# Chapter 9

# Adaptive Integration of Parabolic Problems

In the introductory Chapter 1 we made the acquaintence of several elementary time-dependent PDEs: in Section 1.2 the diffusion equation, in Section 1.3 the wave equation, and in Section 1.4 the Schrödinger equation. All of them can be written as *abstract Cauchy problems*, e.g., in the linear special case as

$$u' = Au, \quad u(0) = u_0, \tag{9.1}$$

where $A$ is an unbounded spatial operator, into which the differential operator as well as the boundary conditions enter. The classical prototype of a parabolic differential equation is the diffusion equation. There the operator $-A$ is elliptic and has a spectrum on the positive real half-axis with accumulation point at infinity. For the Schrödinger equation the spectrum lies on the imaginary axis with an accumulation point at infinity, too. In a first step, we want to discuss important aspects of time discretization at these two elementary example PDEs, which are also meaningful for more general cases.

## 9.1 Time Discretization of Stiff Differential Equations

In Sections 9.1.1 and 9.1.2 we treat – in a crash course – the discretization of initial value problems for *ordinary* differential equations; a more extensive treatment of this topic can be found, e.g., in Volume 2. In Section 9.1.3 we will then discuss the transition to parabolic PDEs.

On the background of the above Cauchy problem (9.1) we are especially interested in "stiff" ODEs. For an explanation of this concept we study the following scalar nonautonomous example:

$$u' = g'(t) + \lambda(u - g(t)), \quad u(0) = g(0) + \epsilon_0. \tag{9.2}$$

As analytical solution we immediately get

$$u(t) = g(t) + \epsilon_0 \exp \lambda t. \tag{9.3}$$

Obviously, the function $g(t)$ is the *equilibrium solution*, which we assume to be comparatively smooth. In Figure 9.1 the two cases $\lambda \ll 0$ and $\lambda \gg 0$ are depicted for some $g$. If $\Re(\lambda) \ll 0$, a small deviation $\epsilon_0$ will rapidly lead back to the equilibrium solution; this is exactly the case of "stiff" initial value problems, often just called stiff ODEs. One should keep in mind from the beginning that in the *transient phase*, i.e., the regime

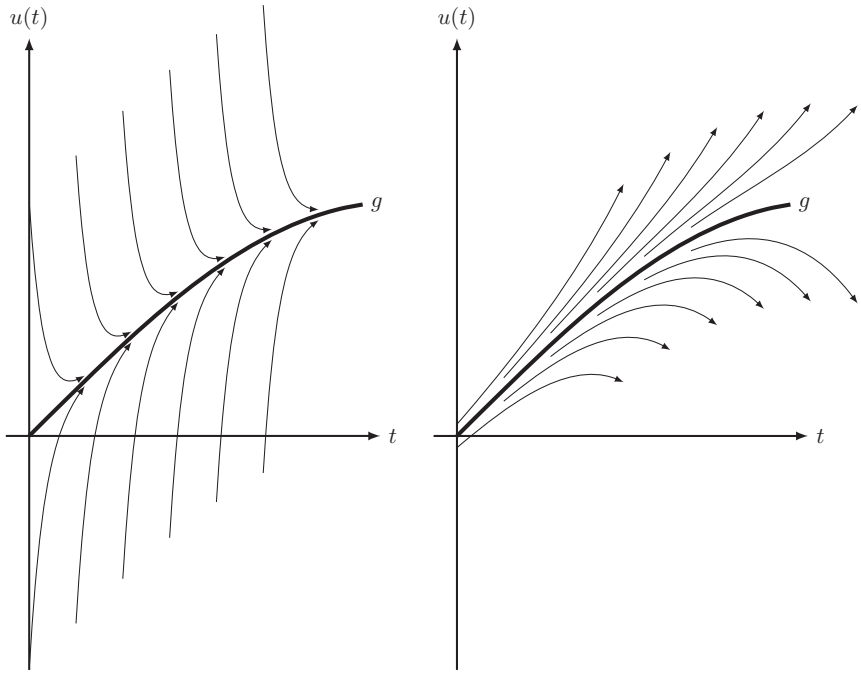


**Figure 9.1.** Example (9.3) for equilibrium solution $g = \sin(t), t \in [0, 1.5]$. *Left*: "stiff" problem: $\lambda = -16,\ \epsilon_0 = 1$. *Right*: inherently unstable problem: $\lambda = 3,\ \epsilon_0 = 0.05$.

before the equilibrium solution is reached, the problem is not "stiff". For $\Re(\lambda) \gg 0$ deviations are rapidly blown up, one speaks of *inherently unstable* initial value problems, often just from unstable ODEs – to be clearly distinguished from stiff ODEs (although the classical paper by C. F. Curtiss and J. O. Hirschfelder [58] from 1952, in which the term "stiff ODE" was coined, is inexact in this very point).

### 9.1.1 Linear Stability Theory

After the simple example we now investigate the general initial value problem for $N$ ordinary differential equations

$$u' = F(u), \quad u(0) = u_0, \tag{9.4}$$

where we, as a preparation for later sections, already envision the nonlinear case. Now let $g$ denote a selected nominal solution of (9.4), assumed to be sufficiently smooth, and $u$ a different solution. Then, for the deviation

$$\delta u(t) = u(t) - g(t)$$

we have – in linearized form – the *variational equation*

$$\delta u' = F_u\big(g(t)\big)\,\delta u, \quad \delta u(0) = \delta u_0\ ,$$

to prescribed "initial deviation" $\delta u_0$. In general this is a nonautonomous (i.e., the time explicitly containing) linear differential equation with time-dependent *Jacobian matrix* $J(t) = F_u(g(t))$ of dimension $N$. However, the analysis of nonautonomous linear systems is nearly as complex as the one of general nonlinear systems.

In order to simplify our investigation, we therefore replace the variational equation by the autonomous differential equation

$$\delta u' = J\ \delta u, \quad \delta u(0) = \delta u_0 \tag{9.5}$$

with *constant* matrix $J$. Here the equilibrium solution is just $\delta u(t) \equiv 0$. In order to gain some insight into the structure of the problem class, we transform (similar to Section 8.1) the variable

$$\delta u = C\ \overline{\delta u}$$

by an *arbitrary* nonsingular matrix $C$ . Thus we obtain the transformed differential equation

$$\overline{\delta u}' = \overline{J}\ \overline{\delta u}, \quad \overline{J} = C^{-1}\ J\ C.$$

Invariants of this similarity transformation are the eigenvalues $\lambda(J)$, representative for all of the thus defined matrices is the *Jordan canonical form*. In a further step of simplification we now assume that $J$ is diagonalizable.[1] Then the linear system (9.5) splits into $N$ decoupled scalar differential equations of the form

$$y' = \lambda y, \quad y(0) = y_0, \quad \lambda \in \mathbb{C}. \tag{9.6}$$

**Dahlquist Test Equation.** If we choose $y_0 = 1$ in (9.6), we arrive at a scalar equation suggested by G. Dahlquist in 1956 as a test equation useful to distinguish different discretization methods [59]. Let $\tau > 0$ denote a prescribed timestep. Then (9.6) can be written as

$$y(\tau) = \mathrm{e}^z \quad \text{with } z = \lambda\tau \in \mathbb{C}.$$

To begin with, we collect several properties of the complex exponential function. The imaginary axis is a separating line in the following sense:

$$\begin{aligned} |\mathrm{e}^z| \geq 1 \quad &\Leftrightarrow \quad \Re(z) \geq 0, \\ |\mathrm{e}^z| \leq 1 \quad &\Leftrightarrow \quad \Re(z) \leq 0, \\ |\mathrm{e}^z| = 1 \quad &\Leftrightarrow \quad \Re(z) = 0. \end{aligned} \tag{9.7}$$

For our stability analysis this property is important with respect to the question of whether a deviation from the equilibrium solution would asymptotically lead back to it. Moreover, the function possesses an *essential singularity* in the point $z = \infty$,

[1] The subsequent theory also holds for nondiagonalizable matrices (see Exercise 9.1).

i.e., when approaching this point, its value depends on the path of approach. In particular, we have

$$z \to \infty : \quad |\mathrm{e}^z| \to \begin{cases} \infty & \text{for } \Re(z) > 0, \\ 0 & \text{for } \Re(z) < 0, \\ 1 & \text{for } \Re(z) = 0. \end{cases} \tag{9.8}$$

This property is predominantly important for large stepsizes $\tau \gg |\lambda|^{-1}$.

**One-step Methods.** We now want to investigate what happens with these two properties by time discretization of the test equation (9.6). Among the discretizations we restrict our attention to one-step methods, since they have certain advantages with respect to time dependent PDEs (see, e.g., the more extensive exhibition in Volume 2). In one-step methods we simply have to consider one step with timestep $\tau$; we may therefore drop the running index of the timestep and always start at $u_0$. As examples we select four elementary discretizations, first applied to the general differential equation $y' = f(y)$, using the short notation $y_1$ for the result after one step as well as $z = \lambda\tau \in \mathbb{C}$:

- explicit euler method (EE):

$$y_1 = y_0 + \tau f(y_0) \quad \Rightarrow \quad y_1 = 1 + z;$$

- implicit Euler method (IE):

$$y_1 = y_0 + \tau f(y_1) \quad \Rightarrow \quad y_1 = \frac{1}{1-z};$$

- implicit trapezoidal rule (ITR):

$$y_1 = y_0 + \tau \frac{f(y_1) + f(y_0)}{2} \quad \Rightarrow \quad y_1 = \frac{1 + z/2}{1 - z/2};$$

- implicit midpoint rule (IMP):

$$y_1 = y_0 + \tau f\left(\frac{y_1 + y_0}{2}\right) \quad \Rightarrow \quad y_1 = \frac{1 + z/2}{1 - z/2}.$$

For linear problems the methods ITR and IMP are identical. In the nonlinear case, however, the IMP preserves quadratic functionals such as the discrete *energy* (see Exercise 9.2), which is why in concrete problems it is usually preferable. From our elementary examples it becomes clear that all one-step methods can be subsumed under the general scheme

$$y_1 = R(z) \ ,$$

where $R$ is a complex *rational* function that might have poles in $\mathbb{C}$, but no essential singularities.

**Order.** For "small' $z \to 0$ the concept of order $p$ according to

$$R(z) = \mathrm{e}^z + \mathcal{O}(z^{p+1})$$

is appropriate. It easy to verify that $p = 1$ holds for EE and IE, $p = 2$ for ITR and IMP.

**A-stability.** If we want to inherit the separation property (9.7) exactly into the discrete setting, from our four sample discretizations merely ITR and IMP are left. This is why G. Dahlquist [59] in 1956 has introduced the concept of A-stability, to be able to save at least part of the important property (9.7) also for other discretizations. A one-step method is called A-stable if and only if

$$|R(z)| \le 1 \quad \text{for} \quad \Re(z) \le 0.$$

However, this concept captures the behavior for $z \to \infty$ only insufficiently. Upon taking the fact into account that complex rational functions in $\mathbb{C}$ are unique, the comparison with (9.8) will direct us to the following discrimination:

$$z \to \infty : \quad |R(z)| \to \begin{cases} \infty & \text{for EE,} \\ 0 & \text{for IE,} \\ 1 & \text{for ITR and IMP.} \end{cases}$$

As expected, it is not possible to mimic the essential singularity $z = \infty$ of the complex exp-function by a single complex rational function. We thus have to decide which limit we prefer for a special problem. Obviously, for problems like the Schrödinger equation, with eigenvalues on the imaginary axis, we will select discretizations of the type IMP. For inherently unstable problems or also for stiff problems in the transient phase an explicit discretization like EE will be considered. For stiff problems in a neighborhood of the equilibrium solution methods like IE will be chosen.

**L-stability.** This concept was introduced by B. L. Ehle [83] in 1969 to select the appropriate limit especially for stiff ODEs. A one-step method is called L-stable if it is A-stable and satisfies the additional property

$$R(\infty) = 0 \, .$$

From our four elementary candidates this additional condition selects only the implicit Euler discretization (IE). This, however, has the disadvantage that it damps too much along the imaginary axis and even in the positive complex half-plane, which is why one also speaks of undesirable *"superstability"*.

**Stability Regions.** The presentation so far has made clear that for an assessment of discretization methods compromises are necessary. As a useful tool one defines the stability region $\mathcal{S}$ of a one-step method by

$$\mathcal{S} = \big\{ z \in \mathbb{C} : |R(z)| \le 1 \big\}.$$

Obviously, A-stability is characterized by the condition

$$\mathbb{C}^- \subset \mathcal{S}.$$

In Figure 9.2 we show the stability regions for the four elementary one-step methods.

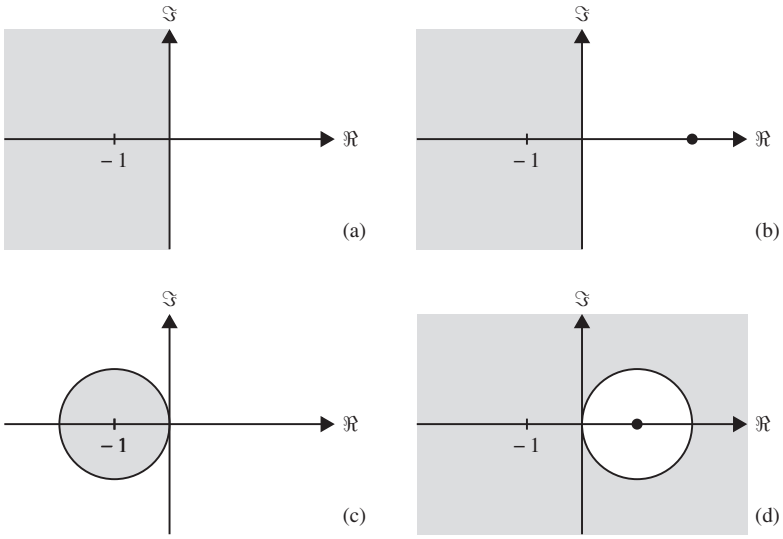


**Figure 9.2.** Stability regions in $\mathbb{C}$ (with poles •) for (a) solution $e^z$, (b) implicit trapezoidal rule (ITR) equivalent to implicit midpoint rule (IMP), (c) explicit Euler discretization (EE), (d) implicit Euler discretization (IE).

For a better understanding of the concept we give, in Figure 9.3, a few examples of discrete solutions $y_k = (R(z))^k$ for selected points $z$ in comparison with the continuous solution $y_k = (e^z)^k$.

