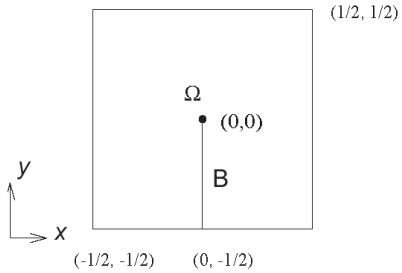

Figure 7.2: Definition region for rotating cone problem.

Consider the region Ω sketched in Figure 7.2.
The region consists of a square with a cut B. In the inner region we suppose that the concentration satisfies the convection-diffusion equation

$$-\varepsilon \triangle c + \mathbf{u} \cdot \nabla c = 0 \,, \tag{7.4.10}$$

where ε is chosen equal to 10^{-6} and the velocity $\mathbf{u}$ is such that the flow rotates counter clockwise. This is achieved by setting $\mathbf{u} = \begin{bmatrix} -y \\ x \end{bmatrix}$. At the outer boundary we use the boundary condition

$$c|_\Gamma = 0 \,. \tag{7.4.11}$$

On the starting curve B the concentration c is set equal to

$$c|_B = \cos(2\pi(y + \frac{1}{4})) \,, \tag{7.4.12}$$

and due to the small diffusion one expects that the concentration at the end curve is nearly the same. The end curve has the same co-ordinates as B but the nodal points differ, which means that the solution may be different from the starting one. Since no boundary condition is given at the outflow curve "B" implicitly the boundary condition

$$\varepsilon \frac{\partial c}{\partial n}|_B = 0 \,, \tag{7.4.13}$$

is prescribed. (Why?)

If we make contour lines (i.e. lines with equal values) of the concentration, we expect concentric circles. However, if we subdivide the region into 20×20 squares, each of which is subdivided into two triangles by drawing the diagonal in arbitrary direction we get a very irregular set of contour lines as can be seen in Figure 7.3.
If we solve the same problem with SUPG the result is much smoother as is shown in Figure 7.4. The fact that circles in this picture are not completely closed is due to the numerical diffusion.

7.5 An example of a system of coupled PDEs

We finish this chapter with the plane stress example of Section 5.4.3. This problem can be formulated in terms of a minimization problem, see Equation (5.4.5), but in this case we shall use the weak formulation.

Figure 7.3: Equi-concentration lines for rotating cone problem computed by SGA.

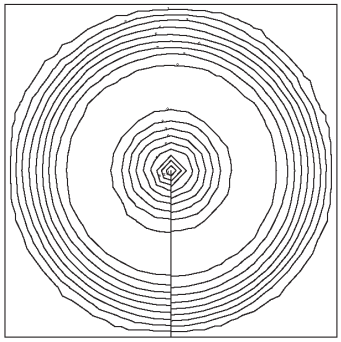

Figure 7.4: Equi-concentration lines for rotating cone problem computed by SUPG.

In Exercise 5.7.3 we have derived that the displacements (u, v) satisfy the equation

$$\begin{aligned} -\frac{\partial \sigma_{xx}}{\partial x} - \frac{\partial \tau_{xy}}{\partial y} &= 0\,, \\ -\frac{\partial \tau_{xy}}{\partial x} - \frac{\partial \sigma_{yy}}{\partial y} &= 0\,. \end{aligned} \tag{7.5.1}$$

Exercise 7.5.1
Show with the information of Section 5.4.3 that the boundary conditions for Equation (7.5.1) are given by:

$u = v = 0$ *on* Γ_1
$\sigma_{xx} n_x + \tau_{xy} n_y = t_1$, $\tau_{xy} n_x + \sigma_{yy} n_y = t_2$ *on* Γ_2
$\sigma_{xx} n_x + \tau_{xy} n_y = 0$, $\tau_{xy} n_x + \sigma_{yy} n_y = 0$ *on all other boundaries,*

with (n_x, n_y) *the normal on the boundary.* □

To derive the weak formulation we multiply the first equation of 7.5.1 by a test function δu and the second equation by δv and integrate over the domain Ω.

Exercise 7.5.2 *Show using the divergence theorem, that the weak formulation corresponding to Equation (7.5.1) with the boundary conditions given in Exercise 7.5.1 can be written*

as:

$$\int_\Omega \{\sigma_{xx}\frac{\partial \delta u}{\partial x} + \tau_{xy}\frac{\partial \delta u}{\partial y}\}\, d\Omega = \int_{\Gamma_2} t_1 \delta u\, d\Gamma, \tag{7.5.2}$$
$$\int_\Omega \{\tau_{xy}\frac{\partial \delta v}{\partial x} + \sigma_{yy}\frac{\partial \delta v}{\partial y}\}\, d\Omega = \int_{\Gamma_2} t_2 \delta v\, d\Gamma.$$

□

The next step is to apply the Galerkin method. We approximate u and v by a linear combination of basis functions

$u^n = \sum_{j=1}^{n} u_j \varphi_j(\mathbf{x}), v^n = \sum_{j=1}^{n} v_j \varphi_j(\mathbf{x}).$
The same basis functions for u and v are used. δu is replaced by $\varphi_i(\mathbf{x})$ for i in Σ_1, the set of non-prescribed u-velocity components and δv by $\varphi_i(\mathbf{x})$ for i in Σ_2, the set of non-prescribed v-velocity components.

In this particular example Σ_1 and Σ_2 are identical.

Exercise 7.5.3 *Show that the system of Galerkin equations corresponding to the weak formulation (7.5.2) is given by*

$$\sum_{j=1}^{n} u_j \int_\Omega \{A\frac{\partial \varphi_j}{\partial x}\frac{\partial \varphi_i}{\partial x} + B\frac{\partial \varphi_j}{\partial y}\frac{\partial \varphi_i}{\partial y}\}\, d\Omega + \sum_{j=1}^{n} v_j \int_\Omega \{\nu A\frac{\partial \varphi_j}{\partial y}\frac{\partial \varphi_i}{\partial x} + B\frac{\partial \varphi_j}{\partial x}\frac{\partial \varphi_i}{\partial y}\}\, d\Omega$$
$$= \int_{\Gamma_2} t_1 \varphi_i\, d\Gamma,\ i \in \Sigma_1 \tag{7.5.3}$$
$$\sum_{j=1}^{n} u_j \int_\Omega \{B\frac{\partial \varphi_j}{\partial y}\frac{\partial \varphi_i}{\partial x} + \nu A\frac{\partial \varphi_j}{\partial x}\frac{\partial \varphi_i}{\partial y}\}\, d\Omega + \sum_{j=1}^{n} v_j \int_\Omega \{B\frac{\partial \varphi_j}{\partial x}\frac{\partial \varphi_i}{\partial x} + A\frac{\partial \varphi_j}{\partial y}\frac{\partial \varphi_i}{\partial y}\}\, d\Omega$$
$$= \int_{\Gamma_2} t_2 \varphi_i\, d\Gamma,\ i \in \Sigma_2$$

□

To apply the FEM, the region Ω is subdivided into triangular elements. The same linear basis functions as in Section 6.3.4 are used. In each point there are two unknowns so the size of the element matrix must be 6×6 and the element vector has 6 entries.

Suppose we order the triangle unknowns u_1, u_2, u_3, v_1, v_2, v_3 with 1, 2 and 3 the local node numbers of the triangle. The rows of the element matrix are of course ordered in the same way. Then the element matrix can be split into 4 submatrices:

$$S^{e_k} = \begin{bmatrix} S^{e_k}_{uu} & S^{e_k}_{uv} \\ S^{e_k}_{vu} & S^{e_k}_{vv} \end{bmatrix}. \tag{7.5.4}$$

Exercise 7.5.4 *Compute the elements of the four subelement matrices under the condition that ν and E are constant.* □

7.6 Mathematical theory

Existence and uniqueness of solutions of the weak formulation are in general much more difficult to prove than in case of a minimization problem. However, for a subclass of problems, it is possible to give some general theory.

Consider a linear PDE of order $2m$ of the form

$$\mathbf{Lu} = \mathbf{f}. \tag{7.6.1}$$

The weak formulation can be written as a bilinear form:

$$a(u, v) = (f, v), \tag{7.6.2}$$

where $a(u, v)$ contains only derivatives of order m. This means that we have removed all higher order derivatives by integration by parts. The solution must be found in some Hilbert space (more precisely a Sobolev space H^m) and the test functions are arbitrary functions in that space. Now we can prove the following theorem

Theorem 7.6.1 *Let V_0 be a real Hilbert space with inner product $(.,.)_0$. Let V_1 be a closed subspace of V_0 with inner product $(.,.)_1$. Let $a(u, v)$ be a positive continuous bilinear form mapping $V_1 \times V_1 \to R$, which means*

$$\begin{aligned}
a(u, v + w) &= a(u, v) + a(u, w), && (7.6.3)\\
a(u + w, v) &= a(u, v) + a(w, v), && (7.6.4)\\
a(\lambda u, v) &= a(u, \lambda v) = \lambda a(u, v), && (7.6.5)\\
|a(u, v)| &\leq K\|u\|_1\|v\|_1, && (7.6.6)\\
a(u, u) &\geq \gamma_1\|u\|_1^2, && (7.6.7)\\
\|u\|_1 &\geq \gamma_0\|u\|_0 && (7.6.8)
\end{aligned}$$

(7.6.3) to (7.6.5) imply linearity, (7.6.6) is continuity and (7.6.7) means positiveness. Let f be an element of V_0. Then the weak formulation

$$\textit{find } u \in V_1 \textit{ such that } a(u, v) = (f, v)_0 \ \forall v \in V_1, \tag{7.6.9}$$

has exactly one solution in V_1.

Proof
The Lax-Milgram theorem (see for example [49], page 92) states that, under conditions (7.6.3) to (7.6.7), for each linear functional $F(v)$ on V_1 there is precisely one element $w \in V_1$ such that

$$a(w, v) = F(v), \ \forall v \in V_1. \tag{7.6.10}$$

According to (7.6.8) for a given $f \in V_0$ the inner product $(f, v)_0$ is a bounded linear functional on V_1, since

$$(f, v)_0 \leq \|f\|_0\|v\|_0 \leq \frac{1}{\gamma_0}\|f\|_0\|v\|_1. \tag{7.6.11}$$

This proves the theorem. □

Remark: if we compare (7.6.3) to (7.6.7) with (5.9.5) to (5.9.7), then we see that the main difference is the symmetry requirement. This is necessary for the minimization problem, but not for the existence of the solution of the weak formulation. (7.6.6), (7.6.7) are necessary to guarantee that $a(u, u)^{\frac{1}{2}}$ is an equivalent norm of

$\|.\|_1$. In (5.9.5) to (5.9.7) $a(.,.)$ created itself V_1, and hence $a(u,u)^{\frac{1}{2}}$ was the norm on V_1. In that case(7.6.6), (7.6.7) are satisfied automatically.

In fact now we require that the symmetrical part of $a(.,.)$ is positive definite.

The next theorem gives an error estimate in terms of the "1"-norm, for our finite dimensional approximation.

Theorem 7.6.2 *Let V_0, V_1 and $a(.,.)$ be defined as in Theorem 7.6.1. Let V_{1h} be a finite dimensional subspace of V_1. Let u be the solution of the weak problem*

$$a(u,v) = (f,v)_0 \ \ \forall v \in V_1, \tag{7.6.12}$$

and let u_h be the solution of the finite dimensional problem

$$a(u_h, v_h) = (f, v_h)_0 \ \ \forall v_h \in V_{1h}. \tag{7.6.13}$$

Then we have the following estimate:

$$\|u - u_h\|_1 \leq \frac{K}{\gamma_1} \min_{v_h \in V_{1h}} \|u - v_h\|_1. \tag{7.6.14}$$

Proof
Since V_{1h} is a subspace of V_1

$$a(u, v_h) = (f, v_h)_0 \ \ \forall v_h \in V_{1h}, \tag{7.6.15}$$

and from (7.6.13)

$$a(u - u_h, v_h) = 0 \ \ \forall v_h \in V_{1h}, \tag{7.6.16}$$

From (7.6.7) and (7.6.16) it follows that

$$\gamma_1 \|u - u_h\|^2 \leq a(u - u_h, u - u_h) = a(u - u_h, u). \tag{7.6.17}$$

Because of (7.6.16) we get

$$\gamma_1 \|u - u_h\|^2 \leq a(u - u_h, u - v_h) \ \ \forall v_h \in V_{1h}, \tag{7.6.18}$$

and due to the continuity of a (7.6.6)

$$\gamma_1 \|u - u_h\|^2 \leq K \|u - u_h\| \|u - v_h\| \ \ \forall v_h \in V_{1h}. \tag{7.6.19}$$

So

$$\|u - u_h\| \leq \frac{K}{\gamma_1} \|u - v_h\| \ \ \forall v_h \in V_{1h}. \tag{7.6.20}$$

which proves the theorem. □

Theorem 7.6.2 shows that the error in the solution in energy norm is smaller than a constant times the interpolation error in energy norm. So the error estimate is of the same type as for minimization problems, although the constant is in general larger than 1. So the estimate is not as sharp as it is for a minimization problem.

7.7 Summary of Chapter 7

Instead of deriving a minimization problem corresponding to a PDE, one may also derive a weak formulation by multiplying the PDE by a test function. Integration by parts may be used to get rid of higher order derivatives. Natural boundary conditions are automatically part of the weak formulation due to this integration

by parts. This method can be applied to general PDEs and is therefore much more applicable than the minimization approach.

To approximate the solution, Galerkin's method is applied, which is a direct generalization of the Ritz method. The solution is approximated by a finite set of basis functions and the test function runs through the same set. Application of the FEM to Galerkin is completely identical to the use of FEM in case of Ritz.

By using test functions that are different from the basis functions one gets the Petrov-Galerkin method. A typical application is SUPG, the finite element equivalent of upwind differencing.

Chapter 8

Extension of the FEM

Objectives

In the previous Chapters 6 and 7 we have introduced the FEM as a technique to construct basis functions for Ritz and Galerkin. Until now we have limited ourselves to linear interpolation polynomials in $\mathbb{R}^1$ and $\mathbb{R}^2$. In $\mathbb{R}^2$ this means automatically that we have to use triangles.

In this chapter we shall extend the theory to higher order polynomials. Furthermore we shall show how quadrilaterals can be handled. In each case we have to check whether these elements satisfy the continuity requirements formulated in Chapter 6. This will lead to the *isoparametric transformations*. Finally we shall show that these requirements, in the case of fourth order PDEs, lead to complicated elements. In practice this is solved by reducing the fourth order problem as a set of two second order problems. In this way the continuity requirements may be reduced.

8.1 (Straight) quadratic triangles

One may expect that the linear interpolations we have used so far, lead to errors of the order h^2, provided the solution is smooth enough. If we want a higher order accuracy, it is necessary to use higher order polynomials. In this section we shall derive the basis functions corresponding to quadratic interpolation. Elements using quadratic interpolation polynomials are usually addressed as *quadratic elements*.

In $\mathbb{R}^1$ the situation is simple. A quadratic interpolation polynomial over an element with vertices $x_1$and x_3 and midpoint x_2 can be written as (compare with (6.2.4) - (6.2.5))

$$u(x) = l_1(x)u_1 + l_2(x)u_2 + l_3(x)u_3, \tag{8.1.1}$$

with $l_i(x)$ the quadratic Lagrangian polynomials defined by

$$l_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}, \tag{8.1.2a}$$

$$l_2(x) = \frac{(x - x_3)(x - x_1)}{(x_2 - x_3)(x_2 - x_1)}, \tag{8.1.2b}$$

$$l_3(x) = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2).} \tag{8.1.2c}$$

Again these polynomials satisfy

$$l_j(x_i) \;=\; \delta_{ij}. \tag{8.1.3}$$

So the basis functions $\varphi_i(x)$ can be defined by

$$\begin{aligned} &\varphi_i(\mathbf{x}) \quad \text{quadratic,} \\ &\varphi_i(\mathbf{x}_j) = \delta_{ij}, \quad i,j,= 1,2,\cdots,n+1\,. \end{aligned} \tag{8.1.4}$$

Figure 8.1 shows a "vertex" basis function and Figure 8.2 a basis function corresponding to the mid point of an element.

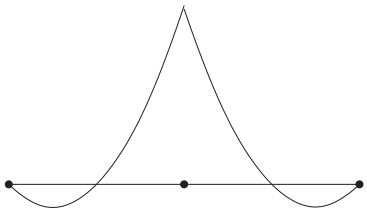

Figure 8.1: Basis function for a vertex.

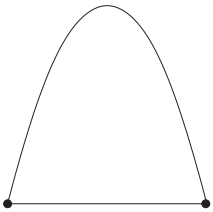

Figure 8.2: Midpoint basis function.

Theorem 8.1.1 *The Newton Cotes integration rule for the quadratic elements in* $\mathbb{R}^1$ *is given by*

$$\int_e f(x)\,dx \;=\; \frac{h}{6}[f(x_1) + 4f(x_2) + f(x_3)], \tag{8.1.5}$$

with h the length of the interval $[x_1, x_3]$ *and* x_2 *the mid point. This is of course Simpson's rule.*

Exercise 8.1.1 *Prove Theorem (8.1.1).* □

Exercise 8.1.2 *Compute the element matrix and element vector for Poisson's equation (6.2.1) using quadratic polynomials. Use the Newton Cotes rule (8.1.5) to compute the integrals.* □

The extension to $\mathbb{R}^2$ is more complex. First we shall limit ourselves to quadratic triangles with straight sides. The extension to curved sides is treated in Section (8.4).

A quadratic polynomial in $\mathbb{R}^2$ is uniquely defined by 6 parameters (why?). In Chapter 6 we have seen that for second order PDEs, it is necessary that the basis functions are continuous. In order to get continuity, it is necessary that the values of the interpolation on the common edge of two adjacent triangles are equal for both triangles. An edge is one dimensional. A quadratic polynomial in $\mathbb{R}^1$ is uniquely defined by 3 parameters, hence we need three nodal points on each edge of the triangle. So it is natural to use vertices and the midside points as nodes of the triangle. See Figure 8.3.

The 6 basis functions on the triangle are implicitly defined by the requirements (8.1.4). If we write the quadratic polynomial $\varphi_i(x)$ as

$$\varphi_i(x) = \alpha_i + \beta_i x + \gamma_i y + \delta_i x^2 + \epsilon_i xy + \eta_i y^2, \tag{8.1.6}$$

the coefficients $\alpha_i, \ldots \eta_i$ can be computed by solving a 6×6 system of linear equations. To do this element by element is relatively expensive. A more subtle approach is to express the basis functions $\varphi_i(\mathbf{x})$ in terms of the linear basis functions

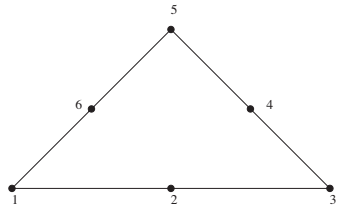

Figure 8.3: Nodes of quadratic triangle.

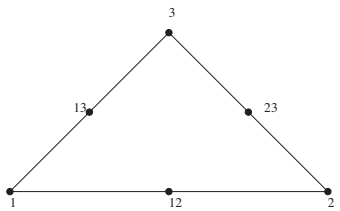

Figure 8.4: Special numbering of nodes.

$\lambda_i(\mathbf{x})$ (6.3.22) corresponding to the vertices of the triangle. In this particular case it is easier to use the local numbering of Figure 8.4 to define the basis functions. The basis functions corresponding to the vertices are denoted by $\psi_i(\mathbf{x})$ and the basis functions corresponding to the midpoints by $\psi_{ij}(\mathbf{x})$. The relation between φ_i and ψ_i is trivial.

First we consider the basis functions $\psi_i(\mathbf{x})$ corresponding to the vertices.

Since $\psi_i(\mathbf{x}_j) = \delta_{ij}$ an obvious choice is to define $\psi_i = \lambda_i v_i$ with v_i a linear function. (Why?).
This function satisfies 3 of the six equations automatically.
From $\psi_i(\mathbf{x}_i) = 1$ it follows that $v_i(\mathbf{x}_i) = 1$.
From $\psi_i(\mathbf{x}_{kl}) = 0$ it follows that $v_i(\mathbf{x}_{kl}) = 0$ for $k = i$ or $l = i$.
Because the points x_{ij} are in the middle of the sides this implies that
$v_i(\mathbf{x}_j) = -1$ and $v_i(\mathbf{x}_k) = -1$.
So v_i can be written as $\lambda_i - \lambda_j - \lambda_k = 2\lambda_i - 1$ (Theorem 6.3.3) $i \neq j,\ i \neq k,\ j \neq k$.

Next consider the mid points $\mathbf{x}_{ij}$.

Since $\psi_{ij}(\mathbf{x}_k) = 0$, $\psi_{ij}(\mathbf{x}_{kl}) = 0$ if $ij \neq kl$ we have $\psi_{ij} = 0$ on the sides not containing point ij. So a natural choice is $\psi_{ij} = \alpha\lambda_i\lambda_j$. From $\psi_{ij}(\mathbf{x}_{ij}) = 1$ it follows that $\alpha = 4$.

In conclusion in each element the quadratic basis functions can be expressed in the linear basis functions by

$$\begin{aligned} \psi_i &= \lambda_i(2\lambda_i - 1), \\ \psi_{ij} &= 4\lambda_i\lambda_j. \end{aligned} \tag{8.1.7}$$

Theorem 8.1.2 *The Newton Cotes formula for the quadratic triangle is given by*

$$\int_e Int(\mathbf{x})\, d\Omega = \frac{|\Delta|}{6}[Int(\mathbf{x}_{12}) + Int(\mathbf{x}_{23}) + Int(\mathbf{x}_{13})]. \tag{8.1.8}$$

Exercise 8.1.3 *Prove Theorem (8.1.2).* □

Exercise 8.1.4 *Compute the element matrix and vector corresponding to the Poisson equation (6.3.1) for a quadratic triangle. Use the Newton Cotes rule (8.1.8).* □

8.2 Linear triangles revisited

In Section 6.3.2 we have introduced linear triangles and computed the basis functions by direct solution of the system of equations (6.3.10). An alternative approach is to map the triangle in (x, y)-space on a so-called standard triangle in (ξ, η)-space, with coordinates (0,0), (1,0) and (0,1). Although there is no need to do so for the

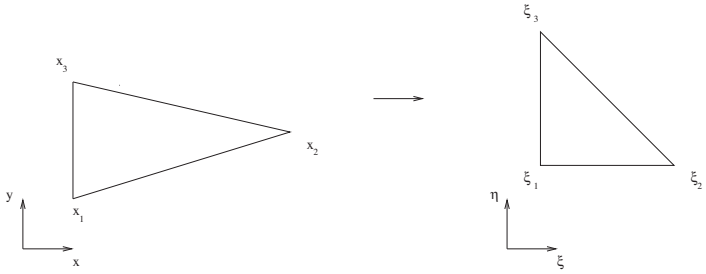

Figure 8.5: General triangle Standard triangle

linear triangle, we shall follow this approach here, since mapping is necessary for a number of elements that will be treated later on.
Consider the standard and general triangle in Figure 8.5. The basis functions on the standard triangle must satisfy $\phi_i(\xi_j, \eta_j) = \delta_{ij}$ and one immediately verifies that they are given by

$$\begin{aligned} \phi_1(\xi,\eta) &= 1-\xi-\eta, \\ \phi_2(\xi,\eta) &= \xi, \\ \phi_3(\xi,\eta) &= \eta. \end{aligned} \tag{8.2.1}$$

The linear transformation from the general triangle to the standard triangle is given by

$$\mathbf{x}_1 \rightarrow (0,0),\ \mathbf{x}_2 \rightarrow (1,0),\ \mathbf{x}_3 \rightarrow (0,1), \tag{8.2.2}$$

hence

$$\begin{aligned} x &= x_1 + (x_2 - x_1)\xi + (x_3 - x_1)\eta, \\ y &= y_1 + (y_2 - y_1)\xi + (y_3 - y_1)\eta. \end{aligned} \tag{8.2.3}$$

In order that the transformation is applicable it must be invertible. So the Jacobian of the transformation must be non-singular for each $\mathbf{x}$ in the triangle.

The Jacobian matrix $\mathbf{J}$ is defined by

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}. \tag{8.2.4}$$

Theorem 8.2.1 *The determinant of* $\mathbf{J}$ *is equal to:*

$$det(\mathbf{J}) = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1), \tag{8.2.5}$$

and this is precisely the parameter Δ in (6.3.11).
Integration of some function $f(x,y)$ over the general triangle can be simplified by transformation to the standard triangle:

$$\int_{e_{xy}} f(x,y)\, d\Omega_{xy} = \int_{e_{\xi\eta}} f(\xi,\eta)|det(J)|\, d\Omega_{\xi\eta}. \tag{8.2.6}$$

Exercise 8.2.1 *Prove that* $|\Delta|$ *is twice the area of the original triangle.*
Hint: use (8.2.6) *with* $f = 1$. □

Since $|\Delta|$ is only zero if the triangle has area zero, the transformation can not be singular.
To show how this transformation can be utilized to compute an element matrix or vector, we consider the simple example of the Poisson equation (6.2.1).
The element matrix has elements $s_{ij} = \int_e \nabla\varphi_i(\mathbf{x}) \cdot \nabla\varphi_j(\mathbf{x})\, d\Omega$. We transform this integral to an integral in the (ξ, η)-plane. Hence:

$$s_{ij} = \int_{e_{xy}} \nabla\varphi_i(\mathbf{x}) \cdot \nabla\varphi_j(\mathbf{x})\, dxdy = \int_{e_{\xi\eta}} \nabla\varphi_i \cdot \nabla\varphi_j |det(J)| d\xi\, d\eta. \tag{8.2.7}$$

Since $|det(J)|$ is constant the integral reduces to

$$s_{ij} = |det(J)| \int_e (\nabla\varphi_i \cdot \nabla\varphi_j) d\xi\, d\eta. \tag{8.2.8}$$

To compute the values of $\nabla\varphi_i$ we express the derivatives to x and y into derivatives of ξ and η:

$$\begin{aligned} \frac{\partial\varphi_k}{\partial x} &= \frac{\partial\varphi_k}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\varphi_k}{\partial\eta}\frac{\partial\eta}{\partial x}, \\ \frac{\partial\varphi_k}{\partial y} &= \frac{\partial\varphi_k}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial\varphi_k}{\partial\eta}\frac{\partial\eta}{\partial y}. \end{aligned} \tag{8.2.9}$$

To compute these derivatives, we need the values of $\frac{\partial\xi}{\partial x}$ and so on.

Theorem 8.2.2 *The matrix*

$$\begin{bmatrix} \frac{\partial\xi}{\partial x} & \frac{\partial\xi}{\partial y} \\ \frac{\partial\eta}{\partial x} & \frac{\partial\eta}{\partial y} \end{bmatrix}, \tag{8.2.10}$$

is the inverse of the matrix

$$\begin{bmatrix} \frac{\partial x}{\partial\xi} & \frac{\partial x}{\partial\eta} \\ \frac{\partial y}{\partial\xi} & \frac{\partial y}{\partial\eta} \end{bmatrix}. \tag{8.2.11}$$

Exercise 8.2.2 *Prove Theorem (8.2.2).* □

Exercise 8.2.3 *Show that*

$$\begin{bmatrix} \frac{\partial\xi}{\partial x} & \frac{\partial\xi}{\partial y} \\ \frac{\partial\eta}{\partial x} & \frac{\partial\eta}{\partial y} \end{bmatrix} = \frac{1}{det(J)} \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix} \tag{8.2.12}$$

□

Exercise 8.2.4 *Show from these formulas that*

$$\nabla\phi_i = \begin{bmatrix} \beta_i \\ \gamma_i \end{bmatrix}, \tag{8.2.13}$$

as defined in (6.3.13). □

Exercise 8.2.5 *Show that* s_{ij} *in* (8.2.8) *is given by* $s_{ij} = \frac{|\Delta|}{2}(\beta_i\beta_j + \gamma_i\gamma_j)$. □

Exercise 8.2.6 *Show that* $\int_{e_{\xi\eta}} \phi_i(\xi, \eta) d\xi d\eta = \frac{1}{6}$ *for* $i = 1, 2, 3$.
Use this result and (8.2.6) *to derive the Newton Cotes formula* (6.3.26). □

Since the jacobian as well as the derivatives for the linear basis functions are constant a lot of this work is superfluous. However, in the next sections we shall use this approach for more complex elements.

8.3 Quadrilaterals

Until now the derivation of the basis functions for triangles was relatively simple. Once we use quadrilaterals, however, things become more complicated. Due to the continuity requirement, it is not trivial what the basis functions look like.

Let us first start with the simple case of a rectangle with all sides in the coordinate directions. Such a rectangle may be considered as the product of two one-dimensional elements in x and y-direction respectively. The most simple element is the one with the 4 vertices as nodes and a bilinear approximation.

Theorem 8.3.1 *Consider the rectangle* $(x_1, x_2) \times (y_1, y_2)$ *and local node numbers 1 to 4, (Figure 8.6).*

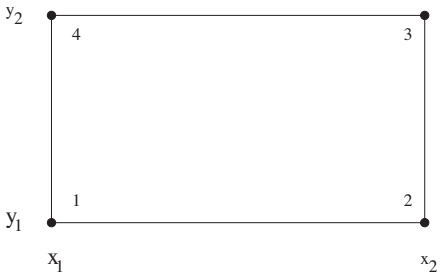

Figure 8.6: Nodes of rectangle.

The four basis functions are defined by:

$$\begin{aligned} \varphi_1(x,y) &= \lambda_1(x)\lambda_1(y), \\ \varphi_2(x,y) &= \lambda_2(x)\lambda_1(y), \\ \varphi_3(x,y) &= \lambda_2(x)\lambda_2(y), \\ \varphi_4(x,y) &= \lambda_1(x)\lambda_2(y), \end{aligned} \tag{8.3.1}$$

with $\lambda_i(x)$ *the one-dimensional basis functions in x-direction and* $\lambda_j(y)$ *in y-direction.*

Exercise 8.3.1 *Prove Theorem (8.3.1).*
Why are these basis functions continuous across element boundaries? □

One easily verifies that the basis functions $\varphi_i(x,y)$ in (8.3.1) have the shape

$$\varphi_i(x,y) = \alpha_i + \beta_i x + \gamma_i y + \delta_i xy. \tag{8.3.2}$$

Unfortunately basis functions of the shape (8.3.2) are not continuous for a general quadrilateral (Why?). Since it is not clear what the general shape of the basis functions must be, we have to use some special construction method.

The standard technique used in the literature is known under the name *isoparametric transformations*. The idea is as follows: one does not know what the basis functions look like for a general quadrilateral but for a square with sides in x and y-direction it is obvious. Therefore one transforms the general quadrilateral element in the x-y-plane with a coordinate transformation $(x,y) \rightarrow (\xi,\eta)$ to a *standard element* (the unit square) in the ξ-η-plane. Such a transformation is called isoparametric if it satisfies the following properties:

1. The nodes $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k$ are transformed to fixed points $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, ..., \boldsymbol{\xi}_k$, i.e. the points in the reference element are always the same.
2. Straight sides in the original element remain straight in the reference element.
3. If the basis functions in the transformed element are given by $\varphi_1(\mathbf{x})$, $\varphi_2(\mathbf{x})$, ... , $\varphi_k(\mathbf{x})$ then the inverse transformation $(\xi, \eta) \rightarrow (x, y)$ is given by

$$\mathbf{x} = \sum_{l=1}^{k} \mathbf{x}_l \varphi_l(\xi, \eta), \tag{8.3.3}$$

and the interpolation by

$$u(\mathbf{x}) = \sum_{l=1}^{k} u_l \varphi_l(\xi, \eta). \tag{8.3.4}$$

In other words we use the same elements for transformation and interpolation.

Note that the basis functions are only known explicitly in the reference element, to compute their values in the original element we have to do a back-transformation. In fact also only the back-transformation is given explicitly.

Figure 8.7 shows the transformation of the quadrilateral element to a unit square in (ξ, η) space.

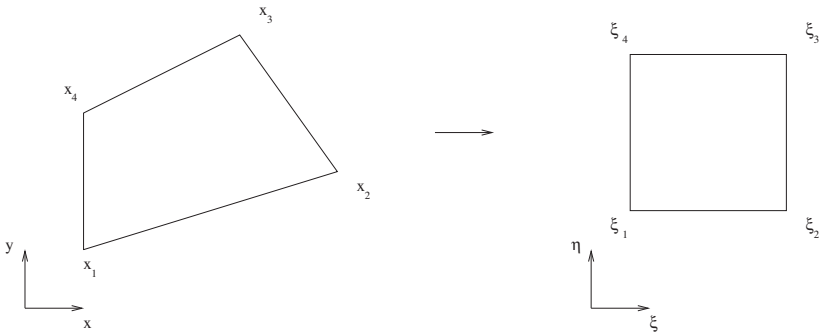

Figure 8.7: Transformation of quadrilateral to unit square.

In this case the isoparametric transformation is a *bilinear transformation*. The nodes $\mathbf{x}_i$ of the quadrilateral are transformed to the vertices of the unit square in the following way:

$$\mathbf{x}_1 \rightarrow (0,0),\ \mathbf{x}_2 \rightarrow (1,0),\ \mathbf{x}_3 \rightarrow (1,1),\ \mathbf{x}_4 \rightarrow (0,1). \tag{8.3.5}$$

The basis functions in the (ξ, η)-plane are bi-linear and defined by

$$\varphi_1 = (1-\xi)(1-\eta),\ \varphi_2 = \xi(1-\eta),\ \varphi_3 = \xi\eta,\ \varphi_4 = (1-\xi)\eta. \tag{8.3.6}$$

Note that the transformation (8.3.3) transforms straight sides of the reference element into straight sides of the quadrilateral in (x, y)-space. Moreover the function $u(\mathbf{x})$ defined by (8.3.4) reduces to a straight line on the sides of the quadrilateral (Why?). Hence continuity of the interpolation is satisfied.

In order that the transformation is applicable it must be invertible, in other words for each $\mathbf{x}$ in the quadrilateral we must have a unique $\boldsymbol{\xi}$. So the Jacobian of the transformation must be non-singular for each $\mathbf{x}$ in the quadrilateral.

Theorem 8.3.2 *The transformation (8.3.3) is given by*

$$\begin{aligned} x &= x_1 + (x_2 - x_1)\xi + (x_4 - x_1)\eta + (x_1 - x_2 + x_3 - x_4)\xi\eta, \\ y &= y_1 + (y_2 - y_1)\xi + (y_4 - y_1)\eta + (y_1 - y_2 + y_3 - y_4)\xi\eta. \end{aligned} \tag{8.3.7}$$

Theorem 8.3.3 *The determinant of* $\mathbf{J}$ *is equal to:*

$$det(\mathbf{J}) = (x_2 - x_1 + A_x\eta)(y_4 - y_1 + A_y\xi) - (x_4 - x_1 + A_x\xi)(y_2 - y_1 + A_y\eta), \tag{8.3.8}$$

with $A_x = x_1 - x_2 + x_3 - x_4$ *and* $A_y = y_1 - y_2 + y_3 - y_4$.

Exercise 8.3.2 *Prove Theorem (8.3.2).* □

Exercise 8.3.3 *Prove Theorem (8.3.3).* □

Theorem 8.3.4 *The transformation (8.3.4) is invertible if and only if all angles of the quadrilateral are less than* π*, i.e. the quadrilateral is convex.*

Proof

Since the terms of second degree cancel, d$et(\mathbf{J})$ is linear. In other words if the sign of d$et(\mathbf{J})$ is the same at each vertex, it is impossible that it is zero inside the element (Why?).

The value of the d$et(\mathbf{J})$ in point (0,0) is equal to $(x_2 - x_1)(y_4 - y_1) - (x_4 - x_1)(y_2 - y_1)$ which is equal to the outer product $(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_4 - \mathbf{x}_1)$.

Since $\mathbf{v}_1 \times \mathbf{v}_2 = ||\mathbf{v}_1||\ ||\mathbf{v}_2||\sin\varphi$, with φ the angle between the two vectors. This is positive if the angle φ (counter-clockwise) is less than π. So in point 1 we must have an angle less than π.

The same holds for all other points and corresponding angles.

In conclusion for a convex quadrilateral the transformation is regular, if one of the angles is larger than π the transformation is singular.

To show how this transformation can be utilized to compute an element matrix or vector, we consider the simple example of the Poisson equation (6.2.1).

The element matrix has elements $s_{ij} = \int_e \nabla\varphi_i(\mathbf{x}) \cdot \nabla\varphi_j(\mathbf{x})\ d\Omega$. Since the basis functions are only known in the reference element we have to transform this integral to an integral in the (ξ, η)-plane as given in (8.2.7). In general it is complicated to compute the integral (8.2.7) exactly so one uses a numerical integration rule.

Theorem 8.3.5 *The Newton Cotes rule corresponding to the reference element 8.3.5 is given by the two-dimensional equivalent of the Trapezoid rule:*

$$\int_0^1 \int_0^1 Int(\xi, \eta)\ d\xi d\eta = \frac{1}{4} \sum_{k=1}^{4} Int(\xi_k, \eta_k). \tag{8.3.9}$$

Exercise 8.3.4 *Prove Theorem (8.3.5).* □

Approximation of Equation (8.2.7) by the Newton Cotes rule (8.3.9) leads to

$$s_{ij} \approx \frac{1}{4} \sum_{k=1}^{4} (\nabla\varphi_i \cdot \nabla\varphi_j |det(J)|)(\xi_k, \eta_k). \tag{8.3.10}$$

The values of $|det(J)|$ in the integration points in the reference element can be computed immediately from Equation (8.3.8). To compute the values of $\nabla\varphi_i$ we have to express the derivatives to x and y into derivatives of ξ and η, since φ_i is only known in the (ξ, η)-plane, see (8.2.9).

Exercise 8.3.5 *Compute the values of $\frac{\partial\xi}{\partial x}$ in the integration points.* □

So the easiest way to approximate the Integrals (8.2.7) in the element matrix is to create a number of tables and combine these tables into the Sum (8.3.9).

In FEM programs the standard sequence to do it is the following:

1. Make a table of the values of ξ and η in the integration points. In this case this table is very simple, but if one uses Gauss integration the numbers are less trivial.

2. Make a table of the weights of the numerical integration. The weights include the factor $|det(J)|$, hence in this particular case we have $w_k = \frac{1}{4}|det(J(\mathbf{x}_k))|$.

3. Make a table of $\frac{\partial x}{\partial\xi}$ and so on in the integration points.

4. Use this table also to create $\nabla\varphi_i$.

Note that only the results of steps 2 and 4 are needed to compute the integrals (8.2.7).

The original weights of the numerical integration ($\frac{1}{4}$) are dimensionless, but due to the multiplication by $|det(J)|$, the weights get the dimension of an area.

Exercise 8.3.6 *Compute the element vector corresponding Poisson's equation (6.3.1), in the case of arbitrary quadrilateral. Use Newton Cotes integration.* □

8.4 Curved quadratic triangles

In Section 8.1 we have shown how basis functions for a straight quadratic triangle can be expressed in terms of linear basis functions. When the boundary of the domain is curved, it is necessary to approximate the boundary of the region by a piecewise quadratic polynomial in order to get the same order of accuracy one expects by using quadratic elements (See Section 8.7). So in practice we may have quadratic elements with a curved boundary.

The derivation of the basis functions for these elements is very similar to that of quadrilaterals. It is hard to find the general expression for the basis functions on the curved triangle and therefore we use an isoparametric transformation to map the curved triangle on a reference triangle with straight sides and vertices (0,0), (1,0) and (0,1). See Figure 8.8.

Let $\varphi_i(\mathbf{x})$ be the basis functions corresponding to the straight triangle (see Formula (8.1.7), with $\lambda_1 = 1 - \xi - \eta$, $\lambda_2 = \xi$ and $\lambda_3 = \eta$). The isoparametric transformation is defined by

$$\mathbf{x} = \sum_{k=1}^{6} \mathbf{x}_k \varphi_k(\xi, \eta). \tag{8.4.1}$$

Due to this transformation the boundaries of the triangle will be polynomials of degree two at most.

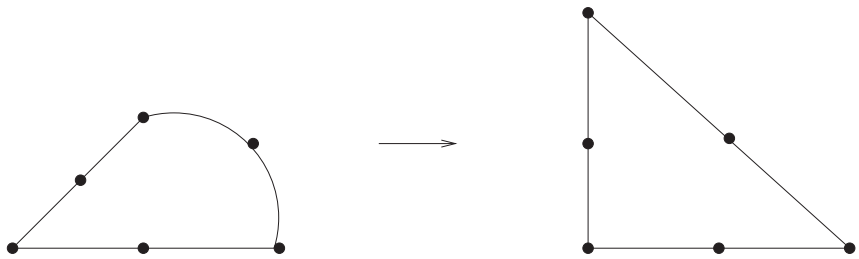

Figure 8.8: Transformation of curved triangle to reference element.

The transformation is non-singular if the determinant of the Jacobian has the same sign on the whole triangle. Unfortunately it is not easy to give a general rule about the restrictions an element has to satisfy in order that the transformation is regular. In general, angles of the triangle should not be too large ($< 135°$) and the "mid"points of the sides must be close to the actual middle of the edge. Furthermore the triangle may not be too curved. Usually it suffices to check the determinant in the integration points. If the sign of the determinant is the same in all integration points, one assumes that the mapping is invertible.

From Equation (8.4.1) it follows that

$$\frac{\partial x}{\partial \xi} = \sum_{k=1}^{6} \mathbf{x}_k \frac{\partial}{\partial \xi} \varphi_k(\xi, \eta) \,, \quad \frac{\partial x}{\partial \eta} = \sum_{k=1}^{6} \mathbf{x}_k \frac{\partial}{\partial \eta} \varphi_k(\xi, \eta) \,. \tag{8.4.2}$$

Exercise 8.4.1 *Show that the Newton Cotes integration rule for the quadratic reference element is given by*

$$\int_e Int(\xi, \eta)\, d\xi\, d\eta = \frac{1}{6}(Int(\frac{1}{2}, 0) + Int(\frac{1}{2}, \frac{1}{2}) + Int(0, \frac{1}{2})). \tag{8.4.3}$$

□

Exercise 8.4.2 *Compute the (ξ, η) derivatives of the basis functions in the Newton Cotes integration points of the reference element.* □

Exercise 8.4.3 *Show how the Jacobian matrix can be computed in the Newton Cotes integration points.* □

Exercise 8.4.4 *Indicate how the weights for the Newton Cotes integration in the original curved element can be computed.* □

Exercise 8.4.5 *Show how the x and y derivatives of the basis functions in the Newton Cotes integration points can be computed.* □

8.5 Application to the Stokes equations

The Stokes equations are derived from the Navier-Stokes equations (2.4.4a,2.4.4b) by removing the convective terms. These equations are only valid in case of small velocities. If the viscosity is a constant, the Stokes equations for incompressible

flow may be formulated as:

$$\begin{aligned} -\mu\Delta u + \frac{\partial p}{\partial x} &= f_x, \\ -\mu\Delta v + \frac{\partial p}{\partial y} &= f_y, \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0. \end{aligned} \tag{8.5.1}$$

$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix}$ is the velocity vector and p the pressure. In order to have a unique solution it is necessary to give boundary conditions for each velocity component on the complete boundary. One can prove that no explicit boundary condition for the pressure is required.

As an example we consider flow in straight channel, see Figure (8.9).

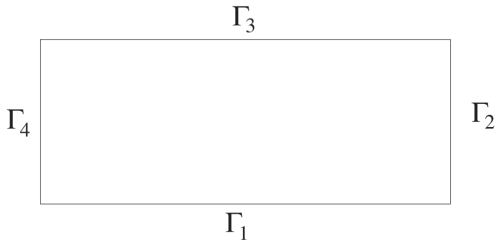

Figure 8.9: Straight channel with boundaries.

Boundary conditions are a given velocity vector on the inflow boundary Γ_4, no-slip boundary conditions on Γ_1 and Γ_3 and *outflow boundary conditions* on Γ_2. The mathematical formulation of the boundary conditions reads

$$\begin{array}{lll} u = 0, & v = 0 & \text{on } \Gamma_1, \\ p - \mu\frac{\partial u}{\partial n} = 0, & v = 0 & \text{on } \Gamma_2, \\ u = 0, & v = 0 & \text{on } \Gamma_3, \\ u = u(y), & v = 0 & \text{on } \Gamma_4. \end{array} \tag{8.5.2}$$

For reasons that go beyond the scope of this book, the polynomials to approximate the pressure are in general one degree lower than those of the velocity components. Both velocity components are approximated in the same way. In order to derive the weak formulation we multiply the first equation in (8.5.1) by a test function δu, the second one by δv and the last one by δp. These test functions belong to the same spaces as u, v and p respectively.

Exercise 8.5.1 *Show that the weak formulation of (8.5.1) with boundary conditions (8.5.2) can be written as*

$$\begin{aligned} \int_\Omega \mu\nabla u \cdot \nabla\delta u \, d\Omega - \int_\Omega p\frac{\partial \delta u}{\partial x} \, d\Omega &= \int_\Omega f_x \delta u \, d\Omega, \\ \int_\Omega \mu\nabla v \cdot \nabla\delta v \, d\Omega - \int_\Omega p\frac{\partial \delta v}{\partial y} \, d\Omega &= \int_\Omega f_y \delta v \, d\Omega, \\ \int_\Omega (\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})\delta p \, d\Omega &= 0. \end{aligned} \tag{8.5.3}$$

What are the boundary conditions for δu, δv and δp? □

Exercise 8.5.2 *Give the continuity requirements for u v, δu and δv.*

Are there any restrictions for p and δp? □

One of the most simple elements found in the literature for the (Navier-)Stokes equations for incompressible flow is the *bi-linear velocity, constant pressure quadrilateral*. The velocity components are approximated by bi-linear polynomials in the way described in Section (8.3). The pressure is approximated by a constant per element.

Exercise 8.5.3 *Show that the Galerkin equations corresponding to the weak formulation (8.5.3) are given by*

$$\begin{aligned}
\sum_{j=1}^{n} u_j \int_\Omega \mu \nabla\varphi_j \cdot \nabla\varphi_i \, d\Omega - \sum_{j=1}^{m} p_j \int_\Omega \psi_j \frac{\partial\varphi_i}{\partial x} \, d\Omega &= \int_\Omega f_x \varphi_i \, d\Omega \;\; (i = 1, ..., n_u), \\
\sum_{j=1}^{n} v_j \int_\Omega \mu \nabla\varphi_j \cdot \nabla\varphi_i \, d\Omega - \sum_{j=1}^{m} p_j \int_\Omega \psi_j \frac{\partial\varphi_i}{\partial y} \, d\Omega &= \int_\Omega f_y \varphi_i \, d\Omega \;\; (i = 1, ..., n_v), \\
\sum_{j=1}^{n} u_j \int_\Omega \frac{\partial\varphi_j}{\partial x} \psi_i \, d\Omega + \sum_{j=1}^{n} v_j \int_\Omega \frac{\partial\varphi_j}{\partial y} \psi_i \, d\Omega &= 0 \;\; (i = 1, ..., m). \qquad (8.5.4)
\end{aligned}$$

Here n is the number of u or v velocity unknowns and m the number of pressure unknowns. n_u is the number of non-prescribed u velocity unknowns and n_v is the number of non-prescribed v velocity unknowns. □

Exercise 8.5.4 *What is the number of u-velocity unknowns per element? And the number of p unknowns?*

Give an expression for the pressure basis functions ψ_i per element. □

Exercise 8.5.5 *Suppose we order the unknowns per element in the sequence u_1, u_2, ... ,v_1, v_2, ... , p_1, ...*
Then the element matrix can be split into 9 parts according to:

$$\mathbf{S}^{e_k} = \begin{bmatrix} \mathbf{S}_{uu} & \mathbf{S}_{uv} & \mathbf{S}_{up} \\ \mathbf{S}_{vu} & \mathbf{S}_{vv} & \mathbf{S}_{vp} \\ \mathbf{S}_{pu} & \mathbf{S}_{pv} & \mathbf{S}_{pp} \end{bmatrix} \qquad (8.5.5)$$

Give the sizes of the subelement matrices.
Give the formulas of the elements of each subelement matrix in integral form. □

8.6 Circle symmetry

Most real world problems are three-dimensional. But solving 3D problems is laborious and time-consuming and on top of that post-processing, like graphically representing results is much more difficult for 3D- than for 2D-problems. Therefore one often tries to reduce a problem from three to two dimensions by assuming certain symmetries in the solution. One such possibility is the assumption that a solution does not depend on a certain coordinate. (Translation symmetry). Another possibility is, if a region is cylindrically shaped, to assume that the solution has circular symmetry. For that to be possible all data, including boundary conditions, must have circular symmetry. In that case it is possible to reduce the

three-dimensional problem to a two-dimensional one by introducing cylinder coordinates (r, θ, z), defined by

$$\begin{aligned} x &= r\cos\theta, \\ y &= r\sin\theta, \\ z &= z. \end{aligned} \tag{8.6.1}$$

In finite difference methods, the standard approach is to transform the PDE in (x, y, z) to a PDE in (r, z).

Exercise 8.6.1 *Show that Poisson's equation $-\Delta u = f$ in cylinder coordinates (r, z) can be written as:*

$$\frac{\partial^2 u}{\partial r^2} + \frac{\partial^2 u}{\partial z^2} + \frac{1}{r}\frac{\partial u}{\partial r} = f. \tag{8.6.2}$$

Suppose that $r = 0$ is part of the region. What is the boundary condition in $r = 0$? Why do we need a boundary condition in $r = 0$? □

Of course in FEM it is also possible to solve the transformed Poisson equation (8.6.2) with boundary condition as defined in Exercise 8.6.1. However, in that case we have to take care of the singularity in $r = 0$. Also we have to take the artificial boundary condition in $r = 0$ into account.

A more natural approach is the following. We derive the weak formulation and Galerkin equations for the original 3D problem. This does not contain any singularity in $r = 0$, nor do we need an artificial boundary condition. Afterwards we take into account that the solution is constant in θ-direction by assuming that the basis functions are constant in that direction. So integration in the θ-direction is trivial. This approach also leads to a 2D formulation, but without special requirements.

Exercise 8.6.2 *Let u satisfy Poisson's equation $-\Delta u = f$ on a 3D circle-symmetric region. Let $u = g$ at the boundary with g independent of θ. Show that the Galerkin equations corresponding to this problem are given by*

$$\sum_{j=1}^{n} u_j \int_{\Omega} \nabla\varphi_i \cdot \nabla\varphi_j \, d\Omega = \int_{\Omega} f\varphi_i \, d\Omega. \tag{8.6.3}$$

□

In order to compute the element matrix and vector on this element we have to transform the Expression (8.6.3) from (x, y, z)-coordinates to (r, z)-coordinates.

