10 Partial Differential Equations: Time-Dependent Problems

Read *sections 11.1, 11.2*
Review questions *11.1, 11.4 – 11.9, 11.10 – 11.12, 11.14 – 11.17*

10.1 Introduction

The differential equations we considered so far included only *one* independent variable, so that only derivatives with respect to this single variable were present. Such differential equations are called *ordinary* ones. The independent variable was either time (mostly in the context of initial value problems) or a one-dimensional space variable (mostly in the context of boundary value problems). In this section, we start joining the two dependencies. The unknown function (or, dependent variable) u shall depend on both time t and space x,

$$u = u(t,x).$$

The equations for determining u contain partial derivatives of u with respect to both variables. Therefore, such equations are called *partial differential equations*. Many basic laws of science are expressed in terms of partial differential equations, including

- Maxwell's equations describing electromagnetic fields,
- Navier-Stokes equations describing the flow of a fluid,
- linear elasticity equations,
- Schrödinger equations of quantum mechanics,
- Einstein's equations of general relativity.

The aim of this chapter is the introduction of the basic ideas for the numerical approximation of partial differential equations. Therefore, we will concentrate on a few simple, but typical, examples. Making things even easier, we assume that the space variable is only one-dimensional. It is common to use a short-hand notation for the derivatives of the dependent variable u. Let us define

$$u_t = \frac{\partial}{\partial t}u, \quad u_x = \frac{\partial}{\partial x}u, \quad u_{tx} = \frac{\partial^2}{\partial t \partial x}u, \quad u_{tt} = \frac{\partial^2}{\partial t^2}u, \quad u_{xx} = \frac{\partial^2}{\partial x^2}u,$$

and so on. Our model examples will be:

Heat equation This equation is given by

$$u_t = cu_{xx}, \quad c > 0.$$

The heat equation is *the* model of a *parabolic* equation.

Wave equation Here, the equation reads

$$u_{tt} = cu_{xx}, \quad c > 0.$$

The wave equation is a model of a *hyperbolic* equation.

Poisson equation Formally, we are changing the sign of c in the previous equation and set $c = -1$. Physically, t is no longer interpretable as a time variable but as a second space variable. Therefore, we will change the notation,

$$u_{xx} + u_{yy} = 0.$$

Such an equation is called *elliptic*.

There is a formal definition of parabolic, hyperbolic, and elliptic problems. However, it is of rather limited value from the application point of view. More heuristically, the following characterization may give some feeling for the physical contents of these notions:

- Hyperbolic partial differential equations describe time-dependent, conservative physical processes, such as convection, that *are not* evolving toward a steady state.
- Parabolic partial differential equations describe time-dependent, dissipative physical processes, such as diffusion, that *are* evolving toward a steady state.
- Elliptic partial differential equations describe systems that have already reached a steady state, or equilibrium, and hence are time-independent.

In the present chapter, we are concerned with parabolic and hyperbolic problems. Elliptic equations will be considered in the next chapter.

10.1.1 Boundary Values for the Heat Equation

As in the case of ordinary differential equations, a unique solvability of the partial differential equation requires additional conditions with respect to both the time variable and the space variable. Having the physical characterization of parabolic problems as evolving process in mind, we will expect that we need initial conditions with respect to time and boundary conditions with respect to space. Assuming, rather arbitrary,

$$0 \le x \le 1, \quad 0 \le t \le T,$$

the initial condition at $t = 0$ becomes

$$u(0,x) = f(x), \quad 0 \le x \le 1.$$

The function $f(x)$ describes the initial state of the system. As boundary conditions with respect to x, we choose

$$u(t,0) = \alpha, \quad u(t,1) = \beta.$$

According to our classification, these boundary conditions are both of Dirichlet type. The physical meaning is that the temperature at the boundary is fixed (by active measures like cooling or heating). It is also possible to pose Neumann boundary conditions, for example

$$u_x(t,1) = 0,$$

which means that there is no heat exchange over the boundary $t = 1$.

The coefficient c will be taken to be constant. In practice, it may vary over the domain such that $c = c(x)$. It is even not uncommon that it depends on the solution, $c = c(x,u)$, giving rise to nonlinear problems.

