14 Generalizations of the simple Heat equation

In this Lecture, we will consider the following generalizations of the IBVP (12.1)–(12.3), based on the simple Heat equation:

- Derivative (Neumann and mixed-type) boundary conditions;
- The linear Heat equations with variable coefficients;
- Nonlinear parabolic equations.

14.1 Boundary conditions involving derivatives

Let us consider the modified IBVP (12.1)–(12.3) where the only modification concerns the boundary condition at $x = 0$:

$$u_t = u_{xx} \qquad 0 < x < 1, \quad t > 0\,; \tag{14.1}$$

$$u(x, t = 0) = u_0(x) \qquad 0 \le x \le 1\,; \tag{14.2}$$

$$u_x(0, t) + p(t)u(0, t) = q(t), \qquad t \ge 0\,; \tag{14.3}$$

$$u(1, t) = g_1(t), \qquad t \ge 0\,. \tag{14.4}$$

The boundary condition involving the derivative can be handled by either of the two methods described in Section 8.4 for one-dimensional BVPs. Below we will describe in detail how the first of those methods can be applied to the Heat equation. We will proceed in two steps, whereby we will first consider a modification of the simple explicit scheme (12.12) and then, a modification for the Crank–Nicolson method (13.8), for the boundary condition (14.3).

Modification of the simple explicit scheme (12.12)

For $n = 0$, i.e. for $t = 0$, U^0_m, $m = 0, 1, \dots, M-1, M$ are given by the initial condition (14.2). Then, discretizing (14.3) with the second order of accuracy in x as

$$\frac{U^0_1 - U^0_{-1}}{2h} + p^0 U^0_0 = q^0, \tag{14.5}$$

one immediately finds U^0_{-1} (because $p^0 \equiv p(0)$ and $q^0 \equiv q(0)$ are given by the boundary condition (14.3)). Thus, at the time level $n = 0$, one knows U^0_m, $m = -1, 0, 1, \dots, M-1, M$.

For $n = 1$, we first determine U^1_m for $m = 0, 1, \dots, M-1$ as prescribed by the scheme:

$$U^1_m = U^0_m + r\left(U^0_{m-1} - 2U^0_m + U^0_{m+1}\right) . \tag{14.6}$$

(Note that the value U^0_{-1} is used to determine the value of U^1_0.) Having thus found U^1_0 and U^1_1, we next find U^1_{-1} from the equation analogous to (14.5):

$$\frac{U^1_1 - U^1_{-1}}{2h} + p^1 U^1_0 = q^1. \tag{14.7}$$

Finally, U^1_M is given by the boundary condition (14.4).

For $n \ge 2$, the above step is repeated.

Remark We used the second-order accurate approximation for u_x in (14.5) and its counterparts for $n > 0$ because we wanted the order of the error at the boundary to be consistent with the order of the error of the scheme, which is $O(h^2)$.

Modification of the Crank–Nicolson scheme (13.8)

For $n = 0$, one finds U^0_{-1} from Eq. (14.5).

For $n = 1$, one has,
from the boundary condition (14.3):

$$\frac{U^1_1 - U^1_{-1}}{2h} + p^1 U^1_0 = q^1\,; \tag{14.7}$$

from the scheme (13.7):

$$U^1_m - \frac{r}{2}\left(U^1_{m-1} - 2U^1_m + U^1_{m+1}\right) = U^0_m + \frac{r}{2}\left(U^0_{m-1} - 2U^0_m + U^0_{m+1}\right), \qquad m = 0, 1, \ldots, M-1. \tag{14.8}$$

Equations (14.7) and (14.8) yield $M+1$ equations for the $M+1$ unknowns U^1_{-1}, U^1_0, U^1_1, …, U^1_{M-1}. This system of linear equations can, in principle, be solved. However, as we know from Sec. 8.4 (see Remark 2 there), the coefficient matrix in such a system will not be tridiagonal, which would preclude a straightforward application of the time-efficient Thomas algorithm. The way around that problem was also indicated in the aforementioned Remark. Namely, one needs to eliminate U^1_{-1} from (14.7) and the Eq. (14.8) with $m = 0$. For example, we can solve (14.7) for U^1_{-1} and substitute the result into Eq. (14.8) with $m = 0$. This yields:

$$U^1_0 - \frac{r}{2}\left(\left[U^1_1 - 2h(q^1 - p^1U^1_0)\right] - 2U^1_0 + U^1_1\right) = U^0_0 + \frac{r}{2}\left(\left[U^0_1 - 2h(q^0 - p^0U^0_0)\right] - 2U^0_0 + U^0_1\right), \tag{14.9}$$

where on the r.h.s. we have also used (14.5). Upon simplifying the above equation, one can write the linear system for the vector

$$\vec{\mathbf{U}}^n = \left[U^n_0,\, U^n_1,\, \ldots,\, U^n_{M-1}\right]^T, \qquad n = 0 \text{ or } 1$$

in the form:

$$A\vec{\mathbf{U}}^1 = B\vec{\mathbf{U}}^0 + \vec{\mathbf{b}}, \tag{14.10}$$

where

$$A = \begin{pmatrix} 1 + r(1 - hp^1) & -r & 0 & 0 & \cdot & 0 \\ -r/2 & 1+r & -r/2 & 0 & \cdot & 0 \\ 0 & -r/2 & 1+r & -r/2 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & -r/2 & 1+r & -r/2 \\ 0 & \cdot & 0 & 0 & -r/2 & 1+r \end{pmatrix} \tag{14.11}$$

and

$$B = \begin{pmatrix} 1 - r(1 - hp^0) & r & 0 & 0 & \cdot & 0 \\ r/2 & 1-r & r/2 & 0 & \cdot & 0 \\ 0 & r/2 & 1-r & r/2 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & r/2 & 1-r & r/2 \\ 0 & \cdot & 0 & 0 & r/2 & 1-r \end{pmatrix}, \qquad \vec{\mathbf{b}} = \begin{pmatrix} -rh(q^0 + q^1) \\ 0 \\ 0 \\ \cdot \\ 0 \\ \frac{r}{2}(g^0_1 + g^1_1) \end{pmatrix}. \tag{14.12}$$

System (14.10) with the tridiagonal matrix A given by (14.11) can now be efficiently solved by the Thomas algorithm.

For $n \geq 2$, the above step is repeated.

14.2 Linear parabolic PDEs with variable coefficients

Generalization of the explicit scheme (12.12) to such PDEs is straighforward. For example, if instead of the Heat equation (14.1) we have a PDE

$$u_t = a(x,t)u_{xx}, \tag{14.13}$$

then we use the following obvious discretization:

$$a(x,t)u_{xx} \quad \to \quad a_m^n \frac{\delta_x^2 U_m^n}{h^2}\,. \tag{14.14}$$

For the CN method, only slightly more effort is required. Note that the main concern here is to maintain the $O(\kappa^2+h^2)$ accuracy of the method. Maintaining this accuracy is achieved by using the (well-known to you by now) fact that

$$\begin{gathered} \frac{f(X+H)-f(X-H)}{2H} = f\,'(X) + O(H^2), \\ \text{or, equivalently,} \\ \frac{f(X+H)-f(X)}{H} = f\,'\left(X+\frac{H}{2}\right) + O(H^2), \end{gathered} \tag{14.15a}$$

where $f(X)$ is any sufficiently smooth function, and X can stand for either x or t (then H stands for either h or κ, respectively). Similarly, using the Taylor expansion, you will be asked in a QSA to show that

$$\frac{f(X+H)+f(X)}{2} = f\left(X+\frac{H}{2}\right) + O(H^2). \tag{14.15b}$$

In other words, we can use values $f(X)$ and $f(X+H)$ to approximate the values of the function and its derivative at $(X+\frac{H}{2})$ — the *midpoint* between X and $X+H$ — with accuracy $O(H^2)$. Using the idea expressed by (14.15), the schemes that we will list below can be shown to have the required accuracy of $O(\kappa^2+h^2)$.