Exercise 8.6.3 *Show that the determinant of the Jacobian matrix of the transformation is equal to r.* □

Exercise 8.6.4 *Show that the elements of the element matrix are given by*

$$s_{ij} = 2\pi \int_{e_{rz}} \left(\frac{\partial\varphi_i}{\partial r}\frac{\partial\varphi_j}{\partial r} + \frac{\partial\varphi_i}{\partial z}\frac{\partial\varphi_j}{\partial z}\right) r \, dr dz. \tag{8.6.4}$$

Hint: transform $\frac{\partial\varphi_i}{\partial x}$, $\frac{\partial\varphi_i}{\partial y}$ and $\frac{\partial\varphi_i}{\partial z}$ into $\frac{\partial\varphi_i}{\partial r}$ and $\frac{\partial\varphi_i}{\partial z}$, using the transformation (8.6.1) □

8.7 Theoretical remarks

In the mathematical parts of Chapters 5 to 7 it has been shown that the error made by the finite element method (measured in energy norm), is smaller than a constant C times the error we would have if we interpolated the exact solution by the same type of approximation:

$$\|u - u_h\|_L \leq C\|u - u_I\|_L, \tag{8.7.1}$$

with u the exact solution, u_h the FEM-solution, u_I the interpolation of u and $\| \ \|_L$ the energy norm. In a minimization problem $C = 1$, proving that the FEM solution is the best approximation in the energy norm. In a general weak formulation, the constant C is related to the ratio of the symmetric and the anti-symmetric part of the operator L. For example in a convection-dominated flow C is large, whereas for diffusion dominated problems C is close to 1.

From (8.7.1) it follows that if we want to estimate the FEM error it is necessary to estimate the interpolation error. Suppose we use an approximation by k^{th} degree polynomials. Then one can prove, under certain (geometrical) conditions, that the error in L^2 norm is of the order h^{k+1}. So for a linear approximation, the interpolation error in L^2 norm is $O(h^2)$.

If the order of the differential equation is $2m$ (Poisson $m = 1$, bi-harmonic $m = 2$), then the energy norm contains derivatives of order m. In general, for each derivative, the interpolation error is reduced by an order 1. So the interpolation error in energy norm of an $2m$-th order operator using k-th degrees polynomials is of order h^{k+1-m}. This means that also the FEM error in energy norm is of order h^{k+1-m}. Under certain conditions one can prove that this error in L^2 norm is again of order h^{k+1}, which is comparable to the interpolation error.

In the above text we mentioned "under certain (geometrical) conditions". One can prove that the elements must satisfy some requirements in order that these estimates can be applied. It goes beyond the scope of this book to give exact formulations, but the following rules are generally valid.

- In case of triangles, the largest angle must not be too close to π. A practical bound is that all angles must be smaller than 135°. A large angle gives a bad approximation of the derivatives and thus a large error. Sharp corners, on the other hand, do not pose problems.
- In case of quadrilaterals, it is necessary that the mapping to a standard square is invertible. In fact the size of the Jacobian must be not too large. In practice this also implies that corners must not be much larger than 135°.

All before mentioned estimates are valid in case of exact integration and under the condition that the whole region is completely covered by elements. In practice, however, one uses numerical integration and the boundary will be approximated by polynomials. It is very difficult to make an estimation of the effect of these approximations. Under certain special conditions, one can prove some theorems about this matter ([11]), but these proofs are very complicated.

In general one can state the following:

If we use polynomials of degree k to approximate the solution, it is also necessary to approximate the boundary of the region by polynomials of the same order. Otherwise the order of accuracy is reduced to lower order. So with linear elements, it is sufficient to approximate the boundary of the region by piecewise linear polynomials, i.e. straight lines. But with quadratic elements, it is necessary to use a quadratic approximation of the boundary.

With respect to the numerical integration, the rules are more complicated.

In Cartesian coordinates it is necessary that the numerical integration is exact for polynomials of degree $2k - 2m$, otherwise the accuracy of the global approximation is reduced. Hence for a second order differential equation ($m = 1$), we have the following requirements:

- Linear elements ($k = 1$), the integration must be exact for constant polynomials.
- Quadratic elements ($k = 2$), the integration must be exact for quadratic polynomials.
- Third degree elements ($k = 3$), the integration must be exact for fourth order polynomials.

For that reason Newton Cotes integration can only be applied to linear and quadratic elements.

In other types of coordinate systems, like for example cylindrical coordinates, the situation is more complex. Actually the rules above remain valid if we adapt the integration rules to reflect the type of coordinates. For example if we incorporate a factor r in the integration rules for cylindrical coordinates then the same type of rules apply. If we do not adapt the integration rules, the numerical integration must be exact for polynomials of degree $2k - 2m + 1$. So in that case the Newton Cotes rule can only be applied to linear elements.

8.8 Fourth order problems

Until now we have limited ourselves to second order problems. This is not without a reason, fourth order problems are much more difficult to solve. An extended description of the various methods to handle this kind of problems is beyond the scope of this book. Nevertheless we shall show some basic techniques used in the literature. These methods will be shown using a very simple example, the clamped beam.

8.8.1 The clamped beam

Consider the beam sketched in Figure 8.10, clamped in both ends 0 and l.

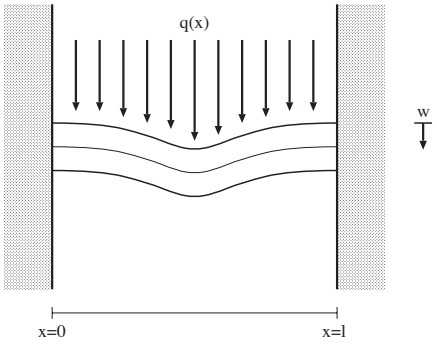

Figure 8.10: Clamped beam with load q.

On the beam we have a given load $q(x)$. The transverse displacement w of the neutral line with respect to the equilibrium satisfies the differential equation:

$$EI\frac{d^4w}{dx^4} = q, \tag{8.8.1}$$

with EI the *flexural rigidity* of the beam. Boundary conditions are

$$w(0) = w(l) = 0, \quad \frac{dw}{dx}(0) = \frac{dw}{dx}(l) = 0. \tag{8.8.2}$$

Note that for this fourth order problem we have to give 2 boundary conditions on the whole boundary.
The transverse displacement w is the solution of the minimization problem

$$\min_w \int_0^l \frac{1}{2}EI(\frac{d^2w}{dx^2})^2 - qw\,dx. \tag{8.8.3}$$

Exercise 8.8.1 *Prove (8.8.3).* □

In order to solve the minimization problem (8.8.3), Ritz's method may be applied and we use the FEM to construct the basis functions $\varphi_i(x)$.
Suppose that we approximate w by a linear combination of basis functions $\varphi_i(x)$

$$w_h(x) = \sum_{j=1}^{n} \alpha_j \varphi_j(x). \tag{8.8.4}$$

Then the Ritz equations corresponding to 8.8.3 are given by

$$\sum_{j=1}^{n} \alpha_j \int_0^l EI\frac{d^2\varphi_j}{dx^2}\frac{d^2\varphi_i}{dx^2}\,dx = \int_0^l q\varphi_i\,dx. \tag{8.8.5}$$

Exercise 8.8.2 *Prove (8.8.5).* □

The basis functions φ_i must satisfy the boundary conditions (8.8.2), but also they have to be continuously differentiable, i.e. $\varphi_i(x) \in C^1(0,l)$. Why?

The simplest element in the FEM, that can be used to construct the basis functions is a 2 node element. In order to ensure the continuity of the derivatives over the element boundaries, *Hermitian interpolation* is applied. In each node i we introduce two unknowns w_i and $(w_x)_i$ and write the interpolation per element as

$$w_h(x) = \sum_{j=1}^{2} w_j\psi_{j0}(x) + \sum_{j=1}^{2}(w_x)_j\psi_{j1}(x), \tag{8.8.6}$$

with $\psi_{j0}(x)$ and $\psi_{j1}(x)$ third degree polynomials, satisfying

$$\psi_{j0}(x_i) = \delta_{ij}, \;\; \frac{d\psi_{j0}}{dx}(x_i) = 0, \;\; \psi_{j1}(x_i) = 0, \;\; \frac{d\psi_{j1}}{dx}(x_i) = \delta_{ij} \tag{8.8.7}$$

So the parameters α_j in (8.8.4) are either w_j or $(w_x)_j$.

Exercise 8.8.3 *Express the basis functions $\psi_{j0}(x)$ and $\psi_{j1}(x)$ in terms of the linear basis functions $\lambda_i(x)$.* □

Exercise 8.8.4 *Compute with the basis functions ψ_{j0} and ψ_{j1} the element matrix corresponding to (8.8.5) for this element.*
What is the size of the element matrix?
Suppose that q is a constant. Compute the element vector. □

In order to get continuity of the first derivatives, we had to introduce the first derivatives of the unknown as parameters. There are in fact alternatives but they are more complicated.

If we extend the example to 2D or even 3D problems, construction of basis functions satisfying continuity of the first derivatives is much more cumbersome. For example if we want a complete polynomial (i.e. a polynomial containing all terms until a certain degree) on a triangle satisfying continuity of the first derivatives, it is necessary to use a fifth degree polynomial with 21 parameters. The reason is that we need continuity both in tangential and normal direction. If we drop the demand for a complete polynomial, the degree may be lowered a bit, but still the element is very complicated.

For that reason one can find many attempts in the literature to get rid of the C^1 requirement. In fact one can find two major solution strategies:

- violate the C^1 continuity requirement and carry out the assembly procedure as if there is no problem, in other words use non-conforming elements.
- Use a mixed formulation

The non-conforming approach is wrong in general, but can be justified if special conditions are met. These conditions are formulated in terms of the *Patch test* of Irons ([21]). It is not simple to apply this condition for general PDEs.

The mixed formulation is easier to generalize and we shall do the beam problem as a simple example.

8.8.2 A simple example of the mixed approach

The idea of the mixed approach is simple. We formulate the minimization problem (8.8.3) in terms of two variables w and β ($=\frac{dw}{dx}$), instead of one (w). These variables are considered to be independent, but we relate them by the constraint

$$\beta - \frac{dw}{dx} = 0. \tag{8.8.8}$$

So we can rewrite the minimization problem as

$$\min_{w,\beta} \int_0^l \frac{1}{2} EI(\frac{d\beta}{dx})^2 - qw \, dx, \tag{8.8.9}$$

under the constraint (8.8.8). A well known technique from the theory of minimization with constraints is the *penalty approach*. We multiply the squared of the constraint by a number $\frac{\alpha}{2}$ and add this to the minimization problem:

$$\min_{w,\beta} \int_0^l \frac{1}{2} EI(\frac{d\beta}{dx})^2 + \frac{\alpha}{2}(\beta - \frac{dw}{dx})^2 - qw \, dx, \tag{8.8.10}$$

If α is large the minimum is reached for $\beta - \frac{dw}{dx}$ small. So the constraint is satisfied approximately. The final step is to solve the penalized minimization problem by the FEM. Both β and w are approximated by linear polynomials per element

$$\beta_h = \sum_{j=1}^{n} \beta_j \varphi_j(x), \quad w_h = \sum_{j=1}^{n} w_j \varphi_j(x). \tag{8.8.11}$$

The boundary conditions can be formulated in terms of w and β.

Exercise 8.8.5 *Show that the Ritz equations corresponding to (8.8.10) using the approximation (8.8.11) are given by*

$$\sum_{j=1}^{n} w_j \int_0^l \alpha \frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx}\,dx - \sum_{j=1}^{n} \beta_j \int_0^l \alpha\varphi_j \frac{d\varphi_i}{dx}\,dx = \int_0^l q\varphi_i\,dx, \quad i = 1,\ldots,n, \tag{8.8.12}$$

$$-\sum_{j=1}^{n} w_j \int_0^l \alpha\varphi_i \frac{d\varphi_j}{dx}\,dx + \sum_{j=1}^{n} \beta_j \int_0^l (EI\frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx} + \alpha\varphi_i\varphi_j)\,dx = 0, \quad i = 1,\ldots,n. \tag{8.8.13}$$

Is the stiffness matrix symmetrical? □

Exercise 8.8.6 *Order the unknowns in the sequence $w_1, w_2, \beta_1, \beta_2$. Compute the element matrix and vector under the condition that E, I, α and q are constant.* □

The formulation given above is only one of the many different mixed formulations one can find in the literature. It is just a demonstration, of how by introducing new variables, one can effectively reduce the order of the equations. In practice, however, one must be careful with this kind of approximations, since in many cases the various unknowns must approximated with different types of polynomial. A discussion of this subject goes beyond the scope of this book.

8.9 Summary of Chapter 8

In Chapters 6 and 7 linear interpolation functions were used. In this chapter it has been demonstrated how the basis functions for higher order elements can be derived.
Also quadrilaterals, which require a special approach to ensure continuity, have been treated. For quadrilaterals as well as curved elements, the standard technique is mapping an arbitrary element onto a standard (reference) element. All integrals are evaluated on this reference element.

Fourth order problems have been introduced for a very simple 1D example, just to show the difficulties and possible solutions.

Chapter 9

Solution of large systems of equations

Objectives

In this chapter we will focus on the solution of systems of linear equations resulting from the discretization of PDE's. The corresponding matrices are generally large and sparse. There are two classes of methods to solve such systems of equations: direct and iterative methods. All direct solvers are variants of Gaussian elimination.
We shall first deal with the direct solvers. Special storage techniques to reduce the amount of memory like band methods and profile methods are treated. Using a special renumbering technique, the size of the matrix can be made semi-optimal.
Direct methods sometimes fall short. This happens mostly, when the number of unknowns becomes excessive like in large 3D problems. Even with optimal numbering the fill-in becomes huge and the L and U matrices no longer fit into memory. That is where iterative methods enter the picture. Historically there has been a trade off between memory and computing time. This certainly is true for the earlier iteration methods like *Jacobi*, *Gauss-Seidel* and *Successive Overrelaxation (SOR))*. But in the early seventies two very different methods became popular fast: the *Conjugate Gradient Method (CG)*, later generalized to *Bi-CGStab* and the *Multigrid Method*. These methods were so successful in fact, that they challenged the direct methods in their own back yard: computation time. The Multigrid method even achieves theoretically the best result possible: the number of computations increases *linearly* with the number of unknowns.

We start with classical iteration methods and derive convergence and stop criteria for them. Then we turn our attention to Krylov space methods like CG and Bi-CGStab. We shall see, that the success of these methods much depends on the choice of *preconditioner* and that classical iteration methods may serve as a preconditioner. Finally we shall describe a very powerful preconditioner: Incomplete LU decomposition.

Finally we shall turn our attention to Multigrid. There too we shall see that the success of the method will depend on various strategic choices. The preconditioners are called *smoothers* in Multigrid.

Often Multigrid and CG like methods are presented as competitors. There is no need for that: Multigrid is an excellent preconditioner for Bi-CGStab.

9.1 Direct methods

9.1.1 Introduction

The discretization of an elliptic PDE leads always to a system of (non-)linear equations. As will be seen in Section 9.7, non-linear systems are solved by a series of linear problems with the same structure. So a fast solution of systems of linear equations is of great importance for the discretization of PDE's.
The matrices resulting from discretization are in general large and sparse. If a suitable numbering is applied, these matrices have also a band structure. Matrices for which only elements within the band are stored are referred to as *band matrices*. Matrices for which only the non-zero elements are stored are known as *compact matrices*. They require extra information about the position of the non-zero elements.

Exercise 9.1.1 *Assume that we discretize the Poisson equation with Dirichlet boundary conditions on a rectangular domain by a central finite difference discretization. Suppose that the number of nodes in each coordinate direction is equal to $n + 2$.*
Show that the size of the discretization matrix is equal to $n \times n$ in $\mathbb{R}^1$, $n^2 \times n^2$ in $\mathbb{R}^2$ and $n^3 \times n^3$ in $\mathbb{R}^3$. □

Exercise 9.1.2 *Suppose that we use a natural numbering.*
Show that the band width of the matrices in Exercise 9.1.1 is equal to 3 in $\mathbb{R}^1$, $2n + 1$ in $\mathbb{R}^2$ and $2n^2 + 1$ in $\mathbb{R}^3$. □

Exercise 9.1.3 *Show that the number of non-zero elements per row for the matrices in Exercise 9.1.1 is equal to 3 in $\mathbb{R}^1$, 5 in $\mathbb{R}^2$ and 7 in $\mathbb{R}^3$.* □

Exercise 9.1.4 *Compute the number of entries that we have to store for the matrices in Exercise 9.1.1 in case of a full matrix, a band matrix and a compact matrix for n = 10, 100 and 1000 respectively. How many bytes is this, if a real takes 8 bytes?* □

The previous exercises show that for $n = 10$ the band matrices in all three dimensions can be stored in the internal memory of the computer, but that for $n = 100$ only the 1D and 2D matrices fit into memory. In the case of $n = 1000$ even the 2D band matrix is too large and the 3D compact matrix fits only in very large 64 bits computers.
In the next sections we shall treat some basis techniques for direct and linear solvers.

9.1.2 Gaussian elimination

As mentioned in Section 9.1.1 all direct solvers are variants of Gaussian elimination. In numerical applications, Gaussian elimination is carried out in the form of a *LU-decomposition*. In case of a *band matrix*, which arises if we apply a discretization technique on a structured (rectangular) grid, all elements outside the band are zero. This property is kept after Gaussian elimination, provided rows and columns are not interchanged. For unstructured meshes, like in FEM, a more sophisticated approach is necessary: the *profile method*. A good numbering of the equations is essential to keep the number of elements to be stored as low as possible.
First we start by explaining the LU-decomposition, then band methods are treated, followed by profile methods. Finally we make a few remarks about automatic optimal renumbering techniques.
For completeness we give a short description of the Gaussian elimination process.

Consider the system of linear equations:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2, \\ \vdots \qquad\qquad & \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n, \end{aligned} \tag{9.1.1}$$

or in matrix vector notation

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{9.1.2}$$

Gaussian elimination transforms system (9.1.2) into an upper triangular matrix by elementary row operations.
Consider the matrix $\mathbf{A}^0$ extended with the right-hand side $\mathbf{b}$.

$$\mathbf{A}^{(0)} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} & b_n \end{pmatrix}. \tag{9.1.3}$$

Subtract the first row multiplied by a suitable constant from the other rows, such that the first column becomes zero from row two. Define:

$$\begin{aligned} m_{j1} &= \tfrac{a_{j1}}{a_{11}} \qquad\qquad j = 2, 3, \ldots n. \\ a_{jk}^{(1)} &= a_{jk} - m_{j1}a_{1k} \quad k = 1, \ldots, n \\ \text{and} \quad b_j^{(1)} &= b_j - m_{j1}b_1. \end{aligned} \tag{9.1.4}$$

One easily verifies that $a_{j1}^{(1)} = 0$, $j = 2, 3, \ldots, n$. This produces a new extended matrix

$$\mathbf{A}^{(1)} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \ldots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & a_{32}^{(1)} & a_{33}^{(1)} & \ldots & a_{3n}^{(1)} & b_3^{(1)} \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & a_{n3}^{(1)} & \ldots & a_{nn}^{(1)} & b_n^{(1)} \end{pmatrix}. \tag{9.1.5}$$

It can be verified easily that the system of equations, described by this new extended matrix has the same solution as the original system 9.1.2. Now subtract the second row times a constant from the next rows such that the second column becomes zero from row number 2. Hence:

$$\begin{aligned} m_{j2} &= \tfrac{a_{j2}^{(1)}}{a_{22}^{(1)}} \qquad\qquad j = 3, 4, \ldots n. \\ a_{jk}^{(2)} &= a_{jk}^{(1)} - m_{j2}a_{2k}^{(1)} \quad k = 2, \ldots, n \\ \text{and} \quad b_j^{(2)} &= b_j^{(1)} - m_{j2}b_2^{(1)}. \end{aligned} \tag{9.1.6}$$

This gives

$$\mathbf{A}^{(2)} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \ldots & a_{2n}^{(1)} & b_2^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \ldots & a_{3n}^{(2)} & b_3^{(2)} \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \ldots & a_{nn}^{(2)} & b_n^{(2)} \end{pmatrix}. \tag{9.1.7}$$

The i^{th} step of the iteration:

$$\begin{aligned} m_{ji} &= \frac{a_{ji}^{(i-1)}}{a_{ii}^{(i-1)}} && j = i+1, i+2, \ldots n. \\ a_{jk}^{(i)} &= a_{jk}^{(i-1)} - m_{ji} a_{ik}^{(i-1)} && k = i, i+1, \ldots, n \\ \text{and} \quad b_j^{(i)} &= b_j^{(i-1)} - m_{ji} b_i^{(i-1)}. \end{aligned} \tag{9.1.8}$$

If we proceed this process until $i = n-1$, we get the matrix

$$\mathbf{A}^{(n-1)} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \ldots & a_{2n}^{(1)} & b_2^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \ldots & a_{3n}^{(2)} & b_3^{(2)} \\ \vdots & \vdots & 0 & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & a_{nn}^{(n-1)} & b_n^{(n-1)} \end{pmatrix}. \tag{9.1.9}$$

This is an upper triangular system. The solution can be determined immediately by back substitution. The quantities m_{ji} are called *multiplicators* and the quantities a_{ii}^{i-1} *pivots*. Since in the i^{th} step we have to subdivide by the pivot a_{ii}^{i-1} to compute the multiplicators m_{ji}, a necessary and sufficient condition for the application of the Gaussian elimination process is that none of the pivots a_{ii}^{i-1} is zero. Usually one interchanges rows and/or columns of the matrix to avoid zero (or small) pivots. However, in this book we shall not use this interchanging process called *pivoting* .

9.1.3 LU-decomposition

The Gaussian elimination process, transforms the matrix A with elements a_{ij} by elementary row operations into an upper triangular matrix U:

$$U = \begin{pmatrix} a_{11} & a_{12} & \ldots & \ldots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \ldots & a_{2n}^{(1)} \\ \vdots & 0 & a_{33}^{(2)} & \ldots & a_{3n}^{(2)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & a_{nn}^{(n-1)} \end{pmatrix}. \tag{9.1.10}$$

Besides, that we can store the multiplicators m_{ji}, which are used to create the zero lower triangle, into a lower triangular matrix L:

$$L = \begin{pmatrix} 1 & 0 & \ldots & \ldots & 0 \\ m_{21} & 1 & \ddots & & \ldots \\ \vdots & m_{32} & \ddots & \ddots & \ldots \\ \vdots & & \ddots & \ddots & 0 \\ m_{n1} & m_{n2} & \ldots & m_{n,n-1} & 1 \end{pmatrix}. \tag{9.1.11}$$

One easily verifies that

$$A = LU. \tag{9.1.12}$$

Once we have constructed the L and U matrix, the solution of $A\mathbf{x} = \mathbf{b}$ is straight forward. Substitution of (9.1.12) gives

$$LU\mathbf{x} = \mathbf{b}. \tag{9.1.13}$$

Define $U\mathbf{x} = \mathbf{y}$, then the sequence of solving is:

$$L\mathbf{y} = \mathbf{b}, \qquad U\mathbf{x} = \mathbf{y}. \tag{9.1.14}$$

As the form of the matrices L and U is triangular, these equations can be solved immediately.

Exercise 9.1.5 *Let L have elements l_{ij} with*

$$l_{ij} = \begin{cases} 0, & \text{if } i < j, \\ 1, & \text{if } i = j, \\ l_{ij}, & \text{if } i > j. \end{cases} \tag{9.1.15}$$

Show that the solution of $L\mathbf{y} = \mathbf{b}$ is given by

$$y_i = b_i - \sum_{k=1}^{i-1} l_{ik} y_k. \tag{9.1.16}$$

□

Exercise 9.1.6 *Let U have elements u_{ij} with*

$$u_{ij} = \begin{cases} 0, & \text{if } i > j, \\ 1, & \text{if } i = j, \\ u_{ij}, & \text{if } i < j. \end{cases} \tag{9.1.17}$$

Show that the solution of $U\mathbf{x} = \mathbf{y}$ is given by

$$x_i = y_i - \sum_{k=i}^{n} u_{ik} x_k. \tag{9.1.18}$$

In which sequence do we have to compute x_i? □

An alternative way to compute the matrices L and U is by direct substitution. Define the matrices L and U as in Exercises 9.1.5 and 9.1.6. Since $A = LU$ we have

$$a_{ij} = \sum_{k=1}^{n} l_{ik} u_{kj} = \sum_{k=1}^{\min(i,j)} l_{ik} u_{kj}. \quad \text{Why?} \tag{9.1.19}$$

Exercise 9.1.7 *Show that l_{ij} and u_{ij} are defined by the following relations*

$$\begin{aligned} u_{ii} &= a_{ii} - \sum_{k=1}^{i-1} l_{ik} u_{ki}, \\ u_{ij} &= (a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}) / u_{ii}, \\ l_{ij} &= (a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}) / u_{jj}. \end{aligned} \tag{9.1.20}$$

Give the sequence in which u_{ij} and l_{ij} can be computed. □

From this exercise we see that the LU-decomposition is unique iff the diagonal elements u_{ii} are non-zero. This is equivalent to having non-zero pivots during Gaussian elimination. In case the pivots are small it may be necessary to interchange rows and/or columns. However, in case of discretization methods, it is common practice to avoid pivoting, since this destroys the structure of the matrix. In many cases the discretization matrix has the property that pivoting is not required.

9.1.4 Band method

When we discretize a PDE on a rectangular structured grid, the matrix we get is always a band matrix, provided a natural numbering is chosen. The band width of such a matrix defines the amount of storage needed as well as the amount of work required to solve the system of equations.

Exercise 9.1.8 *Let the band width of the matrix* $\mathbf{A}$ *be equal to* $2b+1$, *i.e.* $a_{ij} = 0$ *if* $|i-j| > b$.
Prove by induction that $l_{ij} = 0$ *if* $i > j+b$ *and* $u_{ij} = 0$ *if* $j > i+b$. □

From Exercise 9.1.8 it follows that $\mathbf{L}$ and $\mathbf{U}$ are zero for elements outside the band. So it is indeed sufficient to store only the elements inside the band. Band matrices are stored column-wise, hence $(2b+1) \times n$ positions for non-symmetrical and $(b+1) \times n$ positions for symmetrical matrices are needed. So the storage of a typical non-symmetrical band matrix looks like:

$$\mathbf{A} = \begin{pmatrix} 0 & \dots & 0 & 0 & a_{11} & a_{12} & a_{13} & \dots & a_{1,1+b} \\ 0 & \dots & 0 & a_{21} & a_{22} & a_{23} & a_{24} & \dots & a_{2,2+b} \\ 0 & \dots & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & \dots & a_{3,3+b} \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ a_{n,n-b} & \dots & a_{n,n-2} & a_{n,n-1} & a_{nn} & 0 & 0 & \dots & 0 \end{pmatrix} \tag{9.1.21}$$

9.1.5 Profile method

Consider a matrix $\mathbf{A}$, which is the result of discretizing a PDE either by FEM, FVM or FDM. Let i be an arbitrary node in the grid with neighbors $j, k, l, \dots, m$ (Figure 9.1).

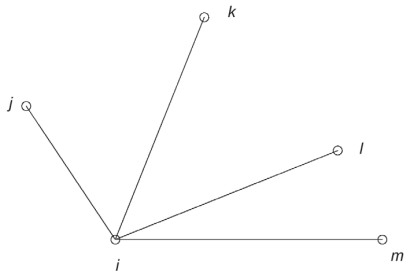

Figure 9.1: Nodal point i with neighbors.

Node i is connected to all of its neighbors, which implies that in general the elements $a_{ij}, a_{ik}, \dots, a_{im}$ are unequal to zero. If node n is not connected to i, then a_{in}=a_{ni}=0. Such elements in the matrix are called *essential zeros*. The fact that elements are essentially zero is a property of the grid and not of the specific PDE. Knowledge of essential zeros can be used to solve the system of equations efficiently. A typical example is the band method treated in Section 9.1.4. Another example is the *profile method*.

By *profile of a matrix* we mean the following:

Consider the i^{th} row. Let a_{ij} be the first essential non-zero element in this row counted from left to right. Hence j is the smallest column number in row i corresponding to an essential non-zero element. Then all elements $a_{ij}, a_{i,j+1}, \dots, a_{ii}$ belong to the *profile* or *envelope* of the matrix.

Consider on the other hand the i^{th} column of $\mathbf{A}$. Let j be the smallest row number in column j corresponding to an essential non-zero element. Then all elements $a_{ji}, a_{j,i+1}, \ldots, a_{ii}$ belong to the profile of the matrix. Mark that essential non-zero elements may be zero by coincidence. However, these elements are still considered to be non-zero. So actually a profile may be seen as a variable band. See Figure 9.2

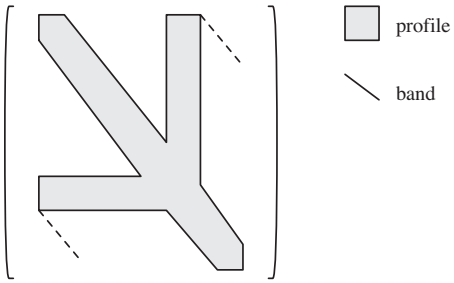

Figure 9.2: Example of a profile.

Exercise 9.1.9 *Show that the profile of a matrix arising from the discretization of a PDE is symmetrical.* □

Exercise 9.1.10 *Show that, if an LU-decomposition is applied, all elements outside the profile remain zero. Elements inside the profile of the L and U matrix will be non-zero in general .* □

From Exercise 9.1.10 it is clear that it is sufficient to store only those elements of the matrix that are *inside the profile*. Of course this requires a special storage scheme, otherwise the amount of memory needed is not better than for a band matrix. A standard storage method due to George [17] is the following one.

Let $\mathcal{L}$ be the lower triangle of the matrix $\mathbf{A}$ (without diagonal), $\mathcal{D}$ the diagonal and $\mathcal{U}$ the upper triangle. Hence

$$\mathbf{A} = \mathcal{L} + \mathcal{D} + \mathcal{U}. \tag{9.1.22}$$

The matrix $\mathbf{A}$ is stored in a one-dimensional array in the sequence

$$a_{11}, a_{21}, a_{22}, a_{12}, a_{31}, a_{32}, a_{33}, a_{23}, a_{13}, \ldots$$

where all the elements outside the profile are skipped. So the storage can be expressed as follows:
Start with diagonal element a_{11}.
Next store all elements of row 2 of $\mathcal{L}$ from left to right, followed by the diagonal element a_{22}, followed by all elements of column 2 of $\mathcal{U}$ from the diagonal to the top.
This process is repeated for all next rows and columns.
So the i^{th} row/column is stored as:

$$a_{i,p_i}, a_{i,p_i+1}, \ldots, a_{ii}, a_{i-1,i}, a_{i-2,i}, \ldots, a_{p_i,i},$$

with p_i the index of the first non-zero element in row i. In case of a symmetrical matrix, of course the upper triangle is not stored.
To keep track of the start of each new row, it is sufficient to store the position of the diagonal elements in the 1D array. This requires one extra integer array of length n (why?).

Exercise 9.1.11 *Show how to find an arbitrary element a_{ij} in the lower triangular matrix $\mathcal{L}$.* □

Exercise 9.1.12 *Show how to find an arbitrary element a_{ij} in the upper triangular matrix $\mathcal{U}$.* □

To perform an LU-decomposition on a profile matrix, it is necessary to apply an adapted method: the *profile method*. Special in this method is the sequence in which the elements of the LU-decomposition are computed. The sequence used is:

$$d_{11}, l_{21}, d_{22}, u_{12}, l_{31}, l_{32}, d_{33}, u_{23}, u_{13}, \ldots$$

So this is precisely the sequence of the matrix storage.

Exercise 9.1.13 *Give the formulas to compute* **L**, **D** *and* **U**, *utilizing the profile structure.* □

In the literature sometimes other names for the profile method are used, like *wave front method* and *frontal solution method*.
A simple example of a profile matrix is created by a one-dimensional problem with periodical boundary conditions as sketched in Figure 9.3. In this problem point i is connected to points $i-1$ and $i+1$ leading to a band width of 3. However, because of the periodical boundary conditions, point n and 1 have the same unknown and

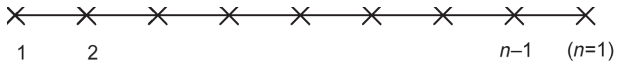

Figure 9.3: One-dimensional mesh, for problem with periodical boundary conditions.

point 1 is connected to both $n-1$ and 2. Point $n-1$ connected to $n-2$ and 1. The corresponding matrix gets the structure as sketched in Figure 9.4. The band width of this matrix is equal to $n-1$, which means that in case of a band storage, the matrix is full. The profile sketched in Figure 9.4b is much smaller.

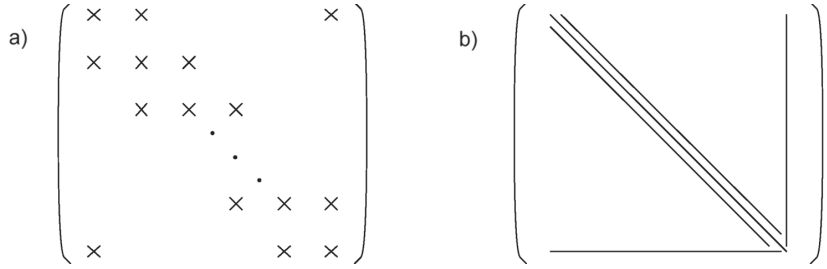

Figure 9.4: a) Non-zero pattern of one-dimensional problem with periodical boundary conditions, b) corresponding profile.

A good numbering may reduce the band width or the profile of the matrix considerably. In the next section we shall deal with a simple but effective renumbering algorithm.

9.1.6 Renumbering techniques

For finite element methods various renumbering algorithms have been constructed. Many of them are variants of the so-called Cuthill-McKee renumbering algorithm. The Cuthill-McKee [14] algorithm is a renumbering technique developed to reduce the band-width or envelope of a matrix. For an extended description see for example George and Liu [17]. Of course it is always possible to compute the optimal storage, but in general computation is so expensive that it takes more time than solving the system of equations. All renumbering techniques are therefore semi-optimal, in the sense that they try to optimize the storage, without performing to many operations.

The idea of Cuthill-McKee is that the local envelope of a matrix is minimal if all neighboring nodes have a number as close as possible to the node itself. Suppose we have a starting node or a set of starting nodes. This starting set has node numbers 1 to n. Then the idea is to give all direct neighbors of this starting set the node numbers from $n + 1$. This process is repeated until the complete set of nodes is exhausted. There are a number of variants of this algorithm all of which try to improve the envelope or profile, but the basic idea is the same.

Of course the difficult part of this process is how to find the starting set of nodes. This may be done automatically, which may be relatively difficult or by hand. In the last case one usually chooses a starting node (for example a corner node), or a starting curve.

Figure 9.5 shows the result of the first 9 steps of Cuthill-McKee in a rectangular grid. In the first step we start with the lower left point,indicated by a black circle. In the next step the three surrounding nodes are added (white circles). Each next set of nodes in the consecutive steps of the Cuthill-McKee algorithm, has been marked with a circle with different fill-in. In step 9 we arrive at a set of vertical nodes, which means that we are almost at an optimal numbering.

In *reversed Cuthill-McKee* we repeat the process in reversed order, starting with the last set of nodes found. In the example this will lead to a natural ordering.

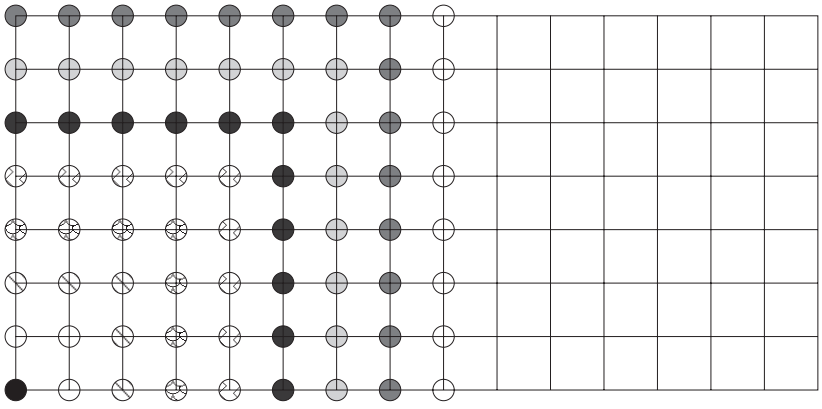

Figure 9.5: Example of the Cuthill-McKee algorithm.

9.2 Generic iterative process.

In this and the subsequent sections we shall consider *iterative methods*. We start out with a system of linear equations we want to solve: $A\mathbf{x} = \mathbf{b}$. Then we choose a *start value* $\mathbf{x}_0$ and we calculate the *residual* $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$. Now we have to improve $\mathbf{x}_0$ in some way, by adding a *correction* vector $\mathbf{c}_0$ to obtain a new estimate $\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{c}_0$. If we have been doing things right, the new residual $\mathbf{r}_1$ will in some way be smaller than $\mathbf{r}_0$. After that the process repeats itself, until the residual has become sufficiently small. Now the central question is of course: how to determine that correction step $\mathbf{c}_k$ to obtain $\mathbf{x}_{k+1}$ from $\mathbf{x}_k$?

Exercise 9.2.1 *Prove that when we take* $\mathbf{c}_k = A^{-1}\mathbf{r}_k$, $\mathbf{x}_{k+1}$ *solves* $A\mathbf{x} = \mathbf{b}$. □

So apparently $\mathbf{c}_k = A^{-1}\mathbf{r}_k$ would be the perfect choice, but unfortunately that is not easier to solve than our original system. However, it sets us on a trail. We may use an *approximation* $P^{-1}\mathbf{r}_k$ to $A^{-1}\mathbf{r}_k$ for $\mathbf{c}_k$. Such a matrix P is called a *preconditioner*. Different preconditioners generate different iterative methods.

9.3 Defect correction

9.3.1 Algorithm

The *defect correction* or *standard iteration* algorithm is the direct implementation of the above idea. It may be summarized in the following few lines of *pseudo code*

Defect correction algorithm

Presets: $\mathbf{x}_0 = 0; \mathbf{r}_0 = \mathbf{b}; k = 0$
while $\|\mathbf{r}_k\|_\infty > \varepsilon\|\mathbf{b}\|_\infty$ **do**
  $\mathbf{c}_k = P^{-1}\mathbf{r}_k$
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{c}_k$
  $\mathbf{r}_{k+1} = \mathbf{r}_k - A\mathbf{c}_k$
  $k = k + 1$
end while

Exercise 9.3.1 *Prove that the expression for* $\mathbf{r}_{k+1}$ *in this algorithm is equivalent to* $\mathbf{r}_{k+1} = \mathbf{b} - A\mathbf{x}_{k+1}$. □

Exercise 9.3.2 *Let* S *be an* $N \times N$ *matrix with* $s_{j,j+1} = 1, j = 1, \ldots, N-1, s_{jk} = 0$ *otherwise and let* I *be the identity matrix. We wish to solve* $A\mathbf{x} = \mathbf{f}$, *with* $A = 2I - S - S^T$ *and* $f_k = 1, k = 1, \ldots, N$. *Use defect correction with* $P^{-1} = \frac{1}{2}I$. *Use Matlab. Compare the number of iterations with N=10, N=100 and N=1000 to arrive at a residual with* $\|\mathbf{r}_k\|_\infty < 10^{-4}$. □

9.3.2 Convergence of defect correction

Let ξ be the solution to $A\mathbf{x} = \mathbf{b}$ and $\varepsilon_k = \xi - \mathbf{x}_k$ the error in the k-th iterate. We may combine various lines of Algorithm (9.3.1) to obtain

$$\mathbf{r}_{k+1} = (I - AP^{-1})\mathbf{r}_k, \tag{9.3.1}$$

$$\varepsilon_{k+1} = (I - P^{-1}A)\varepsilon_k. \tag{9.3.2}$$

Exercise 9.3.3 *Prove the two relations from Equation* (9.3.1) □

Exercise 9.3.4 *Prove that for any two matrices A, B that if*

$$\lim_{k\to\infty}(AB)^k = 0, \tag{9.3.3}$$

then

$$\lim_{k\to\infty}(BA)^k = 0. \tag{9.3.4}$$

□

Exercise 9.3.5 *Prove that* $\mathbf{r}_k = (I - AP^{-1})^k\mathbf{r}_0$. □

Exercise 9.3.6 *Prove that* $\varepsilon_k = (I - P^{-1}A)^k\varepsilon_0$. □

For the algorithm to converge, we must apparently have, that $\varepsilon_k \to 0$ or equivalently $\mathbf{r}_k \to 0$. This is expressed by the following theorem.

Theorem 9.3.1 *Let* $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq \lambda_N$ *be the eigenvalues of* $I - P^{-1}A$. *Then the defect correction method with matrix A and preconditioner* P^{-1} *converges if and only if* $|\lambda_1| < 1$.

Proof
We prove the theorem for non defect matrices. Assume $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N$ are the N linear independent eigenvectors of $I - P^{-1}A$, belonging to $\lambda_1, \lambda_2, \ldots, \lambda_N$. Now ε_0 can be written as a linear combination of those eigenvectors:

$$\boldsymbol{\varepsilon_0 = \sum_{j=1}^{N} \alpha_j \mathbf{v}_j,} \tag{9.3.5}$$

and since every multiplication with $I - P^{-1}A$ multiplies $\mathbf{v}_j$ with α_j we have:

$$\boldsymbol{\varepsilon_k = (I - P^{-1}A)^k\varepsilon_0 = \sum_{j=1}^{N} \alpha_j \lambda_j^k \mathbf{v}_j.} \tag{9.3.6}$$

Now since λ_1 is the largest eigenvalue in absolute value and $|\lambda_1| < 1$, each term in this sum will vanish eventually. Alternatively, if $\lambda_1 \geq 1$ the first term in the sum will never vanish. □

The value $|\lambda_1|$ is also called the *spectral radius* of the matrix $I - P^{-1}A$ and denoted $\rho(I - P^{-1}A)$.

9.3.3 Error estimate for defect correction

A closer look at Equation (9.3.6) reveals, that at the end of the day there will be only one term left in that sum: the first. Let us assume that λ_1 is an eigenvalue with multiplicity 1 and let us also assume there is no eigenvalue $-\lambda_1$. Then in the long run

$$\boldsymbol{\varepsilon_{k+1} \approx \lambda_1 \varepsilon_k.} \tag{9.3.7}$$

This enables us to estimate the error for a defect correction process.

Theorem 9.3.2 $\varepsilon_{k+1} \approx \frac{\lambda_1}{1-\lambda_1}(\mathbf{x}_{k+1} - \mathbf{x}_k)$.

Proof We subtract $\lambda_1\varepsilon_{k+1}$ from both sides of Equation (9.3.7) and note that $\varepsilon_k - \varepsilon_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_k$ to obtain

$$\boldsymbol{(1 - \lambda_1)\varepsilon_{k+1} \approx \lambda_1(\mathbf{x}_{k+1} - \mathbf{x}_k),} \tag{9.3.8}$$

and dividing both sides by $1 - \lambda_1$ gives the result.

The miracle is, that we can estimate the error in terms of things we know, viz $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$. Things we know?? What about λ_1 then? A little patience, we are coming to that.

9.3.4 Estimate of the spectral radius

We subtract Equation (9.3.7) with index k from that with index $k+1$ to obtain

$$\boldsymbol{\varepsilon_{k+1} - \varepsilon_k \approx \lambda_1(\varepsilon_k - \varepsilon_{k-1}).} \tag{9.3.9}$$

But $\varepsilon_k - \varepsilon_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_k$ and the above expression transforms into

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \lambda_1(\mathbf{x}_k - \mathbf{x}_{k-1}). \tag{9.3.10}$$

Or letting $\mathbf{d}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$, we find by applying standard least squares technique:

$$\lambda_1 \approx \frac{(\mathbf{d}_{k+1}, \mathbf{d}_k)}{(\mathbf{d}_k, \mathbf{d}_k)}. \tag{9.3.11}$$

Exercise 9.3.7 *Show that for large k*

$$\varepsilon_{k+N} = \lambda_1^N \varepsilon_k. \tag{9.3.12}$$

Infer from this that to gain one decimal digit you need

$$N = -\frac{1}{^{10}\log \lambda_1} \tag{9.3.13}$$

iterations. □

9.3.5 M-matrices

An important class of matrices for which preconditioners can be constructed such that the defect correction iteration converges are *M-matrices*. These often occur in the context of partial differential equations.

Definition 9.3.1 *A matrix A is called an M-matrix if*

1. $a_{jk} \leq 0, \quad$ *if* $j \neq k$,
2. $A^{-1} \geq 0$.

Exercise 9.3.8 *Show that a diagonally dominant Z-matrix is an M-matrix. Use the discrete maximum principle of Chapter 3.* □

Exercise 9.3.9 *Show that an M-matrix has nonnegative diagonal elements. Use a contradiction argument. Assume $a_{kk} < 0$ and consider $A\mathbf{e}_k$, with $\mathbf{e}_k$ the k-th unit vector.* □

Exercise 9.3.10 *Show that an upper triangular Z-matrix is an M-matrix if the diagonal elements are positive.* □

If $A^{-1} \geq 0$ (so in particular if A is an M matrix) we can construct a convergent preconditioner as follows. Split A into

$$A = P - Q, \tag{9.3.14}$$

in which P^{-1} and Q have only nonnegative entries, also denoted by $P^{-1} \geq 0, Q \geq 0$. Such a splitting is called *regular*. A famous theorem by Varga [42] guarantees that $\rho(P^{-1}Q) < 1$ for regular splittings. In other words, the defect correction process converges with P as preconditioner.

Exercise 9.3.11 *Let the matrix $B \geq 0$. Show that for any eigenvalue λ of B with corresponding eigenvector* $\mathbf{v}$

$$(I - B)|\mathbf{v}| \leq (1 - |\lambda|)|\mathbf{v}|, \tag{9.3.15}$$

with $|\mathbf{v}|_i = |v_i|$. Show, that if in addition $(I - B)^{-1} \geq 0$ that

$$|\mathbf{v}| \leq (1 - |\lambda|)(I - B)^{-1}|\mathbf{v}|, \tag{9.3.16}$$

and hence that $|\lambda| < 1$. □

Exercise 9.3.12 *Prove Varga's theorem. Use the result of the previous exercise.* □

9.4 Classical preconditioners

In this section we shall introduce various classical preconditioners that have been in wide use. Today they are not used much any longer in a context of defect correction. The reason for that we will see in our chapter on Krylov space methods and Multi Grid methods.

9.4.1 Jacobi

The oldest and arguably simplest iterative method known to man is that of Jacobi. Let's write $A = D - L - U$ in which D is the diagonal, L the lower triangular part and U the upper triangular part of the matrix A. For *Jacobi's method* we take $P = D$. The iteration matrix becomes:

$$M = I - P^{-1}A = I - D^{-1}(D - L - U) = D^{-1}(L + U). \tag{9.4.1}$$

Exercise 9.4.1 *Show, that Jacobi's method may be written as*

$$D\mathbf{x}_{k+1} = (L + U)\mathbf{x}_k + \mathbf{b}. \tag{9.4.2}$$

□

Exercise 9.4.2 *Apply Jacobi's method to the problem of Exercise 9.3.2. Estimate λ_1 and show from the numerical results that $\lambda_1 = 1 - kh^2$, with k a positive constant.* □

Exercise 9.4.3 *Prove using Gershgorin's Theorem that Jacobi's method converges if A is diagonally dominant.* □

Exercise 9.4.4 *Prove that Jacobi's method converges if A is an M-matrix.* □

9.4.2 Gauss-Seidel

With the same notation as in the previous section we take $P = D - L$ to obtain the method of Gauss-Seidel. Note that we in general do not calculate P^{-1} itself, since the defect correction algorithm only requires us to solve the set $P\mathbf{c}_k = \mathbf{r}_k$. That is easy in this case, because it only requires backsubstitution.

Exercise 9.4.5 *Prove that Gauss-Seidel's method can be written as*

$$(D - L)\mathbf{x}_{k+1} = U\mathbf{x}_k + \mathbf{b}. \tag{9.4.3}$$

□

Exercise 9.4.6 *Prove that Gauss-Seidel's iteration matrix is given by*

$$M = I - P^{-1}A = I - (D-L)^{-1}(D-L-U) = (D-L)^{-1}U. \tag{9.4.4}$$

□

Exercise 9.4.7 *Prove that Gauss-Seidel converges if A is an M-matrix.* □

Gauss-Seidel's method converges for an important class of practical problems, the positive definite problems. Roughly all those that come from a minimization problem.

