Lecture Notes
Introduction to PDEs and Numerical Methods

Winter Term 2002/03

Hermann G. Matthies

Oliver Kayser-Herold

Institute of Scientific Computing
Technical University Braunschweig

Contents

Chapter 1

An Introductory Example

In this introductory chapter we will go through the steps of setting up a mathematical model for heat conduction. This will be derived from basic physical principles and will lead to the integral form of a partial differential equation. We will look at exact solutions for very idealised situations, in order to see the typical behaviour. For more complicated circumstances we have to resort to numerical methods. These will again be studied in a very idealised setting.

1.1 Derivation of the PDE

To illustrate the way how to derive a partial differential equation describing a physical system out of the basic laws of physics we will consider a simple rod consisting of a normal material (Fig. 1.1).

The rod should be insulated against any heat loss on the whole length. Only at the ends it can gain or loose heat. We are interested in the temperature distribution inside this rod at a specific time. As for most dynamical systems we must know the exact state at a given time t_0. And certainly the temperature at both ends is important, too.

1.1.1 Energy Conservation

The conservation law that seems right for this problem is the conservation of energy because the temperature is equivalent to the motion energy of the molecules that build the rod. First we have the heat (or thermal energy) in the rod. Its density per unit length is:

$$A \cdot \rho \cdot c \cdot \theta(x,t) \tag{1.1}$$

Here ρ is the density of the material, c the specific heat capacity, and A the cross sectional area of the rod. They could all be functions of space and temperature, but for the sake

of simplicity we shall assume them to be constant. The function $\theta(x,t)$ describes the temperature at a given point in space and time. And so the change of energy of an arbitrary piece of rod from a to b is:

$$R_1 = \frac{\partial}{\partial t}\int_a^b A \cdot \rho \cdot c \cdot \theta(x,t)dx \tag{1.2}$$

Next we will take a look at the energy that goes into the rod or out of it. As mentioned before this can only happen at the two ends of the rod.

There we have the heat flow which is described by the function $q(x,t)$. So the energy that goes into the rod is:

$$R_2 = A \cdot (q(a,t) - q(b,t)) = -A \cdot (q(b,t) - q(a,t)) \tag{1.3}$$

This equation can be transformed with the fundamental theorem of calculus into:

$$-A \cdot (q(b,t) - q(a,t)) = -A\int_a^b \left(\frac{\partial}{\partial x}q(x,t)\right)dx \tag{1.4}$$

Finally we assume an internal source of heat. This effect should model something similar to a microwave oven which heats something from the inside. We introduce the function $h(x,t)$ which describes the power density of additional heat sources.

$$R_3 = \int_a^b A \cdot h(x,t)dx \tag{1.5}$$

Conservation of energy means that we must have

$$R_1 = R_2 + R_3 \tag{1.6}$$

Inserting the equations again into the short form gives:

$$\frac{\partial}{\partial t}\int_a^b Ac\rho\theta(x,t)dx = -A\int_a^b \frac{\partial}{\partial x}q(x,t)dx + A\int_a^b h(x,t)dx \tag{1.7}$$

Separating the parts of the equation with known functions and the parts with unknown functions leads to:

$$Ac\rho\int_a^b \frac{\partial}{\partial t}\theta(x,y)dx + A\int_a^b \frac{\partial}{\partial x}q(x,t)dx = A\int_a^b h(x,t)dx \tag{1.8}$$

Finally we obtain for any $a, b \in [0,l]$:

$$\int_a^b \left[c\rho \frac{\partial}{\partial t}\theta(x,t) + \frac{\partial}{\partial x} q(x,t) \right] dx = \int_a^b h(x,t)dx \tag{1.9}$$

This is the integral form of the PDE.

1.1.2 From the Integral Form to the PDE

Up to this point all equations were integral equations which gave some restrictions on the solution. Now the following lemma allows the transformation of these Integrals into a PDE under some conditions:

Lemma 1 (Fundamental lemma of calculus of variations) *Let φ be a continuous function $\varphi : [A, B] \to \mathbb{R}$. If for arbitrary $a, b \in [A, B]$ with $b > a$*

$$\int_a^b \varphi(x)dx = 0, \tag{1.10}$$

then

$$\forall x \in [a, b], \quad \varphi(x) = 0 \tag{1.11}$$

Proof (by contradiction) :

Assume $\exists x_0 : \varphi(x_0) > 0$

φ continuous $\Rightarrow$ there is a neighbourhood of x_0, $([x_0 - \varepsilon, x_0 + \varepsilon])$ where $\varphi(x) \geq \delta > 0$. Then with $a = x_0 - \varepsilon, b = x_0 + \varepsilon$

$$\int_a^b \varphi(x)dx \geq \int_a^b \delta dx = \delta \int_a^b dx = \delta(b - a) = 2\delta\varepsilon > 0 \tag{1.12}$$

in contradiction to Eq. (1.10).

Going back to the relations describing the heat transfer in the rod, we have the following equation:

$$\int_a^b \underbrace{\left[c\rho \frac{\partial}{\partial t}\theta(x,t) + \frac{\partial}{\partial x} q(x,t) - h(x,t) \right]}_{\varphi(x)} dx = 0 \tag{1.13}$$

If we assume that $\varphi(x)$ is continuous then the fundamental lemma of variational calculus gives directly the differential or pointwise form of the PDE:

$$c\rho \frac{\partial}{\partial t}\theta(x,t) + \frac{\partial}{\partial x} q(x,t) = h(x,t) \tag{1.14}$$

With given boundary conditions $q(a,t)$, $q(b,t)$ and initial conditions $\theta(x,0)$.

One important aspect of this assumption is that the expression under the integral has to be continuous. In contrast the original integral equation can also be satisfied by a discontinuous function which may appear in real life problems. So one must keep in mind that the partial differential equations come originally from the integral form and therefore the strict continuity requirements of the PDE may sometimes be neglected. In fact, in the sequel unless stated otherwise, write the differential form – as it is simpler – but we will mean the integral form.

1.1.3 Constitutive Laws

To get a solvable equation one of the two unknown functions must be replaced by a known function. Often this is done with a constitutive law which connects two physical properties with a function. For the heat equation the *Fourier Law* provides this kind of function.

$$q(x,t) = -\lambda \frac{\partial}{\partial x}\theta(x,t) \tag{1.15}$$

Where λ is the heat conductivity. This again could be a function of temperature or position, but again for simplicity we shall assume it constant. Inserting this constitutive law into the PDE gives finally the well known heat equation:

$$\rho \frac{\partial}{\partial t}\theta(x,t) - \frac{\partial}{\partial x}\left[\lambda \frac{\partial}{\partial x}\theta(x,t)\right] = h(x,t) \tag{1.16}$$

sorting the constants gives:

$$\frac{\partial}{\partial t}\theta(x,t) - \left(\frac{\lambda}{c\rho}\right)\frac{\partial^2}{\partial x^2}\theta(x,t) = \frac{h(x,t)}{c\rho} = \eta(x,t) \tag{1.17}$$

The time derivative will be abbreviated with a superposed dot:

$$\frac{\partial}{\partial t}\theta(x,t) = \dot{\theta}(x,t) \tag{1.18}$$

Another possible constitutive law which can be applied in this context is the law of convective transport. While Fourier's law describes a slow diffusive transport of energy, the convective transport is similar to putting a cup of hot water into a river. The energy is transported with the speed of the water flowing in the river:

$$q(x,t) = c\rho\theta v \tag{1.19}$$

where v is the velocity of the transport medium.

1.1.4 Initial and Boundary Conditions

Most PDEs have an infinite number of admissible solutions. Thus the PDE alone is not sufficient to get a unique solution. Usually some boundary conditions and initial conditions are required.

For the heat equation the simplest boundary conditions are fixed temperatures at both ends:

$$\theta(0,t) = h_1(t) \tag{1.20}$$

$$\theta(l,t) = h_2(t) \tag{1.21}$$

where l is the length of the rod and $h_1(t)$ the temperature at the first end and $h_2(t)$ the temperature at the second end.

The initial conditions specify an arbitrary initial temperature distribution inside the rod:

$$\theta(x,0) = \theta_0(x) \tag{1.22}$$

.

1.1.5 General Way of Modelling Physical Systems

Basically many PDEs in mathematical physics are derived in the way, shown in the example. So if we have a quantity with density u which should be conserved, the change of that quantity for an arbitrary piece $[a,b]$ is:

$$\frac{\partial}{\partial t}\int_a^b u dx \tag{1.23}$$

It is equal to the amount going in or out through the boundary with flow density p:

$$-p|_a^b \tag{1.24}$$

and the amount generated or consumed inside the domain:

$$\int_a^b j(x)dx \tag{1.25}$$

which finally gives us the general form of a conservation law:

$$\frac{\partial}{\partial t}\int_a^b u dx = -p|_a^b + \int_a^b j(x)dx \tag{1.26}$$

$$\Rightarrow \frac{\partial}{\partial t}\int_a^b u dx = -\int_a^b \frac{\partial}{\partial x} p + \int_a^b j(x) dx \tag{1.27}$$

The situation does not change if the domain is part of a multidimensional space like $\mathbb{R}^2$ or $\mathbb{R}^3$. Only the flux into the domain changes a little bit when going from 1D to higher dimensions. If we consider a domain Ω in $\mathbb{R}^2$ or $\mathbb{R}^3$, and an arbitrary part V with a given flux field p on the boundary ∂V (see Fig. 1.2) the amount which goes into the domain through a point on ∂V is exactly $p \cdot n$ where n is the normal vector in that point. Here ∂V denotes the boundary of V.

So the conservation law becomes:

$$\frac{\partial}{\partial t}\int_V u \, dV = -\int_{\partial V} p \cdot n \, dS + \int_V j \, dV \tag{1.28}$$

This equation must also be satisfied on every small subdomain V of Ω. Applying the Gauss-Theorem to the integral over the boundary in Eq. (1.29) gives finally for any subdomain $V \subset \Omega$:

$$\int_V \overbrace{\dot{u} + \operatorname{div} p - j}^{\varphi} \, dV = 0 \tag{1.29}$$

This is again the integral form of the PDE. If the expression under the integral in Eq. (1.29) – th function φ – is continuous, we may again use the fundamental lemma of the calculus of variations (suitably modified for higher dimensions), to arrive at the differential form:

$$\dot{u} + \operatorname{div} p - j = 0 \tag{1.30}$$

If we introduce the characteristic function of the subdomain V which is defined as:

$$\chi_V(x) = \begin{cases} 1 & \text{if} \quad x \in V \\ 0 & \text{otherwise} \end{cases} \tag{1.31}$$

the condition that the conservation is also satisfied on every subdomain can be written as:

$$\int_\Omega \chi_V (\dot{u} + \operatorname{div} p - j) \, dV = 0, \quad \forall \chi_V \tag{1.32}$$

The integral is now over the complete domain Ω. If we take linear combinations of different χ_V, and with certain continuity arguments we may deduce that instead of χ_V in Eq. (1.32) we may take any function ψ such that the integral

$$\int_\Omega \psi (\dot{u} + \operatorname{div} p - j) \, dV = 0, \quad \forall \psi \tag{1.33}$$

is still meaningful. This is the so called *weak form* of the PDE.

1.2 Analytical Solutions of PDEs

Although most Partial Differential Equations have no closed solution on complex domains, it is possible to find solutions for some basic equations on simple domains. They are especially important to verify the accuracy and correctness of numerical methods.

1.2.1 Heat equation

We will start again with the heat equation for the rod from section 1.1. It can be written – without convection – in a simplified form as:

$$\frac{\partial u}{\partial t} - \beta^2 \frac{\partial^2 u}{\partial x^2} = 0 \tag{1.34}$$

Initial and boundary conditions are also required. But these conditions are not necessary for the first steps. The first thing on the way towards a solution is an idea how the function which satisfies the PDE should look like. Here we assume that the solution is the product of two unknown functions $A(x)$ and $B(t)$ – a so called product-ansatz:

$$u(x,t) = A(x) \cdot B(t) \tag{1.35}$$

After that the partial derivatives of u with respect to t and x can be computed:

$$\frac{\partial u}{\partial t} = \dot{u} = A(x) \cdot \dot{B}(t) \tag{1.36}$$

$$\frac{\partial u}{\partial x} = u' = A'(x) \cdot B(t) \tag{1.37}$$

$$\frac{\partial^2 u}{\partial x^2} = u'' = A''(x) \cdot B(t) \tag{1.38}$$

(A dot means the time derivative while the prime denotes the spatial derivate). Inserting these derivatives into the original PDE gives the following result:

$$A(x) \cdot \dot{B}(t) - \beta^2 A''(x) \cdot B(t) = 0 \tag{1.39}$$

$$\text{or} \qquad A(x) \cdot \dot{B}(t) = \beta^2 A''(x) \cdot B(t) \tag{1.40}$$

Obviously the trivial solution $u(x,t) = 0$ satisfies the PDE, but we are not interested in the trivial solution, so we can assume that $u(x,t) = A(x)B(t) \neq 0$ and thus multiply with $\frac{1}{AB}$:

$$\frac{\dot{B}(t)}{B(t)} = \beta^2 \frac{A''(x)}{A(x)} \tag{1.41}$$

This equation can only be satisfied if both sides are constant. So it is possible to introduce a constant κ^2:

$$\frac{\dot{B}(t)}{B(t)} = \beta^2 \frac{A''(x)}{A(x)} = -\kappa^2 \tag{1.42}$$

From this we get the following equation:

$$\dot{B}(t) = -\kappa^2 B(t) \tag{1.43}$$

It is easy to see that the solution of that equation is the exponential function:

$$B(t) = B_0 e^{-\kappa^2 t} \tag{1.44}$$

Applying the same steps to the second part of Eq. (1.42) gives:

$$A''(x) = -\frac{\kappa^2}{\beta^2} A(x) \tag{1.45}$$

with the solutions

$$A(t) = \cos\frac{\kappa}{\beta}x, \quad A(t) = \sin\frac{\kappa}{\beta}x \tag{1.46}$$

Finally going back to the ansatz Eq. (1.35) we get:

$$A(x)B(t) = B_0 e^{-\kappa^2 t} \cos\frac{\kappa}{\beta}x \tag{1.47}$$

$$\text{and} \qquad A(x)B(t) = B_0 e^{-\kappa^2 t} \sin\frac{\kappa}{\beta}x \tag{1.48}$$

as solutions for the heat equation.

1.2.2 Boundary Conditions

If we want to impose the boundary conditions $u(0,t) = 0$ and $u(l,t) = 0$ on the beginning and the end of the rod, the parameters κ and β have to satisfy certain conditions depending on the length of the rod l:

$$A(0)B(t) = B_0 e^{-\kappa^2 t} \sin 0 = 0 \tag{1.49}$$

$$A(L)B(t) = B_0 e^{-\kappa^2 t} \sin\frac{\kappa}{\beta}l = 0 \tag{1.50}$$

Condition (1.49) is always satisfied but Eq. (1.50) leads to the following relation between κ and an arbitrary integer k :

$$\frac{\kappa_k}{\beta}L = k\pi \Rightarrow \kappa_k = \frac{\beta}{L}k\pi \tag{1.51}$$

1.2.3 General Solution

Because the heat equation is a linear PDE the sum of two functions satisfying the PDE is also a solution of the PDE. This leads to the following equation:

$$\theta(x,t) = \sum_{k=1}^{\infty} B_k e^{\kappa_k^2 t} \sin(\frac{\kappa_k}{\beta}x) \tag{1.52}$$

The type of solution is only valid for some special boundary conditions (i.e. $u(0,t) = 0$ and $u(l,t) = 0$). But by also using the cosine functions it is possible to satisfy arbitrary boundary conditions.

The function which defines the initial conditions must be decomposed into sines and cosines by a Fourier analysis to find the parameters B_k for the initial conditions.

Another solution can be obtained by integrating the solution from $-\infty$ to $+\infty$:

$$\int_{-\infty}^{+\infty} e^{-\kappa^2 t} \cos\frac{\kappa}{\beta}x \, d\kappa = \frac{1}{2\beta\sqrt{\pi t}} e^{-\frac{x^2}{4\beta^2 t}} \tag{1.53}$$

This solution is called the *fundamental solution* of the heat equation (cf. also Fig. 1.3). Introducing a coordinate transform gives the following more general form of the fundamental solution:

$$\theta(x,t) = \frac{1}{\sqrt{4\beta^2\pi t}} e^{\frac{-(x-\xi)^2}{4\beta^2 t}} \tag{1.54}$$

Here ξ is the parameter which specifies the distance the function is shifted along the x-axis. Although it might seem that the function disappears slowly the following equation holds:

$$\forall t > 0 \int_{-\infty}^{\infty} \theta(x,t)\, dx = 1 \tag{1.55}$$

As $t \to 0+$ the fundamental solution approaches the so called *Delta Function* denoted by $\delta(x)$, which is not a function in the classical meaning. Looking at the graph of the function (Fig. 1.3) one might guess what it looks like. At an infinitely small part of the X-axis centred around zero the function has an infinite value.

It is only defined in a weak sense. That means only the integral of this function together with another function $v(x) \in C^0(\mathbb{R})$ has a defined value:

$$\int_{-\infty}^{+\infty} \delta(x)v(x)\,dx = v(0) \tag{1.56}$$

$$\text{and} \qquad \int_{-\infty}^{+\infty} \delta(x-\xi)v(x)\,dx = v(\xi) \tag{1.57}$$

Using the following limit:

$$\lim_{t\to 0} \int_{-\infty}^{\infty} \theta(x,t)v(x)\,dx = v(0) \tag{1.58}$$

shows that $\theta(x,0)$ must be the Delta Function.

1.2.4 Solutions with Source Terms and Initial Conditions

Using the property that $\theta(x,0)$ is the Delta Function and the linearity of the Laplace operator allows the construction of analytical solutions which satisfy arbitrary initial conditions or functions generating energy or heat.

Without more explanations the following equations come out:

a.) and internal sources $h(t,x)$

$$\hat{\theta}(t,x) = \frac{x}{\sqrt{4\beta^2\pi}} \int_0^t h(t-\tau)\tau^{3/2} \exp(-\frac{x^2}{4\beta^2\tau})d\tau \tag{1.59}$$

b.) initial conditions $\theta(0,x) = f(x)$ and no internal sources.

$$\tilde{\theta}(t,x) = \frac{1}{\sqrt{4\beta^2\pi t}} \int_{-\infty}^{+\infty} f(\xi) \exp(-\frac{(x-\xi)^2}{4\beta^2 t})d\xi \tag{1.60}$$

1.3 Non-Dimensional Form of the Heat Equation

In this section the behaviour of the PDE for different scales should be examined. One example may be the diffusion of some chemical substances in the sea which a large ship looses through a leakage, another may be one drop of milk in a cup of coffee. First step in this examination is the introduction of a coordinate transformation, to make all quantities in the equation non-dimensional

$$\theta = \vartheta(x, y, z, t)\bar{\theta} \tag{1.61}$$

with

$$\begin{aligned} x &= \xi L & (1.62)\\ y &= \eta L & (1.63)\\ z &= \zeta L & (1.64)\\ t &= \tau \cdot T & (1.65) \end{aligned}$$

where L is a reference length and T a reference time.

This time we will consider the heat equation together with convective transport:

$$\dot{\theta} - \beta^2 \Delta\theta + v^T \nabla\theta = 0 \tag{1.66}$$

Here v is the velocity of the convective transport. Perhaps the gulf stream or stirring the cup of coffee. Now the partial derivatives in Eq. (1.66) must be replaced by the derivatives with respect to the new variables ξ, η, ζ and τ.

$$\begin{aligned} \frac{\partial}{\partial x} &= \frac{1}{L}\frac{\partial}{\partial \xi} \Rightarrow \frac{\partial^2}{\partial x^2} = \frac{1}{L^2}\frac{\partial^2}{\partial \xi^2} & (1.67)\\ \frac{\partial}{\partial y} &= \frac{1}{L}\frac{\partial}{\partial \eta} \Rightarrow \frac{\partial^2}{\partial x^2} = \frac{1}{L^2}\frac{\partial^2}{\partial \eta^2} & (1.68)\\ \frac{\partial}{\partial z} &= \frac{1}{L}\frac{\partial}{\partial \zeta} \Rightarrow \frac{\partial^2}{\partial x^2} = \frac{1}{L^2}\frac{\partial^2}{\partial \zeta^2} & (1.69)\\ \frac{\partial}{\partial t} &= \frac{1}{T} \cdot \frac{\partial}{\partial \tau} & (1.70) \end{aligned}$$

And the velocity of the convective flow must obviously also be adapted to the new scales:

$$v = \upsilon \frac{L}{T} \tag{1.71}$$

With these equations the gradient and the Laplacian become:

$$\nabla_\xi = \left(\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta}, \frac{\partial}{\partial \zeta}\right)^T \quad \text{and} \quad \Delta_\xi = \left(\frac{\partial^2}{\partial \xi^2} + \frac{\partial^2}{\partial \eta^2} + \frac{\partial^2}{\partial \zeta^2}\right) \tag{1.72}$$

and the heat equation thus:

$$\frac{1}{T}\frac{\partial}{\partial\tau}\vartheta\bar{\theta} - \frac{\beta^2\bar{\theta}}{L}\Delta_\xi\vartheta + \frac{\bar{\theta}}{T}v^T\cdot\nabla_\xi\vartheta = 0 \tag{1.73}$$

Multiplying with T and dividing by $\bar{\theta}$ gives:

$$\frac{\partial}{\partial\tau}\vartheta - \frac{1}{Pe}\Delta_\xi\vartheta + v^T\cdot\nabla_\xi\vartheta = 0 \tag{1.74}$$

Where $Pe = \frac{L}{\beta^2 T}$. In this equation the reference time and length totally disappeared except for the factor $1/Pe$ in front of the Laplacian. As $\beta^2 = \frac{\lambda}{c\rho}$, we have $Pe = \frac{c\rho\cdot L}{\lambda\cdot T}$. It is a non-dimensional number like in many other areas (Reynolds number, Mach number, ...). All scales of the actual configuration go into that number. So physical phenomena on domains with totally different sizes and different materials can have the same behaviour if their Peclet number is the same.

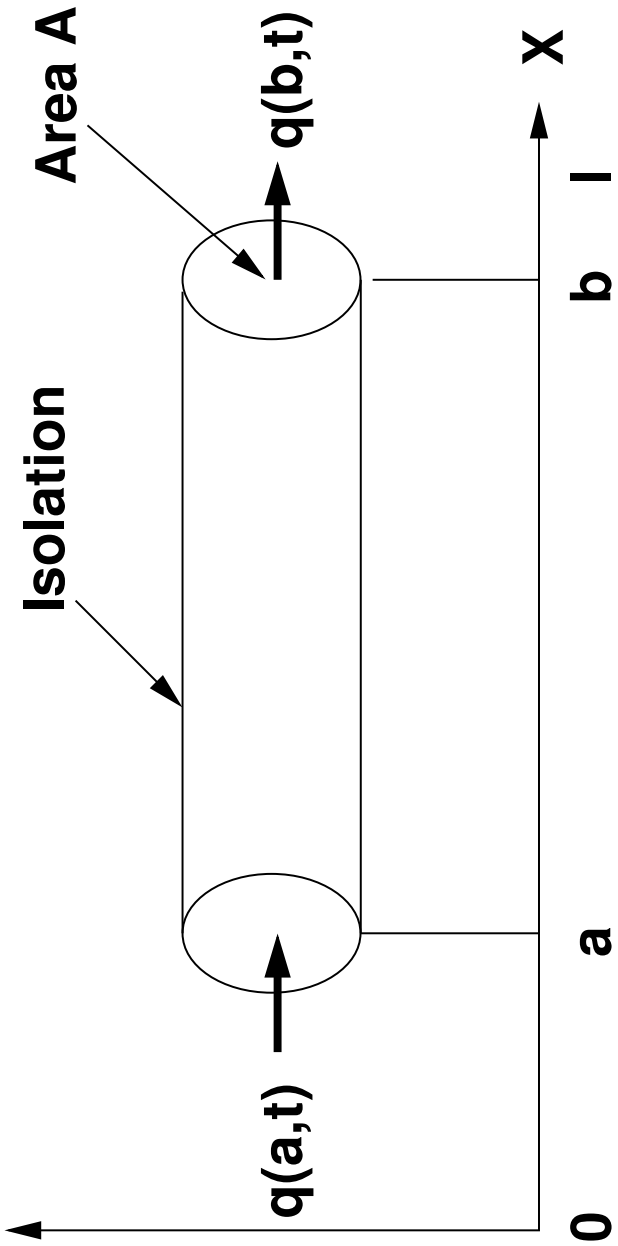


Figure 1.1: Insulated rod

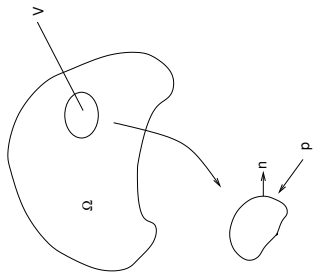


Figure 1.2: The domain Ω and a part V

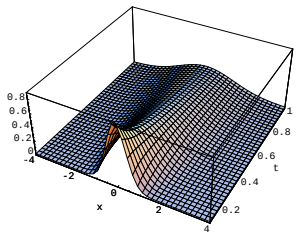


Figure 1.3: Fundamental solution of the heat equation

1.4 Finite Difference methods

One result of the last section was the PDE which describes the heat transfer in a insulated rod. Furthermore, several analytical solutions of this PDE were presented. But these solutions satisfied very special initial and boundary conditions. If we want to solve real problems with arbitrary boundary and initial conditions, it will almost be impossible to find analytical solutions.