**A($\alpha$)- and L($\alpha$)-stability.** In the construction of methods of order $p > 1$ one strives to achieve an approximation as good as possible of the negative half-plane $\mathbb{C}^-$ by the stability region $\mathcal{S}$. In order to be able to measure the quality of the approximation in a single number, one introduces, for $\alpha \in [0, \pi/2]$, the notation

$$\overline{\mathcal{S}}(\alpha) = \{z \in \mathbb{C} : |\arg(-z)| \leq \alpha\}$$

for the angular region around the negative real half-axis and extends the two concepts of A-stability and L-stability: a one-step method is called A($\alpha$)-*stable* if

$$\overline{\mathcal{S}}(\alpha) \subset \mathcal{S}.$$

Obviously, A-stability is equivalent to A($\pi/2$)-stability. In a similar way a one-step method is called L($\alpha$)-*stable* if

$$\overline{\mathcal{S}}(\alpha) \subset \mathcal{S} \quad \text{and} \quad R(\infty) = 0.$$

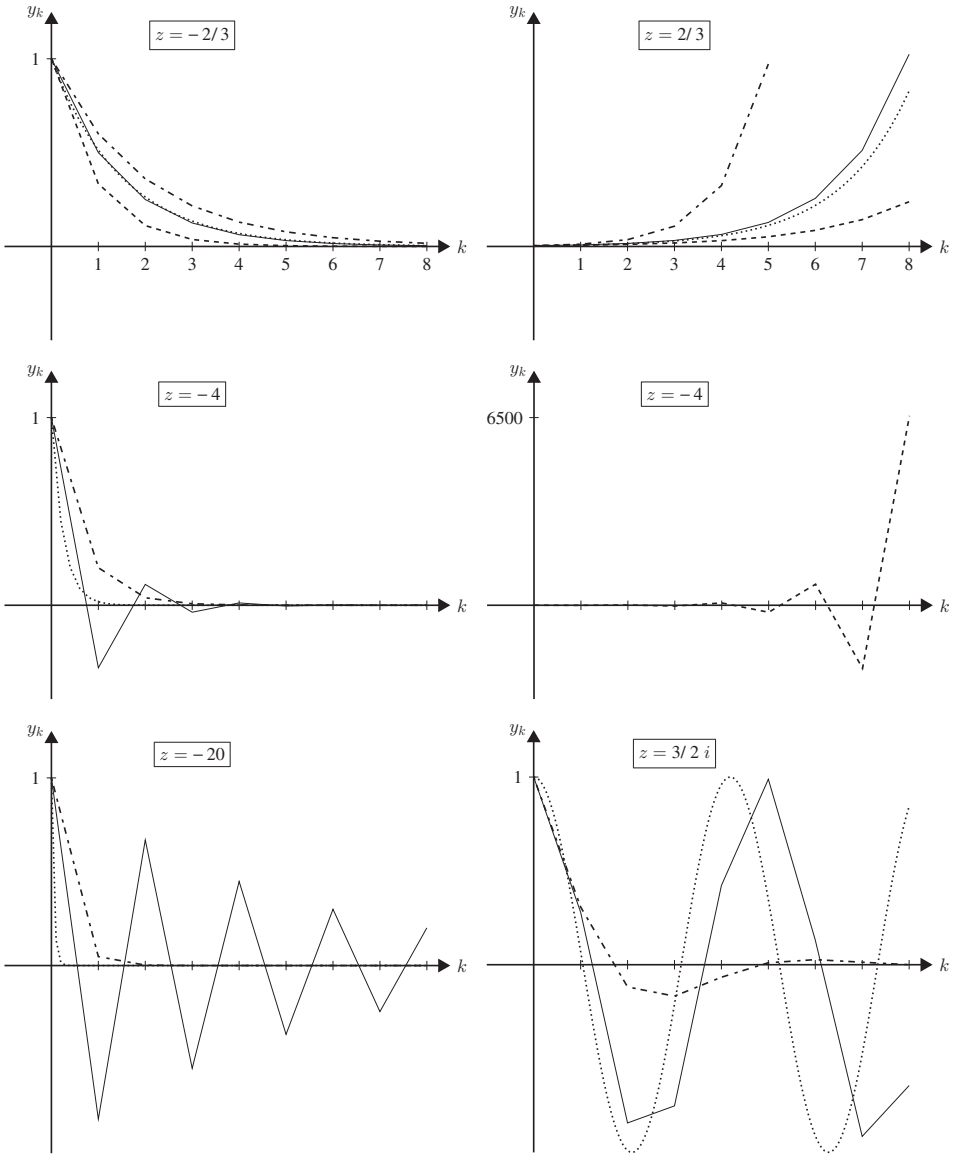


**Figure 9.3.** Solution $y_k = (\mathrm{e}^z)^k$ compared with three discrete solutions $y_k = (R(z))^k$. Notations: ····· solution, - - - - - EE, - - - - - IE, ——— IMP. *Top*: For small $z \in \mathbb{C}$ the higher order of IMP versus EE and IE pays off. *Center and bottom left*: in the "stiff" model problem only IE supplies a qualitatively correct discrete solution, IMP supplies weak, EE strong oscillations (which for $z = -20$ would leave the scale of the figure). *Bottom right*: along the imaginary axis the continuous solution exhibits undamped oscillations, which by IE are incorrectly damped ("superstability"), while IMP captures them qualitatively correctly, though with a phase shift (which, e.g., is undesirable for the Schrödinger equation; see Section 1.4).

Clearly, L-stability is equivalent to L($\pi/2$)-stability. In both concepts one has, due to $\overline{\mathcal{S}}(\alpha_1) \subset \overline{\mathcal{S}}(\alpha_2)$ for $\alpha_1 \leq \alpha_2$: the larger the angle, the more "stable" the discretization. For a more detailed presentation, we refer to Volume 2 or the textbook of E. Hairer and G. Wanner [117].

**Nonlinear Stability Concepts.** There are plenty of extensions of linear stability concepts towards nonlinear stability concepts, such as B-stability, B-convergence, algebraic stability etc. (for a comprehensive discussion of these concepts see again [117]). A different concept based on a simplified Newton method for the nonlinear initial value problem was worked out in [65] as well as in [68], Section 6.2.2. The theory for this was presented in $\mathbb{R}^N$, but applies in function space as well.

### 9.1.2 Linearly Implicit One-step Methods

In this section we give a brief survey on a class of adaptive stiff integrators. For a general theoretical background of this class we refer to Volume 2, Section 6.4; a more detailed treatment of this class can be found in the monograph [195] of K. Strehmel and R. Weiner.

Let us return to the initial value problem (9.4) for $N$ ODEs, here of the form

$$M u' = F(u), \quad u(0) = u_0, \tag{9.9}$$

where $M$ is an invertible $(N, N)$-matrix. If (9.9) originated from a spatial discretization of a parabolic PDE, then, in general, $N$ is large and the system is "stiff". For finite difference methods as spatial discretization, we usually have $M = I$, for Galerkin methods $M$ is the symmetric positive definite mass matrix. For the Jacobian matrix $J(u) = F'(u)$ we assume a spectrum bounded from above, i.e., $v^T J v \leq \lambda_{\max} |v|$. For finite element discretizations the matrices $J$ and $M$ are sparse. In each step of such a discretization large linear equation systems of dimension $N$ arise in the form

$$(M - \gamma \tau J)\, v = w, \quad \gamma > 0. \tag{9.10}$$

The question of how to solve them numerically is postponed, for the time being.

**Implicit Integrators.** Before we deal with the actual topic of this section, the linearly-implicit one-step methods, we want to quickly screen the class of purely implicit one-step and multi-step methods.

- *Implicit one-step methods*. This class contains *Runge–Kutta methods* (see, e.g., Volume 2, Sections 6.2 and 6.3), whose coefficients are determined by the solution of a set of algebraic equations. In each timestep, implicit Runge–Kutta methods require the solution of *several nonlinear* equation systems, whose number depends on the stage number $s$. Newton-like iteration for their solution leads to linear equation systems of the type (9.10). Particularly efficient are *Radau* methods (see Volume 2, Section 6.3), special L-stable collocation methods, realized,

e.g., in the adaptive program Radau53 by E. Hairer and G. Wanner [117]. However, they have the reputation of being too costly for parabolic PDEs.
Formally speaking, *extrapolation methods* are a special subclass of Runge–Kutta methods. However, they are not generated via the solution of algebraic equation systems for the coefficients, but from a simple basis discretization by extrapolation with respect to the timestep $\tau$. Purely implicit extrapolation methods, based on the implicit trapezoidal rule or the implicit midpoint rule, play only a minor role for parabolic PDEs, but are well used in connection with the Schrödinger equation.

- *Implicit multi-step methods*. Among them the *BDF-method* (see Volume 2, Sections 7.3.2 and 7.4.2) certainly is, for both theoretical and practical reasons, the most popular one. An adaptive implementation is realized, e.g., in the program DASSL by L. Petzold [170]. In each timestep this method requires the numerical solution of *one nonlinear* equation system, which is performed by some Newton-like iteration thus leading to linear equations of the type (9.10). Multi-step methods gain their efficiency by exploiting the "history" of the trajectory, which requires a certain smoothness over several timesteps; in our context, this is also a certain weakness, since time dependent adaptation of spatial meshes (see the subsequent Section 9.2) will perturb such a smoothness.

**Structure of Linearly Implicit One-step Methods.** This type of method is based on the simple idea to subtract on both sides of (9.9) a linear autonomous term $Ju$, i.e.,

$$Mu' - Ju = F(u) - Ju, \tag{9.11}$$

and discretize only the left-hand part implicitly, but the right-hand part explicitly. This means, of course, a structural simplification compared to implicit methods. For this type of methods two variants exist:

- *Methods with exact Jacobian matrix*, i.e., with $J = F'(u_0)$. This variant is realized in *Rosenbrock-Wanner methods* (in short: *ROW-methods*). The information of the exact Jacobian matrix enters into the algebraic equations for the method.
- *Methods with inexact Jacobian matrix*, i.e., with $J \approx F'(u_0)$. This variant is realized in so-called *W-methods*. Formally speaking, *linearly implicit extrapolation methods* are a subclass of such methods, even though historically they have originated independently. As the term $Ju$ occurs on both sides of (9.11) in the same way, the choice of $J$ is relatively free. If some detailed insight into the coupling of the various components of the ODE is available, then this class of methods permits to drop "weak" couplings in the ODE system, which arise as "small" entries in the Jacobian matrix, by setting them deliberately to zero (*deliberate sparsing*); of course, scaling needs to be carefully done, when deciding about which elements are to be regarded as "small".

If the linear equation systems (9.10) are so large that they are only accessible to an iterative numerical solution, then linearly-implicit one-step methods require only *two* nested loops (*outer loop*: time discretization, *inner loop*: iteration for linear equation system) versus *three* loops in implicit one-step of multi-step methods (*outer loop*: time discretization, *medium loop*: Newton-like iteration, *inner loop*: iteration for linear equation system). In very large systems this is an important advantage.

**Linearly Implicit Runge–Kutta Methods.** This class contains both ROW- and W-methods. They are constructed by establishing $N_p$ algebraic conditions for the coefficients of the RK-discretization scheme, depending on the order $p$; a more detailed examination reveals that for *exact* Jacobian matrix the number of conditions remains the same as for Runge–Kutta methods, while their structure changes considerably: The information about the Jacobian matrix directly enters into the condition equations. The number of coefficients depends on the stage number $s$, which is usually larger than the number of coefficients. Hence, one arrives at underdetermined systems of algebraic equations, which means that additional wishes can be fulfilled, such as: the stage number $s$ should be as small as possible, the constructed method should be robust in the presence of inconsistent initial data, it should also work for singularly perturbed differential equations (see, e.g., Volume 2, Section 2.5) or for differential-algebraic equations up to index 1 (see again Volume 2, Section 6). Further additional equations aim at avoiding order reduction for nonlinear parabolic differential equations; see Section 9.1.3 below).

**ROW-methods.** In Volume 2, Section 6.4.1 we discussed this class of linearly implicit Runge–Kutta methods. With the here introduced notation one step of this type of method can be written in the form:

$$\begin{aligned}
&\text{(a)}\quad J = F'(u_0),\\
&\text{(b)}\quad i = 1,\dots,s:\\
&\qquad (M - \gamma_{ii}\tau J)k_i = \tau \sum_{j=1}^{i-1} (\gamma_{ij} - \alpha_{ij}) J k_j + F\Big(u_0 + \tau \sum_{j=1}^{i-1} \alpha_{ij} k_j\Big), \qquad (9.12)\\
&\text{(c)}\quad u_1 = u_0 + \tau \sum_{i=1}^{s} b_i k_i .
\end{aligned}$$

Note that the unknowns $k_1, \dots, k_s$ can be worked down recursively. If one sets, as is usual,

$$\gamma_{ii} = \gamma > 0, \quad i = 1,\dots,s,$$

then all of the above matrices are identical so that a single LU-decomposition is sufficient. (Of course, in an iterative solution mode this advantage is of minor importance.)

For the determination of the coefficients the following successive order conditions occur (exemplified for $p \le 4$)

$$
\begin{aligned}
p=1: &\quad \sum_{i=1}^{s} b_i = 1, && p=4: &\quad \sum_{i=1}^{s} b_i \alpha_i^3 = \frac{1}{4},\\
p=2: &\quad \sum_{i,k=1}^{s} b_i a_{ik} = \frac{1}{2}, && &\quad \sum_{i,k,l=1}^{s} b_i \alpha_{ik} a_{kl} \alpha_i^2 = \frac{1}{8},\\
p=3: &\quad \sum_{i=1}^{s} b_i \alpha_i^2 = \frac{1}{3}, && &\quad \sum_{i,k,l,m=1}^{s} b_i a_{ik} \alpha_{kl} \alpha_{km} = \frac{1}{12},\\
&\quad \sum_{i,k,l=1}^{s} b_i a_{ik} a_{kl} = \frac{1}{6}, && &\quad \sum_{i,k,l,m=1}^{s} b_i a_{ik} a_{kl} a_{lm} = \frac{1}{24}.
\end{aligned}
$$

Here we have simplified the notation by introducing

$$\alpha_{ij} = 0,\ j \ge i, \quad \gamma_{ij} = 0,\ j > i$$

as well as the following abbreviations:

$$
\begin{aligned}
&a_{ij} = \alpha_{ij} + \gamma_{ij}, && \mathcal{A} = (a_{ij})_{i,j=1}^{s},\\
&b = (b_1, \ldots, b_s)^T, && \mathbf{e} = (1, \ldots, 1)^T \in \mathbb{R}^s,\\
&\alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}, && \alpha^k = (\alpha_1^k, \ldots, \alpha_s^k), \qquad \gamma_i = \sum_{j=1}^{i} \gamma_{ij}.
\end{aligned}
\tag{9.13}
$$

For higher order the number of conditions increases rapidly, as shown below in the second row of Table 9.1.

In recent years, by subtle investigation of the solution structure of the order conditions, the following adaptive ROW-integrators have emerged:

- ROS3P by J. Lang and J. Verwer [141]: this method, published in 2001, has order $p = 3$ and stage number $s = 3$, is A-stable, suitable up to index 1, and does not suffer from order reduction for nonlinear parabolic differential equations (a topic that we will treat below in Section 9.1.3); in 2008 it was improved by J. Lang and D. Teleaga [140] to ROS3PL with $p = 3$ and $s = 4$, which is L-stable and robust against approximation errors of the Jacobian matrix;
- RODAS, one of the classical integrators by E. Hairer and G. Wanner [117], possesses order $p = 4$ and stage number $s = 6$, but suffers from order reduction; the method was improved in 2001 by G. Steinebach and P. Rentrop [192] to RODASP, it does not suffer from order reduction, but for linear parabolic equations only.

**W-methods.** None of the above mentioned linearly implicit RK-methods is a W-method, i.e., independent of the choice of the matrix $J$. In the context of parabolic

**Table 9.1.** Linearly-implicit RK-methods: number of algebraic conditions for order $p$.

| $p$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $N_p$ (ROW-methods) | 1 | 2 | 4 | 8 | 17 | 37 | 85 | 200 |
| $N_p$ (W-methods) | 1 | 3 | 8 | 21 | 58 | 166 | 498 | 1 540 |

PDEs, however, the "Jacobian matrix" as spatial discretization of the Fréchet-derivative will inevitably contain discretization errors, which is why W-methods seem to have a structural advantage for this problem class. Compared to ROW-methods, however, the number of algebraic conditions for the coefficients grows significantly (see Table 9.1). The theoretical reason for this stronger increase is that now terms of the kind $F'F$ and $JF$ need to be distinguished, which leads to more "elementary differentials" and thus to more condition equations. Correspondingly, the required stage number gets higher. This background explains why this strand of methods has still been neglected in research. An alternative are the linearly implicit extrapolation methods to be presented directly after, which, by construction, can be conveniently extended to higher orders.

**Timestep Control.** As in the spatial discretization in Section 6.2.1 an optimal relation of computational amount and discretization error is achieved by timesteps for which the local error contributions are *equilibrated*. For the construction of such adaptive time grids we thus need a localized error estimator and a refinement mechanism.