For the PDE

$$u_t = a(x,t)u_{xx} + b(x,t)u_x + c(x,t)u, \tag{14.16}$$

we discretize the terms in a rather obvious way:

$$\begin{aligned} u_t \quad &\to \quad \frac{1}{\kappa}\delta_t U_m^n\,, \\ a(x,t)u_{xx} \quad &\to \quad \frac{1}{2h^2}\left(a_m^n \delta_x^2 U_m^n + a_m^{n+1}\delta_x^2 U_m^{n+1}\right)\,, \\ b(x,t)u_x \quad &\to \quad \frac{1}{4h}\left(b_m^n (U_{m+1}^n - U_{m-1}^n) + b_m^{n+1}(U_{m+1}^{n+1} - U_{m-1}^{n+1})\right)\,, \\ c(x,t)u \quad &\to \quad \frac{1}{2}\left(c_m^n U_m^n + c_m^{n+1} U_m^{n+1}\right)\,. \end{aligned} \tag{14.17}$$

Let us explain the origin of the expressions on the r.h.s.'es of the first and third lines above. The term on the first line approximates u_t with accuracy $O(\kappa^2)$ at the virtual node $(mh, (n+\frac{1}{2})\kappa)$; this is just a straightforward corollary of the second line of (14.15a). The term on the third line has two parts. The first part (with $1/(2h)$ factored into it) approximates bu_x with accuracy $O(h^2)$ at the node $(mh, n\kappa)$; this is just a straightforward corollary of the first line of (14.15a).

Similarly, the second term approximates bu_x with accuracy $O(h^2)$ at the node $(mh, (n+1)\kappa)$. Hence the average of these two parts approximates bu_x with accuracy $O(\kappa^2 + h^2)$ at the virtual node $(mh, (n+\frac{1}{2})\kappa)$; this is a straightforward corollary of (14.15b). (If you still have difficulty following these explanations, draw the stencil for the CN method and then draw all the nodes mentioned above.)

Often, the PDE arises in a physical problem in the form

$$\gamma(x,t)u_t = (\alpha(x,t)u_x)_x + \beta(x,t)u\,. \tag{14.18}$$

Instead of manipulating the terms so as to transform this to the form of (14.16) and then use the discretization (14.17), one can discretize (14.18) directly:

$$\begin{aligned}
\gamma(x,t)u_t \quad &\to \quad \frac{1}{2}(\gamma_m^n + \gamma_m^{n+1})\frac{1}{\kappa}\delta_t U_m^n, \quad \text{or} \quad \gamma_m^{n+\frac{1}{2}}\frac{1}{\kappa}\delta_t U_m^n, \\
(\alpha(x,t)u_x)_x \quad &\to \quad \frac{1}{2h}\left(\alpha_{m+\frac{1}{2}}^n \frac{\delta_x U_m^n}{h} - \alpha_{m-\frac{1}{2}}^n \frac{\delta_x U_{m-1}^n}{h}\right) + \frac{1}{2h}\left(\alpha_{m+\frac{1}{2}}^{n+1} \frac{\delta_x U_m^{n+1}}{h} - \alpha_{m-\frac{1}{2}}^{n+1} \frac{\delta_x U_{m-1}^{n+1}}{h}\right), \\
\beta(x,t)u \quad &\to \quad \frac{1}{2}\left(\beta_m^n U_m^n + \beta_m^{n+1} U_m^{n+1}\right).
\end{aligned} \tag{14.19}$$

Here we only explain the term on the r.h.s. of the second line, since the other two discretizations are analogous to those presented in (14.17). The first term in the first parentheses approximates au_x with accuracy $O(h^2)$ at the virtual node $((m+\frac{1}{2})h, n\kappa)$; this is a corollary of the second line of (14.15a). Similarly, the second term in the first parentheses approximates au_x with accuracy $O(h^2)$ at the virtual node $((m-\frac{1}{2})h, n\kappa)$. Consequently, the entire expression in the first parentheses with $1/h$ factored in it approximates $(\alpha u_x)_x$ with accuracy $O(h^2)$ at the node $(mh, n\kappa)$; this is a corollary of the first line of (14.15a). Finally, the entire expression on the r.h.s. of the second line of (14.19) approximates $(\alpha u_x)_x$ with accuracy $O(\kappa^2 + h^2)$ at the virtual node $(mh, (n+\frac{1}{2})\kappa)$.

14.3 Von Neumann stability analysis for PDEs with variable coefficients

Let us recall that the idea of the von Neumann analysis was to expand the error of the PDE with constant coefficients into a set of exponentials $\rho^n \exp(i\beta x) = \rho^n \exp(i\beta m h)$, each of which exactly satisfies the discretized PDE for a certain ρ. Note also that for both the simple explicit scheme (12.12) and the modified Euler-like scheme considered in Problem 4 for Homework # 12, the harmonics $\exp(i\beta m h)$ that would first become unstable should the stability condition for the scheme be violated, are those with the largest spatial frequency, i.e. with $\beta = \pi/h$ (see the figure at the end of Sec. 12.3). The same appears to be true for most other conditionally stable schemes.

Now let us consider the PDE (14.13) (or either of (14.16) and (14.18)) where *the coefficient(s) does(do) not vary too rapidly*. Then, such a coefficient can be considered to be *almost constant* in comparison to the highest-frequency harmonic that can potentially cause the instability. This simple consideration suggests that **for PDEs with sufficiently smooth coefficients, the von Neumann analysis can be carried out without any changes, while assuming that at each point in space and time, the coefficients are constant.** This approximation is known as the ***principle of frozen coefficients***; it was proposed by von Neumann around 1950.

For example, the principle of frozen coefficients yields the following stability criterion for the simple explicit method applied to (14.13):

$$r \le \frac{1}{2a(x,t)}\,. \tag{14.20}$$

This can be interpreted in the following two different ways.
(i) If the user decides to employ constant values for κ and h, and hence r, over the entire grid, then he/she should ensure that

$$r \le \frac{1}{2\max_{x,t} a(x,t)} \tag{14.21}$$

for the scheme to be stable.
(ii) If the user decides to vary the time step κ, then at every time level, κ is to be chosen so as to satisfy the condition

$$r(t) \le \frac{1}{2\max_{x} a(x,t)}\,. \tag{14.22}$$

The principle of frozen coefficients works often, but sometimes it can be strongly violated. One example of this is pointed out in Sec. 14.5.3.