Thus this section will show one possible way to find a numerical approximation for the solution of the PDE. Next the properties of this approximation will be compared with the properties of the analytical solution. At the end some other schemes will be introduced and analysed.

1.4.1 Spatial approximation of the heat equation

If we consider again the heat equation:

$$\begin{aligned} \frac{\partial u}{\partial t} - \beta^2 \Delta u &= f, && (1.75)\\ \forall x \quad u(x,0) &= \tilde{u}_0(x) \quad \text{given}, && (1.76)\\ \forall t > 0 \quad u(0,t) &= \hat{u}_0(t), && (1.77)\\ u(l,t) &= 0. && (1.78) \end{aligned}$$

we see two partial derivatives. One with respect to time and the other with respect to spatial variables. Although some newer methods (Time-Space Finite Elements) treat the time derivatives in the same way as the spatial derivatives, most classical approaches separate the time and space directions and start with a numerical approximation of the space derivative.

Because the real solution $u(x,t)$ of the PDE is defined on infinitely many points inside the domain, it is impossible to handle the complete function inside the computer. So we must limit our solution to a finite number of points in space. For simplicity we assume these points are distributed equidistant on the domain. So each point has a distance of h to its left and right neighbour.

The goal of the approximation is to find an expression for $\frac{\partial^2 u}{\partial x^2}$, which depends only on some neighbour points. One way to derive this expression is a Taylor expansion of u around a given point x. The first approximation is used for the right neighbour:

$$u(x+h) = u(x) + \frac{\partial u}{\partial x}(x)h + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}h^2 + \frac{1}{3!}\frac{\partial^3 u}{\partial x^3}h^3 + \frac{1}{4!}\frac{\partial^4 u}{\partial x^4}h^4 + O(h^5), \tag{1.79}$$

the second one for the left neighbour of point x:

$$u(x-h) = u(x) - \frac{\partial u}{\partial x}(x)h + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}h^2 - \frac{1}{3!}\frac{\partial^3 u}{\partial x^3}h^3 + \frac{1}{4!}\frac{\partial^4 u}{\partial x^4}h^4 - O(h^5). \tag{1.80}$$

Adding Eq. (1.79) and Eq. (1.80) results in:

$$u(x+h) + u(x-h) = 2u(x) + 0 + \frac{\partial^2 u}{\partial x^2}h^2 + 0 + \frac{2}{4!}\frac{\partial^4 u}{\partial x^4}h^4 + O(h^6). \tag{1.81}$$

Dividing by h^2 and rearranging gives:

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{h^2}(u(x+h) - 2u(x) + u(x-h)) - \frac{1}{12}\frac{\partial^4 u}{\partial x^4}h^2 + O(h^4). \tag{1.82}$$

As we only want to use the values at the points $x-h, x, x+h$, we may shorten this to

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{h^2}(u(x+h) - 2u(x) + u(x-h)) + O(h^2). \tag{1.83}$$

Because we have a finite number of equidistant points it is possible to label these points from 0 to N, where $h \cdot N = l$. At a typical point $x_j = x_0 + j \cdot h$ we introduce the notation

$$u_j := u(x_j) \tag{1.84}$$

$$\frac{\partial u_j}{\partial x} := \frac{\partial u_(x_j)}{\partial x}, \quad \text{etc.} \tag{1.85}$$

Introducing this numbering gives for an arbitrary point x_j:

$$\frac{\partial^2 u_j}{\partial x^2} = \frac{1}{h^2}(u_{j+1} - 2u_j + u_{j-1}) + O(h^2) \tag{1.86}$$

This equation provides already an error estimate. Reducing the distance between two points to one half of the original distance reduces the error to roughly one quarter of the previous value.

Another way to derive this equation is to use the well known relation that the second derivative of a function is the derivative of the first derivative of this function. The same applies to the differences. Here we take the difference between the first forward difference and the first backward difference.

$$\frac{1}{h}\left(\frac{u_{j+1} - u_j}{h} - \frac{u_j - u_{j-1}}{h}\right) = \frac{1}{h^2}(u_{j+1} - 2u_j + u_{j-1}) \tag{1.87}$$

1.4.2 Method of Lines / Semi-Discrete Approximation

By inserting the approximation for the second derivative in Eq. (1.75) we obtain approximately:

$$\frac{\partial u_j}{\partial t} - \frac{\beta^2}{h^2}(u_{j-1} - 2u_j + u_{j+1}) = f_j(t), \quad j \in [1..N-1] \tag{1.88}$$

The PDE has now become a system of ODEs. Introducing the vector $\mathbf{u}$

$$\mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_j(t) \\ \vdots \\ u_{N-1}(t) \end{bmatrix} \tag{1.89}$$

allows us to write the system of ODEs in matrix form:

$$\frac{d}{dt}\mathbf{u}(t) = \mathbf{A}\mathbf{u}(t) + \mathbf{f}(t) \tag{1.90}$$

with

$$\mathbf{A} = -\frac{\beta^2}{\Delta x^2}\begin{bmatrix} 2 & -1 & 0 & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ 0 & & & -1 & 2 \end{bmatrix} \quad \text{and} \quad \mathbf{f}(t) = \begin{bmatrix} f_1(t) + \frac{\beta^2}{\Delta x^2}\hat{u}_0(t) \\ f_2(t) \\ \vdots \\ f_{N-1}(t) \end{bmatrix} \tag{1.91}$$

One problem occurs at the boundarys which lie at the points u_0 and u_N. Here we have circumvented it by assuming the simple boundary conditions in Eq. (1.75), where the first (inhomogeneous one) at x_0 gives a contribution to the vector $\mathbf{f}$. Other boundary conditions will be treated later.

The name *Method of Lines* comes from the fact that we have reduced the original problem of finding a solution $u(x,t)$ at an infinite number of points in the space-time domain to the problem of finding solutions $u_j(t)$ on a finite number of lines in the space-time domain (cf. Fig. 1.4). These solutions can be obtained by solving the system of ODEs analytically or by using another numerical method to discretise these ODEs as well in time.

1.4.3 Analysis of the Spatial Discretisation

In this section a general analytical solution for the system of ODEs which came from the spatial discretisation will be derived. For simplicity we consider the heat equation with boundary conditions as in Eq. (1.75), with $f \equiv 0$ and $\hat{u}_0 \equiv 0$.

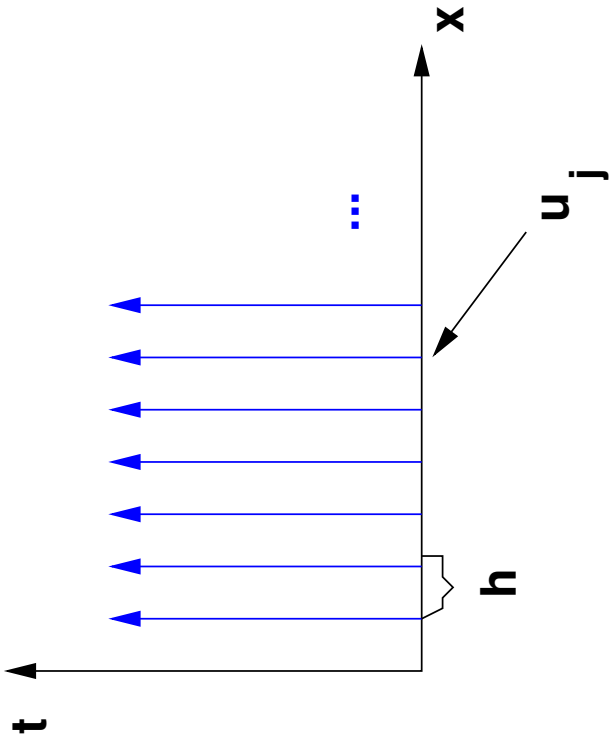


Figure 1.4: Scheme of the Method of Lines

The spatially discrete system Eq. (1.90) from the method of lines then simply reads

$$\dot{\mathbf{u}} = \mathbf{A}\mathbf{u} \tag{1.92}$$

where the matrix A in Eq. (1.91) is symmetric $A = A^T$ and thus has the following properties:

- A has $N-1$ orthogonal eigenvectors which form a basis of $\mathbb{R}^{N-1}$
- A has real eigenvalues

For our analysis we need an analytical solution for Eq. (1.92). We start with the following Ansatz:

$$\mathbf{u}(t) = \mathbf{v} \cdot e^{\alpha t} \tag{1.93}$$

where α is a number and $\mathbf{v}$ a vector. Inserting Eq. (1.93) into Eq. (1.92) gives:

$$\alpha \mathbf{v} e^{\alpha t} = e^{\alpha t}\mathbf{A}\mathbf{v} \Rightarrow \mathbf{A}\mathbf{v} = \alpha \mathbf{v} \tag{1.94}$$

and hence $\mathbf{v}$ and α have to be eigenvector and eigenvalue of $\mathbf{A}$ in order that Eq. (1.93) is a solution of Eq. (1.92). One problem with this solution is that it does not satisfy the initial conditions $u(x,0) = \tilde{u}_0(x)$.

It is possible to overcome this problem because the eigenvectors of $\mathbf{A}$ provide an orthogonal basis. Every vector of initial conditions can then be build up from the eigenvectors:

$$\mathbf{u}(0) = \begin{bmatrix} u_1(0) \\ \vdots \\ u_{N-1}(0) \end{bmatrix} = \sum_{j=1}^{N-1} \beta_j^0 \mathbf{v}_j \tag{1.95}$$

The solution vector at an arbitrary time is decomposed in the same way:

$$\mathbf{u}(t) = \begin{bmatrix} u_1(0) \\ \vdots \\ u_{N-1}(t) \end{bmatrix} = \sum_{j=1}^{N-1} \beta_j(t) \mathbf{v}_j \tag{1.96}$$

Obviously this solution must satisfy the system of ODEs which gives the following relation:

$$\sum_{j=1}^{N-1} \dot{\beta}_j(t) \mathbf{v}_j = \mathbf{A} \left(\sum_{j=1}^{N-1} \beta_j(t) \mathbf{v}_j \right) = \sum_{j=1}^{N-1} \beta_j(t) \mathbf{A} \mathbf{v}_j = \sum_{j=1}^{N-1} \beta_j(t) \lambda_j \mathbf{v}_j \tag{1.97}$$

This leads to the following condition for the variables β_j:

$$\sum_{j=1}^{N-1} (\dot{\beta}_j(t) - \beta_j(t) \lambda_j) \mathbf{v}_j = 0 \tag{1.98}$$

As $\{\mathbf{v}_j\}$ is a basis, this is only possible if the parenthesised term vanishes for each j.

With this basis transformation it is possible to split the original system of coupled ODEs into a set of uncoupled linear ODEs:

$$\dot{\beta}_j(t) = \lambda_j \beta_j(t), \quad \beta_j(0) = \beta_j^0 \tag{1.99}$$

with the analytical solutions:

$$\beta_j(t) = \beta_j^0 e^{\lambda_j t} \tag{1.100}$$

After this preparation we have everything together to analyse the behaviour of the analytical solution of the system of ODEs which we obtained from the spatial discretisation of the heat equation. One very important thing about the solutions of the heat equation was the fact that all solutions were decaying if no internal heat sources were present. If

our spatial discretisation can not guarantee that these properties remain in the solutions of the ODEs it will be not very useful, because the goal of our work is to get a method which can be used to compute reliable predictions.

From Eq. (1.100) it can be seen that the eigenvalues λ_j of the matrix $\mathbf{A}$ are essential for the solutions. If $\lambda_j > 0$ it is clear that the exponent will grow as time increases and thus the solution will also grow. So a decaying solution requires that all λ_j are smaller than zero. To find out if this is true for our matrix $\mathbf{A}$ we need a general eigenvalue analysis of the matrix $\mathbf{A}$. Fortunately a closed formula exists for the eigenvalues of a tridiagonal symmetric matrix.

Lemma 2 (Eigenvalues of a tridiagonal matrix) *Let* $\mathbf{A}$ *be a symmetric tridiagonal matrix of size* $N-1 \times N-1$ *with the following structure:*

$$\mathbf{A} = \begin{bmatrix} a & b & & & \\ b & a & b & & \\ & \ddots & \ddots & \ddots & \\ & & b & a & b \\ & & & b & a \end{bmatrix}$$

Then the eigenvalues λ_j *of* $\mathbf{A}$ *are:*

$$\lambda_j = a + 2b\cos\left(\frac{j\pi}{N}\right), \quad j = [1 \dots N-1]$$

and the eigenvectors $\mathbf{v}_j$ *of* $\mathbf{A}$ *are:*

$$\mathbf{v}_j = \begin{bmatrix} v_j^1 \\ \vdots \\ v_j^k \end{bmatrix}, \qquad v_j^k = \sin\left(\frac{kj\pi}{N}\right), \qquad k, j = [1 \dots N-1]$$

In our case we have:

$$a = -\frac{2\beta^2}{h^2}, \quad b = \frac{\beta^2}{h^2} \tag{1.101}$$

So we obtain:

$$\lambda_j = -\frac{2\beta^2}{h^2} + \frac{2\beta^2}{h^2}\cos\frac{j\pi}{N} = \frac{2\beta^2}{h^2}\left(\left(\cos\frac{j\pi}{N}\right) - 1\right) \tag{1.102}$$

The first part of Eq. (1.102) is just a positive constant. So whether the largest eigenvalue is greater than zero is determined by the last part, which can only become zero if the cosine becomes one. Because the expression j/N never becomes zero the cosine never reaches 1

and the eigenvalues λ_j are always negative. This shows that the analytical solutions of the ODEs will always decay and thus reproduce qualitatively the original behaviour of the PDE.

1.4.4 Time Discretisation

Although we have found an analytical solution for the system of ODEs coming from the spatial discretisation this task will become more difficult and most often impossible if we consider more complex domains. Therefore we need another discretisation which approximates the time derivative and allows us to solve the ODEs numerically (See Fig. 1.5)

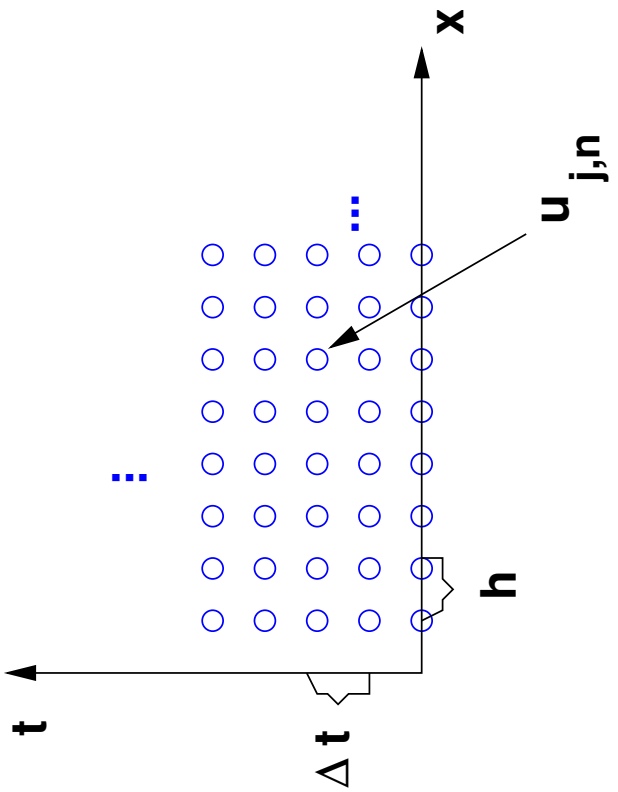


Figure 1.5: Scheme of a full discretisation

Forward Differences

To approximate the time derivative we use again a Taylor series expansion of u around a given time t. Let Δt denote the time step size, then we have:

$$u(t + \Delta t) = u(t) + \left.\frac{\partial u}{\partial t}\right|_t \Delta t + O(\Delta t^2), \tag{1.103}$$

or

$$\left.\frac{\partial u}{\partial t}\right|_t = \frac{u(t+\Delta t) - u(t)}{\Delta t} + O(\Delta t). \tag{1.104}$$

If we insert this approximation of the time derivative into the spatially discretised heat equation, we obtain:

$$\frac{\mathbf{u}(t+\Delta t) - \mathbf{u}(t)}{\Delta t} = \mathbf{A}\mathbf{u}(t) \tag{1.105}$$

The two approximation errors of size $O(\Delta t)$ for the time discretisation and $O(h^2)$ for the space discretisation bring a total discretisation error of $O(\Delta t) + O(h^2) = O(\Delta t + h^2)$.

Assuming that the size of the time steps stays constant it is possible to number the different discrete time points:

$$t_n = t_0 + n \cdot \Delta t \tag{1.106}$$

Together with the spatial discretisation we have a solution vector at every time point:

$$\mathbf{u}_n = \begin{bmatrix} u_1(t_n) \\ \vdots \\ u_j(t_n) \end{bmatrix} = \begin{bmatrix} u_{1,n} \\ \vdots \\ u_{j,n} \end{bmatrix} \tag{1.107}$$

With these vectors the discrete heat equation can be written as:

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t \mathbf{A}\mathbf{u}_n = \underbrace{(\mathbf{I} + \Delta t \mathbf{A})}_{\mathbf{B}}\mathbf{u}_n \tag{1.108}$$

This method for ODEs is also known as the *Euler forward method*. It is now a fully discrete linear dynamical system of difference equations with matrix $\mathbf{B}$.

An important question is now whether the numerical solutions of this difference equation also decay. To find an answer another eigenvalue analysis with the matrix $\mathbf{B}$ is necessary. Again the matrix is tridiagonal which makes the eigenvalue analysis easy.

$$\mathbf{B} = \begin{bmatrix} 1-2\beta^2\frac{\Delta t}{h^2} & \beta^2\frac{\Delta t}{h^2} & 0 & 0 & \dots \\ \beta^2\frac{\Delta t}{h^2} & 1-2\beta^2\frac{\Delta t}{h^2} & \beta^2\frac{\Delta t}{h^2} & 0 & \dots \\ 0 & \beta^2\frac{\Delta t}{h^2} & 1-2\beta^2\frac{\Delta t}{h^2} & \beta^2\frac{\Delta t}{h^2} \quad 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \end{bmatrix} \tag{1.109}$$

Here $a = 1 - 2r$ and $b = r$ with $r = \beta^2\frac{\Delta t}{h^2}$ and thus:

$$\lambda_j = 1 - 2r + 2r\cos\frac{j\pi}{N} = 1 - 2r\left(1 - \cos\frac{j\pi}{N}\right) \tag{1.110}$$

The solution of linear difference equations is growing if the absolute value of one eigenvalue is greater than one. Therefore we must look if one of the eigenvalues is greater than one or less than one. During the analysis of the spatial approximation we already saw that $\cos\frac{j\pi}{N}$ never becomes zero. From this fact we see that Eq. (1.110) is always less then one. The other "dangerous" value is -1. If we set $j = N - 1$ the cosine approaches its maximum negative value:

$$\lambda_{N-1} = 1 - 2r\left(1 - \cos\frac{(N-1)\pi}{N}\right) \tag{1.111}$$

To guarantee decreasing solutions we can make the condition a little bit stronger by requiring:

$$\lambda_{N-1} > \lambda_N = 1 - 4r > -1, \quad \text{or} \quad r < \frac{1}{2}, \tag{1.112}$$

which gives the following relation for β, h and Δt:

$$\Delta t < \frac{h^2}{2\beta^2} \tag{1.113}$$

Satisfying this relation guarantees a stable behaviour with decaying solutions. One interesting thing about this equation is the fact that the time step size depends on the spatial discretisation. So reducing the distance between the points in space requires a reduction of the time step, but with a quadratic dependence !. If we want the solution to be four times as accurate, we have to double the number of spatial points ($O(h^2)$), and divide the time step by 4, both for accuracy ($O(\Delta t)$) and stability (Eq. (1.113)) reasons.

θ - Methods

To overcome the restrictions of the forward differences in time, other time discretisation schemes must be used. One idea is to use not only the forward difference but to take also the backward difference.

The forward difference is defined as:

$$\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{t=t_n} = \frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\Delta t} + O(\Delta t) \tag{1.114}$$

This difference leads, as we already know, to the *Euler forward* method for ODEs. Inserting this finite difference approximation into the original system of ODEs results in:

$$\frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\Delta t} = \mathbf{A}\mathbf{u}_n \tag{1.115}$$

The backward difference is:

$$\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{t=t_{n+1}} = \frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\Delta t} + O(\Delta t) \tag{1.116}$$

This leads to the *Euler backward* method for ODEs. We insert this approximation into the original system of ODEs to obtain:

$$\frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\Delta t} = Au_{,n+1} \tag{1.117}$$

The class of *θ-methods* is based on a linear combination of the forward and backward difference formulas. Introducing a weighting parameter θ we get:

$$\theta\text{backw.} + (1-\theta)\text{forw.} \approx \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{t=t_{n+\theta}} + O(\Delta t^p). \tag{1.118}$$

For $\theta = 1/2$ the order of the method is $p = 2$. All other methods achieve only an order of $p = 1$. Inserting Eq. (1.115) and Eq. (1.117) into Eq. (1.118) gives:

$$\frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\Delta t} = \theta\mathbf{A}\mathbf{u}_{n+1} + (1-\theta)\mathbf{A}\mathbf{u}_n \tag{1.119}$$

By solving for $\mathbf{u}_{,n+1}$ we obtain:

$$(\mathbf{I} - \theta\Delta t\mathbf{A})\mathbf{u}_{n+1} = (\mathbf{I} + (1-\theta)\Delta t\mathbf{A})\mathbf{u}_n \tag{1.120}$$

In this equation we can observe several properties of the θ-method. Non astonishingly for $\theta = 0$ it is exactly the same as the *Euler forward* method. Furthermore we can see that the system of linear equations which must be solved to get the next solution vector $\mathbf{u}_{n+1}$ is non-trivial for all $\theta > 0$. Hence larger timesteps through better stability properties of the method have to be bought at the expense of more floating point operations per time step.

To see if we may use larger time steps with the θ-methods we need the same type of analysis as for the finite difference method.

As both $(\mathbf{I} - \theta\Delta t\mathbf{A}) = \mathbf{B}_1$ and $(\mathbf{I} + (1-\theta)\Delta t\mathbf{A}) = \mathbf{B}_2$ are tridiagonal and symmetric, they have the same eigenvectors and may be diagonalised simultaneously, with

$$\lambda_i(\mathbf{B}_1) = 1 + 2r\theta - 2r\theta\cos\frac{j\pi}{N} = 1 + 2r\theta(1 - \cos\frac{j\pi}{N}) \tag{1.121}$$

and

$$\lambda_i(\mathbf{B}_2) = 1 - 2r(1-\theta)(1-\cos\frac{j\pi}{N}). \tag{1.122}$$

The system in Eq. (1.120) can be written as

$$\mathbf{u}_{n+1} = \mathbf{B}_1^{-1}\mathbf{B}_2\mathbf{u}_n = \mathbf{B}\mathbf{u}_n \tag{1.123}$$

and hence $\mathbf{B}$ has eigenvalues

$$\lambda_j(\mathbf{B}) = \frac{\lambda_j(\mathbf{B}_2)}{\lambda_j(\mathbf{B}_1)} = \frac{1 - 2r(1-\theta)(1-\cos\frac{j\pi}{N})}{1 + 2r\theta(1-\cos\frac{j\pi}{N})} \tag{1.124}$$

(and the same eigenvectors as $\mathbf{B}_1$ and $\mathbf{B}_2$). We require that

$$-1 < \lambda_j(\mathbf{B}) < 1. \tag{1.125}$$

The right inequality leads to $1 - 2r(1-\cos\frac{j\pi}{N}) < 1$ which is satisfied for all j, and the left inequality gives the requirement

$$r(1-\cos\frac{j\pi}{N})(1-2\theta) < 1. \tag{1.126}$$

This is certainly satisfied if $\theta \geq 1/2$, and hence those θ-methods are stable for any combination of Δt and h; this is called *unconditionally stable*. For $\theta < 1/2$ the inequality is certainly satisfied if $r \cdot 2 \cdot (1-2\theta) < 1$, or $r < \frac{1}{2(1-2\theta)}$. For $\theta = 0$ this is relation Eq. (1.112).

1.4.5 Von Neumann Stability Analysis

Some error estimates were obtained by the Taylor series expansion of the PDE in time and space (Eq. (1.118)). These error estimates showed the *consistency* of the numerical approximation which, means that the numerical solution is an approximation to the solution of the PDE.

But consistency is not enough to get correct solutions for the PDE. Another requirement is the *stability* of the numerical solution. The condition for stability Eq. (1.113) was derived by the matrix stability analysis. Stability and consistency guarantee together that the numerical solution *converges* to the real solution of the PDE.