*Error estimator.* For local error estimation two discrete solutions $u_\tau, \hat{u}_\tau$ of order $p+1$ and $p$ are computed in parallel such that

$$u_\tau(\tau) = u(\tau) + \mathcal{O}(\tau^{p+2}), \quad \hat{u}_\tau(\tau) = u(\tau) + \mathcal{O}(\tau^{p+1}).$$

Then

$$[\hat{\epsilon}_\tau] := \|u_\tau(\tau) - \hat{u}_\tau(\tau)\| \doteq \hat{\epsilon}_\tau = \|u(\tau) - \hat{u}_\tau(\tau)\| \doteq C\ \tau^{p+1} \tag{9.14}$$

is an estimator of the actual $\hat{\epsilon}_\tau$ of $\hat{u}_\tau(\tau)$ measured in a suitable norm $\|\cdot\|$

*Refinement.* Unlike the case of spatial meshes, the timesteps are not determined by subdivision. It is more efficient to suggest a nearly optimal timestep for the next step in advance. Thereby one exploits the fact that the optimal timestep will change rarely.

Now let $\tau$ denote the presently selected timestep. We search for an "optimal" stepsize $\tau^*$, for which

$$\hat{\epsilon}_{\tau^*} \leq \mathrm{TOL}_\tau.$$

By insertion of both $\tau$ and $\tau^*$ into (9.14), we arrive at the estimation formula

$$\tau^* = \sqrt[p+1]{\frac{\rho\ \mathrm{TOL}_\tau}{[\hat{\epsilon}_\tau]}}\ \tau \tag{9.15}$$

with a safety factor $\rho \approx 0.9$. If $[\hat{\epsilon}_\tau] \leq \mathrm{TOL}_\tau$, then the present step is accepted and $\tau^*$ is used in the next step; otherwise the present step is rejected and repeated with timestep $\tau^*$. In the successful case the more accurate value $u_\tau(\tau)$ will be used to start the next step.

In the context of parabolic differential equations usually *embedded* methods are realized. Therein one computes, from common intermediate values $k_1, \ldots, k_s$ due to (9.12), the two different solutions

$$\begin{aligned} u_\tau(\tau) &= u_0 + \tau \sum_{i=1}^{s} b_i k_i, \\ \hat{u}_\tau(\tau) &= u_0 + \tau \sum_{i=1}^{s} \hat{b}_i k_i. \end{aligned}$$

Then, obviously, we get

$$[\hat{\epsilon}_\tau] = \|u_\tau(\tau) - \hat{u}_\tau(\tau)\| = \tau \Big\| \sum_{i=1}^{s} (b_i - \hat{b}_i) k_i \Big\|.$$

**Linearly Implicit Extrapolation Methods.** The first such method was the *semiimplicit midpoint rule* (today generally called *linearly implicit midpoint rule*) suggested by G. Bader and P. Deuflhard [14] as an extension of the explicit midpoint rule. This method permits $\tau^2$-extrapolation, but includes intermediate steps with $|R(\infty)| = 1$, which is why it is not suitable for index 1 and thus does not seem to be recommendable for parabolic PDEs. Instead the $\tau$-extrapolation method based on the linearly implicit Euler discretization

$$(M - \tau J)\,(u_{k+1} - u_k) = \tau F(u_k), \quad k = 0, 1, \ldots \tag{9.16}$$

has prevailed, which realizes $R(\infty) = 0$ throughout all intermediate steps. It is implemented in the code LIMEX due to [71, 74]. If $J$ is the exact Jacobian matrix $F'(u_k)$, then this method can be subsumed under (9.12) with

$$\gamma_{ii} = \frac{1}{n_i},$$

where $\mathcal{F} = \{n_1, n_2, \ldots\}$ is the subdivision sequence chosen for this extrapolation method. In practice, the simple harmonic sequence

$$\mathcal{F}_H = \{1, 2, 3, \ldots\}$$

has turned out to be efficient. By construction, this type of method is embedded so that a timestep control according to (9.15) can be realized in an elementary way; it can be complemented by an adaptive order control (see Volume 2, Section 6.4.2). In the integrator LIMEX a maximal order $p_{\max} = 5$ is mostly imposed.

In passing we note that in this framework W-methods of higher order $p$ can be conveniently constructed without having to solve the highly complex (and many!) algebraic equations for the coefficients. Such methods, however, are not optimized with respect to the number of stages. But they are $L(\alpha)$-stable with $\alpha \approx \pi/2$ also for higher order and can also be applied to differential-algebraic equations, for details see [71, 74].

**Exponential Integrators.** This special class of numerical integrators was suggested by M. Hochbruck and C. Lubich [121, 122] in 1997. Like for linearly implicit integrators, here one also starts from the form (9.11) of the ODE, i.e.,

$$u' - Ju = F(u) - Ju, \quad u(0) = u_0,$$

where we have set $M = I$ for simplicity. The basic idea is to integrate the left-hand side *exactly*, which for *linear* $F$ and $J = F'$ would lead to

$$u(t) = \exp(Jt)u_0.$$

For *nonlinear* $F$ one will make a "variation of constants" ansatz,

$$u(t) = \exp(Jt)v(t), \quad u(0) = u_0.$$

After some short calculation one then obtains the ODE

$$\exp(Jt)v' = F(\exp(Jt)v) - J\exp(Jt)v.$$

Starting from a discretization of the ODE for $v$ and a backtransformation to $u$ the authors construct discretization methods with respect to $u$. The simplest candidate is the *exponential Euler method*

$$u_1 = u_0 + \tau\varphi(J\tau)F(u_0), \tag{9.17}$$

where

$$\varphi(z) = \frac{\mathrm{e}^z - 1}{z}$$

and $\varphi(0) = 1$. The method is of order $p = 2$. For higher order the authors worked out ROW-methods for *exact* Jacobian matrix $J = F'(u_0)$ as well as W-methods for *inexact* Jacobian matrix $J \approx F'(u_0)$. In both cases the matrix exponential arises via the form $\varphi(\gamma\tau J)$, where the parameter $\gamma$ must be suitably chosen. For (9.17) then $\gamma = 1$ will hold. In [122] the program exp4 to order $p = 4$ with stage number $s = 7$ and $\gamma = 1/3$ was suggested, into which a method of order $p = 3$ is embedded so that an adaptive timestep control is possible (see Volume 2, Section 5.4). For differential-algebraic problems (with index 1) the order reduces to $p = 3$, for inexact Jacobian with

$$J = F'(u_0) + \mathcal{O}(\tau)$$

to order $p = 2$. In total, the method is not only adaptive, but rather robust.

The knack of this approach, however, comes to bear with high dimension $N$, where the matrix-exponential is approximated by Krylov space methods (cf., for example, Volume 1, Section 8.3 and 8.5). It is shown in [121] that the Krylov approximation of $\varphi(\gamma J\tau)w$ (applied to some vector $w$) in general converges even faster than the one for $(I - \gamma\tau J)^{-1}w$, which would be required for (9.10). These methods are well-suited for "mildly stiff" problems and problems with eigenvalues along the imaginary axis. In the context of PDEs, however, one must take into account that Krylov space methods belong to *fixed* dimension $N$, i.e., they are structurally inconsistent with adaptive multigrid methods: By extension of the discretization spaces from some coarse to a finer mesh the information of the Krylov basis cannot be transferred – which definitely is a structural disadvantage of this type of method.

**Explicit Integrators in Transient Phase.** As already exemplified in the simple test problem (9.2), the problem is not "stiff" in the transient phase, since one is not yet close to the smooth solution $g$; as a consequence, here even an explicit discretization is efficient, which formally may be expressed as W-method by $J = 0$. To make this a robust numerical technique, however, one would need a reliable criterion for switching between "stiff" and "nonstiff". Switching from "stiff" to "nonstiff" uses information of the Jacobian matrix, which is thus needed before; therefore this variant is not able to save very much in terms of computing time and storage, at best in the direct solution of the equation systems (9.10). Switching from "nonstiff" to "stiff" can only use information from the right-hand sides of the ODE to recognize whether the adaptively suggested timesteps are restricted by stability conditions; such strategies have been suggested several times, but do not seem to have been sufficiently robust in practical tests. Therefore, quite often the decision "stiff" or "nonstiff" is made on the basis of a priori insight into the problem to be solved. A "stiff" problem in the transient phase may then be discretized by some explicit method with small timesteps. A comparison with a stiff integrator will then have to weight the larger number of timesteps versus the reduced computational amount per timestep.

### 9.1.3 Order Reduction

When moving from stiff ODEs to parabolic PDEs an important restriction for the time discretization is encountered: instead of the order $p$ of a method, one obtains an *effective* order $p^* \leq p$. The reason for this is that the spectrum of the spatial differential operators is unbounded, whereas for ODEs the corresponding spectrum is bounded. For illustration, we begin with a simple test problem.

**Example 9.1.** Consider the scalar linear parabolic PDE

$$u_t = u_{xx} \quad \text{for } x \in \Omega = ]0, \pi[ \tag{9.18}$$

to homogeneous Dirichlet boundary conditions $u(0,t) = u(\pi,t) = 0$ and starting values $u(x,0) = u_0(x)$. As elaborated in Section 1.2, the problem (9.18) may be split into independent Fourier modes

$$u(x,t) = \sum_{k=1}^{\infty} a_k \mathrm{e}^{\lambda_k t} \sin(kx) \quad \text{to eigenvalues } \lambda_k = -k^2. \tag{9.19}$$

If we consider the problem in function space, we have the "spatial modes" (eigenfunctions) $\{\sin(kx)\} \in L^2(\Omega)$ with time-dependent coefficients $a_k \mathrm{e}^{\lambda_k t}$. These coefficients each satisfy the Dahlquist test equation. If we discretize the problem first with respect to time by means of a (linearly) implicit Runge–Kutta method, then, due to the linearity, we get a decomposition similar to (9.19) of the form

$$u_1 = \sum_{k=1}^{\infty} a_k R(\lambda_k \tau) \sin(kx),$$

where $R$ denotes the rational stability function of the selected one-step method (cf. Section 9.1.1). For the consistency error we obtain pointwise

$$\epsilon(\tau) = u(x,\tau) - u_1 = \sum_{k=1}^{\infty} a_k (\mathrm{e}^{\lambda_k \tau} - R(\lambda_k \tau)) \sin(kx).$$

Now, let the one-step method be L-stable ($R(\infty) = 0$). Then the continuous function $|\mathrm{e}^z - R(z)|$ has a maximum on the negative real axis $]-\infty, 0]$ at some point $\bar{z} < 0$, say. Thus, for stepsizes $\tau_k = \bar{z}/\lambda_k$, we have

$$\|\epsilon(\tau_k)\|^2_{L^2(\Omega)} = \frac{\pi}{2} \sum_{l=1}^{\infty} a_l^2 (\mathrm{e}^{\lambda_l \tau_k} - R(\lambda_l \tau_k))^2 \geq a_k^2 (\mathrm{e}^{\bar{z}} - R(\bar{z}))^2 = \mathcal{O}(a_k^2),$$

where we have used the $L^2$-orthogonality of the eigenfunctions and picked out one term of the sum, where the expression in brackets has its maximum. Note that this result is independent of the consistency order $p$ of the method.

As initial condition we now choose $u_0(x) = x(\pi - x)$. For this we obtain the coefficients

$$a_k = \frac{8}{\pi k^3} = \frac{8}{\pi |\bar{z}|^{3/2}} \tau_k^{3/2} \quad \text{for odd } k,$$

as well as $a_k = 0$ for even $k$. Insertion then yields

$$\|\epsilon(\tau_k)\|_{L^2(\Omega)} \geq \mathcal{O}(\tau_k^{3/2}). \tag{9.20}$$

For comparison: for the consistency order of Runge–Kutta methods one has $p \geq 1$, which means that in the case of ODEs we would obtain $\mathcal{O}(\tau^{p+1})$ as consistency error. The result (9.20) thus represents an effective order $p^* = 1/2$ for the whole class of Runge–Kutta methods. It seems worth noting that this restriction of the consistency

order also occurs for the very smooth initial values that we chose above. Obviously, the reason for the order reduction lies in the fact that the spectrum of the Laplace operator, in contrast to the case of ordinary differential equations, is unbounded and distributed along the whole negative real axis.

This effect is already echoed in differential-algebraic problems; but these have a large "gap" in the spectrum between the bounded differential part and the algebraic part $\lambda \to -\infty$, which is why the consistency order of L-stable methods with $e^z - R(z) \to 0$ is preserved for $z \to -\infty$ (see Volume 2, Section 6.4.2).

The phenomenon of order reduction was probably first detected by K. Strehmel and R. Weiner in the course of their $B$-convergence studies, and presented in 1992 in their monograph [195]; as it turs out, however, their order bounds were not sharp. This is why we give a short account of results here which were proved in 1993 by A. Ostermann and M. Roche [165] and in 1995 by C. Lubich and A. Ostermann [147, 148]. There adaptivity does not play a role, neither in space nor in time; the quoted theoretical results were derived in the framework of *uniform* spatial and temporal grids. Starting point of these analyses is the insight that in successively finer space discretization asymptotically the structure of the underlying Cauchy problem in function space will show up. As a consequence, in [147, 148, 165] only the condition equations for Runge–Kutta methods in Hilbert spaces have been investigated. At the same time it was assumed that one is already on the smooth equilibrium solution, i.e., after the transient phase (see above).

In our simple linear example (9.18) we had only considered the consistency error, i.e., the *local* error; the high frequency modes contained therein, however, are rapidly damped in the course of time and thus do not play a role in the *global* error. As a consequence, in this example the reduction of the consistency order will not end up in a reduction of the convergence order on fixed time intervals. This damping effect, however, will vanish for problems where the high frequency modes are repeatedly renewed. Such a coupling of modes occurs for linear nonautonomous as well as for nonlinear problems; the results of the corresponding theory are now briefly summarized, without going too much into technical detail.

**Linear Nonautonomous Parabolic PDEs.** In [165], ROW-methods (9.12) for abstract Cauchy problems of the type

$$u' = Au + f(t) \tag{9.21}$$

were considered, where the operator $-A$, which includes the boundary conditions, was assumed to be elliptic. As a prerequisite for the theory, the initial value $u_0$ should already be on the stationary solution, which in the reality of scientific computing need not be satisfied – cf., for example, the electrocardiological example worked out in Section 9.3. Under this assumption one obtains the effective order

$$p^* = \min\{p, s + 2 + \beta\}, \tag{9.22}$$

where $s$ is the stage number and the quantity $\beta > 0$ can be determined from the underlying spatial elliptic problem. Due to [148] one has (with $\varepsilon > 0$ arbitrarily small)

$$\beta = \begin{cases} 3/4 - \varepsilon & \text{for homogeneous Dirichlet boundary conditions,} \\ 5/4 - \varepsilon & \text{for natural boundary conditions} \quad (d = 1), \\ 1/4 - \varepsilon & \text{for natural boundary conditions} \quad (d > 1). \end{cases} \tag{9.23}$$

It is interesting to note that for *periodic* boundary conditions no order reduction is to be expected. In the case of realistic application problems, however, there is no hope of getting hold of the value for $\beta$, so that in these cases the essential message is that $\beta > 0$.

In contrast to the adaptive control in one-step methods, where only *local* errors enter, the authors give a *global* convergence analysis for constant stepsizes $\tau$ over a fixed interval $T = n\tau$. Beyond that they work out further conditions that permit an extension of Rosenbrock methods to order $p \geq 3$. In order to break the barrier (9.22) by achieving, say, effective order $p^* = 3$ independent of the spatial regularity, the following conditions (in the notation (9.13))

$$b^T \mathcal{A}^j (2\mathcal{A}^2 \mathbf{e} - \alpha^2) = 0 \quad \text{for} \quad 1 \leq j \leq s - 1 \tag{9.24}$$

must be satisfied; note that they depend on the stage number $s$. These conditions are identical to the ones in [195].

**Example 9.2.** Consider the simple example

$$u_t = u_{xx} + x\, \mathrm{e}^{-t}, \quad x \in ]0, 1[, \quad t \in [0, 0.1] \tag{9.25}$$

with homogeneous Dirichlet-boundary conditions $u(0, t) = u(1, t) = 0$. In this example, Ostermann and Roche conducted numerical experiments. For that purpose they chose a spatial meshsize $h = 1/N$ for fixed $N = 1000$ and timesteps $\tau = 0.1/n$ to variable $n = 2, 4, \ldots, 128$. They generated stationary initial values by the condition $u'|_{t=0} = 0$, which leads to $u(x, 0) = \frac{1}{6}x(1 - x^2)$. For the popular Rosenbrock method GRK4T of P. Kaps and P. Rentrop [133] a reduction of the classical order $p = 4$ to $p^* = 3.25$ was achieved, whereas a method of S. Scholz [182][2], which takes extensions of the additional conditions (9.24) into account, would achieve order $p^* = p = 4$.

