Applied Numerical Methods

William McLean and Kassem Mustapha, UNSW

March 1, 2023

Contents

Chapter 1

Introduction

1.1 PDEs and numerical solutions

Many computational engineering and science applications start with some law of physics that applies to some physical problem. This is mathematically expressed as a partial differential equation (PDE). PDEs arise in the mathematical modeling of physical, chemical and biological phenomena and many diverse subject areas such as fluid dynamics, electromagnetism, material science, astrophysics, etc. PDEs provide the natural mathematical description for pretty much everything in the universe that varies continuously in space and time. The relevant PDE must be complemented by the specific boundary and initial conditions that describe any particular application. In simple cases with a high degree of spatial symmetry, it might be possible to find a closed-form expression for the solution. However, the closed form might anyway be an infinite series or an integral that cannot be easily evaluated to obtain numerical values. So, seeking numerical approximate solutions are therefore essential, particularly for engineering applications that require reasonably precise numerical values for the solution and related quantities.

Current challenges include

- Multi-physics applications
- Multiscale modelling
- Stochastic PDEs/uncertainty quantification
- Parametric prediction
- PDE-constrained optimization,
- Data assimilation for PDE models.

Handling the above challenges requires a combination of mathematical analysis (forward, inverse, etc.), numerical methods, fast algorithms, high-performance computing and application-specific knowledge. In this course, we focus on learning different aspects of finite difference and finite element methods for solving heat (diffusion) equations starting with a simple one-dimensional (1D) steady-state model. Noting that the techniques learned have wide applicability. For instance, in the last two chapters we use these techniques for solving advection-diffusion and wave models.

Whenever dealing with numerical methods, one should address these questions which arise naturally:

1. Does the linear system arising from the numerical approximation always have a unique solution?

2. If so, what is the efficient way to compute the numerical solution?

3. Is our approximate solution stable?

4. How accurate is the approximation?

1.2 Some applications of PDEs

In this section, we give some examples on the applications of the PDEs.

Example 1 (Euler 1757): Ideal fluid flow. Here $\mathbf{u} = (u, v, w)$ is the fluid velocity at position $\mathbf{x} = (x, y, z)$. For incompressible flow, $\nabla \cdot \mathbf{u} = 0$, that is,

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0\,.$$

It can be written in a compact form as:

$$u_x + v_y + w_z = 0\,.$$

For an irrotational flow, $\nabla \times \mathbf{u} = 0$. Noting that, irrotational implies $\mathbf{u} = \nabla\phi$ for a velocity potential ϕ. So, if a flow is both irrotational and incompressible, then the velocity potential is harmonic, and so, it satisfies the Laplace equation $\nabla^2\phi = 0$. That is,

$$\phi_{xx} + \phi_{yy} + \phi_{zz} = 0.$$

Example 2 (Laplace 1782, Poisson 1813): Gravitational potential. Mass distribution with density $\rho(\mathbf{x})$, such as a planet. Newtonian potential

$$V(\mathbf{x}) = \int \frac{\rho(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|}\mathrm{d}\mathbf{y}\,.$$

Laplace observed that in free space, where $\rho \equiv 0$, the Newtonian potential satisfies $\nabla^2 V = 0$, however, Poisson showed that in fact $\nabla^2 V = \rho$ everywhere.

Example 3 (Fourier 1807): Heat equation. Here $u = u(x, t)$ is the temperature at position x and time t. From Fourier's law, the heat flux vector (heat flow) is

$$\mathbf{q} = -a\nabla u,$$

where a is the thermal conductivity. That is, heat flows is proportional to negative temperature gradient. Let c denote the specific heat capacity and ρ the mass density. Conservation of thermal energy implies, via the divergence theorem, that u satisfies the (heat) equation:

$$c\rho u_t + \nabla \cdot \mathbf{q} = 0.$$

In case the thermal conductivity a is constant, then, the above heat equation reduces to

$$c\rho u_t - a\nabla^2 u = 0.$$

Example 4 (d'Alembert 1746, Euler 1756): The second order wave equation (water waves, sound waves, seismic waves, etc.). It arises in fields like acoustics, electromagnetics, and fluid dynamics. The second order (two-way) wave equation describes the superposition of an incoming wave and an outgoing wave. However, the 1st order (one-way) wave equation describes

a single wave with predefined wave propagation direction. Mathematically, the simple second order (linear) wave equation takes the form:

$$u_{tt} - c^2 \nabla^2 u = 0.$$

Here $u = u(x, t)$ is the displacement at position x and time t, and the constant c comes from mass density and elasticity (as expected in Newton's and Hooke's laws).

Example 5 (Cauchy 1822, Navier 1821): Linear elasticity. Let $u(x, t)$ ($\mathbf{u} = (u_1, u_2, u_3)$) denote the displacement (from a reference configuration) of a material particle located at x in an elastic medium at time t. Denote the strain tensor by $\mathbf{e}$, the stress tensor by σ, the stiffness tensor by $\mathbf{C}$, and the body force by $\mathbf{F}$. The strain-displacement equation is:

$$e_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i),$$

which is equivalent to

$$\mathbf{e} = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right).$$

The stress–strain relation (generalized Hooke's law) is given by

$$\sigma_{ij} = C_{ijkl} e_{kl}, \quad \text{equivalent to} \quad \sigma = \mathbf{C} : \mathbf{e}.$$

Equation of motion (Newton's second law) is: with ρ being the mass density,

$$\rho \mathbf{u}_{tt} = \nabla \cdot \sigma + \mathbf{F}.$$

Example 6 (Maxwell 1861): Electromagnetics. Here,

- $\mathbf{E}$ is the electric field,
- $\mathbf{D}$ is the electric displacement,
- $\mathbf{H}$ is the magnetic field,
- $\mathbf{B}$ is the magnetic flux density,
- ρ is the charge density,
- $\mathbf{J}$ is the current density.

Gauss's law

$$\nabla \cdot \mathbf{E} = \rho.$$

Faraday's law of induction

$$\nabla \times \mathbf{E} + \mathbf{B}_t = \mathbf{0}.$$

No magnetic monopoles

$$\nabla \cdot \mathbf{B} = 0.$$

Ampére's circuital law

$$\nabla \times \mathbf{H} - \mathbf{D}_t = \mathbf{J}.$$

In a vacuum (in this case, $\rho = 0$ and $\mathbf{J} = \mathbf{0}$), $\mathbf{D} = \epsilon_0 \mathbf{E}$ and $\mathbf{H} = \frac{1}{\mu_0}\mathbf{B}$, where ϵ_0 is the permittivity and μ_0 is the permeability of the material.

Here are some more examples:

- Navier–Stokes equation in fluid mechanics (Navier 1822, Stokes 1851)
- Schrodinger equation in quantum mechanics (Schrodinger 1925)
- Black–Scholes equation in financial market (option pricing)
- Pattern formation (Turing 1952)
- Susceptible-Exposed-Infectious-Recovered (SEIR) model in epidemiology (Aron and Schwartz, 1984)

1.3 Taylor's theorem and useful estimates

To derive a finite difference approximation (in other words, to approximate the continuous derivatives via finite differences), we mainly rely on Taylor's theorem. This is the topic of the current section where we show different estimate that will be used in the forthcoming chapters to investigate the accuracy of the numerical solutions that will be developed later.

Theorem 1.1. *Let* $\delta > 0$. *If* $f \in C^{k+1}[x, x+\delta]$ *then*

$$f(x+\delta) = \sum_{j=0}^{k} \frac{1}{j!} f^{(j)}(x)\, \delta^j + (R_k f)(x, \delta), \tag{1.1}$$

where the remainder term is given by

$$(R_k f)(x, \delta) = \frac{1}{k!} \int_x^{x+\delta} (x+\delta-y)^k f^{(k+1)}(y)\, dy \tag{1.2}$$

and satisfies

$$|(R_k f)(x, \delta)| \leqslant \frac{\delta^{k+1}}{(k+1)!} \max_{x \leqslant y \leqslant x+\delta} |f^{(k+1)}(y)|. \tag{1.3}$$

Similarly, if $f \in C^{k+1}[x-\delta, x]$, *then*

$$f(x-\delta) = \sum_{j=0}^{k} \frac{(-1)^j}{j!} f^{(j)}(x)\, \delta^j + (R_k f)(x, -\delta)$$

where

$$(R_k f)(x, -\delta) = \frac{(-1)^{k+1}}{k!} \int_{x-\delta}^{x} (y+\delta-x)^k f^{(k+1)}(y)\, dy$$

and

$$|(R_k f)(x, -\delta)| \leqslant \frac{\delta^{k+1}}{(k+1)!} \max_{x-\delta \leqslant y \leqslant x} |f^{(k+1)}(y)|.$$

Proof. We use induction on k. If $k = 0$, then the formulas (1.1) and (1.2) reduce to

$$f(x+\delta) = f(x) + (R_0 f)(x, \delta) \quad \text{where} \quad (R_0 f)(x, \delta) = \int_x^{x+\delta} f'(y)\, dy,$$

which follows from the fundamental theorem of calculus. Let $k \geqslant 0$ and make the induction hypothesis that (1.1) holds with $R_k f$ given by (1.2). Integrating by parts, we have

$$\begin{aligned}(R_k f)(x, \delta) &= \left[-\frac{(x+\delta-y)^{k+1}}{(k+1)!} f^{(k+1)}(y) \right]_{y=x}^{y=x+\delta} + \int_x^{x+\delta} \frac{(x+\delta-y)^{k+1}}{(k+1)!} f^{(k+2)}(y)\, dy \\ &= \frac{\delta^{k+1}}{(k+1)!} f^{(k+1)}(x) + (R_{k+1} f)(x, \delta),\end{aligned}$$

implying that (1.1) and (1.2) hold with k replaced by $k+1$, as required. The estimate (1.3) is an immediate consequence of the inequality

$$\begin{aligned}|(R_k f)(x,\delta)| &\leqslant \int_x^{x+\delta} \frac{(x+\delta-y)^k}{k!}\,|f^{(k+1)}(y)|\,dy \\ &\leqslant \left(\max_{x\leqslant y\leqslant x+\delta} |f^{(k+1)}(y)|\right)\int_x^{x+\delta}\frac{(x+\delta-y)^k}{k!}\,dy.\end{aligned}$$

The expansion for $f(x-y)$ and the bound for $(R_k f)(x,-\delta)|$ follow in a similar fashion. □

A straight forward manipulation of such Taylor expansions shows that

$$f''(x) = \frac{f(x+\delta)-2f(x)+f(x-\delta)}{\delta^2} + O(\delta^2) \quad \text{as } \delta\to 0;$$

more precisely, the following result holds.

Theorem 1.2. *If $f\in C^4[x-\delta,x+\delta]$, then*

$$\left|f''(x) - \frac{f(x+\delta)-2f(x)+f(x-\delta)}{\delta^2}\right| \leqslant \frac{\delta^2}{12}\max_{x-\delta\leqslant y\leqslant x+\delta}|f^{(4)}(y)|.$$

Proof. Since

$$\begin{aligned}f(x+\delta) &= f(x)+f'(x)\delta+\tfrac{1}{2}f''(x)\delta^2+\tfrac{1}{3!}f'''(x)\delta^3+(R_3f)(x,+\delta),\\ f(x-\delta) &= f(x)-f'(x)\delta+\tfrac{1}{2}f''(x)\delta^2-\tfrac{1}{3!}f'''(x)\delta^3+(R_3f)(x,-\delta),\end{aligned}$$

we see that

$$f(x+\delta)+f(x-\delta) = 2f(x)+f''(x)\delta^2+(R_3f)(x,\delta)+(R_3f)(x,-\delta).$$

Thus,

$$\frac{f(x+\delta)-2f(x)+f(x-\delta)}{\delta^2} - f''(x) = \frac{(R_3f)(x,\delta)+(R_3f)(x,-\delta)}{\delta^2} \tag{1.4}$$

and since

$$\begin{aligned}|(R_3f)(x,\delta)+(R_3f)(x,-\delta)| &\leqslant \frac{\delta^4}{4!}\max_{x\leqslant y\leqslant x+\delta}|f^{(4)}(y)| + \frac{\delta^4}{4!}\max_{x-\delta\leqslant y\leqslant x}|f^{(4)}(y)| \\ &\leqslant \frac{2\delta^4}{4!}\max_{x-\delta\leqslant y\leqslant x+\delta}|f^{(4)}(y)|\end{aligned}$$

the result follows. □

We also need to approximate the first derivative f'. Once again, we apply Taylor expansions, this time showing that

$$\frac{f(x+\delta)-f(x-\delta)}{2\delta} = f'(x)+O(\delta^2) \quad \text{as } \delta\to 0.$$

More precisely, the following holds.

Theorem 1.3. *If f is $C^3[x-\delta,x+\delta]$, then*

$$\left|f'(x) - \frac{f(x+\delta)-f(x-\delta)}{2\delta}\right| \leqslant \frac{\delta^2}{6}\max_{x-\delta\leqslant y\leqslant x+\delta}|f'''(y)|.$$

Proof. Since

$$f(x \pm \delta) = f(x) \pm f'(x)\delta + \tfrac{1}{2}f''(x)\delta^2 + (R_2 f)(x, \pm\delta),$$

we have

$$f(x+\delta) - f(x-\delta) = 2f'(x)\delta + (R_2 f)(x,\delta) - (R_2 f)(x,-\delta).$$

Thus,

$$\frac{f(x+\delta) - f(x-\delta)}{2\delta} - f'(x) = \frac{(R_2 f)(x,\delta) - (R_2 f)(x,-\delta)}{2\delta},$$

and since

$$\begin{aligned}\big|(R_2 f)(x,\delta) - (R_2 f)(x,-\delta)\big| &\leqslant \frac{\delta^3}{3!} \max_{x\leqslant y\leqslant x+\delta} |f'''(y)| + \frac{\delta^3}{3!} \max_{x-\delta\leqslant y\leqslant x} |f'''(y)| \\ &\leqslant \frac{2\delta^3}{3!} \max_{x-\delta\leqslant y\leqslant x+\delta} |f'''(y)|,\end{aligned}$$

the result follows at once. □

We derive a different estimate in the next theorem.

Theorem 1.4. *If* $f \in C^2[x-\delta, x+\delta]$, *then*

$$\left| f(x) - \frac{f(x+\delta) + f(x-\delta)}{2} \right| \leqslant \frac{\delta^2}{2} \max_{x-\delta\leqslant y\leqslant x+\delta} |f''(y)|.$$

Proof. Since

$$f(x+\delta) = f(x) + f'(x)\delta + (R_1 f)(x,\delta) \quad \text{and} \quad f(x-\delta) = f(x) - f'(x)\delta + (R_1 f)(x,-\delta),$$

we have

$$f(x+\delta) + f(x-\delta) = 2f(x) + (R_1 f)(x,\delta) + (R_1 f)(x,-\delta).$$

Thus,

$$\frac{f(x+\delta) + f(x-\delta)}{2} - f(x) = \frac{(R_1 f)(x,\delta) + (R_1 f)(x,-\delta)}{2},$$

so the result follows from Theorem 1.1. □

1.4 Exercises

1.1. Use Taylor expansion to find the coefficient c such that, as $x \to 0$,

(i) $\dfrac{x}{1-x^2} - \sin x = cx^3 + O(x^5)$.

(ii) $\log\cos x = cx^2 + O(x^4)$.

(iii) $\dfrac{x - \sinh x}{x^3} = c + O(x^2)$.

1.2. Use Taylor expansion to find the coefficient c such that

(i) $\dfrac{3u(x) - 4u(x-h) + u(x-2h)}{2h} = u'(x) + cu'''(x)\,h^2 + O(h^3)$.

(ii) $\dfrac{-u(x+2h) + 4u(x+h) - 3u(x)}{2h} = u'(x) + cu'''(x)\,h^2 + O(h^3)$.

1.3. Use Theorem 1.1 to show that if $f \in C^2[x, x+\delta]$, then

$$\left| f'(x) - \frac{f(x+\delta) - f(x)}{\delta} \right| \leqslant \frac{\delta}{2} \max_{x \leqslant y \leqslant x+\delta} |f''(y)|,$$

and, similarly, if $f \in C^2[x-h, x]$, then

$$\left| f'(x) - \frac{f(x) - f(x-\delta)}{\delta} \right| \leqslant \frac{\delta}{2} \max_{x-\delta \leqslant y \leqslant x} |f''(y)|,$$

Chapter 2

Finite differences for 1D stationary models

Our aim in this chapter is to develop and analyze finite difference methods for solving *two-point boundary-value problems*, more precisely, *second-order ordinary* differential equations subject to two boundary conditions. The numerical approximation of the exact solution u is computed at a set of grid points in the interval of solution.

In the next section, we will focus mostly on a simple model problem.

2.1 Finite differences for a simple model

Consider the following model:

$$-u''(x) = f(x) \quad \text{for } 0 < x < L, \quad \text{with } u(0) = \gamma_0 \text{ and } u(L) = \gamma_L, \tag{2.1}$$

in which the *source term* $f(x)$ and *boundary data* γ_0, γ_L are given, and we seek the *unknown solution* $u(x)$. The simplest physical interpretation of (2.1) is as a steady-state heat equation, so that $u(x)$ is the temperature at x and $f(x)$ gives the density of heat sources.

For a constant source term $f(x) = c$, the solution is given by

$$u(x) = \frac{1}{L}\big((L-x)\gamma_0 + x\gamma_L\big) + \frac{c}{2}x(L-x) \quad \text{for } 0 \leqslant x \leqslant L; \tag{2.2}$$

For a general f,

$$u(x) = \frac{L-x}{L}\left(\gamma_0 + \int_0^x y f(y)\,dy\right) + \frac{x}{L}\left(\gamma_L + \int_x^L (L-y) f(y)\,dy\right) \quad \text{for } 0 \leqslant x \leqslant L. \tag{2.3}$$

To set up a finite difference scheme for (2.1), we choose a positive integer P and define a uniform grid on $[0, L]$,

$$x_i = i\,h \quad \text{for } 0 \leqslant i \leqslant P, \quad \text{where } h = \frac{L}{P}.$$

In this way, we divide $[0, L]$ into P subintervals, namely $[x_{i-1}, x_i]$ for $1 \leqslant i \leqslant P$, each of length h. The finite difference solution consists of $P+1$ numbers $U_0, U_1, \ldots, U_P$ that approximate $u(x)$ at the $P+1$ grid points, that is,

$$U_i \approx u(x_i) \quad \text{for } 0 \leqslant i \leqslant P.$$

Noting that $x_{i\pm1} = x_i \pm h$ and using Theorem 1.2, we have

$$\frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} = \frac{u(x_i + h) - 2u(x_i) + u(x_i - h)}{h^2} = u''(x_i) + O(h^2) \tag{2.4}$$

which suggests the following finite difference approximation to the solution u of problem (2.1).

$$-\frac{U_{i+1}-2U_i+U_{i-1}}{h^2}=f(x_i) \quad \text{for } 1\leqslant i\leqslant P-1. \tag{2.5}$$

We can satisfy the boundary conditions $u(x_0)=u(0)=\gamma_0$ and $u(x_P)=u(L)=\gamma_L$ exactly, by putting

$$U_0=\gamma_0 \quad \text{and} \quad U_P=\gamma_L.$$

When $i=1$ in (2.5) we move $U_0=\gamma_0$ to the right-hand side,

$$\frac{2U_1-U_2}{h^2}=f(x_1)+\frac{\gamma_0}{h^2},$$

and similarly when $i=P-1$ we move $U_P=\gamma_L$ to the right-hand side,

$$\frac{-U_{P-2}+2U_{P-1}}{h^2}=f(x_{P-1})+\frac{\gamma_L}{h^2}.$$

2.1.1 Implementations

For computation purposes, we write the above finite difference scheme in a matrix form as:

$$\frac{1}{h^2}\mathbf{A}\mathbf{U}=\mathbf{f}+\frac{1}{h^2}\mathbf{g},$$

where the $(P-1)$-by-$(P-1)$ matrix $\mathbf{A}$ is defined by

$$\mathbf{A}=\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}, \quad \mathbf{U}=\begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_{P-2} \\ U_{P-1} \end{bmatrix}, \quad \mathbf{f}=\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{P-2} \\ f_{P-1} \end{bmatrix}, \quad \mathbf{g}=\begin{bmatrix} \gamma_0 \\ 0 \\ \vdots \\ 0 \\ \gamma_L \end{bmatrix},$$

with $f_i=f(x_i)$. It is clear that the tridiagonal matrix $\mathbf{A}$ is diagonally dominant, that is, $|a_{i,i}|\geqslant\sum_{\substack{j=1\\ j\neq i}}^{P-1}|a_{i,j}|$, and symmetric. Moreover, $\mathbf{A}$ is positive definite, $V^T\mathbf{A}V>0$ for any non-zero column vector $V\in\mathbb{R}^{P-1}$. This can be easily observed after noting that $V^T\mathbf{A}V=v_1^2+\sum_{i=1}^{P-2}(v_{i+1}-v_i)^2+v_{P-1}^2$, where $v_1,v_2,\cdots,v_{P-1}$ are the entries of the column vector V. Since the matrix $\mathbf{A}$ is tridiagonal, symmetric and positive definite, it can be decomposed as:

$$\mathbf{A}=\mathbf{L}\mathbf{D}\mathbf{L}^T, \tag{2.6}$$

where the $(P-1)$-by-$(P-1)$ matrices $\mathbf{L}$ and $\mathbf{D}$ of the form

$$\mathbf{L}=\begin{bmatrix} 1 & & & & & \\ \ell_1 & 1 & & & & \\ & \ell_2 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & \ell_{P-3} & 1 & \\ & & & & \ell_{P-2} & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{D}=\begin{bmatrix} d_1 & & & & \\ & d_2 & & & \\ & & \ddots & & \\ & & & d_4 & \\ & & & & d_5 \end{bmatrix}. \tag{2.7}$$

Given a right-hand side vector $\mathbf{b}$, we can solve the linear system $\mathbf{A}\mathbf{x}=\mathbf{b}$ by solving in sequence the three linear systems

$$\mathbf{L}z=\mathbf{b}, \qquad \mathbf{D}\mathbf{y}=z, \qquad \mathbf{L}^T\mathbf{x}=\mathbf{y}, \tag{2.8}$$

because it will then follow that

$$\mathbf{A}\boldsymbol{x} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathsf{T}}\boldsymbol{x} = \mathbf{L}\mathbf{D}\boldsymbol{y} = \mathbf{L}\boldsymbol{z} = \mathbf{b}.$$

Since $\mathbf{L}$ is *lower triangular* and $\mathbf{D}$ is *diagonal*, we can easily compute first $\boldsymbol{z}$, then $\boldsymbol{y}$ and finally $\boldsymbol{x}$.

In the general $n \times n$ case,

$$\mathbf{A} = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix}, \tag{2.9}$$

Algorithm 1 computes the factorization (2.6) and Algorithm 2 uses this factorization to solve the linear system $\mathbf{A}\boldsymbol{x} = \mathbf{b}$.

Algorithm 1 Compute the factorization (2.6).

Require: $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]$ is the main diagonal of $\mathbf{A}$.
Require: $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_{n-1}]$ is the off-diagonal of $\mathbf{A}$.

function FACTORIZE($\boldsymbol{\alpha}, \boldsymbol{\beta}$)
  Allocate storage for $\mathbf{d} = [d_1, d_2, \ldots, d_n]$ and $\boldsymbol{\ell} = [\ell_1, \ell_2, \ldots, \ell_{n-1}]$
  $d_1 = \alpha_1$
  for $j = 1 : n-1$ **do**
    $\ell_j = \beta_j / d_j$
    $d_{j+1} = \alpha_{j+1} - \ell_j^2 d_j$
  end for
  return $\mathbf{d}$, $\boldsymbol{\ell}$
end function

Once d_j has been computed, the value of α_j is never used in subsequent steps of Algorithm 1. Similarly, once ℓ_j has been computed, the value of β_j is never used subsequently.

For a non-singular n-by-n matrix A, the algebraic linear system $AX = B$ can be solved by direct Gaussian elimination where the cost is $2(n^3 - n)/3 + n^2$ flops (floating point operations). Noting that the flop counts grow with the increase in the operation complexity. Direct Gaussian elimination might not be a practical choice when n is large which is most likely the case, for example, when dealing with three-dimensional time-dependent models. Thus, efficient solvers of the system $AX = B$ are needed. To this end, if the matrix A possesses special structure such as sparsity, then, (1) Cholesky factorization and (2) QR factorization can be two useful alternatives of the direct Gaussian elimination method. To use (1), A must be symmetric and positive-definite, then it can be decomposed uniquely as: $A = R^{\mathsf{T}}R$ (Cholesky factorization) where R is an upper triangular matrix. This requires $n^3/3$ flops. Then split $AX = B$ as: $R^{\mathsf{T}}Y = B$ and $RX = Y$. We solve the first system for Y by forward substitution which requires n^2 flops, and then solve the second system for X by backward substitution also requires n^2 flops. The QR factorization depends on a so-called Householder transformation, it works even when A is not a square matrix. A will be decomposed as, $A = QR$ where A is an m-by-n matrix, Q is an orthogonal m-by-n matrix ($Q^{\mathsf{T}}Q = I_n$), and R is an n-by-n upper triangular matrix. See the book by Golub and Loan, Matrix Computation, for more details about these methods.

Before moving to the finite difference methods for a general boundary value problem, we discuss in details the Cholesky factorization in the next section.

Algorithm 2 Solve a symmetric, tridiagonal linear system $\mathbf{A}\boldsymbol{x} = \mathbf{b}$ given the factorization (2.8).

Require: $\mathbf{b} = [b_1, b_2, \ldots, b_n]$ the right-hand side vector.
Require: $\mathbf{d} = [d_1, d_2, \ldots, d_n]$ the main diagonal of $\mathbf{D}$.
Require: $\boldsymbol{\ell} = [\ell_1, \ell_2, \ldots, \ell_{n-1}]$ the first sub-diagonal of $\mathbf{L}$.

function SOLVE($\mathbf{b}, \mathbf{d}, \boldsymbol{\ell}$)
  Allocate storage for $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$, $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]$ and $\boldsymbol{z} = [z_1, z_2, \ldots, z_n]$.
  $z_1 = b_1$
  for $j = 1 : n-1$ **do**
    $z_{j+1} = b_{j+1} - \ell_j z_j$
  end for
  for $j = 1 : n$ **do**
    $y_j = z_j / d_j$
  end for
  $x_n = y_n$
  for $j = n-1 : -1 : 1$ **do**
    $x_j = y_j - \ell_j x_{j+1}$
  end for
  return $\boldsymbol{x}$
end function

A simple MATLAB code to compute $\mathbf{L}$ and $\mathbf{D}$ with $n = 5$.

```
alpha = [alpha1 alpha2 alpha3 alpha4 alpha5];
beta = [beta1 beta2 beta3 beta4];
 A = diag(alpha) + diag(beta,1) + diag(beta,-1);
                   d(1)=alpha(1);
          for j=1:n-1;
          l(j) = beta(j)/d(j);
          d(j+1) = alpha(j+1)-l(j)^2*d(j);
          end
          D = diag(d);
          L = diag(ones(1,n))  + diag(l,-1);
```

2.2 Band Cholesky factorization

A matrix $\mathbf{A} = [a_{ij}]$ has *upper bandwidth* β if $a_{ij} = 0$ whenever $j - i > \beta$, or equivalently if $\mathbf{A}$ has β non-zero superdiagonals. Similarly, $\mathbf{A}$ has *lower bandwidth* β if $a_{ij} = 0$ whenever $i - j > \beta$, so that there are β non-zero subdiagonals. For example, the following matrix has upper bandwidth 2 and lower bandwidth 3:

$$\begin{bmatrix} 5 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 6 & 0 & 8 & 0 & 0 & 0 \\ 2 & 3 & 9 & 1 & 8 & 0 & 0 \\ -3 & 4 & 7 & 2 & 5 & 8 & 0 \\ 0 & 0 & 2 & 8 & 3 & -7 & 1 \\ 0 & 0 & 4 & -5 & 6 & 9 & 3 \end{bmatrix}.$$

If a matrix is symmetric, then its upper and lower bandwidths must be equal, so we refer to both as just the *bandwidth*.

Let $\mathbf{A}$ be an $n \times n$ symmetric, positive-definite matrix with bandwith β. We will seek a *band Cholesky factorization*

$$\mathbf{A} = \mathbf{R}^T\mathbf{R}, \tag{2.10}$$

where the $n \times n$ matrix $\mathbf{R} = [r_{ij}]$ is upper triangular with upper bandwidth β. By the definition of matrix multiplication,

$$\left(\mathbf{R}^T\mathbf{R}\right)_{ij} = \sum_{k=1}^{n} \left(\mathbf{R}^T\right)_{ik} \mathbf{R}_{kj} = \sum_{k=1}^{n} r_{ki} r_{kj}.$$

Since $\mathbf{R}$ is upper triangular, $r_{ki}r_{kj} = 0$ if $k > i$ or $k > j$, that is, if $k > \min(i, j)$, so in fact

$$\left(\mathbf{R}^T\mathbf{R}\right)_{ij} = \sum_{k=1}^{\min(i,j)} r_{ki} r_{kj}.$$

In addition, $\mathbf{R}$ has upper bandwidth β so $r_{ki}r_{kj} = 0$ if $i > k + \beta$ or $j > k + \beta$. Thus, taking account of symmetry, (2.10) is satisfied iff

$$a_{ij} = \sum_{k=\max(1,j-\beta)}^{i} r_{ki} r_{kj} \quad \text{for } \max(1, j - \beta) \leqslant i \leqslant j \leqslant n.$$

We split the last term off the sum, writing

$$a_{ij} = r_{ii} r_{ij} + \sum_{k=\max(1,j-\beta)}^{i-1} r_{ki} r_{kj} \quad \text{for } \max(1, j - \beta) \leqslant i < j \leqslant n,$$

in the off-diagonal case, and

$$a_{jj} = r_{jj}^2 + \sum_{k=\max(1,j-\beta)}^{j-1} r_{kj}^2 \quad \text{for } 1 \leqslant j \leqslant n,$$

in the diagonal case. Rearranging these equations leads to the formulae

$$r_{ij} = \frac{1}{r_{ii}} \left(a_{ij} - \sum_{k=\max(1,j-\beta)}^{i-1} r_{ki} r_{kj} \right) \quad \text{for } \max(1, j - \beta) \leqslant i \leqslant j \leqslant n,$$

Algorithm 3 Compute the band Cholesky factorization (2.10).

Require: $\mathbf{A} = [a_{ij}]$ is a real, $n \times n$, symmetric positive-definite matrix with bandwidth β.

function FACTORIZE($\mathbf{A}$)
  Allocate storage for the $n \times n$ Cholesky factor $\mathbf{R} = [r_{ij}]$ and initialize to zero.
  for $j = 1 : n$ **do**
    for $i = \max(1, j - \beta) : j$ **do**
      $s = 0$
      for $k = \max(1, j - \beta) : i - 1$ **do**
        $s = s + r_{ki} r_{kj}$
      end for
      $r_{ij} = (a_{ij} - s)/r_{ii}$
    end for
    $s = 0$
    for $k = \max(1, j - \beta) : j - 1$ **do**
      $s = s + r_{kj}^2$
    end for
    if $a_{jj} \leqslant s$ **then**
      Error: $\mathbf{A}$ is not positive-definite.
    end if
    $r_{jj} = \sqrt{a_{jj} - s}$
  end for
  return $\mathbf{R}$
end function

Algorithm 4 Solve the linear system $\mathbf{A}x = \mathbf{b}$ given the band Cholesky factorization (2.10).

Require: $\mathbf{b} = [b_i]_{i=1}^n$ is the right-hand side vector.
Require: $\mathbf{R} = [r_{ij}]_{i,j=1}^n$ is the band Cholesky factor of $\mathbf{A}$, computed via Algorithm 3.

function SOLVE($\mathbf{b}, \mathbf{R}$)
  Allocate storage for $x = [x_i]_{i=1}^n$ and $y = [y_i]_{i=1}^n$.
  for $i = 1 : n$ **do**
    $s = b_i$
    for $k = \max(1, i - \beta) : i - 1$ **do**
      $s = s - r_{ji} y_j$
    end for
    $y_i = s/r_{ii}$
  end for
  for $i = n : -1 : 1$ **do**
    $s = y_i$
    for $j = i + 1 : \min(n, i + \beta)$ **do**
      $s = s - r_{ij} x_j$
    end for
    $x_i = s/r_{ii}$
  end for
  return x
end function

and

$$r_{jj} = \sqrt{a_{jj} - \sum_{k=\max(1,j-\beta)}^{j-1} r_{kj}^2} \quad \text{for } 1 \leqslant j \leqslant n,$$

which yield Algorithm 3.

Once the Cholesky factor $\mathbf{R}$ is known, we can solve a linear system $\mathbf{Ax} = \mathbf{b}$ by first solving the lower triangular system $\mathbf{R}^T\mathbf{y} = \mathbf{b}$, and then solving the upper triangular system $\mathbf{Rx} = \mathbf{y}$, so that

$$\mathbf{Ax} = \mathbf{R}^T\mathbf{Rx} = \mathbf{R}^T\mathbf{y} = \mathbf{b}.$$

Since $r_{ji} = 0$ if $j > i$ or $i - j > s$,

$$(\mathbf{R}^T\mathbf{y})_i = \sum_{j=1}^{n} r_{ji}y_j = \sum_{j=\max(1,i-\beta)}^{i} r_{ji}y_j = r_{ii}y_i + \sum_{j=\max(1,i-\beta)}^{i-1} r_{ji}y_j$$

and so $\mathbf{R}^T\mathbf{y} = \mathbf{b}$ iff

$$y_i = \frac{1}{r_{ii}}\left(b_i - \sum_{j=\max(1,i-\beta)}^{i-1} r_{ji}y_j\right) \quad \text{for } 1 \leqslant i \leqslant n.$$

Similarly, since $r_{ij} = 0$ if $i > j$ or $j - i > \beta$,

$$(\mathbf{Rx})_i = \sum_{j=1}^{n} r_{ij}x_j = \sum_{j=i}^{\min(n,i+\beta)} r_{ij}x_j = r_{ii}x_i + \sum_{j=i+1}^{\min(n,i+\beta)} r_{ij}x_j$$

and so $\mathbf{Rx} = \mathbf{y}$ iff

$$x_i = \frac{1}{r_{ii}}\left(y_i - \sum_{j=i+1}^{\min(n,i+\beta)} r_{ij}x_j\right) \quad \text{for } 1 \leqslant i \leqslant n.$$

These formula lead to Algorithm 4 for solving $\mathbf{Ax} = \mathbf{b}$.

2.3 General two-point boundary-value problem

Now consider a general second-order linear differential operator,

$$\mathcal{L}u = -a(x)u'' + b(x)u' + c(x)u,$$

where, for simplicity, we will assume that the coefficients a, b and c are continuous on $[0, L]$, and that the *leading coefficient* a is *strictly positive* on $[0, L]$, so there is a constant $a_{\min}$ such that

$$a(x) \geqslant a_{\min} > 0 \quad \text{for } 0 \leqslant x \leqslant L. \tag{2.11}$$

The general *two-point boundary-value problem* is to find $u = u(x)$ satisfying

$$\begin{aligned} \mathcal{L}u &= f(x) && \text{for } 0 < x < L, \\ \alpha_0 u' + \beta_0 u &= \gamma_0 && \text{at } x = 0, \\ \alpha_L u' + \beta_L u &= \gamma_L && \text{at } x = L, \end{aligned} \tag{2.12}$$

where, for the boundary conditions to make sense, we assume that at least one of α_0 and β_0 is not zero, and likewise at least one of α_L and β_L is not zero. For simplicity, we will also assume that the function f is continuous on $[0, L]$. The simple problem (2.1) is just the special case

$$a(x) = 1, \quad b(x) = 0, \quad c(x) = 0, \quad \alpha_0 = 0, \quad \beta_0 = 1, \quad \alpha_L = 0, \quad \beta_L = 1.$$

Algorithm 5 Solve in place the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ given the band Cholesky factorization (2.10).

Require: $\mathbf{x} = [b_i]_{i=1}^n$ stores the right-hand side vector.
Require: The $(\beta+1)\times n$ array $\mathbf{R}_{\text{band}} = [r_{ij}^{\text{band}}]$ stores the band Cholesky factor $\mathbf{R} = [r_{ij}]$ of $\mathbf{A}$, as computed via Algorithm 3 so that $r_{ij} = r_{\beta+1+i-j,j}^{\text{band}}$.

```
function SOLVE(x, R_band)
    for i = 1 : n do
        s = x_i
        for j = max(1, i − β) : i − 1 do
            s = s − r^band_{β+1+j−i,i} x_j
        end for
        x_i = s / r^band_{β+1,i}
    end for
    for i = n : −1 : 1 do
        s = x_i
        for j = i + 1 : min(n, i + β) do
            s = s − r^band_{β+1+i−j,j} x_j
        end for
        x_i = s / r^band_{β+1,i}
    end for
    return x
end function
```

Motivated by the achieved estimate in Theorem 1.3, $u'(x_i)$ is approximated as:

$$\frac{u(x_{i+1}) - u(x_{i-1})}{2h} = \frac{u(x_i + h) - u(x_i - h)}{2h} \approx u'(x_i) \tag{2.13}$$

and hence, the approximations (2.4) suggests the following discrete approximation to $(\mathcal{L}u)(x_i)$,

$$(\mathcal{L}_h U)_i = -a_i \frac{U_{i+1} - 2U_i + U_{i-1}}{h^2} + b_i \frac{U_{i+1} - U_{i-1}}{2h} + c_i U_i, \tag{2.14}$$

where we have used the abbreviations $a_i = a(x_i)$, $b_i = b(x_i)$ and $c_i = c(x_i)$. The central difference approximation to u' can also be used in the boundary conditions by allowing "ghost" grid points $x_{-1} = -h$ and $x_{P+1} = L + h$ lying just outside the interval $[0, L]$, so that

$$u'(0) \approx \frac{U_1 - U_{-1}}{2h} \quad \text{and} \quad u'(L) \approx \frac{U_{P+1} - U_{P-1}}{2h}. \tag{2.15}$$

Four cases can occur.