In this section another method to find the stability conditions for a method will be presented. This method starts with an assumption about the analytical solutions. These solutions consist of sine and cosine functions of different frequencies at each time instance:

$$u(x) = \cos(k \cdot x) + i \sin(k \cdot x) = e^{ikx} \tag{1.127}$$

Here i is the imaginary unit and k is the *wavenumber*. For this analysis we also assume that the number of discrete points is infinite. Then looking at this function at our discrete grid points where $x = j \cdot h$ reveals:

$$u(j) = e^{ikjh} \tag{1.128}$$

Currently this Ansatz captures only the spatial structure of the solution. From the analytical solution we know that the time evolution of the function is an exponential function. In the discrete case this exponential function is approximated by the gain factor, $G(k)^n$ where:

$$G(k) = e^{\alpha(k)} \tag{1.129}$$

Bringing Eq. (1.128) and Eq. (1.129) together gives the following ansatz function for the solution in one of the discrete points:

$$u_{n,j} = G(k)^n e^{ikjh} \tag{1.130}$$

Using again $r = \frac{\beta^2 \Delta t}{h^2}$, the general form of the Theta-methods can be written as:

$$-\theta r u_{n+1,j-1} + (1+2\theta r) u_{n+1,j} - \theta r u_{n+1,j+1} = (1-\theta) r u_{n,j-1} + (1-2(1-\theta)r) u_{n,j} + (1-\theta) r u_{n,j+1} \tag{1.131}$$

Inserting the Ansatz Eq. (1.130) into the difference formula gives:

$$\begin{aligned}(1+2\theta r)G(k)^{n+1}e^{ikjh} - \theta r(G(k)^{n+1}e^{ik(j+1)h} + G(k)^{n+1}e^{ik(j-1)h}) \\ =(1-2(1-\theta)r)G(k)^n e^{ikjh} + (1-\theta)r(G(k)^n e^{ik(j+1)h} + G(k)^n e^{ik(j-1)h})\end{aligned} \tag{1.132}$$

Dividing by $G(k)^n e^{ikjh}$, which is nonzero, simplifies the equation to:

$$(1+2\theta r)G(k) - \theta r G(k)(e^{ikh} + e^{-ikh}) = (1-2(1-\theta)r) + (1-\theta)r(e^{ikh} + e^{-ikh}) \tag{1.133}$$

From $e^{i\xi} = \cos\xi + i\sin\xi$ it is easy to derive the following two formulae:

$$\cos\xi = \frac{1}{2}\left(e^{i\xi} + e^{-i\xi}\right) \tag{1.134}$$

$$\sin\xi = \frac{1}{2i}\left(e^{i\xi} - e^{-i\xi}\right) \tag{1.135}$$

Using the first of these gives:

$$(1+2\theta r - 2\theta r\cos(kh))G(k) = 1 - 2(1-\theta)r + 2(1-\theta)r\cos(kh) \tag{1.136}$$

Solving for $G(k)$, we finally arrive at the following expression for the gain factor:

$$G(k) = \frac{1-2(1-\theta)r(1-\cos(kh))}{1+2\theta r(1-\cos(kh))} \tag{1.137}$$

Obviously the gain factor depends on the wave number and the spatial discretisation. For stability the following condition must be satisfied:

$$|G(k)| \leq 1 \tag{1.138}$$

Another important component in the stability analysis is the highest wavenumber k which will be included in our examination. This wavenumber is naturally given by the spatial discretisation with alternating values at successive grid points. This means the upper limit is $k_{max} = \frac{\pi}{h}$. Higher frequencies appear as lower frequencies. This effect is known as *aliasing* and follows directly from Shannon's theorem about the discretisation of signals.

The extreme values of G which are important for the stability analysis depend mainly on the cosine in the quotient of Eq. (1.137). Demanding $\cos(kh) = 1$ leads to $k = 0$ which is the lowest possible frequency and thus:

$$G(0) = \frac{1-0}{1+0} = 1 \tag{1.139}$$

This extreme value does not cause any trouble (it is actually necessary for consistency) because it only reaches the stability limit. Now we have to examine the other extreme value $\cos(kh) = -1$, $kh = \pi \Rightarrow k = \frac{\pi}{h}$:

$$G(\frac{\pi}{k}) = \frac{1 - 4(1-\theta)r}{1 + 4\theta r} \tag{1.140}$$

While the first limit exactly measures the amplification of the lowest frequency, the lower limit corresponds to the amplification of the highest frequencies which can be resolved with the given spatial discretisation. And the second limit can become less than -1 and is thus the "dangerous" limit which needs further investigation:

$$\frac{1 - 4(1-\theta)r}{1 + 4\theta r} \geq -1 \Rightarrow (1 - 2\theta)r \leq \frac{1}{2} \tag{1.141}$$

For $\theta \geq \frac{1}{2}$ we get an *unconditionally stable* method for all $r > 0$. If $\theta < \frac{1}{2}$ a restriction on the time step must be imposed to get a stable method ($r < 1/2(1-2\theta)$). Comparing this stability result with the matrix stability analysis for the Euler method ($\theta = 0$) shows that we get the same restriction on r.

If $G < 0$ the first factor G^n of the discrete solution will change its sign with every time step. These solutions are called *oscillatory solutions*. Because the analytical solution does not show this behaviour it would be nice to avoid also this unwanted characteristic. Inserting this requirement into the equation for the gain factor reveals:

$$\frac{1 - 4(1-\theta)r}{1 + 4\theta r} \geq 0 \Rightarrow r \leq \frac{1}{4(1-\theta)} \tag{1.142}$$

A last conditions can be derived from the numerical schemes. It is called *positivity* and should prevent the solution from becoming negative. Looking at Fig. 1.6 shows how the solution at a given point depends on the neighbour points:

$$\begin{aligned} u_{n+1,j} &= u_{n,j} + (1-\theta)r(u_{n,j-1} - 2u_{n,j} + u_{n,j+1}) \\ &= \underbrace{(1 - 2(1-\theta)r)}_{a} u_{n,j} + (1-\theta)r(u_{n,j-1} + u_{n,j+1}) \end{aligned} \tag{1.143}$$

The important criteria for positivity is the part a in Eq. (1.143) because the rest of the equation is always positive, if the algorithm is started with positive initial conditions. It follows that:

$$(1 - 2(1-\theta)r) \geq 0 \Rightarrow r \leq \frac{1}{2(1-\theta)} \tag{1.144}$$

In summary we have found the following three criteria which can be used to find the right parameters for the numerical solution:

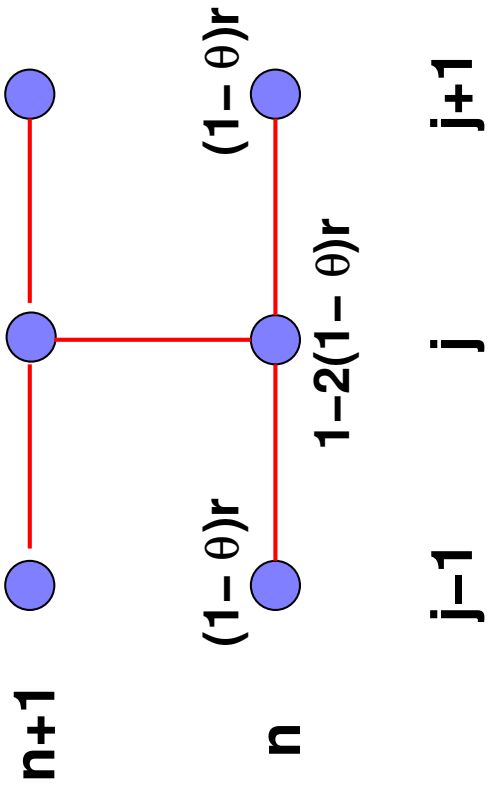


Figure 1.6: Computational molecule or difference star for the theta methods

- Stability : $r \leq \frac{1}{2(1-2\theta)}$
- Positivity : $r \leq \frac{1}{2(1-\theta)}$
- No oscillations : $r \leq \frac{1}{4(1-\theta)}$

For the three schemes which are used most the result are shown in Table 1.1

	Euler fwd.	Trap.Rule/Crank Nicholson	Euler bwd.
Stability	$r \leq 1/2$	$r \leq \infty$	$r \leq \infty$
Positivity	$r \leq 1/2$	$r \leq 1$	$r \leq \infty$
No oscill.	$r \leq 1/4$	$r \leq 1/2$	$r \leq \infty$

Table 1.1: Limits for $\theta = 0, \theta = 1/2, \theta = 1$

1.4.6 Stability and Consistency

In the previous section we analysed the stability of the Theta-methods, which were fortunately consistent. Otherwise the methods could have been stable and nevertheless been

producing wrong results. The meaning of *consistency*, *stability* and *convergence* will be illustrated in the next chapter with some examples which show the need for these criteria.

Well posedness

A very useful demand on PDEs is the *well posedness*. Following the definition of Hadamard a PDE

$$L(u) = f \tag{1.145}$$

is well posed if it possesses three properties:

- the solution exists
- the solution is unique
- the solution depends continuously on auxiliary data

To show the existence of a solution may be a difficult problem, but usually depends on the proper formulation of the problem. It requires the operator L is surjective, i.e. for any f there is at least one u satisfying Eq. (1.145). For the uniqueness of the solution the operator L must be injective, i.e. there is at most one u satisfying Eq. (1.145). The last requirement can be satisfied if L and also L^{-1} are continuous.

Although well posed problems are very nice, not all physical phenomena can be described by a well posed PDE. A simple example is an elastic rod with one fixed end and an increasing force acting in the direction of the rod on the other end. For small forces the problem is well posed. The deformation of the rod follows simply Hooke's law. But at a certain point, when the rod starts buckling, the problem is no longer well posed because the rod can buckle to an arbitrary direction. So infinitely many solutions which are all physically correct can exist.

Convergence

The most important criterium for a numerical approximation is the *convergence* which demands that the approximate solution gets closer to the exact solution as the discretisation is made finer.

Let $L(u) = f$ define the exact solution and $L_h(u_h) = f_h$ be the discrete approximation. Then convergence is:

$$u_h \to u, \quad \text{as} \quad (h \to 0) \tag{1.146}$$

With this definition one open question remains. How to measure if a function approaches another function. For this purpose the concept of norms, which is known from finite dimensional spaces, is transferred to function spaces. A first basic norm is the L_2 norm which is defined by:

$$||u||_{L_2} = \sqrt{\int u(x)^2 dx} \tag{1.147}$$

Utilising an arbitrary norm the convergence can be written as:

$$||u_h - u|| \to 0, \quad \text{as} \quad (h \to 0) \tag{1.148}$$

A weaker criterium than the convergence is the *consistency*, which requires that the discrete system approaches the continuous one as $h \to 0$ (with fixed u) !

$$\begin{matrix} L_h(u) \to L(u) \\ f_h \to f \end{matrix}, \quad \text{as} \quad (h \to 0) \tag{1.149}$$

The last important thing is the stability of a method, which was examined in the previous sections. Formally it can be written as (the inverse or solution operator is uniformly bounded):

$$||L_h^{-1}|| \leq C, \qquad \forall h > 0 \tag{1.150}$$

Where we shall now assume that both L and L_h are linear operators. These three conditions are brought together by the following theorem.

Theorem 1 *Consistency* and *Stability* $\Leftrightarrow$ *Convergence*

Proof:

$$||u - u_h|| = ||L_h^{-1}(L_h(u) - L(u)) + L_h^{-1}(f - f_h)|| \tag{1.151}$$

With the triangle inequality we can find the following upper bound:

$$\begin{aligned} &\leq ||L_h^{-1}(L_h(u) - L(u))|| + ||L_h^{-1}(f - f_h)|| && (1.152)\\ &\leq ||L_h^{-1}|| \cdot ||(L_h(u) - L(u))|| + ||L_h^{-1}|| \cdot ||(f - f_h)|| && (1.153)\\ &= ||L_h^{-1}||(||(L_h(u) - L(u))|| + ||(f - f_h)||) && (1.154) \end{aligned}$$

Stability allows us to introduce another bound:

$$\leq C(||(L_h(u) - L(u))|| + ||(f - f_h)||) \tag{1.155}$$

From consistency we get that:

$$\begin{array}{c} ||L_h(u) - L(u)|| \to 0 \\ ||f_h - f|| \to 0 \end{array} \quad \text{as} \quad (h \to 0) \tag{1.156}$$

and thus:

$$C(||(L_h(u) - L(u))|| + ||(f - f_h)||) \to 0, \quad \text{as} \quad (h \to 0) \tag{1.157}$$

which shows the convergence. The other direction needs some deeper results from functional analysis, and will not be given here.

Richardson scheme

As we have seen in one of the previous sections, approximating the time derivative with forward or backward differences gives only an accuracy of $O(\Delta t)$ in time. To overcome this shortcoming, Richardson developed another scheme which has second order accuracy in time. He simply replaced the forward difference by a difference over two time steps at a given point:

$$\frac{\partial u}{\partial t} \approx \frac{u_{n+1,j} - u_{n-1,j}}{2\Delta t} \tag{1.158}$$

Including this approximation into the spatial discretisation of the heat equation generates the following scheme:

$$\frac{u_{n+1,j} - u_{n-1,j}}{2\Delta t} - \frac{\beta^2}{h^2}(u_{n,j-1} - 2u_{n,j} + u_{n,j+1}) = 0 \tag{1.159}$$

The stability is examined again with a von Neumann stability analysis. We start with the ansatz:

$$u_{n,j} = G(k)^n \cdot e^{ikjh} \tag{1.160}$$

Inserting this Ansatz into the difference scheme gives:

$$\frac{1}{2\Delta t}(G(k)^{n+1}e^{ikjh} - G(k)^{n-1}e^{ikjh}) + \frac{\beta^2}{h^2}G(k)^n[-e^{ikh(j+1)} + 2e^{ikhj} - e^{ikh(j-1)}] = 0 \tag{1.161}$$

Dividing by $G(k)^n e^{ikjh} = u_{n,j}$:

$$\frac{1}{2\Delta t}(G(k) - G(k)^{-1}) + \frac{\beta^2}{h^2}[-e^{ihk} + 2 - e^{-ihk}] = 0 \tag{1.162}$$

Replacing again $\cos(x) = \frac{1}{2}e^{ix} + e^{-ix}$:

$$G(k) - G(k)^{-1} = 4r(\cos(kh) - 1) = 4r(-2\sin^2(\frac{kh}{2})) = -8r\sin^2(\frac{kh}{2}) \tag{1.163}$$

Multiplying with $G(k)$ gives the following quadratic equation:

$$G(k)^2 - 1 = -8rG(k)\sin^2(\frac{kh}{2}) \tag{1.164}$$

with solutions:

$$G(k)_{1,2} = -4r\sin^2(\frac{kh}{2}) \pm \sqrt{1 + 16r^2\sin^4(\frac{kh}{2})} \tag{1.165}$$

The expression below the square root is always positive because of the square and the fourth power and larger than 1. Furthermore the first part of Eq. (1.165) is always negative. Thus the dangerous limit is -1 and it is clear that the Richardson method will always have a gain factor less than -1. As a consequence the Richardson method is *unconditionally unstable*. No choice of time step or spatial discretisation can make this method stable. Therefore the only useful application of the Richardson method is as an example for an unstable method.

DuFort-Frankel scheme

One reason for the instability of the Richardson method is probably the fact that the time step where the spatial derivative is computed is not coupled to the time steps where the time derivative is computed. The DuFort-Frankel scheme tries to overcome this problem by replacing the midpoint of the Richardson scheme u_n, j with the average of u_{n-1}, j and u_{n+1}, j. Written in the normal way the DuFort-Frankel scheme takes the following form:

$$\frac{u_{n+1,j} - u_{n-1,j}}{2\Delta t} - \frac{\beta^2}{h^2}(u_{n,j-1} - (u_{n-1,j} + u_{n+1,j}) + u_{n,j+1}) = 0 \tag{1.166}$$

The von Neumann stability analysis shows that this scheme is unconditionally stable. But this method has another drawback which can be analysed by a *consistency analysis*. Using Taylor expansions for the points used in Eq. (1.166):

$$u_{n+1,j} = u(t+\Delta t, x) = u_{n,j} + \frac{\partial u}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 u}{\partial t^2}\Delta t^2 + O(\Delta t^3) \quad (1.167)$$

$$u_{n-1,j} = u(t-\Delta t, x) = u_{n,j} - \frac{\partial u}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 u}{\partial t^2}\Delta t^2 + O(\Delta t^3) \quad (1.168)$$

$$u_{n,j+1} = u(t, x+h) = u_{n,j} + \frac{\partial u}{\partial x}h + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}h^2 + O(h^3) \quad (1.169)$$

$$u_{n,j-1} = u(t, x+h) = u_{n,j} - \frac{\partial u}{\partial x}h + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}h^2 + O(h^3) \quad (1.170)$$

and inserting these equations into Eq. (1.166) we obtain:

$$\frac{2\frac{\partial u}{\partial t}\Delta t + O(\Delta t^3)}{2\Delta t} - \frac{\beta^2(2u_{n,j} + \frac{\partial^2 u}{\partial x^2}h^2 + O(h^3))}{h^2} - \frac{\beta^2(2u_{n,j} + \frac{\partial^2 u}{\partial t^2}\Delta t^2 + O(\Delta t^3))}{h^2} = 0 \quad (1.171)$$

Some simplifications give:

$$\frac{\partial u}{\partial t} - \beta^2\frac{\partial^2 u}{\partial x^2} + E = 0 \quad (1.172)$$

with

$$E = \frac{\beta^2\Delta t^2}{h^2}\frac{\partial^2 u}{\partial t^2} + O(\Delta t^2) + O(h) \quad (1.173)$$

Looking at the error reveals that we do not only have the normal and unavoidable discretisation errors, but also an additional term which does not exist in the original PDE. If we use the DuFort-Frankel scheme without any restrictions, we will get the solution for a different PDE. This is called *inconsistency*. If we use the method to solve the heat equation, we have to require that $\frac{\Delta t}{h} \to 0$ as $\Delta t, h \to 0$, which is incidentally satisfied by the stability requirements we saw earlier, with $\Delta t = O(h^2)$.

1.5 FD Methods in More Dimensions

The numerical solution of 1D problems serves as an introduction to the treatment of problems in higher dimensions. As soon as it comes to the solution of 2 or 3 dimensional problems, the use of numerical solution methods is almost always unavoidable. Here we will cover the basic ideas of finite difference methods in more dimensions.

1.5.1 Basic Ideas

If we consider again the instationary heat equation, but this time in 2 dimensions, we obtain:

$$\frac{\partial u}{\partial t} - \beta^2(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) = f \tag{1.174}$$

Recalling that in the one dimensional case the second spatial derivative was replaced by a finite difference, this idea can be applied straightforward to the 2 dimensional equation. Prior to doing this we again have to introduce a discretisation of the domain (See Fig. 1.7)

Missing figure!!!

Figure 1.7: Scheme of the 2D discretisation

The coordinates can be expressed in terms of the indices j and l:

$$x = j \cdot \Delta x \tag{1.175}$$
$$y = l \cdot \Delta y. \tag{1.176}$$

Then the partial derivatives can be replaced by finite differences:

$$\frac{\partial^2 u_{j,l}}{\partial x^2} = \frac{1}{\Delta x^2}(u_{j-1,l} - 2u_{j,l} + u_{j+1,l}) \tag{1.177}$$
$$\frac{\partial^2 u_{j,l}}{\partial y^2} = \frac{1}{\Delta y^2}(u_{j,l-1} - 2u_{j,l} + u_{j,l+1}) \tag{1.178}$$

Inserting these expressions into Eq. (1.174) we obtain:

$$\frac{\partial u_{j,l}}{\partial t} - \frac{\beta^2}{\Delta x^2}(u_{j-1,l} - 2u_{j,l} + u_{j+1,l}) - \frac{\beta^2}{\Delta y^2}(u_{j,l-1} - 2u_{j,l} + u_{j,l+1}) = f \tag{1.179}$$

Going one dimension up to three dimensional problems, the basic idea stays the same. Introducing another coordinate z:

$$z = k \cdot \Delta z \tag{1.180}$$

we get the following approximation for the second partial derivative with respect to z:

$$\frac{\partial^2 u_{j,l,k}}{\partial z^2} = \frac{1}{\Delta z^2}(u_{j,l,k-1} - 2u_{j,l,k} + u_{j,l,k+1}) \tag{1.181}$$

The semi-discretisation of the three dimensional instationary heat equation then obviously becomes:

$$\begin{aligned}\frac{\partial u_{j,l,k}}{\partial t} &- \frac{\beta^2}{\Delta x^2}(u_{j-1,l,k} - 2u_{j,l,k} + u_{j+1,l,k}) - \frac{\beta^2}{\Delta y^2}(u_{j,l-1,k} - 2u_{j,l,k} + u_{j,l+1,k}) \\ &- \frac{\beta^2}{\Delta z^2}(u_{j,l,k-1} - 2u_{j,l,k} + u_{j,l,k+1}) = f.\end{aligned} \tag{1.182}$$

1.5.2 Computational Molecules/Stencils

Another simplification is to use the same step size in both space directions. This leads then in 2D to the following expression with $h = \Delta x = \Delta y$ being the unique discretisation parameter:

$$\frac{\partial u_{j,l}}{\partial t} - \frac{\beta^2}{h^2}(-4u_{j,l} + u_{j-1,l} + u_{j+1,l} + u_{j,l-1} + u_{j,l+1}) = f. \tag{1.183}$$

or in 3d to:

$$\frac{\partial u_{j,l,k}}{\partial t} - \frac{\beta^2}{h^2}(-8u_{j,l,k} + u_{j-1,l,k} + u_{j+1,l,k} + u_{j,l-1,k} + u_{j,l+1,k} + u_{j,l,k-1} + u_{j,l,k+1}) = f. \tag{1.184}$$

A very nice way to visualise these schemes is to draw the points used in the schemes with their weights in the original computational domain. For the two schemes shown here one obtains pictures as shown in Fig. 1.8 and Fig. 1.9. These are often referred to as *Computational Molecules* or *Stencils*.

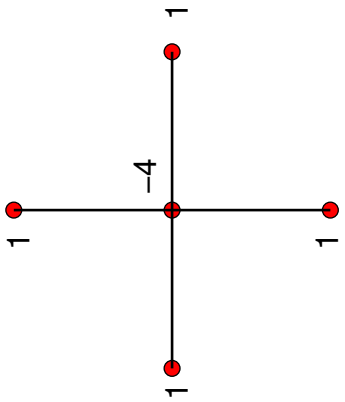


Figure 1.8: Stencil for 2D Laplace operator

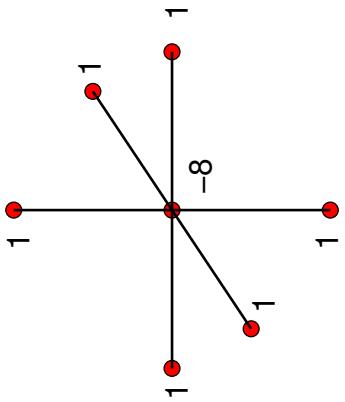


Figure 1.9: Stencil for 3D Laplace operator

1.5.3 Boundary Treatment

As we know from 1-D problems, the solution is only completely specified if the boundary conditions are satisfied along with the differential equation. These boundary conditions have to be discretised also for a numerical treatment. When the boundaries are not straight, this becomes a cumbersome procedure for finite difference methods. We will not treat these here, and refer to specialist texts. We will also see that this problem is much easier with the finite element method, which will be treated next in the more general context of weighted residual methods.

1.5.4 Time Discretisation

Similar to the one dimensional case, the resulting system of ordinary differential equations normally does not possess an analytical solution, which makes the use of numerical methods necessary. Using the θ-method as an example we obtain the following system of equations:

$$\begin{aligned}\frac{u_{j,l}^{n+1}-u_{j,l}^{n}}{\Delta t} = \frac{\beta^2}{h^2}((1-\theta)(-4u_{j,l}^{n}+u_{j-1,l}^{n}+u_{j+1,l}^{n}+u_{j,l-1}^{n}+u_{j,l+1}^{n})+ \\ \theta(-4u_{j,l}^{n+1}+u_{j-1,l}^{n+1}+u_{j+1,l}^{n+1}+u_{j,l-1}^{n+1}+u_{j,l+1}^{n+1}))+f\end{aligned} \tag{1.185}$$

For three and higher dimensional problems the idea and implementation is straightforward. But it should be noted that the computational effort increases extremely fast with higher dimensions. While in one dimension, taking $h = 0.01$ for a unit interval leads to approximately 100 points, the same discretisation size for a unit cube in three dimensions leads to 1000000 points!. Hence for higher dimensional problems often the practical implementation becomes the real problem.

Chapter 2

Equilibrium Equation and Iterative Solvers

The solution of the homogeneous heat equation with constant boundary conditions approaches a stationary state.

$$\frac{\partial}{\partial t}u(x,y,z,t) - \beta^2 \Delta u(x,y,z,t) = f(x,y,z) \tag{2.1}$$

$$u(x,y,z,t) \to \tilde{u}(x,y,z) \quad \text{as} \quad t \to \infty \tag{2.2}$$

This is the steady state of the instationary heat equation and also the solution of the equilibrium equation.

$$\frac{\partial}{\partial t}\tilde{u}(x,y,z) = 0 \quad \Rightarrow \quad -\beta^2 \Delta \tilde{u}(x,y,z) = f(x,y,z) \tag{2.3}$$

In this chapter this equilibrium equation or stationary heat equation will be introduced. After that some methods to find a solution for this equation will be introduced.

2.1 Equilibrium equation

The general form of the heat equation was:

$$\frac{\partial}{\partial t}u(x,y,z,t) - \beta^2 \Delta u(x,y,z,t) = f(x,y,z,t) \tag{2.4}$$

Other physical phenomena like diffusion can also be modelled with this type of equation. This equation is a member of the family of *parabolic* equations. A more detailed description of the different classes of partial differential equations will follow in a later chapter.

In order to have a unique solution of this equation we need some boundary and initial conditions. After spatial discretisation we have the following system of ODEs:

$$\frac{\partial}{\partial t}\mathbf{u} + \mathbf{A}\mathbf{u} = \mathbf{f} \tag{2.5}$$

If the right hand side term $\mathbf{f}$ is independent of the time, and all boundary conditions are also constant in time, the solution of Eq. (2.4) will converge to a steady state as $t \to \infty$.

