Chapter 1

Parabolic partial differential equations

Parabolic PDE (partial differential equations) belong to problems often encountered in chemical engineering. Non-stationary heat conduction or mass transport by diffusion lead to parabolic equations. Many problems are described by linear parabolic equations (simple problems in diffusion and heat conduction) which can be solved by classical analysis. The solution is in the form of an infinite series of special functions (e.g. Bessel and Hankel functions) and these functions must be evaluated which may be expensive. Thus even linear equations are often solved numerically. Many problems involve nonlinear parabolic equations (heat and mass exchange with exothermic reaction, adsorption, non-stationary heat exchange with radiation etc.). Nonlinear parabolic equations must always be solved numerically. The aim of this chapter is to give introduction to numerical analysis used in parabolic equations - the implicit and explicit difference schemes.

1.1 Canonical form of second order equations with two independent variables

Consider a quasilinear equation of the second order with two independent variables x and y in a given domain $D \subset R^2$:

$$A\frac{\partial^2 u}{\partial x^2} + 2B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + F\Big(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\Big) = 0, \tag{1.1}$$

where the coefficients A, B and C are functions of x and y and have continuous derivatives up to order at least 2. Suppose that at least one of them is always nonzero. Corresponding to equation (1.1) we can write the quadratic form

$$At_1^2 + 2Bt_1t_2 + Ct_2^2\,. \tag{1.2}$$

Depending on the values of A, B and C we distinguish three types of equation (1.1), see Tab.1.1.

Table 1.1: Types of equations

Type	Condition
hyperbolic	$B^2 - AC > 0$
parabolic	$B^2 - AC = 0$
elliptic	$B^2 - AC < 0$

We can introduce two new independent variables (X, Y) instead of (x, y) by the functions

$$X = X(x, y)\,, \qquad Y = Y(x, y)\,, \tag{1.3}$$

which are assumed to be twice continuously differentiable and to have nonzero Jacobian

$$\frac{D(X,Y)}{D(x,y)} = \begin{vmatrix} \dfrac{\partial X}{\partial x} & \dfrac{\partial X}{\partial y} \\ \dfrac{\partial Y}{\partial x} & \dfrac{\partial Y}{\partial y} \end{vmatrix} \neq 0 \tag{1.4}$$

in the domain D considered.

Putting (1.3) into (1.1), equation (1.1) changes to

$$\bar{A}\frac{\partial^2 u}{\partial X^2} + 2\bar{B}\frac{\partial^2 u}{\partial X \partial Y} + \bar{C}\frac{\partial^2 u}{\partial Y^2} + \bar{F}(X, Y, u, \frac{\partial u}{\partial X}, \frac{\partial u}{\partial Y}) = 0\,, \tag{1.5}$$

where

$$\begin{aligned} \bar{A}(X,Y) &= A\left(\frac{\partial X}{\partial x}\right)^2 + 2B\frac{\partial X}{\partial x}\frac{\partial X}{\partial y} + C\left(\frac{\partial X}{\partial y}\right)^2, \\ \bar{C}(X,Y) &= A\left(\frac{\partial Y}{\partial x}\right)^2 + 2B\frac{\partial Y}{\partial x}\frac{\partial Y}{\partial y} + C\left(\frac{\partial Y}{\partial y}\right)^2, \\ \bar{B}(X,Y) &= A\frac{\partial X}{\partial x}\frac{\partial Y}{\partial x} + B\left(\frac{\partial X}{\partial x}\frac{\partial Y}{\partial y} + \frac{\partial X}{\partial y}\frac{\partial Y}{\partial x}\right) + C\frac{\partial X}{\partial y}\frac{\partial Y}{\partial y}\,. \end{aligned} \tag{1.6}$$

It is easy to show that

$$\bar{B}^2 - \bar{A}\bar{C} = (B^2 - AC)\left(\frac{\partial X}{\partial x}\frac{\partial Y}{\partial y} - \frac{\partial X}{\partial y}\frac{\partial Y}{\partial x}\right)^2 \tag{1.7}$$

thus transformation (1.3) does not change the type of equation (1.1). Transformation (1.3) can be chosen so that exactly one of the following three conditions holds

$$\bar{A} = 0 \quad \wedge \quad \bar{C} = 0\,, \tag{1.8a}$$

$$\bar{A} = 0 \quad \wedge \quad \bar{B} = 0 \ \text{ or } \ \bar{B} = 0 \quad \wedge \quad \bar{C} = 0\,, \tag{1.8b}$$

$$\bar{A} = \bar{C} \quad \wedge \quad \bar{B} = 0\,. \tag{1.8c}$$

In each of these three cases (which differ in the sign of the expression $(B^2 - AC)$) equation (1.5) can be written in simple (canonical) form:

1. $(B^2 - AC) > 0$ *hyperbolic equation*
   The canonical form is

   $$\frac{\partial^2 u}{\partial X \partial Y} = F_1\left(X, Y, u, \frac{\partial u}{\partial X}, \frac{\partial u}{\partial Y}\right). \tag{1.9}$$

   Often another form is used as the canonical one, namely

   $$\frac{\partial^2 u}{\partial \xi^2} - \frac{\partial^2 u}{\partial \eta^2} = F_2\left(\xi, \eta, u, \frac{\partial u}{\partial \xi}, \frac{\partial u}{\partial \eta}\right); \tag{1.10}$$

   this equation can be derived from (1.9) by the transformation

   $$X = \xi + \eta\,, \qquad Y = \xi - \eta\,.$$

   These types of equations appear seldom in chemical engineering so we will not consider them in this text.

2. $(B^2 - AC) = 0$ *parabolic equation*
   The canonical form is

   $$\frac{\partial^2 u}{\partial Y^2} = F_3\left(X, Y, u, \frac{\partial u}{\partial X}, \frac{\partial u}{\partial Y}\right). \tag{1.11}$$

3. $(B^2 - AC) < 0$ *elliptic equation*
   The canonical form is

$$\frac{\partial^2 u}{\partial X^2} + \frac{\partial^2 u}{\partial Y^2} = F_4\left(X, Y, u, \frac{\partial u}{\partial X}, \frac{\partial u}{\partial Y}\right). \tag{1.12}$$

Numerical solution of elliptic equations are discussed in chapter **??**, hyperbolic equations of order one are discussed in chapter **??**. The current chapter deals with numerical solution of parabolic equations.

1.2 Numerical solution of parabolic equations with two independent variables

Numerical solution of parabolic equations in two dimensions (or in one spatial coordinate x and one time coordinate t) is thoroughly treated in literature (as opposed to higher dimensional cases). As chemical engineering problems often lead to equations in time and one spatial coordinate, one section is devoted to this problem. Let us start with the linear equation. Later we will see that almost all the conclusion for the linear equation can be used for the nonlinear one as well.

1.2.1 Grid methods for linear problems

Let us start with the linear parabolic equation with constant coefficients

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}. \tag{1.13}$$

A more general equation (describing heat conduction or mass diffusion)

$$\frac{\partial u}{\partial \tau} = \sigma \frac{\partial^2 u}{\partial x^2} \tag{1.14}$$

can be converted to (1.13) by the substitution $t = \sigma\tau$.

The solution of equation (1.13) is often searched for on a rectangle $D = [0,1] \times [0,T]$ shown in Fig. 1.1.
The solution $u(x,t)$ must satisfy the initial condition (the function $\varphi(x)$ is given)

$$u(x,0) = \varphi(x), \qquad 0 < x < 1, \tag{1.15}$$

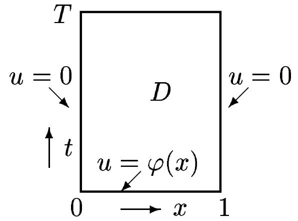

Figure 1.1: The rectangle D where the solution of the parabolic equation (1.13) is defined.

and a boundary condition, e.g.

$$u(0,t) = u(1,t) = 0\,. \tag{1.16}$$

Other problems may contain other boundary conditions, e.g.

$$x = 0: \quad \frac{\partial u(0,t)}{\partial x} = 0\,, \tag{1.17}$$

$$x = 1: \quad u(1,t) = 1 \tag{1.18}$$

or other.

1.2.1.1 Simple explicit formula

The most common approach to equation (1.13) is the difference method also called the grid method. There is a wide range of difference methods, let us start with the simplest one. Let us divide the interval $[0,1]$ in x into n subintervals by equidistant grid points

$$x_0 = 0,\ x_1 = h,\ x_2 = 2h,\ \dots,\ x_{n-1} = 1-h,\ x_n = 1\,,$$

where $h = 1/n$ and $x_i = ih$, $i = 0, 1, \dots, n$. Similarly the interval $[0,T]$ in t is divided into r equal parts by the grid points

$$t_0 = 0,\ t_1 = k,\ \dots,\ t_r = T,$$

where the time step is $k = T/r$ and $t_j = jk$, $j = 0, 1, \dots, r$. The set of nodes - the intersections of the lines $x = ih$, $i = 0, 1, \dots, n$, and the lines $t = jk$, $j = 0, 1, \dots, r$, forms a rectangular grid denoted by $D^{(h)}$ (see Fig.1.2). On this grid we can approximate the derivatives of the function u by the difference formulas (see chapter **??**) for $i = 1, \dots, n-1$, $j = 0, \dots, r-1$:

$$\left.\frac{\partial u}{\partial t}\right|_{(x_i,t_j)} = \frac{u_i^{j+1} - u_i^j}{k} + \mathcal{O}(k)\ , \tag{1.19}$$

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x_i,t_j)} = \frac{u_{i-1}^j - 2u_i^j + u_{i+1}^j}{h^2} + \mathcal{O}(h^2)\,, \tag{1.20}$$

where we denote $u(ih, jk) = u(x_i, t_j) \doteq u_i^j$.

