# 9

# Numerical approximation of boundary-value problems

Boundary-value problems are differential problems set in an interval $(a, b)$ of the real line or in an open multidimensional region $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ for which the value of the unknown solution (or its derivatives) is prescribed at the end-points $a$ and $b$ of the interval, or on the boundary $\partial\Omega$ of the multidimensional region.

In the multidimensional case the differential equation will involve *partial derivatives* of the exact solution with respect to the space coordinates. Equations depending also on time (denoted with $t$), like the heat equation and the wave equation, are called initial-boundary-value problems. In that case initial conditions at $t = 0$ need to be prescribed as well.

Some examples of boundary-value problems are reported below.

1. *Poisson equation*:

$$-u''(x) = f(x),\ x \in (a, b), \tag{9.1}$$

or (in several dimensions)

$$-\Delta u(\mathbf{x}) = f(\mathbf{x}),\ \mathbf{x} = (x_1, \ldots, x_d)^T \in \Omega, \tag{9.2}$$

where $f$ is a given function and $\Delta$ is the so-called *Laplace operator*:

$$\Delta u = \sum_{i=1}^{d} \frac{\partial^2 u}{\partial x_i^2}.$$

The symbol $\partial \cdot / \partial x_i$ denotes partial derivative with respect to the $x_i$ variable, that is, for every point $\mathbf{x}^0$

$$\frac{\partial u}{\partial x_i}(\mathbf{x}^0) = \lim_{h \to 0} \frac{u(\mathbf{x}^0 + h\mathbf{e}_i) - u(\mathbf{x}^0)}{h}, \tag{9.3}$$

where $\mathbf{e}_i$ is $i$th unitary vector of $\mathbb{R}^d$.

A. Quarteroni et al., *Scientific Computing with MATLAB and Octave*,
Texts in Computational Science and Engineering 2,
DOI 10.1007/978-3-642-45367-0_9, © Springer-Verlag Berlin Heidelberg 2014

2. *Heat equation*:

$$\frac{\partial u(x,t)}{\partial t} - \mu \frac{\partial^2 u(x,t)}{\partial x^2} = f(x,t),\ x \in (a,b),\ t > 0, \tag{9.4}$$

or (in several dimensions)

$$\frac{\partial u(\mathbf{x},t)}{\partial t} - \mu \Delta u(\mathbf{x},t) = f(\mathbf{x},t),\ \mathbf{x} \in \Omega,\ t > 0, \tag{9.5}$$

where $\mu > 0$ is a given coefficient representing the thermal diffusivity, and $f$ is again a given function.

3. *Wave equation*:

$$\frac{\partial^2 u(x,t)}{\partial t^2} - c \frac{\partial^2 u(x,t)}{\partial x^2} = 0,\ x \in (a,b),\ t > 0,$$

or (in several dimensions)

$$\frac{\partial^2 u(\mathbf{x},t)}{\partial t^2} - c \Delta u(\mathbf{x},t) = 0,\ \mathbf{x} \in \Omega,\ t > 0,$$

where $c$ is a given positive constant.

For a more complete description of general partial differential equations, the reader is referred for instance to [Eva98], [Sal08], and for their numerical approximation to [Qua13], [QV94], [EEHJ96] or [Lan03].

## 9.1 Some representative problems

**Problem 9.1 (Hydrogeology)** The study of filtration in groundwater can lead, in some cases, to an equation like (9.2). Consider a portion $\Omega$ occupied by a porous medium (like ground or clay). According to the Darcy law, the water velocity filtration $\mathbf{q} = (q_1, q_2, q_3)^T$ is equal to the variation of the water level $\phi$ in the medium, precisely

$$\mathbf{q} = -K \nabla \phi, \tag{9.6}$$

where $K$ is the constant hydraulic conductivity of the porous medium and $\nabla\phi$ denotes the spatial gradient of $\phi$. Assume that the fluid density is constant; then the mass conservation principle yields the equation $\mathrm{div}\mathbf{q} = 0$, where $\mathrm{div}\mathbf{q}$ is the *divergence* of the vector $\mathbf{q}$ and is defined as

$$\mathrm{div}\mathbf{q} = \sum_{i=1}^{3} \frac{\partial q_i}{\partial x_i}.$$

Thanks to (9.6) we therefore find that $\phi$ satisfies the Poisson problem $\Delta\phi = 0$ (see Exercise 9.8). ■

**Problem 9.2 (Thermodynamics)** Let $\Omega \subset \mathbb{R}^d$ be a volume occupied by a continuous medium. Denoting by $\mathbf{J}(\mathbf{x}, t)$ and $T(\mathbf{x}, t)$ the heat flux and the temperature of the medium, respectively, the Fourier law states that heat flux is proportional to the variation of the temperature $T$, that is

$$\mathbf{J}(\mathbf{x}, t) = -k\nabla T(\mathbf{x}, t),$$

where $k$ is a positive constant expressing the thermal conductivity coefficient. Imposing the conservation of energy, that is, the rate of change of energy of a volume equals the rate at which heat flows into it, we obtain the heat equation

$$\rho c \frac{\partial T}{\partial t} = k\Delta T, \tag{9.7}$$

where $\rho$ is the mass density of the continuous medium and $c$ is the specific heat capacity (per unit mass). If, in addition, heat is produced at the rate $f(\mathbf{x}, t)$ by some other means (e.g., electrical heating), (9.7) becomes

$$\rho c \frac{\partial T}{\partial t} = k\Delta T + f. \tag{9.8}$$

The coefficient $\mu = k/(\rho c)$ is the so-called *thermal diffusivity*. For the solution of this problem see Example 9.4. ■

**Problem 9.3 (Communications)** We consider a telegraph wire with resistance $R$ and self-inductance $L$ per unit length. Assuming that the current can drain away to ground through a capacitance $C$ and a conductance $G$ per unith length (see Figure 9.1), the equation for the voltage $v$ is

$$\frac{\partial^2 v}{\partial t^2} - c\frac{\partial^2 v}{\partial x^2} = -\alpha\frac{\partial v}{\partial t} - \beta v, \tag{9.9}$$

where $c = 1/(LC)$, $\alpha = R/L + G/C$ and $\beta = RG/(LC)$. Equation (9.9) is an example of a second order hyperbolic equation and it is known as *telegrapher's equation* (or just telegraph equation) (see [Str07]). The solution of this problem is given in Example 9.8. ■

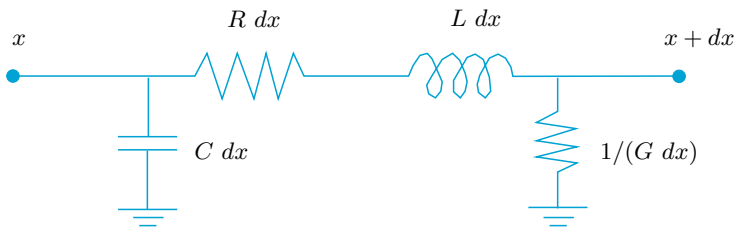


**Figure 9.1.** An element of cable of length $dx$

## 9.2 Approximation of boundary-value problems

The differential equations presented so far feature an infinite number of solutions. With the aim of obtaining a unique solution we must impose suitable conditions on the boundary $\partial\Omega$ of $\Omega$ and, for the time-dependent equations, suitable initial conditions at time $t = 0$.

In this section we consider the Poisson equations (9.1) or (9.2). In the one-dimensional case (9.1), to fix the solution one possibility is to prescribe the value of $u$ at $x = a$ and $x = b$, obtaining

$$\begin{aligned} &-u''(x) = f(x) \quad \text{for } x \in (a,b), \\ &u(a) = \alpha, \qquad u(b) = \beta \end{aligned} \tag{9.10}$$

where $\alpha$ and $\beta$ are two given real numbers. This is a *Dirichlet boundary-value problem*, and is precisely the problem that we will face in the next section.
Performing double integration it is easily seen that if $f \in C^0([a,b])$, the solution $u$ exists and is unique; moreover it belongs to $C^2([a,b])$.

Although (9.10) is an ordinary differential problem, it cannot be cast in the form of a Cauchy problem for ordinary differential equations since the value of $u$ is prescribed at two different points.

Instead to set Dirichlet boundary conditions $(9.10)_2$ we can impose $u'(a) = \gamma$, $u'(b) = \delta$ (where $\gamma$ and $\delta$ are suitable constants such that $\gamma - \delta = \int_a^b f(x)\mathrm{d}x$). A problem with these boundary conditions is named *Neumann* problem. Note that its solution is known up to an additive constant.

In the two-dimensional case, the Dirichlet boundary-value problem takes the following form: being given two functions $f = f(\mathbf{x})$ and $g = g(\mathbf{x})$, find a function $u = u(\mathbf{x})$ such that

$$\begin{aligned} &-\Delta u(\mathbf{x}) = f(\mathbf{x}) \qquad &&\text{for } \mathbf{x} \in \Omega, \\ &u(\mathbf{x}) = g(\mathbf{x}) \qquad &&\text{for } \mathbf{x} \in \partial\Omega \end{aligned} \tag{9.11}$$

Alternatively to the boundary condition on (9.11), we can prescribe a value for the partial derivative of $u$ with respect to the normal direction to the boundary $\partial\Omega$, that is

$$\frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}) = \nabla u(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = h(\mathbf{x}) \quad \text{for } \mathbf{x} \in \partial\Omega,$$

where $h$ is a suitable function such that $\displaystyle\int_{\partial\Omega} h = -\int_{\Omega} f$ (see Figure 9.2), in which case we will get a *Neumann boundary-value problem*.

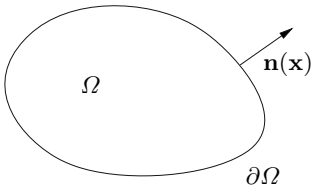


**Figure 9.2.** A two-dimensional domain $\Omega$ and the unit outward normal versor to $\partial\Omega$

It can be proven that if $f$ and $g$ are two continuous functions and the boundary $\partial\Omega$ of the region $\Omega$ is regular enough, then the Dirichlet boundary-value problem (9.11) has a unique solution (while the solution of the Neumann boundary-value problem is unique up to an additive constant).

The numerical methods which are used for its solution are based on the same principles used for the approximation of the one-dimensional boundary-value problem. This is the reason why in Sections 9.2.1 and 9.2.3 we will make a digression on the numerical solution of problem (9.10) with either finite difference and finite element methods, respectively.

With this aim we introduce on $[a, b]$ a partition into intervals $I_j = [x_j, x_{j+1}]$ for $j = 0, \ldots, N$ with $x_0 = a$ and $x_{N+1} = b$. We assume for simplicity that all intervals have the same length $h = (b - a)/(N + 1)$.

### 9.2.1 Finite difference approximation of the one-dimensional Poisson problem

The differential equation (9.10) must be satisfied in particular at any point $x_j$ (which we call *nodes* from now on) internal to $(a, b)$, that is

$$-u''(x_j) = f(x_j), \qquad j = 1, \ldots, N.$$

We can approximate this set of $N$ equations by replacing the second derivative with a suitable finite difference as we have done in Chapter 4 for the first derivatives. In particular, we observe that if $u : [a, b] \to \mathbb{R}$ is a sufficiently smooth function in a neighborhood of a generic point $\bar{x} \in (a, b)$, then the quantity

$$\delta^2 u(\bar{x}) = \frac{u(\bar{x} + h) - 2u(\bar{x}) + u(\bar{x} - h)}{h^2} \tag{9.12}$$

provides an approximation to $u''(\bar{x})$ of order 2 with respect to $h$ (see Exercise 9.3). This suggests the use of the following approximation to problem (9.10): find $\{u_j\}_{j=1}^N$ such that

$$-\frac{u_{j+1}-2u_j+u_{j-1}}{h^2}=f(x_j),\qquad j=1,\dots,N \tag{9.13}$$

with $u_0=\alpha$ and $u_{N+1}=\beta$. Obviously, $u_j$ will be an approximation of $u(x_j)$. Equations (9.13) provide a linear system

$$\mathrm{A}\mathbf{u}_h=h^2\mathbf{f}, \tag{9.14}$$

where $\mathbf{u}_h=(u_1,\dots,u_N)^T$ is the vector of unknowns, $\mathbf{f}=(f(x_1)+\alpha/h^2, f(x_2),\dots,f(x_{N-1}), f(x_N)+\beta/h^2)^T$, and A is the tridiagonal matrix

$$\mathrm{A}=\mathrm{tridiag}(-1,2,-1)=\begin{bmatrix} 2 & -1 & 0 & \dots & 0\\ -1 & 2 & \ddots & & \vdots\\ 0 & \ddots & \ddots & -1 & 0\\ \vdots & & -1 & 2 & -1\\ 0 & \dots & 0 & -1 & 2 \end{bmatrix}. \tag{9.15}$$

This system admits a unique solution since A is symmetric and positive definite (see Exercise 9.1). Moreover, it can be solved by the Thomas algorithm introduced in Section 5.6. We note however that, for small values of $h$ (and thus for large values of $N$), A is ill-conditioned. Indeed, $K(\mathrm{A})=\lambda_{max}(\mathrm{A})/\lambda_{min}(\mathrm{A})=Ch^{-2}$, for a suitable constant $C$ independent of $h$ (see Exercise 9.2). Consequently, the numerical solution of system (9.14), by either direct or iterative methods, requires special care. In particular, when using iterative methods a suitable preconditioner ought to be employed.

It is possible to prove (see, e.g., [QSS07, Chapter 12]) that if $f\in C^2([a,b])$ then

$$\max_{j=0,\dots,N+1}|u(x_j)-u_j|\le\frac{(b-a)^2h^2}{96}\max_{x\in[a,b]}|f''(x)| \tag{9.16}$$

that is, the finite difference method (9.13) converges with order two with respect to $h$.

In Program 9.1 we solve the following boundary-value problem (the so-called *diffusion-convection-reaction problem*)

$$\begin{cases} -\mu u''(x)+\eta u'(x)+\sigma u(x)=f(x) & \text{for } x\in(a,b),\\ u(a)=\alpha & u(b)=\beta, \end{cases} \tag{9.17}$$

$\mu>0$, $\eta$ and $\sigma>0$ constants, which is a generalization of problem (9.10).

For this problem the finite difference method, which generalizes (9.13), reads:

$$\begin{cases} -\mu\dfrac{u_{j+1}-2u_j+u_{j-1}}{h^2}+\eta\dfrac{u_{j+1}-u_{j-1}}{2h}+\sigma u_j=f(x_j), & j=1,\ldots,N,\\ u_0=\alpha, & u_{N+1}=\beta. \end{cases}$$

The input parameters of Program 9.1 are the end-points `a` and `b` of the interval, the number `N` of internal nodes, the constant coefficients $\mu, \eta$ and $\sigma$ and the function handle `bvpfun` associated with the function $f(x)$. Finally, `ua` and `ub` represent the values that the solution should attain at `x=a` and `x=b`, respectively. Output parameters are the vector of nodes `xh` and the computed solution `uh`. Notice that the solutions can be affected by spurious oscillations if $h \geq 2\mu/\eta$ (see next Section).