In the steady state the solution does not change anymore, so $\frac{\partial u}{\partial t} = 0$ and thus the steady state will also satisfy the following partial differential equation:

$$-\beta^2 \Delta u(x, y, z) = f(x, y, z) \tag{2.6}$$

together with the boundary conditions. Now the equation is of *elliptic* type. Several other problems like the stationary state of mechanical systems like displacement of the membrane of a drum or the displacement of a simple beam can be described by elliptic equations.

If we apply finite difference approximation for the spatial derivative we obtain a system of linear equations:

$$\mathbf{A}\mathbf{u} = \mathbf{f} \tag{2.7}$$

with

$$\mathbf{A} = \frac{\beta^2}{h^2}\begin{bmatrix} 2 & -1 & 0 & & \dots \\ -1 & 2 & -1 & 0 & \dots \\ 0 & \ddots & \ddots & \ddots & \\ \vdots & & & & \\ & & & -1 & 2 \end{bmatrix}. \tag{2.8}$$

This matrix is tridiagonal and hence very sparse (its entries are mostly zeros). Tridiagonal matrices can be factorised by direct elimination in $O(n)$ operations (the so called *Thomas algorithm*).

Normally, the discretisation of PDEs leads to sparse and often very large matrices with solution vectors of several million unknowns, because the solution becomes more accurate if the spatial and temporal discretisation is refined.

For not too large systems of linear equations the fastest solution if often to use a direct solution method like Gaussian elimination. Especially for one dimensional problems one can achieve a numerical complexity of $O(n)$ where n is the number of unknowns. But for higher dimensional problems the complexity of efficient direct solvers becomes $O(n^2)$ for typical grid problems in 3D. This makes the use of an alternative approach for very large systems of equations necessary.

2.2 Iterative methods

While the direct solvers try to find the solution of the system of equations in a finite and predetermined number of steps, the iterative solution methods start with an initial guess of the solution, and try then to get closer to the correct solution with each iteration. One then usually stops the iteration when the iteration error is in the same order of magnitude as the discretisation error. All the iterative methods replace the direct solution of the original system of equations with the direct solution of a simpler system, which has to be iterated over and over.

2.2.1 Timestepping, Richardson's Method

If we look back to the instationary heat equation we can already identify a first iterative method. We found out that the steady state solution as $t \to \infty$ of the instationary heat equation is a solution of Eq. (2.6) and thus also a solution of Eq. (2.7). Starting with the initial conditions the Euler forward method allows us to come closer to the stationary solution without solving any systems of equations.

Obviously it will not be possible to come to $t = \infty$ with finite time steps, but assuming we have chosen a stable time step size we can be sure that every iteration brings the approximation closer to the correct solution of the equilibrium equation. Hence an arbitrary accuracy can be achieved after a finite number of time steps.

The Euler forward method for

$$\dot{\mathbf{u}} + \mathbf{A}\mathbf{u} = \mathbf{f} \tag{2.9}$$

was

$$\frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\Delta t} + \mathbf{A}\mathbf{u} = \mathbf{f}. \tag{2.10}$$

Rewriting it in matrix form gives:

$$\mathbf{u}_{n+1} = (\mathbf{I} - \Delta t\mathbf{A})\mathbf{u}_n + \Delta t\mathbf{f} \tag{2.11}$$

This method is equivalent to Richardson's method for solving a linear system of equations $\mathbf{A}\mathbf{u} = \mathbf{f}$:

$$\mathbf{u}_{n+1} = (\mathbf{I} - \vartheta\mathbf{A})\mathbf{u}_n + \vartheta\mathbf{f} = \mathbf{u}_n + \vartheta(\mathbf{f} - \mathbf{A}\mathbf{u}_n) \tag{2.12}$$

with a parameter ϑ which must be sufficiently small.

2.2.2 Jacobi's Method

A slightly different view to Eq. (2.12) reveals that every iteration is the solution of a very simple system of linear equations:

$$\mathbf{I}\mathbf{u}_{n+1} = \vartheta\mathbf{f} - (\vartheta\mathbf{A} - \mathbf{I})\mathbf{u}_n \tag{2.13}$$

Jacobi's method may be seen as replacing the identity matrix with a matrix of similar complexity which is closer to the original system of linear equations. The diagonal matrix $D = \mathrm{diag}(A)$ has the same structure as the identity matrix but is closer to the original system of linear equations and is thus used for the Jacobi method:

$$\mathbf{D}\mathbf{u}_{n+1} = \vartheta\mathbf{f} - (\vartheta\mathbf{A} - \mathbf{D})\mathbf{u}_n \tag{2.14}$$

Another view, and the one initially motivating Jacobi, of the same method is illustrated in Fig. 2.1.

Assuming the solution is known on all nodes except our current node j, we simply solve the system of equations for that node:

$$\sum_{i=1}^{N} a_{ji}u^{(i)} = f^{(j)} \Leftrightarrow a_{jj}u^{(j)} = f^{(j)} - \sum_{i\neq j} a_{ji}u^{(i)} \Rightarrow u_{n+1}^{(j)} = \frac{1}{a_{jj}}f^{(j)} - \frac{1}{a_{jj}}\sum_{i\neq j} a_{ji}u_n^{(i)} \tag{2.15}$$

Obviously, Eq. (2.15) is equivalent to Eq. (2.14) with $\vartheta = 1$.

2.2.3 Matrix Splitting methods

One common principle of the Eq. (2.13) and Eq. (2.14) was the solution of a simpler system of equations in each iteration. This principle is generalised in the matrix splitting methods. Instead of solving the original system of equations one time, simpler systems which are similar to the original system of equations are solved several times to approximate the solution.

A formal derivation starts with the original system of linear equations:

$$\mathbf{A}\mathbf{u} = \mathbf{f} \tag{2.16}$$

Multiplying the system with a factor ω and adding $\mathbf{M}\mathbf{u}$ gives:

$$\mathbf{M}\mathbf{u} = \mathbf{M}\mathbf{u} + \omega(\mathbf{f} - \mathbf{A}\mathbf{u}) \tag{2.17}$$

From this system of equations, which is equivalent to the original system, the iterative method is derived as:

$$\mathbf{M}\mathbf{u}_{n+1} = \mathbf{M}\mathbf{u}_n + \omega(\mathbf{f} - \mathbf{A}\mathbf{u}_n) \tag{2.18}$$

For $\omega = 1$ we have

$$\mathbf{M}\mathbf{u}_{n+1} = \mathbf{f} - (\mathbf{M} - \mathbf{A})\mathbf{u}_n = \mathbf{f} + (\mathbf{A} - \mathbf{M})\mathbf{u}_n \tag{2.19}$$

So we see the matrix $\mathbf{A}$ is split into the parts $\mathbf{A}$ and $\mathbf{A} - \mathbf{M}$.

It is important to have a matrix $\mathbf{M}$ which allows a fast solution of the system of equations. A broad class of very popular methods is based on the splitting of $\mathbf{A}$ into the strictly lower triangular part $\mathbf{E}$, the strictly upper triangular part $\mathbf{E}^T$ and the diagonal part $\mathbf{D}$ (See Fig. 2.2):

$$\mathbf{A} = \mathbf{D} - \mathbf{E} - \mathbf{E}^T \tag{2.20}$$

Gauss-Seidel

If we consider again the Jacobi method, written with the splitting matrices, we obtain (with $\omega = 1$):

$$\mathbf{D}\mathbf{u}_{n+1} = \mathbf{D}\mathbf{u}_n + \mathbf{f} - (\mathbf{D} - \mathbf{E} - \mathbf{E}^T)\mathbf{u}_n = \mathbf{f} + \mathbf{E}\mathbf{u}_n + \mathbf{E}^T\mathbf{u}_n \tag{2.21}$$

Under the assumption that our algorithm starts at the first unknown u_1 and goes down to the last unknown u_N, we have already new values for $u_i, i = 1 \dots j-1$ at position j. The Gauss-Seidel algorithm takes this into account by using these new values as soon as they are available. From Eq. (2.15) we have

$$a_{jj}u_{n+1}^{(j)} = f^{(j)} - \sum_{i<j} a_{ji}u_{n+1}^{(i)} - \sum_{i>j} a_{ji}u_n^{(i)}, \tag{2.22}$$

or in matrix form:

$$\mathbf{D}\mathbf{u}_{n+1} = \mathbf{f} + \mathbf{E}\mathbf{u}_n + \mathbf{E}^T\mathbf{u}_{n+1} \tag{2.23}$$

But as often the advantage of a faster convergence has some disadvantages. For large scale applications it is often necessary to use parallel computers. The Jacobi method allows an almost trivial parallelisation of the algorithm. Each processor gets some unknowns and can compute the next iteration independently of the other processors. After each iteration the new results must be distributed.

In contrast the Gauss-Seidel algorithm cannot be parallelised in its original form because the steps $j+1..N$ can only be started after the results $1..j$ are known. To overcome this problem algorithms like the Block-Gauss-Seidel method were developed.

A typical implementation of the Gauss-Seidel method is shown below:

```
fct gauss_seidel (A,f,u)
  for k := 1 to convergence
    for j := 1 to N
      u(j)k+1 = 1/ajj (fj - sum_{i=1}^{j-1} aji u(i)k+1 - sum_{i=j+1}^{N} aji u(i)k )
    end
  end
end.
```

Successive Over-Relaxation (SOR)

Another acceleration of the solution process is achieved by the SOR method which is the abbreviation for *Successive Over Relaxation*. Here the assumption is that each iteration brings the solution closer to the right solution by a small amount $\Delta\mathbf{u}$. So for the Jacobi or Gauss Seidel method we have something like:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \Delta\mathbf{u}_k \tag{2.24}$$

If $\Delta\mathbf{u}$ points into the direction of the solution we can come even closer to the solution if we go a little bit further in that direction. Hence the SOR method uses:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \omega\Delta\mathbf{u}_k \tag{2.25}$$

Beside the same parallelisation problems as in the Gauss-Seidel method the optimal choice of ω is another problem with the SOR method. For most problems relaxation parameters like $\omega = 1.1$ can already bring a slight improvement.

A last variation is the SSOR method which changes the direction after each iteration. The first iteration goes from $j = 1$ and the next from $j = N$.

Summary

In previous sections we saw some of the basic ideas of iterative solvers. The following table gives an overview about the most popular matrix splitting methods:

- **Richardson $\mathbf{M = I}$**
- **Jacobi $\mathbf{M = D}$**

- **Gauss-Seidel** $\mathbf{M} = \mathbf{D} - \mathbf{E}$ or $\mathbf{M} = \mathbf{D} - \mathbf{E}^T$
- **SOR** $\mathbf{M} = \frac{1}{\omega}\mathbf{D} - \mathbf{E}$ or $\mathbf{M} = \frac{1}{\omega}\mathbf{D} - \mathbf{E}^T$
- **SSOR** once $\mathbf{M} = \frac{1}{\omega}\mathbf{D} - \mathbf{E}$ and once $\mathbf{M} = \frac{1}{\omega}\mathbf{D} - \mathbf{E}^T$

2.3 Multigrid methods

To increase the accuracy of numerical solutions of PDEs is to increase the number of unknowns. This leads to huge systems of linear or nonlinear equations which must be solved efficiently. Direct solvers and the simple iterative solvers shown in the previous section reach their limits at roughly several thousands of unknowns. Complexity analysis shows this behaviour and will be introduced in short in the last subsection.

Because problems like fluid dynamics need even more unknowns they often use a more sophisticated iterative solution strategy called Multigrid. The basic ideas and concepts will be shown in the next section.

2.3.1 Idea

The basis of multigrid method is the clever usage of the so called *smoothing property* of most iterative solvers for systems of linear equations. Considering the system of linear equations coming from the stationary heat equation we have several values along the X-axis. Starting with a random initial guess for the solution vector $\mathbf{u}$ the residuum $\mathbf{r} = \mathbf{f} - \mathbf{A}\mathbf{u}$ along the X-axis looks very irregular (See Fig. 2.3). Interpreting the solution as a time series all frequencies are included.

If we start iterating with the Gauss-Seidel method we can observe that each iteration makes the curve of the residual look more smooth. This means that the higher spatial frequencies (wavenumbers) are diminished (See Fig. 2.4).

Continuing with the iteration at some time only a smooth residual is left which decreases very slowly (See Fig. 2.5). Looking at the norm of the residual vector shows also that the error decreases very fast in the beginning and quite slowly at the end (See Fig. 2.6).

From this observation the basic idea is not far away. Transferring the residual on the fine grid to a coarser grid by an arbitrary *restriction operator* lets it look "rougher" to the iterative solver on the coarser grid, which performs better as a consequence.

After the smooth parts of the residual were decimated on the coarse grid, the correction to the solution is transferred back to the finer grid with an *interpolation operator*. Here only the rough parts are left and can be smoothed away by the iterative solver. This grid-transfer process is then repeated again and again.

2.3.2 Algorithm

The simplest implementation of the idea is the *Twogrid iteration* It uses a coarse and a fine grid. Furthermore an interpolation and a restriction operator are required. Probably the simplest restriction operator is to take only every second node of the grid. For interpolation an easy and often used method is the linear interpolation which takes the average of the two neighbouring points.

Twogrid iteration

For the variables the subset index h or H denotes if the variable is defined on the fine or coarse grid. The superset is used for the iteration number.

The current solution vector is denoted by $\mathbf{v}$, the matrix is called $\mathbf{A}$ and the residuum $\mathbf{r}$. As the exact solution $\mathbf{u}$ satisfies $\mathbf{Au} = \mathbf{f}$, the error $\mathbf{e} = \mathbf{u} - \mathbf{v}$ satisfies $\mathbf{Ae} = \mathbf{Au} - \mathbf{Av} = \mathbf{f} - \mathbf{Av} = \mathbf{r}$

With these definitions we get the following algorithm to compute the next iteration $k+1$ of the solution vector v_h^k:

1. Smooth e, $v_h^k \rightarrow \tilde{v_h^k}$
2. Compute residual $r_h^k = f_h - A_h \tilde{v_h^k}$
3. Transfer $r_h^k \rightarrow r_H^k$ (restriction)
4. On Grid H solve $A_H e_H^k = r_H^k$
5. Transfer $e_H^k \rightarrow e_h^k$ (prolongation)
6. $v_h^{k+1} = \tilde{v}_h^k + e_h^k$
7. Optionally smooth v_h

Graphically this algorithm can be visualised as shown in Fig. 2.7. Especially for more complicated iteration schemes this visualisation becomes useful for understanding the algorithm.

Multigrid iteration

One point in the two grid algorithm is not totally satisfying. In the fourth step the direct solution of a smaller system of equations is required. For large problems this system of equations may again be too large to solve directly. So the idea of the multigrid iteration is to introduce another two grid scheme to solve this system of equations. Applying this recursion several times gives a complete hierarchy of levels.

The variable names are the same as in the twogrid algorithm. Instead subscripts of h and H an index variable l is introduced. Additionally we need a stopping criterion for the recursion which is given by the number of levels lev.

Starting with an initial guess v_1 on the fine grid we call the function `MG(1,v,lev)`

```
fct x = MG(l,v_l^k, f_l,lev)
if l = lev
  Solve directly A_l x_l = f_l
else
  Smooth v_l^k → ṽ_h^k
  Compute residual r_l^k = f_l − A_l ṽ_h^k
  Transfer r_l^k → r_{l+1}^k (restriction)
  On grid l+1, e_{l+1}^{k+1} = MG(l+1,e_{l+1}^k, r_{l+1}^k, lev)
  Transfer e_{l+1}^{k+1} → e_l^{k+1} (prolongation)
end
```

A graphical visualisation of the multigrid algorithm is shown in Fig. 2.8. Because of its V-shape in the visualisation, a complete iteration is often called a V-cycle.

Full Multigrid V-Cycle (FMV)

Another improvement to the multigrid idea is the Full Multigrid V-Cycle which starts on the coarsest level and takes several iterations limited to the two coarsest grids. This gives the iteration on finer grids good starting values. After that the number of levels included into the iteration is increased by one. This process continues until all levels are involved in the V-Cycle (See Fig. 2.9). Empirical analysis shows that the FMV algorithm is one of the most efficient algorithms for several problem types.

2.3.3 Complexity

An important issue regarding solvers for systems of linear equations is their complexity. This is a function which describes the asymptotical runtime of the algorithm depending on one or more variables describing the size of the problem.

Table 2.1 provides an overview about the complexity of several solvers for systems of linear equations coming from a typical test problem, a finite difference discretisation of the Laplace equation on a regular grid. The value in the table represents the exponent k in the complexity function $O(n^k)$ where n is the number of unknowns. Because the structure of the matrix plays an important rule in the runtime behaviour of the solvers the dimension of the test problems appears in the first row.

A first observation is that the complexity of iterative solvers decreases with increasing dimension, while the complexity of the direct solver increases. As a rule of thumb direct

Dimension/Method	1D	2D	3D
Jacobi/GS	3	2	5/3
SOR	2	3/2	4/3
FMV	1	1	1
Direct	1	3/2	2
PCG	3/2	5/4	7/6

Table 2.1: Complexity of linear solvers

solvers perform well for 1 and 2 dimensional problems but are often unusable for large problems in 3 dimensions. Iterative solvers become better for higher dimensional problems and a large number of unknowns. But the performance of iterative solvers depend heavily on the matrix, whereas direct solvers depend only on the structure of the matrix and are therefore more robust.

Full Multigrid solvers seem to be perfectly suited for problems in any dimension and also achieve the optimal performance. But they are generally not usable as "Blackbox" solvers. Often the adaption to a special problem is very difficult. So most time they are used in programs which can cope only with a special kind of problem like fluid solvers.

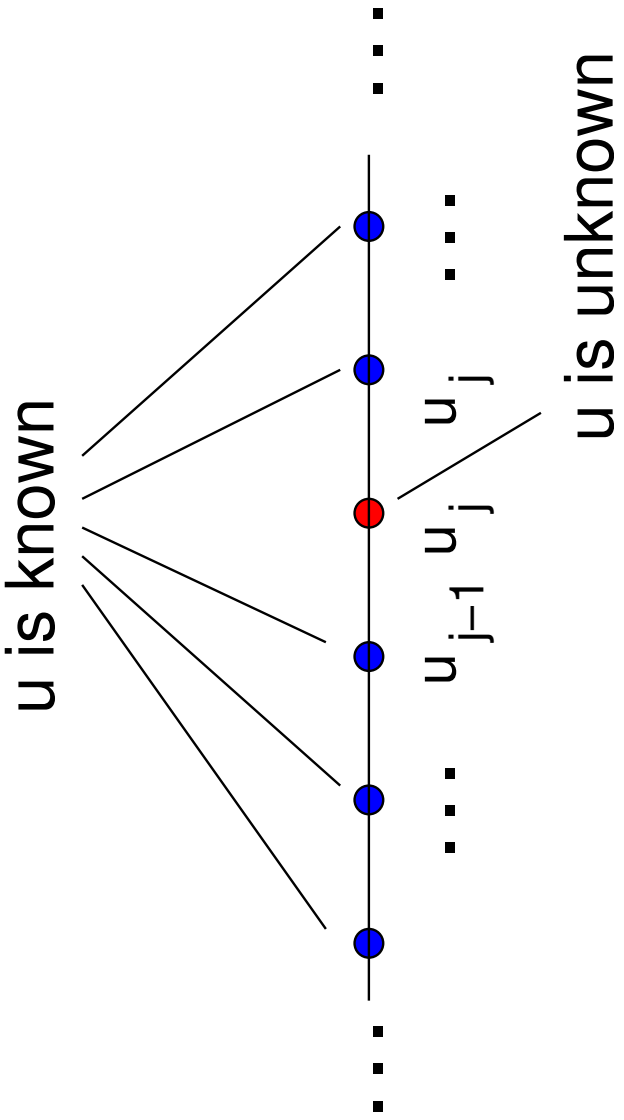


Figure 2.1: Scheme of the Jacobi method

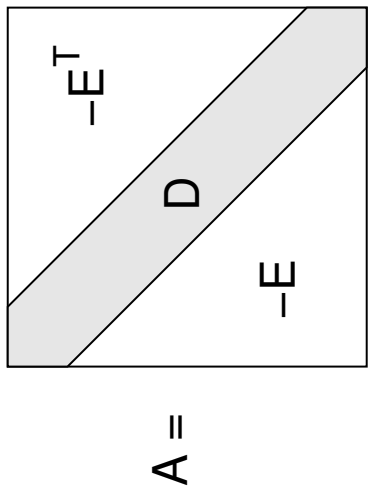


Figure 2.2: Matrix splitting

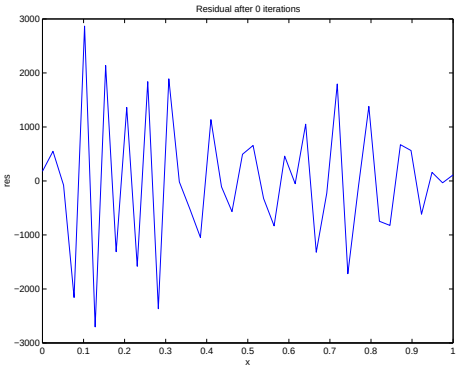


Figure 2.3: Initial Residual

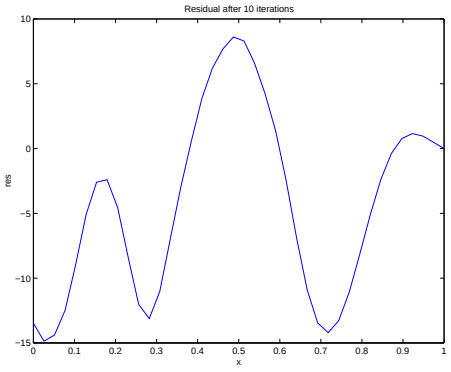


Figure 2.4: Residual after 10 it.

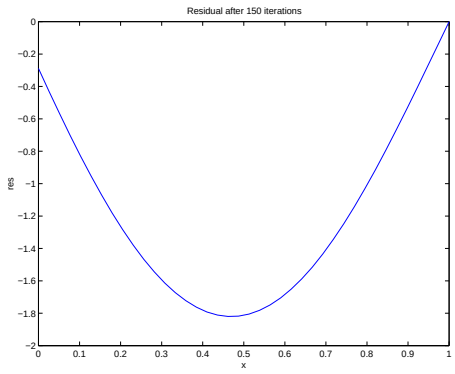


Figure 2.5: Residual after 150 it.