Theorem 9.4.1 *Let $A = D - L - L^T$ be positive definite. Then Gauss-Seidel's method converges.*

Proof
We have to show that all eigenvalues of $M = (D-L)^{-1}L^T$ are in absolute value less than 1. Let λ be an eigenvalue of M, with corresponding eigenvector $\mathbf{v}$. We have:

$$(D-L)^{-1}L^T\mathbf{v} = \lambda\mathbf{v}, \tag{9.4.5}$$

or equivalently

$$L^T\mathbf{v} = \lambda(D-L)\mathbf{v}. \tag{9.4.6}$$

This eigenvalue λ may be *complex* and the corresponding eigenvector will also be complex in that case. The conjugate complex quantities $\bar{\lambda}$ and $\bar{\mathbf{v}}$ will be eigenvalue and eigenvector too, since A is a real matrix. We define the ordinary inner product on complex spaces:

$$(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{n} \bar{x}_k y_k. \tag{9.4.7}$$

(Observe that $(\mathbf{v}, \mathbf{v}) > 0$ unless $\mathbf{v} = 0$ and that $(\mathbf{v}, A\mathbf{v}) > 0$ unless $\mathbf{v} = 0$.) Consider

$$\begin{aligned}
(\mathbf{v}, A\mathbf{v}) &= (\mathbf{v}, (D-L-L^T)\mathbf{v}), && (9.4.8)\\
&= (\mathbf{v}, (D-L)\mathbf{v}) - \lambda(\mathbf{v}, (D-L)\mathbf{v}), \text{by Equation (9.4.6)}, && (9.4.9)\\
&= (1-\lambda)(\mathbf{v}, (D-L)\mathbf{v}). && (9.4.10)
\end{aligned}$$

We also have:

$$\begin{aligned}
(\mathbf{v}, A\mathbf{v}) &= (\mathbf{v}, (D-L-L^T)\mathbf{v}), && (9.4.11)\\
&= (\mathbf{v}, (D-L^T)\mathbf{v}) - (\mathbf{v}, L\mathbf{v}), && (9.4.12)\\
&= (\mathbf{v}, (D-L^T)\mathbf{v}) - (L^T\mathbf{v}, \mathbf{v}), && (9.4.13)\\
&= (\mathbf{v}, (D-L^T)\mathbf{v}) - (\lambda(D-L)\mathbf{v}, \mathbf{v}), && (9.4.14)\\
&= (\mathbf{v}, (D-L^T)\mathbf{v}) - \bar{\lambda}(\mathbf{v}, (D-L^T)\mathbf{v}), && (9.4.15)\\
&= (1-\bar{\lambda})(\mathbf{v}, (D-L^T)\mathbf{v}). && (9.4.16)
\end{aligned}$$

Because A is positive definite λ cannot be equal to 1. Hence

$$\begin{aligned}
\left(\frac{1}{1-\lambda} - \frac{1}{1-\bar{\lambda}}\right)(\mathbf{v}, A\mathbf{v}) &= (\mathbf{v}, (2D-L-L^T)\mathbf{v}), && (9.4.17)\\
&= (\mathbf{v}, A\mathbf{v}) + (\mathbf{v}, D\mathbf{v}), && (9.4.18)
\end{aligned}$$

and therefore

$$\left(\frac{1}{1-\lambda} - \frac{1}{1-\bar{\lambda}} - 1\right)(\mathbf{v}, A\mathbf{v}) = (\mathbf{v}, D\mathbf{v}). \tag{9.4.19}$$

Because A is positive definite, so is D (why?) and we find

$$\frac{1}{1-\lambda} - \frac{1}{1-\bar{\lambda}} - 1 > 0, \tag{9.4.20}$$

$$\frac{1}{1-\lambda} - \frac{1}{1-\bar{\lambda}} > 1, \tag{9.4.21}$$

$$\frac{1-\bar{\lambda}+1-\lambda}{(1-\lambda)(1-\bar{\lambda})} > 1, \tag{9.4.22}$$

$$\frac{2-2\Re(\lambda)}{1-2\Re(\lambda)+|\lambda|^2} > 1, \tag{9.4.23}$$

in which $\Re(\lambda)$ denotes the real part of λ. The denominator in Inequality (9.4.23) is always positive (why?) hence

$$2-2\Re(\lambda) > 1-2\Re(\lambda)+|\lambda|^2, \tag{9.4.24}$$

$$|\lambda|^2 < 1. \tag{9.4.25}$$

□

9.4.3 Successive Overrelaxation SOR

Successive overrelaxation (SOR) has been devised as an improvement on Gauss-Seidel's method. The preconditioner of choice is $P = \frac{1}{\omega}(D - \omega L)$ in which $\omega > 0$ is a parameter that still has to be chosen. It comes down to multiplying the Gauss-Seidel correction *in each point* by ω before applying it. Since the back substitution in Gauss-Seidel uses already updated points this works recursively in a fairly complex way. The term *overrelaxation* really applies only for values $\omega > 1$, for $\omega < 1$ you have *underrelaxation.*

Exercise 9.4.8 *Show that the SOR process can be expressed as*

$$(D - \omega L)\mathbf{x}_{k+1} = ((1-\omega)D + \omega U)\mathbf{x}_k + \omega\mathbf{b}. \tag{9.4.26}$$

□

Exercise 9.4.9 *The* iteration matrix *M for a defect correction method is $I - P^{-1}A$. Show for the SOR iteration matrix M_ω:*

$$M_\omega = (D - \omega L)^{-1}((1-\omega)D + U). \tag{9.4.27}$$

□

Theorem 9.4.2 *If A is positive definite and $0 < \omega < 2$, then SOR converges.*

Exercise 9.4.10 *Prove Theorem 9.4.2 in the same way as Theorem 9.4.1* □

Apparently it is important how to choose ω. There are a number of theoretical results on that, that are valid for matrices A of a special structure: *diagonally block tridiagonal*. That is, the matrix consists of *blocks* and only the *diagonal*, *superdiagonal* and *subdiagonal* blocks may be nonzero. Moreover, the diagonal blocks must be diagonal matrices themselves.

Exercise 9.4.11 *Let V be a rectangle with a regular grid. A* checker board numbering *of the nodes is constructed as follows. The points are painted white and black alternately in the same pattern as the squares of a checkerboard. Now first the black points are numbered and after that the white points. Show that for the 5-point Laplace molecule the resulting matrix is a 2×2 block matrix with diagonal block matrices in the diagonal blocks.* □

Exercise 9.4.12 *Let V be a rectangle with a regular grid that has been obliquely numbered. Show that for the 5-point Laplace molecule the resulting matrix is diagonally block tridiagonal.* □

For diagonally blocktridiagonal matrices there is a functional relationship between the eigenvalues $\lambda_{\omega,k}$ of M_ω, the iteration matrix of SOR and the corresponding eigenvalues $\lambda_{1,k}$ of M_1 the iteration matrix of Gauss-Seidel.

$$\left(\frac{\lambda_\omega - 1 + \omega}{\omega}\right)^2 = \lambda_\omega \lambda_1. \tag{9.4.28}$$

For a proof see [5].

Now let $|\lambda_{1,1}| = \rho(M_1)$ the spectral radius of the Gauss-Seidel iteration matrix. For diagonally block tridiagonal matrices A the following expression for the optimal value of ω holds:

$$\omega_{\text{opt}} = \frac{2}{1 + \sqrt{1 - \lambda_{1,1}}}. \tag{9.4.29}$$

So to estimate the optimum value of ω we have to know the value of $\lambda_{1,1}$. It is possible though, to estimate this value during the SOAR process using Equation (9.3.11) to estimate $\lambda_{\omega,1}$ and subsequently use Equation (9.4.28) to estimate $\lambda_{1,1}$ Care has to be taken however, that the eigenvalue belonging to the spectral radius does not become complex, because in that case the use of estimate (9.3.11) is no longer justified.

Exercise 9.4.13 *Let A be a real matrix with complex eigenvalues λ_1 and $\bar{\lambda}_1$ such that $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \ldots$. Show that* two *components survive in the error:*

$$\varepsilon_k \approx a_1 \lambda_1^k \mathbf{v}_1 + \bar{a}_1 \bar{\lambda}^k \bar{\mathbf{v}_1}. \tag{9.4.30}$$

□

In the remaining exercises of this section you may assume that A is a positive definite diagonally blocktridiagonal matrix.

Exercise 9.4.14 *Show from Equation* (9.4.28) *that $\lambda_{\omega,k}$ is complex if*

$$\lambda_{1,k} < \frac{4(\omega - 1)}{\omega^2}. \tag{9.4.31}$$

Show from Equation (9.4.28) *that if $\lambda_{\omega,k}$ is complex, then $|\lambda_{\omega,k}| = |\omega - 1|$. (Hint: What is the product of the roots of a quadratic equation)* □

Exercise 9.4.15 *Show that when $\omega = \omega_{\text{opt}}$ that*

$$\lambda_{1,1} = \frac{4(\omega - 1)}{\omega^2}. \tag{9.4.32}$$

Show using Exercise 9.4.14 that when $\omega > \omega_{\text{opt}}$ all eigenvalues of M_ω lie on a circle in the complex plane with radius $\omega - 1$. □

Exercise 9.4.16 *Suppose $\lambda_1 = 0.9999$. Estimate the number of iterations to gain one decimal digit using Gauss-Seidel. (Use Exercise 9.3.7) Calculate $\lambda_{\omega,1} = 1 - \omega_{\text{opt}}$ and estimate the number of iterations to gain one decimal using optimal SOAR.* □

9.4.4 Block variations

Block variations of Jacobi, Gauss Seidel and SOAR can be used if the matrix A in the system of equations $A\mathbf{x} = \mathbf{b}$ is a block matrix and the *diagonal* blocks can be solved easily, for example if they are tridiagonal. The convergence properties of block variations are a bit better than those of the standard methods.

9.4.5 Operation count

In numerical approximations of PDE's the number of unknowns per equation is fixed. In that case it is easy to estimate the operation count *per iteration*. Let N be the number of unknowns, then clearly the number of operations for a matrix vector multiplication takes kN operations (multiplication + addition), with k the number of unknowns per equation. Solving the preconditioner equation takes mN operations, $m < k$. If we take the two matrix additions into account and the calculation of $\|r_k\|$ for the stop criterion we end up with $(k + m + 3)N$ operations per iteration.

How many iterations do we need? Basically this depends on the accuracy we demand, but a good measure for that is the number of iterations n to gain one extra accurate decimal. This number n is given by (see Exercise 9.3.7)

$$n = -\frac{1}{^{10}\log \rho}, \tag{9.4.33}$$

or using natural logarithms

$$n = -\frac{\ln 10}{\ln \rho} = -\frac{2.3}{\ln \rho}, \tag{9.4.34}$$

with ρ the spectral radius of the iteration matrix $I - P^{-1}A$. This spectral radius is for PDE matrices and the Jacobi and Gauss-Seidel methods $1 - k_p h^2$. Here k_p is a constant depending on the problem, the form of the region and the method.

Exercise 9.4.17 *Show, using Equation* (9.4.29) *and Exercise 9.4.15 that for diagonal block-tridiagonal matrices the spectral radius of the SOAR iteration matrix for optimal ω is $\rho = 1 - k'_p h$.* □

Since roughly $h^{-1} = N^{\frac{1}{2}}$ for two dimensional problems and $h^{-1} = N^{\frac{1}{3}}$ for three dimensional problems we find

$$n = \frac{2.3}{k_p h^2} = K_p N, \tag{9.4.35}$$

for 2D Jacobi and Gauss Seidel and

$$n = \frac{2.3}{k_p h} = K_p N^{\frac{1}{2}}, \tag{9.4.36}$$

for 2D optimal SOAR. Since these are the number of iterations and the operation count per iteration is $O(N)$ the total operation count to gain one digit (2D) is $nO(N) = KN^2$ for Gauss Seidel and Jacobi and $nO(N) = KN^{\frac{3}{2}}$ for optimal SOAR. K depends on the method. For 3D these numbers are: $KN^{\frac{5}{3}}$ and $KN^{\frac{4}{3}}$ respectively.

9.5 Krylov Space Methods

Our treatment of Krylov space methods has to be superficial. We shall only consider CG in detail and out of the numerous other possibilities we shall only present BiCG-Stab in algorithmic form. The reader who wants to have a more thorough understanding of the subject should consult [41].

9.5.1 Introduction

We first consider simple standard iteration without preconditioning on $A\mathbf{x} = \mathbf{b}$.

Let us consider the form of the error after $n+1$ iterations (see Exercises 9.3.5 and 9.3.6):

$$\mathbf{r}_{n+1} = (I - A)^{n+1}\mathbf{r}_0 = P_{n+1}(A)\mathbf{r}_0, \tag{9.5.1}$$

$$\varepsilon_{n+1} = (I - A)^{n+1}\varepsilon_0 = P_{n+1}(A)\varepsilon_0. \tag{9.5.2}$$

$P_{n+1}(A)$ is an $n+1$-st degree matrix polynomial. Let us assume, that A has eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ and corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$. We may take $\mathbf{x}_0 = 0$ and $\mathbf{r}_0 = \mathbf{b}$ without loss of generality and express $\mathbf{r}_0$ as a linear combination of eigenvectors:

$$\mathbf{r}_0 = \sum_{k=1}^{N} \alpha_k \mathbf{v}_k, \tag{9.5.3}$$

and because $A\mathbf{v}_k = \lambda_k \mathbf{v}_k$ we have

$$\mathbf{r}_n = (I - A)^n r_0 = \sum_{k=1}^{N} \alpha_k (1 - \lambda_k)^n \mathbf{v}_k, \tag{9.5.4}$$

and in general for any matrix polynomial $P_n(A)$:

$$P_n(A)\mathbf{r}_0 = \sum_{k=1}^{N} \alpha_k P_n(\lambda_k)\mathbf{v}_k. \tag{9.5.5}$$

Immediately we deduce that for convergence of the standard iteration all λ_k must be within a circle in the complex plane with radius 1 and real midpoint 1 (see Figure 9.6) The polynomial $(1-\lambda)^n$ is not specifically chosen to make the residual as small as possible. On the contrary, let us draw a picture of $(1-x)^{10}$ on the interval $(0, 2)$.
You can see, that for eigenvectors with eigenvalues between 0.4 and 1.6 there is a very good convergence, but for eigenvalues close to 0 and close to 2 the convergence is rather bad.

There are two things that could be done about that. First you could try to find a better polynomial than $(1-\lambda)^n$. That is what *Krylov (sub)space methods* do. Secondly you could try to treat the remaining parts of the spectrum differently. That is what *Multigrid methods* do.

9.5.2 The Krylov Space

The *Krylov subspace of dimension k* is spanned by the vectors $\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots, A^{k-1}\mathbf{r}_0$ and denoted by $\mathcal{K}^k(A; \mathbf{r}_0)$.

Exercise 9.5.1 *Show that in standard iteration* $\mathbf{r}_k \in \mathcal{K}^{k+1}(A; \mathbf{r}_0)$. *Infer from this that* $\mathbf{x}_{k+1} \in \mathcal{K}^{k+1}(A; \mathbf{r}_0)$. □

Krylov subspace methods all try to optimize the approximate solution in the k-dimensional subspace $\mathcal{K}^k(A; \mathbf{r}_0)$. This can be done in various ways of which we will consider only two: Conjugate Gradients and Bi-CGStab.

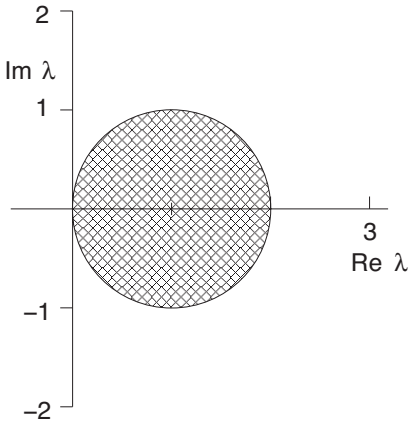

Figure 9.6: Region of convergence.

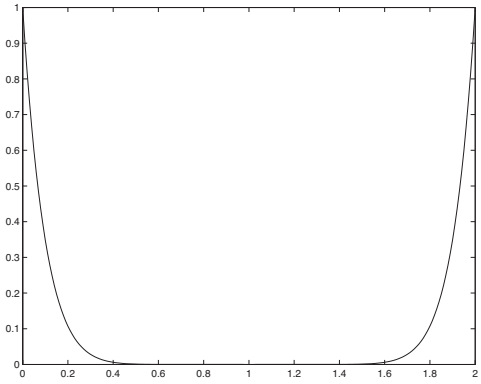

Figure 9.7: Graph of $(1-x)^{10}$.

9.5.3 Conjugate Gradients

The Conjugate Gradient method can best be explained for *A positive definite*. In that case solving $A\mathbf{x} = \mathbf{b}$ is equivalent to a minimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x}. \tag{9.5.6}$$

Exercise 9.5.2 *Show that for positive definite matrices A these formulations are equivalent. Use the same type of argument as in Chapter 5.* □

Starting from $\mathbf{x}_0 = 0$, $\mathbf{r}_0 = \mathbf{b}$ we now seek $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k$, with $\mathbf{s}_j \in \mathcal{K}^j(A;\mathbf{b})$ and solve the minimization problem in the Krylov subspace $\mathcal{K}^k$

$$\min_{\mathbf{x}\in\mathcal{K}^k} \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x}. \tag{9.5.7}$$

Exercise 9.5.3 *Let $A\xi = \mathbf{b}$, in other words ξ is the solution to the linear system. Show that Problem 9.5.7 is equivalent to*

$$\min_{\mathbf{x}\in\mathcal{K}^k} \frac{1}{2}(\mathbf{x}-\xi)^T A(\mathbf{x}-\xi). \tag{9.5.8}$$

□

Exercise 9.5.4 *Let $A\xi = \mathbf{b}$, in other words ξ is the solution to the linear system. Show that Problem 9.5.7 is equivalent to*

$$\min_{\mathbf{x}\in\mathcal{K}^k} \frac{1}{2}\mathbf{r}^T A^{-1}\mathbf{r}, \tag{9.5.9}$$

in which $\mathbf{r} = \mathbf{b} - A\mathbf{x}$. □

So in a way we do the best we can, but you may ask yourself the question if this is really an easier problem than the original one.

Let us write $\mathbf{x}_k = \sum_{j=0}^{k-1} \alpha_j \mathbf{s}_j$. Let us solve minimization Problem 9.5.7. This is just Ritz's method, so what we get is a $k \times k$ set of equations of the form

$$\sum_{j=0}^{k-1} \sigma_{mj}\alpha_j = \beta_m, \quad m = 0, 1, \ldots, k-1, \tag{9.5.10}$$

in which

$$\sigma_{mj} = (\mathbf{s}_m, A\mathbf{s}_j), \tag{9.5.11a}$$
$$\beta_m = (\mathbf{s}_m, \mathbf{b}), \tag{9.5.11b}$$

or in matrix vector notation $\Sigma\alpha = \beta$.

Exercise 9.5.5 *Explain that Equations* (9.5.11) *are an analogue for Galerkin's method.* □

Exercise 9.5.6 *Prove that $\mathbf{r}_k$ is orthogonal to span $\mathbf{s}_0, \mathbf{s}_2, ..., \mathbf{s}_{k-1}$, hence orthogonal to $\mathcal{K}^k(A; \mathbf{b})$.* □

The remark could be made that this is not much of an iterative method. That is right, so far it is not. But we have some freedom left in the choice of $\mathbf{s}_j$ which will make it one. If you consider Equation (9.5.10) you will see, that in every new step *all α_j will change*, unless you make the matrix Σ *diagonal*. In that case the addition of a new dimension to the Krylov subspace will not touch already calculated α_j's . And *that* makes it a truly iterative method. So we have to make sure that $(\mathbf{s}_k, A\mathbf{s}_j) = 0, k \neq j$.

9.5.4 CG algorithm

We summarize all this in the following algorithm:
Conjugate Gradient Algorithm (CG)
Require: A positive definite
1: Presets: $\mathbf{x}_0 = 0$, $\mathbf{r}_0 = \mathbf{b}$, $\mathbf{s}_0 = \mathbf{r}_0$, $k = 0$
2: **while** $\|\mathbf{r}_k\| > \varepsilon\|\mathbf{b}\|$ **do**
3: $\alpha_k = (\mathbf{r}_k, \mathbf{r}_k)/(\mathbf{s}_k, A\mathbf{s}_k)$ {See Exercise 9.5.8}
4: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{s}_k$
5: $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{s}_k$
6: $\beta_k = (\mathbf{r}_{k+1}, \mathbf{r}_{k+1})/(\mathbf{r}_k, \mathbf{r}_k)$
7: $\mathbf{s}_{k+1} = \mathbf{r}_{k+1} + \beta_k\mathbf{s}_k$
8: $k = k + 1$
9: **end while**

The algorithm generates residual vectors $\mathbf{r}_k$ in subsequent Krylov spaces and they are *mutually orthogonal*. This follows in fact from Galerkin's condition (see Exercise 9.5.6). Moreover, the search directions $\mathbf{s}_k$ are mutually $A-$orthogonal ensuring the diagonality of the Galerkin matrix Σ. The positive definiteness of A guarantees that the algorithm will not crash: the denominator of α_k cannot vanish.

The proof of these claims is the subject of the subsequent exercises.

Exercise 9.5.7 *Show from lines 5 and 7, that if* $\mathbf{r}_k, \mathbf{s}_k \in \mathcal{K}^m(A;\mathbf{b})$ *then* $\mathbf{r}_{k+1}, \mathbf{s}_{k+1} \in \mathcal{K}^{m+1}(A;\mathbf{b})$*, independent of the values of* α_k *and* β_k*. Infer by induction that* $\mathbf{r}_k, \mathbf{s}_k \in \mathcal{K}^{k+1}(A;\mathbf{b})$ *for all* k*.*

Explain that if

$$\mathcal{K}^{k+1}(A;\mathbf{b}) \subset \text{span}\,\{\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_k\}, \tag{9.5.12}$$

then

$$\mathcal{K}^{k+2}(A;\mathbf{b}) \subset \text{span}\,\{\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{k+1}\} \text{ if } \alpha_k \neq 0. \tag{9.5.13}$$

(Hint: it is sufficient to show that $A^{k+1}\mathbf{b} \in \text{span}\,\{\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{k+1}\}$*. Use lines 5 and 7 of the algorithm.)* □

Exercise 9.5.8 *Assume*

$$(\mathbf{r}_k, \mathbf{v}) = 0,\ \forall \mathbf{v} \in \mathcal{K}^k(A;\mathbf{b}) \quad \textit{(Galerkin's condition)} \tag{9.5.14}$$

and

$$(\mathbf{s}_k, A\mathbf{v}) = 0,\ \forall \mathbf{v} \in \mathcal{K}^k(A;\mathbf{b}) \quad \textit{(diagonality condition)}. \tag{9.5.15}$$

1. *Show that by line 5 of the algorithm* $(\mathbf{r}_{k+1}, \mathbf{v}) = 0,\ \forall \mathbf{v} \in \mathcal{K}^k(A;\mathbf{b})$ independent of α_k*. Infer from this and the previous exercise, that to satisfy Galerkin's condition for* $\mathcal{K}^{k+1}$ *it is sufficient to ensure that* $(\mathbf{r}_{k+1}, \mathbf{r}_k) = 0$*. Show that line 3 of the algorithm does just that. Use line 7 and the diagonality condition.*

2. *Show that by line 7 of the algorithm* $(\mathbf{s}_{k+1}, A\mathbf{v}) = 0,\ \forall \mathbf{v} \in \mathcal{K}^k$ independent of β_k*. Use the diagonality condition and the fact that if* $\mathbf{v} \in \mathcal{K}^k$ *then* $A\mathbf{v} \in \mathcal{K}^{k+1}$*. Infer from this and the previous exercise that to satisfy the diagonality condition for* $\mathcal{K}^{k+1}$ *it is sufficient to ensure that* $(\mathbf{s}_{k+1}, A\mathbf{s}_k) = 0$*. Show that line 6 of the algorithm does just that. Use lines 5 and 3.*

□

Exercise 9.5.9 *From Equation* (9.5.11) *you would expect*

$$\alpha_k = (\mathbf{s}_k, \mathbf{b})/(\mathbf{s}_k, A\mathbf{s}_k) \tag{9.5.16}$$

on line 3. Show that

$$(\mathbf{s}_k, \mathbf{b}) = (\mathbf{s}_k, \mathbf{r}_k). \tag{9.5.17}$$

(Use the diagonality condition). Next show that

$$(\mathbf{s}_k, \mathbf{r}_k) = (\mathbf{r}_k, \mathbf{r}_k). \tag{9.5.18}$$

(Use line 7 of the algorithm) □

Exercise 9.5.10 *Various inner products in the CG algorithm are used multiple times. It is a waste to calculate them each time anew. The same is true for the matrix vector product* $A\mathbf{s}_k$*. Reformulate the algorithm in such a way that each iteration only needs two inner products and one matrix vector multiplication.* □

9.5.5 Preconditioning

In practice CG is always used with a preconditioner P, but we have to be careful. It is not a good idea to apply the algorithm to the preconditioned system $P^{-1}A\mathbf{x} = P^{-1}\mathbf{b}$, because in general $P^{-1}A$ will no longer be symmetric even if P is. If we have a *factorization* of P:

$$P = LL^T, \tag{9.5.19}$$

we can construct a symmetric preconditioned system as follows:

$$L^{-1}AL^{-1^T}\mathbf{y} = L^{-1}\mathbf{b}. \tag{9.5.20}$$

This has various drawbacks, most notably, that the preconditioner must be available in factored form and that the solution must be backtransformed later on. There is a better way to go about this. We have defined CG with respect to the classical inner product $(\mathbf{x}, \mathbf{y}) = \sum x_k y_k$ but in fact every positive definite matrix B generates an inner product $(\mathbf{x}, \mathbf{y})_B = (\mathbf{x}, B\mathbf{y})$.

Exercise 9.5.11 *Show that $(.,.)_B$ is a proper inner product.* □

Exercise 9.5.12 *Show that with P and A symmetric positive definite $(P^{-1}A\mathbf{x}, \mathbf{y})_P = (\mathbf{x}, P^{-1}A\mathbf{y})_P$.* □

Using this inner product we may formulate the *preconditioned CG algorithm*:

Require: A, P positive definite
Presets: $\mathbf{x}_0 = 0$, $\mathbf{r}_0 = \mathbf{b}$, $\mathbf{t}_0 = \mathbf{s}_0 = P^{-1}\mathbf{r}_0$, $k = 0$
while $(\mathbf{r}_k, \mathbf{t}_k) > \varepsilon^2(\mathbf{r}_0, \mathbf{t}_0)$ **do**
  $\alpha_k = (\mathbf{r}_k, \mathbf{t}_k)/(\mathbf{s}_k, A\mathbf{s}_k)$
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{s}_k$
  $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{s}_k$
  Solve $P\mathbf{t}_{k+1} = \mathbf{r}_{k+1}$
  $\beta_k = (\mathbf{r}_{k+1}, \mathbf{t}_{k+1})/(\mathbf{r}_k, \mathbf{t}_k)$
  $\mathbf{s}_{k+1} = \mathbf{t}_{k+1} + \beta_k \mathbf{s}_k$
  $k = k + 1$
end while

The algorithm uses ordinary inner products, but careful analysis will show that it is in fact CG applied to $P^{-1}A\mathbf{x} = P^{-1}\mathbf{b}$, with the $(.,.)_P$ inner product.

Exercise 9.5.13 *Show that minimizing $(\varepsilon, P^{-1}A\varepsilon)_P$ is the same as minimizing $(\varepsilon, A\varepsilon)$.* □

Exercise 9.5.14 *Show that the preconditioned CG as described above minimizes over the Krylov space $\mathcal{K}^k(P^{-1}A; P^{-1}\mathbf{b})$.* □

Exercise 9.5.15 *(Symmetric Gauss Seidel preconditioner) Let $A = D - L - L^T$ be positive definite, with D diagonal and L lower triangular. Show that $P = (D - L)D^{-1}(D - L^T)$ is positive definite and symmetric.Show that $P\mathbf{t} = \mathbf{r}$ can be solved in three easy steps, of which the first is: solve $(D - L)\mathbf{y} = \mathbf{r}$.* □

Exercise 9.5.16 *Show that the symmetric Gauss Seidel preconditioner generates a regular splitting if A is an M-matrix.* □

Exercise 9.5.17 *Do Exercise 9.5.10 for the preconditioned CG algorithm.* □

9.5.6 Convergence

In theory CG is a finite algorithm (why?) but that is not the reason for its usefulness. Also as an iteration process it has very good properties. In order to understand that we take a closer look at approximations in the Krylov space. Elements of the Krylov space $\mathcal{K}^{k+1}(A;\mathbf{b})$ can be written as

$$\mathbf{y} = \sum_{j=0}^{k} a_j A^j \mathbf{b} = Q_k(A)\mathbf{b}, \tag{9.5.21}$$

in which Q_k is a k^{th} degree polynomial. In CG approximations $a_0 = 1$ hence $Q(0) = 1$ and in fact, the i^th iteration can be written as

$$\varepsilon_i = Q_i(A)\varepsilon_0, \tag{9.5.22}$$

in which $\varepsilon_i = \mathbf{x} - \mathbf{x}_i$ is the error in the i^{th} iteration step. Since (see Exercise 9.5.3) CG minimizes $(\varepsilon_i, A\varepsilon_i)$, or equivalently $(\varepsilon_i, \varepsilon_i)_A$. Apparently the expression

$$(Q_i(A)\varepsilon_0, Q_i(A)\varepsilon_0)_A \tag{9.5.23}$$

is minimized over all possible polynomials Q_i of degree i with $Q_i(0) = 1$. Let us call this optimal polynomial P_i. Let $\lambda_1, \lambda_2, \ldots, \lambda_N$ be the eigenvalues of A in increasing order, with corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N$. We have:

$$\mathbf{b} = \sum_{j=1}^{N} b_j \mathbf{v}_j, \tag{9.5.24}$$

$$\mathbf{r}_i = P_i(A)\mathbf{r}_0 = \sum_{j=1}^{N} P_i(\lambda_j) b_j \mathbf{v}_j, \tag{9.5.25}$$

$$\varepsilon_i = A^{-1}\mathbf{r}_i = \sum_{j=1}^{N} P_i(\lambda_j)/\lambda_j b_j \mathbf{v}_j, \tag{9.5.26}$$

$$(\varepsilon_i, \varepsilon_i)_A = \sum_{j=1}^{N} P_i^2(\lambda_j)/\lambda_j b_j^2. \tag{9.5.27}$$

By comparing P_i with specific polynomials we may obtain an error estimate.

$$(\varepsilon_i, \varepsilon_i)_A = \sum_{j=1}^{N} P_i^2(\lambda_j)/\lambda_j b_j^2 \leq \sum_{j=1}^{N} Q_i^2(\lambda_j)/\lambda_j b_j^2, \tag{9.5.28}$$

$$\leq \max_{\lambda_j} Q_i^2(\lambda_j) \sum_{j=1}^{N} b_j^2/\lambda_j = \max_{\lambda_j} Q_i^2(\lambda_j)(\varepsilon_0, \varepsilon_0)_A. \tag{9.5.29}$$

Q_i is an arbitrary polynomial with $Q_i(0) = 1$.

Exercise 9.5.18 *If A has an eigenvalue with multiplicity $N-1$ then CG needs two iterations to find the exact solution. Explain why.* □

Exercise 9.5.19 *Explain why it is a good thing to have clusters of eigenvalues and a bad thing to have the eigenvalues evenly spread out over the spectrum.* □

We now derive a famous upperbound for the error in the $\|.\|_A$ norm by using scaled, shifted and mirrored Chebyshev polynomials. Chebyshev polynomials $T_n(x)$ are connected to $\cos(n\phi)$. You can expand $\cos(n\phi)$ into an n-th degree polynomial in $\cos\phi$. See exercise 9.5.20

Exercise 9.5.20 *Show that* $\cos n\phi$ *satisfies the following recurrence relation:*

$$\cos(n+1)\phi = 2\cos\phi\cos n\phi - \cos(n-1)\phi, \quad n \geq 2. \tag{9.5.30}$$

Infer from this and $\cos 0 = 1$, *that* $\cos n\phi$ *is a polynomial in* $\cos\phi$ □

$T_n(x)$ is now obtained by substituting $\cos\phi = x$ into that polynomial. That explains its behavior for $-1 \leq x \leq 1$: $|T_n(x)| \leq 1, \quad -1 \leq x \leq 1$. And for values outside that interval? The general solution for the recurrence relation:

$$T_{n+1} = 2xT_n - T_{n-1}, \quad T_0 = 1, T_1 = x, \tag{9.5.31}$$

is given by

$$T_n(x) = \frac{1}{2}((x + \sqrt{x^2-1})^n + (x + \sqrt{x^2-1})^{-n}). \tag{9.5.32}$$

Exercise 9.5.21 *Show that the general solution* u_n *of the recurrence relation*

$$u_{n+1} = 2xu_n - u_{n-1} \tag{9.5.33}$$

satisfies $u_n = A\rho_1^n + B\rho_2^n$ *with* A *and* B *arbitrary and* ρ_1 *and* ρ_2 *the solutions of the quadratic equation* $\rho^2 - 2x\rho + 1 = 0$. *Calculate from this the expression in Equation* (9.5.32).
Clearly this expression is valid for $x \geq 1$. *Is it also valid for* $x < 1$? □

We now cleverly take as our comparison polynomial:

$$Q_i(\lambda) = \frac{T_i(\frac{2\lambda-(\lambda_1+\lambda_N)}{\lambda_1-\lambda_N})}{T_i(-\frac{\lambda_1+\lambda_N}{\lambda_1-\lambda_N})}, \tag{9.5.34}$$

in which λ_1 is the minimum and λ_N the maximum eigenvalue of A. Observe, that we have scaled in such a way that $Q_i(0) = 1$. Let us introduce the condition number $K = \lambda_N/\lambda_1$ and the quantity $B = (K+1)/(K-1)$. By inequality (9.5.29) we have

$$(\varepsilon_i, \varepsilon_i)_A \leq \max_{\lambda_j} Q_i^2(\lambda_j)(\varepsilon_0, \varepsilon_0)_A, \tag{9.5.35}$$

$$\leq \frac{1}{T_i^2(B)}(\varepsilon_0, \varepsilon_0)_A, \tag{9.5.36}$$

$$\leq \frac{2}{(B + \sqrt{B^2-1})^i}(\varepsilon_0, \varepsilon_0)_A, \tag{9.5.37}$$

$$\leq 2(B - \sqrt{B^2-1})^i(\varepsilon_0, \varepsilon_0)_A, \tag{9.5.38}$$

$$\leq 2\left(\frac{\sqrt{K}-1}{\sqrt{K}+1}\right)^i(\varepsilon_0, \varepsilon_0)_A. \tag{9.5.39}$$

This famous error estimate is on the pessimistic side, because as the process continues the *effective* condition number will get gradually smaller, because the extremal eigenvalues of the spectrum will have been sufficiently well approximated by the polynomial P_i. Compared to the standard iteration methods CG performs at least as well as SOAR, but is applicable to more general matrices. In fact if we use a preconditioner like Incomplete LU (See section 9.5.8.2) CG will outperform SOAR by a considerable margin.

9.5.7 Krylov space methods for non symmetric matrices.

9.5.7.1 Bi-CG and CGS

The Bi-CG method can be viewed ([41], pg 98) as an application of the preconditioned CG method on the problem

$$\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{x}} \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \hat{\mathbf{b}} \end{pmatrix}, \tag{9.5.40}$$

with preconditioner $P = \left(\begin{smallmatrix} 0 & I \\ I & 0 \end{smallmatrix}\right)$. $\hat{\mathbf{b}}$ must be some suitably chosen vector. The big problem with that is, that neither the system matrix $B = \left(\begin{smallmatrix} 0 & A \\ A^T & 0 \end{smallmatrix}\right)$nor the preconditioner P is positive definite, so the inner product $(.,.)_P$ is not a proper inner product and the algorithm may break down.

The CGS method tries to improve the convergence speed of Bi-CG by a clever trick on the matrix polynomials $P_i(A)$. For a derivation we refer once more to [41]. If both algorithms converge, CGS converges about twice as fast for the same operation count per iteration. CGS unfortunately has a rather irregular convergence behavior.

9.5.7.2 BiCG-stab

BiCG-stab stabilizes the convergence behavior of CGS and maintains the improved convergence of this method. For the derivation of this method, which is well beyond the scope of this book see [41]. We shall just present the algorithm.

BiCG-Stab without preconditioning
Presets: $\mathbf{x}_0$, $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, $\tilde{\mathbf{r}} \neq 0$, $k = 0$, $\beta_0 = 0$, $\mathbf{p}_0 = 0$, $\omega_0 = 1$, $\mathbf{v}_0 = 0$.
$\rho_0 = (\tilde{\mathbf{r}}, \mathbf{r}_0)$
while not converged **do**
  if $\rho_k = 0$ or $\omega_k = 0$ **then**
    Break {Failure}
  end if
  $\mathbf{p}_{k+1} = \mathbf{r}_k + \beta_k(\mathbf{p}_k - \omega_k \mathbf{v}_k)$
  $\mathbf{v}_{k+1} = A\mathbf{p}_{k+1}$
  $\alpha_{k+1} = \rho_k / (\tilde{\mathbf{r}}, \mathbf{v}_{k+1})$
  $\mathbf{s} = \mathbf{r}_k - \alpha_{k+1}\mathbf{v}_{k+1}$
  $\mathbf{t} = A\mathbf{s}$
  $\omega_{k+1} = (\mathbf{t}, \mathbf{s}) / (\mathbf{t}, \mathbf{t})$
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_{k+1}\mathbf{p}_{k+1} + \omega_{k+1}\mathbf{s}$
  $\mathbf{r}_{k+1} = \mathbf{s} - \omega_{k+1}\mathbf{t}$
  $\rho_{k+1} = (\tilde{\mathbf{r}}, \mathbf{r}_{k+1})$
  $\beta_{k+1} = (\rho_{k+1}/\rho_k)(\alpha_{k+1}/\omega_{k+1})$
  $k = k + 1$
end while

9.5.8 Preconditioners

The success of CG and BiCG-stab largely depends on the application of preconditioners. The easy preconditioners are based on Jacobi and Gauss-Seidel, but the most powerful preconditioners are *incomplete factorizations*.

9.5.8.1 Jacob and Gauss-Seidel

Let A be given by $D - L - U$ in which D is diagonal, L is lower triangular and U is upper triangular. Now the Jacob preconditioner is just given by $P = D$. In this particular case it is also possible (provided A is symmetric and positive definite) to do simple CG on a modified system:

$$TAT\mathbf{y} = T\mathbf{b}, \quad \mathbf{x} = T\mathbf{y}, \tag{9.5.41}$$

with $T = \sqrt{D^{-1}} = \text{diag}(d_{kk}^{-\frac{1}{2}})$.

The Jacob-preconditioner is useful when diagonal elements of A differ by several orders of magnitude. If all diagonal elements are roughly the same size this preconditioner has very little effect.

A simple Gauss-Seidel preconditioner is given by $P = D - L$. A factorized variation, based on symmetric GS to use with simple CG is given by

$$(D - L)^{-1}SAS(D - L^T)^{-1}\mathbf{y} = (D - L)^{-1}S\mathbf{b}, \tag{9.5.42}$$

with $S = \sqrt{D}$.

9.5.8.2 Incomplete LU factorization

An LU factorization (see previous chapter) of a large sparse matrix A can be prohibitively expensive because of the fill in, but a very powerful preconditioner can be constructed by making an *approximate* factorization that uses the sparsity structure of the matrix A or allows a very limited fill in only. It works like this. First of all we define a set of matrix coefficients that we are going to use in our incomplete factorization. Usually this set includes all coefficients of A that are non zero. In any case all *diagonal* elements of A must belong to this set. The complement of this set the *neglected set* is denoted by $\mathcal{S}$. Now an incomplete factorization follows the same procedure as a normal decomposition (see Section 9.1.2), with some important exceptions. The matrix A^0 we start the decomposition with *has zeros for indices in the neglected set* and otherwise is equal to A. In actual practice this often will not make any difference, because usually we will choose the neglected set in such a way that $A^0 = A$. Now consider the equations for the update (9.1.8). In incomplete decompositions we leave $a_{jk}^{(i)}$ unchanged if either $(j,k) \in \mathcal{S}$, $(j,i) \in \mathcal{S}$ or $(i,k) \in \mathcal{S}$. In other words, all coefficients in the update equation must belong to the complement of the neglected set, otherwise there is no update.

Exercise 9.5.22 *Show that in incomplete decomposition*

1. $\ell_{ij} = 0$, if $(i,j) \in S$,
2. $u_{i,j} = 0$, if $(i,j) \in S$.

□

The first question is under which circumstances such factorizations are useful. It turns out that for M-matrices A an incomplete factorization will create a regular splitting. This will guarantee the convergence of CG or BiCG-stab using a preconditioner like that (why?).

We first show, that if A is an M matrix, both factors L and U in the LU-factorization will be M-matrices too. We shall show this in steps. Consider the first step in the LU factorization of A:

$$A^{(1)} = L^{(1)}A, \tag{9.5.43}$$

with $L^{(1)}$ lower triangular, with ones on the diagonal and the multiplicators

$$m_{j1} = -a_{j1}/a_{11}, j = 2, \ldots, N$$

in the first column. Observe that $L^{(1)}$ is nonnegative.

Theorem 9.5.1 *If A is an M-matrix, so is $A^{(1)}$.*

Proof.
The elements of $A^{(1)}$ are given by $a_{jk}^{(1)} = a_{jk} + m_{j1}a_{1k}, j = 2, \ldots, N$ and the first row is unchanged. Apparently all off-diagonal elements of $A^{(1)}$ remain non-positive, since m_{j1} is nonnegative and a_{1k} is nonpositive for $k > 1$. So $A^{(1)}$ is still a Z-matrix.

To show that ${A^{(1)}}^{-1}$ is nonnegative we consider the solution of

$$A^{(1)}\mathbf{x}^{(k)} = \mathbf{e}_k, \tag{9.5.44}$$

where $\mathbf{e}_k$ is the k-th unit vector hence $\mathbf{x}^{(k)}$ is just the k-th column of the updated inverse. For $k = 1$ we have that

$$A^{(1)}\mathbf{x}^{(1)} = \mathbf{e}_1 \tag{9.5.45}$$

has solution $\mathbf{x}^{(1)T} = (1/a_{11}, 0, \ldots, 0)$, which is clearly nonnegative. For all other $\mathbf{e}_k$ the solution of

$$A^{(1)}\mathbf{x}^{(k)} = \mathbf{e}_k \tag{9.5.46}$$

is the same as that of

$$A\mathbf{x}^{(k)} = \mathbf{e}_k, \tag{9.5.47}$$

since the right hand side is unaffected by the Gauss step. □

Exercise 9.5.23 *Show from the proof above, that extending an M-matrix on the left with a zero column and at the top with a row with positive diagonal element and nonpositive off-diagonal generates another M-matrix. Infer from this result, that at each state of the Gaussian elimination the intermediate $A^{(k)}$ is an M-matrix if A is an M-matrix.* □

Making a diagonal element *larger* or an off diagonal element *less negative* leaves the M-property untouched, as may be readily shown.

Exercise 9.5.24 *Let $A = (a_{jk})$ be an M-matrix. Let $b_{11} = a_{11} + \alpha$ with $\alpha > 0$, and $b_{jk} = a_{jk}$ otherwise. Show that $B = (b_{jk})$ is also an M-matrix.* □

Exercise 9.5.25 *Let $A = (a_{jk})$ be an M-matrix. Let $b_{12} = a_{12} + \alpha$ with $0 \leq \alpha \leq |a_{12}|$, and $b_{jk} = a_{jk}$ otherwise. Show that $B = (b_{jk})$ is also an M-matrix.* □

This important property gives us the possibility to ignore fill-in in the elimination process, while the intermediate matrix is still an M-matrix.

Exercise 9.5.26 *Show that ${L^{(k)}}^{-1}$ is obtained by multiplying all off diagonal elements of L^k by -1. Show that ${L^{(k)}}^{-1}$is an M-matrix.* □

After $N - 1$ steps of Gaussian elimination we have:

$$U = L^{(N-1)}L^{(N-2)} \ldots L^{(1)}A, \tag{9.5.48}$$

from which we see, that $L^{-1} = L^{(N-1)}L^{(N-2)} \ldots L^{(1)}$ in the decomposition $A = LU$.

Exercise 9.5.27 *Show that if A is an M-matrix both U and L are M-matrices. Show that this remains true if part of the fill in is neglected.* □

Let us formalize this result in a theorem. We denote the index set of neglected fill in by $\mathcal{S}$. $\mathcal{S}$ cannot include diagonal elements.

Theorem 9.5.2 *Let A be an M-matrix. There exists for every index set $\mathcal{S}$ a lower triangular $\tilde{L}$ with unit diagonal. upper triangular $\tilde{U}$ and remainder N such that*

1. $\ell_{kj} = 0, u_{kj} = 0$ *if* $(k,j) \in \mathcal{S}$,
2. $n_{kj} = 0$ *if* $(k,j) \notin \mathcal{S}$,

such that $A = \tilde{L}\tilde{U} - N$. The factors $\tilde{U}$ and $\tilde{L}$ are completely determined by $\mathcal{S}$. $\tilde{L}$ and $\tilde{U}$ are both M-matrices hence $P = \tilde{L}\tilde{U}$ has nonnegative inverse. Since $N \geq 0$ the splitting is regular and generates a convergent iteration process.

A common strategy for choosing $\mathcal{S}$ is to take for the LU decomposition the same sparsity pattern as A has:

$$\mathcal{S} = \{(k,j) |\, a_{k,j} = 0\}. \tag{9.5.49}$$

Exercise 9.5.28 *Show that $\mathcal{S} = \{(k,j) \mid k \neq j\}$ leads to the Jacob preconditioner.* □

We conclude this section with an algorithmic description of the incomplete LU factorization.

ILU algorithm

Require: A is M-matrix
  Presets: $\mathcal{S}$ set of neglected updates
  for $k = 1..N-1$ **do**
    for $j = k+1..N$ **do**
      if $(j,k) \notin S$ **then**
        $\ell_{jk} = a_{jk}/a_{kk}$ {Store the multiplicator in L}
        for $m = k+1..N$ **do**
          if $(j,m) \notin S$ and $(k,m) \notin S$ **then**
            $a_{jm} = a_{jm} - \ell_{jk}a_{km}$
          end if
        end for{m}
      end if
    end for{j}
  end for{k}

Apart from the two tests whether the update should be performed at all, the ILU algorithm is the same as a normal LU decomposition algorithm.

9.6 The multigrid algorithm

The multigrid algorithm (MG) has become one of the most successful iterative techniques in the past 30 years. Its main strength is, that the operation count increases *linearly* with the number of unknowns N or in other words that the number of iterations is independent of the stepsize. Its main weakness is, that it usually needs a lot of fine-tuning before this is realized in practice. We can only give the briefest of introductions into this very interesting subject. For further information we refer the reader to [48], [18] and [40].

9.6.1 A one-dimensional example

Although MG is never applied to one-dimensional problems, the ideas behind it can perfectly well be illustrated with a one-dimensional example. We consider the one-dimensional boundary problem on the interval $(0,1)$:

$$-\frac{d^2u}{dx^2} = f, \quad u(0) = 0, u(1) = 0. \tag{9.6.1}$$

We discretize this on a grid of $N = 2^p + 1$ points to obtain a familiar set of equations:

$$\begin{aligned}
2u_1 - u_2 &= h^2 f_1 && \text{(9.6.2a)}\\
-u1 + 2u_2 + u_3 &= h^2 f_2 && \text{(9.6.2b)}\\
\vdots \qquad\qquad & \vdots \qquad\qquad\\
-u_{k-1} + 2u_k - u_{k+1} &= h^2 f_k && \text{(9.6.2c)}\\
\vdots \qquad\qquad & \vdots \qquad\qquad\\
-u_{N-3} + 2u_{N-2} - u_{N-1} &= h^2 f_{N-2} && \text{(9.6.2d)}\\
-u_{N-2} + 2u_{N-1} &= h^2 f_{N-1} && \text{(9.6.2e)}
\end{aligned}$$

or $A\mathbf{u} = h^2\mathbf{f}$ in which A is an $(N-1)\times(N-1)$ tridiagonal matrix with 2's on the diagonal and -1's on subdiagonal and super diagonal. The eigenvalues and eigen vectors of such a matrix can be calculated exactly.

Theorem 9.6.1 *Let the $(N-1)\times(N-1)$ matrix A be defined as above, The eigenvalues λ_k are given by*

$$\lambda_k = 4\sin^2\frac{k\pi}{2N}, \quad k = 1, 2, \ldots, N-1, \tag{9.6.3}$$

and the components v_{kj} of the corresponding eigenvectors $\mathbf{v}_k$ by

$$v_{kj} = \sqrt{\frac{2}{N}}\sin\frac{kj\pi}{N}. \tag{9.6.4}$$

Proof

Consider the three term recurrence relation

$$-u_{j-1} + (2-\lambda)u_j - u_{j+1} = 0, \quad u_0 = 0, u_N = 0. \tag{9.6.5}$$

From the theory of linear recurrence relations we know, that the general solution is of the form $u_k = a\rho_1^k + b\rho_2^k$, where $\rho_{1,2}$ are the solutions of the quadratic:

$$-\rho^2 + (2-\lambda)\rho - 1 = 0. \tag{9.6.6}$$

Because A is symmetric, the eigenvalues are real and from Gershgorin's theorem they should lie in the interval $(0,4)$. Therefore the discriminant of Equation (9.6.6) is negative and the roots are conjugate complex. Since $\rho_1\rho_2 = 1$ (why?) we set $\rho_1 = e^{i\phi}$ and $\rho_2 = e^{-i\phi}$. This substituted in Equation (9.6.6) gives

$$\begin{aligned}
2 - \lambda &= e^{i\phi} + e^{-i\phi},\\
\lambda &= 2 - 2\cos\phi,\\
&= 4\sin^2\frac{1}{2}\phi.
\end{aligned} \tag{9.6.7}$$

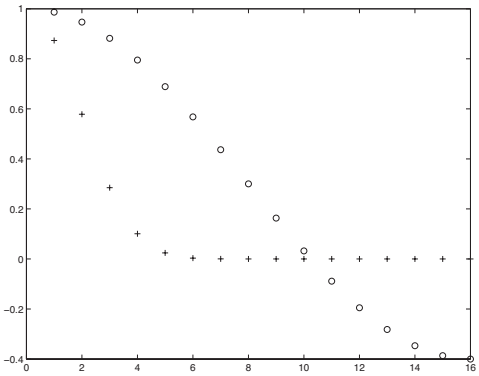

Figure 9.8: Eigenvalues 'o' and 10-th powers of eigenvalues '+'.

The general solution therefore is $u_j = ae^{ij\phi} + be^{-ij\phi}$ and since $u_0 = 0$ we get $a = -b$ or $u_j = c \sin j\phi$. Because $u_N = 0$, $\sin N\phi = 0$ or $\phi = \dfrac{k\pi}{N}$.

$c = \sqrt{2/N}$ follows from a normalization argument. See Exercise 9.6.1. □

Exercise 9.6.1 *Show that*

$$\sum_{j=1}^{N} \sin^2 \frac{jk\pi}{N} = \frac{1}{2}N. \tag{9.6.8}$$

(Hint: write $\sin\phi = (e^{i\phi} - e^{-i\phi})/(2i)$*.)* □

Exercise 9.6.2 *Show that*

$$\sum_{j=1}^{N-1} \sin \frac{jk\pi}{N} \sin \frac{j\ell\pi}{N} = 0, \quad \text{if } k \neq \ell. \tag{9.6.9}$$

□

Exercise 9.6.3 *Show that for large N the* smallest *eigenvalue of A*

$$\lambda_1 \approx \frac{\pi^2}{N^2} = \pi^2 h^2 \tag{9.6.10}$$

□

9.6.2 Smooth and rough part of the spectrum

For this example let us look at a classic iteration process called *damped Jacob.* The preconditioner is given by $P = \alpha^{-1}D$, $\alpha < 1$ and the iteration process by

$$\mathbf{c}^k = D^{-1}\mathbf{r}_k \tag{9.6.11a}$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha\mathbf{c}^k \tag{9.6.11b}$$

$$\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha A\mathbf{c}^k \tag{9.6.11c}$$

The eigenvalues μ_k of the iterationmatrix $M = I - P^{-1}A$ are given by $\mu_k = 1 - \frac{1}{2}\alpha\lambda_k$ in which λ_k the eigenvalues of A. They have been pictured in Figure 9.8.

In the same figure the eigenvalues to the power 10 have been plotted. This is what remains of the components of the error after 10 iterations. As you can

see, all components have disappeared except for those belonging to the smallest eigenvalues λ_k of the matrix A corresponding to those closest to 1 in the iteration matrix M. These eigenvalues are called the *smooth part* of the spectrum and the eigenvalues that are damped out the *rough part* for reasons that become clear when you look at Figure 9.9.