**Program 9.1. bvp**: approximation of a two-point diffusion-convection-reaction problem by the finite difference method

```
function [xh,uh]=bvp(a,b,N,mu,eta,sigma,bvpfun,...
                     ua,ub,varargin)
%BVP  Solves two-point boundary value problems.
%  [XH,UH]=BVP(A,B,N,MU,ETA,SIGMA,BVPFUN,UA,UB)
%  solves the boundary-value problem
%     -MU*D(DU/DX)/DX+ETA*DU/DX+SIGMA*U=BVPFUN
%  on the interval (A,B) with boundary conditions
%  U(A)=UA and U(B)=UB, by the centered finite
%  difference method at N equispaced nodes
%  internal to (A,B). BVPFUN is a function handle.
%  [XH,UH]=BVP(A,B,N,MU,ETA,SIGMA,BVPFUN,UA,UB,...
%  P1,P2,...) passes the additional parameters
%  P1, P2, ... to the function BVPFUN.
%  XH contains the nodes of the discretization,
%  including the boundary nodes.
%  UH contains the numerical solutions.
h = (b-a)/(N+1);
xh = (linspace(a,b,N+2))';
hm = mu/h^2;
hd = eta/(2*h);
e =ones(N,1);
A = spdiags([-hm*e-hd (2*hm+sigma)*e -hm*e+hd],...
    -1:1, N, N);
xi = xh(2:end-1);
f =bvpfun(xi,varargin{:});
f(1) =  f(1)+ua*(hm+hd);
f(end) = f(end)+ub*(hm-hd);
uh = A\f;
uh=[ua; uh; ub];
return
```

### 9.2.2 Finite difference approximation of a convection-dominated problem

We consider now the following generalization of the boundary-value problem (9.10)

$$\boxed{\begin{aligned} &-\mu u''(x) + \eta u'(x) = f(x) \quad \text{for } x \in (a,b), \\ &u(a) = \alpha, \qquad\qquad u(b) = \beta, \end{aligned}} \tag{9.18}$$

$\mu$ and $\eta$ being positive constants. This is the so-called *convection-diffusion problem* since the terms $-\mu u''(x)$ and $\eta u'(x)$ are responsible of diffusion and convection of the unknown function $u(x)$, respectively. The *global Péclet number*, associated to equation (9.18), is defined as

$$\mathbb{Pe}_{gl} = \frac{\eta(b-a)}{2\mu}, \tag{9.19}$$

and it provides a measure of how much the convective term prevails over the diffusive one. A problem featuring $\mathbb{Pe}_{gl} \gg 1$ will be named *convection-dominated problem*.

A possible discretization of (9.18) reads

$$\begin{cases} -\mu \dfrac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + \eta \dfrac{u_{j+1} - u_{j-1}}{2h} = f(x_j),\ j = 1, \ldots, N, \\ u_0 = \alpha, \quad u_{N+1} = \beta, \end{cases} \tag{9.20}$$

in which the centered finite difference scheme (4.9) has been used to approximate the convection term. As for the Poisson equation, one can prove that the error between the solution of the discrete problem (9.20) and that of the continuous problem (9.18) satisfies the following estimate

$$\max_{j=0,\ldots,N+1} |u(x_j) - u_j| \leq Ch^2 \max_{x \in [a,b]} |f''(x)|. \tag{9.21}$$

The constant $C$ is proportional to $\mathbb{Pe}_{gl}$, therefore it is very large when the convection dominates the diffusion. Thus, if the discretization step $h$ is not small enough, the numerical solution computed by the scheme (9.20) may be highly inaccurate and exhibit strong oscillations which are far from satisfying the continuous problem. For a more detailed analysis of this phenomenon we introduce the so-called *local Péclet number* (also named "grid" Péclet number)

$$\mathbb{Pe} = \frac{\eta h}{2\mu}. \tag{9.22}$$

One can prove that the solution of the discrete problem (9.20) does not exhibit oscillations if $\mathbb{Pe} < 1$ (see [Qua13, Chap. 5]). Thus, in order to

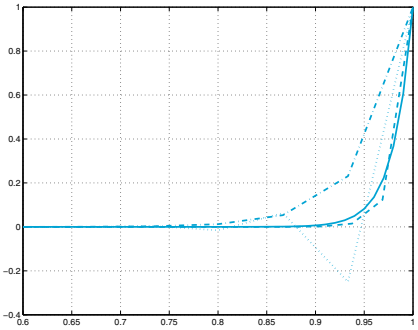


**Figure 9.3.** Exact solution (*solid line*), centered finite difference approximation with $h = 1/15$ ($\mathbb{P}\mathrm{e} > 1$) (*dotted line*), centered finite difference approximation with $h = 1/32$ ($\mathbb{P}\mathrm{e} < 1$) (*dashed line*), upwind finite difference approximation with $h = 1/15$ (*dashed-dotted line*) of the solution of problem (9.18) with $a = 0$, $b = 1$, $\alpha = 0$, $\beta = 1$, $f(x) = 0$, $\mu = 1/50$ and $\eta = 1$. For clearness, numerical solutions have been plotted on the interval $[0.6, 1]$ instead of $[0, 1]$

ensure a good numerical solution, we have to choose a discretization step $h < 2\mu/\eta$. Unfortunately, such a choice is not convenient when the ratio $2\mu/\eta$ is very small.

A possible alternative consists in choosing a different approximation of the convective term $u'$; precisely, instead to use the centered finite difference (4.9), we can employ the backward finite difference (4.8), so that the system (9.20) is replaced by

$$\begin{cases} -\mu\dfrac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + \eta\dfrac{u_j - u_{j-1}}{h} = f(x_j), & j = 1, \ldots, N, \\ u_0 = \alpha, & u_{N+1} = \beta, \end{cases} \tag{9.23}$$

which is known as *upwind* scheme. It is possible to prove that if (9.18) is approximated by (9.23), then the yielded numerical solution will not exhibit any oscillation, as the graphs reported in Figure 9.3 confirm.

### 9.2.3 Finite element approximation of the one-dimensional Poisson problem

The *finite element method* represents an alternative to the finite difference method for the approximation of boundary-value problems and is derived from a suitable reformulation of the differential problem (9.10).

Let us consider again (9.10) and multiply both sides of the differential equation by a generic function $v \in C^1([a, b])$. Integrating the corresponding equality on the interval $(a, b)$ and using integration by parts we obtain

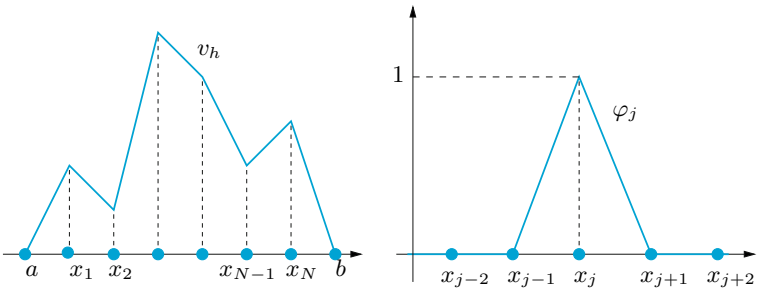


**Figure 9.4.** At left, a generic function $v_h \in V_h^0$. At right, the basis function of $V_h^0$ associated with the $j$th node

$$\int_a^b u'(x)v'(x)\ dx - [u'(x)v(x)]_a^b = \int_a^b f(x)v(x)\ dx.$$

By making the further assumption that $v$ vanishes at the end-points $x = a$ and $x = b$, problem (9.10) becomes: find $u \in C^1([a,b])$ such that $u(a) = \alpha$, $u(b) = \beta$ and

$$\int_a^b u'(x)v'(x)\ dx = \int_a^b f(x)v(x)\ dx \tag{9.24}$$

for each $v \in C^1([a,b])$ such that $v(a) = v(b) = 0$. This is called *weak formulation* of problem (9.10). (Indeed, both $u$ and the test function $v$ can be less regular than $C^1([a,b])$, see, e.g. [Qua13], [QSS07], [QV94].)

Its finite element approximation is defined as follows:

$$\begin{aligned}&\text{find } u_h \in V_h \text{ such that } u_h(a) = \alpha, u_h(b) = \beta \text{ and}\\ &\sum_{j=0}^{N} \int_{x_j}^{x_{j+1}} u_h'(x)v_h'(x)\ dx = \int_a^b f(x)v_h(x)\ dx, \qquad \forall v_h \in V_h^0\end{aligned} \tag{9.25}$$

where

$$V_h = \left\{ v_h \in C^0([a,b]) :\ v_{h|I_j} \in \mathbb{P}_1, j = 0, \ldots, N \right\}, \tag{9.26}$$

i.e. $V_h$ is the space of continuous functions on $[a,b]$ whose restrictions on every sub-interval $I_j$ are linear polynomials. Moreover, $V_h^0$ is the subspace of $V_h$ of those functions vanishing at the end-points $a$ and $b$. $V_h$ is called space of finite-elements of degree 1.

The functions in $V_h^0$ are piecewise linear polynomials (see Figure 9.4, left). In particular, every function $v_h$ of $V_h^0$ admits the representation

$$v_h(x) = \sum_{j=1}^{N} v_h(x_j)\varphi_j(x),$$

where for $j = 1, \ldots, N$,

$$\varphi_j(x) = \begin{cases} \dfrac{x - x_{j-1}}{x_j - x_{j-1}} & \text{if } x \in I_{j-1}, \\ \dfrac{x - x_{j+1}}{x_j - x_{j+1}} & \text{if } x \in I_j, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, $\varphi_j$ is null at every node $x_i$ except at $x_j$ where $\varphi_j(x_j) = 1$ (see Figure 9.4, right). The functions $\varphi_j$, $j = 1, \ldots, N$ are called *shape functions* and provide a basis for the vector space $V_h^0$.

Consequently, to fulfill (9.25) for any function in $V_h$ is equivalent to fulfill it only for the shape functions $\varphi_j$, $j = 1, \ldots, N$. By exploiting the fact that $\varphi_j$ vanishes outside the intervals $I_{j-1}$ and $I_j$, from (9.25) we obtain

$$\int_{I_{j-1} \cup I_j} u_h'(x)\varphi_j'(x)\, dx = \int_{I_{j-1} \cup I_j} f(x)\varphi_j(x)\, dx, \quad j = 1, \ldots, N. \tag{9.27}$$

On the other hand, we can write $u_h(x) = \sum_{j=1}^{N} u_j\varphi_j(x) + \alpha\varphi_0(x) + \beta\varphi_{N+1}(x)$, where $u_j = u_h(x_j)$, $\varphi_0(x) = (x_1 - x)/(x_1 - a)$ for $a \le x \le x_1$, and $\varphi_{N+1}(x) = (x - x_N)/(b - x_N)$ for $x_N \le x \le b$, while both $\varphi_0(x)$ and $\varphi_{N+1}(x)$ are zero otherwise. By substituting this expression in (9.27), we find:

$$u_1 \int_{I_0 \cup I_1} \varphi_1'(x)\varphi_1'(x)\, dx + u_2 \int_{I_1} \varphi_2'(x)\varphi_1'(x)\, dx = \int_{I_0 \cup I_1} f(x)\varphi_1(x)\, dx + \frac{\alpha}{x_1 - a},$$

$$u_{j-1} \int_{I_{j-1}} \varphi_{j-1}'(x)\varphi_j'(x)\, dx + u_j \int_{I_{j-1} \cup I_j} \varphi_j'(x)\varphi_j'(x)\, dx + u_{j+1} \int_{I_j} \varphi_{j+1}'(x)\varphi_j'(x)\, dx = \int_{I_{j-1} \cup I_j} f(x)\varphi_j(x)\, dx, \quad j = 2, \ldots, N-1,$$

$$u_{N-1} \int_{I_{N-1}} \varphi_{N-1}'(x)\varphi_N'(x)\, dx + u_N \int_{I_{N-1} \cup I_N} \varphi_N'(x)\varphi_N'(x)\, dx = \int_{I_{N-1} \cup I_N} f(x)\varphi_j(x)\, dx + \frac{\beta}{b - x_N}.$$

In the special case where all intervals have the same length $h$, then $\varphi'_{j-1} = -1/h$ in $I_{j-1}$, $\varphi'_j = 1/h$ in $I_{j-1}$ and $\varphi'_j = -1/h$ in $I_j$, $\varphi'_{j+1} = 1/h$ in $I_j$. Consequently, we obtain

$$\begin{aligned} 2u_1 - u_2 &= h \int_{I_0 \cup I_1} f(x)\varphi_1(x)\ dx + \alpha, \\ -u_{j-1} + 2u_j - u_{j+1} &= h \int_{I_{j-1} \cup I_j} f(x)\varphi_j(x)\ dx, \qquad j = 2, \ldots, N-1, \\ -u_{N-1} + 2u_N &= h \int_{I_{N-1} \cup I_N} f(x)\varphi_N(x)\ dx + \beta. \end{aligned}$$

The yielded linear system has unknowns $\{u_1, \ldots, u_N\}$ and shares the same matrix (9.15) as the finite difference system, however it has a different right-hand side (and a different solution too, in spite of coincidence of notation). Finite difference and finite element solutions share however the same accuracy with respect to $h$ when the nodal maximum error is computed.

We notice that 2nd-order convergence with respect to $h$ is guaranteed for finite difference approximation if $f \in C^2([a, b])$ (see (9.21)), while for finite elements it is sufficient that $f$ be a square-integrable function in $(a, b)$, i.e.,

$$\int_a^b f^2(x)dx < +\infty.$$

Obviously the finite element approach can be generalized to problems like (9.17) (also in the case when $\mu$, $\eta$ and $\sigma$ depend on $x$) and (9.18).

To approximate the convection-dominated problem (9.18), the upwind scheme used for finite differences can be reproduced also for finite-elements. More precisely, by noting that

$$\frac{u_i - u_{i-1}}{h} = \frac{u_{i+1} - u_{i-1}}{2h} - \frac{h}{2}\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2},$$

we can conclude that decentralizing finite differences is equivalent to perturb the centered incremental ratio by a term corresponding to a second-order derivative. This additional term can be interpreted as an *artificial viscosity*. In other words, using upwind with finite-elements is equivalent to solve, by the (centered) Galerkin method, the following perturbed problem

$$-\mu_h u''(x) + \eta u'(x) = f(x), \tag{9.28}$$

where $\mu_h = (1 + \mathbb{Pe})\mu$ is the augmented viscosity.

A further generalization of linear finite element methods consists of using piecewise polynomials of degree greater than 1, allowing the achievement of higher convergence orders. In these cases, the finite element matrix does not coincide anymore with that of finite differences.

See Exercises 9.1-9.7.

### 9.2.4 Finite difference approximation of the two-dimensional Poisson problem

Let us consider the Poisson problem (9.2), in a two-dimensional region $\Omega$.

The idea behind finite differences relies on approximating the partial derivatives that are present in the PDE again by incremental ratios computed on a suitable grid (called the computational grid) made of a finite number of nodes. Then the solution $u$ of the PDE will be approximated only at these nodes.

The first step therefore consists of introducing a computational grid. Assume for simplicity that $\Omega$ is the rectangle $(a, b) \times (c, d)$. Let us introduce a partition of $[a, b]$ in subintervals $(x_i, x_{i+1})$ for $i = 0, \ldots, N_x$, with $x_0 = a$ and $x_{N_x+1} = b$. Let us denote by $\Delta_x = \{x_0, \ldots, x_{N_x+1}\}$ the set of end-points of such intervals and by $h_x = \max_{i=0,\ldots,N_x} (x_{i+1} - x_i)$ their maximum length.

In a similar manner we introduce a discretization of the $y$-axis $\Delta_y = \{y_0, \ldots, y_{N_y+1}\}$ with $y_0 = c$, $y_{N_y+1} = d$ and $h_y = \max_{j=0,\ldots,N_y} (y_{j+1} - y_j)$. The cartesian product $\Delta_h = \Delta_x \times \Delta_y$ provides the computational grid on $\Omega$ (see Figure 9.5), and $h = \max\{h_x, h_y\}$ is a characteristic measure of the grid-size. We are looking for values $u_{i,j}$ which approximate $u(x_i, y_j)$. We will assume for the sake of simplicity that the nodes be uniformly spaced, that is, $x_i = x_0 + ih_x$ for $i = 0, \ldots, N_x + 1$ and $y_j = y_0 + jh_y$ for $j = 0, \ldots, N_y + 1$.