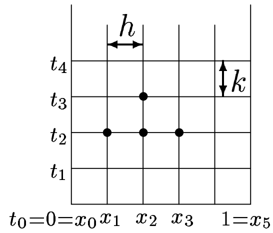

Figure 1.2: The grid $D^{(h)}$, n = 5 and the approximation (1.21) for $i = 2, j = 2$

Consider the equation (1.13) in one node $(x_i, t_j) \in D^{(h)}$ and the approximation using (1.19) and (1.20):

$$\frac{u_i^{j+1} - u_i^j}{k} = \frac{u_{i-1}^j - 2u_i^j + u_{i+1}^j}{h^2} + \mathcal{O}(k + h^2)\,. \tag{1.21}$$

This is illustrated in Fig.1.2. Neglecting $\mathcal{O}(k + h^2) = \mathcal{O}(k) + \mathcal{O}(h^2)$, which is called the approximation error and using the initial condition (1.15) and the boundary conditions (1.16) we get the following difference problem:

$$u_i^{j+1} = \frac{k}{h^2}\left(u_{i-1}^j + u_{i+1}^j\right) + \left(1 - \frac{2k}{h^2}\right)u_i^j\,, \qquad \begin{matrix} i = 1, 2, \ldots, n-1 \\ j = 0, 1, \ldots, r-1\,, \end{matrix} \tag{1.22}$$

$$u_i^0 = \varphi(ih)\,, \qquad i = 1, 2, \ldots, n-1\,, \tag{1.23}$$

$$u_0^j = 0\,, \qquad u_n^j = 0\,, \qquad j = 0, 1, \ldots, r\,. \tag{1.24}$$

If $u(x_i, t_j)$ is the solution of (1.13) with the initial condition (1.15) and the boundary condition (1.16), then the error of the solution computed by (1.22), (1.23) and (1.24) is

$$\varepsilon_i^j = u(x_i, t_j) - u_i^j\,. \tag{1.25}$$

Similarly as for ordinary differential equations (ODE) we require that making the grid finer, i.e. $h \to 0$, $k \to 0$, results in $\varepsilon_i^j \to 0$ in $D^{(h)}$. If this is the case we say that the solution of (1.22), (1.23) and (1.24) converges to the exact solution of (1.13), (1.15) and (1.16). It is obvious that if the numerical solution does not converge to the exact solution then the difference method is useless. The difference approximation (1.22) is called the explicit three point difference scheme. This name tells that the value u_i^{j+1} is computed explicitly from the values $u_{i-1}^j, u_i^j, u_{i+1}^j$. The relations (1.22), (1.23) and (1.24) are iterated. The vector $\boldsymbol{u}^j = (u_0^j, u_1^j, \ldots, u_n^j)$ is called the j-th profile. In (1.22) the j-th profile is called the old (the known) profile, and the $j+1$-st profile is called the new profile. To sum up, the new profile is computed point-wise from the old profile.

1.2.1.2 Stability of the difference scheme

We denote by u_i^j the exact solution of the difference problem (1.22), (1.23) and (1.24) and we denote by $\widetilde{u}_i^j$ the numerically computed solution. These differ due to round-off errors introduced in each arithmetical operation done on a digital computer. We want this round-off error not to grow too much in the course of computation. We want the errors

$$\varrho_i^j = u_i^j - \widetilde{u}_i^j \tag{1.26}$$

to go to zero or at least to stay bounded for increasing j. This requirement presents the stability condition of the difference scheme. The total error of the numerical solution can be estimated by

$$|\varepsilon_i^j| + |\varrho_i^j|\,, \tag{1.27}$$

where $|\varrho_i^j|$ is small and negligible compared to the error of the method $|\varepsilon_i^j|$ for stable schemes. Unstable schemes are useless for practical computation because we can never compute with infinite number of decimal digits.

Let us explain the problem of stability for the scheme (1.22) in more detail. It is easy to rewrite (1.22), (1.23) and (1.24) using profiles as

$$\begin{aligned} \boldsymbol{u}^{j+1} &= \mathbf{A}_1 \boldsymbol{u}^j\,, \\ \boldsymbol{u}^0 &= \Big[0, \varphi(h), \varphi(2h), \ldots, \varphi((n-1)h), 0\Big]^T\,, \end{aligned} \tag{1.28}$$

where the matrix $\mathbf{A}_1$ is three-diagonal

$$\mathbf{A}_1 = \begin{pmatrix} 0 & 0 & & & & & \\ \alpha & (1-2\alpha) & \alpha & & & & \\ & \alpha & (1-2\alpha) & \alpha & & 0 & \\ & & \ddots & \ddots & \ddots & & \\ & 0 & & \alpha & (1-2\alpha) & \alpha & \\ & & & & \alpha & (1-2\alpha) & \alpha \\ & & & & & 0 & 0 \end{pmatrix}, \tag{1.29}$$

where

$$\alpha = \frac{k}{h^2}\,. \tag{1.30}$$

After denoting $\bar{\boldsymbol{u}}^j = (u_1^j, u_2^j, \ldots, u_{n-1}^j)$, and using $u_0^j = u_n^j = 0$, we can rewrite (1.28) as

$$\bar{\boldsymbol{u}}^{j+1} = \mathbf{A}\bar{\boldsymbol{u}}^j\,, \qquad \bar{\boldsymbol{u}}^0 = \Big[\varphi(h), \varphi(2h), \ldots, \varphi((n-1)h)\Big]^T\,, \tag{1.31}$$

where the matrix $\mathbf{A}$ is of type $(n-1)\times(n-1)$:

$$\mathbf{A}=\begin{pmatrix} (1-2\alpha) & \alpha & & & \\ \alpha & (1-2\alpha) & \alpha & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & \alpha & (1-2\alpha) & \alpha \\ & & & \alpha & (1-2\alpha) \end{pmatrix}. \tag{1.32}$$

Consider now a small deviation of the initial condition (introduced by the round-off error) $\bar{\boldsymbol{\varrho}}^0$:

$$\bar{\boldsymbol{\varrho}}^0=\bar{\boldsymbol{u}}^0-\bar{\boldsymbol{u}}'^0\,. \tag{1.33}$$

Here the prime does not mean derivative, it just denotes another profile. Equation (1.31) with the initial condition $\bar{\boldsymbol{u}}'^0$ becomes

$$\bar{\boldsymbol{u}}'^{j+1}=\mathbf{A}\bar{\boldsymbol{u}}'^j\,,\quad \bar{\boldsymbol{u}}'^0=\bar{\boldsymbol{u}}^0-\bar{\boldsymbol{\varrho}}^0\,. \tag{1.34}$$

The error $\bar{\boldsymbol{\varrho}}^j=\bar{\boldsymbol{u}}^j-\bar{\boldsymbol{u}}'^j$ evolves as

$$\bar{\boldsymbol{\varrho}}^{j+1}=\mathbf{A}\bar{\boldsymbol{\varrho}}^j\,, \tag{1.35}$$

giving

$$\bar{\boldsymbol{\varrho}}^j=\mathbf{A}^j\bar{\boldsymbol{\varrho}}^0\,. \tag{1.36}$$

The norm of the effect of the initial deviation $\bar{\boldsymbol{\varrho}}^0$ can be estimated by

$$\|\bar{\boldsymbol{\varrho}}^j\|\le\|\mathbf{A}\|^j\|\bar{\boldsymbol{\varrho}}^0\|\,, \tag{1.37}$$

where the norms can be defined (see **??**)

$$\|\boldsymbol{v}\|=\max_i|v_i|\,, \tag{1.38}$$

$$\|\mathbf{A}\|=\max_i\sum_{s=1}^{n-1}|a_{is}|\,. \tag{1.39}$$

The estimate (1.37) gives:
If

$$\|\mathbf{A}\|\le 1\,, \tag{1.40}$$

then the deviation $\bar{\boldsymbol{\varrho}}^0$ of initial condition does not grow in the course of computation.

Similarly, a deviation $\bar{\boldsymbol{\varrho}}$ in the j-th profile (instead of in the first one) can be treated by considering this j-th profile as the initial condition and the conclusions are the same. In a real computation round-off errors appear in each profile. Thanks to the linearity of (1.31) the total error stays bounded if (1.40) remains valid.

It is easy to see that if the elements in the main diagonal of the matrix $\mathbf{A}$ are non-negative i.e. if

$$\alpha \le \frac{1}{2}\,, \tag{1.41}$$

then due to (1.39) we have $||\mathbf{A}|| = 1$. Thus (1.41) is a sufficient condition for the stability of method (1.22).

Let us see whether this condition is also necessary. The necessary condition requires that for the least norm of the matrix $\mathbf{A}$ the non-equality (1.40) holds. As for any matrix norm it holds $\varrho(\mathbf{A}) = \max|\lambda_i| \le \|\mathbf{A}\|$, where λ_i are the eigenvalues of the matrix $\mathbf{A}$, the necessary and sufficient condition is

$$|\lambda_i| \le 1 \quad i = 1, 2, \dots, n-1\,. \tag{1.42}$$

The matrix (1.32) has eigenvalues

$$\lambda_i = 1 - 4\alpha \sin^2 \frac{i\pi}{2n}\,, \quad i = 1, \dots, n-1\,,$$

then the condition (1.42) is equivalent to the condition (1.41).

If the original equation (1.13) has non-constant coefficients (as functions of x) then the rows of the matrix $\mathbf{A}$ differ. Then the eigenvalues cannot be expressed analytically, they must be found numerically (see chapter **??**), which is expensive for large n. Then it is better to use the sufficient stability condition (1.40) where $\|\mathbf{A}\|$ is defined according to (1.39).

Sometimes the stability is estimated by the Fourier (von Neumann) method. We describe this method briefly for the explicit differential scheme (1.22). This method ignores boundary conditions which is no problem in our case, since the boundary conditions specify zero values of u at the boundaries. In cases where the boundary conditions influence the stability this method has a limited validity. On the other hand the method using the spectral radius of the matrix is still valid, although sometimes difficult to apply.

Assume that the solution of the differential equations can be written as $u_i^j = z_j y_i$, and let us choose one harmonics $e^{\mathrm{i}\beta x}$ from its Fourier representation. Here i is the imaginary unit; to avoid confusion we denote the imaginary unit by i written in ordinary font while we denote the index by i written in mathematical font. The solution of the difference equation is assumed in the form $e^{\omega t}e^{\mathrm{i}\beta x}$ and we want to find conditions for the expression in t not to grow (ω may be complex).

