at moderately large Peclet numbers. Moreover, it is assumed that the computational mesh is sufficiently regular and does not contain inordinately stretched/deformed elements. Advanced mathematical theory and nonlinear high-resolution schemes for computing nonoscillatory solutions to problems with steep fronts on arbitrary meshes will be developed in the next two chapters.

2.1.1 Group Finite Element Formulation

The starting point for a finite element discretization of the scalar conservation law (2.1) is its weak form that corresponds to the weighted residual formulation

$$\int_\Omega w\left(\frac{\partial u}{\partial t}+\nabla\cdot\mathbf{f}\right)\mathrm{d}\mathbf{x}=\int_\Omega ws\,\mathrm{d}\mathbf{x}. \tag{2.3}$$

This relation must hold for any test function w vanishing on the boundary part Γ_D, see Section 1.2.5. To get started, consider a linear flux function of the form

$$\mathbf{f}=\mathbf{v}u-\varepsilon\nabla u, \tag{2.4}$$

where $\mathbf{v}$ is the velocity vector and ε is a constant diffusion coefficient. After substitution of the so-defined flux $\mathbf{f}$ and integration by parts, equation (2.3) becomes

$$\int_\Omega\left(w\frac{\partial u}{\partial t}+w\nabla\cdot(\mathbf{v}u)+\nabla w\cdot(\varepsilon\nabla u)\right)\mathrm{d}\mathbf{x}=\int_\Omega ws\,\mathrm{d}\mathbf{x}+\int_{\Gamma_N}w(\varepsilon\nabla u)\cdot\mathbf{n}\,ds. \tag{2.5}$$

The integral over Γ_N is evaluated using the diffusive flux given by the Neumann boundary conditions. If the total flux is prescribed on Γ_N, then integration by parts should also be applied to the convective term. The corresponding weak form is

$$\int_\Omega\left(w\frac{\partial u}{\partial t}-\nabla w\cdot(\mathbf{v}u-\varepsilon\nabla u)\right)\mathrm{d}\mathbf{x}=\int_\Omega ws\,\mathrm{d}\mathbf{x}-\int_{\Gamma_N}w(\mathbf{v}u-\varepsilon\nabla u)\cdot\mathbf{n}\,ds. \tag{2.6}$$

The next step is to represent the approximate solution u_h as a linear combination of piecewise-polynomial basis functions $\{\varphi_i\}$ spanning a finite-dimensional space

$$u_h=\sum_j u_j\varphi_j. \tag{2.7}$$

This definition yields a convenient separation of variables, such that the spatial and temporal variations of u_h are associated with $u_j(t)$ and $\varphi_j(\mathbf{x})$, respectively.

In the course of differentiation, time derivatives are applied to the nodal values u_j, whereas gradient and divergence operators are applied to the basis functions φ_j. For example, the approximate diffusive flux at a given point $\mathbf{x}$ is proportional to

$$\nabla u_h=\sum_j u_j\nabla\varphi_j. \tag{2.8}$$

The conventional approximation of the convective flux involves the multiplication of (2.7) by the instantaneous velocity that can be interpolated in the same way

$$\mathbf{v}_h=\sum_j \mathbf{v}_j\varphi_j. \tag{2.9}$$

For the time being, the nodal values $\mathbf{v}_j$ are assumed to be known. In real-life applications, they are computed numerically from a momentum balance equation.

The straightforward product rule is not the only and often not the best way to approximate the convective flux that appears in (2.5) and (2.6). Instead of introducing separate trial solutions for each variable, it is possible to interpolate the nodal values of a function that depends on a group of variables. This approximation technique is known as the *group finite element formulation* [109, 110, 111]. It eliminates the need for dealing with products of basis functions and leads to simpler discretized equations, which results in significant savings especially in the case of highly nonlinear problems. In particular, this kind of approximation provides a natural treatment of inviscid fluxes in conservation laws. In many situations, it turns out to be more accurate than the independent approximation of the involved variables [109, 110].

The group finite element approximation of the convective flux is given by

$$(\mathbf{v}u)_h = \sum_j (\mathbf{v}_j u_j)\varphi_j. \tag{2.10}$$

Obviously, this function is easier to differentiate than the product $\mathbf{v}_h u_h$. In particular, the divergence of the convective flux in equation (2.5) is approximated by

$$\nabla \cdot (\mathbf{v}u)_h = \sum_j u_j (\mathbf{v}_j \cdot \nabla \varphi_j). \tag{2.11}$$

That is, the conservative semi-discrete form of the convective term is the sum of nodal values multiplied by the convective derivatives of the basis functions.

If formulation (2.5) is adopted, the resultant semi-discrete problem is as follows

$$\begin{aligned} \int_\Omega w_h \frac{\partial u_h}{\partial t}\,\mathrm{d}\mathbf{x} + \int_\Omega \left(w_h \nabla \cdot (\mathbf{v}u)_h + \nabla w_h \cdot (\varepsilon \nabla u_h)\right)\mathrm{d}\mathbf{x} \\ = \int_\Omega w_h s\,\mathrm{d}\mathbf{x} + \int_{\Gamma_N} w_h (\varepsilon \nabla u_h)\cdot \mathbf{n}\,ds, \end{aligned} \tag{2.12}$$

where the weighting function w_h is taken from a suitable finite-dimensional space.

The finite element discretization based on alternative weak form (2.6) reads

$$\begin{aligned} \int_\Omega w_h \frac{\partial u_h}{\partial t}\,\mathrm{d}\mathbf{x} - \int_\Omega \nabla w_h \cdot \left((\mathbf{v}u)_h - \varepsilon \nabla u_h\right)\mathrm{d}\mathbf{x} \\ = \int_\Omega w_h s\,\mathrm{d}\mathbf{x} - \int_{\Gamma_N} w_h \left((\mathbf{v}u)_h - \varepsilon \nabla u_h\right)\cdot \mathbf{n}\,ds. \end{aligned} \tag{2.13}$$

Remark 2.1. In the case $w_h \equiv 1$ and $\Gamma_N = \Gamma$, the divergence theorem can be used to show that both discretizations reduce to the integral form of the conservation law

$$\frac{\partial}{\partial t}\int_\Omega u_h \mathrm{d}\mathbf{x} + \int_\Gamma \left((\mathbf{v}u)_h - \varepsilon \nabla u_h\right)\cdot \mathbf{n}\,ds = \int_\Omega s\,\mathrm{d}\mathbf{x},$$

which proves that the group finite element formulation is globally conservative.

In the Galerkin method, the weighting functions w_h are taken from the same finite-dimensional space as the basis functions. Substitution of (2.7), (2.8), and (2.11) into (2.12) with $w_h = \varphi_i$ yields the following equation for nodal value u_i

$$\sum_j \left[\int_\Omega \varphi_i \varphi_j \, \mathrm{d}\mathbf{x} \right] \frac{\mathrm{d}u_j}{\mathrm{d}t} + \sum_j \left[\int_\Omega (\varphi_i (\mathbf{v}_j \cdot \nabla \varphi_j) + \nabla \varphi_i \cdot (\varepsilon \nabla \varphi_j)) \, \mathrm{d}\mathbf{x} \right] u_j = \int_\Omega \varphi_i s \, \mathrm{d}\mathbf{x} - \int_{\Gamma_N} \varphi_i h \, ds. \tag{2.14}$$

The value of the flux $h = -(\varepsilon \nabla u) \cdot \mathbf{n}$ is furnished by the Neumann boundary conditions prescribed on Γ_N. If the nodal value u_i is associated with a point lying on $\Gamma_D = \Gamma \backslash \Gamma_N$, then equation (2.14) should be replaced by the algebraic relation $u_i = g_i$ that follows from the Dirichlet boundary conditions. In a practical implementation, it is convenient to do so after the matrix assembly process has been fully completed. Therefore, we will first consider the discretized equations before the imposition of Dirichlet boundary conditions but bear in mind that some of these equations are actually redundant and need to be eliminated to obtain a well-posed discrete problem.

The system of equations (2.14) for all nodes can be written in matrix form as

$$M_C \frac{\mathrm{d}u}{\mathrm{d}t} = Ku + r, \tag{2.15}$$

where $M_C = \{m_{ij}\}$ is the consistent mass matrix, $K = \{k_{ij}\}$ is the negative of the discrete transport operator, and $r = \{r_i\}$ is given by the right-hand side of (2.14)

$$r_i = \int_\Omega \varphi_i s \, \mathrm{d}\mathbf{x} - \int_{\Gamma_N} \varphi_i h \, ds. \tag{2.16}$$

The coefficients of the matrices M_C and K can also be inferred from (2.14)

$$m_{ij} = \int_\Omega \varphi_i \varphi_j \, \mathrm{d}\mathbf{x}, \qquad k_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{v}_j - \varepsilon d_{ij}, \tag{2.17}$$

where the nodal velocities $\mathbf{v}_j$ may vary with time, the diffusion coefficient ε is constant by assumption, and the value of k_{ij} depends on the discretized derivatives

$$\mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \, \mathrm{d}\mathbf{x}, \qquad d_{ij} = \int_\Omega \nabla \varphi_i \cdot \nabla \varphi_j \, \mathrm{d}\mathbf{x}. \tag{2.18}$$

For the discretization based on (2.13), the values of m_{ij} and d_{ij} are the same but

$$k_{ij} = \mathbf{c}_{ji} \cdot \mathbf{v}_j - \varepsilon d_{ij} \tag{2.19}$$

and the contribution of the surface integral to (2.16) should be assembled using the total flux $h = (\mathbf{v}u - \varepsilon \nabla u) \cdot \mathbf{n}$ prescribed on the Neumann boundary part Γ_N.

Remark 2.2. If the boundary value $u = g$ rather than the flux is available, it can be treated as Neumann boundary condition with $h = (\mathbf{v} \cdot \mathbf{n})g$. The advantages of weakly imposed Dirichlet boundary conditions have been explored in [25, 176, 212, 293].

In the notation of definitions (2.16)–(2.19) the $i-$th equation assumes the form

$$\sum_j m_{ij} \frac{\mathrm{d}u_j}{\mathrm{d}t} = \sum_j k_{ij} u_j + r_i, \tag{2.20}$$

where the actual values of k_{ij} and r_i depend on whether (2.12) or (2.13) is employed.

At the initialization stage, the coefficients m_{ij}, $\mathbf{c}_{ij}$, and d_{ij} need to be evaluated using numerical integration. In finite element codes, the matrix assembly process is fully automatic and involves transformations to a reference element on which the local basis functions are defined [76, 176]. In the case of a time-dependent velocity field, the convective part of the discrete operator K must be updated at each time step. On a fixed Eulerian mesh, this can be accomplished in a very efficient way since the coefficients $\mathbf{c}_{ij}$ and d_{ij} do not change as time evolves. If they are stored, the computation of k_{ij} from the above formulas can be performed at a fraction of the cost that the repeated use of element-by-element matrix assembly would require. This is one of the remarkable features peculiar to the group finite element formulation.

The above derivation of the semi-discrete equations is valid for Lagrangian elements of arbitrary shape and order. Traditionally, linear and bilinear approximations have been the workhorse of finite element methods for the equations of fluid dynamics. In many cases, the lack of coercivity or the presence of unresolvable small-scale effects makes the use of higher-order basis functions impractical. Also, the intricate coupling between the degrees of freedom makes it more difficult to control the behavior of numerical solutions than in the case of linear or bilinear elements.

2.1.2 Properties of Discrete Operators

Consistency requires that approximations (2.7) and (2.8) be exact at least for constant functions. To this end, the sum of basis functions must equal 1 everywhere

$$\sum_j \varphi_j \equiv 1. \tag{2.21}$$

Differentiating (2.21), we deduce that the gradients of φ_j must sum to zero

$$\sum_j \nabla \varphi_j \equiv 0. \tag{2.22}$$

This property ensures that the gradient of u_h is zero if u_h is a constant function

$$u_j = c, \quad \forall j \quad \Rightarrow \quad u_h \equiv c, \qquad \nabla u_h \equiv 0.$$

Due to (2.21), the $i-$th row sum of the consistent mass matrix $M_C = \{m_{ij}\}$ is

$$\sum_j m_{ij} = \int_\Omega \varphi_i \sum_j \varphi_j \, \mathrm{d}\mathbf{x} = \int_\Omega \varphi_i \, \mathrm{d}\mathbf{x}.$$

Furthermore, summation over all indices yields the volume (area) of the domain Ω

$$\sum_i \sum_j m_{ij} = \int_\Omega \sum_i \varphi_i \, d\mathbf{x} = |\Omega|. \tag{2.23}$$

The discrete Laplacian $D = \{d_{ij}\}$ is symmetric with zero row and column sums

$$\sum_j d_{ij} = \int_\Omega \nabla\varphi_i \cdot \sum_j \nabla\varphi_j \, d\mathbf{x} = 0, \tag{2.24}$$

$$\sum_i d_{ij} = \int_\Omega \sum_i \nabla\varphi_i \cdot \nabla\varphi_j \, d\mathbf{x} = 0. \tag{2.25}$$

The discrete gradient/divergence operator $\mathbf{C} = \{\mathbf{c}_{ij}\}$ is nonsymmetric and

$$\sum_j \mathbf{c}_{ij} = \int_\Omega \varphi_i \sum_j \nabla\varphi_j \, d\mathbf{x} = 0, \tag{2.26}$$

$$\sum_i \mathbf{c}_{ij} = \int_\Omega \sum_i \varphi_i \nabla\varphi_j \, d\mathbf{x} = \int_\Omega \nabla\varphi_j \, d\mathbf{x}. \tag{2.27}$$

The zero row sum property of $\mathbf{C}$ and D is dictated by the second basic rule from Section 1.6.3. It ensures that the result of numerical differentiation equals zero if all nodal values of the numerical solution coincide and define a constant field.

By definition, finite element basis functions have compact support. That is, φ_i and all of its derivatives are zero outside the set of elements which contain node $\mathbf{x}_i$. Likewise, products of basis functions and/or their derivatives are nonvanishing only on a small patch Ω_{ij} of elements, as shown in Fig. 2.1 for a triangular mesh. Hence, all volume and surface integrals shrink to those over Ω_{ij} and its boundary Γ_{ij}, respectively. This property guarantees that the resulting matrices are sparse.

For example, the assembly of the mass matrix M_C involves the computation of

$$m_{ij} = \int_\Omega \varphi_i \varphi_j \, d\mathbf{x} = \int_{\Omega_{ij}} \varphi_i \varphi_j \, d\mathbf{x}$$

which can be performed element-by-element using numerical or exact integration.

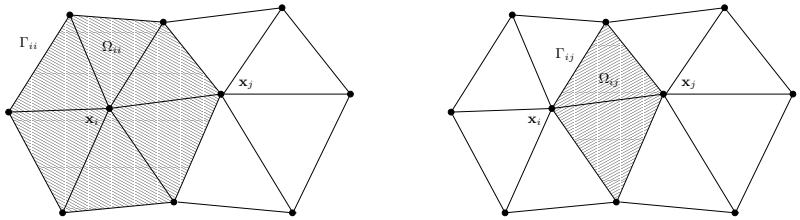


Fig. 2.1 Integration regions for linear basis functions on a triangular mesh.

Assume that at least the mean values of $\varphi_i\varphi_j$ are continuous across interelement boundaries. Then integration by parts within each cell $\Omega_k \in \Omega_{ij}$ reveals that

$$\mathbf{c}_{ij} + \mathbf{c}_{ji} = \int_{\Omega_{ij}} (\varphi_i \nabla\varphi_j + \varphi_j \nabla\varphi_i)\, \mathrm{d}\mathbf{x} = \int_{\Gamma_{ij}\cap\Gamma} \varphi_i\varphi_j\, \mathbf{n}\, \mathrm{d}s. \tag{2.28}$$

The integrals of $\varphi_i\varphi_j$ over an interface between any pair of cells that belong to Ω_{ij} cancel out due to continuity. The integral over the outer boundary of Ω_{ij} is nonvanishing only on $\Gamma_{ij} \cap \Gamma$ since the product of the two basis functions is zero elsewhere. In the case of linear and bilinear elements, formula (2.28) implies that $\mathbf{c}_{ji} = -\mathbf{c}_{ij}$ unless both nodes lie on the boundary Γ and belong to the same element. In particular, $\mathbf{c}_{ii} = 0$ for any interior node i due to the fact that φ_i vanishes on the whole outer boundary Γ_{ii} of the integration region Ω_{ii}, as depicted in Fig. 2.1.

Due to the local support property, relation (2.28) can be reformulated as follows

$$\mathbf{c}_{ij} + \mathbf{c}_{ji} = \mathbf{b}_{ij}, \qquad \mathbf{b}_{ij} = \int_{\Gamma} \varphi_i\varphi_j\, \mathbf{n}\, \mathrm{d}s. \tag{2.29}$$

The vector-valued weights $\mathbf{b}_{ij}$ distribute the surface area between pairs of nodes in much the same way as the coefficients m_{ij} distribute the volume of the domain Ω.

By virtue of (2.29) and (2.26), the column sums (2.27) of $\mathbf{C}$ are equal to

$$\sum_i \mathbf{c}_{ij} = \sum_i \mathbf{b}_{ij} = \mathbf{b}_j, \qquad \mathbf{b}_j = \int_{\Gamma} \varphi_j\, \mathbf{n}\, \mathrm{d}s. \tag{2.30}$$

If the basis function φ_j is associated with an internal node, then it vanishes on the boundary Γ and the $j-$th column sum $\mathbf{b}_j$ equals zero. Otherwise the result is

$$\mathbf{b}_j = \mathbf{n}_j s_j, \qquad \mathbf{n}_j = \frac{1}{s_j}\int_{\Gamma} \varphi_j\, \mathbf{n}\, \mathrm{d}s, \quad s_j = \int_{\Gamma} \varphi_j\, \mathrm{d}s, \tag{2.31}$$

where s_j is the surface area associated with a boundary node j and $\mathbf{n}_j$ is the corresponding unit outward normal vector. Since the basis functions sum to unity

$$\sum_j s_j = \int_{\Gamma} \sum_j \varphi_j\, \mathrm{d}s = |\Gamma|.$$

The above properties of the coefficient matrices M_C, $\mathbf{C}$ and D are not only of purely theoretical interest. As we will see shortly, they are of fundamental importance for the analysis and design of finite element approximations on unstructured meshes.

2.1.3 Conservation and Mass Lumping

The fact that the finite element discretization (2.20) of the transport equation is globally conservative can be easily inferred from the above properties of coefficient

matrices. The total 'mass' of the numerical solution u_h is defined as the integral

$$m(t) = \int_\Omega u_h(\mathbf{x},t)\,\mathrm{d}\mathbf{x} = \sum_i m_i u_i, \tag{2.32}$$

where the row sum m_i represents the share of the domain Ω associated with node i

$$m_i = \int_\Omega \varphi_i\,\mathrm{d}\mathbf{x} = \sum_j m_{ij}. \tag{2.33}$$

Summing equations (2.20) over i, one obtains the global mass balance equation

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \sum_j \sum_i k_{ij} u_j + \sum_i r_i.$$

If the coefficients k_{ij} are defined by (2.19), then the properties of $\mathbf{C}$ and D imply

$$\sum_j \sum_i k_{ij} u_j = \sum_j (\mathbf{v}_j u_j) \cdot \sum_i \mathbf{c}_{ji} - \sum_j u_j \sum_i \varepsilon d_{ij} = 0.$$

Thus, only source terms and fluxes across the boundary may affect the total mass.

The semi-discrete scheme with k_{ij} defined by (2.17) is also conservative since

$$\begin{aligned} \sum_j \sum_i k_{ij} u_j &= -\sum_j (\mathbf{v}_j u_j) \cdot \sum_i (\mathbf{b}_{ij} - \mathbf{c}_{ji}) - \sum_j u_i \sum_i \varepsilon d_{ij} \\ &= -\sum_j (\mathbf{v}_j u_j) \cdot \sum_i \mathbf{b}_{ij} = -\sum_j (\mathbf{v}_j u_j) \cdot \mathbf{b}_j. \end{aligned} \tag{2.34}$$

By definition of the convective flux (2.11) and of the integrated normal vector $\mathbf{b}_j$

$$\sum_j (\mathbf{v}_j u_j) \cdot \mathbf{b}_j = \int_\Gamma (\mathbf{v}u)_h \cdot \mathbf{n}\,\mathrm{d}s.$$

Again, the total mass may change only due to internal sources and boundary fluxes.

Remark 2.3. The imposition of Dirichlet boundary conditions in the strong sense may apparently invalidate the proof of global conservation. Indeed, the elimination of the corresponding equations from the algebraic system means that $w_h = \sum_i \varphi_i \equiv 1$ is no longer an admissible test function, and the column sums of the resultant coefficient matrices are different. This problem can be rectified by including a set of equations for the boundary fluxes weighted by the omitted basis functions [158].

Remark 2.4. Clearly, the global conservation statement remains valid for Dirichlet boundary conditions imposed in a weak sense, e.g., as proposed in Remark 2.2.

It is common practice to approximate the consistent mass matrix M_C by a diagonal matrix M_L, so as to update the solution in a fully explicit way or make an implicit algorithm more robust and efficient. This trick is particularly useful in steady-state

computations since the approximation of the time derivative has no influence on the final solution. In the case of time-dependent problems, mass lumping is usually undesirable since it may adversely affect the phase accuracy of the finite element scheme. Moreover, it may result in a loss of the intrinsic conservation property.

In general, a diagonal mass matrix M_L is a conservative approximation to M_C if

$$\sum_i (M_L u)_i = \sum_i (M_C u)_i \tag{2.35}$$

for an arbitrary vector u. In particular, the sum of its elements should be equal to the volume/area of the computational domain, as required by (2.23) and (2.35) with $u \equiv 1$. These requirements are clearly satisfied by the *row-sum* mass lumping technique, whereby M_L is defined as a diagonal matrix of weights given by (2.33)

$$M_L = \operatorname{diag}\{m_i\}, \qquad m_i = \sum_j m_{ij}. \tag{2.36}$$

By definition, the sum of all nodal values multiplied by the diagonal coefficients of the so-defined lumped mass matrix M_L is equal to the total mass (2.32).

Mass lumping can also be performed using diagonal scaling or inexact evaluation of the coefficients m_{ij} by a nodal quadrature rule that produces a diagonal matrix. In fact, the integrals given by (2.33) can be interpreted as weights of a Newton-Cotes numerical integration scheme. This observation makes it possible to estimate the quadrature error and quantify the loss of accuracy due to mass lumping [134].

The quadrature-based approach to mass lumping is attractive from the viewpoint of analysis but a word of caution is in order. It turns out that only the row-sum formula (2.33) conserves mass in the above sense [134]. Indeed, if we suppose that there is another diagonal mass matrix $\tilde{M}_L = \operatorname{diag}\{\tilde{m}_i\}$ satisfying (2.35) then

$$\sum_i (M_L u)_i - \sum_i (\tilde{M}_L u)_i = \sum_i (m_i - \tilde{m}_i) u_i = 0.$$

Since u is arbitrary, this means that $m_i = \tilde{m}_i$ for all i, which proves the uniqueness of the lumped mass matrix M_L that enjoys the global conservation property.

Linear and bilinear finite element approximations give rise to mass matrices with

$$m_{ij} \geq 0, \quad m_i > 0, \qquad \forall i, j$$

since the basis functions are nonnegative everywhere. In higher-order approximations, the off-diagonal entries of M_C are of variable sign, so row-sum lumping may produce M_L with zero or negative diagonal entries. For example, the lumped mass matrix for the six-node quadratic finite element approximation on a triangular mesh assigns zero masses to all vertex-based degrees of freedom [73]. Such a lumped-mass Galerkin discretization can be interpreted as a finite volume method with edge-centered degrees of freedom [271]. A possible remedy is to enrich the finite element basis by adding a bubble function associated with the center of gravity [73].

2.1.4 Variational Gradient Recovery

In many situations, numerical differentiation of the approximate solution is required to calculate certain derived quantities such as fluxes or curvatures. By definition, the first derivatives of u_h are given by (2.8) but the so-defined gradient ∇u_h is a piecewise-constant function which is discontinuous at interelement boundaries. Therefore, a direct evaluation of the derivatives at nodes is not feasible, and some kind of postprocessing is needed to extract information from the available data.

A continuous approximation to the gradient of the solution u_h can be defined as

$$\mathbf{g}_h = \sum_j \mathbf{g}_j \varphi_j, \tag{2.37}$$

where $\mathbf{g}_j$ is a suitable approximation to the vector of first derivatives at node j. It can be determined, for example, by fitting a polynomial to the values of the piecewise-constant consistent gradient ∇u_h evaluated at a set of points surrounding node j. These points can be placed so as to exploit the *superconvergence* phenomenon [190] and increase the accuracy of approximation by orders of magnitude [362, 363, 364]. Superconvergent postprocessing techniques are frequently employed for purposes of *a posteriori* error estimation and adaptive mesh refinement, see Chapter 7.

Another way to determine the nodal values of $\mathbf{g}_h$ is to perform the L_2-projection

$$\int_\Omega \varphi_i \mathbf{g}_h \,\mathrm{d}\mathbf{x} = \int_\Omega \varphi_i \nabla u_h \,\mathrm{d}\mathbf{x}, \quad \forall i. \tag{2.38}$$

This approach is a classical example of *variational gradient recovery* which can be used repeatedly to calculate the nodal values of higher-order space derivatives.

Remark 2.5. The L_2-projection can also be carried out using a different set of basis/test functions $\{\psi_i\}$ for the representation of the averaged gradient $\mathbf{g}_h$. However, the choice $\psi_i = \varphi_i$ is usually the most natural and economical one [3, 261].

Substitution of (2.8) and (2.37) into (2.38) yields a linear system of the form

$$M_C \mathbf{g} = \mathbf{C}u, \tag{2.39}$$

where $M_C = \{m_{ij}\}$ is the consistent mass matrix and $\mathbf{C} = \{\mathbf{c}_{ij}\}$ is the discrete gradient operator analyzed in the previous sections. As usual, the boldface notation means that the number on unknowns per mesh node equals that of space dimensions.

The linear system (2.39) can be solved, e.g., using Richardson's iteration preconditioned by the lumped mass matrix M_L which is a usable approximation to M_C

$$\hat{\mathbf{g}}^{(m+1)} = \hat{\mathbf{g}}^{(m)} + M_L^{-1}[\mathbf{C}u - M_C \hat{\mathbf{g}}^{(m)}], \qquad m = 0, 1, 2, \ldots. \tag{2.40}$$

Since M_L is a diagonal matrix, its inversion is trivial. Due to the diagonal dominance and positive-definiteness of M_C, convergence is typically fast. In practical calculations, a fixed number (up to 3) of iterations are usually performed. A single iteration

with initial zero guess corresponds to the lumped-mass version of system (2.39)

$$\mathbf{g} = M_L^{-1}\mathbf{C}u. \tag{2.41}$$

This kind of gradient reconstruction is very cheap and its accuracy is sufficient for most practical purposes. Also, the lumped-mass L_2-projection (2.41) is less likely to generate undershoots and overshoots than gradient recovery based on (2.39).

Due to the zero row sum property, the diagonal coefficients of $\mathbf{C}$ are given by

$$\mathbf{c}_{ii} = -\sum_{j\neq i}\mathbf{c}_{ij}.$$

Hence, the equation for the nodal gradient $\mathbf{g}_i$ admits the following representation

$$\mathbf{g}_i = \frac{1}{m_i}\sum_j \mathbf{c}_{ij}u_j = \frac{1}{m_i}\sum_{j\neq i}\mathbf{c}_{ij}(u_j - u_i).$$

Variational gradient recovery, such as the above lumped-mass L_2-projection, can also be used to approximate the divergence of a vector field. For example

$$(\nabla\cdot\mathbf{g})_i = \frac{1}{m_i}\sum_j \mathbf{c}_{ij}\cdot\mathbf{g}_j = \frac{1}{m_i}\sum_{j\neq i}\mathbf{c}_{ij}\cdot(\mathbf{g}_j - \mathbf{g}_i) \tag{2.42}$$

yields an approximate value of the Laplacian Δu at node i. As an alternative, second-order derivatives can be recovered directly using integration by parts [226]

$$\int_\Omega \varphi_i(\Delta u)_h\,\mathrm{d}\mathbf{x} = \int_\Gamma \varphi_i(\mathbf{n}\cdot\nabla u_h)\,\mathrm{d}s - \int_\Omega \nabla\varphi_i\cdot\nabla u_h\,\mathrm{d}\mathbf{x}, \quad \forall i. \tag{2.43}$$

After row-sum mass lumping, the $i-$th equation in this system can be written as

$$(\Delta u)_i = \frac{1}{m_i}\sum_j h_{ij}u_j - \frac{1}{m_i}\sum_{j\neq i} d_{ij}(u_j - u_i),$$

where d_{ij} and h_{ij} represent the contributions of the volume and surface integrals to the right-hand side of (2.43), respectively. Neumann boundary conditions (if any) can be built into the integral over Γ_N. This approach to recovery of $(\Delta u)_i$ is faster and typically more accurate than repeated application of first derivatives.

The divergence of the velocity field $\mathbf{v}_h$ can be approximated in the same way as that of the averaged gradient in equation (2.42). Interestingly enough, the result is proportional to the row sums of the discrete transport operator $K = \{k_{ij}\}$ with coefficients $k_{ij} = -\mathbf{c}_{ij}\cdot\mathbf{v}_j - \varepsilon d_{ij}$. Indeed, using property (2.24) one obtains

$$\sum_j k_{ij} = -\sum_j \mathbf{c}_{ij}\cdot\mathbf{v}_j \approx -m_i(\nabla\cdot\mathbf{v})_i.$$

Loosely speaking, the velocity field is discretely divergence-free if the group finite element approximation (2.10) of the convective flux leads to K with zero row sums.

It is instructive to decompose the $i-$th component of the vector Ku as follows

$$\sum_j k_{ij}u_j = \sum_{j\neq i} k_{ij}(u_j - u_i) + u_i \sum_j k_{ij}. \tag{2.44}$$

In light of the above, the following approximations are associated with each term

$$\sum_j k_{ij}u_j \approx -m_i(\nabla\cdot(\mathbf{v}u))_i - m_i(\varepsilon\Delta u)_i, \tag{2.45}$$

$$\sum_{j\neq i} k_{ij}(u_j - u_i) \approx -m_i(\mathbf{v}\cdot\nabla u)_i - m_i(\varepsilon\Delta u)_i, \tag{2.46}$$

$$u_i\sum_j k_{ij} = -u_i\sum_j \mathbf{c}_{ij}\cdot\mathbf{v}_j \approx -m_i u_i(\nabla\cdot\mathbf{v})_i. \tag{2.47}$$

In conclusion, the sum over $j\neq i$ is the 'incompressible' part of the discrete transport operator K. The 'compressible' part (2.47) vanishes if K has zero row sums.

2.1.5 Treatment of Nonlinear Fluxes

In many practical applications, the diffusive terms can be neglected, while the flux function $\mathbf{f}(u)$ depends on the unknown solution u in a nonlinear way. In this case, equation (2.1) represents a hyperbolic conservation law which can be discretized in much the same way as the convective part of a linear transport equation. Let

$$\mathbf{f}_h = \sum_j \mathbf{f}_j\varphi_j, \qquad \mathbf{f}_j = \mathbf{f}(u_j) \tag{2.48}$$

be the group finite element approximation of the inviscid flux to be inserted into

$$\int_\Omega w_h\frac{\partial u_h}{\partial t}\,\mathrm{d}\mathbf{x} + \int_\Omega w_h\nabla\cdot\mathbf{f}_h\,\mathrm{d}\mathbf{x} = \int_\Omega w_h s\,\mathrm{d}\mathbf{x}. \tag{2.49}$$

Remark 2.6. As before, the global balance equation can be inferred from (2.49) with $w_h = \sum_i\varphi_i \equiv 1$ by applying the divergence theorem element-by-element

$$\frac{\partial}{\partial t}\int_\Omega u_h\,\mathrm{d}\mathbf{x} + \int_\Gamma \mathbf{f}_h\cdot\mathbf{n}\,\mathrm{d}s = \int_\Omega s\,\mathrm{d}\mathbf{x}.$$

The semi-discrete Galerkin equation that corresponds to $w_h = \varphi_i$ is given by

$$\sum_j m_{ij}\frac{\mathrm{d}u_j}{\mathrm{d}t} = -\sum_j \mathbf{c}_{ij}\cdot\mathbf{f}_j + r_i. \tag{2.50}$$

Note that the right-hand side of this equation contains a linear combination of nodal fluxes $\mathbf{f}_j = \mathbf{f}(u_j)$ that depend on the solution values u_j at the corresponding nodes. If an explicit time-stepping strategy is adopted, the fluxes can be evaluated using

the available solution from the previous time level. The matrix assembly process for implicit algorithms involves a suitable linearization of the discretized fluxes.

Due to the zero row sum property (2.26) of the matrix $\mathbf{C} = \{\mathbf{c}_{ij}\}$, we have

$$\sum_j m_{ij} \frac{\mathrm{d}u_j}{\mathrm{d}t} = -\sum_{j \neq i} \mathbf{c}_{ij} \cdot (\mathbf{f}_j - \mathbf{f}_i) + r_i. \tag{2.51}$$

Furthermore, it is possible to transform this equation into the quasi-linear form

$$\sum_j m_{ij} \frac{\mathrm{d}u_j}{\mathrm{d}t} = -\sum_j \mathbf{c}_{ij} \cdot \mathbf{v}_{ij} (u_j - u_i) + r_i, \tag{2.52}$$

where $\mathbf{v}_{ij}$ is a characteristic velocity such that (2.51) and (2.52) are equivalent

$$\mathbf{v}_{ij} = \begin{cases} \frac{\mathbf{f}(u_j) - \mathbf{f}(u_i)}{u_j - u_i}, & \text{if } u_j \neq u_i, \\ \mathbf{f}'(u_i), & \text{if } u_j = u_i. \end{cases} \tag{2.53}$$

As in the one-dimensional case, this definition guarantees that shocks, if any, move at correct speed satisfying the Rankine-Hugoniot condition (see Chapter 3).

In fact, the analytical derivative $\mathbf{f}'(u_i)$ with respect to u can be replaced by zero. The following equivalence relation holds no matter how $\mathbf{v}_{ij}$ is defined for $u_j = u_i$

$$\mathbf{c}_{ij} \cdot \mathbf{v}_{ij} (u_j - u_i) = \mathbf{c}_{ij} \cdot (\mathbf{f}_j - \mathbf{f}_i).$$

To convert (2.52) into the usual form (2.20), the entries of $K = \{k_{ij}\}$ are defined as

$$k_{ii} = -\sum_{j \neq i} k_{ij}, \qquad k_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{v}_{ij}, \quad \forall j \neq i. \tag{2.54}$$

Remark 2.7. Since $\mathbf{v}_{ij} = \mathbf{v}_{ji}$, the relationship between the off-diagonal entries of K follows from (2.29). For any pair of internal nodes, $\mathbf{c}_{ji} = -\mathbf{c}_{ij}$ implies $k_{ji} = -k_{ij}$. Likewise, the symmetric boundary part $\mathbf{b}_{ij}$ makes a symmetric contribution to K.

Remark 2.8. For a linear convective flux $\mathbf{f} = \mathbf{v}u$, the coefficients $k_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{v}_{ij}$ may differ from those given by (2.17) with $\varepsilon = 0$. On the one hand, the above-mentioned preservation of (skew-)symmetry speaks in favor of definition (2.54). On the other hand, the formula $k_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{v}_j$ is certainly more economical for linear problems.

2.1.6 Conservative Flux Decomposition

In finite difference and finite volume methods for conservation laws, the distribution of mass inside the domain satisfies local balance equations formulated in terms of numerical fluxes that transport the mass from one node into another. The mass associated with a given node depends on the sum of incoming and outgoing fluxes.

The advantages of a flux-based formulation are twofold. First, it is guaranteed to be conservative and reflects the physical nature of convection and diffusion processes. Second, the rules of the game are remarkably simple: (i) each pair of neighboring nodes may trade mass, (ii) the mass added to one node is subtracted from another, (iii) mass can be imported or exported across the boundaries of the domain.

The integral nature of the finite element method makes it difficult to ascertain how the mass exchange between individual nodes takes place. Due to global conservation, a decomposition of the residual into numerical fluxes is possible. However, their definition is not as obvious as in the case of finite differences or finite volumes. The discretized convective and/or diffusive terms represent the net amount of mass received by node i from its neighbors. Unfortunately, it is difficult to identify the contribution of each neighbor, and multiple solutions to this problem may exist.

In the early 1990s, a proper definition of numerical fluxes was found for continuous piecewise-linear Galerkin approximations on triangular and tetrahedral meshes [17, 18, 22, 269, 300]. This was a major breakthrough in the development of edge-based finite element methods for compressible flows [225, 226, 239, 259]. Most of such algorithms are fully explicit and rely heavily on certain properties of linear basis functions. As a consequence, they are not suitable for bilinear approximations and other finite elements. This lack of generality rules out the use of quadrilateral, hexahedral, and hybrid meshes which might be desirable for various reasons.

A very useful flux decomposition technique was introduced by Selmin et al. [300, 301, 302] who developed a unified framework for the implementation of unstructured grid methods based on finite element and finite volume discretizations. It provides a simple way to generate numerical fluxes and artificial diffusion operators for Galerkin approximations based on Lagrangian finite elements of various shape and order. Furthermore, element-based data structures are only needed to assemble the matrices of coefficients involved in the computation of numerical fluxes. The transition to a flux-based representation leads to efficient algorithms that operate with node pairs, as in the case of finite difference and finite volume schemes.

Following Selmin et al. [300, 301, 302] and Löhner [226], we consider a finite element approximation to equation (2.1) and define the corresponding fluxes in terms of the off-diagonal coefficients $\mathbf{c}_{ij}$ and d_{ij} associated with the discretization of first-order and second-order derivatives, respectively. The flux decomposition to be presented can serve as a vehicle for extending classical high-resolution schemes to unstructured meshes. Many examples of such generalizations can be found in [239, 243], where the edge-based data structure of Peraire et al. [269] was used to perform upwinding and flux limiting on a triangular mesh of linear finite elements.

2.1.6.1 Inviscid Fluxes

For any linear or nonlinear flux function $\mathbf{f}(u)$, the group finite element discretization of the conservation law (2.1) yields an equation of the form (2.50). The task is to express the right-hand side of this semi-discrete equation in terms of internodal fluxes that take the mass from one node and give it to another node without changing

the global balance. The zero row sum property (2.26) leads to the representation

$$\sum_j \mathbf{c}_{ij}\cdot\mathbf{f}_j = \mathbf{f}_i\cdot\sum_j \mathbf{c}_{ij} + \sum_j \mathbf{c}_{ij}\cdot\mathbf{f}_j = \sum_j \mathbf{c}_{ij}\cdot(\mathbf{f}_i+\mathbf{f}_j). \tag{2.55}$$

Furthermore, the coefficient $\mathbf{c}_{ij}$ can be decomposed into the symmetric internal part $\mathbf{a}_{ij}$ and the skew-symmetric boundary part $\mathbf{b}_{ij}$ given by formula (2.29)

$$\mathbf{c}_{ij} = \frac{\mathbf{a}_{ij}+\mathbf{b}_{ij}}{2}, \qquad \mathbf{a}_{ij} = \mathbf{c}_{ij}-\mathbf{c}_{ji}, \qquad \mathbf{b}_{ij} = \mathbf{c}_{ij}+\mathbf{c}_{ji}. \tag{2.56}$$

The sum of terms proportional to $\mathbf{a}_{ij}$ admits the following flux decomposition

$$\sum_j \mathbf{a}_{ij}\cdot\frac{\mathbf{f}_i+\mathbf{f}_j}{2} = \sum_{j\neq i} f_{ij}, \qquad f_{ij} = \mathbf{a}_{ij}\cdot\frac{\mathbf{f}_i+\mathbf{f}_j}{2}, \quad \forall j\neq i. \tag{2.57}$$

The average of $\mathbf{f}_j$ and $\mathbf{f}_i$ indicates that this formula is a finite element counterpart of the central difference approximation. Since $\mathbf{a}_{ji} = -\mathbf{a}_{ij}$, the flux $f_{ji} = -f_{ij}$ has the same magnitude and opposite sign. This property guarantees mass conservation.

According to (2.29) and (2.48), the contribution of $\mathbf{b}_{ij}$ to (2.55) is represented by

$$\sum_j \mathbf{b}_{ij}\cdot\frac{\mathbf{f}_i+\mathbf{f}_j}{2} = f_{ii}, \qquad f_{ii} = \int_\Gamma \varphi_i \frac{\mathbf{f}_i+\mathbf{f}_h}{2}\cdot\mathbf{n}\,ds, \quad \forall i. \tag{2.58}$$

Here and below, the index $j = i$ is reserved for boundary fluxes. On the Neumann boundary part Γ_N, integration should be performed using $h = \mathbf{f}\cdot\mathbf{n}$ in place of $\mathbf{f}_h\cdot\mathbf{n}$. On the Dirichlet boundary part Γ_D, the $i-$th equation is overwritten by the constraint $u_i = g_i$ which should be used to calculate $\mathbf{f}_i = \mathbf{f}(g_i)$ in (2.57) and elsewhere.

Remark 2.9. The boundary flux f_{ii} can also be evaluated using inexact nodal quadrature or approximation $\mathbf{f}_h \approx \mathbf{f}_i$ which has the same effect as mass lumping [301]

$$f_{ii} \approx \mathbf{b}_i\cdot\mathbf{f}_i, \qquad \mathbf{b}_i = \sum_j \mathbf{b}_{ij} = \mathbf{n}_i s_i, \tag{2.59}$$

where the unit outward normal vector $\mathbf{n}_i$ and the surface area s_i are defined by (2.31). In the worst-case scenario, approximation (2.59) generates an error of order h. On the other hand, the unique definition of the normal at boundary nodes makes the numerical treatment of boundary conditions remarkably simple and natural [301].

Summarizing the results, the sum of internal and boundary fluxes is given by

$$\sum_j \mathbf{c}_{ij}\cdot\mathbf{f}_j = \sum_j f_{ij}. \tag{2.60}$$

The same flux decomposition is obtained if integration by parts is performed to shift the derivatives onto the test function. Indeed, property (2.29) confirms that

$$\sum_j \mathbf{b}_{ij}\cdot\mathbf{f}_j - \sum_j \mathbf{c}_{ji}\cdot\mathbf{f}_j = \sum_j f_{ij}. \tag{2.61}$$

Remark 2.10. Another possibility to define the flux f_{ij} is to take [200, 201, 203]

$$f_{ij} = \mathbf{a}_{ij} \cdot \frac{\mathbf{f}_i + \mathbf{f}_j}{2} - \mathbf{b}_{ij} \cdot \frac{\mathbf{f}_j - \mathbf{f}_i}{2} = \mathbf{c}_{ij} \cdot \mathbf{f}_i - \mathbf{c}_{ji} \cdot \mathbf{f}_j, \qquad \forall j \neq i. \tag{2.62}$$

The difference between the fluxes (2.57) and (2.62) for $j \neq i$ implies that

$$f_{ii} = \sum_j \mathbf{b}_{ij} \cdot \frac{\mathbf{f}_i + \mathbf{f}_j}{2} + \sum_j \mathbf{b}_{ij} \cdot \frac{\mathbf{f}_j - \mathbf{f}_i}{2} = \sum_j \mathbf{b}_{ij} \cdot \mathbf{f}_j, \qquad \forall i.$$

The so-defined boundary flux f_{ii} represents the first term in the left-hand side of (2.61) and approximates the surface integral that results from integration by parts

$$\sum_j \mathbf{b}_{ij} \cdot \mathbf{f}_j = \int_\Gamma \varphi_i \mathbf{f}_h \cdot \mathbf{n} \, \mathrm{d}s.$$

In the interior of the domain, the two definitions of f_{ij} are equivalent since $\mathbf{b}_{ij} = 0$.

Remark 2.11. Flux decompositions (2.60) and (2.61) are valid not only for linear and bilinear basis functions but also for higher-order (quadratic, cubic) finite elements. However, the straightforward two-point flux approximation may cease to reflect the physical nature of transport processes. As the polynomial order increases, so does the number of nonzero matrix entries and possible flux decompositions. Ideally, the definition of numerical fluxes for higher-order FEM should (i) involve solution values at more than two nodes and (ii) allow mass exchange only between nearest neighbors, as in the case of finite difference and finite volume schemes [193].

2.1.6.2 Viscous Fluxes

Several alternative approaches to the numerical treatment of diffusive fluxes such as

$$\mathbf{g} = -\varepsilon \nabla u$$

have been explored by finite element practitioners [226, 239, 300, 301]. The first possibility is to determine the nodal values $\mathbf{g}_i \approx -(\varepsilon \nabla u)_i$ using the L_2-projection

$$\mathbf{g}_i = -\frac{\varepsilon}{m_i} \sum_j \mathbf{c}_{ij} u_j = -\frac{\varepsilon}{m_i} \sum_{j \neq i} \left(\mathbf{a}_{ij} \frac{u_j + u_i}{2} + \mathbf{b}_{ij} \frac{u_j + u_i}{2} \right)$$

(cf. Section 2.1.4) or another gradient recovery technique. The corresponding fluxes g_{ij} can be derived in the same way as their inviscid counterparts, that is,

$$g_{ii} \approx \mathbf{b}_i \cdot \mathbf{g}_i, \qquad g_{ij} = \mathbf{a}_{ij} \cdot \left(\frac{\mathbf{g}_i + \mathbf{g}_j}{2} \right), \qquad \forall j \neq i. \tag{2.63}$$

The use of gradient recovery makes such a formulation akin to finite difference and finite volume schemes. The combined treatment of convective and diffusive terms

leads to conceptually simple and computationally efficient algorithms. A potential drawback to this approach is that the evaluation of g_{ij} involves information from two layers of points [239, 301]. The consistent Galerkin approximation achieves superior accuracy with a compact stencil, i.e., using data from nearest neighbors only. The result is a sparse matrix D which contains the coefficients of diffusive fluxes.

A symmetric matrix $D = \{d_{ij}\}$ is called a *discrete diffusion operator* if it has zero row and column sums [205, 200]. That is, its entries should satisfy the relations

$$\sum_j d_{ij} = \sum_i d_{ij} = 0, \qquad d_{ij} = d_{ji}, \quad \forall i, j. \tag{2.64}$$

For example, consider the standard Galerkin discretization of the diffusive term

$$-\int_\Omega w\nabla\cdot(\varepsilon\nabla u)\,\mathrm{d}\mathbf{x} = \int_\Omega \nabla w\cdot(\varepsilon\nabla u)\,\mathrm{d}\mathbf{x} - \int_{\Gamma_N} w\mathbf{n}\cdot(\varepsilon\nabla u)\,\mathrm{d}s.$$

If the diffusion coefficient ε is constant, this weak statement leads to the formula

$$\int_\Omega \nabla\varphi_i\cdot(\varepsilon\nabla u_h)\,\mathrm{d}\mathbf{x} = \varepsilon\sum_j d_{ij}u_j, \qquad d_{ij} = \int_\Omega \nabla\varphi_i\cdot\nabla\varphi_j\,\mathrm{d}\mathbf{x}.$$

Due to (2.24) and (2.25) the resultant matrix is a typical representative of discrete diffusion operators. The contribution of the surface integral vanishes if homogeneous Neumann boundary conditions are imposed. In general, it can be evaluated using the known value of $\mathbf{n}\cdot(\varepsilon\nabla u)$ or the compact approximation [301]

$$g_{ii} = -\int_{\Gamma_N} \varphi_i\mathbf{n}\cdot(\varepsilon\nabla u)_h\,\mathrm{d}s \approx \mathbf{g}_i\cdot\mathbf{n}_i,$$

where $\mathbf{n}_i$ denotes the outward normal vector defined in the same way as in (2.59).

Remark 2.12. Interestingly enough, the difference between the consistent mass matrix $M_C = \{m_{ij}\}$ and its lumped counterpart $M_L = \mathrm{diag}\{m_i\}$ is also a discrete diffusion operator in the above sense. By definition, $m_{ij} = m_{ji}$ and $m_i = \sum_j m_{ij}$ so that

$$\sum_j (m_{ij} - m_i) = \sum_i (m_{ij} - m_i) = 0.$$

The diffusive nature of $M_C - M_L$ is well known [134] and has been exploited to design various stabilization terms for finite element schemes [26, 87, 232, 301, 302].

Any discrete diffusion operator D defines the conservative flux decomposition

$$(Du)_i = \sum_j d_{ij}u_j = \sum_{j\neq i} g_{ij}, \qquad g_{ij} = d_{ij}(u_j - u_i). \tag{2.65}$$

For any pair of nodes i and j, the skew-symmetric fluxes g_{ij} and $g_{ji} = -g_{ij}$ are proportional to the difference between the two nodal values, which results in smoothing or steepening of solution profiles depending on the sign of the coefficient d_{ij}. As we

will see later, these properties make it possible to adjust the magnitude of the flux g_{ij} so as to enforce monotonicity or remove excessive numerical diffusion.

2.1.7 Relationship to Finite Volumes

To illustrate the relationship between finite element and finite volume discretizations of a conservation law like (2.1), consider an arbitrary control volume V_i with boundary $S_i = \bigcup_j S_{ij}$ which consists of several patches S_{ij}. For all $j \neq i$, the interface between the control volumes V_i and V_j is denoted by $S_{ij} = S_i \cap S_j$. The notation $S_{ii} = S_i \cap \Gamma$ refers to the union of boundary patches that belong to Γ, if any.

The local mass balance for an arbitrary control volume V_i can be written as

$$\frac{\partial}{\partial t}\int_{V_i} u\,\mathrm{d}\mathbf{x} + \int_{S_i} \mathbf{f}\cdot\mathbf{n}\,\mathrm{d}s = \int_{V_i} s\,\mathrm{d}\mathbf{x}. \tag{2.66}$$

The volume of the integration region V_i and the mean value of u are given by

$$m_i = \int_{V_i} \mathrm{d}\mathbf{x} = |V_i|, \qquad u_i = \frac{1}{m_i}\int_{V_i} u\,\mathrm{d}\mathbf{x}.$$

Let r_i denote the right-hand side of (2.66). Splitting the surface integral, one obtains

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} + \sum_j \int_{S_{ij}} \mathbf{f}\cdot\mathbf{n}\,\mathrm{d}s = r_i.$$

In finite volume methods, numerical integration and interpolation are employed to approximate the integral over each portion S_{ij} of the control surface S_i by

$$f_{ij} \approx \int_{S_{ij}} \mathbf{f}\cdot\mathbf{n}\,\mathrm{d}s.$$

A second-order approximation of central difference type is based on the definition

$$f_{ii} = \mathbf{b}_i \cdot \mathbf{f}_i, \qquad f_{ij} = \mathbf{a}_{ij} \cdot \frac{\mathbf{f}_j + \mathbf{f}_i}{2}, \quad \forall j \neq i.$$

The so-defined numerical fluxes f_{ij} are of the same form as those for the group finite element formulation but the metric vectors $\mathbf{n}_i$ and $\mathbf{a}_{ij}$ are defined as

$$\mathbf{b}_i = \int_{S_{ii}} \mathbf{n}\,\mathrm{d}s, \qquad \mathbf{a}_{ij} = \int_{S_{ij}} \mathbf{n}\,\mathrm{d}s, \quad \forall j \neq i. \tag{2.67}$$

It is easy to verify that these quantities possess the following properties [301]

$$\sum_{j\neq i} \mathbf{a}_{ij} + \mathbf{b}_i = 0, \qquad \mathbf{a}_{ji} = -\mathbf{a}_{ij}, \qquad \forall j \neq i. \tag{2.68}$$

The first property reflects the fact that a constant flux cannot change the value of u_i at an interior node. It can be readily inferred from (2.67) and the integral relation

$$\int_{S_i} \mathbf{n}\,\mathrm{d}s = \int_{V_i} \nabla 1\,\mathrm{d}\mathbf{x} = 0, \qquad S_i = \bigcup_j S_{ij}.$$

The second property in (2.68) is a consequence of the fact that the 'outward' normal for V_i is the negative of that for V_j. The skew-symmetry of the coefficient vectors $\mathbf{a}_{ij}$ and $\mathbf{a}_{ji}$ ensures that $f_{ji} = -f_{ij}$, as required by local and global conservation.

Remark 2.13. So far no restrictions have been imposed on the shape of V_i. The degrees of freedom u_i are typically associated with cells or vertices of the underlying mesh. In the latter case, a dual mesh of control volumes covers the whole domain.

Remark 2.14. The original and/or dual mesh are only needed to identify the connections between nodes and generate coefficients that depend on the geometric properties of control volumes. These coefficients are assigned to nodes or node pairs which participate in the mass exchange. A typical implementation involves visiting each pair of nodes and sending the fluxes f_{ij} to the corresponding equations [300].

On an unstructured triangular mesh, the control volume V_i can be built around each node i as sketched in Fig. 2.2. First, each triangle is subdivided into six subelements delimited by the medians whose intersection lies at the center of gravity. Then V_i is defined as the union of subelements having the point $\mathbf{x}_i$ as a vertex. This yields a *median dual mesh* which represents a subdivision of the computational domain into nonoverlapping polygonal cells. The area of each cell V_i equals the diagonal entry m_i of the lumped mass matrix $M_L = \mathrm{diag}\{m_i\}$ for a piecewise linear Galerkin discretization on a triangulation with the same vertices [8, 300]. Moreover, it turns out that the corresponding metric quantities $\mathbf{a}_{ij}$ and $\mathbf{n}_i$ are identical [226, 301]. The same relationship between linear finite elements and vertex-centered finite volume approximations holds for tetrahedral meshes in the three-dimensional case [301].

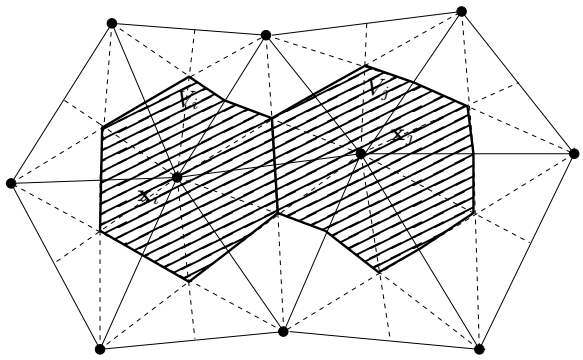


Fig. 2.2 Control volume V_i for a vertex-centered FVM on a triangular mesh.

The bridge between the finite element and finite volume approaches makes it possible to combine their advantages and develop a unified framework for the discretization of transport equations on arbitrary meshes [301]. The flux for a finite element scheme can be manipulated in the same way as that in finite volume methods. Conversely, the consistent mass matrix and the Galerkin approximation of diffusive terms become portable to a finite volume code. Furthermore, the metric coefficients $\mathbf{a}_{ij}$, $\mathbf{b}_i$, and m_i can be calculated using element-by-element matrix assembly or the alternative geometric approach. In the case of bilinear finite elements, there is no equivalent finite volume representation since it is impossible to construct a dual mesh with the same connectivity pattern. However, a conservative flux decomposition and the use of data structures that operate with node pairs are still feasible.

2.1.8 Edge-Based Data Structures

One of the basic tasks involved in the practical implementation of an unstructured grid method is the assembly of sparse matrices and/or matrix-vector products that constitute the algebraic system to be solved. A lion's share of computer time is spent on retrieving information from large arrays and locating array elements which need to be updated. The search process depends on the links between individual elements, nodes, fluxes, and other data items. Hence, the overall performance of the code is strongly influenced by the choice of data structures and storage techniques.

In most finite element codes, global matrices like M_C, $\mathbf{C}$, and D are assembled by visiting all elements sequentially and collecting their contributions to the integrals that define the coefficients of discrete operators. The three basic operations are: *gathering* nodal data pertaining to each element, *processing* this data, and *scattering* the results back to nodes ([226], p. 187). To perform numerical integration, one needs to know the numbers and locations of nodes that belong to a given element. Thus, a typical element-based data structure includes arrays in which the Cartesian coordinates of each node and the connectivity list of each element are stored.

After the matrix assembly, all information about the geometrical and topological properties of the underlying mesh can be discarded. Furthermore, the evaluation of right-hand sides and matrix-vector products can be performed using an edge-based data structure which operates with pairs of nodes, as in the case of finite volume schemes. Such data structures offer algorithmic simplicity, low memory requirements, and a major reduction in indirect addressing [22, 229, 225, 269, 301]. Moreover, they are amenable to large-scale parallel computing [56, 226, 245].

Throughout this book, the term 'edge' refers to a pair of nodes associated with a pair of nonzero off-diagonal matrix entries, i.e., with an edge $\overrightarrow{ij}$ of the graph that represents the sparsity pattern (stencil) of the numerical scheme. In the finite element context, any pair of basis functions with overlapping supports defines such an edge. On a simplex mesh, the number of internodal links for linear finite elements equals the number of physical mesh edges. Bilinear approximations on rectangular

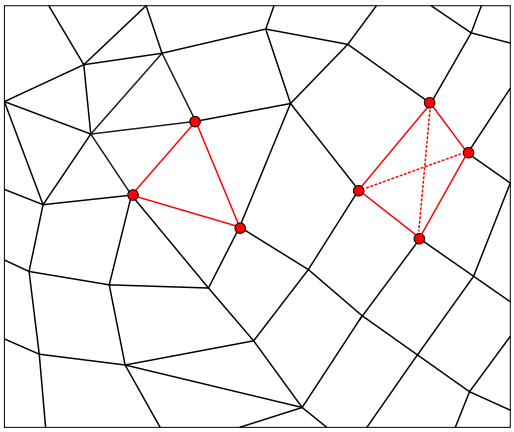


Fig. 2.3 Internodal links for linear and bilinear finite elements.

or mixed meshes give rise to extra diagonal links, see Fig. 2.3. Hence, there is no one-to-one correspondence between the graph of the matrix and that of the mesh.

To get started with edge-based data structures, consider a sparse matrix $A = \{a_{ij}\}$ with `neq` rows/columns and `nnz` entries that may assume nonzero values. If storage is allocated for a_{ij} and/or a_{ji} with $j > i$, then the node pair $\{i, j\}$ defines an edge in the above algebraic sense. The list of edges is stored in a two-dimensional connectivity array $\texttt{kedge}(1:2, 1:\texttt{nedge})$ with two rows and `nedge` columns such that the two node numbers associated with the edge number $\texttt{iedge} = 1, \ldots, \texttt{nedge}$ are

$$\begin{aligned} i &= \texttt{kedge}(1, \texttt{iedge}), \\ j &= \texttt{kedge}(2, \texttt{iedge}). \end{aligned}$$

Remark 2.15. The columnwise storage of edge data is adopted since matrices are stored by columns in MATLAB, FORTRAN, and other programming environments.

Another array $\texttt{aedge}(1:2, 1:\texttt{nedge})$ is used to store the off-diagonal matrix entries

$$\begin{aligned} a_{ij} &= \texttt{aedge}(1, \texttt{iedge}), \\ a_{ji} &= \texttt{aedge}(2, \texttt{iedge}). \end{aligned}$$

The first row of `aedge` is associated with the upper triangular part ($j > i$), while the second row contains the nonzero entries of the lower triangular part ($j < i$). For symmetric matrices, a one-dimensional array `aedge` is enough. A matrix with zero row/column sums is completely defined by the contents of `kedge` and `aedge`. Otherwise, an extra array $\texttt{adiag}(1:\texttt{neq})$ is used to store the diagonal entries

$$a_{ii} = \texttt{adiag}(\texttt{i}).$$

This one-dimensional array is sufficient to represent a diagonal matrix such as M_L.

In summary, an elementary edge-based data structure includes the edge connectivity list and arrays that store the diagonal and/or off-diagonal part of a matrix with up to $\mathtt{nnz} = \mathtt{neq} + 2 * \mathtt{nedge}$ nonzero entries. In what follows, this representation of sparse matrices will be referred to as the Compressed Edge Storage (CES) format.

The following examples illustrate some possibilities to evaluate the matrix-vector product *Au* and add the result to the residual or right-hand side vector *b*. Here and below, pseudo-code segments are written in the FORTRAN 90 terminology which we believe is self-explanatory and readily portable to other programming languages.

Example 2.1. Given a nonsymmetric matrix $\mathtt{A} = \{\mathtt{adiag}, \mathtt{aedge}, \mathtt{kedge}\}$ with non-vanishing row sums, its diagonal entries are processed in a loop over nodes

```
do i = 1,neq
  b(i) = b(i) + adiag(i) * u(i)
end do
```

while the contribution of off-diagonal entries is included in a loop over edges

```
do iedge = 1,nedge
  i = kedge(1,iedge)
  j = kedge(2,iedge)
  b(i) = b(i) + aedge(1,iedge) * u(j)
  b(j) = b(j) + aedge(2,iedge) * u(i)
end do
```

Example 2.2. If the matrix *A* has zero row sums, then a loop over edges is enough

```
do iedge = 1,nedge
  i = kedge(1,iedge)
  j = kedge(2,iedge)
  diff = u(j) - u(i)
  b(i) = b(i) + aedge(1,iedge) * diff
  b(j) = b(j) - aedge(2,iedge) * diff
end do
```

Example 2.3. If the matrix *A* is symmetric with zero row sums, then it represents a discrete diffusion operator. Hence, a conservative flux decomposition is feasible

```
do iedge = 1,nedge
  i = kedge(1,iedge)
  j = kedge(2,iedge)
  flux = aedge(iedge) * (u(j) - u(i))
  b(i) = b(i) + flux
  b(j) = b(j) - flux
end do
```

Other kinds of internodal fluxes can be inserted into the vector *b* in the same way.

The edge-based formulation offers an efficient way to perform sparse matrix-vector multiplications in explicit algorithms and iterative solvers. In the latter case, preconditioners should also be applied edge-by-edge if the CES format is adopted. The design of such preconditioners is addressed in [57, 58]. Reportedly, they are more efficient than their element-by-element counterparts. A further reduction in indirect addressing can be achieved using advanced concepts like stars, superedges, and chains which represent extensions of the single-edge data structure [56, 225].

2.1.9 Compressed Row Storage

A complete transition to an edge-based data structure might be impractical in an existing finite element code that employs a different format for storage of sparse matrices in assembly routines and/or linear solvers. However, the edge-by-edge evaluation of internodal fluxes is nontrivial if the coefficients $\mathbf{c}_{ij}$ and d_{ij} are not available in the CES format. Moreover, the entries of global matrices may depend on edge data, such as the mean velocity $\mathbf{v}_{ij}$ in definition (2.54). Therefore, some parts of the code call for an edge-based implementation and it is essential to have fast access to all matrix entries associated with a given edge. Alternatively, element matrices can be disassembled into edge contributions and combined to form edge matrices [56].

The Compressed Row Storage (CRS) and Compressed Column Storage (CCS) formats are often used in unstructured mesh codes. In either case, all nonzero entries of a matrix A are packed into a one-dimensional array $\texttt{aval}(\texttt{nnz})$. This array is filled row-by-row or column-by-column in the CRS and CCS versions, respectively. In other words, the CRS form of a matrix A is equivalent to the CCS form of A^T. Hence, the decision as to which format is better suited for a given application is a matter of whether the matrix itself or its transposed is needed more frequently. Without loss of generality, we assume the former and discuss the CRS format as implemented in the open-source finite element software library FEAT2D [40].

Given the array $\texttt{aval}(\texttt{nnz})$ of nonzero entries stored row-by-row, two auxiliary integer arrays, $\texttt{kcol}(1:\texttt{nnz})$ and $\texttt{kptr}(1:\texttt{neq}+1)$, make it possible to find an entry with given row and column numbers. The value of $\texttt{j} = \texttt{kcol}(\texttt{ij})$ is the column number of $a_{ij} = \texttt{aval}(\texttt{ij})$ for any index from the range $\texttt{ij} = 1,\ldots,\texttt{nnz}$. The beginning of the i-th row is indicated by $\texttt{ii} = \texttt{kptr}(\texttt{i})$. That is, if $\texttt{i} = 1,\ldots,\texttt{neq}$ is the row number of $a_{ij} = \texttt{aval}(\texttt{ij})$, then $\texttt{kptr}(\texttt{i}) \le \texttt{ij} \le \texttt{kptr}(\texttt{i}+1)-1$. The last element $\texttt{kptr}(\texttt{neq}+1) = \texttt{nnz}+1$ of the row pointer array kptr corresponds to the position where the next row would begin. This entry is introduced for programming convenience in order to avoid treating the last row differently from the others.

For our purposes, it is worthwhile to store the diagonal entry of each row first [40] so that $\texttt{kcol}(\texttt{ii}) = \texttt{i}$ if $\texttt{ii} = \texttt{kptr}(\texttt{i})$. This convention provides fast access to the diagonal part. The remaining column indices are stored in ascending order so that $\texttt{kptr}(\texttt{i})+1 \le \texttt{ij} < \texttt{ik} \le \texttt{kptr}(\texttt{i}+1)-1$ implies $\texttt{kcol}(\texttt{ij}) < \texttt{kcol}(\texttt{ik})$. While this restriction is not a part of the standard CRS format, it plays a pivotal role in the development of the edge-based assembly algorithms to be presented below.