The second order partial derivatives of a function can be approximated by a suitable incremental ratio, as we did for ordinary derivatives. In the case of a function of two variables, we define the following incremental ratios:

$$\begin{aligned} \delta_x^2 u_{i,j} &= \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h_x^2}, \\ \delta_y^2 u_{i,j} &= \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h_y^2}. \end{aligned} \tag{9.29}$$

They are second order accurate with respect to $h_x$ and $h_y$, respectively, for the approximation of $\partial^2 u/\partial x^2$ and $\partial^2 u/\partial y^2$ at the node $(x_i, y_j)$. If we replace the second order partial derivatives of $u$ with the formula

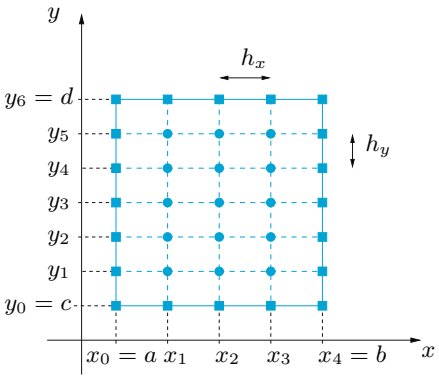


**Figure 9.5.** The computational grid $\Delta_h$ with only 15 internal nodes on a rectangular domain

(9.29), by requiring that the PDE is satisfied at all internal nodes of $\Delta_h$, we obtain the following set of equations:

$$-(\delta_x^2 u_{i,j} + \delta_y^2 u_{i,j}) = f_{i,j}, \quad i = 1, \ldots, N_x, \ j = 1, \ldots, N_y. \tag{9.30}$$

We have set $f_{i,j} = f(x_i, y_j)$. We must add the equations that enforce the Dirichlet data at the boundary, which are

$$u_{i,j} = g_{i,j} \quad \forall i, j \text{ such that } (x_i, y_j) \in \partial\Delta_h, \tag{9.31}$$

where $\partial\Delta_h$ indicates the set of nodes belonging to the boundary $\partial\Omega$ of $\Omega$. These nodes are indicated by small squares in Figure 9.5. If we make the further assumption that the computational grid is uniform in both cartesian directions, that is, $h_x = h_y = h$, instead of (9.30) we obtain

$$-\frac{1}{h^2}(u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i,j+1} + u_{i+1,j}) = f_{i,j}, \quad i = 1, \ldots, N_x, \ j = 1, \ldots, N_y \tag{9.32}$$

The system given by equations (9.32) (or (9.30)) and (9.31) allows the computation of the nodal values $u_{i,j}$ at all nodes of $\Delta_h$. For every fixed pair of indices $i$ and $j$, equation (9.32) involves five unknown nodal values as we can see in Figure 9.6. For that reason this finite difference scheme is called *the five-point scheme* for the Laplace operator. We note that the unknowns associated with the boundary nodes can be eliminated using (9.31) and therefore (9.30) (or (9.32)) involves only $N = N_x N_y$ unknowns.

The resulting system can be written in a more interesting form if we adopt the *lexicographic* order according to which the nodes (and,

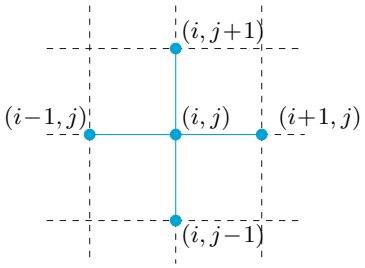


**Figure 9.6.** The stencil of the five point scheme for the Laplace operator

correspondingly, the unknown components) are numbered by proceeding from left to right and from the bottom to the top. By so doing, we obtain a system like (9.14), with a matrix $\mathrm{A} \in \mathbb{R}^{N\times N}$ which takes the following block tridiagonal form:

$$\mathrm{A} = \mathrm{tridiag}(\mathrm{D}, \mathrm{T}, \mathrm{D}). \tag{9.33}$$

There are $N_y$ rows and $N_y$ columns, and every entry (denoted by a capital letter) consists of a $N_x \times N_x$ matrix. In particular, $\mathrm{D} \in \mathbb{R}^{N_x\times N_x}$ is a diagonal matrix whose diagonal entries are $-1/h_y^2$, while $\mathrm{T} \in \mathbb{R}^{N_x\times N_x}$ is a symmetric tridiagonal matrix

$$\mathrm{T} = \mathrm{tridiag}(-\frac{1}{h_x^2}, \frac{2}{h_x^2} + \frac{2}{h_y^2}, -\frac{1}{h_x^2}).$$

A is symmetric since all diagonal blocks are symmetric. It is also positive definite, that is $\mathbf{v}^T\mathrm{A}\mathbf{v} > 0\ \forall \mathbf{v} \in \mathbb{R}^N$, $\mathbf{v} \neq \mathbf{0}$. Actually, by partitioning $\mathbf{v}$ in $N_y$ vectors $\mathbf{v}_k$ of length $N_x$ we obtain

$$\mathbf{v}^T\mathrm{A}\mathbf{v} = \sum_{k=1}^{N_y} \mathbf{v}_k^T\mathrm{T}\mathbf{v}_k - \frac{2}{h_y^2}\sum_{k=1}^{N_y-1} \mathbf{v}_k^T\mathbf{v}_{k+1}. \tag{9.34}$$

We can write $\mathrm{T} = 2/h_y^2\mathrm{I} + 1/h_x^2\mathrm{K}$ where K is the (symmetric and positive definite) matrix given in (9.15) and I is the identity matrix. Consequently, using the identity $2a(a-b) = a^2 - b^2 + (a-b)^2$ and some algebraic manipulation, (9.34) reads

$$\mathbf{v}^T\mathrm{A}\mathbf{v} = \frac{1}{h_x^2}\sum_{k=1}^{N_y-1} \mathbf{v}_k^T\mathrm{K}\mathbf{v}_k + \frac{1}{h_y^2}\left(\mathbf{v}_1^T\mathbf{v}_1 + \mathbf{v}_{N_y}^T\mathbf{v}_{N_y} + \sum_{k=1}^{N_y-1}(\mathbf{v}_k - \mathbf{v}_{k+1})^T(\mathbf{v}_k - \mathbf{v}_{k+1})\right),$$

which is a strictly positive real number since K is positive definite and at least one vector $\mathbf{v}_k$ is non-null.

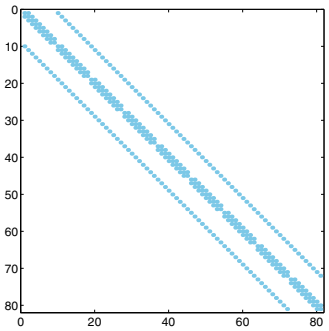


**Figure 9.7.** Pattern of the matrix associated with the five-point scheme using the lexicographic ordering of the unknowns

Having proven that A is non-singular we can conclude that the finite difference system admits a unique solution $\mathbf{u}_h$.

The matrix A is *sparse*; as such, it will be stored in the format `sparse` of MATLAB (see Section 5.3). In Figure 9.7 (obtained by using the command `spy(A)`) we report the structure of the matrix corresponding to a uniform grid of $11 \times 11$ nodes, after having eliminated the rows and columns associated to the nodes of $\partial\Delta_h$. It can be noted that the only nonzero elements lie on five diagonals.

Since A is symmetric and positive definite, the associated system can be solved efficiently by either direct or iterative methods, as illustrated in Chapter 5. Finally, it is worth pointing out that A shares with its one-dimensional analog the property of being ill-conditioned: indeed, its condition number grows like $h^{-2}$ as $h$ tends to zero.

In the Program 9.2 we construct and solve the system (9.30)-(9.31) (using the command \, see Section 5.8). The input parameters `a`, `b`, `c` and `d` denote the endpoints of the intervals generating the domain $\Omega = (a, b) \times (c, d)$, while `nx` and `ny` denote the values of $N_x$ and $N_y$ (the case $N_x \neq N_y$ is admitted). Finally, the two function handles `fun` and `bound` are associated with the right-hand side $f = f(x, y)$ (otherwise called the source term) and the Dirichlet boundary data $g = g(x, y)$, respectively. The output variable `uh` is a matrix whose $j, i$th entry is $u_{i,j}$, while `xh` and `yh` are vectors whose components are the nodes $x_i$ and $y_j$, respectively, all including the nodes of the boundary. The numerical solution can be visualized by the command `mesh(x,y,u)`. The (optional) input function `uex` stands for the exact solution of the original problem for those cases (of theoretical interest) where this solution is known. In such cases the output parameter `error` contains the nodal relative error between the exact and numerical solution, which is computed as follows:

mesh

$$\texttt{error} = \max_{i,j} |u(x_i, y_j) - u_{i,j}| / \max_{i,j} |u(x_i, y_j)|.$$

**Program 9.2. poissonfd**: approximation of the Poisson problem with Dirichlet boundary data by the five-point finite difference method

```
function [xh,yh,uh,error]=poissonfd(a,b,c,d,nx,ny,...
                        fun,bound,uex,varargin)
%POISSONFD two-dimensional Poisson solver
%  [XH,YH,UH]=POISSONFD(A,B,C,D,NX,NY,FUN,BOUND) solves
%  by the five-point finite difference scheme the
%  problem -LAPL(U) = FUN in the rectangle (A,B)X(C,D)
%  with Dirichlet boundary conditions U(X,Y)=BOUND(X,Y)
%  at any (X,Y) on the boundary of the rectangle.
%  [XH,YH,UH,ERROR]=POISSONFD(A,B,C,D,NX,NY,FUN,...
%  BOUND,UEX) computes also the maximum nodal error
%  ERROR with respect to the exact solution UEX.
%  FUN,BOUND and UEX are function handles.
%  [XH,YH,UH,ERROR]=POISSONFD(A,B,C,D,NX,NY,FUN,...
%  BOUND,UEX,P1,P2, ...) passes the optional arguments
%  P1,P2,... to the functions FUN,BOUND,UEX.
if nargin == 8
    uex = @(x,y)0+0*x+0*y;
end
nx1 = nx+2; ny1=ny+2; dim = nx1*ny1;
hx = (b-a)/(nx+1); hy = (d-c)/(ny+1);
    hx2 = hx^2;       hy2 = hy^2;
kii = 2/hx2+2/hy2; kix = -1/hx2;  kiy = -1/hy2;
K = speye(dim,dim);  rhs = zeros(dim,1);
y = c;
for m = 2:ny+1
 x = a; y = y + hy;
 for n = 2:nx+1
   i = n+(m-1)*nx1; x = x + hx;
   rhs(i) = fun(x,y,varargin{:});
   K(i,i) = kii; K(i,i-1) = kix; K(i,i+1) = kix;
   K(i,i+nx1) = kiy;   K(i,i-nx1) = kiy;
 end
end
rhs1 = zeros(dim,1); xh = [a:hx:b]'; yh = [c:hy:d];
rhs1(1:nx1) = bound(xh,c,varargin{:});
rhs1(dim-nx-1:dim) = bound(xh,d,varargin{:});
rhs1(1:nx1:dim-nx-1) = bound(a,yh,varargin{:});
rhs1(nx1:nx1:dim) = bound(b,yh,varargin{:});
rhs = rhs - K*rhs1;
nbound = [[1:nx1],[dim-nx-1:dim],[1:nx1:dim-nx-1],...
    [nx1:nx1:dim]];
ninternal = setdiff([1:dim],nbound);
K = K(ninternal,ninternal);
rhs = rhs(ninternal);
utemp = K\ rhs;
u = rhs1; u (ninternal) = utemp;
k = 1; y = c;
for j = 1:ny1
    x = a;
    for i = 1:nx1
        uh(j,i) = u(k);          k = k + 1;
        ue(j,i) = uex(x,y,varargin{:});
        x = x + hx;
    end
    y = y + hy;
```

```
end
if nargout == 4 & nargin >= 9
    error = max(max(abs(uh-ue)))/max(max(abs(ue)));
elseif nargout == 4 & nargin ==8
    warning('Exact solution not available');
    error = [ ];
end
end
```

**Example 9.1** The transverse displacement $u$ of an elastic membrane from the reference plane $z = 0$, under a load whose intensity is $f(x, y) = 8\pi^2 \sin(2\pi x)$ $\cos(2\pi y)$, satisfies a Poisson problem like (9.2) in the domain $\Omega = (0, 1)^2$. The Dirichlet value of the displacement is prescribed on $\partial\Omega$ as follows: $g = 0$ on the sides $x = 0$ and $x = 1$, and $g(x, 0) = g(x, 1) = \sin(2\pi x)$, $0 < x < 1$. This problem admits the exact solution $u(x, y) = \sin(2\pi x)\cos(2\pi y)$. In Figure 9.8 we show the numerical solution obtained by the five-point finite difference scheme on a uniform grid. Two different values of $h$ have been used: $h = 1/10$ (*left*) and $h = 1/20$ (*right*). When $h$ decreases the numerical solution improves, and actually the nodal relative error is 0.0292 for $h = 1/10$ and 0.0081 for $h = 1/20$. ■

Also the finite element method can be easily extended to the two-dimensional case. To this end the problem (9.2) must be reformulated in an integral form and the partition of the interval $(a, b)$ in one dimension must be replaced by a decomposition of $\Omega$ by polygons (typically, triangles) called *elements*. The generic shape function $\varphi_k$ will still be a continuous function, whose restriction on each element is a polynomial of degree 1 on each element, which is equal to 1 at the $k$th vertex (or node) of the triangulation and 0 at all other vertices. For its implementation one can use the MATLAB toolbox `pde`.

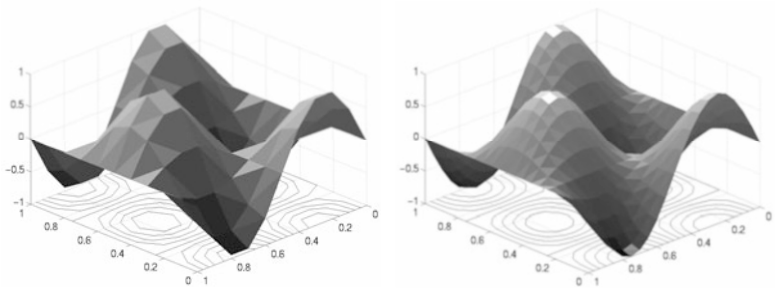

**Figure 9.8.** Transverse displacement of an elastic membrane computed on two uniform grids, coarser at left and finer at right. On the horizontal plane we report the isolines of the numerical solution. The triangular partition of $\Omega$ only serves the purpose of the visualization of the results

### 9.2.5 Consistency and convergence of finite difference discretization of the Poisson problem

In the previous section we have shown that the solution of the finite difference problem exists and is unique. Now we investigate the approximation error. We will assume for simplicity that $h_x = h_y = h$. If

$$\max_{i,j} |u(x_i, y_j) - u_{i,j}| \to 0 \quad \text{as } h \to 0 \tag{9.35}$$

the method used to compute $u_{i,j}$ is called convergent.

As we have already pointed out (see Remark 8.1), consistency is a necessary condition for convergence. A method is *consistent* if the residual, that is the error obtained when the exact solution is plugged into the numerical scheme, tends to zero when $h$ tends to zero. If we consider the five point finite difference scheme, at every internal node $(x_i, y_j)$ of $\Delta_h$ we define

$$\tau_h(x_i, y_j) = -f(x_i, y_j)$$

$$-\frac{1}{h^2}\left[u(x_{i-1}, y_j) + u(x_i, y_{j-1}) - 4u(x_i, y_j) + u(x_i, y_{j+1}) + u(x_{i+1}, y_j)\right].$$

This is the *local truncation error* at the node $(x_i, y_j)$. By (9.2) we obtain

$$\begin{aligned}\tau_h(x_i, y_j) = &\left\{\frac{\partial^2 u}{\partial x^2}(x_i, y_j) - \frac{u(x_{i-1}, y_j) - 2u(x_i, y_j) + u(x_{i+1}, y_j)}{h^2}\right\} \\ &+ \left\{\frac{\partial^2 u}{\partial y^2}(x_i, y_j) - \frac{u(x_i, y_{j-1}) - 2u(x_i, y_j) + u(x_i, y_{j+1})}{h^2}\right\}.\end{aligned}$$

Thanks to the analysis that was carried out in Section 9.2.4 we can conclude that both terms vanish as $h$ tends to 0. Thus

$$\lim_{h \to 0} \tau_h(x_i, y_j) = 0, \quad (x_i, y_j) \in \Delta_h \setminus \partial\Delta_h,$$

that is, the five-point method is consistent.