10.1.2 Boundary Values for the Wave Equation

The reasoning is similar to that for the heat equation. The only difference is that we need two initial conditions since the wave equation is of second order in t,

$$u(0,x) = f(x), \quad u_t(0,x) = g(x), \quad 0 \le x \le 1.$$

As in the case of the heat equation, we choose Dirichlet boundary conditions with respect to the space variable,

$$u(t,0) = \alpha, \quad u(t,1) = \beta.$$

10.2 An Analytical Solution Method: Separation of Variables

The method of separation of variables was introduced as an analytical method for the solution of partial differential equations. The basic idea consists of the *ansatz*

$$u(t,x) = v(t)w(x).$$

Let us use this decomposition in the heat equation where we assume homogeneous Dirichlet conditions $\alpha = \beta = 0$. We obtain

$$\frac{v'(t)}{v(t)} = c\frac{w''(x)}{w(x)}.$$

Since the left-hand side depends only on t while the right-hand side depends only on x, both sides must necessarily be equal to a constant, say μ,

$$v'(t) = \mu v(t), \quad cw''(x) = \mu w(x).$$

The general solution of the first equation is

$$v(t) = c_1 e^{\mu t}, \quad c_1 \text{ constant.}$$

Taking into account the boundary conditions for w, the nontrivial solutions of the second equation are given by

$$w(x) = \sin k\pi x, \quad 0 \le x \le 1, \quad k = 1,2,\ldots,$$

and the possible values for μ are given by

$$\mu = -ck^2\pi^2, \quad k = 1, 2, \ldots.$$

Hence, every function of the form

$$u_k(t,x) = e^{-ck^2\pi^2 t} \sin k\pi x, \quad k = 1, 2, \ldots$$

is a solution of the heat equation. These solutions fulfill the boundary conditions, but not necessarily the initial condition. Since every u_k is a solution of this linear differential equation, every superposition

$$u(t,x) = \sum_{k=1}^{n} u_k(t,x)$$

is a solution, too. Under some circumstances, taking the limit $n \to \infty$ is possible: If the initial condition has a convergent Fourier expansion,

$$f(x) = \sum_{k=1}^{\infty} a_k \sin k\pi x,$$

the solution of the heat equation subject to the initial condition

$$u(0,x) = f(x), \quad 0 \le x \le 1$$

has the (unique) solution

$$u(t,x) = \sum_{k=1}^{\infty} a_k e^{-ck^2\pi^2 t} \sin k\pi x.$$

In case of the wave equation, the same procedure can be repeated. Let the boundary conditions be homogeneous Dirichlet conditions again, and $g(x) \equiv 0$ such that the initial conditions read

$$u(0,x) = f(x), \quad u_t(0,x) = 0, \quad 0 \le x \le 1.$$

If

$$f(x) = \sum_{k=1}^{\infty} a_k \sin k\pi x,$$

then the solution can be represented by

$$u(t,x) = \sum_{k=1}^{\infty} a_k \cos(k\pi\sqrt{c}t) \sin(k\pi x).$$

The method of separation of variables is *not* used as a basis for numerical methods.[1] The important point is that *qualitative properties* of the equations and their solutions can be derived. From the representation of the solution of the heat equation and because of $c > 0$, we see that the solution converges toward zero for $t \to \infty$. Similarly, the solution of the wave equation indicates undamped oscillations as time evolves.

[1] In very rare cases, it can be used nevertheless.

10.3 Semi-discretization

The method of separation of variables for solving partial differential equations suggests to try something similar when discretizing such equations. Let us start from the *ansatz*

$$u(t,x) = v(t)w(x).$$

In a discretization we could try to mimic this idea:

- The differential equation for w gives rise to a boundary value problem of the type we already considered in previous paragraphs. Take, for example, a projection method. Then the space dependency is expressed by a linear combination of basis functions $\phi_1(x), \ldots, \phi_n(x)$. This gives rise to

  $$\hat{u}(t,x) = \sum_{k=1}^{n} \alpha_k(t)\phi_k(x).$$

  Equations for the time-dependent coefficients $\alpha_1, \ldots, \alpha_n$ can be derived using one of the principles introduced earlier (collocation, least-squares, Galerkin methods).

  An idea related to this approach is to split the equation with respect to time and space variables. In that case, finite difference approximations can be used with respect to the space variable while the time-dependency remains.

  As we will see later, the result of this approach is an initial value problem for ordinary differential equations. The latter can be solved by standard methods for that purpose. In this way, the huge library of sophisticated routines can be used efficiently.

  The resulting equations form semidiscretizations because one of the independent variables appears undiscretized. The resulting methods are usually called *method(s) of lines*. In a narrower sense, this notion is only used in the context of finite difference discretizations with respect to space.

- It is possible to exchange the rôle of space and time in the previous approach. More precisely, a finite difference scheme is applied with respect to time. In every time step, a discretization of a boundary value problem with respect to space is necessary. An implementation of this approach is much more involved, but efficient adaptive methods can be constructed using this idea.