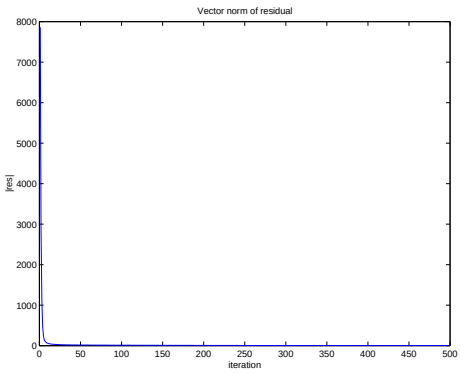


Figure 2.6: Norm of the residual

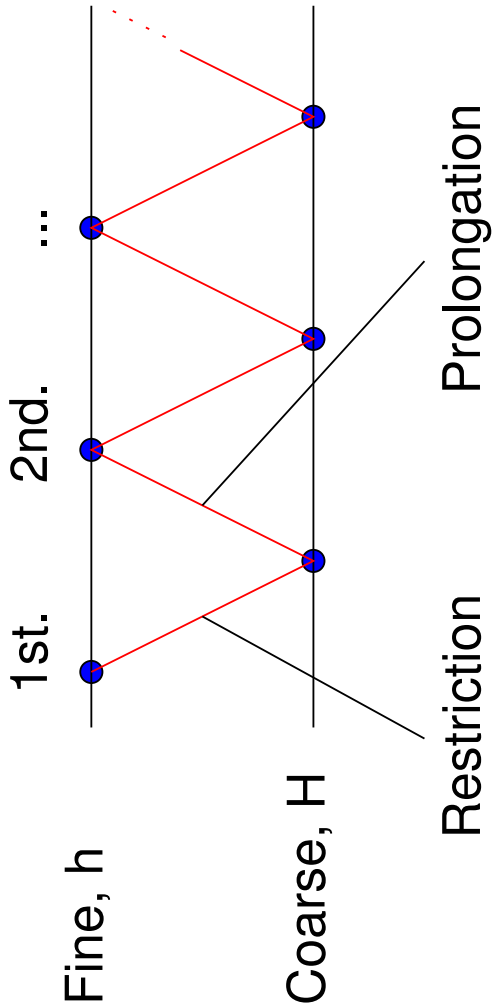


Figure 2.7: Twogrid algorithm

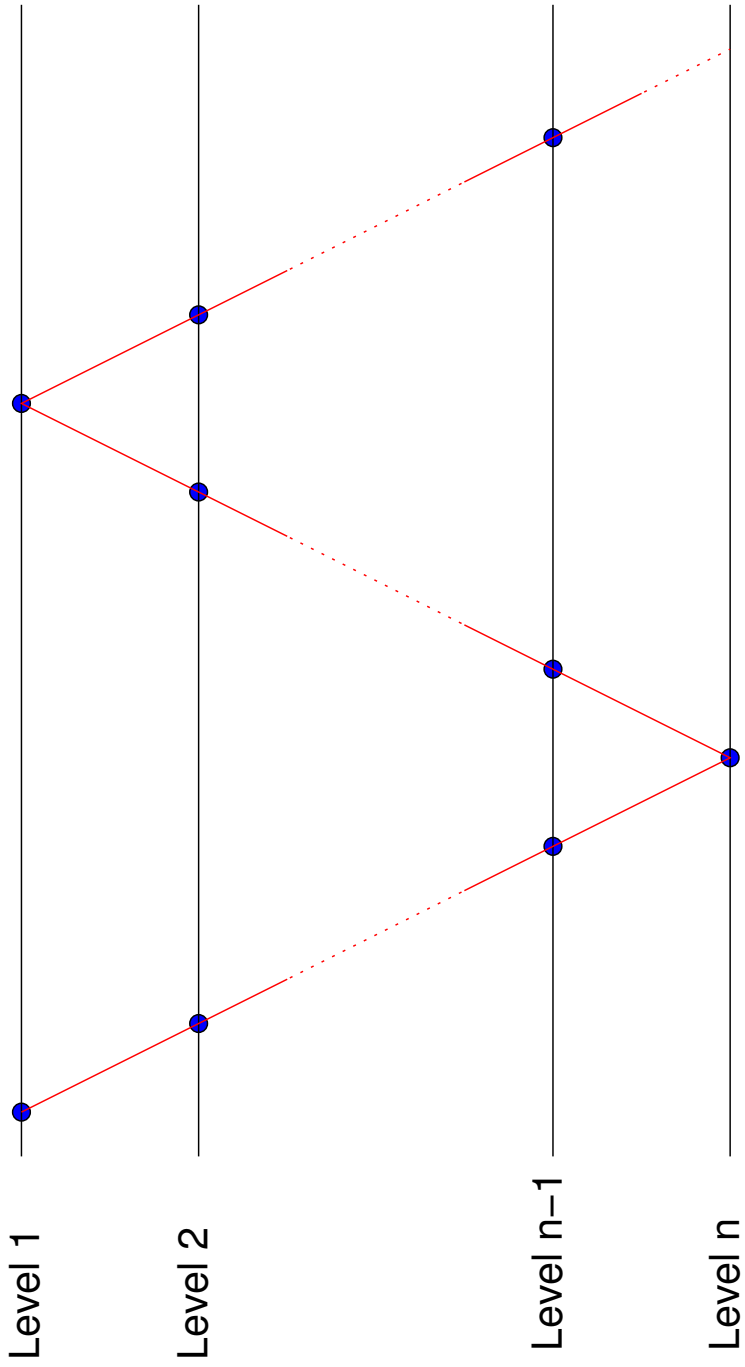


Figure 2.8: Multigrid algorithm

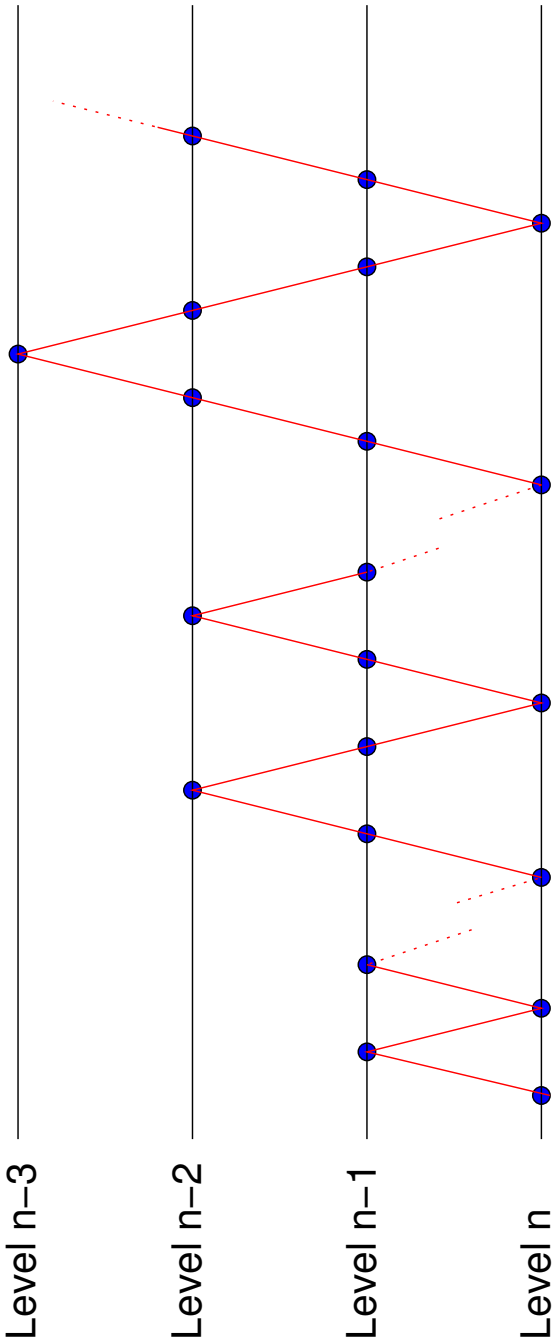


Figure 2.9: Full Multigrid V-Cycle

Chapter 3

Weighted residual methods

In this section another method, or more precisely a general receipt for a couple of methods, will be introduced. Although the finite difference method is convincing by its intuitive approach and its simplicity it becomes quite difficult if it is applied to irregular domains. Another disadvantage is the missing general framework for theoretical analysis which is available for the weighted residual methods and thus gives some insight and a deeper understanding of this class of methods.

3.1 Basic theory

As a simple example we will consider the stationary heat equation (often also called Poisson-equation):

$$-\Delta u = f \tag{3.1}$$

3.1.1 Weak form

The main idea is now to multiply the partial differential equation with a *weighting function* φ and to integrate over the whole domain Ω:

$$\Rightarrow \int_\Omega -\Delta u \cdot \varphi \, d\Omega = \int_\Omega f \cdot \varphi \, d\Omega, \qquad \forall \varphi \in V \tag{3.2}$$

If Eq. (3.2) holds for every φ it is equivalent to Eq. (3.1). For sake of simplicity we assume that $u = 0$ on $\partial\Omega$. Then with Gauss' theorem the following equation can be derived:

$$\int_\Omega (\nabla u)^T \cdot \nabla \varphi \, d\Omega = \int_\Omega f \cdot \varphi \, d\Omega, \qquad \forall \varphi \tag{3.3}$$

3.1.2 Variational formulation

An alternative use of the Poisson equation is to describe the displacement of an elastic bar under load. It is well recognised that elastic structures minimise their internal energy. So mechanical systems possess a natural minimisation principle:

$$-\Delta u = f \Leftrightarrow \min \underbrace{\left(\frac{1}{2}\int ||\nabla u||^2 + \int uf\right)}_{\text{Energy}p} \tag{3.4}$$

To minimise this functional it is necessary that its first variation becomes zero.

$$\begin{aligned} p(u+v) &= \frac{1}{2}\int ||\nabla u + \nabla v||^2 - f(u+v) \\ &= p(u) + \underbrace{\int (\nabla u \cdot \nabla v - fv)}_{=0 \quad \text{for min.}} + \frac{1}{2}\int (\nabla v)^2 \end{aligned} \tag{3.5}$$

If u minimises p, then

$$\int_\Omega (\nabla u)^T \cdot \nabla v \, d\Omega = \int_\Omega f \cdot v \, d\Omega, \qquad \forall v \tag{3.6}$$

which is equivalent to Eq. (3.3). The solution obtained by using the weighted residual methods is thus equivalent to minimising the energy of the system.

3.1.3 Numerical methods

To solve the weak form (Eq. (3.3)) it is necessary to introduce an approximation of the function u. In the most general form this approximation is the sum of several *ansatzfunctions* N_i which are multiplied with coefficients u_i:

$$u(x) \approx u_h(x) = \sum_{i=1}^{N} u_i N_i(x) \tag{3.7}$$

If this Approximation is put into Eq. (3.3) it is not possible to satisfy the equation for all φ. Instead a finite subspace $V_h \subset V$ must also be selected for the weighting functions. This subspace may only have as much spanning functions as the space of Ansatzfunctions in order to have a solution for Eq. (3.3). So the weightingfunction φ can be expressed similarly as:

$$\varphi(x) \approx \varphi_h(x) = \sum_{i=1}^{N} \varphi_i(x) \tag{3.8}$$

Depending on the type of weightingfunctions the numerical methods have different names.

Bubnov Galerkin methods

The characteristic of Bubnov Galerkin methods is that the weightingfunctions are the same as the ansatzfunctions.

$$\varphi_i = N_i \tag{3.9}$$

It is one of the most popular weighted residual methods. Often the name *Finite Element Method* or FEM is used synonymous with this type of weightingfunctions.

Petrov Galerkin methods

Petrov Galerkin methods are all weighted residual methods where the weightingfunctions are different from the ansatzfunctions. It is obvious that this includes all methods which are not Bubnov Galerkin. Nevertheless in literature most of the methods got different names.

One choice for the weightingfunctions is the delta function:

$$\varphi_i = \delta(x - x_i) \tag{3.10}$$

Because integrating with the delta functions gives the function value at one point this method is called *pointwise collocation*. Another choice is the characteristic function of some subdomain Ω_i inside the original domain Ω:

$$\varphi_i = \chi_{\Omega_i} \tag{3.11}$$

From obvious reasons this method is called *subdomain collocation*. It was independently developed for conservation laws and is therefore often also called the *Finite Volume Method*.

Least Squares

Although the Least Squares method can also be seen as a Bubnov Galerkin method it has some special properties. The idea is to apply the differential operator twice. One time to the ansatzfunctions and once to the weightingfunctions. If we consider an abstract differential operator L the least squares formulation is:

$$\int_\Omega (Lu - f)(L\varphi)\, d\Omega = 0 \tag{3.12}$$

This method causes some difficulties when applied directly to higher order partial differential equations. Hence the most common approach is to convert the partial differential equation into a first order system first.

Types of ansatzfunctions

Beside the different choices for the weightingfunctions there are also several possible ways to choose the ansatzfunctions N_i. Some are:

- Polynomials: $N_i = x^i$
- First N eigenfunctions of L: $LN_i = \lambda_i \cdot N_i$
- Trigonometric functions: $N_i = \sin(ix), N_i = \cos(ix)$
- Piecewise polynomials: $N_i = x^i \quad \text{on}\Omega_i$

Not every set of functions is well suited for the solution of partial differential equations. And functions which may be good from the analytical point of view may cause problems in the numerical treatment. The most popular choice today are piecewise polynomials because they have some very useful properties. For special problems like weather simulation also the trigonometric functions are used. These methods are called *spectral methods*.

3.2 Example: The Finite Element method

Now the ingredients are complete to find an approximate solution for the partial differential equation. Inserting the ansatzfunctions and the weightingfunctions into Eq. (3.3) gives:

$$\int_\Omega \frac{\partial}{\partial x} \sum_{i=1}^{N} u_i N_i \cdot \frac{\partial}{\partial x} \sum_{j=1}^{N} N_j \, d\Omega = \int_\Omega f \sum_{j=1}^{N} N_j \, d\Omega \tag{3.13}$$

Evaluating these integrals for every index pair $(i, j) \in [1 \ldots N] \times [1 \ldots N]$ transforms this equation into a system of linear equations:

$$\Rightarrow \mathbf{Au} = \mathbf{f} \tag{3.14}$$

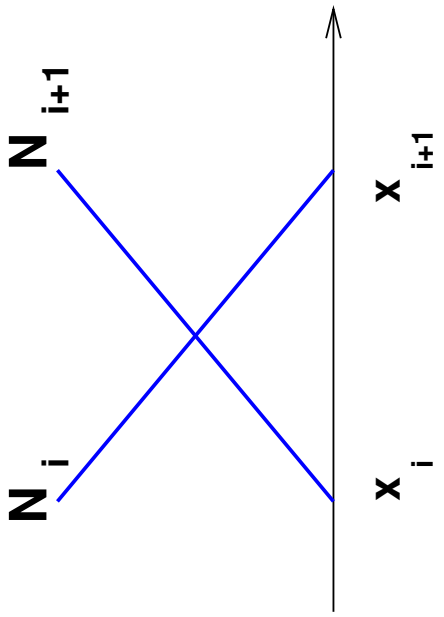


Figure 3.1: Linear ansatzfunctions in 1D

3.2.1 Nodal basis

For the sparsity, which means the matrix $\mathbf{A}$ has only few nonzero entries, a local definition of the ansatzfunctions is necessary. Piecewise polynomials are widely used for this purpose. Here we will look at piecewise linear functions in one dimension. The one dimensional domain is then subdivided into several smaller parts $\Omega_i = [x_i \dots x_{i+1}$. The ansatzfunctions on this interval are then (see Fig. 3.1) :

$$N_i(x) = \begin{cases} (x - x_i)/l & x \in [x_i, x_{i+1}] \\ 0 & \text{else} \end{cases} \tag{3.15}$$

$$N_{i+1}(x) = \begin{cases} (x_{i+1} - x)/l & x \in [x_i, x_{i+1}] \\ 0 & \text{else} \end{cases} \tag{3.16}$$

where $l = x_{i+1} - x_i$ is the length of the interval. The complete domain is then covered by these functions (see Fig. 3.2).

It can easily be seen that the ansatzfunction N_i is one at the point x_i and zero at all other points $x_j, j \neq i$. So if we find a solution vector $\mathbf{u}$ the value of our approximate solution (Eq. (3.7)) at the point x_i is equal to the value of the coefficient u_i. For the interpretation of the solution this property is very helpful because it makes the reconstruction of the approximate solution unnecessary. The points x_i are often called *nodes* which also gives the name for this type of ansatzfunctions.

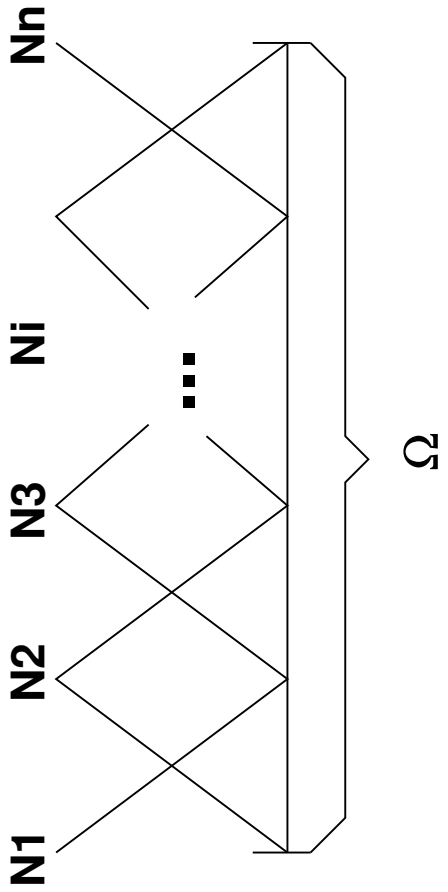


Figure 3.2: Ansatzfunctions on the whole domain

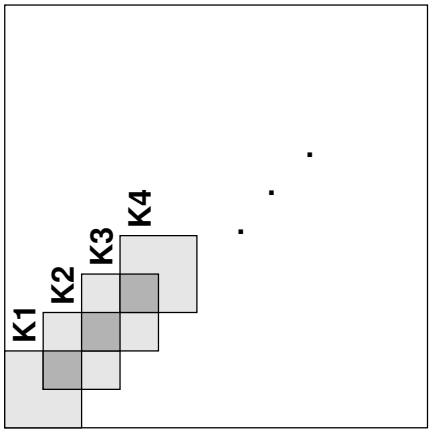


Figure 3.3: Assembly of the global matrix K

3.2.2 Matrix assembly

Another advantage of the nodal basis was the local definition of the ansatzfunctions. This property allows the easy evaluation of Eq. (3.13). Because the ansatzfunctions are only nonzero inside the local subdomain, the product of two ansatzfunctions can also be nonzero only in the local subdomain. So the common way to get the global matrix in Eq. (3.14) is to assemble it from the distributions of the small subdomains Ω_i which are called *elements* in the Finite Element method.

Consider the subdomain Ω_i going from x_i to x_{i+1} with length $l_i = x_{i+1} - x_i$. The local system of equations is then:

$$\begin{pmatrix} \int_{x_i}^{x_{i+1}} \frac{\partial}{\partial x} N_i \frac{\partial}{\partial x} N_i \, dx & \int_{x_i}^{x_{i+1}} \frac{\partial}{\partial x} N_{i+1} \frac{\partial}{\partial x} N_i \, dx \\ \int_{x_i}^{x_{i+1}} \frac{\partial}{\partial x} N_i \frac{\partial}{\partial x} N_{i+1} \, dx & \int_{x_i}^{x_{i+1}} \frac{\partial}{\partial x} N_{i+1} \frac{\partial}{\partial x} N_{i+1} \, dx \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \int_{x_i}^{x_{i+1}} f(x) N_i \, dx \\ \int_{x_i}^{x_{i+1}} f(x) N_{i+1} \, dx \end{pmatrix} \tag{3.17}$$

Solving the integrals we obtain:

$$\underbrace{\left(\frac{1}{l_i} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \right)}_{K_i} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \int_{x_i}^{x_{i+1}} f(x) N_i \, dx \\ \int_{x_i}^{x_{i+1}} f(x) N_{i+1} \, dx \end{pmatrix} \tag{3.18}$$

Summing up these local systems of equations gives the global system of linear equations (see Fig. 3.3).

$$\mathbf{K}\mathbf{u} = \mathbf{f} \tag{3.19}$$

3.3 Example: The Finite Volume method

The original idea for the finite volume methods came from the conservation laws written in integral form. But as already mentioned in subsection 3.1.3 it can also be interpreted as subdomain collocation. If we look at the heat equation again we get the following integral form with subdomains $\Omega_i = [x_i, x_{i+1}]$:

$$\int_\Omega \frac{\partial^2 u}{\partial x^2} \chi_{\Omega_i}\, dx = \int_\Omega f \chi_{\Omega_i}\, dx \quad \forall i \tag{3.20}$$

Using the properties of the characteristic function these integrals can be written as:

$$\int_{x_i}^{x_{i+1}} \frac{\partial^2 u}{\partial x^2} \cdot 1\, dx = \int_{x_i}^{x_{i+1}} f\, dx \tag{3.21}$$

With partial integration we obtain:

$$\left[\frac{\partial u}{\partial x}\right]_{x_i}^{x_{i+1}} - \underbrace{\int_{x_i}^{x_{i+1}} \frac{\partial u}{\partial x} \cdot 1'\, dx}_{=0} = \int_{x_i}^{x_{i+1}} f\, dx \tag{3.22}$$

It follows directly that:

$$\frac{\partial u}{\partial x}(x_{i+1}) - \frac{\partial u}{\partial x}(x_i) = \int_{x_i}^{x_{i+1}} f\, dx \tag{3.23}$$

This equation represents the original idea of the finite volume method. On the left side it has the flux $\partial u/\partial x$ on both sides of the small subdomain Ω_i (this subdomain is called *control volume* in the Finite Volume method) and the source term on the right hand side. So what goes into the control volume and does not go out must be equal to the amount coming from the source term f.

Inserting locally defined piecewise linear functions which have the same boundarys as the subdomains Ω_i we get the following result (shown for Ω_1):

$$(u_1 \frac{\partial N_1}{\partial x}(x_2) + u_2 \frac{\partial N_2}{\partial x}(x_2)) - (u_1 \frac{\partial N_1}{\partial x}(x_1) + u_2 \frac{\partial N_2}{\partial x}(x_1)) = \int_{x_1}^{x_2} f\, dx \tag{3.24}$$

$$\Rightarrow 0 = \int_{x_1}^{x_2} f\, dx \tag{3.25}$$

It is clear that Eq. (3.24) is not very helpful. One possible way to get around this problem is to put the control volume boundarys not onto the nodes of the ansatzfunctions but to put them around the nodes (see Fig. 3.4).

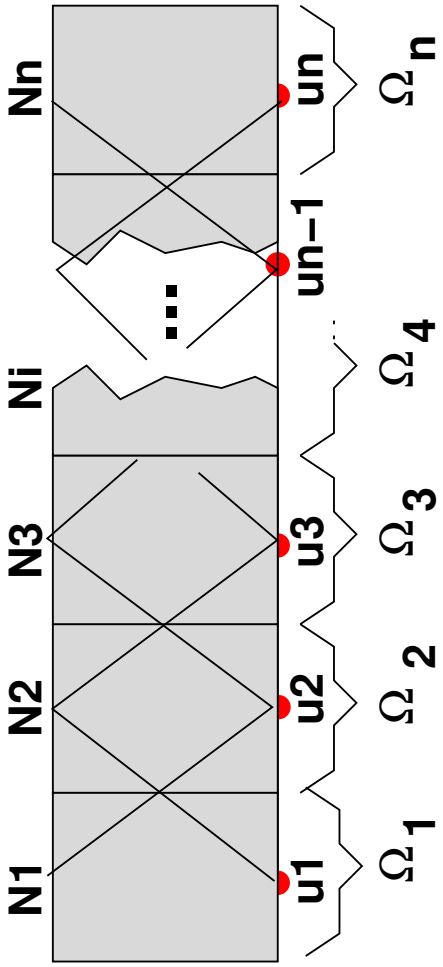


Figure 3.4: Position of the subdomains Ω_i for the FVM

With this ansatz we get the following equations for a control volume Ω_i inside the domain Ω:

$$\begin{aligned}(u_{i-1}\frac{\partial N_{i-1}}{\partial x}(x_{i+1}) + u_i\frac{\partial N_i}{\partial x}(x_{i+1}) + u_{i+1}\frac{\partial N_{i+1}}{\partial x}(x_{i+1}))- \\ (u_{i-1}\frac{\partial N_{i-1}}{\partial x}(x_i) + u_i\frac{\partial N_i}{\partial x}(x_i) + u_{i+1}\frac{\partial N_{i+1}}{\partial x}(x_i)) = \int_{x_1}^{x_2} f\,dx\end{aligned} \tag{3.26}$$

Looking at Fig. 3.4 it is easy to find the appropriate values for the derivatives (assuming the nodes of the ansatzfunctions are equidistant):

$$\begin{aligned}((u_{i-1}\cdot 0) + (u_i\cdot -\frac{1}{l}) + (u_{i+1}\cdot\frac{1}{l}))- \\ ((u_{i-1}\cdot -\frac{1}{l}) + (u_i\cdot\frac{1}{l}) + (u_{i+1}\cdot 0)) = \int_{x_1}^{x_2} f\,dx\end{aligned} \tag{3.27}$$

Finally we get:

$$\frac{1}{l}(u_{i-1} - 2u_i + u_{i+1}) = \int_{x_i}^{x_{i+1}} f\,dx \tag{3.28}$$

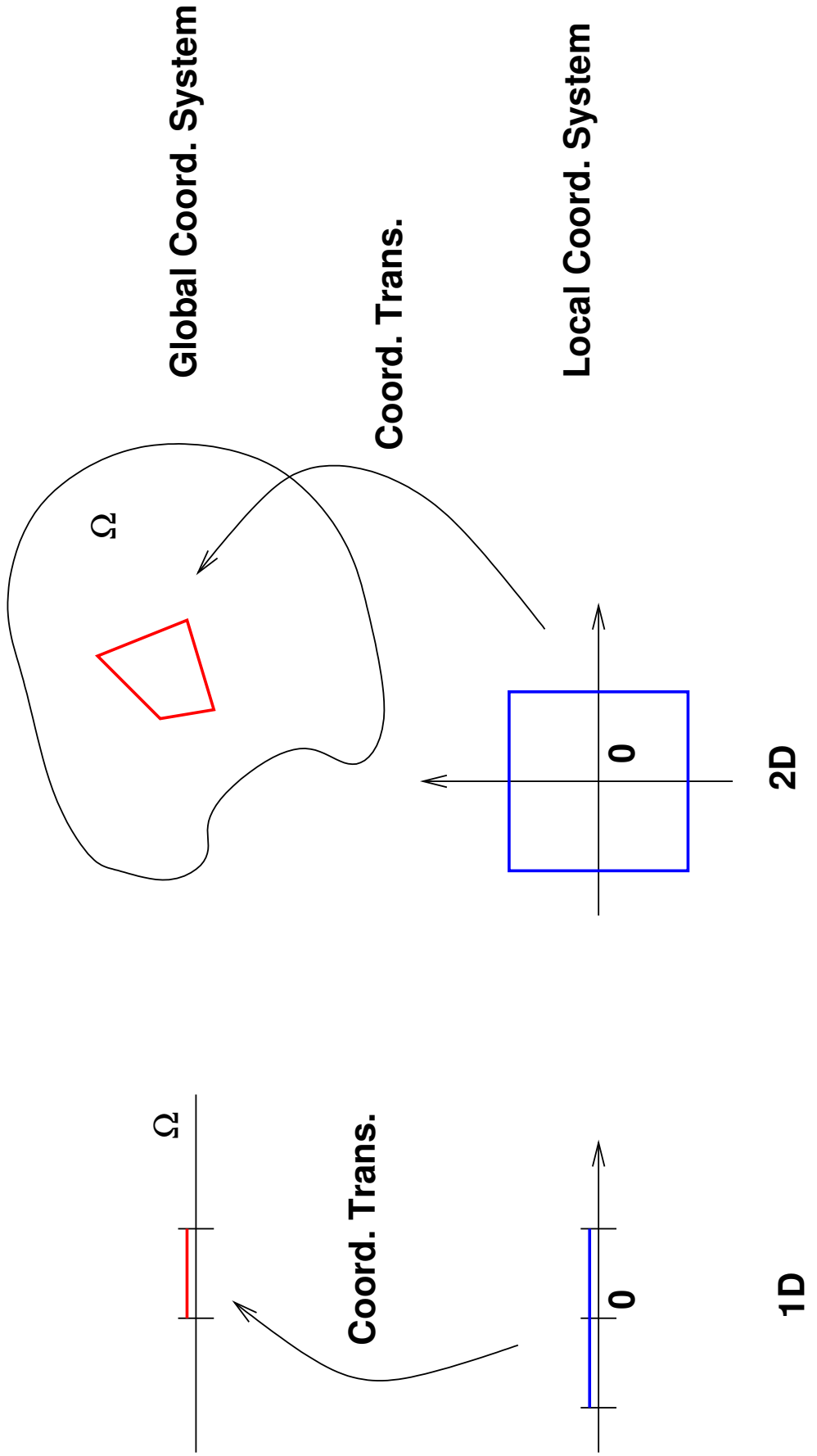
which is exactly the same system of equations as in the finite difference method.