Let us now point out another issue, unrelated to the above one. It can occur, e.g., for PDE (14.16) with $c \neq 0$. Namely, note that (14.16) may have exponentially growing solutions. For example, if each of a, b, and c is constant, then Eq. (14.16) has a solution $u = \exp(ct)$. If $c > 0$, this solution grows in time. In such a case, when carrying out the von Neumann analsysis, one should not require that $|\rho| \le 1$ for the stability of the scheme, because this would preclude obtaining the above exponentially growing solution. Instead, one should stipulate that the *largest*[25] value of $|\rho|$ satisfy (for the above example)

$$\max|\rho| = 1 + c\kappa + \text{ "smaller terms"}, \tag{14.23}$$

while all the other ρ's must be strictly less than the r.h.s. of (14.23) in absolute value. Equation (14.23) allows the (largest) amplification factor ρ corresponding to very *low*-frequency harmonics (i.e. those with $\beta \approx 0$) to be greater than 1 because of the *true* nature of the solution. If one does not include the term into $c\kappa$ into the modified definition of stability, Eq. (14.23), then it would not be possible to find a range for r where the scheme (12.12) could be stable.

For the above example of Eq. (14.16) with constant coefficients a, b, and c, the condition on r based on this modified stability criterion can be shown, by a straightforward but somewhat lengthy calculation, to be

$$r \le \frac{2 + c\kappa - \frac{1}{2}r^2b^2\pi^2h^4}{4a} \approx \frac{1 + (c\kappa/2)}{2a}, \tag{14.24}$$

i.e. almost the same as (14.20).

[25] if more than one value of ρ for a given β exists, as for a multi-level scheme

14.4 Nonlinear parabolic PDEs: I. Explicit schemes, and the Newton–Raphson method for implicit schemes

14.4.1 Explicit schemes

Explicit schemes for nonlinear parabolic PDEs can be consructed straightforwardly. For example, for the PDE

$$u_t = \left(u^2 u_x\right)_x , \tag{14.25}$$

the simple explicit scheme is

$$\frac{\delta_t U_m^n}{\kappa} = \frac{1}{h^2}\left[\left(\frac{U_{m+1}^n + U_m^n}{2}\right)^2 (U_{m+1}^n - U_m^n) - \left(\frac{U_m^n + U_{m-1}^n}{2}\right)^2 (U_m^n - U_{m-1}^n)\right) . \tag{14.26}$$

The von Neumann stability analysis can no longer be rigorously justified for (most) nonlinear PDEs, but it can often be justified approximately, if one assumes that the solution $u(x,t)$ (and hence its numerical counterpart U_m^n) *does not vary too rapidly*. This is analogous to the condition on the coefficients of linear PDEs mentioned in Sec. 14.3. Below we provide an intuitive explanation for this claim using (14.25) as a model problem, and then write down the stability criterion for that PDE.

Recall that to define stability in Lecture 4, we looked at the evolution of two "nearby" solutions: see Eq. (14.16) in Lecture 4 and Eq. (5.35) in Lecture 5. We defined a numerical method as stable if for a differential equation *whose analytical solution is stable*, the numerical solutions that were close initially remain close at all times. Let us, therefore, consider two "nearby" solutions, u and v, of (14.25). Their difference satisfies:

$$(u-v)_t = (u^2 u_x)_x - (v^2 v_x)_x . \tag{14.27}$$

Note that the r.h.s. of this equation is a counterpart of $f(x,y) - f(x,u)$ in (4.16). To approximate such a term, in Lectures 4 and 5 we *linearized it* about one of the solutions. Here, we have to linearize the r.h.s. of (14.27). Below we show how to do so, considering that the counterpart of the nonlinear function $f(x,u)$ — i.e. $(u^2 u_x)_x$ in (14.25) — depends on both u and u_x. In fact, we will first do it for an arbitrary function $f(u, u_x, u_{xx}, \ldots)$ and then illustrate it for $f = (u^2 u_x)_x$.

The Chain Rule for a function of several variables in the form that, most likely, you remember it from Calculus is:

$$\frac{d\, f\big(A(t), B(t), \ldots\big)}{dt} = \frac{\partial f}{\partial A}\frac{dA}{dt} + \frac{\partial f}{\partial B}\frac{dB}{dt} + \cdots . \tag{Chain Rule}$$

An equivalent form of the same rule written for differentials is:

$$df = f_A\, dA + f_B\, dB + \cdots . \tag{14.28a}$$

If differentials are replaced by small but finite increments, then (14.28a) becomes simply

$$\Delta f \approx f_A\, \Delta A + f_B\, \Delta B + \cdots . \tag{14.28b}$$

In what follows we will replace the "≈" with "=". Now substitute u for A, u_x for B, $(u-v) \equiv \Delta u$ for ΔA, and $(u_x - v_x) \equiv \Delta u_x$ for ΔB, etc., to obtain:

$$\Delta f(u, u_x, u_{xx}, \ldots) = f_u\, \Delta u + f_{u_x}\, \Delta u_x + f_{u_{xx}}\, \Delta u_{xx} \cdots . \tag{14.29a}$$

where

$$\Delta f(u, u_x, u_{xx}, \ldots) = f(u, u_x, u_{xx}, \ldots) - f(v, v_x, v_{xx}, \ldots)\,. \tag{14.29b}$$

We will now compute the r.h.s. of (14.27) based on Eqs. (14.29):

$$\Delta\big(u^2 u_x\big)_x = \big(\,(2uu_x)\,\Delta u + u^2\,Du_x\,\big)_x \overset{\text{Product Rule}}{=} 2u_x^2\Delta u + 2uu_{xx}\,\Delta u + 4uu_x\,\Delta u_x + u^2\,Du_{xx}\,. \tag{14.30}$$

In a QSA you will be asked to verify that if you first differentiate $(u^2u_x)_x$ and then apply (14.29), you will reobtain (14.30).

We are now ready to continue with the von Neumann analysis for the model problem (14.25). Combining (14.27) and (14.30) and rearranging terms in the latter equation, we obtain:

$$\Delta u_t = u^2\,\Delta u_{xx} + 2(u^2)_x\,\Delta u_x + (u^2)_{xx}\,\Delta u\,. \tag{14.31}$$

This equation is the *linearization of* (14.25) *on the background of solution* u, where we may now assume that u is the exact solution of (14.25). Thus, **a small deviation Δu between two solutions of a nonlinear PDE satisfies a linearization of that PDE on the background of an exact solution.** This is a universal fact.

Equation (14.31) is a *linear* equation for Δu that has the form of Eq. (14.16) if we pretend for the moment that we know the exact solution $u(x,t)$. Indeed, then, in the notations of (14.16), $a(x,t) \equiv u^2$, $b(x,t) \equiv 2(u^2)_x$, and $c = (u^2)_{xx}$. Then the stability condition is given by (14.24) with $a \equiv u^2$:

$$r \le \frac{1}{2u^2(x,t)}\,. \tag{14.32}$$

In practice, one knows $u(x,t)$ only at a given time level (and, of course, at previous levels), but not for all times in advance. Therefore, condition (14.32) means that the step size κ needs to be adjusted according to that condition at each time level so as to maintain the stability of the scheme.