It is also convergent, as stated in the following Proposition (for its proof, see, e.g., [IK66]):

**Proposition 9.1** *Assume that the exact solution $u \in C^4(\bar{\Omega})$, i.e. all its partial derivatives up to the fourth order are continuous in the closed domain $\bar{\Omega}$. Then there exists a constant $C > 0$ such that*

$$\max_{i,j} |u(x_i, y_j) - u_{i,j}| \leq CMh^2 \tag{9.36}$$

*where $M$ is the maximum absolute value attained by the fourth order derivatives of $u$ in $\bar{\Omega}$.*

**Example 9.2** Let us experimentally verify that the five-point scheme applied to solve the Poisson problem of Example 9.1 converges with order two with respect to $h$. We start from $h = 1/4$ and, then we halve subsequently the value of $h$, until $h = 1/64$, through the following instructions:

```
a=0;b=1;c=0;d=1;
f=@(x,y) 8*pi^2*sin(2*pi*x).*cos(2*pi*y);

g=@(x,y) sin(2*pi*x).*cos(2*pi*y);
uex=g; nx=4; ny=4;
for n=1:5
  [xh,yh,uh,error(n)]=poissonfd(a,b,c,d,nx,ny,f,g,uex);
    nx = 2*nx; ny = 2*ny;
```

The vector containing the error is

```
format short e; error
   1.3565e-01   4.3393e-02   1.2308e-02   3.2775e-03   8.4557e-04
```

As we can verify using the following commands (see formula (1.12))

```
log(abs(error(1:end-1)./error(2:end)))/log(2)
  1.6443e+00   1.8179e+00   1.9089e+00   1.9546e+00
```

this error decreases as $h^2$ when $h \to 0$. ■

### 9.2.6 Finite difference approximation of the one-dimensional heat equation

We consider the one-dimensional heat equation (9.4) with homogeneous Dirichlet boundary conditions $u(a,t) = u(b,t) = 0$ for any $t > 0$ and initial condition $u(x,0) = u^0(x)$ for $x \in [a,b]$.

To solve this equation numerically we have to discretize both the $x$ and $t$ variables. We can start by dealing with the $x$-variable, following the same approach as in Section 9.2.1. We denote by $u_j(t)$ an approximation of $u(x_j,t)$, $j = 0,\ldots,N+1$, and approximate the Dirichlet problem (9.4) by the scheme: for all $t > 0$

$$\begin{cases} \dfrac{du_j}{dt}(t) - \dfrac{\mu}{h^2}(u_{j-1}(t) - 2u_j(t) + u_{j+1}(t)) = f_j(t), \quad j = 1,\ldots,N, \\ u_0(t) = u_{N+1}(t) = 0, \end{cases}$$

where $f_j(t) = f(x_j,t)$ and, for $t = 0$,

$$u_j(0) = u^0(x_j), \qquad j = 0,\ldots,N+1.$$

This is actually a *semi-discretization* of the heat equation, yielding a system of ordinary differential equations of the following form

$$\begin{cases} \dfrac{d\mathbf{u}}{dt}(t) = -\dfrac{\mu}{h^2}\mathrm{A}\mathbf{u}(t) + \mathbf{f}(t) \qquad \forall t > 0, \\ \mathbf{u}(0) = \mathbf{u}^0, \end{cases} \tag{9.37}$$

where $\mathbf{u}(t) = (u_1(t), \ldots, u_N(t))^T$ is the vector of unknowns, $\mathbf{f}(t) = (f_1(t), \ldots, f_N(t))^T$, $\mathbf{u}^0 = (u^0(x_1), \ldots, u^0(x_N))^T$, and A is the tridiagonal matrix introduced in (9.15). Note that for the derivation of (9.37) we have assumed that $u^0(x_0) = u^0(x_{N+1}) = 0$, which is coherent with the homogeneous Dirichlet boundary conditions.

A popular scheme for the integration in time of (9.37) is the so-called *θ−method*. Let $\Delta t > 0$ be a constant time-step, and denote by $v^k$ the value of a variable $v$ referred at the time level $t^k = k\Delta t$. Then the $\theta$-method reads

$$\boxed{\begin{aligned} &\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} = -\frac{\mu}{h^2}\mathrm{A}(\theta\mathbf{u}^{k+1} + (1-\theta)\mathbf{u}^k) + \theta\mathbf{f}^{k+1} + (1-\theta)\mathbf{f}^k, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k = 0, 1, \ldots \\ &\mathbf{u}^0 \text{ given} \end{aligned}} \tag{9.38}$$

or, equivalently,

$$\left(\mathrm{I} + \frac{\mu}{h^2}\theta\Delta t\mathrm{A}\right)\mathbf{u}^{k+1} = \left(\mathrm{I} - \frac{\mu}{h^2}\Delta t(1-\theta)\mathrm{A}\right)\mathbf{u}^k + \mathbf{g}^{k+1}, \tag{9.39}$$

where $\mathbf{g}^{k+1} = \Delta t(\theta\mathbf{f}^{k+1} + (1-\theta)\mathbf{f}^k)$ and I is the identity matrix of order $N$.

For suitable values of the parameter $\theta$, from (9.39) we can recover some familiar methods that have been introduced in Chapter 8. For example, if $\theta = 0$ the method (9.39) coincides with the forward Euler scheme and we can obtain $\mathbf{u}^{k+1}$ explicitly; otherwise, a linear system (with constant matrix $\mathrm{I} + \mu\theta\Delta t\mathrm{A}/h^2$) needs to be solved at each time level.

Regarding stability, when $f = 0$ the exact solution $u(x,t)$ tends to zero for every $x$ as $t \to \infty$. Then we would expect the discrete solution to have the same behavior, in which case we would call our scheme (9.39) *asymptotically stable*, this being coherent with the absolute stability concept defined in Section 8.6 for ordinary differential equations.

In order to study asymptotic stability, let us consider the equation (9.39) with $\mathbf{g}^{(k+1)} = \mathbf{0}\ \forall k \geq 0$.

If $\theta = 0$, it follows that

$$\mathbf{u}^k = (\mathrm{I} - \mu\Delta t\mathrm{A}/h^2)^k\mathbf{u}^0, \qquad k = 1, 2, \ldots$$

whence $\mathbf{u}^k \to \mathbf{0}$ as $k \to \infty$ iff

$$\rho(\mathrm{I} - \mu\Delta t\mathrm{A}/h^2) < 1. \tag{9.40}$$

On the other hand, the eigenvalues $\lambda_j$ of A are given by

$$\lambda_j = 2 - 2\cos(j\pi/(N+1)) = 4\sin^2(j\pi/(2(N+1))), \quad j = 1, \ldots, N$$

(see Exercise 9.2). Then (9.40) is satisfied if

$$\Delta t < \frac{1}{2\mu} h^2.$$

As expected, the forward Euler method is conditionally asymptotically stable, under the condition that the time-step $\Delta t$ should decay as the square of the grid spacing $h$.

In the case of the backward Euler method ($\theta = 1$), we would have from (9.39)

$$\mathbf{u}^k = \left[(\mathrm{I} + \mu \Delta t \mathrm{A}/h^2)^{-1}\right]^k \mathbf{u}^0, \qquad k = 1, 2, \ldots$$

Since all the eigenvalues of the matrix $(\mathrm{I} + \mu \Delta t \mathrm{A}/h^2)^{-1}$ are real, positive and strictly less than 1 for every value of $\Delta t$, this scheme is unconditionally asymptotically stable. More generally, the $\theta$-scheme is unconditionally asymptotically stable for all the values $1/2 \leq \theta \leq 1$, and conditionally asymptotically stable if $0 \leq \theta < 1/2$ (see, for instance, [QSS07, Chapter 13]).

As far as the accuracy of the $\theta$-method is concerned, its local truncation error behaves like $\Delta t + h^2$ if $\theta \neq \frac{1}{2}$ while it is of the order of $\Delta t^2 + h^2$ if $\theta = \frac{1}{2}$. The latter is the *Crank-Nicolson method* (see Section 8.4) and is therefore unconditionally asymptotically stable; the corresponding global (in both space and time) discretization scheme is second-order accurate with respect to both $\Delta t$ and $h$.

The same conclusions hold for the heat equation in a two-dimensional domain. In this case in the scheme (9.38) one must substitute to the matrix $\mathrm{A}/h^2$ the finite difference matrix defined in (9.33).

Program 9.3 solves numerically the heat equation on the time interval $(0, T)$ and on the domain $\Omega = (a, b)$ using the $\theta$-method. The input parameters are the vectors `xspan=[a,b]` and `tspan=[0,T]`, the number of discretization intervals in space (`nstep(1)`) and in time (`nstep(2)`), the scalar `mu` which contains the positive real coefficient $\mu$, the function handles `u0`, `fun` and `g` associated with the initial function $u^0(x)$, the right hand side $f(x, t)$ and the Dirichlet datum $g(x, t)$, respectively. Finally, the variable `theta` contains the coefficient $\theta$. The output variable `uh` contains the numerical solution at the final time $t = T$.

**Program 9.3. heattheta**: $\theta$-method for the one-dimensional heat equation

```
function [xh,uh]=heattheta(xspan,tspan,nstep,mu,...
                  u0,g,f,theta,varargin)
%HEATTHETA Solves the heat equation with the
% theta-method.
% [XH,UH]=HEATTHETA(XSPAN,TSPAN,NSTEP,MU,U0,G,F,THETA)
% solves the heat equation D U/DT - MU D^2U/DX^2 = F
% in (XSPAN(1),XSPAN(2)) X (TSPAN(1),TSPAN(2)) using
% the theta-method with initial condition U(X,0)=U0(X)
```

```
% and Dirichlet boundary conditions U(X,T)=G(X,T) at
% X=XSPAN(1) and X=XSPAN(2).
% MU is a positive constant, F=F(X,T), G=G(X,T) and
% U0=U0(X) are function handles.
% NSTEP(1) is the number of space integration intervals
% NSTEP(2) is the number of time-integration intervals
% XH contains the nodes of the discretization.
% UH contains the numerical solutions at time TSPAN(2).
% [XH,UH]=HEATTHETA(XSPAN,TSPAN,NSTEP,MU,U0,G,F,...
% THETA,P1,P2,...) passes the additional parameters
% P1,P2,...to the functions U0,G,F.

h  = (xspan(2)-xspan(1))/nstep(1);
dt = (tspan(2)-tspan(1))/nstep(2);
N = nstep(1)+1;
e = ones(N,1);
D = spdiags([-e 2*e -e],[-1,0,1],N,N);
I = speye(N);
A = I+mu*dt*theta*D/h^2;
An = I-mu*dt*(1-theta)*D/h^2;
A(1,:) = 0; A(1,1) = 1;
A(N,:) = 0; A(N,N) = 1;
xh = (linspace(xspan(1),xspan(2),N))';
fn = f(xh,tspan(1),varargin{:});
un = u0(xh,varargin{:});
[L,U]=lu(A);
for t = tspan(1)+dt:dt:tspan(2)
    fn1 = f(xh,t,varargin{:});
    rhs = An*un+dt*(theta*fn1+(1-theta)*fn);
    temp = g([xspan(1),xspan(2)],t,varargin{:});
    rhs([1,N]) = temp;
    uh = L\rhs; uh = U\uh; fn = fn1; un = uh;
end
return
```

**Example 9.3** We consider the heat equation (9.4) in $(a, b) = (0, 1)$ with $\mu = 1$, $f(x,t) = -\sin(x)\sin(t) + \sin(x)\cos(t)$, initial condition $u(x, 0) = \sin(x)$ and boundary conditions $u(0,t) = 0$ and $u(1,t) = \sin(1)\cos(t)$. In this case the exact solution is $u(x,t) = \sin(x)\cos(t)$. In Figure 9.9 we compare the behavior of the errors $\max_{i=0,\ldots,N} |u(x_i, 1) - u_i^M|$ with respect to the time-step on a uniform grid in space with $h = 0.002$. $\{u_i^M\}$ are the values of the finite difference solution computed at time $t^M = 1$. As expected, for $\theta = 0.5$ the $\theta$-method is second order accurate until when the time-step is so small that the spatial error dominates over the error due to the temporal discretization. ■

**Example 9.4 (Thermodynamics)** We consider a homogeneous, three meters long aluminium bar with uniform section. We are interested in simulating the evolution of the temperature in the bar starting from a suitable initial condition, by solving the heat equation (9.5). If we impose adiabatic conditions on the lateral surface of the bar (i.e. homogeneous Neumann conditions), and Dirichlet conditions at the end sections of the bar, the temperature only depends on the axial space variable (denoted by $x$). Thus the problem can be modeled by the one-dimensional heat equation (9.7) with $f = 0$, completed by the initial condition at $t = t_0$ and by Dirichlet boundary conditions at the endpoints of the reduced computational domain $\Omega = (0, L)$

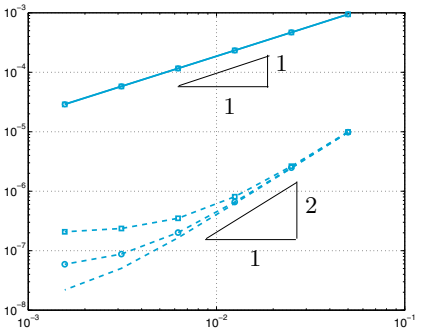


**Figure 9.9.** Error versus $\Delta t$ for the $\theta$-method (for $\theta = 1$, *solid line*, and $\theta = 0.5$ *dashed line*), for three different values of $h$: 0.008 (□), 0.004 (○) and 0.002 (*no symbols*)

($L = 3$m). Pure aluminium has thermal conductivity $k = 237$ W/(m K), density $\rho = 2700\text{kg/m}^3$ and specific heat capacity $c = 897$ J/(kg K), then its thermal diffusivity is $\mu = 9.786 \cdot 10^{-5}\text{m}^2/\text{s}$. Finally we consider the initial condition $T(x, 0) = 500$ K if $x \in (1, 2)$, 250 K otherwise and the Dirichlet boundary conditions $T(0, t) = T(3, t) = 250$ K. In Figure 9.10 we report the evolution of the temperature starting from the initial data, computed by the backward Euler method ($\theta = 1$, *left*) and by the Crank-Nicolson method ($\theta = 0.5$, *right*) (using Program 9.3).
The results show that when the time-step is large ($\Delta t = 20$sec), the Crank-Nicolson method is unstable because of the low smoothness of the initial datum (about this point, see also [QV94, Chapter 11]). On the contrary, the implicit Euler method provides a stable solution because it is more dissipative than Crank-Nicolson. Both methods compute a solution that decays to the correct value 250 K as $t \to \infty$. ■

### 9.2.7 Finite element approximation of the one-dimensional heat equation

The space discretization of the heat equation (9.4) with homogeneous Dirichlet boundary conditions $u(a, t) = u(b, t) = 0$, $\forall t > 0$ can be accomplished using the Galerkin finite element method by proceeding as we did in Section 9.2.3 for the Poisson equation. First, for all $t > 0$ we multiply (9.4) by a test function $v = v(x) \in C^1([a, b])$ and we integrate the resulting equation over $(a, b)$. For all $t > 0$ we therefore look for a function $t \to u(x, t) \in C^1([a, b])$ such that

$$\int_a^b \frac{\partial u}{\partial t}(x,t)v(x)\mathrm{d}x + \int_a^b \mu\frac{\partial u}{\partial x}(x,t)\frac{\mathrm{d}v}{\mathrm{d}x}(x)\mathrm{d}x = \tag{9.41}$$

$$= \int_a^b f(x)v(x)\mathrm{d}x \qquad \forall v \in C^1([a,b]),$$

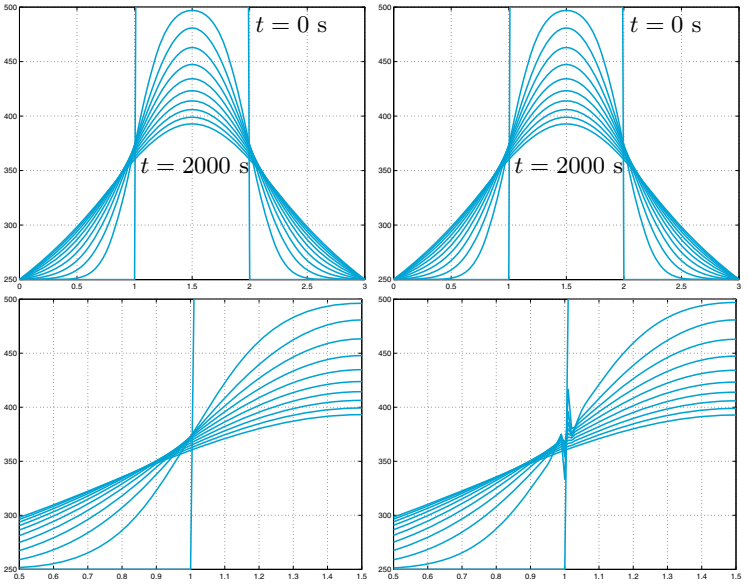


**Figure 9.10.** Temperature profiles in an aluminium bar at different time levels (from $t = 0$ to $t = 2000$ seconds with time-step $\Delta t$ of 0.25 seconds (*top*) and 20 seconds (*bottom*)), obtained using the backward Euler method (*left*) and the Crank-Nicolson method (*right*). In both cases, the space discretization is carried out by centered finite differences with steplength $h = 0.01$. The zoom on the solutions for $\Delta t = 20$sec (at bottom) shows instability of the Crank-Nicolson scheme

with $u(x, 0) = u^0(x)$. To simplify notations, from now on the dependence on variable $x$ in $u$, $v$ and $f$ will be understood.