3.4 Higher dimensional elements

In one dimension the advantages of the Finite Element method seem not really overwhelming. But already in two dimensions it is possible to model complex geometries without difficulties which cannot be handled anymore by the finite difference method. The next sections will cover the basic ideas to create finite elements of arbitrary spatial dimension and arbitrary high order although naturally most time the dimension will be less than four and higher order elements do not always have advantages.

3.4.1 Isoparametric mapping

For the simple 1D elements it was easy to find the ansatzfunctions on an element directly in the *global coordinate system*. In more dimensions this task becomes quite difficult. One solution is to define ansatzfunctions on a convenient domain and to introduce a coordinate transformation from this domain or *local coordinate system* to the global coordinate system (see Fig. 3.5).


Global Coord. System
Coord. Trans.
Local Coord. System
Ω
0
2D
Ω
Coord. Trans.
0
1D

The two most used intervals for the local coordinate system are either the interval $[-1 \dots 1]$ or $[0 \dots 1]$. In higher dimensions the products of these intervals are used. It is also clear that these intervals define lines, quadrilaterals and cubes in 1,2 and 3 dimensions. For triangular elements slightly different domains are used.

In this lecture note the interval $[-1 \dots 1]$ will be used. For 1D elements we get the following ansatzfunctions on the local coordinate system:

$$N_1(\xi) = \frac{1}{2}(1-\xi) \tag{3.29}$$

$$N_2(\xi) = \frac{1}{2}(1+\xi) \tag{3.30}$$

Often this interval together with the ansatzfunctions is called *Master-* or *Urelement* because it is the basis to derive all local elements.

Now a coordinate transformation from the interval $[-1 \dots 1]$ to an arbitrary interval $[x_i \dots x_{i+1}]$ is required. The class of *isoparametric elements* uses the same ansatzfunctions for the coordinate transformation. Other choices are the ansatzfunctions of lower order (lower polynomial degree) which then give *subparametric* elements or ansatzfunctions of higher order which result in *superparametric* elements. The latter two element classes can cause trouble and thus are not used very often. For the isoparamtric elements we then get the following coordinate transformation from the Masterelement to the element i with the coordinates x_i, x_{i+1} in the global coordinate system:

$$x_{glob}(\xi) = x_i h_1(\xi) + x_{i+1} h_2(\xi) \tag{3.31}$$

Going back to the weak form of the heat equation we had the following equation for the element stiffness matrix K:

$$\mathbf{K}_{ij} = \int_{x_i}^{x_{i+1}} \frac{\partial N_j}{\partial x} \cdot \frac{\partial N_i}{\partial x}\, dx \quad i,j \in [1,2] \tag{3.32}$$

Inserting the coordinate transformation we get:

$$\mathbf{K}_{ij} = \int_{-1}^{1} \left(\frac{\partial N_j(x_{glob}(\xi))}{\partial x} \cdot \frac{\partial N_i(x_{glob}(\xi))}{\partial x} \right) \left| \frac{dx_{glob}(\xi)}{d\xi} \right| d\xi \quad i,j \in [1,2] \tag{3.33}$$

One little problem remains in Eq. (3.33). The partial derivatives of the ansatzfunctions are still with respect to the global coordinate system. With the chain rule we obtain the following equation:

$$\frac{\partial N}{\partial \xi} = \frac{\partial N}{\partial x}(x_{glob}(\xi)) \cdot \frac{\partial x_{glob}}{\partial \xi} \Leftrightarrow \left(\frac{\partial x_{glob}}{\partial \xi} \right)^{-1} \frac{\partial N}{\partial \xi} = \frac{\partial N}{\partial x}(x_{glob}(\xi)) \cdot \tag{3.34}$$

Inserting this into Eq. (3.33) gives finally the integral equation for one element stiffness matrix on the master element:

$$\mathbf{K}_{ij} = \int_{-1}^{1} \left(\left(\frac{\partial x_{glob}}{\partial \xi} \right)^{-1} \frac{\partial N_j}{\partial \xi} \cdot \left(\frac{\partial x_{glob}}{\partial \xi} \right)^{-1} \frac{\partial N_i}{\partial \xi} \right) \left| \frac{dx_{glob}(\xi)}{d\xi} \right| d\xi \qquad i,j \in [1,2] \tag{3.35}$$

Computing this integral shows that it is equivalent to the equation obtained by integrating in the global domain. For higher dimensions the integral equations on the master element are derived exactly the same way.

3.4.2 Quadrilateral elements

In higher dimensions ansatzfunctions which have only local support are again required to get sparse matrices. The simplest idea to get ansatzfunctions is thus to use the same functions as in 1D in each spatial direction. Doing this we get the following ansatzfunctions on the master element $[-1 \ldots 1] \times [-1 \ldots 1]$ (see Fig. 3.6) :

$$N_1(\xi,\eta) = \frac{1}{4}(\xi-1)(\eta-1) \tag{3.36}$$

$$N_2(\xi,\eta) = \frac{1}{4}(\xi+1)(\eta-1) \tag{3.37}$$

$$N_3(\xi,\eta) = \frac{1}{4}(\xi+1)(\eta+1) \tag{3.38}$$

$$N_4(\xi,\eta) = \frac{1}{4}(\xi-1)*\eta+1) \tag{3.39}$$

They look similar to a pyramid around a node (see Fig. 3.7). The isoparametric coordinate transformation then becomes:

$$\begin{pmatrix} x_{glob} \\ y_{glob} \end{pmatrix}(\xi,\eta) = N_1(\xi,\eta) \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + N_2(\xi,\eta) \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + N_3(\xi,\eta) \begin{pmatrix} x_3 \\ y_3 \end{pmatrix} + N_4(\xi,\eta) \begin{pmatrix} x_4 \\ y_4 \end{pmatrix} \tag{3.40}$$

where $x_1, \ldots, y_4$ are the global coordinates of the corner nodes of the quadrilateral. Some difficulties appear when going to higher dimensions. Again the heat equation should illustrate the use of the coordinate transformation. In 2D we have for the element stiffness matrix:

$$\mathbf{K}_{ij} = \int_{\Omega_{elm}} \begin{pmatrix} \frac{\partial N_j}{\partial x} \\ \frac{\partial N_j}{\partial y} \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{pmatrix} d\Omega_{elm} \quad i,j \in [1,\ldots,4] \tag{3.41}$$

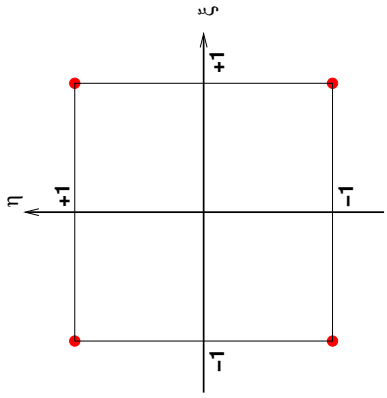


Figure 3.6: Masterelement for quadrilaterals

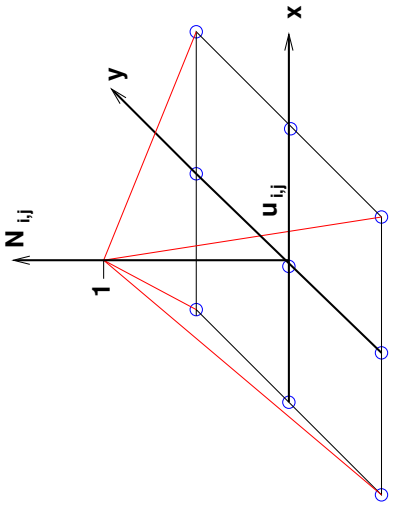


Figure 3.7: Schematic view of the Ansatzfunction $N_{i,j}$

Inserting the coordinate transformation we get:

$$\begin{aligned}\mathbf{K}_{ij} = \int_{-1}^{1}\int_{-1}^{1} \begin{pmatrix} \frac{\partial N_j}{\partial x} \\ \frac{\partial N_j}{\partial y} \end{pmatrix} (x_{glob}(\xi,\eta), y_{glob}(\xi,\eta)) \cdot \\ \begin{pmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{pmatrix} (x_{glob}(\xi,\eta), y_{glob}(\xi,\eta)) \, |\mathbf{J}(\xi,\eta)| \, d\xi \, d\eta \quad i,j \in [1,\ldots,4]\end{aligned} \tag{3.42}$$

Here $|\mathbf{J}|$ should denote the determinant of $\mathbf{J}$, which is the Jacobian of the coordinate transformation:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x_{glob}}{\partial \xi} & \frac{\partial x_{glob}}{\partial \eta} \\ \frac{\partial y_{glob}}{\partial \xi} & \frac{\partial y_{glob}}{\partial \eta} \end{pmatrix} \tag{3.43}$$

Now we can again apply the chain rule to the spatial derivatives of the ansatzfunctions in the masterelement:

$$\frac{\partial N}{\partial \xi} = \frac{\partial N}{\partial x}\frac{\partial x_{glob}}{\partial \xi} + \frac{\partial N}{\partial y}\frac{\partial y_{glob}}{\partial \xi} \tag{3.44}$$

$$\frac{\partial N}{\partial \eta} = \frac{\partial N}{\partial x}\frac{\partial x_{glob}}{\partial \eta} + \frac{\partial N}{\partial y}\frac{\partial y_{glob}}{\partial \eta} \tag{3.45}$$

With the Jacobian $\mathbf{J}$ it can be written more compact:

$$\begin{pmatrix} \frac{\partial N}{\partial \xi} \\ \frac{\partial N}{\partial \eta} \end{pmatrix} = \mathbf{J}^T \begin{pmatrix} \frac{\partial N}{\partial x} \\ \frac{\partial N}{\partial y} \end{pmatrix} \tag{3.46}$$

Bringing the Jacobian to the left side gives:

$$\mathbf{J}^{-T} \begin{pmatrix} \frac{\partial N}{\partial \xi} \\ \frac{\partial N}{\partial \eta} \end{pmatrix} = \begin{pmatrix} \frac{\partial N}{\partial x} \\ \frac{\partial N}{\partial y} \end{pmatrix} \tag{3.47}$$

So the derivatives with respect to the global coordinate system in Eq. (3.42) can be replaced by derivatives in the local coordinate system:

$$\mathbf{K}_{ij} = \int_{-1}^{1} \int_{-1}^{1} \mathbf{J}^{-T} \begin{pmatrix} \frac{\partial N_j}{\partial \xi} \\ \frac{\partial N_j}{\partial \eta} \end{pmatrix} \cdot \mathbf{J}^{-T} \begin{pmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{pmatrix} |\mathbf{J}| \, d\xi \, d\eta \quad i,j \in [1,\ldots,4] \tag{3.48}$$

Higher dimensional elements can be treated in the same way. One point causing some trouble in practical implementations is the term $\mathbf{J}^{-T}$. It implies some requirements for the coordinate transformation. First the Jacobian must always and everywhere be invertible. Furthermore a Jacobian with negative or zero determinant should be avoided.

A common problem in that context is the wrong ordering of the nodes Eq. (3.40). For 2 dimensional quadrilaterals the nodes in the global coordinate system must be ordered counterclockwise to have a positive determinant of the Jacobian.

Another cause for a negative Jacobian can be a highly distorted element where the angle at one corner is greater than 180 degrees. Sometimes this can happen together with automatic mesh deformation.

3.4.3 Triangular elements

The other fundamental element type beside the quadrilateral and its higher dimensional relatives is the triangular element. It was also the first finite element ever. Isoparametric mapping can be used for the triangular elements as well. The ansatzfunctions in the master element (see Fig. 3.8) are:

$$N_1(\xi,\eta) = \xi \tag{3.49}$$
$$N_2(\xi,\eta) = \eta \tag{3.50}$$
$$N_3(\xi,\eta) = 1-\xi-\eta \tag{3.51}$$

For the isoparametric coordinate transformation we get:

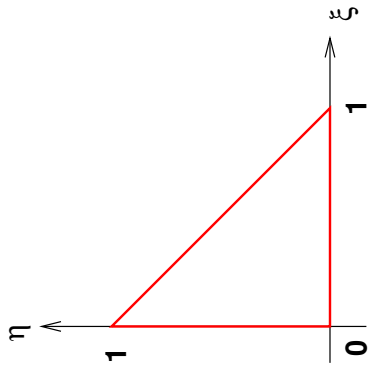


Figure 3.8: Masterelement for triangular elements

$$\begin{pmatrix} x_{glob} \\ y_{glob} \end{pmatrix}(\xi,\eta) = N_1(\xi,\eta)\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + N_2(\xi,\eta)\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + N_3(\xi,\eta)\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} \tag{3.52}$$

The element stiffness matrix can then be derived the same way as shown for the quadrilateral. From the numerical point of view the quadrilateral elements achieve a higher accuracy with the same number of nodes. In mechanical system the triangular elements also tend to be to stiff. Nevertheless in several areas triangular elements are still used because they have some advantages. First thing is that they are quite robust. This means they do not fail numerically when they undergo large deformations. If they become degenerated they loose accuracy but they don't cause trouble like the quadrilaterals, which cannot withstand inner angles greater than 180 degrees. Another advantage is the availability of powerful automatic mesh generators. Research is going on in the field of mesh generation tools for quadrilaterals or cubes, but the automatic generation of triangular or tetrahedral meshes is still more powerful and robust.

3.4.4 Higher order elements

Consider a triangulation[1] of an arbitrary domain Ω where the partial differential equation should be solved. The accuracy of the approximate solution which can be computed with the finite element method depends on the size of the elements which are used in the discretisation. To describe this size, the diameter of the smallest circle that completely covers the element is used in 2D. For 3D elements it is the diameter of the smallest ball. The diameter will be named h.

Let the error between the exact solution u and the finite element approximation u_h be measured in the W_1^2 norm:

[1]The word triangulation is used for general element patterns which are used to discretise a domain. It does not always mean that the discretisation uses triangles

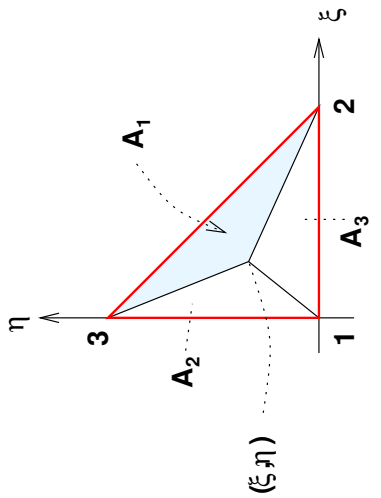


Figure 3.9: Area coordinates for triangular elements

$$||u - u_h||_1^2 = \int_\Omega ||\nabla(u - u_h)||^2 + \int_\Omega |u - u_h|^2 \tag{3.53}$$

For the Laplacian, the following estimate for the error $||u - u_h||_1^2$ can be found:

$$||u - u_h||_1^2 \leq C \cdot h^p \tag{3.54}$$

where C is a constant and p depends on the order of the ansatzfunctions. From Eq. (3.54) it can be seen that the error can be reduced either by increasing the number of elements and thus reducing h or by increasing the order of the ansatzfunctions p. In the next part methods to get elements with high order ansatzfunctions will be shown for triangles and quadrilaterals.

Triangles

Another convenient way to write the ansatzfunctions for triangles is in terms of area coordinates. These are defined as the quotient of the area of the triangles, which can be constructed from a point inside the triangle, and the area of the complete triangle (see Fig. 3.9):

$$L_j = \frac{A_j}{A_{tot}} \tag{3.55}$$

where A_j denotes the area of triangle A_j in Fig. 3.9. With these functions the ansatzfunctions in the triangle can easily be written as:

$$N_1(\xi,\eta) = 1-\xi\eta = L_1(\xi,\eta) \tag{3.56}$$
$$N_2(\xi,\eta) = \xi = L_2(\xi,\eta) \tag{3.57}$$
$$N_3(\xi,\eta) = \eta = L_3(\xi,\eta) \tag{3.58}$$

To get a higher order element it is necessary to put some new nodes into the element. For the next step, the midpoints of the edges of the triangle are a good choice. The ansatzfunctions on these points must be constructed such that they are zero on all other nodes and one at that point. For the fourth node, which should be located between node 1 and node 2, the product of L_1 and L_2 satisfies this conditions. Both are zero at node 3 and at node 1 or 2 one of these functions vanishes. At node 4 L_1 and L_2 are $1/2$ so a correction factor of must also be added. Hence:

$$N_4(\xi,\eta) = 4\cdot L_1(\xi,\eta)\cdot L_2(\xi,\eta) \tag{3.59}$$

The ansatzfunctions for node 5 and 6 can be constructed similarly. After that some corrections must be applied to the old functions N_1 to N_3 because they must now become zero on the additional nodes 4 to 6. This can be done by subtracting the newly created functions N_4 to N_6.

Pascal's triangle can be used to determine the number and position of the nodes in advance. It includes all the terms which appear in the $(x+y)^n$. In Fig. 3.10 the relation can be seen.

Lagrange basis

It was easy to derive the quadrilateral and hexahedral elements from the 1D ansatzfunctions by simply taking the products of these function. To get higher order quadrilaterals it is therefore only necessary to look at the 1D elements. On these elements the ansatzfunctions of arbitrary order can be computed using the Lagrange interpolation formulas, which give also the name for this basis:

$$l_k(\xi) = \frac{\prod_{j\neq k}(\xi-\xi_j)}{\prod_{j\neq k}(\xi_k-\xi_j)} \tag{3.60}$$

Here the ξ_k are the interpolation points or the nodal points in the finite element language. The $l_k(\xi)$ is zero on all nodal points except for the kth where it is exactly one. For quadratic elements with nodes at $-1, 0, 1$ in the masterelement we get :

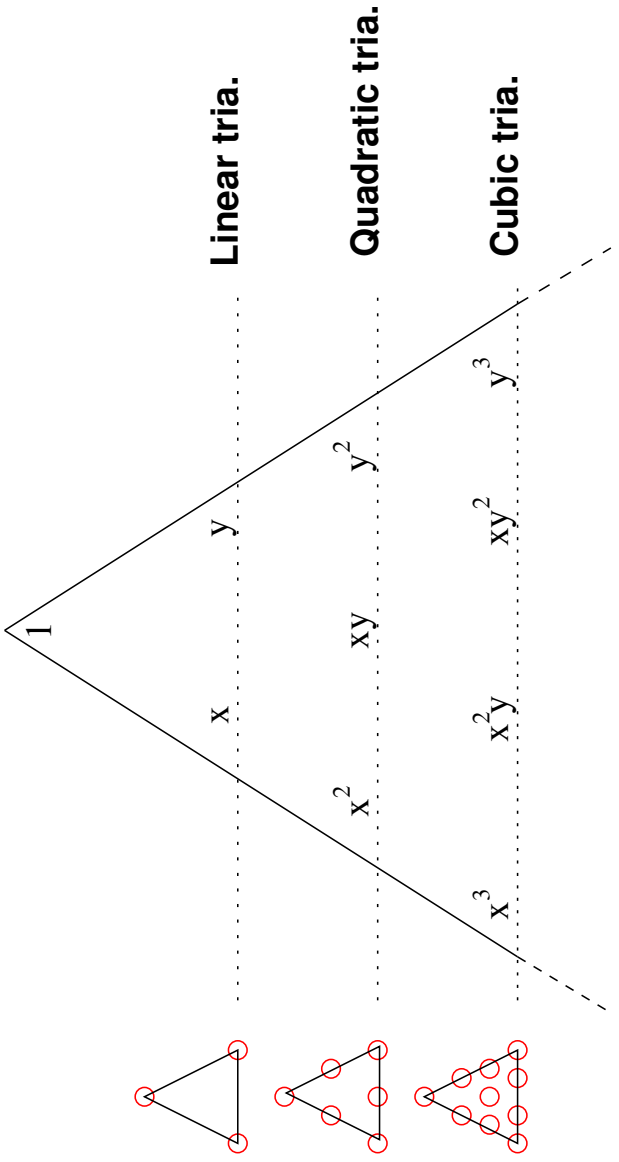


Figure 3.10: Pascal's triangle

$$N_1^{lin}(\xi) = \frac{(\xi-0)(\xi-1)}{(-1-0)(-1-1)} = \frac{1}{2}(\xi^2-\xi) \tag{3.61}$$

$$N_2^{lin}(\xi) = \frac{(\xi+1)(\xi-1)}{(0-1)(0+1)} = 1-\xi^2 \tag{3.62}$$

$$N_3^{lin}(\xi) = \frac{(\xi-0)(\xi+1)}{(1+1)(1-0)} = \frac{1}{2}\xi^2+\xi \tag{3.63}$$

With these functions the ansatzfunctions in the quadrilateral masterelement become:

$$N_1^{quad}(\xi,\eta) = N_1^{lin}(\xi)\cdot N_1^{lin}(\eta) \tag{3.64}$$

$$N_2^{quad}(\xi,\eta) = N_1^{lin}(\xi)\cdot N_2^{lin}(\eta) \tag{3.65}$$

$$N_3^{quad}(\xi,\eta) = N_1^{lin}(\xi)\cdot N_3^{lin}(\eta) \tag{3.66}$$

$$N_4^{quad}(\xi,\eta) = N_2^{lin}(\xi)\cdot N_1^{lin}(\eta) \tag{3.67}$$

$$\vdots \quad \vdots \tag{3.68}$$

At last some remarks about the higher order elements. In most finite element codes quadratic elements will be the highest order elements available. One point is that beside being more accurate higher order elements are much more expensive. That means they need more computational time. One reason is the higher number of nodes (a quadratic hexahedron has already 27 nodes). Most elements cannot be evaluated analytically anymore, so numerical integration formulas are used. These formulas must also become more accurate and thus expensive, if the ansatzfunctions have higher order. So at a certain order the theoretical benefits of higher elements are eaten up by their higher numerical costs.

Another disadvantage is that the elements become numerically less robust. So moving the mid node on the edges to far away from the geometrical centre of the edge can cause a failure of the isoparametric mapping and the element.

3.5 Time dependent problems

For time dependent problems the weighted residual methods can be used exactly the same way as for stationary problems. Consider the instationary heat equation:

$$\dot{u} - \Delta u = f \tag{3.69}$$

together with boundary conditions and initial conditions. Applying a weighted residual method we get:

$$\int_\Omega \dot{u}\varphi \, d\Omega + \int_\Omega (\nabla u)^T \cdot \nabla\varphi \, d\Omega = \int_\Omega f\varphi \, d\Omega \quad \forall\varphi \tag{3.70}$$

From Eq. (3.70) two methods for the time discretisation can be derived. One is the time-space finite element method, which will not be treated here and the other is the method of lines which separates time- and space discretisation. Looking at the approximation of u:

$$u \approx u_h = \sum_{i=1}^{N} u_i N_i \tag{3.71}$$

it is clear that also:

$$\dot{u} \approx \dot{u}_h = \sum_{i=1}^{N} \dot{u}_i N_i \tag{3.72}$$

holds. Inserting this ansatz into Eq. (3.70) allows us to transform the instationary partial differential equation into a system of ODEs for the coefficients u_i.

$$\sum_{i=1}^{N} \left[\underbrace{\left(\int_\Omega N_j N_i \, d\Omega \right)}_{\mathbf{M}_{ij}} \dot{u}_i + \underbrace{\left(\int_\Omega (\nabla N_j)^T \cdot \nabla N_i \, d\Omega \right)}_{\mathbf{K}_{ij}} u_i \right] = \underbrace{\left(\int_\Omega N_j f \, d\Omega \right)}_{\mathbf{f}_j(t)} \quad \forall j \tag{3.73}$$

Writing this system in matrix form we obtain:

$$\mathbf{M}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) + \mathbf{f}(t) \Rightarrow \dot{\mathbf{u}} = -\mathbf{M}^{-1}\mathbf{K}\mathbf{u} + \mathbf{M}^{-1}\mathbf{f} \tag{3.74}$$

So instead of a system of linear equations for the discretisation of a stationary problem we get a system of ordinary differential equations. This is called a *semidiscretisation* because it discretises only the spatial directions. Often the matrix M is called the *mass matrix*. This name stems from the analysis of mechanical systems, where the matrix M is related to the mass of a mechanical system.

In most cases another numerical method is required to find a solution which satisfies the system of ordinary differential equations. One possible choice is the Euler forward method:

$$\mathbf{u}(t + \Delta t) \;=\; \mathbf{u}(t) + \Delta t(\mathbf{M}^{-1}\mathbf{K}\mathbf{u}(t) + \mathbf{M}^{-1}\mathbf{f}(t)) \tag{3.75}$$

Although the Euler method is an explicit method for this system it involves the solution of system of linear equations (instead of computing the inverse of M which should never be done in real applications). So the disadvantages of the explicit Euler method stay, while

the advantage of not having to solve a system of equations is lost. To circumvent this problem often a *lumped* mass matrix is used instead of the correct matrix. The lumped matrix is a diagonal matrix which is easy to invert. Its diagonal elements are simply the sum of all entries in the row of the diagonal element.

$$\mathbf{M}_L = diag(m_i), \qquad m_k = \sum_{j=1}^{N} \mathbf{M}_{ij} \tag{3.76}$$

Because it is known that the mass matrix is responsible for the inertia the error in the description of the physical system is tolerable for many applications.