14.4.2 Newton–Raphson method for implicit schemes

As far as *implicit* methods for nonlinear PDEs are concerned, there are quite a few possibilities in which such methods can be designed. Here we will discuss in detail an equivalent of the Newton–Raphson method considered in Lecture 8. In Sec. 14.5 we will introduce other methods whose counterparts we have not yet encountered in this course.

The main difficulty that one faces with the Newton–Raphson method is, similarly to Lecture 8, the need to solve systems of algebraic nonlinear equations to obtain the solution at the "new" time level. We will now discuss approaches to this problem using Eq. (14.25) as the model PDE.

To begin, we can use the following slight modification of scheme (14.19) for the PDE (14.18), where now $\alpha = u^2$, $\beta = 0$, and $\gamma = 1$:

$$\begin{aligned}
u_t \quad &\to \quad \frac{1}{\kappa}\delta_t U_m^n\,; \\
(u^2u_x)_x \quad &\to \quad \frac{1}{2h}\left(\frac{(U_m^n)^2 + (U_{m+1}^n)^2}{2}\,\frac{\delta_x U_m^n}{h} - \frac{(U_{m-1}^n)^2 + (U_m^n)^2}{2}\,\frac{\delta_x U_{m-1}^n}{h}\right) + \\
&\qquad \frac{1}{2h}\left(\frac{(U_m^{n+1})^2 + (U_{m+1}^{n+1})^2}{2}\,\frac{\delta_x U_m^{n+1}}{h} - \frac{(U_{m-1}^{n+1})^2 + (U_m^{n+1})^2}{2}\,\frac{\delta_x U_{m-1}^{n+1}}{h}\right).
\end{aligned} \tag{14.33}$$

Next, we substitute the discretized derivatives in (14.33) into Eq. (14.25). You will be asked to write down the resulting scheme in a homework problem. This scheme, which is just a nonlinear algebraic system of equations for U_m^{n+1} with $m = 1, \ldots, M-1$, can be solved by any of the iterative methods of Sec. 8.6. We will show the details for the Newton–Raphson method. In fact, that method is essentially the linearization used when arriving at Eq. (14.31) from Eq. (14.27); see especially Eq. (14.30). Namely, the Newton–Raphson method (as any other iterative method) requires one to use an initial guess for U_m^{n+1}, and an obvious candidate for such a guess is the known value of U_m^n. Then we let

$$\vec{\mathbf{U}}^{n+1} = \vec{\mathbf{U}}^n + \vec{\varepsilon}^{(0)}, \qquad \|\vec{\varepsilon}^{(0)}\| \ll \|\vec{\mathbf{U}}^n\|\,; \tag{14.34}$$

compare this with (8.84). Upon substituting (14.33) and (14.34) into (14.25) and discarding terms $O((\varepsilon^{(0)})^2)$, one obtains (see the explanation below):

$$\begin{aligned} \varepsilon_m^{(0)} - \frac{\kappa}{2h}\, \Bigg(& \left(\varepsilon_m^{(0)} U_m^n + \varepsilon_{m+1}^{(0)} U_{m+1}^n\right) \frac{\delta_x U_m^n}{h} - \left(\varepsilon_{m-1}^{(0)} U_{m-1}^n + \varepsilon_m^{(0)} U_m^n\right) \frac{\delta_x U_{m-1}^n}{h} + \\ & \frac{(U_m^n)^2 + (U_{m+1}^n)^2}{2}\, \frac{\delta_x \varepsilon_m^{(0)}}{h} - \frac{(U_{m-1}^n)^2 + (U_m^n)^2}{2}\, \frac{\delta_x \varepsilon_{m-1}^{(0)}}{h} \Bigg) \\ = \; & \frac{\kappa}{h} \left(\frac{(U_m^n)^2 + (U_{m+1}^n)^2}{2}\, \frac{\delta_x U_m^n}{h} - \frac{(U_{m-1}^n)^2 + (U_m^n)^2}{2}\, \frac{\delta_x U_{m-1}^n}{h} \right). \end{aligned} \tag{14.35}$$

Let us outline how the above expression is obtained; you will be asked to fill in the missing details in a homework problem. Although (14.35) can be obtained by direct multiplication of terms in (14.33), the easier, and "mathematically literate", way is to use the following form of the familiar Product Rule from Calculus:

$$\Delta(fg) \equiv (f + \Delta f)(g + \Delta g) - fg \approx f\Delta g + g\Delta f, \qquad \text{where} \quad \Delta f \ll f, \quad \Delta g \ll g\,. \tag{Product Rule}$$

As a preliminary step of the calculation that you will need to complete on your own, consider the term $(U_m^{n+1})^2$. Let us use the substitution (14.34) and denote $U_m^n \equiv f$ and $U_m^{n+1} = U_m^n + \varepsilon_m^{(0)} \equiv f + \Delta f$. Then, using the form of the Product Rule stated above with $g = f$, you can write

$$(U_m^{n+1})^2 \approx (U_m^n)^2 + 2U_m^n \varepsilon_m^{(0)}\,. \tag{14.36}$$

Note that this is nothing but a linearization of the function $\left(U_m^n + \varepsilon_m^{(0)}\right)^2$, as was considered in Sec. 14.4.1.

Next, consider the first term in the first large parentheses in (14.33) and denote $(\,(U_m^n)^2 + (U_{m+1}^n)^2\,)$ by f and $\delta_x U_m^n/h$ by g. (So, you denote f, g, Δf, and Δg anew each time that you need to use the Product Rule.) Then it is reasonable to use the following names for the corresponding quantities in the second large parentheses in (14.33):

$$(U_m^{n+1})^2 + (U_{m+1}^{n+1})^2 \;\equiv\; f + \Delta f, \qquad \delta_x U_m^{n+1}/h \;\equiv\; g + \Delta g, \tag{14.37}$$

where Δf and Δg are proportional to $\varepsilon^{(0)}$. At home you will obtain the form of Δf in (14.37) using Eq. (14.34). Directly from Eq. (14.34) you will be able to obtain the Δg. Then all that remains is to use the Product Rule on these f, g, Δf, and Δg. The remaining terms in (14.33) should be handled similarly.