1. If $\alpha_0 = 0$ and $\alpha_L = 0$, then we require

   $$(\mathcal{L}_h U)_i = f_i \quad \text{for } 1 \leqslant i \leqslant P-1, \quad \text{with } \beta_0 U_0 = \gamma_0 \text{ and } \beta_L U_P = \gamma_L.$$

   Eliminating $U_0 = \gamma_0/\beta_0$ and $U_P = \gamma_L/\beta_L$ leads to a linear system for $U_1, U_2, \ldots, U_{P-1}$.

2. If $\alpha_0 \neq 0$ and $\alpha_L = 0$, then we require

   $$(\mathcal{L}_h U)_i = f_i \quad \text{for } 0 \leqslant i \leqslant P-1, \quad \text{with } \alpha_0 \frac{U_1 - U_{-1}}{2h} + \beta_0 U_0 = \gamma_0 \text{ and } \beta_L U_P = \gamma_L.$$

   Eliminating $U_{-1} = U_1 + 2h(\beta_0 U_0 - \gamma_0)/\alpha_0$ and $U_P = \gamma_L/\beta_L$ leads to a linear system for $U_0, U_1, \ldots, U_{P-1}$.

3. If $\alpha_0 = 0$ and $\alpha_L \neq 0$, then we require

$$(\mathcal{L}_h U)_i = f_i \quad \text{for } 1 \leqslant i \leqslant P, \quad \text{with } \beta_0 U_0 = \gamma_0 \text{ and } \alpha_L \frac{U_{P+1} - U_{P-1}}{2h} + \beta_L U_P = \gamma_L.$$

Eliminating $U_0 = \gamma_0/\beta_0$ and $U_{P+1} = U_{P-1} + 2h(\gamma_L - \beta_L U_P)/\alpha_L$ leads to a linear system for $U_1, U_2, \ldots, U_P$.

4. If $\alpha_0 \neq 0$ and $\alpha_L \neq 0$, then we require

$$(\mathcal{L}_h U)_i = f_i \quad \text{for } 0 \leqslant i \leqslant P,$$

with

$$\alpha_0 \frac{U_1 - U_{-1}}{2h} + \beta_0 U_0 = \gamma_0 \quad \text{and} \quad \alpha_L \frac{U_{P+1} - U_{P-1}}{2h} + \beta_L U_P = \gamma_L.$$

Eliminating U_{-1} and U_{P+1} leads to a linear system for $U_0, U_1, \ldots, U_P$.

In case 1, the first equation is

$$-a_1 \frac{U_2 - 2U_1 + U_0}{h^2} + b_1 \frac{U_2 - U_0}{2h} + c_1 U_1 = f_1$$

so $U_0 = \gamma_0/\beta_0$ gives

$$a_1 \frac{2U_1 - U_2}{h^2} + b_1 \frac{U_2}{2h} + c_1 U_1 = f_1 + \left(\frac{a_1}{h^2} + \frac{b_1}{2h}\right)\frac{\gamma_0}{\beta_0}.$$

Similarly, the last equation is

$$-a_{P-1} \frac{U_P - 2U_{P-1} + U_{P-2}}{h^2} + b_{P-1} \frac{U_P - U_{P-2}}{2h} + c_{P-1} U_{P-1} = f_{P-1},$$

so $U_P = \gamma_L/\beta_L$ gives

$$a_{P-1} \frac{-U_{P-2} + 2U_{P-1}}{h^2} - b_{P-1} \frac{U_{P-2}}{2h} + c_{P-1} U_{P-1} = f_{P-1} + \left(\frac{a_{P-1}}{h^2} - \frac{b_{P-1}}{2h}\right)\frac{\gamma_L}{\beta_L}.$$

For example, if $P = 6$ we obtain a 5×5 linear system

$$\mathbf{A}\mathbf{U} + \mathbf{B}\mathbf{U} + \mathbf{C}\mathbf{U} = \mathbf{f} + \mathbf{g}, \tag{2.16}$$

where

$$\mathbf{A} = \frac{1}{h^2}\begin{bmatrix} 2a_1 & -a_1 & & & \\ -a_2 & 2a_2 & -a_2 & & \\ & -a_3 & 2a_3 & -a_3 & \\ & & -a_4 & 2a_4 & -a_4 \\ & & & -a_5 & 2a_5 \end{bmatrix}, \qquad \mathbf{B} = \frac{1}{2h}\begin{bmatrix} 0 & b_1 & & & \\ -b_2 & 0 & b_2 & & \\ & -b_3 & 0 & b_3 & \\ & & -b_4 & 0 & b_4 \\ & & & -b_5 & 0 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} c_1 & & & & \\ & c_2 & & & \\ & & c_3 & & \\ & & & c_4 & \\ & & & & c_5 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} g_1 \\ 0 \\ 0 \\ 0 \\ g_5 \end{bmatrix},$$

where

$$g_1 = \left(\frac{a_1}{h^2} + \frac{b_1}{2h}\right)\frac{\gamma_0}{\beta_0} \quad \text{and} \quad g_5 = \left(\frac{a_{P-1}}{h^2} - \frac{b_{P-1}}{2h}\right)\frac{\gamma_L}{\beta_L}.$$

In case 4, the first equation is

$$-a_0 \frac{U_1 - 2U_0 + U_{-1}}{h^2} + b_0 \frac{U_1 - U_{-1}}{2h} + c_0 U_0 = f_0$$

so $U_{-1} = U_1 + 2h(\beta_0 U_0 - \gamma_0)/\alpha_0$ gives

$$-2a_0 \frac{U_1 - U_0}{h^2} + c_0 U_0 - \frac{\beta_0(2a_0 + b_0 h)}{\alpha_0 h} U_0 = f_0 - \frac{2a_0 + b_0 h}{\alpha_0 h} \gamma_0.$$

Similarly, the last equation is

$$-a_P \frac{U_{P+1} - 2U_P + U_{P-1}}{h^2} + b_P \frac{U_{P+1} - U_{P-1}}{2h} + c_P U_P = f_P,$$

so $U_{P+1} = U_{P-1} + 2h(\gamma_L - \beta_L U_P)/\alpha_L$ gives

$$-2a_P \frac{-U_P + U_{P-1}}{h^2} + c_P U_P + \frac{\beta_L(2a_P - b_P h)}{\alpha_L h} U_P = f_P + \frac{2a_P - b_P h}{\alpha_L h} \gamma_L.$$

For example, if $P = 4$ then we obtain a 5×5 linear system of the form (2.16) but now

$$\mathbf{A} = \frac{1}{h^2} \begin{bmatrix} 2a_0(1 - \beta_0 h/\alpha_0) & -2a_0 & & & \\ -a_1 & 2a_1 & -a_1 & & \\ & -a_2 & 2a_2 & -a_2 & \\ & & -a_3 & 2a_3 & -a_3 \\ & & & -2a_4 & 2a_4(1 + \beta_L h/\alpha_L) \end{bmatrix},$$

$$\mathbf{B} = \frac{1}{2h} \begin{bmatrix} -2b_0\beta_0 h/\alpha_0 & & & & \\ -b_1 & 0 & b_1 & & \\ & -b_2 & 0 & b_2 & \\ & & -b_3 & 0 & b_3 \\ & & & & -2b_P\beta_L h/\alpha_L \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} c_0 & & & & \\ & c_1 & & & \\ & & c_2 & & \\ & & & c_3 & \\ & & & & c_4 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} g_0 \\ 0 \\ 0 \\ 0 \\ g_4 \end{bmatrix},$$

where

$$g_0 = -\frac{2a_0 + b_0 h}{\alpha_0 h} \gamma_0 \quad \text{and} \quad g_4 = \frac{2a_4 - b_4 h}{\alpha_L h} \gamma_L.$$

The procedure in cases 2 and 3 should now be clear.

2.4 Maximum principle

It will be convenient to use the standard notation

$$\|f\|_\infty = \max_{0 \leqslant x \leqslant L} |f(x)|$$

for the *maximum norm* of any (continuous) function f over the interval $[0, L]$, and to let

$$u^+(x) = \max\{u(x), 0\} \quad \text{and} \quad u^-(x) = \min\{u(x), 0\}.$$

We begin by proving a *maximum principle* that follows from our *ellipticity* assumption (2.11) and simple calculus.

Lemma 2.1. *Assume that* $u \in C^2(0,L)$. *If* $c \geqslant 0$ *and* $\mathcal{L}u < 0$ *on* $(0,L)$, *then* u *cannot attain a non-negative local maximum in* $(0,L)$.

Proof. Suppose for a contradiction that there exist $x_0 \in (0,L)$ and $\delta > 0$ such that

$$u(x_0) \geqslant 0 \quad \text{and} \quad u(x) \leqslant u(x_0) \text{ for } x \in (x_0-\delta, x_0+\delta) \subseteq (0,L).$$

Since u has an interior local maximum at x_0, it follows that $u'(x_0) = 0$ and $u''(x_0) \leqslant 0$ so

$$(\mathcal{L}u)(x_0) = -a(x_0)u''(x_0) + c(x_0)u(x_0) \geqslant 0,$$

contradicting our assumption that $\mathcal{L}u < 0$ on $(0,L)$. □

Theorem 2.2. *Assume that* $u \in C[0,L] \cap C^2(0,L)$. *If* $c \geqslant 0$ *and* $\mathcal{L}u \leqslant 0$ *on* $(0,L)$, *then*

$$u(x) \leqslant \max\{u^+(0), u^+(L)\} \quad \textit{for } 0 < x < L.$$

Proof. Let $\epsilon > 0$ and $\mu > 0$, and put $w(x) = u(x) + \epsilon e^{\mu x}$. Since

$$(\mathcal{L}w)(x) = (\mathcal{L}u)(x) + \epsilon\big[-a(x)\mu^2 + b(x)\mu + c(x)\big]e^{\mu x} \leqslant -\epsilon\big[a_{\min}\mu^2 - \mu\|b\|_\infty - \|c\|_\infty\big]e^{\mu x}$$

by choosing μ sufficiently large we can ensure that $\mathcal{L}w < 0$ on $(0,L)$. By Lemma 2.1, the function w cannot attain a non-negative local maximum in $(0,L)$, implying that

$$u(x) \leqslant w(x) \leqslant w^+(x) \leqslant \max\{w^+(0), w^+(L)\} \leqslant \max\{u^+(0)+\epsilon, u^+(L)+\epsilon e^{\mu L}\}$$

for $0 < x < L$. Since this inequality holds for any $\epsilon > 0$, the result follows. □

Theorem 2.3. *Assume that* $u \in C[0,L] \cap C^2(0,L)$. *If* $c \geqslant 0$ *and* $\mathcal{L}u = f$ *on* $(0,L)$, *then*

$$\max_{[0,L]} u \leqslant \max\{u^+(0), u^+(L)\} + \cosh(\mu L/2) \max_{[0,L]} f^+$$

and

$$\min_{[0,L]} u \geqslant \min\{u^-(0), u^-(L)\} + \cosh(\mu L/2) \min_{[0,L]} f^-,$$

where $\mu = \max\big\{1, (1+\|b\|_\infty)/a_{\min}\big\}$.

Proof. Let

$$w(x) = \max\{u^+(0), u^+(L)\} + v(x) \max_{[0,L]} f^+ \quad \text{where} \quad v(x) = \cosh(\mu L/2) - \cosh\mu(x - L/2),$$

and observe that $v \geqslant 0$ on $[0,L]$ with

$$\begin{aligned}(\mathcal{L}v)(x) &= a(x)\mu^2 \cosh\mu(x-L/2) - b(x)\mu\sinh\mu(x-L/2) \\ &\quad + c(x)\big(\max\{u^+(0), u^+(L)\} + v(x)\big) \\ &\geqslant \big(a_{\min}\mu^2 - \|b\|_\infty\mu\big)\cosh\mu(x-L/2) \geqslant (a_{\min}\mu - \|b\|_\infty)\mu \geqslant 1\end{aligned}$$

for $0 < x < L$, so

$$\mathcal{L}(u-w) = f - c(x)\max\{u^+(0), u^+(L)\} - (\mathcal{L}v)\max_{[0,L]} f^+ \leqslant 0 \quad \text{on } (0,L).$$

By Theorem 2.2,

$$u(x) - w(x) \leqslant \max\{(u-w)^+(0), (u-w)^+(L)\} \quad \text{for } 0 < x < L,$$

and since $v(0) = 0 = v(L)$ we see that

$$(u - w)(0) = u(0) - \max\{u^+(0), u^+(L)\} \leqslant 0$$

and

$$(u - w)(L) = u(L) - \max\{u^+(0), u^+(L)\} \leqslant 0.$$

Thus, $u - w \leqslant 0$ on $(0, L)$, and therefore

$$\max_{[0,L]} u \leqslant \max_{[0,L]} w(x) = w(L/2) = \max\{u^+(0), u^+(L)\} + v(L/2) \max_{[0,L]} f,$$

proving the first inequality. The second follows because $\mathcal{L}(-u) = -f$, $(-u)^+ = -u^-$ and $(-f)^+ = -f^-$. □

Now consider the two-point boundary-value problem (2.12) in the case when $\alpha_0 = 0 = \alpha_L$ and $\beta_0 = 1 = \beta_L$:

$$\mathcal{L}u = f \quad \text{on } (0, L), \quad \text{with } u(0) = \gamma_0 \text{ and } u(L) = \gamma_L. \tag{2.17}$$

As an immediate consequence of Theorem 2.3 we have the following *a priori* estimate, which serves to bound the solution u in terms of the data f, γ_0 and γ_L.

Theorem 2.4. *If $c \geqslant 0$ on $(0, L)$, then any solution u of (2.17) satisfies*

$$\|u\|_\infty \leqslant \max\{|\gamma_0|, |\gamma_L|\} + \cosh(\mu L/2)\|f\|_\infty.$$

Using Theorem 2.4 it can be shown that the two-point boundary-value problem (2.17) is *well-posed*, that is, the problem in (2.17) has a unique solution for each choice of the data f, γ_0 and γ_L, and small changes in these data lead to only small changes in the solution.

To explain the second part, suppose that we perturb the data to $\tilde{f}$, $\tilde{\gamma}_0$ and $\tilde{\gamma}_L$, and let $\tilde{u}$ denote the solution of the resulting perturbed problem,

$$\mathcal{L}\tilde{u} = \tilde{f} \quad \text{on } (0, L), \quad \text{with } \tilde{u}(0) = \tilde{\gamma}_0 \text{ and } \tilde{u}(L) = \tilde{\gamma}_L.$$

If we denote the changes in the solution and the data by

$$\delta u = \tilde{u} - u, \quad \delta f = \tilde{f} - f, \quad \delta\gamma_0 = \tilde{\gamma}_0 - \gamma_0, \quad \delta\gamma_L = \tilde{\gamma}_L - \gamma_L,$$

then we need to show that δu is small whenever δf, $\delta\gamma_0$ and $\delta\gamma_L$ are all small.

Theorem 2.5. *If $c \geqslant 0$ on $(0, L)$, then (2.17) has a unique solution $u \in C[0, L] \cap C^2(0, L)$. Moreover, when the problem is perturbed as described above,*

$$\|\delta u\|_\infty \leqslant \max\{|\delta\gamma_0|, |\delta\gamma_L|\} + \cosh(\mu L/2)\|\delta f\|_\infty.$$

Proof. We will not prove the hard part, namely existence. Since the $\mathcal{L}$ is linear, the proof of uniqueness follows easily from Theorem 2.4. In fact, suppose that u_1 and u_2 are solutions, that is,

$$\mathcal{L}u_1 = f = \mathcal{L}u_2 \quad \text{on } (0, L), \quad \text{with } u_1(0) = \gamma_0 = u_2(0) \text{ and } u_1(L) = \gamma_L = u_2(L).$$

The difference $v = u_1 - u_2$ satisfies

$$\mathcal{L}v = \mathcal{L}(u_1 - u_2) = \mathcal{L}u_1 - \mathcal{L}u_2 = f - f = 0 \quad \text{on } (0, L),$$

with $v(0) = u_1(0) - u_2(0) = \gamma_0 - \gamma_0 = 0$ and $v(L) = u_1(L) - u_2(L) = \gamma_L - \gamma_L = 0$, so $\|v\|_\infty \leqslant 0$ by Theorem 2.4, which means that $u_1 = u_2$ on $[0, L]$.

Similarly, since $\mathcal{L}$ is linear,

$$\mathcal{L}\delta u = \delta f \quad \text{on } (0, L), \quad \text{with } \delta u(0) = \delta\gamma_0 \text{ and } \delta u(L) = \delta\gamma_L,$$

and Theorem 2.4 implies the desired estimate for $\|\delta u\|_\infty$. □

2.5 Discrete maximum principle

We will now establish a maximum principle for our finite difference approximation to the two-point boundary-value problem (2.12) in the case

$$b(x) = 0, \qquad \alpha_0 = 0, \quad \beta_0 = 1, \quad \alpha_L = 0, \quad \beta_L = 1. \tag{2.18}$$

Thus,

$$(\mathcal{L}u)(x) = -a(x)u''(x) + c(x)u(x) = f(x) \quad \text{for } 0 < x < L, \tag{2.19}$$

with $u(0) = \gamma_0$ and $u(L) = \gamma_L$. Likewise,

$$(\mathcal{L}_h U)_i = -a_i \frac{U_{i+1} - 2U_i + U_{i-1}}{h^2} + c_i U_i \quad \text{for } 1 \leqslant i \leqslant P-1,$$

and our finite difference method is (case 1 in section 2.3)

$$(\mathcal{L}_h U)_i = f_i \quad \text{for } 1 \leqslant i \leqslant P-1, \quad \text{with } U_0 = \gamma_0 \text{ and } U_P = \gamma_L. \tag{2.20}$$

The following discrete analogue of Theorem 2.1 holds.

Lemma 2.6. *Assume* (2.18). *If* $c_i \geqslant 0$ *and* $(\mathcal{L}_h U)_i < 0$ *for* $1 \leqslant i \leqslant P-1$, *then the is no* i^* *in the range* $1 \leqslant i^* \leqslant P-1$ *for which*

$$U_{i^*} \geqslant 0, \qquad U_{i^*-1} \leqslant U_{i^*} \qquad \textit{and} \qquad U_{i^*+1} \leqslant U_{i^*}.$$

Proof. If such a i^* exists, then $U_{i^*+1} + U_{i^*-1} \leqslant 2U_{i^*}$ so

$$(\mathcal{L}_h U)_{i^*} = a_{i^*} \frac{-U_{i^*+1} + 2U_{i^*} - U_{i^*-1}}{h^2} + c_{i^*} U_{i^*} \geqslant 0,$$

contradicting the second hypothesis of the lemma. □

A discrete analogue of Theorem 2.2 then follows.

Theorem 2.7. *Assume* (2.18). *If* $c_i \geqslant 0$ *and* $(\mathcal{L}_h U)_i \leqslant 0$ *for* $1 \leqslant i \leqslant P-1$, *then*

$$U_i \leqslant \max\{U_0^+, U_P^+\} \quad \textit{for } 1 \leqslant i \leqslant P-1.$$

Proof. Let $\epsilon > 0$ and $\mu > 0$, and put $W_i = U_i + \epsilon e^{\mu x_i}$. We see from (1.4) that the function $g(x) = e^{\mu x}$ satisfies

$$\frac{g(x_{i+1}) - 2g(x_i) + g(x_{i-1})}{h^2} - g''(x_i) = \frac{1}{h^2}\big[(R_3 g)(x_i, h) + (R_3 g)(x_i, -h)\big],$$

where

$$(R_3 g)(x_i, h) = \frac{1}{4!}\int_{x_i}^{x_{i+1}} (x_{i+1} - y)^3 g^{(4)}(y)\, dy,$$
$$(R_3 g)(x_i, -h) = \frac{1}{4!}\int_{x_{i-1}}^{x_i} (y - x_{i-1})^3 g^{(4)}(y)\, dy.$$

Since $g^{(4)}(y) = \mu^4 e^{\mu y} \geqslant 0$ for all y, it follows that $(R_3 g)(x_i, \pm h) \geqslant 0$ and thus

$$\frac{g(x_{i+1}) - 2g(x_i) + g(x_{i-1})}{h^2} \geqslant g''(x_i) = \mu^2 e^{\mu x_i}.$$

Hence,

$$\begin{aligned}(\mathcal{L}_h W)_i &= (\mathcal{L}_h U)_i + \epsilon\left(-a_i\,\frac{g(x_{i+1}) - 2g(x_i) + g(x_{i-1})}{h^2} + c_i g(x_i)\right)\\ &\leqslant (\mathcal{L}_h U)_i + \epsilon\left(-a_i\mu^2 e^{\mu x_i} + c_i e^{\mu x_i}\right) \leqslant 0 - \epsilon(a_{\min}\mu^2 - \|c\|_\infty)e^{\mu x_i},\end{aligned}$$

and by choosing $\mu^2 > \|c\|_\infty/a_{\min}$ we can ensure that $(\mathcal{L}_h U)_i < 0$ for $1 \leqslant i \leqslant P-1$. By Lemma 2.6, we conclude that

$$U_i \leqslant W_i \leqslant W_i^+ \leqslant \max\{W_0^+, W_P^+\} = \max\{U_0^+ + \epsilon, U_P^+ + \epsilon e^{\mu L}\}$$

for $1 \leqslant i \leqslant P-1$. Since this inequality holds for any $\epsilon > 0$, the result follows. □

Next is a discrete version of Theorem 2.3.

Theorem 2.8. *Assume* (2.18). *If* $c_i \geqslant 0$ *and* $(\mathcal{L}_h U)_i = f_i$ *for* $1 \leqslant p \leqslant L$, *then*

$$\max_{0\leqslant i\leqslant P} U_i \leqslant \max\{U_0^+, U_P^+\} + \frac{L^2}{8a_{\min}} \max_{1\leqslant q\leqslant P-1} f_q^+$$

and

$$\min_{0\leqslant i\leqslant P} U_i \geqslant \min\{U_0^+, U_P^+\} + \frac{L^2}{8a_{\min}} \max_{1\leqslant q\leqslant P-1} f_q^-$$

Proof. Let

$$W_i = \max\{U_0^+, U_P^+\} + V_i \max_{1\leqslant q\leqslant P-1} f_q^+ \quad \text{where} \quad V_i = \frac{x_i(L-x_i)}{2a_{\min}}.$$

Since the second-order central difference formula is exact for a quadratic polynomial,

$$\frac{V_{i+1} - 2V_i + V_{i-1}}{h^2} = \frac{-1}{a_{\min}}$$

and thus, noting that $V_i \geqslant 0$, we conclude that

$$(\mathcal{L}_h V)_i = \frac{a_i}{a_{\min}} + c_i V_i \geqslant 1 \quad \text{for } 1 \leqslant i \leqslant P-1.$$

It follows that

$$(\mathcal{L}_h W)_i = \max\{U_0^+, U_P^+\}(\mathcal{L}_h 1)_i + \left(\max_{1\leqslant q\leqslant P-1} f_q^+\right)(\mathcal{L}_h V)_i \geqslant \max_{1\leqslant q\leqslant P-1} f_q^+$$

and so

$$\big(\mathcal{L}_h(U-W)\big)_i = f_i - (\mathcal{L}_h W)_i \leqslant f_i - \max_{1\leqslant q\leqslant P-1} f_q^+ \leqslant 0 \quad \text{for } 1 \leqslant i \leqslant P-1,$$

with

$$(U-W)_0 = U_0 - \max\{U_0^+, U_P^+\} \leqslant 0 \quad \text{and} \quad (U-W)_P = U_P - \max\{U_0^+, U_P^+\} \leqslant 0.$$

By Theorem 2.7,

$$U_i - W_i \leqslant \max\{(U-W)_0^+, (U-W)_P^+\} \leqslant 0 \quad \text{for } 1 \leqslant i \leqslant P-1,$$

and the first inequality follows because

$$U_i \leqslant W_i \leqslant \max\{U_0^+, U_P^+\} + \left(\max_{0\leqslant x\leqslant L} \frac{x(L-x)}{2a_{\min}}\right)\left(\max_{1\leqslant q\leqslant P-1} f_q^+\right).$$

The second follows because $\big(\mathcal{L}_h(-U)\big)_i = -f_i$, $(-U)_i^+ = -U_i^-$ and $(-f)_i^+ = -f_i^-$. □

Our final result for this section is a discrete version of Theorem 2.4.

Theorem 2.9. *Assume* (2.18). *If* $c \geqslant 0$ *on* $(0, L)$, *then the finite difference equations* (2.20) *have a unique solution* U_i, *and*

$$\max_{0\leqslant i\leqslant P} |U_i| \leqslant \max\{|\gamma_0|, |\gamma_L|\} + \frac{L^2}{8a_{\min}} \max_{1\leqslant q\leqslant P-1} |f_q|.$$

Proof. The *a priori* estimate for U_i follows at once from Theorem 2.8. To prove existence and uniqueness, we write the finite difference equations in matrix form, as in (2.16) except that now $\mathbf{B} = 0$ so

$$(\mathbf{A} + \mathbf{C})\mathbf{U} = \mathbf{f} + \mathbf{g}.$$

The *a priori* estimate shows that if $f_i = 0$ for $1 \leqslant p \leqslant P - 1$ and $\gamma_0 = 0 = \gamma_L$, then $U_i = 0$ for $0 \leqslant p \leqslant P$. In other words if $\mathbf{f} = 0 = \mathbf{g}$, then $\mathbf{U} = 0$. Thus, the homogeneous linear system admits only the trivial solution, which implies that the matrix $\mathbf{A} + \mathbf{C}$ is non-singular and so the linear system is uniquely solvable for any f_i, γ_0 and γ_L. □

2.6 Stability and error bounds

The finite difference method defined in section 2.3 is said to be *stable* if a unique solution U_i exists for any choice of the data f_i, γ_0 and γ_L, and if there is a constant C — independent of f_i, γ_0, γ_L and h — such that

$$\max_{0\leqslant i\leqslant P} |U_i| \leqslant C\Big(|\gamma_0| + |\gamma_L| + \max_{1\leqslant i\leqslant P-1} |f_i|\Big).$$

For example, by Theorem 2.9, the conditions (2.18) are sufficient to ensure stability.

Suppose $\alpha_0 = 0 = \alpha_L$ (that is, case 1). We define the *local truncation error* T_i by

$$T_i = f_i - (\mathcal{L}_h u)_i = (\mathcal{L}u)_i - (\mathcal{L}_h u)_i,$$

and note that

$$T_i = -a_i\left(u''(x_i) - \frac{u(x_i + h) - 2u(x_i) + u(x_i - h)}{h^2}\right) + b_i\left(u'(x_i) - \frac{u(x_i + h) - u(x_i - h)}{2h}\right)$$

so, by Theorems 1.2 and 1.3,

$$|T_i| \leqslant \left(\frac{|a_i|}{12}\|u^{(4)}\|_\infty + \frac{|b_i|}{6}\|u^{(3)}\|_\infty\right) h^2. \tag{2.21}$$

Since $(\mathcal{L}_h U)_i = f_i = (\mathcal{L}u)_i$ and since the finite difference operator $\mathcal{L}_h$ is linear, it follows that the *solution error*,

$$E_i = U_i - u(x_i)$$

satisfies

$$(\mathcal{L}_h E)_i = (\mathcal{L}_h U)_i - (\mathcal{L}_h u)_i = T_i \quad \text{for } 1 \leqslant i \leqslant P - 1.$$

Moreover, since $U_0 = \gamma_0 = u(x_0)$ and $U_P = \gamma_L = u(x_P)$ we have $E_0 = 0 = E_P$. Therefore, the stability property applies with U_i, f_i, γ_0 and γ_L replaced by E_i, T_i, 0 and 0, respectively, and so

$$\max_{0\leqslant i\leqslant P} |E_i| \leqslant C \max_{1\leqslant i\leqslant P-1} |T_i|.$$

Combining this estimate with (2.21) we obtain the error bound

$$|U_i - u(x_i)| \leqslant C\left(\frac{\|a\|_\infty}{12}\,\|u^{(4)}\|_\infty + \frac{\|b\|_\infty}{6}\,\|u^{(3)}\|_\infty\right) h^2 \quad \text{for } 0 \leqslant i \leqslant P.$$

In the above estimate, the power of h is called the convergence order or the convergence rate. So, our finite difference approximation is second order accurate. To compute the convergence rate numerically, we let $\mathcal{E}(h) = \max|E_i| = Ch^r$, which is the maximum nodal error corresponding to the mesh element of size h. Then, $\mathcal{E}(2h) = \max|E_i| = C(2h)^r$. It follows that we can estimate the value of r by computing the ratio

$$\frac{\mathcal{E}(2h)}{\mathcal{E}(h)} \approx 2^r$$

and taking logarithms (to base 2):

$$r \approx \log_2 \frac{\mathcal{E}(2h)}{\mathcal{E}(h)}.$$

2.7 Exercises

2.1. Choose $f(x) = 5e^{-x}$, $L = 2$, $\gamma_0 = -1$, and $\gamma_L = 5/2$. Then, compute the maximum nodal error from the finite difference approximation scheme in (2.5) and also the illustrate the second order convergence rate.

2.2. Verify that (2.2) satisfies (2.1) if $f(x) = c$.

2.3. By following the steps below, use variation of parameters to verify that (2.3) solves (2.1) for a general $f(x)$. Let

$$u_1(x) = L - x \quad \text{and} \quad u_2(x) = x$$

and write

$$u(x) = v_1(x)u_1(x) + v_2(x)u_2(x).$$

(i) Verify that u_1 and u_2 are solutions of the homogeneous equation $u'' = 0$, and find their Wronskian

$$W = \begin{vmatrix} u_1 & u_2 \\ u_1' & u_2' \end{vmatrix}.$$

(ii) Hence, noting that $u'' = -f(x)$, determine

$$v_1'(x) = -\frac{u_2(x)[-f(x)]}{W(x)} = \frac{u_2(x)f(x)}{W(x)} \quad \text{and} \quad v_2'(x) = \frac{u_1(x)[-f(x)]}{W(x)} = -\frac{u_1(x)f(x)}{W(x)}.$$

(iii) Find v_1 and v_2, using the boundary conditions $u(0) = \gamma_0$ and $u(L) = \gamma_L$ to determine the constants of integration.

2.4. Consider the two-point boundary-value problem

$$-u'' = f(x) \quad \text{for } 0 < x < L, \quad \text{with } -u'(0) = \gamma_0 \text{ and } u'(L) = \gamma_L.$$

(i) Show that if $u(x)$ is a solution then so is $u(x) + C$ for any constant C.

(ii) Show that if $u(x)$ exists, then $\gamma_0 + \gamma_L + \int_0^L f(x)\,dx = 0$.

(iii) Use central difference approximations of the form (2.4) and (2.13) to derive a $(P+1)\times(P+1)$ linear system $\mathbf{AU} = \mathbf{f}+\mathbf{g}$ that yields $U_i \approx u(x_i)$ for $0 \leqslant i \leqslant P$. When $P = 5$ you should obtain

$$\frac{1}{h^2}\begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix}\begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ \frac{1}{2}f_5 \end{bmatrix} + \frac{1}{h}\begin{bmatrix} \gamma_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \gamma_L \end{bmatrix}.$$

(iv) Show that $\mathbf{U} = [U_i]$ is a solution, then so is $[U_i + C]$.

(v) Show that if a solution $\mathbf{U}$ exists, then

$$\gamma_0 + \gamma_L + \left(\tfrac{1}{2}f_0 + f_1 + f_2 + \cdots + f_{P-1} + \tfrac{1}{2}f_P\right)h = 0.$$

(vi) If we interpret the ODE as a steady-state heat equation, what is the physical meaning of the condition in (ii)?

2.5. How can we use the LDL^T factorization of an $n \times n$, symmetric tridiagonal matrix $\mathbf{A}$ to compute its determinant?

2.6. Find the LDL^T factorization of the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 4 & 6 & 0 \\ 0 & 6 & 14 & 6 \\ 0 & 0 & 6 & 22 \end{bmatrix}.$$

2.7. If $\mathbf{A}$ is symmetric tridiagonal matrix, then instead of the factorization (2.6) we can seek a tridiagonal, upper triangular matrix $\mathbf{R}$ such that $\mathbf{A} = \mathbf{R}^T\mathbf{R}$. We call $\mathbf{R}$ the *Cholesky factor* of $\mathbf{A}$.

(i) Show that if a matrix $\mathbf{A}$ has a Cholesky factorization, then $\mathbf{A}$ is necessarily symmetric and positive semidefinite.

(ii) Show that if the Cholesky factor is non-singular, then $\mathbf{A}$ must be *strictly* positive-definite.

(iii) Consider the 5×5 case and denote the entries of $\mathbf{A}$ as in (2.9). Determine a sequence of formulae to compute the entries of

$$\mathbf{R} = \begin{bmatrix} d_1 & u_1 & & & \\ & d_2 & u_2 & & \\ & & d_3 & u_3 & \\ & & & d_4 & u_4 \\ & & & & u_5 \end{bmatrix}.$$

(iv) Hence formulate an algorithm to compute $\mathbf{R}$ in the general, $n \times n$ case.

(v) Formulate an in-place version of the algorithm in part (iv).

(vi) How is the Cholesky factorization used to solve a linear system $\mathbf{Ax} = \mathbf{b}$?

2.8. Suppose that the matrices $\mathbf{A}$ and $\mathbf{B}$ have upper bandwidth $s_{\mathbf{A}}$ and $s_{\mathbf{B}}$, respectively, and that the number of columns of $\mathbf{A}$ equals the number of rows of $\mathbf{B}$. Show that the matrix product $\mathbf{AB}$ has upper bandwidth $\max(s_{\mathbf{A}}, s_{\mathbf{B}})$. State and prove the corresponding result for lower bandwidths.

2.9. Consider the boundary value problem

$$\begin{aligned} -u'' + 2u' - u &= 1 \quad && \text{for } 0 < x < 2, \\ u &= 1 && \text{at } x = 0, \\ u &= -1 && \text{at } x = 2. \end{aligned}$$

(i) Set up a finite difference approximation as explained in lectures with $P = 4$ (so $h = 1/2$).

(ii) Eliminate the variables U_0 and U_4 to obtain a 3×3 linear system.

(iii) Solve the linear system to find U_1, U_2 and U_3. (You can use for favourite software.)

(iv) Find the exact solution $u(x)$, and hence compute the errors $E_i = U_i - u(x_i)$ in your numerical solution.

(v) Compare graphically between the exact and the finite difference solutions when $h = 1/4$.

(vi) Illustrate the second order of accuracy of the finite difference scheme by computing the errors and the convergence rates for $h = 1/4, 1/8, 1/16, 1/32, 1/64$.

2.10. Consider the ODE in *divergence form*:

$$-\frac{d}{dx}\left(a(x)\frac{du}{dx}\right) = f(x) \quad \text{for } 0 < x < L. \tag{2.22}$$

(i) Find $w(x)$ such that

$$\begin{aligned} &\frac{1}{h}\left(a(x+\tfrac{1}{2}h)\frac{u(x+h)-u(x)}{h} - a(x-\tfrac{1}{2}h)\frac{u(x)-u(x-h)}{h}\right) \\ &\qquad = \frac{d}{dx}\left(a(x)\frac{du}{dx}\right) + w(x)h^2 + O(h^4) \quad \text{as } h \to 0. \end{aligned}$$

Hint: for $\delta = \frac{1}{2}h$, let

$$v(x) = a(x)\frac{u(x+\delta)-u(x-\delta)}{2\delta},$$

and so, the left-hand side of the equation in (i) equals

$$\frac{v(x+\delta)-v(x-\delta)}{2\delta}.$$

Now, use the result

$$\frac{v(x+\delta)-v(x-\delta)}{2\delta} = v'(x) + \frac{1}{6}v'''(x)\delta^2 + O(\delta^4) \quad \text{as } h \to 0.$$

(ii) Hence devise a finite difference scheme with $h = L/P$ to solve (2.23) subject to the Dirichlet boundary conditions $u(0) = \gamma_0$ and $u(L) = \gamma_L$.

(iii) Write out the linear system in matrix form when $P = 5$.

2.11. Consider the following steady-state convection-diffusion model:

$$-au''(x) + bu'(x) = 0 \quad \text{for } 0 < x < L, \tag{2.23}$$

with $u(0) = 0$ and $u(L) = 1$, where the diffusivity coefficient a is a strictly positive constant, and the stream velocity b is also a positive constant. The exact solution is

$$u(x) = \frac{e^{cx} - 1}{e^{cL} - 1}, \quad \text{with } c = \frac{b}{a}.$$

(i) Set a finite difference scheme as explained in lectures with $h = \frac{L}{P}$. Then, write your scheme in a matrix form.

(ii) Comment on the behavior of exact and finite difference solutions when c is relatively large.

2.12. Consider the following model:

$$-u''(x) + c(x)u(x) = f(x) \quad \text{for } 0 < x < L, \quad \text{with } u(0) = \gamma_0 \quad \text{and} \quad u(L) = \gamma_L,$$

where c and f are smooth functions with $c(x) \geqslant 0$.

