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 1: 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 straightforward. 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 fact, verified by the Taylor expansion around X, that

$$\frac{f(X+H)-f(X-H)}{2H} = f\,'(X) + \frac{H^2}{6} f\,'''(X) + O(H^4), \tag{14.15a}$$

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). You have actually seen this formula in Lectures 3, 8, and 13. If we now "center" the above formula on $(X+H/2)$ instead, then it becomes:

$$\frac{f(X+H)-f(X)}{H} = f\,'\left(X+\frac{H}{2}\right) + \frac{H^2}{24} f\,'''\left(X+\frac{H}{2}\right) + O(H^4). \tag{14.15b}$$

Similarly, using the Taylor expansion, you will be asked in a QSA to show that

$$\frac{f(X+H)+f(X)}{2} = f\left(X+\frac{H}{2}\right) + \frac{H^2}{8} f\,''\left(X+\frac{H}{2}\right) + O(H^4). \tag{14.15c}$$

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:

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

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 (14.15b). The term on the third line has two

parts. The first part approximates bu_x with accuracy $O(h^2)$ at the node $(mh, n\kappa)$; this is just a straightforward corollary 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 (14.15c). (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:

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

Here we only outline the explanation of 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 (14.15b). 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 (14.15a).[42] 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 $\rho^n \exp(i\beta x) = \rho^n \exp(i\beta mh)$, 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 $\exp(i\beta mh)$ 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**

[42] Here, more accurate work, still based on (14.15), is needed to show that $(O(h^2) - O(h^2))/h = O(h^2)$ and *not* $O(h)$, as it generically would have been assumed. You can do this in a bonus problem.

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 analysis, 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* [43] 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).

[43] 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 constructed straightforwardly. As the example of a nonlinear "Heat-like" equation, in this Section (i.e., in all subsections of 14.5), we will use the PDE

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

The simple explicit scheme, based on (14.19), for it is:

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

If you are confused as to how this scheme can be justified, review Eqs. (14.15) and the explanation found after (14.19). Understanding this will be again required when reading the next subsection. So: pause, take a deep breath, and verify, following the guidelines two sentences ago, that the r.h.s. of (14.26) indeed yields a $O(h^2)$-accurate discretization of the r.h.s. of (14.25).

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 the model problem, and then write down the stability criterion for that PDE.

Importantly, in the process, we will show that ***for nonlinear equations, the error no longer satisfies the same equation as the solution of the PDE***. This is in contrast to what we stated about the error for a linear PDE; see Section 12.3. Instead, ***the error of a nonlinear PDE satisfies a <u>linearized</u> version of that equation***.

We will now work out an example.

Recall that to define stability in Lecture 4, we looked at the evolution of two "nearby" solutions: see Eq. (4.16) in Lecture 4 and Eq. (5.40) 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 is:[44]

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

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

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

where subscripts denote partial derivatives. 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:

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

where

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

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

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

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 re-obtain (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, again:

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, 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, in the spirit of the **principle of frozen coefficients**, the stability condition is given by (14.24) with $a\equiv u^2$:

$$r \le \frac{1+(u^2)_{xx}\kappa/2}{2u^2(x,t)}\,. \tag{14.32}$$

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.

[44] You studied this in Calculus III.

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. In the Appendix, we present the general methodology of how this can be done for equations of the form

$$u_t = f(u, u_x, \ldots), \tag{14.33}$$

while below in this subsection we will show how this general methodology can be applied to the particular model, Eq. (14.25).

To ensure the least painful experience when reading the text in the remainder of this subsection, you should first *review* the material in the following order. First, review section 14.2. Next, review subsection 14.4.1, where you will need to make sure that you understand both Eq. (14.26) and the derivation of the linearized equation for Δu. (**Do not proceed without understanding** scheme (14.26), as the cumbersome formulas below will simply not make sense.) Finally, work through the Appendix; after that, the cumbersome formulas in the remainder of this subsection should no longer look to you as gory as they may look now.