Example 2.4. Consider the following square matrix with $\texttt{nnz} = 12$ nonzero entries

$$A = \begin{bmatrix} \underline{1} & 2 & 0 & 7 \\ 2 & \underline{4} & 3 & 0 \\ 0 & 3 & \underline{6} & 5 \\ 7 & 0 & 5 & \underline{8} \end{bmatrix}.$$

In the sorted CSR format, this 4×4 matrix is represented by the three arrays

$$\begin{aligned} \texttt{aval} &= (\underline{1},2,7,\underline{4},2,3,\underline{6},3,5,\underline{8},7,5), \\ \texttt{kcol} &= (1,2,4,2,1,3,3,2,4,4,1,3), \\ \texttt{kptr} &= (1,4,7,10,13). \end{aligned} \tag{2.69}$$

Note that the underlined diagonal entries of A are stored in positions indicated by kptr and followed by other elements of the same row traversed from left to right.

Example 2.5. Given a matrix $\texttt{A} = \{\texttt{aval},\texttt{kcol},\texttt{kptr}\}$ in the CSR format, the following algorithm returns the sum of the matrix-vector product Au and a vector b

```
do i = 1,neq
  do ij = kptr(i),kptr(i+1)-1
    j = kcol(ij)
    b(i) = b(i) + aval(ij) * u(j)
  end do
end do
```
(2.70)

Example 2.6. If A has zero row sums, an alternative way to evaluate and add Au is

```
do i = 1,neq
  do ij = kptr(i)+1,kptr(i+1)-1
    j = kcol(ij)
    b(i) = b(i) + aval(ij) * (u(j) - u(i))
  end do
end do
```
(2.71)

The edge-by-edge assembly/modification of the matrix $\texttt{A} = \{\texttt{aval},\texttt{kcol},\texttt{kptr}\}$ or its conversion into the CES format $\texttt{A} = \{\texttt{adiag},\texttt{aedge},\texttt{kentry}\}$ involves visiting all node pairs $\{i, j\}$ with $j > i$ and searching for elements of aval that correspond to matrix entries with row/column indices i and j. The positions of the diagonal entries $a_{ii} = \texttt{aval(ii)}$ and $a_{jj} = \texttt{aval(jj)}$ are given by $\texttt{ii} = \texttt{kptr(i)}$ and $\texttt{jj} = \texttt{kptr(j)}$, respectively. Let all node pairs $\{i, j\}$ be processed in the same order in which they are encountered in (2.70) and (2.71). Then the off-diagonal entry $a_{ij} = \texttt{aval(ij)}$ is readily available but the location of $a_{ji} = \texttt{aval(ji)}$ is more difficult to find. In principle, it can be determined by jumping to the beginning of row j and checking if $\texttt{kcol(ji)} = \texttt{i}$ for $\texttt{ji} = \texttt{kptr(j)}+1,\ldots,\texttt{kptr(j+1)}-1$. However, the fact that the elements of kcol are sorted makes it possible to get the value of ji for free.

Let ksep(1 : neq) be an integer array that will serve as a pointer to the position of a_{ji} for the current node pair. The edge-by-edge processing of a matrix and its storage in the CES format can be performed by the following algorithm [254]

```
iedge = 0
ksep(1 : neq) = kptr(1 : neq)
do i = 1,neq
  ii = kptr(i); adiag(i) = aval(ii)
  do ij = ksep(i) + 1,kptr(i + 1) − 1
    j = kcol(ij); ksep(j) = ksep(j) + 1
    ji = ksep(j); jj = kptr(j)
    ...
    iedge = iedge + 1
    kedge(1,iedge) = i
    kedge(2,iedge) = j
    aedge(1,iedge) = aval(ij)
    aedge(2,iedge) = aval(ji)
    ...
  end do
end do
```
(2.72)

Initially, the separator array ksep points to the beginning of each row, where the diagonal entry is stored. For $i = 1$, the next row element (if any) belongs to the upper triangular matrix since there is no $j < 1$. All rows are visited sequentially in the outer loop. First, the value of $a_{ii} =$ aval(ii) is retrieved and packed into adiag(i). Then the lists of edges and off-diagonal entries with row or column index i are extruded in the inner loop. Due to the above sorting convention, the column indices $j =$ kcol(ij) increase with ij. When a new entry $a_{ij} =$ aval(ij) with $j > i$ is encountered, the edge counter iedge and the entry ksep(j) are incremented by 1. The new value of ji = ksep(j) marks the position of $a_{ji} =$ aval(ji). When row j is reached in the outer loop, all rows with $i < j$ have already been processed, so that ksep(j) + 1 is the position of the first row element from the upper triangular part.

Remark 2.16. As an exercise that illustrates how ksep evolves, it is instructive to consider the sample matrix (2.69) and unroll the loops using paper and pencil.

Remark 2.17. Algorithm (2.72) differs from the original version proposed by the author and cited in [254] in that it operates with the upper triangular rather than lower triangular part. This eliminates the need for an extra auxiliary array.

The body of the inner loop may include further statements that involve manipulations with matrix entries indexed by i and j. In particular, the edge-by-edge assembly of aval for a discrete transport operator of the form (2.54) and/or the evaluation of numerical fluxes like (2.57) and (2.65) can be performed in this way. Since the values a_{ij} and a_{ji} are not stored contiguously in aval, the permanent 'jumping' may incur a high overhead cost due to slow memory access. Therefore, it is advisable to use the CES format for storage and edge-by-edge processing of sparse matrices.

2.2 Stabilization of Convective Terms

The standard Galerkin method produces accurate solutions to elliptic and parabolic transport equations as long as the Peclet number is relatively small (some notable exceptions to this rule are considered in Section 4.5). However, the presence of convective terms deprives the Galerkin FEM of the *best approximation property* which it is known to possess in the case of self-adjoint (symmetric) operators.

At high Peclet numbers, the discrete transport operator K is dominated by the nonsymmetric convective part and exhibits very unfavorable properties. Since the Galerkin discretization of convective terms is akin to a central difference approximation, it tends to produce spurious oscillations, also known as *wiggles*. Moreover, an iterative algorithm or an explicit time integration scheme may become unstable.

The lack of robustness can be rectified by adding some artificial diffusion or using modified test functions to construct an upwind-biased finite element scheme. In either case, the Galerkin operator K is replaced by its stabilized counterpart $\bar{K}$ which may or may not be associated with a continuous bilinear form. The difference between the two matrices represents a stabilization operator $D = \{d_{ij}\}$. That is,

$$\bar{K} = K + D, \qquad \bar{k}_{ij} = k_{ij} + d_{ij}. \tag{2.73}$$

Of course, it is essential to ensure that the modified scheme remains consistent and conservative. Furthermore, the application of D should not make it too diffusive.

In this section, we review some traditional stabilization tools developed in an attempt to suppress the wiggles or, at least, to keep them small (bounded). The presentation of linear stabilized FEM is followed by a brief introduction to residual-based shock-capturing techniques for problems with discontinuities and steep fronts. Last but not least, a prototype of nonoscillatory finite element schemes to be presented in Chapter 6 is constructed building on the concept of *modulated dissipation*.

2.2.1 First-Order Upwinding

In one space dimension, the first-order upwind difference scheme (UDS) constitutes a nonoscillatory, albeit inaccurate, alternative to the second-order central difference (CDS) and Galerkin finite element methods (GFEM) which are plagued by numerical instabilities at mesh Peclet numbers greater than 2. The 1D examples in Chapter 2 illustrate the relationship between these basic discretization techniques.

In an unstructured mesh environment, the design of UDS-like finite element methods is typically based on a vertex-centered finite volume approach to the discretization of convective terms [8, 11, 167, 288, 322]. Instead, the inviscid flux (2.57) associated with the standard Galerkin approximation can be replaced by [300]

$$f_{ij} = \mathbf{a}_{ij} \cdot \frac{\mathbf{f}_i + \mathbf{f}_j}{2} - \frac{|\mathbf{a}_{ij} \cdot \mathbf{v}_{ij}|}{2}(u_j - u_i), \qquad \forall j \neq i, \tag{2.74}$$

where $\mathbf{v}_{ij}$ is the averaged velocity defined by (2.53). Substitution into (2.74) yields

$$f_{ij} = \begin{cases} \mathbf{a}_{ij} \cdot \mathbf{f}_i, & \text{if } \mathbf{a}_{ij} \cdot \mathbf{v}_{ij} > 0, \\ \mathbf{a}_{ij} \cdot \mathbf{f}_j, & \text{if } \mathbf{a}_{ij} \cdot \mathbf{v}_{ij} < 0, \end{cases} \qquad \forall j \neq i.$$

The so-defined multidimensional generalization of UDS proves nonoscillatory for arbitrary Peclet numbers [300]. Due to the equivalence between linear finite elements and vertex-centered finite volume CDS, it leads to the same set of discrete equations as the geometric approach based on the construction of a dual mesh.

It is worth mentioning that the practical implementation of first-order upwinding in a finite element code does not require a reformulation of the discrete problem in terms of numerical fluxes. In fact, the replacement of (2.57) by (2.74) is equivalent to adding a discrete diffusion operator $D = \{d_{ij}\}$ with coefficients given by

$$d_{ii} = -\sum_{j \neq i} d_{ij}, \qquad d_{ij} = \frac{|\mathbf{a}_{ij} \cdot \mathbf{v}_{ij}|}{2}, \qquad \forall j \neq i.$$

Note that the matrix D is symmetric with zero row and column sums, as required by (2.64). A general approach to the design of such artificial diffusion operators and to 'discrete upwinding' for FEM on unstructured meshes is presented in Chapter 6.

2.2.2 Artificial Diffusion

First-order upwinding is rarely used in the realm of finite elements because it results in strong smearing and does not fit into the usual variational framework. Instead, the Galerkin transport operator K is typically stabilized by manipulating the bilinear form $a(\cdot,\cdot)$ that defines $k_{ij} = -a(\varphi_i, \varphi_j)$ for any pair of nodes i and j. Consider

$$\bar{a}(w,u) := a(w,u) + b(w,u), \tag{2.75}$$

where $b(u,v)$ represents a linear or nonlinear stabilization term. Ideally, this part should vanish if u is the exact solution of the continuous problem. For practical purposes, it is sufficient to make sure that $b(u,v) \to 0$ as the mesh h goes to zero.

The matrix $\bar{K}$ with coefficients $\bar{k}_{ij} = -\bar{a}(\varphi_i, \varphi_j)$ assumes the form (2.73), where

$$d_{ij} = -b(\varphi_i, \varphi_j).$$

In a classical artificial diffusion method, the stabilization operator is defined by

$$b(w,u) = \sum_k \int_{\Omega_k} \nabla w \cdot (\mathscr{D} \nabla u)\, \mathrm{d}\mathbf{x}, \tag{2.76}$$

where $\mathscr{D}$ denotes a tensor diffusivity. Typically, the amount of artificial diffusion depends on the local mesh size h and on the magnitude of the velocity vector $\mathbf{v}$.

The simplest way to offset the intrinsic negative diffusion of the Galerkin scheme is to apply *isotropic* balancing dissipation of the form $\mathscr{D} = \delta \mathscr{I}$, where

$$\delta = \alpha \frac{|\mathbf{v}|h}{2}, \tag{2.77}$$

is a scalar diffusion coefficient and $\mathscr{I}$ is the identity tensor. The free parameter α determines the magnitude of δ and that of the additional discretization error.

Using a constant value $\alpha \in [0,1]$ is feasible but the resulting scheme is at most first-order accurate, no matter how high the degree of basis functions $\{\varphi_i\}$ is. This dramatic loss of accuracy is clearly undesirable. Therefore, the parameter α is typically defined as a monotonically increasing function of the mesh Peclet number

$$\mathrm{Pe}_h = \frac{|\mathbf{v}|h}{\varepsilon}$$

and evaluated separately for each element Ω_k. The local mesh size h may refer, e.g., to the longest edge or to the diameter of the area/volume equivalent circle (sphere).

As a rule of thumb, the solution is well-resolved and no stabilization is required for $\mathrm{Pe}_h \leq 2$. A larger value of Pe_h indicates that the flow is too fast or the mesh is too coarse. Hence, it is necessary to refine the mesh or apply some artificial diffusion.

Example 2.7. Consider the steady one-dimensional convection-diffusion equation

$$v\frac{\mathrm{d}u}{\mathrm{d}x} + \varepsilon \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} = 0 \tag{2.78}$$

discretized by linear finite elements on a uniform mesh of size $h = \Delta x$. The artificial diffusion method with $\alpha \equiv 1$ corresponds to the first-order upwind approximation of the convective term. On the other hand, the solution to equation (2.78) with constant coefficients is nodally exact if α is defined by [86]

$$\alpha = \coth\left(\frac{\mathrm{Pe}_h}{2}\right) - \frac{2}{\mathrm{Pe}_h}. \tag{2.79}$$

To reduce the computational cost associated with repeated evaluation of α, it is common practice to replace (2.79) by the 'doubly asymptotic' approximation

$$\alpha = \min\left\{1, \frac{\mathrm{Pe}_h}{6}\right\}. \tag{2.80}$$

Neither (2.79) nor (2.80) is guaranteed to be a perfect choice in the case of multidimensional transport equations, variable coefficients, and unstructured meshes. Still, these definitions of the parameter α are frequently used by default in FEM codes [86]. Many other formulas have been proposed but the optimal value of α is highly problem-dependent and difficult to determine from *a priori* considerations.

In advanced artificial diffusion methods, the definition of α may depend not only on the local Peclet number Pe_h but also on the derivatives of the velocity or pressure

fields, on the residuals of the continuous problem, and on other quantities that measure the smoothness of a given solution [170, 172, 226]. The design of such schemes can be based on empirical principles or backed by solid mathematical theory. Both approaches have been employed in CFD computations with considerable success.

2.2.3 Streamline Upwinding

In multidimensional problems, both convection and diffusion transport information in certain directions. The velocity field $\mathbf{v}$ determines the direction and speed of convective transport, whereas the net diffusive flux depends on the definition of $\mathscr{D}$. In the above example, the stabilization term $b(w,u)$ consists of element contributions associated with a weak form of $\delta\Delta u$. The results are frequently polluted by numerical crosswind diffusion that could be removed without making the scheme unstable. This has led the developers of stabilized FEM to introduce *anisotropic balancing dissipation* that acts along the streamlines of $\mathbf{v}$ but not transversely [157, 181].

A family of widespread streamline upwind (SU) finite element methods is based on (2.76) with an anisotropic tensor diffusivity of the form [47, 86, 176, 322]

$$\mathscr{D} = \tau\mathbf{v}\otimes\mathbf{v} = \{\mathscr{D}_{\xi\eta}\}, \tag{2.81}$$

where τ is the so-called *intrinsic time*. The componentwise form of (2.81) reads

$$\mathscr{D}_{\xi\eta} = \tau v_\xi v_\eta, \qquad \forall \xi,\eta \in \{x,y,z\}.$$

After some rearrangements, the corresponding stabilization term (2.76) becomes

$$b(w,u) = \sum_k \int_{\Omega_k} \tau(\mathbf{v}\cdot\nabla w)(\mathbf{v}\cdot\nabla u)\,\mathrm{d}\mathbf{x}. \tag{2.82}$$

Since the integrand represents a weak form of the second convective derivative

$$(\mathbf{v}\cdot\nabla)^2 u = \mathbf{v}\cdot\nabla(\mathbf{v}\cdot\nabla u),$$

the term $b(w,u)$ incorporates *streamline diffusion* (SD) into the Galerkin scheme.

The stabilization parameter τ for the streamline upwind method is defined as

$$\tau = \frac{\delta}{|\mathbf{v}|^2} = \frac{\alpha h}{2|\mathbf{v}|},$$

where δ is given by (2.77) and depends on the choice of α. The default is (2.79) or (2.80), while $\alpha \equiv 1$ corresponds to first-order upwinding along the streamlines.

The lack of crosswind diffusion results in a smaller deviation from the original Galerkin scheme than the use of $\mathscr{D} = \delta\mathscr{I}$. However, the SD stabilization fails to remove undershoots and overshoots in the vicinity of steep layers [172]. Some remedies to this deficiency of streamline upwinding are discussed in Section 2.2.6.

2.2.4 Petrov-Galerkin Methods

Instead of modifying the Galerkin bilinear form $a(\cdot,\cdot)$, streamline diffusion can be introduced within the framework of a consistent Petrov-Galerkin method. Let

$$\mathscr{L}u = \mathbf{v}\cdot\nabla u - \nabla\cdot(\varepsilon\nabla u) + \sigma u = s \qquad \text{in } \Omega \tag{2.83}$$

be the conservative ($\sigma = \nabla\cdot\mathbf{v}$) or nonconservative ($\sigma = 0$) form of a stationary convection-diffusion equation. Using the usual notation for the L_2 scalar product

$$(w,u) = \int_\Omega wu\,\mathrm{d}\mathbf{x},$$

the Galerkin weak form of the above model problem can be written as follows

$$a(w,u) = (w,s).$$

For a pure Dirichlet problem, $w = 0$ on Γ and the bilinear form $a(\cdot,\cdot)$ reads

$$a(w,u) = \sum_k \int_{\Omega_k} (w\mathbf{v}\cdot\nabla u + \nabla w\cdot(\varepsilon\nabla u) + w\sigma u)\,\mathrm{d}\mathbf{x}. \tag{2.84}$$

The inclusion of a SU stabilization term (2.82) transforms this integral form into

$$\begin{aligned}\bar{a}(w,u) &= a(w,u) + \sum_k \int_{\Omega_k} \tau(\mathbf{v}\cdot\nabla w)(\mathbf{v}\cdot\nabla u)\,\mathrm{d}\mathbf{x}\\ &= \sum_k \int_{\Omega_k} (\bar{w}\mathbf{v}\cdot\nabla u + \nabla w\cdot(\varepsilon\nabla u) + w\sigma u)\,\mathrm{d}\mathbf{x},\end{aligned}$$

where the convective term is multiplied by the nonconforming test function

$$\bar{w} = w + \tau\mathbf{v}\cdot\nabla w. \tag{2.85}$$

In the *streamline upwind / Petrov-Galerkin* (SUPG) method [47], this test function is applied to all components of $\mathscr{L}u$ and the bilinear form is redefined as

$$\bar{a}(w,u) = a(u,\bar{w}). \tag{2.86}$$

Unlike classical SU methods, this approach to stabilization of convective terms ensures that the residual of the associated weak form vanishes if u is the exact solution of (2.83). This desirable property is called *strong consistency* and can be exploited to maintain optimal accuracy in a given finite-dimensional space ([276], p. 269).

The advent of SUPG was followed by the development of other Petrov-Galerkin methods that differ in the definition of the operator $\mathscr{P}$ for the stabilization term

$$b(w,u) = \sum_k \int_{\Omega_k} \tau(\mathscr{P}w)(\mathscr{L}u - s)\,\mathrm{d}\mathbf{x}. \tag{2.87}$$

In unsteady problems, the transport operator $\mathscr{L}$ includes the contribution of a (discretized) time derivative. The original SUPG formulation corresponds to

$$\mathscr{P}w = \mathbf{v} \cdot \nabla w.$$

The resulting stabilization term $b(w,u)$ is nonsymmetric, unless $\mathscr{L} = \mathscr{P}$. The symmetry can be restored by using the *Galerkin / least squares* (GLS) method [159]

$$\mathscr{P}w = \mathscr{L}w.$$

Reportedly, a better weighting of the reactive term σu is offered by the formula

$$\mathscr{P}w = -\mathscr{L}^* w,$$

where $\mathscr{L}^*$ stands for the adjoint of $\mathscr{L}$. In this basic *subgrid scale* (SGS) method [86] and in its predecessors [92, 113] the sign of the symmetric terms is reversed.

Remark 2.18. All of the above Petrov-Galerkin methods reduce to the classical SUPG stabilization if $\sigma = 0$ and the test function w is linear inside each element.

Many other definitions of the operator $\mathscr{P}$ are possible. In most cases, the differences between the resulting solutions are marginal. For a comprehensive review and a detailed comparative study of stabilized FEM based on various generalizations or modifications of SUPG, we refer to [70, 86, 172, 276] and references therein.

Remark 2.19. The use of edge-based data structures in the context of SUPG is addressed in [56]. It is shown that element matrices and residuals associated with the stabilized bilinear form can be decomposed into edge contributions. Hence, global matrix assembly and the computation of matrix-vector products can be performed in a loop over edges rather than elements. This approach to implementation of SUPG offers significant savings in terms of the CPU time and memory requirements [56].

2.2.5 Taylor-Galerkin Methods

A major drawback of SU methods and their Petrov-Galerkin counterparts is the uncertainty regarding the choice of the stabilization parameter. The default setting given by formula (2.79) is designed to produce a nodally exact solution to (2.78) but might be inappropriate for more involved convection-diffusion-reaction equations. Guessing the right value of α for time-dependent problems is particularly difficult since both amplitude and phase errors must be taken into account. Fortunately, there is a conceptually simple and parameter-free alternative to streamline upwinding.

An accurate and stable finite element approximation to the transport equation

$$\frac{\partial u}{\partial t} + \mathscr{L}u = s \qquad \text{in } \Omega \tag{2.88}$$

can be constructed within the framework of Taylor-Galerkin (TG) methods [85, 86, 90, 275]. The main advantages of these finite element schemes are their inherent stability, improved phase accuracy, and the absence of free parameters.

2.2.5.1 Second-Order TG Approximation

In Taylor-Galerkin methods, the time discretization plays a pivotal role. Instead of manipulating the Galerkin space discretization, the time-stepping method is chosen so as to stabilize it in natural way, perhaps, under a certain time step restriction. TG algorithms do not contain free parameters and are directly applicable to multidimensional transport problems. To give a simple example that illustrates their relationship to SU/SD methods, consider the second-order accurate approximation [70, 74]

$$\frac{u^{n+1}-u^n}{\Delta t} = \left(\frac{\partial u}{\partial t}\right)^n + \frac{\Delta t}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)^{n+\theta} \tag{2.89}$$

which leads to the explicit ($\theta = 0$) or semi-implicit ($\theta = 1$) Lax-Wendroff / TG2 method. The stability limit (if any) for the time step Δt depends on the structure of the spatial differential operator $\mathscr{L}$ and on the number of space dimensions [89].

Invoking the governing equation (2.88) and assuming that all of its coefficients are independent of t, the involved time derivatives are expressed as follows

$$\frac{\partial u}{\partial t} = s - \mathscr{L}u, \qquad \frac{\partial^2 u}{\partial t^2} = -\mathscr{L}\frac{\partial u}{\partial t} = \mathscr{L}(\mathscr{L}u - s) \tag{2.90}$$

and plugged into equation (2.89). Next, the residual of the semi-discrete scheme

$$\frac{u^{n+1}-u^n}{\Delta t} + \mathscr{L}u^n + \frac{\Delta t}{2}\mathscr{L}(s - \mathscr{L}u^{n+\theta}) = s \tag{2.91}$$

is multiplied by the test function w and integrated over the domain Ω. This yields

$$\left(w, \frac{u^{n+1}-u^n}{\Delta t}\right) + a(u^n, w) + b(u^{n+\theta}, w) = (w, s), \tag{2.92}$$

where the contributions of the time derivatives (2.90) are represented by the terms

$$a(w,u) = (w, \mathscr{L}u - s), \qquad b(w,u) = \frac{\Delta t}{2}(w, \mathscr{L}(s - \mathscr{L}u)).$$

The latter represents an explicit or implicit correction to the forward Euler / Galerkin discretization which makes it more stable and second-order accurate in time.

After integration by parts, the following representation of $b(w,u)$ is obtained

$$b(w,u) = \frac{\Delta t}{2}(\mathscr{L}^* w, s - \mathscr{L}u) = \frac{\Delta t}{2}\int_\Omega (\mathscr{L}^* w)(s - \mathscr{L}u)\,\mathrm{d}\mathbf{x} \tag{2.93}$$

for a pure Dirichlet problem such that $w = 0$ on the boundary Γ. This term is identical to (2.87) with $\tau = \Delta t/2$ and $\mathscr{P} = -\mathscr{L}^*$. Like many other stable FEM approximations, (2.92) turns out to be a disguised Petrov-Galerkin method [70, 290].

Remark 2.20. The equivalence between Taylor-Galerkin and SUPG-like discretizations can be exploited to define the stabilization parameter τ for transient computations, or the local time step Δt for marching the solution to a steady state [38, 74].

2.2.5.2 Linear Hyperbolic Equations

A standard model problem is the linear hyperbolic equation given by (2.88) with

$$\mathscr{L}u = \mathbf{v} \cdot \nabla u.$$

Substitution of the streamline diffusion operator $\mathscr{L}^2$ for $\frac{\partial^2}{\partial t^2}$ in (2.89) leads to

$$b(w,u) = \frac{\Delta t}{2}\left[\int_\Omega \nabla \cdot (\mathbf{v}w)(\mathbf{v}\cdot\nabla u)\,\mathrm{d}\mathbf{x} - \int_\Gamma w\mathbf{v}\cdot\mathbf{n}\,(\mathbf{v}\cdot\nabla u)\,ds\right].$$

In incompressible flow problems, the velocity field is divergence-free ($\nabla\cdot\mathbf{v} = 0$) and the volume integral reduces to the SU stabilization term (2.82) with $\tau = \Delta t/2$

$$b(w,u) = \frac{\Delta t}{2}\left[\int_\Omega (\mathbf{v}\cdot\nabla w)(\mathbf{v}\cdot\nabla u)\,\mathrm{d}\mathbf{x} - \int_\Gamma w\mathbf{v}\cdot\mathbf{n}\,(\mathbf{v}\cdot\nabla u)\,ds\right].$$

When a Dirichlet boundary condition is imposed at the inlet $\Gamma_- = \{\mathbf{x}\in\Gamma\,|\,\mathbf{v}\cdot\mathbf{n} < 0\}$, as required by the nature of hyperbolic problems, the surface integral reduces to that over $\Gamma_+ = \{\mathbf{x}\in\Gamma\,|\,\mathbf{v}\cdot\mathbf{n} > 0\}$. Such boundary terms are missing in the original SU formulation (2.82) but have been incorporated into the GLS method [95]. The failure to include them might cause spurious reflections at outflow boundaries [38, 89].

2.2.5.3 Nonlinear Hyperbolic Equations

Next, consider a generic conservation law of the form (2.1) which corresponds to

$$\mathscr{L}u = \nabla\cdot\mathbf{f}(u)$$

with a nonlinear inviscid flux $\mathbf{f}(u)$. Invoking approximation (2.89), one obtains

$$u^{n+1} = u^n + \Delta t(s - \nabla\cdot\mathbf{f}^n) + \frac{(\Delta t)^2}{2}\nabla\cdot\mathbf{f}_t^{n+\theta}, \tag{2.94}$$

where $\mathbf{f}^n = \mathbf{f}(u^n)$ and $\mathbf{f}_t^n$ is the corresponding time derivative. By the chain rule

$$\mathbf{f}_t = \left(\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}u}\right)\frac{\partial u}{\partial t} = -\mathbf{a}\nabla\cdot\mathbf{f}, \qquad \mathbf{a} = \left(\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}u}\right). \tag{2.95}$$

Again, the Galerkin weak form of the semi-discrete scheme (2.94)–(2.95) admits representation (2.92). Integration by parts in the bilinear form $a(\cdot,\cdot)$ yields

$$a(w,u) = \int_\Gamma w\mathbf{f}(u)\cdot\mathbf{n}\,\mathrm{d}s - \int_\Omega \nabla w\cdot\mathbf{f}(u)\,\mathrm{d}\mathbf{x}. \tag{2.96}$$

By virtue of (2.95), the bilinear form $b(\cdot,\cdot)$ for the second-order term is given by

$$b(w,u) = \frac{\Delta t}{2}\left[\int_\Omega (\mathbf{a}\cdot\nabla w)\nabla\cdot\mathbf{f}(u)\,\mathrm{d}\mathbf{x} - \int_\Gamma w\,(\mathbf{a}\cdot\mathbf{n})\nabla\cdot\mathbf{f}(u)\,\mathrm{d}s\right]. \tag{2.97}$$

In an implicit TG algorithm, the term $b(u^{n+\theta},w)$ may be linearized using [226]

$$\mathbf{f}^{n+\theta} \approx \mathbf{f}^n + \theta\mathbf{a}^n(u^{n+1}-u^n).$$

For $\theta = 0$, the contribution of $a(u^n,w)$ and $b(u^n,w)$ to (2.92) can be written as [87]

$$\bar{a}(u^n,w) = \int_\Gamma w\mathbf{f}^{n+1/2}\cdot\mathbf{n}\,\mathrm{d}s - \int_\Omega \nabla w\cdot\mathbf{f}^{n+1/2}\,\mathrm{d}\mathbf{x}, \tag{2.98}$$

where the flux $\mathbf{f}^{n+1/2}$ represents a second-order accurate approximation to $\mathbf{f}(u^{n+1/2})$

$$\mathbf{f}^{n+1/2} = \mathbf{f}^n + \frac{\Delta t}{2}\mathbf{f}_t^n = \mathbf{f}^n - \frac{\Delta t}{2}\mathbf{a}^n\nabla\cdot\mathbf{f}^n. \tag{2.99}$$

Note that there is no need to differentiate the (possibly discontinuous) flux function in (2.96) and (2.98). However, the evaluation of (2.97) and (2.99) might be cumbersome and expensive, especially in the case of nonlinear hyperbolic systems in which the scalar characteristic speed **a** is replaced by a Jacobian matrix [230]. This is why a fractional-step approach to the treatment of such problems is frequently adopted.

2.2.5.4 Two-Step Implementation

An alternative implementation [6, 231] of the explicit second-order Taylor-Galerkin method for (2.1) is based on the two-step Runge-Kutta time-stepping scheme

$$u^{n+1/2} = u^n + \frac{\Delta t}{2}\left(\frac{\partial u}{\partial t}\right)^n = u^n + \frac{\Delta t}{2}(s-\nabla\cdot\mathbf{f}^n), \tag{2.100}$$

$$u^{n+1} = u^n + \Delta t\left(\frac{\partial u}{\partial t}\right)^{n+1/2} = u^n + \Delta t(s-\nabla\cdot\mathbf{f}^{n+1/2}). \tag{2.101}$$

In contrast to (2.99), the flux $\mathbf{f}^{n+1/2}$ is evaluated using the intermediate solution

$$\mathbf{f}^{n+1/2} = \mathbf{f}(u^{n+1/2}). \tag{2.102}$$

The finite difference discretization of (2.100)–(2.101) is known as the *Richtmyer scheme*. In the 1980s, a finite element version of this popular predictor-corrector algorithm was developed [6, 7, 231] and combined with the flux-corrected transport (FCT) methodology to enforce monotonicity on unstructured meshes [232, 233].

The weak form of the midpoint rule corrector (2.101) can be written as follows

$$\left(w, \frac{u^{n+1}-u^n}{\Delta t}\right) + \bar{a}(u^n, w) = (w, s), \tag{2.103}$$

where $\bar{a}(\cdot,\cdot)$ is the bilinear form defined by (2.98). The only difference as compared to the one-step TG2 method (2.92) is that the intermediate flux $\mathbf{f}^{n+1/2}$ is defined by (2.102) rather than (2.99). Donea and Huerta ([86], p. 158) perform matrix assembly for the discrete counterpart of (2.103) using the strong form of (2.100) to calculate the divergence of $\mathbf{f}^{n+1/2}$ at the numerical integration points. Furthermore, they illustrate the influence of various flux representations by numerical examples.

Alternatively, the Richtmyer-TG scheme can be implemented using a usual finite element discretization for the first step. The corresponding weak form reads

$$\left(w, \frac{u^{n+1/2}-u^n}{\Delta t/2}\right) + a(u^n, w) = (w, s), \tag{2.104}$$

where $a(\cdot,\cdot)$ is the Galerkin bilinear form (2.96). The second step is given by

$$\left(w, \frac{u^{n+1}-u^n}{\Delta t}\right) + a(u^{n+1/2}, w) = (w, s). \tag{2.105}$$

The approximate solutions $u^{n+1/2}$ and u^{n+1} may belong to different finite dimensional spaces. The use of equal-order interpolations widens the stencil of the discrete scheme and is not suitable for steady-state computations since the standard Galerkin approximation $a(w,u) = (w,s)$ is recovered for $u = u^n = u^{n+1/2} = u^{n+1}$. Therefore, it is common practice to use piecewise-constant basis/test functions for $u^{n+1/2}$ and (multi-)linear elements for u^{n+1} [87, 231]. This strategy eliminates the need for extra stabilization at steady state [226]. For the linear convection equation in 1D, it leads to the same algebraic system as the TG scheme based on (2.99).

The reader is referred to Löhner et al. [226, 231, 232] for a detailed presentation of the two-step Taylor-Galerkin method (2.104)–(2.105) including practical implementation details and examples that demonstrate the advantages of this approach.

2.2.5.5 Edge-Based Formulation

In an edge-based finite element code, the Richtmyer-TG scheme can be formulated in terms of numerical fluxes. The discrete counterpart of (2.102) is [226, 228]

$$f_{ij}^{n+1/2} = \mathbf{a}_{ij} \cdot \mathbf{f}(u_{ij}^{n+1/2}), \tag{2.106}$$

where $\mathbf{a}_{ij}$ denotes the vector of weights given by (2.56) and $u_{ij}^{n+1/2}$ is defined by

$$u_{ij}^{n+1/2} = \frac{u_i^n + u_j^n}{2} + \frac{\Delta t}{2}\left(\frac{\partial u}{\partial t}\right)_{ij}^n.$$

The value of the time derivative at the edge midpoint can be approximated, e.g., by

$$\left(\frac{\partial u}{\partial t}\right)_{ij} = \frac{s_i + s_j}{2} - \frac{(\nabla\cdot\mathbf{f})_i + (\nabla\cdot\mathbf{f})_j}{2},$$

where $(\nabla\cdot\mathbf{f})_i$ and $(\nabla\cdot\mathbf{f})_j$ are obtained using (2.42) with $\mathbf{f}$ in place of $\mathbf{g}$. Note that the resulting flux (2.106) depends on the solution values at more than two nodes.

As a cheap alternative, the following simplification can be envisaged [226, 228]

$$\left(\frac{\partial u}{\partial t}\right)_{ij} = \frac{s_i + s_j}{2} - \frac{(\mathbf{x}_i - \mathbf{x}_j)\cdot(\mathbf{f}_i - \mathbf{f}_j)}{|\mathbf{x}_i - \mathbf{x}_j|^2}. \tag{2.107}$$

This definition involves a straightforward extension of the 1D approximation

$$\left(\frac{\partial f}{\partial x}\right)_{ij} \approx \frac{f_j - f_i}{\Delta x}, \qquad j = i+1.$$

While the use of formula (2.107) in two or three dimensions seems to lack a theoretical justification, it makes the edge-based TG algorithm very efficient [226].

2.2.6 Discontinuity Capturing

Linear stability of a numerical scheme is insufficient to guarantee that discontinuities and fronts are resolved in a nonoscillatory fashion. All finite element methods that rely on streamline diffusion (SD) stabilization of convective terms are known to produce undershoots and overshoots in regions where the solution gradients are steep and not aligned with the flow direction. In some situations, imperfections are small in magnitude and can be tolerated. In other cases, it is essential to ensure that the numerical solution remains nonnegative and/or devoid of spurious oscillations.

Nonoscillatory finite element approximations can be constructed by increasing the amount of numerical dissipation in regions where linear stabilization is insufficient. This process is commonly referred to as *discontinuity-capturing* or *shock-capturing* [69, 161, 290] even if the solution is differentiable but exhibits abrupt changes across thin layers. Since the location of layers is generally unknown, it must be determined using suitable smoothness sensors such as gradients or residuals.

Discontinuity capturing (DC) involves the construction of a nonlinear artificial diffusion operator and adding its contribution to the stabilization term

$$\overline{b}(w,u) = b(w,u) + c(w,u).$$

The extra dissipation $c(w,u)$ should be designed so as to control the solution gradient ∇u not only along the streamlines but in all relevant directions. Consider [86]

$$c(w,u) = \sum_k \int_{\Omega_k} \hat{\tau}(\hat{\mathbf{v}} \cdot \nabla w) \mathscr{R}(u) \, \mathrm{d}\mathbf{x}, \tag{2.108}$$

where $\hat{\tau}$ is another free parameter, $\hat{\mathbf{v}}$ is a solution-dependent vector function, and

$$\mathscr{R}(u) = \mathscr{L}u - s$$

is the residual of the governing equation. Obviously, an additional term of the form (2.108) does not destroy the strong consistency of a Petrov-Galerkin formulation.

The definition of $\hat{\mathbf{v}}$ proposed by Hughes and Mallet [161] for $|\nabla u| \neq 0$ reads

$$\hat{\mathbf{v}} = \left(\frac{\mathbf{v} \cdot \nabla u}{|\nabla u|^2} \right) \nabla u, \tag{2.109}$$

which corresponds to a projection of the velocity $\mathbf{v}$ onto the gradient of u. The resultant vector field $\hat{\mathbf{v}}$ is called the *effective transport velocity* [290] since

$$\hat{\mathbf{v}} \cdot \nabla u = \mathbf{v} \cdot \nabla u. \tag{2.110}$$

Of course, there is no need for any discontinuity capturing if the solution u is constant. Therefore, $\hat{\mathbf{v}} = \mathbf{0}$ is the natural setting for the degenerate case $|\nabla u| = 0$.

The combination of (2.108) and (2.109) gives rise to isotropic artificial dissipation, as shown by the following representation of the nonlinear DC term

$$c(w,u) = \sum_k \int_{\Omega_k} \nu(u) \nabla w \cdot \nabla u \, \mathrm{d}\mathbf{x}, \tag{2.111}$$

where the coefficient $\nu(u)$ depends on the gradient and residual of u as follows

$$\nu(u) = \begin{cases} \hat{\tau} \left(\frac{\mathbf{v} \cdot \nabla u}{|\nabla u|^2} \right) \mathscr{R}(u), & \text{if } |\nabla u| \neq 0, \\ 0, & \text{if } |\nabla u| = 0. \end{cases}$$

An obvious drawback to this kind of artificial viscosity is the fact that it becomes negative and may destabilize the finite element discretization if $(\mathbf{v} \cdot \nabla u)\mathscr{R}(u) < 0$.

To ensure that the sign of $\nu(u)$ is correct, the corresponding vector $\hat{\mathbf{v}}$ can be redefined using the residual instead of the convective derivative [117, 136, 177]

$$\hat{\mathbf{v}} = \left(\frac{\mathscr{R}(u)}{|\nabla u|^2} \right) \nabla u.$$

Substitution into (2.108) yields a dissipative term of the form (2.108), where

$$\nu(u) = \begin{cases} \hat{\tau} \left(\frac{\mathscr{R}(u)}{|\nabla u|} \right)^2, & \text{if } |\nabla u| \neq 0, \\ 0, & \text{if } |\nabla u| = 0. \end{cases}$$

As before, the nonlinearity of $\nu(u)$ stems from the definition of $\hat{\mathbf{v}}$ in terms of ∇u.

In the context of SUPG methods, a reasonable value of $\hat{\tau}$ is even more difficult to find than that of the linear stabilization parameter τ. A typical setting is [160]

$$\hat{\tau} = \max\{0, \tau(\hat{\mathbf{v}}) - \tau(\mathbf{v})\},$$

where the dependence of $\tau(\mathbf{v})$ on $\mathbf{v}$ is the same as in the linear term $b(w,u)$. Thus, no extra numerical diffusion is added in the case when $\hat{\mathbf{v}} = \mathbf{v}$ is parallel to ∇u.

Alternatively, an anisotropic DC operator can be designed to act in the crosswind direction only. The removal of streamline diffusion from (2.111) leads to [69]

$$c(w,u) = \sum_k \int_{\Omega_k} \nu(u) \left(\nabla w \cdot \nabla u - \frac{(\mathbf{v}\cdot\nabla w)(\mathbf{v}\cdot\nabla u)}{|\mathbf{v}|^2} \right) \mathrm{d}\mathbf{x}.$$

This nonlinear crosswind diffusion term is equivalent to (2.76) with $\mathscr{D}$ given by

$$\mathscr{D} = \nu(u) \left(\mathscr{I} - \frac{\mathbf{v}\otimes\mathbf{v}}{|\mathbf{v}|^2} \right),$$

where $\mathscr{I}$ denotes the unit tensor and it is tacitly assumed that $|\mathbf{v}| \neq 0$. The use of anisotropic artificial diffusion makes it possible to avoid excessive damping.

De Sampaio and Coutinho [290] combine the streamline diffusion and discontinuity capturing operators within the framework of Petrov-Galerkin and Taylor-Galerkin methods for unsteady convection-diffusion equations. To this end, they blend the flow velocity $\mathbf{v}$ and effective transport velocity $\hat{\mathbf{v}}$ as follows

$$\bar{\mathbf{v}} = \gamma\mathbf{v} + (1-\gamma)\hat{\mathbf{v}}, \qquad 0 \leq \gamma \leq 1.$$

To ensure that $\hat{\mathbf{v}}$ is well defined, the setting $\gamma = 1$ is used whenever $|\nabla u| = 0$. The dissipative term $\bar{b}(w,u)$ differs from its linear counterpart $b(w,u)$ in that the convective derivatives are evaluated using the above *combined velocity* $\bar{\mathbf{v}}$ rather than $\mathbf{v}$. This replacement is consistent since $\bar{\mathbf{v}}\cdot\nabla u = \mathbf{v}\cdot\nabla u$ at the continuum level [290].

Many alternative definitions of $\bar{b}(w,u)$ and of the involved parameters can be found in the literature. We refer to John et al. [172, 174] for a comprehensive review of the state of the art and a detailed comparative study of available techniques.

2.2.7 Interior Penalty Methods

A promising new approach to linear stabilization and discontinuity capturing involves the addition of *interior penalty* terms that control the jumps of the solution gradient. This far-reaching idea was originally proposed by Douglas and Dupont [91] and recently revived by Burman and Hansbo [49, 50] who called it *edge stabilization* and put it on a firm theoretical basis. In recent years, the finite element community has come to recognize and explore the potential of such schemes [293, 327].

Edge stabilization methods penalize the jumps of the normal derivative via [50]

$$b(w,u) = \sum_{k\neq l} \int_{\Gamma_{kl}} \gamma h^2 [\mathbf{n}\cdot\nabla w]\,[\mathbf{n}\cdot\nabla u]\,\mathrm{d}s, \tag{2.112}$$

where γ is a free parameter and $[\![q]\!]$ denotes the jump of a given quantity q across the common boundary $\Gamma_{kl} = \bar{\Omega}_k \cap \bar{\Omega}_l$ of two adjacent mesh elements Ω_k and Ω_l.

In 3D problems, the interface Γ_{kl} is not an edge but a face of the computational mesh. Hence, *continuous interior penalty* (CIP) is a more appropriate name for stabilization via jump terms of the form (2.112). It is also worth mentioning that the (geometric) definition of an edge differs from that given earlier in this section.

In contrast to Petrov-Galerkin and Taylor-Galerkin methods, the symmetric CIP term (2.112) is independent of the underlying equation and vanishes if $[\![\mathbf{n}\cdot\nabla u]\!] = 0$. The amount of stabilization is determined solely by the smoothness of the numerical solution and the results are typically rather insensitive to the choice of γ. The price to be paid for these benefits is a denser matrix with a nonstandard sparsity pattern.

The discontinuity capturing term for a CIP method can be defined as follows [52]

$$c(w,u) = \sum_{k\neq l} \delta(u) \sum_{E\in\Gamma_{kl}} \int_E \frac{(\mathbf{e}\cdot\nabla w)(\mathbf{e}\cdot\nabla u)}{|\mathbf{e}\cdot\nabla u|}\,\mathrm{d}e, \tag{2.113}$$

where $\mathbf{e}$ is a vector parallel to an edge E of Γ_{kl}. In two space dimensions, the second sum consists of a single term that represents a line integral over $\Gamma_{kl} = E$.

Burman and Ern [52] employ the following definition of the smoothness sensor

$$\delta(u) = \hat{\gamma} h \int_{\Gamma_{kl}} |[\![\mathbf{n}\cdot\nabla u]\!]|\,\mathrm{d}s \tag{2.114}$$

and prove that no spurious maxima/minima are generated if $\hat{\gamma}$ is sufficiently large.

Remark 2.21. The edge/face integrals that appear in (2.112)–(2.114) are commonly evaluated by the midpoint rule which is exact in the case of linear finite elements.

2.2.8 Modulated Dissipation

In the late 1980s, the diffusive nature of row-sum mass lumping for (multi-)linear finite elements was exploited to construct explicit Taylor-Galerkin schemes that introduce *modulated dissipation* in the vicinity of discontinuities and steep fronts. In these methods, the added *mass diffusion* admits a conservative flux decomposition which facilitates an extension of flux-corrected transport (FCT) algorithms and total variation diminishing (TVD) schemes to finite elements [87, 298, 299]. As explained in Chapter 4, the basic idea behind such high-resolution schemes is to suppress dispersive ripples using a combination of (linear) first-order numerical diffusion and (nonlinear) balancing *antidiffusion*. The latter must be controlled by a suitable sensor/limiter capable of detecting discontinuities and steep gradients.

2.2.8.1 High-Order Scheme

Consider a time-dependent conservation law (2.1) and its weak form associated with a linearly stable (Taylor-Galerkin or Petrov-Galerkin) finite element scheme

$$\left(w, \frac{u^{n+1}-u^n}{\Delta t}\right) + \overline{a}(w, u^n) = (w, s).$$

Inserting the finite-dimensional (linear or multilinear) counterparts of u and w, one ends up with a linear system that relates the vectors of new and old nodal values

$$M_C u^{n+1} = M_C u^n + \Delta t r^n, \tag{2.115}$$

where M_C is the consistent mass matrix and r is the vector of nodal increments

$$m_{ij} = (\varphi_i, \varphi_j), \qquad r_i = \sum_j \overline{k}_{ij} u_j, \qquad \overline{k}_{ij} = \overline{a}(\varphi_i, \varphi_j).$$

The replacement of M_C by its lumped counterpart M_L, as defined in (2.36), is feasible but degrades the phase accuracy of FEM for transient problems. Note that the use of M_C in the left-hand side of (2.115) results in an implicit coupling of the degrees of freedom, although the underlying time discretization is fully explicit. However, the symmetry and diagonal dominance of M_C make it possible to solve (2.115) efficiently by the following iterative algorithm proposed by Donea et al. [85]

$$M_L u^{(m+1)} = M_C u^n + \Delta t r^n + (M_L - M_C) u^{(m)}, \qquad m = 0, \ldots, L-1. \tag{2.116}$$

The first and last iterate are given by $u^{(0)} = u^n$ and $u^{n+1} = u^{(L)}$, respectively. The derivation of this preconditioned Richardson's scheme is based on an approximate factorization of the consistent mass matrix [85, 275]. In the context of explicit Taylor-Galerkin schemes, the three-pass solver ($L = 3$) was found to be optimal.

2.2.8.2 Low-Order Scheme

Next, consider (2.116) with $L = 1$ and $u^{(0)} = 0$. The resulting explicit approximation

$$M_L u^{n+1} = M_C u^n + \Delta t r^n \tag{2.117}$$

corresponds to the original finite element scheme (2.115) with mass lumping in the left-hand side only. For our purposes, it is worthwhile to write (2.117) in the form

$$M_L u^{n+1} = M_L u^n + \Delta t r^n + (M_C - M_L) u^n. \tag{2.118}$$

As already mentioned in Section 2.1.6.2, the matrix $M_C - M_L$ represents a discrete diffusion operator. Its contribution renders the solution to problem (2.118) nonoscillatory (for sufficiently small time steps) but results in a dramatic loss of accuracy.

Remark 2.22. For 1D hyperbolic problems, the low-order counterpart (2.118) of the TG2 scheme corresponds to the Lax-Wendroff method with added dissipation. It is stable and monotone [140, 298] for Courant numbers ν in the range $|\nu| \leq \sqrt{2/3}$.

2.2.8.3 Selective Mass Lumping

As explained in Section 2.1.6.2, the explicit mass diffusion built into the right-hand side of (2.118) can be expressed as a conservative sum of internodal fluxes

$$(M_C u - M_L u)_i = \sum_j m_{ij} u_j - m_i u_i = \sum_{j \neq i} m_{ij}(u_j - u_i). \tag{2.119}$$

The amount of numerical diffusion can be reduced using *modulation coefficients* $\alpha_{ij} \in [0,1]$ to replace $M_C - M_L$ by another symmetric matrix $D = \{d_{ij}\}$ such that

$$d_{ii} = -\sum_{j \neq i} d_{ij}, \qquad d_{ij} = (1 - \alpha_{ij}) m_{ij}, \quad \forall j \neq i.$$

The setting $\alpha_{ij} \equiv 0$ corresponds to $D = M_C - M_L$, while the use of $0 < \alpha_{ij} \leq 1$ leads to a less diffusive approximation. The matrix form of the resulting scheme reads

$$M_L u^{n+1} = M_L u^n + \Delta t r^n + D u^n. \tag{2.120}$$

The last term vanishes for $\alpha_{ij} \equiv 1$, which corresponds to standard mass lumping

$$M_L u^{n+1} = M_L u^n + \Delta t r^n. \tag{2.121}$$

Varying the modulation coefficients α_{ij} between 0 and 1, it is possible to switch between (2.117) and (2.121) in a conservative fashion. This strategy can be interpreted as selective mass lumping in the right-hand side of the low-order scheme (2.117).

2.2.8.4 Two-Stage Implementation

If the problem at hand is truly time-dependent, it is advantageous to split a finite element scheme with modulated dissipation into two stages, so as to separate convective transport from added mass diffusion as follows [298, 299, 87]

$$M_C u^H = M_C u^n + \Delta t r^n, \tag{2.122}$$

$$M_L u^{n+1} = (M_L + D) u^H. \tag{2.123}$$

The first stage corresponds to (2.115) and inherits its superb phase accuracy. The superscript H stands for "high-order." At the second stage, the amplitudes are corrected by adding a certain amount of mass diffusion in regions where steep gradients are detected by the smoothness sensor built into the definition of the coefficients α_{ij}.

The high-order solution $u^{n+1} = u^H$ is obtained for $D = 0$. The associated low-order scheme corresponds to $D = M_C - M_L$ and $u^{n+1} = M_L^{-1} M_C u^H$. Importantly, neither first-order nor modulated dissipation introduces any phase errors. That is, the predictor u^H is smeared but not displaced. This is what makes (2.122)–(2.123) more attractive than the one-step implementation (2.120) of selective mass lumping.

2.2.8.5 Modulation Coefficients

In essence, the use of modulated mass diffusion is a discontinuity capturing technique that operates at the fully discrete level. The overall complexity and performance of the resulting scheme depend on the philosophy behind the computation of the modulation coefficients α_{ij}. First or second derivatives of flow variables can be used to construct empirical smoothness sensors but the presence of a free parameter undermines the practical utility of such schemes. Alternatively, the values of α_{ij} can be determined using Zalesak's fully multidimensional FCT algorithm [226, 232, 299] or symmetric TVD limiters [87, 299]. Below we outline the former approach since it is easier to implement and more accurate for unsteady problems.

Explicit FCT algorithms add limited antidiffusion to a nonoscillatory low-order solution u^L. In the present context, this "transported and diffused" solution is computed from (2.120) or (2.123) with $D = M_C - M_L$. It follows that u^{n+1} is given by

$$m_i u_i^{n+1} = m_i u_i^L + \sum_{j \neq i} \alpha_{ij} m_{ij} (u_i - u_j). \tag{2.124}$$

The last term consists of limited antidiffusive fluxes which are evaluated using the solution $u = u^n$ for (2.120) and $u = u^H$ for (2.123). The correction factors α_{ij} are chosen so that u_i^{n+1} is bounded by the local extrema $u_i^{\max}$ and $u_i^{\min}$ defined by

$$u_i^{\max} = \max_{j \in \mathscr{S}_i} u_j^L, \qquad u_i^{\min} = \min_{j \in \mathscr{S}_i} u_j^L,$$

where $\mathscr{S}_i = \{ j \,|\, m_{ij} \neq 0 \}$ is the stencil of node i. The definition of α_{ij} for (2.124) is based on the following algorithm [226, 355] which is known as *Zalesak's limiter*

1. Compute the sums of positive/negative raw antidiffusive fluxes $f_{ij} = m_{ij}(u_i - u_j)$

$$P_i^+ = \sum_{j \neq i} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{j \neq i} \min\{0, f_{ij}\}.$$

2. Define the upper/lower bounds such that no spurious maxima/minima can emerge

$$Q_i^+ = m_i (u_i^{\max} - u_i^L), \qquad Q_i^- = m_i (u_i^{\min} - u_i^L).$$

3. Evaluate the correction factors α_{ij} and α_{ji} for each pair of antidiffusive fluxes

$$R_i^{\pm} = \min\left\{1, \frac{Q_i^{\pm}}{P_i^{\pm}}\right\}, \quad \alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\}, & \text{if } f_{ij} \geq 0, \\ \min\{R_i^-, R_j^+\}, & \text{if } f_{ij} < 0. \end{cases}$$

This parameter-free definition of α_{ij} gurantees that u^{n+1} given by (2.124) satisfies

$$u_i^{\min} \leq u_i^{n+1} \leq u_i^{\max}, \quad \forall i.$$

We refrain from going into detail at this point because an in-depth presentation of finite element FCT schemes based on Zalesak's limiter will follow in Section 4.4.

2.3 Discontinuous Galerkin Methods

Discontinuous Galerkin (DG) methods [66, 67, 108, 147] represent one of the most promising current trends in computational fluid dynamics. The frequently mentioned advantages of this approach include local conservation and the ease of constructing high-order approximations on unstructured meshes. Moreover, DG methods are well suited for *hp*-adaptivity and parallelization.

One of the major bottlenecks in the design of high-order DG methods for convection-dominated transport problems is the lack of reliable mechanisms that ensure nonlinear stability and effectively suppress spurious oscillations. A number of successful discontinuity capturing and slope limiting techniques are available for DG finite element methods [37, 48, 68, 154, 186, 188, 320] and their finite difference/volume counterparts [19, 251, 350, 342]. However, no universally applicable methodology has been developed to date. Since the accuracy of monotonicity-preserving schemes degenerates to first order at local extrema, free parameters or heuristic indicators are frequently employed to distinguish between troubled cells and regions where the solution varies smoothly. In some cases, the results leave a lot to be desired. Also, the use of limiters may cause severe convergence problems in steady state computations [342].

In this section, we present a parameter-free, non-clipping slope limiter [195] for high-resolution DG-FEM on arbitrary meshes. A hierarchical approach to adaptive *p*-coarsening is pursued. The Taylor series form [237, 251, 350] of a polynomial shape function is considered, and the involved derivatives are limited so as to control the variations of lower-order terms. The corresponding upper and lower bounds are defined using the data from elements sharing a vertex. This strategy yields a remarkable gain of accuracy, as compared to traditional compact limiters that search the von Neumann (common face) neighbors of a given element [19, 68, 188].

2.3.1 Upwind DG Formulation

A simple model problem that will serve as a vehicle for our presentation of slope-limited DG approximations is the linear convection equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0 \qquad \text{in } \Omega, \tag{2.125}$$

where $u(\mathbf{x},t)$ is a scalar quantity transported by a continuous velocity field $\mathbf{v}(\mathbf{x},t)$. Let $\mathbf{n}$ denote the unit outward normal to the boundary Γ of the domain Ω. The initial and boundary conditions are given by

$$u|_{t=0} = u_0, \qquad u|_{\Gamma_{\text{in}}} = g, \qquad \Gamma_{\text{in}} = \{\mathbf{x} \in \Gamma \,|\, \mathbf{v}\cdot\mathbf{n} < 0\}.$$

Multiplying (2.125) by a sufficiently smooth test function w, integrating over Ω, and using Green's formula, one obtains the following weak formulation

$$\int_\Omega \left(w\frac{\partial u}{\partial t} - \nabla w \cdot \mathbf{v}u \right) \Delta x + \int_\Gamma wu\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}s = 0, \qquad \forall w. \tag{2.126}$$

In the discontinuous Galerkin method, the domain Ω is decomposed into a finite number of cells Ω_e, and a local polynomial basis $\{\varphi_j\}$ is employed to define the restriction of the approximate solution $u_h \approx u$ to Ω_e via

$$u_h(\mathbf{x},t)|_{\Omega_e} = \sum_j u_j(t)\varphi_j(\mathbf{x}), \qquad \forall \mathbf{x} \in \Omega_e. \tag{2.127}$$

The globally defined u_h is piecewise-polynomial and may have jumps at interelement boundaries. The meaning of the coefficients u_j depends on the choice of the basis functions. A local version of (2.126) can be formulated as

$$\int_{\Omega_e} \left(w_h\frac{\partial u_h}{\partial t} - \nabla w_h \cdot \mathbf{v}u_h \right) \Delta x + \int_{\Gamma_e} w_h\hat{u}_h\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}s = 0, \qquad \forall w_h, \tag{2.128}$$

where w_h is an arbitrary test function from the DG space spanned by φ_i. Since u_h is multiply defined on Γ_e, the surface integral is calculated using the solution value $\hat{u}_h$ from the upwind side of the interface, that is,

$$\hat{u}_h(\mathbf{x},t)|_{\Gamma_e} = \begin{cases} \lim\limits_{\delta\to+0} u_h(\mathbf{x}+\delta\mathbf{n},t), & \mathbf{v}\cdot\mathbf{n} < 0,\ \mathbf{x} \in \bar{\Omega}\backslash\Gamma_{\text{in}}, \\ g(\mathbf{x},t), & \mathbf{v}\cdot\mathbf{n} < 0,\ \mathbf{x} \in \Gamma_{\text{in}}, \\ \lim\limits_{\delta\to+0} u_h(\mathbf{x}-\delta\mathbf{n},t), & \mathbf{v}\cdot\mathbf{n} \geq 0,\ \mathbf{x} \in \bar{\Omega}. \end{cases} \tag{2.129}$$

In the case of a piecewise-constant approximation, the result is equivalent to the first-order accurate upwind finite volume scheme. The DG formulation for general conservation laws and systems thereof is described, e.g., in [67, 68].

2.3.2 Taylor Basis Functions

In a discontinuous Galerkin method of degree $p \geq 0$, the shape function $u_h|_{\Omega_e}$ is given by (2.127), where the number of basis functions depends on p. Clearly, many alternative representations are possible, and some choices are better than others. For accuracy and efficiency reasons, it is worthwhile to consider an orthogonal basis

such that M is a diagonal matrix and its inversion is trivial. For example, tensor products of Legendre polynomials are commonly employed on quadrilaterals and hexahedra [37]. The Gram-Schmidt orthonormalization procedure [108, 365], Dubiner's basis functions [48, 147], and Bernstein-Bézier [102] polynomials are suitable for the construction of hierarchical approximations on triangular meshes. In general, one set of basis functions may be used for matrix assembly and another for limiting or visualization purposes. Due to the local nature of DG methods, conversion between a pair of alternative bases is straightforward and relatively efficient.

Following Luo *et al.* [237], we restrict our discussion to quadratic polynomials $u_h|_{\Omega_e} \in P_2(\Omega_e)$ and consider the 2D Taylor series expansion

$$\begin{aligned} u_h(x,y) = u_c &+ \left.\frac{\partial u}{\partial x}\right|_c (x-x_c) + \left.\frac{\partial u}{\partial y}\right|_c (y-y_c) + \left.\frac{\partial^2 u}{\partial x^2}\right|_c \frac{(x-x_c)^2}{2} \\ &+ \left.\frac{\partial^2 u}{\partial y^2}\right|_c \frac{(y-y_c)^2}{2} + \left.\frac{\partial^2 u}{\partial x \partial y}\right|_c (x-x_c)(y-y_c) \end{aligned} \tag{2.130}$$

about the centroid (x_c, y_c) of a cell Ω_e. Introducing the volume averages

$$\bar{u}_h = \frac{1}{|\Omega_e|}\int_{\Omega_e} u_h \Delta x, \qquad \overline{x^n y^m} = \frac{1}{|\Omega_e|}\int_{\Omega_e} x^n y^m \Delta x,$$

the quadratic function u_h can be expressed in the equivalent form [237, 251, 350]

$$\begin{aligned} u_h(x,y) = \bar{u}_h &+ \left.\frac{\partial u}{\partial x}\right|_c (x-x_c) + \left.\frac{\partial u}{\partial y}\right|_c (y-y_c) \\ &+ \left.\frac{\partial^2 u}{\partial x^2}\right|_c \left[\frac{(x-x_c)^2}{2} - \overline{\frac{(x-x_c)^2}{2}}\right] + \left.\frac{\partial^2 u}{\partial y^2}\right|_c \left[\frac{(y-y_c)^2}{2} - \overline{\frac{(y-y_c)^2}{2}}\right] \\ &+ \left.\frac{\partial^2 u}{\partial x \partial y}\right|_c \left[(x-x_c)(y-y_c) - \overline{(x-x_c)(y-y_c)}\right]. \end{aligned} \tag{2.131}$$

This representation has led Luo *et al.* [237] to consider the local Taylor basis

$$\begin{aligned} &\varphi_1 = 1, \quad \varphi_2 = \tfrac{x-x_c}{\Delta x}, \quad \varphi_3 = \tfrac{y-y_c}{\Delta y}, \quad \varphi_4 = \tfrac{(x-x_c)^2}{2\Delta x^2} - \overline{\tfrac{(x-x_c)^2}{2\Delta x^2}}, \\ &\varphi_5 = \tfrac{(y-y_c)^2}{2\Delta y^2} - \overline{\tfrac{(y-y_c)^2}{2\Delta y^2}}, \quad \varphi_6 = \tfrac{(x-x_c)(y-y_c) - \overline{(x-x_c)(y-y_c)}}{\Delta x \Delta y}. \end{aligned} \tag{2.132}$$

The scaling by $\Delta x = (x_{\max} - x_{\min})/2$ and $\Delta y = (y_{\max} - y_{\min})/2$ is required to obtain a well-conditioned system [237]. The normalized degrees of freedom are proportional to the cell mean value $\bar{u}_h$ and derivatives of u_h at (x_c, y_c)

$$\begin{aligned} u_h(x,y) = \bar{u}_h\varphi_1 &+ \left(\left.\frac{\partial u}{\partial x}\right|_c \Delta x\right)\varphi_2 + \left(\left.\frac{\partial u}{\partial y}\right|_c \Delta y\right)\varphi_3 + \left(\left.\frac{\partial^2 u}{\partial x^2}\right|_c \Delta x^2\right)\varphi_4 \\ &+ \left(\left.\frac{\partial^2 u}{\partial y^2}\right|_c \Delta y^2\right)\varphi_5 + \left(\left.\frac{\partial^2 u}{\partial x \partial y}\right|_c \Delta x \Delta y\right)\varphi_6. \end{aligned} \tag{2.133}$$

Note that the cell averages are decoupled from other degrees of freedom since

$$\int_{\Omega_e} \varphi_1^2 \Delta x = |\Omega_e|, \qquad \int_{\Omega_e} \varphi_1 \varphi_j \Delta x = 0, \qquad 2 \le j \le 6.$$

On a uniform mesh of rectangular elements, the whole Taylor basis (2.132) is orthogonal, as shown by Cockburn and Shu [68]. On a triangular mesh, this is not the case even for the linear part $\{\varphi_1, \varphi_2, \varphi_3\}$ since the L_2 inner product of φ_2 and φ_3 is nonvanishing. However, the consistent mass matrix M may be 'lumped' by setting all off-diagonal entries equal to zero. In contrast to the case of a typical Lagrange basis, this modification is conservative because it does not affect the decoupled equation for the mean value of u_h in Ω_e.

2.3.3 The Barth-Jespersen Limiter

The above Taylor series representation is amenable to p-adaptation and limiting. In the context of finite volume and DG finite element methods, a slope limiter is a postprocessing filter that constrains a polynomial shape function to stay within certain bounds. Many unstructured grid codes employ the algorithm developed by Barth and Jespersen [19] for piecewise-linear data. Given a cell average $\overline{u}_h = u_c$ and the gradient $(\nabla u)_c$, the goal is to determine the maximum admissible slope for a constrained reconstruction of the form

$$u_h(\mathbf{x}) = u_c + \alpha_e (\nabla u)_c \cdot (\mathbf{x} - \mathbf{x}_c), \qquad 0 \le \alpha_e \le 1, \quad \mathbf{x} \in \Omega_e. \tag{2.134}$$

Barth and Jespersen [19] define the correction factor α_e so that the final solution values at a number of control points $\mathbf{x}_i \in \Gamma_e$ are bounded by the maximum and minimum centroid values found in Ω_e or in one of its neighbors Ω_a having a common boundary (edge in 2D, face in 3D) with Ω_e. That is,

$$u_e^{\min} \le u(\mathbf{x}_i) \le u_e^{\max}, \qquad \forall i. \tag{2.135}$$

Due to linearity, the solution u_h attains its extrema at the vertices $\mathbf{x}_i$ of the cell Ω_e. To enforce condition (2.135), the correction factor α_e is defined as [19]

$$\alpha_e = \min_i \begin{cases} \min\left\{1, \frac{u_e^{\max} - u_c}{u_i - u_c}\right\}, & \text{if } u_i - u_c > 0, \\ 1, & \text{if } u_i - u_c = 0, \\ \min\left\{1, \frac{u_e^{\min} - u_c}{u_i - u_c}\right\}, & \text{if } u_i - u_c < 0, \end{cases} \tag{2.136}$$

where $u_i = u_c + (\nabla u)_c \cdot (\mathbf{x}_i - \mathbf{x}_c)$ is the unconstrained solution value at $\mathbf{x}_i$.