(i) Define a second-order accurate finite difference solution $U_i \approx u(x_i)$ over a uniform mesh of size h.

(ii) State the definition of the truncation error $\mathcal{T}_i$. Show that $|\mathcal{T}_i| = O(h^2)$.

(iii) Assume that the finite difference solution satisfies the following stability property:

$$|U_i| \leqslant C\Big(|\gamma_0| + |\gamma_L| + \max_{1 \leqslant j \leqslant P-1} |f(x_j)|\Big), \quad 1 \leqslant i \leqslant P-1.$$

Prove the uniqueness of the finite difference solution U_i for $1 \leqslant i \leqslant P-1$.

(iv) Show that $|U_i - u(x_i)| \leqslant C\,h^2$ for $1 \leqslant i \leqslant P-1$.

Chapter 3

Finite differences for 2D stationary models

In this chapter, we study the numerical solution of the Dirichlet boundary-value problem for the Poisson equation. Let Ω be a bounded, open subset of $\mathbb{R}^2$, with a piecewise smooth boundary $\Gamma = \partial\Omega$. Given suitable functions $f(x, y)$ and $g(x, y)$, we see $u = u(x, y)$ satisfying

$$\begin{aligned} -\nabla^2 u(x, y) &= f(x, y) \quad \text{for } (x, y) \in \Omega, \\ u(x, y) &= g(x, y) \quad \text{for } (x, y) \in \Gamma. \end{aligned} \tag{3.1}$$

Here, the *Laplacian* is the second-order elliptic differential operator defined by

$$\nabla^2 u = \nabla \cdot (\nabla u) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = u_{xx} + u_{yy}.$$

3.1 Five-point difference scheme

For simplicity, we now restrict our attention to the case when the spatial domain is a rectangle,

$$\Omega = (0, L_x) \times (0, L_y). \tag{3.2}$$

To set up the spatial finite difference grid, we fix positive integers P and Q, define the step sizes

$$h_x = \frac{L_x}{P} \quad \text{and} \quad h_y = \frac{L_y}{Q},$$

and define the grid (nodal) points

$$(x_i, y_j) = (i\,h_x, j\,h_y) \quad \text{for } 0 \leqslant i \leqslant P \text{ and } 0 \leqslant j \leqslant Q. \tag{3.3}$$

Our aim is to compute a finite difference solution $U_{i,j} \approx u_{i,j} := u(x_i, y_j)$ where u is the solution of (3.1).

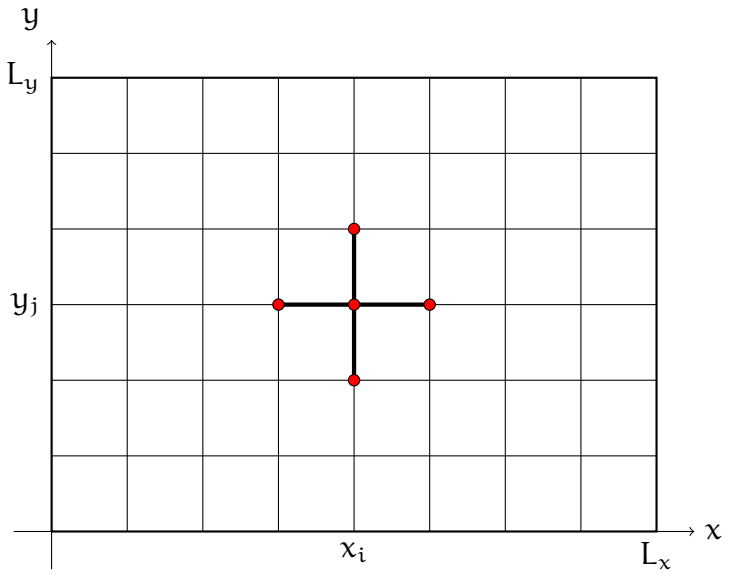
Let δ_x^2 and δ_y^2 denote the second-order, central difference operators in the x- and y-directions, respectively; that is,

$$\delta_x^2 u(x, y) = \frac{u(x + h_x, y) - 2u(x, y) + u(x - h_x, y)}{h_x^2} = u_{xx}(x, y) + O(h_x^2)$$

and

$$\delta_y^2 u(x, y) = \frac{u(x, y + h_y) - 2u(x, y) + u(x, y - h_y)}{h_y^2} = u_{yy}(x, y) + O(h_y^2).$$

Figure 3.1: Five-point finite difference stencil for the discrete Poisson equation (3.4).



Following these notations,

$$\delta_x^2 U_{i,j} = \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h_x^2} \quad \text{and} \quad \delta_y^2 U_{i,j} = \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{h_y^2}.$$

With this notation, our finite difference scheme can be written compactly as

$$\begin{aligned} -\left(\delta_x^2 U_{i,j} + \delta_y^2 U_{i,j}\right) &= f_{i,j} \quad \text{for } (x_i, y_j) \in \Omega, \\ U_{i,j} &= g_{i,j} \quad \text{for } (x_i, y_j) \in \Gamma, \end{aligned} \tag{3.4}$$

with the obvious abbreviations $f_{i,j} = f(x_i, y_j)$ and $g_{i,j} = g(x_i, y_j)$. Notice that $(x_i, y_j) \in \Omega$ for $1 \leqslant i \leqslant P-1$ and $1 \leqslant j \leqslant Q-1$, so there are

$$M = (P-1)(Q-1)$$

unknown values of $U_{i,j}$ at the interior grid points. The remaining $(P+1)(Q+1) - M = 2P + 2Q$ values are given directly by the Dirichlet boundary condition. The finite difference approximation provides one equation for each interior grid points, and hence one equation for each unknown, to yield an $M \times M$ linear system. Figure 3.1 shows the stencil for the scheme, which involves 5 grid points: (x_i, y_j) and its four nearest neighbours (x_{i-1}, y_j), (x_{i+1}, y_j), $(x_{i,j-1})$ and (x_i, y_{j+1}).

To describe the $M \times M$ system explicitly, we need to arrange the unknowns $U_{i,j}$ into a column vector of length M. This will be discussed in the next example.

Example 3.1. *Suppose* $P = 5$ *and* $Q = 4$*, with* $h_x = h = h_y$*. The finite difference method* (3.4) *amounts to*

$$-\frac{1}{h^2}\left(U_{i+1,j} - 2U_{i,j} + U_{i-1,j} + U_{i,j+1} - 2U_{i,j} + U_{i,j-1}\right) = f_{i,j},$$

or equivalently,

$$\frac{1}{h^2}\left(-U_{i,j-1} - U_{i-1,j} + 4U_{i,j} - U_{i+1,j} - U_{i,j+1}\right) = f_{i,j}, \quad 1 \leqslant i \leqslant 4, \quad 1 \leqslant j \leqslant 3,$$

and from the boundary conditions: $U_{\ell,j} = g_{\ell,j}$ *and* $U_{i,\ell} = g_{i,\ell}$, *for* $1 \leqslant i \leqslant 4$, $1 \leqslant j \leqslant 3$, *and* $\ell \in \{0, P\}$. *In a matrix form, it can be written as:*

$$\frac{1}{h^2}\mathbf{A}\mathbf{U} = \mathbf{f} + \frac{1}{h^2}\mathbf{g}, \tag{3.5}$$

where

$$\mathbf{A} = \left[\begin{array}{cccc|cccc|cccc} 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 \end{array}\right],$$

$$\mathbf{U} = \left[\begin{array}{c} U_{1,1} \\ U_{2,1} \\ U_{3,1} \\ U_{4,1} \\ \hline U_{1,2} \\ U_{2,2} \\ U_{3,2} \\ U_{4,2} \\ \hline U_{1,3} \\ U_{2,3} \\ U_{3,3} \\ U_{4,3} \end{array}\right], \quad \mathbf{f} = \left[\begin{array}{c} f_{1,1} \\ f_{2,1} \\ f_{3,1} \\ f_{4,1} \\ \hline f_{1,2} \\ f_{2,2} \\ f_{3,2} \\ f_{4,2} \\ \hline f_{1,3} \\ f_{2,3} \\ f_{3,3} \\ f_{4,3} \end{array}\right], \quad \mathbf{g} = \left[\begin{array}{c} g_{01} + g_{10} \\ g_{20} \\ g_{30} \\ g_{40} + g_{51} \\ \hline g_{02} \\ 0 \\ 0 \\ g_{52} \\ \hline g_{03} + g_{14} \\ g_{24} \\ g_{34} \\ g_{44} + g_{53} \end{array}\right].$$

To proceed, we introduce the following matrices: let $\mathbf{I}_x$ and $\mathbf{0}_x$ be the 4-by-4 (recall that, $4 = P - 1$) identity and zero matrices, respectively, and let

$$\mathbf{A}_x = \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 2 \end{bmatrix},$$

In a similar fashion, we define the 3-by-3 (recall that $3 = Q - 1$) matrices $\mathbf{I}_y$, $\mathbf{0}_y$ and $\mathbf{A}_y$. The next definition is also needed.

Definition 3.2. *Given matrices* $E \in \mathbb{R}^{M\times N}$ *and* $\mathbf{B} \in \mathbb{R}^{P\times Q}$, *the* Kronecker product $E \otimes \mathbf{B} \in \mathbb{R}^{(MP)\times(NQ)}$ *is the* $M \times N$ *block matrix whose* ij*-block equals* $a_{ji}\mathbf{B}$, *that is,*

$$E \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1N}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2N}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1}\mathbf{B} & a_{M2}\mathbf{B} & \cdots & a_{MN}\mathbf{B} \end{bmatrix}.$$

Using the above matrices and then the the *Kronecker product* product concept, $\mathbf{A}$ can be decomposed as: $\mathbf{A} = \mathbf{A_1} + \mathbf{A_2}$, with

$$\mathbf{A_1} = \left[\begin{array}{c|c|c} \mathbf{A_x} & \mathbf{0_x} & \mathbf{0_x} \\ \hline \mathbf{0_x} & \mathbf{A_x} & \mathbf{0_x} \\ \hline \mathbf{0_x} & \mathbf{0_x} & \mathbf{A_x} \end{array}\right] = \left[\begin{array}{c|c|c} 1 \times \mathbf{A_x} & 0 \times \mathbf{A_x} & 0 \times \mathbf{A_x} \\ \hline 0 \times \mathbf{A_x} & 1 \times \mathbf{A_x} & 0 \times \mathbf{A_x} \\ \hline 0 \times \mathbf{A_x} & 0 \times \mathbf{A_x} & 1 \times \mathbf{A_x} \end{array}\right] = \mathbf{I_y} \otimes \mathbf{A_x},$$

and

$$\mathbf{A_2} = \left[\begin{array}{c|c|c} 2\mathbf{I_x} & -\mathbf{I_x} & \mathbf{0_x} \\ \hline -\mathbf{I_x} & 2\mathbf{I_x} & -\mathbf{I_x} \\ \hline \mathbf{0_x} & -\mathbf{I_x} & 2\mathbf{I_x} \end{array}\right] = \left[\begin{array}{c|c|c} 2 \times \mathbf{I_x} & -1 \times \mathbf{I_x} & 0 \times \mathbf{I_x} \\ \hline -1 \times \mathbf{I_x} & 2 \times \mathbf{I_x} & -1 \times \mathbf{I_x} \\ \hline 0 \times \mathbf{I_x} & -1 \times \mathbf{I_x} & 2 \times \mathbf{I_x} \end{array}\right] = \mathbf{A_y} \otimes \mathbf{I_x}\,.$$

Therefore,

$$\mathbf{A} = \mathbf{I_y} \otimes \mathbf{A_x} + \mathbf{A_y} \otimes \mathbf{I_x}\,. \tag{3.6}$$

Now, for general P and Q, the column vectors U, f, and g will be appropriately adjusted, and the matrix structure $\mathbf{A}$ in (3.6) remains valid with

$$\mathbf{A_x} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{(P-1)\times(P-1)}$$

and

$$\mathbf{A_y} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{(Q-1)\times(Q-1)}.$$

The identity matrix $\mathbf{I_x}$ and the zero matrix $\mathbf{0_x}$ are in $\mathbb{R}^{(P-1)\times(P-1)}$, while the identity matrix $\mathbf{I_y}$ and the zero matrix $\mathbf{0_y}$ are in $\mathbb{R}^{(Q-1)\times(Q-1)}$.

3.2 Well-posedness of the finite difference solution

We rely on the matrix structure introduced in the preceding section to show the existence and uniqueness of the finite difference solution in (3.4). If the matrix $\mathbf{A}$ is non-singular, then the linear system $\frac{1}{h^2}\mathbf{A}U = \mathbf{f} + \frac{1}{h^2}g$ has a unique solution. So, it is sufficient to show that $\mathbf{A}$ is non-singular to achieve the goal. In fact, we claim that $\mathbf{A}$ is positive-definite, and thus, it is non-singular.

The next theorem gives a key relation between the Kronecker product and the ordinary matrix product that will be used later.

Theorem 3.3. *If* $\mathbf{E} \in \mathbb{R}^{M\times N}$, $\mathbf{B} \in \mathbb{R}^{P\times Q}$, $\mathbf{C} \in \mathbb{R}^{N\times R}$ *and* $\mathbf{D} \in \mathbb{R}^{Q\times S}$, *then*

$$(\mathbf{E} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{EC}) \otimes (\mathbf{BD}).$$

Proof. The ij-block of $(\mathbf{E} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D})$ equals

$$\sum_{k=1}^{N} (a_{ik}\mathbf{B})(c_{kj}\mathbf{D}) = \left(\sum_{k=1}^{N} a_{ik}c_{kj}\right)(\mathbf{BD}) = (\mathbf{EC})_{i,j}(\mathbf{BD}),$$

which is also the ij-block of $(\mathbf{EC}) \otimes (\mathbf{BD})$. □

Theorem 3.4. *The matrix* $\mathbf{A}$ *given in* (3.6) *is positive-definite, that is,* $V^T\mathbf{A}V > 0$ *for any non-zero column vector* $V \in \mathbb{R}^{(P-1)(Q-1)}$.

Proof. First, noting that any non-zero vector V in $\mathbb{R}^{(P-1)(Q-1)}$ can be expressed as: $V = V_y \otimes V_x$ for some non-zero vectors V_y and V_x in $\mathbb{R}^{Q-1}$ and $\mathbb{R}^{P-1}$, respectively. Then, by using Theorem thm: Kronecker A B C D, we get

$$V^T(\mathbf{I}_y \otimes \mathbf{A}_x)V = (V_y^T \otimes V_x^T)(\mathbf{I}_y V_y) \otimes (\mathbf{A}_x V_x) = (V_y^T \mathbf{I}_y V_y) \otimes (V_x^T \mathbf{A}_x V_x)\,.$$

However, $\mathbf{I}_y$ and $\mathbf{A}_x$ are positive definite, then, $V_y^T \mathbf{I}_y V_y > 0$ and $V_x^T \mathbf{A}_x V_x > 0$, and so is $V^T(\mathbf{I}_y \otimes \mathbf{A}_x)V$. Therefore, $\mathbf{I}_y \otimes \mathbf{A}_x$ is positive-definite. In a similar fashion, one can show that $\mathbf{A}_y \otimes \mathbf{I}_x$ is also positive-definite. Since $\mathbf{A} = \mathbf{I}_y \otimes \mathbf{A}_x + \mathbf{A}_y \otimes \mathbf{I}_x$, the proof of positive-definiteness of $\mathbf{A}$ is completed. □

3.3 Truncation error

We define the local truncation error in the usual way, as

$$\mathcal{T}_{i,j} \;=\; f_{i,j} - \Big(-\delta_x^2 u_{i,j} - \delta_y^2 u_{i,j}\Big) \;=\; -(u_{xx}(x_i, y_j) + u_{yy}(x_i, y_j)) + \Big(\delta_x^2 u_{i,j} + \delta_y^2 u_{i,j}\Big),$$

and note that

$$\begin{aligned}\mathcal{T}_{i,j} = &-\left(u_{xx}(x_i, y_j) - \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2}\right)\\ &-\left(u_{yy}(x_i, y_j) - \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2}\right),\end{aligned} \tag{3.7}$$

so, by Theorem 1.2,

$$|\mathcal{T}_{i,j}| \leqslant \left(\frac{\|u_{xxxx}\|_\infty}{12}\, h_x^2 + \frac{\|u_{yyyy}\|_\infty}{12}\, h_y^2\right) \tag{3.8}$$

With the help of the discrete maximum principle, the *solution error*

$$|U_{i,j} - u_{i,j}| \leqslant C\left(\frac{\|u_{xxxx}\|_\infty}{12}\, h_x^2 + \frac{\|u_{yyyy}\|_\infty}{12}\, h_y^2\right).$$

3.4 Exercises

3.1. Find $A \otimes B$ and $B \otimes A$ in each case.

(i)

$$A = \begin{bmatrix} 2 & 0 \\ 1 & -7 \\ 4 & 6 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & -1 \\ 2 & 0 \end{bmatrix}.$$

(ii)

$$A = \begin{bmatrix} 2 & 5 \\ 0 & -1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 8 \\ -9 \end{bmatrix}.$$

(iii)

$$A = \begin{bmatrix} 3 & -2 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & -4 \\ 3 & 2 \end{bmatrix}.$$

3.2. Prove the following properties of the Kronecker product.

(i) $A \otimes (B + C) = A \otimes B + A \otimes C$.

(ii) $(A + B) \otimes C = A \otimes C + B \otimes C$.

(iii) $A \otimes (B \otimes C) = (A \otimes B) \otimes C$.

(iv) $(A \otimes B)^T = A^T \otimes B^T$.

3.3. Choose $f(x, y) = 8 \sin(2x) \sin(2y)$, $L_x = L_y = \pi$. In this case, the exact solution of the 2D steady-state model (3.1) is $u(x, y) = 8 \sin(2x) \sin(2y)$.

(i) Use the finite difference scheme in (3.4) to surf the approximate solution U when $P = Q = 40$.

(ii) Justify numerically that the convergence rate of this scheme is 2, by choosing different iterations. For instance, choose $P = Q = 10, 20, 40, 80$.

3.4. Let (r, θ) denote the usual polar coordinates, and consider a domain Ω of the form

$$a < r < b \quad \text{and} \quad \alpha < \theta < \beta,$$

with $a > 0$ and $\beta - \alpha \leqslant 2\pi$. It is natural to define polar grid points

$$(r_i, \theta_j) = (a + i\, h_r, \alpha + j\, h_\theta), \quad h_r = \frac{b - a}{P}, \quad h_\theta = \frac{\beta - \alpha}{Q},$$

for $0 \leqslant i \leqslant P$ and $0 \leqslant j \leqslant Q$. Recall that

$$\nabla^2 u = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial u}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2}.$$

(i) Formulate a finite difference approximation to the Poisson problem

$$\begin{aligned} -\nabla^2 u &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

by setting up a $P \times Q$ rectangular grid in the (r, θ)-plane.

(ii) Suppose that Ω is a disk of radius b (so $a = 0$, $\alpha = -\pi$ and $\beta = \pi$). Formulate a finite difference method with $1 + (P - 1)Q$ unknowns. Hint: you will need an equation at the origin.

3.5. Consider the following steady-state reaction-diffusion equation

$$\begin{aligned} -u_{xx} - u_{yy} + u &= f \quad \text{on } \Omega, \\ \partial_n u &= 0 \quad \text{on } \Gamma, \end{aligned}$$

where $\Omega = (0, \pi) \times (0, \pi)$ and Γ is the boundary of Ω. Here, $\partial u / \partial n$ is the derivative of u in the direction of the *outward unit normal* $\mathbf{n}$ for Ω, that is,

$$\frac{\partial u}{\partial n}(x_1, x_2) = \mathbf{n}(x_1, x_2) \cdot \nabla u(x_1, x_2) \quad \text{for } (x_1, x_2) \in \Gamma.$$

(i) As in the lecture notes, develop a 5-point second order finite difference method for the given model using uniform meshes of size $h = \pi/P$ in x and y directions.

(ii) Following the calculations used to obtain (3.5), we should be able to write our finite difference scheme in a matrix form as:

$$\mathbf{A}\mathbf{U} = \mathbf{f},$$

where

$$\mathbf{A} = \frac{1}{h^2}[\mathbf{I}_x \otimes \mathbf{A}_x + \mathbf{A}_x \otimes \mathbf{I}_x] + \mathbf{I}_x \otimes \mathbf{I}_x,$$

with $\mathbf{I}_x$ being the $(P+1) \times (P+1)$ identity matrix,

$$\mathbf{A}_x = \begin{bmatrix} 2 & -2 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -2 & 2 \end{bmatrix} \in \mathbb{R}^{(P+1)\times(P+1)},$$

and $\mathbf{f}$ is the transpose of the row vector

$$[f_{0,0},\ f_{1,0},\ \ldots\ f_{P,0},\ f_{0,1},\ f_{1,1},\ \ldots\ f_{P,P}].$$

Use the matrix form to discuss the existence and uniqueness of the finite difference solution.

(iii) Let $P = 20$ and $f(x,y) = 9\cos(2x)\cos(2y)$, the exact solution in this case is $u(x,y) = \cos(2x)\cos(2y)$. Plot both the approximate and the exact solutions in separate figures.

(iv) Let $f(x,y) = 9\cos(2x)\cos(2y)$. Justify numerically that the developed scheme in (i) is second order accurate.

3.6. Consider the Poisson problem,

$$\begin{aligned} -(u_{xx} + u_{yy}) &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega, \end{aligned}$$

where Ω is the triangular region

$$0 \leqslant x \leqslant 1, \quad 0 \leqslant y \leqslant 1 - x.$$

We define grid points

$$(x_i, y_j) = (i\,h, j\,h) \quad \text{for } 0 \leqslant p \leqslant P,\ 0 \leqslant q \leqslant P - p,$$

where $h = 1/P$, as illustrated in triangular figure for the case $P = 5$, and seek $U_{ij} \approx u(x_i, y_j)$.

(i) Write down the usual 5-point finite difference approximation for the Poisson equation, based on second-order central difference approximations to u_{xx} and u_{yy}. Use the abbreviation $f_{ij} = f(x_i, y_j)$

(ii) Write down the equation at (x_1, y_1), after the known boundary values are moved to the right-hand side. You can use the abbreviation $g_{ij} = g(x_i, y_j)$.

(iii) Assume now that $P = 5$, and put

$$\begin{aligned} \mathbf{U} &= [U_{11} \quad U_{21} \quad U_{31} \quad U_{12} \quad U_{22} \quad U_{13}]^\top, \\ \mathbf{f} &= [f_{11} \quad f_{21} \quad f_{31} \quad f_{12} \quad f_{22} \quad f_{13}]^\top. \end{aligned}$$

Find the matrix $\mathbf{A}$ and vector $\mathbf{g}$ with

$$\mathbf{A}\mathbf{U} = h^2\mathbf{f} + \mathbf{g}.$$

Figure 3.2: Finite difference grid for a triangular domain.

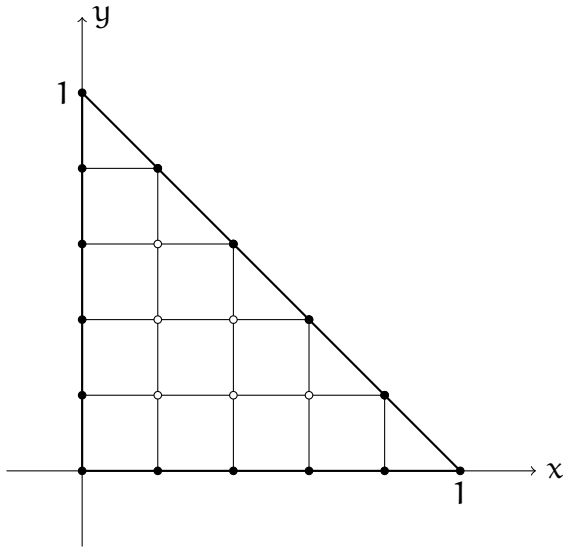
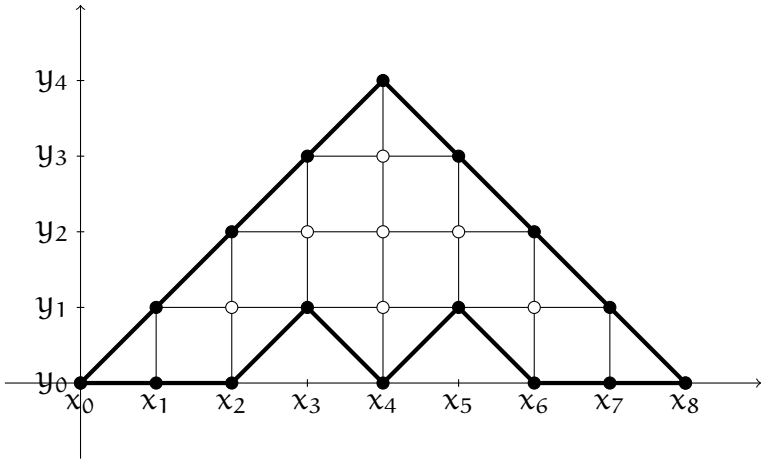


Figure 3.3: Finite difference grid for a triangular domain with two wedges.



3.7. Consider the following steady-state model,

$$\begin{cases} -\nabla^2 u(x,y) &= f(x,y) \quad \text{for } (x,y) \in \Omega, \\ u(x,y) &= g(x,y) \quad \text{for } (x,y) \in \partial\Omega, \end{cases} \tag{3.9}$$

where (open domain) Ω is the polygonal region shown in the above figure with a finite difference grid.

(i) Write out the usual finite difference scheme for the given model at the uniform grid point $(x_i, y_j) = (i\,h, j\,h)$. Use the standard abbreviations $U_{i,j} \approx u(x_i, y_j)$, $f_{i,j} = f(x_i, y_j)$, and $g_{i,j} = g(x_i, y_j)$.

(ii) Express $U_{2,1}$ and $U_{6,1}$ in terms of the relevant $f_{i,j}$ and $g_{i,j}$.

(iii) Assuming

$$\mathbf{U} = [U_{4,1} \quad U_{3,2} \quad U_{4,2} \quad U_{5,2} \quad U_{4,3}]^\top,$$
$$\mathbf{f} = [f_{4,1} \quad f_{3,2} \quad f_{4,2} \quad f_{5,2} \quad f_{4,3}]^\top,$$

find the matrix $\mathbf{A}$ and vector $\mathbf{G}$ such that

$$\mathbf{AU} = h^2\mathbf{f} + \mathbf{G}.$$

Chapter 4

Finite differences for time-dependent diffusion models

Consider the following *initial-boundary value problem* for $u = u(x,t)$,

$$\begin{aligned} u_t(x,t) - a u_{xx}(x,t) &= f(x,t) \quad \text{for } 0 < x < L \text{ and } 0 < t < T, \\ u(0,t) = \gamma_0(t), \quad u(L,t) &= \gamma_L(t) \quad \text{for } 0 < t < T, \\ u &= u_0(x) \quad \text{for } 0 < x < L \text{ when } t = 0, \end{aligned} \tag{4.1}$$

where $u_t = \partial u/\partial t$ and $u_{xx} = \partial^2 u/\partial x^2$. For simplicity, we assume that the coefficient a is a positive constant. The problem (4.1) provides a model of heat conduction in 1D, where $u(x,t)$ is the temperature at position x and time t. The coefficient $a > 0$ is the *thermal conductivity*: the value of a will be large for a material that conducts heat well, but small for a material that conducts heat poorly (that is, for a thermal insulator). The *source term* $f(x,t)$ gives the density of any heat sources in the material, the *boundary conditions* specify the temperatures γ_0 and γ_L at the two edges of the spatial domain $[0, L]$, and the *initial condition* gives the temperature field $u_0(x)$ when $t = 0$. In this 1D model, the temperature does not vary in the y and z directions.

4.1 Explicit Euler method

To obtain a fully-discrete scheme, we introduce uniform grids in both time and space:

$$t_n = n\tau \quad \text{for } 0 \leqslant n \leqslant N, \quad \text{where } \tau = \frac{T}{N}, \tag{4.2}$$

and

$$x_i = i h \quad \text{for } 0 \leqslant i \leqslant P, \quad \text{with } h = \frac{L}{P}.$$

For convenience, we introduce the following notations:

$$u_i^n = u(x_i, t_n), \quad f_i^n = f(x_i, t_n), \quad \gamma_0^n = \gamma_0(t_n), \quad \gamma_L^n = \gamma_L(t_n), \quad u_{0i} = u_0(x_i).$$

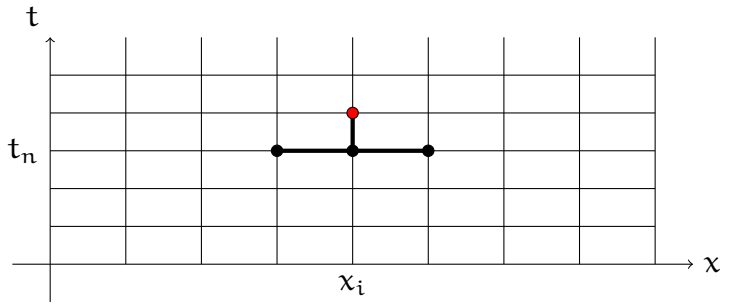
Our explicit scheme is based on the *forward difference* approximation in time and the second-order central difference approximation in space. To this end, from our model problem (4.1),

$$u_t(x_i, t_n) - a u_{xx}(x_i, t_n) = f_i^n .$$

The *forward difference* approximation in time,

$$u_t(x_i, t_n) \approx \frac{u_i^{n+1} - u_i^n}{\tau},$$

Figure 4.1: Computational stencil for the explicit Euler method (4.5).



and the second-order central difference approximation in space,

$$u_{xx}(x_i, t_n) \approx \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2},$$

leads to

$$\frac{u_i^{n+1} - u_i^n}{\tau} - a\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} \approx f_i^n .$$

The *explicit Euler method* finite difference solution for (4.1): $U_i^n \approx u_i^n$,

$$\begin{aligned} \frac{U_i^{n+1} - U_i^n}{\tau} - a\,\frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{h^2} = f_i^n \quad & \text{for } 1 \leqslant i \leqslant P-1 \text{ and } 0 \leqslant n \leqslant N-1, \\ U_0^n = \gamma_0^n, \quad U_P^n = \gamma_L^n \quad & \text{for } 0 \leqslant n \leqslant N, \\ U_i^0 = u_{0i} \quad & \text{for } 1 \leqslant i \leqslant P-1. \end{aligned} \tag{4.3}$$

We multiply both sides of the finite difference equation by τ and put

$$\rho = \frac{a\,\tau}{h^2} \tag{4.4}$$

to obtain

$$U_i^{n+1} = f_i^n\,\tau + \rho U_{i-1}^n + (1-2\rho)U_i^n + \rho U_{i+1}^n. \tag{4.5}$$

In this way, the scheme provides an *explicit* formula for the approximate solution at (x_i, t_{n+1}) given the values at the three nearest grid points (x_{i-1}, t_n), (x_i, t_n) and (x_{i+1}, t_n) from the previous time level. We say the these four points constitute the *stencil* for the method; see Fig. 4.1. Algorithm 6 shows how the formula (4.5) is used to compute the finite difference solution U_i^n by advancing from one time level to the next, starting at $t_0 = 0$.

We now show that the explicit Euler method is stable provided the time step τ is sufficiently small compared to the spatial grid size h, and so, the scheme is conditionally stable. For brevity, the notation

$$\|g_{i:j}\|_\infty = \max_{i \leqslant \ell \leqslant j} |g_\ell|$$

is used.

Theorem 4.1. *If* h *and* τ *satisfy* $\tau \leqslant \frac{h^2}{2a}$, *that is,* $\rho \leqslant 1/2$, *then the explicit Euler method* (4.3) *is stable:*

$$\|U_{0:P}^n\|_\infty \leqslant \max\{\|(u_0)_{1:P-1}\|_\infty, \|\gamma_0^{0:n}\|_\infty, \|\gamma_L^{0:n}\|_\infty\} + \sum_{j=0}^{n-1} \tau \|f_{1:P-1}^j\|_\infty \quad \textit{for } 0 \leqslant n \leqslant N.$$

Algorithm 6 Explicit Euler method.

Allocate storage for x_i, t_n and U_i^n, where $0 \leqslant i \leqslant P$ and $0 \leqslant n \leqslant N$.
$h = L/P$
$\tau = T/N$
for $i = 0 : P$ **do**
  $x_i = i\,h$
end for
for $n = 0 : N$ **do**
  $t_n = n\,\tau$
  $U_0^n = \gamma_0^n$
  $U_P^n = \gamma_L^n$
end for
$\rho = a\,\tau/h^2$
for $i = 1 : P - 1$ **do**
  $U_i^0 = u_0(x_i)$
end for
for $n = 0 : N - 1$ **do**
  for $i = 1 : P - 1$ **do**
    $U_i^{n+1} = f_i^n\,\tau + \rho U_{i-1}^n + (1 - 2\rho)U_i^n + \rho U_{i+1}^n$
  end for
end for

Proof. We use finite induction on n. When $n = 0$, we have $U_i^0 = u_{0i}$ for $1 \leqslant i \leqslant P - 1$, with $U_0^0 = \gamma_0^0$ and $U_P^0 = \gamma_L^0$, so

$$\|U_{0:P}^0\|_\infty = \max\big\{\|(u_0)_{1:P-1}\|_\infty, |\gamma_0^0|, |\gamma_L^0|\big\},$$

which agrees with the formula since the empty sum vanishes. Now, assume that the stability property holds true at level n, and let us prove it at the level $n + 1$. From (4.5) and the assumption $\rho \leqslant 1/2$ (that is, $|1 - 2\rho| = 1 - 2\rho$), we observe that

$$|U_i^{n+1}| \leqslant \tau\,|f_i^n| + \rho|U_{i-1}^n| + |1 - 2\rho|\,|U_i^n| + \rho|U_{i+1}^n| \quad \text{for } 1 \leqslant i \leqslant P - 1,$$

and thus,

$$\|U_{1:P-1}^{n+1}\|_\infty \leqslant \tau\|f_{1:P-1}^n\|_\infty + \|U_{0:P}^n\|_\infty\,.$$

Recall that $U_0^{n+1} = \gamma_0^{n+1}$ and $U_P^{n+1} = \gamma_L^{n+1}$, and so, $\|U_{0:P}^{n+1}\|_\infty \leqslant \max\{\|U_{1:P-1}^{n+1}\|_\infty, |\gamma_0^{n+1}|, |\gamma_L^{n+1}|\}$. If $\|U_{0:P}^{n+1}\|_\infty \leqslant \max\{|\gamma_0^{n+1}|, |\gamma_L^{n+1}|\}$, then we have nothing to show. However, if $\|U_{0:P}^{n+1}\|_\infty \leqslant \|U_{1:P-1}^{n+1}\|_\infty$, then using the above achieved bound, we get

$$\|U_{0:P}^{n+1}\|_\infty \leqslant \tau\|f_{1:P-1}^n\|_\infty + \|U_{0:P}^n\|_\infty,$$

and thus, using the stability bound at the level n (by the induction hypothesis), we complete the proof. □

The *local trunction error* for the explicit Euler method is defined by: for $1 \leqslant i \leqslant P - 1$ and for $0 \leqslant n \leqslant N - 1$,

$$\mathcal{T}_i^n = \frac{u_i^{n+1} - u_i^n}{\tau} - a\,\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} - f_i^n, \tag{4.6}$$

and measures the extent to which the solution of the continuous problem (4.1) fails to satisfy the finite difference equation. In the next lemma, we estimate $\mathcal{T}_i^n$ by Taylor expansion.

Lemma 4.2. *If* u_{tt}, $u_{xxxx} \in C([0,L]\times[0,T])$, *then*

$$|\mathcal{T}_i^n| \leqslant \frac{\tau}{2}\max_{[0,L]\times[0,T]}|u_{tt}| + a\,\frac{h^2}{12}\max_{[0,L]\times[0,T]}|u_{xxxx}|$$

for $1 \leqslant i \leqslant P-1$ *and* $0 \leqslant n \leqslant N-1$.

Proof. Since $u_t - au_{xx} = f$,

$$\begin{aligned}|\mathcal{T}_i^n| &= \left|u_t(x_i,t_n) - \frac{u_i^{n+1}-u_i^n}{\tau} - a\left(u_{xx}(x_i,t_n) - \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2}\right)\right| \\ &\leqslant \left|u_t(x_i,t_n) - \frac{u_i^{n+1}-u_i^n}{\tau}\right| + a\left|u_{xx}(x_i,t_n) - \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2}\right|,\end{aligned}$$

and, so by Exercise 1.3 and Theorem 1.2, implies the stated estimate for $\mathcal{T}_i^n$. □

The next theorem shows that, provided the restriction $\tau \leqslant \frac{h^2}{2a}$ on the step sizes is satisfied, the explicit Euler method is first-order accurate in time and second-order accurate in space.