Equation (9.41) keeps holding also for functions $v$ less regular than $C^1([a, b])$, e.g., like those of the space $V_h$ defined in (9.26). Therefore, we consider the following Galerkin formulation: $\forall t > 0$, find $u_h(t) \in V_h$ such that

$$\int_a^b \frac{\partial u_h}{\partial t}(t) v_h \mathrm{d}x + \int_a^b \mu \frac{\partial u_h}{\partial x}(t) \frac{\mathrm{d}v_h}{\mathrm{d}x} \mathrm{d}x = \int_a^b f(t) v_h \mathrm{d}x \quad \forall v_h \in V_h, \tag{9.42}$$

where $u_h(0) = u_h^0$ and $u_h^0 \in V_h$ is a convenient approximation of $u^0$. Formulation (9.42) is called *semi-discretization* of problem (9.41), since only the space discretization (not yet the time) was carried out.

For what concerns the finite element discretization of (9.42), let us consider the basis functions $\varphi_j$ introduced in Section 9.2.3. Then, the solution $u_h$ of (9.42) can be sought under the form

$$u_h(t) = \sum_{j=1}^{N} u_j(t) \varphi_j,$$

where $\{u_j(t)\}$ are the unknown coefficients and $N$ is the dimension of $V_h$.
Then, from (9.42) we obtain

$$\int_a^b \sum_{j=1}^N \frac{\mathrm{d}u_j}{\mathrm{d}t}(t)\varphi_j\varphi_i \mathrm{d}x + \mu \int_a^b \sum_{j=1}^N u_j(t) \frac{\mathrm{d}\varphi_j}{\mathrm{d}x}\frac{\mathrm{d}\varphi_i}{\mathrm{d}x}\mathrm{d}x =$$

$$= \int_a^b f(t)\varphi_i \mathrm{d}x, \qquad i = 1, \ldots, N$$

that is,

$$\sum_{j=1}^N \frac{\mathrm{d}u_j}{\mathrm{d}t}(t) \int_a^b \varphi_j\varphi_i \mathrm{d}x + \mu \sum_{j=1}^N u_j(t) \int_a^b \frac{\mathrm{d}\varphi_j}{\mathrm{d}x}\frac{\mathrm{d}\varphi_i}{\mathrm{d}x}\mathrm{d}x =$$

$$= \int_a^b f(t)\varphi_i \mathrm{d}x, \qquad i = 1, \ldots, N.$$

Using the same notations as in (9.37) we obtain

$$\mathrm{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}(t) + \mathrm{A_{fe}}\mathbf{u}(t) = \mathbf{f}_{\mathrm{fe}}(t), \tag{9.43}$$

where $(\mathrm{A_{fe}})_{ij} = \mu\int_a^b \frac{\mathrm{d}\varphi_j}{\mathrm{d}x}\frac{\mathrm{d}\varphi_i}{\mathrm{d}x}\mathrm{d}x$, $(\mathbf{f}_{\mathrm{fe}}(t))_i = \int_a^b f(t)\varphi_i \mathrm{d}x$ and $\mathrm{M}_{ij} = (\int_a^b \varphi_j\varphi_i \mathrm{d}x)$ for $i, j = 1, \ldots, N$. M is called the *mass matrix*. Since it is not singular, the system of ordinary differential equations (9.43) can be written in normal form as

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}(t) = -\mathrm{M}^{-1}\mathrm{A_{fe}}\mathbf{u}(t) + \mathrm{M}^{-1}\mathbf{f}_{\mathrm{fe}}(t). \tag{9.44}$$

To solve (9.43) approximately, we can still apply the $\theta$-method and obtain

$$\mathrm{M}\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} + \mathrm{A_{fe}}\left[\theta\mathbf{u}^{k+1} + (1-\theta)\mathbf{u}^k\right] = \theta\mathbf{f}_{\mathrm{fe}}^{k+1} + (1-\theta)\mathbf{f}_{\mathrm{fe}}^k. \tag{9.45}$$

As usual, the upper index $k$ means that the quantity at hand is computed at time $t^k = k\Delta t$, $\Delta t > 0$ being the time discretization step. As in the finite difference case, for $\theta = 0, 1$ and $1/2$, we respectively obtain the forward Euler, backward Euler and Crank-Nicolson methods, the latter being the only one which is second-order accurate with respect to $\Delta t$.
For each $k$, (9.45) is a linear system whose matrix is

$$\mathrm{K} = \frac{1}{\Delta t}\mathrm{M} + \theta\mathrm{A_{fe}}.$$

Since both matrices M and $\mathrm{A_{fe}}$ are symmetric and positive definite, the matrix K is also symmetric and positive definite. Moreover, K is independent of $k$ and then it can be factorized once at $t = 0$. For the

one-dimensional case that we are handling, this factorization is based on the Thomas method (see Section 5.6) and it requires a number of operation proportional to $N$. In the multidimensional case the use of the Cholesky factorization $\mathrm{K} = \mathrm{R}^T\mathrm{R}$, R being an upper triangular matrix (see (5.17)), will be more convenient. Consequently, at each time level the following two linear triangular systems, each of size equal to $N$, must be solved:

$$\begin{cases} \mathrm{R}^T\mathbf{y} = \left[\dfrac{1}{\Delta t}\mathrm{M} - (1-\theta)\mathrm{A}_{\mathrm{fe}}\right]\mathbf{u}^k + \theta\mathbf{f}_{\mathrm{fe}}^{k+1} + (1-\theta)\mathbf{f}_{\mathrm{fe}}^k, \\ \mathrm{R}\mathbf{u}^{k+1} = \mathbf{y}. \end{cases}$$

When $\theta = 0$, a suitable diagonalization of M would allow to decouple the system equations (9.45). The procedure is carried out by the so-called *mass-lumping* in which we approximate M by a non-singular diagonal matrix $\widetilde{\mathrm{M}}$. In the case of piecewise linear finite elements, $\widetilde{\mathrm{M}}$ can be obtained using the composite trapezoidal formula over the nodes $\{x_i\}$ to evaluate the integrals $\int_a^b \varphi_j\varphi_i \, \mathrm{d}x$, obtaining $\tilde{m}_{ij} = h\delta_{ij}$, $i, j = 1, \ldots, N$.

If $\theta \geq 1/2$, the $\theta$-method is unconditionally stable for every positive value of $\Delta t$, while if $0 \leq \theta < 1/2$ the $\theta$-method is stable only if

$$0 < \Delta t \leq \frac{2}{(1-2\theta)\lambda_{\max}(\mathrm{M}^{-1}\mathrm{A}_{\mathrm{fe}})},$$

to this aim see [Qua13, Chap. 5]. Moreover, it is possible to prove that there exist two positive constants $c_1$ and $c_2$, independent of $h$, such that

$$c_1 h^{-2} \leq \lambda_{\max}(\mathrm{M}^{-1}\mathrm{A}_{\mathrm{fe}}) \leq c_2 h^{-2}$$

(see [QV94, Section 6.3.2] for a proof). Thanks to this property, if $0 \leq \theta < 1/2$ the method is stable only if

$$0 < \Delta t \leq C_1(\theta)h^2, \tag{9.46}$$

where $C_1(\theta)$ is a suitable constant independent of both discretization parameters $h$ and $\Delta t$.

## 9.3 Hyperbolic equations: a scalar pure advection problem

Let us consider the following scalar hyperbolic problem

$$\begin{cases} \dfrac{\partial u}{\partial t} + a\dfrac{\partial u}{\partial x} = 0, & x \in \mathbb{R},\ t > 0, \\ u(x,0) = u^0(x), & x \in \mathbb{R}, \end{cases} \tag{9.47}$$

where $a$ is a positive real number. Its solution is given by

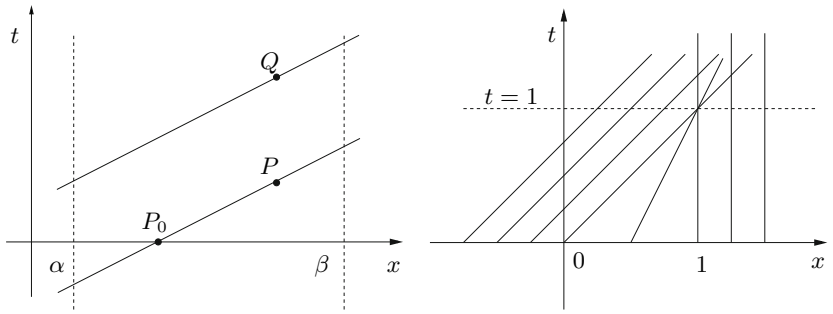


**Figure 9.11.** At left: examples of characteristics which are straight lines issuing from the points $P$ and $Q$. At right: characteristic straight lines for the Burgers equation (9.51)

$$u(x,t) = u^0(x - at),\ t \geq 0,$$

and represents a wave travelling with velocity $a$. The curves $(x(t), t)$ in the plain $(x, t)$, that satisfy the following scalar ordinary differential equation

$$\begin{cases} \dfrac{dx}{dt}(t) = a, & t > 0, \\ x(0) = x_0, \end{cases} \tag{9.48}$$

are called *characteristic curves* (or, simply, *characteristics*), and are the straight lines $x(t) = x_0 + at$, $t > 0$. The solution of (9.47) remains constant along them since

$$\frac{du}{dt} = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}\frac{dx}{dt} = 0 \qquad \text{on } (x(t), t).$$

For the more general problem

$$\begin{cases} \dfrac{\partial u}{\partial t} + a\dfrac{\partial u}{\partial x} + a_0 u = f, & x \in \mathbb{R}, \quad t > 0, \\ u(x,0) = u^0(x), & x \in \mathbb{R}, \end{cases} \tag{9.49}$$

where $a$, $a_0$ and $f$ are given functions of the variables $(x, t)$, the characteristic curves are still defined as in (9.48). In this case, the solutions of (9.49) satisfy along the characteristics the following differential equation

$$\frac{du}{dt} = f - a_0 u \qquad \text{on } (x(t), t).$$

Let us now consider problem (9.47) on a bounded interval $[\alpha, \beta]$

$$\begin{cases} \dfrac{\partial u}{\partial t} + a\dfrac{\partial u}{\partial x} = 0, & x \in (\alpha, \beta),\ t > 0, \\ u(x,0) = u^0(x), & x \in (\alpha, \beta). \end{cases} \tag{9.50}$$

Let us start with $a > 0$. Since $u$ is constant along the characteristics, from Figure 9.11, left, we deduce that the value of the solution at $P$ attains the value of $u^0$ at $P_0$, the foot of the characteristic issuing from $P$. On the other hand, the characteristic issuing from $Q$ intersects the straight line $x(t) = \alpha$ at a certain time $t = \bar{t} > 0$. Thus, the point $x = \alpha$ is an *inflow* point and it is necessary to assign there a boundary value for $u$, for every $t > 0$. Notice that if $a < 0$ then the inflow point is $x = \beta$ and it is necessary to assign there a boundary value for $u$, for every $t > 0$.

Referring to problem (9.47) it is worth noting that if $u^0$ is discontinuous at a point $x_0$, then such a discontinuity propagates along the characteristics issuing from $x_0$. This process can be made rigorous by introducing the concept of *weak solutions* of hyperbolic problems, see e.g. [GR96]. Another reason for introducing weak solutions is that in the case of nonlinear hyperbolic problems the characteristic lines can intersect: in this case the solution cannot be continuous and no classical solution does exist.

**Example 9.5 (Burgers equation)** Let us consider the Burgers equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0, \qquad x \in \mathbb{R}, \quad t > 0, \tag{9.51}$$

which is perhaps the simplest nontrivial example of a nonlinear hyperbolic equation. Taking as initial condition

$$u(x,0) = u^0(x) = \begin{cases} 1, & x \le 0, \\ 1 - x, & 0 < x \le 1, \\ 0, & x > 1, \end{cases}$$

the characteristic line issuing from the point $(x_0, 0)$ is given by

$$x(t) = x_0 + tu^0(x_0) = \begin{cases} x_0 + t, & x_0 \le 0, \\ x_0 + t(1 - x_0), & 0 < x_0 \le 1, \\ x_0, & x_0 > 1. \end{cases}$$

Notice that the characteristic lines do not intersect only if $t < 1$ (see Figure 9.11, right). ■

### 9.3.1 Finite difference discretization of the scalar transport equation

The half-plane $\{(x,t) :\ -\infty < x < \infty,\ t > 0\}$ is discretized by choosing a spatial grid size $\Delta x > 0$ (the parameter named $h$ until now), a temporal step $\Delta t > 0$ and the grid points $(x_j, t^n)$ as follows

$$x_j = j\Delta x, \qquad j \in \mathbb{Z}, \qquad t^n = n\Delta t, \qquad n \in \mathbb{N}.$$

Let us set

$$\lambda = \Delta t/\Delta x,$$

and define $x_{j+1/2} = x_j + \Delta x/2$. We look for discrete solutions $u_j^n$ which approximate the values $u(x_j, t^n)$ of the exact solution for any $j$, $n$. Quite often, explicit methods are employed for advancing in time hyperbolic initial-value problems.
Any explicit finite-difference method can be written in the form

$$u_j^{n+1} = u_j^n - \lambda(h_{j+1/2}^n - h_{j-1/2}^n), \tag{9.52}$$

where $h_{j+1/2}^n = h(u_j^n, u_{j+1}^n)$ for every $j$ and $h(\cdot,\cdot)$ is a function, to be properly chosen, that is called the *numerical flux*.