We put

$$u_i^j = e^{\omega jk}e^{\mathrm{i}\beta ih} \tag{1.43}$$

into (1.22), $\alpha = k/h^2$:

$$e^{\omega(j+1)k}e^{\mathrm{i}\beta ih} = (1-2\alpha)e^{\omega jk}e^{\mathrm{i}\beta ih} + \alpha(e^{\omega jk}e^{\mathrm{i}\beta(i-1)h} + e^{\omega jk}e^{\mathrm{i}\beta(i+1)h})\,.$$

After simplification we get

$$e^{\omega k} = 1 - 4\alpha\sin^2(\frac{1}{2}\beta h)\,,$$

and the condition $|e^{\omega k}| \le 1$ gives $\alpha \le \frac{1}{2}$.

Table 1.2: Difference scheme (1.22), error propagation for $\alpha = \frac{1}{2}$

$u_i^{j+1} = \frac{1}{2}(u_{i-1}^j + u_{i+1}^j)$									
$j=4$	$\varepsilon/16$	0	$\varepsilon/4$	0	$3\varepsilon/8$	0	$\varepsilon/4$	0	$\varepsilon/16$
$j=3$	0	$\varepsilon/8$	0	$3\varepsilon/8$	0	$3\varepsilon/8$	0	$\varepsilon/8$	0
$j=2$	0	0	$\varepsilon/4$	0	$\varepsilon/2$	0	$\varepsilon/4$	0	0
$j=1$	0	0	0	$\varepsilon/2$	0	$\varepsilon/2$	0	0	0
$j=0$	0	0	0	0	ε	0	0	0	0

The error propagation is illustrated in Tables 1.2 and 1.3. The initial error in a single node is denoted by ε. The first case is for $\alpha = \frac{1}{2}$ and the deviation is damped. In the other case $\alpha = 10$ and the error grows quickly.

Note that for the stability of the difference scheme it is necessary that the original differential equations are stable in a certain sense, i.e. a small change in the initial condition results in a small deviation in the exact solution. To show an example where this is not the case, consider the diffusion equation in backward time

$$\frac{\partial u}{\partial t} = -\frac{\partial^2 u}{\partial x^2}$$

which we get from (1.13) by changing t to $-t$. Now the method (1.22) is unstable for any $\alpha > 0$ and a similar result holds for further methods.

Table 1.3: Difference scheme (1.22), error propagation for $\alpha = 10$,

$u_i^{j+1} = -19u_i^j + 10(u_{i-1}^j + u_{i+1}^j)$							
$j=3$	1000ε	-5700ε	13830ε	-18259ε	13830ε	-5700ε	1000ε
$j=2$	0	100ε	-380ε	561ε	-380ε	100ε	0
$j=1$	0	0	10ε	-19ε	10ε	0	0
$j=0$	0	0	0	ε	0	0	0

Table 1.4: Exact solution of (1.13),(1.15),(1.16) and (1.44)

t	x=0.3	x = 0.5	x=0.7
0.005	0.5966	0.8404	0.5966
0.01	0.5799	0.7743	0.5799
0.02	0.5334	0.6809	0.5334
0.10	0.2444	0.3021	0.2444

As an illustration we give an example of a stable and of an unstable scheme for equation (1.13) with boundary conditions (1.16) and with the initial condition (1.15) where

$$\varphi(x) = \begin{cases} 2x & \text{for } 0 \le x \le \frac{1}{2} \\ 2(1-x) & \text{for } \frac{1}{2} \le x \le 1. \end{cases} \tag{1.44}$$

Analytic solution can be found in the form

$$u = \frac{8}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} \left(\sin \frac{n\pi}{2} \right) \left(\sin n\pi x \right) e^{(-n^2 \pi^2 t)} \tag{1.45}$$

and the values of this solution are given in Table 1.4. We use the difference scheme (1.22) for $h = 0.1$ and α equal to 0.1 and 0.5. The results are summarized in Table 1.5. Compare the achieved accuracy. Note that for $x = 0.5$ the agreement is worse because the initial condition (1.44) has at this point non-continuous derivative. The solution is symmetric in x around $x = 0.5$.

Figs. 1.3 and 1.4 show the agreement of numerical ($h = 0.1$) and of the analytic solution for $\alpha < 0.5$ and for $\alpha > 0.5$, i.e. for stable and for unstable scheme.

Table 1.5: Solution of (1.13),(1.15), (1.16) and (1.44) by explicit method for $h = 0.1$

			$x = 0.3$	$x = 0.5$	$x = 0.7$
$\alpha = 0.1$	$t = 0.01$	$(j = 10)$	0.5822	0.7867	0.5822
$k = 0.001$	$t = 0.02$	$(j = 20)$	0.5373	0.6891	0.5373
$\alpha = 0.5$	$t = 0.01$	$(j = 2)$	0.6000	0.8000	0.6000
$k = 0.005$	$t = 0.02$	$(j = 4)$	0.5500	0.7000	0.5500
	$t = 0.1$	$(j = 20)$	0.2484	0.3071	0.2484

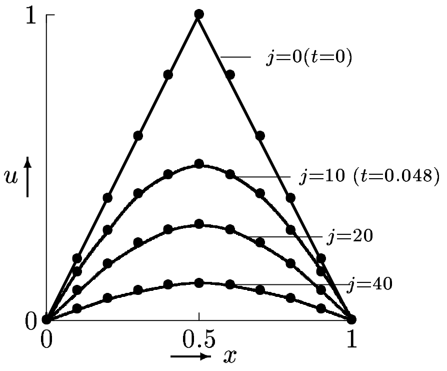

Figure 1.3: Numerical (●) and exact (—–) solution for $\alpha = 0.48$, $h = 0.1$

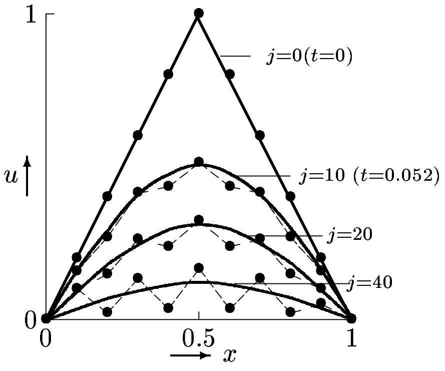

Figure 1.4: Numerical (– ● –) and exact (—–) solution for $\alpha = 0.52$, $h = 0.1$

1.2.1.3 Simple implicit formula

Let us discuss a more complicated form of the difference formula with a parameter w

$$\frac{u_i^{j+1}-u_i^j}{k} = w\frac{u_{i-1}^{j+1}-2u_i^{j+1}+u_{i+1}^{j+1}}{h^2} + (1-w)\frac{u_{i-1}^j-2u_i^j+u_{i+1}^j}{h^2}. \tag{1.46}$$

It is easy to see that for $w=0$ this equation simplifies to (1.21) and this special case represents the above discussed simple explicit scheme. For $w=1$ we have the opposite case - a simple implicit difference scheme

$$-\alpha u_{i-1}^{j+1} + (1+2\alpha)u_i^{j+1} - \alpha u_{i+1}^{j+1} = u_i^j\,. \tag{1.47}$$

For $w=\frac{1}{2}$ we get an "averaged" scheme called Crank-Nicolson:

$$-\frac{\alpha}{2}u_{i-1}^{j+1} + (1+\alpha)u_i^{j+1} - \frac{\alpha}{2}u_{i+1}^{j+1} = \frac{\alpha}{2}u_{i-1}^j + (1-\alpha)u_i^j + \frac{\alpha}{2}u_{i+1}^j. \tag{1.48}$$

Biographical note: Phyllis Nicolson (21 September 1917 - 6 October 1968) was a British mathematician most known for her work on the Crank-Nicolson scheme together with John Crank. John Crank (6 February 1916 - 3 October 2006) was a British mathematical physicist, best known for his work on the numerical solution of partial differential equations.

Similarly as for the explicit scheme it can be shown that approximation error is $\mathcal{O}(k+h^2)$ for (1.47) thus method (1.47) is of similar accuracy as the explicit formula. For the Crank-Nicolson scheme (1.48) it can be shown that the error is $\mathcal{O}(k^2+h^2)$, this method is more precise than the methods (1.22) and (1.47). This can be explained by the fact that the time derivative is approximated in the point $(t_{j+1}+t_j)/2$ corresponding to a central three-point difference formula (see Tab. **??**) with the error $\mathcal{O}((\frac{k}{2})^2)$. The Crank-Nicolson scheme (1.48) is stable for any $\alpha = k/h^2$. The formula (1.46) is stable for any α if $w \in [\frac{1}{2}, 1]$. If $w < \frac{1}{2}$ then this method is stable if

$$\alpha \le \frac{1}{2(1-2w)}\,. \tag{1.49}$$

For $w=0$ this gives again condition (1.41).

Unlike the explicit method, where each value of the new profile $\boldsymbol{u}^{j+1}$ was computed explicitly one after another from the old profile $\boldsymbol{u}^j$, the equation (1.46) for $w \neq 0$ represents a system of linear equations for unknowns $u_i^{j+1}, i = 1, 2, \ldots, n-1$, which must be solved simultaneously.

The matrix of this system is three-diagonal for a general w (for $w = 0$ the matrix is diagonal and the method is explicit). The system may be solved by factorization (see chapter **??**) when computing a new profile from the old one. Let us compare the Crank-Nicolson method with the explicit method using the equations (1.13), (1.15), (1.16) and (1.44) for $h = 0.1$. The results are given in Table 1.6. It is easy to see that the error of the results of the explicit method for $\alpha = 0.1$ are similar to those of the Crank-Nicolson method with the step $k = 0.01$ (where $\alpha = 1$). The explicit method requires to compute ten times more profiles, although the computation was easier because it was not necessary to solve a system of linear equations with a three-diagonal matrix. When we compare the number of arithmetic operations then the Crank-Nicolson method is more efficient.