Theorem 4.3. *Assume that* h *and* τ *satisfy* $\tau \leqslant \frac{h^2}{2a}$, *and that* u_{tt}, $u_{xxxx} \in C([0,L]\times[0,T])$. *Then the error for the explicit Euler method satisfies*

$$|U_i^n - u_i^n| \leqslant Ct_n(\tau + h^2) \quad \textit{for } 0 \leqslant i \leqslant P-1 \textit{ and } 0 \leqslant n \leqslant N,$$

where

$$C = \max\left(\frac{1}{2}\max_{[0,L]\times[0,T]}|u_{tt}|, \frac{1}{12}\max_{[0,L]\times[0,T]}|u_{xxxx}|\right). \tag{4.7}$$

Proof. By the definition (4.6) of the local truncation error,

$$\frac{u_i^{n+1}-u_i^n}{\tau} - a\,\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} = f_i^n + \mathcal{T}_i^n.$$

Subtracting this equation from the finite difference equation in (4.3), we obtain a finite difference equation for the error $E_i^n = U_i^n - u_i^n$ in the finite difference solution, namely

$$\frac{E_i^{n+1}-E_i^n}{\tau} - a\,\frac{E_{i+1}^n - 2E_i^n + E_{i-1}^n}{h^2} = -\mathcal{T}_i^n \quad \text{for } 1 \leqslant i \leqslant P-1 \text{ and } 0 \leqslant n \leqslant N-1.$$

In addition, the boundary and initial conditions satisfied by U_i^n and u imply that

$$\begin{aligned} E_0^n &= U_0^n - u_0^n = \gamma_0^n - \gamma_0^n = 0 && \text{for } 0 \leqslant n \leqslant N, \\ E_P^n &= U_P^n - u_P^n = \gamma_L^n - \gamma_L^n = 0 && \text{for } 0 \leqslant n \leqslant N, \\ E_i^0 &= U_i^0 - u(x_i,0) = u_0(x_i) - u_0(x_i) = 0 && \text{for } 1 \leqslant i \leqslant P-1. \end{aligned}$$

Thus, E_i^n is the explicit Euler solution obtained when f_i^n, γ_0^n, γ_L^n and u_{0i} are replaced by $-\mathcal{T}_i^n$, 0, 0 and 0, respectively. The restriction on the step sizes allows us to apply the stability estimate of Theorem 4.1 to deduce that

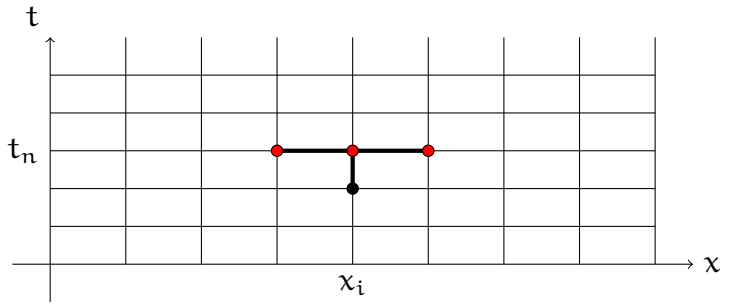
$$\|E_{0:P}^n\|_\infty \leqslant \sum_{j=0}^{n-1}\tau\,\|\mathcal{T}_{1:P-1}^n\|_\infty \quad \text{for } 0 \leqslant n \leqslant N.$$

By Lemma 4.2, $\|\mathcal{T}_{1:P-1}^n\|_\infty \leqslant C(\tau + h^2)$, so

$$\|E_{0:P}^n\|_\infty \leqslant C\,n\,\tau(\tau+h^2) = Ct_n\,(\tau+h^2),$$

as claimed. □

Figure 4.2: Computational stencil for the implicit Euler method (4.9).



4.2 Implicit Euler method

Instead of using a forward difference, we can approximate u_t by a *backward difference*,

$$u_t(x_i, t_n) \approx \frac{u_i^n - u_i^{n-1}}{\tau},$$

which leads to the *im*plicit Euler method,

$$\begin{aligned} \frac{U_i^n - U_i^{n-1}}{\tau} - a\,\frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{h^2} = f_i^n \quad &\text{for } 1 \leqslant i \leqslant P-1 \text{ and } 1 \leqslant n \leqslant N, \\ U_0^n = \gamma_0^n, \quad U_P^n = \gamma_L^n \quad &\text{for } 0 \leqslant n \leqslant N, \\ U_i^0 = u_{0i} \quad &\text{for } 1 \leqslant i \leqslant P-1. \end{aligned} \tag{4.8}$$

Multiplying both sides of the finite difference equation by τ, and defining ρ as before in (4.4), we obtain

$$-\rho U_{i-1}^n + (1+2\rho)U_i^n - \rho U_{i+1}^n = U_i^{n-1} + \tau f_i^n \tag{4.9}$$

for $1 \leqslant i \leqslant P-1$ and $1 \leqslant n \leqslant N$. Thus, at the nth time level we have a system of linear equations for the unknown $U_1^n, U_2^n, \ldots, U_{P-1}^n$, with the right-hand sides involving the solution at the previous time level. The stencil for the scheme is shown in Section 4.2.

In a matrix form, the above implicit finite difference method can be written in a matrix form as:

$$\mathbf{U}^n - \mathbf{U}^{n-1} + \rho \mathbf{A}\mathbf{U}^n = \tau \mathbf{f}^n + \rho \mathbf{g}^n \quad \text{for } 1 \leqslant n \leqslant N, \tag{4.10}$$

where

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix},$$

and

$$\mathbf{U}^n = \begin{bmatrix} U_1^n \\ U_2^n \\ \vdots \\ U_{P-2}^n \\ U_{P-1}^n \end{bmatrix}, \qquad \mathbf{f}^n = \begin{bmatrix} f_1^n \\ f_2^n \\ \vdots \\ f_{P-2}^n \\ f_{P-1}^n \end{bmatrix}, \qquad \mathbf{g}^n = \begin{bmatrix} \gamma_0^n \\ 0 \\ \vdots \\ 0 \\ \gamma_L^n \end{bmatrix}.$$

Rearranging (4.10) we find that

$$(\mathbf{I} + \rho\,\mathbf{A})\mathbf{U}^n = \mathbf{U}^{n-1} + \tau\,\mathbf{f}^n + \rho \mathbf{g}^n \quad \text{for } 1 \leqslant n \leqslant N,$$

where $\mathbf{I}$ is a $(P-1)$-by-$(P-1)$ identity matrix. Here, the coefficient matrix $\mathbf{I}+\rho\mathbf{A}$ is symmetric, tridiagonal and positive-definite, so it can be factorized as $\mathbf{L}^T\mathbf{D}\mathbf{L}$, and then, we can compute $\mathbf{U}^n$ as shown in Algorithm 7. Furthermore, since $\mathbf{I}+\rho\mathbf{A}$ is positive-definite, then the above algebraic linear system of equations is uniquely solvable and so, the proposed implicit Euler finite difference scheme in (4.8) possess a unique solution.

Algorithm 7 Implicit Euler method.

Allocate storage for x_i, t_n and U_i^n, where $0 \leqslant i \leqslant P$ and $0 \leqslant n \leqslant N$.
Allocate storage for vectors $[d_1, d_2, \ldots, d_{P-1}]^T$ and $[\ell_1, \ell_2, \ldots, \ell_{P-2}]^T$.
$h = L/P$
$\tau = T/N$
for $i = 0 : P$ **do**
  $x_i = i\,h$
end for
for $n = 0 : N$ **do**
  $t_n = n\,\tau$
  $U_0^n = \gamma_0^n$
  $U_P^n = \gamma_L^n$
end for
$\rho = a\,\tau/h^2$
for $i = 1 : P-1$ **do**
  $d_i = 1 + 2\rho$
end for
for $i = 1 : P-2$ **do**
  $\ell_i = -\rho$
end for
FACTORIZE!($\mathbf{d}$, $\boldsymbol{\ell}$) $\mathbf{I} + \tau\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^T$ in place.
for $i = 1 : P-1$ **do**
  $U_i^0 = u_0(x_i)$
end for
for $n = 1 : N$ **do**
  for $i = 1 : P-1$ **do**
    $U_i^n \leftarrow U_i^{n-1} + \tau\,f(x_i, t_n)$
  end for
  $U_1^n \leftarrow U_1^n + a\gamma_0/h^2$
  $U_{P-1}^n \leftarrow U_{P-1}^n + a\gamma_L/h^2$
  SOLVE!($\mathbf{U}_{1:P-1}^n$, $\mathbf{d}$, $\boldsymbol{\ell}$) for $\mathbf{U}_{1:P-1}^n$ in place.
end for

Theorem 4.4. *The implicit Euler method is unconditionally stable: for any choice of* h *and* τ,

$$\|U_{0:P}^n\|_\infty \leqslant \max\{\|(u_0)_{1:P-1}\|_\infty, \|\gamma_0^{0:n}\|_\infty, \|\gamma_L^{0:n}\|_\infty\} + \sum_{j=1}^{n} \tau\,\|f_{1:P-1}^j\|_\infty \quad \textit{for } 0 \leqslant n \leqslant N. \tag{4.11}$$

Proof. We use finite induction on n. When $n = 0$, we have $U_i^0 = u_{0i}$ for $1 \leqslant i \leqslant P-1$, with $U_0^0 = \gamma_0^0$ and $U_P^0 = \gamma_L^0$, so

$$\|U_{0:P}^0\|_\infty = \max\{\|(u_0)_{1:P-1}\|_\infty, |\gamma_0^0|, |\gamma_L^0|\},$$

which agrees with the formula since the empty sum vanishes. Now, assume that

$$\|U_{0:P}^{n-1}\|_\infty \leqslant \max\{\|(u_0)_{1:P-1}\|_\infty, \|\gamma_0^{0:n-1}\|_\infty, \|\gamma_L^{0:n-1}\|_\infty\} + \sum_{j=1}^{n-1} \tau\,\|f_{1:P-1}^j\|_\infty, \tag{4.12}$$

and the task is to show (4.11) at the level n. We see from (4.9) that, for $1 \leqslant i \leqslant P-1$,

$$(1+2\rho)U_i^n = \rho U_{i-1}^n + \rho U_{i+1}^n + U_i^{n-1} + \tau f_i^n$$

and so

$$(1+2\rho)\|U_{1:P-1}^n\|_\infty \leqslant 2\rho\|U_{0:P}^n\|_\infty + \|U_{1:P-1}^{n-1}\|_\infty + \tau\|f_{1:P-1}^n\|_\infty .$$

Recall that, $U_0^n = \gamma_0^n$ and $U_P^n = \gamma_L^n$, and so, $\|U_{0:P}^n\|_\infty \leqslant \max\{\|U_{1:P-1}^n\|_\infty, |\gamma_0^n|, |\gamma_L^n|\}$. If

$$\|U_{0:P}^n\|_\infty \leqslant \max\{|\gamma_0^n|, |\gamma_L^n|\}, \tag{4.13}$$

then we have nothing to show. However, if $\|U_{0:P}^n\|_\infty \leqslant \|U_{1:P-1}^n\|_\infty$, then by the above achieved bound,

$$(1+2\rho)\|U_{0:P}^n\|_\infty \leqslant 2\rho\|U_{0:P}^n\|_\infty + \|U_{1:P-1}^{n-1}\|_\infty + \tau\|f_{1:P-1}^n\|_\infty .$$

After canceling $2\rho\|U_{0:P}^n\|_\infty$ and using (4.12), it follows that

$$\begin{aligned}
|U_{0:P}^n\|_\infty &\leqslant \|U_{1:P-1}^{n-1}\|_\infty + \tau\|f_{1:P-1}^n\|_\infty \\
&\leqslant \max\big\{\|(u_0)_{1:P-1}\|_\infty, \|\gamma_0^{0:n-1}\|_\infty, \|\gamma_L^{0:n-1}\|_\infty\big\} + \sum_{j=1}^{n-1}\tau\|f_{1:P-1}^j\|_\infty + \tau\|f_{1:P-1}^n\|_\infty \\
&= \max\big\{\|(u_0)_{1:P-1}\|_\infty, \|\gamma_0^{0:n-1}\|_\infty, \|\gamma_L^{0:n-1}\|_\infty\big\} + \sum_{j=1}^{n}\tau\|f_{1:P-1}^j\|_\infty .
\end{aligned}$$

From this and the bound in (4.13), the required stability estimate is achieved. □

The local truncation error for the *im*plicit Euler method is defined by: for $1 \leqslant i \leqslant P-1$, and for $1 \leqslant n \leqslant N$,

$$\mathcal{T}_i^n = \frac{u_i^n - u_i^{n-1}}{\tau} - a\,\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} - f_i^n,$$

and we can again estimate $\mathcal{T}_i^n$ by Taylor expansion (Exercise 1.3 and Theorem 1.2):

$$|\mathcal{T}_i^n| \leqslant \frac{\tau}{2}\max_{[0,L]\times[0,T]}|u_{tt}| + a\,\frac{h^2}{12}\max_{[0,L]\times[0,T]}|u_{xxxx}|$$

for $1 \leqslant i \leqslant P-1$ and $1 \leqslant n \leqslant N$, which leads to the following global error bound where the proof is similar to the one constructed in the preceding section (Theorem 4.3) for showing the global error of the implicit Euler finite difference scheme.

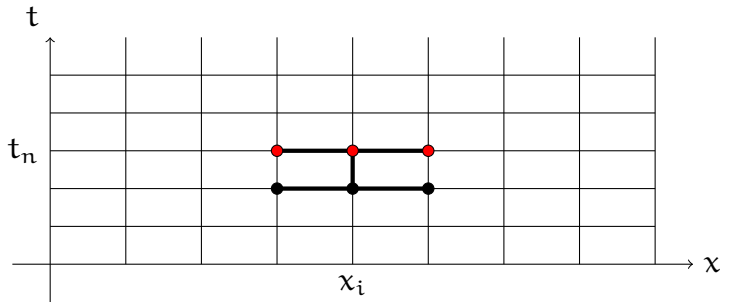
Theorem 4.5. *Assume that* $u_{tt}, u_{xxxx} \in C([0,L]\times[0,T])$. *Let* U *be the implicit numerical solution defined by* (4.8). *Then the error satisfies*

$$|U_i^n - u_i^n| \leqslant Ct_n(\tau + h^2) \quad \textit{for } 1 \leqslant i \leqslant P-1 \textit{ and } 0 \leqslant n \leqslant N .$$

4.3 Crank–Nicolson method

The developed scheme in this section is again implicit and also unconditionally stable as in the case of implicit Euler scheme discussed in the previous section. However, the current scheme is second-order accurate in both time and space $O(\tau^2 + h^2)$, and not only in space, and thus, it is more accurate.

Figure 4.3: Computational stencil for the Crank–Nicolson method (4.14).



Recall that the implicit Euler method is only first-order accurate in time because the first-order backward difference approximates $u_t(x, t_n)$ only to first order in τ, that is,

$$\frac{u(x,t_n) - u(x,t_{n-1})}{\tau} = u_t(x,t_n) + O(\tau).$$

However, viewed as an approximation to $u_t(x, t_{n-1/2})$, with $t_{n-1/2} = \frac{t_n + t_{n-1}}{2} = t_n - \frac{1}{2}\tau$, the backward difference becomes a central difference (with step-size $\frac{1}{2}\tau$), and is therefore second-order accurate by Theorem 1.3:

$$\frac{u(x,t_n) - u(x,t_{n-1})}{\tau} = u_t(x,t_{n-1/2}) + O(\tau^2).$$

Furthermore, an application of Theorem 1.4 shows that

$$\frac{u(x,t_n) + u(x,t_{n-1})}{2} = u(x_i,t_{n-1/2}) + O(\tau^2).$$

This observation leads us to consider the *Crank–Nicolson method* for (4.1):

$$\frac{U_i^n - U_i^{n-1}}{\tau} - a\,\frac{U_{i+1}^{n-1/2} - 2U_i^{n-1/2} + U_{i-1}^{n-1/2}}{h^2} = f_i^{n-1/2} \tag{4.14}$$

for $1 \leqslant i \leqslant P-1$ and $1 \leqslant n \leqslant N$, where we have used the notation

$$U_i^{n-1/2} = \frac{U_i^n + U_i^{n-1}}{2},$$

and, on the right-hand side,

$$f_i^{n-1/2} = \frac{f_i^n + f_i^{n-1}}{2} \approx f(x_i, t_{n-1/2}) \quad \text{or} \quad f_i^{n-1/2} = f(x_i, t_{n-1/2}).$$

The usual discrete boundary conditions are imposed,

$$U_0^n = \gamma_0^n \quad \text{and} \quad U_P^n = \gamma_L^n \quad \text{for } 0 \leqslant n \leqslant N,$$

along with the usual discrete initial condition

$$U_i^0 = u_{0i} \quad \text{for } 1 \leqslant i \leqslant P-1.$$

Fig. 4.3 shows the stencil for the scheme. In matrix notation, the Crank–Nicolson method takes the form

$$\mathbf{U}^n - \mathbf{U}^{n-1} + \rho\mathbf{A}\mathbf{U}^{n-1/2} = \tau\mathbf{f}^{n-1/2} + \rho\mathbf{g}^{n-1/2}, \tag{4.15}$$

where the $(P-1)\times(P-1)$ matrix $\mathbf{A}$ and is same as in (4.10), with

$$g^{n-1/2} = \frac{g^n + g^{n-1}}{2} \quad \text{or} \quad g^{n-1/2} = g(t_{n-1/2})\,.$$

Rearranging the terms in (4.15) yields a linear system that must be solved at the nth time step,

$$\big(\mathbf{I} + \tfrac{\rho}{2}\mathbf{A}\big)\mathbf{U}^n = \big(\mathbf{I} - \tfrac{\rho}{2}\mathbf{A}\big)\mathbf{U}^{n-1} + \tau\, f^{n-1/2} + \rho\, g^{n-1/2}.$$

The proposed implicit Crank–Nicolson method is unconditionally stable. More precisely, by adapting the proof of Theorem 4.4, one can show that

Theorem 4.6. *The implicit Euler method is unconditionally stable: for any choice of* h *and* τ*,*

$$\|U^n_{0:P}\|_\infty \leqslant \max\big\{\|(u_0)_{1:P-1}\|_\infty, \|\gamma_0^{0:n}\|_\infty, \|\gamma_L^{0:n}\|_\infty\big\} + \sum_{j=1}^{n} \tau\, \|f^{n-\frac{1}{2}}_{1:P-1}\|_\infty \quad \textit{for } 0 \leqslant n \leqslant N.$$

In the next theorem we show that the local truncation error the Crank–Nicolson method,

$$\mathcal{T}_i^n = f_i^{n-1/2} - \frac{u_i^n - u_i^{n-1}}{\tau} + \frac{a}{h^2}\left(\frac{u_{i+1}^n + u_{i+1}^{n-1}}{2} - 2\,\frac{u_i^n + u_i^{n-1}}{2} + \frac{u_{i-1}^n + u_{i-1}^{n-1}}{2}\right),$$

is $O(\tau^2 + h^2)$. For simplicity, we choose $f_i^{n-1/2} = f(x_i, t_{n-1/2})\,.$

Theorem 4.7. *Assume that* $u_{tt}, u_{xxxx} \in C([0,L]\times[0,T])$*. Then the truncation error for the Crank-Nicolson method satisfies*

$$|\mathcal{T}_i^n| \leqslant C(\tau^2 + h^2) \quad \textit{for } 0 \leqslant i \leqslant P-1 \textit{ and } 0 \leqslant n \leqslant N,$$

Proof. Since $f_i^{n-1/2} = u_t(x_i, t_{n-1/2}) - a u_{xx}(x_i, t_{n-1/2})$,

$$\begin{aligned}
\mathcal{T}_i^n = &\left(u_t(x_i, t_{n-1/2}) - \frac{u_i^n - u_i^{n-1}}{\tau}\right) \\
&+ a\left(\frac{u_{xx}(x_i, t_n) + u_{xx}(x_i, t_{n-1})}{2} - u_{xx}(x_i, t_{n-1/2})\right) \\
&+ \frac{a}{2}\left(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} - u_{xx}(x_i, t_n)\right) \\
&+ \frac{a}{2}\left(\frac{u_{i+1}^{n-1} - 2u_i^{n-1} + u_{i-1}^{n-1}}{h^2} - u_{xx}(x_i, t_{n-1})\right).
\end{aligned}$$

Applying Theorems 1.3 and 1.4 with $\delta = \frac{1}{2}\tau$, and Theorem 1.2 with $\delta = h$, we conclude that

$$|\mathcal{T}_i^n| \leqslant \frac{\tau^2}{24} \max_{[0,L]\times[0,T]} |u_{ttt}| + a\,\frac{\tau^2}{8} \max_{[0,L]\times[0,T]} |u_{xxtt}| + a\,\frac{h^2}{12} \max_{[0,L]\times[0,T]} |u_{xxxx}|$$

and hence $\mathcal{T}_i^n = O(\tau^2 + h^2)$. □

By using the stability property of our scheme and the truncation error results from the above theorems, we can that the global error $|U_i^n - u_i^n| \leqslant C(\tau^2 + h^2)$.

4.4 Diffusion model in 2D

Consider the following 2D initial-boundary value problem,

$$\begin{aligned} u_t(x,y,t) - a\nabla^2 u(x,y,t) &= f(x,y,t) \quad \text{for } (x,y)\in\Omega \text{ and } 0<t<T,\\ u(x,y,t) &= g(x,y,t) \quad \text{for } (x,y)\in\partial\Omega \text{ and } 0<t<T,\\ u(x,y,t) &= u_0(x,y) \quad \text{for } (x,y)\in\Omega \text{ when } t=0, \end{aligned} \tag{4.16}$$

where, for simplicity, we assume that the coefficient a is a positive constant.

Here $\Omega = (0,L_x)\times(0,L_y)$ (as in (3.2)). We consider the time levels (4.2) and the spatial grid (3.3). By adapting the notations of the previous chapter, the finite difference solution $U_{i,j}^n \approx u_{i,j}^n = u(x_i,y_j,t_n)$ for the 2D problem (4.16) can be defined as: for $\theta\in\{0,1/2,1\}$,

$$\frac{U_{i,j}^n - U_{i,j}^{n-1}}{\tau} - a\Big(\delta_x^2 U_{i,j}^{n-\theta} + \delta_x^2 U_{i,j}^{n-\theta}\Big) = f_{i,j}^{n-\theta},$$

for $1\leqslant n\leqslant N$, $1\leqslant i\leqslant P-1$ and $1\leqslant j\leqslant Q-1$, with

$$\delta_x^2 U_{i,j}^{n-\theta} = \frac{U_{i+1,j}^{n-\theta} - 2U_{i,j}^{n-\theta} + U_{i-1,j}^{n-\theta}}{h_x^2}$$

and

$$\delta_y^2 U_{i,j}^{n-\theta} = \frac{U_{i,j+1}^{n-\theta} - 2U_{i,j}^{n-\theta} + U_{i,j-1}^{n-\theta}}{h_y^2}.$$

The boundary and initial conditions are

$$\begin{aligned} U_{i,j}^n &= g_{i,j}^n \qquad \text{for } 1\leqslant n\leqslant N \text{ and } (x_i,y_j)\in\partial\Omega,\\ U_{i,j}^0 &= u_0(x_i,y_j) \quad \text{for } (x_i,y_j)\in\Omega. \end{aligned} \tag{4.17}$$

It is clear that for $\theta = 1$, the above scheme is the explicit Euler finite difference method. In this case, the terms can be rearranged as:

$$U_{i,j}^n - U_{i,j}^{n-1} - \rho_x\big(U_{i+1,j}^{n-1} - 2U_{i,j}^{n-1} + U_{i-1,j}^{n-1}\big) - \rho_y\big(U_{i,j+1}^{n-1} - 2U_{i,j}^{n-1} + U_{i,j-1}^{n-1}\big) = f_{i,j}^{n-1}$$

with

$$\rho_x = \frac{a\,\tau}{h_x^2} \quad \text{and} \quad \rho_y = \frac{a\,\tau}{h_y^2}\,.$$

Hence

$$U_{i,j}^n = f_{i,j}^{n-1} + \rho_x U_{i-1,j}^{n-1} + \rho_y U_{i,j-1}^{n-1} + (1-2\rho_x-2\rho_y)U_{i,j}^{n-1} + \rho_y U_{i,j+1}^{n-1} + \rho_x U_{i+1,j}^{n-1}.$$

The proof of Theorem 4.1 generalizes to show that the scheme is stable if $1-2\rho_x-2\rho_y\geqslant 0$, or equivalently if

$$\tau \leqslant \frac{h_x^2\,h_y^2}{2a(h_x^2+h_y^2)}\,.$$

For $\theta = 0$ and $\theta = 1/2$, we have implicit Euler and Crank-Nicolson finite difference methods, respectively. Both of them are unconditionally stable. For the matrix form, we will discuss the case $\theta = 0$, (implicit Euler). The backward Euler method for the 2D problem (4.16) is

$$\frac{U_{i,j}^n - U_{i,j}^{n-1}}{\tau} - a\left(\frac{U_{i+1,j}^n - 2U_{i,j}^n + U_{i-1,j}^n}{h_x^2} + \frac{U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n}{h_y^2}\right) = f_{i,j}^n,$$

for $1 \leqslant n \leqslant N$, $1 \leqslant i \leqslant P-1$ and $1 \leqslant j \leqslant P-1$, with the boundary and initial conditions again given by (4.17). Rearranging the finite difference equation gives

$$-\rho_y U^n_{i,j-1} - \rho_x U^n_{i-1,j} + (1 + 2\rho_x + 2\rho_y) U^n_{i,j} - \rho_x U^n_{i+1,j} - \rho_y U^n_{i,j+1} = U^{n-1}_{i,j} + \tau f^n_{i,j}.$$

In matrix notation, the backward Euler method looks the same as in 1D,

$$\mathbf{U}^n - \mathbf{U}^{n-1} + \mathbf{A}\mathbf{U}^n = \tau \mathbf{f}^n + \tau \mathbf{g}^n, \quad \text{for } 1 \leqslant n \leqslant N,$$

where the column vector

$$\mathbf{U}^0 = [(u_0)_{1,1}, (u_0)_{2,1}, \cdots, (u_0)_{P-1,1}, (u_0)_{1,2}, \cdots, (u_0)_{P-1,Q-1}]^T.$$

Here, $\mathbf{A} = \rho_x \mathbf{I}_y \otimes \mathbf{A}_x + \rho_y \mathbf{A}_y \otimes \mathbf{I}_x$, where $\mathbf{I}_x, \mathbf{I}_y, \mathbf{A}_x$ are $\mathbf{A}_y$ are now given by (3.6). Of course, the meanings of the vectors $\mathbf{U}^n$, $\mathbf{f}^n = \mathbf{f}(t_n)$ and $\mathbf{g}^n = \mathbf{g}(t_n)$ are different from those in (4.10). Thus, at the nth time step we must solve the linear system

$$(\mathbf{I} + \mathbf{A})\mathbf{U}^n = \mathbf{U}^{n-1} + \tau \mathbf{f}^n + \tau \mathbf{g}^n.$$

The coefficient matrix is again symmetric and positive-definite.

4.5 Exercises

4.1. The temperature u of a metal bar, two meters long, satisfies the heat equation $u_t - u_{xx} = 0$, knowing that the temperature does not vary in the y and z directions.

Case 1. Initially, the temperature was varying quadratically and measured to be 10 at the center of the bar, while it is always measured to be zero at the ends of the bar.

(i) Write the mathematical model over the time duration $(0, 1)$.

(ii) With $x_i = ih$ and $t_n = n\tau$ where $h = 2/P$ and $\tau = 1/N$, define the usual explicit (first order) Euler finite difference solution $U^n_i \approx u^n_i = u(x_i, t_n)$.

(iii) Is your scheme stable when $P = 10$ and $N = 50$?

(iv) Show that the numerical approximation of the temperature of the bar does not exceed 10 when $2\tau \leqslant h^2$. Is this compatible with the theoretical results?

(v) Simulate graphically the finite difference solution.

(vi) Graph the approximate total temperature at each time level t_n and then comment on it.

Case 2. With $\Omega = (-1, 1)$, the initial temperature is measured to be $10(1-x)(1+x)$, and the temperature flux at the ends of the bar is always measured to be zero.

(vii) Write the mathematical model over the time duration $(0, 1)$ and show that the total temperature is conservative.

(viii) With $x_i = ih$ and $t_n = n\tau$ where $h = 2/P$ and $\tau = 1/N$, define the explicit (first order) Euler finite difference solution $U^n_i \approx u^n_i = u(x_i, t_n)$.

(ix) Plot the finite difference solution for some appropriate choices of P and N.

(x) Is your numerical solution in part (ix) accurate? (Justify your answer.)

4.2. Consider the following diffusion model:

$$u_t(x,t) - \kappa u_{xx}(x,t) = 0, \quad \text{for } 0 < x < L \text{ and } 0 < t < T,$$

(κ is a positive constant) subject to homogeneous Dirichlet boundary conditions, and with initial data $v(x)$. The BDF2 (backward differentiation formula, second-order) scheme for this model is defined as:

$$\frac{3U_i^n - 4U_i^{n-1} + U_i^{n-2}}{2\tau} - \kappa\,\frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{h^2} = 0,$$

where $1 \leqslant i \leqslant P-1$ and $2 \leqslant n \leqslant N$, with

$$U_0^n = 0 = U_P^n \quad \text{for } 0 \leqslant n \leqslant N,$$

and initial conditions $U_i^0 = v(x_i)$ for $0 \leqslant i \leqslant P$.

(i) Draw the stencil for the BDF2 scheme.

(ii) Define the local truncation error T_i^n for this scheme and show that $T_i^n = O(\tau^2 + h^2)$.

(iii) Write the BDF2 scheme in matrix–vector form, and hence describe the linear system that must be solved at the nth time step for $2 \leqslant n \leqslant N$.

(iv) To compute U_i^n when $n = 1$, we use the implicit Euler scheme,

$$\frac{U_i^1 - U_i^0}{\tau} - \kappa\,\frac{U_{i+1}^1 - 2U_i^1 + U_{i-1}^1}{h^2} = 0.$$

Why does the use of this method ensure that

$$U_i^1 = u(x_i, t_1) + O(\tau^2 + h^2),$$

when we saw in lectures that the implicit Euler method is only first-order accurate in time?

(v) Let $v(x) = (1 - x/5)(1 + x/5)e^{-x^2}$, $L = 5$, $T = 1$. Choose $P = 100$ and $N = 50$. Simulate the numerical solution U.

4.3. Let $u = u(x,t)$ be the solution of the heat equation,

$$\begin{aligned} u_t(x,t) - u_{xx}(x,t) &= 0 && \text{for } 0 < x < L \text{ and } 0 < t < T, \\ u(0,t) = u(L,t) &= 0 && \text{for } 0 \leqslant t \leqslant T, \\ u(x,0) &= u_0(x) && \text{for } 0 < x < L\,. \end{aligned} \tag{4.18}$$

With $h = L/P$ and $\tau = T/N$, define the grid points

$$(x_i, t_n) = (i\,h, n\,\tau) \quad \text{for } 0 \leqslant i \leqslant P \text{ and } 0 \leqslant n \leqslant N\,.$$

(i) Write down the equations involving $U_i^n \approx u(x_i, t_n)$ that define the **implicit Euler finite difference method** for (4.18).

(ii) Let $\|U^n\|_\infty := \max_{1 \leqslant i \leqslant P-1} |U_i^n|$ for $1 \leqslant n \leqslant N$, and $\|U^0\|_\infty = \|u_0\|_\infty : \max_{1 \leqslant i \leqslant P-1} |u_0(x_i)|$. Show by induction on n the following stability property:

$$\|U^n\|_\infty \leqslant \|u_0\|_\infty, \quad \text{for } 0 \leqslant n \leqslant N.$$

(iii) State the definition of the local truncation error T_i^n, and then show that $T_i^n = O(\tau + h^2)$.

(iv) Let the error $E_i^n = U_i^n - u_i^n$, with $u_i^n = u(x_i, t_n)$. Prove that

$$\|E^n\|_\infty \leqslant \|E^{n-1}\|_\infty + C\tau\,(\tau + h^2), \quad \text{for } 1 \leqslant n \leqslant N\,.$$

(v) Use the above part to claim that $\|E^n\|_\infty \leqslant C(\tau + h^2)$ for $1 \leqslant n \leqslant N$.

Chapter 5

Finite elements for 1D stationary models

Finite element methods (FEMs) represent a powerful and general class of techniques for the approximate solution of PDEs. They were proposed in the work of Courant in 1943; unfortunately, the relevance of this article was not recognized at the time and the idea was forgotten. In the early 1950's the method was rediscovered by engineers, but the mathematical analysis of finite element approximations began much later, in the 1960's. The term finite element was first used by Clough in 1960 and has become one of the most important numerical method for the numerical solutions of PDEs, particularly for equations of elliptic and parabolic types. In 1960s, engineers used the FEM for approximate solutions of problems in stress analysis, fluid flow, heat transfer, and other areas. FEM is applicable for a wide range of engineering problems including solid mechanics, dynamics, heat problems, fluids, and electrostatic problems. Most commercial FEM software packages originated in the 1970s (Abaqus, Adina, Ansys, etc.). FEM can handle very complex geometry, it is more easily adapted to the geometry of the underlying domain than the finite difference method. For symmetric positive definite elliptic problems, FEM is reduces to a finite linear system with a symmetric positive definite matrix.

The idea of FEM is based on the variational form of the model problem and approximates the exact solution by a piecewise polynomial function. The structure or the physical domain will be broke up into several elements (cell) of simple geometric structure such as segments, triangles, quadrilaterals, tetrahedrons, etc. Then reconnects elements at "nodes" as if nodes were pins or drops of glue that hold elements together. This process results in a system of simultaneous algebraic equations. Solving the algebraic system is often challenging especially when dealing with nonlinear models in three dimensions.

There are different classes of FEMs, in this and the forthcoming chapters, we focus on the piecewise-linear *continuous-conforming* FEM, the solution is a linear polynomial on each cell of the conforming finite element mesh of the physical domain. The first step towards this aim summarizes in introducing the weak formulation of the given model problem. Second, define the finite element spaces such as the trial set (where we sought the approximate solution) and the test space. Then, define the finite element solution. We shall be concerned with the mathematical aspects of finite element approximation, including stability and accuracy. The present chapter is devoted to discuss FEM for solving a steady-state model by starting with a simple model in the next section. The concepts and notations conventions introduced here will be used systematically in the next chapters.

5.1 FEM for a simple model

In this section, we discuss the FEM for solving the following simple two-point boundary-value problem:

$$-u'' = f(x) \quad \text{for } 0 < x < L, \quad \text{with } u(0) = \gamma_0 \text{ and } u(L) = \gamma_L. \tag{5.1}$$

5.1.1 Weak formulation

Multiply both sides of the above model by a *test function* v and integrate to obtain

$$-\int_0^L u''(x)v(x)\,dx = \int_0^L f(x)v(x)\,dx.$$

Assuming v' exists and is continuous on $[0, L]$, we can integrate by parts,

$$-\int_0^L u''(x)v(x)\,dx = -\big[u'(x)v(x)\big]_0^L + \int_0^L u'(x)v'(x)\,dx, \tag{5.2}$$

and conclude that

$$\int_0^L u'(x)v'(x)\,dx = \int_0^L f(x)v(x)\,dx \quad \text{provided } v(0) = 0 = v(L). \tag{5.3}$$

In the finite element method, this *weak formulation* is used as the basis for a discretisation of (5.1).

5.1.2 Finite element space

Suppose that we partition the interval $[0, L]$ into P subintervals uniformly (for simplicity) by choosing grid points, or *nodes*,

$$0 = x_0 < x_1 < x_2 < \cdots < x_P = L. \tag{5.4}$$

Denote the length of each subinterval, or *element*, by h. A function $\phi : [0, L] \to \mathbb{R}$ is *piecewise-linear* (with respect to the chosen grid points x_i) if there exist polynomials ϕ_1, ϕ_2, …, ϕ_P, each of degree at most 1, such that

$$\phi(x) = \phi_i(x) \quad \text{for } x_{i-1} < x < x_i \text{ and } 1 \leqslant i \leqslant P. \tag{5.5}$$

Such a function ϕ is continuous on $[0, L]$ if and only if the P polynomials satisfy

$$\phi_i(x_i) = \phi_{i+1}(x_i) \quad \text{for } 1 \leqslant i \leqslant P - 1. \tag{5.6}$$

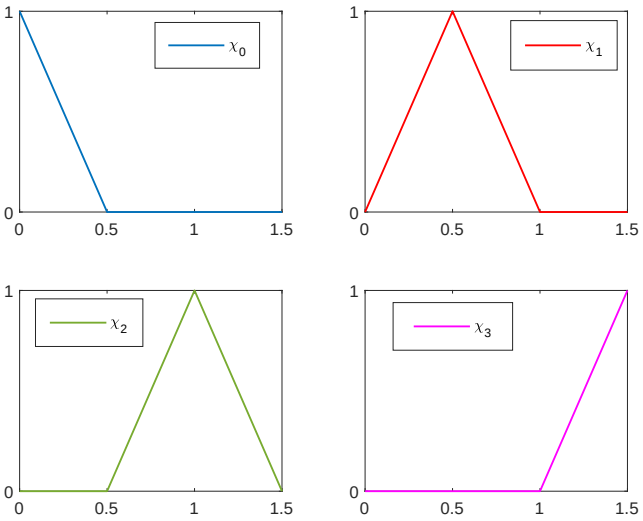
Let

$$V_h = \{\phi \in C[0, L],\ \phi|_{[x_i, x_{i+1}]} \text{ is a linear polynomial}\},$$

that is, V_h is the (vector) space of all continuous, piecewise-linear functions. Noting that, $\dim V_h = P + 1$, and $\{\chi_0, \cdots, \chi_P\}$ forms a set of hat basis functions of V_h, where these functions are defined by:

$$\chi_i(x) = \begin{cases} (x - x_{i-1})/h, & \text{for } x_{i-1} \leqslant x \leqslant x_i, \\ (x_{i+1} - x)/h, & \text{for } x_i \leqslant x \leqslant x_{i+1}, \\ 0, & \text{otherwise,} \end{cases} \tag{5.7}$$

Figure 5.1: An example of hat basis functions with $P = 3$, and $x_i = 0.5\,i$ for $i = 0, 1, 2, 3$.



whereas

$$\chi_0(x) = \begin{cases} (x_1 - x)/h, & \text{for } 0 = x_0 \leqslant x \leqslant x_1, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\chi_P(x) = \begin{cases} (x - x_{P-1})/h, & \text{for } x_{P-1} \leqslant x \leqslant x_P = L, \\ 0, & \text{otherwise.} \end{cases}$$

For the graphical illustration of the above hat basis functions, see the above figure when $P = 3$. So, any function $v \in V_h$, can be written as:

$$v(x) = \sum_{i=0}^{P} c_i \chi_i(x) \quad \text{for } 0 \leqslant x \leqslant L\,,$$

for some constants $c_0, c_1, \cdots, c_P$. Since, $\chi_i(x_i) = 1$ and $\chi_i(x_q) = 0$ whenever $q \neq i$, v has the representation

$$v(x) = \sum_{i=0}^{P} v(x_i)\chi_i(x) \quad \text{for } 0 \leqslant x \leqslant L\,.$$

5.1.3 Finite element solution

In the FEM of our model problem (5.1), we define the *solution set* (or *trial set*)

$$S_h = \{v \in V_h : v(0) = \gamma_0 \text{ and } v(L) = \gamma_L\} \tag{5.8}$$

and the *test space*

$$T_h = \{v \in V_h : v(0) = 0 \text{ and } v(L) = 0\}.$$

Notice that T_h is subspace of V_h with $\dim T_h = P - 1$ and $\{\chi_1, \cdots, \chi_{P-1}\}$ forms a set of basis functions of T_h, whereas S_h is not a subspace unless $\gamma_0 = 0 = \gamma_L$.