- Instead of discretizing only in one direction at a time, replace all appearing partial derivatives by finite differences. After discretization, only algebraic equations remain. This approach is very often used in practice because of its inherent simplicity.

In the present section, we will have a look at the method of lines, both for finite difference and projection discretizations. The third approach will be considered in the next section.

10.3.1 Method of Lines

Consider the heat equation

$$u_t = cu_{xx}$$

subject to the boundary and initial conditions

$$\begin{aligned} u(0,x) &= f(x), \quad 0 \le x \le 1, \\ u(t,0) &= u(t,1) = 0, \quad t \ge 0. \end{aligned}$$

In order to use finite differences with respect to x, let a grid

$$0 = x_0 < x_1 < \cdots < x_n < x_{n+1} = 1$$

be given, where the step size is assumed to be constant for simplicity, $x_i - x_{i-1} = \Delta x \equiv \text{const}$. The standard approximation is

$$u(t,x_i)_{xx} \approx \frac{u(t,x_{i+1}) - 2u(t,x_i) + u(t,x_{i-1})}{(\Delta x)^2}.$$

We will use the approximations $y_i(t) \approx u(t,x_i)$ which are obtained by replacing the second derivative with respect to x by its finite difference approximation,

$$y_i'(t) = \frac{c}{(\Delta x)^2}(y_{i+1}(t) - 2y_i(t) + y_{i-1}(t)), \quad i = 1,\ldots,n, \quad t \ge 0,$$

with $y_0(t) = y_{n+1}(t) \equiv 0$ for all $t \ge 0$. The initial conditions for $\mathbf{y} = (y_1,\ldots,y_n)^T$ are given by

$$y_i(0) = f(x_i), \quad i = 1,\ldots,n.$$

The system can be written in vector notation as

$$\mathbf{y}'(t) = -\frac{c}{(\Delta x)^2}\mathbf{A}\mathbf{y}(t)$$

with the well-known matrix

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

We got an initial value problem for ordinary differential equations. In principle, we can throw it into any initial value solver which we think is appropriate. It is useful to contemplate a little bit longer about the system. The eigenvalues λ_i and eigenvectors $\mathbf{w}^{(i)}$ of the matrix $\mathbf{A}$ are easy to compute,

$$\lambda_i = 2 - 2\cos(i\pi\Delta x), \quad i = 1,\ldots,n,$$

and

$$\mathbf{w}^{(i)} = (\sin(i\pi\Delta x), \sin(2i\pi\Delta x), \ldots, \sin(ni\pi\Delta x))^T, \quad i = 1, \ldots, n.$$

Since all the eigenvalues are positive, the solution converges toward a steady state as we expected from the analytical considerations. Moreover, for $\Delta x \to 0$, $\lambda_1 = O((\Delta x)^2)$ and $\lambda_n = O(1)$ such that the system becomes very stiff for small step sizes Δx. Hence, implicit methods for stiff systems must be used.

10.3.2 Semi-discretization by Collocation

As in the case of boundary value problems for ordinary differential equations, let $\phi_1(x), \ldots, \phi_n(x)$ be a set of basis functions which fulfill the boundary conditions

$$\phi_k(0) = \phi_k(1) = 0, \quad k = 1, \ldots, n.$$

The approximate solution is searched in the form

$$\hat{u}(t,x) = \sum_{k=1}^{n} \alpha_k(t)\phi_k(x).$$

The coefficients $\alpha_k(t)$ are determined by a collocation principle. For that, let $0 \le x_1 < \cdots < x_n \le 1$ be n given collocation points. We require that $\hat{u}$ fulfills the differential equation at these points. Since

$$\hat{u}_t(t,x) = \sum_{k=1}^{n} \alpha_k'(t)\phi_k(x), \quad \hat{u}_{xx}(t,x) = \sum_{k=1}^{n} \alpha_k(t)\phi_k''(x),$$

this amounts to

$$\sum_{k=1}^{n} \alpha_k'(t)\phi_k(x_j) = c \sum_{k=1}^{n} \alpha_k(t)\phi_k''(x_j), \quad j = 1, \ldots, n.$$

Let us introduce the two $n \times n$-matrices

$$\mathbf{M} = (\phi_j(x_i))_{i,j=1,\ldots,n}, \quad \mathbf{N} = (\phi_j''(x_i))_{i,j=1,\ldots,n}.$$