From (14.35), the vector $\vec{\varepsilon}^{(0)}$ can be solved for in a time-efficient manner (since the coefficient matrix is tridiagonal). In most circumstances, one iteration (14.34) is sufficient, but if need be,

the iterations can be continued in complete analogy with the procedure described at the end of Sec. 8.6. Namely, we first compute

$$\vec{\mathbf{U}}^{(1)} \equiv \vec{\mathbf{U}}^n + \vec{\varepsilon}^{(0)} \tag{14.38}$$

and then seek a correction to *that* solution in the form

$$\vec{\mathbf{U}}^{n+1} = \vec{\mathbf{U}}^{(1)} + \vec{\varepsilon}^{(1)}, \qquad \|\vec{\varepsilon}^{(1)}\| \ll \|\vec{\mathbf{U}}^{(1)}\| \,. \tag{14.39}$$

Substituting (14.39) along with (14.33) into (14.25), we obtain an equation similar to (14.35):

$$\begin{aligned} \varepsilon_m^{(1)} - \frac{\kappa}{2h} \, & \left(\left(\varepsilon_m^{(1)} U_m^{(1)} + \varepsilon_{m+1}^{(1)} U_{m+1}^{(1)} \right) \frac{\delta_x U_m^{(1)}}{h} - \left(\varepsilon_{m-1}^{(1)} U_{m-1}^{(1)} + \varepsilon_m^{(1)} U_m^{(1)} \right) \frac{\delta_x U_{m-1}^{(1)}}{h} + \right. \\ & \left. \frac{(U_m^{(1)})^2 + (U_{m+1}^{(1)})^2}{2} \, \frac{\delta_x \varepsilon_m^{(1)}}{h} - \frac{(U_{m-1}^{(1)})^2 + (U_m^{(1)})^2}{2} \, \frac{\delta_x \varepsilon_{m-1}^{(1)}}{h} \right) \\ = \quad & -\varepsilon_m^{(0)} + \frac{\kappa}{2h} \left(\frac{(U_m^n)^2 + (U_{m+1}^n)^2}{2} \, \frac{\delta_x U_m^n}{h} - \frac{(U_{m-1}^n)^2 + (U_m^n)^2}{2} \, \frac{\delta_x U_{m-1}^n}{h} \right) + \\ & \frac{\kappa}{2h} \left(\frac{(U_m^{(1)})^2 + (U_{m+1}^{(1)})^2}{2} \, \frac{\delta_x U_m^{(1)}}{h} - \frac{(U_{m-1}^{(1)})^2 + (U_m^{(1)})^2}{2} \, \frac{\delta_x U_{m-1}^{(1)}}{h} \right) . \end{aligned} \tag{14.40}$$

Recall that here, $U^{(1)}$, U^n, and $\varepsilon^{(0)}$ are known, and one's goal is to solve this linear equation for $\varepsilon^{(1)}$. This can be done time-efficiently, because the coefficient matrix of the equation for $\varepsilon^{(1)}$ is tridiagonal. Once $\varepsilon^{(1)}$ has been found, one can define, and solve for, $\varepsilon^{(2)}$, etc. These iterations can be carried out in the above manner as many times as need be.

As we have seen above, the *strength* of the Newton–Raphson method is that it can be applied to programming an implicit numerical scheme for *any* nonlinear equation or system of equations. However, a *drawback* of this method is that it is quite cumbersome (see, e.g., (14.35) and (14.40)). Therefore, a considerable amount of research has been done on finding other methods which, on one hand, would to large extent retain the good stability properties of implicit methods while, one the other hand, would be much easier to program. Two such systematic alternatives to the Newton–Raphson method, which can be applied to a very wide class of equations and which do not require the solution of a system of nonlinear equations, are described in the next Section.

To conclude this Section, we will point out one issue that is specific to discretization of nonlinear differential equations.

Remark Let us continue using (14.25) as the model problem. Note that it can be written in an equivalent form:

$$u_t = \frac{1}{3} (u^3)_{xx} \,. \tag{14.41}$$

We can use the following discretization that has the accuracy of $O(\kappa^2 + h^2)$:

$$\frac{1}{\kappa} \delta_t U_m^n \; = \; \frac{1}{3} \cdot \frac{1}{2h^2} \left(\delta_x^2 (U^3)_m^n + \delta_x^2 (U^3)_m^{n+1} \right) \; ; \tag{14.42}$$

recall the definition (13.3) of the operator δ_x^2. The point we want to make here is that the nonlinear system (14.42) is *different* from the nonlinear system obtained upon substitution of (14.33) into (14.25)!

The issue we have encountered can be understood from the following simple example, pertaining to a single time level (hence we omit the superscript of the functions). Consider a nonlinear function u^3. Obviously,

$$(u^3)_x = 3u^2 u_x \,. \tag{14.43}$$

With the second-order accuracy, the l.h.s. can be discretized as, e.g.,

$$(u^3)_x \;\to\; \frac{(U_{m+1})^3 - (U_{m-1})^3}{2h} = \frac{(U_{m+1} - U_{m-1})(U_{m+1}^2 + U_{m+1}U_{m-1} + U_{m-1}^2)}{2h} \,. \tag{14.44}$$

Using the same — central-difference — formula to discretize the derivative on the r.h.s. of (14.43), one obtains

$$3u^2 \cdot u_x \;\to\; 3U_m^2 \cdot \frac{U_{m+1} - U_{m-1}}{2h}, \tag{14.45}$$

which, obviously, does not equal the r.h.s. of (14.44), although differs from it by an amount $O(h^2)$.

Thus, a nonlinear term can have several representations, which are equivalent in the continuous limit (like the l.h.s. and r.h.s. of (14.43)). However, these different representations, when discretized *using the same rule*, can still lead to distinct finite-difference equations, as illustrated by (14.44) and (14.45). For Hamiltonian equations, this ambiguity can be utilized to construct methods that preserve specified conserved quantities (like the symplectic Euler and Verlet methods almost preserved the Hamiltonian in Lecture 5). This is explored in a recent paper by M. Dahlby and B. Owren "A general framework for deriving integral preserving numerical methods for PDEs," posted next to this Lecture.

14.5 Nonlinear parabolic PDEs: II. Semi-implicit, implicit-explicit (IMEX), and other methods

14.5.1 A semi-implicit method

Let us present a simple alternative to the Newton–Raphson method using (14.41) as the model problem. With the accuracy of $O(\kappa^2)$, the u^3 term can be discretized as follows:

$$u^3 \;\to\; \left(U^{n+\frac{1}{2}}\right)^2 \frac{U^n + U^{n+1}}{2} \,. \tag{14.46}$$

The r.h.s. of (14.46) is now linear with respect to U^{n+1}, but the problem is that we do not yet know $U^{n+\frac{1}{2}}$. The latter can be approximated by an explicit method that should have the *local* truncation error $O(\kappa^2)$, and hence the global accuracy of one order less, i.e. only $O(\kappa)$. That is, one can first compute $U^{n+\frac{1}{2}}$ and then use it as a known value in (14.46). A simple, $O(\kappa^2)$-accurate way to compute $U^{n+\frac{1}{2}}$ is by a multi-step method similar to (3.4):

$$U^{n+\frac{1}{2}} = U^n + \frac{1}{2}\,(U^n - U^{n-1}) = \frac{3}{2}U^n - \frac{1}{2}U^{n-1} \,. \tag{14.47}$$

Then the scheme

$$\delta_t U_m^n = \frac{r}{6}\delta_x^2 \left(\left(U_m^{n+\frac{1}{2}}\right)^2 (U_m^n + U_m^{n+1}) \right), \tag{14.48}$$

becomes an implicit scheme for the *linear* equation.