Based on the weak formulation in (5.3), the *finite element solution* $u_h \in S_h$ of (5.1) is determined by requiring that

$$\int_0^L u_h'(x)v'(x)\,dx = \int_0^L f(x)v(x)\,dx \quad \text{for all } v \in T_h. \tag{5.9}$$

5.1.4 Matrix form

Since $\{\chi_i\}_{i=1}^{P-1}$ is a basis for T_h, u_h satisfies

$$\int_0^L u_h'(x)\chi_i'(x)\,dx = \int_0^L f(x)\chi_i(x)\,dx \quad \text{for } 1 \leqslant i \leqslant P-1. \tag{5.10}$$

Moreover,

$$u_h(x) = \sum_{j=0}^{P} \alpha_j \chi_j(x) \quad \text{for } 0 \leqslant x \leqslant L, \quad \text{where } \alpha_j = u_h(x_j), \tag{5.11}$$

so

$$\int_0^L u_h'(x)\chi_i'(x)\,dx = \int_0^L \left(\sum_{j=0}^{P} \alpha_j \chi_j'(x)\right)\chi_i'(x)\,dx = \sum_{j=0}^{P} \alpha_j \left(\int_0^L \chi_i'(x)\chi_j'(x)\,dx\right).$$

Therefore, by letting

$$\mathbf{a}_{i,j} = \int_0^L \chi_i'(x)\chi_j'(x)\,dx \quad \text{and} \quad \mathbf{f}_i = \int_0^L f(x)\chi_i(x)\,dx,$$

we can write (5.10) as

$$\sum_{j=0}^{P} \mathbf{a}_{i,j}\alpha_j = \mathbf{f}_i \quad \text{for } 1 \leqslant i \leqslant P-1.$$

Since $u_h \in S_h$, we have $\alpha_0 = u_h(x_0) = u_h(0) = \gamma_0$ and $\alpha_P = u_h(x_P) = u_h(L) = \gamma_L$, leading to a $(P-1)\times(P-1)$ linear system for the unknowns $\alpha_1, \alpha_2, \ldots, \alpha_{P-1}$, namely

$$\sum_{j=1}^{P-1} \mathbf{a}_{i,j}\alpha_j = \mathbf{f}_i - a_{i,0}\gamma_0 - a_{i,P}\gamma_L \quad \text{for } 1 \leqslant i \leqslant P-1. \tag{5.12}$$

The α_j are called the *nodal values* of u_h, and the coefficients $\mathbf{a}_{i,j}$ form the *stiffness matrix*. From (5.12), we reach the following system

$$\frac{1}{h}\mathbf{A}\alpha = \mathbf{f} + \frac{1}{h}\mathbf{g}, \tag{5.13}$$

where the $(P-1)$-by-$(P-1)$ matrix $\mathbf{A} = [\mathbf{a}_{i,j}]$,

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{P-2} \\ \alpha_{P-1} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_{P-2} \\ \mathbf{f}_{P-1} \end{bmatrix}, \quad \text{and} \quad \mathbf{g} = \begin{bmatrix} \gamma_0 \\ 0 \\ \vdots \\ 0 \\ \gamma_L \end{bmatrix}.$$

Since the supports of χ_i' and χ_j' overlap if and only if $|j-i| \leqslant 1$; otherwise, if $|j-i| \geqslant 2$, then $\mathbf{a}_{i,j} = 0$. To compute the non-zero entries in the tridiagonal matrix $\mathbf{A}$, we see from (5.7) that if $1 \leqslant i \leqslant P-1$ then

$$\chi_i'(x) = \begin{cases} 1/h, & \text{for } x_{i-1} < x < x_i, \\ -1/h, & \text{for } x_i < x < x_{i+1}, \\ 0, & \text{otherwise,} \end{cases}$$

with the first case missing if $i = 0$, and the second if $i = P$. The diagonal values $\mathbf{A}$ are

$$a_{0,0} = \int_0^L \chi_0'(x)^2\,dx = \int_{x_0}^{x_1} \left(\frac{1}{h}\right)^2 dx = \frac{x_1 - x_0}{h^2} = \frac{1}{h}$$

and

$$a_{i,i} = \int_{x_{i-1}}^{x_i} \left(\frac{1}{h}\right)^2 dx + \int_{x_i}^{x_{i+1}} \left(\frac{-1}{h}\right)^2 dx = \frac{2}{h} \quad \text{for } 1 \leqslant i \leqslant P-1,$$

with

$$a_{P,P} = \int_0^L \chi_P'(x)^2 \, dx = \int_{x_{P-1}}^{x_P} \left(\frac{-1}{h}\right)^2 dx = \frac{x_P - x_{P-1}}{h^2} = \frac{1}{h}.$$

The off-diagonal values are

$$a_{i-1,i} = \int_0^L \chi_i'(x)\chi_{i-1}'(x) \, dx = \int_{x_{i-1}}^{x_i} \left(\frac{-1}{h}\right)\left(\frac{1}{h}\right) dx = \frac{-1}{h}, \quad \text{for } 1 \leqslant i \leqslant P.$$

Therefore, the $(P-1)$-by-$(P-1)$ matrix

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}.$$

5.1.5 Existence and uniqueness

The $(P-1) \times (P-1)$ stiffness matrix $\mathbf{A}$ is tridiagonal and symmetric. Furthermore, $\mathbf{A}$ is positive-definite (see the proof below), so, it is non-singular. Therefore, the finite element linear system in (5.13) is uniquely solvable for α, which can be computed using the algorithms described in section 2.1.1. Consequently, the finite element solution $u_h(x) = \sum_{j=0}^{P} \alpha_j \chi_j(x)$ exists and is unique.

To show that $\mathbf{A}$ is positive-definite, let $\mathbf{V} = [w_1, w_2, \cdots, w_{P-1}]^T$ be a non-zero column vector. One can show that

$$\mathbf{V}^T\mathbf{A}\mathbf{V} = w_1^2 + \sum_{i=1}^{P-2} (w_{i+1} - w_i)^2 + w_{P-1}^2 \,,$$

and so, $\mathbf{V}^T\mathbf{A}\mathbf{V} > 0$, as required.

Here is an alternative approach for showing that $\mathbf{A}$ is positive-definite. We define $\psi \in S_h$ by $\psi(x) = \sum_{i=1}^{P-1} w_i \chi_i(x)$ and observe that

$$\begin{aligned} \mathbf{V}^T\mathbf{A}\mathbf{V} &= \sum_{j=1}^{P-1}\sum_{i=1}^{P-1} w_j a_{j,i} w_i = \sum_{j=1}^{P-1}\sum_{i=1}^{P-1} w_j w_i \int_0^L \chi_i'(x)\chi_j'(x) \, dx \\ &= \int_0^L \left(\sum_{q=0}^{P-1} w_i \chi_i'(x)\right)\left(\sum_{p=0}^{P-1} w_j \chi_j'(x)\right) dx = \int_0^L \big(\psi'(x)\big)^2 \, dx \geqslant 0. \end{aligned}$$

If $\int_0^L \big(\psi'(x)\big)^2 \, dx = 0$ so $\psi' = 0$ and thus ψ is constant on $[0, L]$. Since $\psi(0) = 0 = \psi(L)$, the function ψ must be identically zero, and hence $w_i = 0$ for $1 \leqslant i \leqslant P-1$. This contradicts our assumption that $\mathbf{V}$ is a non-zero vector. Therefore, $\mathbf{V}^T\mathbf{A}\mathbf{V} > 0$ and consequently, $\mathbf{A}$ is positive-definite.

5.2 Steady-state reaction-diffusion models

5.2.1 Model problem and finite element solution

Define the following a second-order linear differential operator $\mathcal{L}$ as:

$$\mathcal{L}u(x) = -\big(au'\big)'(x) + (c\,u)(x). \tag{5.14}$$

Here, the coefficients a and c are functions of x, and assumed to have the same properties as in Chapter 2; in particular, $a(x)$ must satisfy the lower bound (2.11). For such an $\mathcal{L}$, consider the following two-point boundary-value problem with *mixed boundary conditions*,

$$\mathcal{L}u(x) = f(x) \quad \text{for } 0 < x < L, \quad \text{with } u(0) = \gamma_0 \text{ and } a(L)u'(L) = \gamma_L. \tag{5.15}$$

Here, at the left end of the interval $[0, L]$ we have imposed a *Dirichlet boundary condition* that fixes the value of $u(0)$, whereas at the right end we have imposed a *Neumann boundary condition* that fixes the value of $u'(L)$.

Integration by parts implies that

$$\int_0^L \mathcal{L}u(x)v(x)\,dx = -\big[a(x)u'(x)v(x)\big]_0^L + \int_0^L \big(a(x)u'(x)v'(x) + c(x)u(x)v(x)\big)\,dx, \tag{5.16}$$

so any solution u of (5.15) must satisfy

$$\int_0^L \big(a(x)u'(x)v'(x) + c(x)u(x)v(x)\big)\,dx = \gamma_L v(L) + \int_0^L f(x)v(x)\,dx \quad \text{provided } v(0) = 0. \tag{5.17}$$

Before proceeding and for convenience, let us introduce the vector space $L^2(0, L)$ of square-integrable functions $v : [0, L] \to \mathbb{R}$ with its usual inner product

$$\langle v, w\rangle = \int_0^L v(x)w(x)\,dx,$$

together with the corresponding norm

$$\|v\| = \sqrt{\langle v, v\rangle} = \Big(\int_0^L |v(x)|^2\,dx\Big)^{1/2}.$$

With this notation, we can re-write (5.17) as

$$\langle au', v'\rangle + \langle cu, v\rangle = \gamma_L v(L) + \langle f, v\rangle \quad \text{provided } v(0) = 0. \tag{5.18}$$

Given nodes (5.4), we therefore define the solution set S_h and test space T_h by

$$S_h = \{\, v \in V_h : v(0) = \gamma_0 \,\} \quad \text{and} \quad T_h = \{\, v \in V_h : v(0) = 0 \,\}.$$

The finite element solution $u_h \in S_h$ satisfies

$$\langle a\,u_h', v'\rangle + \langle c\,u_h, v\rangle = \gamma_L v(L) + \langle f, v\rangle \quad \text{for all } v \in T_h. \tag{5.19}$$

Equivalently, since $\{\chi_i\}_{i=1}^P$ is a basis for T_h, we require

$$\langle a\,u_h', \chi_i'\rangle + \langle c\,u_h, \chi_i\rangle = \gamma_L \chi_i(L) + \langle f, \chi_i\rangle \quad \text{for } 1 \leqslant i \leqslant P.$$

Inserting the representation (5.11) yields the system of linear equations

$$\sum_{j=0}^{P} \big(\mathbf{a}_{i,j}\alpha_j + \mathbf{c}_{i,j}\alpha_j\big) = \gamma_L \chi_i(L) + \mathbf{f}_i \quad \text{for } 1 \leqslant i \leqslant P,$$

where

$$\mathbf{a}_{i,j} = \langle a\,\chi_j', \chi_i'\rangle, \quad \mathbf{c}_{i,j} = \langle c\,\chi_j, \chi_i\rangle, \quad \mathbf{f}_i = \langle f, \chi_i\rangle.$$

Moving $\alpha_0 = \gamma_0$ to the right-hand side leads to a $P \times P$ linear system,

$$\sum_{j=1}^{P} (\mathbf{a}_{i,j} + \mathbf{c}_{i,j})\alpha_j = \gamma_L \chi_i(L) + \mathbf{f}_i - (\mathbf{a}_{i,0} + \mathbf{c}_{i,0})\gamma_0 \quad \text{for } 1 \leqslant i \leqslant P,$$

or, in matrix notation,

$$\big(\mathbf{A} + \mathbf{C}\big)\alpha = \mathbf{f} + \mathbf{g}, \tag{5.20}$$

where $\mathbf{A} = [\mathbf{a}_{i,j}]_{i,j=1}^{P}$, $\mathbf{C} = [\mathbf{c}_{i,j}]_{i,j=1}^{P}$, and the column vectors $\alpha = [\alpha_i]_{i=1}^{P}$, $\mathbf{f} = [\mathbf{f}_i]_{i=1}^{P}$ and $\mathbf{g} = [g_i]_{i=1}^{P}$, with

$$g_1 = -(\mathbf{a}_{1,0} + \mathbf{c}_{1,0})\gamma_0, \qquad g_i = 0 \text{ for } 2 \leqslant i \leqslant P-1, \qquad g_P = \gamma_L.$$

We again refer to $\mathbf{A}$ as the stiffness matrix; $\mathbf{C}$ is called the *mass matrix*. On the right-hand side, $\mathbf{f}$ is called the *load vector*. This terminology reflects the historical origins of finite element methods in structural engineering.

As in the previous section, since the supports of χ_i and χ_j overlap if and only if $|j-i| \leqslant 1$; otherwise, if $|j-i| \geqslant 2$, then $\mathbf{a}_{i,j} = 0 = \mathbf{c}_{i,j}$. In addition, since $\mathbf{a}_{i,j} = \mathbf{a}_{j,i}$ and $\mathbf{c}_{i,j} = \mathbf{c}_{j,i}$, $\mathbf{A}$ and $\mathbf{C}$ are tridiagonal symmetric matrices.

5.2.2 Matrix assembly element-by-element

In the previous subsection, we used the nodal basis functions χ_j to set up the linear system (6.5), but this approach becomes very complicated in 2D or 3D, or even in 1D when using piecewise-polynomials of higher degree. Instead, a simpler method is to assemble the matrices $\mathbf{A}$ and $\mathbf{C}$, and the vector $\mathbf{f}$, element-by-element.

For $1 \leqslant i \leqslant P$, we put $n_1^i = x_{i-1}$ and $n_2^i = x_i$ so that the ith element is $[x_{i-1}, x_i] = [n_1^i, n_2^i]$. The *linear shape functions* for this element are defined by

$$\psi_1^i(x) = \frac{x_i - x}{h} \quad \text{and} \quad \psi_2^i(x) = \frac{x - x_{i-1}}{h} \quad \text{for } x_{i-1} \leqslant x \leqslant x_i,$$

and satisfy

$$\psi_j^i(n_k^i) = \delta_{jk} \quad \text{for } j,\, k \in \{1, 2\},$$

so that for any $v \in V_h$,

$$v(x) = v(n_1^i)\psi_1^i(x) + v(n_2^i)\psi_2^i(x) \quad \text{for } x \in [n_1^i, n_2^i].$$

The *element stiffness matrix* is defined by

$$\mathbf{A}^i = \begin{bmatrix} a_{11}^i & a_{12}^i \\ a_{21}^i & a_{22}^i \end{bmatrix} \quad \text{where} \quad a_{jk}^i = \int_{x_{i-1}}^{x_i} a(x)\big(\psi_k^i\big)'(x)\big(\psi_j^i\big)'(x)\,dx,$$

the *element mass matrix* by

$$\mathbf{C}^i = \begin{bmatrix} c_{11}^i & c_{12}^i \\ c_{21}^i & c_{22}^i \end{bmatrix} \quad \text{where} \quad c_{jk}^i = \int_{x_{i-1}}^{x_i} c(x)\psi_k^i(x)\psi_j^i(x)\,dx,$$

and the *element load vector* by

$$\mathbf{f}^i = \begin{bmatrix} f_1^i \\ f_2^i \end{bmatrix} \quad \text{where} \quad f_j^i = \int_{x_{i-1}}^{x_i} f(x)\psi_j^i(x)\,dx.$$

Example 5.1. *We will show that for the constant coefficients* $a(x) = 1$ *and* $c(x) = 1$*, the element stiffness and mass matrices are simply*

$$\mathbf{A}^i = \frac{1}{h}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \textit{and} \quad \mathbf{C}^i = \frac{h}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

In fact, since $(\psi_1^i)'(x) = -1/h$ *and* $(\psi_2^i)'(x) = 1/h$*, we see that*

$$a_{11}^i = a_{22}^i = \int_{x_{i-1}}^{x_i} \frac{1}{h^2}\,dx = \frac{x_i - x_{i-1}}{h^2} = \frac{1}{h}$$

and

$$a_{12}^i = a_{21}^i = \int_{x_{i-1}}^{x_i} \frac{-1}{h^2}\,dx = \frac{-1}{h}.$$

For the element mass matrix,

$$c_{11}^i = \int_{x_{i-1}}^{x_i} \left(\frac{x_i - x}{h}\right)^2 dx = \left[-\frac{(x_i - x)^3}{3h^2}\right]_{x_{i-1}}^{x_i} = \frac{(x_i - x_{i-1})^3}{3h^2} = \frac{h}{3}$$

and similarly

$$c_{22}^i = \int_{x_{i-1}}^{x_i} \left(\frac{x - x_{i-1}}{h}\right)^2 dx = \left[-\frac{(x - x_{i-1})^3}{3h^2}\right]_{x_{i-1}}^{x_i} = \frac{(x_i - x_{i-1})^3}{3h^2} = \frac{h}{3},$$

whereas, integrating by parts,

$$\begin{aligned} c_{12}^i = c_{21}^i &= \int_{x_{i-1}}^{x_i} \left(\frac{x_i - x}{h}\right)\left(\frac{x - x_{i-1}}{h}\right) dx \\ &= \frac{1}{h^2}\left(\left[(x_i - x)\frac{(x - x_{i-1})^2}{2}\right]_{x_{i-1}}^{x_i} - \int_{x_{i-1}}^{x_i} (-1)\frac{(x - x_{i-1})^2}{2}\,dx\right) \\ &= \frac{1}{2h^2}\int_{x_{i-1}}^{x_i} (x - x_{i-1})^2\,dx = \frac{1}{2h^2}\left[\frac{(x - x_{i-1})^3}{3}\right]_{x_{i-1}}^{x_i} = \frac{h}{6}. \end{aligned}$$

We enumerate the nodes of the mesh so that the *free nodes precede the fixed nodes*, where the latter are those at which the value of the solution is fixed by a Dirichlet boundary condition. For our problem (5.15), the only fixed node is x_0, so we put

$$n_i = x_i \quad \text{for } 1 \leqslant i \leqslant P, \quad \text{and} \quad n_{P+1} = x_0.$$

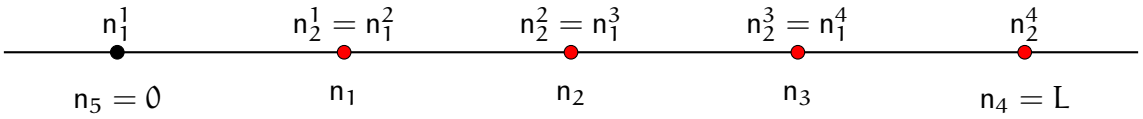
The $2 \times (P+1)$ *connectivity matrix* $\mathbf{T} = [t_{ji}]$ is defined by

$$t_{ji} = r \quad \text{iff} \quad n_j^i = n_r; \tag{5.21}$$

Example 5.2. *If* $P = 4$ *then*

$$\mathbf{T} = \begin{bmatrix} 5 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix}.$$

Figure 5.2: Node numbering from Example 5.2 with $P = 4$.



Writing $U_k^i = u_h(n_k^i)$ and $V_j^i = v(n_j^i)$, we see that

$$u_h(x) = \sum_{k=1}^{2} U_k^i \psi_k^i(x) \quad \text{and} \quad v(x) = \sum_{j=1}^{2} V_j^i \psi_j^i(x) \quad \text{for } x \in [x_{i-1}, x_i].$$

Thus,

$$\int_0^L f(x)v(x)\,dx = \sum_{i=1}^{P} \int_{x_{i-1}}^{x_i} f(x) \sum_{j=1}^{2} V_j^i \psi_j^i(x)\,dx = \sum_{i=1}^{P} \sum_{j=1}^{2} f_j^i V_j^i$$

and

$$\begin{aligned}\int_0^L a(x)u_h'(x)v'(x)\,dx &= \sum_{i=1}^{P} \int_{x_{i-1}}^{x_i} a(x) \left(\sum_{k=1}^{2} U_k^i (\psi_k^i)'(s) \right) \left(\sum_{j=1}^{2} V_j^i (\psi_j^i)'(x) \right) dx \\ &= \sum_{i=1}^{P} \sum_{k=1}^{2} \sum_{j=1}^{2} U_k^i a_{jk}^i V_j^i;\end{aligned}$$

likewise

$$\int_0^L c(x)u_h(x)v(x)\,dx = \sum_{i=1}^{P} \sum_{k=1}^{2} \sum_{j=1}^{2} U_k^i c_{jk}^i V_j^i \quad \text{and} \quad v(L) = V_2^i.$$

Therefore, (5.19) holds iff

$$\sum_{i=1}^{P} \sum_{j=1}^{2} V_j^i \sum_{k=1}^{2} (a_{jk}^i + c_{jk}^i) U_k^i = \gamma_L V_2^i + \sum_{i=1}^{P} \sum_{j=1}^{2} V_j^i f_j^i \quad \text{for all } v \in T_h.$$

For $v \in V_h$, put

$$V_r = v(n_r^i) \quad \text{for } 1 \leqslant r \leqslant P+1,$$

so that $V_r = V_j^i$ iff $r = t_{ji}$, and hence

$$\sum_{i=1}^{P+1} \sum_{j=1}^{2} V_j^i f_j^i = \sum_{r=1}^{P} V_r f_r,$$

where

$$f_r = \sum_{j \in \mathcal{J}_r} f_j^i \quad \text{and} \quad \mathcal{J}_r = \{\, (i,j) : t_{ji} = r \,\}.$$

Similarly, put $U_s = u_h(n_s)$ so that $U_s = U_k^i$ iff $s = t_{ki}$, and hence

$$\sum_{i=1}^{P} \sum_{j=1}^{2} V_j^i \sum_{k=1}^{2} (a_{jk}^i + c_{jk}^i) U_k^i = \sum_{r=1}^{P+1} V_r \sum_{s=1}^{P+1} (a_{rs} + c_{rs}) U_s$$

where

$$a_{rs} = \sum_{(i,j,k)\in\mathcal{I}_{rs}} a^i_{jk} \qquad \text{and} \qquad c_{rs} = \sum_{(i,j,k)\in\mathcal{I}_{rs}} c^i_{jk},$$

with

$$\mathcal{I}_{rs} = \{\,(i,j,k) : t_{ji} = r \text{ and } t_{ki} = s\,\}.$$

In this way, (5.19) holds iff

$$\sum_{r=1}^{P} V_r \sum_{s=1}^{P+1} (a_{rs} + c_{rs}) U_s = \gamma_L V_P + = \sum_{r=1}^{P} V_r f_r,$$

remembering that $V_{P+1} = v(n_{P+1}) = v(0) = 0$. Forming the $P \times (P+1)$ matrices $\boldsymbol{A} = [a_{i,j}]$ and $\boldsymbol{C} = [c_{i,j}]$, and the P-dimensional vector $\boldsymbol{f}$, we see that

$$\boldsymbol{V}^T(\boldsymbol{A} + \boldsymbol{C})\boldsymbol{U} = \gamma_L V_P + \boldsymbol{V}^T\boldsymbol{f} \quad \text{for all } \boldsymbol{V} \in \mathbb{R}^P, \tag{5.22}$$

and therefore

$$(\boldsymbol{A} + \boldsymbol{C})\boldsymbol{U} = \gamma_L \boldsymbol{e}_P + \boldsymbol{f}. \tag{5.23}$$

Partition the matrices as $\boldsymbol{A} = [\boldsymbol{A}^{\text{free}} \quad \boldsymbol{A}^{\text{fix}}]$ and $\boldsymbol{C} = [\boldsymbol{C}^{\text{free}} \quad \boldsymbol{C}^{\text{fix}}]$, where $\boldsymbol{A}^{\text{free}}$ and $\boldsymbol{C}^{\text{free}}$ are $P \times P$, and so $\boldsymbol{A}^{\text{fix}}$ and $\boldsymbol{C}^{\text{fix}}$ are $P \times 1$. Likewise partition the vector $\boldsymbol{U} = [\boldsymbol{U}^{\text{free}} \quad U^{\text{fix}}]^T$ where $\boldsymbol{U}^{\text{free}}$ is $P \times 1$ and so U^{fix} is 1×1, that is, a scalar. Since $U^{\text{fix}} = U_{P+1} = u_h(n_{P+1}) = u_h(0) = \gamma_0$,

$$(\boldsymbol{A} + \boldsymbol{C})\boldsymbol{U} = \begin{bmatrix}(\boldsymbol{A}^{\text{free}} + \boldsymbol{C}^{\text{free}}) & (\boldsymbol{A}^{\text{fix}} + \boldsymbol{C}^{\text{fix}})\end{bmatrix} \begin{bmatrix}\boldsymbol{U}^{\text{free}} \\ U^{\text{fix}}\end{bmatrix} = (\boldsymbol{A}^{\text{free}} + \boldsymbol{C}^{\text{free}})\boldsymbol{U}^{\text{free}} + \gamma_0(\boldsymbol{A}^{\text{fix}} + \boldsymbol{C}^{\text{fix}}),$$

yielding the $P \times P$ linear system

$$(\boldsymbol{A}^{\text{free}} + \boldsymbol{C}^{\text{free}})\boldsymbol{U}^{\text{free}} = \boldsymbol{f} - \gamma_0(\boldsymbol{A}^{\text{fix}} + \boldsymbol{C}^{\text{fix}}) + \gamma_L \boldsymbol{e}_P. \tag{5.24}$$

Algorithm 8 shows how, using the connectivity matrix $\boldsymbol{T} = [t_{pm}]$, the global load vector $\boldsymbol{f}$ can be assembled from the element load vectors $\boldsymbol{f}^i$. Likewise, algorithm 9 shows how the global stiffness matrix $\boldsymbol{A}$ can be assembled from the element stiffness matrices $\boldsymbol{A}^i$. The global mass matrix $\boldsymbol{C}$ is assembled in the same way. In practice, $\boldsymbol{A}$ and $\boldsymbol{C}$ are constructed an appropriate *sparse matrix* data structure.

Algorithm 8 Assemble the load vector $\boldsymbol{f}$ from (5.23).

```
Allocate storage for f = [f_i] ∈ R^P.
for i = 1 : P do
    f_i = 0
end for
for i = 1 : P do
    Compute f^i
    for j = 1 : 2 do
        r = t_ij                                   ▷ n_r = n^i_j
        if r ⩽ M then
            f_r ← f_r + f^i_j
        end if
    end for
end for
```

Algorithm 9 Assemble the stiffness matrix $\mathbf{A}$ from (5.23).

Allocate storage for $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{P\times(P+1)}$
for $i = 1 : P$ **do**
  for $j = 1 : P + 1$ **do**
    $a_{ij} = 0$
  end for
end for
for $i = 1 : P$ **do**
  Compute $\mathbf{A}^i$
  for $j = 1 : 2$ **do**
    $r = e_{ji}$
    if $r \leqslant P$ **then**
      for $k = 1 : 2$ **do**
        $s = t_{ki}$
        $a_{rs} \leftarrow a_{rs} + a^i_{jk}$
      end for
    end if
  end for
end for

Example 5.3. *Let* $L = 4$ *and consider a uniform grid with* $P = 4$ *subintervals, and suppose that* $a(x) = 1 = c(x)$. *Thus, by Example 5.1,*

$$h = 1, \quad \mathbf{A}^i = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{C} = \frac{1}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

and by Example 5.2,

$$\mathcal{I}_1 = \{(1,2),(2,1)\}, \quad \mathcal{I}_2 = \{(2,2),(3,1)\}, \quad \mathcal{I}_3 = \{(3,2),(4,1)\}, \quad \mathcal{I}_4 = \{(4,2)\},$$

so

$$f_1 = f^1_2 + f^2_1, \quad f_2 = f^2_2 + f^3_1, \quad f_3 = f^3_2 + f^4_1, \quad f_4 = f^4_2.$$

Furthermore,

$$\begin{aligned}
&\mathcal{I}_{11} = \{(1,2,2),(2,1,1)\}, && \mathcal{I}_{12} = \{(2,1,2)\}, && \mathcal{I}_{15} = \{(1,2,1)\},\\
&\mathcal{I}_{21} = \{(2,2,1)\}, && \mathcal{I}_{22} = \{(2,2,2),(3,1,1)\}, && \mathcal{I}_{23} = \{(3,1,2)\},\\
&\mathcal{I}_{32} = \{(3,2,1)\}, && \mathcal{I}_{33} = \{(3,2,2),(4,1,1)\}, && \mathcal{I}_{34} = \{(4,1,2)\},\\
&\mathcal{I}_{43} = \{(4,2,1)\}, && \mathcal{I}_{44} = \{(4,2,2)\}, &&
\end{aligned}$$

with $\mathcal{I}_{rs} = \emptyset$ *otherwise, and hence*

$$\begin{aligned}
&a_{11} = a^1_{22} + a^2_{11}, && a_{12} = a^2_{12}, && a_{15} = a^1_{21},\\
&a_{21} = a^2_{21}, && a_{22} = a^2_{22} + a^3_{11}, && a_{23} = a^3_{12},\\
&a_{32} = a^3_{21}, && a_{33} = a^3_{22} + a^4_{11}, && a_{34} = a^4_{12},\\
&a_{43} = a^4_{21}, && a_{44} = a^4_{22}. &&
\end{aligned}$$

Assembling the 4×5 *stiffness matrix* $\mathbf{A} = [\mathbf{A}^{\text{free}} \quad \mathbf{A}^{\text{fix}}]$ *amounts to computing*

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right] + \left[\begin{array}{cccc|c} 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right] + \left[\begin{array}{cccc|c} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right] + \left[\begin{array}{cccc|c} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{array}\right],$$

resulting in

$$\mathbf{A} = \left[\begin{array}{cccc|c} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{array}\right], \qquad \mathbf{A}^{\text{free}} = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \qquad \mathbf{A}^{\text{fix}} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Similarly, we find that the mass matrix $\mathbf{C} = [\mathbf{C}^{\text{free}} \quad \mathbf{C}^{\text{fix}}]$ *is given by*

$$\mathbf{C} = \frac{1}{6}\left[\begin{array}{cccc|c} 4 & 1 & 0 & 0 & 1 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 2 & 0 \end{array}\right], \qquad \mathbf{C}^{\text{free}} = \frac{1}{6}\begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}, \qquad \mathbf{C}^{\text{fix}} = \frac{1}{6}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

5.2.3 Polynomial interpolations and errors

In this subsection, the error from the finite element approximation in the preceding subsection will be discussed. We start by introducing some properties of the L_2 inner product, $\langle \cdot, \cdot \rangle$, and the associated L_2 norm, $\| \cdot \|$, in addition to some other useful inequalities. We also discuss the notion of linear interpolation which is needed in the finite element error analysis.

Properties and inequalities: Let w_1, w_2 and w_3 be in $L_2(0, L)$.

- Symmetric: $\langle w_1, w_2 \rangle = \langle w_2, w_1 \rangle$
- Bilinear: $\langle w_1 + w_2, w_3 \rangle = \langle w_1, w_3 \rangle + \langle w_2, w_3 \rangle$
- If $0 \leqslant w_1 \leqslant w_2$, then $\|w_1\| \leqslant \|w_2\|$
- Triangle inequality: $\|w_1 + w_2\| \leqslant \|w_1\| + \|w_2\|$
- Cauchy-Schwarz inequality: $|\langle w_1, w_2 \rangle| \leqslant \|w_1\| \, \|w_2\|$
- Poincaré inequality: $\|w_1\| \leqslant L\|w_1'\|$, ($L$ is the length of the interval $[0, L]$), provided that $w_1(0) = 0$ or $w_1(L) = 0$.

Linear interpolations: Let $g : [0, L] \to \mathbb{R}$ be a continuous function. We say that a piecewise linear polynomial $\hat{g} \in V_h$ *interpolates* g at the $x_0, x_1, \ldots, x_P$ if

$$\hat{g}(x_j) = g(x_j) \quad \text{for } 0 \leqslant j \leqslant P. \tag{5.25}$$

It is easy to see that such a $\hat{g}$ exists, explicitly, for $x_{i-1} \leqslant x \leqslant x_i$, with $1 \leqslant i \leqslant P$,

$$\hat{g}(x) = g(x_{i-1})\frac{x_i - x}{h} + g(x_i)\frac{x - x_{i-1}}{h} \tag{5.26}$$

However, for $x_{i-1} \leqslant x \leqslant x_i$, with $1 \leqslant i \leqslant P$,

$$g(x) = g(x)\frac{x_i - x}{h} + g(x)\frac{x - x_{i-1}}{h},$$

and so,

$$\hat{g}(x) - g(x) = [g(x_{i-1}) - g(x)]\frac{x_i - x}{h} + [g(x_i) - g(x)]\frac{x - x_{i-1}}{h}.$$

From Taylor theorem, we have

$$g(q) = g(x) + (q-x)g'(x) + \int_x^q (q-s)g''(s)\,ds\,.$$

Thus, applying this for $q = x_{i-1}$ and $q = x_i$,

$$\hat{g}(x) - g(x) = \frac{x_i - x}{h}\int_{x_{i-1}}^{x} (s - x_{i-1})g''(s)\,ds + \frac{x - x_{i-1}}{h}\int_x^{x_i} (x_i - s)g''(s)\,ds\,.$$

Consequently,

$$|\hat{g}(x) - g(x)| \leqslant h\int_{x_{i-1}}^{x} |g''(s)|\,ds + h\int_x^{x_i} |g''(s)|\,ds = h\int_{x_{i-1}}^{x_i} |g''(s)|\,ds, \quad \text{for } x_{i-1} \leqslant x \leqslant x_i\,.$$

Using this bound, we have

$$\begin{aligned}
\|\hat{g} - g\|^2 &= \sum_{i=1}^{P}\int_{x_{i-1}}^{x_i} |\hat{g}(x) - g(x)|^2\,dx \\
&\leqslant h^2\sum_{i=1}^{P}\int_{x_{i-1}}^{x_i}\Big(\int_{x_{i-1}}^{x_i}|g''(s)|\,ds\Big)^2\,dx \\
&\leqslant h^3\sum_{i=1}^{P}\int_{x_{i-1}}^{x_i}\int_{x_{i-1}}^{x_i}|g''(s)|^2\,ds\,dx \\
&= h^4\sum_{i=1}^{P}\int_{x_{i-1}}^{x_i}|g''(s)|^2\,ds = h^4\int_0^L |g''(s)|^2\,ds = h^4\|g''\|^2,
\end{aligned} \tag{5.27}$$

and so, the linear interpolation error is second-order accurate and satisfies:

$$\|\hat{g} - g\| \leqslant h^2\|g''\|\,. \tag{5.28}$$

Next, we show that

$$\|(\hat{g})' - g'\| \leqslant h\|g''\|, \tag{5.29}$$

which is also needed in the forthcoming finite element error estimates.

Differentiating both sides of (5.26), for $x_{i-1} < x < x_i$,

$$(\hat{g})'(x) - g'(x) = \frac{1}{h}[g(x_i) - g(x_{i-1})] - g'(x) = \frac{1}{h}\int_{x_{i-1}}^{x_i}[g'(s) - g'(x)]\,ds = \frac{1}{h}\int_{x_{i-1}}^{x_i}\int_x^s g''(q)\,dq\,ds$$

From this, we deduce that

$$|(\hat{g})'(x) - g'(x)| \leqslant \frac{1}{h}\int_{x_{i-1}}^{x_i}\int_{x_{i-1}}^{x_i}|g''(q)|\,dq\,ds = \int_{x_{i-1}}^{x_i}|g''(q)|\,dq\,.$$

Now, by following the steps in (5.27), the interpolation bound in (5.29) is achieved.