In Figure 9.4 we analyze the situation in more detail for differently fine spatial meshes. As in [165] we only consider the error at the final time point. Due to the damping of errors from intermediate points one expects a convergence error $\mathcal{O}(\tau^5)$ at the final point. This is actually achieved, but only for sufficiently small timesteps, corresponding to the fineness of the spatial discretization. In [165] a value of $N = 1000$ had been chosen, which would, for the selected timesteps $\tau$, be close to the effective order $p^* = 3.25$.

[2] For a correction of the printing errors with respect to the coefficients, however, see [165].

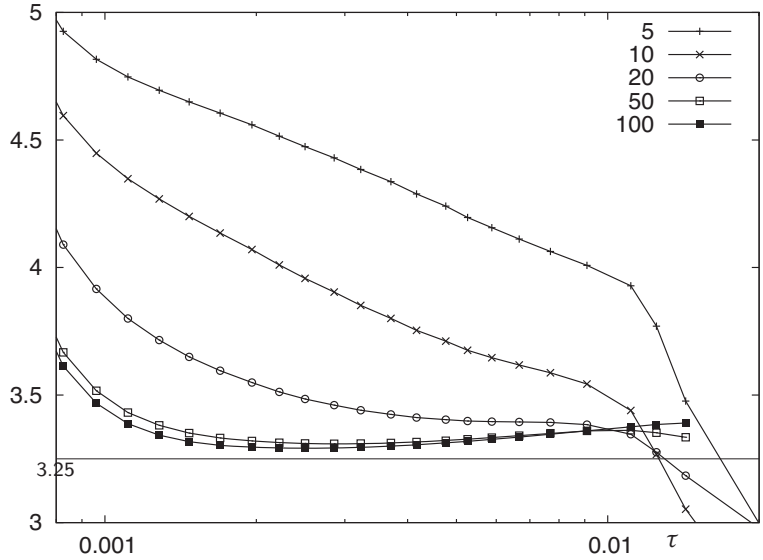


**Figure 9.4.** Test problem (9.25): numerically estimated convergence order of the ROW-integrator GRK4T. Spatial meshsizes $h = 1/N$ for $N = 5, \ldots, 100$, uniform timesteps $\tau = 0.1/n$ for $n = 2, 4, \ldots, 128$. Expected order $p = 5$, effective order $p^* = 3.25$.

**Quasilinear Parabolic PDEs.** This problem class may again be written as an abstract Cauchy problem

$$u' = A(u)u + f(x,t), \quad u(0) = u_0, \quad x \in \Omega \subset \mathbb{R}^d, d \geq 1.$$

The nonlinear operator $A$ again contains the boundary conditions. This nonlinear PDE is regarded as parabolic, whenever the corresponding variational equation, known to be nonautonomous, is parabolic in the sense of (9.21).

**Example 9.3.** Let $\Omega \subset \mathbb{R}^d$ denote a simply connected domain with smooth boundaries (mind you: no reentrant corners, no holes). Let the PDE have the form

$$u_t = \operatorname{div}(a(u)\nabla u) + f, \quad x \in \Omega, \quad 0 \leq t \leq T,$$

with symmetric diffusion matrix $a(u) \in \mathbb{R}^{d\times d}$ and associated natural boundary conditions

$$n^T a(u)\nabla u = 0, \quad x \in \partial\Omega, \quad 0 \leq t \leq T,$$

where $n$ is again the normal unit vector of the boundary $\partial\Omega$.

Applied to this problem class for *implicit Runge–Kutta methods*, the theory [148] supplies an effective order

$$p^* = \min\{p, s + 1 + \beta\} \tag{9.26}$$

with $\beta$ as in (9.23). In [147] the theory was modified for the case of *linearly implicit Runge–Kutta methods*. Here the even stronger restriction

$$p^* = \min\{p, 2 + \beta\},$$

with $\beta$ as in (9.23) comes up. For *ROW-methods*, as in the linear case, an increase of the effective order is possible, if additional conditions of the kind (9.24) can be satisfied.

For *W-methods*, however, such an increase of order is *not* possible: here one essentially gets stuck at the barrier $p^* = 2$. In particular, this also holds for linearly implicit extrapolation methods on the basis of the linearly implicit Euler discretization (LIMEX). At first, this result seemed to contradict the high efficiency of the method as observed in challenging problems. A more detailed analysis in [147] revealed the surprising result that, on the one hand, the order for this method is restricted, but, on the other hand, the coefficients to second order decrease significantly with each extrapolation step. On the basis of these subtle investigations the authors suggested to replace the harmonic subdivision sequence $\mathcal{F}_H = \{1, 2, 3, \ldots\}$ by the sequence $\mathcal{F}_H^+ = \{2, 3, \ldots\}$. Extensive numerical tests, however, have shown that the thus gained reduction of coefficients is not compensated by the increased amount of computation in $\mathcal{F}_H^+$ over $\mathcal{F}_H$.

**Timestep Control.** In Volume 2, Section 6.4, order reduction were already theoretically described for extrapolation methods and, more general, for implicit and linearly implicit Runge–Kutta methods: it occurred in the numerical integration of singularly perturbed ODE systems of the kind

$$u' = f(u, v), \quad \epsilon v' = g(u, v) \quad \text{for} \quad \epsilon \to 0.$$

The question, there as well as here, is: Which order should be used in formula (9.15) to control the timesteps adaptively? In the practical use of extrapolation methods it has turned out that the robust controller property of the timestep control (as worked out, e.g., in Volume 2, Section 5.2) actually permits implementation of the adaptive algorithms with the highest achievable order $p$ in an efficient way. For parabolic PDEs J. Lang [139] recommends a modification. We again start from an *error estimate* at intermediate step $t_n \to t_{n+1} = t_n + \tau_n$, written as

$$[\epsilon_\tau]_{n+1} \doteq \|u_{n+1}^{p+1} - u_{n+1}^{p}\| \doteq C_n \, \tau_n^{p+1}.$$

Formula (9.15) has been derived on the basis of the model assumption $C_{n+1} = C_n$. If one wants to estimate an effective order $p^*$ therefrom, then one gets

$$p^* + 1 \doteq \log \frac{[\epsilon_\tau]_{n+1}}{[\epsilon_\tau]_n} / \log \frac{\tau_n}{\tau_{n-1}}$$

and replaces, in formula (9.15), the value $p$ by the above $p^*$. Obviously, here the information of the two previous timesteps enters. An alternative strategy suggested by K. Gustafsson, M. Lundh, and G. Söderlind [109] uses the model assumption

$$\frac{C_{n+1}}{C_n} = \frac{C_n}{C_{n-1}}.$$

This yields the estimation formula for an "optimal" timestep

$$\tau^* = \frac{\tau_n}{\tau_{n-1}} \sqrt[p+1]{\frac{\rho \ \mathrm{TOL}_\tau \ [\epsilon_\tau]_n}{[\epsilon_\tau]_{n+1}^2}} \ \tau_n$$

with a safety factor $\rho < 1$. Here, too, the information of the two previous discretization steps enters, but with slightly more freedom in the coefficients $C_n$.

## 9.2 Space-time Discretization of Parabolic PDEs

Time dependent PDEs are to be discretized with respect to both time and space. The probably most popular candidate among coupled space-time discretizations with constant spatial mesh size and fixed timesteps is the *Crank–Nicolson scheme* [57], which in time realizes an implicit trapezoidal rule, in space a simple symmetric finite difference scheme. We will not consider it any further here, since it is not adaptive; note, however, that this scheme inherits the instabilities of the trapezoidal rule (cf. Figure 9.3, bottom left), which may show up as unwanted oscillations.

If one discretizes *space first*, which is the presently most popular variant, then one obtains, in general, a large block structured *initial value problem for ODEs* (see the schematic representation in Figure 9.5, left). This approach is called *method of lines* and will be presented in Section 9.2.1. In this approach, adaptivity w.r.t. time (as timestep and order control) is nearly as easy implementable as in ODEs. But adaptivity w.r.t. space is rather restricted in its realization, in an effective way at best only in space dimension $d = 1$.

If one discretizes *time first*, then one comes up with a sequence of *boundary value problems for stationary ordinary ($d = 1$) or partial ($d > 1$) differential equations*, as depicted schematically in Figure 9.5, right. This approach is called *method of time*

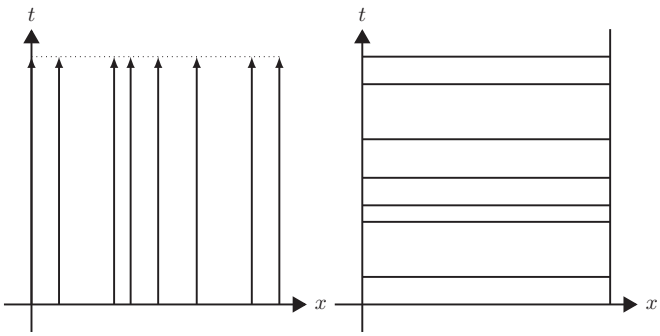


**Figure 9.5.** Ordering in space-time discretization. *Left: method of lines.* Initial value problem for ODEs. *Right: method of time layers*, or also *Rothe method*. Sequence of boundary value problems for stationary differential equations.

*layers* or also *Rothe method*. At this point we want to mention that this approach permits a natural combination of adaptive timestepping and adaptive multigrid methods. This will be worked out in Section 9.2.2. In Figure 9.6, we compare both approaches in a common diagram, where $M_h$ again denotes the earlier introduced *mass matrix*. If one restricts the discretization to *uniformly fixed space grids and constant timesteps*, which is still the standard in the engineering world, then the diagram commutes. In *adaptive* discretizations, however, as treated in this book, the discretization ordering does play an essential role.

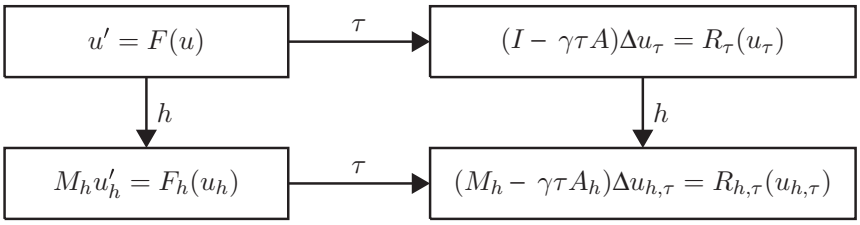


**Figure 9.6.** *Method of lines*: first space discretization ($h$), then time discretization ($\tau$), top left $\rightarrow$ bottom left $\rightarrow$ bottom right. *Method of time layers* or *Rothe method*: first time discretization ($\tau$), then space discretization ($h$), top left $\rightarrow$ top right $\rightarrow$ bottom right.

### 9.2.1 Adaptive Method of Lines

In this approach the space discretization is performed first, in sufficiently large application problems by means of a mesh generator. In *finite difference methods* one then obtains *initial value problems* for large ODE systems of fixed dimension $N$ (for simplicity, we drop the suffix $h$ throughout this section)

$$u' = F(u), \quad u(0) = u_0. \tag{9.27}$$

In *Galerkin methods* one obtains large linearly implicit ODE systems

$$Mu' = F(u), \quad u(0) = u_0, \tag{9.28}$$

where $M$ denotes the arising, usually sparse, $(N, N)$-mass matrix. In both cases, (9.27) was well as (9.28), the space discretization leads to block structured ODE systems. In principle they can be solved by any efficient stiff integrator with adaptive timestep control. However, for the method of lines in particular, it has become clear that linearly implicit integrators of low order are preferable (see Section 9.1.2 above; for a deeper understanding see, e.g., Volume 2, Section 6.4.2, under the keyword "method of lines").

**Dynamic Regridding ($d = 1$).** For problems with "moving fronts", also called *traveling waves*, a transformation onto a "moving grid" is recommended, i.e., an extension $x \to x(t)$, where the number $N$ of nodes remains fixed and a unique correspondence of nodes over all timesteps is possible; in this framework, a change of grid will not cause any interpolation errors (in contrast to static regridding; see below). This method is often denoted as $r$-adaptivity (from *relocation*) – in contrast to $h$-adaptivity for hierarchical mesh refinement or $hp$-adaptivity in case of additional order adaptation. For ease of presentation, we restrict ourselves to one scalar PDE in one space dimension.

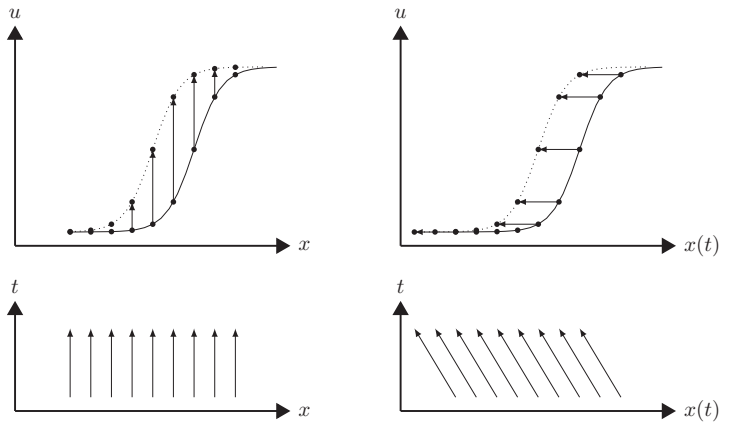


**Figure 9.7.** Moving front of a solution $u$, one time step (due to [163]). *Left*: fixed mesh, solution varies strongly. *Right*: moving mesh, solution remains nearly constant.

In Figure 9.7 the basic idea of the moving fronts is depicted graphically: When integrating on a fixed mesh, the values at all nodes in the front change, seemingly indicating some high dynamics of the problem. In moving coordinates, however, the values of the solution remain nearly constant. As a consequence, timestep control with the fixed grids will suggest "small" stepsizes, with moving grids "large" stepsizes. In moving coordinates, the solution $v(t) = u(x(t), t)$ has the time derivative

$$v' = u_x \, x' + u',$$

which gives rise to a convection term in an extended ODE (9.28) of the form

$$M(v' - u_x \, x') = F(v). \tag{9.29}$$

Obviously, here we need an additional equation for the determination of $x'$. For its derivation, there are quite a number of suggestions, from which we select two here.

The common basic idea in all adaptive methods of lines is to transform the "physical" space variable $x \in [a, b]$ onto a "virtual" space variable $\xi \in [0, 1]$, in which a *uniform* mesh with constant meshsize $\Delta\xi = 1/N$ is introduced. The transformation

has to be chosen such that for the function $x = x(\xi)$ an inverse function $\xi = \xi(x)$ exists. Then, via backtransformation, an in general *nonuniform* mesh in $x$ is generated. In Figure 9.8 the situation is depicted graphically, where the chosen mapping $x(\xi)$ is obviously monotone.

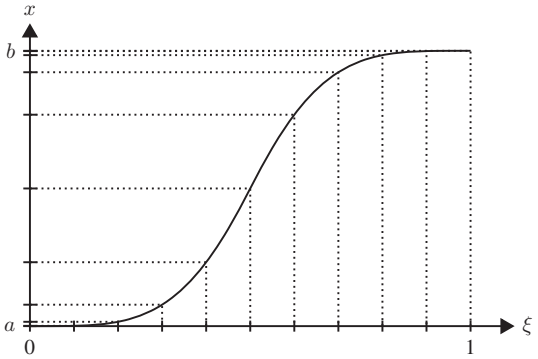


**Figure 9.8.** Uniform "virtual" mesh in $\xi \in [0, 1]$ transformed onto a nonuniform "physical" mesh in $x \in [a, b]$.