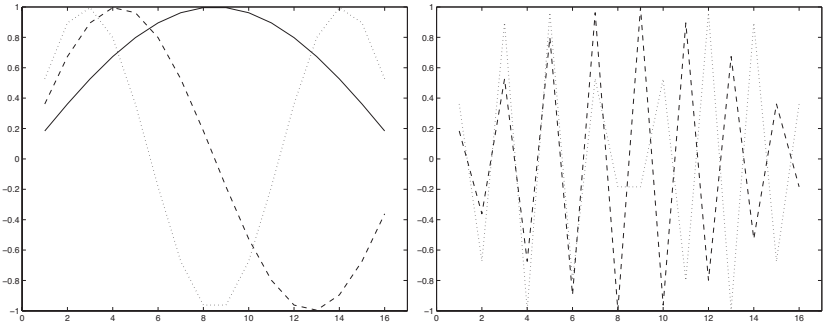

Figure 9.9: Smooth ($\mathbf{v}_{1,2,3}$) and rough ($\mathbf{v}_{15,16}$) eigenvectors.

The contribution to the error belonging to the rough part of the spectrum is annihilated very soon by the preconditioner that for that reason is called a *smoother*. The contribution belonging to the smooth part, however, is not annihilated at all and is the reason the classic iteration process converges so slowly.

9.6.3 Two grid algorithm

The central idea of the MG algorithm is to obtain the smooth parts of the solution in a different way, notably from a solution to a related problem *on a coarser grid*. Suppose we somehow had the solution in the even numbered points $(x_0, x_2, .., x_{16})$, $(u_0, u_2, ...u_{16})$. We could obtained an initial estimate by linear interpolation: $u^0_{2k+1} = (u_{2k} + u_{2k+2})/2$. Since the error in this estimate is 0 in the even points and has the same sign in the odd points as the second derivative it looks something like Figure 9.10.

Apparently this error has a large component in the rough part of the spectrum, exactly where our smoother is most effective. This is the central idea of the *two grid algorithm*. We define a *coarse grid* $\mathcal{G}_H : x_0, x_2, x_4, x_{2k} \ldots x_N$ and a fine grid $\mathcal{G}_h : x_0, x_1, \ldots, x_N$. Our original problem $A_h\mathbf{u}_h = \mathbf{f}_h$ lives on the fine grid. On the coarse grid we can calculate the solution to a related problem $A_H\mathbf{u}_H = \mathbf{f}_H$.

Apparently there must be a mapping from fine grid to coarse grid $R_{Hh}\mathbf{f}_h = \mathbf{f}_H$. This mapping is called the *restriction* in MG speak. And the interpolation operator to get from $\mathbf{u}_H$ to $\mathbf{u}_h$ is called the *prolongation* $P_{hH}\mathbf{u}_H = \mathbf{u}_h$. The prolongation operator is easy in this case. It is the matrix

$$\begin{pmatrix} \frac{1}{2} & 0 & \ldots & \ldots & \ldots & \ldots \\ 1 & 0 & \ldots & \ldots & \ldots & \ldots \\ \frac{1}{2} & \frac{1}{2} & \ldots & \ldots & \ldots & \ldots \\ 0 & 1 & 0 & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & 0 & 1 & 0 \\ \ldots & \ldots & \ldots & \ldots & \frac{1}{2} & \frac{1}{2} \\ \ldots & \ldots & \ldots & \ldots & 0 & 1 \\ \ldots & \ldots & \ldots & \ldots & 0 & \frac{1}{2} \end{pmatrix} \tag{9.6.12}$$

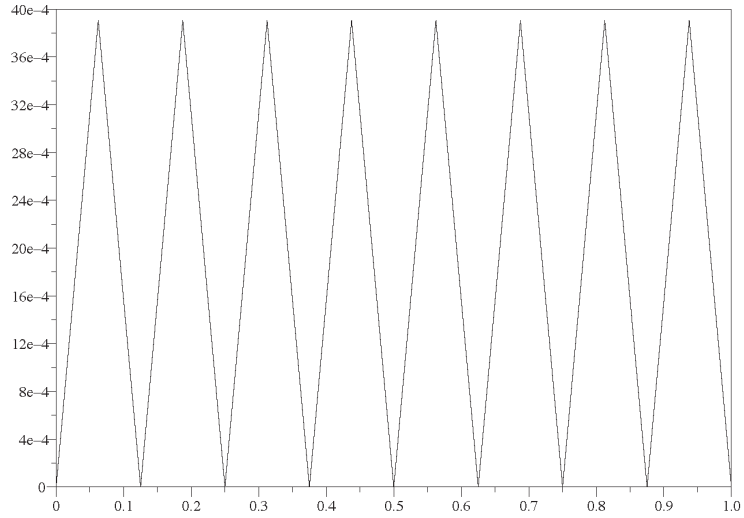

Figure 9.10: Typical error by interpolation.

Since we basically try to solve a set of equations in a space of reduced dimension an obvious strategy is to apply the Galerkin method, in other words take $R = P^T$. Other choices are possible too, and come down to Petrov Galerkin.

Exercise 9.6.4 *Show, that $A_H = P_{hH}^T A_h P_{hH}$ is a tridiagonal matrix with 1's on the diagonal and $-\frac{1}{2}$'s on the sub and super diagonal. Show that $f_{H,k} = \frac{1}{2} f_{h,2k-1} + f_{h,2k} + \frac{1}{2} f_{h,2k+1}$.* □

We present the two grid algorithm in algorithmic form:
Two grid algorithm

1: Presets: $\mathbf{u}_h^0$, $\mathbf{r}_h^0 = \mathbf{f}_h - A\mathbf{u}_h^0$
2: $\mathbf{u}_h^{\text{prs}} = S(\mathbf{u}_h^0, \mathbf{b}, A, n_0)$ {Presmoothing}
3: $\mathbf{r}_H = R_{Hh}\mathbf{r}_h$
4: Solve $A_H \mathbf{c}_H = \mathbf{r}_H$
5: $\mathbf{u}_h^{\text{cgc}} = \mathbf{u}_h^p + P_{hH}\mathbf{c}_H$ {Coarse Grid Correction}
6: $\mathbf{u}_h^{\text{pos}} = S(\mathbf{u}_h^{\text{cgc}}, \mathbf{b}, A, n_1)$ {Postsmoothing}

So the two grid algorithm consist of a pre-smoothing stage, a coarse grid correction and a post-smoothing stage. The notation $S(\mathbf{u}_h^0, \mathbf{b}, A, n)$ means: do n steps with the preferred smoother, with initial estimate $\mathbf{u}_h^0$, right hand side $\mathbf{b}$ and matrix A.

To get an impression how effective the two grid algorithm is, we look at the contributions of the rough and smooth spectra to the original error and the error after coarse grid correction. Only odd modes are shown, because the even modes happen to vanish in this specific example ($f(x) = 1$). See Table 9.11. As you can see,

	smooth				rough			
mode	1	3	5	7	9	11	13	15
initial $\times 10^{-4}$	2580	96	21	7	3	2	1	0
after corr $\times 10^{-4}$	25	9	6	5	5	6	9	25

Figure 9.11: Error reduction by coarse grid correction.

the coarse grid correction action is complementary to that of the smoother. The

contribution of the smooth part of the spectrum is reduced very rapidly, whereas the rough part is increased. Presmoothing is not very efficient in this particular case, since the contribution of the rough component to the initial error is almost negligible.

Exercise 9.6.5 *Explain why we use* damped *Jacob as smoother and not original vintage Jacob. (Hint: consider Figure 9.7)* □

9.6.4 From two grid to multigrid

The *Multigrid* algorithm consists of applying the two grid algorithm recursively to step 4. To solve the coarse grid problem $A_H\mathbf{c}_H = \mathbf{r}_H$, we define an even coarser grid
$x_0, x_4, x_8, x_{12}, x_{16}$, and because we run out of fonts in which to print H to express the fact that this is an even coarser grid it is maybe a good idea to let the notation reflect the *coarsening level*. Let our finest grid be coarsening level 0 and let each application coarsening increase the level by 1. So in level 0 we have $2^p + 1$ points, in level 1 $2^{p-1} + 1$ points, in level k $2^{p-k} + 1$ points. We denote the matrix and vectors on level ℓ with A_ℓ and $\mathbf{v}_\ell$ respectively. R_ℓ and P_ℓ operate from this level to the next higher and lower levels respectively. If we have three points left in our grid we must stop, because we only have one unknown left. The other two are boundaries. Hence we should stop at level $p-1$ or earlier. At level $p-1$ we can solve the problem $A_{p-1}\mathbf{c}_{p-1} = \mathbf{r}_{p-1}$ directly. This gives us the following algorithm:
MGRecursive $(A_\ell, \mathbf{r}_\ell, \mathbf{c}_\ell, \ell)$
  if $\ell < p-1$ **then**
    $\mathbf{c}_\ell = S(\mathbf{0}, \mathbf{r}_\ell, A_\ell, n_0)$ {Presmoothing}
    $\mathbf{r}_{\ell+1} = R_\ell(\mathbf{r}_\ell - A_\ell\mathbf{c}_\ell)$ {Calculate coarse grid residual}
    $A_{\ell+1} = R_\ell A_\ell P_{\ell+1}$ {Calculate coarse grid matrix}
    call MGRecursive $(A_{\ell+1}, \mathbf{r}_{\ell+1}, \mathbf{c}_{\ell+1}, \ell+1)$
    $\mathbf{c}_\ell = \mathbf{c}_\ell + P_{\ell+1}\mathbf{c}_{\ell+1}$ {Coarse grid correction}
    $\mathbf{c}_\ell = S(\mathbf{c}_\ell, \mathbf{r}_\ell, A_\ell, n_1)$ {Postsmoothing}
  else
    Solve $A_{p-1}\mathbf{c}_{p-1} = \mathbf{r}_{p-1}$ {Direct solution on coarsest level}
  end if

For clearness of presentation the calculation of the coarse grid matrix has been put into the algorithm. This is definitely not a good idea in practice, because the algorithm will be used several times and these coarse grid operators do not change. It is a better idea, to do a preliminary stage in which the restriction, prolongation and matrix on all levels are calculated and stored.

The analysis of the multigrid algorithm is far more involved than that of the two grid algorithm, but the analysis remains qualitatively valid.

Exercise 9.6.6 *Explain, that if $n_0 = 0$ (no presmoothing), the right hand side on the coarsest level is given by*

$$\mathbf{r}_{p-1} = R_{p-2}R_{p-3}\dots R_1R_0\mathbf{r}_0. \tag{9.6.13}$$

Exercise 9.6.7 *Calculate the right hand side on the coarsest grid for $p = 3$ and $f(x) = 1$. (No presmoothing). Calculate the matrix on the coarsest grid for the model problem.*

9.6.5 Convergence of the two grid algorithm

For ease of presentation we consider the two grid algorithm with post smoothing only. It consists of two steps, the coarse grid correction and the postsmoothing. We

will first study the effect coarse grid correction. Let us call the coarse grid space Σ_H and the fine grid space Σ_h. The example matrix A is positive definite so the problem to solve in Σ_h is:

$$\min_{\mathbf{c}_h \in \Sigma_h} \frac{1}{2}(\mathbf{c}_h, A\mathbf{c}_h) - (\mathbf{c}_h, \mathbf{r}) \tag{9.6.14}$$

but instead we solved:

$$\min_{\mathbf{c}_H \in \Sigma_H} \frac{1}{2}(P_{hH}\mathbf{c}_H, AP_{hH}\mathbf{c}_H) - (P_{hH}\mathbf{c}_H, \mathbf{r}) \tag{9.6.15}$$

and after that put $\tilde{\mathbf{c}}_h = P_{hH}\mathbf{c}_H$. So we approximated the solution on a subspace but on that subspace we obtained the best possible solution in the sense that it minimizes the quadratic form (9.6.15). In particular, if $\hat{\mathbf{c}}_H$ is the exact solution in the coarse grid points you would still have, from Equation (9.6.15) that

$$\frac{1}{2}(\tilde{\mathbf{c}}, A\tilde{\mathbf{c}}) - (\tilde{\mathbf{c}}, \mathbf{r}) \leq \frac{1}{2}(P_{hH}\hat{\mathbf{c}}_H, AP_{hH}\hat{\mathbf{c}}_H) - (P_{hH}\hat{\mathbf{c}}_H, \mathbf{r}). \tag{9.6.16}$$

And since $\mathbf{r} = A\hat{\mathbf{c}}$ this transforms, after adding $\frac{1}{2}(\hat{\mathbf{c}}, A\hat{\mathbf{c}})$ to both sides into

$$\frac{1}{2}(\tilde{\mathbf{c}} - \hat{\mathbf{c}}, A(\tilde{\mathbf{c}} - \hat{\mathbf{c}})) \leq \frac{1}{2}(P_{hH}\hat{\mathbf{c}} - \hat{\mathbf{c}}, AP_{hH}(\hat{\mathbf{c}} - \hat{\mathbf{c}})). \tag{9.6.17}$$

Hence

$$\|\tilde{\mathbf{c}} - \hat{\mathbf{c}}\|_A \leq \|P_{hH}\hat{\mathbf{c}} - \hat{\mathbf{c}}\|_A. \tag{9.6.18}$$

This very interesting result (analogous to the Finite Element error estimate) tells us, that the error after the coarse grid correction is no worse than the error in linear interpolation of the exact solution *measured in the* $\|.\|_A$ *norm.*

Let us have a look at the residual after coarse grid correction:

$$\tilde{\mathbf{r}} = \mathbf{r} - A_h P_{hH} A_H^{-1} R_{Hh}\mathbf{r}, \tag{9.6.19}$$

$$= (I - A_h P_{hH} A_H^{-1} R_{Hh})\mathbf{r} = Q\mathbf{r}. \tag{9.6.20}$$

Exercise 9.6.8 *Show that* $Range(Q) = Ke(R_{Hh})$. *Infer from this that repeated coarse grid corrections are useless without intermediate smoothing.* □

Finally we shall look in detail to the reduction of the various modes in the spectrum. We shall show that the smooth part of the spectrum is reduced by the coarse grid correction by a fixed amount, independent of the stepsize h. We already know that the rough part of the spectrum is reduced by a fixed amount, irrespective of the stepsize. Together we conclude that one twogrid iteration reduces the error by a fixed amount, irrespective of the stepsize. This means that we need only a fixed number of iterations to get to a certain accuracy *and that the total number of operations is only dependent on the number of operations per iteration*.

The analysis we are about to perform is made easy by the fact, that the smooth part of the eigenvectors of the coarse grid corrector Q and the damped Jacob smoother S are the same: $v_{kj} = \sin jk\pi/N$. Usually this is not the case and this makes convergence analysis a lot harder. For a more general treatment of convergence properties of the multigrid algorithm we refer the reader to [48], [18] and [40].

We perform the analysis in several steps in a number of exercises.

Exercise 9.6.9 *Let A_h be an $N_h - 1 \times N_h - 1$ matrix, $N_h = 2^{p+1}$, with diagonal $2N_h$ and sub and superdiagonals $-N_h$. Show that A_H is an $N_H - 1 \times N_H - 1$ matrix with $N_H = 2^p$ with diagonal $2N_H$ and sub and super diagonals $-N_H$.*

Calculate the eigenvalues of A_H and A_h using Theorem 9.6.1. Calculate the eigenvectors too. □

Exercise 9.6.10 *Show that the smooth part of the eigenvectors of A_h (i.e. $k < N_H$) is completely represented in the eigenvectors of A_H, if you realize that $\mathbf{v}_{Hk}$ contain only the even vectorcomponents of $\mathbf{v}_{hk}$. For instance $\mathbf{v}_{h1}$ has components $\sin j\pi/N_h$ and the even components of that vector just make up $\mathbf{v}_{H1}$ with components $\sin j\pi/N_H = \sin 2j\pi/N_h$.* □

The rough part of the eigenvectors of A_h (i.e. $k \geq N_H$) is not easily represented by eigenvectors with $k_H = k_h \mod N_H$. For instance $\mathbf{v}_{hN_H+1}$ has components $\sin((N_H+1)j\pi/N_h)$ and the even components of that vector just make up

$$\begin{aligned}\sin((N_H+1)2j\pi/N_h) &= \sin((N_H+1)j\pi/N_H),\\ &= \sin(j\pi + j\pi/N_H),\\ &= (-1)^j \sin j\pi/N_H.\end{aligned} \tag{9.6.21}$$

Exercise 9.6.11 *Show that for eigenvectors $\mathbf{v}_{hk}$ of A_h belonging to the smooth spectrum*

$$R_{Hh}\mathbf{v}_{hk} = (1 + cosk\pi/N_h)\mathbf{v}_{Hk}. \tag{9.6.22}$$

□

Exercise 9.6.12 *Show, that for eigenvectors $\mathbf{v}_{Hk}$ of A_H:*

$$P_{hH}\mathbf{v}_{Hk} = \mathbf{v}_{hk} + \text{rough part}, \tag{9.6.23}$$

in which the rough part has even components 0 and odd components

$$v_{k,2j-1} = (1 - \cos k\pi/N_h)\sin k(2j-1)\pi/N_h \tag{9.6.24}$$

□

From Exercises 9.6.9, 9.6.10 9.6.11 and 9.6.12, we conclude, that $Q\mathbf{v}_{hk}$ with k belonging to the smooth part of the spectrum has a smooth spectrum component of:

$$Q\mathbf{v}_{hk} = \left(1 - (1 + \cos k\pi/N)\frac{\lambda_{hk}}{\lambda_{Hk}}\right)\mathbf{v}_{hk} \tag{9.6.25}$$

Now $\lambda_{hk} = 4N_h sin^2 k\pi/(2N_h)$ and $\lambda_{Hk} = 4N_H sin^2 k\pi/(2N_H)$ and after short manipulation:

$$\begin{aligned}2(1 + cos\phi)\frac{\sin^2\phi/2}{\sin^2\phi} &= 2(2 - 2sin^2\phi/2)\frac{\sin^2\phi/2}{\sin^2\phi}\\ &= \frac{4\cos^2\phi/2\sin^2\phi/2}{sin^2\phi}\\ &= 1\end{aligned} \tag{9.6.26}$$

in other words, the smooth part of the spectrum is completely annihilated by the coarse grid correction. That is not entirely true, by the way, because there is some crossover between rough and smooth components.

9.6.6 Restriction and prolongation in two dimensions

The one-dimensional example is in a way special, because many aspects of the algorithm can be calculated exactly. Nevertheless, the main components of the algorithm remain true in more dimensions. The only thing that needs special attention are the restriction and prolongation operators. A straightforward generalization to 2 and 3 dimensions would be bi- or trilinear interpolation and that would fit the bill just fine, but for a small problem.

Let us assume we are working on a rectangular region with 2^p cells in x-direction and 2^q cells in y-direction. We define the bilinear restrictions and prolongations as follows: $P = P_x P_y$ in which P_x is the one-dimensional prolongation operator in x-direction applied to each single row of the region and P_y is is the one-dimensional prolongation operator in y-direction applied to each single column.

Exercise 9.6.13 *Express P_x and P_y in matrix form. Show that P_x and P_y commute.* □

We take again $R = P^T$. However, if you calculate the coarse grid operator A_H from a fine grid operator A_h coming from the five point Laplace molecule, you will see, that the foot print increases from a five point to a nine point molecule. Fortunately, on going to even coarser grids the foot print does not increase.

Exercise 9.6.14 *Show this. Show also that in 3 dimensions the foot print increases from a 7-point molecule to a 27-point molecule.* □

9.6.7 Concluding remarks about MG

There is a vast amount of literature on the subject of MG algorithms. [48] and [18] are classics that are recently reprinted, [40] is recent. All contain pointers to publications that may be of further interest.

A couple of remarks is in order:

1. Damped Jacob is a good smoother, but not the only one. Gauss Seidel is also good and the incomplete *LU* factorizations are very good.
2. The use of powers of 2 as number of cells is widely spread, but not really necessary. The algorithm has been successfully applied for any number of grid points.
3. There are other variations that we have not treated in this short exposition. Most notably other interpolation strategies (cell centered versus vertex centered) and recursion strategies (F-, V- and W cycles) have not been covered. We only have shown a simple vertex centered V-cycle.
4. One multigrid cycle is really a *preconditioner* that can be used with defect-correction or Bi-CGStab. (It is hard to use with CG because of the symmetry requirement). The latter choice is by far the best.
5. There are still unsolved problems with MG in unstructured grids or rapidly changing coefficients. Also applications to 3D problems have still some unchartered waters.

9.7 Non-linear equations

The discretization of non-linear PDEs leads to non-linear algebraic equations. Although many methods to solve non-linear algebraic system are available in the

mathematical literature, we will only treat two classical iterative processes: *Picard iteration* and *Newton iteration*. These two methods usually respectively exhibit linear and quadratic convergence.

9.7.1 Picard iteration

First we consider a class of problems that are small perturbations of linear problems. For instance

$$-\text{div grad}\, u + f(u) = 0, \quad \text{on } \Omega, \tag{9.7.1}$$

and $u = 0$ on Γ. If you discretize this the standard way, you end up with a set of equations of the form

$$A\mathbf{u} + \mathbf{f}(\mathbf{u}) = 0, \tag{9.7.2}$$

in which $f_k(\mathbf{u}) = f(u_k)$. To approximate the solution of the above equation, we generate an array $\mathbf{u}^n$ with the goal that $\mathbf{u}^n \to \mathbf{u}$ as $n \to \infty$. The estimates $\mathbf{u}^n$ are obtained by solving a *linear* system of equations. Since we are only able to solve linear problems as $A\mathbf{u} = \mathbf{b}$, a natural way to go about this is to start out with an initial estimate $\mathbf{u}^0$ and solve the following iteratively:

$$A\mathbf{u}^{n+1} = -\mathbf{f}(\mathbf{u}^n). \tag{9.7.3}$$

Such an iterative process is known as *Picard iteration*.

Exercise 9.7.1 *Show that if* $\mathbf{u}$ *is the solution of* (9.7.2) *and* $\varepsilon^n = \mathbf{u} - \mathbf{u}^n$, *with* $\mathbf{u}^n$ *solution of* (9.7.3) *that*

$$A\varepsilon^{n+1} = D(\mathbf{u})\varepsilon^n + O(\|\varepsilon^n\|^2), \tag{9.7.4}$$

in which D *is a diagonal matrix with* $d_{kk}(\mathbf{u}) = -f'(u_k)$. *Show that this process cannot converge if at least one eigenvalue of* $A^{-1}D$ *is larger than 1 in absolute value.* □

An other example concerns the case of an elliptic equation in which the coefficients depend on the solution u. Let us consider the following equation

$$-\text{div}\ (D(u)\text{grad}\, u) = f(\mathbf{x}). \tag{9.7.5}$$

If $D(u)$ is not a constant, for instance $D(u) = u$, then the above equation is nonlinear. To solve the above equation, we generate a sequence of approximations u^n as in the previous example. Here the above equation is solved by iterating

$$-\text{div}\ \left(D(u^n)\text{grad}\, u^{n+1}\right) = f(\mathbf{x}). \tag{9.7.6}$$

After construction of an appropriate discretization, a linear system to obtain u^{n+1} has to be solved. In general if one wants to solve a nonlinear problem using Picard's method, convergence is not always guaranteed. One needs to use commonsense to solve the problem.

So a natural way to obtain an iterative process to a non-linear set of equations $\mathbf{f}(\mathbf{x}) = 0$ is to reform it to a *fixed point form* $\mathbf{x} = G(\mathbf{x})$ with the same solution. On this fixed point form you graft an iterative process:

$$\mathbf{x}^{k+1} = G(\mathbf{x}^k) \tag{9.7.7}$$

There is a famous convergence result due to Banach on such processes.

Theorem 9.7.1 *Let* $\mathcal{D}$ *be a closed subset of* $\mathbb{R}^n$ *and let* G *be a mapping* $\mathbb{R}^n \to \mathbb{R}^n$ *such that*

1. *if* $\mathbf{x} \in \mathcal{D}$ *then* $G(\mathbf{x}) \in \mathcal{D}$
2. $\|G(\mathbf{x}) - G(\mathbf{y})\| \leq \alpha\|\mathbf{x} - \mathbf{y}\|$ *for all* $\mathbf{x}, \mathbf{y}, \in \mathcal{D}$, $\alpha < 1$

then $\mathcal{D}$ *contains precisely one fixed point of* G.

Proof
Choose $\mathbf{x}^0 \in \mathcal{D}$. By elementary induction it will be clear, that the whole sequence generated by $\mathbf{x}^{k+1} = G\mathbf{x}^k$ is in $\mathcal{D}$. Apparently $\|\mathbf{x}^{k+1} - \mathbf{x}^k\| = \|G(\mathbf{x}^k) - G(\mathbf{x}^{k-1})\| \leq \alpha\|\mathbf{x}^k - \mathbf{x}^{k-1}\|... \leq \alpha^k\|\mathbf{x}^1 - \mathbf{x}^0\|$. Hence the sequence converges to a limit which lies in $\mathcal{D}$ because $\mathcal{D}$ is closed.

There cannot be two different fixed points ξ and η. If there were, $\|G(\xi) - G(\eta)\| \leq \alpha\|\xi - \eta\|$, which is clearly impossible, since $G(\xi) = \xi$ and $G(\eta) = \eta$. □

A mapping that satisfies the conditions of Theorem 9.7.1 is called a *contraction* or a *contractive mapping*.

Exercise 9.7.2 *Let* $\mathbf{x}^{k+1} = G(\mathbf{x}^k)$ *be an iterative process with limit* ξ. *G has continuous partial derivatives in a neighborhood D of* ξ *and* $\|G'(\mathbf{x}))\| < 1, \mathbf{x} \in D$. G' *is the matrix with*

$$g'_{kj} = \frac{\partial g_k}{\partial x_j}. \tag{9.7.8}$$

Show that D contains a subset on which G is a contraction. □

9.7.2 Newton's method in more dimensions

In order to find a faster converging solution process to the set of non linear equations

$$\mathbf{f}(\mathbf{x}) = 0, \quad \mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^n \tag{9.7.9}$$

we try to find an analogue to Newton's method for functions of one variable:

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}. \tag{9.7.10}$$

In the neighborhood of the root ξ we have by Taylors theorem:

$$0 = f(\xi) = f(x) + (\xi - x)f'(x) + O((\xi - x)^2), \tag{9.7.11}$$

for functions of one variable. We arrive at Newton's formula by neglecting the second order term. We try something similar in n dimensions. In the neighborhood of the root ξ we have:

$$0 = f_1(\xi) = f_1(\mathbf{x}) + \sum_{j=1}^{n}(\xi_j - x_j)\frac{\partial f_1}{\partial x_j}(\mathbf{x}) + O(\|\xi - \mathbf{x}\|^2), \tag{9.7.12a}$$

$$0 = f_2(\xi) = f_2(\mathbf{x}) + \sum_{j=1}^{n}(\xi_j - x_j)\frac{\partial f_2}{\partial x_j}(\mathbf{x}) + O(\|\xi - \mathbf{x}\|^2), \tag{9.7.12b}$$

$$\vdots$$

$$0 = f_n(\xi) = f_n(\mathbf{x}) + \sum_{j=1}^{n}(\xi_j - x_j)\frac{\partial f_n}{\partial x_j}(\mathbf{x}) + O(\|\xi - \mathbf{x}\|^2). \tag{9.7.12c}$$

Neglecting the second order term Equations (9.7.12) we arrive at an iteration process that is analogous to (9.7.10):

$$f_1(\mathbf{x}^k) + \sum_{j=1}^{n} (x_j^{k+1} - x_j^k) \frac{\partial f_1}{\partial x_j}(\mathbf{x}^k) = 0, \tag{9.7.13a}$$

$$f_2(\mathbf{x}^k) + \sum_{j=1}^{n} (x_j^{k+1} - x_j^k) \frac{\partial f_2}{\partial x_j}(\mathbf{x}^k) = 0, \tag{9.7.13b}$$

$$\vdots$$

$$f_n(\mathbf{x}^k) + \sum_{j=1}^{n} (x_j^{k+1} - x_j^k) \frac{\partial f_n}{\partial x_j}(\mathbf{x}^k) = 0. \tag{9.7.13c}$$

We can put this into vector notation:

$$f'(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k) = -\mathbf{f}(\mathbf{x}^k), \tag{9.7.14}$$

where $f'(\mathbf{x})$ is the Jacobian matrix

$$f'(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} (\mathbf{x}). \tag{9.7.15}$$

We now present the algorithmic form.

Newton's method for multivariate functions

1: Presets: $\mathbf{x}^0$ {initial estimate}, $\mathbf{r}^0 = \mathbf{f}(\mathbf{x}^0)$, $k = 0$
2: **while** $\|\mathbf{r}^k\| > \varepsilon$ **do**
3: Solve $f'(\mathbf{x}^k)\mathbf{c}^k = -\mathbf{r}^k$
4: $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{c}^k$
5: $\mathbf{r}^{k+1} = \mathbf{f}(\mathbf{x}^{k+1})$
6: $k = k + 1$
7: **end while**

The calculation of the Jacobian is often very time consuming and various schemes have been proposed to improve on that. For the solution of the linear system on line 3 we can use any type of solver. The structure of the Jacobian often has the same sparsity pattern as the corresponding linearization of the PDE.

Example 9.7.1 *We consider the following differential equation in one spatial dimension, with boundary conditions:*

$$u(1-u)\frac{d^2u}{dx^2} + x = 0, \qquad u(0) = u(1) = 0. \tag{9.7.16}$$

A finite difference discretization, with equidistant grid-spacing h and n unknowns ($h = 1/(n+1)$), gives

$$f_i(\mathbf{u}) = u_i(1-u_i)\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} + x_i = 0, \text{ for } i \in \{1, \ldots, n\}. \tag{9.7.17}$$

Note that for $i = 1$ and $i = n$, the boundary conditions are used. This system of n equations with n unknowns is seen as a system of non-linear equations. Using the Picard fixed point or Newton method requires an initial guess for the solution. This initial guess could be chosen by solving the linearized system or by choosing a vector that reflects the

values at a Dirichlet boundary (if there is any). Let $\mathbf{u}^k$ *represent the solution at the k-th iterate, then, one way of using the Picard fixed point method is the following:*

$$u_i^k(1-u_i^k)\frac{u_{i-1}^{k+1}-2u_i^{k+1}+u_{i+1}^{k+1}}{h^2}+x_i=0, \text{ for } i\in\{1,\dots,n\}. \tag{9.7.18}$$

This requires the solution of a system of linear equation at each iterate.

If one prefers to use the Newton method, then, the calculation of the Jacobian matrix is necessary. Considering the i-th row of the Jacobian matrix, all entries are zero, except the one on and the ones adjacent to the main diagonal, that is

$$\begin{aligned}
\frac{\partial f_i}{\partial u_{i-1}}(\mathbf{u}^k) &= \frac{u_i^k(1-u_i^k)}{h^2},\\
\frac{\partial f_i}{\partial u_i}(\mathbf{u}^k) &= \frac{2u_i^k(1-u_i^k)}{h^2}+(1-2u_i^k)\frac{u_{i-1}^k-2u_i^k+u_{i+1}^k}{h^2},\\
\frac{\partial f_i}{\partial u_{i+1}}(\mathbf{u}^k) &= \frac{u_i^k(1-u_i^k)}{h^2}.
\end{aligned} \tag{9.7.19}$$

The rest of the procedure is straightforward. □

Exercise 9.7.3 *Consider the discretization of*

$$-\operatorname{div}\operatorname{grad} u+\mathrm{e}^u=0, \tag{9.7.20}$$

on the square $(0,1)\times(0,1)$. *Calculate* $f'(\mathbf{u})$. *Compare the structure of the Jacobian to the matrix generated by the discretization of the Laplacian.* □

Exercise 9.7.4 *Consider the discretization of*

$$\operatorname{div}\left(\frac{\operatorname{grad} u}{\sqrt{1+u_x^2+u_y^2}}\right)=0, \tag{9.7.21}$$

on the square $(0,1)\times(0,1)$ *by finite volume method. What is the sparsity structure of* $f'(\mathbf{x})$*?* □

9.7.3 Starting values

Although Newton's method converges quadratically in a neighborhood of the root, convergence is often very sensitive to good initial estimates. These are suggested sometimes by the technical context, but if obtaining an initial estimate appears to be a problem the following trick, known as *homotopy method* may be applied.

Suppose the solution to some other problem, say $\mathbf{g}(\mathbf{x})=0$ is known (e.g. a linearization of the original). Consider the following set of problems:

$$(1-\lambda)\mathbf{g}(\mathbf{x})+\lambda\mathbf{f}(\mathbf{x})=0,\quad \lambda\in(0,1). \tag{9.7.22}$$

For $\lambda=1$ we have our original problem, for $\lambda=0$ we have our auxiliary problem. Now the idea is to proceed in small steps h from $\lambda_0=0,\lambda_1=h,\lambda_2=2h$ to $\lambda_N=Nh=1$, using Newton's method as solver and always taking the solution to the problem with λ_k as initial estimate to the problem with λ_{k+1}. This is an expensive method but somewhat more robust than simple Newton.

9.8 Summary of Chapter 9

In this chapter we have studied methods to solve linear and non-linear sets of equations. Direct methods are important particularly for not too large two dimensional problems. In general an LU-decomposition is used. For a structured grid a band method is optimal, but for unstructured girds, profile methods, generally require less memory and computing time.

Renumbering techniques, like Cuthill-McKee reduce the size of the matrix to an almost optimal one.

Iterative methods become important for large problems, where direct methods may be too expensive or do not fit into memory. We first looked at *defect correction* or *standard iteration methods* like Jacob, Gauss Seidel and Successive Overrelaxation. After that we met *Krylov space methods* like Conjugate Gradients (CG) and BiCG-stab. We found that the standard methods could be used as *preconditioner*. We also met a more powerful preconditioner *incomplete LU factorization*.

We learned about the Multigrid algorithm, how its convergence is independent of the stepsize of the approximation, by using a *coarse grid correction* to get rid of the smooth part of the spectrum.

We briefly looked at non linear problems and met simple Picard iteration and a generalization of *Newton's method* to $\mathbb{R}^n$. The *homotopy* method can be used to find a starting value if all other inspiration fails.

Chapter 10

The heat- or diffusion equation

Objectives

In this chapter several numerical methods to solve the heat equation are considered. Since this equation also describes diffusion, the equation is referred to as the diffusion equation. The equation describes very common processes in physics and engineering and we would like our numerical models to inherit certain properties of the physics. The most important aspect - and typical for diffusion equations - is the property that the solution tends to an equilibrium solution as time proceeds. If the coefficients in the heat equation and the boundary conditions do not depend on time, there exists exactly one equilibrium solution, and the solution of the heat equation tends to this equilibrium solution independent of the initial condition.

10.1 A fundamental inequality

The next theorem states this result more precisely.

Theorem 10.1.1 *Let Ω be a bounded domain in $\mathbb{R}^2$, let Δ be given by*

$$\Delta = \text{div grad } = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \tag{10.1.1}$$

Let $u_E(\mathbf{x})$ be the solution of

$$\Delta u + f(\mathbf{x}) = 0, \tag{10.1.2}$$

with boundary conditions

$$u(\mathbf{x}) = g_1(\mathbf{x}), \quad \mathbf{x} \in \Gamma_1 \tag{10.1.3}$$

$$\frac{\partial u}{\partial n}(\mathbf{x}) = g_2(\mathbf{x}), \quad \mathbf{x} \in \Gamma_2 \tag{10.1.4}$$

$$(\sigma u)(\mathbf{x}) + \frac{\partial u}{\partial n}(\mathbf{x}) = g_3(\mathbf{x}), \quad \mathbf{x} \in \Gamma_3 \tag{10.1.5}$$

Further, let $u(\mathbf{x}, t)$ be the solution of the initial value problem

$$\frac{\partial u}{\partial t} = \Delta u + f(\mathbf{x}), \tag{10.1.6}$$

with initial condition $u(\mathbf{x}, t_0) = u_0(\mathbf{x})$ and boundary conditions (10.1.3)–(10.1.5). Let $R(t)$ be the quadratic residual, which is

$$R(t) = \int_\Omega \left(u(\mathbf{x}, t) - u_E(\mathbf{x})\right)^2 d\Omega, \tag{10.1.7}$$

then, there is a $\gamma > 0$ *such that*

$$R(t) < R(t_0)e^{-\gamma(t-t_0)}, \quad \forall t > t_0. \tag{10.1.8}$$

Proof
Apparently, u_E is a solution of (10.1.6) with $\partial u_E/\partial t = 0$. The difference $v = u_E - u$ satisfies:

$$\frac{\partial v}{\partial t} = \Delta v, \tag{10.1.9}$$

with initial condition $v(\mathbf{x}, t_0) = v_0 = u_E - u_0$ and boundary conditions

$$v(\mathbf{x}) = 0, \quad \mathbf{x} \in \Gamma_1 \tag{10.1.10}$$

$$\frac{\partial v}{\partial n}(\mathbf{x}) = 0, \quad \mathbf{x} \in \Gamma_2 \tag{10.1.11}$$

$$(\sigma v)(\mathbf{x}) + \frac{\partial v}{\partial n}(\mathbf{x}) = 0, \quad \mathbf{x} \in \Gamma_3. \tag{10.1.12}$$

Multiplication of equation (10.1.9) by v and subsequent integration over Ω, gives

$$\int_\Omega v\frac{\partial v}{\partial t}\, d\Omega = \int_\Omega v\Delta v\, d\Omega \tag{10.1.13}$$

$$\int_\Omega \frac{1}{2}\frac{\partial v^2}{\partial t}\, d\Omega = -\int_\Omega \|\text{grad}\, v\|^2\, d\Omega + \int_\Gamma v\frac{\partial v}{\partial n}\, d\Gamma. \tag{10.1.14}$$

Here the right-hand side follows from Green's Theorem 1.3.12. We interchange the order of integration over Ω, differentiate with respect to time and apply the boundary conditions to get:

$$\frac{1}{2}\frac{dR}{dt} = -\int_\Omega \|\text{grad}\, v\|^2\, d\Omega - \int_{\Gamma_3} \sigma v^2\, d\Gamma. \tag{10.1.15}$$

According to Poincaré's Lemma [1], (provided $\Gamma \neq \Gamma_2$), there exists a $\gamma_0 > 0$ such that

$$\int_\Omega \|\text{grad}\, v\|^2\, d\Omega > \gamma_0 \int_\Omega v^2\, d\Omega = \gamma_0 R. \tag{10.1.16}$$

Letting $\gamma = 2\gamma_0$, we obtain:

$$\frac{dR}{dt} < -\gamma R, \tag{10.1.17}$$

hence

$$\frac{dR}{dt} + \gamma R < 0. \tag{10.1.18}$$

This inequality holds for all $t > t_0$. We multiply this equation by $e^{\gamma t}$ to get

$$e^{\gamma t}\left(\frac{dR}{dt} + \gamma R\right) = \frac{d\left(e^{\gamma t}R\right)}{dt} < 0. \tag{10.1.19}$$

After integration from t_0 to t this yields

$$e^{\gamma t}R(t) - e^{\gamma t_0}R(t_0) < 0, \tag{10.1.20}$$

hence

$$R(t) < e^{-\gamma(t-t_0)}R(t_0). \tag{10.1.21}$$

This proves the theorem. □

Remarks

1. The quadratic residual tends to zero exponentially, hence the time dependent solution tends to the equilibrium solution exponentially.
2. If a *Neumann*-boundary condition is given on the *entire* boundary, a compatibility condition (which?) has to be satisfied in order that a physical equilibrium is possible. For this particular case the conditions of the theorem have to be adapted. If the compatibility condition condition is not satisfied, the solution of the time dependent problem is unbounded. Depending on the sign of the net heat production temperature goes to $\pm\infty$.
3. This theorem, proved for the Laplace operator, also holds for the general elliptic operator
$$L = \sum_{\alpha}^{n}\sum_{\beta}^{n} \frac{\partial}{\partial x_\alpha} K_{\alpha\beta} \frac{\partial}{\partial x_\beta},$$
with K positive definite.
4. In a similar way, it is possible to establish *analytical stability* for this problem, i.e. one can demonstrate well-posedness in relation to the initial conditions: Given two solutions u and v with initial conditions u_0 and $u_0 + \epsilon_0$ respectively, then, for $\epsilon(\mathbf{x}, t) = (v - u)(\mathbf{x}, t)$, we have
$$\left(\int_\Omega \epsilon^2 \, d\Omega \right)(t) < e^{-\gamma(t-t_0)} \int_\Omega \epsilon_0^2 \, d\Omega. \tag{10.1.22}$$
Hence, for this problem, we have absolute (asymptotic) stability, because the error tends to zero as $t \to \infty$.

□

Exercise 10.1.1 *Prove Theorem (10.1.1) for the general elliptic operator*
$$L = \sum_{\alpha}^{n}\sum_{\beta}^{n} \frac{\partial}{\partial x_\alpha} K_{\alpha\beta} \frac{\partial}{\partial x_\beta}, \quad K \text{ positive definite.}$$
(Hint: For any symmetric matrix K, $(\mathbf{x}, K\mathbf{x}) \geq \lambda_0(\mathbf{x}, \mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n \backslash \{\mathbf{0}\}$, where λ_0 represents the smallest eigenvalue of K.)

□

Exercise 10.1.2 *Demonstrate the analytic absolute stability of (10.1.6).* □

10.2 Method of lines

A very general method to solve time dependent problems is the *method of lines*. In this method we start with the *spatial* discretization of the problem
$$\frac{\partial u}{\partial t} = \Delta u + f. \tag{10.2.1}$$
This spatial discretization can be based on Finite Differences, Finite Volumes or on Finite Elements. The spatial discretization results in a system of ordinary differential equations the size of which is determined by the number of parameters used to approximate u. Formally, this system can be written as
$$M \frac{d\mathbf{u}_h}{dt} = S\mathbf{u}_h + \mathbf{f}_h. \tag{10.2.2}$$

The quantities with index $_h$ represent the discrete approximations of the continuous quantities. Note the matrix M, the *mass matrix*, in the left-hand side. It is the identity matrix in Finite Differences, but has different structure in Finite Volumes or Finite Elements. M represents the scaling of the equations in the discretization. The matrix S is a (possibly scaled) discrete representation of the elliptic operator L and for the FEM it is the same as the stiffness matrix of the corresponding elliptic problem. We illustrate the method with a few examples.

10.2.1 One dimensional examples

In this section we consider the following equation with one space coordinate:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(x,t), \quad x \in [0,1], \tag{10.2.3}$$

with initial condition $u(x,t_0) = u_0(x)$. We look at two different discretization methods.

Example 10.2.1 FDM, Dirichlet
We use as boundary conditions: $u(0) = u(1) = 0$. Similarly as in Chapter 3, the interval $(0,1)$ is divided into sub-intervals of size h, such that $Nh = 1$. The second order derivative is discretized using the second divided difference in each gridnode. In each gridnode x_j, $j = 0 \ldots N$, there is a u_j, which, of course, also depends on time. From the boundary conditions, it follows that $u_0 = 0 = u_N$, hence the remaining unknowns are $u_1, \ldots, u_{N-1}$. After elimination of u_0 and u_N we obtain the following system of ordinary differential equations:

$$\frac{d\mathbf{u}_h}{dt} = S\mathbf{u}_h + \mathbf{f}_h, \tag{10.2.4}$$

with

$$S = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & 0 & \ldots & \ldots & 0 \\ 1 & -2 & 1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 & -2 & 1 \\ 0 & \ldots & \ldots & 0 & 1 & -2 \end{pmatrix}, \tag{10.2.5}$$

$$\mathbf{u}_h = \begin{pmatrix} u_1 \\ \vdots \\ u_{N-1} \end{pmatrix} \quad \text{and} \quad \mathbf{f}_h = \begin{pmatrix} f_1 \\ \vdots \\ f_{N-1} \end{pmatrix}. \tag{10.2.6}$$

in which $\mathbf{u}_h$ and $\mathbf{f}_h$ both depend on t. □

Example 10.2.2 FVM, right-hand boundary point Neumann
We take as boundary conditions $u(0) = 0$, $u'(1) = 0$. Further, a non-equidistant grid is used with N grid nodes, and $h_i = x_{i+1} - x_i$. As a control volume around the node x_i, the interval $V_i = (x_i - 1/2h_{i-1}, x_i + 1/2h_i)$ is used. Subsequently, we integrate the differential equation over the control volume. This gives:

$$\int_{x_i - 1/2h_{i-1}}^{x_i + 1/2h_i} \frac{\partial u}{\partial t}\, dx = \int_{x_i - 1/2h_{i-1}}^{x_i + 1/2h_i} \frac{\partial^2 u}{\partial x^2} + f\, dx, \tag{10.2.7}$$

hence

$$\frac{\partial}{\partial t}\int_{x_i-1/2h_{i-1}}^{x_i+1/2h_i} u\,dx = \frac{\partial u}{\partial x}\Big|_{x_{i+1/2h}} - \frac{\partial u}{\partial x}\Big|_{x_{i-1/2h}} + \int_{x_i-1/2h_{i-1}}^{x_i+1/2h_i} f\,dx. \tag{10.2.8}$$

For the integrals

$$\int_{x_i-1/2h_{i-1}}^{x_i+1/2h_i} u\,dx \quad \text{and} \quad \int_{x_i-1/2h_{i-1}}^{x_i+1/2h_i} f\,dx,$$

the mid-point rule will be used. □

Exercise 10.2.1 *Give the mass matrix and stiffness matrix for this problem, so that the discretization can be written as*

$$M\frac{d\mathbf{u}_h}{dt} = S\mathbf{u}_h + \mathbf{f}_h. \tag{10.2.9}$$

□

10.2.2 Two-dimensional example

In this section we consider the following equation in two spatial coordinates:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, \tag{10.2.10}$$

with initial condition $u(\mathbf{x}, t_0) = u_0(\mathbf{x})$.

Example 10.2.3 FEM, Neumann, Robin
Take Ω bounded, $\partial u/\partial n = 0$ on Γ_1, $\partial u/\partial n + \sigma u = 0$ on Γ_2, with $\Gamma_1 \cup \Gamma_2 = \Gamma$. We distribute Ω into triangles, multiply (10.2.10) by ϕ_k, integrate by parts and obtain:

$$\begin{aligned}\frac{d}{dt}\sum_{i=1}^{N} u_i \int_\Omega \phi_i\phi_k\,d\Omega = &-\sum_{i=1}^{N} u_i \int_\Omega (\operatorname{grad}\phi_i, \operatorname{grad}\phi_k)\,d\Omega \\ &+ \int_\Gamma \phi_k \frac{\partial u}{\partial n}\,d\Gamma + \int_\Omega f\phi_k\,d\Omega.\end{aligned} \tag{10.2.11}$$

After taking the boundary conditions into account, one obtains:

$$\begin{aligned}\frac{d}{dt}\sum_{i=1}^{N} u_i \int_\Omega \phi_i\phi_k\,d\Omega = &-\sum_{i=1}^{N} u_i \int_\Omega (\operatorname{grad}\phi_i, \operatorname{grad}\phi_k)\,d\Omega \\ &- \sum_{i=1}^{N} u_i \int_{\Gamma_2} \sigma\phi_k\phi_i\,d\Gamma + \int_\Omega f\phi_k\,d\Omega.\end{aligned} \tag{10.2.12}$$

This gives a system of ordinary differential equations of the form

$$M\frac{d\mathbf{u}}{dt} = S\mathbf{u} + \mathbf{f}. \tag{10.2.13}$$

with

$$m_{ki} = \int_{\Omega} \phi_k \phi_i \, d\Omega, \tag{10.2.14}$$

$$s_{ki} = -\int_{\Omega} (\operatorname{grad} \phi_k, \operatorname{grad} \phi_i) \, d\Omega - \int_{\Gamma_2} \sigma \phi_k \phi_i \, d\Gamma, \tag{10.2.15}$$

$$f_k = \int_{\Omega} f \phi_k \, d\Omega. \tag{10.2.16}$$

□

We note that if Newton-Cotes integration is applied to the coefficients of the mass matrix, the mass matrix becomes diagonal. This process is called *lumping*.

10.3 Consistency of the spatial discretization

In Chapter 3 consistency of a discretization of a differential operator was treated. For the FVM and FEM discretization of the diffusion equation, it is necessary to include the scaling of the mass matrix M. This means that consistency of the discretization implies that $M^{-1}Sy$ tends to Ly as h tends to zero. In practical situations this can be hard to verify. In order to determine the order of consistency, it suffices to multiply each equation from a FVM discretization by the area of the control volume. For a FEM discretization it is cumbersome to determine *the order* of the consistency of the approximation of the differential operator. However, a conforming FEM approach is always consistent. Each classical definition is pessimistic about the order of the accuracy (if one uses the rule of thumb: order of consistency = accuracy of the numerical solution). Roughly speaking, the accuracy of the numerical solution is $O(h^{p+1})$ for interpolation polynomials of the order p. For convenience this order of the accuracy of the solution is used as the 'definition' of the consistency of the solution.

We will demonstrate that the truncation error of the spatial discretization, of the system ordinary differential equations, causes an error of the same order for the time dependent PDE. We suppose that the *exact* solution of the heat equation, can be substituted into the *discrete* approximation, to obtain:

$$M\frac{d\mathbf{y}}{dt} = S\mathbf{y} + \mathbf{f} + M\mathbf{E}(t), \tag{10.3.1}$$

where $E_k(t) = O(h^p)$ is the error of the kth equation, which, of course, depends on t. The generic discretization parameter (for instance the diameter of the largest element) is denoted by h and p represents the order of the consistency. In the remaining part of this section, the following properties of S and M will be used:

- M and S are symmetric.
- M is positive definite, S is negative definite (i.e. $(\mathbf{x}, S\mathbf{x}) < 0$, for $\mathbf{x} \neq 0$).
- There is a $\gamma_0 > 0$ such that

$$\frac{(\mathbf{x}, S\mathbf{x})}{(\mathbf{x}, M\mathbf{x})} < -\gamma_0. \tag{10.3.2}$$

Now we will show that the difference between the exact solution of the heat equation and the solution of the system of ordinary differential equations is bounded by the error $\mathbf{E}(t)$. Since M is a positive matrix the expression $\|\mathbf{x}\|_M$ defined by $\|\mathbf{x}\|_M = (\mathbf{x}, M\mathbf{x})^{\frac{1}{2}}$ is a proper vector norm. We formulate our result in this norm.