5.2.4 Convergence analysis

This subsection is devoted to investigate the error estimates from the finite element approximations. From the weak formulation in (5.18) and the finite element scheme in (5.19), we have

$$\langle a\,(u_h' - u'), v'\rangle + \langle c\,(u_h - u), v\rangle = 0 \quad \text{for all } v \in T_h. \tag{5.30}$$

Subtracting and adding $\hat{u}$ ($\hat{u} \in V_h$ interpolates u at the mesh nodes.),

$$\langle a\,(u_h' - \hat{u}'), v'\rangle + \langle c\,(u_h - \hat{u}), v\rangle = \langle a\,(u' - \hat{u}'), v'\rangle + \langle c\,(u - \hat{u}), v\rangle \quad \text{for all } v \in T_h.$$

Choosing $v = u_h - \hat{u}$ and then, using the inequality $\langle c\,(u_h - \hat{u}), u_h - \hat{u}\rangle = \|\sqrt{c}\,(u_h - \hat{u})\|^2 \geqslant 0$, in additions to the Poincaré inequality, we notice that

$$\|\sqrt{a}(u_h' - \hat{u}')\|^2 \leqslant C\|u' - \hat{u}'\|\,\|u_h' - \hat{u}'\| + C\|u - \hat{u}\|\,\|u_h - \hat{u}\| \leqslant C\|u' - \hat{u}'\|\,\|u_h' - \hat{u}'\|,$$

Recall that $a \geqslant a_{\min} > 0$, and $c \geqslant 0$. Using this and the interpolation errors, yield

$$\|u_h' - \hat{u}'\|^2 = \frac{1}{a_{\min}}\|\sqrt{a_{\min}}(u_h' - \hat{u}')\|^2 \leqslant \frac{1}{a_{\min}}\|\sqrt{a}(u_h' - \hat{u}')\|^2 \leqslant Ch\|u''\|\,\|u_h' - \hat{u}'\|,$$

and after simplifying,

$$\|u_h' - \hat{u}'\| \leqslant Ch\|u''\|\,.$$

However, by the triangle inequality,

$$\|u_h' - u'\| = \|(u_h' - \hat{u}') + (\hat{u}' - u')\| \leqslant \|u_h' - \hat{u}'\| + \|\hat{u}' - u'\|,$$

and so, we obtain the following convergence estimate.

Theorem 5.4. *Let u be the solution of problem (5.15) and let u_h be the finite solution defined by (5.19). Then, we have the following error estimate*

$$\|u_h' - u'\| \leqslant Ch\|u''\|\,.$$

We show in the next theorem that the finite element solution u_h is second order accurate.

Theorem 5.5. *Let u be the solution of problem (5.15) and let u_h be the finite solution defined by (5.19). Then, we have the following error estimate*

$$\|u_h - u\| \leqslant Ch^2\|u''\|\,.$$

Proof. A duality argument will be used in the proof. Let $e = u_h - u$ and let θ be the solution of the boundary-value problem

$$\mathcal{L}\theta = e \quad \text{for } 0 < x < L, \quad \text{with } \theta(0) = 0 \text{ and } \theta'(L) = 0\,.$$

Taking the inner product of both sides with e, then, integrating by parts and using $e(0) = u_h(0) - u(0) = \gamma_0 - \gamma_0 = 0$, we reach

$$\langle a\,e', \theta'\rangle + \langle c\,e, \theta\rangle = \|e\|^2\,. \tag{5.31}$$

Recall that, from (5.18) and (5.19),

$$\langle a\,e', v'\rangle + \langle c\,e, v\rangle = 0, \quad \text{for } \ v \in T_h.$$

Choose $v = \hat{\theta}$ ($\hat{\theta} \in S_h$, interpolates θ at the mesh nodes) leads to

$$\langle a\,e', \hat{\theta}'\rangle + \langle c\,e, \hat{\theta}\rangle = 0. \tag{5.32}$$

Subtracting (5.32) from (5.31),

$$\begin{aligned}
\|e\|^2 &= \langle a\,e', \theta' - \hat{\theta}'\rangle + \langle c\,e, \theta - \hat{\theta}\rangle \\
&\leqslant \|a\,e'\|\,\|\theta' - \hat{\theta}'\| + \|c\,e\|\,\|\theta - \hat{\theta}\| \\
&\leqslant a_{\max}\|\,e'\|\,\|\theta' - \hat{\theta}'\| + c_{\max}\,L^2\|e'\|\,\|\theta' - \hat{\theta}'\| \\
&\leqslant Ch^2\|u''\|\,\|\theta''\|\,.
\end{aligned}$$

Since $(a\theta')' = c\theta - e$, $\theta'' = (c\theta - a'\theta' - e)/a$. Owing to this and also to the assumption that $a(x) \geqslant a_{\min} > 0$, we deduce that

$$\|\theta''\| \leqslant C\Big(\|\theta\| + \|\theta'\| + \|e\|\Big) \leqslant C(\|\theta'\| + \|e\|)\,.$$

On the other hand,

$$a_{\min}\|\theta'\|^2 \leqslant \|\sqrt{a}\theta'\|^2 + \|\sqrt{c}\theta\|^2 = \langle -(a\theta')' + c\theta, \theta\rangle = \langle e, \theta\rangle \leqslant \|e\|\,\|\theta\| \leqslant C\|e\|\,\|\theta'\|,$$

which implies, $\|\theta'\| \leqslant C\|e\|$. Thus,

$$\|\theta''\| \leqslant C\|e\|\,.$$

Therefore,

$$\|e\|^2 \leqslant Ch^2\|u''\|\,\|\theta''\| \leqslant Ch^2\|u''\|\,\|e\|\,.$$

Simplifying both sides by $\|e\|$ completes the proof. □

5.3 Numerical integration

In the case of variable coefficients and source term, computing the integrals occurring in the finite element solution process analytically (exactly) is often not feasible. To approximate them, an additional error is expected. It is important to make sure that the error from the finite element discretization is dominant, so the second order convergence rate can be maintained. One option is to use the 2-point composite Gauss quadrature rules which is in principle fourth order accurate. To proceed with this choice, and for convenience, we introduce first the 2-point Gauss quadrature rule on the interval $[0, 1]$. It is defined as: for a given function g,

$$\int_0^1 g(\xi)\,d\xi \approx \sum_{i=1}^{2} w_i g(\xi_i)$$

with *weights* w_1 and w_2, and *points* $0 < \xi_1 < \xi_2 < 1$. To compute the four unknowns w_1, w_2, ξ_1 and ξ_2, the Gauss quadrature rule is assumed to be exact for all polynomials of degree $\leqslant 3$, that is,

$$\int_0^1 (\xi)^j\,d\xi = \sum_{i=1}^{2} w_i(\xi_i)^j, \quad \text{for } j = 0, 1, 2, 3\,.$$

These leads to the following four equations:

$$w_1 + w_2 = 1, \;\; w_1\xi_1 + w_2\xi_2 = \frac{1}{2}, \;\; w_1\xi_1^2 + w_2\xi_2^2 = \frac{1}{3}, \;\; w_1\xi_1^3 + w_2\xi_2^3 = \frac{1}{4}\,.$$

Solving this system implies that

$$w_1 = w_2 = \frac{1}{2}, \quad \xi_1 = \frac{1}{2}(1 - \frac{1}{\sqrt{3}}), \quad \xi_2 = \frac{1}{2}(1 + \frac{1}{\sqrt{3}})\,.$$

To define the composite quadrature rule on the interval $[0, L]$,, we apply the above approach on each cell of the finite element mesh. That is,

$$\int_0^L g(x)\,dx = \sum_{j=1}^{P} \int_{x_{j-1}}^{x_j} g(x)\,dx\,.$$

The change of variable $\xi = \frac{x - x_{j-1}}{h}$ leads to

$$\int_{x_{j-1}}^{x_j} g(x)\,dx = h\int_0^1 g(x_{j-1} + h\xi)\,d\xi \approx \sum_{i=1}^{2} h w_i g(x_{j-1} + h\xi_i),$$

and hence,

$$\int_0^L g(x)\,dx = \sum_{j=1}^{P}\int_{x_{j-1}}^{x_j} g(x)\,dx \approx h\sum_{j=1}^{P}\sum_{i=1}^{2} w_i g(\xi_{i,j}), \quad \text{with} \;\; \xi_{i,j} = x_{j-1} + h\xi_i\,.$$

The above composite quadrature approximation satisfies the estimate

$$\Big|\int_0^L g(x)\,dx - h\sum_{j=1}^{P}\sum_{i=1}^{2} w_i g(\xi_{i,j})\Big| \leqslant Ch^4|g^{(4)}(\zeta)|, \quad \text{for some} \;\; \zeta \in [0, L]\,.$$

Now, we are ready to approximate the occurred integrals in the entries $\mathbf{a}_{i,j}$, $\mathbf{c}_{i,j}$, and $\mathbf{f}_i$ of the matrices $\mathbf{A}$, $\mathbf{C}$, and $\mathbf{f}$, respectively, as follows:

$$\mathbf{a}_{i,j} = \langle a\chi_j', \chi_i'\rangle \approx h\sum_{q=1}^{P}\sum_{\ell=1}^{2} w_\ell (a\chi_j'\chi_i')(\xi_{\ell,q}) =: \mathbf{a}_{i,j}^h,$$

$$\mathbf{c}_{i,j} = \langle c\chi_j, \chi_i\rangle \approx h\sum_{q=1}^{P}\sum_{\ell=1}^{2} w_\ell (c\chi_j\chi_i)(\xi_{\ell,q}) =: \mathbf{c}_{i,j}^h,$$

$$\mathbf{f}_i = \langle f, \chi_i\rangle \approx h\sum_{q=1}^{P}\sum_{\ell=1}^{2} w_\ell (f\chi_i')(\xi_{\ell,q}) =: \mathbf{f}_i^h,$$

To estimate the errors in the terms $|\mathbf{a}_{i,j} - \mathbf{a}_{i,j}^h|$, $|\mathbf{c}_{i,j} - \mathbf{c}_{i,j}^h|$ and $|\mathbf{f}_i - \mathbf{f}_i^h|$, we may use the following result: for any piecewise polynomial Q_m of degree $\leqslant m$ on each finite mesh element, with $0 \leqslant m \leqslant 2$, we have

$$\Big|\int_0^L g(x)Q_m(x)\,dx - h\sum_{j=1}^{P}\sum_{i=1}^{2} w_i (g\,Q_m)(\xi_{i,j})\Big| \leqslant Ch^s\|g^{(s)}\|\,\|Q_m\|, \quad \text{with} \;\; s + m \leqslant 4,$$

where $g^{(s)}$ is the s^{th} derivative of g.

5.4 Exercises

5.1. Show that the trial set (5.8) is a *convex subset* of V_h, that is, if v, $w \in S_h$ then $\lambda v + \mu w \in S_h$ for all $\lambda \geqslant 0$ and $\mu \geqslant$ such that $\lambda + \mu = 1$.

5.2. Consider a differential operator $\mathcal{L}u = -(au')' + bu' + cu$. The weak formulation of the mixed boundary-value problem (5.15) changes from (5.17) to

$$\langle au', v'\rangle + \langle bu', v\rangle + \langle cu, v\rangle = \gamma_L v(L) + \langle f, v\rangle \quad \text{provided } v(0) = 0,$$

with $u(0) = \gamma_0$, and the linear system (6.5) is now of the form

$$(\mathbf{A} + \mathbf{B} + \mathbf{C})\mathbf{U} = \mathbf{f} + \mathbf{g}.$$

(i) How are the entries $b_{i,j}$ of the matrix $\mathbf{B}$ defined in terms of the hat basis functions?

(ii) Assume now that $b(x) \equiv 1$ and we use piecewise-linear elements with $P = 5$. Compute all entries of the matrix $\mathbf{B}$ in the case $P = 5$.

5.3. Consider a quadrature rule of the form

$$\int_{-1}^{1} f(x)\,dx \approx w_1 f(-1) + w_2 f(-a) + w_2 f(a) + w_1 f(1) \quad \text{with } 0 < a < 1,$$

and denote the error by

$$Ef = \int_{-1}^{1} f(x)\,dx - \big(w_1 f(-1) + w_2 f(-a) + w_2 f(a) + w_1 f(1)\big).$$

(i) Why does $Ef = 0$ if f is an odd function, that is, if $f(-x) = -f(x)$?

(ii) Simplify Ef in the case when f is an even function, that is, when $f(-x) = f(x)$.

(iii) Determine the values of w_1, w_2 and a for which $Ef = 0$ for any polynomial f of degree $\leqslant 5$.

5.4. Repeat the steps of Exercise 5.3 for a quadrature rule of the form

$$\int_{-1}^{1} f(x)\,dx \approx w_1 f(-a) + w_2 f(0) + w_1 f(a).$$

5.5. Consider the following steady-state model:

$$-(au')'(x) = \sin(x), \quad \text{for } 0 < x < \pi,$$

with $u(0) = u(\pi) = 0$, and where $a(x) = 1/(x+1)$. The exact solution of this model is

$$u(x) = (1+x)\sin(x) + \cos(x) + \frac{2}{\pi(\pi+2)}x(x+2) - 1.$$

(i) Define the piecewise linear finite element solution u_h using a uniform mesh consists of P elements.

(ii) Compare graphically between u and u_h when $P = 5$.

(iii) Compute the maximum nodal errors and the convergence rates when $P = 10, 20, 40, 80, 160$.

5.6. Consider the following steady-state model:

$$-(au')'(x) + \pi^2 u(x) = \pi\sin(\pi x) + \pi^2(2+x)\cos(\pi x), \quad \text{for } 0 < x < 1,$$

with $u'(0) = u'(1) = 0$, and where $a(x) = x + 1$. The exact solution of this model is $u(x) = \cos(\pi x)$.

(i) Define the weak formulation of this model.

(ii) Define the piecewise linear finite element solution u_h using a uniform mesh consists of P elements.

(iii) Compare graphically between u and u_h when $P = 5$.

(iv) Compute the maximum nodal errors and the convergence rates when $P = 10, 20, 40, 80, 160$.

5.7. Consider the following model:

$$-u''(x) - 2u'(x) = f(x) \quad \text{for } 0 < x < 1, \quad \text{with } u(0) = 0 \text{ and } u'(1) + u(1) = 0,$$

where f is a given smooth function.

(i) Show the following weak formulation

$$\langle u' + 2u, v' \rangle = u(1)\, v(1) + \langle f, v \rangle, \quad \text{provided that } v(0) = 0.$$

(ii) Define the finite element solution u_h on a uniform mesh with a step size $h = 1/P$, including the description of V_h, S_h and T_h.

(iii) By a simple integration, verify that $2\langle u_h, u_h' \rangle = \Big(u_h(1)\Big)^2$.

(iv) Use the above part and the inequality $\|u_h\| \leqslant \|u_h'\|$ to show the stability property: $\|u_h\| \leqslant \|f\|$.

(v) Write the numerical scheme in a matrix form.

(vi) Show the existence and uniqueness of u_h.

Chapter 6

Finite elements for time-dependent diffusion models

In this chapter, we focus on the numerical solution of the following time-dependent diffusion model with mixed boundary conditions,

$$\begin{aligned} u_t(x,t) + \mathcal{L}u(x,t) &= f(x,t) \quad \text{for } 0 < x < L \text{ and } 0 < t < T, \\ u(0,t) &= \gamma_0(t) \quad \text{for } 0 < t < T, \\ a(L)u_x(L,t) &= \gamma_L(t) \quad \text{for } 0 < t < T, \\ u(x,0) &= u_0(x) \quad \text{for } 0 < x < L, \end{aligned} \tag{6.1}$$

where $\mathcal{L}u = -\big(a(x)u_x\big)_x + c(x)u$ as before in (5.14). Of course, the developed numerical approach can be easily extended for the cases of pure Dirichlet or pure Neumann boundary conditions. We discretize in space via piecewise linear finite elements, while (first order) forward-backward Euler and Crank-Nicolson methods will be used for the time discretization.

6.1 Numerical method

First, we discretize in space via FEM which will define a semidiscrete scheme. The starting point here is the weak formulation of (6.1).

Recall that, $\langle \cdot, \cdot \rangle$ is the $L_2(0,L)$ inner product, that is, $\langle \phi, \psi \rangle = \int_0^L \phi(x)\psi(x)\,dx$ for any ϕ, $\psi \in L_2(0,L)$.

The identity (5.16) implies that, for any test function $v(x)$,

$$\langle u_t, v \rangle + \langle a u_x, v_x \rangle + \langle cu, v \rangle = \gamma_L(t)v(L) + \langle f, v \rangle \quad \text{provided } v(0) = 0. \tag{6.2}$$

We will use this relation to formulate a semidiscrete finite element solution $u_h(x,t) \approx u(x,t)$ and then apply finite difference approximations in time to derive some fully-discrete schemes.

Let V_h denote the space of continuous, piecewise-linear functions for a given mesh (5.4) on the spatial interval $[0,L]$. We define the solution set (now dependent on t) and test space by

$$S_h(t) = \{ v \in V_h : v(0) = \gamma_0(t) \} \quad \text{and} \quad T_h = \{ v \in V_h : v(0) = 0 \}.$$

We also choose $u_{0h} \in V_h$ such that $u_{0h} \approx u_0$; the simplest choice would be the piecewise linear interpolant $u_{0h} = \hat{u}_0$. The semidiscrete finite element solution $u_h(x,t)$ is then defined for $0 \leqslant t \leqslant T$ by requiring that $u_h(\cdot,t) \in S_h(t)$ and

$$\langle (u_h)_t, v \rangle + \langle a(u_h)_x, v_x \rangle + \langle cu_h, v \rangle = \gamma_L(t)v(L) + \langle f, v \rangle \quad \text{for all } v \in T_h, \tag{6.3}$$

with $u_h(0) = u_{0h}$.

To discretize in time, putting

$$t_n = n\tau \quad \text{for } 0 \leqslant n \leqslant N, \quad \text{where } \tau = \frac{T}{N},$$

and seeking $U_h^n \in S_h^n$ to approximate $U_h(t_n)$ and consequently, approximating $u(t_n)$, where

$$S_h^n = \{ v \in V_h : v(0) = \gamma_0(t_n) \}$$

For $\sigma \in \{0, 1/2, 1\}$, $U_h^n \approx u(\cdot, t_n)$ is defined as: for $1 \leqslant n \leqslant N$,

$$\frac{1}{\tau}\langle U_h^n - U_h^{n-1}, v\rangle + \langle a(U_h)_x^{n-\sigma}, v_x\rangle + \langle cU_h^{n-\sigma}, v\rangle = \gamma_L^{n-\sigma} v(L) + \langle f^{n-\sigma}, v\rangle \quad \text{for all } v \in T_h, \tag{6.4}$$

with $U_h^0 = u_{0h}$. For $\sigma \in \{0, 1\}$,

$$\gamma_L^j = \gamma_L(t_j), \quad f^j = f(t_j), \quad \text{for } j = 0, 1, \cdots, N.$$

For $\sigma = 1/2$, with $t_{n-1/2} = (t_n + t_{n-1})/2$,

$$(U_h)_x^{n-\sigma} = \frac{(U_h)_x^n + (U_h)_x^{n-1}}{2}, \quad \gamma_L^{n-\sigma} = \frac{\gamma_L(t_n) + \gamma_L(t_{n-1})}{2} \text{ or } \gamma_L(t_{n-1/2}),$$

and

$$f^{n-\sigma} = \frac{f(t_n) + f(t_{n-1})}{2} \text{ or } f(t_{n-1/2}).$$

For $\sigma = 0, 1/2$ and 1, our numerical scheme (in time) reduces to (first order) backward Euler, Crank-Nicolson, and (first order) forward Euler methods, respectively.

6.2 Matrix form

Recall that $\{\chi_i\}_{i=1}^P$ are the hat basis for T_h, so, our scheme in (6.4) is equivalent to: for $1 \leqslant n \leqslant N$,

$$\frac{1}{\tau}\langle U_h^n - U_h^{n-1}, \chi_i\rangle + \langle a(U_h)_x^{n-\sigma}, (\chi_i)_x\rangle + \langle cU_h^{n-\sigma}, \chi_i\rangle = \gamma_L^{n-\sigma}\chi_i(L) + \langle f^{n-\sigma}, \chi_i\rangle \quad \text{for } 1 \leqslant i \leqslant P.$$

Inserting the following representation

$$U_h^n(x) = \sum_{j=0}^{P} \alpha_j^n \chi_j(x) \quad \text{for } 0 \leqslant x \leqslant L, \quad \text{where } \alpha_j^n = U_h^n(x_j),$$

yields the system of linear equations

$$\sum_{j=0}^{P} \left(\mathbf{d}_{i,j}[\alpha_j^n - \alpha_j^{n-1}] + \tau[\mathbf{a}_{i,j} + \mathbf{c}_{i,j}]\alpha_j^{n-\sigma} \right) = \tau\gamma_L^{n-\sigma}\chi_i(L) + \tau\mathbf{f}_i^{n-\sigma} \quad \text{for } 1 \leqslant i \leqslant P,$$

where

$$\mathbf{d}_{i,j} = \langle \chi_j, \chi_i\rangle, \quad \mathbf{a}_{i,j} = \langle a\,(\chi_j)_x, (\chi_i)_x\rangle, \quad \mathbf{c}_{i,j} = \langle c\,\chi_j, \chi_i\rangle, \quad \mathbf{f}_i^{n-\sigma} = \langle f^{n-\sigma}, \chi_i\rangle.$$

Since $\alpha_0^n = U_h^n\Big|_{x=0} = \gamma_0^n$ for $n = 0, \cdots, N$,

$$\sum_{j=1}^{P} \left(\mathbf{d}_{i,j}[\alpha_j^n - \alpha_j^{n-1}] + \tau[\mathbf{a}_{i,j} + \mathbf{c}_{i,j}]\alpha_j^{n-\sigma} \right)$$
$$= \tau\gamma_L^{n-\sigma}\chi_i(L) + \tau\mathbf{f}_i^{n-\sigma} - \left(\mathbf{d}_{i,0}[\gamma_0^n - \gamma_0^{n-1}] + \tau[\mathbf{a}_{i,0} + \mathbf{c}_{i,0}]\gamma_0^{n-\sigma} \right), \quad \text{for } 1 \leqslant i \leqslant P.$$

In a matrix notation,

$$\mathbf{D}[\alpha^n - \alpha^{n-1}] + \tau\big(\mathbf{A}+\mathbf{C}\big)\alpha^{n-\sigma} = \tau\mathbf{f}^{n-\sigma} + \tau\mathbf{g}^{n-\sigma}, \quad \text{for } 1 \leqslant n \leqslant N, \tag{6.5}$$

where the stiffness matrix $\mathbf{A}$ and mass matrix $\mathbf{C}$ are the same as in the stationary problem (section 5.2), $\mathbf{D}$ is obtained from $\mathbf{C}$ by replacing c with 1, and the column vectors $\alpha^n = [\alpha_i^n]_{i=1}^P$, $\mathbf{f}^{n-\sigma} = [\mathbf{f}_i^{n-\sigma}]_{i=1}^P$ and $\mathbf{g}^{n-\sigma} = [g_i^{n-\sigma}]_{i=1}^P$, with

$$g_1^{n-\sigma} = -\frac{\mathbf{d}_{1,0}}{\tau}(\gamma_0^n - \gamma_0^{n-1}) - (\mathbf{a}_{1,0}+\mathbf{c}_{1,0})\gamma_0^{n-\sigma}, \qquad g_i^{n-\sigma} = 0 \text{ for } 2 \leqslant i \leqslant P-1, \qquad g_P^{n-\sigma} = \gamma_L^{n-\sigma}.$$

From the initial approximation $U_h^0 = u_{0h}$, the column vector $\alpha^0 = [u_{0h}(x_i)]_{i=1}^P$.

Based on above, the *forward Euler method* ($\sigma = 1$) is

$$\mathbf{D}[\alpha^n - \alpha^{n-1}] + \tau\big(\mathbf{A}+\mathbf{C}\big)\alpha^{n-1} = \tau\mathbf{f}^{n-1} + \tau\mathbf{g}^{n-1}, \quad \text{for } 1 \leqslant n \leqslant N, \quad \text{with } \alpha^0 = [u_{0h}(x_i)]_{i=1}^P.$$

Notice that this method is not actually explicit, due to the presence of the matrix $\mathbf{C}$: at the nth time step we have to solve the linear system

$$\mathbf{D}\alpha^n = \big(\mathbf{D} - \tau\,(\mathbf{A}+\mathbf{C})\big)\alpha^{n-1} + \tau\,\big(\mathbf{f}^{n-1} + \mathbf{g}^{n-1}\big), \quad \text{for } 1 \leqslant n \leqslant N.$$

The *backward Euler method* ($\sigma = 0$) is

$$\mathbf{D}\,[\alpha^n - \alpha^{n-1}] + \tau(\mathbf{A}+\mathbf{C})\alpha^n = \tau\mathbf{f}^n + \tau\mathbf{g}^n \quad \text{for } 1 \leqslant n \leqslant N, \quad \text{with } \alpha^0 = [u_{0h}(x_i)]_{i=1}^P,$$

which requires that we solve the linear systems

$$\big(\mathbf{D} + \tau\,(\mathbf{A}+\mathbf{C})\big)\alpha^n = \mathbf{D}\alpha^{n-1} + \tau\,\big(\mathbf{f}^n + \mathbf{g}^n\big), \quad \text{for } 1 \leqslant n \leqslant N.$$

The *Crank–Nicolson method* ($\sigma = 1/2$) is is

$$\mathbf{D}\,[\alpha^n - \alpha^{n-1}] + \tau(\mathbf{A}+\mathbf{C})\alpha^{n-1/2} = \tau\mathbf{f}^{n-1/2} + \tau\mathbf{g}^{n-1/2} \quad \text{for } 1 \leqslant n \leqslant N, \quad \text{with } \alpha^0 = [u_{0h}(x_i)]_{i=1}^P,$$

which requires that we solve the linear systems

$$\big(\mathbf{D} + \tfrac{1}{2}\tau\,(\mathbf{A}+\mathbf{C})\big)\alpha^n = \big(\mathbf{D} - \tfrac{1}{2}\tau\,(\mathbf{A}+\mathbf{C})\big)\alpha^{n-1} + \tau\,\big(\mathbf{f}^{n-1/2} + \mathbf{g}^{n-1/2}\big).$$

6.3 Stability and error analysis

This section focuses on the stability and error analysis of the backward Euler finite element scheme, that is, for the case $\sigma = 0$. For simplicity, we assume that the reaction coefficient $c = 0$, and that $\gamma_0(t) = \gamma_L(t) = 0$. In this case,

$$S_h^n = \{\, v \in V_h : v(0) = 0 \,\} = S_h,$$

that is, S_h^n does not change with n anymore. Moreover, $S_h = T_h$. So, the finite element scheme in (6.4) can be reformulated as: Find $U_h^n \in S_h$ such that

$$\frac{1}{\tau}\langle U_h^n - U_h^{n-1}, v\rangle + \langle a(U_h)_x^n, v_x\rangle = \langle f^n, v\rangle \quad \text{for all } v \in S_h, \tag{6.6}$$

for $1 \leqslant n \leqslant N$, with $U_h^0 = u_{0h}$.

6.3.1 Stability

Choose $v = U_h^n$ in (6.6),

$$\|U_h^n\|^2 - \langle U_h^{n-1}, U_h^n\rangle + \tau\langle a(U_h)_x^n, (U_h)_x^n\rangle = \tau\langle f^n, U_h^n\rangle, \quad \text{for } 1 \leqslant n \leqslant N,$$

and then, rearranging the terms and using the fact that $\langle a(U_h)_x^n, (U_h)_x^n\rangle \geqslant 0$ in addition to the Cauchy-Schwarz inequality,

$$\|U_h^n\|^2 \leqslant \|U_h^{n-1}\|\,\|U_h^n\| + \tau\|f^n\|\,\|U_h^n\|\,.$$

After simplifying,

$$\|U_h^n\| \leqslant \|U_h^{n-1}\| + \tau\|f^n\|, \quad \text{for } 1 \leqslant n \leqslant N\,.$$

Summing over n, we obtain the following unconditional stability property:

$$\|U_h^n\| \leqslant \|u_{0h}\| + \tau\sum_{j=1}^{n}\|f^j\|, \quad \text{for } 1 \leqslant n \leqslant N\,.$$

Noting that the proof of the unconditional stability property of the Crank-Nicolson is more technical. As a hint, choose $v = U_h^{n-1/2}$ and use $2\langle U_h^n - U_h^{n-1}, U_h^{n-1/2}\rangle = \|U_h^n\|^2 - \|U_h^{n-1}\|^2$. Another thing, as we have previously seen in Chapter 3, the *forward Euler method*, associated with $\sigma = 1$, is conditionally stable, see Exercise 6.2 for more details.

6.3.2 Error estimates

We decompose the error as:

$$U_h^n - u(t_n) = [U_h^n - R_h u(t_n)] - [u(t_n) - R_h u(t_n)] =: \theta^n - \eta^n$$

where R_h is the Ritz projection on S_h defined by: for each fixed t,

$$\langle a(R_h u)_x, v_x\rangle = \langle a u_x, v_x\rangle, \quad \text{for all} \quad v \in S_h.$$

The Ritz projection satisfies the following error estimate:

$$\|u(t) - R_h u(t)\| \leqslant Ch^2\|u_{xx}(t)\|\,. \tag{6.7}$$

For later use, from the definition of the Ritz projection, we conclude that

$$\langle a(R_h u_t)_x, v_x\rangle = \langle a(u_t)_x, v_x\rangle, \quad \forall v \in S_h,$$

and so, we have the following error estimate:

$$\|u_t(t) - R_h u_t(t)\| \leqslant Ch^2\|u_{txx}(t)\|\,. \tag{6.8}$$

From above, we deduce that $\|\eta^n\| \leqslant Ch^2\|u_{xx}(t_n)\|$ and thus, the remaining task is to estimate θ^n.

From (6.6), the weak formulation in (6.2) ($\langle u_t, v\rangle + \langle a u_x, v_x\rangle = \langle f, v\rangle$), and the definition of the Ritz projection, we get

$$\begin{aligned}\langle U_h^n - U_h^{n-1}, v\rangle &+ \tau\langle a(U_h)_x^n, v_x\rangle \\ &= \tau\langle u_t(t_n), v\rangle + \tau\langle a R_h u_x(t_n), v_x\rangle \\ &= \int_{t_{n-1}}^{t_n}\langle u_t(t_n) - u_t(t), v\rangle\,dt + \langle u(t_n) - u(t_{n-1}), v\rangle + \tau\langle a R_h u_x(t_n), v_x\rangle,\end{aligned}$$

Consequently,

$$\langle \theta^n - \theta^{n-1}, v\rangle + \tau\langle a\theta_x^n, v_x\rangle = \left\langle \eta^n - \eta^{n-1} + \int_{t_{n-1}}^{t_n} u_t(t_n) - u_t(t)\,dt, v\right\rangle$$

Now, adopting the proof of the stability result and using (6.8) yield

$$\begin{aligned}
\|\theta^n\| &\leqslant \|\theta^0\| + \sum_{j=1}^{n}\Big(\|\eta^j - \eta^{j-1}\| + \int_{t_{n-1}}^{t_n}\|u_t(t_j) - u_t(t)\|\,dt\Big)\\
&\leqslant \|\theta^0\| + \sum_{j=1}^{n}\Big(\int_{t_{j-1}}^{t_j}\|\eta_t\|\,dt + \int_{t_{j-1}}^{t_j}\int_t^{t_j}\|u_{tt}(s)\|\,ds\,dt\Big)\\
&\leqslant \|\theta^0\| + C\sum_{j=1}^{n}\Big(h^2\int_{t_{j-1}}^{t_j}\|u_{txx}\|\,dt + \tau\int_{t_{j-1}}^{t_j}\|u_{tt}(s)\|\,ds\Big)\\
&= \|\theta^0\| + C\Big(h^2\int_0^{t_n}\|u_{txx}\|\,dt + \tau\int_0^{t_n}\|u_{tt}\|\,dt\Big).
\end{aligned}$$

Using this and (6.7), we observe that

$$\|U_h^n - u(t_n)\| \leqslant \|\theta^n\| + \|\eta^n\| \leqslant \|\theta^0\| + C\Big(h^2\int_0^{t_n}\|u_{txx}\|\,dt + \tau\int_0^{t_n}\|u_{tt}\|\,dt\Big) + Ch^2\|u_{xx}(t_n)\|,$$

and therefore, our backward Euler finite element scheme is first order accurate in time and second order accurate in space, provided that $\|\theta^0\| = O(h^2)$.

6.4 Approximations of the initial data

For the approximation of the initial data u_0, denoted by u_{0h}, we can use the piecewise linear interpolation. That is, $u_{0h} \in V_h$ interpolate u at the nodes x_i for $0 \leqslant i \leqslant P$. Alternatively, u_{0h} can be chosen to be the Ritz projection of the initial data u_0, that is, $u_{0h} = R_h u_0$. For non-smooth initial data, u_{0h} can be defined via the $L_2(0, L)$ projection. That is, $u_{0h} = P_h u_0$, where $P_h : L_2(0, L) \to V_h$ defined by: for $w \in L_2(0, L)$,

$$\langle P_h w, v\rangle = \langle w, v\rangle, \quad \text{for} \quad v \in V_h.$$

The error $\|u_0 - u_{0h}\| \leqslant Ch^2\|u_{0xx}\|$ can be guaranteed by using any of the above approaches.

6.5 Exercises

6.1. Let $f(\xi) = a_1 + a_2\xi + a_3\xi^2 + \cdots + a_{r+1}\xi^r$ be any real polynomial of degree at most r, and form the associated (column) vector of coefficients $\mathbf{a} = [a_i]_{i=1}^{r+1}$.

(i) Find the matrix $\mathbf{B}$ such that $\int_0^1 |f(\xi)|^2\,d\xi = \mathbf{a}^T\mathbf{B}\mathbf{a}$.

(ii) Prove that $\mathbf{B}$ is symmetric and (strictly) positive-definite.

(iii) Find the matrix $\mathbf{M}$ such that $\int_0^1 |f_x(\xi)|^2\,d\xi = \mathbf{a}^T\mathbf{M}\mathbf{a}$.

(iv) Prove that $\mathbf{M}$ is symmetric and positive-semidefinite.

6.2. Consider the following diffusion model

$$\begin{aligned} u_t - (a(x)u_x)_x &= f(x,t) \quad \text{for } 0 < x < L \text{ and } 0 < t < T, \\ u(0,t) = u(L,t) &= 0 \qquad \text{for } 0 < t < T, \\ u(x,0) &= u_0(x) \quad \text{for } 0 < x < L, \end{aligned} \tag{6.9}$$

where $0 < a_{\min} \leqslant a(x) \leqslant a_{\max} < \infty$ for $0 \leqslant x \leqslant L$.

(i) Define the weak formulation of (6.9).

Assume that $f = 0$ and $a(x) = 1$ in (6.9).

(ii) Let U_h^n be the forward Euler (in time) finite element (in space) approximate solution. Define U_h^n as explained in lectures with $h = \frac{L}{P}$ and $\tau = \frac{T}{N}$.

(iii) Write the numerical scheme in a matrix form.

(iv) Show the following stability property: $\|U_h^n\| \leqslant \sqrt{2}\,\|U_h^0\|$ for $1 \leqslant n \leqslant N$, provided that $\tau \leqslant \frac{h^2}{m^2}$, where m is a positive constant occurred in the inverse inequality $\|(U_h)_x^n\| \leqslant m\,h^{-1}\|U_h^n\|$ (for $1 \leqslant n \leqslant N$).

(v) Show the existence and uniqueness of the numerical solution U_h^n for $1 \leqslant n \leqslant N$.

6.3. Consider the model problem in Exercise 6.2 with $f = 0$ and $a = 1$.

(i) For $1 \leqslant n \leqslant N$, let $U_h^n \in S_h$ be the implicit Euler (in time) finite element (in space) solution of (6.9) . Define U_h^n as explained in lectures with $h = \frac{L}{P}$ and $\tau = \frac{T}{N}$.

(ii) Write the numerical scheme in a matrix form.

(iii) Let $u_0(x) = x(1-x)$, $L = 1$, $T = 1$. Choose $P = N = 40$, simulate the approximate solution.

(iv) Use the data given in (iii) and fix $P = 400$. Compute the errors when $N = 10, 20, 40, 80$ and 160, and illustrate numerically that the proposed numerical scheme is first order accurate in time. Noting that the exact solution in this case is

$$u(x,t) = \frac{8}{\pi^3} \sum_{m=1}^{\infty} \frac{\sin((2m-1)\pi x)}{(2m-1)^3} \exp(-(2m-1)^2\pi^2\, t)\,.$$

6.4. Consider the model problem in Exercise 6.2.

(i) Let $U_h^n \in S_h$ be the Crank-Nicolson (in time) finite element (in space) solution of (6.9). Define U_h^n as explained in lectures with $h = \frac{L}{P}$ and $\tau = \frac{T}{N}$.

(ii) Write the numerical scheme in a matrix form.

(iii) Use the stability estimate $\|U_h^n\| \leqslant \|U_h^0\| + 2\tau \sum_{j=0}^{n} \|f^{j-1/2}\|$ for $1 \leqslant n \leqslant N$ to show the existence and uniqueness of the numerical solution.

(iv) Let $u_0(x) = \sin x$, $f(x,t) = 0$, $a = 1$, $L = \pi$, and $T = 1$. Surf the numerical solution when $P = N = 40$.