	Explicit method (1.22)		Crank-Nicolson method	Analytic solution (1.45)
	$\alpha = \frac{1}{10}$ $k = 0.001$	$\alpha = \frac{1}{2}$ $k = 0.005$	$k = 0.01$	
$t = 0.01$	0.7867	0.8000	0.7691	0.7743
$t = 0.02$	0.6891	0.7000	0.6921	0.6809
$t = 0.10$	0.3056	0.3071	0.3069	0.3021

Table 1.6: Comparison of the explicit and the Crank-Nicolson methods. Values in the point $x = 0.5$ are shown $(h = 0.1)$

1.2.1.4 Multi-step methods

So far, we have considered two-profile methods that contain $\boldsymbol{u}^j$ and $\boldsymbol{u}^{j+1}$ only. We have noted that the discretization in t has the greatest contribution to the error, namely $\mathcal{O}(k)$, or $\mathcal{O}(k^2)$ in special methods. This means we must use a small time step k and this requires a long computation time. Another possibility is (similarly to Adams formulas, see chapter **??**), to approximate the derivative $\frac{\partial u}{\partial t}$ using more than two points. To start such a computation we must know more than just one profile (given by the initial condition). To prepare these profiles another method must be used. One disadvantage of multi-step methods is that it is not easy to adapt the step size k according to how complicated the solution is. Another disadvantage, namely the need of more computer memory to store extra profiles becomes less important with modern hardware. One important advantage of multi-step methods is that we can use a greater step size k because the approximation of $\frac{\partial u}{\partial t}$ is more precise. We show a few multi-step methods for the equation (1.13), using the approximation from table **??** and **??**.

A non-central approximation of $\frac{\partial u}{\partial t}$ gives a three-profile implicit formula

$$\frac{3u_i^{j+1} - 4u_i^j + u_i^{j-1}}{2k} = \frac{u_{i-1}^{j+1} - 2u_i^{j+1} + u_{i+1}^{j+1}}{h^2} . \tag{1.50}$$

This can be rewritten to

$$-2\alpha u_{i-1}^{j+1} + (3 + 4\alpha)u_i^{j+1} - 2\alpha u_{i+1}^{j+1} = 4u_i^j - u_i^{j-1} . \tag{1.51}$$

Similarly a four-profile implicit formula is

$$-6\alpha u_{i-1}^{j+1} + (11 + 12\alpha)u_i^{j+1} - 6\alpha u_{i+1}^{j+1} = 18u_i^j - 9u_i^{j-1} + 2u_i^{j-2} \tag{1.52}$$

and finally a five-profile implicit formula is

$$-12\alpha u_{i-1}^{j+1} + (25+24\alpha)u_i^{j+1} - 12\alpha u_{i+1}^{j+1} = 48u_i^j - 36u_i^{j-1} + 16u_i^{j-2} - 3u_i^{j-3} . \tag{1.53}$$

Formulas (1.51), (1.52) and (1.53) have the error $\mathcal{O}(k^2 + h^2)$, $\mathcal{O}(k^3 + h^2)$ and $O(k^4 + h^2)$ resp. From the computational point of view these formulas are not much more difficult than a simple implicit formula (1.47); the right-hand-side of the system of linear algebraic equations with a three-diagonal matrix contain a few more terms. To start we must prepare three initial profiles (besides the initial condition) using another method with a sufficiently small error.

There exist another multi-step formulas where the approximation of $\frac{\partial^2 u}{\partial x^2}$ is computed from more profiles with appropriate weights with total sum being one. On the other hand, explicit multi-step methods are seldom used, because the stability condition requires a small step size in t, so that the high accuracy of the approximation in t cannot be used (by taking a large step size).

1.2.1.5 Boundary conditions

We have considered boundary conditions of the first kind, i.e. boundary conditions specifying the value of the solution, e.g. for equation (1.13) the boundary condition was (1.16). Often the boundary conditions specify the derivative of the unknown function (for example the boundary between a heat conducting medium and an insulator is described by $\frac{\partial u}{\partial n} = 0$ where n means the normal i.e. perpendicular direction). This type of boundary condition is called the boundary condition of the second kind. The most often case, however, is a linear combination of the function value and its derivative at the boundary. i.e. $C_1 u + C_2 \frac{\partial u}{\partial n} = C_3$. This type of boundary condition is called the boundary condition of the third kind. Nonlinear boundary condition are discussed below.

Consider a general linear boundary condition

$$C_1 u + C_2 \frac{\partial u}{\partial x} = C_3 \tag{1.54}$$

for the equation (1.13) in $x = 0$. Assume $C_2 \neq 0$, i.e. (1.54) is not a condition of the first kind. The simplest approximation of (1.54) is to replace the derivative $\frac{\partial u}{\partial x}$ by a suitable difference formula (see chapter **??**, boundary value problem for ordinary differential equation). Replacing

$$\left.\frac{\partial u}{\partial x}\right|_{\substack{x=0\\ t=(j+1)k}} = \frac{u_1^{j+1} - u_0^{j+1}}{h} + \mathcal{O}(h)\,, \tag{1.55}$$

and putting into (1.54) we get a linear equation for u_0^{j+1} and u_1^{j+1} (upper indexes can be chosen arbitrarily because (1.54) holds for all t)

$$\left(C_1 - \frac{C_2}{h}\right) u_0^{j+1} + \frac{C_2}{h} u_1^{j+1} = C_3\,. \tag{1.56}$$

Using (1.56) for the explicit formula (1.22) is simple: u_0^{j+1} is evaluated by (1.56) based on u_1^{j+1} (computed from u_0^j, u_1^j, u_2^j). Put together we get

$$u_0^{j+1} = \frac{C_3 h}{C_1 h - C_2} - \frac{C_2}{C_1 h - C_2} u_1^{j+1} = \delta + \gamma_0 u_0^j + \gamma_1 u_1^j + \gamma_2 u_2^j\,, \tag{1.57}$$

where

$$\delta = \frac{C_3 h}{C_1 h - C_2}\,, \quad \gamma_0 = \gamma_2 = -\frac{\alpha C_2}{C_1 h - C_2}\,, \quad \gamma_1 = -(1 - 2\alpha)\frac{C_2}{C_1 h - C_2}\,.$$

The first row of the "transformation" matrix $\mathbf{A}_1$ (see (1.29)) changes to

$$(\gamma_0, \gamma_1, \gamma_2, 0, \ldots).$$

It is easy to see that

$$|\gamma_0| + |\gamma_1| + |\gamma_2| = \left|\frac{C_2}{C_1 h - C_2}\right|$$

for $\alpha = k/h^2 \le \frac{1}{2}$ (which must be satisfied for stability reasons). From $h = \sqrt{k/\alpha}$ it follows that for constant α we have $|\gamma_0| + |\gamma_1| + |\gamma_2| = 1 + \mathcal{O}(\sqrt{k})$, which is a sufficient stability condition. Thus the method (1.22) with the boundary condition (1.56) is stable for $\alpha \le \frac{1}{2}$. This is a non-trivial result. Replacement of boundary condition can change the stability. When investigating stability it is always necessary to consider the replacement of boundary conditions as well.

The replacement (1.55) has one big disadvantage both for explicit and for implicit scheme. The error is by one order worse than the error of the equation, thus it is better to use a more precise replacement for $\frac{\partial u}{\partial x}$. There are two possibilities:

1. To use a non-central three-point difference

$$\left.\frac{\partial u}{\partial x}\right|_{\substack{x=0\\ t=(j+1)k}} = \frac{-3u_0^{j+1} + 4u_1^{j+1} - u_2^{j+1}}{2h} + \mathcal{O}(h^2)\,, \tag{1.58}$$

This is no complication for explicit formula. For the implicit formula the resulting system must be converted to a three-diagonal one.

2. To use a central three-point difference

$$\left.\frac{\partial u}{\partial x}\right|_{\substack{x=0\\ t=(j+1)k}} = \frac{u_1^{j+1} - u_{-1}^{j+1}}{2h} + \mathcal{O}(h^2) \tag{1.59}$$

by introducing a fictitious node with index -1. This increases the number of unknowns and we must find one equation for this new unknown. This can be done by approximating equation (1.13) by the implicit formula (1.47) for $i = 0$. The unknown u_{-1}^{j+1} can be expressed from this equation as a function of u_0^j, u_0^{j+1} and u_1^{j+1} and we put the result into the approximation (1.59). For the implicit method (1.47) we get again a system of linear equations with a three-diagonal matrix. This second approach is better because the replacement (1.59) has a smaller error than the replacement (1.58), although they are of the same order (see chapter **??**).

For the implicit or the explicit method the replacement of the boundary condition is easy. For more complex methods it is usually not obvious how to approximate the boundary condition to get the highest accuracy of the resulting replacement. The implicit replacement of the boundary condition usually gives good results.

In some problems the boundary conditions depend on time, e.g.

$$u(0,t) = \sin \omega t$$

is periodic in time t. This type of boundary conditions presents no big complication. We can use the same methods as for time independent boundary conditions. The resulting formula contains time dependent term.

Sometimes we have a linear parabolic equation with a nonlinear boundary condition, e.g. equation (1.13) with boundary conditions

$$\psi_0 \left(u(0,t), \frac{\partial u(0,t)}{\partial x}, t \right) = 0 \ , \qquad \psi_1 \left(u(1,t), \frac{\partial u(1,t)}{\partial x}, t \right) = 0 \qquad (1.60)$$

instead of (1.16).