Then the system can be written as

$$\mathbf{M}\mathbf{y}'(t) = c\mathbf{N}\mathbf{y}(t),$$

where $\mathbf{y} = (\alpha_1, \ldots, \alpha_n)^T$. Assuming that the matrix $\mathbf{M}$ is nonsingular, the system can be formally transformed to an explicit system,

$$\mathbf{y}'(t) = c\mathbf{M}^{-1}\mathbf{N}\mathbf{y}(t).$$

The initial conditions can be derived be requiring

$$\hat{u}(0,x_j) = \sum_{k=1}^{n} \alpha_k(0)\phi_k(x_j) = f(x_j), \quad j = 1, \ldots, n.$$

In matrix notation, this gives

$$\mathbf{M}\mathbf{y}(0) = \mathbf{f}.$$

Standard (stiff) initial value solvers can be applied now. Note that the inverse matrix $\mathbf{M}$ does not need to be computed explicitly. The initial value solvers in MATLAB allow the use of the original system $\mathbf{M}\mathbf{y}'(t) = c\mathbf{N}\mathbf{y}(t)$. If localized basis functions are used, the matrices appearing in this system are sparse such that the computational complexity is comparable to the method of lines.

10.3.3 Semi-discretization for the Wave Equation

Consider the wave equation

$$u_{tt} = cu_{xx}$$

subject to the boundary conditions

$$u(t,0) = u(t,1) = 0, \quad t \geq 0,$$

and the initial conditions

$$u(0,x) = f(x), \quad u_t(0,x) = g(x), \quad 0 \leq x \leq 1.$$

It is usual to transform this system into a system of first-order equations with respect to time. This is comparable to what we did for ordinary differential equations. Let us introduce a function $v(t,x)$ such that

$$u_t = av_x, \quad v_t = au_x,$$

with $a = \sqrt{c}$. If u is smooth enough, it holds

$$u_{tt} = \frac{\partial}{\partial t} av_x = \frac{\partial}{\partial x} av_t = \frac{\partial}{\partial x} a^2 u_x = cu_{xx}.$$

The initial conditions read

$$u(0,x) = f(x), \quad v(0,x) = \frac{1}{a}\int_0^x g(s)ds.$$

The semidiscretizations can now be derived in the same way as for the heat equation. The only difference is that, for the same accuracy, the number of unknown coefficient functions has doubled since we need to approximate u as well as v.[2]

[2]Note that there are even methods which can be directly applied to the wave equation which lead to second order initial value problems.

10.4 Fully Discrete Methods

Now we are interested in deriving numerical methods which lead directly to algebraic equations. That means that we replace *all* appearing derivatives in one step by finite differences. The discretization of the space derivative is rather straightforward. We will always use the standard symmetric second order approximation. Remembering our considerations for finite difference methods for initial value problems, the time discretization may be problematic because of possible stability problems. In principle, explicit or implicit schemes can be applied for time discretizations. Explicit schemes have the nice property that time stepping is a very cheap process while implicit schemes require the solution of a linear system of equations in every time step. Since the number of discrete variables is typically very large, explicit schemes seem to be preferable. On the other hand, in the method of lines, *stiff* initial value problems appeared. So explicit methods lead to step size restrictions. The question arises how serious this restriction is. Otherwise, we are required to use implicit methods which are appropriate for stiff problems.

10.4.1 Explicit Time Discretization

Consider the heat equation

$$u_t = cu_{xx}$$

subject to the boundary conditions

$$u(t,0) = \alpha, \quad u(t,1) = \beta, \quad t \geq 0$$

and the initial condition

$$u(0,x) = f(x), \quad 0 \leq x \leq 1.$$

The first step consists of defining a grid by a space grid size Δx and a time step Δt:

$$x_i = i\Delta x, \quad i = 0,\ldots,n+1,$$
$$t_k = k\Delta t, \quad k = 0,1,\ldots.$$

Replacing u_t with a forward Euler discretization and u_{xx} with the central difference, we obtain

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = c\frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{(\Delta x)^2},$$

where we hope for $u_i^k \approx u(t_k, x_i)$. Rearranging terms leads to

$$u_i^{k+1} = u_i^k + \mu(u_{i+1}^k - 2u_i^k + u_{i-1}^k), \quad i = 1,\ldots,n,$$

where

$$\mu = \frac{c\Delta t}{(\Delta x)^2}.$$

The boundary conditions give

$$u_0^k = \alpha, \quad u_{n+1}^k = \beta, \quad k = 0,1,\ldots,$$

while the initial condition yields

$$u_i^0 = f(x_i), \quad i = 1, \ldots, n.$$

The equations describe how the discrete solution can be computed step by step. Using the initial conditions u_i^0, the values u_i^1 for $i = 1, \ldots, n$ are directly computable. Using these values, u_i^2 can be computed and so on.