Chapter 7

Finite elements for 2D stationary models

In Chapter 5, we investigated the finite element numerical solution for a 1D steady state model with mixed boundary conditions. We focus in this chapter on the 2D extension. The 2D equivalent version of the linear, second-order differential operator (5.14) has the form

$$\begin{aligned}\mathcal{L}u &= -\nabla\cdot\big(a\nabla u\big) + cu \\ &= -\frac{\partial}{\partial x_1}\left(a\,\frac{\partial u}{\partial x_1}\right) - \frac{\partial}{\partial x_2}\left(a\,\frac{\partial u}{\partial x_2}\right) + cu,\end{aligned} \tag{7.1}$$

where the coefficients a and c must be smooth functions of x_1 and x_2, and there is a constant $a_{\min}$ such that

$$a(x_1,x_2) \geqslant a_{\min} > 0 \quad \text{for } (x_1,x_2)\in\Omega\,.$$

Here, as in Chapter 3, Ω is a bounded open subset of $\mathbb{R}^2$ with a piecewise smooth boundary $\Gamma = \partial\Omega$, but we will now suppose that

$$\Gamma = \Gamma_D \cup \Gamma_N,$$

where Γ_D and Γ_N are non-overlapping, relatively closed subsets of $\partial\Omega$ consisting of finitely many smooth curves. (Thus, the intersection $\Gamma_D \cap \Gamma_N$ consists of finitely many *collision points*.)

Consider the following steady-steady state *mixed model*:

$$\begin{aligned}\mathcal{L}u &= f \quad \text{in } \Omega,\\ u &= 0 \quad \text{on } \Gamma_D,\\ \frac{\partial u}{\partial n} &= 0 \quad \text{on } \Gamma_N.\end{aligned} \tag{7.2}$$

Here, $\partial u/\partial n$ is the derivative of u in the direction of the *outward unit normal* n for Ω, that is,

$$\frac{\partial u}{\partial n}(x_1,x_2) = n(x_1,x_2)\cdot\nabla u(x_1,x_2) \quad \text{for } (x_1,x_2)\in\Gamma.$$

The proposed approach can be extended to the case of non-homogeneous mixed boundary conditions.

We refer to Γ_D and Γ_N as the *Dirichlet* and *Neumann* parts of the boundary, respectively, since we specify a Dirichlet boundary condition $u = 0$ on Γ_D and a Neumann boundary condition $\partial u/\partial n = 0$ on Γ_N.

In the special case of a *pure Dirichlet problem*, the Neumann part of the boundary is empty and so u is specified on the whole of Γ (as in Chapter 3). In the opposite case of a *pure Neumann problem*, the Dirichlet part of the boundary is empty and so $\partial u/\partial n$ is specified on the whole of Γ.

Our aim is to use the finite element method to compute numerical solutions of the *mixed model* problem in (7.2). As in the 1D case, the first step towards this aim summarizes in introducing the weak formulation of (7.2). Second, define the finite element space including the trial set (where we sought the approximate solution) and the test space. Then, define the finite element solution.

7.1 Weak formulations

Throughout this chapter, $\overline{\Omega}$ denotes the closure of Ω, that is, $\overline{\Omega} = \Omega \cup \Gamma$. Recall the *divergence theorem* from vector calculus.

Theorem 7.1. *If the vector field $\mathbf{F} : \overline{\Omega} \to \mathbb{R}^2$ is continuously differentiable, then*

$$\int_\Omega \nabla \cdot \mathbf{F} = \int_\Gamma \mathbf{F} \cdot \mathbf{n}.$$

Written out more explicitly, if

$$\mathbf{F}(x_1, x_2) = P(x_1, x_2)\,\mathbf{i} + Q(x_1, x_2)\,\mathbf{j} \quad \text{and} \quad \mathbf{n} = n_{x_1}\,\mathbf{i} + n_{x_2}\,\mathbf{j},$$

then the divergence theorem says that

$$\iint_\Omega \left(\frac{\partial P}{\partial x_1} + \frac{\partial Q}{\partial x_2} \right) dx_1\, dx_2 = \int_\Gamma \left(P\, n_{x_1} + Q\, n_{x_2} \right) ds,$$

where ds is the element of arc length along Γ. We also recall the following vector field identity which is a 2D version of (5.2).

Theorem 7.2 (First Green Identity)**.** *If $u : \overline{\Omega} \to \mathbb{R}$ is twice continuously differentiable, and if $v : \overline{\Omega} \to \mathbb{R}$ is continuously differentiable, then*

$$\int_\Omega \nabla \cdot \left(a \nabla u \right) v = \int_\Gamma a \frac{\partial u}{\partial n} v - \int_\Omega a \nabla u \cdot \nabla v.$$

Proof. Since

$$\nabla \cdot (v a \nabla u) = (\nabla v) \cdot (a \nabla u) + v \nabla \cdot (a \nabla u),$$

$$\nabla \cdot (a \nabla u) v = v \nabla \cdot (a \nabla u) = \nabla \cdot (v a \nabla u) - (\nabla v) \cdot (a \nabla u) = \nabla \cdot (v a \nabla u) - a \nabla u \cdot \nabla v.$$

Applying Theorem 7.1 with $\mathbf{F} = v a \nabla u$, it follows that

$$\int_\Omega \nabla \cdot (a \nabla u) v = \int_\Gamma (v a \nabla u) \cdot \mathbf{n} - \int_\Omega a \nabla u \cdot \nabla v,$$

which gives the desired identity because $(\nabla u) \cdot \mathbf{n} = \partial u / \partial n$. □

We are ready now to introduce the weak formulation of (7.2). Following the 1D case process, we take the $L_2(\Omega)$-inner product of (7.2) with a test function v,

$$\langle -\nabla \cdot (a \nabla u) + c u, v \rangle = \langle f, v \rangle .$$

An application of the first Green identity yields

$$\langle a \nabla u, \nabla v \rangle + \langle c u, v \rangle = \int_\Gamma a \frac{\partial u}{\partial n} v + \langle f, v \rangle .$$

However,

$$\int_\Gamma a \frac{\partial u}{\partial n} v = \int_{\Gamma_D} a \frac{\partial u}{\partial n} v + \int_{\Gamma_N} a \frac{\partial u}{\partial n} v = \int_{\Gamma_D} a \frac{\partial u}{\partial n} v,$$

and so, we obtain the weak formulation of (7.2):

$$\langle a \nabla u, \nabla v \rangle + \langle c u, v \rangle = \langle f, v \rangle \quad \text{provided } v = 0 \text{ on } \Gamma_D. \tag{7.3}$$

Compare this property with its 1D equivalent (5.17).

Figure 7.1: A regular triangulation.

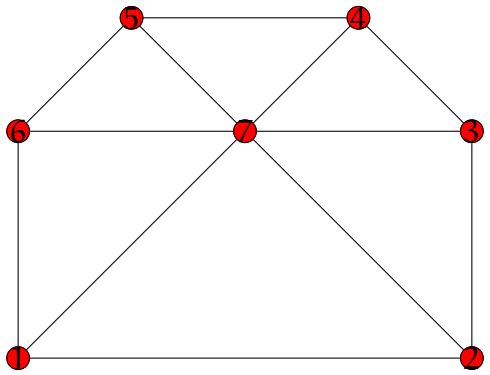
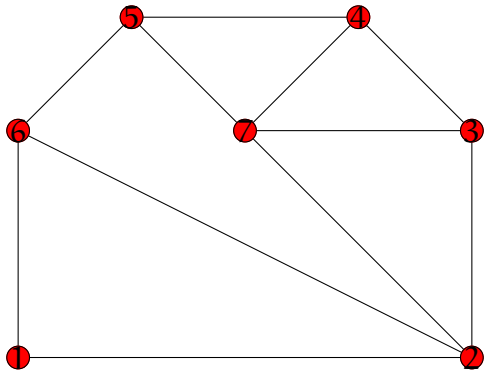


Figure 7.2: A triangulation that fails to be regular.



7.2 Meshes and finite element spaces

Assume now that Ω is a polygon. It follows by induction on the number of vertices that there exists a *triangulation* of Ω, that is, a finite set $\mathcal{T}_h$ of *non-overlapping* (disjoint) closed triangles whose union is $\overline{\Omega}$. A triangulation $\mathcal{T}_h$ is *regular* if the following two conditions are satisfied:

1. No triangle in $\mathcal{T}_h$ is *degenerate*, that is, no $K \in \mathcal{T}_h$ has collinear vertices.

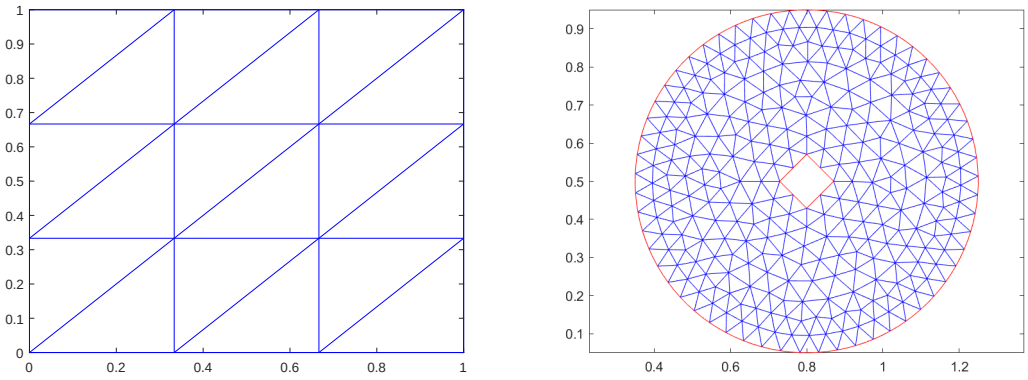
2. The intersection $K_1 \cap K_2$ of any two distinct triangles K_1, $K_2 \in \mathcal{T}_h$ is either empty, a common edge or a common vertex.

For example, Figure 7.1 shows a regular triangulation with 7 vertices, or *nodes* (numbered in red), 6 triangles, or *elements*, and 6-boundary edges. However, the triangulation in Figure 7.2 is not regular. Vertex 7 in Figure 7.2 is said to be a *hanging node*.

A Matlab script for generating the triangular mesh in Figure 7.1.

```
P = [-2  2  2  1  -1  -2  0;
       -2 -2  0  2  2  0  0]; % the x- and y-coordinates of all nodes
T=[1  1  2  3  4  5;7  2  3  4  5  6;6  7  7  7  7  7]; % connectivity matrix
TR=triangulation(T', P');
triplot(TR)
```

Figure 7.3: Some other examples of regular triangulations.



A simple data structure to store a triangulation consists of two arrays, $\mathbf{P}$ and $\mathbf{T}$. The first stores the coordinates of the jth node in its jth column, and the second stores the node numbers of the kth triangle in its kth column. Thus, the triangulation in Figure 7.1 may be described by

$$\mathbf{P} = \begin{bmatrix} -2 & 2 & 2 & 1 & -1 & -2 & 0 \\ -2 & -2 & 0 & 2 & 2 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} 1 & 1 & 2 & 3 & 4 & 5 \\ 7 & 2 & 3 & 4 & 5 & 6 \\ 6 & 7 & 7 & 7 & 7 & 7 \end{bmatrix}.$$

We deal with boundary conditions matrix via the matrix $\mathbf{E}$ where the jth column stores the node numbers of the jth boundary edge. In Figure 7.1,

$$\mathbf{E} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 4 & 5 & 6 & 1 \end{bmatrix}.$$

For a simple example of a uniform triangular mesh, see Figure 7.1.

In general, to simplify some geometric computations, it is best to ensure that the node numbering within each triangle proceeds counterclockwise, and that the edge node numbering follows the induced orientation of $\partial\Omega$. (Even with this restriction, there are 3 possibilities for each column of $\mathbf{T}$.)

Let h be the maximal length of the sides of the triangulation $\mathcal{T}_h$, that is, $h = \max_{K\in\mathcal{T}_h} \operatorname{diam}(K)$. We shall assume sometimes that the triangulations are quasi-uniform in the sense that the triangles of $\mathcal{T}_h$ are of essentially the same size, which we express by demanding that the area of any triangle K in $\mathcal{T}_h$ to be bounded below by $m h^2$ for some constant $m > 0$, independent of h.

Let V_h denote the vector space consisting of those functions $v : \overline{\Omega} \to \mathbb{R}$ that are continuous and piecewise-linear with respect to $\mathcal{T}_h$. Thus, if $v \in V_h$ then for each $K \in \mathcal{T}_h$ there are coefficients c_0^K, c_1^K and c_2^K such that

$$v(x_1, x_2) = c_0^K + c_1^K x_1 + c_2^K x_2 \quad \text{for } (x_1, x_2) \in K. \tag{7.4}$$

Suppose that n_1, n_2, ..., n_N is an enumeration of the nodes of $\mathcal{T}_h$. For $1 \leqslant j \leqslant N$, we define $\chi_j \in V_h$ by requiring

$$\chi_k(n_j) = \delta_{jk} \quad \text{for } j, k \in \{1, 2, \ldots, N\}. \tag{7.5}$$

Figure 7.4 shows an example of such a "tent function". If $v \in V_h$, then

$$v(x_1, x_2) = \sum_{k=1}^{N} v(n_k)\chi_k(x_1, x_2) \quad \text{for } (x_1, x_2) \in \overline{\Omega},$$

Figure 7.4: A piecewise-linear "tent function", equal to 1 at one node, and 0 at all other nodes.

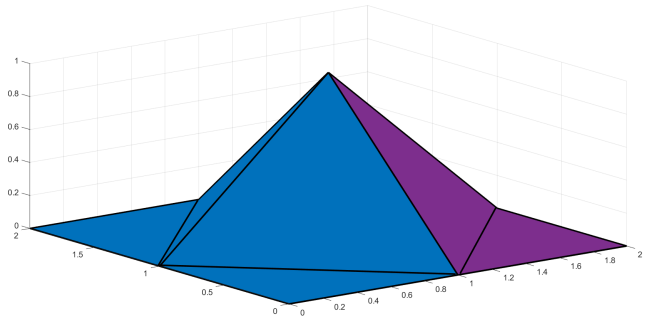
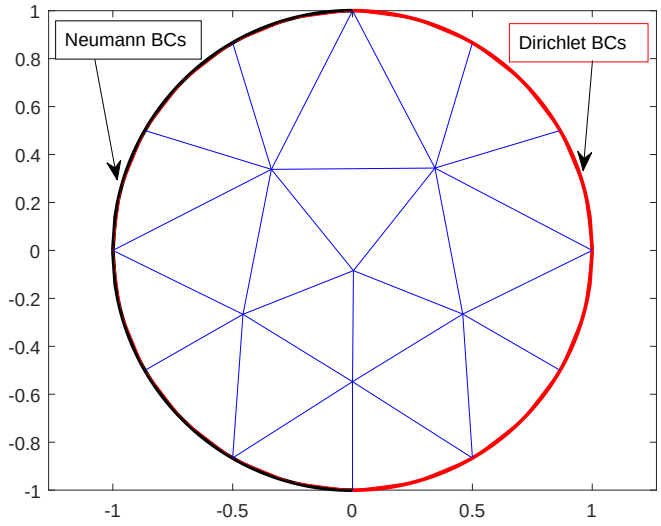


Figure 7.5: A triangular mesh with six interior nodes and twelve boundary nodes.



and we call $\{\chi_1, \chi_2, \ldots, \chi_N\}$ the *nodal basis* for V_h (Exercise 7.5).

Suppose that a regular triangulation $\mathcal{T}_h$ is *aligned* with the decomposition $\Gamma = \Gamma_D \cup \Gamma_N$ of the boundary of Ω. This assumption means that Γ_D is a union of edges of triangles in $\mathcal{T}_h$ (in which case, the same must be true of Γ_N). The vertices lying on the Dirichlet boundary Γ_D are called the *fixed nodes*, because the values of the solution u are fixed at these points. The remaining vertices are called the *free nodes*; these belong to $\Omega \cup \Gamma_N$, but note that the collision points, where Γ_D and Γ_N meet, are among the fixed nodes.

Suppose that there are M free nodes and R fixed nodes. It is convenient to number the nodes so that free nodes come first, followed by the fixed nodes. That is, n_1, n_2, ..., n_M are free, and n_{M+1}, n_{M+2}, ..., n_{M+R} are fixed.

We define the trial space

$$S_h = \{ v \in V_h : v = 0 \text{ on } \Gamma_D \}$$

and the test space $T_h = S_h$ because the Dirichlet boundary conditions are homogeneous. Before proceeding, it is worth to mention that the dimension of V_h, $\dim V_h$, is equal to the total number of nodes, that is, $M + R$. However, $\dim S_h = M$, the number of free nodes. As an example, in the triangular mesh in Figure 7.5, $\dim V_h = 18$ and $\dim S_h = 11$.

7.3 Finite element method

Recalling (7.3), the finite element solution $u_h \in S_h$ is then defined by requiring that

$$\langle a\nabla u_h, \nabla v\rangle + \langle cu_h, v\rangle = \langle f, v\rangle \quad \text{for all } v \in S_h. \tag{7.6}$$

We expand the finite element solution in the nodal basis, to obtain the representation

$$u_h(x_1, x_2) = \sum_{j=1}^{M} \alpha_j \chi_j(x_1, x_2), \quad \text{where} \quad \alpha_j = u_h(n_j). \tag{7.7}$$

Since $\{\chi_1, \chi_2, \ldots, \chi_M\}$ is a nodal basis for the trial space S_h, the requirement (7.6) is equivalent to

$$\langle a\nabla u_h, \nabla \chi_i\rangle + \langle cu_h, \chi_i\rangle = \langle f, \chi_i\rangle \quad \text{for } 1 \leqslant i \leqslant M, \tag{7.8}$$

and so, after inserting the representation (7.7), we obtain an $M \times M$ linear system

$$\sum_{j=1}^{M} \big(a_{i,j} + c_{jk}\big)\alpha_j = f_i \quad \text{for } 1 \leqslant i \leqslant M,$$

where

$$a_{i,j} = \langle a\nabla\chi_j, \nabla\chi_i\rangle, \qquad c_{i,j} = \langle c\chi_j, \chi_i\rangle, \qquad f_i = \langle f, \chi_i\rangle\,.$$

Introduce the column vector $\alpha = [\alpha_j]_{j=1}^{M}$, and then,

$$\big(\mathbf{A} + \mathbf{C}\big)\alpha = \mathbf{f}, \tag{7.9}$$

where $\mathbf{A} = [a_{i,j}]_{i,j=1}^{M}$ is the stiffness matrix, $\mathbf{C} = [c_{i,j}]_{i,j=1}^{M}$ is the mass matrix, $\mathbf{f} = [f_i]_{i=1}^{M}$ is the load vector.

To solve the above system, a special care has to be taken for nodes which belong to triangles with at least one boundary point. This leads to a complicated and slow assembly process. Owing to this, an alternative element-by-element implementation process will be discussed Section 7.4.

Stability. To show the stability of the finite element solution u_h in (7.6), we choose $v = u_h$ and observe that

$$\langle a\nabla u_h, \nabla u_h\rangle + \langle cu_h, u_h\rangle = \langle f, u_h\rangle.$$

Recalling that, $a \geqslant a_{\min} > 0$ and $c \geqslant 0$, and so, by the Cauchy-Schwarz inequality,

$$a_{\min}\|\nabla u_h\|^2 \leqslant \|\sqrt{a}\nabla u_h\|^2 \leqslant \|f\|\,\|u_h\|\,.$$

By the Poincare inequality, $\|u_h\| \leqslant C\|\nabla u_h\|$ (provided that the length of $\Gamma_D > 0$), and so

$$\|\nabla u_h\|^2 \leqslant C\|f\|\,\|\nabla u_h\|\,.$$

Simplifying and using again $\|u_h\| \leqslant C\|\nabla u_h\|$, then we obtain the following stability property

$$\|u_h\| \leqslant C\|\nabla u_h\| \leqslant C\|f\|\,. \tag{7.10}$$

Existence and uniqueness. As demonstrated in (7.9), the finite element scheme in (7.6) amounts into a finite square linear system. Because of this, the existence of the finite element solution u_h follows from its uniqueness. To show the latter, assume that we have two solutions, say u_h and w_h. Then, from (7.6),

$$\langle a\nabla(u_h - w_h), \nabla v\rangle + \langle c(u_h - w_h,)v\rangle = 0 \quad \text{for all } v \in S_h.$$

Applying the stability property in (8.8) gives

$$\|u_h - w_h\| \leqslant 0, \tag{7.11}$$

and therefore, $u_h - w_u = 0$. This completes the proof of uniqueness, and consequently, of the existence of the finite element solution u_h.

Errors. For the error estimate, we subtract the finite element scheme in (7.6) from the weak formulation in (7.3) to get

$$\langle a\nabla(u - u_h), \nabla v\rangle + \langle c(u - u_h), v\rangle = 0 \quad \text{for all } v \in S_h.$$

Decomposing as

$$\langle a\nabla(\hat{u} - u_h), \nabla v\rangle + \langle c(\hat{u} - u_h), v\rangle = \langle a\nabla(\hat{u} - u), \nabla v\rangle + \langle c(\hat{u} - u), v\rangle \quad \text{for all } v \in S_h,$$

where $\hat{u} \in S_h$ is the piecewise linear function interpolates u as the nodes. For a sufficiently regular function u, we have the following interpolation error (Bramble-Hilbert lemma can be used in the proof):

$$\|u - \hat{u}\| + h\|\nabla(u - \hat{u})\| \leqslant Ch^2\,. \tag{7.12}$$

Now, choosing $v = u_h - \hat{u}$, then using $c \geqslant 0$ and $a \geqslant a_{\min} > 0$, the Cauchy-Schwarz inequality, and the interpolation error in (7.12),

$$\begin{aligned}
a_{\min}\|\nabla(\hat{u} - u_h)\|^2 &\leqslant \|\sqrt{a}\nabla(\hat{u} - u_h)\|^2 + \|\sqrt{c}(\hat{u} - u_h)\|^2 \\
&= \langle a\nabla(\hat{u} - u_h), \nabla(\hat{u} - u_h)\rangle + \langle c(\hat{u} - u_h), \hat{u} - u_h\rangle \\
&= \langle a\nabla(\hat{u} - u), \nabla(\hat{u} - u_h)\rangle + \langle c(\hat{u} - u), \hat{u} - u_h\rangle \\
&\leqslant C\|\nabla(\hat{u} - u)\|\,\|\nabla(\hat{u} - u_h)\| + C\|\hat{u} - u\|\,\|\hat{u} - u_h\| \\
&\leqslant C\Big(\|\nabla(\hat{u} - u)\| + \|\hat{u} - u\|\Big)\|\nabla(\hat{u} - u_h)\| \\
&\leqslant Ch\|\nabla(\hat{u} - u_h)\|\,.
\end{aligned}$$

After simplifying,

$$\|\nabla(\hat{u} - u_h)\| \leqslant Ch,$$

and therefore,

$$\|\nabla(u - u_h)\| \leqslant \|\nabla(u - \hat{u})\| + \|\nabla(\hat{u} - u_h)\| \leqslant Ch + Ch \leqslant Ch\,.$$

The proof of the L_2-norm error estimate is more delicate. Aubin-Nitsche duality argument is needed here. With some efforts, the 1D case proof in Theorem 5.5 can be extended to the 2D case and show that

$$\|u - u_h\| \leqslant Ch^2\,.$$

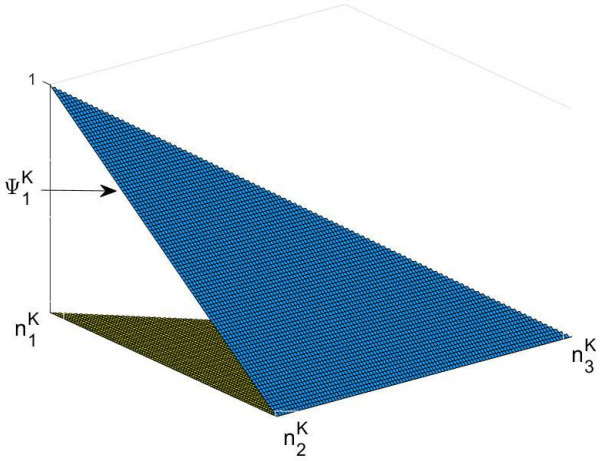
7.4 Element matrices

7.4.1 Local matrices

Let n_1^K, n_2^K, n_3^K denote the vertices of the triangle $K \in \mathcal{T}_h$ that are labeled in the anti-clockwise direction. Let ψ_1^K, ψ_2^K, ψ_3^K denote the unique linear functions satisfying (see the example in Figure 7.6)

$$\psi_p^K(n_q^K) = \delta_{p,q} \quad \text{for } p,\, q \in \{1, 2, 3\}, \tag{7.13}$$

Figure 7.6: An example of the linear function ψ_1^K.



where $\delta_{p,q}$ is the Kronecker delta, that is,

$$\delta_{p,q} = \begin{cases} 1, & \text{if } p = q, \\ 0, & \text{if } p \neq q. \end{cases}$$

The property (7.13) implies that if $v \in V_h$ then

$$v(x_1, x_2) = \sum_{q=1}^{3} v(n_q^K)\psi_q^K(x_1, x_2) \quad \text{for } (x_1, x_2) \in K,$$

showing that v is uniquely determined by its values at the nodes of $\mathcal{T}_h$.

The main task of this section is to assemble the $M \times M$ global stiffness matrix $\mathbf{A}$ from the 3×3 element stiffness matrices

$$\boldsymbol{A}^K = \left[a_{i,j}^K\right]_{i,j=1}^3 \quad \text{where} \quad a_{i,j}^K = \int_K a\nabla\psi_i^K \cdot \nabla\psi_j^K,$$

the $M \times M$ global mass matrix $\mathbf{C}$ from the 3×3 element mass matrices,

$$\mathbf{C}^K = \left[c_{i,j}^K\right]_{i,j=1}^3 \quad \text{where} \quad c_{i,j}^K = \int_K c\psi_i^K\psi_j^K,$$

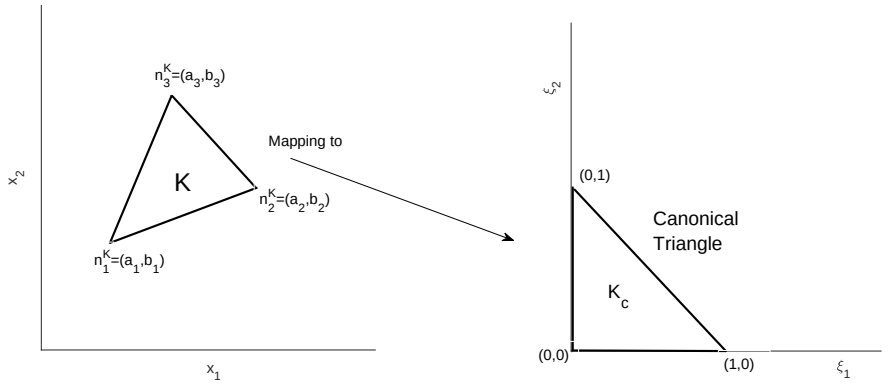
and also to assemble the load vector $\mathbf{f} = [f_i]_{i=1}^M$ from the 3×1 element load vector

$$\mathbf{f}^K = [f_i^K]_{i=1}^3 \quad \text{where} \quad f_i^K = \int_K f\psi_i^K\,.$$

To achieve the above tasks, we consider a so-called local (ξ_1, ξ_2) coordinate system and the canonical (reference) triangle K_c illustrated in Figure 7.7. The coordinate of any point $\mathbf{x} = (x_1, x_2) \in K$ can be written as a combination of the coordinates of the three vertices: with $\phi_1(\xi_1, \xi_2) = 1 - \xi_1 - \xi_2$, $\phi_2(\xi_1, \xi_2) = \xi_1$, and $\phi_3(\xi_1, \xi_2) = \xi_2$,

$$\begin{aligned} \mathbf{x} &= \phi_1(\xi_1, \xi_2)\, n_1^K + \phi_2(\xi_1, \xi_2)\, n_2^K + \phi_3(\xi_1, \xi_2)\, n_3^K \\ &= n_1^K + \xi_1\,(n_2^K - n_1^K) + \xi_2\,(n_3^K - n_1^K)\,. \end{aligned}$$

Figure 7.7: Mapping K into a canonical triangle



The set $\{\phi_1, \phi_2, \phi_3\}$ is called the nodal basis (or local basis) for the set of linear polynomials in terms of the local coordinates. Noting that, $\phi_1 + \phi_2 + \phi_3 = 1$.

The affine transformation (transformation that preserves collinearity) $K_c \to K$;

$$(\xi_1, \xi_2) \mapsto \mathbf{x}(\xi_1, \xi_2) = n_1^K + \xi_1 (n_2^K - n_1^K) + \xi_2 (n_3^K - n_1^K) \tag{7.14}$$

is one-to-one (because the homogeneous linear system $\xi_1 (n_2^K - n_1^K) + \xi_2 (n_3^K - n_1^K) = 0$ has a unique solution) and onto (due to the relation in (7.16)). The Jacobi matrix of this transformation is

$$J = \frac{\partial(x_1, x_2)}{\partial(\xi_1, \xi_2)} = \begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_1} \\ \frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_2} \end{bmatrix} = \begin{bmatrix} (a_2 - a_1) & (b_2 - b_1) \\ (a_3 - a_1) & (b_3 - b_1) \end{bmatrix},$$

and so,

$$|J| = \left| \begin{bmatrix} a_1 & b_1 & 1 \\ a_2 & b_2 & 1 \\ a_3 & b_3 & 1 \end{bmatrix} \right| = 2\,\mathrm{area}(K)\,.$$

It follows that

$$\int_K g(\mathbf{x})\, dx_1 dx_2 = \int_{K_c} g(\mathbf{x}(\xi_1, \xi_2))\, |J|\, d\xi_1\, d\xi_2 = 2\,\mathrm{area}(K) \int_0^1 \int_0^{1-\xi_2} g(\mathbf{x}(\xi_1, \xi_2))\, d\xi_1\, d\xi_2. \tag{7.15}$$

In the next theorem we find ξ_i as a linear functions of x_1 and x_2 for $i = 1, 2$.

Theorem 7.3. *The algebraic linear system in* (7.14) *has the following unique solution:*

$$\xi_1 = \frac{1}{2\,\mathrm{area}(K)} \Big((x_1 - a_1)(b_3 - b_1) - (x_2 - b_1)(a_3 - a_1) \Big),$$
$$\xi_2 = \frac{1}{2\,\mathrm{area}(K)} \Big((x_2 - b_1)(a_2 - a_1) - (x_1 - a_1)(b_2 - b_1) \Big)\,.$$

Proof. Rewrite (7.14) explicitly as:

$$\begin{cases} x_1 &= (1 - \xi_1 - \xi_2) a_1 + \xi_1 a_2 + \xi_2 a_3, \\ x_2 &= (1 - \xi_1 - \xi_2) b_1 + \xi_1 b_2 + \xi_2 b_3\,. \end{cases}$$

Rearranging the terms in the above linear system as:

$$\begin{cases} (a_2 - a_1)\xi_1 + (a_3 - a_1)\xi_2 &= x_1 - a_1, \\ (b_2 - b_1)\xi_1 + (b_3 - b_1)\xi_2 &= x_2 - b_1\,. \end{cases}$$

Applying Cramer's rule yields

$$\xi_i = \frac{|E_i|}{|J|}, \quad \text{for} \quad i = 1, 2,$$

where

$$E_1 = \begin{bmatrix} (x_1 - a_1) & (a_3 - a_1) \\ (x_2 - b_1) & (b_3 - b_1) \end{bmatrix}, \text{ and } E_2 = \begin{bmatrix} (a_2 - a_1) & (x_1 - a_1) \\ (b_2 - b_1) & (x_2 - b_1) \end{bmatrix}.$$

Computing the determinants of E_1 and E_2, and using $|J| = 2\,\text{area}(K)$ will complete the proof. □

One can deduce from Theorem 7.3 that

$$\psi_p^K(\mathbf{x}(\xi_1, \xi_2)) = \phi_p(\xi_1, \xi_2) \quad \text{for } 1 \leqslant p \leqslant 3. \tag{7.16}$$

Furthermore, by using Theorem 7.3, we can easily find that

$$\nabla\xi_1 = \frac{1}{2\,\text{area}(K)} \begin{bmatrix} (b_3 - b_1) \\ -(a_3 - a_1) \end{bmatrix}, \quad \nabla\xi_2 = \frac{1}{2\,\text{area}(K)} \begin{bmatrix} -(b_2 - b_1) \\ (a_2 - a_1) \end{bmatrix}.$$

So, for $1 \leqslant i < j \leqslant 3$, and with $p \neq i, j$,

$$\nabla\psi_p^K = \frac{(-1)^p}{2\,\text{area}(K)} \begin{bmatrix} (b_j - b_i) \\ -(a_j - a_i) \end{bmatrix}, \quad \text{for } 1 \leqslant p \leqslant 3\,.$$

To proceed in our derivation, we introduce the following notations: let

$$\begin{aligned} \delta n_1^K &= n_2^K - n_3^K, \\ \delta n_2^K &= n_3^K - n_1^K, \\ \delta n_3^K &= n_1^K - n_2^K. \end{aligned}$$

Based on above, it is clear that

$$\begin{aligned} 4(\text{area}(K))^2 \nabla\psi_1^K \cdot \nabla\psi_1^K &= \begin{bmatrix} (b_3 - b_2) \\ -(a_3 - a_2) \end{bmatrix} \cdot \begin{bmatrix} (b_3 - b_2) \\ -(a_3 - a_2) \end{bmatrix} \\ &= (b_3 - b_2)^2 + (a_3 - a_2)^2 = |n_3^K - n_2^K|^2 = \delta n_1^K \cdot \delta n_1^K, \end{aligned}$$

and

$$\begin{aligned} 4(\text{area}(K))^2 \nabla\psi_1^K \cdot \nabla\psi_2^K &= \begin{bmatrix} (b_3 - b_2) \\ -(a_3 - a_2) \end{bmatrix} \cdot \begin{bmatrix} (b_3 - b_1) \\ -(a_3 - a_1) \end{bmatrix} \\ &= (b_2 - b_3)(b_3 - b_1) - (a_2 - a_3)(a_3 - a_1) \\ &= (n_2^K - n_3^K) \cdot (n_3^K - n_1^K) = \delta n_1^K \cdot \delta n_2^K\,. \end{aligned}$$

Following similar calculations, we find that

$$4(\text{area}(K))^2 \nabla\psi_i^K \cdot \nabla\psi_j^K = \delta n_i^K \cdot \delta n_j^K, \quad \text{for } 1 \leqslant i, j \leqslant 3\,.$$

Therefore, the entries of the element stiffness matrix are

$$a_{i,j}^K = \frac{\delta n_i^K \cdot \delta n_j^K}{4(\text{area}(K))^2} \int_K a = \frac{\delta n_i^K \cdot \delta n_j^K}{2\,\text{area}(K)} \int_0^1 \int_0^{1-\xi_2} a\big(\mathbf{x}(\xi_1, \xi_2)\big)\, d\xi_1\, d\xi_2,$$

In particular, if $a(x) = 1$ then

$$A^K = \frac{1}{4\,\mathrm{area}(K)}\left[\delta n_i^K \cdot \delta n_j^K\right]_{i,j=1}^3 .$$

The formula in the next theorem allows us to compute $\int_K P$ for any polynomial P. It is useful to compute the entries of the 3×3 element mass matrices $c_{i,j}^K$.

Theorem 7.4. *For all non-negative integers* n, r, s,

$$\int_K \xi_1^n \xi_2^r (1-\xi_1-\xi_2)^s = 2\,\mathrm{area}(K)\,\frac{n!\,r!\,s!}{(n+r+s+2)!}.$$

Proof. By (7.15),

$$\begin{aligned}\int_K \xi_1^n \xi_2^r (1-\xi_1-\xi_2)^s &= 2\,\mathrm{area}(K)\int_0^1 \int_0^{1-\xi_2} \xi_1^n \xi_2^r (1-\xi_1-\xi_2)^s \,d\xi_1\,d\xi_2 \\ &= 2\,\mathrm{area}(K)\int_0^1 \xi_2^r \int_0^{1-\xi_2} \xi_1^n (1-\xi_2-\xi_1)^s \,d\xi_1\,d\xi_2.\end{aligned}$$

Applying the formula (follows from integrating by parts m times): for any $\epsilon > 0$,

$$\int_0^\epsilon \frac{\xi^n}{n!}\,\frac{(\epsilon-\xi)^m}{m!}\,d\xi = \frac{\epsilon^{n+m+1}}{(n+m+1)!},$$

so

$$\begin{aligned}&\int_0^1 \xi_2^r \int_0^{1-\xi_2} \xi_1^n (1-\xi_2-\xi_1)^s \,d\xi_1\,d\xi_2 \\ &\qquad = \frac{n!\,s!}{(n+s+1)!}\int_0^1 \xi_2^r\,(1-\xi_2)^{n+s+1}\,d\xi_2 = \frac{n!\,s!\,r!}{(n+s+r+2)!},\end{aligned}$$

giving the desired formula. □

Explicitly,

$$c_{i,j}^K = \int_K c\psi_i^K\psi_j^K = 2\,\mathrm{area}(K)\int_0^1\int_0^{1-\xi_1} c\big(x(\xi_1,\xi_2)\big)\phi_i\phi_j\,d\xi_1\,d\xi_2\,.$$

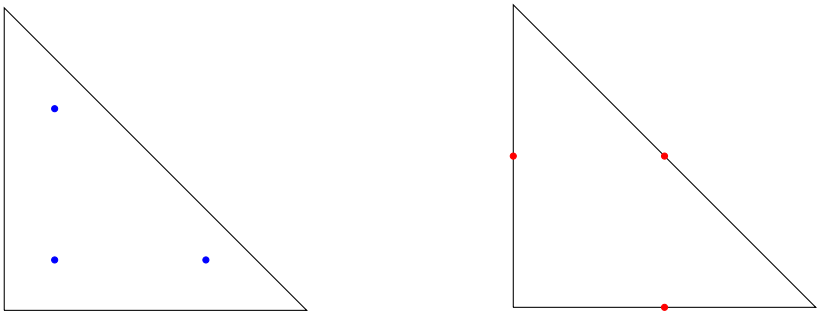
In particular, if $c(x) = 1$, then Theorem 7.4 shows that

$$C^K = \frac{\mathrm{area}(K)}{12}\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}. \tag{7.17}$$

The entries of the 3-dimensional element load vector f^K are

$$f_i^K = \int_K f\psi_i^K = \int_0^1\int_0^{1-\xi_2} f\big(x(\xi_1,\xi_2)\big)\xi_i\,d\xi_1\,d\xi_2 \quad \text{for} \quad i = 1,2,3.$$

Figure 7.8: Third order accurate 3-point quadrature rule on the canonical triangle K_c with vertices $(0,0)$, $(1,0)$ and $(0,1)$. In the left figure, the points are: $(1/6,1/6)$, $(1/6,2/3)$, and $(2/3,1/6)$, and the weights are $w_1 = w_2 = w_3 = 1/6$. In the right figure, the points are: $(1/2,0)$, $(0,1/2)$, and $(1/2,1/2)$, and the weights are $w_1 = w_2 = w_3 = 1/6$. See (7.19) and (7.20) for a related explanation.