This is the case of heat conduction with radiation, or diffusion with surface chemical reaction etc. Let us illustrate this by an example. Consider heat conduction in an insulated bar described by equation (1.13). One end of the bar is kept at a constant temperature and the other end of the bar receives heat by radiation from a source of constant temperature and looses heat by its own radiation. The boundary conditions are

$$x = 0 : \quad u = U_0 \ , \qquad x = 1 : \quad s(1 - u^4) - \frac{\partial u}{\partial x} = 0 \, , \qquad (1.61)$$

and the initial condition is: for $t = 0$ and $x \in [0,1]$ $u = U_0$. Here the temperature is related to the thermodynamic temperature of the radiation source. The dimensionless parameter s contains the fourth power of the source temperature, the Stephan-Boltzmann constant, heat conductivity, the length of the bar and the configuration factor. The partial differential equation can be discretized by the Crank-Nicolson method and the boundary condition (1.61) can be replaced by the implicit method by introducing a fictitious profile $n + 1$:

$$s\Big(1 - (u_n^{j+1})^4\Big) - \frac{u_{n+1}^{j+1} - u_{n-1}^{j+1}}{2h} = 0 \, . \qquad (1.62)$$

We have again a system of n equations for n unknowns $u_1^{j+1}, \ldots, u_n^{j+1}$ with a three-diagonal appearance. The first $n-1$ equations are linear and the last equation is nonlinear in the form

$$au_{n-1}^{j+1} + bu_n^{j+1} = c - d(u_n^{j+1})^4 \,. \tag{1.63}$$

The last equation comes from putting (1.62) into the Crank-Nicolson replacement for $i = n$, the constant c contains u_{n-1}^j, u_n^j. The right-hand-side of the last "linear" equation of the system with a three-diagonal matrix depends on the "parameter" u_n^{j+1}.

The first phase of the factorization and vanishing the bottom diagonal gives the last equation in the form

$$b' u_n^{j+1} = c' - d'(u_n^{j+1})^4 \,. \tag{1.64}$$

This is an algebraic equation for one unknown u_n^{j+1}. This equation can be solved by some method in chapter **??** (we have a good initial approximation u_n^j). Only after solving the equation (1.64) the second phase of the factorization is done.

Exercise: How can we solve the same PDE with the non-linear boundary condition (1.61) on both ends of the bar?

1.2.1.6 Methods with higher accuracy

This section is devoted to algorithms that increase the order of the difference approximation and that allow higher step sizes h and k for the same accuracy. This can be achieved by two ways. The first way is to tune certain parameters in the difference formula so that the order is higher. This way has a big disadvantage that the difference formula is prepared to fit the given PDE and cannot be used for other equations. We do not discuss this type of methods here. The other way uses more nodes for the approximations of derivatives.

Exercise: Find the minimal number of nodes to approximate
$\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial t}$, $\frac{\partial^2 u}{\partial x \partial t}$, $(\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2})$, etc.

To avoid problems with having more unknowns than equations we use non-symmetric difference formulas near boundaries. This is illustrated in Fig. 1.5 where the second derivative in the nodes 2, 3, 4 is approximated by a symmetric formula with 5 nodes and in the nodes 1, 5 by a non-symmetric formula again with 5 nodes. We consider a difference approximation of equation (1.13) where the derivative $\frac{\partial^2 u}{\partial x^2}$ is approximated using 5 points. The case with more nodes is similar. The explicit approximation can be

$$\frac{u_i^{j+1} - u_i^j}{k} = \frac{-u_{i-2}^j + 16u_{i-1}^j - 30u_i^j + 16u_{i+1}^j - u_{i+2}^j}{12h^2} + \mathcal{O}(k + h^4)\,. \quad (1.65)$$

A necessary and sufficient stability condition is now more restrictive in the time step k, namely $\alpha \leq \frac{3}{8}$. On the other hand the spatial step size h can be larger so the restriction in k is not necessarily worse than in the classical explicit method (1.22).

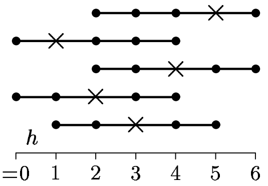

Figure 1.5: Non-symmetric approximations

The reader is invited to write the implicit formula of type (1.65), similarly as the non-symmetric approximation for one node near the boundary (use chapter **??**).

Formula (1.65) and similar ones have one disadvantage - the approximation in the t direction is much worse than in the x direction. One way to remove this disadvantage is to use the Crank-Nicolson approximation, namely

$$\begin{aligned} \frac{u_i^{j+1} - u_i^j}{k} &= \frac{1}{2}\left(\frac{-u_{i-2}^{j+1} + 16u_{i-1}^{j+1} - 30u_i^{j+1} + 16u_{i+1}^{j+1} - u_{i+2}^{j+1}}{12h^2} + \right. \quad (1.66) \\ &\left. + \frac{-u_{i-2}^j + 16u_{i-1}^j - 30u_i^j + 16u_{i+1}^j - u_{i+2}^j}{12h^2} \right) + \mathcal{O}(k^2 + h^4)\,. \end{aligned}$$

The implicit approximation means that we must solve a system of linear equations with a five-diagonal matrix, this can be solved by an algorithm similar to factorization of a three-diagonal matrix.

The other way how to increase the accuracy in the t direction is to use more than two profiles, i.e. to use a multi-step method, see chapter 1.2.1.4.

1.2.2 Grid methods for nonlinear problems

A nonlinear problem can be formulated in general as

$$F\left(t, x, u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial u}{\partial t}\right) = 0\,. \tag{1.67}$$

In chemical engineering we usually solve problems linear both in $\frac{\partial u}{\partial t}$ and in $\frac{\partial^2 u}{\partial x^2}$. These problems are called quasi-linear, e.g.

$$\frac{\partial u}{\partial t} = a\big(t, x, u, \frac{\partial u}{\partial x}\big)\frac{\partial^2 u}{\partial x^2} + b\big(t, x, u, \frac{\partial u}{\partial x}\big)\frac{\partial u}{\partial x} + c\big(t, x, u, \frac{\partial u}{\partial x}\big) \tag{1.68}$$

(the last two terms could be written as a single term, but b and c are often independent of $\frac{\partial u}{\partial x}$, so this form is more convenient).

Some authors use the term quasi-linear for systems with coefficients that do not depend on first derivatives; the terminology is not uniform. It is appropriate to say that unlike linear equations, there is no general approach to nonlinear parabolic equations. Each nonlinear equation (or a system of them) is usually a unique problem for numerical solution. Thus we discuss algorithms that often work in engineering applications, they are not however reliable recipes for all problems.

1.2.2.1 Simple explicit method

If we replace all spatial derivatives and nonlinear coefficients in the old profile in equation (1.68) we get the approximation

$$\begin{aligned}\frac{u_i^{j+1} - u_i^j}{k} &= a\left(t_j, x_i, u_i^j, \frac{u_{i+1}^j - u_{i-1}^j}{2h}\right)\frac{u_{i-1}^j - 2u_i^j + u_{i+1}^j}{h^2} + \\ &\quad + b\left(t_j, x_i, u_i^j, \frac{u_{i+1}^j - u_{i-1}^j}{2h}\right)\frac{u_{i+1}^j - u_{i-1}^j}{2h} + \\ &\quad + c\left(t_j, x_i, u_i^j, \frac{u_{i+1}^j - u_{i-1}^j}{2h}\right), \quad i = 1, 2, ..., n-1\,,\end{aligned} \tag{1.69}$$

which is from the computational point of view similar to the explicit method (1.22). From the known values of $u_0^j, u_1^j, \ldots, u_n^j$ it is possible to compute the right hand side of the approximation (1.69) and then we can get easily u_i^{j+1} for $i = 1, 2, \ldots, n-1$. The problem of approximation of the boundary condition is equivalent to that for linear equation.

Similarly as in the linear case, the steps h and k in the approximation (1.69) cannot be chosen arbitrarily because for some combinations of h and k the replacement (1.69) is unstable. Unlike the linear case it is not possible to get simple analytic condition of stability. The stability of nonlinear problems must be tested experimentally. This is done by computing a few steps for various values of the step k, the instability can be seen clearly. Also, the condition of stability may vary with time t. For equation (1.69) the necessary condition of stability (as the lower order terms have no significant influence on stability) is

$$\frac{k\, a\Big(t_j, x_i, u_i^j, \frac{u_{i+1}^j - u_{i-1}^j}{2h}\Big)}{h^2} < \frac{1}{2}\,. \tag{1.70}$$

In (1.70) the boundary conditions of the first kind are considered; the boundary conditions with derivatives may change the condition substantially. The estimate (1.70) shows that the acceptable step size k may indeed vary with time t and this must be taken into account.

Next, we use the explicit method (1.69) for a problem with a known analytic solution. Consider the partial differential equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{u}{2}\frac{\partial u}{\partial x} + c\,u^2 - e^{-2\pi^2 t}\Big(c \sin^2 \pi x + \frac{\pi}{4}\sin 2\pi x\Big) \tag{1.71}$$

with the boundary condition

$$u(0,t) = u(1,t) = 0 \tag{1.72}$$

and the initial condition

$$u(x,0) = \sin \pi x\,. \tag{1.73}$$

It is easy to check that the analytic solution (for any c) is

$$u(x,t) = e^{-\pi^2 t} \sin \pi x\,. \tag{1.74}$$

Table 1.7 shows results computed by the explicit method (for $c = 1$).