The above algorithm belongs to the most popular and successful limiting techniques, although its intrinsic non-differentiability tends to cause severe convergence problems at steady state [251, 342]. Another potential drawback is the elementwise definition of $u_e^{\max}$ and $u_e^{\min}$ which implies that

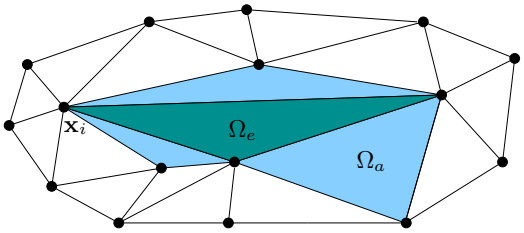


Fig. 2.4 Vertices and neighbors of Ω_e on a triangular mesh.

- the bounds for $u(\mathbf{x}_i)$ satisfying (2.135) at a vertex $\mathbf{x}_i$ depend on the element number e and may be taken from neighbors that do not contain $\mathbf{x}_i$,
- no constraints are imposed on the difference between the solution values in elements meeting at a vertex but having no common edge/face,
- the results are rather sensitive to the geometric properties of the mesh.

In particular, problems are to be expected if Ω_e has sharp angles, as in Fig. 2.4.

2.3.4 The Vertex-Based Limiter

In light of the above, the accuracy of limited reconstructions can be significantly improved if the bounds for variations $u_i - u_c$ at the vertices of Ω_e are constructed using the maximum and minimum values in the elements containing the vertex $\mathbf{x}_i$. The so-defined $u_i^{\max}$ and $u_i^{\min}$ may be initialized by a small/large constant and updated in a loop over elements Ω_e as follows:

$$u_i^{\max} := \max\{u_c, u_i^{\max}\}, \qquad u_i^{\min} := \min\{u_c, u_i^{\min}\}. \tag{2.137}$$

The elementwise correction factors α_e for (2.134) should guarantee that

$$u_i^{\min} \le u(\mathbf{x}_i) \le u_i^{\max}, \qquad \forall i. \tag{2.138}$$

This vertex-based condition can be enforced in the same way as (2.135)

$$\alpha_e = \min_i \begin{cases} \min\left\{1, \frac{u_i^{\max}-u_c}{u_i-u_c}\right\}, & \text{if } u_i - u_c > 0, \\ 1, & \text{if } u_i - u_c = 0, \\ \min\left\{1, \frac{u_i^{\min}-u_c}{u_i-u_c}\right\}, & \text{if } u_i - u_c < 0. \end{cases} \tag{2.139}$$

Obviously, the only difference as compared to the classical Barth-Jespersen (BJ) limiter is the use of $u_i^{\max}$ in place of $u_e^{\max}$. This subtle difference turns out to be the key to achieving high accuracy with p-adaptive DG methods.

In fact, the revised limiting strategy resembles the elementwise version of the finite element flux-corrected transport (FEM-FCT) algorithm developed by Löhner *et al.* [232]. In explicit FCT schemes, $u_i^{\max}$ and $u_i^{\min}$ represent the local extrema of a low-order solution. In accordance with the local discrete maximum principle for unsteady problems, data from the previous time level can also be involved in the estimation of admissible upper/lower bounds.

2.3.5 Limiting Higher-Order Terms

The quality of the limiting procedure is particularly important in the case of a high-order DG method [186]. Poor accuracy and/or lack of robustness restrict the practical utility of many parameter-dependent algorithms and heuristic generalizations of limiters tailored for piecewise-linear functions.

Following Yang and Wang [350], we multiply all derivatives of order p by a common correction factor $\alpha_e^{(p)}$. The limited counterpart of (2.131) becomes

$$\begin{aligned} u_h(x,y) = \bar{u}_h &+ \alpha_e^{(1)} \left\{ \left.\frac{\partial u}{\partial x}\right|_c (x-x_c) + \left.\frac{\partial u}{\partial y}\right|_c (y-y_c) \right\} \\ &+ \alpha_e^{(2)} \left\{ \left.\frac{\partial^2 u}{\partial x^2}\right|_c \left[\frac{(x-x_c)^2}{2} - \overline{\frac{(x-x_c)^2}{2}} \right] + \left.\frac{\partial^2 u}{\partial y^2}\right|_c \left[\frac{(y-y_c)^2}{2} - \overline{\frac{(y-y_c)^2}{2}} \right] \right. \\ &\left. + \left.\frac{\partial^2 u}{\partial x \partial y}\right|_c \left[(x-x_c)(y-y_c) - \overline{(x-x_c)(y-y_c)} \right] \right\}. \end{aligned} \tag{2.140}$$

In our method, the values of $\alpha_e^{(1)}$ and $\alpha_e^{(2)}$ are determined using the vertex-based or standard BJ limiter, as applied to the linear reconstructions

$$u_x^{(2)}(x,y) = \left.\frac{\partial u}{\partial x}\right|_c + \alpha_x^{(2)} \left\{ \left.\frac{\partial^2 u}{\partial x^2}\right|_c (x-x_c) + \left.\frac{\partial^2 u}{\partial x \partial y}\right|_c (y-y_c) \right\}, \tag{2.141}$$

$$u_y^{(2)}(x,y) = \left.\frac{\partial u}{\partial y}\right|_c + \alpha_y^{(2)} \left\{ \left.\frac{\partial^2 u}{\partial x \partial y}\right|_c (x-x_c) + \left.\frac{\partial^2 u}{\partial y^2}\right|_c (y-y_c) \right\}, \tag{2.142}$$

$$u^{(1)}(x,y) = \bar{u}_h + \alpha_e^{(1)} \left\{ \left.\frac{\partial u}{\partial x}\right|_c (x-x_c) + \left.\frac{\partial u}{\partial y}\right|_c (y-y_c) \right\}. \tag{2.143}$$

The last step is identical to (2.134). In the first and second step, first-order derivatives with respect to x and y are treated in the same way as cell averages, while second-order derivatives represent the gradients to be limited.

Since the mixed second derivative appears in (2.141) and (2.142), the correction factor $\alpha_e^{(2)}$ for the limited quadratic reconstruction (2.140) is defined as

$$\alpha_e^{(2)} = \min\{\alpha_x^{(2)}, \alpha_y^{(2)}\}. \tag{2.144}$$

The first derivatives are typically smoother and should be limited using

$$\alpha_e^{(1)} := \max\{\alpha_e^{(1)}, \alpha_e^{(2)}\} \tag{2.145}$$

to avoid the loss of accuracy at smooth extrema. It is important to implement the limiter as a hierarchical p-coarsening algorithm, as opposed to making the assumption [68] that no oscillations are present in u_h if they are not detected in the linear part. In general, we begin with the highest-order derivatives (cf. [186, 350]) and calculate a nondecreasing sequence of correction factors

$$\alpha_e^{(p)} := \max_{p \le q} \alpha_e^{(q)}, \qquad p \ge 1. \tag{2.146}$$

As soon as $\alpha_e^{(q)} = 1$ is encountered, no further limiting is required since definition (2.146) implies that $\alpha_e^{(p)} = 1$ for all $p \le q$. Remarkably, there is no penalty for using the maximum correction factor. At least for scalar equations, discontinuities are resolved in a sharp and nonoscillatory manner.

For a numerical study of the above slope limiter, we refer to [195].

2.4 Summary

In this chapter, we dealt with the design of unstructured grid methods for scalar transport equations. The first part was concerned with the standard Galerkin finite element approximation of conserved variables, fluxes, and derived quantities. The analysis of discrete operators has shown that they possess some interesting and useful properties. The aspects of mass conservation were discussed in some detail. The semi-discrete scheme was expressed in terms of numerical fluxes, and a relationship to finite volumes was established. Last but not least, edge-based algorithms and data structures were introduced as an alternative to the traditional element-by-element programming strategy. The above concepts and tools lend themselves to numerical simulation of compressible flows and convection-dominated transport problems.

The second part was devoted to finite element approximations of convective terms. A survey of representative Petrov-Galerkin and Taylor-Galerkin schemes was included to introduce the key ideas but the reader may want to consult, e.g., the recent book by Donea and Huerta [86] for further details and numerical examples. Also, we have presented a hierarchical slope limiter for discontinuous Galerkin methods. The aspects of flux/slope limiting for continuous finite elements are addressed in Chapter 4, where we pursue an algebraic approach to the design of artificial diffusion operators on the basis of generalized FCT and TVD algorithms.

Chapter 3
Maximum Principles

In this chapter, we elaborate on the qualitative behavior of solutions to multidimensional equations of elliptic, hyperbolic, and parabolic type. We analyze the properties of differential operators and derive *a priori* bounds that depend on the initial and/or boundary conditions. Maximum and minimum principles are formulated for each PDE model. If we include a proof, we try to keep it rigorous but simple. The obtained estimates lead to a set of algebraic and geometric constraints on the coefficients of the numerical scheme and the shape of mesh elements, respectively.

In particular, we consider a handy generalization of Harten's TVD theorem to multidimensional discretizations on unstructured meshes. We show that the nonnegativity of off-diagonal coefficients is sufficient for the space discretization of an unsteady transport equation to be local extremum diminishing (LED) and/or positivity-preserving. Furthermore, we address the implications of the time-stepping method and the properties of discrete operators. The third basic rule, as postulated in Section 1.6.3.3, is reinforced by criteria based on the concept of monotone matrices. The material to be covered provides the theoretical background and useful design criteria for the derivation of algebraic flux correction schemes in the next chapter.

3.1 Properties of Linear Transport Models

The theory of partial differential equations makes it possible to perform a detailed analysis and validation of the mathematical models we are interested in. A particularly useful and important analytical tool is the *maximum principle* which also implies *positivity preservation*. In the absence of zeroth-order terms, solutions to some elliptic PDEs of second order are known to attain their maxima and minima on the boundary of the domain. If a positive source is included, the solution cannot assume a negative value at any interior point if nonnegative boundary data are prescribed. In unsteady problems of hyperbolic and parabolic type, the initial time level represents another inflow boundary of the space-time domain. Therefore, the upper and lower bounds for the exact solution may also be influenced by the choice of initial data.

There are several reasons for the importance of maximum principles. On the one hand, they usually represent certain physical constraints that should apply to a given mathematical model. On the other hand, useful information about the solutions of differential solutions becomes available, even if the solutions themselves are unknown. Upper/lower bounds, uniqueness proofs, and comparison principles can be obtained using elementary calculus. Last but not least, discrete maximum principles play an important role in the development of numerical methods, so we feel that a structured and self-contained review of their continuous counterparts is in order.

In this section, we restrict ourselves to a study of transport problems from Section 1.3. For simplicity, we assume that all coefficients are known and do not depend on the solution. However, the maximum principles to be established are applicable more generally, and the assumption of linearity may be waived in many cases.

3.1.1 The Laplace Operator

The maximum principle for harmonic functions, i.e., functions that satisfy the Laplace equation, was known to Gauss already in 1839. A far-reaching generalization is due to Hopf [152] who proved that if a function satisfies a partial differential inequality of second order and attains a maximum in the interior of the domain, then this function is constant. Building on this result, strong and weak maximum principles have been established for PDEs of different types [120, 211, 273].

For simplicity, let us start with the maximum principle for the Laplace operator

$$\Delta = \frac{\partial^2}{\partial x_1^2} + \ldots + \frac{\partial^2}{\partial x_d^2} = \nabla^2$$

that appears in the left-hand side of the Poisson equation $-\Delta u = f$ to be solved in a bounded domain $\Omega \subset \mathbb{R}^d$, where d is the number of space dimensions.

Consider a twice continuously differentiable function $u \in C^2(\Omega) \cap C^0(\bar{\Omega})$. If u has a local maximum at an interior point $\mathbf{x} \in \Omega$, then the partial derivatives of first and second order must satisfy the following conditions at this point ([273], p. 51)

$$\frac{\partial u}{\partial x_k} = 0, \quad \frac{\partial^2 u}{\partial x_k^2} \leq 0, \qquad \forall k = 1, \ldots, d. \tag{3.1}$$

Obviously, this cannot be the case if the Laplacian of u is strictly positive

$$\Delta u > 0 \quad \text{in } \Omega.$$

It turns out that maxima are attained on the boundary even if this inequality is not strict. This result is known as the weak maximum principle (cf. [120], p. 15).

Theorem 3.1. *If the Laplacian of $u \in C^2(\Omega) \cap C^0(\bar{\Omega})$ satisfies the inequality*

$$\Delta u \geq 0 \quad \text{in } \Omega, \tag{3.2}$$

then the maximum of $u(\mathbf{x})$ over all $\mathbf{x} \in \bar{\Omega}$ is attained on the boundary Γ, that is,

$$\max_{\bar{\Omega}} u = \max_{\Gamma} u. \tag{3.3}$$

Remark 3.1. The corresponding strong maximum principle states that $\mu = \max_{\bar{\Omega}} u$ cannot be attained at any interior point $\mathbf{x} \in \Omega$ unless $u \equiv \mu$ is constant.

Different proofs of Theorem 3.1 can be found in the literature [120, 273]. The following one can be readily extended to steady convection-diffusion equations.

Proof. Following Gilbarg and Trudinger ([120], p. 45), we construct a proof by contradiction. Let $\mu = \max_{\Gamma} u$ be the maximum over $\mathbf{x} \in \Gamma$ and consider the function

$$w = \max\{0, u - \mu\}. \tag{3.4}$$

By construction, $w \geq 0$ in $\bar{\Omega}$ and $w = 0$ on Γ. The theorem requires that $w \equiv 0$ in Ω. Suppose that $w(\mathbf{x}) > 0$ at an interior point $\mathbf{x} \in \Omega$. Due to continuity, there is a neighborhood $\Omega_* \subset \Omega$ such that $w = u - \mu > 0$ in Ω_* and $w = 0$ on its boundary Γ_*.

Since the derivatives of u and w are equal in Ω_*, (3.2) and (3.4) imply that

$$w \Delta w = w \Delta u \geq 0 \quad \text{in } \Omega_*. \tag{3.5}$$

Integrating this product over Ω_*, invoking Green's formula for integration by parts, and using the assumption that w is zero on the boundary Γ_*, we obtain

$$\int_{\Omega_*} w \Delta w \, \mathrm{d}\mathbf{x} = -\int_{\Omega_*} |\nabla w|^2 \, \mathrm{d}\mathbf{x}.$$

Obviously, the right-hand side of this relation cannot be positive, while the left-hand side is nonnegative due to (3.5). This can only be the case if w is constant in $\bar{\Omega}_*$. However, a constant w cannot satisfy $w > 0$ in Ω_* and $w = 0$ on Γ_*. This contradiction proves the weak maximum principle formulated in Theorem 3.1. □

Theorem 3.2. *If the Laplacian of $u \in C^2(\Omega) \cap C^0(\bar{\Omega})$ satisfies the inequality*

$$\Delta u \leq 0 \quad \text{in } \Omega,$$

then the minimum of $u(\mathbf{x})$ over all $\mathbf{x} \in \bar{\Omega}$ is attained on the boundary Γ, that is,

$$\min_{\bar{\Omega}} u = \min_{\Gamma} u. \tag{3.6}$$

Proof. This estimate follows from Theorem 3.1 applied to $-u$. Due to the equivalence of maximum and minimum principles, it is enough to prove the former. □

Corollary 3.1. *Let $u \in C^2(\Omega) \cap C^0(\bar{\Omega})$ be the solution of the Poisson equation*

$$-\Delta u = f \quad \text{in } \Omega. \tag{3.7}$$

Then $\max_{\bar{\Omega}} u = \max_{\Gamma} u$ and/or $\min_{\bar{\Omega}} u = \min_{\Gamma} u$ if $f \leq 0$ and/or $f \geq 0$, respectively.

Definition 3.1. A function $u \in C^2(\Omega) \cap C^0(\bar{\Omega})$ is called *subharmonic* if $\Delta u \geq 0$ in Ω, *superharmonic* if $\Delta u \leq 0$ in Ω, and *harmonic* if $\Delta u = 0$ in Ω [273].

If the right-hand side of (3.7) is zero, then both estimates are applicable. Hence, the weak maximum principle for harmonic functions can be formulated as follows.

Corollary 3.2. *Let $u \in C^2(\Omega) \cap C^0(\bar{\Omega})$ be the solution of the Laplace equation*

$$\Delta u = 0 \quad \text{in } \Omega.$$

Then this harmonic function attains its maxima and minima on the boundary Γ

$$\min_{\Gamma} u \leq u(\mathbf{x}) \leq \max_{\Gamma} u, \qquad \forall \mathbf{x} \in \bar{\Omega}. \tag{3.8}$$

This double inequality gives an *a priori* estimate of $u(\mathbf{x})$ in Ω in terms of its values on Γ which are known if boundary conditions of Dirichlet type are prescribed.

Corollary 3.3. *Let u be subharmonic and v be harmonic in Ω. If $u = v$ on Γ, then*

$$u \leq v \quad \text{in } \Omega.$$

Proof. Consider the function $w = u - v$ which is subharmonic in Ω and vanishes on the boundary Γ. By the maximum principle, this function is nonpositive in Ω. □

Similarly, a superharmonic function u provides an upper bound for its harmonic counterpart v if $u = v$ on Γ. This fact explains the names given in Definition 3.1.

3.1.2 Equations of Elliptic Type

The maximum principle established for the Poisson and Laplace equations can be extended to many other problems including steady transport equations of the form

$$\mathscr{L}u = s \quad \text{in } \Omega, \tag{3.9}$$

where the divergence of convective and/or diffusive fluxes is represented by

$$\mathscr{L}u = \nabla \cdot (\mathbf{v}u - \mathscr{D}\nabla u). \tag{3.10}$$

Definition 3.2. A second-order operator $\mathscr{L}$ of the form (3.10) is called *elliptic* at a point $\mathbf{x}$ if the matrix $\mathscr{D}(\mathbf{x})$ is symmetric positive definite at this point. It is called *uniformly elliptic* in a domain Ω if it is elliptic at each point $\mathbf{x} \in \Omega$.

Ellipticity has some interesting implications. It is known from linear algebra that any symmetric positive definite matrix $\mathscr{D}$ admits the factorization ([273], p. 59)

$$\mathscr{D} = R^{-1}\Lambda R = C^T C, \tag{3.11}$$

where Λ is the diagonal matrix of positive eigenvalues, R is the orthogonal matrix ($R^{-1} = R^T$) of eigenvectors, and $C = \sqrt{\Lambda}R$ defines the linear transformation

$$C^T \tilde{\mathbf{x}} = \mathbf{x}.$$

By the chain rule, the first derivatives of u with respect to $\tilde{\mathbf{x}}$ and $\mathbf{x}$ are related by

$$\tilde{\nabla} u = C \nabla u, \tag{3.12}$$

and the sum of second derivatives with respect to the coordinates $\tilde{\mathbf{x}}$ corresponds to

$$\tilde{\Delta} u = \tilde{\nabla} \cdot (\tilde{\nabla} u) = \nabla \cdot (C^T C \nabla u) = \nabla \cdot (\mathscr{D} \nabla u). \tag{3.13}$$

Therefore, the diffusive term can be expressed in terms of the Laplacian $\tilde{\Delta}$ associated with the $\tilde{\mathbf{x}}$ coordinates under the above linear transformation. This remarkable property indicates that the theory developed for the Poisson and Laplace equations should be applicable to other models based on elliptic PDEs of second order.

If a convective flux is included, its contribution to the transport equation can be decomposed into the streamline derivative and a 'reactive' term as follows

$$\nabla \cdot (\mathbf{v} u) = \mathbf{v} \cdot \nabla u + (\nabla \cdot \mathbf{v}) u. \tag{3.14}$$

The physical meaning of the divergence $\nabla \cdot \mathbf{v}$ is the rate at which the volume of a moving fluid parcel changes as it travels through the flow field (see [4], p. 48). If $\nabla \cdot \mathbf{v} \equiv 0$, then the flow is incompressible and the solution of the transport equation (3.9) is bounded by its boundary values as in the case of the Laplace operator.

Theorem 3.3. *Let the function $u \in C^2(\Omega) \cap C^0(\bar{\Omega})$ satisfy the differential inequality*

$$\mathscr{L} u = \nabla \cdot (\mathbf{v} u - \mathscr{D} \nabla u) \leq 0 \quad \text{in } \Omega.$$

If the diffusion tensor $\mathscr{D}$ is symmetric positive definite and $\nabla \cdot \mathbf{v} \equiv 0$, then

$$\max_{\bar{\Omega}} u = \max_{\Gamma} u. \tag{3.15}$$

Proof. The proof is similar to that of Theorem 3.1. Let $\mu = \max_{\Gamma} u$ and consider

$$w = \max\{0, u - \mu\}$$

such that $w \geq 0$ in Ω and $w = 0$ on Γ. Again, the goal is to prove that $w \equiv 0$ in Ω.

Suppose that there is a subdomain $\Omega_* \subset \Omega$ such that $w = u - \mu > 0$ in Ω_* and $w = 0$ on its boundary Γ_*. Using (3.14) and the fact that $\nabla \cdot \mathbf{v} \equiv 0$, we obtain

$$\mathscr{L} w = \mathbf{v} \cdot \nabla w - \nabla \cdot (\mathscr{D} \nabla w).$$

Since the partial derivatives of u and w are equal in Ω_*, this gives the estimate

$$w \mathscr{L} w = w \mathscr{L} u \leq 0. \tag{3.16}$$

Next, we integrate the convective term by parts and invoke $\nabla \cdot \mathbf{v} \equiv 0$ again to get

$$\int_{\Omega_*} w\mathbf{v} \cdot \nabla w \, \mathrm{d}\mathbf{x} = -\int_{\Omega_*} w \nabla \cdot (\mathbf{v}w) \, \mathrm{d}\mathbf{x} = -\int_{\Omega_*} w\mathbf{v} \cdot \nabla w.$$

Since the left- and right-hand sides are the same up to the sign, this integral is zero.

Therefore, only the diffusive term might make a nonvanishing contribution to

$$\int_{\Omega_*} w\mathscr{L}w \, \mathrm{d}\mathbf{x} = \int_{\Omega_*} \nabla w \cdot (\mathscr{D}\nabla w) \, \mathrm{d}\mathbf{x}. \tag{3.17}$$

The left-hand side of this relation is nonpositive due to (3.16). Since the diffusion tensor $\mathscr{D}$ was assumed to be symmetric positive definite, $\nabla w \cdot (\mathscr{D}\nabla w) > 0$. Thus, the right-hand side of (3.17) is strictly positive, which yields a contradiction. □

In the case of an arbitrary velocity field, convective effects may create internal maxima/minima in regions where the term $(\nabla \cdot \mathbf{v})u$ is nonvanishing. Therefore, we can only prove a weaker result known as *nonnegativity* or *positivity* (preservation).

Theorem 3.4. *Let the function $u \in C^2(\Omega) \cap C^0(\bar{\Omega})$ satisfy the differential inequality*

$$\mathscr{L}u = \nabla \cdot (\mathbf{v}u - \mathscr{D}\nabla u) \geq 0 \quad \text{in } \Omega. \tag{3.18}$$

If the diffusion tensor $\mathscr{D}$ is symmetric positive definite in Ω and $u \geq 0$ on Γ, then

$$u \geq 0 \quad \text{in } \Omega.$$

Proof. We prove this weak minimum principle using a generalization of the idea presented in [289] for a simpler convection-diffusion equation in divergence form.

Suppose, contrary to the theorem, that $u \geq 0$ on Γ and $u(\mathbf{x}) < 0$ at an interior point $\mathbf{x} \in \Omega$. Then there is a subdomain $\Omega_* \subset \Omega$ such that $u < 0$ in Ω_* and $u = 0$ on its boundary Γ_*. By the divergence theorem, we have the integral identity

$$\int_{\Omega_*} \mathscr{L}u \, \mathrm{d}\mathbf{x} = -\int_{\Gamma_*} \mathbf{n} \cdot (\mathscr{D}\nabla u) \, \mathrm{d}s, \tag{3.19}$$

where $\mathbf{n}$ denotes the unit outward normal to Γ_*. The contribution of the convective flux $\mathbf{n} \cdot (\mathbf{v}u)$ to the surface integral vanishes since $u = 0$ on Γ_*. The remainder

$$\mathbf{n} \cdot (\mathscr{D}\nabla u) = \tilde{\mathbf{n}} \cdot \nabla u \tag{3.20}$$

is the rate at which u changes as we approach the boundary Γ_* moving in the direction $\tilde{\mathbf{n}} := \mathbf{n} \cdot \mathscr{D}$. Since we assume that $u < 0$ in Ω_* and $u = 0$ on Γ_*, this rate of change is strictly positive if we go in the outward direction, that is, if $\tilde{\mathbf{n}} \cdot \mathbf{n} > 0$.

Due to the assumption that the matrix $\mathscr{D}$ is symmetric positive definite, we have

$$\tilde{\mathbf{n}} \cdot \mathbf{n} = \mathbf{n} \cdot \mathscr{D} \cdot \mathbf{n} > 0.$$

It follows that the directional derivative (3.20) is positive, whereas $\mathscr{L}u \geq 0$ by assumption (3.18). Therefore, the left- and right-hand sides of equation (3.19) have nonmatching signs, which leads to a contradiction and concludes the proof. □

Remark 3.2. Reversing the sign of u in Theorems 3.3 and 3.4, one can prove the corresponding minimum principle and sign preservation for negative functions, respectively. Following a common convention, we restrict ourselves to the analysis of maximum principles and nonnegativity (positivity) constraints in what follows.

Remark 3.3. All of the above upper and lower bounds reflect the qualitative properties of differential operators and not of any particular boundary value problem. This is why no restrictions have been imposed on the choice of boundary conditions.

Elliptic partial differential equations of second order are usually endowed with boundary conditions of Dirichlet or mixed (Dirichlet-Neumann) type. In the former case, the boundary value problem for the transport equation (3.9)–(3.10) reads

$$\begin{cases} \nabla \cdot (\mathbf{v}u - \mathscr{D}\nabla u) = s, & \text{in } \Omega, \\ u = g & \text{on } \Gamma. \end{cases} \tag{3.21}$$

The weak maximum principle established in Theorems 3.3 and 3.4 yields the following *a priori* estimates of the solution $u \in C^2(\Omega) \cap C^0(\bar{\Omega})$ in terms of $g \in C^0(\Gamma)$.

Theorem 3.5. *Let the diffusion tensor $\mathscr{D}$ be symmetric positive definite and $\nabla \cdot \mathbf{v} \equiv 0$ in Ω. Then a solution of problem (3.21) satisfies the maximum principle*

$$s \leq 0 \quad \Rightarrow \quad \max_{\bar{\Omega}} u = \max_{\Gamma} g.$$

Theorem 3.6. *Let the diffusion tensor $\mathscr{D}$ be symmetric positive definite in Ω. Then a solution of problem (3.21) with arbitrary $\mathbf{v}$ satisfies the positivity constraint*

$$s \geq 0, \quad g \geq 0 \quad \Rightarrow \quad u \geq 0.$$

Here and below, inequalities are meant to hold in the whole range of function values.

Corollary 3.4. *Let the diffusion tensor $\mathscr{D}$ be symmetric positive definite in Ω. Then there is at most one solution to a linear problem of the form (3.21).*

Proof. If we suppose that there exist two different solutions u and v, then the function $w = u - v$ satisfies (3.21) with $s = 0$ and $g = 0$. Due to Theorem 3.6, as applied to w and $-w$, this implies that $0 \leq w \leq 0$ in Ω, which can only be the case if $w \equiv 0$. □

Remark 3.4. The *existence* of a unique solution is not guaranteed by this Corollary.

Corollary 3.5. *Let the linear operator $\mathscr{L}$ be given by (3.10), where $\mathscr{D}$ is symmetric positive definite in Ω. If $\mathscr{L}u \geq \mathscr{L}v$ in Ω and $u \geq v$ on Γ, then*

$$u \geq v \qquad \text{in } \bar{\Omega}.$$

Proof. Since the function $w = u - v$ satisfies the Dirichlet problem (3.21) with $s \geq 0$ and $g \geq 0$, the fact that $w \geq 0$ in Ω follows from Theorem 3.6. □

This relationship between the solutions of the same partial differential equation with different boundary data is called the *comparison principle* ([120], p. 263).

Remark 3.5. If $\nabla \cdot \mathbf{v} \equiv 0$, then $\mathscr{L}u = \mathscr{L}(u+c)$ for any constant c. Hence, if u is the unique solution of the Dirichlet problem (3.21) with $s = 0$, then $u + c$ is the unique solution of the same PDE with the Dirichlet boundary data given by $g + c$.

Theorems 3.3 and 3.4 are also applicable to (3.9) with boundary conditions of Dirichlet-Neumann type. However, the corresponding estimates are of little practical value since the solution u is not known on the whole boundary. For further information on maximum principles for elliptic problems we refer to [120, 180, 273].

3.1.3 Equations of Hyperbolic Type

Convection-reaction models are based on first-order PDEs, and maximum principles are obtained in an entirely different way. Convective transport of information by a prescribed velocity field $\mathbf{v}$ takes place along parametric curves that represent the characteristics of the hyperbolic equation. In the Lagrangian reference frame which corresponds to the viewpoint of an observer moving with the flow velocity, the problem reduces to a set of ODEs to be integrated along the characteristics subject to the prescribed initial and/or boundary conditions. This knowledge makes it possible to predict how the solution evolves as the fluid moves through the flow field.

In experimental fluid dynamics, it is common practice to visualize the flow motion by releasing and tracking markers, such as small particles or colored dye. The flow lines traced by these markers are the physical prototypes of characteristics.

Definition 3.3. A *streamline* is a curve tangent to the velocity vector at each point.

As long as a fluid particle cannot have two different velocities at the same point, streamlines of an instantaneously defined velocity field $\mathbf{v}(\mathbf{x})$ cannot intersect. If the function $\mathbf{v}$ is Lipschitz-continuous, i.e., there is a constant C such that

$$|\mathbf{v}(\mathbf{x}_1) - \mathbf{v}(\mathbf{x}_2)| \leq C|\mathbf{x}_1 - \mathbf{x}_2|, \qquad \forall \mathbf{x}_1, \mathbf{x}_2 \in \Omega,$$

then there is exactly one streamline through a given point [176]. In steady flow, markers released at successive time instants are exposed to the same flow field and follow the same path. Thus, streamlines coincide with the trajectories of tracers.

Definition 3.4. A *pathline* is the trajectory followed by an individual fluid particle.

If the velocity field is known, a pathline is described by a system of ordinary differential equations for the Cartesian coordinates of a moving marker. Hence, it is possible to track the markers and monitor their properties mathematically rather than experimentally. This is the idea behind the method of characteristics that we will use to analyze the properties of steady and unsteady convective transport models.

3.1.3.1 Steady Convective Transport

Steady convection-reaction processes can be described by the hyperbolic equation

$$\nabla \cdot (\mathbf{v}u) = s \quad \text{in } \Omega \tag{3.22}$$

which can also be written in the generic form (3.9) with $\mathscr{L}u = \nabla \cdot (\mathbf{v}u)$ and $\mathscr{D} \equiv 0$.

In the absence of diffusion, boundary conditions can only be prescribed on the inflow boundary, as required by the one-way nature of convective transport. Let

$$u = g \quad \text{on} \quad \Gamma_- = \{\mathbf{x} \in \Gamma \,|\, \mathbf{v} \cdot \mathbf{n} < 0\}. \tag{3.23}$$

No boundary conditions are prescribed on the complementary part $\Gamma_0 \cup \Gamma_+$, where

$$\Gamma_0 = \{\mathbf{x} \in \Gamma \,|\, \mathbf{v} \cdot \mathbf{n} = 0\}, \qquad \Gamma_+ = \{\mathbf{x} \in \Gamma \,|\, \mathbf{v} \cdot \mathbf{n} > 0\}.$$

Although the velocity field $\mathbf{v}(\mathbf{x})$ is assumed to be steady, the fluid is in motion. If a marker is launched at a point $\mathbf{x}_0 \in \Gamma_-$ and time instant t_0, then it will move along the streamline/pathline through $\mathbf{x}_0$ until it leaves the domain Ω at the outlet Γ_+. Let

$$\hat{\mathbf{x}}(t) = \mathbf{x}(t, \mathbf{x}_0, t_0) \tag{3.24}$$

denote the instantaneous position $\mathbf{x}$ of the marker at time $t \geq t_0$. The 'hat' notation will also refer to functions of the position vector $\hat{\mathbf{x}}(t)$, such as the velocity

$$\hat{\mathbf{v}}(t) = \mathbf{v}(\hat{\mathbf{x}}(t)).$$

Since the marker is moving with the prescribed velocity $\hat{\mathbf{v}}(t)$, its pathline is given by the set of points $\hat{\mathbf{x}}(t) \in \Omega$ which satisfy the following Cauchy problem

$$\frac{\mathrm{d}\hat{\mathbf{x}}(t)}{\mathrm{d}t} = \hat{\mathbf{v}}(t), \qquad \hat{\mathbf{x}}(t_0) = \mathbf{x}_0. \tag{3.25}$$

The so-defined parametric curves $\hat{\mathbf{x}}(t)$, as depicted in Fig. 3.1, are the streamlines of the velocity field and the characteristics of the linear hyperbolic equation (3.22).

By the chain rule, the total derivative of the function $\hat{u}(t) = u(\hat{\mathbf{x}}(t))$ is given by

$$\frac{\mathrm{d}\hat{u}}{\mathrm{d}t} = \sum_{k=1}^{d} \frac{\partial u}{\partial x_k} \frac{\mathrm{d}\hat{x}_k}{\mathrm{d}t} = \hat{\mathbf{v}} \cdot \hat{\nabla} u. \tag{3.26}$$

The notation $\hat{\nabla}$ stands for ∇ applied at $\hat{\mathbf{x}}(t)$. Invoking (3.22) and (3.14), we obtain

$$\hat{\mathbf{v}} \cdot \hat{\nabla} u = \hat{s} - (\hat{\nabla} \cdot \mathbf{v})\hat{u}.$$

Therefore, the evolution of $\hat{u}(t)$ along the characteristic through $\mathbf{x}_0$ is governed by

$$\frac{\mathrm{d}\hat{u}}{\mathrm{d}t} + \hat{r}\hat{u} = \hat{s}, \qquad \hat{u}(t_0) = g_0, \tag{3.27}$$

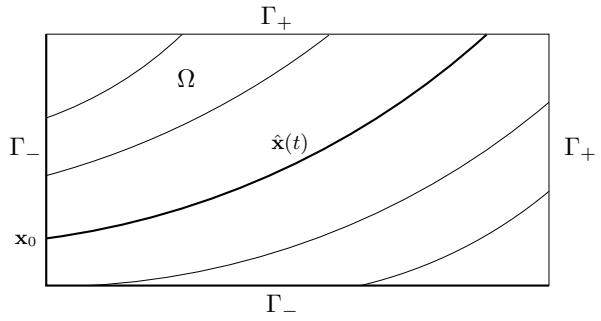


Fig. 3.1 Characteristics of the steady two-dimensional convection equation.

where $\hat{r} = \hat{\nabla} \cdot \mathbf{v}$ represents the rate of volumetric compressibility, and the value of

$$g_0 = g(\mathbf{x}_0)$$

is available from the Dirichlet boundary conditions (3.23) prescribed at the inlet Γ_-.

The analytical solution of the Cauchy problem (3.27) is as follows [86, 88]

$$\hat{u}(t) = \frac{1}{\gamma(t)} \left[g_0 + \int_0^t \gamma(\tau)\hat{s}(\tau)\,\mathrm{d}\tau \right], \tag{3.28}$$

where the contribution of $\hat{r}\hat{u}$ is taken into account by the auxiliary function

$$\gamma(t) = \exp\left(\int_0^t \hat{r}(\tau)\,\mathrm{d}\tau \right).$$

Remark 3.6. In the case $\hat{r} \equiv 0$ and $\hat{s} \equiv 0$, solution (3.28) reduces to the identity

$$\hat{u}(t) = g_0, \qquad \forall t \geq t_0. \tag{3.29}$$

This means that the solution of (3.22) is constant along the streamlines/characteristics.

Remark 3.7. By using the chain rule, we tacitly assumed that u is differentiable. This assumption can be relaxed. If the prescribed boundary condition (3.23) has a jump at some point $\mathbf{x}_0 \in \Gamma_-$, then the weak solution of the linear hyperbolic equation (3.22) will remain discontinuous along the entire characteristic $\hat{\mathbf{x}}(t)$ through $\mathbf{x}_0 = \hat{\mathbf{x}}(t_0)$.

To obtain the solution $u(\mathbf{x})$ of problem (3.22)–(3.23) at a point $\mathbf{x} \in \Omega$, one needs to solve (3.27) along the characteristic that passes through $\mathbf{x}$ and satisfies (3.25). Moreover, the following upper/lower bounds can be readily inferred from (3.28).

Theorem 3.7. *A solution of problem (3.22)–(3.23) with* $\nabla \cdot \mathbf{v} \equiv 0$ *satisfies*

$$s \leq 0 \quad \Rightarrow \quad \max_{\Omega} u = \max_{\Gamma_-} g.$$

In elliptic problems, maxima and minima could be attained anywhere on the boundary. The above theorem takes advantage of the fact that convection is a one-way process, which restricts the possible location of maxima/minima to the inlet Γ_-.

If the velocity field is not divergence-free, local extrema can emerge in the interior of the domain Ω but equation (3.28) still implies positivity preservation.

Theorem 3.8. *A solution of problem (3.22)–(3.23) with arbitrary* $\mathbf{v}$ *satisfies*

$$s \geq 0, \quad g \geq 0 \quad \Rightarrow \quad u \geq 0.$$

Corollary 3.6. *If equation (3.22) is linear, then there is at most one solution.*

Corollary 3.7. *If* $w = u - v$ *satisfies (3.22)–(3.23) with* $s \geq 0$ *and* $g \geq 0$*, then*

$$u \geq v \qquad \text{in } \bar{\Omega}.$$

Linearity is essential for the proof of uniqueness and for the comparison principle.

3.1.3.2 Unsteady Convective Transport

In unsteady problems, the solution $u(\mathbf{x},t)$ is defined in a bounded space-time domain $\Omega \times (0,T)$, and information is 'convected' forward in time with unit velocity. Adding a time derivative to equation (3.22), one obtains its unsteady counterpart

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = s \quad \text{in } \Omega \times (0,T). \tag{3.30}$$

It is also hyperbolic, so boundary conditions are required only at the inlet Γ_-. Let

$$u(\mathbf{x},t) = g(\mathbf{x},t), \quad \forall \mathbf{x} \in \Gamma_-. \tag{3.31}$$

Since the time level $t = 0$ represents another 'inflow boundary' of the space-time domain $\Omega \times (0,T)$, it is also necessary to prescribe a suitable initial condition

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega. \tag{3.32}$$

If the velocity field $\mathbf{v}(\mathbf{x},t)$ is time-dependent, then the trajectory of a marker depends not only on the position $\mathbf{x}_0$ but also on the time t_0 at which it is launched. The characteristics of equation (3.30) are defined as pathlines given by (3.25) with $\hat{\mathbf{v}}(t) = \mathbf{v}(\hat{\mathbf{x}}(t),t)$. The origin of each characteristic $\hat{\mathbf{x}}(t)$ is located either at the inlet ($\mathbf{x}_0 \in \Gamma_-$ for $t_0 > 0$) or at the initial time level ($\mathbf{x}_0 \in \Omega$ for $t_0 = 0$), see Fig. 3.2.

Definition 3.5. The *substantial derivative* is the time rate of change along a pathline

$$\frac{Du}{Dt} := \frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u. \tag{3.33}$$

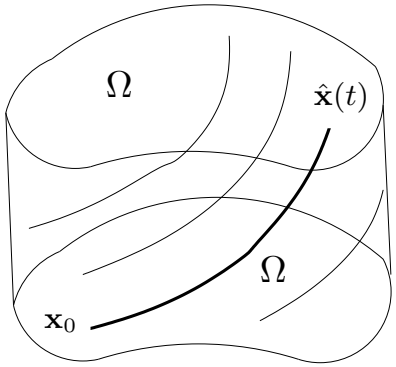


Fig. 3.2 Characteristics of the unsteady two-dimensional convection equation.

Differentiating the function $\hat{u}(t) = u(\hat{\mathbf{x}}(t), t)$ along the characteristics, we obtain

$$\frac{\mathrm{d}\hat{u}(t)}{\mathrm{d}t} = \frac{\partial u}{\partial t} + \sum_{k=1}^{d} \frac{\partial u}{\partial x_k} \frac{\mathrm{d}\hat{x}_k}{\mathrm{d}t} = \frac{D\hat{u}}{Dt}, \tag{3.34}$$

where $\frac{D\hat{u}}{Dt}$ is evaluated at point $\hat{\mathbf{x}}(t)$ and time t. Due to equation (3.30), it satisfies

$$\frac{D\hat{u}}{Dt} = \hat{s} - (\hat{\nabla} \cdot \mathbf{v})\hat{u}.$$

Substitution into the right-hand side of (3.34) leads to a Cauchy problem of the form (3.27), where the value of g_0 is known from the initial/boundary data

$$g_0 = \begin{cases} g(\mathbf{x}_0, t_0), & \text{if } \mathbf{x}_0 \in \Gamma_-, \ t_0 > 0, \\ u_0(\mathbf{x}_0), & \text{if } \mathbf{x}_0 \in \Omega, \ t_0 = 0. \end{cases} \tag{3.35}$$

Again, the instantaneous value of $\hat{u}(t)$ is given by (3.28), and the solution may be discontinuous along the characteristic $\hat{\mathbf{x}}(t)$ if there is a jump at point $\mathbf{x}_0$ and time t_0.

Theorem 3.9. *A solution of problem (3.30)–(3.32) with* $\nabla \cdot \mathbf{v} \equiv 0$ *satisfies*

$$s \le 0 \quad \Rightarrow \quad \max_{\bar{\Omega}} u = \max_{\Gamma_- \times [0,T]} g \quad \text{or} \quad \max_{\bar{\Omega}} u = \max_{\bar{\Omega}} u_0.$$

In contrast to steady convection, the possible locus of maxima and minima includes not only $\Gamma_- \times [0,T]$ but also the 'time inlet' $\Omega \times \{0\}$ of the space-time domain.

Theorem 3.10. *A solution of problem (3.30)–(3.31) with arbitrary* $\mathbf{v}$ *satisfies*

$$s \ge 0, \quad g \ge 0, \quad u_0 \ge 0 \quad \Rightarrow \quad u \ge 0.$$

Corollary 3.8. *If equation (3.30) is linear, then there is at most one solution.*

Corollary 3.9. *Let $w = u - v$ be a solution of (3.30)–(3.32) with $s \geq 0$ and $g \geq 0$. If the corresponding initial data satisfy $u_0 \geq v_0$, then*

$$u \geq v \qquad \text{in } \bar{\Omega} \times [0,T].$$

Linearity is essential for the proof of uniqueness and for the comparison principle.

3.1.4 Equations of Parabolic Type

Unsteady transport equations in which both convection and diffusion are taken into account are of parabolic type. The most general formulation of such a model is

$$\frac{\partial u}{\partial t} + \mathscr{L}u = s \quad \text{in } \Omega \times (0,T), \tag{3.36}$$

where the linear operator $\mathscr{L}$ is the same as that for steady convection-diffusion

$$\mathscr{L}u = \nabla \cdot (\mathbf{v}u - \mathscr{D}\nabla u). \tag{3.37}$$

This model shares some features of its elliptic and hyperbolic counterparts which are obtained by neglecting the time derivative and/or diffusive terms, respectively.

Definition 3.6. If $\mathscr{L}$ is an elliptic operator of second order, then $\frac{\partial}{\partial t} + \mathscr{L}$ is *parabolic*.

Loosely speaking, parabolic equations are elliptic in space and hyperbolic in time. On the one hand, information may travel in arbitrary space directions, so the space variables are *two-way coordinates* [268]. On the other hand, the time is always a *one-way coordinate* since changes that occur at a given instant can only influence the solution at the same or later time. Current happenings depend on the evolution history and affect future events but have no influence on what happened in the past.

Due to the presence of a time derivative and second-order space derivatives, both initial data and boundary conditions are to be prescribed. In contrast to unsteady hyperbolic problems, the distinction between inlets and outlets is irrelevant. A boundary condition is required at each point $\mathbf{x} \in \Gamma$, no matter if $\mathbf{v} \cdot \mathbf{n}$ is positive or negative.

Consider an initial boundary value problem that consists of the parabolic PDE

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u - \mathscr{D}\nabla u) = s \quad \text{in } \Omega \times (0,T) \tag{3.38}$$

supplemented by an initial condition and boundary conditions of Dirichlet type

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega, \tag{3.39}$$

$$u(\mathbf{x},t) = g(\mathbf{x},t), \quad \forall \mathbf{x} \in \Gamma,\ \forall t \in (0,T]. \tag{3.40}$$

Our experience with transport equations of elliptic and hyperbolic type enables us to prove the following maximum principle for the above parabolic problem.

Theorem 3.11. *Let the diffusion tensor $\mathscr{D}$ be symmetric positive definite in Ω. Then a solution of problem (3.38)–(3.40) with $\nabla \cdot \mathbf{v} \equiv 0$ satisfies*

$$s \le 0 \quad \Rightarrow \quad \max_{\bar{\Omega}} u = \max_{\Gamma \times [0,T]} g \quad \text{or} \quad \max_{\bar{\Omega}} u = \max_{\Omega} u_0. \tag{3.41}$$

Proof. Obviously, any values prescribed on Γ and at $t = 0$ satisfy the requirement stated in (3.41). To prove the maximum principle (3.41), it is sufficient to show that no new maxima can be generated in the interior of Ω at any time $t \in (0,T]$.

Consider $g_0 = \max_{\Gamma_- \times [0,T]} g$ or $g_0 = \max_{\Omega} u_0$, whichever is greater. By definition, $g_0 = u(\mathbf{x}_0, t_0)$, where $\mathbf{x}_0 \in \Omega$ and $t_0 = 0$ or $\mathbf{x}_0 \in \Gamma_-$ and $t_0 > 0$. The question is whether the so-defined initial peak will grow or decay as it convected through the flow field subject to diffusion and reaction. Along the corresponding pathline $\hat{\mathbf{x}}(t)$, equation (3.38) can be written in terms of the substantial derivative (3.34) thus:

$$\frac{\mathrm{d}\hat{u}}{\mathrm{d}t} = \hat{\nabla} \cdot (\mathscr{D} \hat{\nabla} u) + \hat{s} - (\hat{\nabla} \cdot \mathbf{v}) \hat{u},$$

where the last term is zero since $\nabla \cdot \mathbf{v} \equiv 0$. Hence, the evolution of $\hat{u}(t)$ along the pathline through $\mathbf{x}_0$ is governed by the following initial value problem

$$\frac{\mathrm{d}\hat{u}}{\mathrm{d}t} = \hat{\nabla} \cdot (\mathscr{D} \hat{\nabla} u) + \hat{s}, \qquad \hat{u}(t_0) = g_0.$$

In incompressible flow problems, convection alone cannot change the amplitude of the peak g_0 but diffusion and reaction surely can. Since the diffusion tensor $\mathscr{D}$ is symmetric positive definite, it admits a factorization of the form (3.11) which leads to (3.13). Under the linear transformation (3.12), we have $\hat{\nabla} \cdot (\mathscr{D} \hat{\nabla} u) = \tilde{\Delta} u$ and

$$\frac{\mathrm{d}\hat{u}}{\mathrm{d}t} = \tilde{\Delta} u + \hat{s}. \tag{3.42}$$

If $\hat{u}(t)$ is a local maximum, then conditions (3.1) hold in the transformed coordinate system, so the right-hand side of (3.42) is nonpositive for $s \equiv 0$. Therefore, the initial peak $\hat{u}(t_0) = g_0$ cannot increase along the pathline $\hat{\mathbf{x}}(t)$, which proves that the maximum $\mu = \max_{\bar{\Omega}} u$ must be attained either in Ω at $t = 0$ or on $\Gamma \times [0,T]$. □

Theorem 3.12. *Let the diffusion tensor $\mathscr{D}$ be symmetric positive definite in Ω. Then a solution of problem (3.38)–(3.40) with arbitrary $\mathbf{v}$ satisfies*

$$s \ge 0, \quad g \ge 0, \quad u_0 \ge 0 \quad \Rightarrow \quad u \ge 0. \tag{3.43}$$

Proof. If the velocity field $\mathbf{v}(\mathbf{x},t)$ is not divergence-free, then the nonvanishing 'reactive' term $(\hat{\nabla} \cdot \mathbf{v})\hat{u}$ must be added to the right-hand side of equation (3.42). As a consequence, a positive minimum $\hat{u}(t_0) = g_0$ may decrease along $\hat{\mathbf{x}}(t)$. However, as soon as its value reaches the zero level, the term $(\hat{\nabla} \cdot \mathbf{v})\hat{u}$ vanishes, so $\hat{u}(t)$ cannot decrease any further for the reasons explained in the proof of Theorem 3.11. □

Corollary 3.10. *If equation (3.38) is linear, then there is at most one solution.*

Corollary 3.11. *Let $w = u - v$ be a solution of (3.38)–(3.40) with $s \geq 0$ and $g \geq 0$. If the corresponding initial data satisfy $u_0 \geq v_0$ in Ω, then*

$$u \geq v \qquad \text{in } \bar{\Omega} \times [0,T].$$

Linearity is essential for the proof of uniqueness and for the comparison principle.

3.1.5 Singularly Perturbed Problems

In hyperbolic problems, boundary conditions are required only at the inlet. If we add a small diffusive term, this hardly makes any difference as far as the partial differential equation itself is concerned. However, extra boundary conditions are required for the so-defined problem which is nominally elliptic or parabolic. Even if the diffusion coefficient is very small, the solution of the perturbed problem may turn out to be a poor approximation to that of the original one and vice versa. Moreover, the right formulation of a maximum/minimum principle changes with the PDE type.

In perturbation theory, an approximation to a *regularly perturbed* problem can be obtained by simply setting the small parameter to zero. Problems that cannot be approximated properly in this way are referred to as *singularly perturbed*.

In fluid dynamics, the small parameter is usually the diffusion coefficient. A classical example is the following singularly perturbed elliptic problem [176, 288]

$$\begin{cases} \nabla \cdot (\mathbf{v}u - \varepsilon \nabla u) = s & \text{in } \Omega, \\ u = g & \text{on } \Gamma, \end{cases} \tag{3.44}$$

where $0 < \varepsilon \ll 1$ is very small. The solution u of this convection-dominated problem is smooth (differentiable) but its derivatives may become very large as $\varepsilon \to 0$.

Definition 3.7. A zone in which u or its derivatives change abruptly is called a *layer*.

In steady transport problems, the location of layers is fixed and sometimes known *a priori*. Their thickness decreases as the ratio $|\mathbf{v}|/\varepsilon$ increases. To identify a possible cause of layers, consider the hyperbolic counterpart of problem (3.44)

$$\begin{cases} \nabla \cdot (\mathbf{v}u) = s & \text{in } \Omega, \\ u = g & \text{on } \Gamma_-. \end{cases} \tag{3.45}$$

Note that the boundary data are prescribed at the inlet Γ_- since the solution of a hyperbolic equation cannot be forced to satisfy any boundary condition elsewhere.

If the solution of problem (3.45) has a jump along a characteristic through a point $\mathbf{x}_0 \in \Gamma_-$, then diffusion will smear it over a zone of finite thickness. As a consequence, the solution of (3.44) will exhibit an *internal layer* of width $\mathcal{O}(\sqrt{\varepsilon})$ around the characteristic [176]. Discontinuous diffusion coefficients and singular sources may also give rise to internal layers. Moreover, a *boundary layer* of width

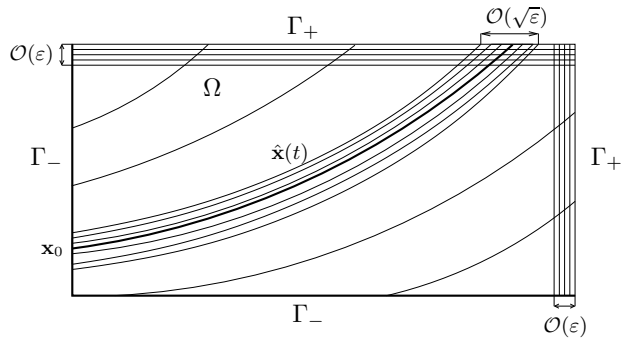


Fig. 3.3 Interior and boundary layers for a singularly perturbed transport equation.

$\mathcal{O}(\varepsilon)$ will form if the boundary conditions prescribed on $\Gamma\backslash\Gamma_-$ are incompatible with the boundary values of the solution to the reduced problem (3.45), see Fig. 3.3.

In the unsteady case, the solution of the singularly perturbed parabolic problem

$$\begin{cases} \frac{\partial u}{\partial t} + \nabla\cdot(\mathbf{v}u - \varepsilon\nabla u) = s & \text{in } \Omega\times(0,T), \\ u = g & \text{on } \Gamma\times(0,T), \\ u = u_0 & \text{in } \Omega \text{ at } t = 0 \end{cases} \tag{3.46}$$

may also have internal and boundary layers. The former may be caused by discontinuities in initial data, inflow boundary conditions, or coefficients. As time evolves, internal layers are convected downstream and smeared by diffusion. Their thickness and the rate of smearing depend on the value of the perturbation parameter ε.

3.2 Matrix Analysis for Steady Problems

If the solution of a given boundary value problem satisfies a maximum principle, then a properly designed approximation should behave in the same way. A numerical scheme that does not generate spurious global extrema in the interior of the computational domain is said to satisfy a *discrete maximum principle* (DMP). As in the continuous case, the precise formulation of this criterion is problem-dependent. In particular, the zero row sum property (second rule from Section 1.6.3) has the same implications as the constraint $\nabla\cdot\mathbf{v}\equiv 0$ in continuous maximum principles.

In the context of finite difference approximations to linear elliptic problems, sufficient conditions of DMP were formulated and proven by Varga [340] as early as in 1966. These conditions are related to the concept of *monotone operators* and, in particular, *M-matrices* which play an important role in numerical linear algebra [339, 354]. A general approach to DMP analysis for finite difference operators was developed by Ciarlet [63]. Its extension to finite elements in [64] features a proof of *uniform convergence*, as well as simple geometric conditions that ensure the validity

of DMP for a piecewise-linear Galerkin discretization of the (linear) model problem

$$\begin{cases} -\Delta u + ru = s, & \text{in } \Omega, \\ u = g & \text{on } \Gamma, \end{cases} \tag{3.47}$$

on a triangular mesh under the assumption that $r \geq 0$ in Ω. The results obtained in [63, 64, 339] have illustrated the significance of DMPs for the analysis and design of numerical approximations. Various extensions and generalizations were published during the past three decades, see [61, 100, 151, 179, 180, 341, 311] and references therein. The frequently cited monograph by Ikeda [167] is devoted entirely to DMP for finite element models of convection-diffusion phenomena.

Some low-order approximations of transport equations are known to satisfy a DMP unconditionally or under rather mild restrictions on the angles or aspect ratios of mesh cells. However, most *a priori* proofs are based on a set of sufficient conditions which become overly restrictive in the case of higher-order discretizations, singularly perturbed convection-diffusion equations, and anisotropic diffusion problems. A possible remedy to this problem is proposed in the next chapter.

In this section, we review the algebraic constraints that ensure DMP and/or positivity preservation for steady transport problems of elliptic and hyperbolic type. A brief summary of the corresponding geometric conditions will also be presented.

3.2.1 The Discrete Problem

Consider the steady transport-reaction equation (3.9) discretized by a finite difference, finite volume, or finite element scheme. Let the approximate solution u_h, where the subscript h refers to the mesh size, be determined by a finite number N of degrees of freedom $u_1, \ldots, u_{\bar{N}}$ that represent pointwise nodal values, control volume averages, or coefficients of piecewise-polynomial basis functions, respectively. Hence, all information about the solution u_h can be packed into a vector $u \in \mathbb{R}^{\bar{N}}$.

Furthermore, the differential operator $\mathscr{L}$ acting on functions defined at infinitely many locations is replaced by a discrete operator $\mathscr{L}_h$ acting on vectors of length $\bar{N}$

$$\mathscr{L}_h : \mathbb{R}^{\bar{N}} \to \mathbb{R}^{\bar{N}}.$$

Regardless of the underlying approximation technique, we define this mapping as

$$\mathscr{L}_h u = Au,$$

where $A = \{a_{ij}\}$ is a sparse $\bar{N} \times \bar{N}$ matrix and $u = \{u_i\}$ is the vector of nodal values.

The sparsity pattern of A depends on the mesh, on the type of discretization, and on the numbering of nodes. Since some nodal values are known from the Dirichlet boundary conditions, the size of the algebraic system reduces accordingly.

Let the first N nodes be associated with the unknown degrees of freedom, and the rest with the Dirichlet boundary values. This numbering convention implies that the

discrete operator A and the vector of nodal values u can be partitioned as follows

$$A = \begin{bmatrix} A_{\Omega\Omega} & A_{\Omega\Gamma} \\ A_{\Gamma\Omega} & A_{\Gamma\Gamma} \end{bmatrix}, \quad u = \begin{bmatrix} u_\Omega \\ u_\Gamma \end{bmatrix}. \tag{3.48}$$

The subscripts Ω and Γ refer to row/column numbers from $\mathcal{N}_\Omega = \{1,\ldots,N\}$ and $\mathcal{N}_\Gamma = \{N+1,\ldots,\bar{N}\}$, respectively. Thus, $u_\Omega = \{u_1,\ldots,u_N\}$ is the vector of unknowns, whereas $u_\Gamma = \{u_{N+1},\ldots,u_{\bar{N}}\}$ is given by the prescribed boundary data

$$u_\Gamma = g. \tag{3.49}$$

In this notation, the system of algebraic equations for the components of u_Ω reads

$$A_{\Omega\Omega}u_\Omega = b_\Omega - A_{\Omega\Gamma}g, \tag{3.50}$$

where b_Ω is the contribution of sources and Neumann boundary conditions, if any.

In a practical implementation, it is convenient to incorporate the Dirichlet boundary conditions into the $\bar{N}\times\bar{N}$ matrix A and solve the extended linear system [63]

$$\bar{A}u = b, \tag{3.51}$$

where the matrix $\bar{A}$ and right-hand side b are defined so as to enforce (3.49)

$$\bar{A} = \begin{bmatrix} A_{\Omega\Omega} & A_{\Omega\Gamma} \\ 0 & I \end{bmatrix}, \quad b = \begin{bmatrix} b_\Omega \\ g \end{bmatrix}. \tag{3.52}$$

In other words, the $\bar{N}\times\bar{N}$ matrix $\bar{A}$ is constructed from A by setting $A_{\Gamma\Omega} := 0$ and $A_{\Gamma\Gamma} := I$, where I denotes the identity matrix with $\bar{N}-N$ rows and columns.

If the solution of the continuous problem satisfies Theorem 3.5 or 3.7, it is natural to require that the maxima of u_Ω be bounded by those of $u_\Gamma = g$. Likewise, all nodal values should be nonnegative if Theorem 3.6 or 3.8 is applicable. To verify the validity of DMP, one needs to analyze the properties of the discrete operator $\bar{A}$.

3.2.2 M-Matrices and Monotonicity

A key ingredient of the mathematical theory behind the discrete maximum principles and positivity preservation is the following monotonicity concept [63, 339].

Definition 3.8. A regular matrix $A = \{a_{ij}\}$ is called *monotone* if $A^{-1} \geq 0$.

This kind of monotonicity is equivalent to the requirement that, for any vector u,

$$Au \geq 0 \quad \Rightarrow \quad u \geq 0.$$

Of course, it is impractical to compute the inverse of A and check the sign of its entries. Instead, we will deal with a special class of matrices which are known to be monotone under certain constraints on the sign and magnitude of their coefficients.

Definition 3.9. A regular matrix $A = \{a_{ij}\}$ is called an *M-matrix* if $A^{-1} \geq 0$ and

$$a_{ij} \leq 0, \qquad \forall j \neq i.$$

In other words, an M-matrix is a monotone matrix with nonpositive off-diagonal entries. These properties ensure positivity and convergence of iterative solvers.

Definition 3.10. A matrix $A = \{a_{ij}\}$ is called *diagonally dominant* (by rows) if

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, \qquad \forall i. \tag{3.53}$$

Such a matrix is called *strictly diagonally dominant* if all inequalities are strict

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \qquad \forall i. \tag{3.54}$$

Definition 3.11. A matrix $A = \{a_{ij}\}$ of size $N \times N$ is called *irreducible* if there is no $N \times N$ permutation matrix P such that the following transformation is possible

$$PAP^T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix},$$

where the size of A_{11} is $M \times M$, the size of A_{22} is $(N-M) \times (N-M)$, and $1 \leq M < N$.

It turns out that a matrix A is irreducible if and only if its directed graph is strongly connected ([339], p. 20) or, equivalently, if and only if for any i and $j \neq i$ there is a sequence of distinct indices $i = n_0, n_1, \ldots, n_l = j$ such that [132]

$$a_{n_{k-1} n_k} \neq 0, \qquad 1 \leq k \leq l.$$

Remark 3.8. In the context of linear systems, irreducibility ensures that it is impossible to extract a subsystem that can be solved independently. Matrices that result from discretization of partial differential equations are irreducible in most cases.

Definition 3.12. A matrix $A = \{a_{ij}\}$ is *irreducibly diagonally dominant* if it is irreducible and diagonally dominant, with strict dominance for at least one row i.

The following theorem yields a set of sufficient conditions which are commonly employed in DMP analysis based on the M-matrix property of discrete operators.

Theorem 3.13. *If $A = \{a_{ij}\}$ is a strictly or irreducibly diagonally dominant $N \times N$ matrix with $a_{ii} > 0$, $\forall i = 1, \ldots, N$ and $a_{ij} \leq 0$, $\forall j \neq i$, then $A^{-1} \geq 0$.*

Proof. A common approach to the proof of this theorem is based on the splitting

$$A = D - C,$$

where the diagonal part $D = \operatorname{diag}(A) > 0$ is nonsingular and $C \geq 0$. The diagonal dominance makes it possible to prove that the spectral radius ρ of $B = D^{-1}C \geq 0$ satisfies $\rho(B) < 1$. This condition holds if and only if the series

$$(I - B)^{-1} = I + B + B^2 + B^3 + \ldots$$

converges, see [131, 339] for technical details. Hence, $A^{-1} = (I - B)^{-1}D^{-1} \geq 0$. □

If all diagonal entries of A are strictly positive and there are no positive off-diagonal ones, then diagonal dominance (3.53) requires that all row sums be nonnegative. The following definition summarizes the corresponding sign conditions.

Definition 3.13. A matrix $A = \{a_{ij}\}$ is said to be of *nonnegative type* [64, 69] if

$$a_{ii} > 0, \qquad \forall i, \tag{3.55}$$

$$a_{ij} \leq 0, \qquad \forall j \neq i, \tag{3.56}$$

$$\sum_j a_{ij} \geq 0, \quad \forall i. \tag{3.57}$$

Corollary 3.12. *By Theorem 3.13, a nonnegative-type matrix A is an M-matrix if inequality (3.57) is strict or A is irreducible and (3.57) is strict for at least one row.*

Note that conditions (3.55)–(3.56) impose the same constraints as the third basic rule from Section 1.6.3. The second basic rule is satisfied if (3.57) holds as equality.

Under the assumptions of Corollary 3.12, the nonnegativity conditions are sufficient (but not *necessary*) for the matrix A to be monotone. Some other useful criteria related to M-matrices and monotonicity can be found in [46, 132, 153, 311, 346].

3.2.3 Discrete Maximum Principles

Given a discretization of the form (3.51), the monotonicity of the matrix $\bar{A}$ makes it possible to prove discrete counterparts of all maximum, minimum, and comparison principles established in Section 3.1. The uniqueness of the solution vector u follows from the regularity of $\bar{A}$. The usual approach to the proof of monotonicity is based on Theorem 3.13 and Corollary 3.12 since the nonnegativity conditions (3.55)–(3.57) are easy to verify for an arbitrary space discretization of the transport equation.

To prove that the solution of problem (3.51) attains its maximum on the set $\mathcal{N}_\Gamma$ of Dirichlet boundary nodes, we need a discrete counterpart of the incompressibility constraint $\nabla \cdot \mathbf{v} \equiv 0$. At the continuous level, it implies that $\mathscr{L}u = \mathscr{L}(u + c)$ for an arbitrary constant c. According to the second basic rule from Section 1.6.3, the discrete operator A should have zero row sums to inherit this property. Thus, the *global* discrete maximum principle for nodal values can be formulated as follows.