Theorem 10.3.1 *The difference $\epsilon = \mathbf{y} - \mathbf{u}$ between the exact solution of the heat equation and the solution of the system of ordinary differential equations (10.3.1), satisfies the following estimate:*

$$\|\epsilon\|_M < \frac{1}{\gamma_0} \sup_{t>t_0} \|\mathbf{E}(t)\|_M. \tag{10.3.3}$$

Proof
The proof is similar to the proof of the fundamental inequality of Theorem 10.1.1. We subtract the solution of

$$M\frac{d\mathbf{u}}{dt} = S\mathbf{u} + \mathbf{f}, \tag{10.3.4}$$

from (10.3.1), to obtain:

$$M\frac{d\epsilon}{dt} = S\epsilon + M\mathbf{E}. \tag{10.3.5}$$

Since $\mathbf{y}$ and $\mathbf{u}$ have the same initial condition, we have $\epsilon(t_0) = 0$. Taking the inner product of the above equation with ϵ we get:

$$\frac{1}{2}\frac{d(\epsilon, M\epsilon)}{dt} = (\epsilon, S\mathbf{ffl}) + (\epsilon, M\mathbf{E}), \text{ or} \tag{10.3.6}$$

$$\|\varepsilon\|_M \frac{d\|\varepsilon\|_M}{dt} = (\epsilon, S\epsilon) + (\epsilon, M\mathbf{E}). \tag{10.3.7}$$

With $(\varepsilon, S\varepsilon) < -\gamma_0(\varepsilon, M\varepsilon)$ and Schwartz's inequality $(\varepsilon, M\mathbf{E}) \leq \|\varepsilon\|_M \|\mathbf{E}\|_M$ this transforms into

$$\frac{d\|\varepsilon\|_M}{dt} < -\gamma_0\|\varepsilon\|_M + \|\mathbf{E}\|_M, \tag{10.3.8}$$

and hence

$$\frac{d}{dt}\left(e^{\gamma_0 t}\|\varepsilon\|_M\right) < e^{\gamma_0 t}\|\mathbf{E}\|_M. \tag{10.3.9}$$

We integrate this expression and use $\epsilon_0 = 0$ to obtain

$$e^{\gamma_0 t}\|\varepsilon\|_M < \int_{t_0}^{t} e^{\gamma_0 \tau}\|\mathbf{E}\|_M \, d\tau. \tag{10.3.10}$$

Hence

$$\|\varepsilon\|_M < \frac{1}{\gamma_0}(1 - e^{-\gamma_0(t-t_0)}) \sup_{t>t_0} \|\mathbf{E}\|_M, \tag{10.3.11}$$

and the theorem follows. □

Remark
If $\tilde{y} = \sum_{i=1}^{N} y_i \phi_i$, (i.e. the interpolated value using the exact solution), and if $\tilde{u}$ represents the FEM approximation, then

$$\int_{\Omega} (\tilde{y} - \tilde{u})^2 \, d\Omega = (\epsilon, M\epsilon), \tag{10.3.12}$$

which is straightforward to show. Something similar holds for the FVM approach.

Exercise 10.3.1 *Prove inequality (10.3.2).*
Hint: Consider

$$\sup_{\mathbf{x}} \frac{(\mathbf{x}, S\mathbf{x})}{(\mathbf{x}, \mathbf{x})} \frac{(\mathbf{x}, \mathbf{x})}{(\mathbf{x}, M\mathbf{x})} < \sup_{\mathbf{x}} \frac{(\mathbf{x}, S\mathbf{x})}{(\mathbf{x}, \mathbf{x})} \sup_{\mathbf{y}} \frac{(\mathbf{y}, \mathbf{y})}{(\mathbf{y}, M\mathbf{y})}$$

□

Exercise 10.3.2 *Positive definiteness of M implies that for all α, β and vectors $\mathbf{x}$ and $\mathbf{y}$*

$$(\alpha\mathbf{x} + \beta\mathbf{y}, M(\alpha\mathbf{x} + \beta\mathbf{y})) = \alpha^2 \|\mathbf{x}\|_M{}^2 + 2\alpha\beta(\mathbf{x}, M\mathbf{y}) + \beta^2 \|\mathbf{y}\|_M{}^2 > 0 \qquad (10.3.13)$$

Use this to prove Schwartz's inequality. □

Exercise 10.3.3 *Prove the fundamental inequality of Theorem 10.1.1 for the solution of*

$$M\frac{d\mathbf{u}}{dt} = S\mathbf{u} + \mathbf{f}. \qquad (10.3.14)$$

□

10.4 Time integration

The next step we have to take is to integrate in time our system of ordinary differential equations, that we obtained by the method of lines. To this end we use well known methods for numerical integration of initial value problems, like Euler, improved Euler, Runge-Kutta or the trapezoidal rule.

Example 10.4.1 *Application of Euler's method gives:*

$$M\frac{\mathbf{u}^{n+1}}{\Delta t} = M\frac{\mathbf{u}^n}{\Delta t} + S\mathbf{u}^n + \mathbf{f}^n, \qquad (10.4.1)$$

in which $\mathbf{u}^{n+1}$ and $\mathbf{u}^n$ represent the solutions on t_{n+1} and t_n respectively, with $t_n = t_0 + n\Delta t$. □

So unless M is diagonal we have to solve a system of equations *in each time step* even when we use an explicit integration scheme. In FDM or FVM M is diagonal, but in FEM M has the complexity of the Laplacian operator. If you nevertheless really want to use an explicit integration method (there are several reasons why you would not) you can diagonalize M by a technique known as *lumping*. See Exercise 10.4.1. This technique can be used only for linear basis functions. If you do not lump the mass matrix you have to solve a system with the complexity of the Laplacian in each time step.

Exercise 10.4.1 *Calculate the element mass matrix for linear basis functions*

$$m^e_{ij} = \int_e \lambda_i \lambda_j \, de \qquad (10.4.2)$$

using Newton Cotes' integration rule. Show that the element mass matrix is diagonal and explain that the large mass matrix has to be diagonal too. □

Exercise 10.4.2 *Formulate the implicit method of Euler (backward) for the system of ordinary differential equations as obtained from the method of lines.* □

Exercise 10.4.3 *Formulate the improved Euler method for this system.* □

Example 10.4.2 *The method of Crank-Nicholson or* the trapezoid rule *for our system of ordinary differential equations is given by:*

$$(\frac{M}{\Delta t} - \frac{1}{2}S)\mathbf{u}^{n+1} = (\frac{M}{\Delta t} + \frac{1}{2}S)\mathbf{u}^n + \frac{1}{2}\left(\mathbf{f}^n + \mathbf{f}^{n+1}\right). \qquad (10.4.3)$$

□

Example 10.4.3 *The* θ-method *for the system of ordinary differential equations is given by:*

$$(\frac{M}{\Delta t} - \theta S)\mathbf{u}^{n+1} = (\frac{M}{\Delta t} + (1-\theta)S)\mathbf{u}^n + (1-\theta)\mathbf{f}^n + \theta\mathbf{f}^{n+1}, \tag{10.4.4}$$

where θ *is a real number in the closed interval between zero and one. Note that* $\theta = 0$, $\theta = 1$ *and* $\theta = \frac{1}{2}$ *correspond to the Forward, Backward Euler and the Crank-Nicholson method respectively.* □

For the θ-method it can be shown that the global error in the time integration is of second order if $\theta = \frac{1}{2}$ and else the order of the error is of first order.

10.5 Stability of the numerical integration

In section 10.1 we demonstrated that the heat equation is *absolutely* stable with respect to the initial conditions. This means that if two solutions have different initial conditions, the difference between these two solutions vanishes as $t \to \infty$. This property also holds for the system of ordinary differential equations obtained by the method of lines (see Exercise 10.5.1). We want to make sure that the numerical time integration inherits this property, so that the numerical time integration is absolutely stable as well. Stability of numerical integration methods in time is treated more extensively in [7]. We state the most important results. The stability of the system of ordinary differential equations:

$$\frac{d\mathbf{u}}{dt} = A\mathbf{u} + \mathbf{f}, \tag{10.5.1}$$

is determined by the 'error-equation'

$$\boldsymbol{\frac{d\epsilon}{dt} = A\epsilon.} \tag{10.5.2}$$

1. The system is absolutely stable if and only if the real part of the eigenvalues λ_k of the matrix A is negative, i.e. $\mathrm{Re}(\lambda_k) < 0$.

2. Each numerical solution procedure has an *amplification matrix* $G(\Delta tA)$, given by the numerical solution of (10.5.2):

   $$\boldsymbol{\epsilon^{n+1} = G(\Delta tA)\epsilon^n.} \tag{10.5.3}$$

   If the error equation is *scalar* (i.e. the system reduces to of one equation only: $\epsilon' = \lambda\epsilon$), the matrix reduces to an *amplification factor* , which is denoted by $C(\Delta t\lambda)$.

3. A numerical solution method is absolutely stable if all eigenvalues μ_k of $G(\Delta tA)$ have the property $|\mu_k| < 1$.

4. The eigenvalues μ_k of $G(\Delta tA)$ can be obtained by substitution of the eigenvalues λ_k of the matrix A into the amplification factor:

   $$\mu_k = C(\Delta t\lambda_k). \tag{10.5.4}$$

   Hence, for stability we need $|C(\Delta t\lambda_k)| < 1$.

Exercise 10.5.1 *The amplification matrices for forward Euler, improved Euler, backward Euler, Crank-Nicholson and the θ-method are given by*

$$\begin{aligned} I + \Delta t A, \\ I + \Delta t A + \frac{1}{2}(\Delta t A)^2, \\ (I - \Delta t A)^{-1}, \\ (I - \frac{1}{2}\Delta t A)^{-1}(I + \frac{1}{2}\Delta t A), \\ (I - \theta \Delta t A)^{-1}(I + (1-\theta)\Delta t A). \end{aligned}$$

Show this. What are the corresponding amplification factors? □

If the mass matrix is not I we have $A = M^{-1}S$, hence, in order to investigate the stability of the numerical time integration, the eigenvalues of $M^{-1}S$ have to be estimated. We note that the eigenvalues of $M^{-1}S$ are the same as the eigenvalues of the generalized eigenvalue problem:
Determine λ and $\mathbf{x} \neq \mathbf{0}$, such that

$$S\mathbf{x} = \lambda M\mathbf{x}. \tag{10.5.5}$$

All eigenvalues of the above generalized eigenvalue are real-valued and negative, since S is negative definite and M is positive definite. (See [36]). For real-valued eigenvalues, the following criterion for stability holds

$$\Delta t < \frac{c}{|\lambda_{max}|}, \tag{10.5.6}$$

with $c = 2$ for Euler and improved Euler and $c = 2.8$ for Runge-Kutta (see [7]). Hence we have to estimate the maximal eigenvalue of the generalized eigenvalue problem. This is treated in the next section.

10.5.1 Gershgorin's circle theorem

The most interesting case to consider is that M is diagonal. We may formulate Gershgorin's Theorem for this case to estimate the lie of the eigenvalues.

Theorem 10.5.1 (Gershgorin)
Let M be diagonal, then, for all eigenvalues λ of $M^{-1}S$ holds:

$$|m_{kk}\lambda - s_{kk}| \leq \sum_{i=1_{i\neq k}}^{N} |s_{ki}|. \tag{10.5.7}$$

Remark:
Eigenvalues may be complex valued in general and for complex eigenvalues $\lambda = \mu + i\nu$, the absolute value is the *modulus:* $|\lambda| = \sqrt{\mu^2 + \nu^2}$. So the eigenvalues are located within a circle in the complex plane and that is the reason why the theorem is also often referred to as Gershgorin's *circle* theorem. But for symmetric M and S the eigenvalues of $M^{-1}S$ are real-valued.

Proof
Let λ be an eigenvalue of the generalized eigenvalue problem with corresponding eigenvector $\mathbf{v}$, then,

$$\sum_i s_{pi}v_i = \lambda m_{pp}v_p, \quad p = 1, \ldots, N. \tag{10.5.8}$$

Let v_k be the component of $\mathbf{v}$ with the largest modulus. For this index l we have

$$\lambda m_{kk} - s_{kk} = \sum_i \sum_{i \neq k} s_{ki} \frac{v_i}{v_k}, \tag{10.5.9}$$

and because $|v_i/v_k| \leq 1$, we get

$$|\lambda m_{kk} - s_{kk}| \leq \sum_i \sum_{i \neq k} |s_{ki}|. \tag{10.5.10}$$

This proves the theorem. □

Example 10.5.1 *For the heat equation in one spatial dimension (see example 10.2.1) the Finite Difference Method gives $M = I$ and hence*

$$|\lambda_{max}| < \frac{4}{h^2}. \tag{10.5.11}$$

From this we obtain a stability criterion for the Forward Euler method:

$$\Delta t < \frac{2h^2}{4} = \frac{1}{2}h^2. \tag{10.5.12}$$

Application of a two dimensional Finite Difference Method (see example 10.4.1) with two spatial coordinates, gives in a similar way:

$$|\lambda_{max}| < \frac{4}{(\Delta x)^2} + \frac{4}{(\Delta y)^2}, \tag{10.5.13}$$

and a stability criterion of the form

$$\Delta t < \frac{\beta^2}{2(1+\beta^2)}(\Delta x)^2, \tag{10.5.14}$$

in which $\Delta y = \beta \Delta x$. □

Example 10.5.2 *Lumping the mass matrix in Example 10.2.2 gives $m_{ii} = \frac{1}{2}(h_{i-1} + h_i)$. Gershgorin's Theorem results in the following estimate:*

$$|\lambda_{max}| < \sup_i \frac{2}{h_{i-1} + h_i} \left(\frac{2}{h_{i-1}} + \frac{2}{h_i} \right) \tag{10.5.15}$$

$$= \sup_i \frac{4}{h_{i-1} h_i}, \tag{10.5.16}$$

and a stability criterion of the form

$$\Delta t < \frac{1}{2} \inf_i (h_{i-1} h_i). \tag{10.5.17}$$

□

In all the examples the time step has to be smaller than the product of a factor times the square of the grid spacing. In practical situations, this could imply that the time step has to be very small. For that reason explicit time integration methods are not popular for the heat equation. Implicit methods such as the Crank-Nicholson method or the implicit Euler (backward) method are usually preferred. *This always implies the solution of a problem with the complexity of the Laplacian in each time step*. In one space dimension, this amounts to the solution of a tridiagonal system of equations in each time step, which is no big deal. Two and more space dimensions however lead to the same type of problems as the Laplacian. For iterative methods the solution on the previous time level is of course an excellent starting value.

For regions with simple geometries some special implicit methods for the heat equation are available. This will be addressed later.

Exercise 10.5.2 *Prove that Euler backward and Crank-Nicholson are absolutely stable for each value of the step-size Δt if $Re(\lambda_k) < 0$.* □

Exercise 10.5.3 *Prove that the θ-method is absolutely stable for all time steps if $\theta \geq \frac{1}{2}$. Derive a condition for stability for the case that $\theta < \frac{1}{2}$.* □

As an illustration of the stability of the numerical solution to the heat problem we consider a Finite Element solution in the square $\Omega = [0,1] \times [0,1]$, on which

$$\frac{\partial u}{\partial t} = 0.5 \Delta u. \tag{10.5.18}$$

We take as initial condition and boundary condition at all the boundaries Γ:

$$\begin{aligned} u(x,y,0) &= \sin(x)\sin(y), (x,y) \in \Omega \\ u(x,y,t) &= \sin(x)\sin(y), (x,y) \in \Gamma. \end{aligned} \tag{10.5.19}$$

Exercise 10.5.4 *Prove that the analytical solution to the above problem is given by:*

$$u(x,y,t) = e^{-t}\sin(x)\sin(y). \tag{10.5.20}$$

□

In Figure 10.1 we show the numerical solution to the above problem as computed by the use of the Forward Euler method with $\Delta t = 0.1$. For this case the stability criterion is violated and hence the solution exhibits unphysical behavior. In Figure 10.2 we show the solution that has been obtained for the same data by the backward Euler method. Now the solution exhibits the expected physical behavior. The contourlines are nice and smooth and are similar to the ones of the analytical solution.

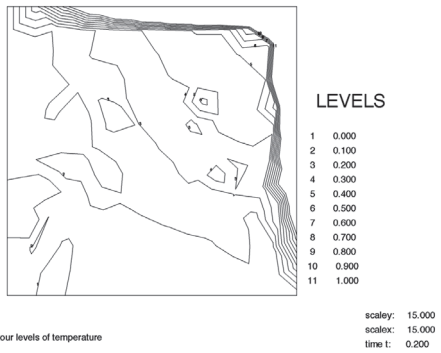

Figure 10.1: Contourlines of the numerical solution to the heat equation with $\Delta t = 0.1$ as obtained by the use of the *Forward* (explicit) Euler method (unstable solution).

10.5.2 Stability analysis of Von Neumann

As an alternative method to estimate the eigenvalues of the matrix $M^{-1}S$ we present a method due to the American mathematician John Von Neumann. This method

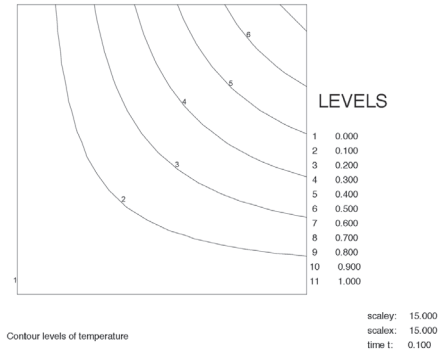

Figure 10.2: Contourlines of the numerical solution to the heat equation with $\Delta t = 0.1$ as obtained by the use of the *Backward* (implicit) Euler method.

has gained much popularity. For equations with *constant coefficients* and *equidistant grids* it can be shown that eigenvectors of $M^{-1}S$ can be written as

$$v_k = e^{i\rho kh} \tag{10.5.21}$$

in one and

$$v_{kl} = e^{i(\rho k\Delta x + \sigma l\Delta y)} \tag{10.5.22}$$

in two space dimensions. The region must be rectangular in 2 D. The numbers ρ and σ depend on the type of boundary conditions. In order to estimate an upper bound for the eigenvalues it is sufficient to substitute these expressions in one single equation of the generalized eigenvalue problem.

Example 10.5.3 *As an example we consider the heat equation with an equidistant grid in one space dimension*

$$\lambda e^{i\rho kh} = \frac{1}{h^2}(e^{i\rho(k-1)h} - 2e^{i\rho kh} + e^{i\rho(k+1)h}). \tag{10.5.23}$$

We divide the left and right-hand sides of this equation by $e^{i\rho kh}$ and obtain using the relation $1/2(e^{i\phi} + e^{-i\phi}) = \cos\phi$:

$$\lambda = \frac{2(\cos(\rho h) - 1)}{h^2} = -4\frac{\sin^2 \rho h/2}{h^2}. \tag{10.5.24}$$

From this we find for the eigenvalues the estimate:

$$|\lambda| \leq \frac{4}{h^2} \tag{10.5.25}$$

and the stability criterion

$$\Delta t < \frac{1}{2}h^2, \tag{10.5.26}$$

for the forward Euler time-integration. □

Remark

1. Von Neumann's original method also uses time dependence and calculates *amplification factors* directly. Our presentation is more in line with the method of lines.

2. The domain of computation, in which the Von Neumann analysis is applied, does not necessarily have to be rectangular. In that case the analysis gives a rough upper bound for the eigenvalues, which in fact holds for the smallest rectangle that encloses the domain of computation. The coefficients in the PDE have to be constant. Furthermore the discretization has to be equidistant, otherwise the analysis is not valid. If both Gershgorin's Theorem and the Von Neumann analysis can be applied, these methods give the same stability criterion. Gershgorin's Theorem can also be applied for non-constant coefficients and non-equidistant grids. But the mass matrix has to be diagonal in that case.

10.6 The accuracy of the time integration

When we use a numerical method for time integration we make an error at each time step. These errors accumulate in general, and you might ask if this accumulation could be disastrous. From [7] we know that in a bounded time interval $(t_0, T]$ a local truncation error of the order $O(h^m)$ gives a global error of the same order. The forward and backward methods of Euler have $m = 1$, whereas the improved Euler method and the method of Crank-Nicholson have $m = 2$. Absolutely stable systems like the heat equation have even better properties. If the numerical integration is stable, the global error is uniformly bounded on the interval (t_0, ∞).

Theorem 10.6.1 *Let $\mathbf{y}(t)$ be the solution of the absolutely stable system*

$$\frac{d\mathbf{y}}{dt} = A\mathbf{y} + \mathbf{f}, \quad \mathbf{y}(t_0) = y_0. \tag{10.6.1}$$

Further, let $\mathbf{u}^n$ be the solution of the numerical method

$$\mathbf{u}^{n+1} = G(\Delta t A)\mathbf{u}^n + I_n(\mathbf{f}), \quad \mathbf{u}^0 = \mathbf{y}_0, \tag{10.6.2}$$

where $I_n(f)$ represents an approximation of

$$\int_{t_n}^{t_{n+1}} \mathbf{f} dt, \tag{10.6.3}$$

so that

1. $$\mathbf{y}(t_{n+1}) = G(\Delta t A)\mathbf{y}(t_n) + I_n(\mathbf{f}) + (\Delta t)^{m+1}\mathbf{p}^n. \tag{10.6.4}$$
   Here $\|p^n\|$ is uniformly bounded for all n and Δt.
2. $\lim_{n\to\infty} G(\Delta t A)^n \to 0, \quad \forall \Delta t < \tau,$

then, the following holds

$$\|\mathbf{y}(t_n) - \mathbf{u}^n\| = O((\Delta t)^m). \tag{10.6.5}$$

In other words: if the local truncation error in time is of order m (after division of equation (10.6.4) by Δt), the global error is also of order m provided the integration is stable.

Proof
We define $\epsilon^n = y(t_n) - u^n$and subtract Equation (10.6.2) from Equation (10.6.4) to get:

$$\boldsymbol{\epsilon^{n+1} = G(\Delta t A)\epsilon^n + (\Delta t)^{m+1}\mathbf{p}^n.} \tag{10.6.6}$$

Now $\epsilon_0 = 0$ and we shall show by induction that

$$\boldsymbol{\epsilon^n = (\Delta t)^{m+1} \sum_{k=0}^{n-1} G(\Delta t A)^{n-k-1} \mathbf{p}^k.} \tag{10.6.7}$$

Equation (10.6.7) holds for $n = 0$. Assume Equation (10.6.7) holds until n. We obtain

$$\epsilon^{n+1} = G(\Delta t A)\epsilon^n + (\Delta t)^{m+1}\mathbf{p}^n \tag{10.6.8}$$

$$= (\Delta t)^{m+1} G(\Delta t A) \sum_{k=0}^{n-1} G(\Delta t A)^{n-k-1}\mathbf{p}^k + (\Delta t)^{m+1}\mathbf{p}^n \tag{10.6.9}$$

$$= (\Delta t)^{m+1} \sum_{k=0}^{n} G(\Delta t A)^{n-k}\mathbf{p}^k. \tag{10.6.10}$$

From this we conclude that (10.6.7) holds for all n. $\|p^n\|$ is uniformly bounded, so there exists a vector $\mathbf{p}_{max}$ with $\|\mathbf{p}^n\| < \|\mathbf{p}_{max}\|$ for all n. Putting this into (10.6.10) we obtain

$$\boldsymbol{\|\epsilon^n\| \leq (\Delta t)^{m+1} \sum_{k=0}^{n-1} \|G(\Delta t A)^{n-k-1}\| \|\mathbf{p}_{max}\|.} \tag{10.6.11}$$

We use the diagonalization of $G(\Delta t A)$:

$$G(\Delta t A) = Q^{-1} M Q, \tag{10.6.12}$$

where Q is a matrix with the eigenvectors of $G(\Delta t A)$ as columns and M is a diagonal matrix with the eigenvalues of $G(\Delta t A)$. For $\|G(\Delta t A)^k\|$ we have

$$G(\Delta t A)^k = Q^{-1} M^k Q \tag{10.6.13}$$

$$\|G(\Delta t A)^k\| \leq |\mu_1^k| \, \|Q^{-1}\| \, \|Q\|. \tag{10.6.14}$$

μ_1 is the eigenvalue of $G(\Delta t A)$ with the largest modulus. This gives

$$\boldsymbol{\|\epsilon^n\| \leq (\Delta t)^{m+1} \frac{1 + |\mu_1^n|}{1 - |\mu_1|} \|Q^{-1}\| \, \|Q\| \, \|\mathbf{p}_{max}\|.} \tag{10.6.15}$$

Since $\mu_1 = C(\lambda_1 \Delta t) = 1 + \lambda_1 \Delta t + O(\Delta t^2)$, we have $1 - \mu_1 = \lambda_1 \Delta t + O(\Delta t^2)$ and we finally obtain

$$\boldsymbol{\|\epsilon^n\| \leq K(\Delta t)^m,} \tag{10.6.16}$$

which proves the theorem. □

10.7 Conclusions for the method of lines

We summarize the results of the methods of lines for the heat/diffusion equation.

- Using the method of lines, the PDE is written as a system of ordinary differential equations by the spatial discretization of the elliptic operator.
- The global error of the *analytic* solution of this system of ordinary differential equations (compared to the solution of the solution of the PDE) is of the same order as the consistency of the FDM and FVM and an order higher than the degree of the interpolation polynomials of the FEM.

- The *numerical* solution of this system has an additional error due to the numerical time integration. This global error is of the order of $K\Delta t^m$, if the local truncation error is of the order $O(\Delta t^m)$ This constant does not depend on time t and this estimate holds at the *entire* time interval (t_0, ∞).
- Explicit (and some implicit) methods have a stability criterion of the form

$$\Delta t < c\Delta x^2 \tag{10.7.1}$$

  and hence these methods are less suitable for the heat equation.

10.8 Special difference methods for the heat equation

The method of lines is a general method, which is applicable to one, two or three spatial dimensions. At each time step, the implicit methods give a problem to be solved with the same complexity as the Poisson problem. Therefore, one has searched for methods that are *stable* but have a simpler complexity than the Poisson problem. We present one example of such a method: The ADI method. This method can only be used with regular grids with a five-point molecule for the elliptic operator. Unfortunately, the ADI method cannot be used if the elliptic operator is discretized using a general Finite Element Method. First we sketch the principle of the ADI method and subsequently a formal description of the ADI method is given.

10.8.1 The principle of the ADI method

The abbreviation ADI means *Alternating Direction Implicit*. This is a fairly accurate description of the working of the method. Suppose that we have to solve the heat equation on a rectangle with length l_x and width l_y and we use a discretization with stepsize Δx and Δy respectively, such that $N_x\Delta x = l_x$ and $N_y\Delta y = l_y$. For convenience we apply Dirichlet boundary conditions at all the boundaries of the domain of computation, where we set $u = 0$. For the time integration of t_n up to t_{n+1} the ADI method uses two steps. The idea is as follows: first we use a half time step with an intermediate auxiliary quantity u^*. To compute u^* we use the implicit Euler time integration method for the derivative with respect to x and the explicit Euler time integration for the derivative with respect to y. In the next half time step, we reverse this process. Hence: The first step, a so-called half time step, computes an auxiliary-quantity u^*_{ij} according to:

$$\begin{aligned} u^*_{ij} = u^n_{ij} + \frac{\Delta t}{2\Delta x^2}(u^*_{i+1,j} - 2u^*_{i,j} + u^*_{i-1,j}) + \\ \frac{\Delta t}{2\Delta y^2}(u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}) + \Delta t f^*_{ij}, \\ i = 1, \ldots N_x - 1, j = 1 \ldots N_y - 1, \end{aligned} \tag{10.8.1}$$

where f^*_{ij} denotes $f(i\Delta x, j\Delta y, t_0 + (n + \frac{1}{2})\Delta t)$. Subsequently u^{n+1} is calculated according to:

$$\begin{aligned} u^{n+1}_{ij} = u^*_{ij} + \frac{\Delta t}{2\Delta x^2}(u^*_{i+1,j} - 2u^*_{i,j} + u^*_{i-1,j}) + \\ \frac{\Delta t}{2\Delta y^2}(u^{n+1}_{i,j+1} - 2u^{n+1}_{i,j} + u^{n+1}_{i,j-1}) + \Delta t f^*_{ij}, \\ i = 1, \ldots N_x - 1, j = 1 \ldots N_y - 1. \end{aligned} \tag{10.8.2}$$

In Equation (10.8.1) for a fixed index j a tridiagonal system of equations has to be solved in $\mathbf{u}_j^*$, with

$$\mathbf{u}_j^* = \begin{pmatrix} u_{1j}^* \\ u_{2j}^* \\ \vdots \\ u_{N_x-1,j}^* \end{pmatrix}. \tag{10.8.3}$$

In total there are $N_y - 1$ systems like this one to be solved in order to determine all the values of $\mathbf{u}_j^*$. Similarly, one has to solve in Equation (10.8.2) for a fixed index i a tridiagonal system of equations in $\mathbf{u}_i^{n+1}$, with

$$\mathbf{u}_i^{n+1} = \begin{pmatrix} u_{i1}^{n+1} \\ u_{i2}^{n+1} \\ \vdots \\ u_{i,N_y-1}^{n+1} \end{pmatrix}. \tag{10.8.4}$$

This is exactly in the other direction, which explains the name of the method. In total we are faced with $N_x - 1$ of such systems. Hence to integrate the heat equation from t_n up to t_{n+1} one has to

- solve $N_y - 1$ tridiagonal systems of size $N_x - 1$
- solve $N_x - 1$ tridiagonal systems of size $N_y - 1$

Exercise 10.8.1 *Verify that the amount of computational effort per time step for the ADI method is proportional to the total number of gridpoints. (Hint: How many operations does it take to solve a $N \times N$ tridiagonal system of equations?) Further, verify that the direct solution of the problem with the method of lines using for instance the method of Crank-Nicholson with a profile-method takes a computational effort which is proportional to $(N_x - 1)^2(N_y - 1)$ of $(N_x - 1)(N_y - 1)^2$, depending on the used numbering of the unknowns.* □

Indeed the computational complexity of the ADI method is better than that of the method of lines. However, the question remains whether this benefit is not at the expense of the accuracy or the stability of the method. To scrutinize this, a formal description of the ADI method is presented in the next section.

10.8.2 Formal description of the ADI method

The ADI method can be seen as a special way to integrate the system of ordinary differential equations

$$\frac{d\mathbf{u}}{dt} = (A_x + A_y)\mathbf{u} + \mathbf{f}, \tag{10.8.5}$$

which arises from a PDE using the method of lines. The ADI method of this system is given by:

$$\mathbf{u}^* = \mathbf{u}^n + \frac{1}{2}\Delta t(A_x\mathbf{u}^* + A_y\mathbf{u}^n + \mathbf{f}^*) \tag{10.8.6}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* + \frac{1}{2}\Delta t(A_x\mathbf{u}^* + A_y\mathbf{u}^{n+1} + \mathbf{f}^*). \tag{10.8.7}$$

From this the intermediate step $\mathbf{u}^*$ can be eliminated:

$$(I - \frac{1}{2}\Delta t A_x)\mathbf{u}^* = (I + \frac{1}{2}\Delta t A_y)\mathbf{u}^n + \frac{1}{2}\Delta t \mathbf{f}^* \tag{10.8.8}$$

$$(I - \frac{1}{2}\Delta t A_y)\mathbf{u}^{n+1} = (I + \frac{1}{2}\Delta t A_x)\mathbf{u}^* + \frac{1}{2}\Delta t \mathbf{f}^* \tag{10.8.9}$$

and from multiplication of the top part of this expression by $I + \frac{1}{2}\Delta t A_x$ and the bottom part with $I - \frac{1}{2}\Delta t A_x$ and from the fact that these matrices commute, one obtains:

$$(I - \frac{1}{2}\Delta t A_x)(I - \frac{1}{2}\Delta t A_y)\mathbf{u}^{n+1} = (I + \frac{1}{2}\Delta t A_x)(I + \frac{1}{2}\Delta t A_y)\mathbf{u}^n + \Delta t \mathbf{f}^*. \tag{10.8.10}$$

Equation (10.8.10) is the basis of our investigations. First, we make a statement about the accuracy.

Theorem 10.8.1 *Equation (10.8.10) differs from Crank-Nicholson's method applied in (10.8.5) by a term of the order of* $O(\Delta t^3)$.

Proof
Crank-Nicholson applied on 10.8.5 gives

$$(I - \frac{1}{2}\Delta t A_x - \frac{1}{2}\Delta t A_y)\mathbf{u}^{n+1} = (I + \frac{1}{2}\Delta t A_x + \frac{1}{2}\Delta t A_y)\mathbf{u}^n + \frac{1}{2}\Delta t(\mathbf{f}^n + \mathbf{f}^{n+1}). \tag{10.8.11}$$

Elaboration of 10.8.10 gives:

$$(I - \frac{1}{2}\Delta t A_x - \frac{1}{2}\Delta t A_y)\mathbf{u}^{n+1} = (I + \frac{1}{2}\Delta t A_x + \frac{1}{2}\Delta t A_y)\mathbf{u}^n + \frac{1}{4}\Delta t^2 A_x A_y(\mathbf{u}^n - \mathbf{u}^{n+1}) + \Delta t \mathbf{f}^*. \tag{10.8.12}$$

Now the theorem immediately follows by noting that $\mathbf{u}^n - \mathbf{u}^{n+1}$ is of order $O(\Delta t)$ and that $\mathbf{f}^* = \frac{1}{2}(\mathbf{f}^n + \mathbf{f}^{n+1}) + O(\Delta t^2)$. Hence the ADI method has the same accuracy as the method of Crank Nicholson, which is $O(\Delta t^2)$. □

It is hard to investigate the stability of the ADI method theoretically. In practical situations, it turns out that the ADI method does not require a stringent stability criterion. In a special case, there is a theoretical justification for the unconditional stability of the ADI method:

Theorem 10.8.2 *If A_x and A_y are commuting matrices (i.e. $A_x A_y = A_y A_x$), then, the ADI method is unconditionally stable.*

Proof
We have to calculate the eigenvalues of

$$(I - \frac{1}{2}\Delta t A_y)^{-1}(I - \frac{1}{2}\Delta t A_x)^{-1}(I + \frac{1}{2}\Delta t A_x)(I + \frac{1}{2}\Delta t A_y)$$

but under the conditions of the conditions of the theorem all these matrices commute. Then, the eigenvalues of these matrices are given by the product of the separate matrices

$$(I - \frac{1}{2}\Delta t A_x)^{-1}(I + \frac{1}{2}\Delta t A_x) \text{ and } (I - \frac{1}{2}\Delta t A_y)^{-1}(I + \frac{1}{2}\Delta t A_y).$$

These eigenvalues are

$$\frac{1 + \frac{1}{2}\Delta t \lambda_x}{1 - \frac{1}{2}\Delta t \lambda_x} \text{ and } \frac{1 + \frac{1}{2}\Delta t \lambda_y}{1 - \frac{1}{2}\Delta t \lambda_y}.$$

Since λ is real-valued is negative, the moduli of all these eigenvalues are less than one. □

Exercise 10.8.2 *Show that the operators A_x and A_y commute on the problem of the rectangle with Dirichlet conditions.* □

Extension of the ADI method to three spatial dimensions is not straightforward. The most straightforward way (three steps, subsequently for the x-,y- and z coordinate) is no longer unconditionally stable. Further, its global error is of the order $O(\Delta t)$. There exist adequate ADI methods for three spatial coordinates, see [26].

10.9 Summary of Chapter 10

In this chapter we paid attention to the numerical solution of the *heat* or *diffusion* equation. We have shown, that with one exception this equation has an equilibrium solution and that independent of the initial values the transient solution tends to this equilibrium solution exponentially fast.

We introduced the *method of lines* for the numerical solution which transforms the PDE into a set of ODE's by discretizing first the spatial differential operators. We estimated the effect of the truncation error of the spatial discretization on the solution of this system of ODE's. We proved that this effect is uniformly bounded. Finite Volume and Finite Element methods generate a *mass matrix*. The mass matrix of the FEM has the same complexity as the Laplacian operator. For that reason even for explicit time integration methods a system of equations of that complexity has to be solved in each time step. This is not necessary if the mass matrix is diagonal and therefore one often lumps the mass matrix, transforming it into a diagonal matrix. This procedure is only possible for linear approximation.

We briefly considered the stability of the explicit integration schemes for which we had to estimate the lie of the eigenvalues of the system matrix. To this end we could use *Gershgorin's circle theorem* or *Von Neumann's stability analysis.*

Finally we considered the *ADI-method*, an unconditionally stable method of considerable less complexity than Crank-Nicolson's method, but with the same accuracy.

Chapter 11

The wave equation

Objectives

In this chapter we shall look at various methods for the time integration of the wave equation. This equation is crucial in applications dealing with electromagnetic radiation, wave propagation, acoustics and seismics (used for oil finding for instance). Before we do this, a conservation principle for the solution of the wave equation is derived. The numerical solution should satisfy this principle as well. Stability in terms of decay and growth of the numerical solution as a function of time is investigated for several methods. Furthermore, the concepts *dispersion* and *dissipation* will be introduced and an illustration of these concepts will be given. Finally a procedure to derive the CFL-criterion, a criterion for the numerical solution to represent the exact solution, will be given by use of the characteristic curves in the x, t-plane.

11.1 A fundamental equality

Consider the wave equation on a domain Ω

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) =: c^2 \Delta u. \tag{11.1.1}$$

In Equation (11.1.1) no *internal* energy source term is taken into account. Further, homogeneous boundary conditions are imposed on the boundaries Γ_1, Γ_2 and Γ_3 of the domain Ω, i.e.

$$\begin{aligned} u &= 0, && (x,y) \in \Gamma_1, \\ \frac{\partial u}{\partial n} &= 0, && (x,y) \in \Gamma_2, \\ \sigma u + \frac{\partial u}{\partial n} &= 0, && (x,y) \in \Gamma_3. \end{aligned} \tag{11.1.2}$$

Hence there is no transport of energy through the boundaries. Therefore the PDE (11.1.1) with boundary conditions (11.1.2) is homogeneous. As initial conditions, we have that u and $\dfrac{\partial u}{\partial t}$ are given at $t = 0$ at all points in the domain of computation. Now we will show that the 'energy' of this equation is preserved in time.

Theorem 11.1.1 *The homogeneous wave equation* (11.1.1) *with homogeneous boundary*

conditions (11.1.2) *satisfies the following conservation principle*

$$\frac{1}{2}\int_{\Omega}\left\{\left(\frac{\partial u}{\partial t}\right)^2 + c^2||\text{grad}\, u||^2\right\}\, d\Omega + \frac{1}{2}\int_{\Gamma_3}\sigma c^2 u^2 d\Gamma = \textit{Constant}. \tag{11.1.3}$$

Proof: We multiply both sides of the equality of Equation (11.1.1) by $\frac{\partial u}{\partial t}$ and integrate the results over the domain Ω to obtain

$$\int_{\Omega}\frac{1}{2}\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial t}\right)^2 d\Omega = \int_{\Omega} c^2\frac{\partial u}{\partial t}\Delta u\, d\Omega = \int_{\Omega} c^2\frac{\partial u}{\partial t}\text{div grad}\, u\, d\Omega. \tag{11.1.4}$$

Assuming that all derivatives are continuous and using the product rule for differentiation, the integrand of the right-hand side can be written as

$$\text{div}\left(\text{grad}\,(u)\frac{\partial u}{\partial t}\right) - \text{grad}\,(u)\cdot\text{grad}\left(\frac{\partial u}{\partial t}\right). \tag{11.1.5}$$

This yields

$$\int_{\Omega}\frac{1}{2}\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial t}\right)^2 d\Omega = \int_{\Omega} c^2\text{div}\left(\text{grad}\, u\frac{\partial u}{\partial t}\right)\, d\Omega - \int_{\Omega} c^2\text{grad}\,(u)\cdot\text{grad}\left(\frac{\partial u}{\partial t}\right)\, d\Omega. \tag{11.1.6}$$

We apply the Divergence Theorem to the first term on the right-hand side and use the product rule for differentiation on the second term of the right-hand side to get

$$\int_{\Omega}\frac{1}{2}\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial t}\right)^2 d\Omega = \int_{\Gamma_1\cup\Gamma_2\cup\Gamma_3} c^2\frac{\partial u}{\partial n}\frac{\partial u}{\partial t}\, d\Gamma - \frac{1}{2}\int_{\Omega} c^2\frac{\partial}{\partial t}\left(\text{grad}\,(u)\cdot\text{grad}\,(u)\right)\, d\Omega. \tag{11.1.7}$$

The boundary integral on the right-hand side vanishes on Γ_1 and Γ_2 due to the boundary conditions. Application of the boundary condition on Γ_3 then transforms Equation (11.1.7) into

$$\int_{\Omega}\frac{1}{2}\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial t}\right)^2 d\Omega = -\int_{\Gamma_3} c^2\sigma u\frac{\partial u}{\partial t}\, d\Gamma - \frac{1}{2}\int_{\Omega} c^2\frac{\partial}{\partial t}\left(\text{grad}\,(u)\cdot\text{grad}\,(u)\right)\, d\Omega. \tag{11.1.8}$$

Finally using a standard differentiation property we get

$$\int_{\Omega}\frac{1}{2}\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial t}\right)^2 d\Omega = -\int_{\Gamma_3}\frac{1}{2}c^2\sigma\frac{\partial u^2}{\partial t}d\Gamma - \frac{1}{2}\int_{\Omega} c^2\frac{\partial}{\partial t}\left(\text{grad}\,(u)\cdot\text{grad}\,(u)\right)\, d\Omega. \tag{11.1.9}$$

Interchanging the differentiation and integration operations in the above expression and subsequent integration over time t proves the theorem. □

Remarks

1. Consider the wave equation with a source term

$$\frac{\partial^2 u}{\partial t^2} = c^2\Delta u + f(\mathbf{x}, t). \tag{11.1.10}$$

The difference between two solutions of Equation (11.1.10) with the same source term f and the same boundary conditions satisfies Equation (11.1.3) and homogeneous boundary conditions (11.1.2).

2. The first term in Equation (11.1.3) gives the kinetic energy of the vibrating medium, whereas the second and a third term involve the potential energy. Therefore, Equation (11.1.3) is commonly referred to as the energy norm.
3. The total amount of energy is entirely defined by the two initial conditions $u(x, y, t_0)$ and $\frac{\partial u}{\partial t}(x, y, t_0)$.
4. The difference in this 'energy-norm', between two solutions with the same boundary conditions and different initial conditions is constant at all stages.

Exercise 11.1.1 *Prove remarks 1 and 4.* □

Exercise 11.1.2 *The solution of the heat equation in the previous chapter tends to an equilibrium solution (i.e. a steady-state) as t tends to infinity. Does the solution of the wave equation tend to a steady state as t tends to infinity?* □

From remark 4 it follows that the solution of the wave equation is neutrally stable, that is an error made in the initial conditions will neither decrease nor increase and hence it persists. This property must also hold for our numerical methods. Otherwise the numerical solution would not exhibit the same physical characteristics as the analytical solution

11.2 The method of lines

In a similar way as we did for parabolic equations we may first discretize only the spatial part of the wave equation. The difference with the previous chapter is that we now have to deal with a second order system with respect to time. After the discretization of Equation (11.1.10), we obtain:

$$M\frac{d^2\mathbf{u}}{dt^2} = c^2 S\mathbf{u} + \mathbf{f}, \quad \mathbf{u}(t_0) = \mathbf{u}_0, \quad \frac{d\mathbf{u}}{dt}(t_0) = \mathbf{v}_0. \tag{11.2.1}$$

Here M and S are the mass and stiffness matrices respectively, just like in the previous chapter. Next, we establish that equation (11.2.1) also conserves the energy if $\mathbf{f} = \mathbf{0}$.

Theorem 11.2.1 *Let* $\mathbf{f} = \mathbf{0}$, *then*

$$\frac{1}{2}\left(\frac{d\mathbf{u}}{dt}, M\frac{d\mathbf{u}}{dt}\right) - \frac{1}{2}c^2(\mathbf{u}, S\mathbf{u}) = \textit{constant}. \tag{11.2.2}$$

Exercise 11.2.1 *Prove this theorem. Hint: Use the symmetry of M and S.* □

11.2.1 The error in the solution of the system

Application of the method of lines generates a truncation error $\mathbf{E}$ in the spatial discretization. This may be defined by

$$M\frac{d^2\mathbf{y}}{dt^2} = c^2 S\mathbf{y} + \mathbf{f} + M\mathbf{E}, \tag{11.2.3}$$

where $\mathbf{y}$ denotes the exact solution to the wave equation. This definition holds for Finite Difference and Finite Volume methods. For the Finite Element Method, the order of the truncation error depends on the approximation properties of the basis functions. Under fairly general assumptions it can be shown that this truncation error is equal to the truncation error of the polynomial interpolation of the basis

functions. This truncation error causes an error in the solution of (11.2.1) of the form Ch^p, where h denotes a generic discretization parameter (such as the diameter of the largest element used in the discretization) and p represents the order of consistency (i.e. for Finite Elements it is the order of the polynomial degree plus one). For the heat equation it was possible to find a constant C, valid for the entire interval of integration (t_0, ∞). For the wave equation this is not possible. The constant C depends linearly on the length of the integration interval (t_0, T). A complete analysis of the error is beyond the scope of the book., but qualitatively the phenomenon is explained as follows: An eigenvibration of (11.1.1) is given by a function of the form of $e^{i\lambda ct}U(x, y)$, where U satisfies the *homogeneous* boundary conditions (note that the boundary conditions can be of several types). Substitution into equation (11.1.1) yields

$$-\lambda^2 c^2 U = c^2 \Delta U. \tag{11.2.4}$$

This is just the eigenvalue problem for the Laplace operator, which has an infinite number of solutions in terms of eigenpairs λ_k and U_k. λ_k is the eigenfrequency of the vibration and U_k the eigenfunction. These quantities depend on the domain of computation Ω. Generally speaking the wavelength of the eigenfunction (i.e. the number of peaks) decreases as the eigenfrequency increases.

Consider the discrete version of Equation (11.1.1), which is given by system (11.2.1). We obtain:

$$-\lambda_h^2 c^2 M\mathbf{U} = c^2 S\mathbf{U}. \tag{11.2.5}$$

The subscript h indicates that eigenvalues of the discretized problem are considered. The discretized system only has a finite number of eigenvalues, or put it differently: the resolution is finite on the discrete grid. The shortest wave that can be represented on a grid has wave length $O(2h)$. For eigenfunctions that can be represented well on the grid we have

$$|\lambda - \lambda_h| = O(h^p) \text{ and } ||\mathbf{U} - \mathbf{U}_h|| = O(h^p). \tag{11.2.6}$$

Since the eigenfrequencies of numerical and exact solution differ, the difference between the numerical solution and the exact solution increases as the simulation proceeds. This results in a *phase-shift error*. Moreover, this phase-shift error differs for the different eigenvibrations. This phenomenon is called *dispersion*. Since each solution can be written as a linear combination of eigenvibrations, there will be dispersion in the solution of equation (11.2.1) in relation to the solution of equation (11.1.1). This dispersion even exists for the eigenfunctions, which are represented well on the grid (i.e. eigenfunctions with a large wavelength, i.e. a small frequency). Therefore, the difference between the solution of (11.2.1) and the exact solution of the wave equation (11.1.1) increases as the interval of the time integration increases. Since the error is of the form $C(T - t_0)h^p$, one has to use a more accurate spatial discretization as T increases if the same absolute accuracy is to be maintained for the final stages of time interval as for the initial stages of the computation process.

As an example, we consider

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2}, \text{ for } 0 < x < 1, \tag{11.2.7}$$

subject to boundary conditions $u(0, t) = 0 = u(1, t)$ and some initial condition. It can be shown that the eigenvalues and eigenfunctions of the spatial differential operator with the given boundary conditions are respectively given by

$$\lambda_k = k\pi \text{ and } U_k = \sin k\pi x, \quad k = 1, 2, \ldots. \tag{11.2.8}$$

Note that $-\lambda_k^2$ are the actual eigenvalues of the spatial differential operator. Once a finite difference method with an equidistant grid for which $h = \frac{1}{N}$ (where h represents the step size) has been used, it follows that the eigenvalues and eigenvectors of the discretized problem are respectively given by

$$\lambda_{hk} = \frac{2\sin\left(\frac{1}{2}k\pi h\right)}{h}, \text{ and } \mathbf{U}_k = \begin{pmatrix} \sin k\pi h \\ \sin 2k\pi h \\ \dots \\ \sin (N-1)k\pi h \end{pmatrix}. \tag{11.2.9}$$

Note that $-\lambda_{hk}^2$ are the actual eigenvalues of the discretized problem. Note that the eigenvectors are exact. It can be demonstrated that $|\lambda_1 - \lambda_{h1}| = O(h^2)$ and that for $k = \frac{N}{2}$ the phase shift error is already significant. In the following exercise, the claims that we made in this paragraph are sustained by a motivation.

Exercise 11.2.2 *Consider the initial boundary value problem in Equation* (11.2.7)*.*

- *Verify by substitution that the eigenfunctions and eigenvalues are respectively given by*
$$U_k = \sin k\pi x \text{ and } \lambda_k = k\pi, \quad k = 1, 2, \dots. \tag{11.2.10}$$
Note that the eigenvalues of the Laplacian operator are given by $-\lambda_k^2$*.*
- *Use the Finite Difference method to create an equidistant discretization for which* $h = \frac{1}{N}$*, with h representing the stepsize.*
- *Verify by substitution that the eigenfunctions and eigenvectors of the discretized problem are respectively given by*
$$\mathbf{U}_k = \begin{pmatrix} \sin k\pi h \\ \sin 2k\pi h \\ \dots \\ \sin (N-1)k\pi h \end{pmatrix}, \text{ and } \lambda_{hk} = \frac{2\sin\left(\frac{1}{2}k\pi h\right)}{h}. \tag{11.2.11}$$
Note that the eigenvectors are exact. Show, further that $|\lambda_1 - \lambda_{h1}| = O(h^2)$ *and that for* $k = \dfrac{N}{2}$ *the phase-shift error is already significant.*

□

11.3 Numerical time integration

One possibility to integrate equation (11.2.1) numerically is to write it as a system of first order differential equations with respect to time:

$$\begin{aligned} \frac{d\mathbf{u}}{dt} &= \mathbf{v}, \\ M\frac{d\mathbf{v}}{dt} &= c^2 S\mathbf{u} + \mathbf{f}, \end{aligned} \tag{11.3.1}$$

with initial conditions $\mathbf{u}(t_0) = \mathbf{u}_0$ and $\mathbf{v}(t_0) = \mathbf{v}_0$. For this system the ordinary numerical methods for initial value problems can be used.

Example 11.3.1 *Forward Euler applied to System* (11.3.1)*, gives*

$$\begin{aligned} \frac{\mathbf{u}^{n+1}}{\Delta t} &= \frac{\mathbf{u}^n}{\Delta t} + \mathbf{v}^n, \\ M\frac{\mathbf{v}^{n+1}}{\Delta t} &= M\frac{\mathbf{v}^n}{\Delta t} + c^2 S\mathbf{u}^n + \mathbf{f}^n. \end{aligned} \tag{11.3.2}$$

Exercise 11.3.1 *Give the equations for* $\mathbf{u}$ *and* $\mathbf{v}$ *when a Crank-Nicholson time integration of System* (11.3.1) *is applied.* □

11.4 Stability of the numerical integration

From the conservation of energy of the solutions of both the wave equation and the discretization based on the method of lines, it follows that asymptotic stability does not make much sense here. A perturbation of the initial conditions will never vanish. A *fundamental solution* of the form $\mathbf{u}(t) = e^{\lambda t}\mathbf{u}, \mathbf{v}(t) = e^{\lambda t}\mathbf{v}$ of system (11.3.1) with $\mathbf{f} = 0$ has a purely imaginary λ as is shown in the next theorem.