1.2.2.2 Method of simple linearization

The explicit method is easy to use, but it has a strong stability restriction which is here a greater disadvantage than for linear equations, because the evaluation of nonlinear functions is usually expensive. We often split nonlinear terms into two parts: a linear part, considered on the new profile and a

Table 1.7: Results for explicit method (1.69) and equation (1.71), values $u(0.5;t)$
for various values of h and k

t	$h = 0.1$			$h = 0.05$		Exact solution
	$k = 0.005$	$k = 0.002$	$k = 0.001$	$k = 0.001$	$k = 0.0005$	(equation (1.74))
0.01	0.9045	0.9059	0.9063	0.9058	0.9060	0.9060
0.05	0.6053	0.6100	0.6115	0.6096	0.6104	0.6105
0.2	0.1341	0.1384	0.1399	0.1381	0.1388	0.1389
0.4	0.0180	0.0192	0.0196	0.0191	0.0193	0.0193

nonlinear part (or a remaining part), considered on the old profile. E.g. u^2 can be split into $u^{j+1}u^j$, similarly u^3 can be split into $u^{j+1}(u^j)^2$, or $(\frac{\partial u}{\partial x})^2$ can be split into $(\frac{\partial u}{\partial x})^{j+1}(\frac{\partial u}{\partial x})^j$ etc. Here superscript 2 or 3 means power, while superscript j or $j+1$ denotes discretized time. This trick is called linearization. Thus equation (1.68) can be approximated by

$$\begin{aligned}\frac{u_i^{j+1}-u_i^j}{k} &= a\left(t_j, x_i, u_i^j, \frac{u_{i+1}^j - u_{i-1}^j}{2h}\right)\frac{u_{i-1}^{j+1} - 2u_i^{j+1} + u_{i+1}^{j+1}}{h^2} + \\ &+b\left(t_j, x_j, u_i^j, \frac{u_{i+1}^j - u_{i-1}^j}{2h}\right)\frac{u_{i+1}^{j+1} - u_{i-1}^{j+1}}{2h} + \qquad (1.75)\\ &+c\left(t_j, x_i, u_i^j, \frac{u_{i+1}^j - u_{i-1}^j}{2h}\right).\end{aligned}$$

The coefficients a, b, c are evaluated in the old profile j and the derivatives $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial u}{\partial x}$ are approximated in the new profile $j+1$. The difference scheme (1.75) is actually an implicit scheme and it gives a system of linear equations for unknowns $u_0^{j+1}, u_1^{j+1}, \ldots, u_n^{j+1}$ (including boundary condition replacement). This is a three-diagonal system and it can be solved by factorization. Approximation (1.75) is implicit for spatial derivatives. Alternatively $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial u}{\partial x}$ could be approximated by the average of the values in the old and in the new profile similarly to the Crank-Nicolson method. Each equation can usually be linearized by various ways, the experience and intuition is important.

1.2.2.3 Extrapolation techniques

Let us try to replace equation (1.68) in pure implicit way, i.e.

$$\begin{aligned}\frac{u_i^{j+1}-u_i^j}{k} &= a_i^{j+1}\frac{u_{i-1}^{j+1}-2u_i^{j+1}+u_{i+1}^{j+1}}{h^2}+b_i^{j+1}\frac{u_{i+1}^{j+1}-u_{i-1}^{j+1}}{2h}+c_i^{j+1}\,,\\ & \quad i=1,2,...,n-1\,. \end{aligned} \tag{1.76}$$

The coefficients a, b, c are functions of the unknowns $\boldsymbol{u}^{j+1}$, e.g.

$$a_i^{j+1}=a\left(t_{j+1},x_i,u_i^{j+1},\frac{u_{i+1}^{j+1}-u_{i-1}^{j+1}}{2h}\right). \tag{1.77}$$

System (1.76) can be solved as a set of nonlinear equations, which will be discussed later. Here we try to predict the values of $a_i^{j+1}, b_i^{j+1}, c_i^{j+1}$ based on the knowledge of a few last profiles. Assuming $u(x,t), a, b, c$ are sufficiently smooth functions we can extrapolate the values of $a_i^{j+1}, b_i^{j+1}, c_i^{j+1}$ linearly for small time step k from the known profiles j and $(j-1)$ according to

$$a_i^{j+1}\approx 2a_i^j-a_i^{j-1} \tag{1.78}$$

(and similarly for b and c). We can extrapolate from more than just two profiles, e.g. quadratic extrapolation gives

$$a_i^{j+1}=a_i^{j-2}-3a_i^{j-1}+3a_i^j\,. \tag{1.79}$$

Approximation (1.76) is implicit, thus the stability restriction is not so severe (if any) as for explicit one. The error introduced by extrapolation is much smaller than the error of linearization as discussed in the previous section. So what is the disadvantage of this approach? It is a multi-step method, meaning the first one or two steps must be computed by another method, e.g. by actual solving the nonlinear equations (1.76).

1.2.2.4 Predictor - corrector technique

In the last section we discussed the prediction of the coefficients a, b, c in the profile $(j+1)$. There is another way: to predict the values of $\bar{\boldsymbol{u}}^{j+1}$ using the explicit method (1.69), where $u_i^{j+1}=\bar{u}_i^{j+1}$, $i=1,2,\ldots,n-1$. This predicted $\bar{\boldsymbol{u}}_i^{j+1}$ can be substituted into the coefficients a, b, c in equation (1.76), e.g.

$$\bar{a}_i^{j+1}=a\left(t_{j+1},x_i,\bar{u}_i^{j+1},\frac{\bar{u}_{i+1}^{j+1}-\bar{u}_{i-1}^{j+1}}{2h}\right). \tag{1.80}$$

Then (1.76) becomes

$$\begin{aligned}\frac{u_i^{j+1}-u_i^j}{k} &= \bar{a}_i^{j+1}\frac{u_{i-1}^{j+1}-2u_i^{j+1}+u_{i+1}^{j+1}}{h^2}+\bar{b}_i^{j+1}\frac{u_{i+1}^{j+1}-u_{i-1}^{j+1}}{2h}+\bar{c}_i^{j+1}\,,\\ & i=1,2,...,n-1\,, \qquad (1.81)\end{aligned}$$

which is a system in linear equations (including boundary conditions) with a three diagonal matrix; the solution being similar as in the linear case.

What advantages and disadvantages has this method as compared to extrapolation methods (which can be regarded as a special case of predictor - corrector methods)? It is not necessary to start with a different method i.e. the computation can start with the knowledge of the initial condition alone. Sometimes the memory requirements are weaker. As opposed to the linear extrapolation this prediction is usually better (even though they both are of order $\mathcal{O}(k)$). On the other hand the computation time can grow. Using a large step size k (from the point of view of stability of the explicit method) is no problem because the implicit method (1.81) eliminates this influence.

It is clear that when using the Crank-Nicolson method instead of (1.81) we must evaluate $\bar{a}_i^{j+1/2}, \bar{b}_i^{j+1/2}, \bar{c}_i^{j+1/2}$, which can be done using an explicit method with the step size $k' = k/2$. When using this predictor - corrector method we can compare $\bar{u}_i^{j+1}$ and u_i^{j+1} (predicted and computed values) in each profile. We want these values to be close. If they differ much we can substitute u_i^{j+1} for $\bar{u}_i^{j+1}$ and repeat the computation according to (1.81). This means we repeat the corrector step, similarly as for ordinary differential equations (see **??**). It would be too difficult to prove the convergence of this method for general a, b, c and arbitrary boundary conditions. The experience tells us that this approach usually converges for sufficiently small k.

1.2.2.5 Newton's method

Consider the system (1.76) including the boundary value replacement as a system of nonlinear equations

$$a_i^{j+1}\frac{u_{i-1}^{j+1}-2u_i^{j+1}+u_{i+1}^{j+1}}{h^2}+b_i^{j+1}\frac{u_{i+1}^{j+1}-u_{i-1}^{j+1}}{2h}+c_i^{j+1}-\frac{u_i^{j+1}}{k}+\frac{u_i^j}{k}=0\,, \quad (1.82)$$

thus

$$f_i(u_{i-1}^{j+1}, u_i^{j+1}, u_{i+1}^{j+1}) = 0\,, \qquad i=1,2,\dots,n-1\,, \qquad (1.83)$$

and possible boundary conditions

$$u_0^{j+1} = u_n^{j+1} = 0\,, \tag{1.84}$$

that allow to eliminate u_0^{j+1} and u_n^{j+1} from equation (1.82). After choosing the initial approximation $u_1^{j+1,0}, u_2^{j+1,0}, \dots, u_{n-1}^{j+1,0}$, the next approximation can be computed by the iteration

$$\mathbf{\Gamma}(\boldsymbol{u}^{j+1,s})\Delta\boldsymbol{u}^{j+1,s} = -\boldsymbol{f}(\boldsymbol{u}^{j+1,s})\,, \tag{1.85}$$

$$\boldsymbol{u}^{j+1,s+1} = \boldsymbol{u}^{j+1,s} + \Delta\boldsymbol{u}^{j+1,s}\,, \tag{1.86}$$

where

$$\mathbf{\Gamma} = \begin{pmatrix} \dfrac{\partial f_1}{\partial u_1^{j+1}} & \dfrac{\partial f_1}{\partial u_2^{j+1}} & \cdots & \dfrac{\partial f_1}{\partial u_{n-1}^{j+1}} \\ \vdots & & & \vdots \\ \dfrac{\partial f_{n-1}}{\partial u_1^{j+1}} & \dfrac{\partial f_{n-1}}{\partial u_2^{j+1}} & \cdots & \dfrac{\partial f_{n-1}}{\partial u_{n-1}^{j+1}} \end{pmatrix}, \quad \boldsymbol{u}^{j+1} = \begin{pmatrix} u_1^{j+1} \\ u_2^{j+1} \\ \vdots \\ u_{n-1}^{j+1} \end{pmatrix}, \quad \boldsymbol{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \end{pmatrix}.$$

From (1.83) we can see that the Jacobi matrix $\mathbf{\Gamma}$ is three diagonal. The Newton's method converges almost always in a few iterations because we have a very good initial approximation u_i^j, $i = 1, 2, \dots, n-1$. The disadvantage is the need to evaluate the Jacobi matrix.

Up to now we considered one nonlinear partial differential equation. In most cases we have a system of partial differential equations and then the Jacobi matrix for the Newton's method is no longer three diagonal, it still has a band structure. We are going to show how appropriate linearization (sometimes called quasi-linearization) can be used to take the advantage of a three diagonal matrix.