To begin (that is, after you have done the reading indicated in the previous paragraph), we can use the following slight modification of scheme (14.19) for the PDE (14.18), where now $\alpha = u^2$, $\beta = 0$, and $\gamma = 1$:

$$\begin{aligned} u_t \quad &\to \quad \frac{1}{\kappa}\delta_t U_m^n\,; \\ (u^2 u_x)_x \quad &\to \quad \frac{1}{2h}\left(\frac{(U_m^n)^2 + (U_{m+1}^n)^2}{2}\,\frac{\delta_x U_m^n}{h} - \frac{(U_{m-1}^n)^2 + (U_m^n)^2}{2}\,\frac{\delta_x U_{m-1}^n}{h}\right) + \\ &\qquad \frac{1}{2h}\left(\frac{(U_m^{n+1})^2 + (U_{m+1}^{n+1})^2}{2}\,\frac{\delta_x U_m^{n+1}}{h} - \frac{(U_{m-1}^{n+1})^2 + (U_m^{n+1})^2}{2}\,\frac{\delta_x U_{m-1}^{n+1}}{h}\right). \end{aligned} \tag{14.34}$$

Note that the discretization of $(u^2 u_x)_x$ on the r.h.s. of (14.34) is a special case of the r.h.s. of (14.69) in Appendix. Also, the following observation is in order.

Remark 2: Note that the value $u^2(x_m + h/2, t_n)$ was represented differently in (14.26) and in (14.34):

$$\begin{aligned} &\text{in (14.26):} \quad u^2(x_m + h/2, t_n) \to \quad \left(\tfrac{U_{m+1}^n + U_m^n}{2}\right)^2; \\ &\text{in (14.34):} \quad u^2(x_m + h/2, t_n) \to \quad \tfrac{(U_{m+1}^n)^2 + (U_m^n)^2}{2}. \end{aligned} \tag{14.35}$$

This was done intentionally, to illustrate an ambiguity of writing out some nonlinear terms. For now, we will just note this fact, but will comment on it more extensively at the end of this subsection.

The scheme resulting from substitution of the discretized derivatives (14.34) into Eq. (14.25) is a special case of scheme (14.69) in Appendix. You will be asked to write it down in a Bonus homework problem. As also noted in Appendix, this scheme is a nonlinear algebraic system of equations for U_m^{n+1} with $m = 1, \ldots, M-1$ and can be solved by the Newton–Raphson via a substitution (14.70a). We will rewrite that latter formula for the entire vector of unknown

solutions on the grid, $\vec{\mathbf{U}}^{n+1} = [U_1^{n+1}, U_2^{n+1}, \ldots, U_{M-1}^{n+1}]^T$:

$$\vec{\mathbf{U}}^{n+1} = \vec{\mathbf{U}}^n + \vec{\varepsilon}^{(0)}, \qquad \|\vec{\varepsilon}^{(0)}\| \ll \|\vec{\mathbf{U}}^n\|\,. \tag{14.36}$$

Note that the superscript '(0)' indicates that the subsequent calculations will reveal that $\vec{\varepsilon}^{(0)}$ is only the first approximation to the change in the solution from time level n to $(n+1)$; compare this with (8.84). More accurate approximations can be found, as shown below (see (14.42) below and compare with (8.88)).

Upon substituting (14.34) and (14.36) into (14.25) and discarding terms $O((\varepsilon^{(0)})^2)$, one obtains (see the explanation below):

$$\begin{aligned} \varepsilon_m^{(0)} - \frac{\kappa}{2h}\, \Bigg(& \left(\varepsilon_m^{(0)} U_m^n + \varepsilon_{m+1}^{(0)} U_{m+1}^n\right) \frac{\delta_x U_m^n}{h} - \left(\varepsilon_{m-1}^{(0)} U_{m-1}^n + \varepsilon_m^{(0)} U_m^n\right) \frac{\delta_x U_{m-1}^n}{h} + \\ & \frac{(U_m^n)^2 + (U_{m+1}^n)^2}{2}\, \frac{\delta_x \varepsilon_m^{(0)}}{h} - \frac{(U_{m-1}^n)^2 + (U_m^n)^2}{2}\, \frac{\delta_x \varepsilon_{m-1}^{(0)}}{h} \Bigg) \\ = \; & \frac{\kappa}{h} \left(\frac{(U_m^n)^2 + (U_{m+1}^n)^2}{2}\, \frac{\delta_x U_m^n}{h} - \frac{(U_{m-1}^n)^2 + (U_m^n)^2}{2}\, \frac{\delta_x U_{m-1}^n}{h} \right). \end{aligned} \tag{14.37}$$

This is a special case of Eq. (14.71) in Appendix (please look at that equation and recall what its notations mean). Below we provide brief details on how the above expression was obtained; you will be asked to fill in the missing details in the aforementioned Bonus homework problem.