Theorem 3.14. *If the matrix $\bar{A}$ is given by (3.52), $A_{\Omega\Omega}$ is monotone, $A_{\Omega\Gamma} \le 0$, and*

$$\sum_{j=1}^{\bar{N}} a_{ij} = 0, \qquad \forall i \in N_\Omega, \tag{3.58}$$

then the solution of (3.51) satisfies the global discrete maximum principle

$$b_\Omega \le 0 \quad \Rightarrow \quad \max_i u_i = \max_j g_j. \tag{3.59}$$

Proof. Let $b_\Omega \le 0$ and consider $w = u - \mu$, where $\mu = \max_j g_j$ is the largest Dirichlet boundary value. Due to (3.51) and the zero-sum property (3.58), we have

$$\sum_{j=1}^{\bar{N}} a_{ij} w_j = \sum_{j=1}^{\bar{N}} a_{ij} u_j - \mu \sum_{j=1}^{\bar{N}} a_{ij} = \sum_{j=1}^{\bar{N}} a_{ij} u_j, \qquad \forall i \in N_\Omega. \tag{3.60}$$

It follows that $A_{\Omega\Omega} w_\Omega + A_{\Omega\Gamma} w_\Gamma = b_\Omega \le 0$, where $A_{\Omega\Gamma} w_\Gamma \ge 0$. Since $A_{\Omega\Omega}$ is monotone, $w_\Omega = A_{\Omega\Omega}^{-1}[b_\Omega - A_{\Omega\Gamma} w_\Gamma] \le 0$ so that $u_i \le \max_j g_j$ for all $i \in N_\Omega$. □

Since the matrix $\bar{A}$ is sparse, only nearest neighbors make a nonzero contribution to the right-hand side of the algebraic equation for an interior node $i \in \mathcal{N}_\Omega$. The following theorem states that u_i is bounded by the solution values at neighbor nodes.

Theorem 3.15. *If the matrix $\bar{A}$ is of nonnegative type and condition (3.58) holds, then the solution of (3.51) satisfies the local discrete maximum principle [20]*

$$b_i \le 0 \quad \Rightarrow \quad u_i \le \max_{j \in \mathcal{N}_i} u_j, \qquad \forall i \in \mathcal{N}_\Omega, \tag{3.61}$$

where $\mathcal{N}_i := \{ j \ne i \mid a_{ij} \ne 0 \}$ is the set of neighbors that form the stencil of node i.

Proof. Let $i \in N_\Omega$ be any interior node. The equation for the nodal value u_i reads

$$a_{ii} u_i = b_i - \sum_{j \in \mathcal{N}_i} a_{ij} u_j. \tag{3.62}$$

The zero sum property (3.58) implies that the involved coefficients satisfy

$$\sum_{j \in \mathcal{N}_i} \frac{a_{ij}}{a_{ii}} = -1.$$

Due to the assumptions that $\bar{A}$ is of nonnegative type and $b_i \le 0$, this yields

$$u_i = \frac{b_i}{a_{ii}} - \sum_{j \in \mathcal{N}_i} \frac{a_{ij}}{a_{ii}} u_j \le -\max_{j \in \mathcal{N}_i} u_j \left(\sum_{j \in \mathcal{N}_i} \frac{a_{ij}}{a_{ii}} \right) = \max_{j \in \mathcal{N}_i} u_j, \tag{3.63}$$

which proves that u_i is bounded by the maximum over the stencil of $i \in N_\Omega$. □

Corollary 3.13. *If $b_i = 0$ and $u_j = \bar{u}$, $\forall j \in \mathcal{N}_i$, then $u_i = \bar{u}$ by the local DMP.*

That is, if the source term is absent and the solution has a constant value μ at all neighboring nodes, then u_i must also assume this value. This property is guaranteed by the zero row sum condition (3.58). Only a poor discretization of the transport equation with $\nabla \cdot \mathbf{v} \equiv 0$ would produce $u_i \neq \bar{u}$ in this situation ([268], p. 39).

Remark 3.9. Since estimate (3.63) holds for any interior node, successive application of the local DMP can be used to prove (3.59) if $A_{\Omega\Omega}$ is irreducible, cf. [69, 344].

If the row sums of $\bar{A}$ are nonvanishing for some $i \in \mathcal{N}_\Omega$, then the DMP may cease to hold but positivity preservation can be inferred from the fact that $\bar{A}$ is monotone.

Theorem 3.16. *If the matrix $\bar{A}$ is given by (3.52), where $A_{\Omega\Omega}$ is monotone and $A_{\Omega\Gamma} \leq 0$, then discretization (3.51) is positivity-preserving, that is,*

$$b \geq 0 \quad \Rightarrow \quad u \geq 0. \tag{3.64}$$

Proof. The matrix $\bar{A}$ given by (3.52) is regular if and only if the block $A_{\Omega\Omega}$ is regular. Furthermore, the inverse matrices $\bar{A}^{-1}$ and $A_{\Omega\Omega}^{-1}$ are related by the formula

$$\bar{A}^{-1} = \begin{bmatrix} A_{\Omega\Omega}^{-1} & -A_{\Omega\Omega}^{-1} A_{\Omega\Gamma} \\ 0 & I \end{bmatrix}. \tag{3.65}$$

If $A_{\Omega\Omega}^{-1} \geq 0$ and $A_{\Omega\Gamma} \leq 0$, then $\bar{A}$ is monotone and $u = \bar{A}^{-1} b \geq 0$ for any $b \geq 0$. □

Remark 3.10. In the case of linear or bilinear finite elements, the interpolant u_h satisfies a local maximum principle inside each cell. That is, the value of the approximate solution at any interior point $\mathbf{x} \in \Omega$ is bounded by the nodal values at the vertices of the cell containing $\mathbf{x}$, so that Theorems 3.14–3.16 can be used to estimate $u_h(\mathbf{x})$.

Remark 3.11. Many other definitions and proofs of DMP can be found in the literature. The following result [311] is one of the most general and elegant formulations.

Theorem 3.17. *If $\bar{A}$ is an M-matrix that satisfies condition (3.57), then*

$$\max_{1 \leq i \leq N} u_i \leq \max_{j \in \mathcal{N}_+} u_j, \qquad \mathcal{N}_+ = \{1 \leq j \leq \bar{N} \mid b_j > 0\}.$$

In the case $N_+ = \emptyset$, the right-hand side of the above inequality is taken to be zero.

This theorem states that maxima occur on a set of nodes where the right-hand side of system (3.51) is positive. Remarkably, the degrees of freedom associated with $\mathcal{N}_\Omega$ and $\mathcal{N}_\Gamma$ are treated equally. For a proof, the interested reader is referred to [311].

Theorem 3.18. *If $\bar{A}$ is a linear monotone operator and $\bar{A}u \geq \bar{A}v$, then $u \geq v$.*

Proof. Due to linearity and monotonicity, $\bar{A}(u - v) \geq 0 \Rightarrow u - v \geq 0$. □

Unfortunately, only some low-order approximations can be both linear and monotone. To achieve higher accuracy, one of these requirements has to be sacrificed.

Theorem 3.19. *Any linear monotone operator that results from a discretization of first-order space derivatives can be at most first-order accurate.*

This order barrier is known as the Godunov theorem [123]. It is responsible for the tradeoff between devastating numerical diffusion and undershoots/overshoots. Since first-order accuracy is usually insufficient, the only way to avoid both side effects is to adjust the coefficients of the discrete transport operator in an adaptive way so as to enforce relevant DMP conditions *a posteriori*, that is, for a given data set.

Theorem 3.20. *Any linear monotone operator that results from a discretization of second-order space derivatives can be at most second-order accurate.*

This result is also disappointing but second-order accuracy if often sufficient for practical purposes. A simple proof of Theorems 3.19 and 3.20 for finite difference schemes is available in [164] on pp. 118-120. The lack of monotonicity for quadratic finite element discretizations of the Laplace operator was first reported in [151].

3.2.4 Desirable Mesh Properties

Some finite element approximations to elliptic problems like (3.47) are known to satisfy the DMP conditions (3.55)–(3.57) on a suitably designed mesh. The derivation of geometric constraints that ensure monotonicity has been one of the primary research directions in the DMP analysis for finite element schemes [64, 100, 179, 180]. Below, we present some useful geometric criteria in the context of linear and bilinear Galerkin discretizations of the Laplace operator in two space dimensions.

Definition 3.14. A triangular mesh is called *strongly acute* if all angles are smaller than $\pi/2$ and *weakly acute* (or *nonobtuse*) if all angles are not greater than $\pi/2$.

Theorem 3.21. *The discrete Laplace operator $\bar{A}$ for the linear finite element approximation on a triangular mesh of weakly acute type is monotone [18, 64].*

This classical result dates back to the paper by Ciarlet and Raviart [64]. In the 3D case, a tetrahedral mesh is said to be of acute type if all internal angles between the faces of tetrahedra are not greater than $\pi/2$. Again, this condition ensures that the discrete Laplace operator is monotone if linear finite elements are employed [189].

Definition 3.15. A triangular mesh is a *Delaunay triangulation* if no vertex of this mesh is inside the circumcircle of any triangle to which it does not belong.

Theorem 3.22. *The discrete Laplace operator $\bar{A}$ for the linear finite element approximation on a Delaunay triangulation is monotone [18].*

Delaunay triangulations maximize the minimum angle as to avoid excessively stretched triangles. It is known that there exists a unique Delaunay triangulation for

any set of points that do not lie on the same line. Moreover, fast algorithms are available for creating such triangulations [55, 115, 226], which makes them very popular with finite element practitioners. Figure 3.4 displays a simple 2D Delaunay triangulation generated using the MATLAB function `delaunay`. In three dimensions, no vertex of a tetrahedron is inside the circumsphere of any other tetrahedron. However, the discrete Laplacian operator for a linear FEM approximation on an arbitrary 3D Delaunay triangulation may fail to be a matrix of nonnegative type [18]. This does not necessarily cause a violation of the DMP but it cannot be ruled out anymore.

Definition 3.16. A rectangular mesh is called *nonnarrow* if the ratio of longest and shortest mesh edge is not greater than $\sqrt{2}$ for any rectangle [100].

Theorem 3.23. *The discrete Laplace operator $\bar{A}$ for the bilinear finite element approximation on a rectangular mesh of nonnarrow type is monotone [61, 100].*

This theorem explains why iterative solution techniques that rely on the M-matrix property may experience convergence problems when applied to discretizations of second-order PDEs on quadrilateral/hexahedral meshes with high aspect ratios.

Geometric DMP conditions for various elliptic and parabolic problems have been formulated building on the above results [100, 116, 179, 180]. Even if convective effects are present, it is desirable to use a sufficiently regular mesh that satisfies the above conditions, so that at least the discrete diffusion operator poses no hazard to DMP. Moreover, it may offset a nonmonotone convective part if the Peclet number is small or a sufficiently large amount of artificial diffusion is added [51, 69]. Alternatively, the *upwind triangle* method or other techniques can be used to construct a monotone approximation of convective terms [11, 167, 200, 288, 348].

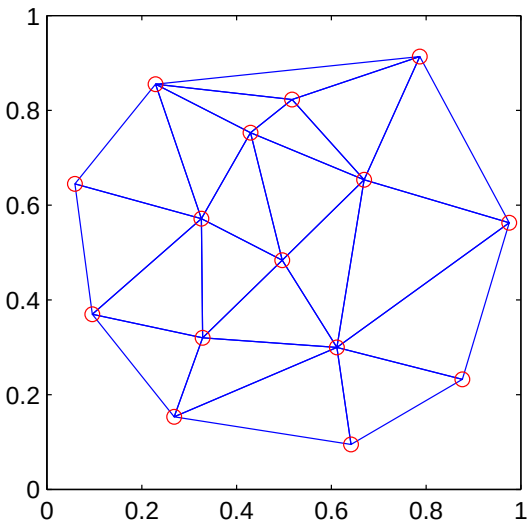


Fig. 3.4 A two-dimensional Delaunay triangulation with 15 points.

Many finite element approximations do not produce a monotone matrix, or more sophisticated tools than Theorem 3.13 are required to prove monotonicity. It is not unusual that spurious maxima or minima of significant amplitude are generated in regions where small-scale features are present and the mesh is too coarse. On the other hand, a well-resolved numerical solution satisfies the DMP even if it violates the *sufficient* monotonicity conditions that impose unrealistically severe restrictions on the properties of the mesh and on the choice of polynomial approximations.

3.3 Matrix Analysis for Unsteady Problems

Unsteady transport processes are governed by equations of parabolic and hyperbolic type in which convective terms may be dominant. The time derivative can also be interpreted as 'convection' in the positive $t-$direction. In inviscid flow problems, the discrete maximum principle and positivity preservation can be enforced within the framework of monotone, monotonicity-preserving, and total variation diminishing (TVD) methods for 1D hyperbolic conservation laws. Many representatives of these schemes are explicit and/or essentially one-dimensional. Their extensions to 2D/3D rely on dimensional splitting, which rules out the use of unstructured meshes.

It turns out that any multidimensional TVD scheme is at most first-order accurate, except in certain trivial cases [124, 216]. The concept of local extremum diminishing (LED) schemes [170, 171] makes it possible to design approximations that enjoy the TVD property in the 1D case and provide a weaker form of monotonicity in multidimensions. Interestingly enough, all of the above DMP criteria impose essentially the same constraints on the coefficients of the space discretization.

In transient computations, the time-stepping method should be chosen so as to keep the solution free of undershoots and overshoots. Since the Godunov order barrier applies to time discretizations as well, only the first-order accurate backward Euler method may be used with arbitrary time steps. In all other cases, monotonicity conditions impose a certain upper bound on the time step. This constraint may be the same or more stringent than the usual stability condition, if any. The use of the consistent mass matrix in finite element discretizations of unsteady hyperbolic and parabolic problems may also cause some complications [33, 100, 116].

3.3.1 Semi-Discrete DMP Constraints

In this section, we start with the DMP analysis for semi-discrete schemes that can be written as a system of differential algebraic equations (DAEs) of the form

$$M\frac{\mathrm{d}u}{\mathrm{d}t} = Cu + r, \tag{3.66}$$

where u is the vector of nodal values, $M = \{m_{ij}\}$ is the mass matrix, $C = \{c_{ij}\}$ is the negative of the discrete transport operator, and r is the vector of nodal sources.

A semi-discrete maximum principle for a space discretization of the form (3.66) is particularly easy to establish if M is a diagonal matrix. Finite difference and finite volume discretizations satisfy this requirement from the outset. In the realm of finite elements, it is commonly enforced using row-sum mass lumping, see Section 2.1.2.

Consider the $i-$th equation in system (3.66) with the mass matrix $M = \mathrm{diag}\{m_i\}$

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_j c_{ij} u_j + r_i. \tag{3.67}$$

Since the coefficient matrix $C = \{c_{ij}\}$ is sparse, only nearest neighbors from the set

$$\mathscr{N}_i = \{ j \neq i \mid c_{ij} \neq 0 \}$$

can contribute to the right-hand side of (3.67). If the coefficients c_{ij} sum to zero

$$\sum_j c_{ij} = 0, \tag{3.68}$$

then it follows that $c_{ii} = -\sum_{j \neq i} c_{ij}$, whence equation (3.67) can be written as

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} c_{ij} (u_j - u_i) + r_i. \tag{3.69}$$

Theorem 3.24. *The following (local) semi-discrete maximum principle holds for the solution of equation (3.69) with $r_i \leq 0$ if $m_i > 0$ and $c_{ij} \geq 0$ for all $j \neq i$*

$$u_i \geq u_j, \quad \forall j \in \mathscr{N}_i \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \leq 0. \tag{3.70}$$

Proof. The theorem states that a maximum $u_i = \max_j u_j$ cannot increase. Indeed,

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \frac{1}{m_i} \sum_{j \neq i} c_{ij} (u_j - u_i) + \frac{r_i}{m_i} \tag{3.71}$$

is nonpositive due to the fact that $m_i > 0,\ r_i \leq 0$, and $c_{ij}(u_j - u_i) \leq 0,\ \forall j \neq i$. □

In a similar vein, a minimum $u_i = \min_j u_j$ cannot decrease if r_i is nonnegative

$$u_i \leq u_j, \quad \forall j \in \mathscr{N}_i \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \geq 0. \tag{3.72}$$

In the case $r_i = 0$, the above semi-DMP states that neither maxima nor minima are enhanced, which has led Jameson [170, 171] to introduce the following definition.

Definition 3.17. A semi-discrete scheme of the form (3.69) with $r_i = 0$ is called *local extremum diminishing* (LED) if estimates (3.70) and (3.72) hold for all i.

Corollary 3.14. *The following conditions are sufficient for (3.69) to be LED*

$$m_i > 0, \quad r_i = 0, \quad c_{ij} \geq 0, \qquad \forall i,\ \forall j \neq i. \tag{3.73}$$

Remark 3.12. Flux-corrected transport (FCT) algorithms [42, 355] impose essentially the same constraints (no new maxima or minima, no growth of existing ones).

Remark 3.13. The term *local* implies that maxima and minima are taken over the stencil of individual nodes. A global semi-discrete maximum principle and, hence, L_∞-stability can be readily inferred from the LED criterion and Theorem 3.24.

The concept of LED schemes provides [171] "a convenient basis for the construction of nonoscillatory schemes on both structured and unstructured meshes." Indeed, conditions like (3.73) are easy to check and enforce for arbitrary discretizations.

In the context of three-point finite difference schemes for 1D problems, the nonnegativity of both off-diagonal coefficients is also required by Harten's TVD conditions [137]. It turns out that there is a close relationship between TVD and LED space discretizations. In one space dimension, the total variation

$$TV(u,t) = \sum_i |u_{i+1}(t) - u_i(t)|, \qquad \forall t \geq 0$$

must be a nonincreasing function of time for a semi-discrete scheme to be TVD. If zero boundary values are prescribed at both endpoints of the 1D domain, then [171]

$$TV(u,t) = 2\left(\sum_{j \in \mathcal{N}_{\max}} u_j(t) - \sum_{k \in \mathcal{N}_{\min}} u_k(t) \right), \tag{3.74}$$

where $\mathcal{N}_{\max}$ and $\mathcal{N}_{\min}$ contain the indices of local maxima and minima, respectively

$$\mathcal{N}_{\max} = \{ j \mid u_j \geq u_{j\pm1} \}, \qquad \mathcal{N}_{\min} = \{ k \mid u_k \leq u_{k\pm1} \}.$$

If the LED constraint holds, then these maxima and minima cannot grow, whence

$$TV(u,t_1) \geq TV(u,t_2), \qquad \forall t_1 \leq t_2.$$

Therefore, a semi-discrete LED scheme enjoys the TVD property in one dimension.

Remark 3.14. Due to Theorems 3.19 and 3.20, a linear LED/TVD discretization of convective/diffusive terms is at most first/second-order accurate, respectively.

It is certainly incorrect to demand that the numerical scheme be local extremum diminishing if the exact solution does not satisfy the LED constraint. However, it is still possible to prove the following property implied by Theorems 3.10 and 3.12.

Definition 3.18. A semi-discrete scheme of the form (3.67) is called *positive* if

$$u_i(0) \geq 0, \quad \forall i \quad \Rightarrow \quad u_i(t) \geq 0, \quad \forall i,\ \forall t > 0. \tag{3.75}$$

To avoid a common misunderstanding, we emphasize that the numerical solution is not forced to be positive if there is $j \neq i$ such that $u_j(0) < 0$. Positivity preservation means that the numerical scheme cannot produce *nonphysical* negative values, i.e., undershoots. Likewise, an initially nonpositive solution should preserve its sign, so that no overshoots are generated. Sink terms may destroy positivity and require special treatment in accordance with the fourth basic rule from Section 1.6.3.

Theorem 3.25. *The following conditions are sufficient for (3.67) to be positive*

$$m_i > 0, \quad r_i \geq 0, \quad c_{ij} \geq 0, \qquad \forall i, \forall j \neq i. \tag{3.76}$$

Proof. Suppose that $u_i(t) = 0$ and $u_j(t) \geq 0$ for all $j \in \mathcal{N}_i$. Then the time derivative of u_i satisfies (3.71) and is nonnegative under the above sufficient conditions. □

Corollary 3.15. *Let u and v be solutions computed by a linear positive scheme using the initial data $u_i(0) \geq v_i(0)$, $\forall i$, all other settings being fixed. Then*

$$u_i(t) \geq v_i(t), \qquad \forall i, \quad \forall t \geq 0.$$

Remark 3.15. In the 1D case, finite difference schemes that satisfy such a comparison principle for initial data are *monotone* by definition [216].

3.3.2 Fully Discrete DMP Constraints

After the discretization in time and implementation of the Dirichlet boundary conditions, the fully discrete counterpart of (3.66) can be written in the form

$$\begin{bmatrix} A_{\Omega\Omega} & A_{\Omega\Gamma} \\ 0 & I \end{bmatrix} \begin{bmatrix} u_\Omega \\ u_\Gamma \end{bmatrix} = \begin{bmatrix} b_\Omega \\ b_\Gamma \end{bmatrix}, \tag{3.77}$$

where u_Ω is the vector of unknowns for the current time step, b_Γ is the vector of prescribed boundary values, and b_Ω depends on the previously computed data

$$b_\Omega = B_{\Omega\Omega} g_\Omega + B_{\Omega\Gamma} g_\Gamma + s_\Omega.$$

The matrix blocks $B_{\Omega\Omega}$ and $B_{\Omega\Gamma}$ contain the coefficients of the explicit part that depends on the vector of old nodal values $g = g_\Omega \cup g_\Gamma$. The remainder s_Ω represents the contribution of source terms and Neumann boundary conditions, if any.

For a two-level time-stepping method, $u = u^{n+1}$ and $g = u^n$, where the superscripts refer to the time levels t^{n+1} and $t^n = t^{n+1} - \Delta t$, respectively. In fractional-step algorithms, a pair of intermediate solutions may also be denoted by u and g.

Definition 3.19. The global discrete maximum principle holds for (3.77) if

$$s_\Omega \leq 0 \quad \Rightarrow \quad u_i \leq \max_j g_j, \qquad \forall i \in N_\Omega. \tag{3.78}$$

Theorem 3.26. *Let the first N_Ω rows sums of A and B be equal (cf. [101]), i.e.,*

$$\sum_{j=1}^{\bar{N}} a_{ij} = \sum_{j=1}^{\bar{N}} b_{ij}, \qquad \forall i \in N_\Omega \tag{3.79}$$

and the block $A_{\Omega\Omega}$ be regular. Then the solution u_Ω of problem (3.77) satisfies the global discrete maximum principle under the following sign conditions

$$A_{\Omega\Omega}^{-1} \geq 0, \quad A_{\Omega\Gamma} \leq 0, \quad B_{\Omega\Omega} \geq 0, \quad B_{\Omega\Gamma} \geq 0. \tag{3.80}$$

Proof. Consider $w = u - \mu$ and $v = g - \mu \leq 0$, where $\mu = \max_j g_j$. Due to (3.79)

$$\sum_{j=1}^{\bar{N}} (b_{ij} v_j - a_{ij} w_j) + s_i = \sum_{j=1}^{\bar{N}} (b_{ij} g_j - a_{ij} u_j) + s_i = 0, \qquad \forall i \in N_\Omega. \tag{3.81}$$

It follows that $A_{\Omega\Omega} w_\Omega + A_{\Omega\Gamma} w_\Gamma = B_{\Omega\Omega} v_\Omega + B_{\Omega\Gamma} v_\Gamma + s_\Omega$, where $s_\Omega \leq 0$ by assumption and $w_\Gamma \leq 0,\ v \leq 0$ by definition. Invoking (3.80), we infer that

$$w_\Omega = A_{\Omega\Omega}^{-1} [B_{\Omega\Omega} v_\Omega + B_{\Omega\Gamma} v_\Gamma - A_{\Omega\Gamma} w_\Gamma + s_\Omega] \leq 0.$$

This estimate proves the global DMP which requires that $u_i \leq \mu$ for all $i \in N_\Omega$. □

As in the case of steady transport equations, it is also possible to estimate the unknown nodal value u_i in terms of the data defined at a few neighboring nodes.

Definition 3.20. The local discrete maximum principle holds for (3.77) if

$$s_i \leq 0 \quad \Rightarrow \quad u_i \leq \mu_i, \quad \forall i \in \mathcal{N}_\Omega, \tag{3.82}$$

where μ_i denotes the maximum taken over $\{u_j \mid a_{ij} \neq 0,\ j \neq i\}$ and $\{g_j \mid b_{ij} \neq 0\}$.

Theorem 3.27. *The solution of (3.77) satisfies (3.82) subject to the row-sum constraint (3.79) and conditions of the third basic rule from Section 1.6.3, i.e.,*

$$a_{ii} > 0, \quad b_{ii} \geq 0, \qquad \forall i, \tag{3.83}$$

$$a_{ij} \leq 0, \quad b_{ij} \geq 0, \qquad \forall j \neq i. \tag{3.84}$$

Proof. Let $i \in N_\Omega$ be any interior node. Following the proof of Theorem 3.26, we introduce the auxiliary functions $w = u - \mu_i$ and $v = g - \mu_i$. By definition of μ_i for the local DMP, $w_j \leq 0$ for all $j \neq i$ such that $a_{ij} \neq 0$, and $v_k \leq 0$ for $1 \leq k \leq \bar{N}$ such that $b_{ik} \neq 0$. The row sum property leads to a relation of the form (3.81), whence

$$a_{ii} w_i = \sum_{j=1}^{\bar{N}} b_{ij} v_j - \sum_{j \neq i}^{\bar{N}} a_{ij} w_j + s_i. \tag{3.85}$$

Conditions (3.83)–(3.84) imply that $a_{ii} > 0$ and the right-hand side of equation (3.85) is nonpositive for $s_i \leq 0$. This proves that $w_i \leq 0$, that is, $u_i \leq \mu_i$. □

Remark 3.16. Note that the steady-state counterpart (3.51) of the local discrete maximum principle (3.77) is recovered for $u = g$. Thus, pseudo-time stepping can be used not only to march the solution to a steady state but also to prove DMP for stationary problems. Of course, a proof of convergence should also be provided.

It remains to formulate sufficient conditions of positivity preservation for discretizations of transport equations in which sources, sinks, or compressibility effects may destroy the 'equal row sum' property (3.79) of the two coefficient matrices. Due to the presence of a strictly diagonally dominant mass matrix, the usual way to establish positivity preservation for time-dependent problems is as follows.

Theorem 3.28. *If the coefficients of (3.77) satisfy conditions (3.83)–(3.84) and*

$$\sum_j a_{ij} > 0, \quad \forall i \in \mathcal{N}_\Omega, \tag{3.86}$$

then such a discretization is guaranteed to be positivity-preserving, that is,

$$s_\Omega \geq 0, \quad g \geq 0 \quad \Rightarrow \quad u_\Omega \geq 0.$$

Proof. Due to (3.83)–(3.84) and (3.86), the block $A_{\Omega\Omega}$ is monotone by Corollary (3.12). Thus, $u_\Omega = A_{\Omega\Omega}^{-1}[B_{\Omega\Omega}g_\Omega + B_{\Omega\Gamma}g_\Gamma - A_{\Omega\Gamma}u_\Gamma + s_\Omega] \geq 0$. □

Corollary 3.16. *If solutions u and v are computed by the same linear positivity-preserving scheme using the data $g \geq h$, all other settings being fixed, then $u \geq v$.*

This comparison principle represents a fully discrete counterpart of Corollary 3.15. As usual, the proof is based on Theorem 3.28 applied to the vector $w = u - v$.

3.3.3 Positive Time-Stepping Methods

The fully discrete maximum principles may turn out to be more restrictive than their semi-discrete counterparts. Even if the space discretization is designed to be monotone, the final algebraic system may fail to satisfy the DMP conditions, especially in the case of a finite element approximation with a nondiagonal mass matrix.

According to Theorem 3.28, the following set of sufficient conditions guarantees positivity preservation for a discretization that can be cast into the form (3.77)

- the diagonal block $A_{\Omega\Omega}$ has (i) positive diagonal entries, (ii) nonpositive off-diagonal entries, and (iii) positive row sums (strict diagonal dominance).
- there are no positive coefficients in $A_{\Omega\Gamma}$ and negative ones in $B_{\Omega\Omega}$ or $B_{\Omega\Gamma}$.

Furthermore, the global and local DMP hold under the additional condition (3.79).

Some time-stepping schemes preserve positivity, at least if the time step is chosen so as to satisfy the above algebraic constraints on the coefficients of discrete operators. For example, let (3.66) be discretized in time by the $\theta-$*scheme*

$$M\frac{u^{n+1}-u^n}{\Delta t} = \theta C u^{n+1} + (1-\theta)Cu^n + r, \tag{3.87}$$

where $0 \le \theta \le 1$ is an implicitness parameter. This formulation combines the forward Euler ($\theta = 0$), Crank-Nicolson ($\theta = \frac{1}{2}$), and backward Euler ($\theta = 1$) methods.

Collecting all terms that depend on $u = u^{n+1}$ and $g = u^n$, we can multiply (3.87) by the time step Δt and write this algebraic system in the equivalent form

$$Au^{n+1} = Bu^n + s,$$

where $s = \Delta t r$ and the involved coefficient matices are defined as follows

$$A = M - \theta \Delta t C, \qquad B = M + (1-\theta)\Delta t C.$$

Therefore, the objective is to check the sign and magnitude of the coefficients

$$a_{ij} = m_{ij} - \theta \Delta t c_{ij}, \qquad b_{ij} = m_{ij} + (1-\theta)\Delta t c_{ij}. \tag{3.88}$$

Suppose that the underlying space discretization is of positive type, that is,

$$c_{ij} \ge 0, \quad \forall i, \forall j \ne i. \tag{3.89}$$

This condition holds for some upwind-type discretizations of convective terms, as well as for centered schemes if there is enough (physical or artificial) diffusion. Moreover, the geometric DMP conditions for finite element approximations of diffusive terms may require that the mesh be of acute/nonnarrow type, see Section 3.2.4. Alternatively, flux/slope limiters can be used to enforce condition (3.89).

If the mass matrix M is diagonal, all off-diagonal coefficients have the right sign

$$m_{ij} = 0, \quad c_{ij} \ge 0 \quad \Rightarrow \quad a_{ij} \le 0, \quad b_{ij} \ge 0, \quad \forall i, \forall j \ne i.$$

In the case of (linear and bilinear) finite elements, the consistent mass matrix has some positive off-diagonal entries. Thus, the corresponding coefficients b_{ij} remain nonnegative but $a_{ij} \le 0$ only if the time step Δt satisfies the following lower bound

$$\Delta t \ge \frac{m_{ij}}{\theta c_{ij}} = \Delta t_{\min}, \qquad \forall i, \forall j \ne i. \tag{3.90}$$

Obviously, this requirement is impossible to fulfil with $\theta = 0$ or $c_{ij} = 0$. Thus, the time-stepping scheme must be implicit ($\theta > 0$) and the diffusive term must be nonvanishing for a consistent-mass finite element method to satisfy the sufficient conditions of positivity preservation [100, 116, 246]. To circumvent this problem, some modifications of the mass matrix were proposed by Berzins [33, 34, 35].

Conditions (3.83) and (3.86) for the diagonal coefficient a_{ii} require that

$$\sum_j [m_{ij} - \theta \Delta t c_{ij}] > 0, \qquad \forall i. \tag{3.91}$$

The diagonal coefficient b_{ii} is nonnegative under the following condition

$$m_{ii} + (1-\theta)\Delta t c_{ii} \geq 0, \qquad \forall i \tag{3.92}$$

which is responsible for stability and boundedness of the explicit part. The backward Euler method satisfies this condition automatically but the Crank-Nicolson scheme is positivity-preserving only for sufficiently small time steps, although it is unconditionally stable. In the case of the forward Euler method, the time step must be small anyway for stability reasons, so positivity can be achieved at no extra cost.

Remark 3.17. Condition (3.92) with $\theta < 1$ is particularly restrictive if the coefficient c_{ii}^n assumes a large negative value. This situation occurs, e.g., if diffusion plays a dominant role or the divergence of the velocity field is large and positive.

Due to (3.91) and (3.92) the upper bound for largest admissible time step is

$$\Delta t \leq \min\left\{ \frac{\sum_j m_{ij}}{\max\{0, \theta \sum_j c_{ij}\}}, \frac{m_{ii}}{(\theta-1)c_{ii}} \right\} = \Delta t_{\max}, \qquad \forall i. \tag{3.93}$$

In the case of a 1D hyperbolic conservation law discretized in space by the explicit first-order upwind scheme, this bound reduces to the standard CFL condition.

Theorem 3.29. *If the space discretization satisfies (3.89), while the time step satisfies (3.90) and (3.93), then the $\theta-$scheme (3.87) is positivity-preserving.*

The restrictions on the choice of θ and Δt are very stringent in the case of consistent-mass FEM since the time step Δt cannot be greater than $\Delta t_{\max}$ and smaller than $\Delta t_{\min}$. The use of diagonal mass matrices and/or implicit algorithms is preferable from the standpoint of positivity preservation, see also [100, 116, 164, 246].

The analysis of the $\theta-$scheme can be extended to other time discretizations. For example, let $M = \mathrm{diag}\{m_i\}$ and consider an explicit L-stage Runge-Kutta method

$$Mu^{(l)} = \sum_{k=0}^{l-1} \gamma_{kl} \left(Mu^{(k)} + \theta_{kl} \Delta t C^{(k)} u^{(k)} \right), \tag{3.94}$$

$$u^{(0)} = u^n, \quad u^{n+1} = u^{(L)}, \quad l = 1, \ldots, L. \tag{3.95}$$

This time discretization is called a *TVD Runge-Kutta method* if it preserves the TVD property of the underlying space discretization for scalar conservation laws in 1D. Such time-stepping schemes were introduced by Shu and Osher [304, 306] and analyzed by Gottlieb and Shu [125]. Other (non-TVD but linearly stable) Runge-Kutta methods can generate spurious oscillations even if the semi-discrete scheme is local extremum diminishing and/or positive, see [125] for a numerical example.

Theorem 3.30. *A Runge-Kutta method of the form (3.94)–(3.95) with*

$$0 \leq \gamma_{kl} \leq 1, \quad \sum_{k=0}^{l-1} \gamma_{kl} = 1, \quad 0 \leq \theta_{kl} \leq 1 \tag{3.96}$$

is positivity-preserving if the time step Δt satisfies condition (3.92) for $\theta = 0$ [125].

Proof. Conditions (3.96) imply that the right-hand side of (3.94) is a convex combination of forward Euler predictors with Δt replaced by $\theta_{kl}\Delta t$, where $\theta_{kl} \in [0,1]$. Therefore, positivity is preserved under condition (3.92) with $\theta = 0$. □

Remark 3.18. The above proof of positivity is only valid for diagonal (lumped) mass matrices. Otherwise, positive off-diagonal coefficients $a_{ij} = m_{ij}$ violate (3.84).

In the review paper by Gottlieb, Shu, and Tadmor [126], explicit high-order time-stepping schemes that comply with the requirements of Theorem 3.30 were renamed into *strong stability-preserving* (SSP) time discretizations. This more suitable term refers to the ability of SSP methods to maintain boundedness not only in the total variation norm but also in other norms. If the forward Euler method is SSP, so are its high-order counterparts, perhaps under a different restriction on the time step [126].

The optimal (in terms of the time step restriction and computational cost) SSP Runge-Kutta scheme of second order is the well-known *Heun method* [125]

$$Mu^{(1)} = Mu^n + \Delta t C^n u^n, \tag{3.97}$$

$$Mu^{n+1} = \frac{1}{2}\left(M(u^n + u^{(1)}) + \Delta t C^{(1)} u^{(1)}\right). \tag{3.98}$$

The final solution u^{n+1} represents the average of the forward Euler predictor $u^{(1)}$ and a backward Euler corrector evaluated using $u^{(1)}$ in place of u^{n+1}.

The optimal SSP Runge-Kutta time discretization of third order is given by [125]

$$Mu^{(1)} = Mu^n + \Delta t C^n u^n, \tag{3.99}$$

$$Mu^{(2)} = \frac{1}{4}\left(M(3u^n + u^{(1)}) + \Delta t C^{(1)} u^{(1)}\right), \tag{3.100}$$

$$Mu^{n+1} = \frac{1}{3}\left(M(u^n + 2u^{(2)}) + 2\Delta t C^{(2)} u^{(2)}\right). \tag{3.101}$$

For a comprehensive review and systematic study of SSP Runge-Kutta / multistep methods that provide high accuracy and low storage, the reader is referred to [126, 305]. Aspects of positivity preservation and monotonicity concepts for numerical integration of initial value problems are also addressed in [164], pp. 185–196.

Since SSP time-stepping methods are usually of the same form and incur approximately the same computational cost per time step as traditional ODE solvers, it is worthwhile to use them whenever possible. Even if the corresponding theoretical restriction on the time step is smaller than the linear stability bound, an SSP method tends to be more stable in the range of time steps that lie in-between [305]. In many cases, there is no penalty for taking time steps far beyond the SSP bound because its derivation is usually based on sufficient (rather than necessary) conditions.

3.4 Summary

The abundance of theorems and proofs in the present chapter makes it very different from the rest of the book. The mastery of this material is not important for programmers and users of CFD codes. However, the presented theory illustrates the close relationship between the physical nature of transport processes and qualitative properties of solutions to partial differential equations of different types. Moreover, numerical approximations were shown to inherit these properties under certain restrictions on the coefficients of the algebraic systems to be solved. This knowledge can contribute to the development of numerical methods for transport equations.

Since most analytical and numerical studies are restricted to a certain class of problems (steady/unsteady, viscous/inviscid etc.), no unified theory of continuous and discrete maximum principles seems to be available to date. This chapter was written in an attempt to fill this gap and illustrate some striking similarities between positivity and monotonicity constraints that have been known under different names and developed independently by different groups of researchers. Another goal was to retrace the basic steps involved in the discretization process and discuss the implications of each step in terms of maximum principles and positivity preservation. As we have seen, a simple analysis of matrix properties provides a link to the governing equation and valuable information about the properties of numerical solutions.

The theoretical framework presented in this chapter rests on a set of sufficient conditions that rule out the onset of spurious undershoots or overshoots. To this end, the left-hand side matrix is required to be (irreducibly or strictly) diagonally dominant with nonpositive off-diagonal coefficients. Its diagonal entries and all coefficients of the explicit part are required to be nonnegative. These conditions are known from classical texts on numerical linear algebra [339, 354] and coincide with the requirements of Patankar's basic rules [268] frequently referred to in this book. As a useful byproduct, one obtains computable bounds for admissible time steps.

Some discretizations of transport equations are guaranteed to be monotone but their accuracy is restricted by the order barriers that apply to linear approximations of convective and diffusive terms. The only way to achieve higher accuracy while maintaining monotonicity is to devise a smart feedback mechanism that extracts information from the approximate solution and constrains the coefficients of the numerical scheme on the basis of this information. This design principle leads to the algebraic flux correction paradigm to be introduced in the next chapter.

Chapter 4
Algebraic Flux Correction

Multidimensional transport problems with interior/boundary layers or discontinuities represent a formidable challenge for numerical techniques, especially in the case of unstructured meshes and implicit time-stepping schemes. As usual, the reason is the tradeoff between spurious oscillations and excessive numerical diffusion. In the realm of finite elements, it is common practice to combat the latter evil by adding some anisotropic artificial diffusion acting in the streamline direction only. Ironically, the Galerkin discretization of diffusive terms may violate the discrete maximum principle (DMP) instead of helping the convective part to satisfy it. This multidimensional side effect is rarely taken into account. Moreover, conventional stabilization techniques operate at the continuous level and involve free parameters which are highly problem-dependent. It is difficult to achieve the M-matrix property and maintain monotonicity in this fashion. This is why even stabilized finite element methods with favorable theoretical properties tend to produce oscillatory results.

In this chapter, we revisit the algebraic constraints that guarantee the validity of a DMP and enforce them in a mass-conserving way using a set of diffusive and antidiffusive fluxes. After a brief presentation of the basic ideas, we introduce the algebraic flux correction (AFC) paradigm which will serve as a general framework for the design of multidimensional flux/slope limiters in an unstructured grid environment. We address the iterative treatment of nonlinear algebraic systems and the optimal choice of the limiting strategy. Finally, we apply the developed tools to finite element discretizations of elliptic, hyperbolic, and parabolic transport problems.

4.1 Nonlinear High-Resolution Schemes

The trend towards the use of unstructured mesh methodologies in general-purpose CFD codes has stimulated a lot of research on fully multidimensional generalizations of classical high-resolution schemes for transport equations. While Godunov-type methods can be readily integrated into finite volume codes, there is no natural extension to continuous (linear or bilinear) Galerkin discretizations. Similarly, the

development of finite element schemes based on algebraic flux limiting techniques requires a major revision, or at least a new interpretation, of their finite difference prototypes which are typically explicit and/or tailored for Cartesian meshes.

In the late 1980s and early 1990s, flux-corrected transport (FCT) and total variation diminishing (TVD) schemes were extended to explicit algorithms based on linear and bilinear Galerkin finite element discretizations [8, 232, 244, 266, 298, 299]. Conservative flux decompositions, edge-based data structures, and the equivalence to finite volumes have made it possible to generalize many one-dimensional concepts, such as 'upwind difference' or 'slope ratio', in a rather straightforward way [239, 243]. These remarkable advances have formed the basis for the development of high-resolution finite element schemes for compressible CFD and aerodynamics. However, they were met with little enthusiasm by the theoretically oriented fraction of the FEM community, perhaps, due to the lack of mathematical rigor and a possible loss of the Galerkin orthogonality as a result of such 'variational crimes.'

As of this writing, most finite element schemes for convection-dominated transport equations still rely on linear stabilization. For decades, the mainstream approach has been represented by the Streamline Upwind Petrov-Galerkin (SUPG) and Galerkin Least Squares (GLS) methods [172]. A variety of 'improvements' and 'optimal' values of free parameters have been proposed. Furthermore, front capturing techniques have been devised for problems with interior and boundary layers in an attempt to suppress spurious oscillations. In most cases, the involved stabilization mechanisms are designed at the continuous level using heuristic arguments and *ad hoc* parameter fitting rather than a rigorous mathematical theory which ensures the validity of the discrete maximum principle. Therefore, undershoots and/or overshoots are to be expected whenever the solution develops steep gradients [172]. Even if the ripples are relatively small, they may cause irrecoverable damage in situations when positivity preservation is a must for physical and numerical reasons.

In recent years, interior penalty (edge stabilization) techniques [51, 50, 293, 327] have become increasingly popular with finite element practitioners. In this approach, the amount of stabilization is proportional to the jumps of the gradient across interelement boundaries. The resulting solutions are not as sensitive to the choice of free parameters as in the case of SUPG-like methods. The inclusion of a nonlinear shock capturing term with a sufficiently large coefficient makes it possible to prove the weak DMP property [52]. Edge stabilization appears to be a very promising methodology but it is not a free lunch since the addition of jump terms leads to a finite element discretization with a wider stencil and a different sparsity pattern.

The latest comparative studies of finite element methods for stationary and time-dependent transport problems [174, 175] speak in favor of high-resolution schemes based on the *algebraic flux correction* (AFC) paradigm [200, 203, 205, 206]. The basic idea is very simple: if a given discretization fails to satisfy the sufficient conditions of the discrete maximum principle, they can be enforced by adding a discrete diffusion operator that adjusts itself adaptively to the local solution behavior. This design principle represents a 'black-box' approach to the construction of constrained high-order discretizations, whereby all the necessary information is inferred from the entries of a given matrix. Algebraic flux correction schemes can be equipped

with symmetric or upwind-biased flux limiters which differ in the definition of local extremum diminishing (LED) upper and lower bounds. It is also possible to constrain the local slopes edge-by-edge so as to limit the jumps of the gradient.

The marriage of implicit FEM-FCT schemes [191, 203, 205] and their multidimensional FEM-TVD counterparts [206] within the framework of algebraic flux correction [192, 200] was followed by the investigation of many related concepts and limiting techniques [194, 196, 204, 258]. As the methodology has evolved and matured, the growing number of publications has made it difficult for the readers to keep track of recent developments and choose the right algorithms. Therefore, the time has come to review the state of the art, summarize the most important results, and give some practical recommendations. This is the goal of the present chapter.

4.1.1 Design Philosophy and Tools

As a standard model problem, consider an unsteady conservation law of the form

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f} = 0 \quad \text{in } \Omega. \tag{4.1}$$

For the linear convection-diffusion equation, the flux function $\mathbf{f}(u)$ is given by

$$\mathbf{f} = \mathbf{v}u - \varepsilon \nabla u,$$

where $\mathbf{v}$ is a known velocity field and ε is a constant diffusion coefficient. The case of a nonuniform diffusion tensor $\mathscr{D}(\mathbf{x},t)$ will be considered in Section 4.5.

The above problem is endowed with appropriate boundary conditions imposed in a strong or weak sense. The initial condition is given by the formula

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega.$$

Modern front-capturing methods for (4.1) constrain the coefficients of a high-order scheme so as to keep it as accurate as possible without generating undershoots or overshoots. The basic ingredients of such nonlinear discretizations are [357]

1. physical or mathematical constraints that guarantee certain qualitative properties;
2. a stable high-order scheme that satisfies 1 only for sufficiently smooth solutions;
3. a monotone low-order scheme which is guaranteed to satisfy 1 for arbitrary data;
4. a simple mechanism for blending 2 and 3 so as to enforce 1 in an adaptive fashion.

Of course, it is also essential to maintain mass conservation. To this end, it is sufficient to express the differences between 2, 3, and 4 in terms of internodal fluxes.

In algebraic flux correction schemes, the constraints to be imposed (item 1) are based on the theory of discrete maximum principles (DMP) and positivity preservation, as presented in Chapter 3. A discrete diffusion operator is employed to convert an accurate high-order discretization (item 2) into a nonoscillatory low-order one

(item 3), after which a limited amount of compensating antidiffusion is applied in order to prevent a global loss of accuracy (item 4). All of these manipulations are performed at the discrete level using the set of sufficient conditions (3.83)–(3.84) to enforce and maintain the M-matrix property which guarantees monotonicity. This methodology represents a generalization of classical FCT and TVD schemes.

Consider equation (4.1) discretized in space by a centered finite difference, finite volume, or finite element method (item 1) on a structured or unstructured mesh

$$\sum_j m_{ij}\frac{\mathrm{d}u_j}{\mathrm{d}t} = \sum_j k_{ij}u_j, \tag{4.2}$$

where u_j is a time-dependent nodal value, m_{ij} is an entry of the mass matrix, and k_{ij} is an entry of the discrete transport operator (see Chapter 2). This approximation is supposed to be conservative, linear, and more than first-order accurate.

Also, consider a low-order discretization (item 2) with a diagonal mass matrix

$$m_i\frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_j l_{ij}u_j. \tag{4.3}$$

The job of (4.2) is to approximate smooth data with high precision, whereas (4.3) must produce nonoscillatory solutions even in the presence of steep fronts.

By definition, a space discretization of equation (4.1) is *positivity-preserving* if

$$u_i(0) \geq 0, \quad \forall i \quad \Rightarrow \quad u_i(t) \geq 0, \quad \forall i,\ \forall t > 0.$$

A sufficient condition for (4.3) to possess this property is given by Theorem 3.25

$$m_i > 0, \quad l_{ij} \geq 0, \qquad \forall i,\ \forall j \neq i. \tag{4.4}$$

If the coefficient matrix $L = \{l_{ij}\}$ has zero row sums, then (4.3) is equivalent to

$$m_i\frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j\neq i} l_{ij}(u_j - u_i). \tag{4.5}$$

Such a semi-discrete scheme proves not only positivity-preserving but also *local extremum diminishing* (LED) under conditions (4.4). Theorem 3.24 states that

$$u_i \geq u_j, \quad \forall j \neq i \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \leq 0,$$

whence a maximum cannot increase. Likewise, a minimum cannot decrease since

$$u_i \leq u_j, \quad \forall j \neq i \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \geq 0.$$

After the discretization in time by the standard $\theta-$scheme or a suitable Runge-Kutta method, conditions (4.4) ensure the validity of a discrete maximum principle, perhaps under additional restrictions on the time step size, see Section 3.3.3.

At large Peclet numbers, both (4.2) and (4.3) may fail to resolve the solution properly. Due to the Godunov theorem [123], a linear high-order discretization of the form (4.2) cannot be positivity-preserving for arbitrary data. In fact, it tends to produce spurious oscillations in the neighborhood of steep fronts. This problem can be cured by adding some artificial diffusion. On the other hand, the accuracy of a linear low-order discretization like (4.3) can be enhanced by removing excessive numerical diffusion. As long as the diffusive and antidiffusive terms admit a conservative flux decomposition, they do not affect the global mass balance but make it possible to improve the distribution of mass and satisfy the discrete maximum principle. The algebraic flux correction methodology to be presented below provides a general approach to finding the right amount of artificial diffusion and antidiffusion.

4.1.2 Artificial Diffusion Operators

For the time being, we assume that the mesh is sufficiently regular for the discretization of the diffusive term to satisfy the LED criterion. Under this assumption, undershoots and overshoots are caused by the contributions of the convective term and/or of a nondiagonal mass matrix. In the case of finite difference and finite volume approximations, the use of first-order upwinding leads to the least diffusive linear positivity-preserving scheme. For linear and bilinear finite element discretizations, the same effect can be achieved by adding a discrete diffusion operator [191, 205].

The system of equations (4.2) for a FEM discretization of (4.1) can be written as

$$M_C \frac{\mathrm{d}u}{\mathrm{d}t} = Ku, \tag{4.6}$$

where u is the vector of time-dependent nodal values, $M_C = \{m_{ij}\}$ is the consistent mass matrix, and $K = \{k_{ij}\}$ is (the negative of) the discrete transport operator. The coefficients of M_C and K are defined and calculated as explained in Chapter 2.

Remark 4.1. The skew-symmetric part $\frac{1}{2}(K - K^T)$ is associated with a centered discretization of $-\mathbf{v}\cdot\nabla$, whereas $\frac{1}{2}(K + K^T) - \mathrm{diag}\{K\}$ is a discrete (anti-)diffusion operator. The latter may include streamline diffusion used for stabilization purposes and/or to achieve better phase accuracy, as in the case of Taylor-Galerkin methods.

A scheme of the form (4.6) is neither LED nor positivity-preserving as long as

$$\exists m_{ij} \neq 0, \quad \exists k_{ij} < 0, \quad j \neq i.$$

In order to enforce the discrete maximum principle for (4.6) it is sufficient to

- perform row-sum mass lumping and replace the consistent mass matrix M_C by

$$M_L = \mathrm{diag}\{m_i\}, \qquad m_i = \sum_j m_{ij}, \tag{4.7}$$

- approximate the discrete transport operator K by its low-order counterpart

$$L = K + D, \qquad l_{ij} \geq 0, \quad \forall j \neq i, \tag{4.8}$$

where $D = \{d_{ij}\}$ stands for an artificial diffusion operator such that

$$\sum_j d_{ij} = \sum_i d_{ij} = 0, \qquad d_{ij} = d_{ji}, \quad \forall i, j. \tag{4.9}$$

These manipulations lead to a linear positivity-preserving scheme of the form

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu. \tag{4.10}$$

For every pair of nonzero off-diagonal entries k_{ij} and k_{ji}, the artificial diffusion coefficient $d_{ij} = d_{ji}$ should ensure that $l_{ij} = k_{ij} + d_{ij} \geq 0$ and $l_{ji} = k_{ji} + d_{ij} \geq 0$, as required by (4.8). Therefore, the lower bound for d_{ij} is [170, 191, 205, 301]

$$d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\} = d_{ji}, \qquad \forall j \neq i. \tag{4.11}$$

This is just enough to eliminate all negative off-diagonal entries of the high-order operator K. Artificial diffusion coefficients that enforce positivity in this way were used to construct low-order schemes for FCT as early as in the mid-1970s [43].

For the row sums of $D = \{d_{ij}\}$ to be zero, its diagonal entries are defined as

$$d_{ii} := -\sum_{j \neq i} d_{ij}, \qquad \forall i. \tag{4.12}$$

By construction, $d_{ij} = d_{ji}$ for all i and j. Hence, the resulting matrix satisfies (4.9) and is a representative of discrete diffusion operators defined in Section 2.1.6.2.

Remark 4.2. If the matrix K is skew-symmetric, as in the case of $k_{ij} = -k_{ji}$ given by (2.54), then the diffusion coefficient (4.11) reduces to $d_{ij} = |k_{ij}|$ for all $j \neq i$.

Remark 4.3. In nonlinear inviscid flow problems, it might happen that $k_{ij} < 0$ and, consequently, $l_{ij} = 0$ for some i and all $j \neq i$. If the matrix K has zero row sums, then $l_{ii} = 0$ as well, which implies that L is reducible and actually singular. The contribution of the mass matrix or 'relaxation by inertia' (see next section) renders the fully discrete problem well-posed but the numerical solution may exhibit spurious kinks. These artifacts typically occur at stagnation points, where the velocity reverses its sign creating an internal 'inlet' with unspecified 'inflow' values. A common remedy is to replace the smallest artificial diffusion coefficient (4.11) by [139, 235]

$$d_{ij} := \begin{cases} d_{ij}, & \text{if } d_{ij} \geq \delta, \\ \frac{d_{ij}^2 + \delta^2}{2\delta}, & \text{if } d_{ij} \leq \delta, \end{cases} \qquad \forall j \neq i,$$

where $\delta > 0$ is a small parameter that does not allow d_{ij} to vanish and produce a row of zero entries. In gas dynamics, this trick is known as the *entropy fix*. The threshold δ may be taken constant or designed to be a function of the local flow conditions.

Remark 4.4. Physical diffusion can be built into the matrices K and L before or after the computation of D. In the former case, the value of the artificial diffusion coefficient d_{ij} given by (4.11) is reduced accordingly. This may or may not be desirable. For example, the negative numerical diffusion inherent to the standard Galerkin discretization of convective terms would offset some physical diffusion and result in artificial steepening of solution profiles. On the other hand, if the high-order scheme contains some background dissipation and its leading truncation error is of a diffusive nature, then it is worthwhile to minimize the amount of numerical diffusion.

In practice, the elimination of negative off-diagonal entries is performed step-by-step without assembling the global matrix D. Instead, artificial diffusion can be built into the operator K in a loop over edges of the sparsity graph, see Section 2.1.8. By definition, each edge is a pair of nodes $\{i, j\}$ that corresponds to a pair of nonzero off-diagonal coefficients k_{ij} and k_{ji}. The required solution update is as follows

$$\begin{aligned} k_{ii} &:= k_{ii} - d_{ij}, \qquad k_{ij} := k_{ij} + d_{ij}, \\ k_{ji} &:= k_{ji} + d_{ij}, \qquad k_{jj} := k_{jj} - d_{ij}. \end{aligned} \tag{4.13}$$

Without loss of generality, the edges of the sparsity graph are oriented so that

$$k_{ij} \le k_{ji}. \tag{4.14}$$

This orientation convention implies that node i is located 'upwind' and corresponds to the row number of the negative off-diagonal entry to be eliminated (if any).

Equation (4.12) implies that the diagonal-entries of $L = K + D$ are given by

$$l_{ii} := k_{ii} - \sum_{j \neq i} d_{ij}. \tag{4.15}$$

Furthermore, it can readily be seen that the row sums of K and $L = K + D$ are equal

$$\sum_j l_{ij} = \sum_j (k_{ij} + d_{ij}) = \sum_j k_{ij}.$$

The ordinary differential equation for the nodal value u_i can be represented as

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} l_{ij}(u_j - u_i) + u_i \sum_j k_{ij}. \tag{4.16}$$

If K has zero row sums, then the low-order scheme is local extremum diminishing

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \neq i} l_{ij}(u_j - u_i), \qquad l_{ij} \ge 0, \quad \forall j \neq i.$$

Otherwise, the second term in the right-hand side of (4.16) represents a nonvanishing discrete counterpart of $-u\nabla \cdot \mathbf{v}$ which is responsible for compressibility effects. As in the continuous case, convective transport by a nonuniform velocity field may

concentrate the mass in certain regions or create zones of low concentration. Then the low-order discretization (4.16) is no longer LED but still positivity-preserving.

Example 4.1. To clarify the implications of (4.13), consider the 1D model problem

$$\frac{\partial u}{\partial t}+v\frac{\partial u}{\partial x}=0, \tag{4.17}$$

where the velocity v is constant and strictly positive. The computational domain is $\Omega=(0,1)$ and a Dirichlet boundary condition $u(0)=g$ is prescribed at the inlet.

On a uniform mesh of linear finite elements, the standard Galerkin method yields

$$K=\frac{1}{2}\begin{bmatrix} \cdots & & & & & & \\ & v & 0 & -v & & & \\ & & v & 0 & -v & & \\ & & & v & 0 & -v & \\ & & & & & & \cdots \end{bmatrix}.$$

Note that this skew-symmetric tridiagonal matrix has zero row sums. The diagonal entries and column sums, except for the first and last one, are also equal to zero.

For any interior node, $m_i=\Delta x$, where Δx is is the constant mesh size. Hence, the lumped-mass version of (4.2) is equivalent to the central difference scheme

$$\frac{\mathrm{d}u_i}{\mathrm{d}t}+v\,\frac{u_{i+1}-u_{i-1}}{2\Delta x}=0.$$

Since $k_{ij}=-\frac{v}{2}$ for $j=i+1$, the artificial diffusion coefficient (4.11) is $d_{ij}=\frac{v}{2}$ and

$$L=\begin{bmatrix} \cdots & & & & & & \\ & v & -v & 0 & & & \\ & & v & -v & 0 & & \\ & & & v & -v & 0 & \\ & & & & & & \cdots \end{bmatrix},$$

which corresponds to the first-order accurate upwind difference approximation

$$\frac{\mathrm{d}u_i}{\mathrm{d}t}+v\,\frac{u_i-u_{i-1}}{\Delta x}=0.$$

Thus, the elimination of negative off-diagonal entries from a skew-symmetric operator K can be interpreted as *discrete upwinding* [200]. For any pair of nodes i and $j=i+1$ numbered in accordance with (4.14), the grid point x_i lies upstream of x_j.

After the discretization in time by the standard $\theta-$scheme, the upwind method is stable and positive under the CFL-like condition (3.92) which reduces to

$$v\frac{\Delta t}{\Delta x}\le\frac{1}{1-\theta},\qquad 0\le\theta<1. \tag{4.18}$$

Of course, there is no time step restriction for the backward Euler method ($\theta = 1$) which corresponds to 'upwinding in time' and is only first-order accurate.

4.1.3 Conservative Flux Decomposition

The replacement of the high-order discretization (4.6) by the low-order one (4.10) ensures positivity preservation but introduces a first-order perturbation error which manifests itself in strong smearing effects. The next step towards the construction of an algebraic flux correction scheme involves a decomposition of this error into internodal fluxes which can be used to restore high accuracy in regions where the solution is well resolved and no modifications of the original scheme are required.

By construction, the difference between the residuals of(4.6) and (4.10) is

$$f = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} - Du. \tag{4.19}$$

The zero row sum property of the artificial diffusion operator D implies that its contribution to the $i-$th component of the vector f can be written in the form

$$(Du)_i = \sum_j d_{ij}u_j = \sum_{j\neq i} d_{ij}(u_j - u_i) \tag{4.20}$$

and looks similar to the right-hand side of a local extremum diminishing scheme.

The error due to row-sum mass lumping can be decomposed in a similar way

$$(M_C u - M_L u)_i = \sum_j m_{ij}u_j - m_i u_i = \sum_{j\neq i} m_{ij}(u_j - u_i). \tag{4.21}$$

Due to (4.20)–(4.21) and symmetry, the total error (4.19) induced by mass lumping and artificial diffusion admits a conservative decomposition into internodal fluxes

$$f_i = \sum_{j\neq i} f_{ij}, \qquad f_{ji} = -f_{ij}. \tag{4.22}$$

The amount of mass transported by the *raw antidiffusive flux* f_{ij} is given by

$$f_{ij} = \left[m_{ij}\frac{\mathrm{d}}{\mathrm{d}t} + d_{ij}\right](u_i - u_j), \qquad \forall j \neq i. \tag{4.23}$$

Every pair of fluxes f_{ij} and f_{ji} can be associated with an edge of the sparsity graph which represents a pair of nonzero off-diagonal entries with indices i and j.

Remark 4.5. After the discretization in time, the derivative with respect to t is replaced by a finite difference. In steady-state problems, its contribution is zero.

Giving the flux f_{ij} to node i and $f_{ji} = -f_{ij}$ to its neighbor j does not create or destroy mass. The addition of raw antidiffusive fluxes (4.23) to the right-hand side of (4.16)

removes the error induced by the row-sum mass lumping and artificial diffusion

$$\begin{aligned} m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} &= \sum_{j\neq i} l_{ij}(u_j - u_i) + u_i \sum_j k_{ij} \\ &\quad - \sum_{j\neq i} d_{ij}(u_j - u_i) - \sum_{j\neq i} m_{ij}\left(\frac{\mathrm{d}u_i}{\mathrm{d}t} - \frac{\mathrm{d}u_j}{\mathrm{d}t}\right) \\ &= \sum_j k_{ij} u_j - \sum_{j\neq i} m_{ij}\left(\frac{\mathrm{d}u_i}{\mathrm{d}t} - \frac{\mathrm{d}u_j}{\mathrm{d}t}\right). \end{aligned}$$

Moving all time derivatives into the left-hand side, one obtains an equation of the form (4.2) which corresponds to the original high-order discretization (4.6).

4.1.4 Limited Antidiffusive Correction

Some of the raw antidiffusive fluxes f_{ij} are harmless but others are responsible for the violation of the positivity constraint by the high-order scheme. Such fluxes need to be canceled or limited so as to keep the scheme positivity-preserving. In the process of flux correction, every antidiffusive flux f_{ij} is multiplied by a solution-dependent correction factor $\alpha_{ij} \in [0,1]$ before it is inserted into the equation

$$m_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_j l_{ij} u_j + \bar{f}_i, \qquad \bar{f}_i = \sum_{j\neq i} \alpha_{ij} f_{ij}. \tag{4.24}$$

Of course, the fluxes f_{ij} and $f_{ji} = -f_{ij}$ must be limited using the same correction factor $\alpha_{ij} = \alpha_{ji}$ to maintain skew-symmetry and, hence, conservation of mass.

By construction, the high-order discretization (4.6) and its low-order counterpart (4.10) are recovered for $\alpha_{ij} = 1$ and $\alpha_{ij} = 0$, respectively. The former setting is usually acceptable in regions where the solution is smooth and well-resolved. However, the magnitude of antidiffusive fluxes may need to be reduced elsewhere, so as to prevent the formation of undershoots or overshoots. As a rule of thumb, the solution-dependent correction factors α_{ij} should be chosen as close to 1 as possible without violating the positivity constraint. Since discretization (4.24) is nonlinear in the choice of α_{ij}, it has the potential of being more than first-order accurate.

The flux-corrected semi-discrete scheme (4.24) can be written in the matrix form

$$\bar{M}_C \frac{\mathrm{d}u}{\mathrm{d}t} = \bar{K} u. \tag{4.25}$$

The coefficients of the partially lumped mass matrix $\bar{M}_C = \{\bar{m}_{ij}\}$ are given by

$$\bar{m}_{ii} := m_i - \sum_{j\neq i} \bar{m}_{ij}, \qquad \bar{m}_{ij} = \alpha_{ij} m_{ij}, \quad \forall j \neq i, \tag{4.26}$$

while the structure of the nonlinear transport operator $\bar{K} = \{\bar{k}_{ij}\}$ is as follows

$$\bar{k}_{ii} := l_{ii} + \sum_{j \neq i} \alpha_{ij} d_{ij}, \qquad \bar{k}_{ij} = l_{ij} - \alpha_{ij} d_{ij}, \quad \forall j \neq i. \tag{4.27}$$

Suppose that the solution to (4.25) satisfies an equivalent nonlinear ODE system

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = \bar{L}u, \tag{4.28}$$

where $\bar{L} = \{\bar{l}_{ij}\}$ has no negative off-diagonal coefficients and is defined so that

$$\bar{L}u = Lu + \bar{f}. \tag{4.29}$$

Such a space discretization is positivity-preserving and so is (4.25) because it was assumed to have the same solution. After the discretization in time, additional restrictions may apply to the time step for an explicit or semi-implicit algorithm.

To ensure the existence of representation (4.28) with $\bar{l}_{ij} \geq 0$ for all $j \neq i$, it is sufficient to find α_{ij} such that the sum of antidiffusive fluxes is constrained by

$$Q_i^- \leq \sum_{j \neq i} \alpha_{ij} f_{ij} \leq Q_i^+, \tag{4.30}$$

where $Q_i^{\pm}$ are local extremum diminishing upper and lower bounds of the form

$$Q_i^+ = \sum_{j \neq i} q_{ij} \max\{0, u_j - u_i\}, \tag{4.31}$$

$$Q_i^- = \sum_{j \neq i} q_{ij} \min\{0, u_j - u_i\}. \tag{4.32}$$

The coefficients q_{ij} must be nonnegative for all $j \neq i$. The best choice of these parameters depends on the problem at hand and on the limiting strategy (see below).

When appropriate values of q_{ij} have been fixed, it remains to define the correction factors $\alpha_{ij} = \alpha_{ji}$. It is always possible to satisfy (4.30) by setting $\alpha_{ij} = 0$ but a properly designed flux limiter returns $\alpha_{ij} \approx 1$ if the fluxes f_{ij} and f_{ji} are harmless.