Consider a system of two equations

$$\frac{\partial u_m}{\partial t} = \frac{\partial^2 u_m}{\partial x^2} + f_m(u_1, u_2)\,, \quad m = 1,\, 2\,.$$

Using the Crank-Nicolson method we get for $m = 1, 2$

$$\frac{u_{m,i}^{j+1} - u_{m,i}^{j}}{k} = \frac{1}{2}\Big(\frac{u_{m,i-1}^{j+1} - 2u_{m,i}^{j+1} + u_{m,i+1}^{j+1}}{h^2} + \frac{u_{m,i-1}^{j} - 2u_{m,i}^{j} + u_{m,i+1}^{j}}{h^2}\Big) + f_{m,i}^{j+\frac{1}{2}}\,. \tag{1.87}$$

If we replace the nonlinear term by the Taylor expansion

$$f_{m,i}^{j+\frac{1}{2}} \doteq f_m(\boldsymbol{u}_i^j) + \frac{\partial f_m(\boldsymbol{u}_i^j)}{\partial u_1} \frac{u_{1,i}^{j+1} - u_{1,i}^j}{2} + \frac{\partial f_m(\boldsymbol{u}_i^j)}{\partial u_2} \frac{u_{2,i}^{j+1} - u_{2,i}^j}{2}, \quad m = 1, 2,$$

we get actually the Newton's method (written in a different way) and the Jacobi matrix will have a band structure with five diagonals (with appropriate ordering of the unknowns and the equations). Doing only a partial linearization

$$\begin{aligned} f_{1,i}^{j+\frac{1}{2}} &\doteq f_1(\boldsymbol{u}_i^j) + \frac{\partial f_1(\boldsymbol{u}_i^j)}{\partial u_1} \frac{u_{1,i}^{j+1} - u_{1,i}^j}{2} \\ f_{2,i}^{j+\frac{1}{2}} &\doteq f_2(\boldsymbol{u}_i^j) + \frac{\partial f_2(\boldsymbol{u}_i^j)}{\partial u_2} \frac{u_{2,i}^{j+1} - u_{2,i}^j}{2}, \end{aligned} \tag{1.88}$$

the system of equations (1.87) splits into two independent subsystems, each one with a three diagonal matrix. The algorithm can be further improved by using $u_{1,i}^{j+1}$ for the computation of $f_{2,i}^{j+1/2}$ and to alternate the order of (1.88).

1.2.3 Method of lines

The method of lines is sometimes called the differential difference method. This name reflects the fact that we replace partial derivatives in one direction by difference formulas while we preserve them in the other direction and consider them as ordinary derivatives. We explain the method using a simple quasi-linear equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + R(u) \tag{1.89}$$

with boundary conditions of the first kind

$$u(0, t) = u(1, t) = 0, \qquad t > 0, \tag{1.90}$$

and the initial condition

$$u(x, 0) = \varphi(x), \qquad x \in (0, 1). \tag{1.91}$$

We replace the spatial derivative using a difference formula

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{x=x_i} \approx \frac{u(x_{i-1}, t) - 2u(x_i, t) + u(x_{i+1}, t)}{h^2}, \quad i = 1, 2, \ldots, n-1, \tag{1.92}$$

where $x_i = ih,\ i = 0, 1, 2, \ldots, n$. We denote

$$u(x_i, t) \doteq u_i(t)\,. \tag{1.93}$$

Along vertical lines (see Fig. 1.6) we get differential equations

$$\frac{\mathrm{d}u_i(t)}{\mathrm{d}t} = \frac{u_{i-1}(t) - 2u_i(t) + u_{i+1}(t)}{h^2} + R\big(u_i(t)\big)\,, \quad i = 1, 2, \ldots, n-1\,, \tag{1.94}$$

by substituting into equation (1.89). To satisfy boundary equations (1.90), it is easy to see that it must be

$$u_0(t) = 0\,, \qquad u_n(t) = 0\,. \tag{1.95}$$

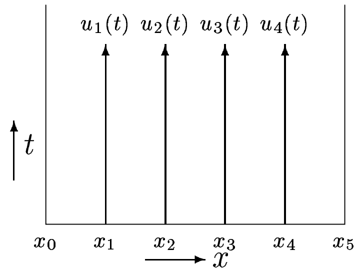

Figure 1.6: Method of lines

Initial condition (1.91) gives initial condition for ordinary differential equations (1.94):

$$u_i(0) = \varphi(x_i) = \varphi(ih)\,, \qquad i = 1, 2, \ldots, n-1\,. \tag{1.96}$$

Method of lines is easy even for more complicated problems. E.g. the equation

$$\frac{\partial u}{\partial t} = F\left(x, t, u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}\right) \tag{1.97}$$

can be transformed into a system of ordinary differential equations (without considering boundary conditions)

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = F\left(x_i, t, u_i, \frac{u_{i+1} - u_{i-1}}{2h}, \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}\right), \quad i = 1, 2, \ldots, n-1\,. \tag{1.98}$$

There is no principal difference between system (1.94) and system (1.98). The method of lines is a general approach both for linear and for nonlinear parabolic equations in two variables. A system of ordinary differential equations was discussed in chapter **??**. Not all numerical methods for ordinary differential equations are appropriate for solution of systems (1.94) or (1.98), but most of them can be used. The system (1.94) has two important properties that must be considered when choosing the integration method:

1. It is a large system. The number of ordinary differential equations may be several hundreds or thousands.

2. It is not necessary to take an extremely precise method for the numerical integration because even a precise solution of this system suffers the error of discretization of the spatial derivative. A method with a similar accuracy to that of the spatial discretization is appropriate.

Having a large number of equations it seems that complicated single step methods (Runge-Kutta methods of a high order) are not good. Using the Euler's method we get the simple explicit formula (1.22). The reader is invited to check this. To integrate this system of ordinary differential equations we often use the Runge-Kutta method of order 2 or 3 or a multi step method or a predictor - corrector method. Then the starting profiles must be computed using Runge-Kutta methods.

Using an explicit integration method brings the problem of stability. We cannot use an arbitrarily long integration step for the Runge-Kutta method. The stability condition must be investigated for each combination of PDE, spatial derivative approximation and integration method separately. Thus it is better to use some implicit method, but this requires iteration or to solve a system of liner algebraic equations for linear PDE.

Treatment of boundary conditions for the method of lines is similar to that of difference methods. We can again introduce a fictitious profile or we can use non-symmetric difference formulas for derivatives in the boundary conditions.

The method of lines with a single step integration is a good starting method for multi profile methods.

The number of nodes in the spatial coordinate is given by the desired accuracy. For problems where the solution in different regions of x differs considerably (e.g. for the wave or front solution, where u changes significantly in a very small interval of x) with an equidistant grid we must choose the step size so small to approximate this sharp transition well. Then small changes of u in the rest of the interval are approximated too precisely and the total number of nodes is too high. For such problems methods with adaptive regulation of non-equidistant spatial grid have been developed (see [?]).

1.3 Numerical solution of parabolic equations with three independent variables

As compared to problems solved above, here we have one more spatial coordinate, so we solve parabolic equations in two spatial and one temporal coordinates. The strategies are similar to those discussed above, numerical realization is more difficult, memory requirements are higher and the computation time is usually much longer.

A typical and the simplest linear parabolic equation in three dimensions is the equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \tag{1.99}$$

describing non-stationary heat conduction in a plane plate or non-stationary diffusion in a plane. Assume the initial condition

$$u(x,y,0) = \varphi(x,y), \quad x \in [0,1], \quad y \in [0,1] \tag{1.100}$$

and the boundary conditions

$$\begin{array}{lll} u(x,0,t) = 1, \quad u(x,1,t) = 0, & x \in [0,1], & t > 0, \\ u(0,y,t) = u(1,y,t) = 0, & y \in [0,1], & t > 0. \end{array} \tag{1.101}$$

This describes warming up a square plate with the initial temperature $\varphi(x,y)$, by keeping three sides at the zero temperature and one side at the unit temperature. In the region $0 \le x, y \le 1, t \ge 0$ we define a grid of nodes $x_i = ih$; $y_j = jh$; $t_m = mk$, where $i, j = 0, 1, \ldots, n$; $m = 0, 1, \ldots$. This grid is given by the step h in the two spatial coordinates x and y and by the temporal step k. Again we define

$$\alpha = \frac{k}{h^2}. \tag{1.102}$$

We denote the value of the numerical solution at a grid point

$$u_{i,j}^m \approx u(x_i, y_j, t_m) = u(ih, jh, mk). \tag{1.103}$$

To keep the formulas simple we define central difference operators of the second order δ_x^2 and δ_y^2 by

$$\delta_x^2 u_{i,j} = u_{i+1,j} - 2u_{i,j} + u_{i-1,j}\,, \quad \delta_y^2 u_{i,j} = u_{i,j+1} - 2u_{i,j} + u_{i,j-1}\,. \tag{1.104}$$

The simple explicit formula then becomes

$$\boldsymbol{u}^{m+1} = (1 + \alpha(\delta_x^2 + \delta_y^2))\boldsymbol{u}^m + \mathcal{O}(k^2 + kh^2)\,, \tag{1.105}$$

or in details

$$u_{i,j}^{m+1} = u_{i,j}^m + \alpha(u_{i-1,j}^m - 2u_{i,j}^m + u_{i+1,j}^m + u_{i,j-1}^m - 2u_{i,j}^m + u_{i,j+1}^m)\,. \tag{1.106}$$

The order of this method is clearly $\mathcal{O}(k + h^2)$ and each point in the new profile is computed from five points in the old profile. It is possible to derive a similar formula

$$\boldsymbol{u}^{m+1} = (1 + \alpha\delta_x^2)(1 + \alpha\delta_y^2)\boldsymbol{u}^m + \mathcal{O}(k^2 + kh^2)\,, \tag{1.107}$$

that uses 9 points in the old profile and that has the same order as formula (1.105). The reader is invited to rewrite (1.107) in the form similar to (1.106).