Method (14.48), (14.47) is a member of a large class of *semi-implicit* methods. It can be straightforwardly generalized to the following class of equations:

$$u_t = a(u, u_x, x, t)u_{xx} + b(u, u_x, x, t)u_x \,, \tag{14.49}$$

where, as stated above, the coefficients a and b may depend on the solution u and its derivative u_x. (Further generalizations of this form are possible, but for the purpose of our brief discussion, form (14.49) is sufficient.) An extension of scheme (14.48), (14.47) for (14.49) is:

$$\begin{aligned} \frac{\delta_t U_m^n}{\kappa} \;=\; & a\left(U_m^{n+\frac12}, (U_m^{n+\frac12})_x, x_m, t_{n+\frac12}\right) \frac{(U_m^n)_{xx} + (U_m^{n+1})_{xx}}{2} \;+ \\ & b\left(U_m^{n+\frac12}, (U_m^{n+\frac12})_x, x_m, t_{n+\frac12}\right) \frac{(U_m^n)_{x} + (U_m^{n+1})_{x}}{2} \,, \end{aligned} \tag{14.50}$$

where $U_m^{n+\frac12}$ is given by (14.47) and $(U_m^n)_x$ denotes the second-order accurate finite-difference approximation of $u_x(x_m, t_n)$, etc..

Since this scheme is not fully implicit, it cannot be unconditionally stable (see Theorem 4.2 at the end of Lecture 4). However, one can show that it is unconditionally stable *on the background of the constant solution*, $u = C$ where C is any constant, of (14.49). To show this, let us consider an ansatz

$$U_m^n = C + \epsilon\, \rho^n\, e^{i\beta m h} \,, \tag{14.51}$$

where ρ is the amplification factor in the von Neumann analysis and $\epsilon \ll 1$ indicates smallness of the perturbation to the exact solution $u = C$. Then, according to (14.47),

$$U_m^{n+\frac12} = C \;+\; \epsilon \left(\frac{3}{2}\rho - \frac{1}{2}\right) \rho^{n-1} e^{i\beta m h} \,. \tag{14.52}$$

When, however, (14.52) is substituted into (14.50) and terms of order $O(\epsilon^2)$ and higher are neglected, the $O(\epsilon)$-term from (14.52) drops out, since the terms that it multiplies are already $O(\epsilon)$ (the C-term is absent from $(U_m^n)_{xx}$ and similar terms due to the x-derivative). Then, in the equation that results from the stability analysis, a and b take on the forms $a(C, 0, x_m, t_{n+\frac12})$ and $b(C, 0, x_m, t_{n+\frac12})$, which is the same as in the case of linear parabolic PDEs with variable coefficients, considered in Section 14.2. Thus, method (14.50) is unconditionally stable *on the background of the constant solution* of Eqs. (14.49). While it may be unstable (for "too large" a time step) on the background of other, non-constant, solutions, it may still be a good first method to try since it is much easier to implement than the Newton–Raphson method.

14.5.2 The idea behind Implicit–Explicit (IMEX) methods

IMEX methods present another attractive alternative to the Newton–Raphson method because, as the semi-implicit method above, they also do not require the solution of a system of nonlinear algebraic equations. They do require the step size κ to be restricted since they are not fully implicit and hence cannot be unconditionally stable (see Lecture 4). However, such a restriction can be significantly weaker than that for a fully explicit method. Below we present only the basic idea of IMEX methods. A more detailed, and quite readable, exposition, as well as references, can be found in Section IV.4 of the book by W. Hundsdorfer and J.G. Verwer, "Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations," (Springer Series in Comput. Math., vol. 33, Springer, 2003).

The idea behind IMEX methods can be explained without any explicit reference to spatial variables. Let the evolution equation that we want to solve have the form

$$u_t = F\left(u(t),t\right) \equiv F_0(u(t),t) + F_1(u(t),t)\,, \tag{14.53}$$

where F_0 is a non-stiff term suitable for explicit time-integration and F_1 is a stiff term that requires implicit treatment. Usually, F_0 and F_1 include, respectively, the advection and diffusion terms (i.e., the second and first terms in (14.16) or (14.49), respectively; recall from Lecture 12 that the simple Heat equation $u_t = u_{xx}$ is a stiff problem). The last term in (14.16), which can be generalized to be some nonlinear function $C(u)$, can belong to either F_0 or F_1. It is usually referred to as the reaction term because it often describes chemical reactions. To make the splitting (14.53) useful for a numerical implementation, which means avoiding the solution of a system of nonlinear equations, it suffices to require that F_1 be linear in u. Below we will proceed with this assumption, but at the end of our discussion will mention a generalization where F_1 may contain nonlinear terms. Let us also note that our consideration applies equally well both to a single Eq. (14.53) and to a system of coupled equations whose r.h.s. can be split as a sum of non-stiff and stiff terms.

A simple first-order accurate IMEX method for (14.53) is:

$$\frac{U^{n+1}-U^n}{\kappa} = F_0(U^n,t_n) + (1-\theta)F_1(U^n,t_n) + \theta F_1(U^{n+1},t_{n+1})\,, \tag{14.54}$$

where θ is a parameter, as in Lecture 13. Note that since, by design, F_1 depends on U^{n+1} linearly, scheme (14.54) does not require its user to solve any nonlinear algebraic equations. The stability analysis for this scheme is done as follows. Instead of the model equation

$$u_t = \lambda u\,, \tag{4.15}$$

which does not distinguish between the stiff and non-stiff parts, one considers a model equation

$$u_t = \lambda_0 u + \lambda_1 u\,, \tag{14.55}$$

where λ_0 and λ_1 correspond to F_0 and F_1. Substituting $U^n = \rho^n$ into scheme (14.54) applied to Eq. (14.55), one finds that

$$\rho \equiv \rho(z_0,z_1) = \frac{1+z_0+(1-\theta)z_1}{1-\theta z_1}\,, \tag{14.56}$$

where $z_0 = \lambda_0\kappa$ and $z_1 = \lambda_1\kappa$. As usual, one requires

$$|\rho(z_0,z_1)| < 1 \tag{14.57}$$

for stability. Inequality (14.57) turns out to impose a set of *two* conditions on the time step κ. We will now explain that this set of conditions can be interepreted in two different ways, depending on what one knows about the family of equations that one wants to solve.

First interpretation of (14.57)

Suppose one has to design a method (14.54) that should be applicable to equations of the form (14.53) where parameters of F_0 cause the values of λ_0 to be anywhere (i.e., *not* just on the negative real line) in some bounded region of the left-half complex plane. Then one should

insist on using the full stability region of the explicit method, i.e., to have $|1+z_0|<1$. Thus, the first condition in the set is:

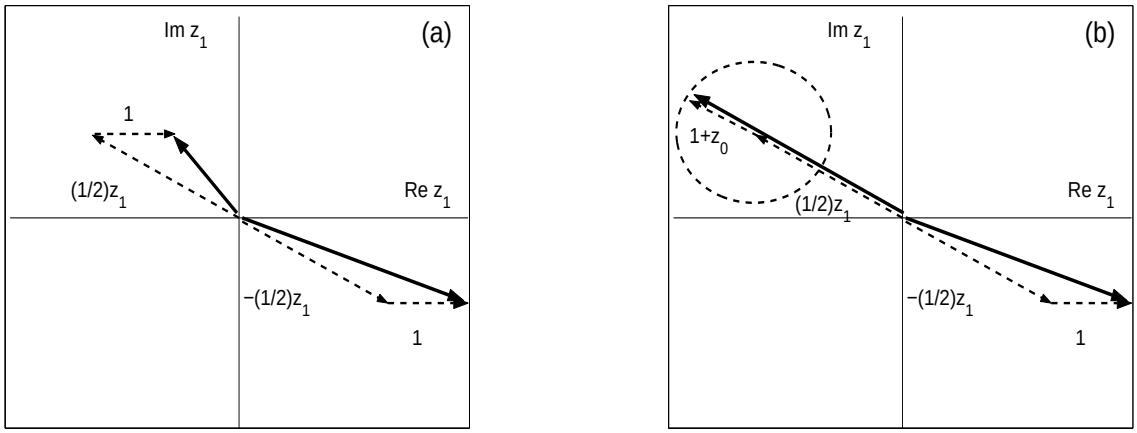
$$\mathcal{D}_0: \qquad |1+\lambda_0\kappa|<1\,, \tag{14.58}$$

where λ_0 has been described in the previous sentence. An example of such a situation is when F_0 contains terms describing nonlinear, but non-stiff, reaction or advection, while F_1 contains the simple diffusion term u_{xx}, for which all λ_1's lie on the negative real axis (see Problem 2 in HW 12). It turns out that enforcing both conditions, (14.57) and (14.58) imposes a restriction on the values of z_1, which in the absence of (14.58) (or, equivalently, if $z_0=0$ in (14.57)) would not have occurred. Let us now explain *why* this restriction on the values of z_1 occurs.

For the sake of argument, consider the value $\theta=1/2$ in (14.54), which would lead to the Crank–Nicolson scheme if F_0 were absent. Since that scheme is nothing but the implementation of the modified implicit Euler method for the Heat equation (see Sec. 13.1), its stability region is the entire left-half complex plane (recall the result of Problem 7 in HW 4). That is,

$$\frac{\left|1+\frac{1}{2}z_1\right|}{\left|1-\frac{1}{2}z_1\right|}\le 1 \tag{14.59}$$

whenever $\mathrm{Re}\,(z_1)\le 0$. Graphically, this is illustrated in Figure (a) below. There, the expressions in the numerator and denominator on the l.h.s. of (14.59) are depicted by the thick vectors in the left-half and right-half planes, respectively. It is clear that the ratio of the lengths of those vectors is indeed *always* less than one.



On the other hand, the stability condition (14.57) with $\theta=1/2$ is

$$\frac{\left|(1+z_0)+\frac{1}{2}z_1\right|}{\left|1-\frac{1}{2}z_1\right|}\le 1, \tag{14.60}$$

which must hold for *all* z_0 such that $|1+z_0|\le 1$. As illustrated in Figure (b) above, condition (14.60) can be violated for some of such z_0 *unless* $\mathrm{Im}\,(z_1)=0$. Thus, if one insists on having the full stability region for the explicit part of the IMEX method (14.54), the stability region of this method *with respect to its implicit part* can be *less* than the corresponding stability region of (14.54) with $F_0\equiv 0$.

In general, in this case one can show that the stability condition (14.57) yields the inequality

$$\mathcal{D}_1: \qquad 1+|(1-\theta)z_1| \;<\; |1-\theta z_1|\,. \tag{14.61}$$

This is the second condition in the set. That is, (14.58) and (14.61) together are equivalent to (14.57).

With some effort, one can further show from (14.61) that the unconditional stability of the IMEX method (14.54) is attained only for $\theta = 1$. For $\theta < 1/2$, scheme (14.54) is unstable. (This should be contrasted with the situation when $F_0 \equiv 0$, for which method (14.54) with $\theta < 1/2$ is conditionally stable, as we showed in Sec. 13.3.) For $\theta = 1/2$, its stability region $\mathcal{D}_1$, given by (14.61), collapses onto the negative real axis: $z_1 < 0$. (Above, we have illustrated this graphically.) However, already for θ just slightly exceeding the critical value of 1/2, the stability region $\mathcal{D}_1$ becomes a sector with a significantly nonzero angle α on both sides of the negative real axis; for example, $\alpha \approx 25^o$ and $\alpha > 50^o$ for $\theta = 0.51$ and $\theta = 0.6$, respectively (see, e.g., Fig. 4.1 in the book by Hundsdorfer and Verwer cited above).

Second interpretation of (14.57)

Alternatively, suppose that (14.53) is a *system of coupled equations* for variables $u^{(1)}$, $u^{(2)}, \ldots$, and suppose that F_1 contains both the diffusion term and the stiff part of the reaction term. Then, the eigenvalues $\lambda_1^{(1)}$, $\lambda_1^{(2)}, \ldots$ (and hence the corresponding values $z_1^{(1)}$, $z_1^{(2)}, \ldots$) of the Jacobian matrix $\partial(F^{(1)}, F^{(2)}, \ldots)/\partial(u^{(1)}, u^{(2)}, \ldots)$ (see Sec. 5.4 in Lecture 5) can be found anywhere in the left half of the complex plane, i.e. $\mathrm{Re}\, z_1^{(j)} \le 0$ for all j. Thus, one may want to know for which complex z_0 one can fulfill condition (14.57) given that z_1 can be allowed anywhere in the left-half complex plane.

Similarly to the previous case, the corresponding nonempty region $\mathcal{D}_0$ exists only for $\theta \ge 1/2$. That is, if $\theta < 1/2$, then the IMEX method (14.54) where z_1 can be found anywhere in the left-half plane, *is unstable for any* $z_0 \neq 0$ *with* $\mathrm{Re}\,(z_0) \le 0$! (This should be contrasted with the situation when $F_1 \equiv 0$, for which method (14.54) — i.e., the simple Euler method — is conditionally stable.) For $\theta < 1$, the stability region $\mathcal{D}_0$ of the IMEX method is smaller than the region $|1 + z_0| < 1$, which would result in the absence of the F_1 term in (14.53). For $\theta = 1/2$, the region $\mathcal{D}_0$ collapses into the segment $-2 < z_0 < 0$ along the negative real axis, while for $\theta = 1$, the stability region of the explicit Euler method, e.g., $\mathcal{D}_0(\theta = 1) = \{\text{All } z_0 \text{ such that } |1 + z_0| < 1\}$, is recovered (see, again, Fig. 4.1 in the book by Hundsdorfer and Verwer).

The book by Hundsdorfer and Verwer provides an overview of higher-order accurate members of the IMEX family, which are preferred in practice over the lowest-order method (14.54). Among them are, for example, IMEX Runge–Kutta and multistep IMEX methods. Below we will list two second-order accurate IMEX methods and briefly comment on their properties.

Second-order IMEX-Adams methods have the form:

$$\begin{aligned} \frac{U^{n+1} - U^n}{\kappa} \;=\; & \frac{3}{2}F_0(U^n, t_n) - \frac{1}{2}F_0(U^{n-1}, t_{n-1}) \;+ \\ & \theta F_1(U^{n+1}, t_{n+1}) + \left(\frac{3}{2} - 2\theta\right) F_1(U^n, t_n) + \left(\theta - \frac{1}{2}\right) F_1(U^{n-1}, t_{n-1})\,. \end{aligned} \tag{14.62}$$

If we insist that it be stable for all z_1 in the left-half plane, its stability region with respect to z_0 depends on θ (similarly to what we discussed above in the second interpretation of (14.57)). For example, for $\theta = 1/2$, this method is stable only when z_0 belongs to a segment along the negative real axis, $z_0 \in [-1,\ 0]$. For $\theta = 1$, the stability region of the second-order Adams–Bashforth method is recovered (see Problem 4 in HW 4). For $\theta = 3/4$, the stability region is an oval part of whose boundary follows the imaginary axis most closely (out of all values of θ). Thus, the IMEX-Adams method with $\theta = 3/4$ is preferred for equations that have z_0 both on, and to the left of, the imaginary axis.