*Monitor function.* Here we follow the survey paper [45] from 2009 as well as the more recent book [125, Chapter 5] by W. Huang and R. D. Russell. The method described there extensively starts from an equilibration with respect to a *monitor function* $\mu(\xi, t)$, which leads to a condition of the form

$$\mu x_\xi = \theta(t) = \frac{\int_{\Omega_x} \mu(x,t)\,\mathrm{d}x}{\int_{\Omega_\xi} \mathrm{d}\xi},$$

where the domains $\Omega_x, \Omega_\xi$ exist in the corresponding coordinate spaces. If one chooses $\mu > 0$ herein, then $x_\xi > 0$ holds, i.e., the mapping $x(\xi)$ is monotone and there exists a unique inverse function. Differentiation w.r.t. $\xi$ then supplies the nonlinear boundary value problem

$$\mu_\xi x_\xi^2 + \mu x_{\xi\xi} = 0, \quad x(0) = a, \quad x(1) = b. \tag{9.30}$$

The only thing left to do is the to determine of $\mu$ from some first principle. An intuitive choice is the *arc length*, i.e.,

$$\mu = \sqrt{1 + u_x^2}.$$

It is known to be not invariant under rescaling of the variable $x \to x/\sigma$, so that the ansatz is generalized according to

$$\mu = \sqrt{1 + \sigma^2 u_x^2}$$

with a scaling quantity $\sigma$ given into the hands of the user. Inserting $\mu$ into the boundary value problem (9.30) one ends up with (9.29), a differential-algebraic system with nonlinear algebraic equality constraint. In the context of *finite difference methods* ($M = I$ in (9.28)) the technique of C. E. Pearson [168] can be applied on the nonuniform spatial grid in $x$ (see Section 3.3.1 above). In this way the time-dependent adaptation of the space grid is fixed.

What is missing, however, is a strategy for the choice of the virtual grid size $\Delta\xi$ or the number $N$ of nodes, respectively, which must come from an estimate of the space discretization error. Should an estimation method for the discretization error not be available, then the interpolation error is estimated instead. However, if $N$ is changed, then the solution values on the actual grid need to be transferred to a new grid, where again the interpolation error creeps in.

The concept presented here can in principle be carried over to higher space dimension $d > 1$ (see, e.g., [125, Chapter 4]). There and in [45] a number of articles are cited in which this has been successfully done. The examples given for the choice of a monitor function $\mu$, however, are chosen more for illustration purposes, and so do not permit any conclusions about the efficiency in the general case.

*Minimization problem.* Here we present suggestions by J. M. Hyman [126] and L. R. Petzold [171] in the framework of finite difference methods, i.e., based on (9.27). In this approach, with the aim of achieving time stepsizes $\tau$ as large as possible, the requirement

$$\|v'\|_2^2 = (u' + u_x x')^T (u' + u_x x') = \min$$

is imposed. Minimization w.r.t. $x'$ leads to the PDE

$$(u' + u_x x')^T u_x = u_x^T v' = 0.$$

Combination with (9.27) then supplies the coupled PDE system

$$\begin{aligned} M(v' - u_x\, x') &= F(v), \\ u_x^T v' &= 0. \end{aligned} \tag{9.31}$$

Space discretization due to C. E. Pearson [168] finally leads to some ODE system in $2N$ variables.

Experience has shown, however, that by this approach a *crossing of nodes* is not systematically avoided. For an illustration of this phenomenon we start from neighboring nodes $x_i < x_{i+1}$ with local velocities $x_i', x_{i+1}'$ that can be computed from the above ODE system. Let $x_{i+1}' < x_i'$. In order to assure that in the next timestep the condition

$$x_i + \tau x_i' < x_{i+1} + \tau x_{i+1}'$$

holds as well, the maximal time stepsize

$$\tau_{\max} < \frac{x_{i+1} - x_i}{x_i' - x_{i+1}'} \tag{9.32}$$

must not be exceeded. Therefore in [156, 171] it has been suggested to couple another functional in, quasi as a "repelling term" that should work to prevent the crossing of nodes. Therefore, in view of (9.32), additionally

$$\sum_{i=2}^{N} \left\| \frac{x_i' - x_{i-1}'}{x_i - x_{i-1}} \right\|_2^2 = \min \tag{9.33}$$

is introduced. This leads to the $N-2$ additional conditions

$$\frac{x_i' - x_{i-1}'}{(x_i - x_{i-1})^2} - \frac{x_{i+1}' - x_i'}{(x_{i+1} - x_i)^2} = 0, \quad i = 2, \ldots, N-1.$$

These equations may be interpreted as space discretization of the PDE $-x'_{xx} = 0$. If this is coupled by means of a Lagrange multiplier $\lambda > 0$, then we obtain, instead of (9.31), alternatively

$$M(v' - u_x\, x') = F(v),$$
$$u_x^T v' + \lambda x'_{xx} = 0, \quad \lambda > 0.$$

Spatial discretization then again yields a system of $2N$ coupled ODEs, which can be solved numerically by a stiff integrator. The parameter $\lambda$, however, must be chosen in each example from scratch by trial and error, a problem-independent robust choice is not known up to now. In this way the case of crossing traveling waves becomes tractable, as illustrated in Figure 9.9; for a not further specified example see [163].

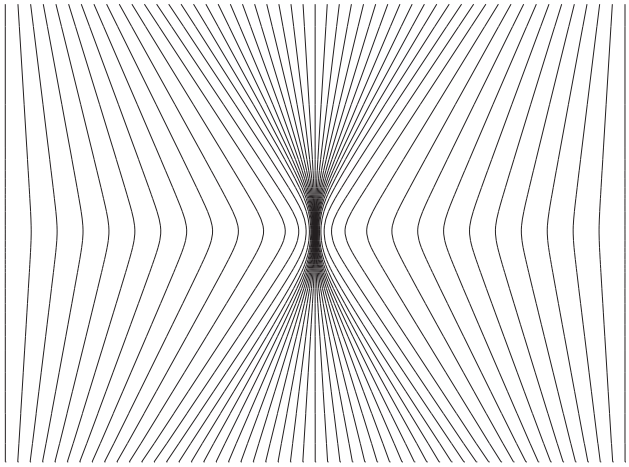

**Figure 9.9.** Adaptive method of lines [163]: dynamic regridding with "repelling functional" (9.33) at the example of crossing fronts. Cf. also Figure 9.13.

This approach to dynamic grid adaptation can be generalized to space dimension $d > 1$, too. For a survey on this topic we refer to the article [145]. However, as documented in [157], a grid "tangling" cannot be fully excluded, despite the addition of the "repelling functional". This might be the reason why this method has not generally prevailed yet.

**Static Regridding.** In this class of methods a new grid is constructed in every timestep (or also after a fixed number of timesteps). The number $N$ of nodes is no longer constant, but varies within prescribed limits $N_{\min} \leq N(t) \leq N_{\max}$. There, inevitably interpolation errors occur, which must be carefully monitored apart from the discretization errors. In [163], two generally applicable strategies are suggested:

- In order to possibly avoid interpolation errors, two global grids are used in parallel, a coarse grid $G$ with virtual grid size $\Delta\xi$ and a finer grid $G^+$ with grid size $\Delta\xi/2$. Thus, in local refinement, the additional value to be inserted can be just taken from $G^+$. For grid coarsening, $G$ is sufficient anyway.
- In order to assure a globally monotone mapping $x(\xi)$, the cubic Hermite interpolation due to [97] is modified such that it remains locally monotone. The usual cubic Hermite interpolation (see, e.g., Volume 1, Section 7.1.2) over the interval $I_i = [x_i, x_{i+1}]$ requires as input the function values and their derivatives at the interval boundaries, i.e., within each time layer the values $u(x_i), u_x(x_i)$, $u(x_{i+1}), u_x(x_{i+1})$. In finite difference methods, however, only *difference quotients* at the boundaries are available as approximations, which reduces the approximation order from $p = 3$ to a poor $p = 1$; for compensation, superconvergence occurs at the corresponding interval midpoints $x_{i+1/2} = x(\xi_i + \Delta\xi/2)$, i.e., pointwise $p = 2$. These are the very nodes additionally held in $G^+$.

The goal of static regridding is to equilibrate the global discretization error by means of the choice of nodes such that in each node roughly the same local error occurs. The construction of adaptive grids is carried out stepwise by virtue of the following two algorithmic pieces:

(a) In each node $x_i$ the discretization error is estimated; let $[\epsilon_i^x]$ denote a computationally accessible estimate of the local discretization error $\epsilon_i^x$. If a good estimate of the discretization error is not available, it is replaced by an estimate of the interpolation error, which, in general, is easily available.

(b) Let $\mathrm{TOL}_h$ denote a user prescribed global upper bound and $\sigma^{\pm}$ local bounds for the spatial error. Then the new grid $G(t+\tau)$ is generated from the actual grid $G(t)$ according to the following *rules*:

- if $[\epsilon_i^x] < \sigma^-$, then the node $x_i$ is eliminated, unless it is a boundary node;
- if $[\epsilon_i^x] \geq \sigma^-$, then the node $x_i$ is kept;
- ff $[\epsilon_i^x] \geq \sigma^+$, then two additional new nodes $x_{i\pm1/2}$ are inserted;
- one ensures globally that the new grid remains *quasi-uniform*, i.e., that neighboring subintervals differ only within prescribed limits in their lengths; this leads to the recipe $h_i / h_{i+1} \in [1/\alpha, \alpha]$ for a parameter $\alpha \approx 2-3$ (to be possibly modified by the user).

With these means one hopes to be able to roughly monitor the quality of the approximations to be computed (for an illustration of these rules see Figure 9.10).

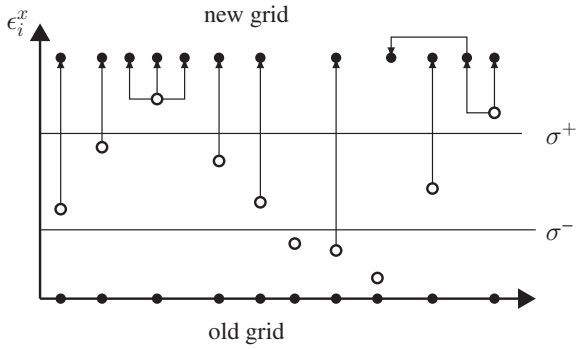


**Figure 9.10.** Method of lines: refinement and coarsening strategies in static regridding.

**Software.** In his dissertation [163] U. Nowak elaborated an efficient adaptive method of lines for parabolic PDEs in one space dimension. There he combined the above presented concepts of dynamic and static regridding by means of cleverly selected heuristics. For the estimation of the spatial error he uses a coupled extrapolation scheme in space and time that builds upon an asymptotic expansion in space and time, which, however, is mathematically not sufficiently backed up (cf. Volume 2, Sections 4.3 and 6.4.2 on extrapolation methods). In the engineering community, his program PDEX1M [164] is highly reputated, due to its efficiency (as an example see, e.g., [91]).

**Example 9.4.** This 1D test problem from combustion chemistry [169] describes a traveling combustion front with strongly varying velocity on a domain $\Omega = ]-25, 10[$ for $t \in [0, 15]$. The PDE system

$$\begin{aligned} T_t &= T_{xx} + R(T, Y), \\ Y_t &= \frac{1}{L} Y_{xx} - R(T, Y) \end{aligned}$$

models the evolution of the temperature $T$ and of the fuel concentration $Y$ with the initial and boundary conditions

$$\begin{aligned} T(x, 0) &= \min(\mathrm{e}^x, 1), & T(-25, t) &= 0, \quad T(10, t) = 0, \\ Y(x, 0) &= \max(1 - \mathrm{e}^{Lx}, 0), & Y(-25, t) &= 1, \quad Y(10, t) = 0. \end{aligned}$$

Therein

$$R(T, Y) = \frac{\beta^2}{2L} Y \exp\left(-\frac{\beta(1 - T)}{1 - \alpha(1 - T)}\right)$$

denotes the reaction rate and $\alpha = 0.8$ (heat release), $\beta = 20$ (activation energy) and $L = 2.0$ (Lewis number) are dimensionless parameters.

Nonadaptive methods of lineshave turned out to be unable to solve this problem with sufficient accuracy. In Figure 9.11, top and center, we illustrate the nodal flux of the

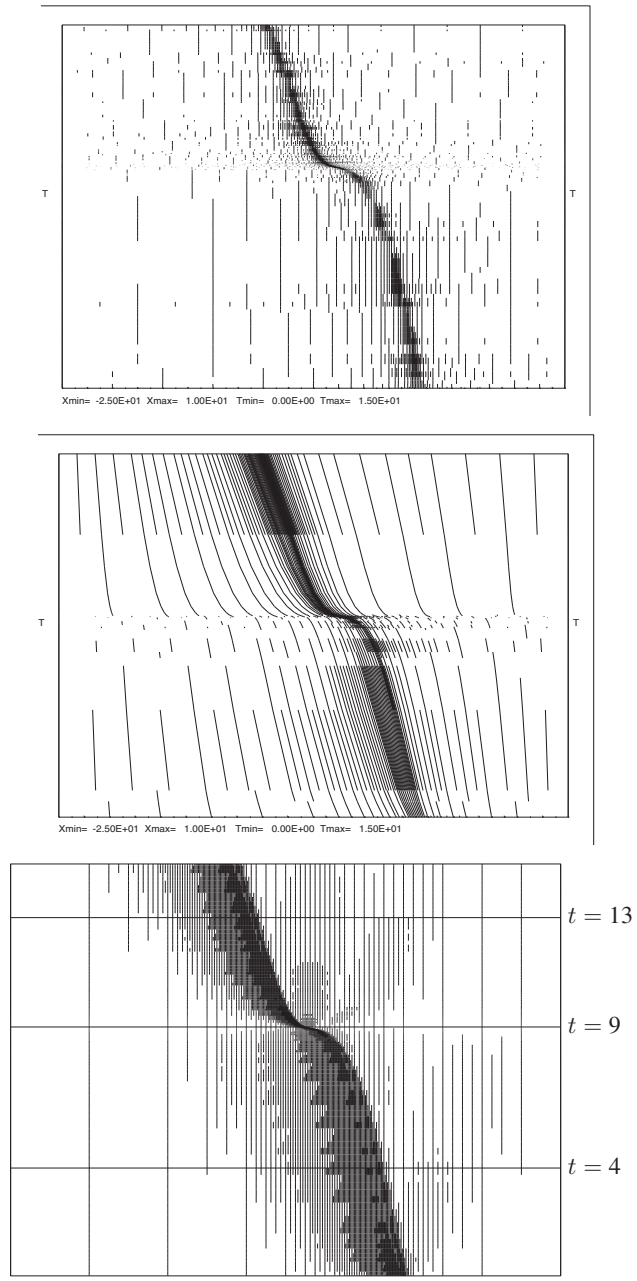


**Figure 9.11.** Nodal flux in Example 9.4. *Top*: adaptive method of lines [163]: static regridding. *Center*: adaptive method of lines [163]: mixed static-dynamic regridding. *Bottom*: adaptive method of time layers [72]. Temperature profiles for $t = 4, 9, 13$ see Figure 9.12.

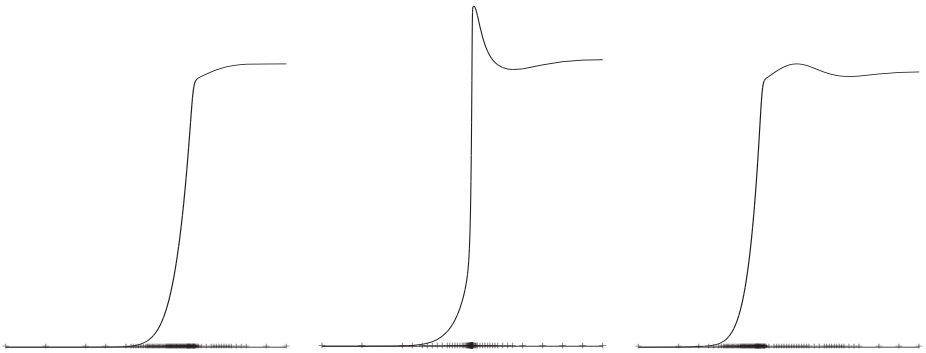

**Figure 9.12.** Example 9.4 (cf. Figure 9.11): temperature profiles $T(x,t)$ at the time layers $t = 4, 9, 13$ with corresponding automatic choice of nodes.

discrete solution in the adaptive method of lines due to U. Nowak [163]. For comparison, we already give at this stage the nodal flux of the method of time layers, which will be discussed below in Section 9.2.2 (see Figure 9.11, bottom). In order to give some insight into the difficulty of the problem, we additionally show the temperature profiles at critical intermediate time layers in Figure 9.11.