The important question is how accurate the approximate solution will be. The general answer to this question is very hard to give. Instead, we will have a look at two main ingredients. Taking into account our experiences from initial value methods, the two main properties will be *consistency* and *stability*.

Consistency Let u be the exact solution which is assumed to be sufficiently many times differentiable. If we insert this solution into the discrete formula, a certain residual will remain,

$$e_{\Delta t, \Delta x}(t,x) = \frac{u(t+\Delta t, x) - u(t,x)}{\Delta t} - c\frac{u(t, x+\Delta x) - 2u(t,x) + u(t, x-\Delta x)}{(\Delta x)^2}.$$

This residual is called the *local truncation error* (or, *local discretization error*). This definition is completely analogous to that what we did earlier. By Taylor expansion, the discretization error can be easily estimated as[3]

$$e_{\Delta t, \Delta x} = O(\Delta t) + O((\Delta x)^2).$$

This result is not surprising, because the forward Euler discretization is first order accurate while the central finite difference is second order accurate. One says that the scheme is first order accurate in time and second order accurate in space.

Stability It is tempting to conclude that the global discretization error

$$E = \max_{\substack{i=1,\ldots,n \\ k=1,\ldots,K}} |u_i^k - u(t_k, x_i)|$$

has the same order of convergence as the local error e if Δt and Δx tend toward 0. For deriving conditions for such a property to hold, we will write down the recursion in vector notation for the special case $\alpha = \beta = 0$. For every $k = 1, 2, \ldots$, define

$$\mathbf{u}^{(k)} = (u_1^k, \ldots, u_n^k)^T.$$

Then we obtain the recursion

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \mu \mathbf{A}\mathbf{u}^{(k)}$$

[3]For this estimate to hold, u must be very smooth!

where again

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

Denote the eigenvalues of $\mathbf{A}$ by λ_j and the eigenvectors by $\mathbf{w}^{(j)}$. The set of eigenvectors $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(n)}$ forms a basis of $\mathbb{R}^n$ such that there exist coefficients $a_1, \ldots, a_n$ with[4]

$$\mathbf{u}^{(0)} = \sum_{j=1}^{n} a_j \mathbf{w}^{(j)}.$$

This gives rise to

$$\begin{aligned} \mathbf{u}^{(1)} &= \mathbf{u}^{(0)} - \mu \mathbf{A}\mathbf{u}^{(0)} \\ &= \sum_{j=1}^{n} a_j \mathbf{w}^{(j)} - \mu \mathbf{A} \sum_{j=1}^{n} a_j \mathbf{w}^{(j)} \\ &= \sum_{j=1}^{n} a_j \mathbf{w}^{(j)} - \mu \sum_{j=1}^{n} a_j \mathbf{A}\mathbf{w}^{(j)} \\ &= \sum_{j=1}^{n} a_j \mathbf{w}^{(j)} - \mu \sum_{j=1}^{n} a_j \lambda_j \mathbf{w}^{(j)} \\ &= \sum_{j=1}^{n} a_j (1 - \mu\lambda_j) \mathbf{w}^{(j)}. \end{aligned}$$

Doing the same computation for $k = 1$, we obtain

$$\begin{aligned} \mathbf{u}^{(2)} &= \mathbf{u}^{(1)} - \mu \mathbf{A}\mathbf{u}^{(1)} \\ &= \sum_{j=1}^{n} a_j (1 - \mu\lambda_j)^2 \mathbf{w}^{(j)}. \end{aligned}$$

More generally, it holds

$$\mathbf{u}^{(k)} = \sum_{j=1}^{n} a_j (1 - \mu\lambda_j)^k \mathbf{w}^{(j)}.$$

From our analytical considerations we know that

$$u(t,x) \to 0 \quad \text{for} \quad t \to \infty.$$

[4]This is nothing else than a discrete version of the separation of variables!