Chapter 4

Hyperbolic equations

In the first chapter the Fourier's law for heat transport or diffusion processes was introduced. A slight variation of this equation was the transport equation which describes convective heat transport. The difference between this two equations might seem small but for the numerical treatment it is quite important. Similar equations also appear in many other physical phenomena. Examples like the wave equation, the telegraph equation and the transport equation will be shown. After that some properties of the solutions of hyperbolic equations will be analysed. Finally finite difference schemes to find an approximate solution will be shown.

4.1 Introduction

Many physical phenomena like sound and electromagnetic fields need to be modelled with waves. Thus the wave equation:

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 0 \Leftrightarrow \ddot{u} - \Delta u = 0 \tag{4.1}$$

is one prototype of a hyperbolic equation. Another one comes from transport processes, as shown in the heat equation with convective transport:

$$\frac{\partial u}{\partial t} - \beta^2 \Delta u + (v^T \cdot \nabla u) = f \tag{4.2}$$

where v^T is a prescribed velocity field. In the extreme case $\beta = 0$, which describes a pure convective transport, the equation becomes the transport equation (shown in 1D):

$$\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} = 0 \tag{4.3}$$

Another example is an elastic string (in a piano, or a guitar). If $u(x,t)$ is the displacement of the string at a certain point the acceleration is $\ddot{u}(x,t)$. According to Newton's the law the force is then $-\rho\frac{\partial^2 u}{\partial t^2}$ where ρ is the density of the string. Assuming small displacements the force from the elastic deformation is $-T\frac{\partial^2 u}{\partial x^2}$ with T being a material constant describing the strength of the string. Putting these terms together with an external force term it follows that:

$$-\rho\frac{\partial^2 u}{\partial t^2} - T\frac{\partial^2 u}{\partial x^2} = f \tag{4.4}$$

So the motion of elastic string can also be described by the wave equation.

4.1.1 Telegraph equation

Now we will consider an example from electrical engineering. It is called the *telegraph equation* because it models the behaviour of electrical signals on a telegraph line. The first step in building the model is to replace a small piece of the wire by a quadrupole build from resistors, capacitors and coils (see Fig. 4.1). Letting the size of this piece go to zero we obtain a partial differential equation describing the behaviour of signals on the wire.

Using Kirchhoff's laws we obtain the following two equation for the quadrupole:

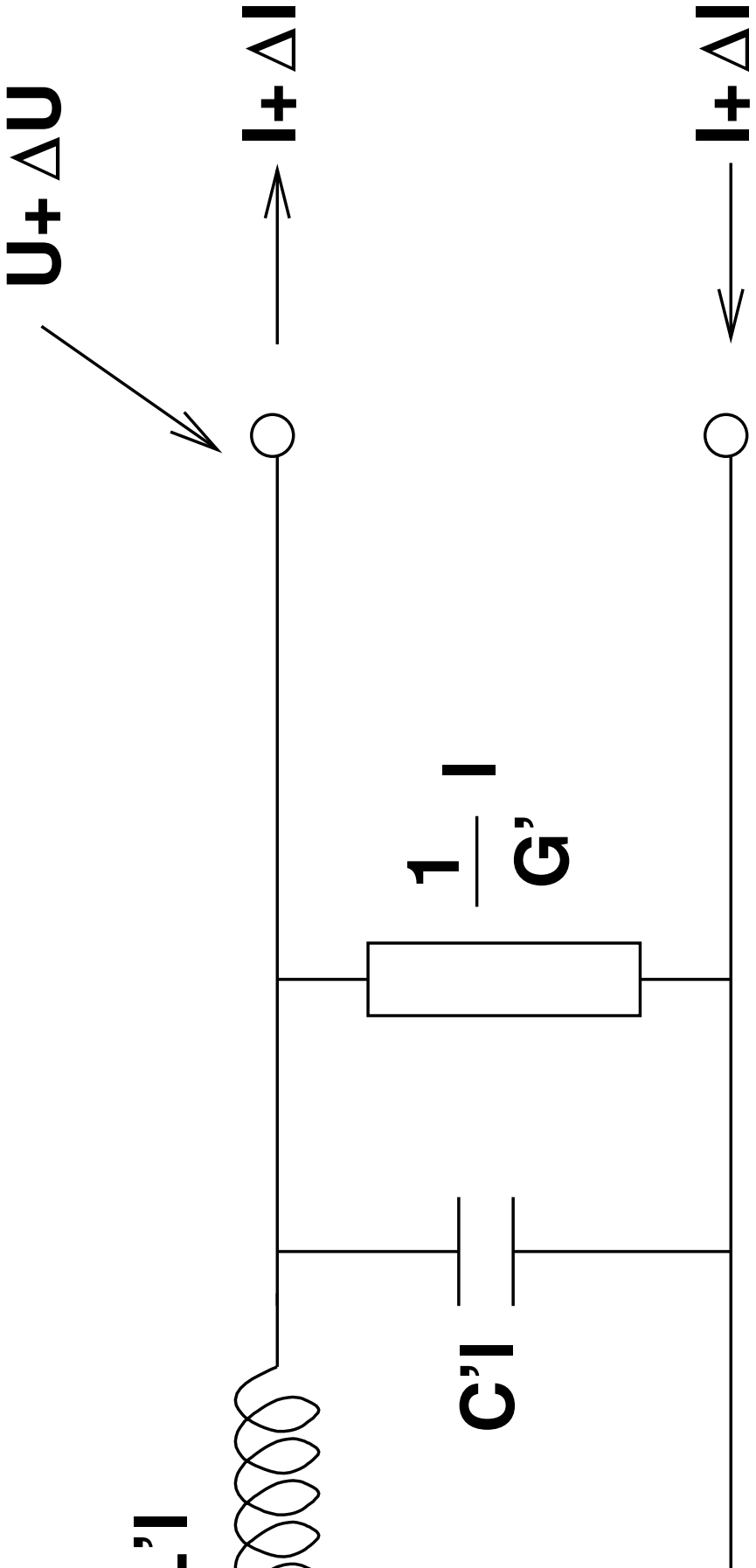
$$\begin{aligned} -U + I\cdot R'l + \frac{dI}{dt}L'\cdot l + U + \Delta U &= 0 && (4.5)\\ I - I_c - I_g - I - \Delta I &= 0 && (4.6) \end{aligned}$$

The two currents at the capacitor C and the conductivity G can be expressed in terms of the voltage change:

$$\begin{aligned} I_c &= (C'l)\frac{d(U+\Delta U)}{dt} && (4.7)\\ I_G &= \frac{l}{G'}(U+\Delta U) && (4.8) \end{aligned}$$

Letting l go to zero and inserting Eq. (4.7) into Eq. (4.5) we get:

$$\begin{aligned} \frac{\partial U}{\partial x} &= -R'\cdot I - L'\frac{\partial I}{\partial t} && (4.9)\\ \frac{\partial I}{\partial x} &= -C\frac{\partial u}{\partial t} - S'U && (4.10) \end{aligned}$$


U+ ΔU
I+ ΔI
I+ ΔI
1 | G'
C'I
L'I

Here S' is a replacement for l/G'. Using matrix notation Eq. (4.9) and Eq. (4.10) can be written as:

$$\begin{bmatrix} C' & 0 \\ 0 & L' \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} U \\ I \end{bmatrix} = - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \frac{\partial}{\partial x} \begin{bmatrix} U \\ I \end{bmatrix} - \begin{bmatrix} S' & 0 \\ 0 & R' \end{bmatrix} \begin{bmatrix} U \\ I \end{bmatrix} \tag{4.11}$$

Multiplying Eq. (4.9) with the partial differential operator $\partial/\partial x$ and Eq. (4.10) with $\partial/\partial t$ gives:

$$\frac{\partial^2 U}{\partial x^2} = -R' \frac{\partial I}{\partial x} - L' \frac{\partial^2 I}{\partial x \partial t} \tag{4.12}$$

$$\frac{\partial^2 I}{\partial x \partial t} = -C' \frac{\partial^2}{\partial t^2} - S' \frac{\partial U}{\partial t} \tag{4.13}$$

Inserting Eq. (4.13) into Eq. (4.12) results in:

$$\frac{\partial^2 U}{\partial x^2} = S'R'U + C'R' \frac{\partial U}{\partial t^2} + S'L' \frac{\partial U}{\partial t} + C'L' \frac{\partial^2 U}{\partial t^2} \tag{4.14}$$

Sorting the terms and adding a source term $v(x, t$ we finally obtain:

$$\frac{\partial^2 U}{\partial t^2} + \left(\frac{R'}{L'} + \frac{S'}{C'} \right) \frac{\partial U}{\partial t} - \frac{1}{C'L'} \frac{\partial^2 U}{\partial x^2} + \frac{S'R'}{C'L'} U = v(x, t) \tag{4.15}$$

Looking at Eq. (4.15) shows that this equation is very similar to the wave equation. Two additional terms $c_1 U$ and $c_2 \partial U/\partial t$ are the only difference. The effect of these terms will be examined later. But the main result is that the propagation of signals on a wire can be seen as a wave phenomenom and thus be described by a hyperbolic equation.

4.1.2 Analytical solutions

Again the analysis of hyperbolic equations should be started with analytical solutions to these equations. Exponential functions in time and space should be a good first ansatz:

$$u(x, t) = A \cdot e^{pt} e^{ikx}, \quad A \neq 0 \tag{4.16}$$

where p and k are some constants. p describes the amplification of the solution in time, while k is the wave number of the solution. Higher k correspond to higher frequencies (perhaps for the telegraph equation the frequency of the input signal).

Transport equation

The partial derivatives of the ansatz with respect to t and x are:

$$\frac{\partial u}{\partial t} = p \cdot u, \qquad \frac{\partial u}{\partial x} = iku \tag{4.17}$$

Inserting these expressions into the transport equation Eq. (4.3), which is the simplest hyperbolic equation, we get:

$$p \cdot u - viku = 0 \Rightarrow p = ivk \tag{4.18}$$

So if Eq. (4.16) satisfies the transport equation the amplification factor p is purely imaginary. Hence it does not describe an amplification but is another wave length. Introducing the circular frequency ω it follows that:

$$i\omega = p = ivk \Rightarrow \omega = vk \tag{4.19}$$

From this relation we also get another form for the analytical solution which satisfies the transport equation:

$$u(x,t) = Ae^{i(\omega t}e^{ikx} = Ae^{i(\omega t+kx)} = Ae^{ik(vt+x)} \tag{4.20}$$

Looking into the spatial direction this solution is a trigonometric function or wave. On the other hand an observer standing at one point of the domain will see that the solution in time also is a wave. If the observer will move with the top of a wave the time and spatial wave must be in constant phase, which means $vt + x = 0$. Therefore the observer must choose his position such that:

$$x = -vt \Leftrightarrow x = c_p t \tag{4.21}$$

where c_p is the *phase velocity* which is in this case equal to $-v$.

Wave equation

Now we will take a look at the wave equation as another typical hyperbolic equation:

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \tag{4.22}$$

The partial derivatives are:

$$\frac{\partial^2 u}{\partial t^2} = (-i\omega)^2 Ae^{i(kx-\omega t)} = -\omega^2 Ae^{i(kx-\omega t)} \tag{4.23}$$

$$\frac{\partial^2 u}{\partial x^2} = (ik)^2 A e^{i(kx-\omega t)} = -k^2 A e^{i(kx-\omega t)} \tag{4.24}$$

Inserting these terms into Eq. (4.22) we obtain:

$$-\omega^2 \cdot u + c^2 k^2 \cdot u = 0 \Rightarrow \omega^2 = c^2 k^2 \Rightarrow \omega = \pm ck \tag{4.25}$$

Eq. (4.25) is called the *dispersion relation* of a wave. The dispersion describes the difference in speed of waves with different frequency. If $\omega/k = \text{const}$ holds, all waves travel with the same speed. So there is no dispersion. A signal build from several waves of different frequencies will travel along the domain unchanged.

Putting the dispersion relation into the ansatz we get for u:

$$u = A e^{i(kx \mp ckt)} \tag{4.26}$$

It can be seen that for the wave equation to phase speeds exist. One with positive sign and the other with negative sign. Information can travel from a point at time t into both directions with the same speed. But for the wave equation it is not possible that information travels faster than the phase speed. Thus it is possible to draw an area in the time space domain which can be influenced by the information at a given point in space and time (see Fig. 4.2).

Because for electromagnetic waves the speed of light is the phase speed this area of influence is often called the *lightcone*. Applying a binomial formula to the wave equation shows that it can be seen as two transport equations with different directions:

$$\left(\frac{\partial}{\partial t} - c\frac{\partial}{\partial x}\right)\left(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\right) u = 0 \tag{4.27}$$

Klein-Gordon equation

A slight variation of the pure wave equation is the Klein-Gordon equation. Although it has it origins in quantum physics it can also be interpreted as a string which oscillates in some foam which damps the oscillations:

$$\frac{\partial^2 u}{\partial t^2} u - c' \frac{\partial^2 u}{\partial x^2} + du = 0 \tag{4.28}$$

The term du is responsible for the damping. Deriving the dispersion relation shows that:

$$\omega = \pm\sqrt{c'k^2 + d} \tag{4.29}$$

Drawing this function together with the dispersion relation of the wave equation (see Fig. 4.3) shows that this time there is dispersion. So waves with longer wavelength travel

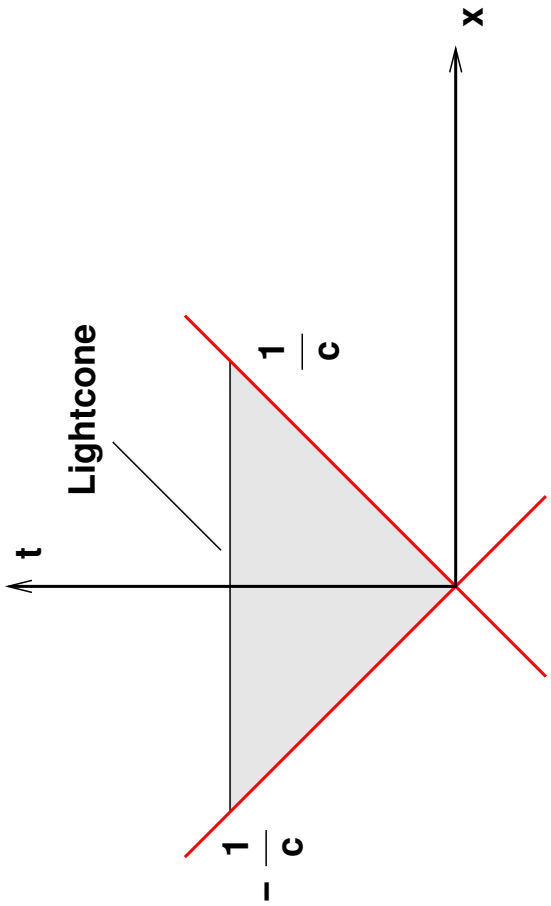


Figure 4.2: Lightcone of the wave equation

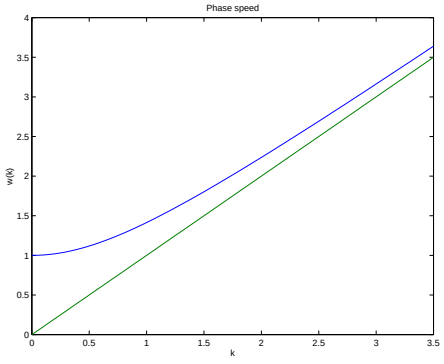


Figure 4.3: Dispersion relation of the Klein-Gordon equation (blue) compared with the dispersion of the wave equation (green)

slower than waves with shorter wavelength. A signal put into that system will become a different signal as time progresses. For the telegraph equation a similar result can be obtained. Therefore it is not possible to transfer information lossless over long distances. After some time a signal having rectangular shape would become unrecognisable.

Beam equation

Another equation which seems to be hyperbolic is the beam equation:

$$\rho \frac{\partial^2 u}{\partial t^2} + \frac{EI}{\rho} \cdot \frac{\partial^4 u}{\partial x^4} = 0 \tag{4.30}$$

Here EI denotes the elastic modulus and ρ is the density of the material while u is the displacement of the beam. Putting all material constants together in one constant a^2 gives:

$$\frac{\partial^2 u}{\partial t^2} + a^2 \cdot \frac{\partial^4 u}{\partial x^4} = 0 \tag{4.31}$$

The partial derivatives of the ansatz Eq. (4.16) are:

$$\frac{\partial^2 u}{\partial t^2} = -\omega^2 e^{i(kx-\omega t)} \tag{4.32}$$

$$\frac{\partial^4 u}{\partial x^4} = k^4 e^{i(kx-\omega t)} \tag{4.33}$$

With Eq. (4.31) we obtain:

$$-\omega^2 e^{i(kx-\omega t)} + a^2 k^4 e^{i(kx-\omega t)} = 0 \tag{4.34}$$

and hence:

$$w(k) = \pm a k^2 \tag{4.35}$$

So the transmission speed is not limited. It can become infinitely large if the frequency is high enough. Actually the equation is not really hyperbolic. It is a parabolic equation which only looks like a hyperbolic equation. For parabolic equations it is known that these allow infinite transmission speeds. But although it looks as if this observation can be used to achieve infinite transmission speeds with the help of beams, it is not possible because the model does not represent the real physics anymore if the frequencies become infinitely high.

4.1.3 Fourier series solution

Also for the hyperbolic equations it is possible to construct a solution for arbitrary initial conditions by a Fourier series approximation. As shown in the previous sections, the exponential function is a solution of the hyperbolic equations. So the integral over different wavenumbers must be also a solution of the hyperbolic equations:

$$\Phi(x) = \int \hat{\Phi}(k) e^{ikx}\, dk \tag{4.36}$$

With the dispersion relation for the equation the time dependent solution can be found:

$$u(x,t) = \sum_{\omega} \int \hat{\Phi}(k) e^{i(kx-\omega(k)t)}\, dk \tag{4.37}$$

It is the sum over the different branches of the dispersion relation. This solution will become quite useful for the stability analysis and for the derivation of the group speed.

4.1.4 D'Alambert's solution

In the previous section the ansatz function was always an exponential function. For the wave and transport equation another analytical solution exists. Taking the following initial conditions:

$$u(x,0) = u_0(x) = \Phi(x) \tag{4.38}$$

the transport equation has the following analytical solution:

$$u(x,t) = \Phi(x+vt) \tag{4.39}$$

For this solution we have the partial derivatives:

$$\frac{\partial}{\partial x}\Phi = \Phi'(x+vt) \tag{4.40}$$

$$\frac{\partial}{\partial t}\Phi = \Phi'(x+vt)v \tag{4.41}$$

Then the transport equation is with these derivatives:

$$v\Phi'(x+vt) - v\Phi'(x+vt) = 0 \quad \Leftrightarrow \quad 0 = 0 \tag{4.42}$$

Obviously this equation is always satisfied and thus Eq. (4.39) a solution of the transport equation. For the wave equation a similar result can be derived. Here the analytical solution is (with the same initial conditions as for the transport equation):

$$\alpha\Phi(x+ct)+\beta\Phi(x-ct) \tag{4.43}$$

For the second partial derivatives we can compute:

$$\frac{\partial^2 u}{\partial x^2} = \alpha\Phi(x+ct)+\beta\Phi(x+ct)) \tag{4.44}$$

$$\frac{\partial^2 u}{\partial t^2} = c^2(\alpha\Phi(x+ct)+\beta\Phi(x+ct)) \tag{4.45}$$

With this solution Eq. (4.22) becomes:

$$c^2(\alpha\Phi(x+ct)+\beta\Phi(x+ct))-c^2(\alpha\Phi(x+ct)+\beta\Phi(x+ct))=0\Leftrightarrow 0=0 \tag{4.46}$$

So Eq. (4.43) satisfies the wave equation. Although the result might seem trivial it is quite useful for the analysis of numerical schemes, because it offers a huge range of analytical solutions which can be used as a test case.

4.1.5 Characteristics of 1st order equations

A first order hyperbolic equation can be written in general form as:

$$a\frac{\partial u}{\partial \xi}+b\frac{\partial u}{\partial \eta}=c \tag{4.47}$$

Geometrically the function $u(\xi,\eta)$ describes a surface in a three dimensional vector space with dimensions u,ξ,η. A normal vector can thus be found in every point of the surface. From analysis it is known that this vector is:

$$\left(\frac{\partial u}{\partial \xi},\frac{\partial u}{\partial \eta},-1\right)^T \tag{4.48}$$

Using the normal scalar product Eq. (4.47) can be written as:

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix}^T \cdot \begin{pmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \\ -1 \end{pmatrix} = 0 \tag{4.49}$$

This allows another interpretation of the partial differential equation. Its solution is then the surface which normal vector is orthogonal to the coefficient vector of the partial differential equation. The idea for the *method of characteristics* is now to find a coordinate transformation which reduces the partial differential equation to an ordinary differential equation. Introducing the parameter s the coordinates become:

$$\xi(s), \eta(s), u(s) \tag{4.50}$$

The geometric interpretation is a line in the three dimensional space. We now choose that u should depend linearly on s:

$$\frac{du}{ds} = c \tag{4.51}$$

Writing u in terms of the coordinates $\xi(s), \eta(s)$ we get

$$u(\xi, \eta) = u(\xi(s), \eta(s)) \tag{4.52}$$

and thus (with the chain rule):

$$\frac{du}{ds} = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial s} + \frac{\partial u}{\partial \eta}\frac{\partial \eta}{\partial s} = c \tag{4.53}$$

Comparing this equation with Eq. (4.47)it is clear that:

$$\frac{d\xi}{ds} = a, \quad \frac{d\eta}{ds} = b \tag{4.54}$$

From these equations it can be seen that the coordinate transformation is a line in the ξ, η plane (at least for constant coefficients a and b). The steepness of this line with respect to time limits the speed at which signals can be transmitted. If $c = 0$ the solution u does not change along this line because $du/ds = c = 0$. So the initial conditions will be transported along the characteristic (see Fig. 4.4).

4.1.6 Group velocity

The dispersion relation showed the theoretical limits for the transmission of waves. Normally it is not very useful to send waves along media because they do not transport information. For practical purposes it is more important to known, how fast the maximal amplitude of a signal will travel along the domain. As already mentioned the solution of hyperbolic equations can be formulated in terms of a Fourier series as:

$$\int \hat{\Phi}(k) e^{ikx} \tag{4.55}$$

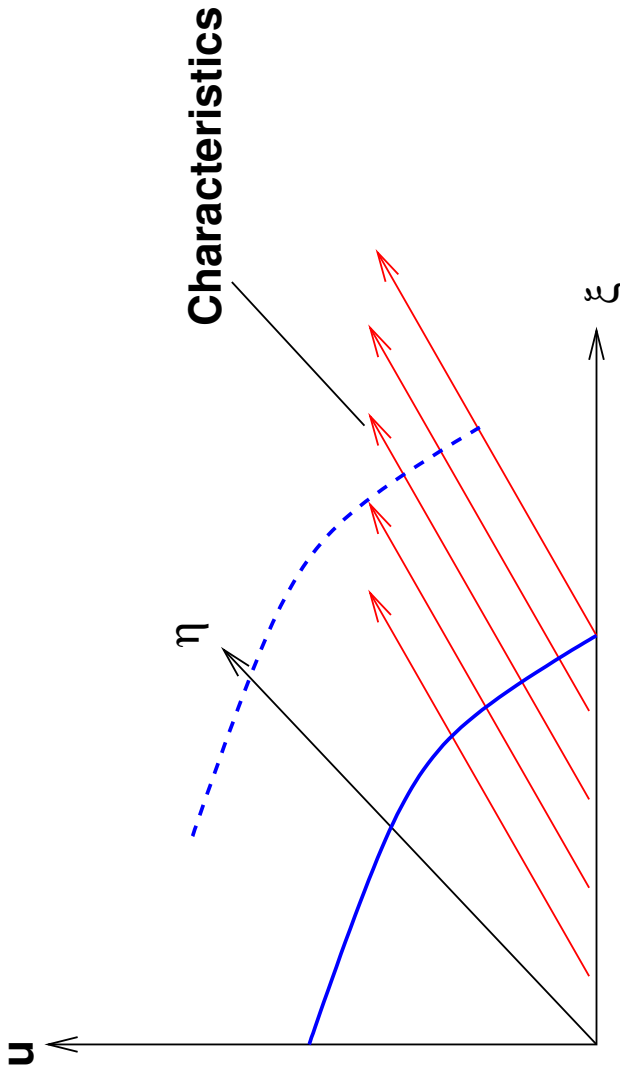


Figure 4.4: Transmission of initial conditions u_0 along the characteristic

Considering two waves with slightly different wavenumber the solution is:

$$e^{i(kx-\omega t)} + e^{i((k+\Delta k)x-(\omega+\Delta\omega)t))} \tag{4.56}$$

A short modification gives:

$$e^{i(kx-\omega t)}(\underbrace{1 + e^{i(\Delta kx-\Delta\omega t)}}_{=2 \quad \text{if} \quad (\Delta kx-\Delta\omega t)=0}) \tag{4.57}$$

So the condition, which must be satisfied at the point of maximal amplitude is $(\Delta kx - \Delta\omega t) = 0$. Bringing the speed x/t to the left side of the equation we obtain:

$$\frac{x}{t} = \frac{\Delta\omega}{\Delta k} \tag{4.58}$$

Letting the Δk go to zero we get the definition of the group speed:

$$\lim_{\Delta k} \frac{\Delta\omega}{\Delta k} = \frac{d\omega(k)}{dk} = c_{gr} \tag{4.59}$$

While the phase speed limits transmission of waves without information, the group speed limits the transfer of information. Applying this result to the wave equation with the dispersion relation $\omega(k) = \pm ck$ we obtain the following speeds:

$$c_{ph} = \frac{\omega(k)}{k} = \pm c, \qquad c_{gr} = \frac{d\omega(k)}{dk} = \pm c \tag{4.60}$$

Here the phase speed is equal to the group speed. The wave equation can thus transport information with the same speed as waves. Looking at d'Alamberts solution this is clear because arbitrary initial conditions are transported without any change. So the maximum of the solution will travel with the same speed as everything else. Taking again a look at the beam equation with the dispersion relation $\omega(k) = \pm ak^2$ we obtain the following speeds:

$$c_{ph} = \frac{\omega(k)}{k} = \pm ak, \qquad c_{gr} = \frac{d\omega(k)}{dk} = \pm 2ak \tag{4.61}$$

It is interesting that here the group speed is even higher than the phase speed. So the maximum travels faster than the waves itself. But as mentioned earlier this is due to the insufficient model which is not good to describe the wave phenomena in beams.