In what follows we will illustrate several instances of explicit methods for the approximation of problem (9.47):

1. *forward Euler/centered*

$$u_j^{n+1} = u_j^n - \frac{\lambda}{2}a(u_{j+1}^n - u_{j-1}^n), \tag{9.53}$$

which can be cast in the form (9.52) by setting

$$h_{j+1/2}^n = \frac{1}{2}a(u_{j+1}^n + u_j^n); \tag{9.54}$$

2. *Lax-Friedrichs*

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{\lambda}{2}a(u_{j+1}^n - u_{j-1}^n), \tag{9.55}$$

which is of the form (9.52) with

$$h_{j+1/2}^n = \frac{1}{2}[a(u_{j+1}^n + u_j^n) - \lambda^{-1}(u_{j+1}^n - u_j^n)]; \tag{9.56}$$

3. *Lax-Wendroff*

$$u_j^{n+1} = u_j^n - \frac{\lambda}{2}a(u_{j+1}^n - u_{j-1}^n) + \frac{\lambda^2}{2}a^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n), \tag{9.57}$$

which can be written in the form (9.52) provided that

$$h_{j+1/2}^n = \frac{1}{2}[a(u_{j+1}^n + u_j^n) - \lambda a^2(u_{j+1}^n - u_j^n)]; \tag{9.58}$$

4. *Upwind (or forward Euler/decentered)*

$$u_j^{n+1} = u_j^n - \frac{\lambda}{2}a(u_{j+1}^n - u_{j-1}^n) + \frac{\lambda}{2}|a|(u_{j+1}^n - 2u_j^n + u_{j-1}^n), \tag{9.59}$$

which fits the form (9.52) when the numerical flux is defined to be

$$h_{j+1/2}^n = \frac{1}{2}[a(u_{j+1}^n + u_j^n) - |a|(u_{j+1}^n - u_j^n)]. \tag{9.60}$$

**Table 9.1.** Artificial viscosity, artificial diffusion flux, and truncation error for Lax-Friedrichs, Lax-Wendroff and upwind methods

| method | $k$ | $h^{diff}_{j+1/2}$ | $\tau(\Delta t, \Delta x)$ |
|---|---|---|---|
| Lax-Friedrichs | $\Delta x^2$ | $-\frac{1}{2\lambda}(u_{j+1} - u_j)$ | $\mathcal{O}\left(\Delta x^2/\Delta t + \Delta t + \Delta x^2\right)$ |
| Lax-Wendroff | $a^2\Delta t^2$ | $-\frac{\lambda a^2}{2}(u_{j+1} - u_j)$ | $\mathcal{O}\left(\Delta t^2 + \Delta x^2 + \Delta t \Delta x^2\right)$ |
| upwind | $\lvert a\rvert \Delta x \Delta t$ | $-\frac{\lvert a\rvert}{2}(u_{j+1} - u_j)$ | $\mathcal{O}(\Delta t + \Delta x)$ |

Each one of the last three methods can be obtained from the forward Euler/centered method by adding a term proportional to the centered finite difference (4.9), so that they can be written in the equivalent form

$$u_j^{n+1} = u_j^n - \frac{\lambda}{2}a(u_{j+1}^n - u_{j-1}^n) + \frac{1}{2}k\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}. \tag{9.61}$$

The last term represents indeed a discretization of the second-order derivative

$$\frac{k}{2}\frac{\partial^2 u}{\partial x^2}(x_j, t^n).$$

The coefficient $k > 0$ plays the role of artificial viscosity. Its expression is given for the three previous cases in Table 9.1. Consequently, the numerical flux for each scheme can be equivalently written as

$$h_{j+1/2} = h^{FE}_{j+1/2} + h^{diff}_{j+1/2},$$

where $h^{FE}_{j+1/2}$ is the numerical flux of the forward Euler/centered scheme (which is given in (9.54)) and the *artificial diffusion flux* $h^{diff}_{j+1/2}$ for the three cases is also reported in Table 9.1.
The most classical implicit method is the *backward Euler/centered* scheme

$$u_j^{n+1} + \frac{\lambda}{2}a(u_{j+1}^{n+1} - u_{j-1}^{n+1}) = u_j^n. \tag{9.62}$$

It can still be written in the form (9.52) provided that $h^n$ is replaced by $h^{n+1}$. In the example at hand, the numerical flux is the same as for the forward Euler/centered method.

### 9.3.2 Finite difference analysis for the scalar transport equation

The convergence analysis of finite difference methods introduced in the previous Section requires that both consistency and stability hold.
Consider for instance, the forward Euler/centered method (9.53). As

done in Section 8.3.1, denoting by $u$ the exact solution of problem (9.47), the *local truncation error* at $(x_j, t^n)$ represents, up to a factor $1/\Delta t$, the error that would be generated by forcing the exact solution to satisfy that specific numerical scheme. In particular for the forward Euler/centered method it is defined as follows

$$\tau_j^n = \frac{u(x_j, t^{n+1}) - u(x_j, t^n)}{\Delta t} + a\frac{u(x_{j+1}, t^n) - u(x_{j-1}, t^n)}{2\Delta x},$$

while the *(global) truncation error* is defined as

$$\tau(\Delta t, \Delta x) = \max_{j,n} |\tau_j^n|.$$

When $\tau(\Delta t, \Delta x)$ goes to zero as $\Delta t$ and $\Delta x$ tend to zero independently, the numerical scheme is said to be *consistent*.

More in general, we say that a numerical method is of *order* $p$ in time and of *order* $q$ in space (for suitable positive values $p$ and $q$) if, for a sufficiently smooth solution of the exact problem,

$$\tau(\Delta t, \Delta x) = \mathcal{O}(\Delta t^p + \Delta x^q).$$

Finally, we say that a numerical scheme is *convergent* (in the maximum norm) if

$$\lim_{\Delta t, \Delta x \to 0} \max_{j,n} |u(x_j, t^n) - u_j^n| = 0.$$

If the exact solution is regular enough, using Taylor's expansion conveniently, we can characterize the truncation error of the methods previously introduced. For the forward (or backward) Euler/centered method it is $\mathcal{O}(\Delta t + \Delta x^2)$. For the other methods, see Table 9.1.

As of stability, we say that a numerical scheme for the approximation of a hyperbolic (either linear or nonlinear) problem is *stable* if, for any time $T$, there exist two constants $C_T > 0$ (possibily depending on $T$) and $\delta_0 > 0$, such that

$$\|\mathbf{u}^n\|_\Delta \leq C_T \|\mathbf{u}^0\|_\Delta, \tag{9.63}$$

for any $n$ such that $n\Delta t \leq T$ and for any $\Delta t$, $\Delta x$ such that $0 < \Delta t \leq \delta_0$, $0 < \Delta x \leq \delta_0$. The symbol $\|\cdot\|_\Delta$ stands for a suitable discrete norm, there are three instances:

$$\|\mathbf{v}\|_{\Delta,p} = \left(\Delta x \sum_{j=-\infty}^{\infty} |v_j|^p\right)^{\frac{1}{p}} \quad \text{for } p = 1, 2, \quad \|\mathbf{v}\|_{\Delta,\infty} = \sup_j |v_j|. \tag{9.64}$$

Courant, Friedrichs and Lewy [CFL28] have proved that a necessary and sufficient condition for any explicit scheme of the form (9.52) to be stable

is that the time and space discretization steps must obey the following condition

$$|a\lambda| \leq 1, \text{ i.e. } \Delta t \leq \frac{\Delta x}{|a|} \tag{9.65}$$

which is known as the *CFL condition*. The adimensional number $a\lambda$ ($a$ is a velocity) is commonly referred to as the *CFL number*. If $a$ is not constant the CFL condition becomes

$$\Delta t \leq \frac{\Delta x}{\sup\limits_{x\in\mathbb{R},\ t>0} |a(x,t)|}.$$

It is possible to prove that

1. the *forward Euler/centered* method (9.53) is unconditionally unstable, i.e. it is unstable for any possible choice of $\Delta x > 0$ and $\Delta t > 0$;
2. the *upwind* method (also called *forward Euler/decentered* method) (9.59) is conditionally stable with respect to the $\|\cdot\|_{\Delta,1}$ norm, i.e.

   $$\|\mathbf{u}^n\|_{\Delta,1} \leq \|\mathbf{u}^0\|_{\Delta,1} \qquad \forall n \geq 0,$$

   provided that the CFL condition (9.65) is satisfied; the same result can be proved also for both *Lax-Friedrichs* (9.55) and *Lax-Wendroff* (9.57) schemes;
3. the *backward Euler/centered* method (9.62) is unconditionally stable with respect to the $\|\cdot\|_{\Delta,2}$ norm, i.e., for any $\Delta t > 0$

   $$\|\mathbf{u}^n\|_{\Delta,2} \leq \|\mathbf{u}^0\|_{\Delta,2} \qquad \forall n \geq 0.$$

   See Exercise 9.11.

For a proof of the these results see, e.g., [QSS07, Chap. 13] and [Qua13, Chap. 12].

We want now to mention two important features of a numerical scheme: *dissipation* and *dispersion*. To this aim, let us suppose that the initial datum $u^0(x)$ of problem (9.47) is $2\pi-$periodic so that it can be expanded in a Fourier series as

$$u^0(x) = \sum_{k=-\infty}^{\infty} \alpha_k e^{ikx},$$

where

$$\alpha_k = \frac{1}{2\pi} \int_0^{2\pi} u^0(x) e^{-ikx} \mathrm{d}x$$

is the $k-$th Fourier coefficient of $u^0(x)$. The exact solution $u$ of problem (9.47) satisfies (formally) the nodal conditions

$$u(x_j, t^n) = \sum_{k=-\infty}^{\infty} \alpha_k e^{ikj\Delta x} (g_k)^n, \quad j \in \mathbb{Z}, n \in \mathbb{N} \tag{9.66}$$

with $g_k = e^{-iak\Delta t}$, while the numerical solution $u_j^n$, computed by one of the schemes introduced in Section 9.3.1, reads

$$u_j^n = \sum_{k=-\infty}^{\infty} \alpha_k e^{ikj\Delta x} (\gamma_k)^n, \quad j \in \mathbb{Z}, \quad n \in \mathbb{N}. \tag{9.67}$$

The form of coefficients $\gamma_k \in \mathbb{C}$ depends on the particular numerical scheme used; for instance, for the scheme (9.53) we can show that $\gamma_k = 1 - a\lambda i \sin(k\Delta x)$.
We notice that, while $|g_k| = 1$ for any $k \in \mathbb{Z}$, the values $|\gamma_k|$ depend on the *CFL number* $a\lambda$, and then also on the chosen discretization. Precisely, by choosing $\|\cdot\|_\Delta = \|\cdot\|_{\Delta,2}$, one can prove that a necessary and sufficient condition for a given numerical scheme to satisfy the stability inequality (9.63) is that $|\gamma_k| \leq 1$, $\forall k \in \mathbb{Z}$. The ratio $\epsilon_a(k) = |\gamma_k|/|g_k| = |\gamma_k|$ is the so-called *dissipation coefficient* (or *amplification coefficient*) of the $k-$th harmonic associated with the numerical scheme. We recall that the exact solution of (9.47) is the travelling wave $u(x,t) = u^0(x - at)$ whose amplitude is independent of time; as of its numerical approximation (9.67), the smaller $\epsilon_a(k)$, the higher the reduction of the wave amplitude and, whence the higher the numerical dissipation. Moreover, if the stability condition is violated, then the wave amplitude will increase and a *blow-up* of the numerical solution will occur at sufficiently large times.

Besides dissipation, numerical schemes introduce also dispersion, that is either a delay or an advance in the wave propagation. To understand this phenomenon we write $g_k$ and $\gamma_k$ as follows:

$$g_k = e^{-ia\lambda\phi_k}, \qquad \gamma_k = |\gamma_k| e^{-i\omega\Delta t} = |\gamma_k| e^{-i\frac{\omega}{k}\lambda\phi_k},$$

$\phi_k = k\Delta x$ being the so-called *phase angle* associated to the $k-$th harmonic.
By comparing $g_k$ with $\gamma_k$ and recalling that $a$ is the propagation velocity of the "exact" wave, we define *dispersion coefficient* associated to the $k$th harmonic the value $\epsilon_d(k) = \frac{\omega}{ak} = \frac{\omega\Delta t}{\phi_k a\lambda}$.

In Figures 9.12 and 9.13 we report the exact solution of problem (9.50) (for $a = 1$) and the numerical solutions obtained by some of the schemes presented in Section 9.3.1. The initial datum is

$$u^0(x) = \begin{cases} \sin(2\pi x/\ell) & -1 \leq x \leq \ell \\ 0 & \ell < x < 3, \end{cases} \tag{9.68}$$

of wavelength $\ell = 1$ (left) and $\ell = 1/2$ (right). In both cases the CFL number is equal to 0.8. For $\ell = 1$ we have chosen $\Delta x = \ell/20 = 1/20$, so

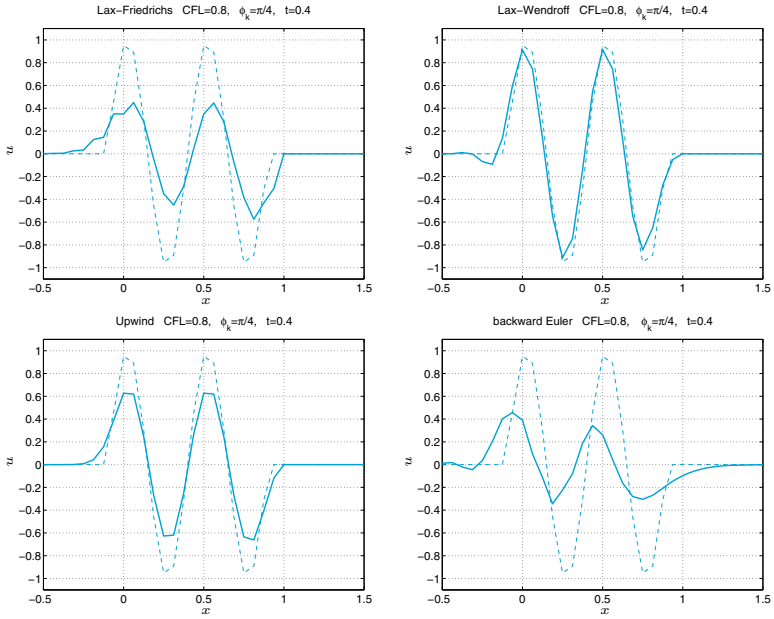


**Figure 9.12.** Exact solution (*dashed line*) and numerical solution (*solid line*) of problem (9.50) at $t = 0.4$, with $a = 1$ and initial datum given by (9.68) with equal wavelength $\ell = 1/2$

that $\phi_k = 2\pi\Delta x/\ell = \pi/10$ and $\Delta t = 1/25$. For $\ell = 1/2$ we have chosen $\Delta x = \ell/8 = 1/16$, so that $\phi_k = \pi/4$ and $\Delta t = 1/20$.

In Figures 9.14 and 9.15 we display the dissipation and dispersion coefficients, respectively, versus the CFL number (at top) and the phase angle $\phi_k = k\Delta x$ (at bottom).
Notice from Figure 9.14 that, when CFL=0.8, the Lax-Wendroff scheme is the least dissipative one, this information is confirmed by the numerical solutions shown in Figure 9.13, for both $\phi_k = \pi/10$ and $\phi_k = \pi/4$. About the dispersion error, still for CFL=0.8, from Figure 9.15 it emerges that the upwind scheme features the lowest dispersion and shows a light phase advance; the Lax-Friederichs scheme has a considerable phase advance, while both Lax-Wendroff and implicit Euler/centered schemes show a phase delay. These conclusions are confirmed by the numerical solution shown in Figure 9.12.

Notice that the dissipation coefficient is responsible for the damping of the wave amplitude, while the dispersion coefficient is responsible for the inexact propagation velocity.