Theorem 11.4.1 *Consider system* (11.3.1) *and let* λ *be an eigenvalue of the generalized eigenvalue problem*

$$\begin{aligned} \lambda\mathbf{u} &= \mathbf{v}, \\ \lambda M\mathbf{v} &= S\mathbf{u}. \end{aligned} \tag{11.4.1}$$

If M *is symmetric positive definite and if* S *is symmetric negative definite, then, the eigenvalues of the above generalized eigenvalue problem are purely imaginary.*

Proof: We substitute the upper equation into the bottom equation, to obtain:

$$\lambda^2 M\mathbf{u} = S\mathbf{u}, \tag{11.4.2}$$

which is the generalized eigenvalue problem for M and S. Next we show that the eigenvalues of the above generalized eigenvalue problem are real-valued and negative.

This amounts to establishing that the eigenvalues of $M^{-1}S$ are negative, real-valued. Since M is symmetric positive definite, the matrix $M^{-1/2}$ exists and $M^{-1}S$ is similar to $M^{1/2}M^{-1}SM^{-1/2} = M^{-1/2}SM^{-1/2}$. This matrix is symmetric and hence all the eigenvalues of $M^{-1}S$ are real-valued (we used the fact that matrices that are similar have the same eigenvalues). Furthermore, S is symmetric negative definite, i.e. $(S\mathbf{x}, \mathbf{x}) < 0$ for all $\mathbf{x} \neq \mathbf{0}$. Hence for $M^{-1/2}SM^{-1/2}$, we have $(M^{-1/2}SM^{-1/2}\mathbf{x}, \mathbf{x}) = (SM^{-1/2}\mathbf{x}, M^{-1/2}\mathbf{x}) < 0$ for all $\mathbf{x} \neq \mathbf{0}$. This implies that the eigenvalues of $M^{-1/2}SM^{-1/2}$ are negative and from similarity the eigenvalues of $M^{-1}S$ are negative as well. Combining this fact with the knowledge that the eigenvalues of $M^{-1}S$ are real-valued, it follows that λ^2 is negative and hence the eigenvalue λ is purely imaginary. This completes the proof. □

With the purely imaginary eigenvalues of the above generalized eigenvalue problem (11.4.1), it follows that the solution of system (11.3.1) is neutrally stable. An absolutely stable time integration method decays the error of the solution and also the solution itself as $t \to \infty$. Whereas an unstable time integration method blows up the error and the solution. This implies that with neither of these time integration methods, the wave equation can be integrated numerically up to any large time t. Hence we have to define an *end time* T and choose the time step Δt accordingly small. If $T = n\Delta t$ and $\lim_{\Delta t \to 0} |C(\lambda \Delta t)|^n = 1$ for a particular method, then, the wave equation can be integrated up to this bounded time T. Note that $n \to \infty$ as $\Delta t \to 0$.

11.5 Total dissipation and dispersion

Since the eigenvalues of (11.4.1) are purely imaginary the solution of (11.3.1) can be written as a linear combination of products of eigenvectors and undamped vi-

brations. Hence it is sufficient to consider a single differential equation of the form

$$\frac{dw}{dt} = i\mu w, \text{ subject to } w(t_0) = w_0. \tag{11.5.1}$$

The behavior of this differential equation qualitatively reflects the behavior of the total system (11.3.1). The exact solution is

$$w(t) = w_0 \mathrm{e}^{i\mu(t-t_0)}. \tag{11.5.2}$$

For the solution at $t^{n+1} = t_0 + (n+1)\Delta t$ we note that

$$w(t^{n+1}) = w(t^n)\mathrm{e}^{i\mu\Delta t}. \tag{11.5.3}$$

Hence the *amplification factor* of the exact solution is given by

$$C(i\mu\Delta t) = \mathrm{e}^{i\mu\Delta t} \Rightarrow |C(i\mu\Delta t)| = 1 \text{ and } \arg(C(i\mu\Delta t)) = \mu\Delta t. \tag{11.5.4}$$

The argument of the amplification factor, $\arg(C(i\mu\Delta t))$, is referred to as the *phase shift*. Hence in each time step there is a phase shift in the exact solution, whereas the modulus of the exact solution does not change.

Exercise 11.5.1 *Show that the complex differential equation* (11.5.1) *is equivalent to the system*

$$\begin{aligned} \frac{du}{dt} &= -\mu v \\ \frac{dv}{dt} &= \mu u, \end{aligned} \tag{11.5.5}$$

where $u = Re\{w\}$ and $v = Im\{w\}$. Show that $w(t) = Constant$ is equivalent to conservation of energy. □

For the numerical method, the following relation holds

$$w^{n+1} = C(i\mu\Delta t)w^n. \tag{11.5.6}$$

If the modulus of the amplification factor is larger than one, the energy increases in each time step. This is called *amplification*. Conversely, if the amplification factor is smaller than one the energy decreases. This is called *dissipation*.

Example 11.5.1 *The modulus of the amplification factor of Euler's method is*

$$|C(i\mu\Delta t)| = \sqrt{1 + (\mu\Delta t)^2}. \tag{11.5.7}$$

So the amplification of the method is $O(\mu^2\Delta t^2)$ accurate.

The phase shift per time step of a numerical method is defined by the argument of the amplification factor, i.e.

$$\Delta\Phi = \arg(C(i\mu\Delta t)) = \arctan\left(\frac{\mathrm{Im}\{C\}}{\mathrm{Re}\{C\}}\right). \tag{11.5.8}$$

Example 11.5.2 *The phase shift of the improved Euler method is given by*

$$\Delta\Phi = \arctan\left(\frac{\mu\Delta t}{1 - \frac{1}{2}(\mu\Delta t)^2}\right). \tag{11.5.9}$$

The phase error or *dispersion* is the difference between the exact and numerical phase shifts. This is referred to as *dispersion* because the phase shifts differ for the different values of μ_k in equation (11.3.1).

Exercise 11.5.2 *Show that the phase error of the improved Euler method per time step is* $O((\mu\Delta t)^3)$. □

The *total dissipation*, $D_n(i\mu\Delta t)$, is the product of the dissipations of all the time steps from t_0 up to the end time T. The *total dispersion*, $\Delta\Phi_n(i\mu\Delta t)$, is the sum over the phase errors of all the time steps. Note that we have $n\Delta t = T - t_0$. The total dissipation and the total dispersion are measures of the error in the numerical solution. As $\Delta t \to 0$ the total dissipation should tend to 1 and the total dispersion should tend to 0.

Exercise 11.5.3 *Why do we need*

$$\lim_{\Delta t \to 0} D_n(i\mu\Delta t) = 1? \tag{11.5.10}$$

□

As an illustration we calculate the total dissipation and total dispersion for the forward Euler method:

$$D_n = (C(i\mu\Delta t))^n = (1 + (\mu\Delta t)^2)^{\frac{T-t_0}{2\Delta t}}. \tag{11.5.11}$$

From a Taylor series of the exponential, we see that

$$1 \leq D_n \leq \left(\mathrm{e}^{(\mu\Delta t)^2}\right)^{\frac{T-t_0}{2\Delta t}}. \tag{11.5.12}$$

Subsequently, from a linearization of the exponential, we get

$$\mathrm{e}^{(\mu\Delta t)^2 \frac{T-t_0}{2\Delta t}} = 1 + O(\mu^2\Delta t). \tag{11.5.13}$$

So the condition $\lim_{\Delta t \to 0} D_n(i\mu\Delta t) = 1$ is satisfied. For the total dispersion we have

$$\begin{aligned} &\Delta\Phi_n(i\mu\Delta t) = n(\mu\Delta t - \Delta\Phi) = n(\mu\Delta t - \arctan(\mu\Delta t)) = \\ &n(\mu\Delta t - (\mu\Delta t + O((\mu\Delta t)^3))) = nO((\mu\Delta t)^3)) = O(\mu^3\Delta t^2). \end{aligned} \tag{11.5.14}$$

Note that $n\Delta t = T - t_0$ and that the exact phase shift is $\mu\Delta t$. This has been used in this expression. It is clear from the expression that the total dispersion tends to zero as the time step tends to zero. In Figures 11.1 and 11.2 the total dissipation and dispersion are plotted as a function of the time-step Δt.

This total dispersion and total dissipation can be investigated for other time integration methods as well. We leave this as an exercise to the reader.

11.6 Direct time integration of the second order system

In principle it is not necessary to write equation (11.3.1) as a system of two first order differential equations. A lot of methods are available to integrate a second order differential equation of the form

$$\frac{d^2\mathbf{y}}{dt^2} = f(\mathbf{y}, t) \tag{11.6.1}$$

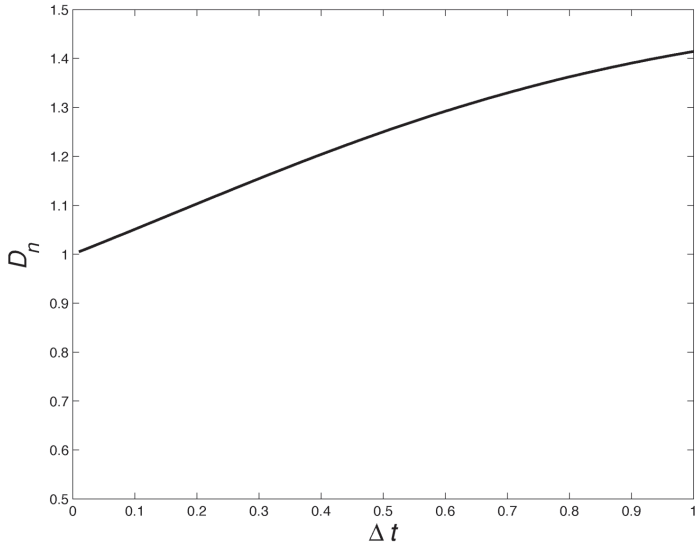

Figure 11.1: Dissipation of the forward Euler method for $\mu = 1$ and $T - t_0 = 1$.

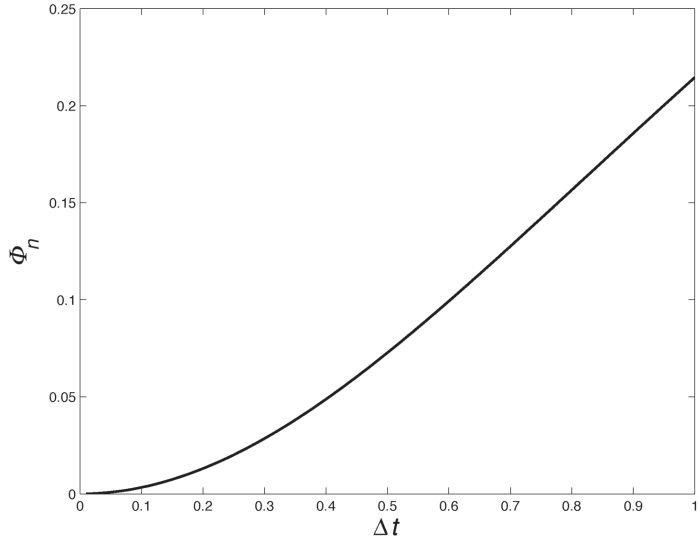

Figure 11.2: Dispersion of the forward Euler method for $\mu = 1$ and $T - t_0 = 1$.

directly. For a comprehensive survey of numerical methods to solve this system of second order differential equations, we refer to [23]. In this course we will treat two example schemes, applied to (11.2.1):

1. Explicitly
$$M\mathbf{u}^{n+1} - 2M\mathbf{u}^n + M\mathbf{u}^{n-1} = \Delta t^2 \left(c^2 S\mathbf{u}^n + \mathbf{f}^n \right). \tag{11.6.2}$$

2. Implicitly
$$\begin{aligned} M\mathbf{u}^{n+1} - 2M\mathbf{u}^n + M\mathbf{u}^{n-1} = \\ \frac{\Delta t^2}{4} \left(c^2 (S\mathbf{u}^{n+1} + 2S\mathbf{u}^n + S\mathbf{u}^{n-1}) + \mathbf{f}^{n+1} + 2\mathbf{f}^n + \mathbf{f}^{n-1} \right). \end{aligned} \tag{11.6.3}$$

Both methods are consistent of $O(\Delta t^2)$ in time. These methods are referred to as *three step* schemes, which implies that before one starts using these schemes, one first has to use an other method, such as Euler explicit:

$$\mathbf{u}_1 = \mathbf{u}_0 + \Delta t \mathbf{v}_0. \tag{11.6.4}$$

Using the explicit Euler method for the first step is satisfactory, since its error for the first step is $O(\Delta t^2)$.

The Equations (11.6.2) and (11.6.3) are special cases of the popular Newmark-(α, β) scheme. This scheme is usually written in a three-level form based on displacement $\mathbf{u}$, velocity $\mathbf{v}$ and acceleration $\mathbf{a}$. It uses a Taylor expansion, where the higher order terms are averaged.
Newmark reads:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \mathbf{v}^n + \frac{\Delta t^2}{2}((1-2\beta)\mathbf{a}^n + 2\beta \mathbf{a}^{n+1}), \tag{11.6.5}$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta t((1-\gamma)\mathbf{a}^n + \gamma \mathbf{a}^{n+1}), \tag{11.6.6}$$

$$M\mathbf{a}^{n+1} + c^2 S\mathbf{u}^{n+1} = \mathbf{f}^{n+1}. \tag{11.6.7}$$

At $t = t_0$ we solve $\mathbf{a}^0$ from the equation of motion (11.6.7). In the following steps we substitute (11.6.5) in (11.6.7) to get an equation for $\mathbf{a}^{n+1}$. Finally (11.6.5) and (11.6.6) are used to compute $\mathbf{u}^{n+1}$ and $\mathbf{v}^{n+1}$.
It is possible to rewrite Newmark as a three-step scheme:

$$\begin{aligned} (M + \beta c^2 \Delta t^2 S)\mathbf{u}^{n+1} - (2M - (\frac{1}{2} + \gamma - 2\beta)c^2 \Delta t^2 S)\mathbf{u}^n + \\ (M + (\frac{1}{2} - \gamma + \beta)c^2 \Delta t^2 S)\mathbf{u}^{n-1} = \Delta t^2 \mathbf{F}, \end{aligned} \tag{11.6.8}$$

with

$$\mathbf{F} = (\frac{1}{2} - \gamma + \beta)\mathbf{f}^{n-1} + (\frac{1}{2} + \gamma - 2\beta)\mathbf{f}^n + \beta \mathbf{f}^{n+1}. \tag{11.6.9}$$

Remark
(11.6.7) can not be used to compute $\mathbf{a}$ at boundaries with prescribed displacements at $t = t_0$. Why not? In practice one often takes $\mathbf{a} = 0$ in that case.
An alternative is to use a Taylor series expansion at $t = t_0 + \Delta t$ and to express $\mathbf{a}^0$ in $\mathbf{u}^0$, $\mathbf{v}^0$, and $\mathbf{u}^1$ at that boundary.

Exercise 11.6.1 *Prove that (11.6.8), (11.6.9) follows from (11.6.5)-(11.6.7).*
Hint: Substitute (11.6.6) in (11.6.5) to eliminate $\mathbf{v}^n$*, and write the equation for the previous timestep to get an expression of the form:*

$$\mathbf{u}^{n-1} = \mathbf{u}^n - \mathbf{v}^n \Delta t + \frac{\Delta t^2}{2}((1 - 2(\gamma - \beta))\mathbf{a}^{n-1} + 2(\gamma - \beta)\mathbf{a}^n). \tag{11.6.10}$$

Add (11.6.5) and (11.6.10) to get an expression for $\mathbf{u}^{n+1}$ *and* $\mathbf{a}^{n+1}$.
Then use the equation of motion (11.6.7). □

Exercise 11.6.2 *Show that the Newmark scheme reduces to the explicit central difference scheme ((11.6.2)) if* $\beta = 0$ *and* $\gamma = \frac{1}{2}$. □

Exercise 11.6.3 *Show that the Newmark scheme reduces to the implicit central difference scheme ((11.6.2)) if* $\beta = \frac{1}{4}$ *and* $\gamma = \frac{1}{2}$. □

Exercise 11.6.4 *Show that the three step implicit scheme (11.6.3) is identical to Crank-Nicholson's method for* (11.3.1)*. (Hint: write out the steps for* n *and* $n + 1$ *and eliminate all the* $\mathbf{v}$*'s.) Note that the first step of the three step method should be taken with Crank-Nicholson's method instead of the previously mentioned Euler explicit method.* □

11.7 The CFL criterion

From the section about the numerical time integration, it is clear that the time step plays an important role in the numerical integration. In general the time step Δt and step size Δx cannot be chosen independently. This was already observed for Euler's method. In 1928 Courant, Friedrichs and Lewy formulated a condition for the time step for the numerical solution to be a representation of the exact solution. Their condition was obtained by using a physical argument. Commonly one refers to it as the CFL criterion. Often this CFL condition is used in relation with stability of a numerical method. Strictly, this is not true since the CFL criterion represents a condition for convergence. In the following text an intuitive justification of the CFL criterion will be given. It is possible though to derive the CFL criterion in full mathematical rigor.

The solution of the wave equation can be represented by a superposition of linear waves, which all have a velocity c. Consider the solution at any node x_i at time t^j, then, within a time interval Δt, this *point source* influences the solution within the distance $c\Delta t$ from position x_i. Within a time interval Δt, the solution at locations with distance larger than $c\Delta t$ from x_i is not influenced by the solution at x_i on t_j. Usually this is referred to as the *region of influence* of $u(x_i, t^j)$. Vice versa, $u(x_i, t^j)$ is determined by the *point sources* of $u(x, t^{j+1} - \tau)$, with $|x - x_i| < c\tau$ for $\tau < \Delta t$. This region of influence is indicated by the dashed part in Figure 11.3. For the explicit time integration of the wave equation, the spatial discretization is done at time t^j. For the finite differences solution with one spatial coordinate at x_i on t^j, one uses $u(x_i, t^j)$, $u(x_{i-1}, t^j)$ and $u(x_{i+1}, t^j)$, i.e.

$$\left.\frac{d^2u}{dx^2}\right|_{t=t^j} = \frac{u(x_{i-1}, t^j) - 2u(x_i, t^j) + u(x_{i+1}, t^j)}{\Delta x^2}. \tag{11.7.1}$$

The CFL criterion of an explicit scheme for the wave equation is as follows: *The region of influence of* $u(x, t^{j+1})$ *with* $|x - x_i| < c\tau$ *for* $\tau < \Delta t$ *(hence around* x_i*), may not contain any locations at* t^j *outside the interval of the grid nodes* $(x_{i-1}, x_i, x_{i+1}$ *here), which are used for the discretization of the second order partial derivatives of the wave equation.*

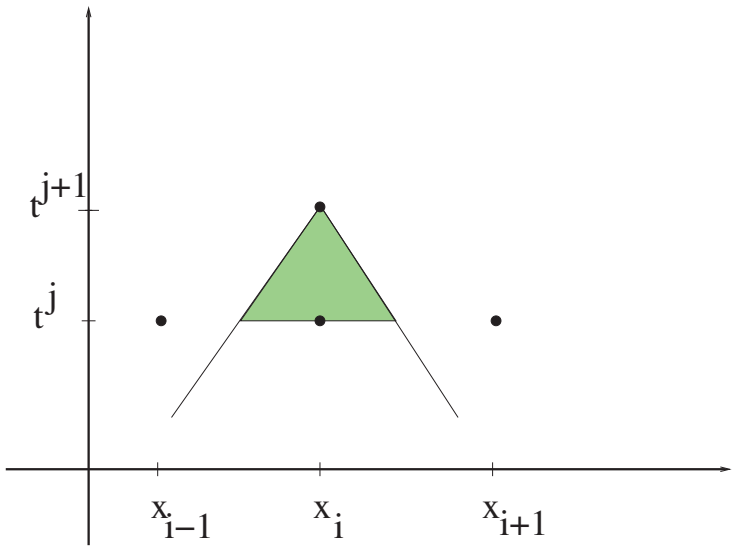

Figure 11.3: The solution at x_i on t^{j+1} is determined by u(x,$t^{j+1} - \tau$) where x $\in$ (x_j − cτ,x_j + cτ) for $\tau < \Delta$t. The region of influence is indicated by the grey region. This situation satisfies the CFL criterion.

The CFL criterion guarantees that the numerical solution is determined only by all the point sources that physically have an influence on this solution. In the case of Figure 11.3, it turns out that the region of influence, for $\tau < \Delta t$, only contains locations within the interval (x_{i-1}, x_{i+1}) and hence for this Δt the CFL criterion is satisfied and convergence of the numerical solution is to be expected. Before an example is treated, the following important aspects should be noted:

Remarks

1. For a *three step* scheme it is sufficient to check the CFL criterion for the final two steps. By induction it follows that the criterion is satisfied for all steps.
2. For an implicit scheme the CFL criterion is irrelevant. Since, then, the entire previous time step determines the solution on the present time step.

An example of the derivation of the CFL criterion is treated:

Example 11.7.1 *Consider the explicit time integration of the wave equation in one dimension with equidistant nodes:*

$$u_i^{n+1} - 2u_i^n + u_i^{n-1} = \left(\frac{c\Delta t}{\Delta x}\right)^2 (u_{i+1}^n - 2u_i^n - u_{i-1}^n). \tag{11.7.2}$$

The region of determination of u_i^{n+1} is then given by the interval $(x_i - c\tau, x_i + c\tau)$ for $\tau < \Delta t$. The nodes for the spatial discretization are x_{i-1}, x_i and x_{i+1}. The interval that is defined by these nodes (hence (x_{i-1}, x_{i+1})) must contain the region of influence $(x_i - c\tau, x_i + c\tau)$ for $\tau < \Delta t$. Hence, the CFL criterion for this case is given by $x_i - c\tau > x_{i-1}$ and $x_i + c\tau < x_{i+1}$ for $\tau < \Delta t$. Since the nodes are equidistant and $\Delta x = x_{i+1} - x_i$, this implies the following CFL-criterion:

$$\frac{c\Delta t}{\Delta x} \leq 1. \tag{11.7.3}$$

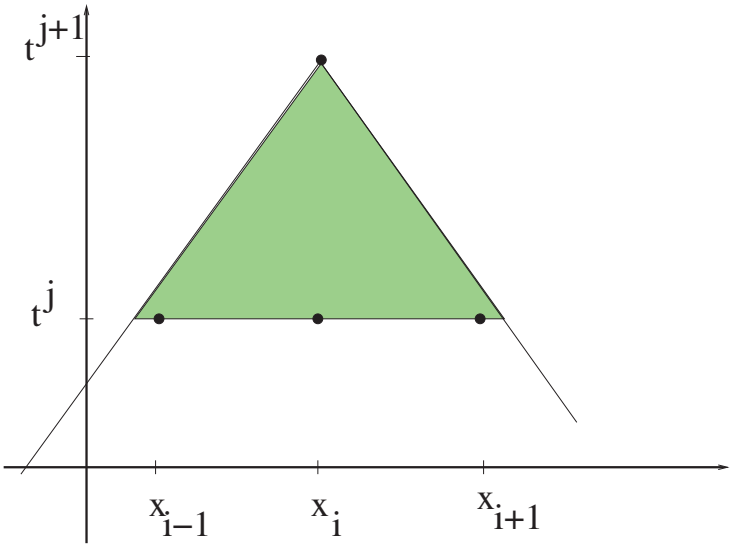

Figure 11.4: The solution at x_i on t^{j+1} is determined by u(x,$t^{j+1}-\tau$) where x $\in$ (x_j $-$ cτ,x_j + cτ) for $\tau < \Delta$t. The region of influence is indicated by the grey region and some part of the region of influence is outside the interval (x_{i-1},x_{i+1}). Hence this situation violates the CFL criterion.

An example of a region of influence for a time step that does not satisfy the CFL criterion is shown in Figure 11.4.

Exercise 11.7.1 *Check that for the wave equation with one spatial coordinate, the Euler forward method by*

$$\begin{aligned} u_i^{n+1} &= u_i^n + \Delta t v_i^n \\ v_i^{n+1} &= v_i^n + \frac{c^2 \Delta t}{\Delta x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n), \end{aligned} \tag{11.7.4}$$

cannot satisfy the CFL criterion. If the first equation is replaced with

$$u_i^{n+1} = u_i^n + \frac{\Delta t}{4}(v_{i-1}^n + 2v_i^n + v_{i+1}^n), \tag{11.7.5}$$

there is a CFL criterion. Give the CFL criterion for this case. □

11.8 Summary of Chapter 11

This chapter has dealt with numerical methods for the solution of the (hyperbolic) wave equation. The hyperbolic nature of the wave equation is important for the nature of the numerical solutions. To solve the PDE the method of lines has been used. It first deals with the spatial derivatives and considers time integration of the resulting system of ODE's as a separate problem.

A direct time integration scheme for the second derivative of the time has also been presented. The numerical amplification factor for the dissipation and the phase shift of the numerical solution have been defined and analyzed. Finally,

the derivation of the CFL-criterion, using the concept of the region of influence in the x, t plane, has been given. This CFL criterion is necessary for the numerical solution to be a representation of the exact solution.

Chapter 12

The transport equation

Objectives

The transport equation is fundamental in modeling (multi-phase) flow in porous media, such as underground oil, gas and water reservoirs. Some engineering disciplines, where the transport equation plays an important role as well, are geosciences, aerospace engineering and naval engineering. The transport equation is also referred to as a first order hyperbolic conservation law and an important application in aerospace engineering involves modeling of air flow around aircraft. The backbone for understanding the nature of the solutions and boundary conditions of the transport equation lies in the analysis of the characteristics of the solutions. Further, some classical numerical methods to solve hyperbolic conservation laws will be presented. The presentation is given for configurations with one spatial coordinate only. Finally, some mathematical theory for the transport equation will be presented. *Traveling wave solutions* for Burgers equation are analyzed and subsequently the nature of the solutions of the Buckley-Leverett equation is discussed.

12.1 Introduction

The transport equation describes transport of one or various components in n dimensions. In this chapter we limit ourselves to transport in one spatial dimension. The most general form of a transport equation in *conservative form* is:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} = \mathbf{g}(\mathbf{u}, x, t). \tag{12.1.1}$$

with $\mathbf{u} = (u_1, \ldots, u_m)^T$ denoting the vector with the transported quantities and $\mathbf{f} = (f_1, \ldots, f_m)^T$ the *flux-vector*. If the vector $\mathbf{g}$ does not depend on x and t, then the problem is called *autonomous*. In many transport problems the right-hand side involves a chemical reaction, whose rate often only depends on the solution. Hence, many transport problems are autonomous.

In the literature the transport problem is often referred to as hyperbolic, e.g. a hyperbolic conservation law. Although the equation certainly does not satisfy the standards for hyperbolicity as in Chapter 2, this classification does make sense, since the solutions of Equation (12.1.1) can be represented by waves, just like those of genuine hyperbolic partial differential equations. This is justified in the following exercise:

Exercise 12.1.1 *Show that the transport equation of two components*

$$\begin{cases} \dfrac{\partial u}{\partial t} + c\dfrac{\partial v}{\partial x} = 0 \\ \dfrac{\partial v}{\partial t} + c\dfrac{\partial u}{\partial x} = 0 \end{cases} \tag{12.1.2}$$

is equivalent to the wave equation $\dfrac{\partial^2 u}{\partial t^2} = c^2 \dfrac{\partial^2 u}{\partial x^2}$.

In non-conservative form, Equation (12.1.1) has the following shape:

$$\frac{\partial \mathbf{u}}{\partial t} + A(\mathbf{u})\frac{\partial \mathbf{u}}{\partial x} = \mathbf{g}, \tag{12.1.3}$$

with $a_{ij} = \frac{\partial f_i}{\partial u_j}$. Hence A is the Jacobian of the flux-vector. For regular transport A must have real eigenvalues. In the literature 12.1.1 and 12.1.3 are commonly referred to as hyperbolic if the Jacobian A has real eigenvalues (i.e. regular transport). Imposing initial and boundary conditions for (12.1.1) and (12.1.3) is usually not trivial. This will be illustrated by use of the characteristics for the transport of one component.

12.2 Characteristics

Let us consider the equation

$$\frac{\partial u}{\partial t} + a(u)\frac{\partial u}{\partial x} = g(u), \tag{12.2.1}$$

which describes transport of one component in non-conservative form. We consider a curve in the (x,t) plane, parameterized by s with the property

$$\frac{dx}{ds} = \rho(s)a(u) \text{ and } \frac{dt}{ds} = \rho(s), \tag{12.2.2}$$

then along this curve we obtain from the total derivative of u with respect to s:

$$\frac{du}{ds} = \frac{\partial u}{\partial x}\frac{dx}{ds} + \frac{\partial u}{\partial t}\frac{dt}{ds} = \rho(s)g(u). \tag{12.2.3}$$

This means that if the value of u is known at a certain point, i.e. $u(x_0,t_0) = u_0$, then, the value over a curve is defined for $(x(s),t(s),u(x(s),t(s))$, is the solution of the coupled system of ordinary differential equations:

$$\frac{dx}{ds} = \rho(s)a(u), \quad \frac{dt}{ds} = \rho(s), \quad \frac{du}{ds} = \rho(s)g(u). \tag{12.2.4}$$

with initial conditions $x(0) = x_0$, $t(0) = t_0$ and $u(0) = u_0$. The (x,t) curve of Equation (12.2.2) is referred to as a *characteristic*, the system (12.2.2) is called the characteristic equation and Equation (12.2.3) is referred to as the characteristic relation. The Equations (12.2.2) and (12.2.3) give the solution of the partial differential equation in the form of a system of coupled ordinary differential equations. One also expresses this property as follows: along the characteristics the PDE changes into an ODE. Note that if $g = 0$, then the solution u is constant along a characteristic and the quantity u is transported along the characteristics. The choice of ρ is arbitrary, it influences the parameterization and not the solution itself. If g does not depend on u then the characteristics can be determined by just solving Equation (12.2.2). However, for cases in which a depends on u, the complete Equation (12.2.4) must be solved. Then, one obtains the characteristic and the solution at the same time.

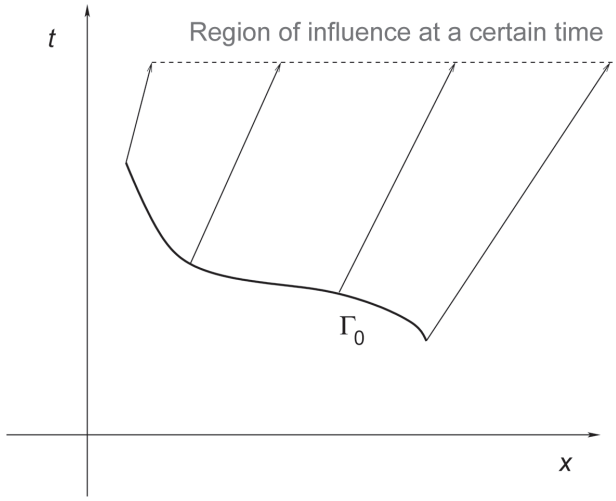

Figure 12.1: The region of influence of Γ_0 at a certain time. This region is indicated by the dashed line. The arrows indicate the characteristics of the solution.

Exercise 12.2.1 *Show that the characteristic equation corresponding to the PDE*

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0. \tag{12.2.5}$$

is given by

$$\frac{dx}{ds} = \rho c \text{ and } \frac{dt}{ds} = \rho. \tag{12.2.6}$$

Show that this gives $\dfrac{dx}{dt} = c$. *Hence all lines of the form* $x - ct =$ *constant are characteristics. Suppose that* u *is given at* $t = 0$ *on the interval* $(0,1)$, *show that then the solution at time* t *is determined on the interval* $x \in (ct, 1+ct)$ *and show that it is given by* $u(x,t) = u_0(x-ct)$.

We formulate the initial value problem as follows:

Let Γ_0 *be a curve in the* (x,t)*-plane, such that each characteristic intersects* Γ_0 *only once. Let* u *be given on* Γ_0. *Then, the solution is determined on a strip* Σ, *which is constructed by the union of all the characteristics that intersect* Γ_0. *The solution on each characteristic is determined by the system of ordinary differential equations* (12.2.4) *with as the initial condition the values of the solution at* Γ_0. *The situation has been pictured in Figure 12.1.*

The strip Σ is called the *region of influence* of Γ_0.

Exercise 12.2.2 *Why is a characteristic not allowed to intersect* Γ_0 *twice for a general initial condition on* Γ_0? *What condition should be satisfied if the characteristic intersects the curve* Γ_0 *twice?*

Exercise 12.2.3 *Given the differential equation*

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = -u, \tag{12.2.7}$$

with initial condition $u(x,0) = u_0(x)$ *on the interval* $0 \le x \le 1$.

1. *What is the equation for the characteristics?*
2. *What is the characteristic relation?*
3. *What is the region of influence of the interval* $0 \leq x \leq 1$*?*

Exercise 12.2.4 *Given the differential equation*

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 1, \tag{12.2.8}$$

with initial conditions on $\Gamma_0 = \Gamma_1 \cup \Gamma_2$ *with*

1. $\Gamma_1 = \{(x,t) : t = 0, 0 \leq x \leq 1\}$
   $\Gamma_2 = \{(x,t) : x = 0, 0 \leq t < \infty\}$.
2. $\Gamma_1 = \{(x,t) : t = 0, 0 \leq x \leq 1\}$
   $\Gamma_2 = \{(x,t) : x = 1, 0 \leq t < \infty\}$.

In which of these two cases is the problem well-posed for a general initial condition? What is the region of influence of u?

Exercise 12.2.5 *Give for the differential equation*

$$\frac{\partial u}{\partial t} + t\frac{\partial u}{\partial x} = f, \tag{12.2.9}$$

the region of influence of the start curve: $\Gamma_0 = \{(x,t) : -1 \leq t \leq 1, x = 0\}$. *Are all choices for u allowed on the curve* Γ_0*?*

In case of smooth solutions of u with respect to x and t, it is necessary that two characteristics, which originate from different locations on Γ_0 with different initial values, do not intersect.

12.3 Some classical numerical procedures

In the past many numerical methods to solve the transport equation were based on the characteristics of the solution. However, the popularity of these methods decreased and therefore they are not treated in this book. These methods were gradually replaced by the *fixed grid* methods. In this section first the classical methods of central and upwind discretization are analyzed. Subsequently, the Lax-Wendroff scheme for smooth solutions and the use of fluxlimiters for discontinuous solutions are introduced as higher order methods to solve the transport equation.

12.3.1 Central discretization and upwind discretization

We consider again the transport equation

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \quad \Longleftrightarrow \quad \frac{\partial u}{\partial t} = -\frac{\partial f(u)}{\partial x}. \tag{12.3.1}$$

In this text an equidistant distribution of the gridnodes is considered.
We use a Finite Volume Method, hence we integrate over a control volume (see Figure 12.2), which gives on time-step t_j:

$$\int_{\Omega_i} \frac{\partial u}{\partial t}(x,t_j)dx = -\int_{\Omega_i} \frac{\partial f(u(x,t_j))}{\partial x}dx =$$

$$= -\left[f(u(x_{i+\frac{1}{2}},t_j)) - f(u(x_{i-\frac{1}{2}},t_j))\right].$$

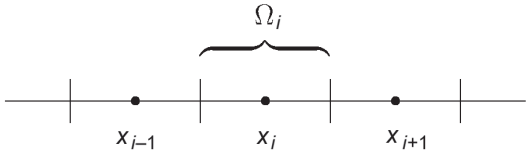

Figure 12.2: A gridcell used for the Finite Volume Discretization.

We define $f^j_{i-\frac{1}{2}} := f(u(x_{i-\frac{1}{2}}, t_j))$ and $f^j_{i+\frac{1}{2}} := f(u(x_{i+\frac{1}{2}}, t_j))$, as the flux on both boundaries of the gridcell Ω_i at time step $j\Delta t$. The flux entering Ω_{i+1} and the flux leaving Ω_i are balanced with the accumulation in Ω_i. The integral on the left-hand side of the above equation is approximated as follows:

$$\int_{\Omega_i} \frac{\partial u}{\partial t}(x, t_j)dx \approx \frac{\partial u}{\partial t}(x_i, t_j)\Delta x \approx \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t}\Delta x.$$

Define $u_i^{j+1} = u(x_i, t_{j+1})$ and $u_i^j = u(x_i, t_j)$, in this way the following discretization of (12.3.1) is obtained:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} + \frac{f^j_{i+\frac{1}{2}} - f^j_{i-\frac{1}{2}}}{\Delta x} = 0. \tag{12.3.2}$$

We analyze difference equation (12.3.2). We consider the simple example $f(u) = u$. We discretize the above equation by the use of the second order *central discretization*. The fluxes on the boundaries of Ω_i are approximated

$$f^j_{i+\frac{1}{2}} := f(u^j_{i+\frac{1}{2}}) = u^j_{i+\frac{1}{2}} \approx \frac{u^j_{i+1} + u^j_i}{2},$$

$$f^j_{i-\frac{1}{2}} := f(u^j_{i-\frac{1}{2}}) = u^j_{i-\frac{1}{2}} \approx \frac{u^j_{i-1} + u^j_i}{2}.$$

With these approximations, a central discretization results from Equation (12.3.2):

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} + \frac{u^j_{i+1} - u^j_{i-1}}{2\Delta x} = 0, \tag{12.3.3}$$

In Chapter 3 it is derived that the truncation error in the spatial discretization is of second order. It turns out that the above discretization is prone to unphysical oscillations. For the stationary convection-diffusion equation this is motivated in Chapter 3. This will be analyzed later in this section by the use of the Von Neumann stability analysis.

Due to this oscillatory behavior it is preferable to use an alternative method. We derive this discretization method by the use of characteristics. The points (x_{i-1}, t^j), (x_i, t^j) and (x_i, t^{j+1}) in the (x,t)-plane are sketched in Figure 12.3. Over each characteristic the value of u is constant. Since, $\frac{dx}{dt} = 1$, the characteristics are parallel to the line $x = t$ in the (x,t)-plane. So following the characteristic through (x_i, t^{j+1}) we end up at the point $(x_i - \Delta t, t^j)$ at the line $t = t^j$ (see Figure 12.3). Hence, we have $u_i^{j+1} = u^j(x_i - \Delta t)$. Since $u^j(x_i - \Delta t)$ is not a value on a node, its value is

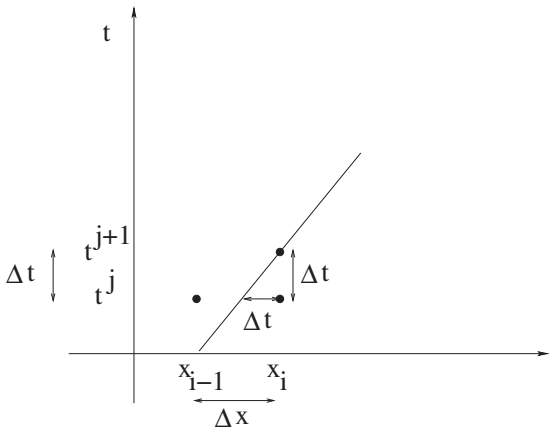

Figure 12.3: Derivation of the first order upwind discretization.

computed by linear interpolation between the values u_{i-1}^j and u_i^j, to obtain

$$u^j(x_i - \Delta t) = u_{i-1}^j \frac{\Delta t}{\Delta x} + u_i^j \frac{\Delta x - \Delta t}{\Delta x} = u_i^j + \frac{\Delta t}{\Delta x}(u_{i-1}^j - u_i^j). \tag{12.3.4}$$

Keeping in mind that $u_i^{j+1} = u^j(x_i - \Delta t)$, the above equation is written as

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} + \frac{u_i^j - u_{i-1}^j}{\Delta x} = 0. \tag{12.3.5}$$

The above equation is commonly referred to as the first order *upwind* discretization.

In the coming text the accuracy and stability issues are investigated for these two discretization for the linear transport equation, where $f(u) = u$.

The Taylor Series around $x = x_i$ and $t = t^j$ for u_i^{j+1} and u_{i-1}^j are given by

$$\begin{aligned} u_{i-1}^j &= u_i^j - \Delta x \cdot \frac{\partial u}{\partial x} + \frac{(\Delta x)^2}{2}\frac{\partial^2 u}{\partial x^2} + \cdots, \\ u_i^{j+1} &= u_i^j + \Delta t \cdot \frac{\partial u}{\partial t} + \frac{(\Delta t)^2}{2}\frac{\partial^2 u}{\partial t^2} + \cdots. \end{aligned}$$

Substitution of the above equations into Equation (12.3.5) gives

$$\frac{u_i^j + \Delta t \cdot \frac{\partial u}{\partial t} + \frac{(\Delta t)^2}{2}\frac{\partial^2 u}{\partial t^2} + \cdots - u_i^j}{\Delta t} +$$

$$+ \frac{u_i^j - u_i^j + \Delta x \cdot \frac{\partial u}{\partial x} - \frac{(\Delta x)^2}{2}\frac{\partial^2 u}{\partial x^2} + \cdots}{\Delta x} = 0.$$

Then, after neglecting higher order terms one obtains:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \frac{\Delta x}{2}\frac{\partial^2 u}{\partial x^2} - \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2}. \tag{12.3.6}$$

We remark that $\frac{\partial u}{\partial t} = -\frac{\partial u}{\partial x} \Longrightarrow \frac{\partial^2 u}{\partial t^2} = -\frac{\partial^2 u}{\partial t \partial x} = -\frac{\partial^2 u}{\partial x \partial t} = -\frac{\partial}{\partial x}(-\frac{\partial u}{\partial x}) = \frac{\partial^2 u}{\partial x^2}$ (provided the second order partial derivatives are continuous). Substitution into (12.3.6)

gives the following equation:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \left(\frac{\Delta x}{2} - \frac{\Delta t}{2}\right)\frac{\partial^2 u}{\partial x^2} = \mathbb{D}\frac{\partial^2 u}{\partial x^2}, \tag{12.3.7}$$

with $\mathbb{D} = \dfrac{\Delta x}{2} - \dfrac{\Delta t}{2}$. This equation is a convection-diffusion equation. From the above equation it is clear that the discretization error for this upwind discretization is first order for the time step and the grid spacing. Therefore, this formula is referred to as first order upwind. The dispersion term, $\mathbb{D}\dfrac{\partial^2 u}{\partial x^2}$, is called the numerical diffusion. We know that the convection-diffusion equation has a stable solution if and only if $\mathbb{D} \geq 0$. So stability is guaranteed if

$$0 \leq \mathbb{D} = \frac{\Delta x}{2} - \frac{\Delta t}{2} \quad \Longleftrightarrow \quad \frac{\Delta t}{\Delta x} \leq 1. \tag{12.3.8}$$

Inequality (12.3.8) is commonly called the *CFL-condition* after Courant-Friedrichs-Lewy. The values of Δt and Δx have to satisfy the CFL-condition. Note that if $\mathbb{D} = 0$ (or $\dfrac{\Delta t}{\Delta x} = 1$), then Equation (12.3.7) reduces to (12.3.1) with $f(u) = u$. For this case there is no numerical diffusion. Then, the error consists of higher order terms only. In most practical situations with variable coefficients or non-linearities, it is impossible to tune Δt and Δx such that $\mathbb{D} = 0$.

Note that we use an explicit time-integration, which has a time-step restriction for stability. For an implicit time integration one obtains:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = -\frac{u_i^{j+1} - u_{i-1}^{j+1}}{\Delta x}.$$

Using a similar procedure with Taylor-expansion as for the explicit scheme, one obtains

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \frac{1}{2}(\Delta t + \Delta x)\frac{\partial^2 u}{\partial x^2},$$

hence $\mathbb{D} := \frac{1}{2}(\Delta t + \Delta x) > 0$ for all $\Delta t, \Delta x > 0$. Above equation always has a stable solution, however, errors due to space discretization and time integration do not tend to cancel each other. Therefore, an implicit time integration method is not widely used.

12.3.1.1 Von Neumann stability analysis

Next, the issue of stability is treated. In the present section the method of Von Neumann is used to analyze stability. This method is based on the estimation of the eigenvalues of the discretization matrix. The procedure can be used for PDE's with constant coefficients and equidistant grids only. The method involves the use of a discrete Fourier series and is formally applicable to rectangular geometries with periodical boundary conditions. In this section only one spatial co-ordinate is considered. Let us consider the function $\hat{u}$ and the domain $x \in [0, 1]$, then, a Fourier series of $\hat{u}$ is given by

$$\hat{u}(x, t) = \sum_{\alpha=1}^{\infty} \rho_\alpha(t) \mathrm{e}^{-2\pi\alpha x i}. \tag{12.3.9}$$

The functions $\rho_\alpha(t)$ are referred to as Fourier coefficients. Let N denote the number of grid nodes. Then, the above relation is written for the function $\hat{u}$ on the grid, i.e.

$\hat{u}_k^n = \hat{u}(k\Delta x, n\Delta t)$:

$$\hat{u}_k^n = \sum_{\alpha=1}^{N-1} \rho_\alpha^n \mathrm{e}^{-2\pi\alpha k\Delta x i}, \tag{12.3.10}$$

where we define $\rho_\alpha^n = \rho_\alpha(n\Delta t)$. The above relation can represent a Discrete Fourier Transform of the perturbed solution. For stability we require that ρ_α^n stays bounded as $n \to \infty$. Consider the following example

$$\frac{u_k^{n+1} - u_k^n}{\Delta t} + \frac{u_{k+1}^n - u_{k-1}^n}{2\Delta x} = 0, \tag{12.3.11}$$

i.e. a central discretization with Euler forward (explicit) time integration. Then, substitution of (12.3.10) into (12.3.11) yields

$$\sum_{\alpha=1}^{N-1} (\rho_\alpha^{n+1} - \rho_\alpha^n)\mathrm{e}^{-2\pi\alpha k\Delta x i} = \frac{\Delta t}{2\Delta x} \sum_{\alpha=1}^{N-1} \rho_\alpha^n \left(\mathrm{e}^{-2\pi\alpha(k-1)\Delta x i} - \mathrm{e}^{-2\pi\alpha(k+1)\Delta x i}\right). \tag{12.3.12}$$

Collecting all terms for a fixed value of α, ρ_α^{n+1} and ρ_α^n, and division by $\mathrm{e}^{-2\pi\alpha k\Delta x i}$, gives

$$\rho_\alpha^{n+1} = \rho_\alpha^n \left[1 + \frac{\Delta t}{2\Delta x}\left(\mathrm{e}^{2\pi\alpha\Delta x i} - \mathrm{e}^{-2\pi\alpha\Delta x i}\right)\right]. \tag{12.3.13}$$

Using $\sin(\theta) = \dfrac{\mathrm{e}^{\theta i} - \mathrm{e}^{-\theta i}}{2i}$, gives

$$\rho_\alpha^{n+1} = \rho_\alpha^n \left[1 + i\frac{\Delta t}{\Delta x}\sin(2\pi\alpha\Delta x)\right]. \tag{12.3.14}$$

The ratio $\dfrac{\rho_\alpha^{n+1}}{\rho_\alpha^n}$ represents an amplification factor. A condition for stability is

$$\left|\frac{\rho_\alpha^{n+1}}{\rho_\alpha^n}\right| \leq 1, \tag{12.3.15}$$

i.e. the modulus of the ratio between the Fourier coefficients ρ_α^n at consecutive time-steps may not be larger than one. Note that the Fourier coefficients are not real, then, for the central explicit discretization follows

$$\left|\frac{\rho_\alpha^{n+1}}{\rho_\alpha^n}\right|^2 = 1 + \left(\frac{\Delta t}{\Delta x}\right)^2 \sin^2\left(\frac{2\pi\alpha}{N}\right) > 1, \tag{12.3.16}$$

and hence the central discretization with Euler-forward time integration is always unstable, regardless the value of Δt and Δx.

An other example with conditional stability is considered in the following exercise, where an explicit time-integration is considered for an upwind discretization scheme.

Exercise 12.3.1 *Consider the discretization method with Euler forward time integration and a first order upwind spatial discretization:*

$$\frac{u_k^{n+1} - u_k^n}{\Delta t} + \frac{u_k^n - u_{k-1}^n}{\Delta x} = 0. \tag{12.3.17}$$

Use the Von Neumann stability of this section to show that the above mentioned formula gives stability if $\dfrac{\Delta t}{\Delta x} < 1$.

Bear in mind that the Fourier coefficients are not real in general.

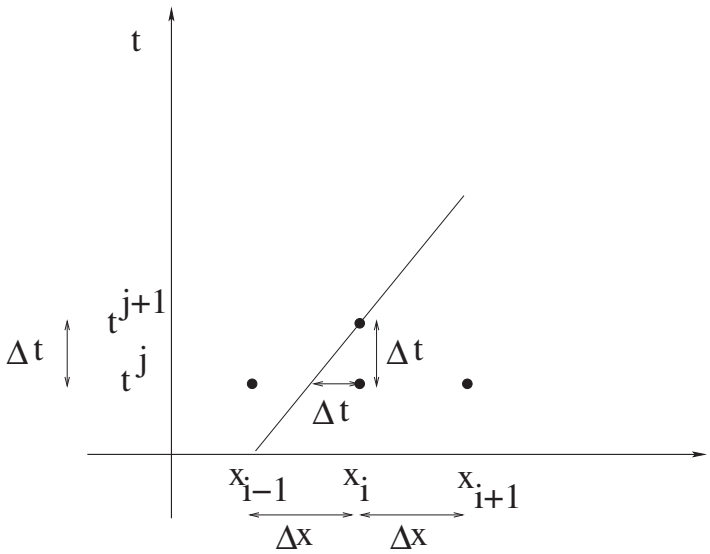

Figure 12.4: Derivation of the Lax-Wendroff scheme.

An alternative and more general method for the analysis of stability is the so-called matrix-method, where the eigenvalues of the discretization matrix, where the time integration is incorporated, are estimated by use of Gershgorin's Theorem. This has been treated in Chapter 10 for the heat equation.