If z_0 are known to lie *only* on the imaginary axis, then the so-called IMEX-CNLF (Crank–Nicolson Leap-frog) method can be used. Its scheme is:

$$\frac{U^{n+1}-U^{n-1}}{2\kappa} = F_0(U^n, t_n) + \frac{1}{2}\Big(F_1(U^{n+1}, t_{n+1}) + F_1(U^{n-1}, t_{n-1})\Big)\,. \tag{14.63}$$

This scheme is stable for all z_1 in the left-half plane and for $z_0 \in [-i,\ i]$. Examples of the non-stiff term F_0 for which λ_0 lies on the imaginary axis is the advection term $b(x,t,u)u_x$. (It is beyond the scope of this course to explain why this is so, but if you are familiar with Fourier analysis, you may figure it out on your own.) Thus, equations of the form (14.49) where a is independent of u can be solved by this method. Another example is the Nonlinear Schrödinger equation (14.64) below.

Finally, we note that the same considerations can often be generalized when F_1 is not a linear function of u. For example, consider Eq. (14.49) where now the coefficient a *does* depend on u. Then one can replace the implict integration in (14.54) with an analogue of the semi-implicit method (14.50). This would still result in the equation for U^{n+1} being linear, and hence easily solvable. Stability properties of such a method are not, however, clear, and may need to be verified by numerical experiments.

14.5.3 Comments on other methods

Let us mention a popular method called a split-step method, which we will illustrate with the example of the celebrated *Nonlinear Schrödinger equation*:

$$iu_t + u_{xx} + 2|u|^2 u = 0, \qquad (\text{note the } i=\sqrt{-1} \text{ in front of } u_t) \tag{14.64}$$

which appears in a great many applications involving propagation of wave packets. The split-step method is based on the observation that the linear and nonlinear parts of this equation can be solved *exactly* (we do not need to consider here *how* this can be done). Then the split-step algorithm is:

$$\begin{aligned}
&\text{Given } \ U^n(x) \equiv u(x,t_n),\\
&\text{Solve } \ iu_t + u_{xx} = 0 \ \text{ from } t_n \text{ to } t_{n+1}; \qquad \Rightarrow \ \text{ get } \ U^{\text{aux}};\\
&\text{Using } \ U^{\text{aux}} \ \text{ as the initial condition,}\\
&\text{Solve } \ iu_t + 2u|u|^2 = 0 \ \text{ from } t_n \text{ to } t_{n+1}; \ \Rightarrow \ \text{ get } \ U^{n+1}.
\end{aligned} \tag{14.65}$$

The split-step method, being explicit, is only conditionally stable. Its numerical stability for a constant-amplitude solution of the Nonlinear Schrödinger equation known as a plane-wave solution:

$$u = A\,e^{2iA^2t}, \qquad \text{where } A \text{ is a real constant,} \tag{14.66}$$

was first considered in a paper by A. Weideman and B. Herbst "Split-step methods for the solution of the nonlinear Schrödinger equation," SIAM J. Numer. Anal., vol. 23, pp. 485 - 507 (1986). The Nonlinear Schrödinger equation has many other solutions, the most well-known of which is the soliton:

$$u = A\,\mathrm{sech}(Ax)\ e^{iA^2t}, \tag{14.67}$$

which hass a bell-like (i.e., localized) profile in x. Numerical stability of *this* solution obtained by the split-step method was considered by me. The most remarkable conclusion of that analysis is that the principle of frozen coefficients, mentioned in Sec. 14.3, is *strongly* violated. For example, no prediction of the numerical stability or instability of the soliton (14.67) can be

made based on the knowledge of numerical stability or instability of the plane wave solution (14.66).

The last class of methods that we will mention are valuable only for PDEs that possess conserved quantities, like energy. Usually, such equations are hyperbolic PDEs or parabolic PDEs with "imaginary time", like the Nonlinear Schrödinger equation (14.64). Such equations are multi-dimensional counterparts of the harmonic oscillator equation. There are classes of numerical schemes that preserve some (or, in rare cases, all!) of the conserved quantities of those equations. Such schemes are relatives of symplectic methods for ODEs, discussed in Lecture 5. One can read about those conservation-laws-based schemes in, e.g., a textbook by J.W. Thomas, "Numerical partial differential equations: Conservation laws and elliptic equations" (Springer, 1999); see also the paper by M. Dahlby and B. Owren mentioned at the end of Sec. 14.4 and posted next to this Lecture. Let us stress that "true" parabolic equations, like the Heat equation or, more generally, any equation with diffusion in real-valued time, do *not* have conserved quantities like energy, and hence conservation-laws-based schemes are not applicable to them.

14.6 Questions for self-assessment

1. In (14.5) and (14.7), why did we *not* use the simpler discretization

$$\frac{U_1^n - U_0^n}{h} + p^n U_0^n = q^n, \qquad n = 0, 1,$$

   which would have eliminated the need to deal with the solution U_{-1}^n at the virtual node?

2. Be able to explain the idea(s) behind handling the derivative boundary condition for both the simple explicit and Crank–Nicolson schemes.

3. Make sure you can obtain (14.9) and hence (14.10)–(14.12).

4. Obtain (14.15b). *Hint*: Expand about $X + (H/2)$, not X.

5. Explain *without calculations* that discretization (14.17) produces a scheme of the accuracy stated in the text. (Drawing the stencil should help.)

6. Same question about (14.19).

7. What condition on the variable coefficients of a linear PDE should hold in order for the von Neumann stability analysis to proceed along the same lines as for the simple Heat equation? Why?

8. Describe two ways in which the person who is numerically solving PDE (14.13) may use the stability condition (14.20).

9. When and why does one need to modify the stability criterion to be (14.23)?

10. What is the order of accuracy of scheme (14.26)?

11. Make sure you can derive the r.h.s. of (14.30).

12. Verify the statement made immediately after (14.30).

13. Explain qualitatively (i.e., without calculations) that discretization (14.33) produces a scheme of the accuracy $O(\kappa^2 + h^2)$. (Drawing the stencil should help.)

14. What is the main difficulty in solving nonlinear PDEs by implicit methods?

15. Using discretizations (14.33) as an example, explain the idea behing the Newton–Raphson method when applied to nonlinear PDEs.

16. Describe the issue about discretization of nonlinear terms, pointed out in Remark 1.

17. Describe the idea behind the semi-implicit method presented in Sec. 14.5.

18. Explain why the r.h.s. of (14.47) approximates the l.h.s. of that equation.

19. Obtain (14.56).

20. What are two possible interpretations of (14.57)?

21. Make sure you can follow the argument made around condition (14.60).

22. Why does method (14.62) have the name 'Adams' in it?

23. When can method (14.63) be used?