In [150], the combustion chemist U. Maas published numerical comparisons of simulations at difficult problems from engine combustion (see also the monograph [44]). His comparisons with respect to speed and accuracy focused on the extrapolation code LIMEX and the multistep code DASSL by L. R. Petzold [170], where the one-step method performed better by far. The reason for this performance difference is that, after a change of grid, the multistep methods must restart and build up from order $k = 1$ to the order $k$ of the previous step (see Volume 2, Section 7.4, under the key word "startup calculation"); as an alternative, a so-called "warm restart" was suggested (see, e.g., [145]), in which, however, a sufficient monitoring of the interpolation error is lacking. As another variant, a restart by means of some one-step method of higher order has been discussed, which, however, in case of frequently changing grids leads to the nearly exclusive use of this very one-step method. For problems with high spatial-temporal dynamics, this structural property of adaptive multistep methods (like DASSL) leads to a drastic slow-down of the computations; this holds in particular for space dimension $d > 1$.

### 9.2.2 Adaptive Method of Time Layers

This method is also called *adaptive Rothe method*, since around 1930 E. Rothe [176] applied such a method – for uniform space and time grids – as a technique of proving existence of the solution of the diffusion equation. Around 1989 F. A. Bornemann [31]

recognized that it constitutes the ideal algorithmic frame to realize *full adaptivity in space and time* for parabolic PDEs.

**Preliminary Considerations.** Let us start from the situation as represented in Figure 9.6, i.e., from the abstract Cauchy problem

$$u' = F(u), \quad u(0) = u_0 \in H^1(\Omega), \quad F : H^1(\Omega) \to H^1(\Omega)^*. \tag{9.34}$$

Such a problem is regarded as parabolic whenever its linearization, i.e., the linear (in general nonautonomous) variational equation, is parabolic. If we discretize this problem first with respect to time, we obtain a sequence of linear boundary value problems with solutions $\{u_\tau(t), t = t_0, t_1, \dots\} \subset H^1(\Omega)$.

As worked out in Section 9.1.2, in the finite dimensional case of PDEs the situation is as follows. Whenever the exact Jacobian matrix $J = F'(u_0)$ can be evaluated, then ROW-methods may be constructed, otherwise W-methods or linearly implicit extrapolation methods. In the infinite dimensional case of interest here, however, an exact functional derivative $F'(u_0)$ is not available, but at best, in the context of FE-methods, a projection on the FE-subspace $S_h$ or an approximation of it. Thus, if we interpret the time integration within the method of time layers as in Section 9.1.3 as *function space method*, we need to consider either W-methods or ROW-methods, where in the latter the quantities to be computed can be approximated *to sufficient accuracy* such that the order of the time integration method is essentially preserved. From (9.26) we know that in W-methods we can expect an effective order of maximally $p^* = 2$. This is why, up to now, in the developments of the method of time layers ROW-methods dominate, where adaptive mesh refinement realizes the sufficiently accurate approximation of $F'(u_0)$. Upon satisfying the additional conditions (9.24) the order reduction elaborated in Section 9.1.3 can be avoided so that time integrators of high effective order can be implemented.

**Timestep Control.** From (9.15) one obtains an optimal timestep via the formula

$$\tau^* = \sqrt[p+1]{\frac{\rho \ \mathrm{TOL}_\tau}{[\hat{\epsilon}_\tau]}} \ \tau \tag{9.35}$$

with $\rho < 1$ and

$$[\hat{\epsilon}_\tau] = \|u_\tau(\tau) - \hat{u}_\tau(\tau)\| \doteq C_p \ \tau^{p+1},$$

where in our context now $u_\tau, \hat{u}_\tau \in H_0^1(\Omega)$. For the time being, the norm $\|\cdot\|$ may remain unspecified. The associated accuracy monitor would then read

$$[\hat{\epsilon}_\tau] \leq \mathrm{TOL}_\tau. \tag{9.36}$$

Obviously, the formula as such cannot be directly evaluated. At best, the norm of the difference $[\hat{\epsilon}_\tau]$ can be approximated by

$$[\hat{\epsilon}_\tau]_h = \|u_{h,\tau}(\tau) - \hat{u}_{h,\tau}(\tau)\|, \quad u_{h,\tau} \in S_h \subset H_0^1(\Omega) \ , \tag{9.37}$$

i.e., by the difference of discretizations $u_{h,\tau}$ and $\hat{u}_{h,\tau}$. In order to realize adaptivity of the algorithm w.r.t. time in a reliable way, one must assure that

$$[\hat{\epsilon}_\tau] \approx [\hat{\epsilon}_\tau]_h \leq \mathrm{TOL}_\tau.$$

This can be done in connection with adaptive multigrid methods, as will be demonstrated below in Section 6.1.

Already at this stage we want to indicate that in the explicit evaluation of $[\hat{\epsilon}_\tau]_h$ special attention must be paid to the possible occurrence of *numerical extinction of leading digits*. For example, if one computes $u_\tau$ and $\hat{u}_\tau$ independently from each other on different meshes, then the evaluation of the above difference indeed induces a massive blow-up of the various discretization errors. In order to cope with this problem, two different strategies have been developed, which we are going to present next.

**Linear problems**

F. Bornemann worked out the Rothe method for linear nonautonomous scalar parabolic PDEs, first in one space dimension [31, 32], later in two space dimensions [33]. That is why we consider this special case first and start from the abstract linear Cauchy problem

$$u' = Au + f(t), \quad u(0) = u_0 \tag{9.38}$$

with $u \in H_0^1(\Omega)$ and $-A(\cdot)$ an elliptic operator. As time discretization he began with the extrapolated linearly implicit Euler method (code LIMEX; see Section 9.1.2), but did not couple the grids for the discrete solutions associated with neighboring orders w.r.t. $\tau$ and thus had to compensate large error amplification factors. Nevertheless, already in this first realization, the principal advantage of an adaptive method of time layers could be seen: in contrast to the method of lines, the space discretization is not frozen over all time layers, so that a crossing of traveling fronts does not exhibit any problem. For illustration, see a test problem suggested by M. Bietermann and I. Babuška [28].

**Example 9.5.** In this example the solution $u$ consists of two components

$$u = u^{(1)} + u^{(2)},$$

where

$$u^{(1)}(x,t) = 0.25\Big(1 + \tanh\big(100(x - 10t)\big)\Big),$$
$$u^{(2)}(x,t) = 0.25\Big(1 + \tanh\big(80(1 - x - 30t)\big)\Big).$$

Obviously, the two solution parts model two "countertraveling" waves with different velocities (10 and 30). In the PDE

$$u_t = u_{xx} + f(t), \quad t > 0,\ x \in [0,1]$$

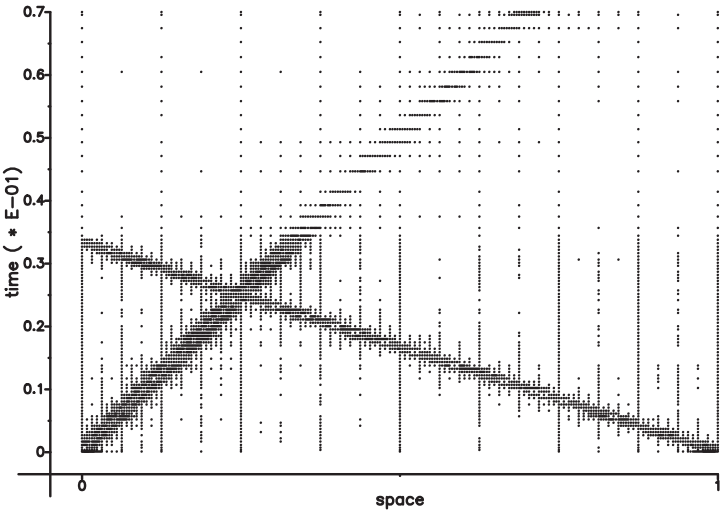


**Figure 9.13.** Adaptive method of time layers [31]: nodal flux for two crossing traveling fronts. Cf. also Figure 9.9.

Dirichlet boundary conditions and the function $f(t)$ are chosen such that the above superposition ends up as the unique solution. In Figure 9.13 we show the nodal flux obtained when applying the adaptive method of time layers based on LIMEX (see Section 9.1.2); in each time layer an adaptive multigrid method has been applied.

In order to circumvent the above problem of unwanted amplification of space discretization errors in $[\epsilon_\tau]_h$ (see (9.37)), F. A. Bornemann [32, 33] developed two specific classes of time discretizations, first only for *autonomous* source terms $f$: Let, for increasing order $p$, solutions $u_\tau^p \in H^1(\Omega)$ be defined according to

$$u_\tau^{p+1} = u_\tau^p + \Delta u_\tau^p, \quad p = 0, 1, \ldots.$$

The special idea here is to compute the differences $\Delta u_\tau^0, \Delta u_\tau^1, \ldots$ *multiplicatively* from one another.

In a linear stability model for stiff ODEs (see above Section 9.1.1) this idea corresponds to the rational functions $r_p(z)$, recursively computed according to

$$r_{p+1}(z) = r_p(z) + \rho_p(z), \quad \rho_{p+1}(z) = \gamma_p \frac{z}{1-z} \rho_p(z), \quad p = 0, 1, \ldots,$$

where $z = \lambda\tau \in \mathbb{C}$ belongs to eigenvalues $\lambda(A)$. As corresponding starting values we have $r_0(z) = \rho_0(z) = 1$. In our context we are interested in the special case $\Re(z) < 0$, which is why the methods are started with an implicit Euler step

$$r_1(z) = \frac{1}{1-z}.$$

A more accurate analysis shows that for the choice of $\rho_1$ only two reasonable possibilities exist,

$$\rho_1^A(z) = -\frac{1}{2}\frac{z^2}{(1-z)^2} \quad \text{and} \quad \rho_1^L(z) = -\frac{1}{2}\frac{z^2}{(1-z)^3},$$

which lead to $A(\alpha)$- and $L(\alpha)$-stable methods. Apart from automatically avoiding the error amplification when computing the difference (9.37) the small stage number $s = p$ is attractive.

A transfer to the linear *nonautonomous* case, i.e., with $f(t)$ in (9.38), is worked out in [32], there, however, only for the first two steps of the $A$-stable discretization, which means up to $s = p = 2$. (Note that the nonautonomous case is included in the general nonlinear case if the usual autonomization trick is applied; see Volume 2).

**Remark 9.6.** For the numerical solution of countable differential equations (roughly: discrete PDEs) the method of time layers is also crucial in the context of *discrete* Galerkin methods. For this problem class, M. Wulkow [220] suggested a transfer of the above mentioned method from the nonautonomous case to the general *nonlinear* case. This method has proved to be successful in practice (see, e.g., [221]); however, for the general case there is still no consistency theory yet, which is why we do not pursue it any further here.

### Nonlinear problems

In 2001, J. Lang [139] managed to extend the adaptive method of times layers efficiently to nonlinear parabolic PDEs (9.34). He employed ROW-methods as linearly implicit time discretizations, avoiding order reduction (see Section 9.1.3) by satisfying the additional conditions (9.24).

Due to (9.12) ROW-methods require the solution of a formal linear system in each time step, here written as operator equation

$$\begin{aligned} &i = 1, \dots, s: \\ &(I - \gamma\tau A)k_i = \tau \sum_{j=1}^{i-1} (\gamma_{ij} - \alpha_{ij}) A k_j + F\Big(u_0 + \tau \sum_{j=1}^{i-1} \alpha_{ij} k_j\Big). \end{aligned} \tag{9.39}$$

There the eigenvalues $\lambda_i$ of the operator $-A = -F'(u_0)$ are restricted according to $\Re(\lambda_i) \le C$ for some $C > 0$; due to $\gamma\tau > 0$ the operator $I - \gamma\tau A$ is then elliptic for sufficiently small timesteps $\tau < 1/\gamma C$. Because of the special structure of ROW-methods the above system (9.39) is recursive, i.e., the $k_1, \dots, k_s \in H_0^1(\Omega)$ can be computed one after the other by solving *linear elliptic boundary value problems*. This is why for nonlinear parabolic PDEs predominantly linearly implicit time integrators are applied.

In an *embedded* ROW-method (see Section 9.1.2) two discrete solutions for neighboring orders can be computed from these intermediate quantities as follows:

$$u_\tau(\tau) \;= u_0 + \tau \sum_{i=1}^{s} b_i k_i \;\in H_0^1(\Omega), \quad \text{order } p+1,$$

$$\hat{u}_\tau(\tau) \;= u_0 + \tau \sum_{i=1}^{s} \hat{b}_i k_i \;\in H_0^1(\Omega), \quad \text{order } p.$$

As error estimator w.r.t. time one obtains, purely formally,

$$[\hat{\epsilon}_\tau] = \|u_\tau(\tau) - \hat{u}_\tau(\tau)\| = \tau \Big\| \sum_{i=1}^{s} (b_i - \hat{b}_i) k_i \Big\| \doteq C_p\, \tau^{p+1}.$$

None of these formulas can be directly evaluated numerically. Instead, sufficiently accurate spatial approximations must be computed, e.g., by adaptive multilevel methods.

For this reason, we turn to the weak formulation in a finite element space $S_h \subset H_0^1(\Omega)$. We obtain a recursive system for the approximations $k_{h,1}, \ldots, k_{h,s} \in S_h$ of the form

$$\begin{aligned} &i = 1, \ldots, s: \\ &\langle (I - \gamma\tau A) k_{h,i}, v \rangle \\ &\quad = \tau \sum_{j=1}^{i-1} (\gamma_{ij} - \alpha_{ij}) \langle A k_{h,j}, v \rangle + \Big\langle F\Big(u_0 + \tau \sum_{j=1}^{i-1} \alpha_{ij} k_{h,j}\Big), v \Big\rangle, \quad v \in S_h. \end{aligned} \tag{9.40}$$

In this FE-approximation we now have

$$u_{h,\tau}(\tau) = u_0 + \tau \sum_{i=1}^{s} b_i k_{h,i} \in S_h,$$

$$\hat{u}_{h,\tau}(\tau) = u_0 + \tau \sum_{i=1}^{s} \hat{b}_i k_{h,i} \in S_h.$$

The space discretization error can be approximately determined by a hierarchical extension $S_h^+ = S_h \oplus S_h^\oplus$, $S_h^\oplus = \operatorname{span}(\varphi_l)_{l=1,\ldots,n}$ defining

$$k_{h,i}^\oplus = \sum_l \eta_{h,i,l} \varphi_l \in S_h^\oplus$$

via a hierarchical error estimator (e.g., the DLY-estimator; see Section 6.1.4):

$$\begin{aligned} \langle (I - \gamma\tau A)\varphi_l, \varphi_l \rangle \eta_{h,i,l} &= \tau \sum_{j=1}^{i-1} (\gamma_{ij} - \alpha_{ij}) \langle A k_{h,j}^+, \varphi_l \rangle \\ &\quad + \Big\langle F\Big(u_0 + \tau \sum_{j=1}^{i-1} \alpha_{ij} k_{h,j}^+\Big), \varphi_l \Big\rangle - \langle (I - \gamma\tau A) k_{h,i}, \varphi_l \rangle. \end{aligned} \tag{9.41}$$

With this, the terms $k_{h,i}^+ = k_{h,i} + k_{h,i}^\oplus$ and

$$u_{h,\tau}^+(\tau) = u_0 + \tau \sum_{i=1}^{s} b_i k_{h,i}^+ \in S_h^+,$$

$$\hat{u}_{h,\tau}^+(\tau) = u_0 + \tau \sum_{i=1}^{s} \hat{b}_i k_{h,i}^+ \in S_h^+$$

may be defined. For ease of reading, all four notations are arranged in Table 9.2.