This property must hold true if we want to expect convergence of our scheme. From the representation of $\mathbf{u}^{(k)}$ we can conclude that this holds if and only if

$$|1-\mu\lambda_j| < 1, \quad j = 1,\ldots,n.$$

Since μ as well as λ_j are positive, this holds if and only if

$$-(1-\mu\lambda_j) < 1, \quad j = 1,\ldots,n.$$

This can be rewritten as

$$\mu < \min_{j=1,\ldots,n} \frac{2}{\lambda_j}.$$

The latter term can be evaluated,

$$\min_{j=1,\ldots,n} \frac{2}{\lambda_j} = \frac{1}{1-\cos(\pi n\Delta x)} = \frac{1}{1+\cos(\pi\Delta x)} > \frac{1}{2}.$$

The last estimate is very tight. If Δx is close to 0, then $\cos(\pi\Delta x) \approx 1$. Using the definition of $\mu = c\Delta t/(\Delta x)^2$, the condition becomes

$$\Delta t < \frac{(\Delta x)^2}{2c}.$$

This condition is a restriction on the relation between Δt and Δx. If this condition is not fulfilled, the solution of the difference scheme diverges for $k \to \infty$ and hence does not provide an accurate approximation of the exact solution. Although derived with an asymptotic (that is, $t \to \infty$) behavior of the solution in mind, this condition is also necessary (and sufficient for a consistent discretization) for the global error converging to zero if both Δx and Δt converge to zero. Because of its importance, the inequality has a special name. It is called the *CFL-condition* after its inventors R. Courant, K. Friedrichs, and H. Lewy. The following table gives an impression of the maximal time step for $c = 1$ and a given space grid size.

Δx	Δt
0.1	$0.5 \cdot 10^{-2}$
0.01	$0.5 \cdot 10^{-4}$
0.001	$0.5 \cdot 10^{-6}$

Example 10.1. We demonstrate the stability issue at hand of a simple example. In the heat equation, let $\alpha = \beta = 0$. The initial condition is chosen to be $f(x) = \sin\pi x$. We use the forward Euler discretization with $n = 10$. According to the CFL-condition, the time step size must be smaller than 0.0042. With the program below, we compute the numerical approximations for the time step sizes $\Delta t = 0.004$ (which is on the stable side) and $\Delta t = 0.0044$ (which is unstable). The time interval chosen is $t \in [0,2]$. The results are plotted in the Figures 1 and 2, respectively. For those who are curious, I also included a plot of the results for $\Delta t = 0.0045$ in Figure 3. □

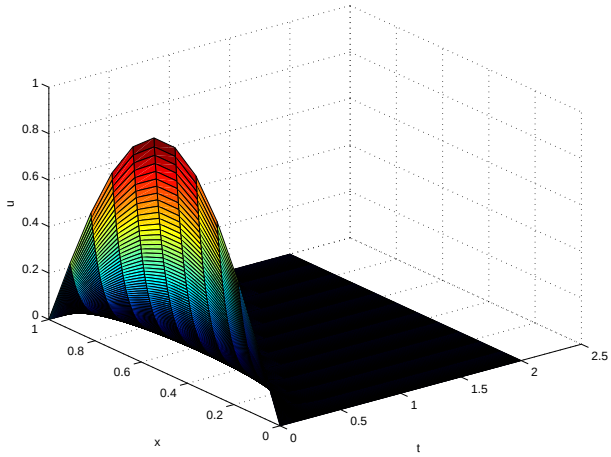

Figure 1: Solution of the example for $\Delta t = 0.004$

```
function cfl = stabtest(T,dt,n)
    dx = 1/(n+1);
    cfl = dx^2/(1+cos(pi*dx));
    mu = dt/dx^2;
    x = linspace(0,1,n+2)';
    u = sin(pi*x);
    A = spdiags([-ones(n,1),2*ones(n,1),-ones(n,1)],[-1,0,1],n,n);
    H = speye(n)-mu*A;
    t = 0;
    tout = t;
    uout = u;
    while t+dt*0.1 < T
        u(2:end-1) = H*u(2:end-1);
        uout = [uout,u];
        t = t+dt;
        tout = [tout,t];
    end
    surf(tout,x,uout)
    xlabel('t')
    ylabel('x')
    zlabel('u')
```

Explicit Time Discretization for the Wave Equation Consider now our hyperbolic model problem, the wave equation,