Due to (4.30)–(4.32), there exists a matrix $\bar{Q} = \{\bar{q}_{ij}\}$ of nonlinear coefficients

$$\bar{q}_{ii} := -\sum_{j \neq i} \bar{q}_{ij}, \qquad 0 \leq \bar{q}_{ij} \leq q_{ij}, \quad \forall j \neq i \tag{4.33}$$

such that the sum of limited antidiffusive fluxes can be expressed in the LED form

$$\bar{f}_i = \sum_{j \neq i} \bar{q}_{ij}(u_j - u_i), \qquad \bar{q}_{ij} \geq 0, \quad \forall j \neq i. \tag{4.34}$$

In essence, this representation guarantees that the term $\bar{f}_i$ is equivalent to a sum of 'diffusive' and, therefore, acceptable edge contributions. Substitution into (4.29) yields $\bar{L} = L + \bar{Q}$, which proves positivity preservation at the semi-discrete level.

The beauty and generality of the above approach to flux limiting lie in the flexible choice of the parameters q_{ij} that define the upper and lower bounds. Any set of nonnegative bounded values ($0 \leq q_{ij} < \infty$, $\forall j \neq i$) is acceptable from the viewpoint of positivity preservation at the semi-discrete level. Hence, the definition of these parameters is dictated by accuracy and efficiency considerations. To get close enough to the high-order solution, the magnitude of q_{ij} should be sufficiently large. On the other hand, it cannot be chosen arbitrarily large for the following reasons:

- In explicit algorithms, the CFL-like positivity condition (3.92) for the largest admissible time step depends on the sum of q_{ij} and may become too restrictive.
- In implicit schemes and steady-state solvers, the nonlinear antidiffusive term is updated in an iterative way (see Section 4.2). Inordinately large values of q_{ij} may cause severe convergence problems and should be avoided, especially if intermediate solutions may fail to conserve mass and/or to stay positivity-preserving.

A number of ways to define q_{ij} will be discussed in Sections 4.3–4.5. As we will see, (quasi)-stationary and time-dependent transport problems may require different treatments. For the time being, let us keep the values of q_{ij} unspecified and present a general approach to the practical computation of the correction factors α_{ij}.

4.1.5 The Generic Limiting Strategy

In the process of flux correction, each node i may receive both positive and negative antidiffusive fluxes from its neighbors. Although some fluxes may cancel out, the formula for α_{ij} should be failsafe even in the worst-case scenario. Given a vector of nodal values u and a set of nonnegative coefficients q_{ij}, the sum of positive fluxes is required to be smaller than the upper bound (4.31), while the sum of negative ones may not fall below the lower bound (4.32). Hence, the admissible portion of a raw antidiffusive flux depends on its sign [355]. In Sections 4.3–4.4, we will consider algebraic flux correction schemes based on the following generic algorithm [192]

1. Compute the sums of positive and negative antidiffusive fluxes to be limited

$$P_i^+ = \sum_{j \neq i} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{j \neq i} \min\{0, f_{ij}\}. \tag{4.35}$$

2. Generate local extremum diminishing upper and lower bounds of the form

$$Q_i^+ = \sum_{j \neq i} q_{ij} \max\{0, u_j - u_i\}, \qquad Q_i^- = \sum_{j \neq i} q_{ij} \min\{0, u_j - u_i\}. \tag{4.36}$$

3. Evaluate the nodal correction factors for the positive and negative part

$$R_i^+ = \min\left\{1, \frac{Q_i^+}{P_i^+}\right\}, \qquad R_i^- = \min\left\{1, \frac{Q_i^-}{P_i^-}\right\}. \tag{4.37}$$

4. Perform flux limiting using edge-based correction factors α_{ij} such that

$$\alpha_{ij} \leq \begin{cases} R_i^+, & \text{if } f_{ij} > 0, \\ R_i^-, & \text{if } f_{ij} < 0, \end{cases} \qquad \alpha_{ji} = \alpha_{ij}. \tag{4.38}$$

Definition (4.38) guarantees that an upper bound of the form (4.30) holds for the sum of limited antidiffusive fluxes into node i. Indeed, it is easy to verify that

$$Q_i^- \leq R_i^- P_i^- \leq \sum_{j \neq i} \alpha_{ij} f_{ij} \leq R_i^+ P_i^+ \leq Q_i^+.$$

Remark 4.6. This multidimensional limiting strategy traces its origins to Zalesak's algorithm [355, 357]. Although classical (two-step) FCT methods do not fit into this framework, we adopt the same notation to highlight the existing similarities.

Remark 4.7. The upper/lower bounds $Q_i^\pm$ may consist of a single term associated with a local maximum $u_i^{\max}$ or minimum $u_i^{\min}$ taken over the stencil of node i

$$Q_i^+ = q_i^+ (u_i^{\max} - u_i), \qquad Q_i^- = q_i^- (u_i^{\min} - u_i). \tag{4.39}$$

The nonnegative coefficients $q_i^\pm$ can be defined as the sums of $q_{ij} \geq 0$ for all $j \neq i$.

It remains to give a formal definition of $\alpha_{ij} \leq R_i^\pm$ and $j \neq i$. This definition depends on whether an upwind-biased or a symmetric limiting strategy is capable of producing larger values of α_{ij}. Here and below, the terms 'upwind' and 'downwind' refer to the order of nodes i and j under convention (4.14) on edge orientation.

- **Symmetric flux limiters** do not distinguish between upwind and downwind nodes, treating them equally. For each pair of nodes $\{i, j\}$, the raw antidiffusive flux f_{ij} is added to the sum $P_i^\pm$ and subtracted from the sum $P_j^\mp$. Flux correction is performed using the minimum of nodal correction factors for i and j

$$\alpha_{ij} = \min\{R_i^\pm, R_j^\mp\}, \qquad \alpha_{ji} = \alpha_{ij}.$$

- **Upwind-biased flux limiters** take advantage of the fact that (a large portion of) the raw antidiffusive flux f_{ji} into a downwind node j is compensated by the diffusive edge contribution $l_{ji}(u_i - u_j)$, where $l_{ji} > 0$. In this approach, only the upwind sum $P_i^\pm$ is incremented, and the correction factors are defined as

$$\alpha_{ij} = R_i^\pm, \qquad \alpha_{ji} := \alpha_{ij}.$$

- **General-purpose flux limiters** compare the magnitude of the raw antidiffusive flux f_{ji} to that of the diffusive edge contribution $l_{ji}(u_i - u_j)$. On the basis of this comparison, an upwind-biased or a symmetric limiter is invoked, cf. [192].

Remark 4.8. Instead of constraining the sums of antidiffusive fluxes, it is possible to adjust the local slopes $u_i - u_j$ in a stand-alone fashion, as in the 1D case. Then the quantities P_i^+, bounds $Q_i^\pm$, and correction factors $R_i^\pm$ are defined edge-by-edge.

4.1.6 Summary of Algorithmic Steps

Let us summarize the basic steps involved in the derivation of an algebraic flux correction scheme. The starting point was a linear high-order discretization

$$M_C \frac{\mathrm{d}u}{\mathrm{d}t} = Ku, \qquad \exists\, j \neq i: \quad m_{ij} \neq 0, \quad k_{ij} < 0. \tag{4.40}$$

To achieve the desired matrix properties, we performed row-sum mass lumping and applied an artificial diffusion operator D designed so as to eliminate all negative off-diagonal coefficients of K. These manipulations have led us to a low-order counterpart of (4.40) which is the least diffusive among linear positivity-preserving ones

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu, \qquad L = K + D, \qquad l_{ij} \geq 0, \quad \forall j \neq i. \tag{4.41}$$

Finally, we removed excessive artificial diffusion using a set of internodal fluxes f_{ij} multiplied by solution-dependent correction factors $\alpha_{ij} \in [0,1]$. The end product is a nonlinear blend of (4.40) and (4.41) with solution-dependent matrix entries

$$\bar{M}_C \frac{\mathrm{d}u}{\mathrm{d}t} = \bar{K}u, \qquad \exists\, j \neq i: \quad \bar{m}_{ij} \neq 0, \quad \bar{k}_{ij} < 0. \tag{4.42}$$

Note that the limited antidiffusive correction reintroduces some off-diagonal entries of wrong sign. However, there exists a matrix $\bar{Q}$ of coefficients given by (4.33) such that problem (4.42) has the same solution as the equivalent nonlinear system

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = \bar{L}u, \qquad \bar{L} = L + \bar{Q}, \qquad \bar{l}_{ij} \geq 0, \quad \forall j \neq i. \tag{4.43}$$

In Sections 4.3–4.5, we will present a number of multidimensional flux limiters which guarantee the existence of the matrix $\bar{L}$ without constructing it explicitly.

The link between representations (4.42) and (4.43) is given by the formula

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = Lu + \bar{f}, \qquad \bar{f}_i = \sum_{j \neq i} \alpha_{ij} f_{ij} = \sum_{j \neq i} \bar{q}_{ij}(u_j - u_i). \tag{4.44}$$

Conservation and positivity of the flux-corrected scheme follow from the fact that

$$\alpha_{ji} = \alpha_{ij}, \qquad f_{ji} = -f_{ij}, \qquad \bar{q}_{ij} \geq 0, \qquad \forall j \neq i. \tag{4.45}$$

Up to now, we have explored the principles of algebraic flux correction in a rather abstract setting, so as develop a general framework in which to work. An in-depth presentation of some upwind-biased and symmetric flux limiters will follow in Sections 4.3–4.5. The main objective of this chapter is not to promote any particular algorithm but to retrace the steps involved in the design process and explain their ramifications. It is hoped that this background information will enable the interested reader to develop new high-resolution schemes building on similar concepts.

4.2 Solution of Nonlinear Systems

The implementation of flux correction in an unstructured mesh code requires further considerations regarding the choice of time-stepping schemes and/or iterative solution methods. In explicit algorithms, the limited antidiffusive term resides in the right-hand side and can be readily evaluated using the solution from the previous time step. The implementation of such a scheme is rather straightforward and will not be discussed here. An implicit time discretization makes it possible to operate with larger time steps but the correction factors α_{ij} must be calculated in an iterative way, even in the case of a linear transport equation. The repeated solution of linear subproblems followed by an update of α_{ij} is likely to pay off only if robust and efficient solution algorithms are available. In this section, we address the numerical treatment of nonlinear algebraic systems in implicit flux correction schemes.

4.2.1 Successive Approximations

After the time discretization by an implicit $\theta-$scheme, equation (4.44) becomes

$$A^{n+1}u^{n+1} = B^n u^n + \Delta t\, \bar{f}(u^{n+1}, u^n). \tag{4.46}$$

The matrices A^{n+1} and B^n represent the contribution of the low-order part. The entries of these matrices depend on the time step Δt and on the parameter $\theta \in (0,1]$

$$A^{n+1} = M_L - \theta \Delta t L^{n+1}, \tag{4.47}$$
$$B^n = M_L + (1-\theta)\Delta t L^n. \tag{4.48}$$

The settings $\theta = \frac{1}{2}$ and $\theta = 1$ correspond to the Crank-Nicolson and backward Euler methods, respectively. The latter is first-order accurate but offers unconditional stability and positivity preservation. The former is second-order accurate and unconditionally stable but positive only under a CFL-like condition of the form (3.92).

In the case of a nonlinear governing equation and/or a nonstationary velocity field, the entries of A^{n+1} and B^n need to be updated as the solution and time evolve. Furthermore, the antidiffusive term $\bar{f}(u^{n+1}, u^n)$ depends on u^{n+1} in a nonlinear way, so an iterative approach to the solution of the algebraic system (4.46) is required.

Consider a sequence of successive approximations $\{u^{(m)}\}$ to $u = u^{n+1}$. At the beginning of the first outer iteration, the value of $u^{(0)}$ is guessed making use of the previously computed data and initial/boundary conditions. A reasonable initial guess for an unsteady transport problem is $u^{(0)} = u^n$ or $u^{(0)} = 2u^n - u^{n-1}$. These settings correspond to the constant and linear extrapolation in time, respectively.

Given an approximation $u^{(m)}$, its successor $u^{(m+1)}$ can be calculated as follows

$$A^{(m)}u^{(m+1)} = B^n u^n + \Delta t\, \bar{f}(u^{(m)}, u^n), \qquad m = 0, 1, \ldots \tag{4.49}$$

The sum of limited antidiffusive fluxes remains in the right-hand side and is evaluated using the current iterate $u^{(m)}$ in place of u^{n+1}. If the coefficients of the matrix $A^{(m)} = M_L - \theta\Delta t L^{(m)}$ depend on the solution, they also needs to be recalculated. This straightforward solution procedure is referred to as *fixed-point iteration*, also known as the Picard iteration and successive approximation (substitution).

A convenient way to enforce Dirichlet boundary conditions for a given node i is to replace the corresponding row of $A^{(m)}$ by that of the identity matrix

$$a_{ii} := 1, \qquad a_{ij} := 0, \qquad \forall j \neq i \tag{4.50}$$

and substitute the boundary value g_i for the i-th element of the right-hand side [322].

In most cases, linear systems (4.49) are also solved iteratively. Since the low-order operator $A^{(m)}$ was designed to meet the requirements of Corollary 3.12, it proves to be an M-matrix. This favorable property ensures that a small number of inner iterations are typically sufficient for practical purposes. The number of outer iteration cycles (4.49) depends on the evolution of the residual (alias *defect*)

$$r^{(m)} = B^n u^n - A^{(m)} u^{(m)} + \Delta t\, \overline{f}(u^{(m)}, u^n). \tag{4.51}$$

All elements of $r^{(m)}$ associated with Dirichlet boundary nodes should be set to zero.

The Euclidean (or maximum) norm $||r^{(m)}||$ of the residual is a good indicator of how close the current approximation $u^{(m)}$ is to the final solution u^{n+1}. A typical set of stopping criteria for an iterative solution procedure like (4.49) is as follows

$$||r^{(m+1)}|| \leq \min\{\varepsilon_1, \varepsilon_2 ||r^{(0)}||\}, \qquad \frac{||u^{(m+1)} - u^{(m)}||}{||u^{(m+1)}||} < \varepsilon_3, \tag{4.52}$$

where ε_1, ε_2, and ε_3 are the tolerances prescribed for defects and relative changes.

Suppose that M outer iterations are required to meet the above stopping criteria and obtain $u^{n+1} := u^{(M)}$. Then the effective *rate of convergence* can be defined as

$$\rho_M = \left(\frac{||r^{(M)}||}{||r^{(0)}||} \right)^{\frac{1}{M}}.$$

In essence, this is the average factor by which the magnitude of the defect shrinks during a single outer iteration. Note that $\rho_M < 1$ is required for convergence.

The above methodology is also suitable for computing stationary solutions with

$$M_L = 0, \quad \theta = 1, \quad \Delta t = 1.$$

Alternatively, the solution can be marched to the steady state by the unconditionally positive backward Euler method with large and, possibly, variable pseudo-time steps. The relative solution changes and/or convergence rates can be used to determine the optimal time step size adaptively. When the solution begins to approach the steady state, the removal of the mass matrix can greatly speed up convergence, while removing it too soon can have the opposite effect [307].

In steady-state computations, it is advisable to switch off the nonlinear antidiffusive term $\underline{f}$ at the startup stage since the low-order solution is usually inexpensive to compute and closer to the flux-corrected one than an arbitrary initial guess $u^{(0)}$.

4.2.2 Defect Correction Schemes

Fixed-point iteration is one of the simplest techniques for solving nonlinear algebraic equations. In many cases, the computational cost can be drastically reduced by using another iterative solver tailored to the properties of the continuous problem and/or of the employed discretization techniques. In addition, special measures may need to be taken to secure convergence to a steady state or the ability to operate with large time steps. Hence, it is worthwhile to consider a general class of iterative solution techniques to which the basic fixed-point iteration (4.49) belongs.

Many iterative solution methods for system (4.46) can be formally written in the following generic form which will be referred to as the *defect correction scheme*

$$u^{(m+1)} = u^{(m)} + [\overline{A}^{(m)}]^{-1} r^{(m)}, \qquad m = 0, 1, 2 \ldots, \tag{4.53}$$

where $\overline{A}^{(m)}$ is a suitable 'preconditioner' and $r^{(m)}$ is the residual given by (4.51). As before, the Dirichlet boundary conditions for node i are implemented by setting

$$\begin{aligned} &\overline{a}_{ii} := 1, \quad \overline{a}_{ij} := 0, \quad \forall j \neq i, \\ &u_i^{(m)} := g_i, \quad r_i^{(m)} := 0. \end{aligned}$$

In practice, the 'inversion' of $\overline{A}^{(m)}$ is performed by solving the linear subproblem

$$\overline{A}^{(m)} \Delta u^{(m+1)} = r^{(m)}, \qquad m = 0, 1, 2, \ldots \tag{4.54}$$

After a few inner iterations, the increment $\Delta u^{(m+1)}$ is applied to the last iterate

$$u^{(m+1)} = u^{(m)} + \Delta u^{(m+1)}. \tag{4.55}$$

The iteration process is terminated when the defect $r^{(m)}$ and the relative solution changes $\Delta u^{(m+1)}$ become sufficiently small in the sense of criteria (4.52).

The implementation of a defect correction cycle involves the following tasks:

- Assembly of $\overline{A}^{(m)}$ and $r^{(m)}$, imposition of Dirichlet boundary conditions.
- Monitoring the residual norms, relative changes, and convergence rates.
- Implicit computation of the solution increments $\Delta u^{(m+1)}$ from (4.54).
- Explicit computation of the new approximation $u^{(m+1)}$ from (4.55).

Ideally, the preconditioner $\overline{A}^{(m)}$ should be designed so that (i) matrix assembly is relatively fast, (ii) linear subproblems (4.54) can be solved efficiently, and (iii) convergence is achieved with a small number of outer iterations. These requirements are often in conflict with one another, so some compromises need to be made.

If the time step Δt is very small, the approximate solution can be updated in a fully explicit fashion using (4.53) with a Jacobi-like diagonal preconditioner

$$\bar{A} = \mathrm{diag}\{m_i - \theta \Delta t\, l_{ii}\}. \tag{4.56}$$

The number of outer iterations for (4.53) preconditioned in this way can be as small as 1 if a good initial guess is available. Thus, the computational cost per time step might be comparable to that for a conditionally stable explicit algorithm.

At intermediate and large time steps, the default preconditioner for (4.53) is

$$\bar{A} = M_L - \theta \Delta t L. \tag{4.57}$$

Substitution of (4.51) and (4.57) into (4.53) reveals that the resulting defect correction scheme is equivalent to the standard fixed-point iteration given by (4.49).

Furthermore, particularly severe nonlinearities can be handled using Newton-like methods in which $\bar{A}$ is a suitable approximation to the Jacobian matrix

$$\bar{A} \approx -\left\{ \frac{\partial r_i}{\partial u_j} \right\}. \tag{4.58}$$

This definition requires (numerical) differentiation of the residual $r(u)$ with respect to each element of the solution vector u. In the context of algebraic flux correction, the matrix $J(u)$ can be approximated by divided differences and assembled edge-by-edge, as proposed by Möller [254]. If properly configured, the resulting discrete Newton method converges much faster than the defect correction scheme (4.53) preconditioned by (4.57). For implementation details we refer to [254, 255, 258].

The optimal choice of iterative solvers for sparse linear systems (4.54) with a nondiagonal and nonsymmetric matrix $\bar{A}^{(m)}$ also depends on the time step size and on the matrix properties. As long as Δt is relatively small, the preconditioner is dominated by the lumped mass matrix and a basic iteration of Jacobi or Gauß-Seidel (SOR) type may suffice. In this case, each inner iteration has approximately the same cost as one step of an explicit algorithm. As the time step and the condition number increase, BiCGSTAB, GMRES, and multigrid methods are to be preferred. The Gauß-Seidel iteration and ILU with Cuthill-McKee renumbering can serve as smoothers/preconditioners for intermediate and large time steps, respectively. Due to the M-matrix property of the low-order operator (4.57), its ILU factorization exists and is unique [249]. It can be used as a preconditioner for inner iterations even if the actual matrix to be 'inverted' is the Jacobian (4.58) for Newton's method [255].

4.2.3 Underrelaxation and Smoothing

In many cases, it is desirable to reduce the changes between two successive iterates so as to stabilize the solution and make it converge 'slowly but surely.' This can be accomplished by limiting the increments $\Delta u^{(m+1)}$ computed in (4.54) before

applying them to the old iterate $u^{(m)}$. To this end, formula (4.55) is replaced by

$$u_i^{(m+1)} = u_i^{(m)} + \omega_i^{(m)} \Delta u_i^{(m+1)}, \tag{4.59}$$

where $0 < \omega_i^{(m)} \leq 1$ for all i. This is a classical example of *explicit underrelaxation* which amounts to combining the latest and previous nodal values [104, 106, 268]. The relaxation is said to be 'heavy' if a larger weight is given to the latter, that is, if $\omega_i^{(m)} < 0.5$. In other cases, 'light' underrelaxation with $\omega_i^{(m)} \geq 0.5$ is appropriate.

Unfortunately, there are no general rules for the choice of relaxation factors. They can be assigned a fixed value (e.g., $\omega = 0.8$) or chosen adaptively so as to control the evolution of residuals [173] or minimize the error in an appropriate norm [322]. Typically, a range of admissible values is specified for a variable relaxation factor

$$0 < \omega^{\min} \leq \omega_i^{(m)} \leq \omega^{\max} \leq 1.$$

Alternatively, *implicit underrelaxation* can be performed to make a given preconditioner $\bar{A}^{(m)}$ more diagonally dominant [104, 268]. For example, let $\gamma_i^{(m)} \geq 1$ and

$$\bar{a}_{ii}^{(m)} := \gamma_i^{(m)} \bar{a}_{ii}^{(m)}. \tag{4.60}$$

The same effect can be achieved by means of 'relaxation through inertia' [268], whereby a nonnegative number $\sigma_i^{(m)} \geq 0$ is added to each diagonal entry

$$\bar{a}_{ii}^{(m)} := \bar{a}_{ii}^{(m)} + \sigma_i^{(m)}. \tag{4.61}$$

Obviously, the additive version is related to the multiplicative one by the formula

$$\sigma_i^{(m)} = (\gamma_i^{(m)} - 1) a_{ii}^{(m)}.$$

In steady-state computations, implicit underrelaxation is equivalent to the use of variable pseudo-time steps [104, 268]. Conversely, local time-stepping can be interpreted as underrelaxation. The optimal values of $\gamma_i^{(m)}$ and $\sigma_i^{(m)}$ are problem-dependent and may change in the course of simulation. It is advisable to perform stronger underrelaxation at the startup stage and gradually adjust relaxation factors so as to control the residuals, relative solution changes, and convergence rates.

Another way to accelerate the defect correction scheme is known as *residual smoothing*. If the residual $r^{(m)}$ exhibits an oscillatory behavior that hampers convergence, it is worthwhile to replace it by a smooth approximation $\bar{r}^{(m)}$. For instance, the latter can be constructed by introducing some implicit mass diffusion [301]

$$m_i \bar{r}_i^{(m)} + \sum_{j \neq i} \omega_{ij} m_{ij} (\bar{r}_i^{(m)} - \bar{r}_j^{(m)}) = m_i r_i^{(m)}, \tag{4.62}$$

where ω_{ij} is a positive weight and $m_{ij} \geq 0$ is an entry of the consistent mass matrix.

In practice, $\bar{r}^{(m)}$ is obtained with a few sweeps of the Jacobi iteration [301]

$$\left(m_i + \sum_{j \neq i} \omega_{ij} m_{ij}\right) \bar{r}_i^{(m,l+1)} = m_i r_i^{(m)} + \sum_{j \neq i} \omega_{ij} m_{ij} \bar{r}_j^{(m,l)}.$$

Remark 4.9. A converged solution does not depend on *how* it was computed. However, underrelaxation and residual smoothing may render intermediate results nonconservative. To avoid stopping outer iterations too soon, it is worthwhile to check the total mass if the tolerances for residuals and relative changes are rather slack.

4.2.4 Positivity-Preserving Solvers

The use of limiters in unstructured grid methods for steady-state problems is frequently associated with severe convergence problems [342]. Sometimes the nonlinearity is so strong that even heavy underrelaxation and/or residual smoothing are of little help. The residuals decrease steadily until the initial error is reduced by several orders of magnitude, after which convergence stalls. Although the discrete maximum principle would hold for the fully converged solution, the linearization inherent to an iterative solver may give rise to undershoots/overshoots. Conversely, the oscillatory solution behavior may be responsible for the lack of convergence. Therefore, it is worthwhile to start with a monotone low-order solution and configure the defect correction scheme so as to ensure that each step is positivity-preserving.

At steady state, the residual of a scalar transport equation discretized by an algebraic flux correction scheme reduces to (4.29) which corresponds to

$$r^{(m)} = L^{(m)} u^{(m)} + \bar{Q}^{(m)} u^{(m)} = L^{(m)} u^{(m)} + \bar{f}^{(m)}, \tag{4.63}$$

where $\bar{Q} = \{\bar{q}_{ij}\}$ is a matrix with zero row sums and nonnegative off-diagonal entries given by (4.33). By definition, the antidiffusive term can be written as

$$\bar{f}_i = \sum_{j \neq i} \alpha_{ij} d_{ij} (u_i - u_j) = \sum_{j \neq i} \bar{q}_{ij} (u_j - u_i). \tag{4.64}$$

After the imposition of Dirichlet boundary conditions, the nodes can be numbered so as to cast the defect correction scheme (4.53) into the partitioned form

$$\begin{bmatrix} u_\Omega^{(m+1)} \\ u_\Gamma^{(m+1)} \end{bmatrix} = \begin{bmatrix} u_\Omega^{(m)} \\ u_\Gamma^{(m)} \end{bmatrix} + \begin{bmatrix} \bar{A}_{\Omega\Omega}^{(m)} & \bar{A}_{\Omega\Gamma}^{(m)} \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} r_\Omega^{(m)} \\ 0 \end{bmatrix}. \tag{4.65}$$

As in Chapter 3, the subscripts Ω and Γ refer to row numbers associated with the vectors of unknowns u_Ω and Dirichlet boundary values $u_\Gamma = g$, respectively.

Again, the iterative solution procedure involves the evaluation of residuals, solution of linear systems, and updating the solution step-by-step. The choice of the preconditioner $\bar{A}^{(m)}$ defines a splitting of the residual (4.63) into a part that varies

linearly with $u^{(m+1)}$ and the remainder $b^{(m)}$ which depends on $u^{(m)}$ only. That is, each defect correction cycle (4.65) represents a linearized problem of the form

$$\begin{bmatrix} \bar{A}^{(m)}_{\Omega\Omega} & \bar{A}^{(m)}_{\Omega\Gamma} \\ 0 & I \end{bmatrix} \begin{bmatrix} u^{(m+1)}_{\Omega} \\ u^{(m+1)}_{\Gamma} \end{bmatrix} = \begin{bmatrix} b^{(m)}_{\Omega} \\ g \end{bmatrix}. \tag{4.66}$$

Theorem 3.16 states that the resulting solution is positivity-preserving if the left-hand side matrix is monotone and the corresponding b_Ω is nonnegative for $g \geq 0$.

Mild nonlinearities incurred by rather diffusive flux limiters can be handled with the stationary counterpart of (4.57). This linearization leads to (4.66) with

$$\bar{A} = -L, \qquad b = \bar{f}. \tag{4.67}$$

By definition, the matrix in the left-hand side of the linearized system (4.66) is monotone but there is no guarantee that $b(u) \geq 0$ if $u \geq 0$. Hence, convergence is required for positivity preservation and, sometimes, vice versa. To give an example of a defect correction scheme that maintains positivity in each step, consider

$$\bar{A} = -(L + \bar{Q}), \qquad b = 0. \tag{4.68}$$

In this case, the monotonicity of $\bar{A}$ is sufficient to keep all solutions to (4.66) nonnegative given $g \geq 0$. However, only the fully converged solution is guaranteed to conserve mass. This is in contrast to the usual situation in which every update is conservative but intermediate solutions may assume nonphysical negative values.

Remark 4.10. In the context of pseudo-time-stepping with implicit TVD schemes, splitting (4.68) corresponds to the *linearized nonconservative implicit* (LNI) form [353]. The preconditioner $\bar{A}$ is defined in terms of the coefficients involved in the proof of Harten's theorem [137]. Aspects of strict positivity preservation in linearized TVD approximations are also investigated in [178].

If a generic limiter of the form (4.35)–(4.38) is employed, the following strategy can be used to determine the coefficients (4.33) of the nonlinear operator Q

1. Compute the limited sums $\bar{P}_i^{\pm}$ of positive and negative antidiffusive fluxes

$$\bar{P}_i^{+} = \sum_{j \neq i} \max\{0, \alpha_{ij} f_{ij}\}, \qquad \bar{P}_i^{-} = \sum_{j \neq i} \min\{0, \alpha_{ij} f_{ij}\}. \tag{4.69}$$

2. Retrieve the local extremum diminishing upper and lower bounds (4.36)

$$Q_i^{+} = \sum_{j \neq i} q_{ij} \max\{0, u_j - u_i\}, \qquad Q_i^{-} = \sum_{j \neq i} q_{ij} \min\{0, u_j - u_i\}. \tag{4.70}$$

3. Assemble $\bar{Q} = \{\bar{q}_{ij}\}$ using (4.33) with off-diagonal coefficients $\bar{q}_{ij}$ given by

$$\bar{R}_i^{+} = \frac{\bar{P}_i^{+}}{Q_i^{+}}, \qquad \bar{R}_i^{-} = \frac{\bar{P}_i^{-}}{Q_i^{-}}, \qquad \bar{q}_{ij} = \begin{cases} \bar{R}_i^{+} q_{ij}, & \text{if } u_j > u_i, \\ \bar{R}_i^{-} q_{ij}, & \text{if } u_j < u_i. \end{cases} \tag{4.71}$$

Remark 4.11. The correction factors applied to q_{ij} are q_{ji} are generally not equal.

Preconditioners of LNI type are to be recommended for steady state computations in which the use of (4.57) would inhibit convergence or require taking impractically small time steps. For linear transport equations, the need to recompute the coefficients $\bar{q}_{ij}$ and update $\bar{A}^{(m)}$ after each flux/defect correction step makes matrix assembly more expensive than that for (4.57). However, the differences are not so pronounced in the case of nonlinear real-life problems since even the discrete transport operator of low order needs to be reassembled after each outer iteration.

In fact, the defect correction scheme (4.65) with (4.68) is not the only and not the cheapest way to keep intermediate solutions nonoscillatory. Another representative of such schemes can be constructed using *negative slope linearization* (cf. the fourth basic rule in Section 1.6.3) in the LED form (4.64) of the antidiffusive term

$$\bar{f}_i^{(m+1)} = \sum_{j\neq i} \bar{q}_{ij}^{(m)} u_j^{(m)} - \sum_{j\neq i} \bar{q}_{ij}^{(m)} u_i^{(m+1)}. \tag{4.72}$$

The defect correction scheme (4.66) proves positivity-preserving if we take [204]

$$\bar{A} = -L + \mathrm{diag}\{\bar{\sigma}_i\}, \qquad \bar{\sigma}_i = \sum_{j\neq i} \bar{q}_{ij} \tag{4.73}$$

such that the i-th component of the right-hand side vector b_Ω is given by

$$b_i = \sum_{j\neq i} \bar{q}_{ij} u_j.$$

Remark 4.12. This definition of the preconditioner $\bar{A}$ can be interpreted as adaptive relaxation through inertia (4.61) as applied to the low-order operator $\bar{A} = -L$.

The values of $\bar{q}_{ij}$ depend on the correction factors $\bar{R}_i^{\pm}$ which need to be recalculated at each outer iteration using (4.69)–(4.71). Alternatively, the inertia term σ_i can be defined in terms of the uncorrected coefficients q_{ij}. The preconditioner

$$\bar{A} = -L + \mathrm{diag}\{\sigma_i\}, \qquad \sigma_i = \sum_{j\neq i} q_{ij} \tag{4.74}$$

differs from (4.73) in the value of the inertia terms σ_i. The right-hand side becomes

$$b_i = \sum_{j\neq i} \bar{q}_{ij} u_j + \sum_{j\neq i} (q_{ij} - \bar{q}_{ij}) u_i = \bar{f} + \sigma_i u_i.$$

Positivity preservation follows from the fact that $0 \leq \bar{q}_{ij} \leq q_{ij}$ for all $j \neq i$. In a practical implementation, the right-hand side vector b_Ω is assembled as follows

$$b_i = \sum_{j\neq i} \alpha_{ij} f_{ij} + \sigma_i u_i, \qquad f_{ij} = d_{ij}(u_i - u_j), \quad \forall j \neq i.$$

In contrast to solvers based on (4.68) and (4.73), there is no need to compute $\bar{q}_{ij}$.

Remarkably, positivity is maintained if negative off-diagonal entries of the preconditioner $\bar{A}$ are set to zero. For example, if an M-matrix $A = \{\bar{a}_{ij}\}$ is replaced by $\bar{A} := \mathrm{diag}\{\bar{a}_{ii}\}$, the corresponding modification of the right-hand side is

$$b_i := b_i - \sum_{j \neq i} \bar{a}_{ij} u_j, \qquad \bar{a}_{ij} \leq 0, \quad j \neq i.$$

Hence, the replacement of $\bar{A}$, as defined in (4.68), (4.73), or (4.73), by its diagonal or triangular part does not destroy positivity but convergence will slow down.

Remark 4.13. A slowly converging defect correction scheme can be accelerated within the framework of a nonlinear full approximation storage / full multigrid (FAS-FMG) solution strategy. In this case, a basic iteration of the form (4.65) with a diagonal or upper/lower triangular preconditioner $\bar{A}$ can serve as a smoother.

4.2.5 Accuracy vs. Convergence

In our experience, there is a tradeoff between the accuracy of the flux limiting procedure and convergence of an iterative defect correction scheme. Any enhancement of the flux limiter that makes it possible to accept more antidiffusion is likely to have an adverse effect on the nonlinear convergence rates. Conversely, more diffusive schemes converge better but the results are less accurate. In many cases, it is worthwhile to sacrifice some accuracy if this would make computations much faster.

In other situations, the use of the least diffusive flux limiters is desirable and more work needs to be invested in the development of a robust iterative solver. The right constellation of parameter settings for inner and outer iterations is almost certain to exist. Therefore, there is often no need to give up or use *ad hoc* tricks (like 'freezing' the correction factors [239, 342]) when convergence problems are encountered.

4.3 Steady Transport Problems

The term *algebraic flux correction* was introduced in [200] as a common name for high-resolution schemes based on node-oriented flux limiters of FCT and TVD type. The striking similarities between the two flux limiting techniques were exploited in [192], where algorithm (4.35)–(4.38) was first presented. The differences between algebraic FCT and TVD schemes were also analyzed and explained. The former approach is readily applicable to finite element discretizations with a consistent mass matrix. The amount of admissible antidiffusion is inversely proportional to the time step, which makes the imposed constraints less restrictive as the time step is refined. This is a desirable feature if the problem at hand is unsteady, and a small time step is required to capture the evolution details. However, the use of large time steps results in a loss of accuracy, which compromises the advantages of unconditionally

stable implicit algorithms. Moreover, severe convergence problems are observed in the steady-state limit and stationary solutions depend on the pseudo-time step.

On the other hand, algebraic flux correction of TVD type is independent of the time step and lends itself to the treatment of stationary transport problems. Of course, it can also be employed in transient computations but the use of small time steps has no direct influence on the correction factors and, therefore, does not lead to a marked improvement. Moreover, the need for mass lumping makes the solutions less accurate than those produced by a consistent-mass FEM-FCT scheme.

Upwind-biased TVD limiters can be integrated into unstructured grid codes and applied edge-by-edge [8, 239] or node-by-node [206], so as to control the slope ratio for a local 3-point stencil or the net antidiffusive flux, respectively. In either case, the resulting scheme proves local extremum diminishing (LED) but, strictly speaking, the use of standard limiter functions like *minmod* or *superbee* does not guarantee that a second-order accurate approximation is recovered in regions of smoothness. Indeed, the raw antidiffusive flux for an algebraic flux correction scheme is uniquely defined by (4.23) rather than by an arbitrary combination of Lax-Wendroff (central difference) and Beam-Warming (second-order upwind) fluxes. Thus, straightforward extensions of classical TVD schemes may exhibit unexpected behavior when applied to multidimensional transport problems on strongly nonuniform meshes.

The above considerations have stimulated the search for alternatives to algebraic FEM-TVD schemes proposed in [206]. Several algorithms [192, 194, 204] were designed to constrain a given high-order discretization and revert to it in regions where no flux limiting is required. In the present section, we review the current state of the art and apply algebraic flux correction to steady convection-dominated transport equations. The design of FCT algorithms for transient problems and the numerical treatment of anisotropic diffusion are addressed in subsequent sections.

4.3.1 Upwind-Biased Flux Correction

An upwind-biased limiting strategy is to be recommended for simulation of stationary and weakly time-dependent transport processes at high Peclet numbers. Before embarking on the development of flux correction schemes for this class of problems, some implications of the edge orientation convention (4.14) need to be discussed.

If the off-diagonal entry k_{ij} is dominated by the skew-symmetric convective part $k'_{ij} = (k_{ij} - k_{ji})/2$ and the edge $\overrightarrow{ij}$ is oriented in accordance with (4.14) then

$$k_{ij} < 0 < k_{ji}, \qquad d_{ij} = -k_{ij} > 0, \qquad 0 = l_{ij} < l_{ji}. \tag{4.75}$$

The correction factor α_{ij} should ensure that bounds of the form (4.30) hold for a given set of parameters $q_{ij} \geq 0$. Instead of checking these bounds for both nodes, the antidiffusive flux received by the downwind node j can be absorbed into

$$\bar{k}_{ji}(u_i - u_j) = l_{ji}(u_i - u_j) - \alpha_{ij} f_{ij}, \tag{4.76}$$

where l_{ji} is positive and $0 \le \alpha_{ij} \le 1$. This edge contribution is of LED type if $\bar{k}_{ji} \ge 0$.

If the problem at hand is stationary or the time-dependent part of the raw antidiffusive flux f_{ij} can be neglected, then definition (4.23) reduces to

$$f_{ij} = d_{ij}(u_i - u_j), \qquad f_{ji} = -f_{ij}. \tag{4.77}$$

Substitution into (4.76) reveals that the flux-corrected coefficient $\bar{k}_{ji}$ is given by

$$\bar{k}_{ji} = l_{ji} - \alpha_{ij} d_{ij} = k_{ji} + (1 - \alpha_{ij}) d_{ij}$$

and proves nonnegative, except at critical points where the velocity changes its sign so that both off-diagonal entries of K are negative (a rather unusual situation).

To make sure that $\bar{k}_{ji} \ge 0$, an arbitrary antidiffusive flux f_{ij} can be replaced by

$$f_{ij} := minmod\{f_{ij}, l_{ji}(u_i - u_j)\}, \qquad f_{ji} := -f_{ij}. \tag{4.78}$$

The *minmod* function returns zero if its arguments do not have the same sign. Otherwise, the argument with the smallest magnitude is returned. That is,

$$minmod\{a, b, \ldots\} = \begin{cases} \min\{a, b, \ldots\}, & \text{if} \quad a > 0,\ b > 0, \ldots \\ \max\{a, b, \ldots\}, & \text{if} \quad a < 0,\ b < 0, \ldots \\ 0, & \text{otherwise.} \end{cases} \tag{4.79}$$

In many situations, minmod prelimiting (4.78) does not change the magnitude of the original antidiffusive flux f_{ij} or reduces it by a small amount. The remaining part can be handled using the upwind-biased version of algorithm (4.35)–(4.38).

The sums of positive and negative antidiffusive fluxes f_{ij} received by node i can be decomposed into the contributions of upwind ($k_{ij} > k_{ji}$) and downwind ($k_{ij} \le k_{ji}$) neighbors. In light of the above, only the latter part still needs to be limited. Let

$$P_i^+ = \sum_{k_{ij} \le k_{ji}} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{k_{ij} \le k_{ji}} \min\{0, f_{ij}\}. \tag{4.80}$$

To enforce the positivity constraint, a set of nonnegative coefficients q_{ij} is employed to define local extremum diminishing upper and lower bounds of the form

$$Q_i^+ = \sum_{j \ne i} q_{ij}(u_j - u_i), \qquad Q_i^- = \sum_{j \ne i} q_{ij}(u_j - u_i). \tag{4.81}$$

The flux f_{ij} is limited using the nodal correction factor for the upwind node, i.e.,

$$R_i^\pm = \min\left\{1, \frac{Q_i^\pm}{P_i^\pm}\right\}, \qquad \alpha_{ij} = \begin{cases} R_i^+, & \text{if } f_{ij} \ge 0, \\ R_i^-, & \text{if } f_{ij} < 0, \end{cases} \tag{4.82}$$

assuming that (4.14) holds. The flux f_{ji} is multiplied by the same correction factor

$$\alpha_{ji} := \alpha_{ij}, \qquad \forall j \ne i, \quad k_{ij} \le k_{ji}.$$

Obviously, the amount or raw antidiffusion that can be accepted by the above upwind-biased flux limiter depends on the definition of q_{ij} in (4.81). Choosing a value that is too large is as bad as choosing one that is too small. It is worthwhile to define the upper/lower bounds so that a classical TVD scheme is recovered for linear convection in 1D. As shown in [192], this requirement is satisfied for

$$q_{ij} := l_{ij} \geq 0, \qquad \forall j \neq i. \tag{4.83}$$

This definition guarantees that (i) limited antidiffusive fluxes have the same effect as the local extremum diminishing low-order part and (ii) their contribution to the residual of the flux-corrected scheme is of the same or smaller magnitude.

In the nontrivial case, $d_{ij} > 0$ and $0 = l_{ij} < l_{ji}$ due to (4.14). Hence, the upper and lower bounds $Q_i^{\pm}$ consist of upstream edge contributions ($k_{ij} > k_{ji}$) only. The contribution of the edge $\overrightarrow{ij}$ can be inserted into the sums $P_i^{\pm}$ and $Q_j^{\pm}$ as follows

$$\begin{aligned} P_i^+ &:= P_i^+ + \max\{0, f_{ij}\}, \qquad & Q_j^+ &:= Q_j^+ + \max\{0, l_{ji}(u_i - u_j)\}, \\ P_i^- &:= P_i^- + \min\{0, f_{ij}\}, \qquad & Q_j^- &:= Q_j^- + \min\{0, l_{ji}(u_i - u_j)\}. \end{aligned}$$

After the computation of the correction factors $\alpha_{ij} = R_i^{\pm}$ from (4.82), the limited antidiffusive fluxes are inserted into the global vector $\bar{f}$ that appears in (4.44)

$$\bar{f}_i := \bar{f}_i + \alpha_{ij} f_{ij}, \qquad \bar{f}_j := \bar{f}_j - \alpha_{ij} f_{ij}.$$

Thus, a typical implementation of algorithm (4.80)–(4.82) involves two loops over edges (one to assemble $P_i^{\pm}$ and $Q_i^{\pm}$, the other to perform flux limiting) and one loop over nodes. The latter is required to evaluate the nodal correction factors $R_i^{\pm}$.

Remark 4.14. If Dirichlet boundary conditions are imposed at node i, then the nodal value u_i is fixed. Therefore, there is no need to limit f_{ij} and we can set $R_i^{\pm} := 1$.

As an alternative to (4.83), the bounds $Q_i^{\pm}$ can be defined in terms of [194]

$$q_{ij} := d_{ij} \geq 0, \qquad \forall j \neq i. \tag{4.84}$$

This version is designed to ensure that (i) limited antidiffusive fluxes have the same effect as artificial diffusion built into the low-order part and (ii) their contribution to the residual of the flux-corrected scheme is of the same or smaller magnitude.

In a loop over edges, a raw antidiffusive flux of the form $f_{ij} = d_{ij}(u_i - u_j)$ is added to the upwind sum $P_i^{\pm}$ and to the upper/lower bounds for both nodes [194]

$$\begin{aligned} Q_i^+ &:= Q_i^+ + \max\{0, -f_{ij}\}, \qquad & Q_j^+ &:= Q_j^+ + \max\{0, f_{ij}\}, \\ Q_i^- &:= Q_i^- + \min\{0, -f_{ij}\}, \qquad & Q_j^- &:= Q_j^- + \min\{0, f_{ij}\}. \end{aligned}$$

The 1D version of this algorithm, as applied to the pure convection equation discretized by linear finite elements, corresponds to the *minmod* limiter which is more diffusive than the 1D counterpart of (4.83). In multidimensions, the differences are not so large, while the flux correction scheme based on (4.84) converges better.

In either case, the imposed constraints (4.81) can be made less restrictive by taking the largest/smallest possible value of the difference $u_j - u_i$, that is,

$$\delta u_i^{\max} := \max_j (u_j - u_i), \qquad \delta u_i^{\min} := \min_j (u_j - u_i). \tag{4.85}$$

The so-defined increments $\delta u_i^{\max}$ and $\delta u_i^{\min}$ can be initialized by zero and updated edge-by-edge in the same loop as the sums of antidiffusive fluxes to be limited

$$\begin{aligned} \delta u_i^{\max} &:= \max\{\delta u_i^{\max}, u_j - u_i\}, \quad & \delta u_j^{\max} &:= \max\{\delta u_j^{\max}, u_i - u_j\}, \\ \delta u_i^{\min} &:= \min\{\delta u_i^{\min}, u_j - u_i\}, \quad & \delta u_j^{\min} &:= \min\{\delta u_j^{\min}, u_i - u_j\}. \end{aligned} \tag{4.86}$$

The resulting 'lumped' upper/lower bounds $Q_i^{\pm}$ are of the form (4.39), where

$$u_i^{\max} = u_i + \delta u_i^{\max}, \qquad u_i^{\min} = u_i + \delta u_i^{\min} \tag{4.87}$$

and the coefficients $q_i^{\pm} = \sum_{j \neq i} q_{ij}$ are nonnegative since $q_{ij} \geq 0$ for all $j \neq i$. This enhancement makes it possible to resurrect a larger portion of the raw antidiffusive flux but may inhibit convergence to the steady state and/or give rise to 'terracing.'

4.3.2 Relationship to TVD Limiters

In fact, algebraic flux correction schemes based on standard TVD limiters [200, 206] can also be written in the form (4.80)–(4.82). The corresponding coefficients q_{ij} are given by (4.83) but the raw antidiffusive flux (4.77) is replaced by

$$f_{ij} = \hat{\Phi}(r_i) d_{ij} (u_i - u_j), \qquad r_i = \begin{cases} r_i^+, & \text{if } u_i > u_j, \\ r_i^-, & \text{if } u_i < u_j, \end{cases}$$

where $\hat{\Phi}(r)$ is a function associated with a linear second-order scheme in Sweby's diagram [216, 315]. The generalized smoothness indicator $r_i^{\pm}$ is defined as the ratio of edge contributions with positive and negative coefficients [192, 200, 206]

$$r_i^{\pm} = \frac{\sum_{j \neq i} \max\{0, k_{ij} - k_{ji}\} \max_{\min} \{0, u_j - u_i\}}{\sum_{j \neq i} \min\{0, k_{ij} - k_{ji}\} \min_{\max} \{0, u_j - u_i\}}.$$

For the 1D model problem (4.17) discretized by linear finite elements on a uniform mesh, the so-defined r_i reduces to the ratio of consecutive gradients and [192]

$$\alpha_{ij} = \Phi(r_i), \quad \Phi(r) = \max\{0, \min\{2, \hat{\Phi}(r), 2r\}\}.$$

The Galerkin flux (4.77) corresponds to $\hat{\Phi}(r) \equiv 1$ and $\alpha_{ij} = \max\{0, \min\{1, 2r_i\}\}$, which leads to the limited central difference scheme. The *minmod* and *superbee*

limiters are associated with $\hat{\Phi}(r) = \min\{1, r_i\}$ and $\hat{\Phi}(r) = \max\{1, r_i\}$, respectively. The graphs of other limiter functions lie somewhere in-between.

Even though the multidimensional version of the FEM-TVD algorithm with $\hat{\Phi}(r) \neq 1$ was found to produce good results even on nonuniform triangular meshes [201], it does not revert to the original Galerkin scheme and may fail to stay second-order accurate for smooth data. Instead of manipulating the raw antidiffusive flux in an uncontrollable manner, it is preferable to add some background diffusion to the underlying high-order scheme or adjust the parameters q_{ij} as explained above.

4.3.3 Gradient-Based Slope Limiting

Ideally, the correction factors α_{ij} should approach unity in regions where the solution is well-resolved. In particular, no diffusion or antidiffusion should be applied, even on a nonuniform grid, if the solution varies linearly in the vicinity of node i. This desirable property is known as *linearity preservation* [54, 250]. It was found to maintain consistency and, normally, second-order accuracy on arbitrary meshes.

Classical TVD schemes are linearity-preserving (LP) if $\Phi(1) = 1$, which is the case for all standard limiter functions. Unfortunately, it is difficult to prove the LP property for a multidimensional algebraic flux correction scheme based on algorithm (4.80)–(4.82). It is easier to do so if the fluxes are constrained individually so as to limit the jump of the directional derivative along the edge. To this end, let the correction factor α_{ij} be defined in terms of the limited slope $\bar{s}_{ij}$ such that

$$\bar{s}_{ij} = \alpha_{ij}(u_i - u_j).$$

To find the right value of $\bar{s}_{ij}$, one needs a LED-type estimate of the solution gradient at node i. As explained in Section 2.1.4, a continuous approximation to nodal gradients can be obtained, for example, using the lumped-mass L_2-projection

$$\mathbf{g}_i = \frac{1}{m_i} \sum_k \mathbf{c}_{ik} u_k, \tag{4.88}$$

where m_i is a diagonal entry of the lumped mass matrix M_L. The coefficient vectors

$$\mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \, \mathrm{d}\mathbf{x}$$

constitute the discrete gradient operator $\mathbf{C} = \{\mathbf{c}_{ij}\}$ which has zero row sums so that

$$\mathbf{g}_i = \frac{1}{m_i} \sum_{k \neq i} \mathbf{c}_{ik}(u_k - u_i).$$

For any pair of nodes i and j, a usable approximation to the difference $u_i - u_j$ is

$$s_{ij} = \mathbf{g}_i \cdot (\mathbf{x}_i - \mathbf{x}_j). \tag{4.89}$$

The slope s_{ij} can be estimated in terms of the maxima and minima (4.87) thus:

$$\gamma_{ij}(u_i^{\min} - u_i) \le s_{ij} \le \gamma_{ij}(u_i^{\max} - u_i), \tag{4.90}$$

where the LED upper and lower bounds depend on the nonnegative coefficients

$$\gamma_{ij} = \frac{1}{m_i} \sum_{k \neq i} |\mathbf{c}_{ik} \cdot (\mathbf{x}_i - \mathbf{x}_j)|, \qquad \forall j \neq i. \tag{4.91}$$

Finally, the constrained slope $\bar{s}_{ij}$ is taken to be $u_i - u_j$ or twice the upper/lower bound (4.90) for the extrapolated value s_{ij}, whichever is smaller in magnitude

$$\bar{s}_{ij} = \begin{cases} \min\{2\gamma_{ij}(u_i^{\max} - u_i), u_i - u_j\}, & \text{if } u_i > u_j, \\ \max\{2\gamma_{ij}(u_i^{\min} - u_i), u_i - u_j\}, & \text{if } u_i < u_j. \end{cases} \tag{4.92}$$

Remark 4.15. A symmetric version of this formula was proposed in [204] in the context of algebraic flux correction schemes for anisotropic diffusion problems.

As the mesh is refined, the difference between the local slopes shrinks and $\bar{s}_{ij}$ approaches $u_i - u_j$. This guarantees consistency and linearity preservation for the high-resolution scheme (4.44) in which the limited antidiffusive fluxes are given by

$$\bar{f}_{ij} = \alpha_{ij} d_{ij}(u_i - u_j) = d_{ij}\bar{s}_{ij}, \qquad \forall j \neq i. \tag{4.93}$$

Furthermore, the sum of limited antidiffusive fluxes can be written in the LED form

$$Q_i^- = q_i^-(u_i^{\min} - u_i) \le \sum_{j \neq i} \alpha_{ij} f_{ij} \le q_i^+(u_i^{\max} - u_i) = Q_i^+, \tag{4.94}$$

where the parameters $q_i^{\pm}$ combine the coefficients of all positive/negative slopes

$$q_i^- = \sum_{u_i < u_j} \bar{q}_{ij}, \quad q_i^+ = \sum_{u_i > u_j} \bar{q}_{ij}, \tag{4.95}$$

$$0 \le \bar{q}_{ij} \le q_{ij} = 2\gamma_{ij} d_{ij}, \quad \forall j \neq i. \tag{4.96}$$

This representation proves that the slope-limited scheme is positivity-preserving.

Remark 4.16. The performance of the slope limiter (4.92) depends on the smoothness of the solution and on the quality of underlying gradient recovery method. If the gradient is discontinuous, the standard lumped-mass L_2-projection (4.88) may converge too slowly. To avoid sampling data from both sides of an internal interface, adaptive gradient reconstruction techniques of ENO type [20] can be employed.

Example 4.2. In one dimension, the lumped-mass L_2-projection (4.88) with $m_i = \Delta x$ and $c_{i\pm 1/2} = \pm 1/2$ reduces to the second-order accurate central difference

$$g_i = \frac{1}{2}\left[\frac{u_i - u_{i-1}}{\Delta x} + \frac{u_{i+1} - u_i}{\Delta x}\right] = \frac{u_{i+1} - u_{i-1}}{2\Delta x}.$$

For any interior node, the local maxima and minima of the 1D grid function u are

$$u_i^{\max} = \max\{u_{i-1}, u_i, u_{i+1}\}, \qquad u_i^{\min} = \min\{u_{i-1}, u_i, u_{i+1}\}.$$

Estimate (4.90) with $\gamma_{ij} = 1$ and $j = i+1$ yields the upper and lower bounds

$$u_i^{\min} - u_i \le \Delta x g_i \le u_i^{\max} - u_i.$$

Finally, the one-dimensional version of formula (4.92) can be written as follows

$$\bar{s}_{ij} = minmod\{2(u_{i-1} - u_i), u_i - u_{i+1}\}. \tag{4.97}$$

The *minmod* function, as defined in (4.79), compares the signs and magnitudes of the two slopes. If a local maximum or minimum is attained at node i, then the slope ratio is negative and, therefore, $\bar{s}_{ij} = 0$. Otherwise, the result is $\bar{s}_{ij} = u_i - u_{i+1}$ or a slope of the same sign and smaller magnitude. Limiting is performed only if the two slopes have opposite signs or their magnitudes differ by a factor of two and more.

Remark 4.17. In the case of the linear convection equation (4.17), the limited antidiffusive flux (4.93) is the same as that obtained with algorithm (4.80)–(4.83).

4.3.4 Reconstruction of Local Stencils

The traditional approach to implementation of high-resolution schemes on unstructured meshes is based on reconstruction of one-dimensional stencils associated with mesh edges [8, 170, 244]. The key idea is to extend the edge connecting the vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ in both directions so as to define a pair of dummy nodes. The interpolated or extrapolated solution values at these points are used to define the local slopes that make it possible to design the diffusive and antidiffusive fluxes as in the 1D case.

In upwind-biased algorithms, just one dummy node is involved. Given a pair of neighboring nodes i and j such that convention (4.14) holds, the position of the third node for an equidistant local stencil is $\mathbf{x}_k = 2\mathbf{x}_i - \mathbf{x}_j$. Alternatively, the dummy node can be placed at the intersection $\mathbf{x}_l$ of the straight line through $\mathbf{x}_i$ and $\mathbf{x}_j$ with the boundary of the adjacent element E located upstream of the point $\mathbf{x}_i$, see Fig. 4.1.

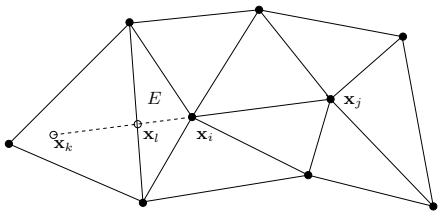


Fig. 4.1 Upwind stencil reconstruction on an unstructured triangular mesh.

4.3.4.1 Construction of SLIP Schemes

The flux correction process involves edge-by-edge reconstruction and limiting of slopes for the local 1D stencil. The slope estimated using the data at $\mathbf{x}_i$ and $\mathbf{x}_k$ is

$$s_{ij} = u_k - u_i. \tag{4.98}$$

If the dummy node is located at the point $\mathbf{x}_l$, then the formula for s_{ij} becomes

$$s_{ij} = \frac{|\mathbf{x}_i - \mathbf{x}_l|}{|\mathbf{x}_j - \mathbf{x}_i|}(u_l - u_i). \tag{4.99}$$

In either case, the limited antidiffusive flux (4.93) can be defined in terms of

$$\bar{s}_{ij} = minmod\{2s_{ij}, u_i - u_j\}. \tag{4.100}$$

This formula is similar to (4.92) and leads to an algorithm that belongs to the class of upstream slope-limited positive (SLIP) schemes introduced by Jameson [170].

A wealth of other high-resolution schemes, including straightforward generalizations of classical TVD and MUSCL methods, can be derived using the reconstructed slope s_{ij} to design the modified numerical flux for a finite element or finite volume discretization on an unstructured mesh [8, 170, 226, 239, 243]. Although such algorithms are sensitive to the orientation of mesh edges and may lack a rigorous theoretical justification, they belong to the most successful unstructured mesh methods for convection-dominated transport problems and hyperbolic conservation laws.

4.3.4.2 Recovery via FEM Interpolation

The quality of a SLIP-like scheme depends on the structure of the slopes s_{ij} given by (4.98) or (4.99). Let the unknown solution value at a dummy node $\bar{\mathbf{x}}$ be interpolated using linear or multilinear finite element basis functions $\{\varphi_i\}$ such that

$$u(\bar{\mathbf{x}}) = \sum_j u_j \varphi_j(\bar{\mathbf{x}}) \tag{4.101}$$

is a positivity-preserving convex average of the nodal values u_j at the vertices of the element that contains the point $\bar{\mathbf{x}}$. For the upwind triangle E depicted in Fig. 4.1, linear interpolation is performed using local basis functions $\{\hat{\varphi}_1, \hat{\varphi}_2, \hat{\varphi}_3\}$. Without loss of generality, assume that $\hat{\varphi}_3 = \varphi_i|_E$ is the one associated with $\hat{u}_3 = u_i$. Since this basis function vanishes on the edge where the point $\bar{\mathbf{x}} = \mathbf{x}_l$ resides, we have

$$u_l = \hat{u}_1\hat{\varphi}_1(\mathbf{x}_l) + \hat{u}_2\hat{\varphi}_2(\mathbf{x}_l) = \xi\hat{u}_1 + (1-\xi)\hat{u}_2, \qquad 0 \le \xi \le 1. \tag{4.102}$$

Likewise, the value of u_k can be determined using local basis functions defined on the actual element to which $\bar{\mathbf{x}} = \mathbf{x}_k$ belongs. This reconstruction technique is recommended in [239] but it may require a costly search for the host element and

sampling data from points that are not nearest neighbors of node i. In any case, if the limited slope $\bar{s}_{ij}$ is proportional to $u(\overline{\mathbf{x}}) - u_i$ and has the same sign, then (4.101) implies that the flux $\bar{f}_{ij} = d_{ij}\bar{s}_{ij}$ is of LED type and there is no threat to positivity.

4.3.4.3 Gradient-Based Reconstruction

Another popular approach to the reconstruction of s_{ij} is gradient-based extrapolation

$$u_k = u_i + s_{ij}, \qquad s_{ij} = (\nabla u)_i \cdot (\mathbf{x}_i - \mathbf{x}_j). \tag{4.103}$$

The nodal gradient $(\nabla u)_i$ can be approximated using the upwind-sided derivatives or gradient averaging. In the latter case, the above formula reduces to (4.89) which may fail to possess the LED property if $(\nabla u)_i = \mathbf{g}_i$ is recovered via the lumped-mass L_2-projection. In the upwind-biased version, the gradient is obtained by differentiating the solution $u_h|_E$ restricted to the first mesh cell crossed by the line $\mathbf{x}_i\mathbf{x}_k$

$$(\nabla u)_i = \sum_k \hat{u}_k \nabla \hat{\varphi}_k(\mathbf{x}_i).$$

As before, $\hat{\varphi}_k$ denotes a local basis function associated with the upwind element E and $\hat{u}_k$ is the corresponding nodal value. For linear triangles, the result is [170, 239]

$$s_{ij} = \hat{\varphi}_1(\hat{u}_1 - \hat{u}_3) + \hat{\varphi}_2(\hat{u}_2 - \hat{u}_3),$$

where $\hat{u}_3 = u_i$ under the node numbering convention adopted in (4.102). Since $\hat{\varphi}_1$ and $\hat{\varphi}_2$ are nonnegative, the slope s_{ij} is LED and so is its limited counterpart $\bar{s}_{ij}$.

For a detailed comparative study of various techniques for reconstruction of local 1D stencils in edge-based finite element codes, we refer to Lyra [239, 243, 240]. Slope limiting based on (4.99)–(4.101) is probably the most attractive among the considered alternatives since it is relatively simple, positivity-preserving, and applicable not only to simplex meshes but also to quadrilateral/hexahedral/hybrid ones.

4.3.5 Background Dissipation

The nondissipative nature of the central difference scheme and Galerkin finite element methods is known to have an adverse effect on the convergence of their flux-limited counterparts. Therefore, it is generally preferable to use another base scheme which incorporates some streamline diffusion or dampens oscillatory modes by means of higher-order dissipation [170, 239, 357]. Alternatively, the stabilization effect can be achieved in the process of flux correction. Consider a pair of slopes

$$s_{ij} = \mathbf{e}_{ij} \cdot (\nabla u)_i, \qquad s_{ji} = \mathbf{e}_{ji} \cdot (\nabla u)_j, \qquad \mathbf{e}_{ij} = \mathbf{x}_i - \mathbf{x}_j \tag{4.104}$$

obtained with any of the above gradient reconstruction techniques. To introduce background dissipation, let the raw antidiffusive flux (4.77) be redefined as [170]

$$f_{ij} = d_{ij}\left(\frac{s_{ij}-s_{ji}}{2}\right) = d_{ij}\mathbf{e}_{ij}\cdot\frac{(\nabla u)_i+(\nabla u)_j}{2}. \tag{4.105}$$

In upwind-biased flux correction schemes, this replacement should be carried out before the minmod prelimiting (4.78) is applied. Furthermore, the reconstructed slope $(s_{ij}-s_{ji})/2$ should be used instead of u_i-u_j in formulas like (4.92) and (4.100), whereas the upper and lower bounds for the limiter remain unchanged.

Remark 4.18. The one-dimensional counterpart of the flux (4.105) is as follows

$$f_{ij} = d_{ij}\left(\frac{u_{i-1}-u_i+u_{i+1}-u_{i+2}}{2}\right), \qquad j = i+1. \tag{4.106}$$

The Taylor series expansion reveals that (4.105) is related to (4.77) via [226]

$$\mathbf{e}_{ij}\cdot\frac{(\nabla u)_i+(\nabla u)_j}{2} \approx u_i-u_j-\frac{|\mathbf{e}_{ij}|^2}{4}(u_j''-u_i''),$$

where $u''=(\mathbf{e}_{ij}\cdot\nabla)^2u$ denotes the second directional derivative of u along the edge.

Hence, a simple way to introduce fourth-order damping into a centered scheme is to augment the corresponding flux (4.23) by a dissipative term proportional to the difference between the approximate nodal values of second derivatives. Such terms extend the stencil of the numerical scheme and are not of LED type but positivity preservation is enforced by the flux/slope limiter applied to f_{ij}. As a result, background dissipation is added in smooth regions and low-order diffusion elsewhere. The use of high-order fluxes with a dissipative component makes the constrained scheme more robust and less susceptible to 'terracing' or similar side effects.

In multidimensions, the scalar-valued Laplacian is cheaper to calculate than the gradient. This is the rationale behind the following definition [226, 250, 301, 302]

$$f_{ij} = d_{ij}(u_i-u_j) - d_{ij}''|\mathbf{x}_j-\mathbf{x}_i|^2((\Delta u)_j-(\Delta u)_i), \tag{4.107}$$

where d_{ij}'' is a nonnegative coefficient associated with fourth-order dissipation. The nodal Laplacians $(\Delta u)_i$ and $(\Delta u)_j$ can be recovered as explained in Section 2.1.4 or approximated using the off-diagonal entries of the consistent mass matrix [301].

Remark 4.19. In the 1D case, definition (4.107) with $d_{ij}''=\frac{d_{ij}}{2}$ reduces to (4.106) if the second derivatives at nodes i and j are approximated by central differences

$$(\Delta u)_i = \frac{u_{i-1}-2u_i+u_{i+1}}{(\Delta x)^2}, \qquad x_i = i\Delta x, \quad \forall i.$$

Remark 4.20. Fourth-order dissipation is linearity-preserving since all second derivatives of a linear function are zero. This property turns out to be a valuable tool for the theoretical analysis of accuracy and consistency on general triangulations [250].

4.3.6 Numerical Examples

To assess the accuracy of flux-limited Galerkin approximations to stationary transport equations, a comparative study is performed for high-resolution schemes of Upwind-LED type. The four methods under investigation are abbreviated by

ULED-0	the low-order scheme which corresponds to $\alpha_{ij} \equiv 0$,
ULED-1	algebraic flux correction of TVD type (4.80)–(4.83),
ULED-2	slope limiting based on gradient recovery and (4.92),
ULED-3	the upstream SLIP scheme given by (4.99)–(4.101).