**Table 9.2.** Four different approximations of the solution $u(\tau) \in H_0^1(\Omega)$.

| $h$ \ $\tau$ | order $p$ | order $p+1$ |
|---|---|---|
| $S_h$ | $\hat{u}_{h,\tau}$ | $u_{h,\tau}$ |
| $S_h^+$ | $\hat{u}_{h,\tau}^+$ | $u_{h,\tau}^+$ |

For the acceptance test (9.36) we replace the unavailable estimate $[\hat{\epsilon}_\tau]$ by the most accurate available spatial estimate

$$[\hat{\epsilon}_\tau]_h^+ = \|u_{h,\tau}^+ - \hat{u}_{h,\tau}^+\| = \tau \Big\| \sum_{i=1}^{s} (b_i - \hat{b}_i) k_{h,i}^+ \Big\| \le \mathrm{TOL}_\tau, \tag{9.42}$$

which we also insert into the timestep control (9.35).

In addition to the pure time discretization error $\hat{\epsilon}_\tau$ we obtain the space discretization error

$$\hat{\epsilon}_h = \|\hat{u}_\tau - \hat{u}_{h,\tau}\| = \tau \Big\| \sum_{i=1}^{s} \hat{b}_i (k_i - k_{h,i}) \Big\|$$

and bound the suitable estimator by virtue of

$$[\hat{\epsilon}_h] = \|\hat{u}_{h,\tau}^+ - \hat{u}_{h,\tau}\| = \tau \Big\| \sum_{i=1}^{s} \hat{b}_i k_{h,i}^\oplus \Big\| \le \mathrm{TOL}_h. \tag{9.43}$$

For the estimation of the total error we replace $u(\tau) \in H_0^1(\Omega)$ by its best available approximation $u_{h,\tau}^+ \in S_h^+$ according to

$$\hat{\epsilon}_{h,\tau} = \|u(\tau) - \hat{u}_{h,\tau}\| \approx \|u_{h,\tau}^+ - \hat{u}_{h,\tau}\| =: [\hat{\epsilon}_{h,\tau}]$$

and thus obtain the desired bound

$$\begin{aligned} [\hat{\epsilon}_{h,\tau}] &= \|u_{h,\tau}^+ - \hat{u}_{h,\tau}\| \\ &\le \|u_{h,\tau}^+ - \hat{u}_{h,\tau}^+\| + \|\hat{u}_{h,\tau}^+ - \hat{u}_{h,\tau}\| = [\hat{\epsilon}_\tau]_h^+ + [\hat{\epsilon}_h] \le \mathrm{TOL}_\tau + \mathrm{TOL}_h \le \mathrm{TOL} \end{aligned}$$

with a user prescribed tolerance TOL.

**Computational Complexity Model.** Following [33] we split the required accuracy into its temporal and spatial parts

$$\mathrm{TOL}_\tau = \sigma \mathrm{TOL}, \quad \mathrm{TOL}_h = (1-\sigma)\mathrm{TOL}$$

with a parameter $0 < \sigma < 1$ to be determined. The choice of $\sigma$ does not influence the size of the total error, but it does very much so the computational complexity. In particular, a reduction of the spatial error mostly goes with a significantly larger computational amount than the corresponding reduction of the temporal error.

For a more detailed analysis we use the following simple model. We start from the situation that the local timesteps have been selected by the timestep control. In Volume 1, Section 9.5.3, we derived the local timestep control (in the simple example of numerical quadrature) on the basis of a model in which the *local amount per unit timestep* has been minimized and thus the error globally equilibrated. This model is applicable to every evolution problem, i.e., also in the present case. When using an optimal multigrid solver, the local computational amount $W$ per timestep $\tau$ depends linearly on the number $N_h = \dim S_h$ of degrees of freedom:

$$W \sim N_h/\tau.$$

Note that also in the *time adaptive* case, this result is equivalent to the model

$$W \sim N_h N_\tau \quad \text{with} \quad N_\tau \sim 1/\tau.$$

For *space adaptive* meshes we have, asymptotically just as in the quasi-uniform special case,

$$\mathrm{TOL}_h \approx \hat{\epsilon}_h \sim N_h^{-q/d} \|\hat{u}_\tau - \hat{u}_{h,\tau}\|_A,$$

where the convergence order $q$ of the finite elements depends both on the ansatz order of the space $S_h$ and on the chosen norm, i.e., in linear finite elements $q = 1$ for the energy norm (Theorem 4.19), $q = 2$ for the $L^2$-norm (Theorem 4.21). Because of $\hat{u}_0 - \hat{u}_{h,0} = 0$ we have $\|\hat{u}_\tau - \hat{u}_{h,\tau}\|_A \sim \tau$, from which we conclude

$$(1-\sigma)\,\mathrm{TOL} = \mathrm{TOL}_h \sim N_h^{-q/d}\tau \quad \Rightarrow \quad N_h \sim \left(\frac{\tau}{(1-\sigma)\,\mathrm{TOL}}\right)^{d/q}.$$

A more detailed consideration leads to the insight that TOL should not be regarded as independent of the timestep $\tau$. In fact, when integrating over a fixed time interval of length $T$ (not discussed further here) the local errors of an order of magnitude of TOL will give rise to a final accuracy $\epsilon(T)$ as discussed extensively in Volume 2, Section 5.5. In parabolic problems, which are known to be dissipative, the local errors will be damped in the stationary phase, so that then $\epsilon(T) \approx \mathrm{TOL}$ arises. In the transient phase, however, the timestep control will, for *fixed final accuracy*, establish the relation

$$\epsilon(T) \approx N_\tau\,\mathrm{TOL} \quad \Rightarrow \quad \mathrm{TOL} \sim \tau.$$

For the local timesteps one gets

$$\tau^{p+1} \sim \mathrm{TOL}_\tau = \sigma\,\mathrm{TOL} \sim \sigma\tau \quad \Rightarrow \quad \tau \sim \sigma^{1/p}.$$

This finally supplies the functional dependence

$$W \sim (1-\sigma)^{-d/q}\sigma^{-1/p} =: \varphi(\sigma).$$

As $\varphi(0)$ and $\varphi(1)$ are unbounded positive, the condition $\varphi'(\sigma) = 0$ yields the minimum in

$$\sigma_{\min} = \frac{q}{q+dp}. \tag{9.44}$$

As expected, the result (9.44) produces a smaller $\sigma$ for larger space dimension, higher order in time and lower order in space, and thus permits a relatively larger spatial part in the total error.

Let us exemplify the case for linear finite elements, i.e., $q = 2$ in the $L^2$-norm or $q = 1$ in the energy norm, respectively. For the implicit Euler discretization ($p = 1$) one then gets $\sigma_{\min} = 1/(1 + d/2)$ (nearly the result in [33]) or $\sigma_{\min} = 1/(1 + d)$, respectively. For ROW-methods of order $p = 4$ and in space dimension $d = 3$ the formula (9.44) supplies the value $\sigma_{\min} = 1/13$ and thus $\mathrm{TOL}_\tau \approx \mathrm{TOL}_h/12$. This low value expresses the fact that a reduction of the spatial error causes a significantly larger amount of work than a reduction of the temporal error. Facing this accuracy discrepancy, it may be asked whether the hierarchical spatial error estimator used in (9.42) will at all permit a reliable determination of the temporal error – after all, it still depends, by a not too large factor, on the parameter $\beta$ of the saturation property (see Definition 6.7). Therefore, in practice, the factor $\sigma$ should be chosen sufficiently large, e.g., by the restriction $\sigma \geq 1/4$. The considerations so far lead us to the following algorithm.

**Algorithm 9.7.** *Adaptive method of time layers.*
$(u^+_{h,\tau}, \tau, \tau^*) := \mathrm{AMOT}(u, \tau^*, \mathrm{TOL}, \sigma)$

**do**
  $\tau := \tau^*$
  **for** $i := 1$ **to** $s$ **do**
    determine $k_{h,i}$ and $k^{\oplus}_{h,i}$ according to (9.40) and (9.41)
  **end for**
  $[\hat{e}_h] := \tau \sum_{i=1}^{s} \hat{b}_i k^{\oplus}_{h,i}$
  **if** $|[\hat{\epsilon}_h]| \geq \mathrm{TOL}_h$ **then**
    refine the mesh, where $|[\hat{e}_h]|$ is large
  **end if**
  compute $\tau^*$ according to (9.35)
**until** $[\hat{\epsilon}_\tau]^+_h \leq \mathrm{TOL}_\tau$ and $[\hat{\epsilon}_h] \leq \mathrm{TOL}_h$

Note that *all* stage vectors $k_{h,i}$ and $k_{h,i}^{\oplus}$ need to be recomputed on a finer mesh if (9.43) is violated. Fortunately, this costly case occurred rather rarely in our numerical tests.

**Mesh Coarsening.** Up to now we have discussed only mesh *refinement*. However, especially for nonlinear time-dependent equations a shift of the spatial solution structure in time is typical. Algorithm 9.7 merely realizes mesh refinement in case of insufficient accuracy so that a large part of the domain will be gradually filled with small elements that were only necessary at some earlier time layer. This phenomenon does not affect the accuracy of the solution, but but it doese affect the computational efficiency. For this reason, a *coarsening* of meshes, where small elements are no longer needed, will be an essential piece of an efficient algorithm. In the context of nonlinear parabolic PDEs two types of coarsening methods have evolved.

*A priori coarsening.* Here the mesh is coarsened before a timestep is computed, either globally by one mesh level as a whole or only locally where the error estimator in the previous timestep supplied particularly small error indicators $[\epsilon_T]$. The adaptive mesh refinement in the next timestep then generates a mesh corresponding to the required accuracy. In order to avoid losing information and thus accuracy by the coarsening, the initial value $u$ of the step on the finer mesh is kept stored, which, however, complicates the implementation significantly.

*A posteriori coarsening.* Here the tolerance is slightly reduced and the thusly gained scope afterwards used to coarsen the solution. In this coarsening the complete information is available, so that the thusly generated error can be exactly determined in advance and be used to control the coarsening process. In this variant the arising systems to be solved become slightly larger.

### 9.2.3 Goal-oriented Error Estimation

As in Section 6.1.5, for parabolic problems, goal-oriented error estimators for quantities of interest $J(u) = \langle j, u\rangle$ can also be expressed via weight functions $z$ that are computed as solutions of the adjoint problem. In contrast to elliptic equations, the information flow here is directed in time, which leads to an interesting structure of the goal-oriented error estimation.

For ease of a simpler derivation of the weight function, we consider the model problem

$$u' = \Delta u + f(u) + r \quad \text{on } \Omega \times [0,T], \qquad u|_{t=0} = u_0, \quad u|_{\partial\Omega} = 0 \tag{9.45}$$

in abstract notation $c(u,r) = 0$, where $r$ is again the residual. To each $r$ there belongs a solution $u(r)$ with $c(u(r),r) \equiv 0$. Taking the derivative w.r.t. $r$ leads to

$$c_u\big(u(r),r\big)u_r(r) + I = 0, \quad \text{i.e.,} \quad u_r = -c_u^{-1}.$$

Let $u = u(0)$ denote the exact solution and $u_h = u(r)$ an approximation. Then, in first order approximation,

$$\epsilon_h = \langle j, u_h - u \rangle \approx \langle j, u_r(r)\, r \rangle = \langle -c_u(u_h)^{-*} j, r \rangle.$$

Thus the weight function $z = -c_u(u_h)^{-*} j$ is again a solution of the adjoint problem

$$c_u(u_h)^* z = -j, \tag{9.46}$$

which can be written as a parabolic equation backward in time (see Exercise 9.5):

$$-z' = \Delta z + f'(u_h)z - j \quad \text{on } \Omega \times [0,T], \qquad z|_{t=T} = 0, \quad z|_{\partial\Omega} = 0. \tag{9.47}$$

In accordance with the sign of the time derivative, a final value is prescribed here (see Section 1.2). The goal-oriented error estimator is then defined in analogy to the elliptic case as

$$[\epsilon_h] = \langle z, r \rangle, \quad [\epsilon_T] = \langle z|_T, r \rangle. \tag{9.48}$$

For point evaluations $J(u) = u(\hat{x}, \hat{t})$ and a simple diffusion with $f'(u_h) = 0$ the weight function is just the fundamental solution (A.18), but in backward temporal direction. In particular we have $z(x,t) \equiv 0$ for $t > \hat{t}$ – in agreement with the causality interpretation that errors arising after $\hat{t}$ cannot have an influence on the value $u(\hat{x}, \hat{t})$.

Other than in the case of pure diffusion, in *nonlinear* parabolic equations the weight function $z$ corresponding to a point evaluation $J$ need not decay rapidly, as shown by the following example.

**Example 9.8.** Consider the equation

$$u_t = 10^{-3} u_{xx} + a f(u) \quad \text{on } \Omega \times [0,T] = ]0,1[ \times [0,10] \tag{9.49}$$

with $f(u) = u(u-0.1)(1-u)$ and initial value $u_0 = \max(0, 1-2x)$ to homogeneous Neumann boundary conditions. For $a = 0$ we obtain the linear diffusion equation familiar from Section 1.2. For $a = 10$, however, equation (9.49) is a simple model for excitable media. The reaction term $f(u)$ has two stable fixed points, 0 (the "state in rest") and 1 (the "excited state"), as well as an unstable fixed point 0.1. By diffusion, domains in rest are lifted by neighboring excited domains over the unstable fixed point and, due to the reaction, aspire to the excited state. In this way the excitation spreads through the domain in the form of a traveling wave – from left to right for this starting value. In an unbounded domain, solutions of the form $u(x,t) = w(x - vt)$ would come up, which will occur as well on bounded domains, as long as boundary effects do not play a dominant role. The *shape* of the solution is stable in the sense that small perturbations, in particular those of high frequencies, are rapidly damped. Not stable, however, is the *position* of the solution. Position errors remain permanent, since both $u(x,t) = w(x - vt)$ and the shifted variant $u(x,t) = w(x - vt + \epsilon)$ are solutions of (9.49). Whenever perturbations have an influence on the position of

the solution, then this is not damped. This is clearly recognized at the weight function $z$ in Figure 9.14, left, where residuals along the whole wave front in the space-time domain have an essentially constant influence on the point evaluation $u(0.9, T)$. The reason for this behavior is that for $0.05 \leq u \leq 0.68$ the reaction derivative $f'(u)$ is positive, which implies that the differential operator $10^{-3}\Delta + f'(u)$ in the adjoint equation (9.47) has positive eigenvalues.

In contrast to this behavior, the influence of perturbations in the case of pure diffusion (Figure 9.14, center) is restricted to spatially and temporally nearby points. This holds all the more for nonlinear problems if the solution to be evaluated is in a stable fixed point of the reaction (Figure 9.14, right), where perturbations are additionally damped.

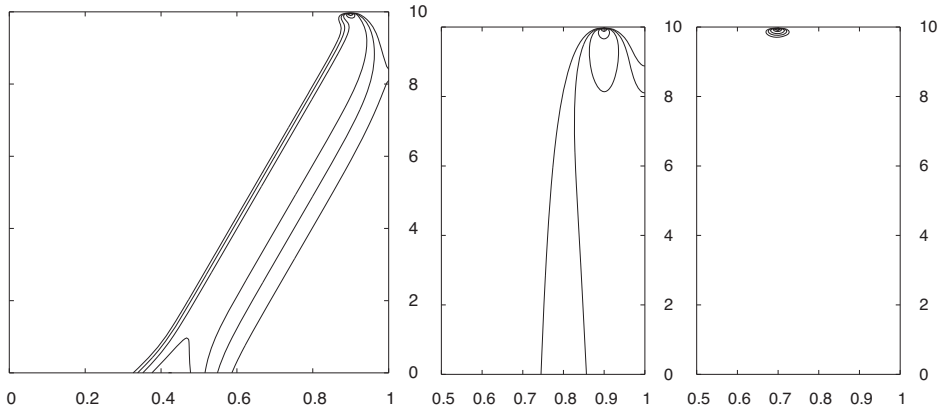


**Figure 9.14.** Isolines of the weight function $z(x, t)$ for the equation (9.49) for point evaluation. The heights of the isolines are 50, 100, 200, 500, 1000. *Left:* $a = 10$, $J = u(0.9, T)$. *Center:* $a = 0$, $J = u(0.9, T)$. *Right:* $a = 10$, $J = u(0.8, T)$.