7.4.2 Numerical integration

In practice, the occurred local integrals on K_c are often not computed exactly even if analytical expressions for a, c and f are available. Instead they are approximated by numerical integration through a quadrature (cubature) formula of the form: for a given function defined on K_c,

$$\int_{K_c} g(\xi_1,\xi_2) \approx \sum_{i=1}^{m} w_i g(\zeta_i),$$

The numbers w_i are called the weights and the points $\zeta_i = (\zeta_{1,i}, \zeta_{2,i})$ the nodes of the quadrature formula.

The quadrature formula should be chosen in such a way that the error in u_h is of the same order as in the original finite element solution. That is, we require to maintain the second order convergence rate result that was proved. An example of such a quadrature formula is the one point barycentric quadrature rule where the exactness is assumed for all linear functions. That is,

$$\int_{K_c} 1 = w_1, \quad \int_{K_c} \xi_1 = w_1\,\zeta_{1,1}, \quad \int_{K_c} \xi_2 = w_1\,\zeta_{2,1}\,.$$

This leads to $w_1 = \mathrm{area}(K_c) = \frac{1}{2}$, and $\zeta_1 = \left(\frac{1}{3},\frac{1}{3}\right)$, which is the centroid of the triangle K_c. Therefore,

$$\int_{K_c} g(\xi_1,\xi_2) \approx \frac{1}{2} g\left(\frac{1}{3},\frac{1}{3}\right).$$

Such a quadrature rule is second order accurate, and so, it should be sufficient in the case of linear finite elements. For high-order finite element methods, more points are needed in the quadrature formula, see for example the 3-point quadrature rule in Figure 7.8. In general, we should aim to use a quadrature formula that provides the desired accuracy by using minimum number of nodes because the evaluation of the coefficient functions is often expensive.

7.4.3 Global matrices

We enumerate the nodes of the mesh so that the *free nodes precede the fixed nodes*, where the latter are those at which the value of the solution is fixed by a Dirichlet boundary condition,

see the sample mesh in Figure 7.9. Recalling that M is the number of free nodes while R is the number of fixed nodes. Also recalling (7.7), that is, for $x \in \overline{\Omega}$,

$$u_h(x) = \sum_{j=1}^{M+R} \alpha_j \chi_j(x), \quad \text{where} \quad \alpha_j = u_h(n_j). \tag{7.18}$$

Owing to the homogeneous Dirichlet boundary conditions, $\alpha_j = 0$ for $M + 1 \leqslant j \leqslant M + R$. Let $K^\ell \in \mathcal{T}_h$ with vertices n_1^ℓ, n_2^ℓ and n_3^ℓ. So, $n_1^\ell = n_i$, $n_2^\ell = n_j$ and $n_3^\ell = n_p$ for some $1 \leqslant i, j, p \leqslant M + R$. The Boolean $(M + R) \times 3$ matrix $\mathbf{L}^\ell$ corresponding to K^ℓ is defined such that the only unitary elements are the i^{th}, j^{th}, and p^{th} entries of columns one, two and three, respectively. Then, the $(M+R) \times (M+R)$ full stiffness matrix $\mathbf{A}^*$ is defined as a sum over the elements K^ℓ in the triangulation $\mathcal{T}_h$ as follows

$$\mathbf{A}^* = \sum_{\ell=1}^{N} \mathbf{L}^\ell \mathbf{A}^\ell (\mathbf{L}^\ell)',$$

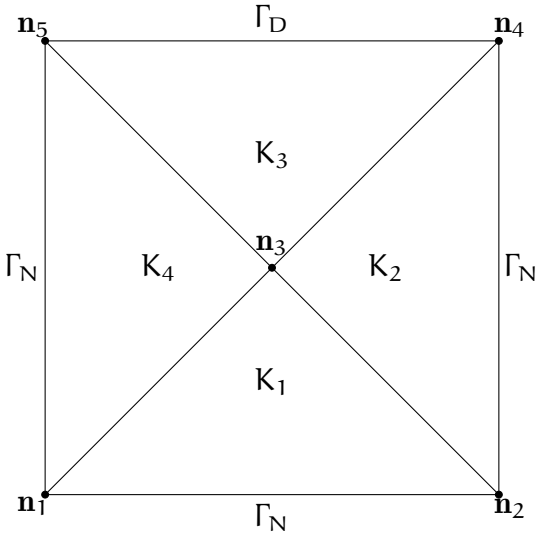
where N is the number of triangles in $\mathcal{T}_h$, $\mathbf{A}^\ell$ is the local stiffness matrix corresponding to triangle K^ℓ, and $(\mathbf{L}^\ell)'$ is the transpose of $\mathbf{L}^\ell$.

In a similar fashion, we build up the $(M + R) \times (M + R)$ full mass matrix $\mathbf{C}^*$ and the $(M + R) \times 1$ full load vector $\mathbf{f}^*$. Whence we compute $\mathbf{A}^*$, $\mathbf{C}^*$ then we dropout the last R columns and R rows from these matrices, to obtain the global matrices $\mathbf{A}$ and $\mathbf{C}$, respectively. However, we remove the last R entries from $\mathbf{f}^*$ to obtain the global load vector $\mathbf{f}$. Then solving the linear system

$$\big(\mathbf{A} + \mathbf{C}\big)\alpha = \mathbf{f},$$

for α. Recalling that the column vector $\alpha = [\alpha_j]_{j=1}^M$, with $\alpha_j = u_h(n_j)$ (that is, the finite element solution at node n_j).

Figure 7.9: A mesh sample.



Example 7.5. *Consider the uniform mesh in Figure 7.9 with area equals to* 1 *of each cell. So*

$$\mathbf{L}^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{L}^2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{L}^3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{L}^4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

If the reaction coefficient $c = 1$*, then recalling* (7.17)*, the local matrices are*

$$\mathbf{C}^\ell = \frac{1}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}, \quad \text{for} \quad \ell = 1, 2, 3, 4.$$

Thus, a tedious calculation leads to the 5×5 *full mass matrix*

$$\mathbf{C}^* = \sum_{\ell=1}^{4} \mathbf{L}^\ell \mathbf{C}^\ell (\mathbf{L}^\ell)' = \frac{1}{12} \begin{bmatrix} 4 & 1 & 2 & 0 & 1 \\ 1 & 4 & 2 & 1 & 0 \\ 2 & 2 & 8 & 2 & 2 \\ 0 & 1 & 2 & 4 & 1 \\ 1 & 0 & 2 & 1 & 4 \end{bmatrix},$$

and consequently, the 3×3 *global mass matrix*

$$\mathbf{C} = \frac{1}{12} \begin{bmatrix} 4 & 1 & 2 \\ 1 & 4 & 2 \\ 2 & 2 & 8 \end{bmatrix}.$$

7.5 Exercises

7.1. Derive the weak formulation of the boundary-value problem (7.2) if we replace the Neumann boundary condition with the more general Robin boundary condition

$$a\frac{\partial u}{\partial n} + bu = g_N \quad \text{on } \Gamma_N.$$

7.2. Derive the weak formulation for the elliptic eigenproblem

$$\begin{aligned} \mathcal{L}\phi &= \lambda b\phi \quad \text{in } \Omega, \\ \phi &= 0 \qquad \text{on } \Gamma_D, \\ a\frac{\partial \phi}{\partial n} &= 0 \qquad \text{on } \Gamma_N, \end{aligned}$$

where, as usual, $\mathcal{L}\phi = -\nabla\cdot(a\nabla\phi) + c\phi$.

7.3. Consider the triangulation $\mathcal{K}$ determined by the nodal coordinate matrix

$$\mathbf{N} = \begin{bmatrix} 1 & 0 & 1 & 0 & -1 & 0 & -1 \\ 1 & 1 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}$$

and the triangle connectivity matrix

$$\mathbf{T}^{\mathcal{K}} = \begin{bmatrix} 1 & 4 & 4 & 3 & 4 & 5 \\ 2 & 3 & 2 & 4 & 5 & 7 \\ 3 & 2 & 5 & 6 & 6 & 6 \end{bmatrix}.$$

(i) Draw $\mathcal{K}$, numbering the nodes and triangles.

(ii) Enumerate the outer edges, given that the edge connectivity matrix is

$$\mathbf{T}^{\mathcal{E}} = \begin{bmatrix} 6 & 3 & 1 & 2 & 5 & 7 \\ 3 & 1 & 2 & 5 & 7 & 6 \end{bmatrix}.$$

(iii) Determine Γ_D and Γ_N assuming $M^{\text{free}} = 4$.

7.4. Suppose that $\Omega = \Omega_1 \cup \Gamma_i \cup \Omega_2$ where the *interface* Γ_i is a piecewise smooth curve, and that

$$a(x,y) = \begin{cases} a_1(x,y) & \text{for } (x,y) \in \Omega_1, \\ a_2(x,y) & \text{for } (x,y) \in \Omega_2. \end{cases}$$

Define corresponding partial differential operators $\mathcal{L}_k u = -\nabla\cdot(a_k\nabla u)$ on Ω_k for $k \in \{1,2\}$. Suppose that

$$\mathcal{L}_k u_k = f_k \quad \text{on } \Omega_k \text{ for } k \in \{1,2\},$$

and define

$$u(x,y) = \begin{cases} u_1(x,y) & \text{for } (x,y) \in \Omega_1, \\ u_2(x,y) & \text{for } (x,y) \in \Omega_2, \end{cases} \qquad \text{and} \qquad f(x,y) = \begin{cases} f_1(x,y) & \text{for } (x,y) \in \Omega_1, \\ f_2(x,y) & \text{for } (x,y) \in \Omega_2. \end{cases}$$

Under what condition(s) on u_1 and u_2 do u and f satisfy

$$\int_\Omega a\nabla u\cdot\nabla v = \int_\Omega fv - \int_{\partial\Omega} a\frac{\partial u}{\partial n}v$$

for any test function v? Hint: let $\mathbf{n}_i$ denote the unit normal along Γ_i, outward to Ω_1 and inward to Ω_2.

7.5. Prove that the functions $\chi_j \in V_h$ satisfying (7.5) form a basis for the piecewise-linear, finite element space V_h, that is, prove that the nodal basis really is a basis.

7.6. Let $\mathbf{A}$ be a nonsingular matrix. Show that $(\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1}$; we denote this *inverse transpose* matrix by $\mathbf{A}^{-T}$.

7.7. Consider a three-point quadrature rule of the form

$$\int_K f \approx \frac{\text{area}(K)}{3} \sum_{p=1}^{3} f(\mathbf{x}_p^K) \tag{7.19}$$

where, for some choice of the parameter $\lambda \in (0,1)$,

$$\begin{aligned} \mathbf{x}_1^K &= (1-2\lambda)\mathbf{a}_1 + \lambda\mathbf{a}_2 + \lambda\mathbf{a}_3, \\ \mathbf{x}_2^K &= \lambda\mathbf{a}_1 + (1-2\lambda)\mathbf{a}_2 + \lambda\mathbf{a}_3, \\ \mathbf{x}_3^K &= \lambda\mathbf{a}_1 + \lambda\mathbf{a}_2 + (1-2\lambda)\mathbf{a}_3. \end{aligned} \tag{7.20}$$

Verify that this rule integrates all quadratic polynomials exactly iff $\lambda = 1/6$ or $1/2$. See Figure **??**.

7.8. To achieve higher accuracy, we can consider a six-point quadrature rule

$$\int_K f \approx \frac{\text{area}(K)}{3}\left(w \sum_{p=1}^{3} f\big(\mathbf{x}_p^K(\lambda_1)\big) + (1-w) \sum_{p=1}^{3} f\big(\mathbf{x}_p^K(\lambda_2)\big) \right),$$

depending on the parameters w, λ_1 and λ_2, with $\mathbf{x}_p^K = \mathbf{x}_p^K(\lambda)$ defined as in (7.20).

7.9. Consider the model

$$-\nabla^2 u = 0 \text{ in } \Omega, \qquad u = g \text{ on } \Gamma_D, \qquad \frac{\partial u}{\partial n} = 0 \text{ on } \Gamma_N,$$

where g is piecewise linear on Γ_D, $\Omega \subset \mathbb{R}^2$ is a polygonal domain. The boundary $\Gamma = \partial\Omega$ is the union of two non-overlapping parts Γ_D and Γ_N.

(i) Define the weak formulation.

(ii) Define the finite element solution u_h on a regular triangular mesh with maximum element diameter h. Assume that $u_h = g$ on Γ_D.

(iii) Show that $\|\nabla u - \nabla u_h\| \leqslant Ch$.

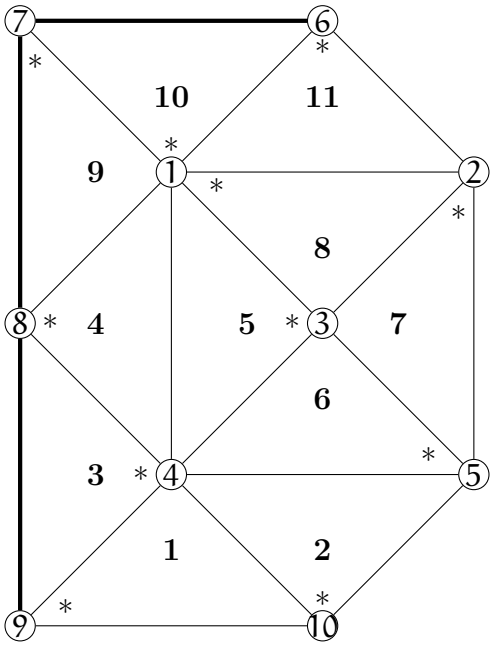
(iv) Give an example of a regular triangulation and a non-regular triangulation of $\Omega := (0,1) \times (0,1)$.

(v) Consider the triangulation shown in Figure 7.9. The global node numbers are circled and the element numbers are in bold. The choice of the first node in each element is indicated with an asterisk, after which the second and third follow **counterclockwise**. The part Γ_D of the boundary where a Dirichlet boundary condition applies is shown in a thicker line.

1) Write out the 3×11 connectivity matrix T

2) Which nodes are free and which are fixed? Then, find the dimensions of V_h and T_h.

Figure 7.10: Triangulation for Exercise 7.9.



7.10. Consider the triangulation shown in Figure 7.11. The global node numbers are circled and the element numbers are in bold. The choice of the first node in each element is indicated with an asterisk, after which the second and third follow **counterclockwise**. The part Γ_D of the boundary where a Dirichlet boundary condition applies is shown in a thicker line (between nodes 6 and 7). Let $\mathbf{f} = [f_r]$ and $\mathbf{A} = [a_{rs}]$ denote the global load vector and the global stiffness matrix, and let $\mathbf{f}^p = [f_j^p]$ and $\mathbf{A}^p = [a_{jk}^p]$ denote the element load vector and element stiffness matrix for the pth element ($1 \leqslant p \leqslant 6$).

(i) Write out the 3×6 connectivity matrix.

(ii) Express f_4 as a sum over entries f_j^p of the element load vectors.

(iii) Express a_{22}, a_{35} and a_{47} as sums over entries a_{jk}^p of the element matrices.

7.11. Consider the finite element method for a boundary-value problem (7.2) using the triangulation shows in Fig. 7.12. Note that Γ_D consists of the bottom and right sides of Ω (that is, the thicker edges numbered 6–8.)

(i) What are M^{free} and M^{fix}, the numbers of free and fixed nodes?

(ii) What is Q_N, the number of edges along Γ_N?

(iii) Write down the triangle connectivity matrix $\mathbf{T}^{\mathcal{K}}$.

(iv) Write down the edge connectivity matrix $\mathbf{T}^{\mathcal{E}}$.

(v) What are the dimensions of the global load vector $\mathbf{f} = [f_r]$, the global stiffness matrix $\mathbf{A} = [a_{rs}]$ and the global Neumann vector $\mathbf{g}_N = [g_{N,r}]$?

(vi) Express each f_r as a sum of entries f_i^p from the element load vectors $\mathbf{f}^1$, $\mathbf{f}^2$, …, $\mathbf{f}^{10}$.

(vii) Express each nonzero a_{rs} as a sum of entries a_{ij}^p from the element stiffness matrices $\mathbf{A}^1$, $\mathbf{A}^2$, …, $\mathbf{A}^{10}$. (From symmetry, it suffices to list the cases with $r \leqslant s$.)

Figure 7.11: Triangulation for Exercise 7.10.

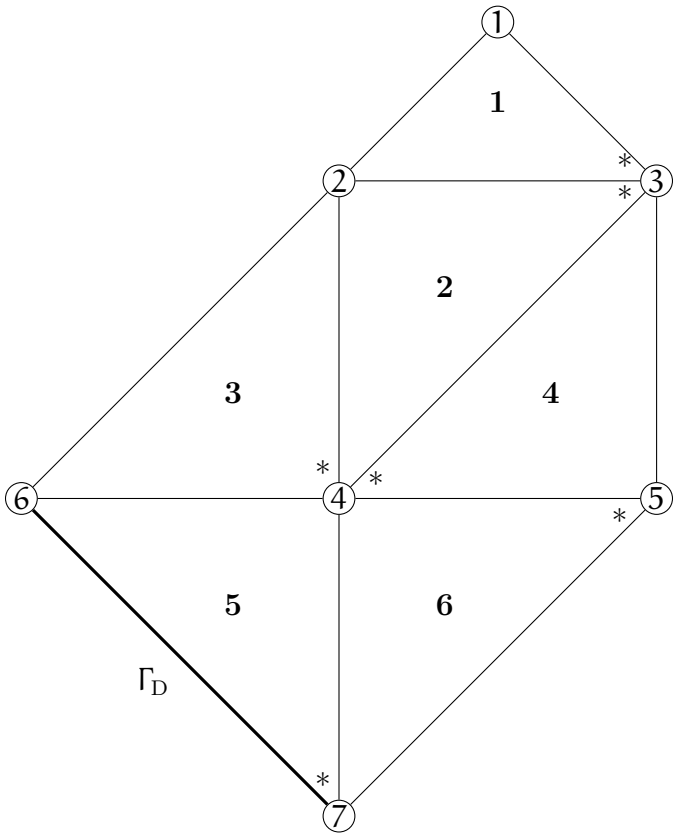
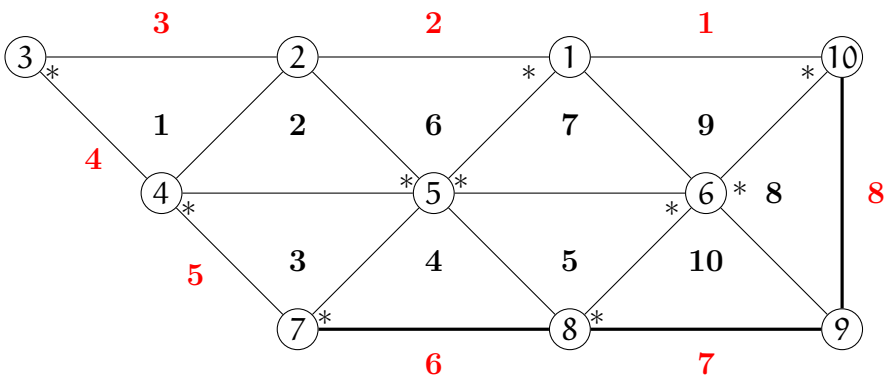


Figure 7.12: Triangulation for Exercise 7.11.



Chapter 8

Numerical solutions for convection and wave models

In this chapter, we briefly discuss the numerical solution via finite element/difference methods for solving two different models: advection-diffusion (convection) and second-order wave equations. More precisely, we apply the first order Euler method for the time-discretization combined with the piecewise linear finite element for the spatial discretization. We follow the notations of the previous chapters.

8.1 Advection-diffusion models

Consider the following advection-diffusion equation

$$u_t - \nabla \cdot \left(a \nabla u - \vec{F} u\right) = 0 \quad \text{for } x \in \Omega \text{ and } 0 < t < T, \tag{8.1}$$

with initial condition $u(x, 0) = g_0(x)$, where u_t denotes $\partial u/\partial t$ and Ω is a convex polyhedral domain in $\mathbb{R}^d$ ($d \geqslant 1$). Here, $a \geqslant a_{\min} > 0$ is the diffusivity coefficient function and $\vec{F}$ is the time-space general driving force.

We impose a homogeneous Dirichlet boundary condition,

$$u(x, t) = 0 \quad \text{for } x \in \Gamma := \partial\Omega \text{ and } 0 < t < T. \tag{8.2}$$

For a zero-flux boundary condition;

$$a \frac{\partial u}{\partial n} - (\vec{F} \cdot \vec{n})\, u = 0 \quad \text{for } x \in \Gamma \text{ and } 0 < t < T, \tag{8.3}$$

where $\vec{n}$ denotes the outward unit normal to Ω, our numerical scheme remains valid with $V_h = S_h = T_h$. Notice that the total mass $\int_\Omega u(\cdot, t)$ within Ω is conserved in this case.

In the weak formulation of (8.1), subject to the homogeneous Dirichlet boundary condition (8.2), we take the $L_2(\Omega)$-inner product of (8.1) with a test function v, and then, applying the first Green identity in Theorem 7.2 yield

$$\langle u_t, v \rangle + \langle a \nabla u, \nabla v \rangle - \langle \vec{F} u, \nabla v \rangle = 0 \quad \text{provided } v = 0 \text{ on } \Gamma. \tag{8.4}$$

8.1.1 Approximate solutions

Recalling that, h denotes the maximum diameter of the elements in a shape-regular triangulation $\mathcal{T}_h$ of Ω, V_h is the the usual space of continuous, piecewise-linear functions on $\mathcal{T}_h$, and

$$S_h = \{ v \in V_h : v = 0 \text{ on } \Gamma \}.$$

The semidiscrete finite element solution $u_h : [0, T] \to S_h$ is then defined by

$$\langle u_{ht}, v\rangle + \langle a\nabla u_h, \nabla v\rangle - \langle \vec{F}u_h, \nabla v\rangle = 0 \quad \text{for all } v \in S_h, \tag{8.5}$$

together with the initial condition $u_h(0) = g_{0h}$, where $g_{0h} \in S_h$ is a suitable approximation to g_0.

For the time discretization, we simply use the first order implicit Euler method, and we obtain the following fully-discrete scheme: $U_h^n \approx u_h(t_n)$, such that

$$\frac{1}{\tau}\langle U_h^n - U_h^{n-1}, v\rangle + \langle a\nabla U_h^n - \vec{F}(t_n)U_h^n, \nabla v\rangle = 0 \quad \text{for all } v \in S_h, \tag{8.6}$$

for $1 \leqslant n \leqslant N$, together with the initial condition $U_h^0 = g_{0h}$, where $g_{0h} \in S_h$ is a suitable approximation to g_0.

Remark If we have a nonlinear advection term, for example, $f(u)\cdot\nabla u$ instead of $\nabla\cdot\big(\vec{F}u\big)$, then it is clear from the matrix form in (8.7) that the obtained algebraic system is nonlinear. So, we have to deal with nonlinear solvers. This can be avoided sometimes by linearizing the scheme. For instance, replacing the term $f(U_h^n)\cdot\nabla U_h^n$ with $f(U_h^{n-1})\cdot\nabla U_h^n$.

8.1.2 Matrix form

Recalling that the dimension of S_h, $\dim S_h$, is equal to the number of interior nodes $\{p_j\}$, say M, and $\{\chi_1, \chi_2, \ldots, \chi_M\}$ is the set of nodal basis for S_h. So, the numerical solution U_h^n has the following representation: for $1 \leqslant n \leqslant N$,

$$U_h^n(\vec{x}) = \sum_{j=1}^{M} \alpha_j^n \chi_j(x), \quad \text{where } \alpha_j^n = U_h^n(\mathbf{p}_j),$$

with $\alpha_j^0 = U_h^0(\mathbf{p}_j) = U_{h0}(\mathbf{p}_j)$. Inserting this representation in (8.6), we obtain the following $M \times M$ linear system

$$\frac{1}{\tau}\sum_{j=1}^{M} d_{i,j}\big(\alpha_j^n - \alpha_j^{n-1}\big) + \sum_{j=1}^{M} a_{i,j}^n \alpha_j^n = 0 \quad \text{for } 1 \leqslant i \leqslant M,$$

and for $1 \leqslant n \leqslant N$, where

$$a_{i,j}^n = \langle a\nabla\chi_j - \vec{F}(t_n)\chi_j, \nabla\chi_i\rangle \quad \text{and} \quad d_{i,j} = \langle \chi_j, \chi_i\rangle\,.$$

Introduce the column vector $\alpha^n = [\alpha_j^n]_{j=1}^M$, and then,

$$\big(\mathbf{D} + \tau\mathbf{A}^n\big)\alpha^n = \mathbf{D}\alpha^{n-1}, \quad \text{for } 1 \leqslant n \leqslant N, \tag{8.7}$$

with $\mathbf{A} = [a_{i,j}^n]_{i,j=1}^M$ and $\mathbf{D} = [d_{i,j}]_{i,j=1}^M$.

8.1.3 Stability

To show the stability of the finite element solution U_h^n in (8.6), we choose $v = U_h^n$ and observe after applying the Cauchy-Schwarz inequality,

$$\begin{aligned}
\|U_h^n\|^2 + \tau\|\sqrt{a}\nabla U_h^n\|^2 &= \langle U_h^{n-1}, U_h^n\rangle + \tau\langle \vec{F}(t_n)/\sqrt{a}\, U_h^n, \sqrt{a}\,\nabla U_h^n\rangle \\
&\leqslant \|U_h^{n-1}\|\,\|U_h^n\| + \Big(\max|\vec{F}(t_n)/\sqrt{a}|\Big)\tau\|U_h^n\|\,\|\sqrt{a}\,\nabla U_h^n\| \\
&\leqslant \frac{1}{2}\|U_h^{n-1}\|^2 + \frac{1}{2}\|U_h^n\|^2 + C\tau\|U_h^n\|^2 + \frac{1}{2}\tau\|\sqrt{a}\,\nabla U_h^n\|^2\,.
\end{aligned}$$

Canceling the similar terms yield

$$\frac{1}{2}\|U_h^n\|^2 + \frac{\tau}{2}\|\sqrt{a}\nabla U_h^n\|^2 \leqslant \frac{1}{2}\|U_h^{n-1}\|^2 + C\tau\|U_h^n\|^2.$$

Replacing n with j,

$$\|U_h^j\|^2 \leqslant \|U_h^{j-1}\|^2 + c_0\tau\|U_h^j\|^2, \quad \text{for } 1 \leqslant j \leqslant N,$$

for some positive constant c_0. Summing over j from 1 through n,

$$\|U_h^n\|^2 \leqslant \|U_h^0\|^2 + c_0\sum_{j=1}^{n}\tau\|U_h^j\|^2, \quad \text{for } 1 \leqslant n \leqslant N\,.$$

Assume that $\tau \leqslant 1/(2c_0)$ gives

$$\|U_h^n\|^2 \leqslant 2\|U_h^0\|^2 + 2c_0\sum_{j=1}^{n-1}\tau\|U_h^j\|^2, \quad \text{for } 1 \leqslant n \leqslant N\,.$$

An application of the discrete Gronwall inequality leads to the following stability property

$$\|U_h^n\|^2 \leqslant C\|g_{0h}\|^2, \quad \text{for } 1 \leqslant n \leqslant N\,.$$

8.1.4 Existence and uniqueness

As demonstrated in (8.7), the finite element scheme in (8.6) amounts into a finite square linear system. Because of this, the existence of the numerical solution U_h^n follows from its uniqueness. To show the latter, assume that we have two solutions, say U_h^n and W_h^n. Let $\eta^n = U_h^n - W_h^n$, then, from (8.6),

$$\frac{1}{\tau}\langle\eta^n - \eta^{n-1}, v\rangle + \langle a\nabla\eta^n - \vec{F}(t_n)\eta^n, \nabla v\rangle = 0 \quad \text{for all } v \in S_h,$$

for $1 \leqslant n \leqslant N$, together with the initial condition $\eta^0 = g_{0h} - g_{0h}$. Hence, applying the stability property in (8.8) gives

$$\|\eta^n\| \leqslant 0, \quad \text{for} \quad 1 \leqslant n \leqslant N, \tag{8.8}$$

and therefore, $U_h^n - W_h^n = 0$. This completes the proof of uniqueness, and consequently, of the existence of the numerical solution U_h^n for $1 \leqslant n \leqslant N$.

In principle, the proposed numerical scheme in (8.6) is first-order accurate in time and second-order accurate in space, that is,

$$\|u(t_n) - U_h^n\| \leqslant C(\tau + h^2)\,.$$

8.2 Wave models

Consider the following linear wave equation

$$u_{tt} - \nabla\cdot\left(a^2\nabla u\right) = 0 \quad \text{for } x \in \Omega \text{ and } 0 < t < T, \tag{8.9}$$

with initial conditions $u(x,0) = g_0(x)$ and $u_t(x,0) = g_1(x)$, and the constant a defines speed at which the wave moves. In this model, u_t is $\partial u/\partial t$, u_{tt} is $\partial^2 u/\partial t^2$, and Ω is a convex polyhedral domain in $\mathbb{R}^d$ ($d \geqslant 1$).

Again, we impose a homogeneous Dirichlet boundary condition,

$$u(x,t) = 0 \quad \text{for } x \in \Gamma := \partial\Omega \text{ and } 0 < t < T. \tag{8.10}$$

8.2.1 Approximate solutions and matrix form

In the weak formulation of (8.9), subject to the homogeneous Dirichlet boundary condition (8.10), we take the $L_2(\Omega)$-inner product of (8.1) with a test function v, and then, applying the first Green identity in Theorem 7.2 yield

$$\langle u_{tt}, v\rangle + \langle a^2\nabla u, \nabla v\rangle = 0 \quad \text{provided } v = 0 \text{ on } \Gamma. \tag{8.11}$$

Recalling that, h denotes the maximum diameter of the elements in a shape-regular triangulation $\mathcal{T}_h$ of Ω, V_h is the the usual space of continuous, piecewise-linear functions on $\mathcal{T}_h$, and

$$S_h = \{v \in V_h : v = 0 \text{ on } \Gamma\}.$$

The semidiscrete finite element solution $u_h : [0, T] \to S_h$ is then defined by

$$\langle u_{htt}, v\rangle + \langle a^2\nabla u_h, \nabla v\rangle = 0 \quad \text{for all } v \in S_h, \tag{8.12}$$

together with the initial condition $u_h(0) = g_{0h}$ and $u_{ht}(0) = g_{1h}$ where g_{0h} and g_{1h} are some suitable approximations of g_0 and g_1 in the finite element space S_h, respectively.

For the time discretization, we use the second central differences to approximate the term u_{htt}. The fully-discrete scheme is defined as: Find $U_h^n \approx u_h(t)$, such that

$$\frac{1}{\tau^2}\langle U_h^{n+1} - 2U_h^n + U_h^{n-1}, v\rangle + \frac{a^2}{4}\langle\nabla(U_h^{n+1} + 2U_h^n + U_h^{n-1}), \nabla v\rangle = 0 \quad \text{for all } v \in S_h,$$

for $1 \leqslant n \leqslant N-1$, with the initial condition $U_h^0 = g_{0h}$ and $U_h^1 \in S_h$ approximates $u(t_1)$ that will be defined later. Alternatively, the above scheme can be rewritten in a compact form as:

$$\frac{1}{\tau}\langle\partial U_h^{n+1} - \partial U_h^n, v\rangle + \frac{a^2}{2}\langle\nabla(U_h^{n+1/2} + U_h^{n-1/2}), \nabla v\rangle = 0 \quad \text{for all } v \in S_h, \tag{8.13}$$

for $1 \leqslant n \leqslant N-1$, with

$$\partial U_h^n = \frac{U_h^n - U_h^{n-1}}{\tau}, \quad \text{and} \quad U_h^{n+1/2} = \frac{U_h^{n+1} + U_h^n}{2}.$$

As in the previous section, the numerical solution U_h^n has the following representation: for $2 \leqslant n \leqslant N$,

$$U_h^n(x) = \sum_{j=1}^{M} \alpha_j^n \chi_j(x), \quad \text{where } \alpha_j^n = U_h^n(\mathbf{p}_j),$$

with $\alpha_j^0 = U_h^0(\mathbf{p}_j) = U_{h0}(\mathbf{p}_j)$ and $\alpha_j^1 = U_h^1(\mathbf{p}_j)$. Inserting this representation in (8.13), we obtain the following $M \times M$ linear system

$$\frac{1}{\tau^2}\sum_{j=1}^{M} d_{i,j}\left(\alpha_j^{n+1} - 2\alpha_j^n + \alpha_j^{n-1}\right) + \frac{1}{4}\sum_{j=1}^{M} a_{i,j}\left(\alpha_j^{n+1} + 2\alpha_j^n + \alpha_j^{n-1}\right) = 0 \quad \text{for } 1 \leqslant i \leqslant M,$$

and for $1 \leqslant n \leqslant N-1$, where

$$a_{i,j} = a^2\langle\nabla\chi_j, \nabla\chi_i\rangle \quad \text{and} \quad d_{i,j} = \langle\chi_j, \chi_i\rangle.$$

Introduce the column vector $\alpha^n = [\alpha_j^n]_{j=1}^M$, and then,

$$(4\mathbf{D} + \tau^2\mathbf{A})\alpha^{n+1} = 4\mathbf{D}(2\alpha^n - \alpha^{n-1}) - \tau^2\mathbf{A}(2\alpha^n + \alpha^{n-1}), \quad \text{for } 1 \leqslant n \leqslant N-1, \tag{8.14}$$

with $\mathbf{A} = [a_{i,j}]_{i,j=1}^{M}$ and $\mathbf{D} = [d_{i,j}]_{i,j=1}^{M}$. To solve the above scheme recursively, we need to compute first α^0 and α^1 by using the initial conditions g_0 and g_1. For example, we may define U_h^0 using the Ritz projection as:

$$\langle \nabla U_h^0, \nabla v \rangle = \langle \nabla g_0, \nabla v \rangle, \quad \text{for all } v \in S_h.$$

The above scheme can be solved for α^0. Since u is not known at time $t = t_1$, the approximate solution U_h^1 cannot be defined directly. From Taylor Theorem, we have

$$u(t_1) = u(0) + \tau u_t(0) + \frac{\tau^2}{2} u_{tt}(0) + O(\tau^3) = g_0 + \tau g_1 + \frac{\tau^2}{2} a^2 \nabla^2 u(0) + O(\tau^3)\,.$$

So, we can approximate $u(t_1)$ by $\lambda := g_0 + \tau g_1 + \frac{\tau^2}{2} a^2 \nabla^2 g_0$. Consequently, we can define the approximate solution U_h^1 via the L_2 projection as:

$$\langle U_h^1, v \rangle = \langle \lambda, v \rangle, \quad \text{for all } v \in S_h.$$

8.2.2 Well-posedness

As in the previous section, our numerical scheme amounts into a finite square linear system, and so, it is enough to show the uniqueness of the numerical solution. Stability wise, we show an unconditional stability property of the numerical scheme in (8.13). To do so, we choose

$$v = \frac{1}{2\tau}(U_h^{n+1} - U_h^{n-1}) = \frac{1}{2}(\partial U_h^{n+1} + \partial U_h^n) = \frac{1}{\tau}(U_h^{n+1/2} - U_h^{n-1/2}).$$

in (8.13) and observe that

$$\mathcal{E}^{n+1} = \mathcal{E}^n, \quad \text{for } \ 1 \leqslant n \leqslant N-1,$$

where

$$\mathcal{E}^{n+1} = \|\partial U_h^{n+1}\|^2 + a^2 \|\nabla U_h^{n+1/2}\|^2\,.$$

Summing over n leads to

$$\mathcal{E}^{n+1} = \mathcal{E}^1, \quad 1 \leqslant n \leqslant N-1$$

that is, the numerical solution U_h^n satisfied the following stability property:

$$\|U_h^{n+1} - U_h^n\|^2 + a^2 \tau \|\nabla U_h^{n+1/2}\|^2 \leqslant \|U_h^1 - U_h^0\|^2 + a^2 \tau \|\nabla U_h^{1/2}\|, \ \text{ for } \ 1 \leqslant n \leqslant N-1\,.$$

This stability property can be used to show that the scheme in (8.13) is $O(\tau^2 + h^2)$ accurate, that is, it is second order accurate in both time and space.