In one space dimension, the last three algorithms are equivalent to one another and produce antidiffusive fluxes proportional to the limited slope (4.97). The use of upper/lower bounds (4.84) instead of (4.83) has little influence on the accuracy and qualitative behavior of ULED-1, at least for the test problems considered here.

Numerical solutions are marched to the steady state using pseudo-time stepping of backward Euler type. Nonlinear systems are solved by the defect correction scheme (4.53) preconditioned by the monotone low-order operator (4.57). Implicit underrelaxation with $\omega \equiv 0.8$ is performed to secure convergence to a steady state.

4.3.6.1 Circular Convection

The first test problem is taken from [156]. Consider the hyperbolic conservation law

$$\nabla \cdot (\mathbf{v}u) = 0 \qquad \text{in } \Omega = (-1,1) \times (0,1). \tag{4.108}$$

This equation describes steady circular convection if the velocity field is defined as

$$\mathbf{v}(x,y) = (y,-x).$$

The exact solution and inflow boundary conditions for this test case are given by

$$u(x,y) = \begin{cases} G(r), & \text{if } 0.35 \le r = \sqrt{x^2+y^2} \le 0.65, \\ 0, & \text{otherwise,} \end{cases}$$

where $G(r)$ is a function that defines the shape of the solution profile along the inflow ($-1 \le x < 0$) and outflow ($0 < x \le 1$) part of the boundary Γ at $y = 0$.

Since equation (4.108) is linear, both smooth and discontinuous data propagate along the characteristics that coincide with the streamlines of the stationary velocity field (see Chapter 3). The ability of a numerical scheme to maintain smooth peaks and discontinuities is tested by imposing inflow boundary conditions defined by

$$G_1(r) = \cos^2\left(5\pi\frac{2r+1}{3}\right), \qquad G_2(r) \equiv 1.$$

The test problems that deal with circular convection of $G = G_1$ and $G = G_2$ will be referred to as CC1 and CC2, respectively. All numerical solutions are computed on a uniform mesh of bilinear finite elements which is successively refined to perform a grid convergence study. The errors are measured in the discrete norms

$$E_{\max} = \max_i |u(\mathbf{x}_i) - u_i| \approx ||u - u_h||_\infty, \tag{4.109}$$

$$E_2 = \sqrt{\sum_i m_i |u(\mathbf{x}_i) - u_i|^2} \approx ||u - u_h||_2, \tag{4.110}$$

where $m_i = \int_\Omega \varphi_i \, \mathrm{d}\mathbf{x}$ denotes a diagonal coefficient of the lumped mass matrix or, equivalently, the area of the control volume associated with the mesh point $\mathbf{x}_i$.

Figure 4.2a displays the steady-state solution to CC1 computed by ULED-2 on a mesh with spacing $h = 1/64$. The outflow profiles produced by the four schemes on this mesh are compared in Fig. 4.2b. The uncorrected low-order solution (ULED-0)

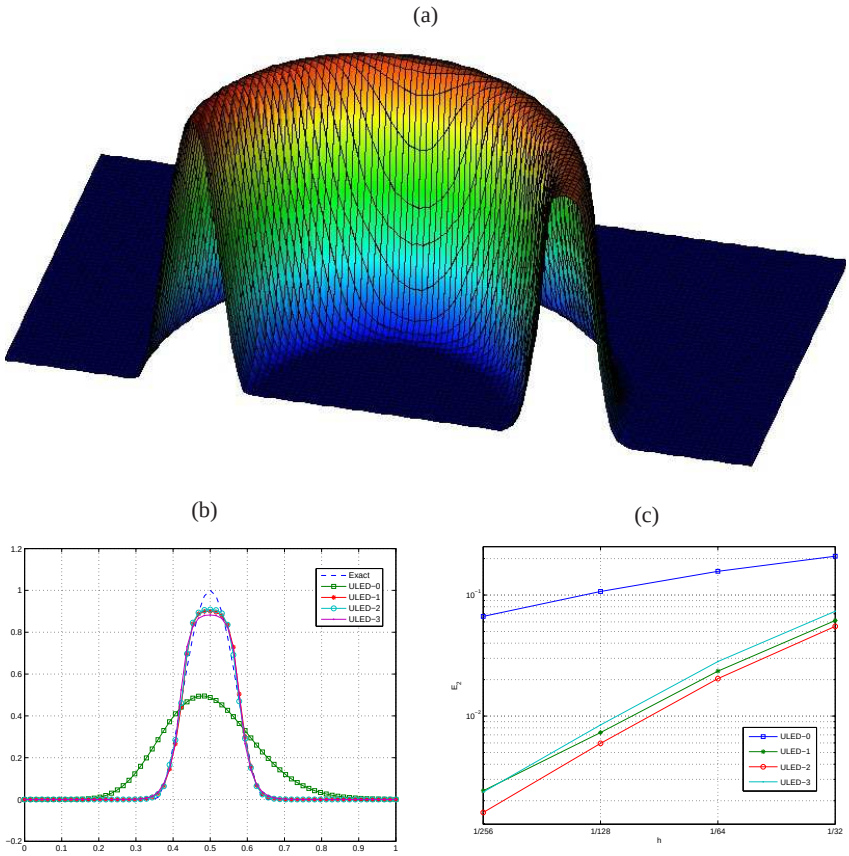


Fig. 4.2 Circular convection, ULED results for smooth data.

is not only strongly smeared by numerical diffusion but also asymmetric. The other three curves are much closer to the dashed line that depicts the exact solution. The highest accuracy is achieved with ULED-2, followed by ULED-1 and ULED-3.

The log-log plot in Fig. 4.2c shows the variation of the E_2 error with the mesh size h. The order of accuracy $p = \log_2(E_2(h)/E_2(h/2))$ estimated on the finest mesh level ($h = 1/128$) equals $\{0.68, 1.59, 1.89, 1.84\}$ for ULED-0 through ULED-3, respectively. The values of E_2 and $E_{\max}$ for all meshes are presented in Table 4.1. The inability of LED methods to distinguish a smooth peak from a spurious maximum/minimum is the reason why even flux-corrected versions fail to attain second-order accuracy in this example. Nevertheless, a large portion of artificial diffusion can be removed in the process of flux correction as long as the solution is smooth.

In the test case CC2, the exact solution is discontinuous at the inlet and remains so along the streamlines of the incompressible velocity field. The numerical solution produced by ULED-2 with $h = 1/64$ is shown in Fig. 4.3a. It is devoid of under-

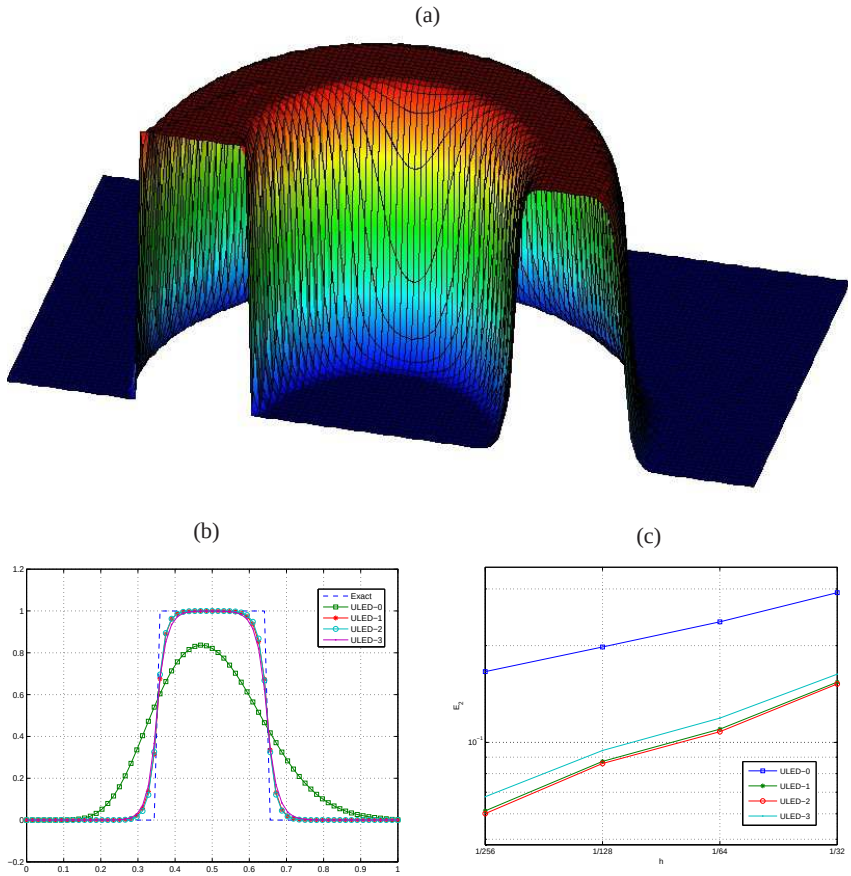


Fig. 4.3 Circular convection, ULED results for discontinuous data.

Table 4.1 Circular convection, smooth data, convergence history.

	ULED-0		ULED-1		ULED-2		ULED-3	
h	E_2	$E_{\max}$	E_2	$E_{\max}$	E_2	$E_{\max}$	E_2	$E_{\max}$
1/32	0.209e0	0.637e0	0.616e-1	0.258e0	0.551e-1	0.235e0	0.734e-1	0.296e0
1/64	0.157e0	0.512e0	0.235e-1	0.998e-1	0.204e-1	0.917e-1	0.282e-1	0.118e0
1/128	0.107e0	0.375e0	0.731e-2	0.375e-1	0.595e-2	0.340e-1	0.845e-2	0.452e-1
1/256	0.666e-1	0.244e0	0.242e-2	0.132e-1	0.160e-2	0.118e-1	0.236e-2	0.162e-1

Table 4.2 Circular convection, discontinuous data, convergence history.

	ULED-0		ULED-1		ULED-2		ULED-3	
h	E_2	$E_{\max}$	E_2	$E_{\max}$	E_2	$E_{\max}$	E_2	$E_{\max}$
1/32	0.292e0	0.600e0	0.154e0	0.605e0	0.152e0	0.597e0	0.163e0	0.584e0
1/64	0.237e0	0.561e0	0.110e0	0.562e0	0.108e0	0.566e0	0.119e0	0.570e0
1/128	0.198e0	0.573e0	0.873e-1	0.667e0	0.860e-1	0.683e0	0.944e-1	0.660e0
1/256	0.166e0	0.540e0	0.613e-1	0.550e0	0.601e-1	0.557e0	0.678e-1	0.569e0

shoots/overshoots and exhibits a fairly high resolution. Figure 4.3b reveals that the qualitative behavior of the four methods is the same as that for CC1. Again, the low-order solution (ULED-0) is strongly smeared and asymmetric, whereas the differences between the results produced by ULED-1 through ULED-3 are marginal. The definition of upper/lower bounds in ULED-2 enables the slope limiter to accept more antidiffusion than in the case of ULED-1 or ULED-3. The latter proves to be the most diffusive among the three flux correction schemes under investigation.

The convergence history presented in Fig. 4.3c and Table 4.2 confirms that the presence of a discontinuous profile has an adverse effect on the overall performance of numerical schemes. In this example, mesh refinement does not necessarily improve the value of $E_{\max}$, while the effective order of accuracy with respect to E_2 deteriorates to $\{0.25, 0.51, 0.52, 0.48\}$ for ULED-0 through ULED-3, respectively. Still, the flux-corrected versions converge twice as fast as the low-order scheme.

In conclusion, no discretization technique can resolve smooth and discontinuous profiles equally well. However, flux correction still pays off as long as it increases the effective order of accuracy by a factor of 2 as compared to the underlying low-order scheme. It is not the absolute value of the error but the ratio of errors and convergence rates that determines which scheme is the best for a given problem.

4.3.6.2 Convection-Diffusion

Another popular test case is the singularly perturbed convection-diffusion equation

$$\nabla \cdot (\mathbf{v}u - \varepsilon \nabla u) = 0 \qquad \text{in } \Omega = (0,1) \times (0,1) \tag{4.111}$$

which is nominally elliptic but may give rise to arbitrarily sharp internal and boundary layers (see Section 3.1.5). Following John and Knobloch [172], we consider

$$\mathbf{v} \equiv \left(\cos\frac{\pi}{3}, -\sin\frac{\pi}{3}\right), \qquad \varepsilon = 10^{-8}$$

and impose a Dirichlet boundary condition which is discontinuous at $\mathbf{x}_0 = (0, 0.7)$

$$u(x,y) = \begin{cases} 0 & \text{if } x = 1 \text{ or } y \leq 0.7, \\ 1 & \text{otherwise.} \end{cases}$$

The exact solution to the above boundary value problem has an internal layer along the streamline through the point $\mathbf{x}_0$ and a boundary layer next to the line $y = 0$.

To our knowledge, John and Knobloch [172] were the first to perform a detailed and systematic comparative study of conventional stabilized FEM for transport equations with small diffusion and sharp layers. It turns out that even the use of a nonlinear shock-capturing viscosity may fail to prevent a violation of the discrete maximum principle. In this section, we contribute the results computed with the four ULED schemes that guarantee the validity of the DMP by construction.

The discretization of (4.111) is performed using linear and bilinear finite elements on three uniform meshes. The first (Grid 1) and second (Grid 2) one are triangular and have the same vertices as a Cartesian mesh (Grid 3) with equal spacing in both coordinate directions. The orientation of mesh edges is shown in Fig. 4.4. The total number of nodes (vertices) is $N = (1 + 1/h)^2$, where h is the mesh size.

The numerical solutions in Fig. 4.5 were produced by ULED-0 and ULED-2 on Grid 1 with 4,225 degrees of freedom, which corresponds to $h = 1/64$. As expected, the low-order solution (left diagram) exhibits a stronger smearing of the interior and boundary layers. To assess the rate of smearing and enable a quantitative comparison of different methods, the following benchmark quantities are introduced in [172]

$$smear_{\text{int}} = x_2 - x_1, \qquad smear_{\text{exp}} = \sqrt{\sum_{\mathbf{x}_i \in \Omega_2} (\min\{0, u_i - 1\})^2},$$

where $\Omega_2 = \{(x,y) \in \Omega \,|\, x \geq 0.7\}$. The points x_1 and x_2 are chosen so that [172]

$$0.1 = u_h(x_1, 0.25) \leq u_h(x, 0.25) \leq u_h(x_2, 0.25) = 0.9, \qquad \forall x \in (x_1, x_2).$$

To determine the values of x_1 and x_2 for the cutline $y = 0.25$, the approximate solution u_h is evaluated on a one-dimensional subgrid with spacing 10^{-5} [172].

The magnitudes of $smear_{\text{int}}$ and $smear_{\text{exp}}$ measure the thickness of the internal and (exponential) boundary layer, respectively. Two additional benchmark quantities (osc_{int} and osc_{exp}) are defined in [172] to quantify the pollution by undershoots and overshoots. In the present study, both of them are identically equal to zero.

Table 4.3 displays the results obtained using algebraic flux correction on the three meshes with $h = 1/64$. The reader is invited to compare these results to those produced by SUPG-like finite element methods in [172]. While the low-order scheme (ULED-0) is characterized by inordinately large values of $smear_{\text{int}}$ and $smear_{\text{exp}}$,

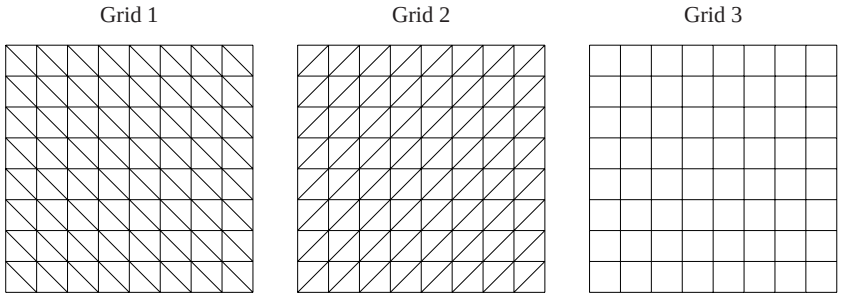


Fig. 4.4 John-Knobloch benchmark: computational meshes for $h = 1/8$.

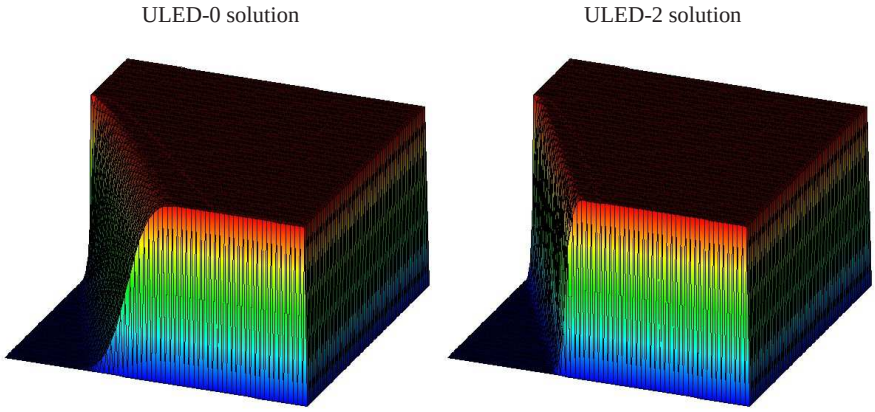


Fig. 4.5 John-Knobloch benchmark: numerical solutions on Grid 1, $h = 1/64$.

the ULED antidiffusive correction is seen to reduce the amount of smearing, while keeping the numerical solution free of spurious oscillations in the vicinity of internal and boundary layers. As in the previous example, the most accurate solutions are obtained with ULED-2. This flux correction scheme receives the highest possible score 10 as defined in [172] for Grid 1. The different pattern of linear triangles in Grid 2 and the bilinear approximation on Grid 3 give rise to stronger smearing but the results are still quite good as compared to other finite element methods [172].

Table 4.3 John-Knobloch benchmark: results for all grids with $h = 1/64$.

	ULED-0		ULED-1		ULED-2		ULED-3	
Grid	$smear_{\text{int}}$	$smear_{\text{exp}}$	$smear_{\text{int}}$	$smear_{\text{exp}}$	$smear_{\text{int}}$	$smear_{\text{exp}}$	$smear_{\text{int}}$	$smear_{\text{exp}}$
1	0.1176	9.1e-6	0.0388	7.8e-6	0.0379	7.9e-6	0.0416	7.9e-6
2	0.2457	1.5494	0.0665	0.4140	0.0615	0.3544	0.0751	0.4180
3	0.1929	0.8525	0.0570	0.2222	0.0566	0.1961	0.0675	0.2240

4.4 Unsteady Transport Problems

In unsteady transport problems, the Galerkin discretization of space derivatives can also be fixed using any of the flux correction schemes presented in the previous section. Within the framework of the method of lines (MOL), integration in time can be performed by an arbitrary explicit or implicit algorithm. However, the use of small time steps in transient computations makes the upper/lower bounds considered so far more restrictive than necessary to keep the scheme positivity-preserving. Moreover, the time-dependent part of the raw antidiffusive flux (4.23) for a finite element discretization may become dominant. Neglecting it would make the mass lumping error irrecoverable and significantly degrade the phase accuracy. The use of an upwind-biased algorithm with minmod prelimiting (4.78) may also result in a serious loss of accuracy. In such situations, algebraic flux correction schemes based on the flux-corrected transport (FCT) methodology typically perform much better.

FCT was the first nonlinear high-resolution scheme to produce sharp and monotone solutions even in the limit of pure convection [41]. The early FCT algorithms of Boris, Book, and Hain [42, 43, 44] involve two basic steps:

1. Advance the solution in time by an explicit low-order scheme that incorporates enough numerical diffusion to suppress undershoots and overshoots.
2. Correct the solution using antidiffusive fluxes limited in such a way that no new maxima or minima can form and existing extrema cannot grow.

This predictor-corrector strategy is typical of diffusion-antidiffusion (DAD) methods [82]. The job of the numerical diffusion built into the low-order scheme is to maintain positivity and provide good phase accuracy. The antidiffusive correction is intended to reduce the amplitude errors in a local extremum diminishing manner.

Zalesak's fully multidimensional FCT algorithm [355] is based on blending explicit high- and low-order approximations so as to constrain the maximum and minimum increments to each nodal value. A detailed presentation of the underlying design philosophy can be found in [357]. Zalesak's limiter has had a significant impact on the development of the algebraic flux correction paradigm and served as a prototype for the generic limiting strategy presented in Section 4.1.5. In contrast to TVD schemes and extensions thereof, flux limiters of FCT type operate at the fully discrete level and are designed to accept as much antidiffusion as possible.

The combination of FCT with finite elements and unstructured meshes dates back to the explicit algorithms of Parrott and Christie [266] and Löhner et al. [232, 233]. Several implicit FEM-FCT schemes were published by the author and his coworkers [191, 203, 205, 258]. The rationale for the use of an implicit time discretization stems from the fact that the CFL stability condition becomes prohibitively restrictive in the case of strongly nonuniform velocity fields and/or locally refined meshes. Woodward and Colella ([347], p. 119) conclude that "adaptive grid schemes have a major drawback – they demand an implicit treatment of the flow equations." This statement reflects a widespread prejudice that implicit schemes are computationally expensive. As a matter of fact, the cost of an implicit algorithm depends on the choice of iterative methods, parameter settings, and stopping criteria. If the time

step is very small, then a good initial guess is available and the sparse linear system can be solved with 1-2 iterations of the Jacobi or Gauß-Seidel method. Thus, the cost per time step approaches that of an explicit finite difference or finite volume scheme. As the time step increases, so does the number of iterations, and more sophisticated linear algebra tools (smoothers, preconditioners) may need to be employed.

When antidiffusive fluxes depend on the unknown solution, the nonlinear algebraic system must be replaced by a sequence of linearized ones in which the antidiffusive term is evaluated using the previously computed data. Sometimes, too many flux/defect correction cycles are required to obtain a fully converged solution, especially if the Courant number is large and the contribution of the consistent mass matrix cannot be neglected. The use of a discrete Newton method [258] makes it possible to accelerate convergence but the computational cost per time step is still rather high as compared to that of a fully explicit algorithm. This is unacceptable since the time step for FCT must be chosen relatively small for accuracy reasons.

In this section, we consider both the nonlinear FEM-FCT procedure and simple predictor-corrector algorithms of diffusion-antidiffusion type. In order to reduce the cost of flux correction, we linearize the antidiffusive fluxes about a nonoscillatory end-of-step solution computed by an explicit or implicit low-order scheme [196]. Going back to the roots of FCT, we correct this "transported and diffused" solution directly instead of modifying the algebraic system and solving it again. This fractional-step approach seems to provide the best cost-accuracy ratio [175, 196].

4.4.1 Nonlinear FEM-FCT Schemes

A family of implicit FEM-FCT algorithms was developed in [191, 203, 205] using the algebraic approach to flux correction. Consider system (4.44) discretized in time by the two-level $\theta-$scheme which yields a nonlinear system of the form (4.46)

$$[M_L - \theta\Delta t L^{n+1}]u^{n+1} = [M_L + (1-\theta)\Delta t L^n]u^n + \Delta t\,\bar{f}(u^{n+1}, u^n), \tag{4.112}$$

where $0 \le \theta \le 1$ is the degree of implicitness. The $i-$th element of the vector $\bar{f}$ is

$$\bar{f}_i = \sum_{j\neq i} \bar{f}_{ij}, \qquad \bar{f}_{ij} = \alpha_{ij} f_{ij}, \quad 0 \le \alpha_{ij} \le 1. \tag{4.113}$$

The raw antidiffusive flux f_{ij} is the fully discrete counterpart of (4.23) defined as

$$\begin{aligned} f_{ij} = {} & [m_{ij}(u_i^{n+1} - u_j^{n+1}) - m_{ij}(u_i^n - u_j^n)]/\Delta t \\ & + \theta d_{ij}^{n+1}(u_i^{n+1} - u_j^{n+1}) + (1-\theta) d_{ij}^n (u_i^n - u_j^n). \end{aligned} \tag{4.114}$$

Interestingly enough, the contribution of the consistent mass matrix to f_{ij} combines a truly antidiffusive implicit part and a diffusive explicit part. Mass diffusion of the form $D = M_C - M_L$ offers a cheap way to construct the nonoscillatory low-order scheme within the framework of explicit FEM-FCT algorithms [205, 232].

However, the associated time step restriction (3.92) is more severe than that for the artificial diffusion operator D with variable coefficients given by (4.11) and (4.12).

The forward Euler version ($\theta = 0$) of (4.112) is not to be recommended. If the underlying high-order discretization ($\alpha_{ij} \equiv 1$) is linearly unstable, then aggressive flux limiting may result in a significant distortion of solution profiles. Also, first-order time accuracy is insufficient for simulation of unsteady phenomena. If a fully explicit solution strategy is preferred, then a Lax-Wendroff/Taylor-Galerkin method or a TVD Runge-Kutta scheme, such as (3.97)–(3.98), should be employed.

The fixed-point iteration method (4.49) transforms (4.112) into a sequence of linear systems for approximations $\{u^{(m)}\}$ to the end-of-step solution u^{n+1}

$$[M_L - \theta \Delta t L^{(m)}]u^{(m+1)} = [M_L + (1-\theta)\Delta t L^n]u^n + \Delta t\, \bar{f}(u^{(m)}, u^n). \tag{4.115}$$

A natural initial guess is $u^{(0)} = u^n$ such that $f_{ij}^{(0)} = d_{ij}^n(u_i^n - u_j^n)$. In our experience, convergence is faster if the contribution of the time derivative is approximated by

$$f_{ij}^{(0)} = [m_{ij}(u_i^n - u_j^n) - m_{ij}(u_i^{n-1} - u_j^{n-1})]/\Delta t + d_{ij}^n(u_i^n - u_j^n).$$

Each solution update of the form (4.115) can be split into three steps [191, 196]

1. Compute an explicit low-order approximation to $u^{n+1-\theta}$ by solving

$$M_L \tilde{u} = [M_L + (1-\theta)\Delta t L^n]u^n. \tag{4.116}$$

2. Apply limited antidiffusive fluxes to the intermediate solution $\tilde{u}$

$$M_L \bar{u} = M_L \tilde{u} + \Delta t\, \bar{f}(u^{(m)}, u^n). \tag{4.117}$$

3. Solve the linear system for the new approximation to u^{n+1}

$$[M_L - \theta \Delta t L^{(m)}]u^{(m+1)} = M_L \bar{u}. \tag{4.118}$$

The auxiliary solution $\tilde{u}$ depends only on u^n and needs to be computed just once at the first outer iteration ($m = 0$). For the explicit update in Step 1 to be positivity-preserving, the time step Δt must satisfy (3.92) with $m_{ii} = m_i$ and $c_{ii} = l_{ii}^n$

$$m_i + (1-\theta)\Delta t l_{ii}^n \geq 0, \qquad 0 \leq \theta < 1. \tag{4.119}$$

The correction factors α_{ij} for Step 2 are determined using Zalesak's multidimensional FCT limiter to be presented below. This algorithm guarantees that $\bar{u} \geq 0$ for $\tilde{u} \geq 0$. Step 3 is positivity-preserving under condition (3.91). In summary,

$$u^n \geq 0 \quad \Rightarrow \quad \tilde{u} \geq 0 \quad \Rightarrow \quad \bar{u} \geq 0 \quad \Rightarrow \quad u^{(m+1)} \geq 0$$

provided that the time step Δt satisfies (3.91) and (3.92) for the given $\theta \in (0, 1]$.

4.4.2 Zalesak's Limiter Revisited

The computation of the correction factors α_{ij} involved in the assembly of (4.113) is based on the fully multidimensional flux-corrected transport algorithm [355]. This limiting strategy represents a symmetric version of (4.35)–(4.38). Again, the objective is to make sure that neither positive nor negative antidiffusive fluxes can conspire to create/enhance a local extremum [355, 357].

4.4.2.1 Prelimiting

Flux correction might be beneficial even in the unlikely case when f_{ij} has the same sign as $\tilde{u}_j - \tilde{u}_i$ and poses no threat to positivity. Such an outlier flattens the solution profile instead of steepening it. As a consequence, numerical ripples may develop within the bounds imposed on the flux-corrected solution. In the Boris-Book limiter [42] and some FEM-FCT algorithms [228], the sign of a defective antidiffusive flux is reversed and the amplitude is limited in the usual way. This trick results in a sharp resolution of discontinuities but may produce excessive antidiffusion elsewhere.

A safer remedy is to cancel f_{ij} if it is directed down the gradient of $\tilde{u}$. That is,

$$f_{ij} := 0, \quad \text{if} \quad f_{ij}(\tilde{u}_j - \tilde{u}_i) > 0. \tag{4.120}$$

Similarly to (4.78), this manipulation should be done before the evaluation of α_{ij}.

Zalesak [355] argued that the effect of (4.120) is marginal and cosmetic in nature since the vast majority of antidiffusive fluxes have the right sign and steepen the solution gradient. This remark might have led many readers to disregard equations (14) and (14′) in [355]. Two decades later, the need for 'prelimiting' of the form (4.120) was emphasized by DeVore [80] who explained its ramifications and demonstrated that it may lead to a marked improvement of accuracy. In our experience, this optional correction step is certainly worth including in FEM-FCT algorithms [205].

In the case of a finite element discretization, the contribution of the consistent mass matrix provides better phase accuracy but may reverse the sign of the antidiffusive spatial part (4.77) or significantly increase its magnitude. For particularly sensitive problems, the *minmod* function (4.79) can be used to redefine f_{ij} as

$$f_{ij} := minmod\{f_{ij}, d_{ij}(\tilde{u}_i - \tilde{u}_j)\}, \tag{4.121}$$

where d_{ij} is a nonnegative coefficient. The default value (4.11) prevents the flux f_{ij} from becoming diffusive or more antidiffusive than its lumped-mass counterpart.

4.4.2.2 Flux Correction

In the context of multidimensional FCT algorithms, the solution-dependent correction factors α_{ij} are chosen so as to ensure that antidiffusive fluxes acting in concert

shall not cause the solution value $\bar{u}_i$ to exceed some maximum value $\tilde{u}_i^{\max}$ or fall below some minimum value $\tilde{u}_i^{\min}$. Assuming the worst-case scenario, positive and negative fluxes should be limited separately, as proposed by Zalesak [355]

1. Compute the sums of positive/negative antidiffusive fluxes into node i

$$P_i^+ = \sum_{j\neq i} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{j\neq i} \min\{0, f_{ij}\}. \tag{4.122}$$

2. Determine the distance to a local maximum/minimum and the bounds

$$Q_i^+ = \frac{m_i}{\Delta t}(\tilde{u}_i^{\max} - \tilde{u}_i), \qquad Q_i^- = \frac{m_i}{\Delta t}(\tilde{u}_i^{\min} - \tilde{u}_i). \tag{4.123}$$

3. Evaluate the nodal correction factors for the net increment to node i

$$R_i^+ = \min\left\{1, \frac{Q_i^+}{P_i^+}\right\}, \qquad R_i^- = \min\left\{1, \frac{Q_i^-}{P_i^-}\right\}. \tag{4.124}$$

4. Check the sign of each raw antidiffusive flux and adjust its magnitude

$$\bar{f}_{ij} = \alpha_{ij} f_{ij}, \qquad \alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\}, & \text{if } f_{ij} > 0, \\ \min\{R_i^-, R_j^+\}, & \text{if } f_{ij} < 0. \end{cases} \tag{4.125}$$

The local maximum $\tilde{u}_i^{\max}$ and minimum $\tilde{u}_i^{\min}$ are defined as in (4.87) and the corresponding solution increments are assembled in a loop over edges using (4.86).

The symmetric limiting strategy (4.122)–(4.125) guarantees that the second step (4.117) of the generic FEM-FCT algorithm is positivity-preserving since

$$\tilde{u}_i^{\min} = \tilde{u}_i + Q_i^- \leq \bar{u}_i \leq \tilde{u}_i + Q_i^+ = \tilde{u}_i^{\max}.$$

Remark 4.21. It is worthwhile to set $R_i^{\pm} := 1$ if Dirichlet boundary conditions are imposed at node i and, therefore, the value of u_i is invariant to the choice of α_{ij}.

Remark 4.22. The need for a repeated evaluation of the nodal correction factors $R_i^{\pm}$ can be avoided if the limited antidiffusive flux $\bar{f}_{ij}$ is redefined as follows [258]

$$\bar{f}_{ij} = minmod\{f_{ij}, \bar{\alpha}_{ij}^n d_{ij}^n (u_i^n - u_j^n)\},$$

where the correction factors $\bar{\alpha}_{ij}^n$ may exceed 1. They are calculated at the beginning of each time step using Zalesak's limiter with $R_i^{\pm} = Q_i^{\pm}/P_i^{\pm}$ instead of (4.124).

The upper and lower bounds $Q_i^{\pm}$ are of the form (4.39), where $q_i^{\pm} = \frac{m_i}{\Delta t}$. The appearance of the time step in the denominator turns out to be a blessing or a curse, depending on the purpose of simulation. On the one hand, the constraints become less restrictive and, consequently, a larger portion of the raw antidiffusive flux f_{ij} is retained as the time step is refined. This makes FCT the method of choice for transient computations. On the other hand, the use of large Δt results in a loss of accuracy, and severe convergence problems may occur in the steady state limit.

A well-known problem associated with flux correction of FCT type is clipping [41, 355]. Since the sum of limited antidiffusive fluxes is forced to become local extremum diminishing, existing peaks lose a little bit of amplitude during each time step. TVD-like methods also fail to recognize smooth extrema, while geometric schemes based on ENO/WENO reconstruction can handle them fairly well [20].

Another infamous byproduct of flux limiting is known as terracing. It manifests itself in a distortion of smooth profiles and represents 'an integrated, nonlinear effect of residual phase errors' [265] or, loosely speaking, 'the ghosts of departed ripples' [41].

Remark 4.23. As shown in [200, 203], it is possible to incorporate accepted antidiffusion into the auxiliary solution $\tilde{u}$ so that only the rejected portion of the flux f_{ij} needs to be constrained in the next cycle. This simplifies the task of the flux limiter and enables it to remove more artificial diffusion. The resulting iterative FEM-FCT algorithm yields a crisp resolution of steep gradients and alleviates peak clipping but might aggravate terracing and slow down the convergence of outer iterations.

4.4.2.3 Slack Bounds

Zalesak's limiter is designed to accept as much antidiffusion as the reference solution $\tilde{u}$ can accommodate. Instead, it is possible to relax the upper/lower bounds $Q_i^{\pm}$ and adjust the time step if this is necessary to satisfy a CFL-like condition. For example, the local extrema of u^n can be used to define $Q_i^{\pm}$ as follows [192]

$$Q_i^+ = \frac{m_i - m_{ii}}{\Delta t}(u_i^{\max} - u_i^n), \qquad Q_i^- = \frac{m_i - m_{ii}}{\Delta t}(u_i^{\min} - u_i^n), \tag{4.126}$$

where $m_i - m_{ii} = \sum_{j \neq i} m_{ij}$ is the difference between the diagonal entries of the consistent and lumped mass matrices. The resulting correction factors α_{ij} will typically be smaller than those obtained with (4.123). However, the difference between the solutions produced by the two versions will shrink and eventually vanish as the time step is refined. As soon as Δt becomes sufficiently small, the accuracy of the results depends solely on the resolving power of the underlying high-order scheme.

The use of slack bounds (4.126) eliminates the need for evaluating an intermediate solution of low order. Combining (4.116) and (4.117), one obtains

$$M_L \bar{u} = [M_L + (1-\theta)\Delta t L^n]u^n + \Delta t\, \bar{f}(u^{(m)}, u^n).$$

Since the sum of limited antidiffusive fluxes satisfies estimate (4.94) with $u = u^n$ and $q_i^{\pm} = (m_i - m_{ii})/\Delta t$, the fully discrete scheme is positivity-preserving if [258]

$$m_{ii} + (1-\theta)\Delta t l_{ii}^n \geq 0, \qquad 0 \leq \theta < 1.$$

This CFL-like condition is more restrictive than (4.119). To rectify this, the upper and lower bounds (4.126) can be defined implicitly using u^{n+1} rather than u^n.

4.4.3 Flux Linearization Techniques

A major drawback of the nonlinear FEM-FCT algorithm is the need to recompute the raw antidiffusive fluxes (4.114) and the correction factors (4.125) after every solution update. The fixed point iteration method (4.115) keeps intermediate solutions positivity-preserving and the convergence of inner iterations is fast owing to the M-matrix property of the low-order preconditioner. Unfortunately, the lagged treatment of limited antidiffusion results in slow convergence of outer iterations. At large time steps, as many as 50 sweeps may be required to obtain a fully converged solution (see the numerical examples below). The lumped-mass version, which corresponds to (4.114) with $m_{ij} = 0$, converges faster but is not to be recommended for strongly time-dependent problems. The convergence of outer iterations can be accelerated using a discrete Newton method [258]. However, the costly assembly of the (approximate) Jacobian operator is unlikely to pay off in transient computations that call for the use of small time steps and, in many cases, explicit algorithms.

A suitable linearization technique can simplify a FEM-FCT algorithm and make it much more efficient. For instance, the implicit part of (4.114) can be evaluated using the solution $u^H \approx u^{n+1}$ of the high-order system ($\alpha_{ij} \equiv 1$) given by

$$[M_C - \theta\Delta t K^{n+1}]u^H = [M_C + (1-\theta)\Delta t K^n]u^n. \tag{4.127}$$

In this case, the right-hand side of system (4.118) needs to be assembled just once per time step. If the underlying continuous problem is linear, then the left-hand side matrix does not change either, and a single iteration of the form (4.118) is required to obtain the flux-corrected end-of-step solution u^{n+1}. Thus, the computational effort reduces to one call of Zalesak's limiter and two linear systems per time step [205, 258]. If the governing equation is nonlinear, so are the two systems to be solved.

In practice, it is usually much more expensive to solve (4.127) than (4.118). The main reason is the lack of the M-matrix property which has an adverse effect on the behavior of linear solvers. As the time step increases, inner iterations may fail to converge, even if advanced linear algebra tools are employed. A robust alternative to the brute-force approach is to compute the high-order predictor u^H using fixed-point iteration (4.115) with $\alpha_{ij} \equiv 1$. However, this linearization strategy is rather inefficient since the flux-limited version of (4.115) tends to converge faster [258].

Another possibility is to linearize f_{ij} about the solution of the low-order system

$$[M_L - \theta\Delta t L^{n+1}]u^L = [M_L + (1-\theta)\Delta t L^n]u^n. \tag{4.128}$$

Unlike (4.127), this linear system can be solved efficiently but the flux-corrected solution u^{n+1} computed with (4.116)–(4.118) turns out too diffusive (see [337], Section 5.2). Moreover, it differs from u^H even if Zalesak's limiter returns $\alpha_{ij} \equiv 1$.

Instead, the flux f_{ij} can be linearized about the solution of a nonlinear system

$$[M_L - \theta\Delta t \bar{K}^{n+1}]\bar{u}^{n+1} = [M_L + (1-\theta)\Delta t \bar{K}^n]u^n \tag{4.129}$$

associated with a flux correction scheme of the form (4.80)–(4.82) which is supposed to be less accurate but more efficient than the nonlinear FEM-FCT algorithm. Since the raw antidiffusive flux is replaced by an accurate and smooth approximation, no ripples are typically generated if the optional prelimiting step is omitted.

4.4.4 Predictor-Corrector Algorithms

After years of research aimed at making implicit FEM-FCT schemes more efficient, the author has come to favor predictor-corrector methods in which the computation of a tentative solution u^L is followed by an explicit flux correction step [196]

$$M_L u^{n+1} = M_L u^L + \Delta t\, \bar{f}(u^L, u^n), \tag{4.130}$$

as in the case of classical diffusion-antidiffusion methods based on FCT [42, 82].

The low-order predictor u^L can be calculated using (4.128) or any other time discretization of (4.10), e.g., the explicit TVD Runge-Kutta method (3.97)–(3.98). The raw antidiffusive fluxes for the flux correction step (4.130) are defined as

$$f_{ij} = m_{ij}(\dot{u}_i^L - \dot{u}_j^L) + d_{ij}^{n+1}(u_i^L - u_j^L), \tag{4.131}$$

where $\dot{u}^L$ denotes a finite difference approximation of the time derivative. This quantity can be computed, e.g., using the leapfrog method as applied to (4.6)

$$\dot{u}^L = M_C^{-1}[K^{n+1}u^L]. \tag{4.132}$$

Since the inverse of M_C is full, let successive approximations to $\dot{u}^L$ be computed using Richardson's iteration preconditioned by the lumped mass matrix [85]

$$\dot{u}^{(m+1)} = \dot{u}^{(m)} + M_L^{-1}[K^{n+1}u^L - M_C\dot{u}^{(m)}], \qquad m = 0, 1, \ldots \tag{4.133}$$

starting with $\dot{u}^{(0)} = 0$ or $\dot{u}^{(0)} = (u^L - u^n)/\Delta t$. Convergence is typically very fast (1-5 iterations are enough) since the consistent mass matrix M_C is well-conditioned.

As an alternative to iterating until (4.133) converges, the first lumped-mass approximation $\dot{u}^{(1)} = M_L^{-1}[K^{n+1}u^L]$ or the corresponding low-order solution

$$\dot{u}^L = M_L^{-1}[L^{n+1}u^L] \tag{4.134}$$

can be used to predict the raw antidiffusive flux f_{ij}. This technique yields a smooth but diffusive approximation to the time derivative. Its accuracy can be enhanced using an upwind-biased flux limiter applied to (4.131) with $\dot{u}^L = 0$. The result is

$$\dot{u}^L = M_L^{-1}[\bar{K}^{n+1}u^L], \tag{4.135}$$

where $\bar{K}$ is the nonlinear operator with built-in antidiffusion. Since the computation of $\dot{u}^L$ is performed just once per time step, the associated overhead cost is acceptable.

The 'high-order' solution u^H produced by (4.130)–(4.131) with $\alpha_{ij} \equiv 1$ satisfies

$$M_L u^H = M_L u^L + \Delta t[(M_L - M_C)\dot{u}^L - D^{n+1}u^L],$$

where the vector of approximate time derivatives $\dot{u}^L$ is given by (4.132), (4.134), or (4.135). This u^H is typically less oscillatory than the solution of (4.127).

The above linearization strategy offers a number of significant advantages. First, the low-order predictor u^L can be calculated by an arbitrary (explicit or implicit) time-stepping method. In the case of an implicit algorithm, iterative solvers are fast due to the M-matrix property of the low-order operator. Second, the leapfrog time discretization of the antidiffusive flux is second-order accurate with respect to the time level t^{n+1} on which u^L and f_{ij} are defined. Third, instead of three different solutions (u^n, u^{n+1}, and $\tilde{u}$) only the smooth predictor u^L is involved in the computation of f_{ij} and α_{ij} for the correction step (4.130). No prelimiting is required for the lumped-mass version ($\dot{u}^L = 0$) since the flux $f_{ij}^L = d_{ij}^{n+1}(u_i^L - u_j^L)$ is truly antidiffusive. For $\dot{u}^L \neq 0$, it can serve as the upper bound for minmod prelimiting (4.121).

4.4.5 Positive Time Integrators

The efficiency of implicit FEM-FCT schemes for unsteady transport problems is hampered not only by their nonlinearity but also by the severe time step restriction (4.119). The second-order accurate Crank-Nicolson version ($\theta = \frac{1}{2}$) is linearly stable for arbitrary time steps but may produce oscillatory solutions if (4.119) does not hold. The backward Euler method ($\theta = 1$) is unconditionally positivity-preserving (UPP) but only first-order accurate in time. An analog of the Godunov theorem [123] states that no linear time integration scheme of higher order can be UPP [126]. The unsatisfactory state of affairs can be rectified by selecting the optimal degree of implicitness $\theta_{ij} \in [0,1]$ for each node pair so as to localize the CFL-like condition [337, 338] or to enforce monotonicity constraints by means of *time limiters* [96].

For a variable-order $\theta-$scheme to stay conservative, not only the diffusive and antidiffusive terms but also the underlying high-order discretization must be expressed in terms of internodal fluxes. The derivation of a conservative flux decomposition for the Galerkin finite element scheme was addressed in Section 2.1.6. After the discretization in time, the corresponding low-order scheme can be written as

$$m_i u^L = m_i u^n - \Delta t \sum_j \hat{f}_{ij}, \tag{4.136}$$

where the numerical flux $\hat{f}_{ij}$ consists of a centered convective part augmented by physical and/or numerical diffusion. Following the notation of previous chapters, the subscript $j = i$ is reserved for fluxes across the boundary of the domain.

Let the discretization in time be performed by the *local θ-scheme* [337, 338]

$$\hat{f}_{ij} = \theta_{ij}\hat{f}_{ij}^{n+1} + (1-\theta_{ij})\hat{f}_{ij}^{n}, \qquad 0 \leq \theta_{ij} \leq 1. \tag{4.137}$$

Since $\hat{f}_{ij}$ is a convex average of $\hat{f}_{ij}^{n+1}$ and $\hat{f}_{ij}^{n}$, a numerical scheme of this form is conservative even if the parameters θ_{ij} vary in space and time. Instead of adjusting the time step size so as to preserve positivity with a constant value of θ, the degree of implicitness can be tailored to the local Courant number and/or to a given solution.

The CFL-like condition (4.119) suggests the following linear combination of the Crank-Nicolson and backward Euler time integration methods, cf. [337, 338]

$$\theta_{ij} = \min\{\theta_i, \theta_j\}, \qquad \theta_i = \max\left\{\frac{1}{2}, 1 + \min\left\{0, \frac{m_i}{\Delta t l_{ii}^n}\right\}\right\}. \tag{4.138}$$

This choice is to be recommended for problems in which local mesh refinement and/or a nonuniform velocity field result in a strong variation of the Courant number.

Furthermore, a tentative solution u^L calculated by the backward Euler method ($\theta_{ij} \equiv 1$) or using the local θ-scheme (4.137)–(4.138) can be improved by including

$$\hat{f}_{ij} - \hat{f}_{ij}^{n+1/2} = \left(\theta_{ij} - \frac{1}{2}\right)(\hat{f}_{ij}^{n+1} - \hat{f}_{ij}^{n})$$

into the raw antidiffusive flux f_{ij} for a nonlinear or linearized FEM-FCT algorithm.

A more sophisticated approach to the design of time limiters for implicit high-resolution schemes is considered in [96]. The nonlinear L-TRAP scheme proposed therein is based on the following definition of the implicitness parameters

$$\theta_{ij} = \frac{\theta_i + \theta_j}{2}, \qquad \theta_i = 1 - \frac{1}{2}\bar{r}_i,$$

where $\bar{r}_i$ is a correction factor that controls the jump of the temporal slope ratio at a given node i. A linear and nonlinear stability analysis is performed for the first-order upwind discretization of the linear convection equation in 1D. The numerical results for scalar conservation laws and hyperbolic systems confirm that a linear time integration scheme of high order can be forced to satisfy monotonicity constraints using an extension of tools and concepts originally developed for space discretizations.

4.4.6 Numerical Examples

A properly designed high-resolution scheme should be (i) at least second-order accurate for smooth data and (ii) capable of resolving arbitrary small-scale features without excessive smearing or steepening of the transported profiles. To evaluate the accuracy and efficiency of FEM-FCT, we consider a suite of time-dependent benchmark problems which are representative and challenging enough to indicate how the methods under investigation would behave in real-life applications [196].

In what follows, a comparative study is performed for both explicit and implicit FEM-FCT algorithms. The ones based on the TVD Runge-Kutta (3.97)–(3.98), Crank-Nicolson ($\theta = \frac{1}{2}$), and backward Euler ($\theta = 1$) time-stepping schemes will be

referred to as RK-FCT-L, CN-FCT-L, and BE-FCT-L, respectively. The last digit in these abbreviations refers to the type of flux linearization, if any. The fully nonlinear version from Section 4.4.1 is assigned the number $L = 1$, whereas $L = 2$ stands for the linearization about the solution u^H of system (4.127). Both of these algorithms require the use of an implicit $\theta-$scheme, so RK-FCT-1 and RK-FCT-2 are not available. The predictor-corrector strategy (4.131) corresponds to $L = 3$ if $\dot{u}^L$ is given by (4.132), while $L = 4$ if approximation (4.134) is adopted. In the former case, the consistent mass matrix M_C is 'inverted' using 5 iterations of the form (4.133).

By default, systems (4.118), (4.127), and (4.128) are solved by the Gauss-Seidel method. BiCGSTAB with ILU preconditioning and Cuthill-McKee renumbering is invoked if convergence fails because of too large a time step. All computations are performed on a laptop computer using the Intel Fortran Compiler for Linux.

In this section, the following quantities serve as the measure of the difference between the analytical solution u and a given numerical approximation u_h

$$E_1 = \sum_i m_i |u(\mathbf{x}_i) - u_i| \approx ||u - u_h||_1, \tag{4.139}$$

$$E_2 = \sqrt{\sum_i m_i |u(\mathbf{x}_i) - u_i|^2} \approx ||u - u_h||_2. \tag{4.140}$$

The goal of the below numerical study is to investigate how the above errors and the CPU times depend on the mesh size h, time step Δt, and linearization technique.

4.4.6.1 Solid Body Rotation

Solid body rotation illustrates the ability of a numerical scheme to transport initial data without distortion [218, 355]. Consider the linear convection equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0 \qquad \text{in } \Omega = (0,1) \times (0,1) \tag{4.141}$$

which is hyperbolic and of the form (4.1). The incompressible velocity field

$$\mathbf{v}(x,y) = (0.5 - y, x - 0.5) \tag{4.142}$$

corresponds to a counterclockwise rotation about the center of the square domain Ω. Homogeneous Dirichlet boundary conditions are prescribed at the inlets.

The exact solution to (4.141)–(4.142) depends solely on the initial state u_0 and reproduces it exactly after each full revolution. Hence, the challenge of this test is to preserve the shape of u_0 as accurately as possible. Following LeVeque [218], we consider a slotted cylinder, a sharp cone, and a smooth hump (Fig. 4.6). Initially, the geometry of each body is given by a function $G(x,y)$ defined within the circle

$$r(x,y) = \frac{1}{r_0}\sqrt{(x - x_0)^2 + (y - y_0)^2} \leq 1$$

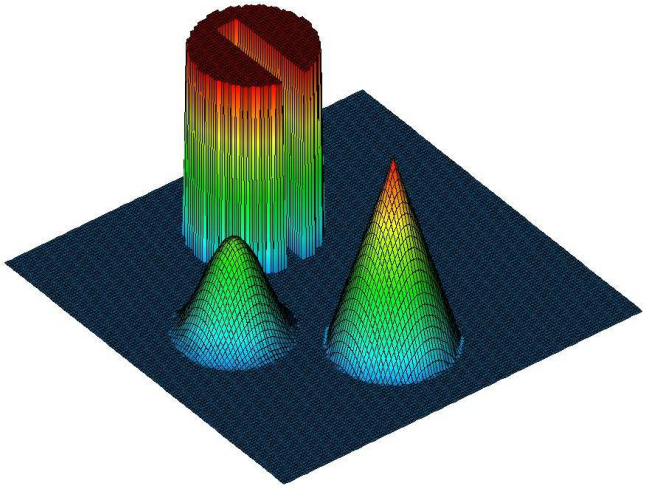


Fig. 4.6 Initial data / exact solution at the final time $t = 2\pi$.

of radius $r_0 = 0.15$ centered at a certain point with Cartesian coordinates (x_0, y_0).

For the slotted cylinder, the reference point is $(x_0, y_0) = (0.5, 0.75)$ and [218]

$$G(x,y) = \begin{cases} 1 & \text{if } |x - x_0| \geq 0.025 \text{ or } y \geq 0.85, \\ 0 & \text{otherwise.} \end{cases}$$

The conical body is centered at $(x_0, y_0) = (0.5, 0.25)$ and its geometry is defined by

$$G(x,y) = 1 - r(x,y).$$

The peak of the hump is located at $(x_0, y_0) = (0.25, 0.5)$ and the shape function is

$$G(x,y) = \frac{1 + \cos(\pi r(x,y))}{4}.$$

In the rest of the domain Ω, the solution of equation (4.141) is initialized by zero.

Figure 4.7 displays the results produced by the four Crank-Nicolson FEM-FCT schemes after one full revolution ($t = 2\pi$). These numerical solutions were computed on a uniform mesh of bilinear elements with $h = 1/128$ and $\Delta t = 10^{-3}$. Prelimiting of the form (4.120) was performed for CN-FCT-1 through CN-FCT-3, while CN-FCT-4 was found to produce ripple-free solutions even without prelimiting.

The diagrams in Fig. 4.8 depict the convergence history for (4.139) and the total CPU time as a function of the mesh size h. It can be seen that the linearized schemes CN-FCT-2 and CN-FCT-3 are almost as accurate as CN-FCT-1. The norms of the error for CN-FCT-4 are larger on all meshes but decrease at a faster rate than those for CN-FCT-3. The effective order of accuracy $p = \log_2(E_1(h)/E_1(h/2))$ estimated

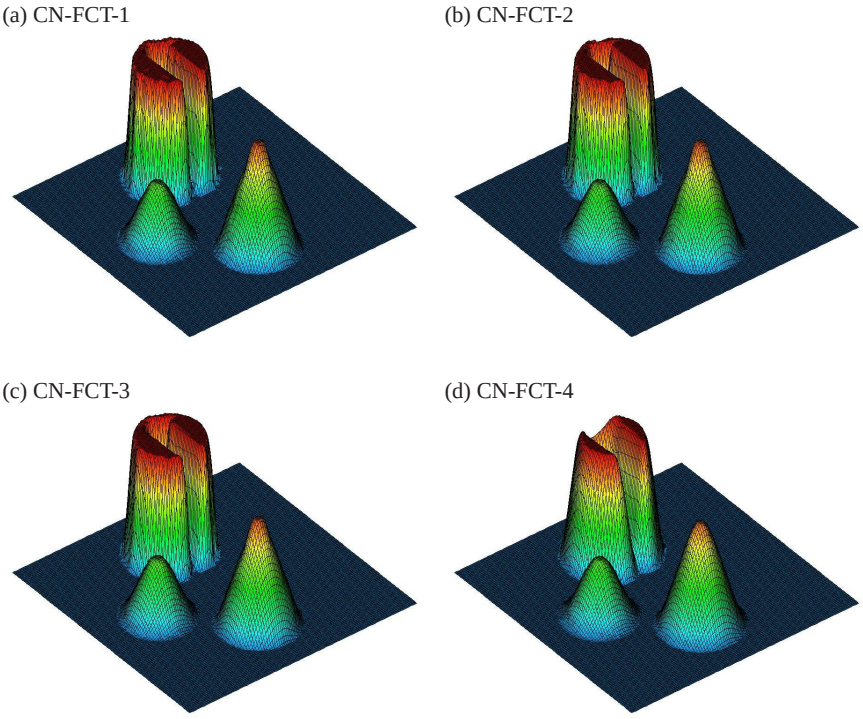


Fig. 4.7 Solid body rotation, CN-FCT schemes, $\mathscr{Q}_1$ elements, $\Delta t = 10^{-3}$, $t = 2\pi$.

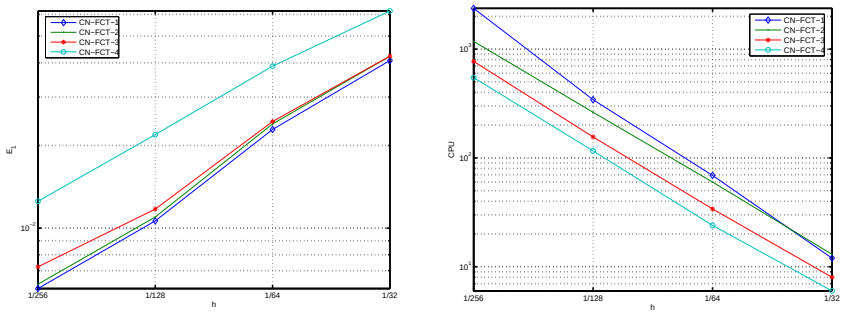


Fig. 4.8 Solid body rotation, convergence history and CPU times for CN-FCT.

using $h = 1/128$ equals $\{0.82, 0.81, 0.70, 0.81\}$ for CN-FCT-1 through CN-FCT-4, respectively. As in the case of steady convection, the presence of a discontinuous profile is the reason why the errors decrease so slowly with mesh refinement.

A comparison of the CPU times illustrates the gain of efficiency offered by the predictor-corrector algorithms CN-FCT-3 and CN-FCT-4. The cost of CN-FCT-2 is significantly higher and even exceeds that for CN-FCT-1 on the coarsest mesh. This

is explained by the slow convergence of the linear solver for the ill-conditioned high-order system (4.127). In the case of CN-FCT-1, defect correction was performed as long as required to make the normalized residual smaller than the tolerance 10^{-5}.

The results in Tables 4.4–4.6 illustrate the performance of different time-stepping methods and flux linearization techniques. The local Courant number $\nu = |\mathbf{v}|\frac{\Delta t}{h}$ varies between zero and $\nu_{\max} = \frac{\Delta t}{2h}$ in each test. The errors and CPU times are measured for the numerical solutions computed on a Cartesian mesh with $h = 1/128$. The entry in the last column is the average number of outer iterations required to reach the above tolerance for CN-FCT-1. In the case of linearized FCT schemes, there is no need for iterative defect correction anymore. This is why NIT equals 1.

For time steps as small as $\Delta t = 10^{-3}$, the second-order accurate RK-FCT and CN-FCT schemes produce essentially the same results (see Table 4.4). The first-order temporal accuracy of BE-FCT is noticeable but spatial errors are clearly dominant. Furthermore, it can be seen that the predictor-corrector approach to flux linearization reduces the difference between the cost (per time step) of explicit and implicit FEM-FCT algorithms. A further gain of efficiency can be achieved using a Jacobi-like iterative method to update the discrete solution in a fully explicit way.

Table 4.5 demonstrates that the errors for BE-FCT become disproportionately large as compared to those for RK-FCT and CN-FCT as Δt is increased by a factor of 10. Since the backward Euler method is equivalent to the first-order backward difference approximation of the time derivative, it turns out overly diffusive at large time steps. The main advantage of BE-FCT is its unconditional stability and positivity preservation for arbitrary time steps. The poor accuracy of the results in Table 4.6 indicates that no time-accurate solutions can be obtained with time steps that violate the CFL stability condition in the whole domain. However, if the Courant number ν exceeds unity only in small subdomains, where the velocity is unusually large and/or adaptive mesh refinement is performed, then a local loss of accuracy is acceptable if the use of large time steps would make the computation much more efficient.

No results for RK-FCT and linearized CN-FCT are presented in Table 4.6 since these schemes turn out to be unstable for the time step $\Delta t = 10^{-1}$ that exceeds

Table 4.4 Solid body rotation: results for $h = 1/128$, $\Delta t = 10^{-3}$, $\nu_{\max} = 0.064$.

	E_1	E_2	CPU	NIT
RK-FCT-3	1.1754e-2	5.9882e-2	127	1.0
RK-FCT-4	2.1913e-2	8.3066e-2	84	1.0
CN-FCT-1	1.0622e-2	5.6411e-2	343	3.5
CN-FCT-2	1.0980e-2	5.7370e-2	263	1.0
CN-FCT-3	1.1729e-2	5.9818e-2	156	1.0
CN-FCT-4	2.1902e-2	8.3045e-2	116	1.0
BE-FCT-1	1.9818e-2	7.5392e-2	280	2.5
BE-FCT-2	2.0069e-2	7.5862e-2	255	1.0
BE-FCT-3	2.1131e-2	7.9686e-2	155	1.0
BE-FCT-4	2.7443e-2	9.2886e-2	110	1.0

Table 4.5 Solid body rotation: results for $h = 1/128,\ \Delta t = 10^{-2},\ \nu_{\max} = 0.64$.

	E_1	E_2	CPU	NIT
RK-FCT-3	1.8289e-2	7.5075e-2	13	1.0
RK-FCT-4	2.4417e-2	8.8419e-2	8	1.0
CN-FCT-1	1.2867e-2	6.2870e-2	173	19.7
CN-FCT-2	1.3552e-2	6.5033e-2	27	1.0
CN-FCT-3	1.7018e-2	7.3535e-2	17	1.0
CN-FCT-4	2.3676e-2	8.7242e-2	13	1.0
BE-FCT-1	5.5943e-2	1.3651e-1	155	15.9
BE-FCT-2	5.6119e-2	1.3675e-1	36	1.0
BE-FCT-3	5.7247e-2	1.3966e-1	17	1.0
BE-FCT-4	5.8198e-2	1.4102e-1	13	1.0

Table 4.6 Solid body rotation: results for $h = 1/128,\ \Delta t = 10^{-1},\ \nu_{\max} = 6.4$.

	E_1	E_2	CPU	NIT
CN-FCT-1	7.3711e-2	1.6587e-1	54	35.0
BE-FCT-1	1.0519e-1	2.0244e-1	92	51.3
BE-FCT-3	1.0504e-1	2.0250e-1	4	1.0
BE-FCT-4	1.0506e-1	2.0251e-1	3	1.0

the upper bound imposed by the CFL-like condition (4.119). CN-FCT-1 remains stable and more accurate than BE-FCT but the solution is no longer guaranteed to be positivity-preserving. The results for BE-FCT-2 are missing due to the failure of the BiCGSTAB solver for the high-order system (4.127). At large time steps, the cost of a nonlinear FEM-FCT algorithm becomes very high due to slow convergence of inner and outer iterations. In the case of BE-FCT-1, more than 50 flux/defect correction steps are required to advance the solution from one time level to the next in Table 4.6. BE-FCT-3 or BE-FCT-4 produce the same results 30 times faster.

4.4.6.2 Swirling Flow

In the next example, we consider the same equation, the same domain, and the same initial data as before but the velocity field is given by the formula [218]

$$\mathbf{v}(x,y,t) = (\sin^2(\pi x)\sin(2\pi y)g(t), -\sin^2(\pi y)\sin(2\pi x)g(t)), \tag{4.143}$$

where $g(t) = \cos(\pi t/T)$ on the time interval $0 \le t \le T$. This incompressible velocity field describes a swirling deformation flow that provides a more severe test for numerical schemes than solid body rotation with a constant angular velocity.

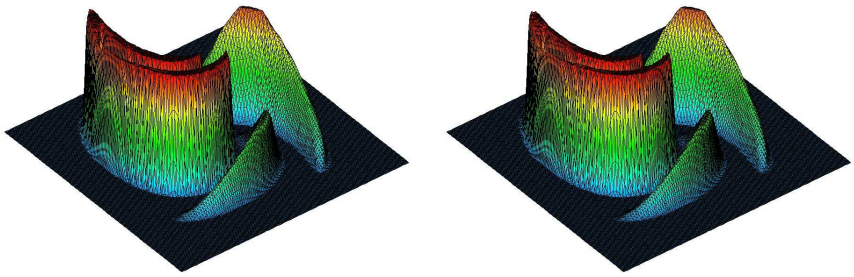
Since the velocity $\mathbf{v}$ vanishes on the boundaries of the domain $\Omega = (0,1)\times(0,1)$, no boundary conditions need to be prescribed in the case of pure convection. The function $g(t)$ is designed so that the flow slows down and eventually reverses its

direction as time evolves. The exact solution at $t = T$ reproduces the initial profile depicted in Fig. 4.6 although the flow field has a fairly complicated structure.

The numerical solutions in Fig. 4.9–4.10 were computed by CN-FCT using linear finite elements and $\Delta t = 10^{-3}$. The underlying triangular mesh has the same vertices and twice as many cells as its quadrilateral counterpart with $h = 1/128$. The snapshots in Fig. 4.9 correspond to the time of maximum deformation $t = T/2$ and those in Fig. 4.10 to the final time $T = 1.5$. Although the solution undergoes significant deformations in the course of simulation, the shape of the initial data is recovered fairly well. As in the case of solid body rotation, erosion of the slotted cylinder is stronger for CN-FCT-4 than for the other three schemes. On the other hand, the artificial steepening of smooth profiles is alleviated since the linearized antidiffusive flux is smooth and does not need to be prelimited in this particular test.

The error norms and CPU times for all FEM-FCT algorithms as applied to the swirling flow problem are presented in Tables 4.7–4.9. Since the velocity field $\mathbf{v}$ is nonstationary, the coefficients of the discrete operators $K = \{k_{ij}\}$ and $D = \{d_{ij}\}$ need to be updated after each time step. As explained in Chapter 2, the group FEM approximation offers a simple and efficient way to do so. Since matrix assembly claims a larger share of the CPU time, the difference between the cost of explicit

(a) CN-FCT-1 (b) CN-FCT-2



(c) CN-FCT-3 (d) CN-FCT-4

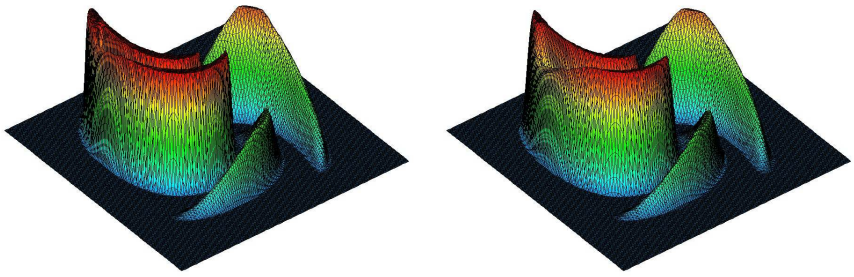


Fig. 4.9 Swirling deformation, CN-FCT schemes, $\mathscr{P}_1$ elements, $\Delta t = 10^{-3}$, $t = 0.75$.