Equation (1.106) can be written by the scheme

$$\begin{tikzcd}
 & \alpha \arrow[d, dash] & \\
\alpha \arrow[r, dash] & (1-4\alpha) \arrow[r, dash] \arrow[d, dash] & \alpha \\
 & \alpha &
\end{tikzcd}$$

and similarly equation (1.107) by the scheme

$$\begin{tikzcd}
\alpha^2 \arrow[r, dash] \arrow[d, dash] & \alpha(1-2\alpha) \arrow[r, dash] \arrow[d, dash] & \alpha^2 \arrow[d, dash] \\
\alpha(1-2\alpha) \arrow[r, dash] \arrow[d, dash] & (1-2\alpha)^2 \arrow[r, dash] \arrow[d, dash] & \alpha(1-2\alpha) \arrow[d, dash] \\
\alpha^2 \arrow[r, dash] & \alpha(1-2\alpha) \arrow[r, dash] & \alpha^2
\end{tikzcd}$$

Formula (1.107) differs from (1.106) by including $\alpha^2\delta_x^2\delta_y^2\boldsymbol{u}^m$. These formulas are illustrated in Fig. 1.7. They both are of order $\mathcal{O}(k+h^2)$; the stability condition of the 5 point formula (1.106) is

$$\alpha \leq \frac{1}{4}\,, \tag{1.108}$$

while the 9 point formula (1.107) is stable for

$$\alpha \leq \frac{1}{2}\,. \tag{1.109}$$

If we take $\alpha = \frac{1}{6}$, the order increases to $\mathcal{O}(k^2+h^4)$ and this formula is appropriate for preparing precise starting profiles for multi profile methods (this is true for equation (1.99) only). Strict stability conditions (1.108) and (1.109) require small temporal step size k resulting in a long computation time which in turn limits the usability of explicit methods (1.105) and (1.107) for numerical solution of three dimensional problems. For four dimensional problems the stability restrictions are even stronger. On the other hand, a big advantage of explicit methods is their generality and ease of use (evaluation of recurrent formulas).

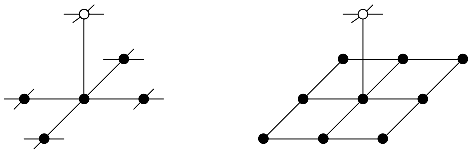

Figure 1.7: Illustration of explicit formulas (1.106) and (1.107)

Du Fort and Frankel derived a stable explicit method by taking (similarly as for a single spatial coordinate) the unstable Richardson formula

$$u_{i,j}^{m+1} = u_{i,j}^{m-1} + 2\alpha(\delta_x^2+\delta_y^2)u_{i,j}^m\,. \tag{1.110}$$

They replaced $u_{i,j}^m$ by the arithmetic mean $\frac{1}{2}(u_{i,j}^{m-1}+u_{i,j}^{m+1})$ and they got

$$(1+4\alpha)u_{i,j}^{m+1} = (1-4\alpha)u_{i,j}^{m-1} + 2\alpha(u_{i-1,j}^m + u_{i+1,j}^m + u_{i,j-1}^m + u_{i,j+1}^m)\,. \tag{1.111}$$

This equation is the Du Fort - Frankel method. The necessary starting values must be computed by another method. The convergence is guaranteed if the parameters of the grid satisfy certain additional condition, e.g. $k/h \to 0$. These conditions decrease the value of this method.

Similarly to the case of a single spatial variable it is possible to derive an explicit - implicit method where the new profile is computed by

$$u_{i,j}^{m+1} = (1 + \alpha(\delta_x^2 + \delta_y^2))u_{i,j}^m\,, \qquad m + i + j \text{ even}\,, \tag{1.112}$$

$$(1 - \alpha(\delta_x^2 + \delta_y^2))u_{i,j}^{m+1} = u_{i,j}^m\,, \qquad m + i + j \text{ odd}\,. \tag{1.113}$$

Formula (1.112) is an explicit one in the form of (1.106) and (1.113) is implicit, where we have all the values $u_{i-1,j}^{m+1}, u_{i+1,j}^{m+1}, u_{i,j-1}^{m+1}, u_{i,j+1}^{m+1}$ in the $(m+1)$-th profile computed by (1.112), thus (1.113) can be used for recurrent evaluation. This algorithm is illustrated in Fig. 1.8. It can be shown that this method is very similar to the Du Fort - Frankel method, so even here we need $k/h \to 0$.

For explicit method the temporal step size k is bounded by the stability condition or by the condition $k/h \to 0$. Thus implicit methods are often used instead. When used for problems described by (1.99) - (1.101) we need to solve a system of linear algebraic equations for $(n-1)^2$ unknowns in each step. The precise form of this system depends strongly on the type of the problem and on the method used; generally these systems are sparse because in each equation only a small number of unknowns appears. So for large n it is unreasonable to use finite methods (e.g. the Gauss elimination) because of memory and computation time demands.

It is possible to prepare a special algorithm with a finite method for a particular problem, but its applicability is restricted to this particular problem so it is not worth the effort.

Often the method called alternating direction implicit (ADI) is used involving two solutions of a three diagonal system of $(n-1)$ equations. The usage is similar to ADI for elliptic problems see chapter **??**. Here, however, the block relaxation ADI is not done for the same time level. Or the point relaxation (upper) method can be used with only a few (usually just one) relaxation cycle for each time level.

Of fundamental meaning is the Crank-Nicolson method (which is always stable for problems (1.99) - (1.101)) with a five point scheme

$$\left(1 - \frac{\alpha}{2}(\delta_x^2 + \delta_y^2)\right)\boldsymbol{u}^{m+1} = \left(1 + \frac{\alpha}{2}(\delta_x^2 + \delta_y^2)\right)\boldsymbol{u}^m + \mathcal{O}(k^3 + kh^2) \tag{1.114}$$

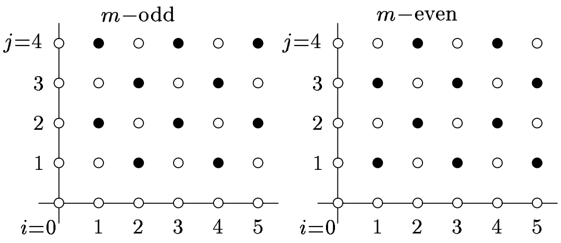

Figure 1.8: Explicit implicit method
● - values computed by (1.112)
○ - values computed by (1.113)
or from boundary condition

or a nine point scheme

$$\left(1-\frac{\alpha}{2}\delta_x^2\right)\left(1-\frac{\alpha}{2}\delta_y^2\right)\boldsymbol{u}^{m+1} = \left(1+\frac{\alpha}{2}\delta_x^2\right)\left(1+\frac{\alpha}{2}\delta_y^2\right)\boldsymbol{u}^{m}+\mathcal{O}(k^3+kh^2)\,. \quad (1.115)$$

They both are of order $\mathcal{O}(k^2+h^2)$. We get the ADI method by introducing additional profile $\boldsymbol{u}^+$ and by appropriate splitting the formula (1.114). This way we get the Peaceman-Rachford method

$$\begin{aligned}\left(1-\frac{\alpha}{2}\delta_x^2\right)\boldsymbol{u}^{+} &= \left(1+\frac{\alpha}{2}\delta_y^2\right)\boldsymbol{u}^{m}\,, & (1.116)\\ \left(1-\frac{\alpha}{2}\delta_y^2\right)\boldsymbol{u}^{m+1} &= \left(1+\frac{\alpha}{2}\delta_x^2\right)\boldsymbol{u}^{+}\,. & (1.117)\end{aligned}$$

If we eliminate the profile $\boldsymbol{u}^+$, from (1.116) and (1.117) by simple manipulation we get (1.114). Fig. 1.9 illustrates the Peaceman-Rachford method.

There are other methods using alternating directions (Djakon method, Douglas-Rachford method etc.). The interested reader is invited to use the original literature.

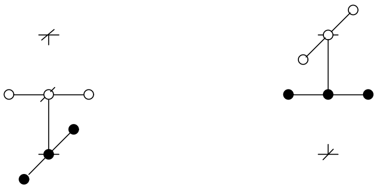

Figure 1.9: Peaceman-Rachford method
● - known values,
○ - unknown values

As the number of unknowns and the number of equations for implicit methods depends heavily on n, namely as $(n-1)^2$, we try to reduce the number of nodes while keeping the accuracy. This can be done by using more nodes to approximate the spatial derivatives e.g.

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{i,j} \approx \frac{-u_{i-2,j}+16u_{i-1,j}-30u_{i,j}+16u_{i+1,j}-u_{i+2,j}}{12h^2} \quad (1.118)$$

or at the boundary

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{1,j} \approx \frac{11u_{0,j} - 20u_{1,j} + 6u_{2,j} + 4u_{3,j} - u_{4,j}}{12h^2}\,. \tag{1.119}$$

This method is illustrated in Fig. 1.10 for both explicit and implicit methods. The order in x and y is $\mathcal{O}(h^4)$, again Crank-Nicolson averaging can be used. Difference formulas of a very high order can be constructed, using up to all $(n-1)$ values of u so that even for small n a good accuracy can be reached in certain cases.

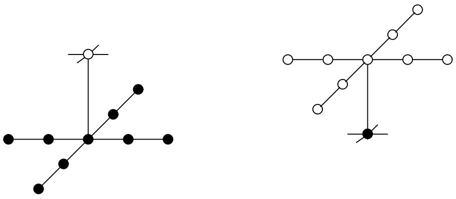

Figure 1.10: Explicit and implicit formula of higher order
● - known values,
○ - unknown values

Solution of nonlinear parabolic equations in three dimensions is similar to two dimensional problems, the resulting implicit linear problems are solved by some method given above, e.g. upper relaxation or ADI.

Similarly as for two independent variables, the method of lines can be used. Consider a quasi-linear equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + R(u)$$

with initial and boundary conditions (1.100), (1.101). Denoting $u_{i,j}(t) \doteq u(x_i, y_j, t)$, and using the simplest three point formulas we get

$$\begin{aligned} \frac{\mathrm{d}u_{i,j}}{\mathrm{d}t} &= \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2} + R(u_{i,j})\,, \\ u_{i,j}(0) &= \varphi(x_i, y_j)\,, \qquad i = 1, \dots, n-1\,,\ j = 1, \dots, n-1\,. \end{aligned}$$

The number of ordinary differential equations is in this case large, proportional to n^2. The advantage of this approach is that it is easy.

* * *

For further study see [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?], [?].