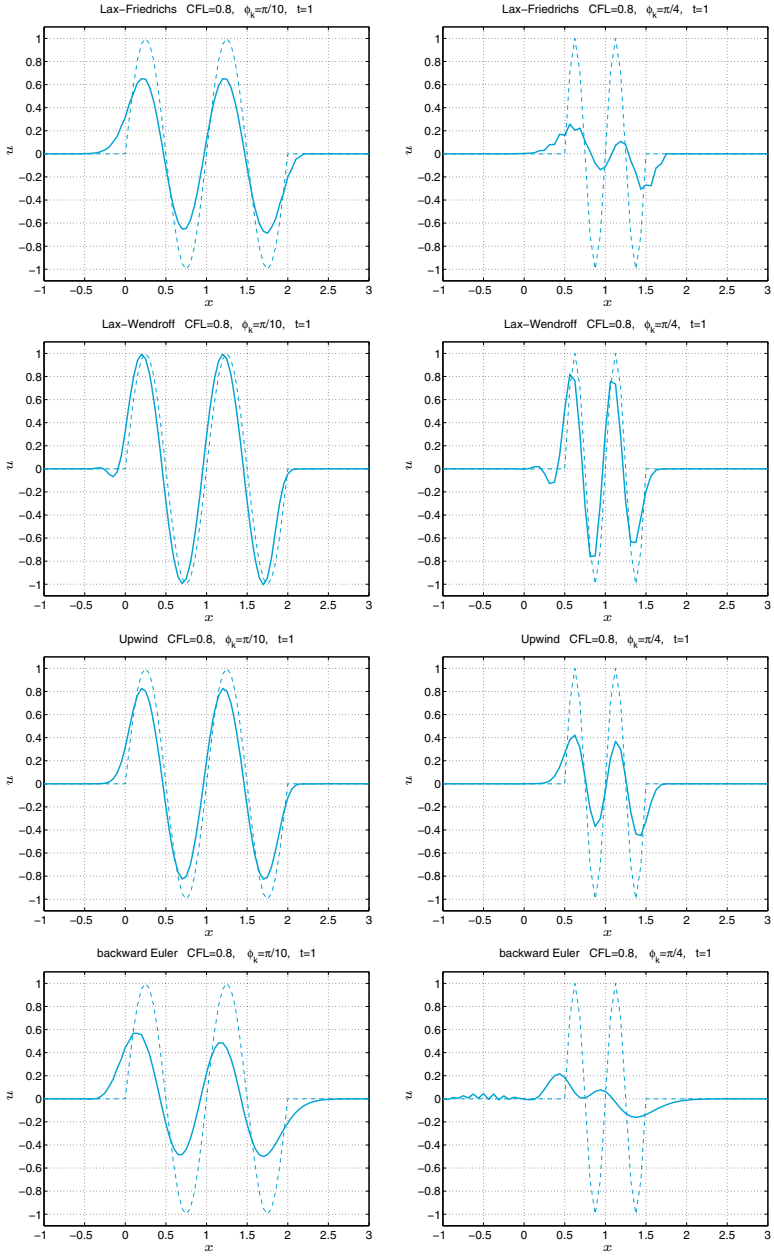


**Figure 9.13.** Exact solution (*dashed line*) and numerical solution (*solid line*) at $t = 1$ of problem (9.50) with $a = 1$ and initial datum given by (9.68) with wavelength $\ell = 1$ (left) and $\ell = 1/2$ (right)

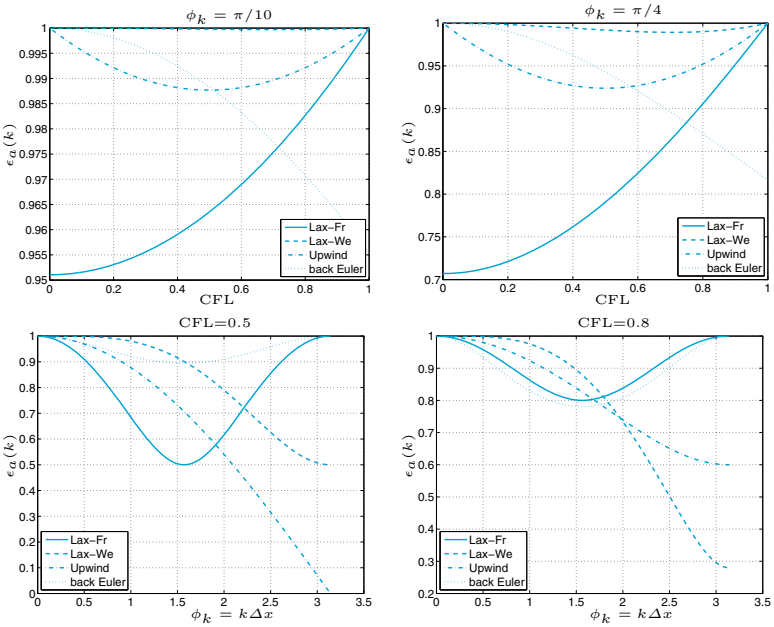


**Figure 9.14.** Dissipation coefficients

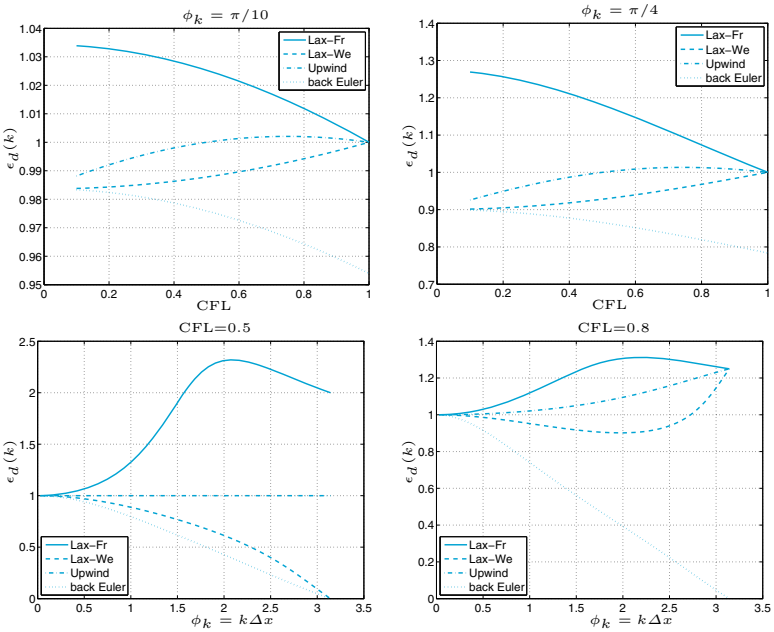


**Figure 9.15.** Dispersion coefficients

### 9.3.3 Finite element space discretization of the scalar advection equation

Following Section 9.2.3, a Galerkin semi-discrete approximation of problem (9.47) can be introduced as follows. Let us assume that $a = a(x) > 0$ $\forall x \in [\alpha, \beta]$, so that the node $x = \alpha$ coincides with the *inflow boundary*. For any $t > 0$, we complete system (9.47) with the boundary condition

$$u(\alpha, t) = \varphi(t), \qquad t > 0, \tag{9.69}$$

where $\varphi$ is a given function of $t$.

After defining the space

$$V_h^{in} = \{v_h \in V_h : \ v_h(\alpha) = 0\},$$

we consider the following finite element approximation of problem (9.47), (9.69): for any $t \in (0, T)$ find $u_h(t) \in V_h$ such that

$$\begin{cases} \displaystyle\int_\alpha^\beta \frac{\partial u_h(t)}{\partial t} v_h \, \mathrm{d}x + \int_\alpha^\beta a \frac{\partial u_h(t)}{\partial x} v_h \, \mathrm{d}x = 0 & \forall \ v_h \in V_h^{in}, \\ u_h(t) = \varphi(t) & \text{at } x = \alpha, \end{cases} \tag{9.70}$$

with $u_h(0) = u_h^0 \in V_h$ being a suitable finite element approximation of the initial datum $u^0$, e.g. its piecewise polynomial interpolant.

The time discretization of (9.70) can be accomplished still by using finite difference schemes. If, for instance, we use the backward Euler method, for any $n \geq 0$, we have: find $u_h^{n+1} \in V_h$ such that

$$\frac{1}{\Delta t} \int_\alpha^\beta (u_h^{n+1} - u_h^n) v_h \, \mathrm{d}x + \int_\alpha^\beta a \frac{\partial u_h^{n+1}}{\partial x} v_h \, \mathrm{d}x = 0 \quad \forall v_h \in V_h^{in}, \tag{9.71}$$

with $u_h^{n+1}(\alpha) = \varphi^{n+1}$.

If $\varphi = 0$, we can conclude that

$$\|u_h^n\|_{\mathrm{L}^2(\alpha,\beta)} \leq \|u_h^0\|_{\mathrm{L}^2(\alpha,\beta)} \quad \forall n \geq 0,$$

which means that the backward Euler scheme is unconditionally stable with respect to the norm $\|v\|_{\mathrm{L}^2(\alpha,\beta)} = \left(\int_\alpha^\beta v^2(x)\mathrm{d}x\right)^{1/2}$.

See Exercises 9.10-9.14.

## 9.4 The wave equation

We consider now the following second-order hyperbolic equation in one dimension

$$\boxed{\frac{\partial^2 u}{\partial t^2} - c\frac{\partial^2 u}{\partial x^2} = f} \tag{9.72}$$

where $c$ is a given positive constant.
When $f = 0$, the general solution of (9.72) is the so-called d'Alembert travelling-wave

$$u(x,t) = \psi_1(\sqrt{c}t - x) + \psi_2(\sqrt{c}t + x), \tag{9.73}$$

for arbitrary functions $\psi_1$ and $\psi_2$.

In what follows we consider problem (9.72) for $x \in (a,b)$ and $t > 0$, therefore we need to complete the differential equation with the initial data

$$u(x,0) = u_0(x) \text{ and } \frac{\partial u}{\partial t}(x,0) = v_0(x),\ x \in (a,b), \tag{9.74}$$

and the boundary data

$$u(a,t) = 0 \text{ and } u(b,t) = 0,\ t > 0. \tag{9.75}$$

In this case, $u$ may represent the transverse displacement of an elastic vibrating string of length $b-a$, fixed at the endpoints, and $c$ is a positive coefficient depending on the specific mass of the string and on its tension. The string is subjected to a vertical force of density $f$. The functions $u_0(x)$ and $v_0(x)$ denote respectively the initial displacement and the initial velocity of the string.

The change of variables

$$\omega_1 = \frac{\partial u}{\partial x}, \qquad \omega_2 = \frac{\partial u}{\partial t},$$

transforms (9.72) into the first-order system

$$\boxed{\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathrm{A}\frac{\partial \boldsymbol{\omega}}{\partial x} = \mathbf{f}, \qquad x \in (a,b),\ t > 0} \tag{9.76}$$

where

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}, \ \mathrm{A} = \begin{bmatrix} 0 & -1 \\ -c & 0 \end{bmatrix}, \ \mathbf{f} = \begin{bmatrix} 0 \\ f \end{bmatrix},$$

and the initial conditions are $\omega_1(x,0) = u_0'(x)$ and $\omega_2(x,0) = v_0(x)$ for $x \in (a,b)$.

In general, we can consider systems of the form (9.76) where $\boldsymbol{\omega}$, $\mathbf{f} : \mathbb{R} \times [0, \infty) \to \mathbb{R}^p$ are two given vector functions and $\mathrm{A} \in \mathbb{R}^{p \times p}$ is a matrix with constant coefficients. This system is said *hyperbolic* if A is diagonalizable and has real eigenvalues, that is, if there exists a nonsingular matrix $\mathrm{T} \in \mathbb{R}^{p \times p}$ such that

$$\mathrm{A} = \mathrm{T} \Lambda \mathrm{T}^{-1},$$

where $\Lambda = \mathrm{diag}(\lambda_1, ..., \lambda_p)$ is the diagonal matrix of the real eigenvalues of A, while $\mathrm{T} = (\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^p)$ is the matrix whose column vectors are the right eigenvectors of A. Thus

$$\mathrm{A}\mathbf{v}^k = \lambda_k \mathbf{v}^k, \qquad k = 1, \ldots, p.$$

Introducing the *characteristic variables* $\mathbf{w} = \mathrm{T}^{-1}\boldsymbol{\omega}$, system (9.76) becomes

$$\frac{\partial \mathbf{w}}{\partial t} + \Lambda \frac{\partial \mathbf{w}}{\partial x} = \mathbf{g},$$

where $\mathbf{g} = \mathrm{T}^{-1}\mathbf{f}$. This is a system of $p$ independent scalar equations of the form

$$\frac{\partial w_k}{\partial t} + \lambda_k \frac{\partial w_k}{\partial x} = g_k, \qquad k = 1, \ldots, p.$$

When $g_k = 0$, its solution is given by $w_k(x,t) = w_k(x - \lambda_k t, 0)$, $k = 1, \ldots, p$. Therefore the solution $\boldsymbol{\omega} = \mathrm{T}\mathbf{w}$ of problem (9.76) (for $\mathbf{f} = \mathbf{0}$) can be written as

$$\boldsymbol{\omega}(x,t) = \sum_{k=1}^{p} w_k(x - \lambda_k t, 0)\mathbf{v}^k.$$

The curve $(x_k(t), t)$ in the plane $(x,t)$ that satisfies $x_k'(t) = \lambda_k$ is the $k$th characteristic curve (see Section 9.3) and $w_k$ is constant along it. Then $\boldsymbol{\omega}(\overline{x}, \overline{t})$ depends only on the initial datum at the points $\overline{x} - \lambda_k \overline{t}$. For this reason, the set of $p$ points that form the feet of the characteristics issuing from the point $(\overline{x}, \overline{t})$,

$$D(\overline{t}, \overline{x}) = \{x \in \mathbb{R} \ : \ x = \overline{x} - \lambda_k \overline{t} \ , \ k = 1, ..., p\}, \tag{9.77}$$

is called the *domain of dependence* of the solution $\boldsymbol{\omega}(\overline{x}, \overline{t})$.

If (9.76) is set on a bounded interval $(a, b)$ instead of on the whole real line, the inflow point for each characteristic variable $w_k$ is determined by the sign of $\lambda_k$. Correspondingly, the number of positive eigenvalues determines the number of boundary conditions that should be assigned at $x = a$, whereas at $x = b$ the number of conditions that must be assigned equals the number of negative eigenvalues.

**Example 9.6** System (9.76) is hyperbolic since A is diagonalizable with matrix

$$\mathrm{T} = \begin{bmatrix} -\dfrac{1}{\sqrt{c}} & \dfrac{1}{\sqrt{c}} \\ 1 & 1 \end{bmatrix}$$

and features two distinct real eigenvalues $\pm\sqrt{c}$ (representing the propagation velocities of the wave). Moreover, one boundary condition needs to be prescribed at every end-point, as in (9.75). ■

### 9.4.1 Finite difference approximation of the wave equation

To discretize in time equation (9.72) we can use the Newmark method formerly proposed in Chapter 8 for second-order ordinary differential equations, see (8.71). Still denoting by $\Delta t$ the (uniform) time-step and using in space the classical finite difference method on a grid with nodes $x_j = x_0 + j\Delta x$, $j = 0, \ldots, N+1$, $x_0 = a$ and $x_{N+1} = b$, the Newmark scheme for (9.72) reads as follows: for any $n \geq 1$ find $\{u_j^n, v_j^n, j = 1, \ldots, N\}$ such that

$$\begin{aligned} u_j^{n+1} &= u_j^n + \Delta t v_j^n \\ &\quad + \Delta t^2 \left[\zeta(cw_j^{n+1} + f(x_j, t^{n+1})) + (1/2 - \zeta)(cw_j^n + f(x_j, t^n))\right], \\ v_j^{n+1} &= v_j^n + \Delta t \left[(1-\theta)(cw_j^n + f(x_j, t^n)) + \theta(cw_j^{n+1} + f(x_j, t^{n+1}))\right], \end{aligned} \tag{9.78}$$

with $u_j^0 = u_0(x_j)$ and $v_j^0 = v_0(x_j)$ and $w_j^k = (u_{j+1}^k - 2u_j^k + u_{j-1}^k)/(\Delta x)^2$ for $k = n$ or $k = n+1$. System (9.78) must be completed by imposing the boundary conditions (9.75).

The Newmark method is implemented in Program 9.4. The input parameters are the vectors `xspan=[a,b]` and `tspan=[0,T]`, the number of discretization intervals in space (`nstep(1)`) and in time (`nstep(2)`), the scalar `c` (corresponding to the positive constant $c$), the function handles `u0` and `v0` associated with the initial data $u_0(x)$ and $v_0(x)$, respectively, and the function handles `g` and `fun` associated with the functions $g(x,t)$ and $f(x,t)$, respectively. Finally, the vector `param` allows to specify the values of the coefficients (`param(1)`=$\theta$, `param(2)`=$\zeta$). This method is second order accurate with respect to $\Delta t$ if $\theta = 1/2$, whereas it is first order if $\theta \neq 1/2$. Moreover, the condition $\theta \geq 1/2$ is necessary to ensure stability (see Section 8.9).