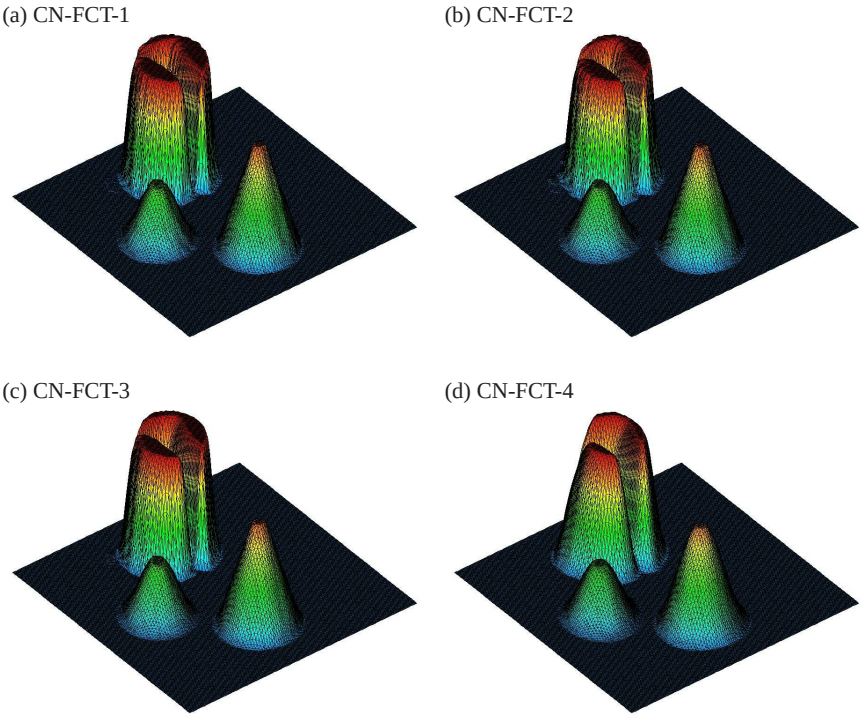


Fig. 4.10 Swirling deformation, CN-FCT schemes, $\mathscr{P}_1$ elements, $\Delta t = 10^{-3}$, $t = 1.5$.

and implicit schemes is smaller than in the case of a stationary velocity field. In Table 4.7, the differences between the CPU times for RK-FCT-4, CN-FCT-4, and BE-FCT-4 are marginal since the convergence of the Gauss-Seidel solver is fast.

At intermediate and large time steps, the convergence of implicit solvers slows down. This is the price to be paid for robustness. Tables 4.8–4.9 summarize the results for $\Delta t = 10^{-2}$ and $\Delta t = 10^{-1}$. Both explicit RK-FCT algorithms turned out unstable, while the linear solver for BE-FCT-2 failed in the latter test. Again, linearization about the low-order predictor u^L proved more efficient than the nonlinear FEM-FCT scheme and the one linearized about the high-order solution u^H. The associated loss of accuracy is acceptable, especially in the case of backward Euler.

4.4.6.3 Convection-Diffusion

To investigate the interplay between physical and numerical diffusion, we apply the four Crank-Nicolson FEM-FCT algorithms to the parabolic equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u - \varepsilon \nabla u) = 0 \qquad \text{in } \Omega = (-1,1) \times (-1,1), \tag{4.144}$$

Table 4.7 Swirling deformation: results for $h = 1/128,\ \Delta t = 10^{-3},\ t = 1.5$.

	E_1	E_2	CPU	NIT
RK-FCT-3	1.4440e-2	6.6023e-2	45	1.0
RK-FCT-4	2.4558e-2	8.9130e-2	36	1.0
CN-FCT-1	1.2043e-2	5.8858e-2	122	5.6
CN-FCT-2	1.3049e-2	6.1370e-2	60	1.0
CN-FCT-3	1.4300e-2	6.5626e-2	50	1.0
CN-FCT-4	2.4493e-2	8.8983e-2	42	1.0
BE-FCT-1	2.4606e-2	8.4485e-2	112	4.8
BE-FCT-2	2.5185e-2	8.5713e-2	60	1.0
BE-FCT-3	2.5334e-2	8.5644e-2	49	1.0
BE-FCT-4	3.1814e-2	1.0039e-1	41	1.0

Table 4.8 Swirling deformation: results for $h = 1/128,\ \Delta t = 10^{-2},\ t = 1.5$.

	E_1	E_2	CPU	NIT
CN-FCT-1	2.2380e-2	8.1277e-2	38	17.9
CN-FCT-2	2.3670e-2	8.4051e-2	10	1.0
CN-FCT-3	2.4119e-2	8.6538e-2	6	1.0
CN-FCT-4	2.8809e-2	9.6268e-2	5	1.0
BE-FCT-1	6.4479e-2	1.4867e-1	53	21.1
BE-FCT-2	6.4621e-2	1.4885e-1	11	1.0
BE-FCT-3	6.3877e-2	1.4760e-1	6	1.0
BE-FCT-4	6.4827e-2	1.4907e-1	5	1.0

Table 4.9 Swirling deformation: results for $h = 1/128,\ \Delta t = 10^{-1},\ t = 1.5$.

	E_1	E_2	CPU	NIT
CN-FCT-1	6.3013e-2	1.3422e-1	12	24.1
CN-FCT-3	6.4958e-2	1.3829e-1	1	1.0
CN-FCT-4	6.3189e-2	1.3556e-1	1	1.0
BE-FCT-1	1.1173e-1	2.0886e-1	11	17.7
BE-FCT-3	1.1155e-1	2.0870e-1	1	1.0
BE-FCT-4	1.1155e-1	2.0870e-1	1	1.0

where $\mathbf{v}(x,y) = (-y,x)$ is the velocity field and $\varepsilon = 10^{-3}$ is the diffusion coefficient.

The initial and boundary conditions are defined using an analytical solution that describes convection and diffusion of a rotating Gaussian hill [214]

$$u(x,y,t) = \frac{1}{4\pi\varepsilon t}\, e^{-\frac{r^2}{4\varepsilon t}}, \qquad r^2 = (x-\hat{x})^2 + (y-\hat{y})^2, \tag{4.145}$$

where $\hat{x}$ and $\hat{y}$ are the time-dependent coordinates of the moving peak

$$\hat{x}(t) = x_0 \cos t - y_0 \sin t, \qquad \hat{y}(t) = -x_0 \sin t + y_0 \cos t.$$

Since $u(x,y,0) = \delta(x_0,y_0)$, where δ is the Dirac delta function, it is worthwhile to start the numerical simulation at $t_0 > 0$. As time goes on, the moving peak is gradually smeared by diffusion and flux limiting becomes redundant. The purpose of this test is to investigate how FEM-FCT algorithms can handle such situations.

Because of phase errors, the peak of an approximate solution u_h may be displaced. Its Cartesian coordinates $\hat{\mathbf{x}}_h = (\hat{x}_h, \hat{y}_h)$ can be estimated as follows

$$\hat{x}_h(t) = \int_\Omega x u_h(x,y,t)\,\mathrm{d}\mathbf{x}, \qquad \hat{y}_h(t) = \int_\Omega y u_h(x,y,t)\,\mathrm{d}\mathbf{x}.$$

The smearing caused by physical and numerical diffusion is quantified via

$$\sigma_h^2(t) = \int_\Omega r_h^2 u_h(x,y,t)\,\mathrm{d}\mathbf{x}, \qquad r_h^2 = (x-\hat{x}_h)^2 + (y-\hat{y}_h)^2.$$

The difference between σ_h^2 and the variance $\sigma^2 = 4\varepsilon t$ of the exact solution (4.145) to the convection-diffusion equation defines the relative dispersion error

$$E_{\mathrm{disp}}(t) = \frac{\sigma_h^2(t) - \sigma^2(t)}{\sigma^2(t)} = \frac{\sigma_h^2(t)}{4\varepsilon t} - 1.$$

Positive values of E_{disp} imply that a given approximate solution is overly diffusive, while negative dispersion errors indicate that some physical diffusion is offset by numerical antidiffusion inherent to a nondissipative space discretization.

As a direct measure of peak clipping, we introduce the relative amplitude error

$$E_{\mathrm{peak}}(t) = \frac{u_h^{\max}(t) - u^{\max}(t)}{u^{\max}(t)} = \frac{u_h^{\max}(t)}{u^{\max}(t)} - 1,$$

where $u^{\max}$ and $u_h^{\max}$ denote the global maxima of the analytical and numerical solutions, respectively. Hence, the value of E_{peak} is positive if the top of the rotating Gaussian hill is too high and negative in the presence of clipping effects.

The numerical experiment begins at $t_0 = \pi/2$ with a peak located at the point $(x_0,y_0) = (0.0, 0.5)$. The initial shape of the Gaussian hill and the solution produced by CN-FCT-4 after one full revolution ($t = 5\pi/2$) are displayed in Fig. 4.11. This simulation was performed on a uniform mesh of bilinear elements using $h = 1/128$ and $\Delta t = 10^{-3}$. Flux correction was applied to the convective part of the discrete transport operator, whereas the diffusive part was left unchanged. While the latter is of nonnegative type (on such a regular mesh), the Galerkin discretization of the convective term is too antidiffusive. Therefore, it is not desirable to minimize the amount of artificial diffusion. On the other hand, physical diffusion may be taken into account if some background dissipation is included in the high-order scheme.

The convergence history and CPU times for $t = 5\pi/2$ and $\Delta t = 10^{-3}$ are presented in Fig. 4.12. Surprisingly enough, the convergence of the nonlinear version slows down as the mesh is refined. On the finest mesh, the solution obtained with CN-FCT-1 is even less accurate than that produced by CN-FCT-4. The difference between the corresponding CPU times is about 50 percent, which is not as much

as in the case of solid body rotation. The convergence behavior of all linearized FEM-FCT schemes is satisfactory. In this test, the estimated order of accuracy $p = \log_2(E_1(h)/E_1(h/2))$ equals $\{1.3, 2.5, 2.4, 2.7\}$ for the solutions computed with CN-FCT-1 through CN-FCT-4 on the two finest meshes. Remarkably, the cheapest algorithm converges at the fastest rate which is higher than second order. It is well known that the presence of the consistent mass matrix makes the 1D Galerkin discretization of pure convection problems fourth-order (!) accurate on a uniform mesh of linear finite elements (see [86], p. 96). This leads to a significant gain of accuracy as compared to centered finite difference or finite volume discretizations.

While fourth-order accuracy is no longer guaranteed for multidimensional problems, nonsmooth data, and nonuniform meshes or velocity fields, mass lumping tends to degrade the accuracy of transient solutions. Likewise, the definition of $\dot{u}^L \neq 0$ has a strong influence on the final solution. The use of a low-order approximation in CN-FCT-4 makes it more efficient but less accurate than CN-FCT-3.

Figure 4.13 shows how the relative dispersion and amplitude errors vary with the mesh size. On the coarsest mesh, all four schemes produce rather diffusive results. Although the peak height predicted by CN-FCT-3 turns out to be very accurate, further mesh refinement reveals that this is just a coincidence. On finer meshes, the relative errors E_{disp} and E_{peak} approach zero in a monotone fashion. The slightly antidiffusive behavior of CN-FCT-1 through CN-FCT-3 is due to the nondissipative nature of the underlying Galerkin discretization. While peak clipping is pronounced on coarse meshes, this hardly justifies the use of *ad hoc* adjustments that increase the complexity of the algorithm and/or may cause a loss of positivity preservation.

(a) initial peak at $t_0 = \frac{\pi}{2}$ (b) CN-FCT solution at $t = \frac{5\pi}{2}$

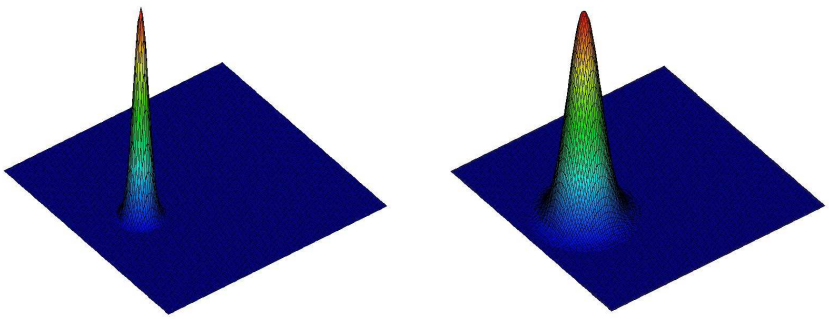


Fig. 4.11 Snapshots of the Gaussian hill: $\mathscr{Q}_1$ elements, $h = 1/128,\ \Delta t = 10^{-3}$.

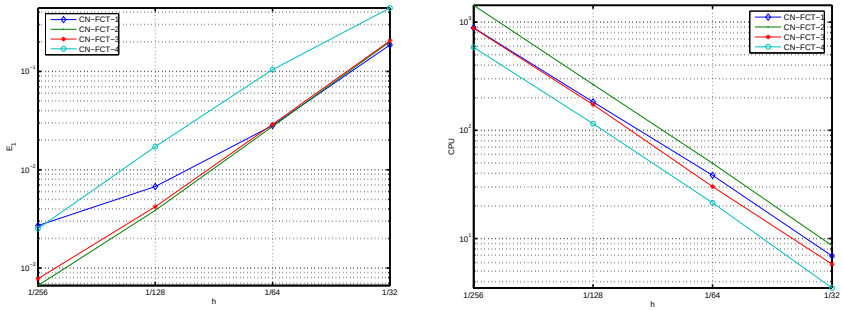


Fig. 4.12 Gaussian hill: convergence history and CPU times for CN-FCT.

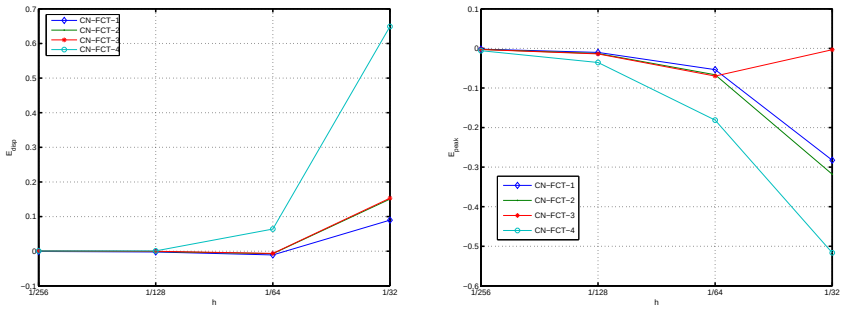


Fig. 4.13 Gaussian hill: dispersion (left) and amplitude (right) errors for CN-FCT.

4.5 Limiting for Diffusion Operators

As a matter of fact, not only convective but also diffusive terms may be the cause of undershoots and overshoots if the computational mesh and/or the diffusion tensor are anisotropic. This is why discrete diffusion operators may also require some kind of flux correction [29, 163, 349]. Monotone finite volume schemes were recently developed for anisotropic diffusion problems on unstructured meshes [215, 221]. However, they are merely positivity-preserving and, generally, do not satisfy the discrete maximum principle. Moreover, their derivation is based on a design philosophy which is not applicable to finite element approximations. The methodology proposed in [222] is based on constrained optimization and requires *a priori* knowledge of the solution bounds. Also, the solution of the constrained optimization problem can become prohibitively expensive as the number of unknowns increases.

In this section, we extend the algebraic flux correction paradigm to anisotropic diffusion problems and enforce the DMP using a symmetric version [204] of the slope limiter based on gradient reconstruction. The resulting discretization is akin to symmetric limited positive (SLIP) schemes [170] but the upper and lower bounds

are given in terms of local maxima and minima, as in the case of FCT algorithms. The numerical study to be presented demonstrates that this kind of slope limiting renders the constrained Galerkin approximation local extremum diminishing and keeps it sufficiently accurate when applied to problems with smooth solutions.

4.5.1 The Galerkin Discretization

Consider an elliptic boundary value problem that describes steady diffusive transport of a scalar quantity u in a multidimensional domain Ω with boundary Γ

$$\begin{cases} -\nabla\cdot(\mathscr{D}\nabla u) = s, & \text{in } \Omega, \\ u = g & \text{on } \Gamma, \end{cases} \tag{4.146}$$

where $\mathscr{D}(\mathbf{x})$ is a (possibly anisotropic) diffusion tensor and $s(\mathbf{x})$ is a source or sink.

The finite element approach to solution of (4.146) is based on the weak form

$$a(w,u) = (s,w), \tag{4.147}$$

where $a(\cdot,\cdot)$ is a bilinear form, u is the weak solution, w is any admissible test function, and $(\cdot,\cdot)$ is the usual shorthand notation for the scalar product in $L_2(\Omega)$

$$(w,u) = \int_\Omega wu\,dx.$$

The bilinear form associated with the weak form of the model problem (4.146) reads

$$a(w,u) = \int_\Omega \nabla w\cdot(\mathscr{D}\nabla u)\,d\mathbf{x}. \tag{4.148}$$

Given a set of finite element basis functions $\{\varphi_i\}$, substitution of $u \approx \sum_j u_j\varphi_j$ and $w = \varphi_i$ into (4.147) yields the standard Galerkin discretization

$$\sum_j a(\varphi_i,\varphi_j)u_j = (\varphi_i,s), \qquad \forall i.$$

This is a linear system of the form $Au = b$ with $A = \{a_{ij}\}$ and $b = \{b_i\}$ given by

$$a_{ij} = a(\varphi_i,\varphi_j), \qquad b_i = (\varphi_i,s). \tag{4.149}$$

Due to (4.148) the extended matrix A is symmetric with zero row and column sums

$$a_{ij} = a_{ji}, \qquad \sum_i a_{ij} = \sum_j a_{ij} = 0. \tag{4.150}$$

However, this discrete diffusion operator may fail to be of *nonnegative type* in the sense of Definition 3.13 if the given mesh and/or the tensor $\mathscr{D}$ are anisotropic. To prevent a violation of the DMP, some coefficients of A may need to be adjusted.

4.5.2 Positive-Negative Splitting

As before, the process of algebraic flux correction starts with conservative elimination of entries that have wrong sign. Now the stiffness matrix A resides in the left-hand side of the linear system, so the 'bad' part is $A^+ = \{a_{ij}^+\}$ with [311]

$$a_{ij}^+ := \max\{0, a_{ij}\}, \qquad \forall j \neq i. \tag{4.151}$$

The diagonal coefficients of A^+ are defined so that it has zero row and column sums

$$a_{ii}^+ := -\sum_{j\neq i} a_{ij}^+, \qquad \forall i. \tag{4.152}$$

The complement $A^- := A - A^+$ represents the 'good' part of the stiffness matrix

$$A = A^+ + A^-. \tag{4.153}$$

By virtue of (4.150)–(4.152), the $i-$th component of the vector $A^\pm u$ is given by

$$(A^\pm u)_i = \sum_j a_{ij}^\pm u_j = \sum_{j\neq i} a_{ij}^\pm (u_j - u_i)$$

and can be expressed in terms of numerical fluxes from one node into another

$$(A^\pm u)_i = -\sum_{j\neq i} f_{ij}^\pm, \qquad f_{ij}^\pm = a_{ij}^\pm (u_i - u_j). \tag{4.154}$$

The fluxes f_{ij}^- and f_{ij}^+ represent the diffusive and antidiffusive edge contributions to row i, respectively. The $i-$th element of the residual vector $r = b - Au$ is

$$r_i = b_i + \sum_{j\neq i} (f_{ij}^+ + f_{ij}^-).$$

To enforce monotonicity constraints, the raw antidiffusive flux f_{ij}^+ is replaced by

$$\bar{f}_{ij}^+ = a_{ij}^+ \bar{s}_{ij} = \alpha_{ij} f_{ij}^+, \tag{4.155}$$

where $\bar{s}_{ij}$ denotes the limited slope and $\alpha_{ij} \in [0,1]$ is the correction factor such that

$$\bar{s}_{ij} = \alpha_{ij}(u_i - u_j).$$

Assuming that the nonnegative-type matrix A^- is irreducible, Corollary 3.12 states that the discrete maximum principle holds at least for $\alpha_{ij} \equiv 0$ and $\bar{s}_{ij} \equiv 0$. However, the so-defined 'low-order' scheme may turn out to be inconsistent since the elimination of positive off-diagonal entries a_{ij}^+ from a discrete diffusion operator introduces a perturbation error of one order lower than that for a convective flux [193]. If the antidiffusive part A^+ of the stiffness matrix is omitted, the modified scheme may converge to a wrong solution as the mesh size h is refined. Thus, an antidiffusive correction is a must and it is essential to guarantee that $\bar{f}_{ij}^+ \to f_{ij}^+$ as $h \to 0$.

4.5.3 Symmetric Slope Limiter

Since it is rather difficult to maintain/prove consistency within the framework of a purely algebraic approach, flux correction is performed using a linearity-preserving slope limiter based on gradient reconstruction. Due to the symmetry of the stiffness matrix, nodes i and j should be treated equally. For example, a modification of formula (4.92) leads to the following definition of the limited slope [204]

$$\bar{s}_{ij} = \begin{cases} \min\{2\gamma_{ij}(u_i^{\max} - u_i), u_i - u_j, 2\gamma_{ji}(u_j - u_j^{\min})\}, & \text{if } u_i > u_j, \\ \max\{2\gamma_{ij}(u_i^{\min} - u_i), u_i - u_j, 2\gamma_{ji}(u_j - u_i^{\max})\}, & \text{if } u_i < u_j. \end{cases} \tag{4.156}$$

The nonnegative coefficients γ_{ij} and γ_{ji} are obtained from a LED estimate of the form (4.90). The corresponding unlimited slopes s_{ij} and s_{ji} are given by (4.104), where the nodal gradients are approximated using the lumped-mass L_2-projection.

Remark 4.24. A symmetric counterpart of the upstream SLIP limiter (4.100) based on the reconstruction of local 1D stencils can be formulated in the same way. The one-dimensional version of formula (4.156) depends on three consecutive slopes and amounts to a double application of the one-sided slope limiter (4.97)

$$\bar{s}_{ij} = minmod\{2(u_{i-1} - u_i), u_i - u_{i+1}, 2(u_{i+1} - u_{i+2})\}.$$

If the sign of $u_i - u_{i+1}$ differs from that of a neighboring slope, then $\bar{s}_{ij} = 0$. Otherwise, the result is at most twice as large in magnitude as $u_{i-1} - u_i$ and $u_{i+1} - u_{i+2}$.

Remark 4.25. If i is an internal node, while j is a node on the boundary, then the positive coefficients a_{ji}^+ and a_{jj}^+ pose no hazard to monotonicity of the discrete problem. Indeed, the corresponding algebraic equation is replaced by the Dirichlet boundary conditions before the linear solver is invoked. Hence, the one-sided slope limiting strategy (4.92) may be employed to constrain the antidiffusive flux f_{ij}^+ into node i.

As the mesh is refined, the difference between the local slopes shrinks and $\bar{s}_{ij}$ approaches $u_i - u_j$. The validity of the discrete maximum principle follows from estimates (4.94)–(4.96) with $d_{ij} = a_{ij}^+$ and limited antidiffusive fluxes satisfying

$$\bar{q}_{ij}(u_k - u_i) = \bar{f}_{ij}^+ = -\bar{f}_{ji}^+ = -\bar{q}_{ji}(u_l - u_j),$$

where $u_k = u_i^{\max}$ or $u_k = u_i^{\min}$ and $u_l = u_j^{\min}$ or $u_l = u_j^{\max}$ depending on the sign of $u_i - u_j$. The edge contributions received by nodes i and j are of LED type since

$$0 \le \bar{q}_{ij} \le q_{ij} = 2\gamma_{ij} a_{ij}^+, \qquad 0 \le \bar{q}_{ji} \le q_{ji} = 2\gamma_{ji} a_{ji}^+. \tag{4.157}$$

Of course, the constrained Galerkin scheme stays conservative since $\bar{f}_{ij}^+ + \bar{f}_{ji}^+ = 0$.

4.5.4 Treatment of Nonlinearities

After slope limiting, the corrected fluxes $\bar{f}_{ij}^+$ are inserted into the residual vector

$$\bar{r} = b - A^- u + \bar{f}^+, \qquad \bar{f}_i^+ = \sum_{j \ne i} \bar{f}_{ij}^+ \tag{4.158}$$

and a defect correction scheme of the form (4.65) is employed to solve the associated nonlinear system. The first guess $u^{(0)}$ can be obtained by solving the linear system $A^- u^L = b$ or $Au^H = b$. The 'low-order' solution $u^{(0)} = u^L$ is guaranteed to be monotone but its accuracy might be very poor due to the lack of consistency. On the other hand, the unconstrained Galerkin solution $u^{(0)} = u^H$ enjoys the 'best approximation property' but may violate the discrete maximum principle. Before proceeding to the next step, it is worthwhile to trim the undershoots and overshoots, if any. This is acceptable since $u^{(0)}$ is just an arbitrary guess which has no influence on the converged solution and, therefore, is not required to be conservative.

The simplest preconditioner for subsequent cycles is, again, the linear operator $\bar{A} = A^-$. It does not need to be reassembled and linear solvers for (4.66) converge rapidly owing to the M-matrix property. However, the convergence of outer iterations tends to be very slow, or even fail, if the anisotropy effects are too strong. Residual smoothing (4.62) makes it possible to achieve monotone convergence but only the final solution is guaranteed to satisfy the discrete maximum principle. The alternative is to trade conservation for positivity and use one of the preconditioners presented in Section 4.2.4. Then intermediate results are devoid of spurious oscillations and converge to the mass-conserving final solution in a monotone fashion.

The relaxation factors for preconditioners (4.73) and (4.74) with $L = -A^-$ depend on the nonnegative coefficients $\bar{q}_{ij}$ and q_{ij} related by (4.157). The former version converges faster but the nonlinear diagonal part of $\bar{A}$ must be updated along with the solution. Anyhow, thousands of defect correction steps may be required to make the Euclidean norm of the residual as small as 10^{-10} on a fine mesh. Then the nonlinearity of the slope-limited Galerkin scheme results in a high overhead cost.

Setting the off-diagonal entries of $\bar{A}$ to zero leads to a fully explicit solution strategy but the number of outer iterations increases further. As in the case of linear systems that result from the discretization of elliptic problems, convergence rates deteriorate as the mesh is refined. Multigrid acceleration of the basic defect correction scheme seems to be a promising way to make computations more efficient.

4.5.5 Numerical Examples

In the below numerical study, we test the ability of the symmetric slope limiter (4.156) to enforce the DMP for (4.146) with an anisotropic diffusion tensor. Also, we present a grid convergence study for test problems with smooth and discontinuous data. The difference between the numerical solution u_h and the exact solution u is measured in the discrete norms (4.109) and (4.110). The defect correction scheme preconditioned by (4.74) is employed as the iterative solver for nonlinear systems.

4.5.5.1 Nonsmooth solutions

To illustrate the benefits of slope limiting, we present two examples in which the existence of steep gradients represents a challenge to the conventional Galerkin discretization. The computational domain $\Omega = (0,1)^2 \backslash [4/9, 5/9]^2$ for the first test problem (TP1) is depicted in Fig. 4.14a. The outer and inner boundary of Ω are denoted by Γ_0 and Γ_1, respectively. Consider the Dirichlet boundary conditions

$$u = -1 \quad \text{on } \Gamma_0, \qquad u = 1 \quad \text{on } \Gamma_1. \tag{4.159}$$

Let the diffusion tensor $\mathscr{D}$ be the symmetric positive definite 2×2 matrix given by

$$\mathscr{D} = \mathscr{R}(-\theta) \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix} \mathscr{R}(\theta), \qquad \mathscr{R}(\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \tag{4.160}$$

where $k_1 = 100$ and $k_2 = 1$ are the diffusion coefficients associated with the axes of the Cartesian coordinate system rotated by the angle $\theta = -\pi/6$. The source/sink term s is taken to be zero. By the continuous maximum principle, the exact solution to the elliptic problem (4.146) is bounded by the Dirichlet boundary data $g = \pm 1$. However, the diffusion tensor (4.160) is highly anisotropic, which may result in a violation of the DMP even if a regular mesh of acute/nonnarrow type is employed.

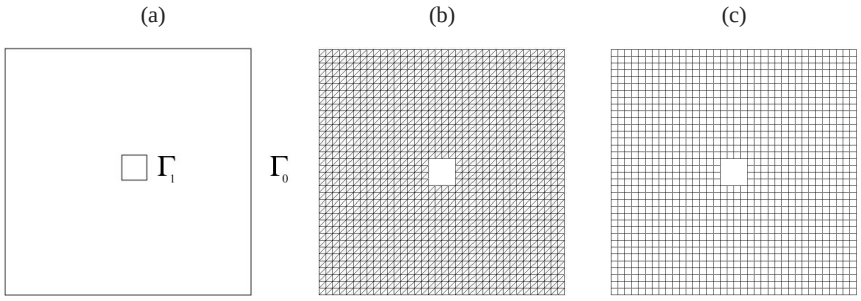


Fig. 4.14 TP1: (a) computational domain Ω, (b) triangular mesh, (c) quadrilateral mesh.

(a) $u_{\min} = -1.055,\ u_{\max} = 1.0$ (b) $u_{\min} = -1.054,\ u_{\max} = 1.0$

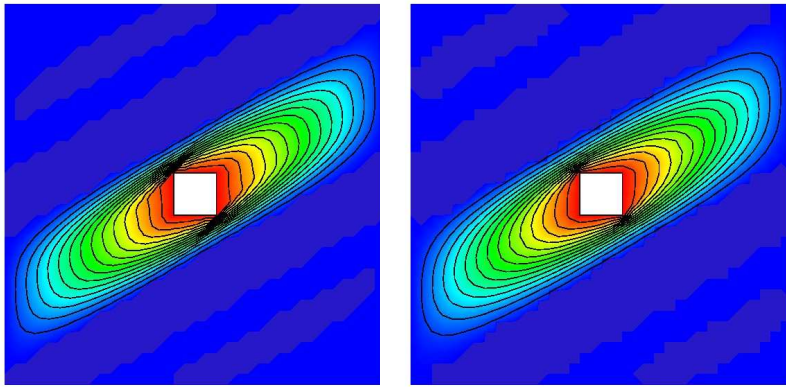


Fig. 4.15 TP1: unconstrained solutions, (a) triangular mesh, (b) quadrilateral mesh.

(a) $u_{\min} = -1.0,\ u_{\max} = 1.0$ (b) $u_{\min} = -1.0,\ u_{\max} = 1.0$

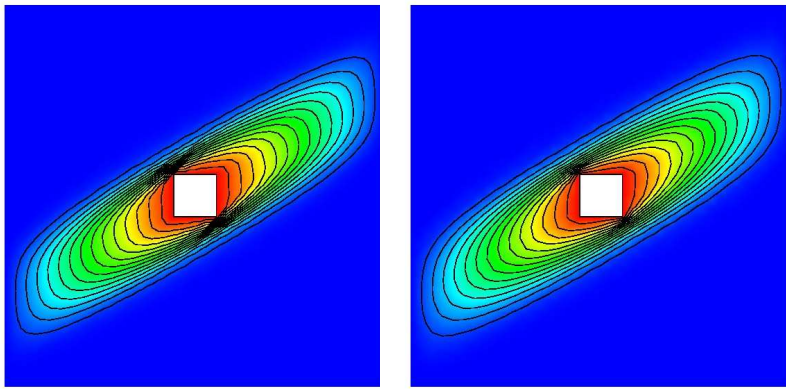


Fig. 4.16 TP1: constrained solutions, (a) triangular mesh, (b) quadrilateral mesh.

The verification of the DMP property is performed for linear and bilinear finite elements on two uniform meshes (see Fig. 4.14b-c). In both cases, the total number of nodes is 1,360. The number of mesh elements equals 2,560 for the triangular mesh and 1,280 for the quadrilateral one. The numerical solutions computed on these meshes by the standard Galerkin method are displayed in Fig. 4.15. Both of them attain correct maximum values but exhibit spurious minima that fall below the theoretical lower bound $u_{\min} = -1$ by about 5%. Although the undershoots are relatively small, they might be totally unacceptable in some situations. For example, if the scalar variable u is responsible for phase transitions, such undershoots can trigger a nonphysical process. Since it is rather difficult to 'repair' a DMP-violating solution [222], it is worthwhile to use a scheme that does not produce undershoots/overshoots in the first place. The constrained Galerkin solutions com-

puted on the same meshes using the slope limiter (4.156) are shown in Fig. 4.16. Both of them satisfy the DMP perfectly, and no other side effects are observed.

The second test problem (TP2) stems from a benchmark suite for anisotropic diffusion problems on general grids ([146], see Test 9: anisotropy and wells). The diffusion tensor is given by (4.160) with $k_1 = 1$, $k_2 = 10^{-3}$, and $\theta = 67.5^\circ$. As before, the source term is zero. The domain $\Omega = (0,1)^2 \backslash (\overline{\Omega}_{4,6} \cup \overline{\Omega}_{8,6})$ has two square holes that correspond to cells $(4,6)$ and $(8,6)$ of a uniform grid with 11×11 cells. The Dirichlet boundary conditions prescribed on $\Gamma_1 = \partial\overline{\Omega}_{4,6}$ and $\Gamma_2 = \partial\overline{\Omega}_{8,6}$ are

$$u = 0 \quad \text{on } \Gamma_1, \qquad u = 1 \quad \text{on } \Gamma_2. \tag{4.161}$$

Homogeneous Neumann boundary conditions are applied at the outer boundary Γ_0 of Ω. For a detailed description of this benchmark problem we refer to [146].

The numerical solutions obtained with 11×11 bilinear finite elements are shown in Fig. 4.17. On such a coarse mesh, the unconstrained Galerkin method produces undershoots and overshoots of about 14%. Other discretization methods compared

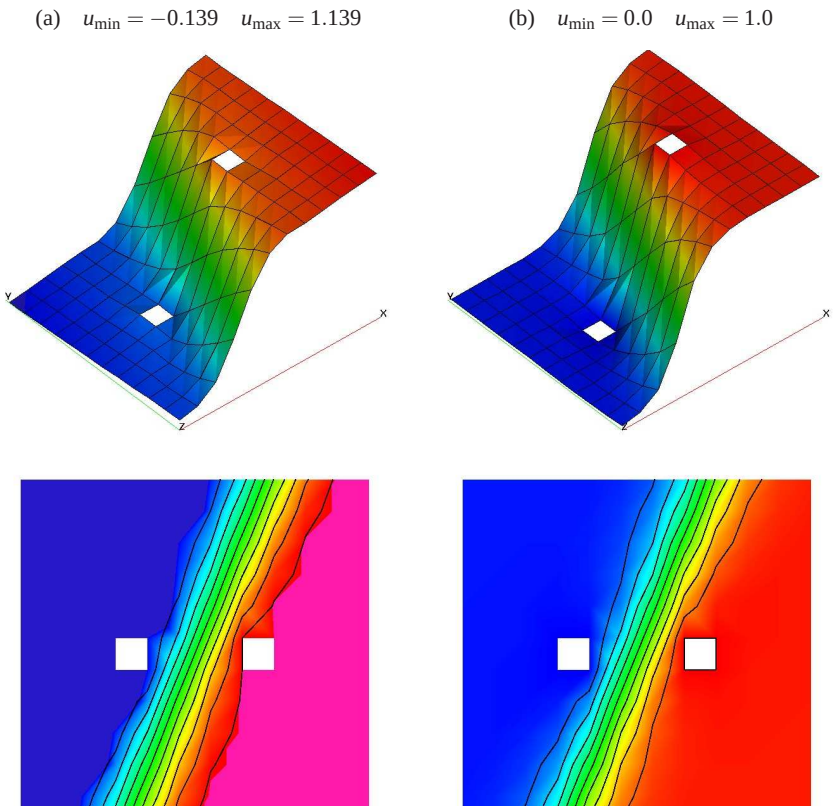


Fig. 4.17 TP2: bilinear elements, (a) unconstrained solution, (b) constrained solution.

in [146] behave in the same way, whereas the slope-limited solution is uniformly bounded by the Dirichlet boundary values, as required by the maximum principle.

4.5.5.2 Smooth solutions

Next, we study the approximation properties of the proposed technique as applied to problems with smooth solutions. Usually, even the conventional Galerkin scheme does not violate the discrete maximum principle for this type of problems. Thus, no slope limiting is actually required for smooth data. The goal of the numerical experiments to be performed is to compare the accuracy and convergence behavior of the constrained nonlinear scheme to those of the underlying Galerkin discretization.

The diffusion tensor and source for the third test problem (TP3) are given by

$$\mathscr{D} = \begin{pmatrix} 100 & 0 \\ 0 & 1 \end{pmatrix}, \qquad s(x,y) = 50.5\sin(\pi x)\sin(\pi y). \tag{4.162}$$

With these parameter settings, the exact solution to the Dirichlet problem (3.21) is

$$u(x,y) = \frac{1}{2\pi^2}\sin(\pi x)\sin(\pi y). \tag{4.163}$$

In accordance with this formula, homogeneous Dirichlet boundary conditions are imposed. The problem is solved on a sequence of distorted triangular and quadrilateral meshes. Given a uniform grid with spacing h, its distorted counterpart is generated by applying random perturbations to the coordinates of internal nodes

$$x := x + \alpha\xi_x h \qquad y := y + \alpha\xi_y h,$$

where ξ_x and ξ_y are random numbers with values in the range from -0.5 to 0.5. The parameter $\alpha \in [0,1]$ quantifies the degree of distortion. The default value $\alpha = 0.4$ was adopted to introduce sufficiently strong grid deformations without tangling.

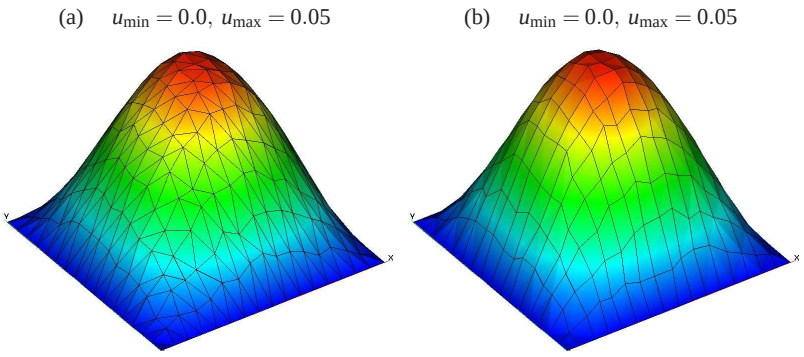


Fig. 4.18 TP3: numerical solutions, (a) triangular mesh, (b) quadrilateral mesh.

Table 4.10 TP3: grid convergence study for the unconstrained Galerkin scheme.

	triangular meshes		quadrilateral meshes	
h	E_2	$E_{\max}$	E_2	$E_{\max}$
1/16	0.158e-3	0.576e-3	0.113e-3	0.396e-3
1/32	0.445e-4	0.154e-3	0.270e-4	0.113e-3
1/64	0.112e-4	0.473e-4	0.693e-5	0.351e-4
1/128	0.320e-5	0.140e-4	0.176e-5	0.789e-5
1/256	0.820e-6	0.467e-5	0.441e-6	0.231e-5

Table 4.11 TP3: grid convergence study for the constrained Galerkin scheme.

	triangular meshes		quadrilateral meshes	
h	E_2	$E_{\max}$	E_2	$E_{\max}$
1/16	0.293e-3	0.136e-2	0.265e-3	0.103e-2
1/32	0.656e-4	0.407e-3	0.616e-4	0.337e-3
1/64	0.146e-4	0.121e-3	0.104e-4	0.847e-4
1/128	0.321e-5	0.140e-4	0.204e-5	0.211e-4
1/256	0.826e-6	0.467e-5	0.468e-6	0.642e-5

In this test, the results produced by the Galerkin scheme and by its slope-limited counterpart are optically indistinguishable. The diagrams in Fig. 4.18 show the solutions computed using linear and bilinear finite elements with $h = 1/16$. The corresponding grid convergence study is presented in Tables 4.10–4.11. On coarse meshes, the slope limiter tends to 'clip' smooth peaks, as in the case of FEM-FCT methods. To alleviate the undesirable decay of admissible local extrema, the sufficient conditions of DMP should be replaced by a weaker monotonicity constraint.

As the mesh is refined and resolution improves, the slope limiter is gradually deactivated, and the error norms approach those for the Galerkin method. The results presented in Tables 4.10–4.11 indicate that slope limiting does not degrade the order of convergence, and peak clipping becomes less pronounced on finer meshes.

4.5.5.3 Heterogeneous diffusion

The last example (TP4) is designed to test the ability of a discretization technique to handle problems with discontinuous coefficients. Let the diffusion tensor $\mathscr{D}$ be a piecewise-constant function defined in the unit square $\Omega = (0,1)^2$ as follows

$$\mathscr{D}(x,y) = \begin{cases} \mathscr{D}_1, & \text{if } x < 0.5, \\ \mathscr{D}_2, & \text{if } x > 0.5, \end{cases} \tag{4.164}$$

where

$$\mathscr{D}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \mathscr{D}_2 = \begin{pmatrix} 10 & 3 \\ 3 & 1 \end{pmatrix}.$$

This heterogeneous diffusion tensor has a jump in value and direction of anisotropy across the line $x = 0.5$. The source term s is also discontinuous along this line

$$s(x,y) = \begin{cases} 4.0, & \text{if } x < 0.5, \\ -5.6, & \text{if } x > 0.5. \end{cases} \tag{4.165}$$

For $\mathscr{D}$ and s defined as above, an analytical solution to problem (4.146) is given by

$$u(x,y) = \begin{cases} 1 - 2y^2 + 4xy + 2y + 6x, & \text{if } x \leq 0.5, \\ b_2 y^2 + c_2 xy + d_2 x + e_2 y + f_2, & \text{if } x > 0.5. \end{cases} \tag{4.166}$$

Substitution into (4.146) yields the following values of the involved coefficients

$$b_2 = -2, \quad c_2 = \frac{4(\mathscr{D}_2(1,2)+1)}{\mathscr{D}_2(1,1)}, \quad d_2 = \frac{6 - 4\mathscr{D}_2(1,2)}{\mathscr{D}_2(1,1)},$$
$$e_2 = \frac{4\mathscr{D}_2(1,1) - 2\mathscr{D}_2(1,2) - 2}{\mathscr{D}_2(1,1)}, \quad f_2 = \frac{4\mathscr{D}_2(1,1) + 2\mathscr{D}_2(1,2) - 3}{\mathscr{D}_2(1,1)}.$$

Again, the discretization is performed using linear and bilinear finite elements on distorted meshes. These meshes are constructed as explained in the previous subsection but nodes that lie on the line $x = 0.5$ are shifted in the $y-$direction only.

The unconstrained Galerkin solutions for $h = 1/16$ are presented in Fig. 4.19. Their limited counterparts look the same but a comparison of the error norms presented in Tables 4.12–4.13 reveals significant differences between the convergence histories of the slope-limited version on triangular and quadrilateral meshes. Although the solution consists of two smooth patches, its gradient is discontinuous across the internal interface $x = 0.5$. Moreover, the kink in the solution profile makes

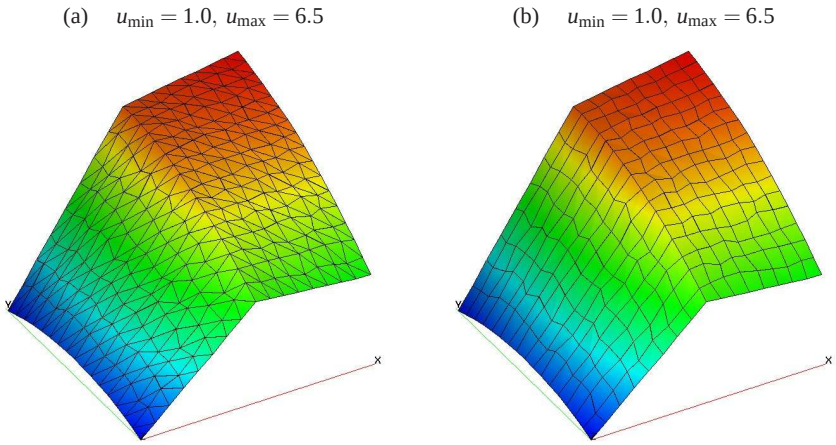


Fig. 4.19 TP4: numerical solutions, (a) triangular mesh, (b) quadrilateral mesh.

Table 4.12 TP4: grid convergence study for the unconstrained Galerkin scheme.

	triangular meshes		quadrilateral meshes	
h	E_2	$E_{\max}$	E_2	$E_{\max}$
1/16	0.101e-2	0.337e-2	0.473e-3	0.167e-2
1/32	0.328e-3	0.133e-2	0.154e-3	0.636e-3
1/64	0.841e-4	0.374e-3	0.426e-4	0.242e-3
1/128	0.211e-4	0.107e-3	0.109e-4	0.514e-4
1/256	0.551e-5	0.351e-4	0.286e-5	0.166e-4

Table 4.13 TP4: grid convergence study for the constrained Galerkin scheme.

	triangular meshes		quadrilateral meshes	
h	E_2	$E_{\max}$	E_2	$E_{\max}$
1/16	0.244e-2	0.155e-1	0.473e-3	0.167e-2
1/32	0.161e-2	0.187e-1	0.154e-3	0.636e-3
1/64	0.101e-2	0.109e-1	0.426e-4	0.242e-3
1/128	0.281e-3	0.438e-2	0.109e-4	0.514e-4
1/256	0.140e-3	0.214e-2	0.286e-5	0.166e-4

the outcome of the slope limiting procedure highly mesh-dependent. Note that the solution is smooth along the y-axis and piecewise-smooth along the $x-$axis. This is why the constrained and unconstrained solutions coincide on quadrilateral meshes.

On the other hand, some edges of the triangular mesh are directed skew to the kink so that the corresponding solution differences are large, whereas the distance to the nearest local maximum or minimum, as defined in (4.87), is small. This places a heavy burden on the slope limiter which is forced to reject a large percentage of the antidiffusive flux in accordance with (4.156). The approximation of discontinuous gradients by means of the standard L_2-projection (4.88) can also be responsible for the relatively slow convergence on distorted triangular meshes. In summary, this test problem turns out to be very easy or rather difficult, depending on the orientation of mesh edges. It was included to identify the limitations of the proposed limiting strategy, discuss their ramifications, and illustrate the need for further research.

4.6 Summary

The algebraic flux correction paradigm considered in this chapter provides a set of general rules, concepts, and tools that make it possible to enforce positivity constraints and the discrete maximum principle in an adaptive way. The presented algorithms are based on a generalization of classical high-resolution schemes. All of them exhibit a common structure but the actual flux/slope limiting procedure can be tailored to the specific properties of the transport problem under investigation.

In the context of finite element methods, an upwind-biased limiting strategy of TVD type is appropriate if the Peclet number is large and the mass lumping error is negligible/acceptable. The inclusion of the consistent mass matrix and the use of symmetric FCT limiters are to be recommended for transient computations with small time steps. Similarly, the best approach to the solution of linear and nonlinear systems depends on the properties of the continuous and discrete problems.

The presentation of algebraic flux correction schemes in this chapter was aimed primarily at finite element practitioners who share our viewpoint (or can be convinced) that positivity preservation is more important than the Galerkin orthogonality when it comes to numerical solution of problems with steep fronts. Instead of manipulating the variational formulation and fitting a free parameter, we have shown that artificial diffusion operators can be constructed at the discrete level so as to control the contribution of matrix entries associated with antidiffusive fluxes.

Remarkably, the same limiter routine can be employed to enforce the positivity constraint for linear and multilinear elements in 2D and 3D, on structured and unstructured meshes. The origin of discrete operators makes no difference as far as the M-matrix property is concerned. However, the flux limiter must be designed to keep the perturbation of the algebraic system as small as possible. The demand for high resolution is particularly difficult to meet in the case of higher-order finite elements because the fluxes may depend on solution values at more than two nodes, and even the construction of an optimal low-order scheme becomes a nontrivial task.

In essence, flux correction is intended to increase the local order of accuracy in smooth regions and decrease it when a front is detected. Hence, a further extension can be envisaged within the framework of a variable-order (*p-adaptive*) finite element scheme with hierarchical basis functions. The basic idea is as follows:

1. Compute a first approximation $\bar{u}$ using (multi-)linear elements and flux limiting.
2. Increase the polynomial order in smooth regions where no limiting is performed.
3. Recalculate the solution using the nonoscillatory predictor $\bar{u}$ as the initial guess.

On the one hand, it is difficult to prove that the discrete maximum principle will continue to hold in regions where the lowest-order Galerkin approximation produces monotone results. On the other hand, it is intuitively clear that the risk of a DMP violation is minimal and worth taking, given the lack of cost-effective alternatives. Also, it is always possible to decrease the order of basis functions if the solution develops local maxima or minima that are not present in $\bar{u}$ computed at stage 1.

The unavoidable errors due to low-order artificial diffusion that cannot be removed in the vicinity of internal or boundary layers can be compensated by means of local mesh refinement. The use of mesh adaptation for improving the accuracy of numerical approximations to transport equations is addressed in Chapter 5.

Chapter 5
Error Estimates and Adaptivity

In this chapter, we discuss some aspects of mesh adaptation for steady transport equations. The goal-oriented error estimator developed in [197, 199] is used as a refinement criterion. The error in the value of a linear target functional is measured in terms of weighted residuals that depend on the solutions to the primal and dual problems. The Galerkin orthogonality error is taken into account and turns out to be dominant whenever flux or slope limiters are activated to enforce monotonicity constraints. The localization of global errors is performed using a natural decomposition of the involved weights into nodal contributions. The developed simulation tools are applied to a linear convection problem in two space dimensions.

5.1 Introduction

The goal-oriented approach to error estimation [14, 27, 185, 295, 309] is applicable not only to elliptic PDEs but also to hyperbolic conservation laws [141, 142, 310]. In most cases, the error in the quantity of interest is estimated using the duality argument, Galerkin orthogonality, and a direct decomposition of the weighted residual into element contributions. The most prominent representative of such error estimators is the Dual Weighted Residual (DWR) method of Becker and Rannacher [27, 28]. The recent paper by Meidner *et al.* [248] is a rare example of a DWR estimate that does not require Galerkin orthogonality or information about the cause of its possible violation.

Kuzmin and Korotov [197] applied the DWR method to steady convection-diffusion equations and obtained a simple estimate of local Galerkin orthogonality errors due to flux limiting or other 'variational crimes.' In contrast to the usual approach, the weighted residuals are decomposed into nodal (rather than element) contributions. In regions of insufficient mesh resolution, the computable Galerkin orthogonality error comes into prominence. The mesh adaptation strategy to be presented below takes advantage of this fact.

5.2 Galerkin Weak Form

Steady convective transport of a conserved scalar quantity u in a domain Ω with boundary Γ can be described by the linear hyperbolic equation

$$\nabla \cdot (\mathbf{v}u) = s \qquad \text{in } \Omega. \tag{5.1}$$

Here $\mathbf{v}$ is a stationary velocity field and s is a volumetric source/sink. Due to hyperbolicity, a Dirichlet boundary condition is imposed at the inlet

$$u = u_D \qquad \text{on} \quad \Gamma_{\text{in}} = \{\mathbf{x} \in \Gamma \,|\, \mathbf{v} \cdot \mathbf{n} < 0\}, \tag{5.2}$$

where $\mathbf{n}$ is the unit outward normal and u_D is the prescribed boundary data.

The weak form of the above boundary value problem can be written as

$$a(w,u) = b(w), \qquad \forall w. \tag{5.3}$$

For brevity, we refrain from an explicit definition of functional spaces. The bilinear form $a(\cdot,\cdot)$ and the linear functional $b(\cdot)$ are defined by

$$a(w,u) = \int_\Omega w \nabla \cdot (\mathbf{v}u) \Delta x - \int_{\Gamma_{\text{in}}} w u \mathbf{v} \cdot \mathbf{n} \, \mathrm{d}s, \tag{5.4}$$

$$b(w) = \int_\Omega w s \Delta x - \int_{\Gamma_{\text{in}}} w u_D \mathbf{v} \cdot \mathbf{n} \, \mathrm{d}s. \tag{5.5}$$

The inflow boundary conditions are imposed weakly via the surface integrals.

The differentiation of $\mathbf{v}u$ in (5.4) can be avoided using integration by parts

$$a(w,u) = \int_\Gamma w u \mathbf{v} \cdot \mathbf{n} \, \mathrm{d}s - \int_\Omega \nabla w \cdot (\mathbf{v}u) \Delta x. \tag{5.6}$$

This representation implies that a discontinuous weak solution u is admissible. In linear hyperbolic problems of the form (5.1), singularities travel along the streamlines of $\mathbf{v}$. They may be caused by a jump in the value of s or u_D.

5.3 Global Error Estimates

Let u_h be a continuous function that may represent an approximate solution to (5.1)–(5.2) or a finite element interpolant of discrete nodal values. The numerical error $e = u - u_h$ can be measured using the residual of (5.3)

$$\rho(w,u_h) = b(w) - a(w,u_h). \tag{5.7}$$

Obviously, the value of $\rho(w,u_h)$ depends not only on the quality of u_h but also on the choice of w. In goal-oriented estimates, this weight carries information about the

quantities of interest. The objectives of a numerical study are commonly defined in terms of a linear output functional, such as [310]

$$j(u) = \int_{\Omega} gu\Delta x + \int_{\Gamma_{\text{out}}} hu\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}s, \qquad g,h \in \{0,1\}. \tag{5.8}$$

The piecewise-constant function g picks out a subdomain, for example, an interior or boundary layer, where a particularly accurate approximation to u is desired. The selector h picks out a portion of the outflow boundary $\Gamma_{\text{out}} = \{\mathbf{x} \in \Gamma \,|\, \mathbf{v}\cdot\mathbf{n} > 0\}$, where the convective flux is to be controlled.

In order to estimate the error $j(e)$ in the numerical value of the output functional, consider the *dual* or *adjoint* problem [27, 28] associated with (5.3)

$$a(z,e) = j(e), \qquad \forall e. \tag{5.9}$$

The surface integral in (5.8) implies the weakly imposed Dirichlet boundary condition $z = h$ on Γ_{out} [310]. The error $j(e)$ and residual (5.7) are related by

$$j(u-u_h) = a(z,u-u_h) = \rho(z,u_h). \tag{5.10}$$

An arbitrary numerical approximation z_h to the exact solution z of the dual problem (5.9) can be used to decompose the so-defined error as follows

$$j(u-u_h) = \rho(z-z_h,u_h) + \rho(z_h,u_h). \tag{5.11}$$

If Galerkin orthogonality holds for the numerical approximation u_h, then $\rho(z_h,u_h) = 0$. Thus, the computable term $\rho(z_h,u_h)$ is omitted in most goal-oriented error estimates for finite element discretizations. However, the orthogonality condition is frequently violated due to numerical integration, round-off errors, slack tolerances for iterative solvers, and flux limiting.

Since the exact dual solution z is usually unknown, the derivation of a computable error estimate involves another approximation $\hat{z} \approx z$ such that

$$j(u-u_h) \approx \rho(\hat{z}-z_h,u_h) + \rho(z_h,u_h). \tag{5.12}$$

The magnitudes of the two residuals can be estimated separately as follows:

$$|\rho(\hat{z}-z_h,u_h)| \le \Phi, \qquad |\rho(z_h,u_h)| \le \Psi, \tag{5.13}$$

where the globally defined bounds Φ and Ψ are assembled from contributions of individual nodes or elements, as explained in the next section.

The reference solution $\hat{z}$ is commonly obtained from z_h using some sort of post-processing. If $\rho(z_h,u_h) = 0$, then the estimate $j(u-u_h) \approx 0$ that follows from (5.12) with $\hat{z} = z_h$ is worthless, hence the need to compute $\hat{z}$ on another mesh or interpolate it using higher-order polynomials [197, 295]. On the other hand, the setting $\hat{z} = z_h$ is not only acceptable but also optimal for nonlinear flux-limited discretizations such that $j(u-u_h) \approx \rho(z_h,u_h) \neq 0$. In situations when the term $\rho(z-z_h,u_h)$ is non-

negligible, extra work needs to be invested into the recovery of a superconvergent approximation $\hat{z} \neq z_h$.

5.4 Local Error Estimates

The global upper bounds Φ and Ψ make it possible to verify the accuracy of the approximate solution u_h but the estimated errors in the quantity of interest must be localized to find the regions where a given mesh is too coarse or too fine. A straightforward decomposition of weighted residuals into element contributions results in an oscillatory distribution and a strong overestimation of local errors. In particular, the restriction of the term $\rho(z_h, u_h)$ to a single element Ω_k can be large in magnitude even if Galerkin orthogonality is satisfied globally (positive and negative contributions cancel out).

Following Schmich and Vexler [295], we decompose Φ and Ψ into local bounds associated with the nodes of the mesh on which z_h is defined. Let

$$z_h = \sum_i z_i \varphi_i, \tag{5.14}$$

where $\{\varphi_i\}$ is a set of Lagrange basis functions such that $\sum_i \varphi_i \equiv 1$ and

$$\hat{z} - z_h = \sum_i w_i, \qquad w_i = \varphi_i(\hat{z} - z_h). \tag{5.15}$$

The contribution of node i to the bounds Φ and Ψ is defined as in [197]

$$\Phi_i = |\rho(w_i, u_h)|, \qquad \Psi_i = |\rho(z_i \varphi_i, u_h)|. \tag{5.16}$$

If the residual is orthogonal to the test function φ_i, then $\Psi_i = 0$. A nonvanishing value of Ψ_i implies a local violation of Galerkin orthogonality.

The magnitude of $j(u - u_h)$ is estimated by the sum of local errors, i.e.,

$$\Phi = \sum_i \Phi_i, \qquad \Psi = \sum_i \Psi_i. \tag{5.17}$$

Finally, an optional conversion into element contributions is performed for mesh adaptation purposes. Introducing the continuous error function

$$\xi = \sum_i \xi_i \varphi_i, \qquad \xi_i = \frac{\Phi_i + \Psi_i}{\int_\Omega \varphi_i \Delta x}, \tag{5.18}$$

the following representation of the total error $\eta = \Phi + \Psi$ is obtained [197]

$$\eta = \sum_k \eta_k, \qquad \eta_k = \int_{\Omega_k} \xi \Delta x. \tag{5.19}$$

In a practical implementation, the midpoint rule is employed to calculate η_k.

5.5 Numerical Experiments

In this section, the presented high-resolution finite element scheme, goal-oriented error estimator, and hierarchical mesh adaptation algorithm are applied to a test problem from [156]. Consider equation (5.1) with $s \equiv 0$ and

$$\mathbf{v}(x,y) = (y,-x) \qquad \text{in} \quad \Omega = (-1,1)\times(0,1).$$

This incompressible velocity field corresponds to steady rotation about $(0,0)$.

The exact solution and inflow boundary conditions are given by [156]

$$u(x,y) = \begin{cases} 1, & \text{if } 0.35 \le \sqrt{x^2+y^2} \le 0.65, \\ 0, & \text{otherwise.} \end{cases}$$

The so-defined discontinuous inflow profile ($-1 \le x < 0,\ y = 0$) undergoes circular convection and propagates along the streamlines of $\mathbf{v}(x,y)$ all the way to the outlet ($0 < x \le 1,\ y = 0$), while its shape remains the same.

Let $j(u)$ be defined by (5.8) with $g = 1$ in $\omega = (-0.1,0.1)\times(0,1)$ and $g = 0$ elsewhere. The function h is defined as the trace of g on Γ_{out}. The exact value of $j(u)$ is $6.04497\text{e}-02$. The solution shown in Fig. 5.1 (a) was computed by the FEM-LED scheme described in Chapter 4 on a uniform mesh of bilinear elements with spacing $h = 1/80$. Owing to algebraic flux correction, the resolution of the discontinuous front is remarkably sharp, and no undershoots or overshoots are observed. However, it is obvious that there is actually no need for such a high resolution beyond $x > 0.1$ if it is enough to have an accurate approximation in the small subdomain ω. Indeed, whatever is happening downstream of ω has no influence on the solution in this subdomain. This is illustrated by Fig. 5.1 (b) which shows the solution to the dual problem computed by the FEM-LED scheme on the same mesh.

Goal-oriented error analysis is performed using estimate (5.12) with $\hat{z} = z_h$. This setting implies that $\Phi = 0$ and $\eta = \Psi$ is the Galerkin orthogonality error caused by flux limiting. Remarkably, the resulting global estimates are in a good agreement with the exact error which is illustrated in Table 5.1 for different grid spacings. The sharpness of the obtained error estimates is measured using the absolute and relative effectivity indices [197]

$$I_{\text{eff}} = \frac{\eta}{|(j(u-u_h)|}, \qquad I_{\text{rel}} = \left| \frac{|(j(u-u_h)| - \eta}{|(j(u)|} \right|.$$

We remark that the value of I_{eff} is unstable and misleading when the denominator is very small or zero, and the evaluation of integrals is subject to rounding errors. The relative effectivity index I_{eff} is free of this drawback and exhibits monotone convergence as the mesh is refined (see Table 5.1).

h	$\lvert j(u-u_h)\rvert$	$\eta(z_h,u_h)$	I_{eff}	I_{rel}
1/10	2.009555e-03	2.115012e-03	1.05	1.744541e-03
1/20	4.401534e-04	3.640322e-04	0.82	1.259248e-03
1/40	1.312391e-04	1.025215e-04	0.78	4.750662e-04
1/80	4.283158e-05	3.535738e-05	0.82	1.236433e-04
1/160	1.254089e-05	1.072697e-05	0.85	3.000709e-05

Table 5.1 Circular convection: exact vs. estimated global error.

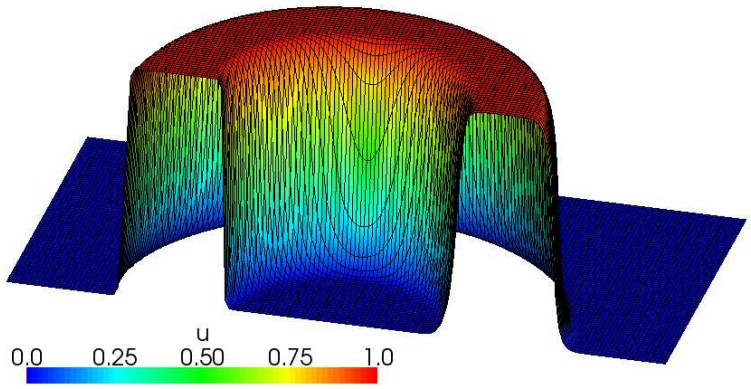
The adaptive hybrid mesh presented in Fig. 5.2 is refined along the discontinuity lines of u but only until they cross the outflow boundary of ω. Using a finer mesh beyond the line $x = 0.1$ would not improve the accuracy of the solution u_h inside ω. The smallest mesh width is $h = 1/320$, which corresponds to more than 200,000 cells in the case of global mesh refinement.

Since the dual weight z_h contains built-in information regarding the transport of errors and goals of simulation, such error estimators furnish a better refinement criterion than, for example, error indicators based on gradient recovery [362]. In the latter case, unnecessary mesh refinement would take place along the discontinuities located downstream of the subdomain ω.

5.6 Summary

A goal-oriented error estimate was derived for LED discretizations of a steady transport equation. The loss of Galerkin orthogonality in the process of flux limiting was shown to provide valuable feedback for mesh adaptation. The local orthogonality error was employed to generate an adaptive mesh for circular convection in a 2D domain. Diffusive terms can be included using gradient recovery to stabilize the residuals and infer a proper distribution of local errors [197]. Further work will concentrate on goal-oriented error estimation for unsteady flow problems.

(a) primal solution



(b) dual solution

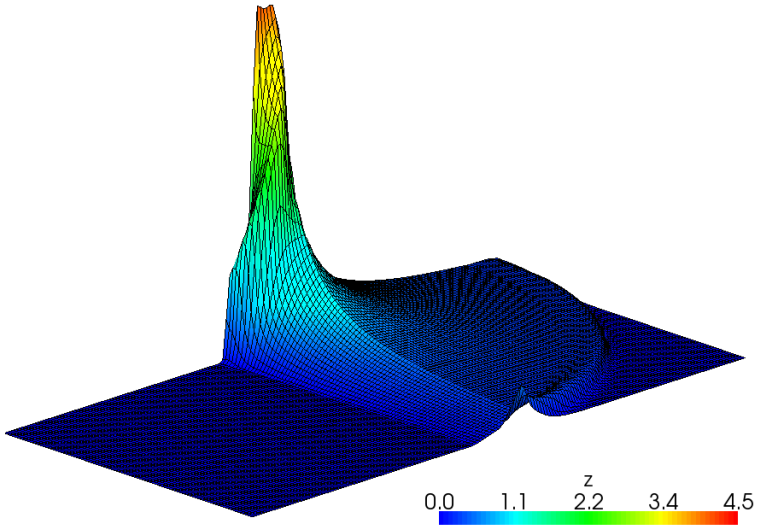
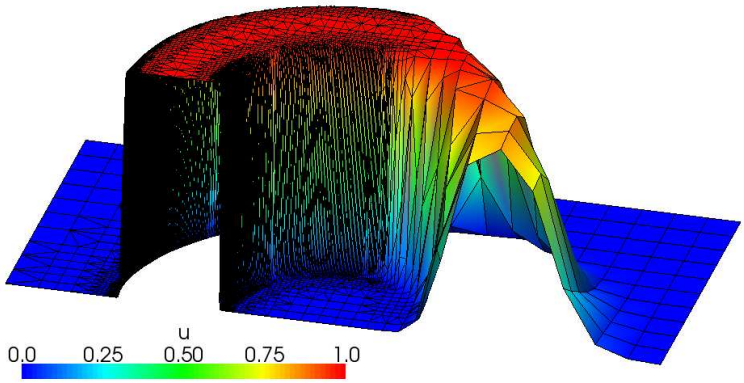


Fig. 5.1 Circular convection: FEM-LED discretization, $h = 1/80$.