12.3.1.2 The Lax-Wendroff scheme

When solutions are smooth the Lax Wendroff scheme is suitable for hyperbolic conservation Laws. We present the Lax-Wendroff scheme for the first order linear advection equation:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0. \tag{12.3.18}$$

To derive the Lax-Wendroff scheme for one spatial coordinate, characteristics are used. The procedure is similar to the derivation of the first-order upwind method. Consider the (x,t)-plane as in Figure 12.4, the points (x_{i-1}, t^j), (x_i, t^j), (x_{i+1}, t^j) and (x_i, t^{j+1}) are indicated. We consider the characteristic that passes through (x_i, t^{j+1}). Note that the characteristics are straight lines for which $\dfrac{dx}{dt} = 1$. Since the solution is constant over the characteristics, we have $u_i^{j+1} = u^j(x_i - \Delta t)$ (see Figure 12.4). Since $x_i - \Delta t$ generally does not co-coincide with a node, quadratic interpolation between u_{i-1}^j, u_i^j and u_{i+1}^j is used to approximate the value $u^j(x_i - \Delta t)$ (see Figure 12.4). By use of $u_i^{j+1} = u^j(x_i - \Delta t)$, the solution at the new time step is determined. Let the CFL-number be given by σ, i.e. $\sigma = \dfrac{\Delta t}{\Delta x}$, then, this procedure gives

$$u_i^{j+1} = \frac{\sigma}{2}(\sigma + 1)u_{i-1}^j - (\sigma^2 - 1)u_i^j + \frac{\sigma}{2}(\sigma - 1)u_{i+1}^j. \tag{12.3.19}$$

Exercise 12.3.2 *Use quadratic interpolation to approximate $u^j(x_i - \Delta t)$ using the values u_{i-1}^j, u_i^j and u_{i+1}^j. Finally, show that Equation* (12.3.19) *follows.*

This scheme due to Lax-Wendroff is more accurate than the previously treated first-order upwind and second-order central discretization methods. However, due to the higher order interpolation, it is suitable for hyperbolic PDE's with smooth solutions only. For shock solutions, it has been proved (see Leveque [25]) that spurious oscillations are introduced by this method and hence for these cases the Lax-Wendroff scheme is not popular. Then, one relies on alternative methods which are suitable for shock capturing. These schemes are based on flux limiters or slope limiters.

12.3.1.3 Flux limiters

From the preceding section it is clear that the upwind scheme causes numerical diffusion as an undesired side effect. The error is of the order of Δx. However, the solution is physical in the sense that no spurious oscillations are introduced. In this section we will consider some higher order methods. We point out that these higher order methods are useful in those parts of the domain of computation where the solution behaves smoothly. At positions where no smoothness is attained we will fall back on the first order upwind method. We turn back to the original first order hyperbolic transport equation with one spatial coordinate

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0. \tag{12.3.20}$$

For the sake of simplicity only equidistant grids for the computational domain are considered. Application of the Finite Volume method over the volume Ω_i around grid node i and division by Δx gives:

$$\int_{\Omega_i} \frac{\partial u}{\partial t} dx = f(u_{i-1/2}) - f(u_{i+1/2}). \tag{12.3.21}$$

Let w_i approximate the mean value of the solution within Ω_i, then we get

$$\frac{w_i^{j+1} - w_i^j}{\Delta t} = \frac{f_{i-1/2} - f_{i+1/2}}{\Delta x}. \tag{12.3.22}$$

For most cases one takes u linear or constant between two consecutive gridnodes and then this average over Ω_i equals the value of u at the particular gridnode, i.e. $w_i = u_i$. The first order upwind approximation gives

$$f_{i+1/2} = f(w_i), \quad f_{i-1/2} = f(w_{i-1}), \tag{12.3.23}$$

and a second order central scheme gives

$$\begin{aligned} f_{i+1/2} &= f(\frac{w_{i+1} + w_i)}{2}) = f(w_i + 1/2(w_{i+1} - w_i)) \\ f_{i-1/2} &= f(\frac{w_{i-1} + w_i}{2}) = f(w_{i-1} + 1/2(w_i - w_{i-1})) \end{aligned} \tag{12.3.24}$$

For the first order upwind scheme, it is known that the numerical solution exhibits no unphysical oscillations. However, it tends to smear out discontinuities due to numerical diffusion. A higher order scheme, such as Lax-Wendroff's scheme, is more accurate but initial conditions with discontinuities can develop into numerical solutions with spurious oscillations. Hence, near shocks one avoids the use of methods which are prone to unphysical oscillations and one does not insist on the higher order accuracy of the solution near discontinuities. Therefore, one tries

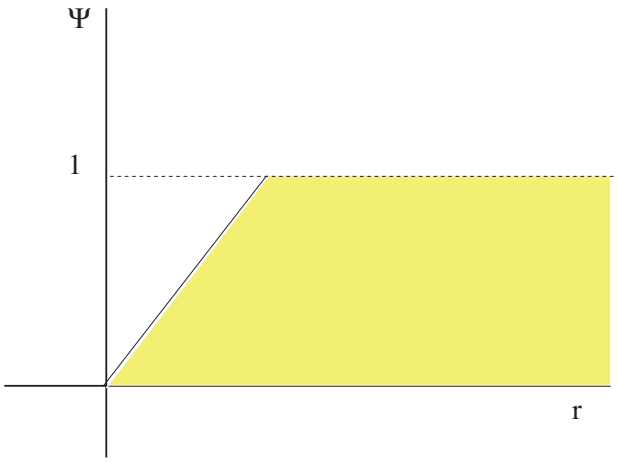

Figure 12.5: The conditions for ψ to avoid unphysical oscillation. This area is indicated by the grey region.

to combine the advantages of both methods: near discontinuities one uses a first order upwind discretization and away from a discontinuity, where the solution is smooth, one uses a higher order method. Van Leer proposes to generalize the relations of $f_{i+1/2}$ to

$$f_{i+1/2} = f(w_i + \psi(r_i)(w_{i+1} - w_i)). \tag{12.3.25}$$

The idea is to let ψ be dependent on the smoothness of the solution and therefore one uses the ratio $r_i := \frac{w_i - w_{i-1}}{w_{i+1} - w_i}$, which is defined by the ratio of the subsequent differences of values of w over neighboring gridnodes. The function ψ is commonly called a *limiter function*. Note that whenever $\psi = \frac{1}{2}$ for both $f_{i-1/2}$ and $f_{i+1/2}$, then, a central scheme is recovered. Whereas, if $\psi = 0$ for both $f_{i-1/2}$ and $f_{i+1/2}$, then, the first order upwind scheme is obtained. Further, if $\psi = 1$ for both $f_{i-1/2}$ and $f_{i+1/2}$, then, a *downwind* discretization is used. The limiter function ψ should be chosen such that first order upwind is obtained near shocks and that the order of discretization is maximal when the solution is smooth. Further, unphysical oscillations are not allowed. Using the concept of conservation of monotonicity and the decrease of total variation (see the appendix of this chapter or Leveque [25], sections 6.7 and 6.12 for the interested reader), one can show that

$$\begin{aligned} 0 \leq \psi(r_i) \leq r_i \\ 0 \leq \psi(r_i) \leq 1, \end{aligned} \tag{12.3.26}$$

provide sufficient conditions to avoid spurious oscillations. This regime is plotted in Figure 12.5 by the colored area.

To make the function ψ more look like a higher order method, a wide variety of functions for ψ has been proposed and investigated. Sweby [38] presents a comparison of the properties of the various choices for ψ. Not aiming at being complete we only mention the limiter due to Van Leer [25]

$$\psi(r) = \frac{r + |r|}{2(1 + |r|)}, \tag{12.3.27}$$

and the 'minmax' limiter due to Kooren [25]

$$\psi(r) = \max\left(0, \min\left(1, \tfrac{1}{3} + \tfrac{1}{6}r, \mu r\right)\right), \quad \text{(Kooren limiter function)}, \quad \mu > 0. \tag{12.3.28}$$

One sees immediately that the second limiter function satisfies all the requirements. It is shown in following exercise that the first limiter satisfies the desired properties as well. Therefore, it is commonly used.

Exercise 12.3.3 *Show that* $\psi(r) = 0$ *for all* $r \leq 0$, $\psi(1) = \frac{1}{2}$ *and that* $\lim_{r\to\infty} \psi(r) = 1$ *and* $\lim_{r\to 0^+} \psi'(r) = 1$. *Further show that* $\psi(r)$ *is monotonic.*

Note that when the profile is almost linear, then, $r \approx 1$. This implies that almost the central discretization is used. Further, when there is a shock between x_{i-1} and x_i, an upwind scheme is obtained. We remark that the situation where both $\psi(r_{i-1}) = 1 = \psi(r_i)$ does not occur for one dimensional geometries. This is nice since this particular case would reflect a *downwind* discretization. Many other limiters are specified by the use of if-statements which is at the expense of computation time. The advantage of these limiters, however, is a slight increase of accuracy.

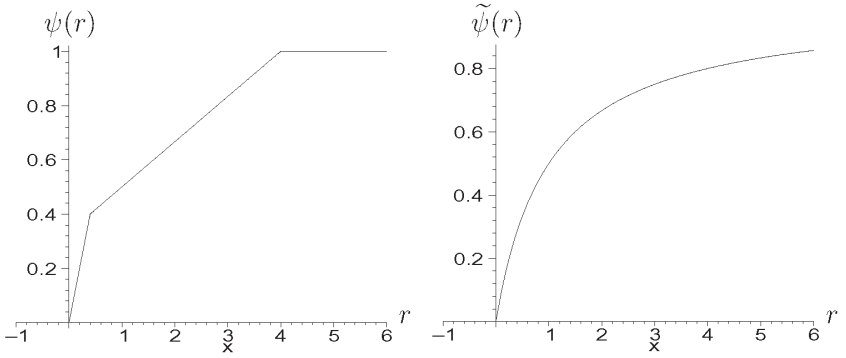

Figure 12.6: Left the Kooren limiter function (12.3.28) and right the Van Leer limiter function.

Further, one can use a predictor-corrector method to improve the accuracy of the solution.

12.4 Mathematical theory for the transport equation

In this section some mathematical background of first order hyperbolic conservation laws is given. This background is commonly used to check the results obtained from numerical simulations. The present treatment is for a scalar conservation law. For systems of hyperbolic PDE's some mathematical theory is presented in Smoller [33].

Since Burgers equation is the simplest case of a non-linear transport equation, first traveling wave solutions for Burgers equation are analyzed. Subsequently, smooth solutions and discontinuous solutions for the Buckley-Leverett with a convex flux function are discussed. Finally, the convexity condition for the Buckley-Leverett equation is relaxed and the construction of analytical solutions is shown.

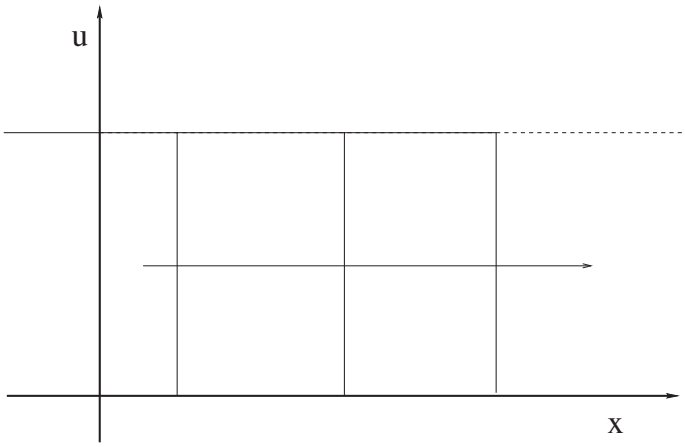

Figure 12.7: Sketch of a shock solution to Burgers equation with $\nu \to 0$ at several times. The shock moves to the right.

12.4.1 Burgers equation

Burgers equation appears in some models from hydrodynamics and aerodynamics. It derivation follows from the conservation of momentum, i.e. a special case of momentum equations due to Euler, in one spatial dimension. This equation is the following:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0. \tag{12.4.1}$$

Exercise 12.4.1 *Given the above Burgers equation, on the unbounded domain* $\mathbb{R}$*, subject to the initial condition*

$$u(x,0) = \begin{cases} 1, & x < 0, \\ 0, & x > 0. \end{cases} \tag{12.4.2}$$

Show by use of the characteristics that the solution of this problem develops into a shock (see Figure 12.7).

One can show that the general hyperbolic conservation Law for smooth $f(u)$ can be transformed into the above Burgers equation, this also motivates that Burgers equation is an important model for which some qualitative properties will be derived. We will see that solutions of Burgers equation are discontinuous (see Figure 12.7 with shocks) or continuous (see Figure 12.8), depending on the initial / boundary conditions. First we consider the equation with the incorporation of an extra diffusive term, which allows us to have smooth solutions of which the derivatives exist:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = \nu\frac{\partial^2 u}{\partial x^2}, \tag{12.4.3}$$

where $\nu > 0$ denotes the viscosity. To get some insight into the structure of the solutions of this equation, we consider the existence of *traveling wave solutions* of the above equation, which are given by $u(x,t) = f(\eta)$ with $\eta = x - st$ (s is to be determined). The solutions that we consider in this section are on the unbounded domain, i.e. $x \in \mathbb{R}$ and $t > 0$. Further, we consider bounded solutions with horizontal asymptotes only, i.e. there exists values u_L and u_R such that

$$\lim_{x\to-\infty} u(x,t) = u_L \text{ and } \lim_{x\to\infty} u(x,t) = u_R. \tag{12.4.4}$$

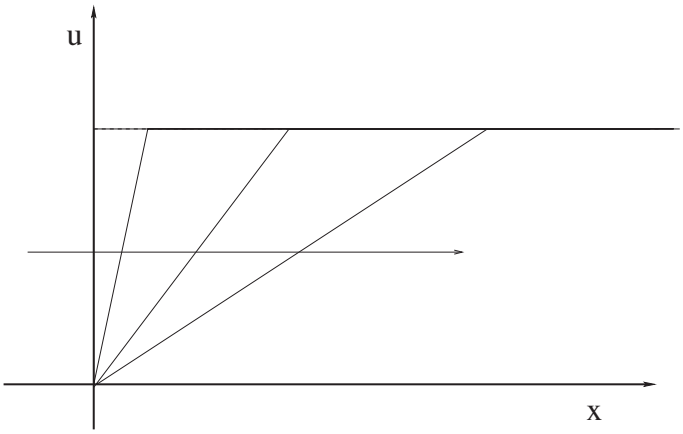

Figure 12.8: Sketch of a continuous solution to Burgers equation with $\nu \to 0$ at several times.

Then Equation (12.4.3) changes into

$$-sf' + \left(\frac{1}{2}f^2\right)' = \nu f''. \tag{12.4.5}$$

After integration of the above equation we obtain

$$-sf + \frac{1}{2}f^2 = \nu f' + A, \tag{12.4.6}$$

where A is a constant of integration. The solution f is assumed to be smooth. Hence, for $|\eta| \to \infty$ we must have $\lim_{|\eta|\to\infty} f'(\eta) = 0$. The numbers s and A are determined from the use of the boundary conditions and $\lim_{|\eta|\to\infty} f'(\eta) = 0$, to obtain

$$-su_L + \frac{1}{2}u_L^2 = A; \tag{12.4.7}$$

$$-su_R + \frac{1}{2}u_R^2 = A. \tag{12.4.8}$$

We solve these equations for s and A to get

$$s = \frac{1}{2}(u_R + u_L) \tag{12.4.9}$$

$$A = \frac{1}{2}u_R u_L. \tag{12.4.10}$$

Note that s defines the velocity of the traveling wave. Substitution of equation (12.4.10) into Equation (12.4.6) gives

$$2\nu f' = (f - u_R)(f - u_L) < 0, \tag{12.4.11}$$

where the inequality holds for f between u_L and u_R. Hence, the solution f should be within the interval between u_L and u_R otherwise the boundary conditions cannot be satisfied unless the solution contains a discontinuity. This implies that if $u = u(x,t)$ satisfies the traveling wave behavior then it is a decreasing function with respect to x. This also implies that $u_L \geq u_R$ only admits traveling wave

solutions , since $u_L < u_R$ requires an increasing function, i.e. $f'(\eta) > 0$, which contradicts Equation (12.4.11). For $u_L \geq u_R$ Equation (12.4.11) is solved by the use of separation of variables to get

$$f(\eta) = u_R + \frac{u_L - u_R}{1 + \exp\left(\frac{u_L - u_R}{2\nu}\eta\right)}, \tag{12.4.12}$$

hence

$$u(x,t) = u_R + \frac{u_L - u_R}{1 + \exp\left(\frac{u_L - u_R}{2\nu}(x - st)\right)}, \tag{12.4.13}$$

with

$$s = \frac{u_L + u_R}{2}. \tag{12.4.14}$$

Note that if $\nu \to 0$ then the solution tends to a shock behavior:

$$\lim_{\nu \to 0} f(\eta) = \begin{cases} u_R, \text{ for } \eta > 0, \\ u_L, \text{ for } \eta < 0. \end{cases} \tag{12.4.15}$$

For the case that $u_R > u_L$ we saw no traveling wave solutions, i.e. $u(x,t) = f(\eta)$ with $\eta = x - st$, does not exist. However, then a solution of a different structure exists. This will be analyzed in the presentation of the Buckley-Leverett equation. The most important conclusions of this section are that the solution of Burgers equation tends to be discontinuous under $u_L > u_R$ as $\nu \to 0$ and that traveling wave solutions exists provided $u_L \geq u_R$.

12.4.2 The Buckley-Leverett equation

The Buckley-Leverett equation plays a crucial role in the flow of two phases in porous media. Its derivation is based on the concept of *relative permeabilities*. For a derivation, we refer to the book of Bear [4]. We consider the Buckley-Leverett equation with (piecewise) smooth solutions, such that the derivatives of the solutions exist. Further, it is assumed that $f(u)$ is a smooth function too. This equation reads as

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \Rightarrow \frac{\partial u}{\partial t} + f'(u)\frac{\partial u}{\partial x} = 0. \tag{12.4.16}$$

We will see that the character of the solution of the above equation depends on the flux function $f(u)$. Consider a point (x_0, t_0) in the x,t-plane and the initial value problem for the characteristics

$$\frac{dx}{ds} = f'(u), \quad \frac{dt}{ds} = 1, \text{ where } \rho(s) = 1, \tag{12.4.17}$$

then we consider characteristics of Equation (12.4.16) in terms of $x(t)$ since

$$\frac{dx}{dt} = f'(u), x(t_0) = x_0.$$

At this curve the following holds after combination with Equation (12.4.17)

$$\frac{d}{dt}u(x(t),t) = \frac{\partial u}{\partial x}(x(t),t)x'(t) + \frac{\partial u}{\partial t}(x(t),t) = \frac{\partial u}{\partial x}f'(u) + \frac{\partial u}{\partial t} = 0. \tag{12.4.18}$$

The first equality follows the Chain Rule for differentiation. Note that for all the differentiations it is necessary that the derivatives exist. We only consider (piecewise) smooth solutions. The method of characteristics is used to study qualitative

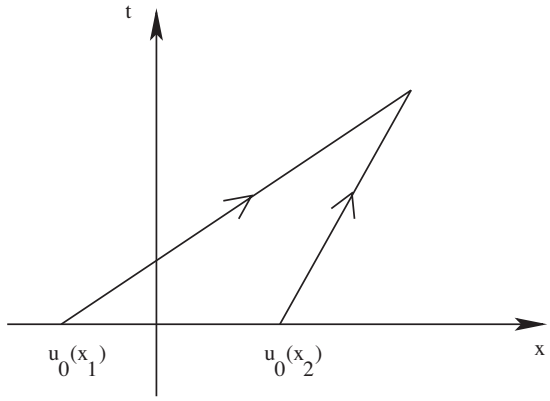

Figure 12.9: Formation of a discontinuity.

aspects of the solution of the Buckley-Leverett equation. As an example we treat the case that $f(u)$, $f'(u)$, $f''(u) > 0$. Suppose that $u(x,0) = u_0(x)$ is given as the initial condition. Now we show by a contradiction argument that no smooth solutions for $u(x,t)$ can exist if $u_0(x)$ is a decreasing function. We recall the characteristic equation (12.4.17) and assume that $u_0(x)$ is a decreasing function. We take two points (x_1, t_0) and (x_2, t_0) with $x_2 > x_1$ (see Figure 12.9. Then, $u(x_2, t_0) < u(x_1, t_0)$ and since $f''(u) > 0$, we obtain

$$\frac{dx_1}{dt} = f'(u(x_1(t), t)) > f'(u(x_2(t), t)) = \frac{dx_2}{dt}. \tag{12.4.19}$$

The last equation is justified since $\frac{d}{dt}u(x(t), t) = 0$ ($u(x,t)$ is constant along its characteristics and hence the characteristics are straight lines). Relation (12.4.19) implies that characteristics intersect and hence the solution becomes multi-valued and the model breaks down. At this point it is possible to assign a large class of solutions to the model problem. Later, in this section, it will turn out that only one solution is physically relevant since it conserves mass. This is a solution with a discontinuity.

As an other example we consider the case $f''(u) > 0$ with the initial condition

$$u(x,0) = u_0(x) = \begin{cases} 0, & \text{for } x < 0, \\ 1, & \text{for } x \geq 0. \end{cases} \tag{12.4.20}$$

Use of the method of characteristics gives with $x_1 \to 0^-$ and $x_2 = 0$ and hence

$$x_1'(t) = f'(u(x_1(t), t)) < f'(u(x_2(t), t)) = x_2'(t). \tag{12.4.21}$$

This implies that the characteristics diverge (see Figure 12.10). For this case we will have a continuous solution. We see that the method of characteristics gives insight into whether or not smooth solutions are possible or whether a discontinuous initial condition stays a shock or develops into a smooth solution. The nature of the continuous and discontinuous solutions will be discussed in the next two sections.

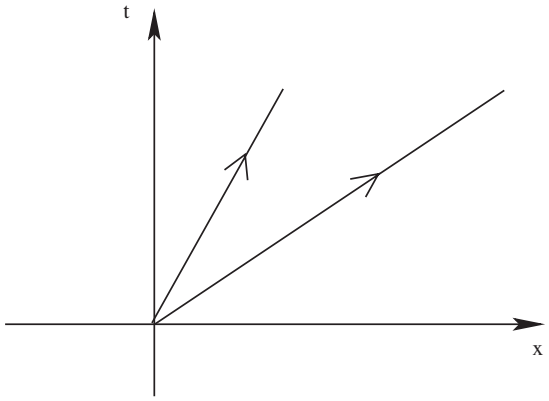

Figure 12.10: Formation of the rarefaction or expansion wave.

12.4.2.1 The smooth solution

For $u_0(x) = \begin{cases} 0, \text{ for } x < 0 \\ 1, \text{ for } x \geq 0 \end{cases}$ and $f''(u) > 0$, then, from the characteristics we saw that the solution loses its discontinuous behavior. Since in a numerical method for a PDE the initial drop occurs over at least one grid-spacing, the numerical initial condition is continuous. The method of characteristics then implies that the solution stays continuous. Further, for $f''(u) < 0$ we will have convergence to a discontinuous solution as time proceeds. This will be the topic in the next section. Now we examine the continuous solution when $f''(u) > 0$. Therefore, we set $u(x,t) = g(\eta)$, $\eta = \dfrac{x}{t}$ with $f'(0) < \eta < f'(1)$, then substitution into the Buckley-Leverett equation (12.4.16) gives with use of the Chain Rule for differentiation

$$-\frac{x}{t^2}g'(\eta) + \frac{1}{t}\frac{d}{d\eta}f(g(\eta)) = 0. \tag{12.4.22}$$

This gives

$$\eta g'(\eta) = f'(g(\eta))g'(\eta). \tag{12.4.23}$$

Hence $g'(\eta) = 0$ (constant state solution) or

$$f'(g(\eta)) = \eta \Rightarrow g(\eta) = (f')^{-1}(\eta). \tag{12.4.24}$$

In the implication we assume that $f'(u)$ is invertible on the domain of consideration, the inverse of $f'(u)$ is denoted by $(f')^{-1}$. For a bounded solution $u \in [0,1]$ we have that

$$u(x,t) = \begin{cases} 0, & \text{for } x < f'(0)t, \\ (f')^{-1}(\frac{x}{t}), & \text{for } f'(0)t < x < f'(1)t, \\ 1, & \text{for } x > f'(1)t. \end{cases} \tag{12.4.25}$$

A solution with the structure of Equation (12.4.25) is called a *rare faction* or an *expansion wave*. Physically, this often amounts a mixing behavior of two phases or *viscous fingering* (as a Saffmann-Taylor instability), see Bear [4].

Example 12.4.1 *For Burgers equation we have* $f(u) = \frac{u^2}{2}$, *hence* $f'(u) = u$. *Therefore*

$(f')^{-1}(\eta) = \eta$ *and the solution becomes with* $f'(0) = 0$ *and* $f'(1) = 1$*:*

$$u(x,t) = \begin{cases} 0, & \text{for } x < 0, \\ \dfrac{x}{t}, & \text{for } 0 \le x \le t, \\ 1, & \text{for } x > t. \end{cases} \tag{12.4.26}$$

12.4.2.2 The discontinuous solution

We consider the case $f'' > 0$ ($f(u)$ is convex), and $u(x,0) = u_0(x) = \begin{cases} 1 \text{ for } x < 0 \\ 0 \text{ for } x \ge 0 \end{cases}$, for the Buckley-Leverett equation. We saw for Burgers equation that the shock speed is given by

$$s = \frac{u_L + u_R}{2}. \tag{12.4.27}$$

We are going to calculate the shock speed by the use of a mass conservation argument for the Buckley-Leverett equation in general. An alternative formal derivation can be given by a consideration of weak solutions and compact support. This is beyond the scope of this book and we refer the interested reader to the book of Smoller [33]. Consider integration over $x \in [a,b]$ of the Buckley-Leverett equation, where a and b are chosen such that the interval contains the shock position:

$$\int_a^b \frac{\partial u}{\partial t} = -\int_a^b \frac{\partial f(u)}{\partial x} dx = f(u(a,t)) - f(u(b,t)). \tag{12.4.28}$$

Since the bounds in the above integral do not depend on t, we may interchange the order of differentiation with respect to time and integration over the fixed interval $[a,b]$. Further, the quantity $\int_a^b u dx$ depends on t only, hence the partial differentiation with respect to t can be written as an ordinary derivative with respect to t, to give

$$\frac{d}{dt}\int_a^b u dx = f(u(a,t)) - f(u(b,t)). \tag{12.4.29}$$

Now suppose that the solution u is discontinuous at a curve $s(t)$ where $a < s(t) < b$, then by the use of Leibniz' Rule and the Chain Rule for differentiation follows

$$\begin{aligned} \frac{d}{dt}\int_a^b u dx &= \frac{d}{dt}\left[\int_a^{s(t)} u dx + \int_{s(t)}^b u dx\right] \\ &= \int_a^{s(t)} \frac{\partial u}{\partial t} dx + u(s^-(t),t)s'(t) + \int_{s(t)}^b \frac{\partial u}{\partial t} dx - u(s^+(t),t)s'(t). \end{aligned} \tag{12.4.30}$$

Here we define $s^-(t)$ and $s^+(t)$ as the positions adjacent to the left and the right side of the shock position respectively. Use of the Buckley-Leverett equation (12.4.16) gives

$$\begin{aligned} \frac{d}{dt}\int_a^b u dx = f(u(a,t)) - f(u(s^-(t),t)) + u(s^-(t),t)s'(t) + \\ f(u(s^+(t),t)) - f(u(b,t)) - u(s^+(t),t)s'(t). \end{aligned} \tag{12.4.31}$$

Since Equation (12.4.29) holds, it follows from Equation (12.4.31) that

$$\big(u(s^-(t),t) - u(s^+(t),t)\big)\, s'(t) = f(u(s^-(t),t)) - f(u(s^+(t),t)). \tag{12.4.32}$$

Let $s'(t)$ be the shockspeed, then if $u(s^-(t),t) - u(s^+(t),t) \neq 0$, then

$$s'(t) = \frac{f(u(s^-(t),t)) - f(u(s^+(t),t))}{u(s^-(t),t) - u(s^+(t),t)} =: \frac{[f(u)]}{[u]}. \tag{12.4.33}$$

This equation is known as the Rankine-Hugoniot condition. Note that if $u(s^+(t),t)$ tends to $u(s^-(t),t)$ (i.e. the continuous case) then the speed of a characteristic is recovered (see exercise 19.1).

Exercise 12.4.2 *Show that, for $u(s^+(t),t)$ tending to $u(s^-(t),t)$, the speed of a characteristic is recovered.*

The case of $f''(u) > 0$ has been examined now, the case of $f''(u) < 0$ can be addressed likewise. This is left as an exercise.

Exercise 12.4.3 *Given the Buckley-Leverett equation with $f''(u) < 0$ with $u(x,0) = \begin{cases} u_L, x < 0 \\ u_R, x \geq 0 \end{cases}$, $u_L \neq u_R$. Under which conditions will be the shock be stable and under which conditions will a rarefaction develop? Motivate this by the use of characteristics.*

Exercise 12.4.4 *Given Burgers equation*

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = 0, \text{ with } u = \begin{cases} 1, x < 0 \\ 0, x \geq 0 \end{cases}. \tag{12.4.34}$$

Describe the solution for $x \in \mathbb{R}$ and $t > 0$.

12.4.2.3 The non-convex case

We abandon the convexity condition for $f(u)$ and consider

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \text{ with } f(u) = \frac{u^2}{u^2 + (1-u)^2}. \tag{12.4.35}$$

This choice of $f(u)$ arises in many applications of two-phase flow in porous media where influences of gravity are neglected. The following initial condition is used

$$u(x,0) = u_0(x) = \begin{cases} 1, & x < 0, \\ 1 - \frac{x}{\epsilon}, & 0 \leq x \leq \epsilon, \\ 0, & x > \epsilon \end{cases} \quad \text{for some } \epsilon > 0. \tag{12.4.36}$$

To gain insight into the qualitative behavior of the solution, characteristics are used:

$$\frac{dt}{ds} = 1, \quad \frac{dx}{ds} = f'(u) \text{ hence } \frac{dx}{dt} = f'(u). \tag{12.4.37}$$

Since $f'(u) = 0$ for $u = 0$ and $u = 1$, it is clear that characteristics, originating from the negative part of the x-axis ($x < 0$) and from $x > \epsilon$, move vertically upward. Further, since $f''(u) = 0$ at $u = \frac{1}{2}$ (corresponding with $x = \frac{\epsilon}{2}$ at $t = 0$) the slope of the characteristics tends to be less vertical as x increases within the interval $0 < x < \frac{\epsilon}{2}$ (at $t = 0$). In the interval $\frac{\epsilon}{2} < x < \epsilon$ at $t = 0$, the characteristics tend to be more vertical again. This is illustrated in Figure 12.11. For the characteristics originating from the x-axis, the following interesting features are observed:

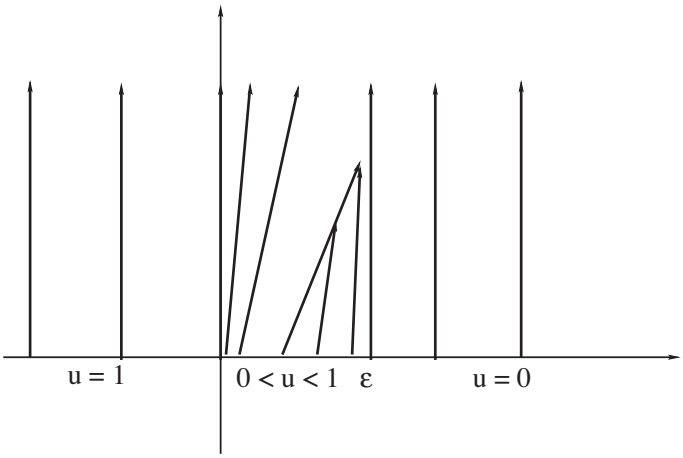

Figure 12.11: A sketch of the characteristics that originate from the x-axis.

- Characteristics originating from the interval $0 < x < \frac{\epsilon}{2}$ diverge and hence a rare faction develops.
- Characteristics originating from the interval $\frac{\epsilon}{2} < x < \epsilon$ converge and intersect and hence a shock develops.

In the remainder of this section some mathematical background for the construction of analytical solutions is presented. For this purpose a traveling wave argument is given for the Buckley-Leverett equation with an added 'viscosity term':

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad \nu > 0. \tag{12.4.38}$$

We only consider solutions for which

$$u(-\infty, t) = u_L, \qquad u(\infty, t) = u_R. \tag{12.4.39}$$

For the traveling wave structure, we set

$$u(x,t) = v(\eta), \text{ with } \eta = \frac{x - ct}{\nu}. \tag{12.4.40}$$

This equation transforms with the 'boundary conditions' into

$$\begin{array}{l} -cv' + (f(v))' = v'' \\ v(-\infty) = u_L \text{ and } v(\infty) = u_R. \end{array} \tag{12.4.41}$$

The above differential equation is integrated to obtain

$$-cv + f(v) = v' + A. \tag{12.4.42}$$

In the above equation c and A are determined from the 'boundary conditions' at $\pm\infty$. Since v has horizontal asymptotes at $|\eta| \to \infty$, it is necessary that $v'(\eta) \to 0$ as $|\eta| \to \infty$. Hence the 'conditions' at $|\eta| \to \infty$ imply

$$\left.\begin{array}{l} -cu_L + f(u_L) = A \\ -cu_R + f(u_R) = A \end{array}\right\} \Rightarrow c = \frac{f(u_R) - f(u_L)}{u_R - u_L}, \quad A = f(u_L) - cu_L. \tag{12.4.43}$$

Hence Equation (12.4.42) changes into

$$v' = f(v) - f(u_L) + c(u_L - v). \tag{12.4.44}$$

Suppose that there exists a $\hat{v}$ between u_L and u_R for which

$$f(\hat{v}) - f(u_L) + c(u_L - \hat{v}) = 0 \Rightarrow v' = 0 \text{ at } v = \hat{v}. \tag{12.4.45}$$

Then we are at an equilibrium point of Equation (12.4.44) and hence $v = \hat{v}$ for $\eta \in \mathbb{R}$ and herewith a contradiction with the boundary conditions is obtained. This implies that v is a strictly monotonic function of η and hence a traveling wave is strictly monotonic. Therewith

$$\begin{array}{l} u_L < u_R \Rightarrow v' > 0 \Rightarrow v > u_L \\ u_L > u_R \Rightarrow v' < 0 \Rightarrow v < u_L. \end{array} \tag{12.4.46}$$

Division of Equation (12.4.44) by $u_L - v$ and use of the above observations gives after some rearrangement

$$c = \frac{f(u_L) - f(v)}{u_L - v} + \frac{v'}{u_L - v} < \frac{f(u_L) - f(v)}{u_L - v}, \tag{12.4.47}$$

since $\frac{v'}{u_L - v} < 0$. The above equation is formulated as

$$c = \frac{f(u_R) - f(u_L)}{u_R - u_L} < \frac{f(u_R) - f(v)}{u_R - v}, \tag{12.4.48}$$

for all v between u_L and u_R. The above inequality is a sufficient and necessary condition for the existence of a traveling wave solution. It is clear that relation (12.4.48) is a consequence of the above arguments and hence (12.4.48) poses a necessary condition for the existence of a traveling wave solution. Next we show that condition (12.4.48) is sufficient for the existence of a traveling wave solution, i.e. (12.4.48) guarantees the existence of a traveling wave, therefore we integrate Equation (12.4.44) to obtain

$$\int_{\frac{u_L + u_R}{2}}^{v(\eta)} \frac{ds}{f(s) - f(u_L) + c(u_L - s)} = \eta. \tag{12.4.49}$$

The traveling wave solution exists if the above integral exists. Condition (12.4.48) implies that the denominator of the above integrand is non-zero for v between u_L and u_R. Hence, the integrand is bounded and therefore the above integral exists. This guarantees the existence of a traveling wave. Now, suppose that $f(u)$ is convex-concave and that $f(0) = 0$ and $f(1) = 1$, we investigate the possibility for traveling waves. See Figure 12.12 for a sketch of $f(u)$. We distinguish the following cases:

- $u_L > u_R = 0$, then $v' < 0$ and hence from Equation (12.4.44) follows

  $$f(v) < f(u_L) - c(u_L - v) = f(u_L) + \frac{f(u_L)}{u_L}(v - u_L), \text{ note that } u_R = 0 = f(u_R). \tag{12.4.50}$$

  The graph of the right-hand side of the above inequality is indicated by the dotted line in Figure 12.12. The position u_1 is the intersection of the dotted line and $f(v)$ and hence the above inequality no longer holds if $v \geq u_1$ and thus traveling waves exists for values of $0 = u_R \leq u_L < u_1$.

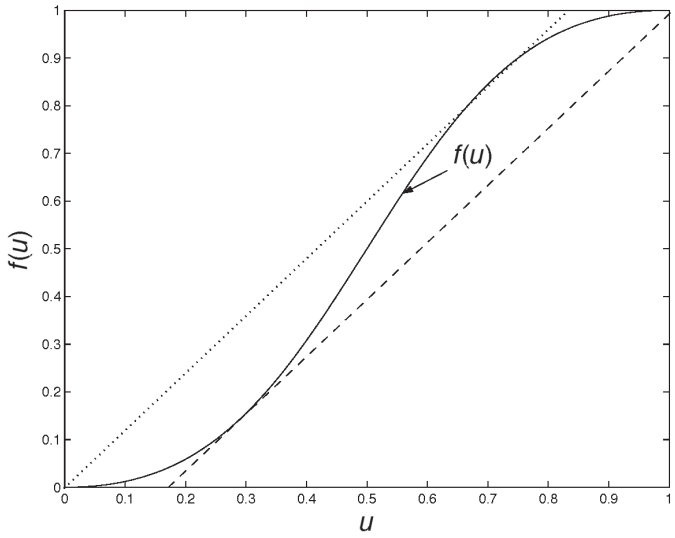

Figure 12.12: A sketch of the function f.

- $u_L < u_R = 1$, then $v' > 0$ and hence from Equation (12.4.44) follows

$$f(v) > f(u_L) + \frac{f(u_L) - 1}{u_L - 1}(v - u_L), \text{ note that } u_R = 1 = f(u_R). \quad (12.4.51)$$

  The graph of the right-hand side of the above inequality is indicated by the dashed line in Figure 12.12. The position u_2 is the intersection of the dashed line and the function $f(v)$ and hence the above inequality no longer holds if $v \leq u_2$ and thus traveling waves exist for values of $u_2 < u_L \leq u_2 = 1$.

Since condition (12.4.48) holds for any $\nu > 0$ (hence also for $\nu \to 0$), it is used as an additional 'entropy condition' for traveling waves. As $\nu \to 0$ this traveling wave becomes a shock, due to the intersection of the characteristics, with velocity

$$c = \frac{f(u) - f(u_L)}{u - u_L} \geq \frac{f(u_R) - f(u_L)}{u_R - u_L} \text{ for } u \text{ between } u_L \text{ and } u_R. \quad (12.4.52)$$

Solutions possibly consist of a combination of a rare faction and a shock.

12.4.2.4 Construction of solutions

Let $f(u)$ have a continuous second order derivative on the interval $[0, 1]$, and let f satisfy the following requirements:

$$\begin{cases} f(s) > 0, f'(s) > 0 \text{ for } s \in (0,1) \\ f(0) = 0, f(1) = 1 \\ f''(s) > 0 \text{ for } s \in [0, \hat{s}) \\ f''(s) < 0 \text{ for } s \in (\hat{s}, 1] \end{cases} \quad (12.4.53)$$

for some $\hat{s} \in (0,1)$. Further, $u(x,t)$ satisfies

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial f(u)}{\partial x} = 0 \\ u(x,0) = u_0(x) = \begin{cases} 1, & x < 0 \\ 0, & x > 0 \end{cases} \end{cases} \quad (12.4.54)$$

We will construct a solution which consists of a rarefaction and a shock. We see that $u_R = 0$. For the traveling wave part one obtains that a traveling wave is allowed for $0 = u_R \leq u_L < u_1 < 1$, where u_1 is the value at which

$$f'(u_1) = \frac{f(u_1) - f(u_R)}{u_1 - u_R} \text{ and } f(u_1) = \frac{f(u_1) - f(u_R)}{u_1 - u_R} u_1, (\text{ note that } u_R = 0). \tag{12.4.55}$$

This follows from the arguments of the preceding section. The right-hand sides of the two equations are consecutively the derivative of the straight line and the line itself at $u = u_1$. Both expressions imply that u_1 satisfies

$$u_1 - \frac{f(u_1)}{u_1} = 0. \tag{12.4.56}$$

This equation can be solved for u_1 by the use of a zero-point method. There is a shock over $u_R = 0$ and $u_L = u_1$ which travels at the constant speed $c = \frac{f(u_1)}{u_1}$. For the part $u \in [u_1, 1]$ no traveling wave exists, there a rarefaction is obtained, i.e. $u = g(\eta)$, $\eta = \frac{x}{t}$, for $u_L = 1$ and $u_R = u_1$ as the respective left- and right state. Substitution of this rarefaction behavior, gives

$$g'(\eta) = 0 \text{ (constant state) or } \eta = f'(g(\eta)). \tag{12.4.57}$$

This implies that

$$g(\eta) = \begin{cases} 1, \text{ for } 0 < \eta < f'(1) \\ (f')^{-1}(\eta) \text{ for } f'(1) < \eta < f'(u_1) \\ 0 \text{ for } \eta > f'(u_1) \end{cases} . \tag{12.4.58}$$

Hence the solution is constructed by

$$u(x,t) = \begin{cases} 1, \text{ for } 0 < x < f'(1)t \\ (f')^{-1}(\frac{x}{t}) \text{ for } f'(1)t < x < f'(u_1)t \\ 0 \text{ for } x > f'(u_1)t \end{cases} . \tag{12.4.59}$$

Exercise 12.4.5 *Construct the analytical solution of the same problem except for the initial condition*

$$u(x,t) = u_0(x) = \begin{cases} 0, \textit{ for } x < 0, \\ 1, \textit{ for } x > 0, \end{cases} \qquad (u_R = 1), \tag{12.4.60}$$

by use of the arguments of the preceding section.

12.5 Summary of Chapter 12

This chapter treats an introduction into the transport equation, which is also commonly referred to as a first order hyperbolic conservation law. To gain insight into the behavior of the solutions and the nature of the boundary and initial conditions, characteristics are treated. Formerly, many numerical techniques were based on a direct solution of the characteristic relation. Further, the most classical numerical methods for the solution of the transport equation are described. Finally, some mathematical aspects of the structure of the solution are presented. First, the flux-function $f(u)$ is assumed to be convex. Subsequently, the convexity condition for $f(u)$ is dropped. This case is crucial when considering two-phase flow in porous media without gravity.

12.6 Appendix: requirements on flux-limiters

In this appendix we comment on the requirements on the limiter function. The requirements

$$\begin{aligned} 0 \leq \psi(r_i) &\leq r_i \\ 0 \leq \psi(r_i) &\leq 1, \end{aligned} \tag{12.6.1}$$

are demonstrated in this section for the linear hyperbolic partial differential equation only, i.e.

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0. \tag{12.6.2}$$

Before we continue the motivation, the concept of the Total Variation (TV) is introduced:

$$TV(w) := \sum_{i=-\infty}^{i=\infty} |w_i - w_{i-1}|. \tag{12.6.3}$$

The total variation is high for oscillatory functions and zero for functions with a constant value. The only attractive numerical methods contain a decreasing total variation. Because these methods will give a damping of unphysical oscillations. Let j denote the time-step, then, it is required that:

$$TV(w^{j+1}) \leq TV(w^j), \tag{12.6.4}$$

then the method is called Total Variation Diminishing (or briefly TVD).

We shall show now that Euler forward time integration is TVD, if conditions (12.6.1) are satisfied. Euler forward time integration gives

$$w_i^{j+1} = w_i^j - \Delta t \frac{f_{i+1/2}^j - f_{i-1/2}^j}{\Delta x}, \tag{12.6.5}$$

with

$$f_{i+1/2}^j = w_i^j + \psi(r_i)(w_{i+1}^j - w_i^j). \tag{12.6.6}$$

Here r_i is a measure of the smoothness of the data, defined by

$$r_i := \frac{w_i^j - w_{i-1}^j}{w_{i+1}^j - w_i^j}. \tag{12.6.7}$$

We write this as

$$w_i^{j+1} = w_i^j - \frac{\Delta t}{\Delta x}(w_{i+1}^j - w_i^j)\left(1 + \frac{\psi(r_i)}{r_i} - \psi(r_i)\right), \tag{12.6.8}$$

where the Definition (12.6.7) for r_i was used. Harten shows that a sufficient condition for TVD in the sense of relation (12.6.4) is given by

$$0 \leq 1 + \frac{\psi(r_i)}{r_i} - \psi(r_{i-1}) \leq 1. \tag{12.6.9}$$

This is motivated as follows. From Equation (12.6.8) follows for w_i^{j+1}:

$$w_{i-1}^{j+1} = w_{i-1}^j - \frac{\Delta t}{\Delta x}(w_i^j - w_{i-1}^j)\left(1 + \frac{\psi(r_{i-1})}{r_{i-1}} - \psi(r_{i-1})\right), \tag{12.6.10}$$

Subtraction of Equation (12.6.8) from (12.6.10), gives

$$w_i^{j+1} - w_{i-1}^{j+1} = (w_i^j - w_{i-1}^j)\left(1 - \frac{\Delta t}{\Delta x}\left(1 + \frac{\psi(r_i)}{r_i} - \psi(r_{i-1})\right)\right) + \\ + \frac{\Delta t}{\Delta x}\left(1 + \frac{\psi(r_{i-1})}{r_{i-1}} - \psi(r_{i-2})\right)(w_{i-1}^j - w_{i-2}^j). \tag{12.6.11}$$

Subsequently we take the absolute value and sum over all indices i and use the triangle identity to obtain:

$$\sum_{i=-\infty}^{i=\infty} |w_i^{j+1} - w_{i-1}^{j+1}| \leq \sum_{i=-\infty}^{i=\infty} |w_i^j - w_{i-1}^j| \left(1 - \frac{\Delta t}{\Delta x}\left(1 + \frac{\psi(r_i)}{r_i} - \psi(r_{i-1})\right)\right) \\ + \sum_{i=-\infty}^{i=\infty} \frac{\Delta t}{\Delta x}\left(1 + \frac{\psi(r_{i-1})}{r_{i-1}} - \psi(r_{i-2})\right)|w_{i-1}^j - w_{i-2}^j| \tag{12.6.12}$$

Next we require condition (12.6.1) to hold and shift the index of the second summation in the right-hand side so that most terms cancel. Hence we are left with

$$\sum_{i=-\infty}^{i=\infty} |w_i^{j+1} - w_{i-1}^{j+1}| \leq \sum_{i=-\infty}^{i=\infty} |w_i^j - w_{i-1}^j|. \tag{12.6.13}$$

Hence the discretization is TVD if condition (12.6.1) is satisfied.

Chapter 13

Moving boundary problems

Objectives

In previous chapters several numerical methods have been presented and applied to model problems: (Navier) Stokes and Euler equations, Transport in porous media, Diffusion problems and Wave equations. In industrial applications a large amount of completely different problems arise. An important class of problems is that of free and moving boundaries. In free-surface problems the boundary (or interface) is not known a-priori but it is a part of the solution. In case of moving boundaries the boundary changes in time. Here an example of a moving boundary problem, the so-called *Stefan problem*, is given. This describes, for example, melting of ice or solidification of liquid metals.

Free/moving boundary problems arise in phase transitions (such as solidification), flow problems, crystal growth, steam injection in oil and gas reservoirs and in bubbly flow. Free boundary problems also occur in finance, where they are solved to determine the price of a call option (the right to purchase shares). In this chapter a moving boundary problem with heat diffusion (Fourier), which is referred to as the classical *Stefan problem*, is presented. The model is applied to freezing of water. Several well-known numerical solution procedures to solve Stefan problems are presented. The advantages and disadvantages of particular methods will be described. For a qualitative picture of the solution a reference is given to exact solutions that hold whenever the domain is of infinite size.

13.1 The formulation of a classical Stefan problem: ice and water

The scientist J. Stefan studied the melting and freezing of the ice-caps near the North pole of the earth [34]. Using his experiments, he formulated a model to describe the area of the ice-caps as a function of time and the classical Stefan problem was born. Weber [47] was, as far as known the first to study the Stefan problem mathematically and he found a so-called 'self-similar' solution. For more historical and mathematical background on the classical Stefan problem and its relating mathematics, we refer to the work of Vuik [46, 45]. In this chapter we consider freezing of water. For more physical background we refer to the textbook of Carslaw and Jaeger [9] . For the sake of illustration, we consider an open rectangular domain $\Omega := \{(x,y) \in \mathbb{R}^2 : 0 < x < 1, 0 < y < 1\}$ that is initially filled with water at temperature T_0. The boundary of Ω is given by $\partial\Omega$, which is divided into $\partial\Omega_1$, $\partial\Omega_2$, $\partial\Omega_3$ and $\partial\Omega_4$. The areas, occupied by ice and water, are respectively

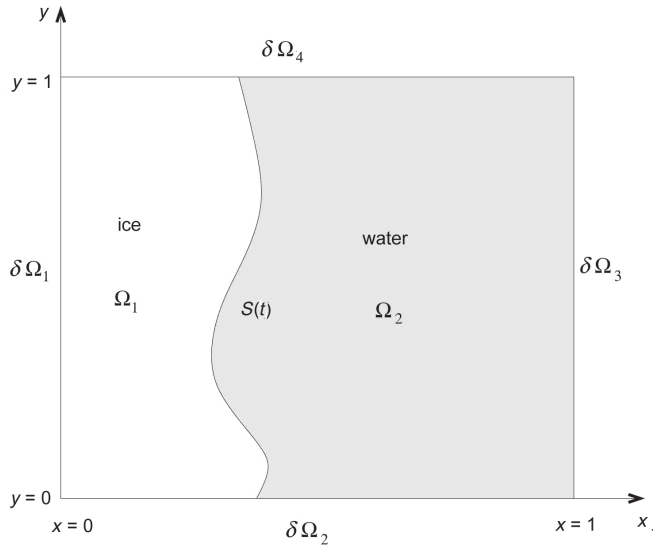

Figure 13.1: Geometry of the domain.

given by the non-overlapping subdomains $\Omega_1(t)$ and $\Omega_2(t)$, where the boundary between water and ice is given by the line $S = S(t)$. This boundary $S(t)$ is also referred to as the (moving) interface. The geometry is shown in Figure 13.1. The temperature in Ω_1 and Ω_2 is denoted by respectively T_1 and T_2, the temperature in Ω is denoted by $T(x, y, t)$.