$$u_{tt} = cu_{xx}$$

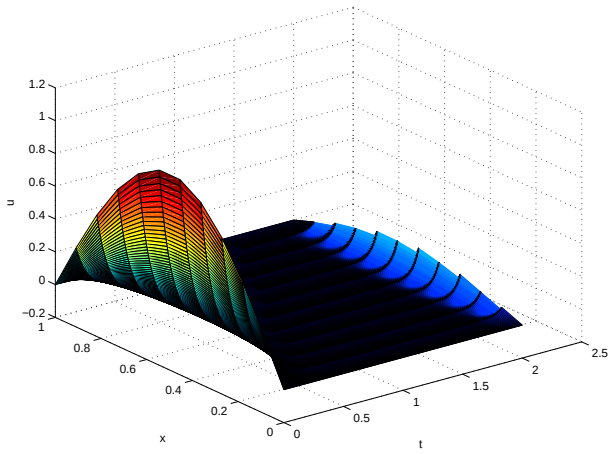

Figure 2: Solution of the example for $\Delta t = 0.0044$

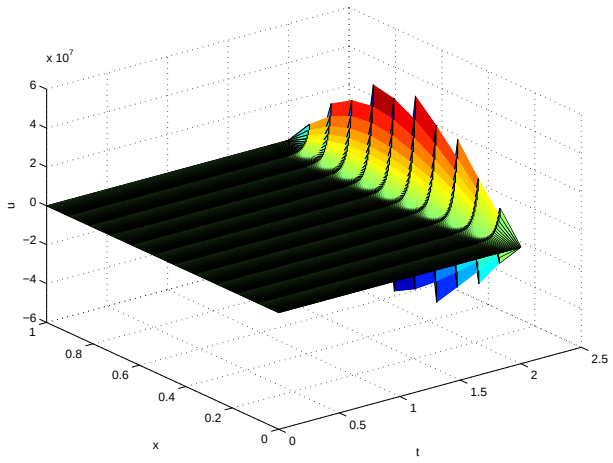

Figure 3: Solution of the example for $\Delta t = 0.0045$

subject to the boundary conditions

$$u(t,0) = \alpha, \quad u(t,1) = \beta, \quad t \geq 0,$$

and the initial conditions

$$u(0,x) = f(x), \quad u_t(0,x) = g(x), \quad 0 \leq x \leq 1.$$

The simplest finite difference method is obtained if we use the standard central differences,

$$\frac{u_i^{k+1} - 2u_i^k + u_i^{k-1}}{(\Delta t)^2} = c\frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{(\Delta x)^2}.$$

In contrast to the heat equation, there are now three time levels involved in the equation. This is similar to multistep methods for solving initial value problems. In order to start the recursion, we need beside

$$u_i^0 = f(x_i), \quad i = 1, \ldots, n,$$

also approximations

$$u_i^1 \approx u(\Delta t, x_i), \quad i = 1, \ldots, n.$$

The latter approximation can be obtained a using an explicit Euler starting step,

$$u_i^1 = f(x_i) + \Delta t g(x_i), \quad i = 1, \ldots, n.$$

The boundary conditions provide

$$u_0^k = \alpha, \quad u_{n+1}^k = \beta, \quad k = 1, 2, \ldots.$$

The values on the $k+1$-st time level are given by

$$u_i^{k+1} = 2u_i^k - u_i^{k-1} + \mu(u_{i+1}^k - 2u_i^k + u_{i-1}^k),$$

where

$$\mu = c\frac{(\Delta t)^2}{(\Delta x)^2}.$$

Similarly as before, one can easily show that for the local discretization error, it holds

$$e_{\Delta t, \Delta x} = O((\Delta t)^2) + O((\Delta x)^2).$$

Stability can now be investigated as before by using the discrete method of separation of the variables. The stability condition (or, CFL-condition) is in the present case

$$\Delta t \leq \frac{\Delta x}{\sqrt{c}}.$$

This restriction is much weaker than the CFL-condition for the heat equation. It says that the time step is only required to be proportional to the space step.

10.4.2 Implicit Time Discretizations for the Heat Equation

Qualitatively, we expected severe restrictions on the time step size for the heat equation, because we found in the semi-discrete case that the stiffness of the system increased with a smaller and smaller space step. A natural idea is to apply implicit methods that are known to work well for stiff systems. The simplest example of such a method is the implicit Euler discretization in time. Its application is, for the heat equation

$$u_t = cu_{xx},$$

given by

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = c\frac{u_{i+1}^{k+1} - 2u_i^{k+1} + u_{i-1}^{k+1}}{(\Delta x)^2}, \quad i = 1, \ldots, n.$$

Although the system looks rather similar to the explicit Euler discretization, the important difference is the appearance of the unknowns on time level $k+1$ on the right-hand side. Let again

$$\mu = \frac{c\Delta t}{(\Delta x)^2}.$$

With the well-known matrix $\mathbf{A}$, the system can be equivalently written as

$$(\mathbf{I} + \mu\mathbf{A})\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{b}, \quad k = 1, 2, \ldots.$$

The vector $\mathbf{b}$ takes care of the Dirichlet boundary conditions: If we require

$$u(t, 0) = \alpha, \quad u(t, 1) = \beta, \quad t \geq 0,$$

we use the discrete boundary conditions

$$u_0^k = \alpha, \quad u_{n+1}^k = \beta, \quad k = 1, 2, \ldots.$$

Then the vector $\mathbf{b}$ contains only zeros with the exception of the first and the last components which are equal to $\mu\alpha$ and $\mu\beta$, respectively. Moreover, the initial condition

$$u(0, x) = f(x), \quad 0 \leq x \leq 1,$$

yields

$$u_i^0 = f(x_i), \quad i = 1, \ldots, n.$$

In every iteration step, we must solve a linear system of equations with a *tridiagonal* matrix $\mathbf{I} + \mu\mathbf{A}$. Even if this can be done in $O(n)$ operations, the amount of work is much larger than for the explicit methods.[5] Is this additional effort worth spending? The answer is yes since we will obtain excellent stability properties.