(a) primal solution



(b) computational mesh

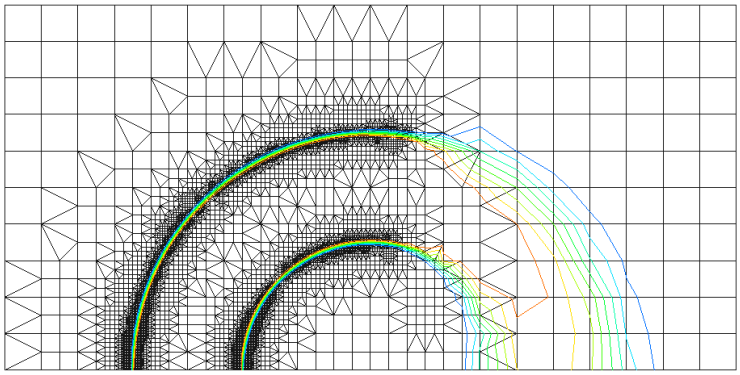


Fig. 5.2 Circular convection: FEM-LED discretization, 5,980 cells.

References

1. M. Aftosmis and N. Kroll, A quadrilateral based second-order TVD method for unstructured adaptive methods. *AIAA Paper*, 91-0124, 1991.
2. M. Ainsworth and J.T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*. John Wiley & Sons, New York, 2000.
3. M. Ainsworth, J.Z. Zhu, A.W. Craig, O.C. Zienkiewicz, Analysis of the Zienkiewicz-Zhu a-posteriori error estimator in the finite element method. *Int. J. Numer. Methods Engrg.* **28**:9 (1989) 2161–2174.
4. J.D. Anderson, Jr., *Computational Fluid Dynamics. The Basics with Applications*. McGraw-Hill, 1995.
5. J.D. Anderson, Jr., *Modern Compressible Flow: With Historical Perspective*, McGraw-Hill, 1990.
6. F. Angrand, A. Derieux, L. Loth, G. Vijayasundaram, Simulation of Euler transonic flows by means of explicit finite element type schemes. *INRIA Research Report* **250**, 1983.
7. F. Angrand and A. Derieux, Some explicit triangular finite element schemes for the Euler equations. *Int. J. Numer. Methods Fluids* **4** (1984) 749–764.
8. P. Arminjon and A. Dervieux, Construction of TVD-like artificial viscosities on 2-dimensional arbitrary FEM grids. *INRIA Research Report* **1111**, 1989.
9. M. Arora and P.L. Roe, A well-behaved TVD limiter for high-resolution calculations of unsteady flow. *J. Comput. Phys.* **132** (1997) 3–11.
10. Athena test suite, `http://www.astro.virginia.edu/VITA/ATHENA/dmr.html`.
11. K. Baba and M. Tabata, On a conservative upwind finite element scheme for convective diffusion equations. *RAIRO Numerical Analysis* **15** (1981) 3–25.
12. I. Babuška and W.C. Rheinboldt, Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.* **15**:4 (1978) 736–354.
13. I. Babuška and W.C. Rheinboldt, A posteriori error estimates for the finite element method. *Int. J. Numer. Methods Engrg.* **12**:10 (1978) 1597–1615.
14. W. Bangerth and R. Rannacher, Adaptive finite element methods for differential equations. Lectures in Mathematics, ETH Zürich, Birkhäuser, 2003.
15. R.E. Bank, A.H. Sherman, A. Weiser, Some refinement algorithms and data structures for regular local mesh refinement. In: R. Stepleman (ed.), *Scientific Computing, Applications of Mathematics and Computing to the Physical Sciences*, IMACS Transactions on Scientific Computation, Vol. I. North-Holland, Amsterdam, 1983, 3–17.
16. R.E. Bank and R.K. Smith, Mesh smoothing using a posteriori error estimates. *SIAM J. Numer. Anal.* **34** (1997) 979–997.
17. T.J. Barth, Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes. *AIAA Paper*, 91-0721, 1991.
18. T.J. Barth, Aspects of unstructured grids and finite volume solvers for the Euler and Navier-Stokes equations. In: Lecture Series 1994-05, von Karman Institute for Fluid Dynamics, Brussels, 1994.

19. T. Barth and D.C. Jespersen, The design and application of upwind schemes on unstructured meshes. *AIAA Paper*, 89-0366, 1989.
20. T. Barth and M. Ohlberger, Finite volume methods: foundation and analysis. In: E. Stein, R. de Borst, T.J.R. Hughes (eds), *Encyclopedia of Computational Mechanics, Volume 1: Fundamentals*. John Wiley & Sons, 2004, 439–474.
21. G.K. Batchelor, *An Introduction to Fluid Dynamics*. Cambridge University Press, 2000.
22. J.D. Baum and R. Löhner, Numerical simulation of pilot/seat ejection from an F-16. *AIAA Paper*, 93-0783, 1993.
23. J.D. Baum, H. Luo, R. Löhner, Validation of a new ALE adaptive unstructured moving body methodology for multi-store ejection simulations. *AIAA Paper*, 95-1792, 1995.
24. C.E. Baumann and T.J. Oden, A discontinuous *hp*-finite element method for the solution of the Euler and Navier-Stokes equations. *Int. J. Numer. Methods Fluids* **31** (1999) 79–95.
25. Y. Bazilevs and T.J.R. Hughes, Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Comput. Fluids* **36**:1 (2007) 12–26.
26. R. Becker and P. Hansbo, A simple pressure stabilization method for the Stokes equation. *Commun. Numer. Methods Engrg.*, in press.
27. R. Becker and R. Rannacher, A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West J. Numer. Math.* **4** (1996) 237–264.
28. R. Becker and R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica* **10** (2001) 1–102.
29. J.-M. Beckers, H. Burchard, E. Deleersnijder, P.P. Mathieu, Numerical discretization of rotated diffusion operators in ocean models. *Monthly Weather Review* **28**:8 (2000) 2711–2733.
30. M.J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.* **82**:1 (1989) 64–84.
31. M.J. Berger and R.J. LeVeque, Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Numer. Anal.* **35**:6 (1998) 1439–1461.
32. M.J. Berger and J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.* **53**:3 (1984) 484–512.
33. M. Berzins, Modified mass matrices and positivity preservation for hyperbolic and parabolic PDEs. *Commun. Numer. Methods Engrg.* **17**:9 (2001) 659–666.
34. M. Berzins, Variable-order finite elements and positivity preservation for hyperbolic PDEs. *Appl. Numer. Math.* **48**:3-4 (2004) 271–292.
35. M. Berzins, Preserving positivity for hyperbolic PDEs using variable-order finite elements with bounded polynomials. *Appl. Numer. Math.* **52** (2005) 197–217.
36. K.S. Bey and J.T. Oden, hp-version discontinuous Galerkin methods for hyperbolic conservation laws. *Comput. Methods Appl. Mech. Engrg.* **133** (1996) 259–286.
37. R. Biswas, K. Devine, and J. E. Flaherty, Parallel adaptive finite element methods for conservation laws. *Appl. Numer. Math.* **14** (1994) 255–284.
38. H. Blank, M. Rudgyard, A. Wathen, Stabilized finite element methods for steady incompressible flow. *Comput. Methods Appl. Mech. Engrg.* **174** (1999) 91–105.
39. J. Blasco, R. Codina, A. Huerta, *Analysis of Fractional Step Finite Element Methods for the Incompressible Navier-Stokes Equations*. Monograph CIMNE **38**, 1997.
40. H. Blum, J. Harig, S. Müller, P. Schreiber, S. Turek, FEAT2D/3D: *Finite Element Analysis Tools*, User Manual, Release 1.3, University of Heidelberg, 1995.
41. D.L. Book, The conception, gestation, birth, and infancy of FCT. In: D. Kuzmin, R. Löhner, S. Turek (eds), *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, Berlin, 2005, 5–28.
42. J.P. Boris and D.L. Book, Flux-Corrected Transport: I. SHASTA, a fluid transport algorithm that works. *J. Comput. Phys.* **11** (1973) 38–69.
43. D.L. Book, J.P. Boris, K. Hain, Flux-Corrected Transport: II. Generalizations of the Method. *J. Comput. Phys.* **18** (1975) 248–283.
44. J.P. Boris and D.L. Book, Flux-Corrected Transport: III. Minimal-error FCT algorithms. *J. Comput. Phys.* **20** (1976) 397–431.
45. J.P. Boris, F.F. Grinstein, E.S. Oran, R.J. Kolbe, New insights into Large Eddy Simulation. *Fluid Dynamics Research* **10**:4–6 (1992) 199–227.

46. J.H. Bramble and B.E. Hubbard, New monotone type approximations for elliptic problems. *Math. Comp* **18** (1964) 349–367.
47. A.N. Brooks and T.J.R. Hughes, Streamline upwind Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.* **32** (1982) 199–259.
48. A. Burbeau, P. Sagaut, and C.-H. Bruneau, A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods. *J. Comput. Phys.* **169** (2001) 111–150.
49. E. Burman, A unified analysis for conforming and nonconforming stabilized finite element methods using interior penalty. *SIAM J. Numer. Anal.* **43** (2005) 2012–2033.
50. E. Burman and P. Hansbo, Edge stabilization for Galerkin approximations of convection-diffusion-reaction problems. *Comput. Methods Appl. Mech. Engrg.* **193** (2004) 1437–1453.
51. E. Burman and A. Ern, Nonlinear diffusion and discrete maximum principle for stabilized Galerkin approximations of the convection-diffusion-reaction equation. *Comput. Methods Appl. Mech. Engrg.* **191** (2002) 3833–3855.
52. E. Burman and A. Ern, Stabilized Galerkin approximation of convection-diffusion-reaction problems: discrete maximum principle and convergence. *Math. Comput.* **74** (2005) 1637–1652.
53. G. Cantin, C. Loubignac, C. Touzot, An iterative scheme to build continuous stress and displacement solutions. *Int. J. Numer. Methods Engrg. 12* (1978) 1493–1506.
54. J.-C. Carette, H. Deconinck, H. Paillère, P.L. Roe, Multidimensional upwinding: Its relation to finite elements. *Int. J. Numer. Methods Fluids* **20**:8-9 (1995) 935–955.
55. G.F. Carey, *Computational Grids: Generation, Adaptation, and Solution Strategies*. Taylor & Francis, 1997.
56. L. Catabriga and A.L.G.A. Coutinho, Implicit SUPG solution of Euler equations using edge-based data structures. *Comput. Methods Appl. Mech. Engrg.* **191** (2002) 3477–3490.
57. L. Catabriga, M.A.D. Martins, A.L.G.A. Coutinho, J.L.D. Alves, An Edge-based Preconditioner for Non-symmetric Finite Element Equations, In: proceedings of the *18th Ibero-Latin American Congress on Computational Methods in Engineering*. Brazil, October 29-31, 1997, Vol. 3, 1249–1255.
58. L. Catabriga, M.A.D. Martins, A.L.G.A. Coutinho, J.L.D. Alves, Clustered edge-by-edge preconditioners for non-symmetric finite element equations. In: CD-ROM proceedings of the *4th World Congress on Computational Mechanics*. Buenos Aires, Argentina, 29 June to 2 July, 1998.
59. K.-Y. Chien, Predictions of channel and boundary-layer flows with a low-Reynolds-number turbulence model. *AIAA Journal* **20** (1982) 33–38.
60. A.J. Chorin, Numerical solution of the Navier–Stokes equations. *Math. Comp.* **22** (1968) 745–762.
61. I. Christie and C. Hall, The maximum principle for bilinear elements. *Int. J. Numer. Methods Engrg.* **20** (1984) 549–553.
62. M.A. Christon, P.M. Gresho, S.B. Sutton, Computational predictability of natural convection flows in enclosures. In: K. J. Bathe (ed.), *Computational Fluid and Solid Mechanics*, Elsevier, 2001, 1465–1468.
63. P.G. Ciarlet, Discrete maximum principle for finite-difference operators. *Aequationes Math.* **4** (1970) 338–352.
64. P.G. Ciarlet and P.-A. Raviart, Maximum principle and convergence for the finite element method. *Comput. Methods Appl. Mech. Engrg.* **2** (1973) 17–31.
65. R. Clift, J.R. Grace, M.E. Weber, *Bubbles, Drops and Particles*. Academic Press, 1978.
66. B. Cockburn, G.E. Karniadakis, C.-W. Shu, The development of discontinuous Galerkin methods. In: *Discontinuous Galerkin Methods. Theory, Computation and Applications*, LNCSE **11**, Springer, 2000, 3–50.
67. B. Cockburn and C.-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.* **16** (2001) 173–261.
68. B. Cockburn and C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V. Multidimensional Systems. *J. Comput. Phys.* **141** (1998) 199–224.

69. R. Codina, A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection-diffusion equation. *Comput. Methods Appl. Mech. Engrg.* **110** (1993) 325–342.
70. R. Codina, Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Comput. Methods Appl. Mech. Engrg.* **156**:1–4 (1998) 185–210.
71. R. Codina and M. Cervera, Block-iterative algorithms for nonlinear coupled problems. In: M. Papadrakakis and G. Bugeda (eds), *Advanced Computational Methods in Structural Mechanics*, Chapter 7. Theory and Engineering Applications of Computational Methods, CIMNE, Barcelona, 1996.
72. R. Codina and O. Soto, Finite element implementation of two-equation and algebraic stress turbulence models for steady incompressible flows. *Int. J. Numer. Methods Fluids* **30**:3 (1999) 309–333.
73. G. Cohen, P. Joly, J.E. Roberts, N. Tordjman, Higher order triangular finite elements with mass lumping for the wave equation. *SIAM J. Numer. Anal.* **38**:6 (2001) 2047–2078.
74. G. Comini, M. Manzan, C. Nonino, Analysis of finite element schemes for convection-type problems. *Int. J. Numer. Methods Fluids* **20** (1995) 443–458.
75. M. Crouzeix and P.A. Raviart, Conforming and non-conforming finite elements for solving the stationary Stokes equations. *R.A.I.R.O. Anal. Numér.* **7** (1973) 33–76.
76. C. Cuvelier, A. Segal, A.A. van Steenhoven, *Finite Element Methods and Navier-Stokes Equations*. Kluwer, 1986.
77. J.F. Dannenhoffer and J.R. Baron, Robust grid adaptation for complex transonic flows. *AIAA Paper*, 86-0495, 1986.
78. D.L. Darmofal and B. van Leer, Local preconditioning: Manipulating mother nature to fool father time. In: D. A. Caughey et al. (eds), *Frontiers of Computational Fluid Dynamics*, Singapore: World Scientific Publishing, 1998, 211–239.
79. H. Deconinck, H. Paillère, R. Struijs, P.L. Roe, Multidimensional upwind schemes based on fluctuation-splitting for systems of conservation laws. *Comput. Mech.* **11**:5-6 (1993) 323–340.
80. C.R. DeVore, An improved limiter for multidimensional flux-corrected transport. *NASA Report*, AD-A360122, 1998.
81. M.J. Díaz, F. Hecht, B. Mohammadi, New progress in anisotropic grid adaptation for inviscid and viscous flows simulations. In: *Proceedings of the 4th Annual International Meshing Roundtable*. Sandia National Laboratories, 1995.
82. G.S. Dietachmayer, A comparison and evaluation of some positive definite advection schemes. In: J. Noyle and R. May (eds), *Computational Techniques and Applications*, Elsevier, 1986, 217–232.
83. J. Donea, A Taylor-Galerkin method for convective transport problems. *Int. J. Numer. Meth. Engrg.* **20** (1984) 101–120.
84. J. Donea, S. Giuliani, H. Laval, L. Quartapelle, Finite element solution of the unsteady Navier-Stokes equations by a fractional step method. *Comput. Methods Appl. Mech. Engrg.* **30** (1982) 53–73.
85. J. Donea, S. Giuliani, H. Laval, L. Quartapelle, Time-accurate solution of advection-diffusion equations by finite elements. *Comput. Methods Appl. Mech. Engrg.* **193** (1984) 123–145.
86. J. Donea and A. Huerta, *Finite Element Methods for Flow Problems*. John Wiley & Sons, Chichester, 2003.
87. J. Donea, V. Selmin, L. Quartapelle, Recent developments of the Taylor-Galerkin method for the numerical solution of hyperbolic problems. *Numerical methods for fluid dynamics III*, Oxford, 1988, 171–185.
88. J. Donea and L. Quartapelle, An introduction to finite element methods for transient advection problems. *Comput. Methods Appl. Mech. Engrg.* **95** (1992) 169–203.
89. J. Donea, L. Quartapelle, V. Selmin, An analysis of time discretization in the finite element solution of hyperbolic problems. *J. Comput. Phys.* **70** (1987) 463–499.
90. J. Donea, B. Roig, A. Huerta, *High-Order Accurate Time-Stepping Schemes for Convection-Diffusion Problems*. Monograph CIMNE **42**, Barcelona, 1998.

91. J. Douglas Jr. and T. Dupont, Interior penalty procedures for elliptic and parabolic Galerkin methods. In: R. Glowinski and J.-L. Lions (eds) *Computing Methods in Applied Sciences*, Lecture Notes in Physics, vol. **58**, Springer, Berlin, 1976, 207–216.
92. J. Douglas, Jr. and J. Wang, An absolutely stabilized finite element method for the Stokes problem. *Math. Comp.* **52** (1989) 495–508.
93. D.A. Drew and S.L. Passman, *Theory of Multicomponent Fluids*. Springer, 1999.
94. D. Drikakis and W. Rider, *High-Resolution Methods for Incompressible and Low-Speed Flows: Fundamentals and Applications*. Springer, 2004.
95. J.-J. Droux and T.J.R. Hughes, A boundary integral modification of the Galerkin least squares formulation for the Stokes problem. *Comput. Methods Appl. Mech. Engrg.* **113** (1994) 173–182.
96. K. Duraisamy, K.D. Baeder, J.-G. Liu, Concepts and application of time-limiters to high resolution schemes. *J. of Sci. Comput.* **19**:1-3 (2003) 139–162.
97. M.S. Engelman, V. Haroutunian, I. Hasbani, Segregated finite element algorithms for the numerical solution of large–scale incompressible flow problems. *Int. J. Numer. Methods Fluids* **17** (1993) 323–348.
98. M.S. Engelman, R.L. Sani, P.M. Gresho, The implementation of normal and/or tangential boundary conditions in finite element codes for incompressible fluid flow. *Int. J. Numer. Methods Fluids* **2** (1982) 225–238.
99. K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Introduction to adaptive methods for differential equations. *Acta Numerica* **4** (1995) 105–158.
100. I. Faragó, R. Horváth, S. Korotov, Discrete maximum principle for linear parabolic problems solved on hybrid meshes. *Appl. Numer. Math.* **53** (2005) 249–264.
101. I. Faragó and R. Horváth, Continuous and discrete parabolic operators and their qualitative properties. *IMA J. Numer. Anal.* In press, doi:10.1093/imanum/drn032.
102. R.T. Farouki, T.N.T. Goodman, and T. Sauer, Construction of orthogonal bases for polynomials in Bernstein form on triangular and simplex domains. *Computer Aided Geometric Design* **20** (2003) 209–230.
103. M. Feistauer, J. Felcman, I. Straškraba, *Mathematical and Computational Methods for Compressible Flow*. Clarendon Press, Oxford, 2003.
104. J.H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics*. Springer, 1996.
105. J.H. Ferziger and M. Perić, Further discussion of numerical errors in CFD. *Int. J. Numer. Methods Fluids* **23** (1996) 1–12.
106. *FIDAP 8 Theory Manual*, December 1998.
107. B. A. Finlayson. *Numerical Methods for Problems with Moving Fronts*. Ravenna Park Publishing, Seattle, 1992.
108. J.E. Flaherty, L. Krivodonova, J.-F. Remacle, and M.S. Shephard, Aspects of discontinuous Galerkin methods for hyperbolic conservation laws. *Finite Elements in Analysis and Design* **38** (2002) 889–908.
109. C.A.J. Fletcher, The group finite element formulation, *Comput. Methods Appl. Mech. Engrg.* **37** (1983) 225–243.
110. C.A.J. Fletcher, A comparison of finite element and finite difference solutions of the one- and two-dimensional Burgers' equations. *J. Comput. Phys.* **51** (1983) 159–188.
111. C.A.J. Fletcher, *Computational Techniques for Fluid Dynamics*. Springer, 1988.
112. P.A. Forsyth. A control volume finite element approach to NAPL groundwater contamination. *SIAM J. Sci. Stat. Comput.* **12** (1991) 1029–1057.
113. L.P. Franca, S.L. Frey, T.J.R. Hughes, Stabilized finite element methods: I. Application to the advective-diffusive model. *Comput. Methods Appl. Mech. Engrg.* **95** (1992) 253–276.
114. L.A. Freitag, On combining Laplacian and optimization-based mesh smoothing techniques. *Trends in Unstructured Mesh Generation* ASME Applied Mechanics Division, **220** (1997) 37–44.
115. P.J. Frey and P.L. George, *Automatic Mesh Generation, Applications to Finite Methods*. Hermes Science Publishing, Oxford, Paris, 2000.

116. H. Fujii, Some remarks on finite element analysis of time-dependent field problems. In: Y. Yamada and R.H. Gallagher (eds), *Theory and Practice in Finite Element Structural Analysis*, Univ. Tokyo Press, Tokyo, 1973, 91–106.
117. A.C. Galeão and E.G.D. do Carmo, A consistent approximate upwind Petrov-Galerkin Method for convection-dominated problems. *Comput. Methods Appl. Mech. Engrg.* **68**:1 (1988) 83–95.
118. P.L. George, *Automatic Mesh Generation, Applications to Finite Methods*. John Wiley & Sons, New York, 1991.
119. J. Geppert, *Adaptive Gitterverfeinerung bei Flux-Limiter-Verfahren*. PhD thesis, University of Hamburg, 1996.
120. D. Gilbarg and N.S. Trudinger, *Elliptic Partial Differential Equations of Second Order* (2nd edition). Grudlehren der Mathematischen Wissenschaften **224**, Springer, 1983.
121. V. Girault and P.A. Raviart, *Finite Element Methods for Navier–Stokes Equations*, Springer, Berlin-Heidelberg, 1986.
122. R. Glowinski, *Finite Element Methods for Incompressible Viscous Flow*. In: P.G. Ciarlet and J.L. Lions (eds), Handbook of Numerical Analysis, Vol. IX: Numerical Methods for Fluids (Part 3), North-Holland, Amsterdam, 2003, 3–1176.
123. S.K. Godunov, Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb.* **47** (1959) 271–306.
124. J.B. Goodman and R.J. LeVeque, On the accuracy of stable schemes for 2D scalar conservation laws. *Math. Comp.* **45** (1985) 15–21.
125. S. Gottlieb and C.-W. Shu, Total Variation Diminishing Runge-Kutta schemes. *Math. Comp.* **67** (1998) 73–85.
126. S. Gottlieb, C.-W. Shu, and E. Tadmor, Strong stability-preserving high-order time discretization methods. *SIAM Review* **43** (2001) 89–112.
127. P.M. Gresho, On the theory of semi–implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix, Part 1: Theory, Part 2: Implementation. *Int. J. Numer. Methods Fluids* **11** (1990) 587–659.
128. P.M. Gresho and R.L. Lee, Don't suppress the wiggles — they're telling you something! In: T.J.R. Hughes (ed.), *Finite Element Methods for Convection Dominated Flows*, American Society of Mechanical Engineers, New York, 1979, 37-61.
129. P.M. Gresho and R.L. Sani, *Incompressible Flow and the Finite Element Method. Vol. 1: Advection-diffusion, Vol. 2: Isothermal laminar flow*. John Wiley & Sons, 2000.
130. F.F. Grinstein and C. Fureby, On monotonically integrated Large Eddy Simulation of turbulent flows based on FCT algorithms. In: D. Kuzmin, R. Löhner, S. Turek (eds) *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, 2005, 79–104.
131. W. Hackbusch, *Multigrid Methods and Applications*. Springer, Berlin, 1985.
132. W. Hackbusch, *Elliptic Differential Equations: Theory and Numerical Treatment*. Springer, Berlin, 1992
133. K. Hain, The partial donor cell method. *J. Comput. Phys.* **73** (1987) 131–147.
134. P. Hansbo, Aspects of conservation in finite element flow computations. *Comput. Methods Appl. Mech. Engrg.* **117** (1994) 423–437.
135. P. Hansbo, A free-Lagrange finite element method using space-time elements. *Comput. Methods Appl. Mech. Engrg.* **188**:1–3 (2000) 347–361.
136. P. Hansbo and C. Johnson, Adaptive streamline diffusion methods for compressible flow using conservation variables. *Comput. Methods Appl. Mech. Engrg.* **87** (1991) 267–280.
137. A. Harten, High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.* **49** (1983) 357–393.
138. A. Harten, On a class of high resolution total-variation-stable finite-difference-schemes. *SIAM J. Numer. Anal.* **21** (1984) 1-23.
139. A. Harten and J. Hyman, Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.* **50** (1983) 235–269.
140. A. Harten, J. Hyman, P. Lax, On finite-difference approximations and entropy conditions for shocks. *Comm. Pure Appl. Math.* **29** (1976) 297–322.

141. R. Hartmann, Adaptive FE methods for conservation equations. In: H. Freistühler and G. Warnecke (eds.), *Hyperbolic Problems: Theory, Numerics, Applications*, ISNM **141**. Birkhäuser, Basel, 2001, 495-503.
142. R. Hartmann and P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.* **183** (2002) 508–532.
143. F.K. Hebeker and Yu. A. Kuznetsov, Unsteady convection and convection-diffusion problems via direct overlapping domain decomposition methods. *Numer. Meth. PDEs* **14**:3 (1998) 387–406.
144. P.W. Hemker and B. Koren, Defect correction and nonlinear multigrid for steady Euler equations. In: W.G. Habashi and M.M. Hafez (eds), *Computational Fluid Dynamics Techniques*. London: Gordon and Breach Publishers, 1995, 699–718.
145. D. Hempel, *Rekonstruktionsverfahren auf unstrukturierten Gittern zur numerischen Simulation von Erhaltungsprinzipien*. PhD thesis, University of Hamburg, 1999.
146. R. Herbin and F. Hubert, Benchmark on discretization methods for anisotropic diffusion problems on general grids. In: R. Eymard and J.-M. Herard (eds), *Finite Volumes for Complex Applications* **V**, 2008, 659–692.
147. J.S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications.* Springer Texts in Applied Mathematics **54**, Springer, New York, 2008.
148. E. Hinton and J. Campbell, Local and global smoothing of discontinuous finite element functions using least squares method. *Int. J. Numer. Methods Engrg.* **8**:3 (1974) 461–480.
149. C. Hirsch, *Numerical Computation of Internal and External Flows. Vol. I: Fundamentals of Numerical Discretization*. John Wiley & Sons, Chichester, 1990.
150. C. Hirsch, *Numerical Computation of Internal and External Flows. Vol. II: Computational Methods for Inviscid and Viscous Flows*. John Wiley & Sons, Chichester, 1990.
151. W. Höhn and H.-D. Mittelmann, Some remarks on the discrete maximum-principle for finite elements of higher order. *Computing* **27**:2 (1981) 145–154.
152. E. Hopf, Elementare Bemerkungen über die Lösungen partieller Differentialgleichungen zweiter Ordnung vom elliptischen Typus, *Sitzungsber. Preuss. Akad. Wissensch.* **19** (1927) 147–152.
153. R. A. Horn and C.R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, 1991.
154. H. Hoteit, Ph. Ackerer, R. Mosé, J. Erhel, and B. Philippe, New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes. *Int. J. Numer. Meth. Engrg.* **61** (2004) 2566–2593.
155. P. Houston, E. Süli, J.A. Mackenzie, G. Warnecke, A posteriori error analysis for numerical approximations of Friedrichs systems. *Numer. Math.* **82**:3 (1999) 433–470.
156. M.E. Hubbard, Non-oscillatory third order fluctuation splitting schemes for steady scalar conservation laws. *J. Comput. Phys.* **222** (2007) 740–768.
157. T.J.R. Hughes and A. Brooks, A multidimensional upwind scheme with no crosswind diffusion. In: T.J.R. Hughes (ed.), *Finite Element Methods for Convection Dominated Flows*, AMD **34**, AMSE, New York, 1979, 19–35.
158. T.J.R. Hughes, G. Engel, L. Mazzei, M.G. Larson, The continuous Galerkin method is locally conservative. *J. Comput. Phys.* **163**:2 (2000) 467–488.
159. T.J.R. Hughes, L. Franca, G. Hulbert, A new finite element formulation for fluid dynamics: VIII. The Galerkin least-squares method for advective-diffusive equations. *Comput. Methods Appl. Mech. Engrg.* **73** (1989) 173–189.
160. T.J.R. Hughes, M. Mallet and A. Mizukami, A new finite element formulation for computational fluid dynamics. II. Beyond SUPG. *Comput. Methods Appl. Mech. Engrg.* **54**:3 (1986) 341–355.
161. T.J.R. Hughes and M. Mallet, A new finite element formulation for computational fluid dynamics. IV. A discontinuity capturing operator for multidimensional advective-diffusive systems. *Comput. Methods Appl. Mech. Engrg.* **58** (1986) 329–336.
162. T.J.R. Hughes and G.N. Wells, Conservation properties for the Galerkin and stabilised forms of the advection-diffusion and incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.* **194**:9-11 (2005) 1141–1159. Erratum: *Comput. Methods Appl. Mech. Engrg.* **195**:9–12 (2006) 1277–1278.

163. W. Hundsdorfer and C. Montijn, A note on flux limiting for diffusion discretizations. *IMA J. of Numer. Anal.* **24**:4 (2004) 635–642.
164. W. Hundsdorfer and J.G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer, 2003.
165. S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan and L. Tobiska, Quantitative benchmark computations of two-dimensional bubble dynamics. *Int. J. Numer. Meth. Fluids*. In press, doi: 10.1002/fld.1934
166. S. Idelsohn and E. Oñate, Finite elements and finite volumes. Two good friends. *Int. J. Numer. Methods Engrg.* **37** (1994) 3323–3341.
167. T. Ikeda, *Maximum Principle in Finite Element Models for Convection-Ciffusion Phenomena*. North-Holland: Mathematics Studies **4**, Kinokuniya, Tokyo, 1983.
168. F. Ilinca, J.-F. Hétu, D. Pelletier, A unified finite element algorithm for two-equation models of turbulence. *Comp. & Fluids* **27**:3 (1998) 291–310.
169. C. Ilinca, X.D. Zhang, J.Y. Trépanier, R. Camarero, A comparison of three error estimation techniques for finite-volume solutions of compressible flows. *Comput. Methods Appl. Engrg.* **189**:4 (2000) 1277–1294.
170. A. Jameson, Computational algorithms for aerodynamic analysis and design. *Appl. Numer. Math.* **13** (1993) 383–422.
171. A. Jameson, Analysis and design of numerical schemes for gas dynamics 1. Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *Int. Journal of CFD* **4** (1995) 171–218.
172. V. John and P. Knobloch, On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations: Part I - A review. *Comput. Methods Appl. Mech. Engrg.* 196:17–20 (2007) 2197–2215.
173. V. John and P. Knobloch, On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations: Part II - Analysis for P_1 and Q_1 finite elements. *Comput. Methods Appl. Mech. Engrg.* **197** (2008) 1997–2014.
174. V. John and E. Schmeyer, Finite element methods for time-dependent convection-diffusion-reaction equations with small diffusion. Preprint, Saarland University, 2008.
`http://www.math.uni-sb.de/ag/john/tcd.pdf`
175. V. John and E. Schmeyer, On finite element methods for 3D time-dependent convection-diffusion-reaction equations with small diffusion. Preprint, Saarland University, 2008.
`http://www.math.uni-sb.de/ag/john/john_ schmeyer.bail2008.pdf`
176. C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Studentlitteratur, Lund, 1987
177. C. Johnson, A. Szepessy, P. Hansbo, On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws. *Math. Comput.* **54** (1990) 107–129.
178. T. Jongen and Y.P. Marx, Design of an unconditionally stable, positive scheme for the $K-\varepsilon$ and two-layer turbulence models. *Comput. Fluids* **26** (1997) no. 5, 469-487.
179. J. Karátson, S. Korotov, M. Křížek, On discrete maximum principles for nonlinear elliptic problems. *Math. Comp. Simulation* **76** (2007) 99–108.
180. J. Karátson and S. Korotov, Discrete maximum principles for finite element solutions of nonlinear elliptic problems with mixed boundary conditions. *Numer. Math.* **99** (2005) 669–698.
181. D.W. Kelly, S. Nakazawa, O.C. Zienkiewicz, J.C. Heinrich, A note on anisotropic balancing dissipation in finite element approximation to convection diffusion problems, *Int. J. Numer. Methods Engrg.* **15** (1980) 1705–1711.
182. J. Kim, *Investigation of Separation and Reattachment of Turbulent Shear Layer: Flow over a Backward Facing Step*. PhD thesis, Stanford University, 1978.
183. J. Kim, P. Moin, R.D. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.*, **177** (1987) 133–166.
184. B. Koren, A robust upwind discretization method for advection, diffusion and source terms. In: C.B. Vreugdenhil et al. (eds), *Numerical methods for advection - diffusion problems*. Braunschweig: Vieweg. *Notes Numer. Fluid Mech.* **45** (1993) 117–138.

185. S. Korotov, P. Neittaanmäki, S. Repin, A posteriori error estimation of goal-oriented quantities by the superconvergence patch recovery. *J. Numer. Math.* **11** (2003) 33-59.
186. L. Krivodonova, Limiters for high-order discontinuous Galerkin methods. *J. Comput. Phys.* **226** (2007) 879–896.
187. L. Krivodonova and M. Berger, High-order accurate implementation of solid wall boundary conditions in curved geometries. *J. Comput. Phys.* **211** (2006) 492–512.
188. L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, and J.E. Flaherty, Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. *Appl. Numer. Math.* **48** (2004) 323–338.
189. M. Křížek and Q. Lin, On diagonal dominance of stiffness matrices in 3D. *East-West J. Numer. Math.* **3** (1995) 59–69.
190. M. Křížek and P. Neittaanmäki, On superconvergence techniques. *Acta Appl. Math.* **9** (1987) 175–198.
191. D. Kuzmin, Positive finite element schemes based on the flux-corrected transport procedure. In: K. J. Bathe (ed.), *Computational Fluid and Solid Mechanics*, Elsevier, 2001, 887–888.
192. D. Kuzmin, On the design of general-purpose flux limiters for implicit FEM with a consistent mass matrix. I. Scalar convection. *J. Comput. Phys.* **219** (2006) 513–531.
193. D. Kuzmin, On the design of algebraic flux correction schemes for quadratic finite elements. *J. Comput. Appl. Math.* **218**:1 (2008) 79–87.
194. D. Kuzmin, Algebraic flux correction for finite element discretizations of coupled systems. In: E. Oñate, M. Papadrakakis, B. Schrefler (eds) *Computational Methods for Coupled Problems in Science and Engineering II*, CIMNE, Barcelona, 2007, 653–656.
195. D. Kuzmin, A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods. *J. Comput. Appl. Math.* **233** (2010) 3077–3085.
196. D. Kuzmin, Explicit and implicit FEM-FCT algorithms with flux linearization. *J. Comput. Phys.* In press, doi:10.1016/j.jcp.2008.12.011
197. D. Kuzmin and S. Korotov, Goal-oriented a posteriori error estimates for transport problems. *Math. Comput. Simul.* **80** (2010) 2674-1683.
198. D. Kuzmin, O. Mierka, S. Turek, On the implementation of the $k-\varepsilon$ turbulence model in incompressible flow solvers based on a finite element discretization. *Int. J. Comp. Sci. Math.* **1**:2–4 (2007) 193-206.
199. D. Kuzmin and M. Möller, Goal-oriented mesh adaptation for flux-limited approximations to steady hyperbolic problems. *J. Comput. Appl. Math.* **233** (2010) 3113–3120.
200. D. Kuzmin and M. Möller, Algebraic flux correction I. Scalar conservation laws. In: D. Kuzmin, R. Löhner, S. Turek (eds) *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, 2005, 155–206.
201. D. Kuzmin and M. Möller, Algebraic flux correction II. Compressible Euler equations. In: D. Kuzmin, R. Löhner, S. Turek (eds) *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, 2005, 207–250.
202. D. Kuzmin, M. Möller and S. Turek, Multidimensional FEM-FCT schemes for arbitrary time-stepping. *Int. J. Numer. Methods Fluids* **42** (2003) 265–295.
203. D. Kuzmin, M. Möller, S. Turek, High-resolution FEM-FCT schemes for multidimensional conservation laws. *Computer Methods Appl. Mech. Engrg.* **193** (2004) 4915–4946.
204. D. Kuzmin, M. Shashkov, D. Svyatskiy, A constrained finite element method satisfying the discrete maximum principle for anisotropic diffusion problems on arbitrary meshes. *Appl. Math. Report* **373**, University of Dortmund, 2008. Submitted to *J. Comput. Phys.*
205. D. Kuzmin and S. Turek, Flux correction tools for finite elements. *J. Comput. Phys.* **175** (2002) 525–558.
206. D. Kuzmin and S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *J. Comput. Phys.* **198** (2004) 131–158.
207. D. Kuzmin and S. Turek, Multidimensional FEM-TVD paradigm for convection-dominated flows. In: P. Neittaanmäki, T. Rossi, K. Majava, O. Pironneau (eds) CD-ROM Proceedings of the *IV European Congress on Computational Methods in Applied Sciences and Engineering* (ECCOMAS 2004), Jyväskylä, Finland, 24-28 July 2004. Vol. II, ISBN 951-39-1869-6.

208. D. Kuzmin and S. Turek, Numerical simulation of turbulent bubbly flows. In: G.P. Celata, P. Di Marco, A. Mariani, R.K. Shah (eds) *Two-Phase Flow Modeling and Experimentation*. Edizioni ETS, Pisa, 2004, Vol. I, 179-188.
209. Yu. A. Kuznetsov, New algorithms for approximate realization of implicit difference schemes. *Soviet. J. Numer. Anal. Math. Modelling* **3** (1988) 99–114.
210. Yu. A. Kuznetsov, Domain decomposition methods for unsteady convection diffusion problems. *Comput. Methods Appl. Sci. Engin.* (Proceedings of the Ninth International Conference, Paris 1990) SIAM, Philadelphia (1990) 211–227.
211. O.A. Ladyzhenskaya and N.N. Ural'tseva, *Linear and Quasilinear Elliptic Equations*. Academic Press, New York, 1968.
212. R.J. Labeur and G.N. Wells, A Galerkin interface stabilisation method for the advection-diffusion and incompressible Navier-Stokes equations. *Computer Meth. Appl. Mech. Engrg.* **196** (2007) 4985–5000.
213. M. Laforest, M.A. Christon, T.E. Voth, A survey of error indicators and error estimators for hyperbolic problems. Available online at `http://www.mgi.polymtl.ca/marc.laforest/pages/sand_rep.ps`.
214. A. Lapin, University of Stuttgart. Private communication, 2001.
215. C. Le Potier, Schema volumes finis monotone pour des operateurs de diffusion fortement anisotropes sur des maillages de triangle non structures. *C.C. Acad. Sci. Paris*, Ser. I, **341** (2005) 787–792.
216. R.J. LeVeque, *Numerical Methods for Conservation Laws*. Birkhäuser, 1992.
217. R.J. LeVeque, CLAWPACK – a software package for solving multi-dimensional conservation laws. In: proceedings of the *5th International Conference on Hyperbolic Problems*, `http://www.amath.washington.edu/~claw/`, 1994.
218. R.J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.* **33** (1996) 627–665.
219. R.J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, 2002.
220. A.J. Lew, G.C. Buscaglia, P.M. Carrica, A note on the numerical treatment of the $k-\varepsilon$ turbulence model. *Int. J. of Comp. Fluid Dyn.* **14** (2001) 201–209.
221. K. Lipnikov, M. Shashkov, D. Svyatskiy, Yu. Vassilevski, Monotone finite volume schemes for diffusion equations on unstructured triangular and shape-regular polygonal meshes. *J. Comput. Phys.* **227** (2007) 492–512.
222. R. Liska and M. Shashkov, Enforcing the discrete maximum principle for linear finite element solutions of second-order elliptic problems. *Commun. Comput. Phys.* **3**:4 (2007) 852–877.
223. R. Löhner, An adaptive finite element scheme for transient problems in CFD. *Comput. Methods Appl. Mech. Engrg.* **61** (1987) 323–338.
224. R. Löhner, Adaptive remeshing for transient problems. *Comput. Methods Appl. Mech. Engrg.* **75** (1989) 195–214.
225. R. Löhner, Edges, stars, superedges and chains. *Comput. Methods Appl. Mech. Engrg.* **111** (1994) 255–263.
226. R. Löhner, *Applied CFD Techniques: An Introduction Based on Finite Element Methods* (2nd edition). John Wiley & Sons, Chichester, 2008.
227. R. Löhner and J.D. Baum, Adaptive h-refinement on 3-D unstructured grids for transient problems. *Int. J. Numer. Methods Fluids* **14** (1992) 1407–1419.
228. R. Löhner and J.D. Baum, 30 Years of FCT: Status and Directions. In: D. Kuzmin, R. Löhner, S. Turek (eds) *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, 2005, 131–154.
229. R. Löhner and M. Galle, Minimization of indirect addressing for edge-based field solvers. *Commun. Numer. Methods Eng.* **18**:5 (2002) 335–343.
230. R. Löhner, K. Morgan, O.C. Zienkiewicz, The solution of non-linear hyperbolic equation systems by the finite element method. *Int. J. Numer. Methods Fluids* **4** (1984) 1043–1063.
231. R. Löhner, K. Morgan, O.C. Zienkiewicz, An adaptive finite element procedure for compressible high speed flows. *Comput. Methods Appl. Mech. Engrg.* **51** (1985) 441–465.

232. R. Löhner, K. Morgan, J. Peraire, M. Vahdati, Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier-Stokes equations. *Int. J. Numer. Meth. Fluids* **7** (1987) 1093–1109.
233. R. Löhner, K. Morgan, M. Vahdati, J.P. Boris, D.L. Book, FEM-FCT: combining unstructured grids with high resolution. *Commun. Appl. Numer. Methods* **4** (1988) 717–729.
234. R. Löhner, H. Luo, J.D. Baum, Selective edge removal for unstructured grids with Cartesian cores. *J. Comput. Phys.* **206**:1 (2005) 208–226.
235. H. Luo, J.D. Baum, R. Löhner, Edge-based finite element scheme for the Euler equations. *AIAA Journal* **32** (1994) 1183–1190.
236. H. Luo, J.D. Baum, and R. Löhner, Fast *p*-multigrid discontinuous Galerkin method for compressible flows at all speeds. *AIAA Journal* **46** (2008) 635–652.
237. H. Luo, J.D. Baum, and R. Löhner, A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids. *J. Comput. Phys.* **227** (2008) 8875–8893.
238. J.F. Lynn, *Multigrid Solution of the Euler Equations with Local Preconditioning*. PhD thesis, University of Michigan, 1995.
239. P.R.M. Lyra, *Unstructured Grid Adaptive Algorithms for Fluid Dynamics and Heat Conduction*. PhD thesis, University of Wales, Swansea, 1994.
240. P.R.M. Lyra, O. Hassan, K. Morgan, Adaptive unstructured grid solutions of hypersonic viscous flows. In: K.W. Morton and M.J. Baines (Eds) *Numerical Methods for Fluid Dynamics* **V**, Oxford University Press, Oxford, 1996, 465–472.
241. P.R.M. Lyra and K. Morgan, A review and comparative study of upwind biased schemes for compressible flow computation. I: 1-D first-order schemes. *Arch. Comput. Methods Eng.* **7**:1 (2000) 19–55.
242. P.R.M. Lyra and K. Morgan, A review and comparative study of upwind biased schemes for compressible flow computation. II: 1-D higher-order schemes. *Arch. Comput. Methods Eng.* **7**:3 (2000) 333–377.
243. P.R.M. Lyra and K. Morgan, A review and comparative study of upwind biased schemes for compressible flow computation. III: Multidimensional extension on unstructured grids. *Arch. Comput. Methods Eng.* **9**:3 (2002) 207–256.
244. P.R.M. Lyra, K. Morgan, J. Peraire, J. Peiro, TVD algorithms for the solution of the compressible Euler equations on unstructured meshes. *Int. J. Numer. Methods Fluids* **19** (1994) 827–847.
245. P.R.M. Lyra, R.B. Willmersdorf, M.A.D. Martins, A.L.G.A. Coutinho, Parallel implementation of edge-based finite element schemes for compressible flow on unstructured grids, Proceedings of the 3rd International Meeting on Vector and Parallel Processing, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 21.-23. Juni 1998.
246. R.J. MacKinnon and G.F. Carey, Positivity preserving flux-limited finite-difference and finite-element methods for reactive transport. *Int. J. Numer. Methods Fluids* **41** (2003) 151–183.
247. G. Medić and B. Mohammadi, NSIKE - an incompressible Navier-Stokes solver for unstructured meshes. *INRIA Research Report* **3644**, 1999.
248. D. Meidner, R. Rannacher and J. Vihharev, Goal-oriented error control of the iterative solution of finite element equations. *J. Numer. Math.* **17** (2009) 143–172.
249. J.A. Meijerink and H.A. van der Vorst, Iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comput.* **31** (1977) 148–162.
250. K. Mer, Variational analysis of a mixed element/volume scheme with fourth-order viscosity on general triangulations. *Comput. Methods Appl. Mech. Engrg.* **153** (1998) 45–62.
251. K. Michalak and C. Ollivier-Gooch, Limiters for unstructured higher-order accurate solutions of the Euler equations. In: Proceedings of the *AIAA Forty-Sixth Aerospace Sciences Meeting*, 2008.
252. B. Mohammadi and O. Pironneau, *Analysis of the k-epsilon Turbulence Model*. John Wiley & Sons, 1994.
253. B. Mohammadi and O. Pironneau, *Applied Shape Optimization for Fluids*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, 2001.

254. M. Möller, Efficient solution techniques for implicit finite element schemes with flux limiters. *Int. J. Numer. Methods Fluids* **55**:7 (2007) 611–635.
255. M. Möller, *Adaptive High-Resolution Finite Element Schemes*. PhD thesis, Dortmund University of Technology, 2008.
256. M. Möller and D. Kuzmin, Adaptive mesh refinement for high-resolution finite element schemes. *Int. J. Numer. Methods Fluids* **52** (2006) 545–569.
257. M. Möller and D. Kuzmin, On the use of slope limiters for the design of recovery based error indicators. In: A. Bermudez de Castro, D. Gomez, P. Quintela, P. Salgado (eds), *Numerical Mathematics and Advanced Applications*, Springer, 2006, 233–240.
258. M. Möller, D. Kuzmin, D. Kourounis, Implicit FEM-FCT algorithms and discrete Newton methods for transient convection problems. *Int. J. Numer. Methods Fluids* **57**:6 (2008) 761–792.
259. K. Morgan and J. Peraire, Unstructured grid finite element methods for fluid mechanics. *Reports on Progress in Physics*. **61**:6 (1998) 569–638.
260. K.W. Morton, *Numerical Solution of Convection-Diffusion Problems*. Chapman & Hall, 1996.
261. J.T. Oden and H.J. Brauchli, On the calculation of consistent stress distributions in finite element approximations. *Int. J. Numer. Methods Engrg.* **3** (1971) 317–325.
262. J.T. Oden and S. Prudhomme, Goal-oriented error estimation and adaptivity for the finite element methods. *Comput. Math. Appl.* **41** (2000) 735–756.
263. J.T. Oden and J.N. Reddy, Note on an approximate method for computing consistent conjugate stresses in elastic finite elements. *Int. J. Numer. Methods Engrg.* **6** (1973) 55–61.
264. J.T. Oden, T. Strouboulis, P. Devloo, Adaptive finite element methods for the analysis of inviscid compressible flow: Part I. Fast refinement/unrefinement and moving mesh methods for unstructured meshes. *Comput. Methods Appl. Mech. Engrg.* **59** (1986) 327–362.
265. E.S. Oran and J.P. Boris, *Numerical Simulation of Reactive Flow* (2nd edition). Cambridge University Press, 2001.
266. A.K. Parrott, M.A. Christie, FCT applied to the 2-D finite element solution of tracer transport by single phase flow in a porous medium. In: *Numerical Methods for Fluid Dynamics*, Oxford Univ. Press, London, 1986, 609–619.
267. G. Patnaik, J.P. Boris, F.F. Grinstein, J.P. Iselin, Large scale urban simulations with FCT. In: D. Kuzmin, R. Löhner, S. Turek (eds), *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, 2005, 105–130.
268. S.V. Patankar, *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill, New York, 1980.
269. J. Peraire, M. Vahdati, J. Peiro, K. Morgan, The construction and behaviour of some unstructured grid algorithms for compressible flows. *Numerical Methods for Fluid Dynamics* **IV**, Oxford University Press, 1993, 221-239.
270. O. Pironneau, *Finite Element Methods for Fluids*. John Wiley & Sons, Chichester; Masson, Paris, 1989.
271. L. Postma and J.-M. Hervouet, Compatibility between finite volumes and finite elements using solutions of shallow water equations for substance transport. *Int. J. Numer. Methods Fluids* **53** (2007) 1495–1507.
272. A. Prohl, *Projection and Quasi-Compressibility Methods for Solving the Incompressible Navier-Stokes Equations*. Advances in Numerical Mathematics. B.G. Teubner, Stuttgart, 1997.
273. M.H. Protter and H.F. Weinberger, *Maximum Principles in Differential Equations*. Prentice-Hall, 1967.
274. S. Prudhomme and J.T. Oden, Computable error estimators. In: T.J. Barth and H. Deconinck (eds), *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, LNCSE **25**. Springer, Berlin, 2002, 207–268.
275. L. Quartapelle, *Numerical Solution of the Incompressible Navier-Stokes Equations*. ISNM **113**, Birkäuser, Basel, 1993.
276. A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*. Springer, Berlin, 1994.

277. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, Oxford, 1999.
278. R. Rannacher and S. Turek, A simple nonconforming quadrilateral Stokes element. *Numer. Meth. PDEs* **8**:2 (1992) 97–111.
279. W. J. Rider, Methods for extending high-resolution schemes to non-linear systems of hyperbolic conservation laws. *Int. J. Numer. Methods Fluids* **17**:10 (1993) 861–885.
280. M.C. Rivara, Design and data structure of fully adaptive multigrid finite element software. *ACM Trans. Math. Software* **10** (1984) 242–264.
281. M.C. Rivara, New mathematical tools and techniques for the refinement and/or improvement of unstructured triangulations. *Proc. 5th Int. Meshing Roundtable 96*, Pittsburgh, 1996: 77-86.
282. M.C. Rivara, Using longest-side bisection techniques for the automatic refinement of delaunay triangulations. *Int. J. Numer. Methods Engrg.* **40** (1997) 581–597.
283. P.J. Roache, A method for uniform reporting of grid refinement studies. *ASME J. Fluids Engrg.* **116** (1994) 405–413.
284. P.J. Roache, *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, New Mexico, 1998.
285. P.L. Roe, Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comput. Phys.* **43** (1981) 357–372.
286. A. Rohde, Eigenvalues and eigenvectors of the Euler equations in general geometries. *AIAA Paper*, 2001-2609, 2001.
287. C.J. Roy, Grid convergence error analysis for mixed-order numerical schemes. *AIAA Journal* **41**:4 (2003) 595–604.
288. H.G. Roos, M. Stynes, L. Tobiska, *Numerical Methods for Singularly Perturbed Differential Equations*. Springer, 1996.
289. A. Samarskii and P. Vabishchevich, *Numerical Methods for Solving Convection-Diffusion Problems* (in Russian). Editorial URSS, Moscow, 1999.
290. P.A.B. de Sampaio and A.L.G.A. Coutinho, A natural derivation of discontinuity capturing operator for convection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.* **190** (2001) 6291–6308.
291. E. Schall, D. Leservoisier, A. Dervieux, B. Koobus, Mesh adaptation as a tool for certified computational aerodynamics. *Int. J. Numer. Methods Fluids* **45** (2004) 179–196.
292. M. Schäfer and S. Turek (with support of F. Durst, E. Krause, R. Rannacher), Benchmark computations of laminar flow around cylinder, In E.H. Hirschel (ed.), *Flow Simulation with High-Performance Computers II*, Vol. 52 von *Notes on Numerical Fluid Mechanics*, Vieweg, 1996, 547–566.
293. F. Schieweck, On the role of boundary conditions for CIP stabilization of higher order finite elements. *Electronic Transactions on Numerical Analysis* **32** (2008) 1–16.
294. R. Schmachtel, *Robuste lineare und nichtlineare Lösungsverfahren für die inkompressiblen Navier-Stokes-Gleichungen*. PhD thesis, University of Dortmund, 2003.
295. M. Schmich and B. Vexler, Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations. *SIAM J. Sci. Comput.* **30**:1 (2008) 369–393.
296. P. Schreiber, *A new finite element solver for the nonstationary incompressible Navier–Stokes equations in three dimensions*, PhD thesis, University of Heidelberg, 1996.
297. P. Schreiber and S. Turek, An efficient finite element solver for the nonstationary incompressible Navier–Stokes equations in two and three dimensions. Proc. Workshop *Numerical Methods for the Navier–Stokes Equations*, Heidelberg, Oct. 25–28, 1993, Vieweg.
298. V. Selmin, Finite element solution of hyperbolic equations. I. One-dimensional case. *INRIA Research Report* **655**, 1987.
299. V. Selmin, Finite element solution of hyperbolic equations. I I. Two-dimensional case. *INRIA Research Report* **708**, 1987.
300. V. Selmin, The node-centred finite volume approach: bridge between finite differences and finite elements. *Comput. Methods Appl. Mech. Engrg.* **102** (1993) 107–138.
301. V. Selmin and L. Formaggia, Unified construction of finite element and finite volume discretizations for compressible flows. *Int. J. Numer. Methods Engrg.* **39** (1996) 1–32.

302. V. Selmin and L. Quartapelle, A unified approach to build artificial dissipation operators for finite element and finite volume discretisations. In: K. Morgan et al. (eds), *Finite Elements in Fluids*, CIMNE / Pineridge Press, 1993, 1329–1341.
303. R.A. Shapiro, *Adaptive Finite Element Solution Algorithm for the Euler Equations*. *Notes on Numerical Fluid Mechanics*, Vol. **32**. Vieweg, Braunschweig-Wiesbaden, 1991.
304. C.-W. Shu, Total-variation-diminishing time discretizations. *SIAM J. Sci. Stat. Comput.* **9** (1988) 1073–1084.
305. C.-W. Shu, A Survey of Strong Stability Preserving High Order Time Discretizations. *Scientific Computing Report Series* **2001-18**, Brown University, 2001.
306. C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* **77** (1988) 439–471.
307. T.M. Smith, R.W. Hooper, C.C. Ober, A.A. Lorber, Intelligent Nonlinear Solvers for Computational Fluid Dynamics. Conference Paper, Presentation at the *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno NV, January 2006.
308. G. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.* **27** (1978) 1–31.
309. P. Šolin and L. Demkowicz, Goal-oriented hp-adaptivity for elliptic problems. *Comput. Methods Appl. Mech. Engrg.* **193** (2004) 449–468.
310. C. Steiner and S. Noelle, On adaptive timestepping for weakly instationary solutions of hyperbolic conservation laws via adjoint error control. *Commun. Numer. Meth. Engrg.* (2009), in press, doi:10.1002/cnm.1183.
311. G. Stoyan, On maximum principles for monotone matrices. *Linear Algebra Appl.* **78** (1986) 147–161.
312. E. Süli, A-posteriori error analysis and adaptivity for finite element approximations of hyperbolic problems. In: M. Ohlberger, D. Kröner, C. Rohde (eds), *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws*, LNCSE **5**, Springer, 1998, 123–194.
313. E. Süli, P. Houston, B. Senior, hp-discontinuous Galerkin finite element methods for nonlinear hyperbolic problems. *Int. J. Numer. Methods Fluids* **40**:1–2 (2001) 153–169.
314. E. Süli and P. Houston, Adaptive finite element approximation of hyperbolic problems. In: T.J. Barth and H. Deconinck (eds) *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*. LNCSE **25**. Springer, Berlin, 2002, 269–344.
315. P.K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.* **21** (1984) 995–1011.
316. S. Thangam and C.G. Speziale, Turbulent flow past a backward-facing step: A critical evaluation of two-equation models. *AIAA Journal* **30**:5 (1992) 1314–1320.
317. J. F. Thompson, B. Soni, and N. Weatherill (Eds) *Handbook of Grid Generation*. CRC Press LLC , Boca Raton, FL, 1998,
318. J.F. Thompson, Z.U.A. Warsi, C.W. Mastin, *Numerical Grid Generation: Foundations and Applications*. North-Holland, Amsterdam, 1985.
319. E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, 1999.
320. S. Tu and S. Aliabadi, A slope limiting procedure in discontinuous Galerkin finite element method for gasdynamics applications. *Int. J. Numer. Anal. Model.* **2** (2005) 163–178.
321. S. Turek, On discrete projection methods for the incompressible Navier-Stokes equations: An algorithmical approach. *Comput. Methods Appl. Mech. Engrg.* **143** (1997) 271–288.
322. S. Turek, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, LNCSE 6, Springer, 1999.
323. S. Turek *et al.*, *FEATFLOW: finite element software for the incompressible Navier-Stokes equations*. User manual, University of Dortmund, 2000. Available at *http://www.featflow.de*.
324. S. Turek, C. Becker, S. Kilian, Hardware-oriented numerics and concepts for PDE software. Special Journal Issue for PDE Software, Elsevier, *International Conference on Computational Science ICCS'2002*, Amsterdam, 2002, FUTURE 1095 (2003), 1-23.
325. S. Turek, C. Becker, S. Kilian, Some concepts of the software package FEAST. In: J.M. Palma, J. Dongarra, V. Hernandes (eds), *VECPAR'98 - Third International Conference for Vector and Parallel Processing*, Lecture Notes in Computer Science, Springer, Berlin, 1999.

326. S. Turek and S. Kilian, An example for parallel ScaRC and its application to the incompressible Navier-Stokes equations, Proc. *ENUMATH'97*, World Science Publ., 1998.
327. S. Turek and A. Ouazzi, Unified edge-oriented stabilization of nonconforming FEM for incompressible flow problems: Numerical investigation. *J. Numer. Math.* **15**:4 (2007) 299–322.
328. S. Turek and R. Schmachtel, Fully coupled and operator-splitting approaches for natural convection. *Int. Numer. Meth. Fluids* **40** (2002) 1109–1119.
329. S. Turek and D. Kuzmin, Algebraic flux correction III. Incompressible flow problems. In: D. Kuzmin, R. Löhner, S. Turek (eds), *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, 2005, 251–296.
330. UMFPACK web site, `http://www.cise.ufl.edu/research/sparse/umfpack/`.
331. P. Vabishchevich, Monotone difference schemes for convection/diffusion problems. *Differential Equations* **30**:3 (1994) 466–474.
332. P. Vabishchevich, Iterative methods for solving convection-diffusion problem. *Comput. Meth. Appl. Math.* **2**:4 (2002) 410–444.
333. A.M.P. Valli, G.F. Carey, A.L.G.A. Coutinho, Finite element simulation and control of nonlinear flow and reactive transport. In: *Proceedings of the 10th International Conference on Numerical Methods in Fluids*. Tucson, Arizona, 1998: 450–455.
334. A.M.P. Valli, G.F. Carey, A.L.G.A. Coutinho, Control strategies for timestep selection in simulation of coupled viscous flow and heat transfer. *Commun. Numer. Methods Eng.* **18**:2 (2002) 131–139.
335. S.P. Vanka, Implicit multigrid solutions of Navier–Stokes equations in primitive variables. *J. Comput. Phys.* **65** (1985) 138–158.
336. J. van Kan, A second–order accurate pressure–correction scheme for viscous incompressible flow. *SIAM J. Sci. Stat. Comp.* **7** (1986) 870–891.
337. P. van Slingerland, *An Accurate and Robust Finite Volume Method for the Advection Diffusion Equation*. M.Sc. thesis, Delft University of Technology, June 2007.
338. P. van Slingerland, M. Borsboom, C. Vuik, A local theta scheme for advection problems with strongly varying meshes and velocity profiles. Report **08-17**, Department of Applied Mathematical Analysis, Delft University of Technology, June 2008.
339. R.S. Varga, *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, 1962.
340. R.S. Varga, On discrete maximum principle. *SIAM J. Numer. Anal.* **3** (1966) 355–359.
341. T. Vejchodsky and P. Solin, Discrete maximum principle for higher-order finite elements in 1D. *Math. Comp.* **76** (2007) 1833–1846.
342. V. Venkatakrishnan, Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *J. Comput. Phys.* **118** (1995) 120–130.
343. R. Verfürth, *A Review of a Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. Wiley-Teubner, 1996.
344. P. Wesseling, *Principles of Computational Fluid Dynamics*. Springer, Berlin-Heidelberg, 2001.
345. P. Wesseling, *An Introduction to Multigrid Methods*. John Wiley & Sons, Chichester, 1992. Corrected Reprint. Philadelpha: R.T. Edwards, Inc., 2004.
346. G. Windisch, *M-matrices in Numerical Analysis*. Teubner, Leipzig, 1989.
347. P.R. Woodward and P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.* **54** (1984) 115–173.
348. J. Xu and L. Zikatanov, A monotone finite element scheme for convection-diffusion equations. *Math. Comput.* **68** (1999) 1429–1446.
349. M. Xue, High-order monotonic numerical diffusion and smoothing. *Monthly Weather Review* **128**:8 (2000) 2853–2864.
350. M. Yang and Z.J. Wang, A parameter-free generalized moment limiter for high-order methods on unstructured grids, AIAA-2009-605.
351. H.C. Yee, Numerical approximations of boundary conditions with applications to inviscid gas dynamics. *NASA Report* TM-81265, 1981.
352. H.C. Yee, Construction of explicit and implicit symmetric TVD schemes and their applications. *J. Comput. Phys.* **43** (1987) 151–179.

353. H.C. Yee, R.F. Warming, A. Harten, Implicit Total Variation Diminishing (TVD) schemes for steady-state calculations. *J. Comput. Phys.* **57** (1985) 327–360.
354. D. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.
355. S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31** (1979) 335–362.
356. S.T. Zalesak, A preliminary comparison of modern shock-capturing schemes: linear advection. In: R. Vichnevetsky and R. Stepleman (eds), *Advances in Computer Methods for PDEs*. Publ. IMACS, 1987, 15–22.
357. S.T. Zalesak, The design of Flux-Corrected Transport (FCT) algorithms for structured grids. In: D. Kuzmin, R. Löhner, S. Turek (eds), *Flux-Corrected Transport: Principles, Algorithms, and Applications*. Springer, 2005, 29–78.
358. X.D. Zhang, D. Pelletier, J.-Y. Tr'epanier, R. Camarero, Numerical assessment of error estimators for Euler equations. *AIAA Journal* **39**:9 (2001) 1706–1715.
359. Z. Zhang and A. Naga, A posteriori error estimates based on polynomial preserving recovery. *SIAM J. Numer. Anal.* **42**:4 (2004) 1780–1800.
360. Z. Zhang and A. Naga, A new finite element gradient recovery method: Superconvergence property. *SIAM J. Sci. Comput.* **26**:4 (2005) 1192–1213.
361. O.C. Zienkiewicz and R.L. Taylor, *The Finite Element Method*. Butterworth-Heinemann, Oxford, 2000.
362. O.C. Zienkiewicz and J.Z. Zhu, A simple error estimator and adaptive procedure for practical engineering analysis. *Int. J. Numer. Methods Engrg.* **24**:2 (1987) 337–357.
363. O.C. Zienkiewicz and J.Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery techniques. *Int. J. Numer. Methods Engrg.* **33**:7 (1992) 1331–1364.
364. O.C. Zienkiewicz and J.Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity. *Int. J. Numer. Methods Engrg.* **33**:7 (1992) 1365–1382.
365. M. Zitka, P. Solin, T. Vejchodsky, and F. Avila, Imposing orthogonality to hierarchic higher-order finite elements. *Math. Comput. Simul.* **76** (2007) 211–217.