As illustrated by this example, nonlinear parabolic problems often exhibit a significant global error transport. Unlike elliptic problems with strongly localized effects, here mesh refinement, due to a local residual or the locally estimated error, generally does not lead to meshes optimally adapted to the problem. Therefore, if we want to apply (9.48) for mesh refinement, we must compute the weight function $z$ from (9.47), where the solution $u_h$ in (9.47) and the corresponding residual $r$ in (9.48) enter. This implies that $u_h$ must be simultaneously available on the whole space-time domain $\Omega \times [0, T]$, which for $d = 3$ may involve an enormous amount of storage. In order to reduce this amount, several approaches like checkpointing [107] or state trajectory compression [214] have been developed.

## 9.3 Electrical Excitation of the Heart Muscle

In the last part of this chapter we present a parabolic problem from a medical application, the excitation of the excitation of the heart muscle. As will turn out, its complexity by far exceeds that of the simple model problems for illustration given so far. However, it nicely indicates how the adaptive algorithmic pieces suggested in this book affect the efficiency of numerical computation.

### 9.3.1 Mathematical Models

In order to pump enough blood through the body, a uniform coherent contraction of the heart muscle is required, which is imposed by an interaction of elastomechanical and electrical processes. Here we skip the elastomechanical part and only look at the electrocardiology.

*Physiological model.* While normal skeleton muscles are excited individually, the muscle cells of the heart are excited collectively. The electrical excitation runs through the muscle tissue in the form of a depolarization wave. In the course of depolarization the transmembrane voltage between the interior of the cell and its exterior changes from a rest value $-85$ mV to an excitation potential of 20 mV. This is followed by a discharge of $Ca^{2+}$, which causes a contraction of the muscle cells. In the healthy heartbeat this wave is triggered by some regularly repeated electrical impulse from the Sine node (see Figure 9.15, left). In some kinds of arrhythmia, out-of-phase impulses arise which totally change the spatio-temporal pattern (see Figure 9.15, right). The propagation of the depolarization front is based on an interplay of passive ion diffusion on the one hand and active ion exchange between cell interior and exterior on the other hand. Ion diffusion takes place both outside the cells and inside, as well as in between. The ion transport between the cells, i.e., through the cell membrane, runs through the ion channels. These may, depending on the transmembrane voltage and the interior states, be open or closed. By an action of the chemical substance adenosintriphosphat (ATP), ions can be transported even against the potential difference, and different kinds of ions can be exchanged.

To avoid a description of individual cells, *continuous* mathematical models are established. The essential quantities there are:

- the electric potentials $\phi_i$ in the cell interior and $\phi_e$ in the cell exterior;
- the transmembrane voltage $u = \phi_i - \phi_e$;
- the interior states $w$ of the ion channels.

**Membrane Models.** The ion current $I_{\text{ion}}$ through the channels as well as the temporal evolution $R$ of the internal states is described by one of the numerous membrane models (see, e.g., [161]). Physiologically-oriented membrane models contain descriptions of the dynamics of real ion channels as well as the concentrations of the various ions ($Na^+$, $K^+$, $Ca^{2+}$). The mathematical formulation of these connections leads to

a system of PDEs plus between one and 100 ODEs, which act at each point of the volume.

**Bidomain Model.** This model is the most elaborate one among the established electrocardiological models. It consists of two PDEs and a set of ODEs:

$$\chi C_m u' = \operatorname{div}(\sigma_i \nabla(u + \phi_e)) - \chi I_{\text{ion}}(u, w), \tag{9.50}$$

$$0 = \operatorname{div}(\sigma_i \nabla u) + \operatorname{div}((\sigma_i + \sigma_e)\nabla \phi_e) + I_e, \tag{9.51}$$

$$w' = R(u, w).$$

The physical quantities used above have the following meaning:

- $\chi$: membrane surface per volume;
- $C_m$: electrical capacity per membrane surface;
- $\sigma_i, \sigma_e \in \mathbb{R}^{d \times d}$: conductivities of the intra- and the extracellular medium, anisotropic because of the fiber structure of the heart muscle (and therefore tensors);
- $I_e$: external excitation current.

The elliptic equation (9.51) obviously has the function of an algebraic equality constraint, so that a differential-algebraic system of index 1 arises (cf. Volume 2, Section 2.6). In most cases one applies isolating boundary conditions

$$n^T \sigma_i \nabla u = -n^T \sigma_i \phi_e,$$
$$n^T \sigma_e \nabla \phi_e = 0.$$

Because of the high computational complexity of the bidomain model, simpler models are often applied (the limitations of which one should keep in mind).

**Monodomain Model.** If the diffusion tensors $\sigma_i$ and $\sigma_e$ are proportional to each other, then the second equation (9.51) may be inserted into (9.50), which produces the differential equation system:

$$\chi C_m u' = \operatorname{div}(\sigma_m \nabla u) - \chi I_{\text{ion}}(u, w),$$
$$w' = R(u, w).$$

Even though the assumption underlying this simplification does not hold in general, this model is taken as a good approximation for a number of questions; moreover, it is also popular due to its lower computational complexity.

### 9.3.2 Numerical Simulation

Changes in the normal excitation propagation due to genetic defects or local tissue damages as a consequence of infarcts, electric shocks, or drug abuse may perturb the

coherent contraction pattern and lead to severe diseases like cardiac insufficiency or tachycardia, as well as to fatal fibrillation. In the future, reliable numerical simulations of the excitation propagation will play an increasing role for the fundamental understanding of the functioning and the prediction of therapeutic effects.

The numerical solution of the above mathematical model equations is a real challenge (see also the monograph [196]). Main reasons for that are:

- *Strong spreading of spatial scales.* While a human heart has a diameter of about 10 cm, the width of a depolarization front lies around 1 mm. In order to obtain a roughly acceptable accuracy of the representation, the spatial resolution at the front position must be less than 0.5 mm.
- *Strong spreading of time scales.* A healthy heart beat takes about 1 s, whereas the transmembrane voltage changes from the rest value to the excitation potential within the order of 2 ms. For a roughly acceptable accuracy an equidistant timestep of 1 ms should not be exceeded.
- *Large systems.* Recent membrane models take into account various ions and their concentrations in several organelles within the cells as well as a large selection of ion channels. Such models may include up to 100 variables (in $w$) with rather different dynamics.
- *Complex geometry and anisotropy.* The lengthy form of the muscle cells and their arrangement in layers lead to a marked spatially dependent and rather diverse anisotropy of the conductivities $\sigma_i$ and $\sigma_e$. In addition, the geometry of the heart is rather complex: it is pervaded by blood vessels that do not propagate the excitation impulse; vice versa, the heart ventricles are pervaded by fiber-like structures that well conduct the depolarization front. Special conduction fibers (His-bundles and the Purkinje system) exhibit a much faster propagation of stimuli.

The numerical simulation of this obviously difficult problem has been approached in two stages. Since hitherto only uniform spatial meshes had been used, the effect of spatial and temporal adaptivity in the solution of the problem was first tested on a quadrilateral piece of tissue (see [53]). This gave rise to 200 adaptive timesteps vs. 10 000 equidistant timesteps that would have been necessary with constant timestep $\tau$. In the space meshes, a factor of approximately 200 could be saved. These results were seen as an encouragement for attacking the problem with realistic geometry (see [69]): the 3D coarse mesh of the heart muscle was taken from [142] consisting of 11 306 nodes for 56 581 tetrahedra. The simulation was done by the adaptive finite element code KARDOS due to J. Lang [139], which realizes an adaptive method of time layers. For time integration, the recently developed linearly implicit integrator ROS3PL [140] was used (see Section 9.1.2). For space discretization, linear finite elements were selected. The adaptive meshes were constructed by means of the hierarchical DLY-error estimator (see Section 6.1.4). The numerical solution of the arising huge systems of linear equations was performed by the iterative solver BI-CGSTAB from [207] with

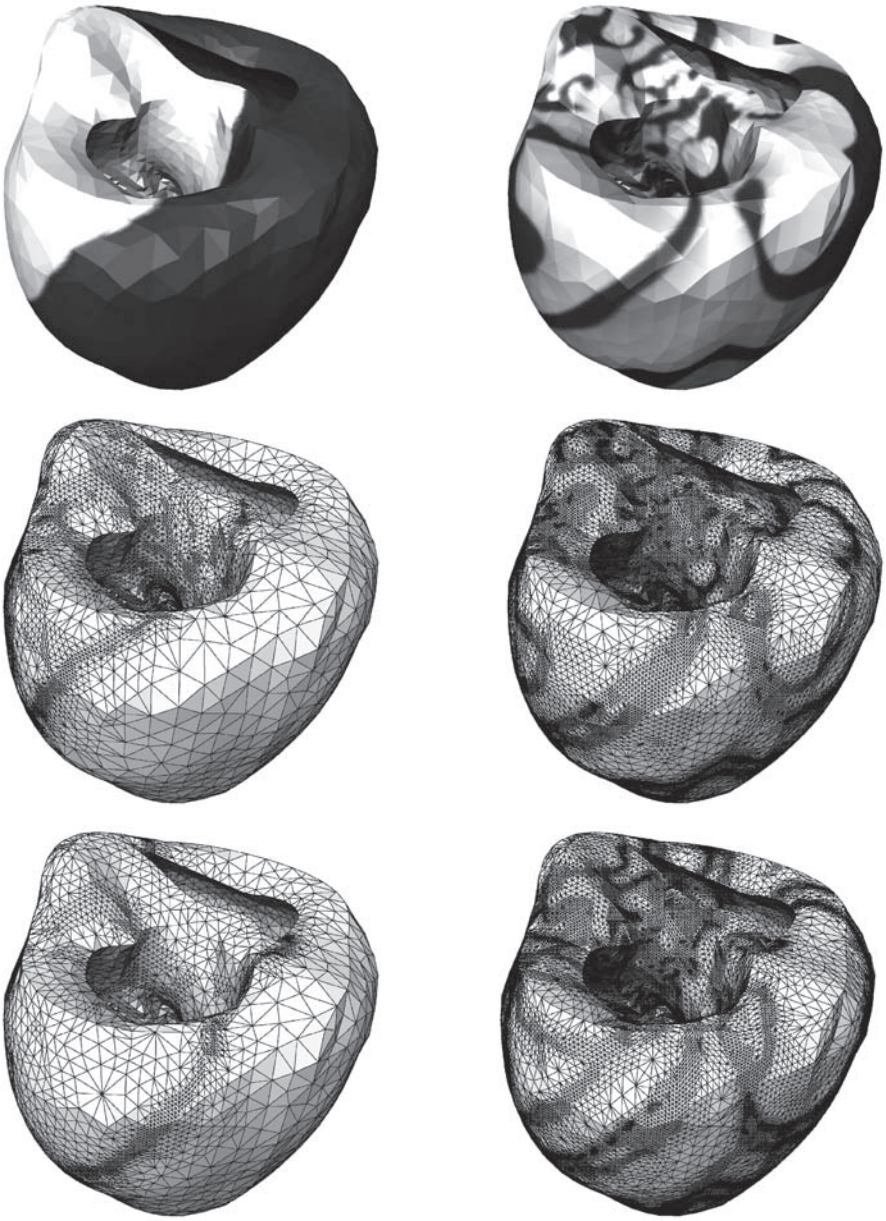

**Figure 9.15.** Adaptive method of time layers: Traveling depolarization fronts in the process of the electrical excitation of the heart. *Left*: normal heart beat: *top*: visualization of the solution at timepoint $t = 40$ ms, *center*: associated spatial mesh with front, *bottom*: spatial mesh at $t = 55$ ms. *Right*: Fibrillation: *top*: visualization of the solution at $t = 710$ ms, *center*: associated spatial mesh with multiple fronts, *bottom*: spatial mesh at $t = 725$ ms.

ILU-preconditioning[3] (see Remark 5.11). This preconditioning is an unsymmetric extension of the incomplete Cholesky decomposition presented in Section 5.3.2; the efficiency of ILU-preconditioners, however, has not yet been proven.

In Figure 9.15, typical transmembrane voltages from the monodomain equations with an Aliev–Panfilov membrane model are shown for one healthy heart beat (left) and for the case of cardiac fibrillation (right). The simple Aliev–Panfilov model was particularly developed for a mathematical description of cardiac arrhythmia. One may clearly observe the local refinement of the meshes along the fronts. When the fronts move through the domain, the position of the mesh refinement also moves. In the case of cardiac fibrillation, with mesh refinements at several parts of the domain, up to 2.1 million nodes were necessary at intermediate timepoints. An equidistant mesh with the same local resolution would have required around 310 million nodes and thus 150 times more degrees of freedom for spatial discretization.

Despite the significant reduction of the number of degrees of freedom by adaptive methods the simulation of the electrical excitation of the heart remains a real challenge. This comes not the least from the fact that the adaptive methods require an additional amount of computing compared to methods with fixed meshes: the computation of the error estimators, the mesh manipulation as such, and in particular the assembly of the mass and stiffness matrices, which need to be repeated after each mesh adaptation.

## 9.4 Exercises

**Exercise 9.1.** The concept of *A-stability* will be examined for the case of nondiagonalizable matrices. For this purpose we consider the linear ODE $y' = Ay$ and a corresponding one-step method $y_{k+1} = R(\tau A)y_k$ with stability function $R(z), z \in \mathbb{C}$. Let

$$|R(z)| \leq 1 \quad \text{for} \quad \Re(z) \leq 0.$$

Show that if the ODE is asymptotically stable, then the sequence $\{y_k\}$ is a null sequence.

*Hint*: Transform $A$ to Jordan canonical form; for the individual Jordan blocks, construct a similarity transformation depending on some sufficiently small parameter $\varepsilon$ such that the off-diagonal elements become arbitrarily small.

**Exercise 9.2.** Let $\mathcal{E}$ denote a first integral of the ODE $u' = f(u)$, i.e., let

$$\mathcal{E}\big(u(t)\big) = \mathcal{E}\big(u(0)\big).$$

With the notations $u_0 = u(0),\ u_k = u_\tau(k\tau),\ k = 0, 1, \ldots$ consider the following one-step methods:

[3] This may not be the last word on the subject, from the point of view of multigrid methods.

- implicit midpoint rule: $u_{k+1} = u_k + \tau f(\frac{1}{2}(u_k + u_{k+1}))$;
- implicit trapezoidal rule: $u_{k+1} = u_k + \frac{\tau}{2}(f(u_k) + f(u_{k+1}))$.

Show that:

1. for the invariant the following result must hold: $\operatorname{grad} \mathcal{E}(u(t))^T f(u(t)) = 0$;
2. the relation $\mathcal{E}(u_\tau(\tau)) = \mathcal{E}(u_0)$ holds for the implicit midpoint rule, but not for the implicit trapezoidal rule.

*Hint*: Introduce the quantity $g(\theta) := \mathcal{E}(u_\tau(\theta\tau))$.

**Exercise 9.3.** For $z = \mathrm{i}\,\sigma$ in the complex plane, represent the continuous solution of the Dahlquist test equation and the corresponding discrete solutions for the one-step methods EE, IE, and IMP.

**Exercise 9.4.** Consider a parabolic PDE in the abstract Cauchy form

$$u' = F(u) = -f_u(u), \quad u(0) = u_0,$$

where $f$ is a strictly convex function. Show that $f$ decreases monotonely along the trajectory $u(t)$.

**Exercise 9.5.** Show that, under the assumption of sufficient regularity, the solution $z$ of the problem (9.46) adjoint to (9.45) satisfies just the parabolic equation (9.47).

**Exercise 9.6.** Consider the case that in an adaptive method of lines in each timestep merely the *local* error is monitored by some prescribed tolerance TOL independent of $\tau$. Derive a decomposition of the time and the space part of the total error.