[5] As we will see in the next chapter, the discrepancy in the amount of work needed for the explicit and implicit methods, respectively, increases considerably when the dimension of the space domain increases to 2 and 3.

Let us do the stability analysis in the same way as before for the explicit method. Set, for simplicity, $\alpha = \beta = 0$. Assume that

$$\mathbf{u}^{(0)} = \sum_{j=1}^{n} a_j \mathbf{w}^{(j)}$$

and, more general,

$$\mathbf{u}^{(k)} = \sum_{j=1}^{n} a_j^{(k)} \mathbf{w}^{(j)}.$$

Introducing these expressions into the recursion, we obtain

$$\begin{aligned}
(\mathbf{I} + \mu\mathbf{A})\mathbf{u}^{(k+1)} &= \mathbf{u}^{(k)} \\
(\mathbf{I} + \mu\mathbf{A}) \sum_{j=1}^{n} a_j^{(k+1)} \mathbf{w}^{(j)} &= \sum_{j=1}^{n} a_j^{(k)} \mathbf{w}^{(j)} \\
\sum_{j=1}^{n} (1 + \mu\lambda_j) a_j^{(k+1)} \mathbf{w}^{(j)} &= \sum_{j=1}^{n} a_j^{(k)} \mathbf{w}^{(j)}.
\end{aligned}$$

This yields

$$(1 + \mu\lambda_j) a_j^{(k+1)} = a_j^{(k)}, \quad j = 1, \ldots, n,$$

such that we finally obtain

$$a_j^{(k+1)} = \frac{1}{(1 + \mu\lambda_j)} a_j^{(k)} =: \gamma_j a_j^{(k)}, \quad j = 1, \ldots, n.$$

Resolving this recursion for all steps, we see that

$$\mathbf{u}^{(k)} = \sum_{j=1}^{n} a_j \gamma_j^k \mathbf{w}^{(j)}.$$

Stability is obtained if, for all $j = 1, \ldots, n$, it holds $|\gamma_j| < 1$. Because of $\lambda_j > 0$ and $\mu > 0$, this is always the case! Hence, the method is stable independently of the relation between the time and space step sizes. A method with this property is called *unconditionally stable*.

The local discretization error can be estimated as before. It is no surprise that

$$e_{\Delta t, \Delta x} = O(\Delta t) + O((\Delta x)^2).$$

For a reasonable approximation, we obtain a restriction on the step size because of accuracy considerations. Assume that

$$e_{\Delta t, \Delta x} \approx c_1 \Delta t + c_2 (\Delta x)^2.$$

Both terms on the right-hand side should have the same order of magnitude. This is the case if

$$\Delta t \approx c_3 (\Delta x)^2.$$

This condition resembles the stability condition for the explicit method. But the interpretation of the condition is different: It became an accuracy condition.

10.4.3 The Crank-Nicolson Method

We conclude the chapter with one of the most often used early methods because this method fits easily in the present context. We want to emphasize at this point that more advanced methods are in common use today which are much more efficient than the Crank-Nicolson method.

The idea is simply to replace the implicit Euler method by the trapezoidal rule. The discretization is given by

$$\frac{u_i^{k+1}-u_i^k}{\Delta t}=\frac{c}{2}\Big(\frac{u_{i+1}^{k+1}-2u_i^{k+1}+u_{i-1}^{k+1}}{(\Delta x)^2}+\frac{u_{i+1}^{k}-2u_i^{k}+u_{i-1}^{k}}{(\Delta x)^2}\Big),\quad i=1,\ldots,n.$$

In matrix notation,

$$(\mathbf{I}+\frac{\mu}{2}\mathbf{A})\mathbf{u}^{(k+1)}=(\mathbf{I}-\frac{\mu}{2}\mathbf{A})\mathbf{u}^{(k)}+\mathbf{b},\quad k=1,2,\ldots.$$

By using the methods above, one can show that this method is unconditionally stable and of second order in both time and space.