4.1.7 Eigenvector decomposition

Let $\mathbf{u}$ be a vector of time dependent functions:

$$\mathbf{u}(x,t) = \begin{bmatrix} u_1 \\ \vdots \\ u_d \end{bmatrix} \tag{4.62}$$

Then a multidimensional hyperbolic equation can be written as:

$$\frac{\partial}{\partial t}\mathbf{u} + \mathbf{A}\frac{\partial}{\partial x}\mathbf{u} = 0 \tag{4.63}$$

where $\mathbf{A} \in \mathbb{R}^d \times \mathbb{R}^d$ is a matrix. With a set of eigenvectors $\{e_1, \ldots, e_d\}$ and eigenvalues $\lambda_i, \ldots, \lambda_d$ the following equation must hold:

$$\mathbf{A}\mathbf{e}_j = \lambda_j \mathbf{e}_j \quad \forall j \in [1, \ldots, d] \tag{4.64}$$

By using the eigenvector basis it is possible to write the function vector $\mathbf{u}$ in terms of the eigenvectors:

$$\mathbf{u}(x,t) = \sum_{j=1}^{d} a_j(x,t)\mathbf{e}_j \tag{4.65}$$

For the partial derivatives we obtain:

$$\frac{\partial}{\partial t}\mathbf{u} = \sum_{j=1}^{d} \frac{\partial}{\partial t} a_j(x,t)\mathbf{e}_j \tag{4.66}$$

$$\frac{\partial}{\partial x}\mathbf{u} = \sum_{j=1}^{d} \frac{\partial}{\partial x} a_j(x,t)\mathbf{e}_j \tag{4.67}$$

Inserting these expressions into Eq. (4.63) the partial differential equation becomes:

$$\sum_{j=1}^{d} \frac{\partial}{\partial t} a_j(x,t)\mathbf{e}_j + \mathbf{A}\sum_{j=1}^{d} \frac{\partial}{\partial x} a_j(x,t)\mathbf{e}_j = 0 \tag{4.68}$$

Using Eq. (4.64) and sorting the terms gives:

$$\sum_{j=1}^{d} \left(\frac{\partial}{\partial t} a_j + \lambda_j \frac{\partial}{\partial x} a_j \right) \mathbf{e}_j = 0 \tag{4.69}$$

Now this equation can be divided into d independent transport equations for the functions a_j:

$$\frac{\partial}{\partial t}a_j - \lambda_j \frac{\partial}{\partial x}a_j = 0 \quad \forall j \in [1, \ldots, d] \tag{4.70}$$

The initial conditions for the functions a_j must be constructed such that:

$$\mathbf{u}(x, 0) = \sum_{j=1}^{d} a_j(x, 0)e_j \tag{4.71}$$

With these equations it is possible to find analytical solutions even for multidimensional hyperbolic equations. But this method is limited to cases without dispersion.

4.2 Numerical methods

Now we will consider numerical methods for hyperbolic equations. First the three simplest finite difference approximations will be shown. After that a stability analysis will show which method may be used. Then some comparisons between numerical and analytical solutions regarding the propagation of will be done. Finally the influence of the used time discretisation will be examined.

4.2.1 Finite difference approximation

The transport equation in 1D consists of a first partial derivative with respect to time and a first partial derivative with respect to x. In analogy to the discretisation of the heat equation the continuous time and space domain is divided into discrete points (compare Fig. 1.5). For the time derivative we use forward differences:

$$\frac{\partial u}{\partial t} \approx \frac{1}{\Delta t}(u_{n+1,j} - u_{n,j}) + O(\Delta t) \tag{4.72}$$

The spatial derivative should be approximated with three different schemes because the properties of these discretisations should be analysed later:

$$\frac{\partial u}{\partial x} \approx \frac{u_{n,j+\epsilon} - u_{n,j-\eta}}{(\epsilon + \eta)}h \tag{4.73}$$

Here ϵ and η are the parameters determining the type of spatial discretisation:

- $\epsilon = 1, \eta = 1$, central differences $O(h^2)$

- $\epsilon = 1, \eta = 0$, forward differences $O(h)$
- $\epsilon = 0, \eta = 1$, backward differences $O(h)$

With these approximations we get the discrete form of the transport equation:

$$\frac{1}{\Delta t}(u_{n+1,j} - u_{n,j}) + c\left(\frac{u_{n,j+\epsilon} - u_{n,j-\eta}}{(\epsilon+\eta)h}\right) = 0 \tag{4.74}$$

4.2.2 Stability analysis

For the stability analysis we use the following ansatz:

$$u_{n,j} = e^{i(kx-\omega t)} \tag{4.75}$$

Inserting the index to coordinate transformations $x = j \cdot h$ and $t = n \cdot \Delta t$ we obtain:

$$u_{n,j} = e^{-i\omega\Delta t n} e^{ikjh} = G(k)^n e^{ikjh} \tag{4.76}$$

where $G(k)$ is the gain factor which describes the amplification of a wave with wavenumber k. Using this ansatz in Eq. (4.74) gives the following expression:

$$G(k)^{n+1}e^{ikjh} - G(k)^n e^{ikjh} + \frac{r}{\epsilon+\eta}(G(k)^n e^{ik(j+\epsilon)h} - G(k)^n e^{ik(j-\eta)h}) = 0 \tag{4.77}$$

where $r = \frac{c\cdot\Delta t}{h}$ is the *Courant-number* which is an important parameter for the numerical solution of hyperbolic equations. Dividing by $G(k)^n e^(ikjh)$ and solving for $G(k)$ the result is:

$$G(k) = 1 - \frac{r}{\epsilon+\eta}(e^{ik\epsilon h} - e^{-ik\eta h}) \tag{4.78}$$

Forward differences

Inserting the parameters $\epsilon = 1, \eta = 0$ for the forward differences into Eq. (4.78) it follows that

$$G(k) = 1 - r(e^{ikh} - 1) = 1 + r(1 - e^{ikh}) = 1 + r - re^{ikh} \tag{4.79}$$

Recalling that for stability $|G(k)| \leq 1$ we can see from Fig. 4.5 that this method is unstable for all $r > 0$. So it is clear that it can not be use for hyperbolic equations.

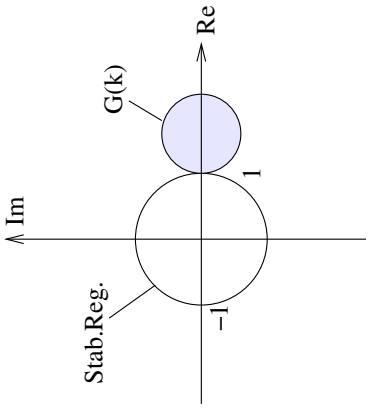


Figure 4.5: Stability region of the forward difference method

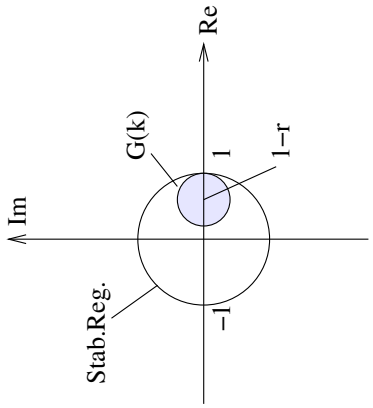


Figure 4.6: Stability region of the backward difference method

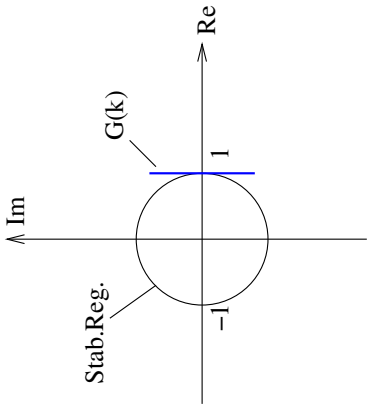


Figure 4.7: Stability region of the central difference method

Taking a look at the computational stencil reveals that the numerical method takes the value in front of the current point (let the front be defined as the direction in which the convection transports the solution). Therefore this method is often referred to as the *downwind method*.

Backward differences

For the backward differences the amplification $G(k)$ becomes:

$$G(k) = 1 - r(1 - e^{-ikh}) = 1 - r + re^{-ikh} \tag{4.80}$$

From Fig. 4.6 it can be seen that this time the amplification factor lies within the stability region. At least for

$$r = \frac{c\Delta t}{h} \leq 1 \tag{4.81}$$

This relation is called the *Courant-Friedrichs-Levy* condition. If it is satisfied, the transport equation can be solved with the backward difference method which is often called, in analogy to the forward differences, the *upwind method* because it uses the value which lies upstream.

It should be noted that similar to the heat equation some information about the relation between convective velocity, time step and spatial discretisation can be seen. If either the velocity is higher or the spatial discretisation becomes finer, the time step size must be reduced.

Central differences

Finally the gain factor for the central difference scheme is:

$$G(k) = 1 - \frac{r}{2}(e^{ikh} - e^{-ikh}) = 1 - ri(\sin(kh)) \tag{4.82}$$

Hence the absolute value will always be greater or equal to one which makes this method unstable. But the central difference scheme is the only one which achieves second accuracy. So some methods are proposed which cure the instability of the central difference scheme while being stable.

4.2.3 Friedrich's method

An intuitive explanation for the absolute instability of the central difference method is similar to the Richardson scheme for the heat equation. The points used for the space discretisation are not attached to the points which are involved in the time discretisation. A workaround which corresponds to the Du Fort Frankel scheme for the heat equation is to replace $u_{n,j}$ in the time derivative by $(1/2)(u_{n,j-1} + u_{n,j+1})$. This scheme is called the *Friedrich's* method:

$$u_{n+1,j} - \frac{1}{2}(u_{n,j-1} - u_{n,j+1}) + r(u_{n,j+1} - u_{n,j-1}) \tag{4.83}$$

Doing a von Neumann stability analysis gives the following gain factor:

$$\begin{aligned} G(k) &= \left(\frac{1}{2} + \frac{r}{2}\right) e^{-ikh} + \left(\frac{1}{2} - \frac{r}{2}\right) e^{ikh} \\ &= \cos(kh) - ir\sin(kh) \end{aligned} \tag{4.84}$$

For the absolute value of G we can get the following expression:

$$|G(k)| = \cos^2(kh) + r^2\sin^2(kh) \leq 1 \quad \text{for} \quad r \leq 1 = 0 \tag{4.85}$$

As a consequence the Friedrich's scheme for the transport equation is stable for courant numbers less than one and also second order accurate in space, which is an advantage over the upwind scheme.

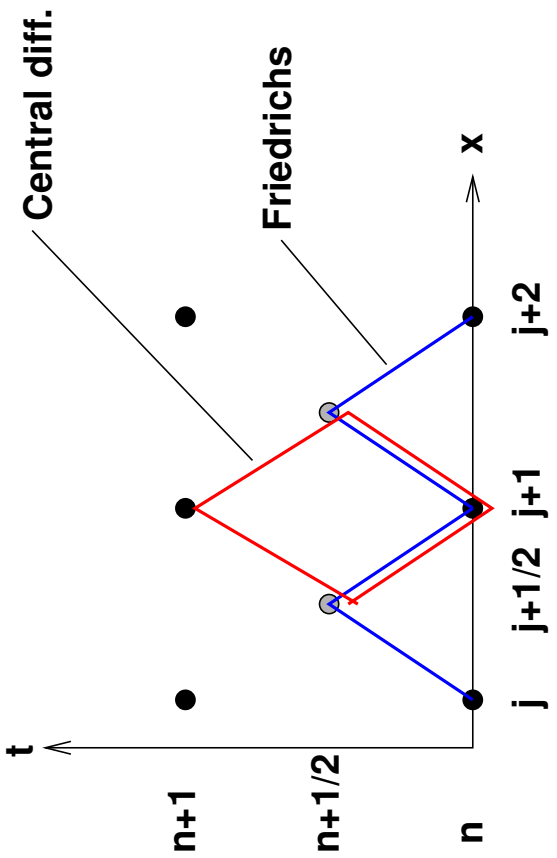


Figure 4.8: Lax-Wendroff method

4.2.4 Lax-Wendroff method

Another proposed method for the transport equation is the Lax-Wendroff method. Its goal is to achieve also second order accuracy in time. Central difference approximations have this property, but simply replacing the forward difference, which is used for the time derivative, by a central difference produces an unstable method. Instead the first point of the idea is to introduce intermediate points, which lie between the discretisation points. This is possible due to the fact that the Friedrich's scheme does not use $u_{n,j}$. So if we insert $u_{n,j}$ and $u_{n,j+1}$ for $u_{n,j-1}$ and $u_{n,j+1}$ the Friedrich's scheme will compute a point $u_{n+1/2,j+1/2}$.

After that has been done for all points, these points can be used as the basis for applying a central difference scheme (see Fig. 4.8).

Writing the three steps (two times Friedrich's method, one time the central difference scheme) we get:

$$\frac{2}{\Delta t}(u_{n+1/2,j+1/2} - \frac{1}{2}(u_{n,j} + u_{n,j+1})) + \frac{c}{h}(u_{n,j+1} - u_{n,j}) = 0 \quad \text{right blue tria} \tag{4.86}$$

$$\frac{2}{\Delta t}(u_{n+1/2,j-1/2} - \frac{1}{2}(u_{n,j} + u_{n,j-1})) + \frac{c}{h}(u_{n,j} - u_{n,j-1}) = 0 \quad \text{left blue tria.} \tag{4.87}$$

$$\frac{1}{\Delta t}(u_{n+1,j} - u_{n,j}) + \frac{c}{h}(u_{n+1/2,j+1/2} - u_{n+1/2,j-1/2}) = 0 \quad \text{cent.diff.} \tag{4.88}$$

After several steps this equations can be brought to the normal form which only includes the real discretisation points:

$$\frac{1}{\Delta t}(u_{n+1,j} - u_{n,j}) + \frac{c}{2h}(u_{n,j+1} - u_{n,j-1}) - \underbrace{\frac{c^2 \Delta t}{2h^2}(u_{n,j+1} - 2u_{n,j} + u_{n,j-1})}_{\approx -\frac{c^2}{2}\Delta t \frac{\partial^2 u}{\partial x^2}} = 0 \tag{4.89}$$

It is easy to see that the Lax-Wendroff scheme adds a term which corresponds to a diffusive part in the partial differential equation although there is no such term in the original equation. This is called *numerical diffusion*. The smoothing property of the Laplace operator stabilises the numerical scheme. Also in the Finite Element method adding a small diffusive part was one of the first methods to handle the problem occurring with hyperbolic equations.

4.2.5 Dispersion of numerical methods

During the analysis of the properties of the analytical solutions it came out that dispersion is an important aspect of hyperbolic equations. Now the reproduction of the dispersion relation by a numerical method should be examined in more detail.

One analytical solution of the transport equation was:

$$u(x,t) = \int \Phi(k) e^{i(kx - \omega(k)t)}\, dk \tag{4.90}$$

where $\Phi(k)$ depends on the initial conditions. The phase speed c_{ph} was defined as:

$$c_{ph} = \frac{\omega(k)}{k} \tag{4.91}$$

and the group speed as:

$$c_{gr} = \frac{d\omega(k)}{dk} \tag{4.92}$$

The discretisation replaced the exponential term in Eq. (4.90) by a discrete counterpart:

$$e^{i(kx-\omega(k)t)} \Rightarrow e^{i(kjh-\omega(k)\Delta t\cdot n)} = \underbrace{\left(e^{-i\omega(k)\Delta t}\right)^n}_{G(k)} e^{ikjh} \tag{4.93}$$

Introducing the numerical dispersion relation $\hat{\omega}(k)$ we get:

$$G(k) = e^{-i\hat{\omega}(k)\Delta t} \Rightarrow \ln G(k) = i\hat{\omega}(k)\Delta t \tag{4.94}$$

and hence:

$$\hat{\omega}(k) = \frac{i}{\Delta t} \ln G(k) \tag{4.95}$$

Now it is trivial to derive the numerical phase and group speed $\hat{c}_{ph}$ and $\hat{c}_{gr}$:

$$\hat{c}_{ph}(k) = \frac{\hat{\omega}(k)}{k} \tag{4.96}$$

$$\hat{c}_{gr}(k) = \frac{d\hat{\omega}(k)}{dk} \tag{4.97}$$

A further investigation of these equations will be given in the next section.

4.3 Time integration

In the previous section two methods were proposed for the discretisation of the time derivative. One was the well known Euler forward method and the other was the Central difference scheme which was only stable together with an artificial diffusion term (in the Lax-Wendroff method). Now the aspects of time integration should be analysed more detailed because they are quite important for the overall behaviour of the numerical solution.

4.3.1 General remarks

During the analysis of the beam equation which is a parabolic equation, it came out that parabolic equations allow infinite transmission speeds. So explicit time integration methods, which compute the result of the next time step only from values which lie close to the computed point, cannot reproduce this infinite transmission speed. In contrast implicit methods like the Euler backward method or the trapezoidal rule can reproduce this behaviour because all points are coupled through the system of linear equations.

Comparing the situation with hyperbolic equations where we have only a finite transmission speed, the explicit methods seem more appropriate. Especially the Upwind method reproduces the behaviour of the transport equation very intuitively. It simply takes the value of the point which lies upstream and "'transports"' its value to the next point. Explicit methods are therefore more "'natural"' for hyperbolic equations than implicit methods. Nevertheless if large timesteps should be used, implicit methods are also necessary for hyperbolic equations. This is clear, because for large timesteps the value of a point may have to be transported over a distance which is longer than the distance between two points. Only an implicit method can do this.

Summarising these results we get the following rules of thumb:

- Parabolic equations with diffusive solution: implicit time integration methods in the method of lines are "'natural"'.
- Parabolic equations with waves: explicit methods are "'natural"' but enforce severe restrictions on the timestep size Δt.
- Hyperbolic equations with wave behaviour: explicit methods are "'natural"' but the time step size must be kept small enough.
- Hyperbolic equations with large time step size: implicit method are "'natural"'

4.3.2 Analysis of the time integration

Speaking of the accuracy of a numerical method is not trivial, because especially for wave phenomena the difference between the exact solution and the numerical solution might be large although the solution is not so bad at all (see Fig. 4.9). Here the distance between the two functions is large but it is easy to see that basically the frequency is slightly different. So for the wave equation the comparison between the numerical and the analytical group or phase speed will probably bring more useful results.

Spatial discretisation

Starting again with the wave equation:

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \tag{4.98}$$

a spatial discretisation (today mostly finite elements) transforms the partial differential equation into a system of second order differential equations:

$$\mathbf{M}\ddot{\mathbf{v}} + \mathbf{K}v = \tilde{\mathbf{f}} \tag{4.99}$$

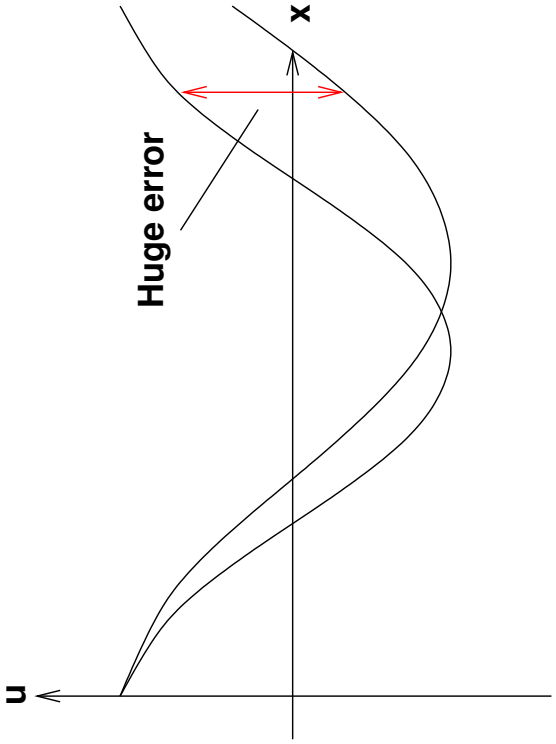


Figure 4.9: Problem of measuring the error between waves

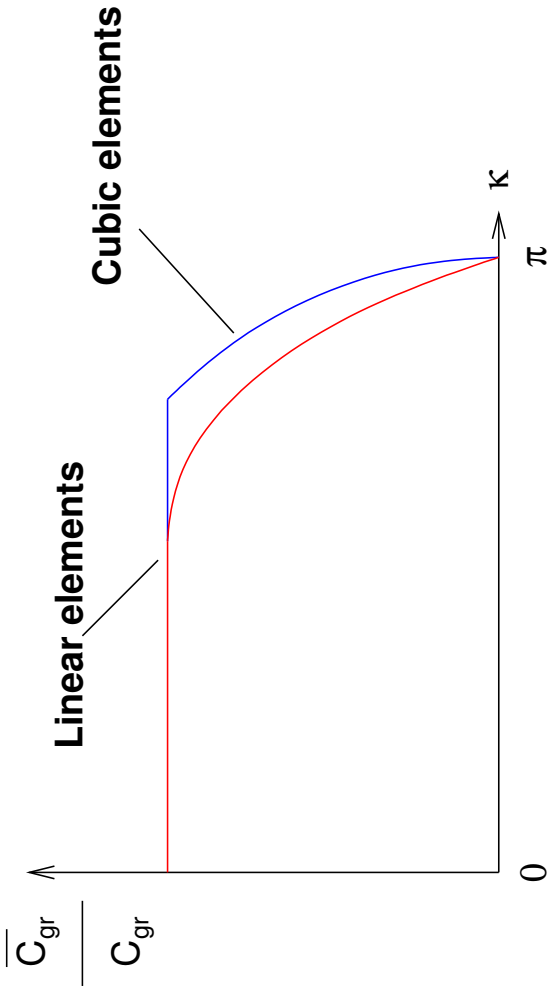


Figure 4.10: Relation between numerical group speed and analytical group speed for finite elements (schematic)

In the previous sections the group and phase speed were defined as:

$$c_{ph} = \frac{\omega}{k}, \qquad c_{gr} = \frac{d\omega}{dk} \tag{4.100}$$

Now we are looking the dispersion relation of spatially discretised wave equation. As an ansatzfunction the exponential function will be used again:

$$\begin{aligned} \mathbf{v}(t) &= \mathbf{v}_0 e^{i\omega t} && (4.101)\\ \Rightarrow \ddot{\mathbf{v}}(t) &= -\omega^2 \mathbf{v}_0 e^{i\omega t} && (4.102) \end{aligned}$$

The discrete equation becomes with this ansatz:

$$\mathbf{K}\mathbf{v}_0 = \omega^2 \mathbf{M}\mathbf{v}_0 \tag{4.103}$$

This can be seen as a generalised eigenvalue problem with $\mathbf{v}_0$ being an eigenvector and ω^2 the eigenvalue. Computing these eigenvalues the dispersion relation for the spatially discretised wave equation can be found. In Fig. 4.10 the relation between the group speed of the analytical solution and the group speed of the numerical solution is shown for different relative wavelengths (relative to the size of the finite elements). If linear elements are used, the especially short waves travel much slower. Cubic elements can improve this behaviour.

Time discretisation

Summarising the results of the previous sections, there exist three dispersion relations:

- $\omega(k)$ dispersion relation of the partial differential equation
- $\bar{\omega}(k)$ dispersion relation of the spatially discretised equation
- $\hat{\omega}(k)$ dispersion relation of the totally discrete equation

Numerical methods for first order ordinary differential equations can be analysed by using the test equation:

$$\dot{x} = \lambda x \tag{4.104}$$

with its analytical solution:

$$x(t) = x_0 e^{\lambda t} \tag{4.105}$$

This equation is not sufficient to examine numerical method for second order differential equations because these equations describe waves in time. Instead the following test equation shows to be very useful:

$$\ddot{x} = -\omega^2 x \tag{4.106}$$

It has the following exact solution:

$$x(t) = Ae^{i\omega t} + Be^{-i\omega t} \tag{4.107}$$

Here the initial conditions go into the parameters A and B. With the starting values $x_0 = x(0)$ and $v_0 = \dot{x}_0 = \dot{x}(0)$ normal numerical method for first order system can be written as:

$$\begin{pmatrix} x_n \\ v_n \end{pmatrix}^h \rightarrow \begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix}^h \tag{4.108}$$

Introducing the operator A which maps the solution at one time step into the solution at the next time step the numerical method can be written:

$$\begin{pmatrix} x_n \\ v_n \end{pmatrix}^h = A^n \begin{pmatrix} x_{n+1} \\ v_{n+1} \end{pmatrix}^h \tag{4.109}$$

Analysing the eigenvalues of the operator A will therefore give some insights about the development of the solution.

(Hier hoeren meine Aufzeichnungen auf ...)