At the interface, $(x, y) \in S(t)$, between water and ice the temperature is $T = T_s$ (freezing temperature) at all times. At the initial stage we have a temperature T_0 and we assume that the whole domain is filled with water: $\Omega_2(0) = \Omega$. At $x = 0$, the temperature is prescribed: $T = T_*$ for $(x, y) \in \partial\Omega_1$, $T_* < T_s < T_0$. Since T_* is below the freezing temperature, the water freezes as time proceeds. The moving interface starts at $x = 0$ and moves to the right. We assume that heat transport in ice and water only takes place by conduction, i.e. heat diffusion. A further assumption is that there is no heat flux across the other boundaries $\partial\Omega_2$ to $\partial\Omega_4$. The normal velocity of the interface is denoted by v_n. Summarizing we have the following mathematical problem where in both ice and water a heat equation is satisfied. At the interface an amount of heat is produced by the freezing of the water (*latent heat*). The differential equation describing the process in the interior of the domain is given by:

$$\rho c \frac{\partial T}{\partial t} = \lambda \Delta T, \qquad \mathbf{x} \in \Omega, \tag{13.1.1}$$

with initial condition

$$T = T_0, \qquad \mathbf{x} \in \Omega. \tag{13.1.2}$$

The boundary conditions are

$$T = T_*, \qquad \mathbf{x} \in \partial\Omega_1, \tag{13.1.3}$$

$$\frac{\partial T}{\partial n} = 0, \qquad \mathbf{x} \in \bigcup_{k=2}^{4} \partial\Omega_k \tag{13.1.4}$$

On the interface we need the following two conditions

$$T = T_s, \qquad \mathbf{x} \in S(t), \tag{13.1.5}$$

and (latent heat)

$$\rho_1 L v_n = \lambda_1 \frac{\partial T_1}{\partial n} - \lambda_2 \frac{\partial T_2}{\partial n}, \qquad \mathbf{x} \in S(t). \tag{13.1.6}$$

We need two interface conditions to calculate the position of the interface. This problem is the classical Stefan problem in a rectangular domain. Equation (13.1.6) gives the rate of the interface. The unknowns are the temperature T and the position of the interface S between both phases. The densities of water and ice are respectively given by ρ_1 and ρ_2. The parameter L represents the latent heat of solidification. The parameters c_1, c_2 and λ_1, λ_2 denote the heat capacities and heat conductivities of respectively ice and water. Existence and uniqueness of a solution pair T and S has been established in Cannon [8] and Vuik [45]. Before treating some numerical solution techniques, the exact solution for a simple one-dimensional case will be given. This solution, also called a self-similar solution, shows the qualitative behavior of this kind of problem.

13.2 An exact (self-similar) solution for an unbounded region

With a *self-similar* solution we mean a solution for the temperature that depends on a pair of x and t, e.g. $T = T(\frac{x}{\sqrt{t}})$. To get some quick insight into the behavior of the solution of the Stefan problem, we present a self-similar solution of the Stefan problem for an unbounded interval: $x \in (0, \infty)$. The solution is for a one-dimensional case and mimics the actual behavior of the solution of the Stefan problem (13.1.1)-(13.1.6) especially in the early stages when the temperature at $\partial\Omega_3$ has not been effected by the freezing front yet. Hence it serves as a test-problem, which can be used to check the behavior of the results from numerical solutions. We require that the solution for the temperature, $T = T(x,t)$ is bounded, continuous and monotone in x and t and hence its derivative with respect to x vanishes as $x \to \infty$. For this purpose we search bounded solutions of the following problem, with the same symbols as in the previous section:

$$\rho c \frac{\partial T}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2} \tag{13.2.1}$$

$$T(x,0) = T_0 \tag{13.2.2}$$

$$T(0,t) = T_* \tag{13.2.3}$$

$$S(0) = 0 \tag{13.2.4}$$

$$\rho_1 L \frac{dS}{dt} = \lambda_1 \frac{\partial T_1}{\partial x} - \lambda_2 \frac{\partial T_2}{\partial x}, \qquad x = S(t) \tag{13.2.5}$$

$$T(S(t),t) = T_s \tag{13.2.6}$$

The above problem, equations (13.2.1)-(13.2.6) admits self-similar solutions in the form $T = T(\frac{x}{\sqrt{t}})$ and $S = k\sqrt{t}$. Explicit formulas for T, S, k can be determined using procedures given in the text-books [9, 13] and the very early paper of Neumann [47].

In Figure 13.2 the temperature profile during freezing of water at consecutive times is shown. The initial water temperature is $T_0 = 2^o = 275K$, freezing temperature, $T_s = 0^o = 273K$. The temperature at $x = 0$ is maintained at $T_* = -17^o = 250K$ at all stages. Further physical data are given in Table 13.1. Figure 13.3 displays the interface position (ice thickness) as a function of time for various temperatures at $x = 0$ (i.e. T_*).

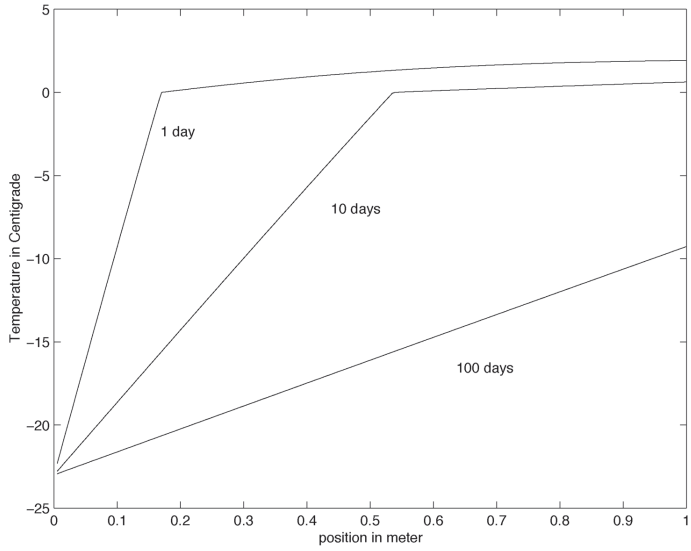

Figure 13.2: The temperature profile of freezing water at consecutive times for T_* = −23 °C. The rate factor k is obtained from the numerical solution, see [9].

Table 13.1. Input data.

Physical quantity	Value	Si-Unit
T_*	250	K
T_s	273	K
T_0	275	K
λ_1	2.2	W/(mK)
λ_2	0.55	W/(mK)
L	33400	J/kg
ρ_1	920	kg/m^3
ρ_2	1000	kg/m^3

13.3 Numerical methods

Various numerical techniques are known to solve Stefan problems. For an overview we refer for instance to the book of Crank [13], where roughly the following methods are distinguished: Front tracking and Fixed domain methods. The main feature of the Fixed domain methods is that the front is defined implicitly and the discretization mesh does not move. Whereas the front is followed explicitly and the mesh moves with the interface in the Front tracking methods. Besides fixed domain and front tracking methods there exists also a hybrid form where a fixed basis grid is used, which in each time step is locally adapted to the front. After the time step the local adaptation is removed.

13.3.1 Moving grid methods

A Front tracking method explicitly tracks the position of the interface. The equations for the temperature are solved using a discretization method in both subdomains and the discrete temperature gradients are substituted into the rate equa-

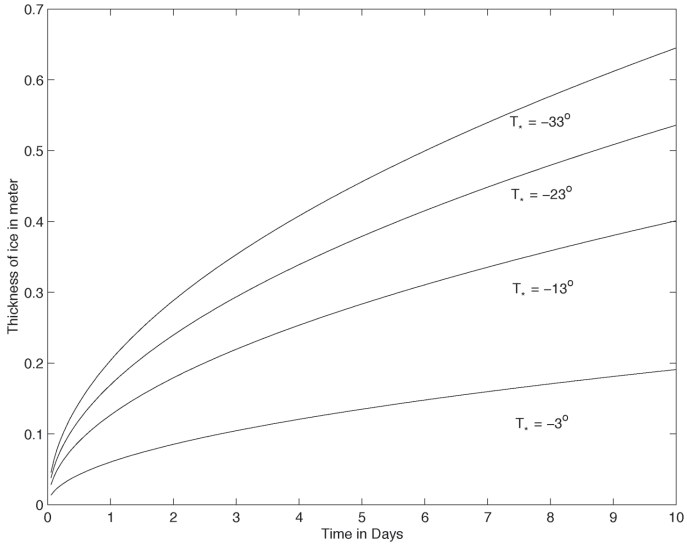

Figure 13.3: The thickness of ice as a function of time for various values of T_*.

tion (see Equation (13.1.6) or (13.2.5)). First we treat a one-dimensional case to illustrate some of the numerical problems that arise in Front tracking methods. Subsequently, we briefly treat a two-dimensional example.

A one-dimensional example

We consider the same situation as in Equations (13.2.1)-(13.2.6) but now the interval in x is bounded, say $x \in [0, 1]$. We again apply a no-flux boundary condition at $x = 1$, i.e.:

$$\lambda_2 \frac{\partial T}{\partial x} = 0, \qquad x = 1. \tag{13.3.1}$$

For now, we assume that the interface position $S(t)$ is given within the interval $0 < S(t) < 1$, and hence Ω_1 and Ω_2 exist and are nonempty sets. Furthermore, we assume that the temperature profile is given at $t = 0$. We use the method of lines to solve the problem. First we deal with the spatial discretization, where we divide Ω_1 and Ω_2 into N grid nodes, i.e.

$$\Delta x_1 = \frac{S(t)}{N}, \qquad \Delta x_2 = \frac{1 - S(t)}{N}. \tag{13.3.2}$$

Note that the grid spacings Δx_1 and Δx_2 depend on time. The index i refers to the grid node:

$$T_{1,i} := T_1(i\,\Delta x_1), \tag{13.3.3}$$

$$T_{2,i} := T_2(S(t) + i\,\Delta x_2). \tag{13.3.4}$$

Further, we use the boundary conditions for $t > 0$

$$T_{1,0} = T_*, \qquad T_{1,N} = T_s, \qquad \text{for } \Omega_1, \tag{13.3.5}$$

$$T_{2,0} = T_s, \quad \lambda_2 \frac{T_{2,N+1} - T_{2,N-1}}{2\,\Delta x_2} = 0, \quad \text{for } \Omega_2. \tag{13.3.6}$$

Note that we add an extra grid point $N + 1$ for Ω_2 to maintain an accuracy of $O(\Delta x_2^2)$ for the global discretization over Ω_2.

Exercise 13.3.1 *Write down the equations from the discretization of the diffusion equation in both subdomains (ice and water) where we use an implicit Euler time integration.*

Since the interface and mesh movement are not incorporated yet into the discretization, we write tildes above the unknowns.

Exercise 13.3.2 *To guarantee a second order spatial accuracy globally, we introduce ghostpoints near the moving interface $S(t)$. Derive expressions for the ghostpoints $\tilde{T}^{j+1}_{1,N+1}$ and $T^{j+1}_{2,-1}$. Hint: treat the discretization of the interface like an internal point in the domain.*

The interface S moves, and the speed is approximated by

$$\rho_1 L \frac{dS}{dt} \approx \rho_1 L \frac{S^{j+1} - S^j}{\Delta t} \approx \lambda_1 \frac{\tilde{T}^{j+1}_{1,N+1} - \tilde{T}^{j+1}_{1,N-1}}{2\,\Delta x_1} - \lambda_2 \frac{\tilde{T}^{j+1}_{2,1} - \tilde{T}^{j+1}_{2,-1}}{2\,\Delta x_2}, \tag{13.3.7}$$

where we use the expressions that were obtained in Exercise 13.3.2. From above expression S^{j+1} is easily computed. We update the grid by computing $\Delta x_1 = \dfrac{S^{j+1}}{N}$ and $\Delta x_2 = \dfrac{1 - S^{j+1}}{N}$, further note that at all stages we have

$$T^{j+1}_{1,N} = T_S = T^{j+1}_{2,0}. \tag{13.3.8}$$

The temperature at the new grid nodes are obtained using linear interpolation. Let $x^j_{k,i}$ denote the position of the i^{th} grid point in Ω_k at time step j, i.e.

$$x^j_{k,i} = \begin{cases} i\,\Delta x_1, \text{ for } k = 1 \\ \\ S^j + i\,\Delta x_2, \text{ for } k = 2 \end{cases}, \tag{13.3.9}$$

and the tildes represent the temperatures that have been computed from the discretization of the heat equation in both regions, then

$$T^{j+1}_{k,i} = \tilde{T}^{j+1}_{k,i} + \frac{\partial \tilde{T}^{j+1}_k}{\partial x}|_i (x^{j+1}_{k,i} - x^j_{k,i}) \approx \tilde{T}^{j+1}_{k,i} + \frac{\tilde{T}^{j+1}_{k,i+1} - \tilde{T}^{j+1}_{k,i-1}}{2\Delta x_k}(x^{j+1}_{k,i} - x^j_{k,i}) \tag{13.3.10}$$

Now the temperature has been updated correctly. Using this interpolation, the time integration can be rewritten, after division by Δt of Equation (13.3.10), by

$$\frac{T^{j+1}_{k,i} - \tilde{T}^{j+1}_{k,i}}{\Delta t} - \frac{x^{j+1}_{k,i} - x^j_{k,i}}{\Delta t} \frac{\tilde{T}^{j+1}_{k,i+1} - \tilde{T}^{j+1}_{k,i-1}}{2\Delta x_k} = 0, \qquad k \in \{1, 2\}. \tag{13.3.11}$$

Equation (13.3.11) represents a discrete convection equation, with mesh velocity

$$v_{\text{mesh}} = \frac{x^{j+1}_{k,i} - x^j_{k,i}}{\Delta t} \tag{13.3.12}$$

that is solved explicitly and using a central discretization. Since the temperature is smooth within Ω_1 and Ω_2, central discretization does not produce any unphysical wiggles (see Chapter 1). However, explicit time integration gives conditional stability: $\dfrac{u_{\text{mesh}} \Delta t}{\Delta x} < 1$ (see Chapter 1). For most practical situations, this is not so limiting since u_{mesh} is usually small. We end the one-dimensional description with some general remarks:

1. We have described the moving-grid method for cases when both Ω_1 and Ω_2 exist. In practice one assumes at $t = 0$ that ice is already present in a very thin layer. When the ice layer is small, one can use fewer grid nodes within Ω_1 and similarly fewer grid nodes within Ω_2 when Ω_2 almost vanishes. This implies that at certain times the grid needs to be regenerated.

2. In the above presentation of the numerical method, the determination of the position of the interface is rather inaccurate. One can use an iterative Trapezium method to improve the accuracy of the position of the interface. We omit the treatment here and refer the interested reader to [43].

A two-dimensional example
For the illustration of the two-dimensional solution of a Stefan problem, we consider an 'ice-disc' in a rectangular domain filled with water. For the sake of illustration we assume that the temperature in the 'ice-disc' is constant in space. Due to a relatively high water temperature the ice starts to melt and the circumference of the 'ice-circle' decreases. This gives a change of topology for the elements attached to the moving boundary. Mathematically we deal with the following problem:

$$\rho_2 c_2 \frac{\partial T}{\partial t} = \lambda_2 \triangle T, \qquad \mathbf{x} \in \Omega \tag{13.3.13}$$

$$T = T_0, \qquad \mathbf{x} \in \Omega, \tag{13.3.14}$$

$$T = T_s, \qquad \mathbf{x} \in S(t), \tag{13.3.15}$$

$$T = T_*, \qquad \mathbf{x} \in \Omega, \tag{13.3.16}$$

$$\frac{\partial T}{\partial n} = 0, \qquad \mathbf{x} \in \bigcup_{k=2}^{4} \partial\Omega_k \tag{13.3.17}$$

$$\rho_1 L v_n = -\lambda_2 \frac{\partial T}{\partial n}, \qquad \mathbf{x} \in S(t). \tag{13.3.18}$$

The equations in (13.3.13)-(13.3.18) are solved using a Finite Element method in both subdomains and the discrete temperature gradients are substituted into the rate Equation (13.3.18) to obtain the speed and position of the grid nodes at the interface as a function of time. An unstructured grid is used for the Finite Element discretization.

Exercise 13.3.3 *Derive a Finite Element formulation for the differential equation for the temperature with boundary and initial conditions in the two-dimensional heat diffusion problem* (13.3.13)-(13.3.17)*, where we have as essential condition $T = T_s$ at the interface S.*

Figure 13.4 shows an example of the initial mesh in subdomain Ω_2. In this subdomain moves the circular inclusion Ω_1. The interface is approximated by a spline. Frequently the number of the nodes at the interface and in the subdomains is kept constant. However, this is not necessary. Once the interface has been moved, the position of the mesh points inside the subdomains are adapted. The value of the temperature at the new positions of the grid nodes are unknown. To obtain these values, one can either use interpolation or a correction for the displacement. Since interpolation is rather expensive, a correction, taking into account the velocity of the grid nodes, is recommended. When we compute the time-derivative between the old and new points, a material derivative, as in fluid mechanics, is used:

$$\frac{dT}{dt} = \frac{\partial T}{\partial t} + \mathbf{u}_{\text{mesh}} \cdot \nabla T, \tag{13.3.19}$$

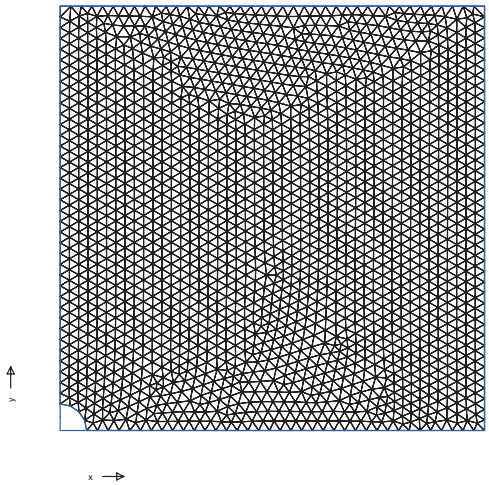

Figure 13.4: The initial mesh in Ω_2.

with the mesh-velocity $\mathbf{u}_{\text{mesh}} = \dfrac{d}{dt}\mathbf{x}$.

The temperature T is determined from

$$\frac{dT}{dt} - \mathbf{u}_{\text{mesh}} \cdot \nabla T = \lambda \Delta T, \text{ for all interior mesh points.} \tag{13.3.20}$$

Above treatment is known as the Arbitrary Lagrangian Eulerian (ALE) method and is very common in fluid dynamics. For a complete description of the Front tracking method for a one-dimensional case we refer to Murray and Landis [28], for two dimensions we refer to Segal et al [19]. The moving mesh is shown in Figure 13.5. During the adaptation of the mesh the quality of the mesh must be checked. As the length of the interface may change, the angles at the mesh-points change as well especially near the moving interface. Further due to the interface movement elements within both subdomains become either stretched or contracted. To avoid ill-shaped elements, remeshing may be necessary. Remeshing is expensive since the values of the temperature at the new mesh points have to be determined using interpolation and a new mesh must be generated.

Figure 13.6 shows an example where the mesh has not been checked at the boundary. Here the elements at the interface have become stretched and hence ill-shaped. An example of remeshing is presented in Figure 13.7, where we display the mesh at a point in time when the subdomain Ω_1 has grown significantly. It can be seen, also from Figure 13.4, that initially there were 5 grid nodes at the moving boundary. For larger values of time, when the subdomain Ω_1 has grown, the number of grid nodes at the interface has increased, see Figure 13.7. Let β be the angle of the elements in the domain, then remeshing has been applied on the basis of the following criterion:

$$\beta_{\min} \leq \beta \leq \beta_{\max},$$

where $\beta_{\min} = 10^o$ and $\beta_{\max} = 120^o$. Whenever some element in the domain does not satisfy this criterion, the domain is remeshed.

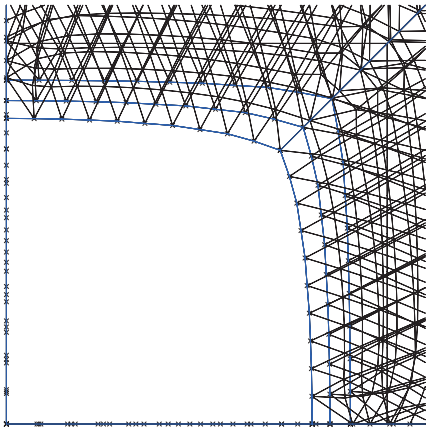

Figure 13.5: The moving mesh of Ω_2. The blue lines represent the moving boundary after t = 0.

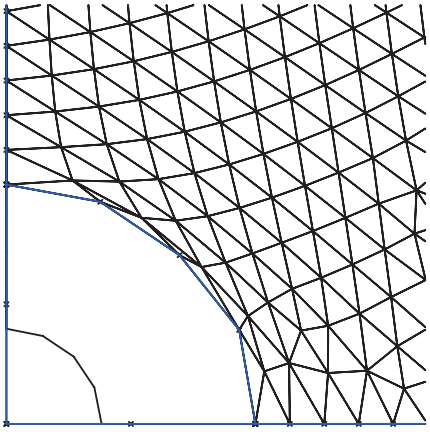

Figure 13.6: The mesh after significant growth of subdomain Ω_1 with the original mesh topology. The blue and black lines respectively represent the moving interface for $t > 0$ and t = 0.

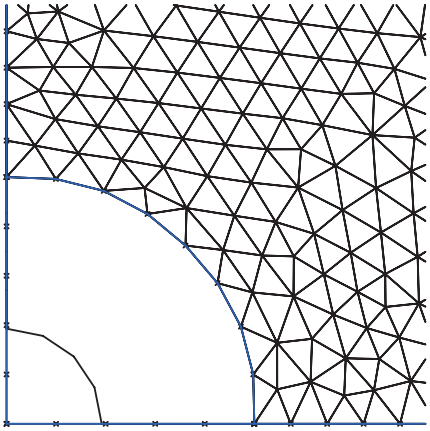

Figure 13.7: The mesh after significant growth of subdomain Ω_1 with mesh topology after remeshing. The blue and black lines respectively represent the moving interface for t $>$ 0 and t = 0. Remeshing is applied whenever one of the angles of the elements falls outside of the range [10,120] degrees.

It will be clear that moving grid methods are rather expensive due to remesh-ing and difficult to program, especially in three dimensional problems. A great advantage, however, is that the interface is part of the boundary of the elements and therefore interface conditions can be satisfied easily.

13.3.2 A fixed domain method: the level set method

As an example of a Fixed grid method we consider the level set method. The method does not track the interface explicitly. The method is conceptually less self-evident than the moving grid method. This is mainly because of the intro-duction of the *level set* function, which is sometimes also referred to as a 'pseudo-temperature'. The method is very powerful especially for three dimensions. The level set method was first introduced by Osher and Sethian [29]. First we outline the level set method in a general way and subsequently we describe an application to a 2D-Stefan problem. The application is for solidification and melting as studied by Chen et al [10]. The idea behind the level set function is as follows: We define an extra unknown, the *pseudo-temperature*. This unknown is only meant to define the interface implicitly. The sign of the pseudo-temperature determines in which phase or subdomain a node is at particular instant of time. Furthermore, the interface position coincides with the zero level of the pseudo-temperature and hence this position is tracked implicitly. Since the interface is convected by some velocity, one can derive a convection equation for the pseudo-temperature, which is solved together with the original PDE. The principles and the concept pseudo-temperature are outlined in the coming subsections.

Application to the Stefan problem
Consider, as before, the heat equation in a solid and a liquid phase Ω_1 and Ω_2 with interface $S(t)$:

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (\lambda \nabla T), \qquad \mathbf{x} \in \Omega, \tag{13.3.21}$$

$$\rho_1 L v_n = \lambda_1 \frac{\partial T_1}{\partial n} - \lambda_2 \frac{\partial T_2}{\partial n}, \qquad \mathbf{x} \in S(t), \tag{13.3.22}$$

with similar initial and boundary conditions as in Equations (13.1.1)-(13.1.6). The complete region is used as computational domain. We define the *level set* function, which is negative in one phase and positive in the other phase, ϕ. At time $t = 0$, ϕ is defined as

$$\phi = \begin{cases} -d(\mathbf{x}), & \text{for } \mathbf{x} \in \Omega_1(0) \\ 0, & \text{for } \mathbf{x} \in S(0) \\ +d(\mathbf{x}), & \text{for } \mathbf{x} \in \Omega_2(0) \end{cases}, \tag{13.3.23}$$

where the most important feature of ϕ is $\phi(\mathbf{x}, 0) = 0$ on $S(0)$ and that $\phi < 0$ on Ω_1 and $\phi > 0$ on Ω_2. Hence the level set function ϕ is used to indicate in which phase a specific gridnode is. The level set function ϕ is also sometimes referred to as a 'pseudo-temperature'. We prescribe the level set function ϕ at $t = 0$ as a signed distance function: the function $d \geq 0$ denotes the minimal distance between a certain point $\mathbf{x} \in \Omega$ and the boundary $S(t)$. Also for $t > 0$ we require $\phi = 0$ at $S(t)$, $\phi < 0$ in Ω_1 and $\phi > 0$ in Ω_2. From the definition of the function ϕ follows that the interface can be determined for each given ϕ:

$$S(t) = \{(x, y) \in D : \phi(x, y, t) = 0\} \text{ for } t \geq 0. \tag{13.3.24}$$

Since $\phi(x(t), y(t), t) = 0$ at the interface, it follows that the total derivative with respect to time vanishes at the moving boundary

$$\frac{D}{Dt}\phi = \phi_t + \nabla\phi \cdot \mathbf{r}' = 0 \text{ for } \mathbf{r} \in S(t).$$

Here $\mathbf{r}'(t)$ represents the speed of a point $\mathbf{r}(t) \in S(t)$ at the interface, where $\phi = 0$. This point coincides with the moving boundary and hence has the same speed as the moving boundary, so

$$\mathbf{r}'(t) = [\lambda \nabla T] := \lambda_1 \nabla T_1 - \lambda_2 \nabla T_2, \text{ for } \mathbf{x} \in S(t). \tag{13.3.25}$$

This implies that the total derivative for points on the interface with respect to time t (where $\phi = 0$) can be written as

$$\phi_t + \nabla\phi \cdot [\lambda \nabla T] = 0 \text{ for } \mathbf{x} \in S(t). \tag{13.3.26}$$

Above equation only holds at the moving interface. In order to have ϕ as a signed continuous function on the whole domain Ω, it must be prescribed in other positions of Ω as well. We do this by the use of the following PDE

$$\phi_t + \mathbf{u} \cdot \nabla\phi = 0 \text{ for } \mathbf{x} \in \Omega, \tag{13.3.27}$$

where the vector function $\mathbf{u} : \mathbb{R}^2 \times R_0^+ \to \mathbb{R}^2$, will be defined as a continuous extension of $[\lambda \nabla T]$ over Ω, i.e.

$$\mathbf{u} \in U := \{\mathbf{u} \in C(\Omega) : \mathbf{u} = [\lambda \nabla T] \text{ for } \mathbf{x} \in S(t)\} \text{ for } t > 0. \tag{13.3.28}$$

We proceed to construct a continuous extension for $\mathbf{u}$ near $S(t)$, by which we mean that $\mathbf{u}$ is continuous at points near the interface position. This is based on the principles which are described in [10].

For the moment we assume that we have an initial value for $\mathbf{u}$ for points not on S. Furthermore, $\mathbf{u}$ is prescribed on S by the initial value of T. This implies that if the components of $\mathbf{u} = (u_1, u_2)$ satisfy a first order hyperbolic (convection) equation, then $\mathbf{u}$ is continuous near $S(t)$. This convection problem is well-posed

as long as the prescribed value is upwind from points away from $S(t)$ where a 'boundary condition' is imposed. Therefore, we set

$$\frac{\partial u_1}{\partial \tau} + sign(\phi\phi_x)\frac{\partial u_1}{\partial x} = 0, \tag{13.3.29}$$

$$\frac{\partial u_2}{\partial \tau} + sign(\phi\phi_y)\frac{\partial u_2}{\partial y} = 0, \tag{13.3.30}$$

$$\text{subject to } u_1 = \left[\lambda\frac{\partial T}{\partial x}\right], \quad u_2 = \left[\lambda\frac{\partial T}{\partial y}\right], \quad \text{for } \mathbf{x} \in S(t), \tag{13.3.31}$$

then $\mathbf{u}$ points away from $S(t)$ and a well-posed definition of $\mathbf{u}$ is obtained. In the above equations τ is a pseudotime, since the reason for the use of the above equation is just to extend the velocity continuously near the interface. Further, for $(x, y) \in S(t)$, implying $\phi = 0$ and hence $sign(\phi\phi_x) = 0$ and $sign(\phi\phi_y) = 0$, we have

$$\frac{\partial}{\partial \tau}\mathbf{u} = 0 \Rightarrow \mathbf{u} = [\lambda \nabla T] \text{ for } \mathbf{x} \in S(t). \tag{13.3.32}$$

In principle we have given a full system of PDE's to solve the Stefan problem using the level set method. The level set function ϕ defines the position of the interface. Further, defining it as a signed distance function, it satisfies nice monotonicity properties and the closer to zero a particular gridnode value is, the closer the corresponding grid node is to the interface. It is particularly important to have information whether a grid node is in subdomain Ω_1 or Ω_2 when the coefficients in the heat equation are determined. Therefore, the sign of the level set function is crucial. We want to have this information without having to track the position of the moving interface explicitly like in the moving grid method. Furthermore, for boundary conditions at the moving interface it is important to know whether a gridnode is a neighbor of the interface.

One-dimensional implementation

As an example of the application of the level set method, we consider the 1D equation

$$\rho c\frac{\partial T}{\partial t} = \lambda\frac{\partial^2 T}{\partial x^2}, \quad x \in \Omega \tag{13.3.33}$$

$$\rho_1 L\frac{dS}{dt} = \lambda_1\frac{\partial T_1}{\partial x} - \lambda_2\frac{\partial T_2}{\partial x}, \quad x = S(t), \tag{13.3.34}$$

$$T = T_*, \quad x = 0, \tag{13.3.35}$$

$$T = T_s, \quad x = S(t) \tag{13.3.36}$$

$$\frac{\partial T}{\partial x} = 0, \quad t > 0, \tag{13.3.37}$$

$$T = T_0, S = 0, \quad t = 0. \tag{13.3.38}$$

We describe the solution procedure at each time-step. We suppose here that ϕ^j, T^j, u^j are known. Let j be the time-step index, $t_j = j\Delta t$ and $h = \frac{1}{N}$ be the grid-spacing. At each time-step we first solve the temperature field in both subdomains using Finite Differences (see Figure 13.8), to obtain

$$\rho_2 c_2\frac{T_i^{j+1} - T_i^j}{\Delta t} = \lambda_2\frac{T_{i+1}^{j+1} - 2T_i^{j+1} + T_{i-1}^{j+1}}{h^2}, \quad i \in \{1, \ldots, p-2\}. \tag{13.3.39}$$

$$\rho_1 c_1\frac{T_i^{j+1} - T_i^j}{\Delta t} = \lambda_1\frac{T_{i+1}^{j+1} - 2T_i^{j+1} + T_{i-1}^{j+1}}{h^2}, \quad i \in \{p+1, \ldots, n\}, \tag{13.3.40}$$

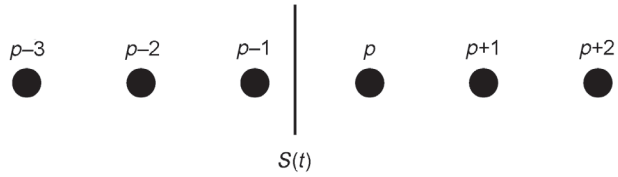

Figure 13.8: The gridpoints in the neighborhood of the moving interface.

For the gridpoints near the interface, we use the fact that ϕ represents a signed distance function. To guarantee a $O(h^2)$ accuracy we use ghost points at both sides of the interface. Let $\hat{T}_p$ and $\hat{T}_{p-1}$ respectively represent ghost points belonging to $\Omega_1 := \{x \in \mathbb{R} : 0 < x < S(t)\}$ and $\Omega_2 := \{x \in \mathbb{R} : S(t) < x < 1\}$. We treat the boundary condition at the right-hand side of the interface, in Ω_2, and the treatment for the boundary condition at the side of the subdomain Ω_1 is left for the reader. For $i = p$, we then have

$$\rho_2 c_2 \frac{T_p^{j+1} - T_p^j}{\Delta t} = \lambda_2 \frac{T_{p+1}^{j+1} - 2T_p^{j+1} + \hat{T}_{p-1}^{j+1}}{h^2} + O(h^2). \tag{13.3.41}$$

The value for $\hat{T}_{p-1}$is computed from a Taylor expansion around $\phi = 0$, i.e. the interface S:

$$\hat{T}_{p-1} = T_S + (x_{p-1} - S)\frac{\partial T}{\partial x}|_S + O(h^2). \tag{13.3.42}$$

The position of the interface, S, is obtained as the zero level of the function ϕ. Having two values of ϕ with opposite sign, it is clear that the interface is between the two corresponding positions. The interface position is then obtained by interpolation using these points. The derivative of T at S is obtained from a Taylor expansion around ϕ_p:

$$\frac{\partial T}{\partial x}|_S = \frac{\partial T}{\partial x}|_{x_p} - (x_p - S)\frac{\partial^2 T}{\partial x^2}|_{\phi_p} + O(h^2) \tag{13.3.43}$$

$$= \frac{T_{p+1} - \hat{T}_{p-1}}{2h} - (x_p - S)\frac{T_{p+1} - 2T_p + \hat{T}_{p-1}}{h^2} + O(h^2) \tag{13.3.44}$$

Equation (13.3.44) is substituted into Equation (13.3.42) where an expression for $\hat{T}_{p-1}$ is obtained (after dropping the $O(h^2)$-terms) and subsequently substituted into equation (13.3.41). A similar procedure is done for the left-hand side of the interface, and the matrix equation is subsequently solved. We need the gradients of the temperature at the interface for the velocity of the interface (see the equation (17.49). When we use for the discretization of the gradient the one-sided discretization formulas

$$\frac{\partial T_2}{\partial x}(S(t), t) = \frac{T_p - T_S}{x_p - S} \tag{13.3.45}$$

$$\frac{\partial T_1}{\partial x}(S(t), t) = \frac{T_S - T_{p-1}}{S - x_{p-1}}, \tag{13.3.46}$$

for the determination of the gradients, it can be seen that once $S \to x_p$ or $S \to x_{p-1}$ division by zero results. This causes the jumps in the velocity of the interface as shown in Figure 13.11. Therefore, we use for the gradients at the interface at both

sides:

$$\frac{\partial T_2}{\partial x} = \frac{T_p - T_S}{x_p - S}\frac{x_p - S}{h} + (1 - \frac{x_p - S}{h})\frac{T_{p+1} - T_p}{h} \tag{13.3.47}$$

$$= \frac{T_p - T_S}{h} + (1 - \frac{x_p - S}{h})\frac{T_{p+1} - T_p}{h} \tag{13.3.48}$$

$$\frac{\partial T_1}{\partial x} = \frac{T_S - T_{p-1}}{S - x_{p-1}}\frac{S - x_{p-1}}{h} + (1 - \frac{S - x_{p-1}}{h})\frac{T_{p-1} - T_{p-2}}{h} \tag{13.3.49}$$

$$= \frac{T_S - T_{p-1}}{h} + (1 - \frac{S - x_{p-1}}{h})\frac{T_{p-1} - T_{p-2}}{h} \tag{13.3.50}$$

This computation of gradients is called the *weighted gradients approach* . The result using the weighted gradient approach is represented by the blue curve in Figure 13.11. Using both gradients, we determine the velocity of the interface. Given the interface velocity, we solve the following equation to obtain the velocity field over the whole domain, with the known level set function ϕ:

$$\frac{\partial u}{\partial \tau} + sign(\phi\phi_x)\frac{\partial u}{\partial x} = 0, \quad x \in (0,1) \tag{13.3.51}$$

$$u(S) = \lambda_1 \frac{\partial T_1}{\partial x}(S) - \lambda_2 \frac{\partial T_2}{\partial x}(S) \tag{13.3.52}$$

Here τ represents a pseudotime since the above equation just artificially extends the velocity continuously. After the update of u we compute the update of the signed distance function from solution of

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0. \tag{13.3.53}$$

Solution of Equations (13.3.53) and (13.3.51), (13.3.52) is done using upwind differences. Note that in the solution of Equation (13.3.53) the discretization of $\frac{\partial \phi}{\partial x}$ depends on the sign of u. The function ϕ was initially chosen to be a signed distance function. However, at the course of the iteration process, the function ϕ looses this property. This is not so bad in general, since only smoothness of ϕ is necessary in all the steps taken until now. Hence, if ϕ is a signed distance function at all time steps, then ϕ is continuous and then all the operations until now are allowed. Therefore, one often requires ϕ to be a distance function although this is not necessary. However, often it is desirable to have ϕ as a signed distance function if local curvatures of the interface are used. This is often done in relation to surface tension. Furthermore, having ϕ as a distance function, guarantees that ϕ is continuous, which is a necessary requirement. Therefore, we iterate

$$\frac{\partial \phi}{\partial \tau} = sign(\phi_0)(1 - |\frac{\partial \phi}{\partial x}|). \tag{13.3.54}$$

In order to get a signed distance function it is necessary that $|\frac{\partial \phi}{\partial x}| = 1$. To satisfy this we consider Φ as the stationary solution of (13.3.54). This equation is solved using the time-step as a kind of convergence parameter. Furthermore, τ is a pseudotime, which is artificial for the convergence towards a stationary solution. We show some results for 100 gridpoints for the freezing of water in Figures 13.9, 13.10. Initially, we have an ice-layer of 3 gridpoints (0.03 m) and we apply a temperature $T_* = -23°C$. The temperature-profile after 1 and 3 days is shown in Figure 13.9. The position of the interface is shown in Figure 13.10. A good correspondence is observed with the analytical solution (see Figure 13.2 and 13.3). However, since the water area is bounded $(S(t), 1)$ for the numerical solution, the ice-thickness

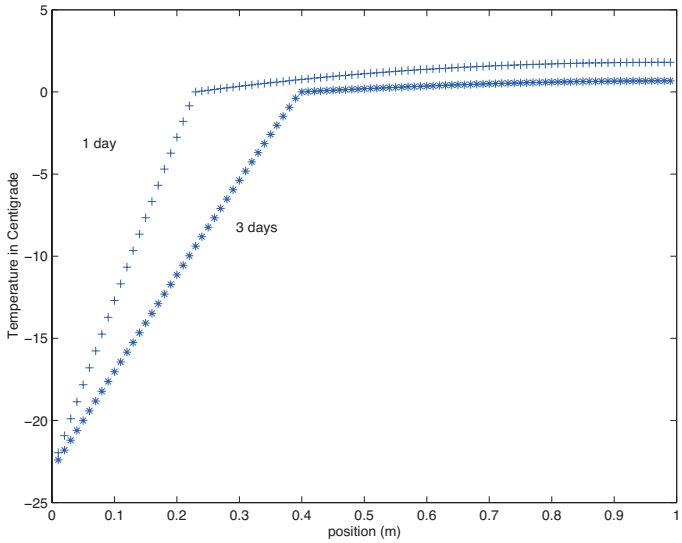

Figure 13.9: Temperature profiles of freezing water at subsequent times. The data have been taken from Table 13.1. The temperature at x = 0 is −23 °C.

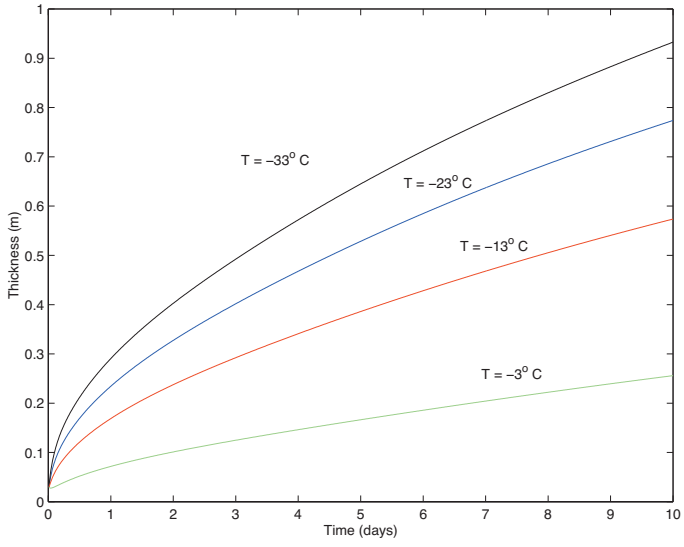

Figure 13.10: The ice-thickness as a function of time for different temperatures at x = 0. Further data have been taken from Table 13.1.

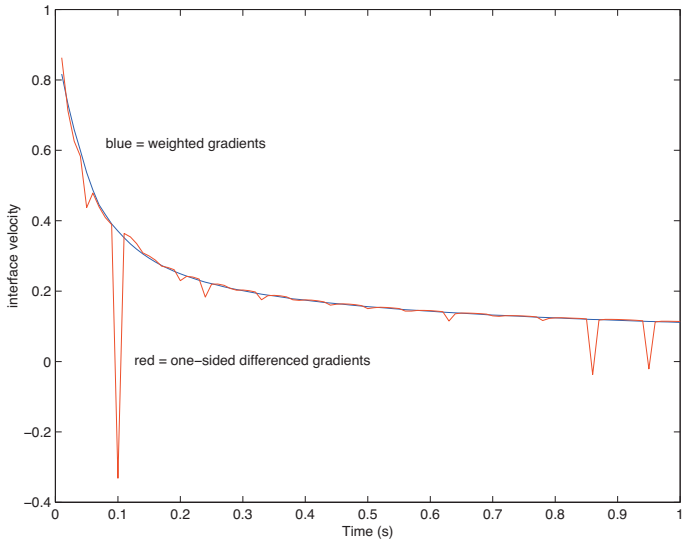

Figure 13.11: The velocity of the interface as a function of time for T = −23 °C at x = 0. Further data are from Table 13.1. The red curve represents the calculation for the one-sided gradients.

evolves faster than for the analytical solution where the water area is unbounded $(S(t), \infty)$. Furthermore, we show the velocity of the interface as a function of time in Figure 13.11. The smooth curve represents the use of the weighted gradients.

13.3.3 Other applications of Stefan problems

Stefan problems occur also, among others, during solidification of metals or in the solid state. Here typical problems of phase transformations involving different lattices (crystals) take place. Some examples are the dissolution or growth of particles in ferrous or non-ferrous alloys or the phase-transformation of ferrite to austenite in steels. Both processes occur during production and optimization of high quality metals and alloys. Models can be found in the book of Visintin [44]. Furthermore, similar numerical techniques are used to solve free boundary problems. We mention the seepage of water through a porous dam as an example of a free boundary problem. This free boundary problem has been described in Crank [13]. Furthermore, it should be noted that Stefan problems are also solved by different methods, such as the phase-field approach, enthalpy method and the method of variational inequalities. These methods are beyond the scope of the book and their principles can be found in textbooks as [44], [13].

13.4 Summary of Chapter 13

In this chapter some examples of Stefan problems are formulated. A moving grid method to solve the Stefan problem is described. This method is conceptually simple, however for cases where several moving boundaries merge, the method fails. Further, the interpolation that has to be applied can be expensive. Next, a fixed grid method, the level set method, is described as a conceptually less obvious method. Here the interface was taken into account in an implicit way and hence the

determination of the exact position of the interface may be less accurate. However, the method is successfully used in cases where several interfaces merge or come close together. Both methods have its benefits and disadvantages. Further, the class of so-called self-similarity solutions have been referred to as a tool to validate the behavior of solutions obtained from numerical methods in a qualitative way.

Bibliography

[1] R.A. Adams. *Sobolev Spaces*. Academic Press, New York, 1975.

[2] Robert A. Adams. *Calculus, a complete course. Fifth Edition.* Addison Wesley Longman, Toronto, 2003.

[3] R. Aris. *Vectors, Tensors and the Basic Equations of Fluid Mechanics*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1962. Reprinted, Dover, New York, 1989.

[4] J. Bear. *Dynamics of Fluids in Porous Media*. American Elsevier publishing company, New York, 1972.

[5] E.K. Blum. *Numerical Analysis and Computation: Theory and Practice*. Addison-Wesley Publishing Company, Reading, Mass, 1972.

[6] A.N. Brooks and T.J.R. Hughes. Stream-line upwind/Petrov Galerkin formulstion for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equation. *Comp. Meth. Appl. Mech. Eng.*, 32:199–259, 1982.

[7] R.L. Burden and J.D. Faires. *Numerical analysis*. Brooks/Cole, Pacific Grove, 2001.

[8] J.R. Cannon. *The one-dimensional heat equation*. Addison-Wesley Publishing company, Menlo park, California, U.S.A., 1984.

[9] H.S. Carslaw and J.C. Jæger. *Conduction of heat in solids*, volume 2. Clarendon Press, Oxford, 1988.

[10] S. Chen, B. Merriman, S. Osher, and P. Smereka. A simple level-set method for solving stefan problems. *J. Comp. Phys.*, 135:8–29, 1997.

[11] Ph.G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland, Amsterdam, 1978.

[12] R. Courant and D. Hilbert. *Methods of Mathematical Physics, Vol. 2. Partial Differential Equations.* Interscience, New York, 1989.

[13] J. Crank. *Free and Moving Boundary Problems*. Clarendon Press, Oxford, 1984.

[14] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proc. 24th Nat. Conf. Assoc. Comput. Mech.*, pages 1–69, New York, 1969. ACM publ.

[15] C. Cuvelier, A. Segal, and A.A. van Steenhoven. *Finite Element Methods and Navier-Stokes Equations*. Reidel Publishing Company, Dordrecht, Holland, 1986.

[16] L.C. Evans. *Partial Differential Equations*. American Mathematical Society, 2010.

[17] A. George and J.W.H. Liu. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, Engelwood Cliffs, New Jersey, (USA), 1981.

[18] Wolfgang Hackbusch. *Multi-grid methods and applications*. Springer, Berlin, 2003 (1986).

[19] C.W. Hirt, A.A. Amsden, and J.L. Cook. An Arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14:227–253, 1974.

[20] I. Holand and K. Bell eds. *Finite Element Methods in Stress Analysis*. Tapir, 1969, Trondheim, Norway, 1969.

[21] B.M. Irons and A. Razazaque. Expericience with the patch test for convergence of finite elements. In A.K. Aziz, editor, *The mathemattical foundations of the finite element method with applications to partial differential equations*, pages 557–587, New-York, 1972. Academic Press.

[22] E. Kreyszig. *Introductory Functional Analysis with Applications*. Wiley, New York, 1989.

[23] J.D. Lambert. *Numerical methods in ordinary differential equations*. John Wiley, Englewood Cliffs, 1991.

[24] David C. Lay. *Linear Algebra and its applications*. Addison Wesley, New York, 1993.

[25] R.J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser, Basel, 1992.

[26] A.R. Mitchell and D.F. Griffiths. *The Finite Difference Method in Partial Differential Equations*. Wiley, Chichester, 1994.

[27] A.R. Mitchell and R. Wait. *The finite element method in partial differential equations*. Wiley, Chichester, 1977.

[28] W.D. Murray and F. Landis. Numerical and machine solutions of transient heat-conduction problems involving melting or freezing. *Trans. ASME (C) J. Heat Transfer*, 81:106–112, 1959.

[29] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[30] M.H. Protter and H.F. Weinberger. *Maximum Principles in Differential Equations*. Prentice-Hall, Englewood Cliffs, 1967.

[31] J.N. Reddy. *An introduction to the finite element method*. McGraw-Hill, New York, 1984.

[32] P.P. Silvester and R.L. Ferrari. *Finite elements for electrical engineers*. Cambridge University Press, Cambridge, 1983.

[33] J. Smoller. *Shock Waves and Reaction-Diffusion Equations*. Springer, New York, 1983.

[34] J. Stefan. Über die Theorie der Eisbildung, insbesondere über die Eisbildung im Polarmeere. *Annalen der Physik und Chemie*, 42:269–286, 1891.

[35] James Stewart. *Calculus. Fifth Edition*. Brooks/Cole, New York, 2002.

[36] G. Strang. *Linear Algebra and its Applications, (third edition)*. Harcourt Brace Jovanovich, San Diego, 1988.

[37] G. Strang and G.J. Fix. *An Analysis of the Finite Element Method*. Prentice Hall Inc., Englewood Clifts, New Jersey, 1973.

[38] P.K. Sweby. High resolution schemes using flux-limiters for hyperbolic conservation laws. *SIAM J. Num. Anal.*, 21:995–1011, 1984.

[39] R. Temam. *Navier-Stokes Equations*. North-Holland, Amsterdam, 1985.

[40] U. Trottenberg, C.W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2001.

[41] H.A. van der Vorst. *Iterative Krylov Methods for Large Linear Systems*. Cambridge University Press, Cambridge, UK, 2003.

[42] R.S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, N.J., 1962.

[43] F. Vermolen and K. Vuik. A numerical method to compute the dissolution of second phases in ternary alloys. *J. Comp. Appl. Math.*, 93:123–143, 1998.

[44] A. Visintin. *Models of phase transitions, Progress in nonlinear differential equations and their applications*, volume 28. Birkhäuser, Boston, 1996.

[45] C. Vuik. *The Solution of a One-Dimensional Stefan Problem*. CWI-tract 90. CWI, Amsterdam, 1993.

[46] C. Vuik. Some historical notes about the stefan problem. Reports of the Faculty of Technical Mathematics and Informatics 93–07, Delft, 1993.

[47] H. Weber. *Die partiellen Differential-Gleichungen der Mathematischen Physik*. Vieweg, Braunschweig, 1901.

[48] Pieter Wesseling. *An introduction to Multigrid Methods*. John Wiley & Sons, Ltd, 1992. Corrected reprint. R.T. Edwards Inc., Philadelphia, 2004.

[49] K. Yosida. *Functional Analysis*. Springer Verlag, Berlin, 1971.

[50] O.C. Zienkewicz. *Finite element method in Engineering Science*. Mc Graw-Hill, 1971.

Index

Numerical Methods in Scientific Computing

Jos van Kan, Guus Segal, Fred Vermolen

This is a book about numerically solving partial differential equations occurring in technical and physical contexts and the authors have set themselves a more ambitious target than to just talk about the numerics. Their aim is to show the place of numerical solutions in the general modeling process and this must inevitably lead to considerations about modeling itself. Partial differential equations usually are a consequence of applying first principles to a technical or physical problem at hand. That means, that most of the time the physics also have to be taken into account especially for validation of the numerical solution obtained. This book aims especially at engineers and scientists who have 'real world' problems. It will concern itself less with pesky mathematical detail. For the interested reader though, we have included sections on mathematical theory to provide the necessary mathematical background. Since this treatment had to be on the superficial side we have provided further reference to the literature where necessary.



Jos van Kan, Retired professor Delft University of Technology, Delft Institute of Applied Mathematics

Guus Segal, Retired professor Delft University of Technology, Delft Institute of Applied Mathematics

Fred Vermolen, University of Hasselt, Department of Mathematics and Statistics, Computational Mathematics Group

ISBN 978-94-6366-740-1
DOI https://doi.org/10.59490/t.2023.009

textbooks.open.tudelft.nl

Cover image:
TU Delft OPEN Publishing.