**Program 9.4. newmarkwave**: Newmark method for the wave equation

```
function [xh,uh]=newmarkwave(xspan,tspan,nstep,param,...
                c,u0,v0,g,f,varargin)
%NEWMARKWAVE solves the wave equation with the Newmark
```

```
% method.
% [XH,UH]=NEWMARKWAVE(XSPAN,TSPAN,NSTEP,PARAM,C,...
% U0,V0,G,F)
% solves the wave equation D^2 U/DT^2 - C D^2U/DX^2 = F
% in (XSPAN(1),XSPAN(2)) X (TSPAN(1),TSPAN(2)) using
% Newmark method with initial conditions U(X,0)=U0(X),
% DU/DX(X,0)=V0(X) and Dirichlet boundary conditions
% U(X,T)=G(X,T) for X=XSPAN(1) and X=XSPAN(2). C is a
% positive constant.
% NSTEP(1) is the number of space integration intervals
% NSTEP(2) is the number of time-integration intervals.
% PARAM(1)=ZETA and PARAM(2)=THETA.
% U0(X), V0(X), G(X,T) and F(x,T) are function handles.
% XH contains the nodes of the discretization.
% UH contains the numerical solutions at time TSPAN(2).
% [XH,UH]=NEWMARKWAVE(XSPAN,TSPAN,NSTEP,PARAM,C,...
% U0,V0,G,F,P1,P2,...) passes the additional parameters
%  P1,P2,...to the functions U0,V0,G,F.
h  = (xspan(2)-xspan(1))/nstep(1);
dt = (tspan(2)-tspan(1))/nstep(2);
zeta = param(1);  theta = param(2);
N = nstep(1)+1;
e = ones(N,1); D = spdiags([e -2*e e],[-1,0,1],N,N);
I = speye(N); lambda = dt/h;
A = I-c*lambda^2*zeta*D;
An = I+c*lambda^2*(0.5-zeta)*D;
A(1,:) = 0; A(1,1) = 1; A(N,:) = 0; A(N,N) = 1;
xh = (linspace(xspan(1),xspan(2),N))';
fn = f(xh,tspan(1),varargin{:});
un = u0(xh,varargin{:});
vn = v0(xh,varargin{:});
[L,U]=lu(A);
alpha = dt^2*zeta; beta = dt^2*(0.5-zeta);
theta1 = 1-theta;
for t = tspan(1)+dt:dt:tspan(2)
    fn1 = f(xh,t,varargin{:});
    rhs = An*un+dt*I*vn+alpha*fn1+beta*fn;
    temp = g([xspan(1),xspan(2)],t,varargin{:});
    rhs([1,N]) = temp;
    uh = L\rhs;    uh = U\uh;
    v = vn + dt*((1-theta)*(c*D*un/h^2+fn)+...
        theta*(c*D*uh/h^2+fn1));
    fn = fn1;    un = uh;    vn = v;
end
```

An alternative to the Newmark method is provided by the following *Leap-Frog* method

$$u_j^{n+1} - 2u_j^n + u_j^{n-1} = c\left(\frac{\Delta t}{\Delta x}\right)^2 (u_{j+1}^n - 2u_j^n + u_{j-1}^n), \tag{9.79}$$

which is obtained by discretizing both time and space derivatives by the centered finite difference formula (9.12).

Both Newmark (9.78) and *Leap-Frog* (9.79) schemes are second order accurate with respect to $\Delta t$ and $\Delta x$. About stability, the *Leap-Frog* method is stable provided that the CFL condition $\Delta t \leq \Delta x/\sqrt{c}$ is

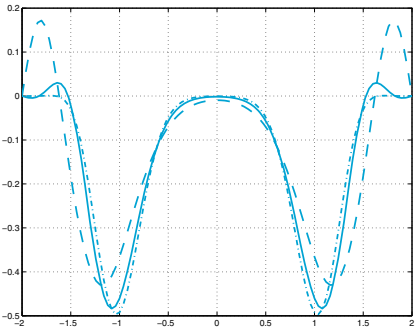


**Figure 9.16.** Comparison between the solutions obtained using the Newmark method for a discretization with $\Delta x = 0.04$ and $\Delta t = 0.15$ (*dashed line*), $\Delta t = 0.075$ (*solid line*) and $\Delta t = 0.0375$ (*dashed-dotted line*)

satisfied, while the Newmark method is unconditionally stable if $2\zeta \geq \theta \geq \frac{1}{2}$ (see [Joh90]).

**Example 9.7** Using Program 9.4 we study the evolution of the initial condition $u_0(x) = e^{-10x^2}$ for $x \in (-2, 2)$, by putting $f = 0$ and $c = 1$ in (9.72). We assume $v_0 = 0$ and homogeneous Dirichlet boundary conditions. In Figure 9.16 we compare the solutions obtained at time $t = 3$ using $\Delta x = 0.04$ and time-steps $\Delta t = 0.15$ (*dashed line*), $\Delta t = 0.075$ (*solid line*) and $\Delta t = 0.0375$ (*dashed-dotted line*). The parameters of the Newmark method are $\theta = 1/2$ and $\zeta = 0.25$, and they ensure a second order unconditionally stable method. ■

**Example 9.8 (Communications)** In this example we use the equation (9.9) to model the way a telegraph wire transmits a pulse of voltage. The equation is a combination of diffusion and wave equations, and accounts for effects of finite velocity in a standard mass transport equation. In Figure 9.17 we compare the evolution of one bump (precisely a cubic B-spline (see [QSS07, Sect. 8.7.2])) centered in $x = 3$ and non-null in the interval (1,5) using the wave equation (9.72) (*dashed line*) and the telegrapher's equation (9.9) (*solid line*), on the interval $(0, 10)$ with $c = 1$, $\alpha = 0.5$ and $\beta = 0.04$. The initial speed is chosen to be $v_0(x) = -cu_0'(x)$ ($v_0(x) = -cu_0'(x) - \alpha/2u_0(x)$, resp.) for the wave (telegrapher's, resp.) equation, so that the bump travels with speed $c$. We have solved both the wave equation and telegrapher's equation by the Newmark scheme using $\Delta x = 0.025$, time-step $\Delta t = 0.1$, $\zeta = 1/4$ and $\theta = 1/2$. To approximate the wave equation we have called Program 9.4, while to solve the telegrapher's equation we have written a different program implementing the Newmark scheme (8.71) applied to equation (9.9). The presence of the dissipation effect is evident in the solution of the telegrapher's equation. ■

An alternative approach consists of discretizing the first-order system (9.76) instead of the (equivalent) second order scalar equation (9.72). When $\mathbf{f} = \mathbf{0}$, Lax-Wendroff and upwind schemes for the hyperbolic system (9.76) are defined as follows:

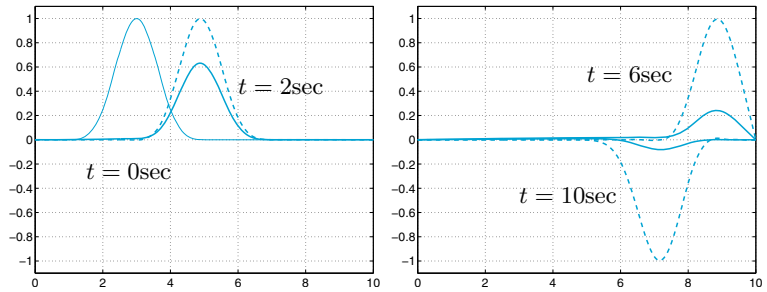


**Figure 9.17.** Propagation of a pulse of voltage using the wave equation (*dashed line*) and the telegrapher's equation (*solid line*). At left, the thin solid line represents the initial condition $u_0(x)$

1. *Lax-Wendroff method*

$$\begin{aligned}\boldsymbol{\omega}_j^{n+1} = \boldsymbol{\omega}_j^n - \frac{\lambda}{2}\mathrm{A}(\boldsymbol{\omega}_{j+1}^n - \boldsymbol{\omega}_{j-1}^n) \\ + \frac{\lambda^2}{2}\mathrm{A}^2(\boldsymbol{\omega}_{j+1}^n - 2\boldsymbol{\omega}_j^n + \boldsymbol{\omega}_{j-1}^n),\end{aligned} \tag{9.80}$$

2. *upwind (or forward Euler/decentered) method*

$$\begin{aligned}\boldsymbol{\omega}_j^{n+1} = \boldsymbol{\omega}_j^n - \frac{\lambda}{2}\mathrm{A}(\boldsymbol{\omega}_{j+1}^n - \boldsymbol{\omega}_{j-1}^n) \\ + \frac{\lambda}{2}|\mathrm{A}|(\boldsymbol{\omega}_{j+1}^n - 2\boldsymbol{\omega}_j^n + \boldsymbol{\omega}_{j-1}^n),\end{aligned} \tag{9.81}$$

where $|\mathrm{A}| = \mathrm{T}|\Lambda|\mathrm{T}^{-1}$ and $|\Lambda|$ is the diagonal matrix of the moduli of the eigenvalues of A.

The Lax-Wendroff method is second order accurate (in both time and space), while the upwind scheme is first order.

About stability, all considerations made in Section 9.3.1 are still valid, provided the CFL condition (9.65) is replaced by

$$\Delta t < \frac{\Delta x}{\rho(\mathrm{A})}. \tag{9.82}$$

As usual, $\rho(\mathrm{A})$ denotes the spectral radius of A.

For the proof of these results see, e.g., [QV94], [LeV02], [GR96], [QSS07, Chapter 13].

See Exercises 9.8-9.9.

### Let us summarize

1. One-dimensional boundary value problems are set up on an interval; boundary conditions on the solution (or on its derivative) must be prescribed at the endpoints of the interval;
2. numerical approximation can be carried out by finite-differences (arising from truncated Taylor series) or by finite-elements (arising from the weak formulation of the differential problem; in this context, both test and trial functions are piecewise polynomials);
3. multidimensional problems can be faced by using similar arguments. For two-dimensional boundary-value problems, finite element approximations make use of piecewise polynomials, where "piecewise" refers to either triangles or quadrilaterals of a grid partitioning the spatial domain;
4. matrices arising from both finite element and finite difference discretizations are sparse and ill-conditioned;
5. initial-boundary-value problems contain time derivatives of the solution which are discretized by finite difference formulas, of either explicit or implicit type;
6. when explicit schemes are used, stability conditions have to be satisfied: the time-step turns out to be bounded by the spatial grid size. On the other hand, when implicit schemes are used, a linear algebraic system (similar to that obtained for stationary problems) has to be solved at each time level;
7. in this Chapter we have presented some simple linear problems of elliptic, parabolic and hyperbolic type. For a more exhaustive treatment of this subject we suggest the reader to refer to the bibliography presented in the next Section.

## 9.5 What we haven't told you

We could simply say that we have told you almost nothing, since the field of numerical analysis which is devoted to the numerical approximation of partial differential equations is so broad and multifaceted to deserve an entire monograph simply for addressing the most essential concepts (see, e.g., [TW98], [EEHJ96]).

We would like to mention that the finite element method is nowadays probably the most widely diffused method for the numerical solution of partial differential equations (see, e.g., [Qua13], [QV94], [Bra97], [BS01]). As already mentioned the MATLAB toolbox `pde` allows the solution of a broad family of partial differential equations by the linear finite element method, in particular for the discretization of space variables.

Other popular techniques are the spectral methods (see, e.g., [CHQZ06], [CHQZ07], [Fun92], [BM92], [KS99]) and the finite volume method (see, e.g., [Krö98], [Hir88] and [LeV02]).

**Octave 9.1** The Octave-Forge package `bim` offers most of the main functionalities of the `pde` toolbox, although its syntax is in general not compatible with that of MATLAB. ■

## 9.6 Exercises

**Exercise 9.1** Verify that matrix (9.15) is positive definite.

**Exercise 9.2** Verify that the eigenvalues of the matrix $\mathrm{A} \in \mathbb{R}^{N\times N}$, defined in (9.15), are

$$\lambda_j = 2(1-\cos(j\theta)), \quad j = 1,\ldots,N,$$

while the corresponding eigenvectors are

$$\mathbf{q}_j = (\sin(j\theta), \sin(2j\theta), \ldots, \sin(Nj\theta))^T,$$

where $\theta = \pi/(N+1)$. Deduce that $K(A)$ is proportional to $h^{-2}$.

**Exercise 9.3** Prove that the quantity (9.12) provides a second order approximation of $u''(\bar{x})$ with respect to $h$.

**Exercise 9.4** Compute the matrix and the right-hand side of the numerical scheme that we have proposed to approximate problem (9.17).

**Exercise 9.5** Use the finite difference method to approximate the boundary-value problem

$$\begin{cases} -u'' + \dfrac{k}{T}u = \dfrac{w}{T} \text{ in } (0,1), \\ u(0) = u(1) = 0, \end{cases}$$

where $u = u(x)$ represents the vertical displacement of a string of length 1, subject to a transverse load of intensity $w(x)$ per unit length. $T$ is the tension and $k$ is the elastic coefficient of the string. For the case in which $w(x) = 1 + \sin(4\pi x)$, $T = 1$ and $k = 0.1$, compute the solution corresponding to $h = 1/i$, with $i = 10, 20, 40$, and deduce the order of accuracy of the method.

**Exercise 9.6** Use the finite difference method to solve problem (9.17) in the case where the following boundary conditions are prescribed at the endpoints (called *Neumann* boundary conditions)

$$u'(a) = \alpha, \qquad u'(b) = \beta.$$

Use the formulae given in (4.11) to discretize $u'(a)$ and $u'(b)$.

**Exercise 9.7** Verify that, when using a uniform grid, the right-hand side of the system (9.14) associated with the centered finite difference scheme coincides, up a factor $h$, with that of the finite element scheme (9.27) provided that the composite trapezoidal formula is used to compute the integrals on the elements $I_{j-1}$ and $I_j$.

**Exercise 9.8** Verify that $\mathrm{div}\nabla\phi = \Delta\phi$, where $\nabla$ is the gradient operator that associates to a function $u$ the vector whose components are the first order partial derivatives of $u$.

**Exercise 9.9 (Thermodynamics)** Consider a square plate whose side length is 20 cm and whose thermal conductivity is $k = 0.2$ cal/(sec·cm·C). Denote by $Q = 5$ cal/(cm$^3$·sec) the heat production rate per unit area. The temperature $T = T(x, y)$ of the plate satisfies the equation $-\Delta T = Q/k$. Assuming that $T$ is null on three sides of the plate and is equal to 1 on the fourth side, determine the temperature $T$ at the center of the plate.

**Exercise 9.10** Verify that the solution of problem (9.72), (9.74) – (9.75) (with $f = 0$) satisfies the identity

$$\int_a^b (u_t(x,t))^2 \mathrm{d}x + c\int_a^b (u_x(x,t))^2 \mathrm{d}x = \tag{9.83}$$
$$\int_a^b (v_0(x))^2 \mathrm{d}x + c\int_a^b (u_{0,x}(x))^2 \mathrm{d}x,$$

provided that $u_0(a) = u_0(b) = 0$.

**Exercise 9.11** Prove that the numerical solution provided by the backward Euler/centered scheme (9.62) is unconditionally stable, that is $\forall \Delta t > 0$,

$$\|\mathbf{u}^n\|_{\Delta,2} \leq \|\mathbf{u}^0\|_{\Delta,2} \qquad \forall n \geq 0. \tag{9.84}$$

**Exercise 9.12** Prove that the solution provided by the upwind scheme (9.59) satisfies the estimate

$$\|\mathbf{u}^n\|_{\Delta,\infty} \leq \|\mathbf{u}^0\|_{\Delta,\infty} \qquad \forall n \geq 0, \tag{9.85}$$

provided that the CFL condition has been verified. The inequality (9.85) is named *discrete maximum principle*.

**Exercise 9.13** Solve problem (9.47) with $a = 1$, $x \in (0, 0.5)$, $t \in (0, 1)$, initial datum $u^0(x) = 2\cos(4\pi x) + \sin(20\pi x)$ and boundary condition $u(0,t) = 2\cos(4\pi t) - sin(20\pi t)$ for $t \in (0, 1)$. Use both Lax-Wendroff (9.57) and upwind (9.59) schemes. Set the CFL number equal to 0.5. Verify experimentally that the Lax-Wendroff scheme is second-order accurate with respect to $\Delta x$ and $\Delta t$, while the upwind scheme is first-order accurate. To evaluate the error use the norm $\|\cdot\|_{\Delta,2}$.

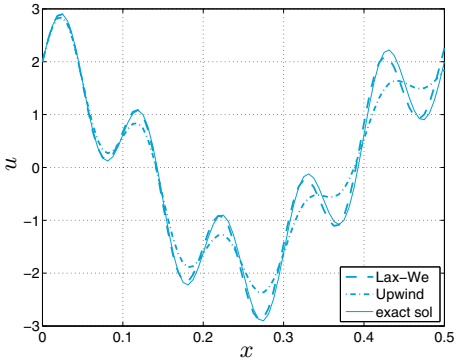


**Figure 9.18.** Numerical solutions at time $t = 5$ for the problem (9.47) by using data of Exercise 9.13. The CFL number is 0.8

**Exercise 9.14** In Figure 9.18 both exact and numerical solutions of problem (9.47) at time $t = 5$ are shown. The latter are computed by the Lax-Wendroff (9.57) and upwind (9.59) schemes, using the same data of Exercise 9.13. By knowing that the CFL number is 0.8 and that we have used $\Delta t = 5.e - 3$, comment on the dissipation and dispersion coefficients that we have obtained.