The partial derivative notations in (14.71) are not particularly illuminating since the explicit dependence of f on u, u_x, etc. is not given there. It is, therefore, instructive to work out the terms in (14.37) using the specific form of the r.h.s. of (14.25). Namely, note that the expression inside the outer parentheses can be written as:

$$u^2 u_x \;\equiv\; A\,B, \qquad \text{where} \quad A \equiv u^2, \;\; B \equiv u_x. \tag{14.38}$$

Then, when we replace $u \to (U + \varepsilon^{(0)})$,[45] we do *not* multiply out all terms, but instead use a "mathematically literate" approach, equivalent to the the following form of the familiar Product Rule from Calculus:

$$\Delta(A\,B) \equiv (A + \Delta A)(B + \Delta B) - A\,B \approx A\Delta B + B\Delta A, \qquad \text{(Product Rule)}$$

where $\Delta A \ll A$ and $\Delta B \ll B$ are the small terms engendered by $\varepsilon^{(0)}$. Note that the quadratically small term $\Delta A \Delta B$ has been omitted. To show how one applies this formula, we first compute ΔA:

$$(U + \varepsilon^{(0)})^2 \approx U^2 + 2U\,\varepsilon^{(0)}, \qquad \Rightarrow \qquad \Delta A \equiv 2U\,\varepsilon^{(0)}. \tag{14.39}$$

Note that the approximate equation in (14.39) is nothing but a linearization of the function $\left(U + \varepsilon^{(0)}\right)^2$, as was considered in Sec. 14.4.1. As for B and ΔB, they are simply:

$$\frac{\delta_x U}{h} + \frac{\delta_x \varepsilon^{(0)}}{h} \;\equiv\; B + \Delta B. \tag{14.40}$$

Multiplying out (14.39) and (14.40) and using (Product Rule), one obtains the term in the largest parentheses on the l.h.s. of (14.34). Again, you can verify this in a Bonus homework problem.

[45] We have dropped indices m and n since they are not essential for what we are about to illustrate.

From (14.37), 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 (14.36) 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

$$\vec{\mathbf{U}}^{(1)} \equiv \vec{\mathbf{U}}^n + \vec{\varepsilon}^{(0)} \tag{14.41}$$

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

$$\vec{\mathbf{U}}^{n+1} = \vec{\mathbf{U}}^{(1)} + \vec{\varepsilon}^{(1)}, \qquad \|\vec{\varepsilon}^{(1)}\| \ll \|\vec{\mathbf{U}}^{(1)}\| \,. \tag{14.42}$$

Substituting (14.42) along with (14.34) into (14.25), we obtain an equation similar to (14.37):

$$\begin{aligned} \varepsilon_m^{(1)} - \frac{\kappa}{2h} \, & \left(\left(\varepsilon_m^{(1)} U_m^{(1)} + \varepsilon_{m+1}^{(1)} U_{m+1}^{(1)} \right) \frac{\delta_x U_m^{(1)}}{h} - \left(\varepsilon_{m-1}^{(1)} U_{m-1}^{(1)} + \varepsilon_m^{(1)} U_m^{(1)} \right) \frac{\delta_x U_{m-1}^{(1)}}{h} + \right. \\ & \left. \frac{(U_m^{(1)})^2 + (U_{m+1}^{(1)})^2}{2} \frac{\delta_x \varepsilon_m^{(1)}}{h} - \frac{(U_{m-1}^{(1)})^2 + (U_m^{(1)})^2}{2} \frac{\delta_x \varepsilon_{m-1}^{(1)}}{h} \right) \\ = \; & -\varepsilon_m^{(0)} + \frac{\kappa}{2h} \left(\frac{(U_m^n)^2 + (U_{m+1}^n)^2}{2} \frac{\delta_x U_m^n}{h} - \frac{(U_{m-1}^n)^2 + (U_m^n)^2}{2} \frac{\delta_x U_{m-1}^n}{h} \right) + \\ & \frac{\kappa}{2h} \left(\frac{(U_m^{(1)})^2 + (U_{m+1}^{(1)})^2}{2} \frac{\delta_x U_m^{(1)}}{h} - \frac{(U_{m-1}^{(1)})^2 + (U_m^{(1)})^2}{2} \frac{\delta_x U_{m-1}^{(1)}}{h} \right) . \end{aligned} \tag{14.43}$$

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., (14.37) and (14.43)). 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 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 2+ : This elaborates on the observation made in Remark 2 above. Let us continue using (14.25) as the model problem. Note that it can be written in an equivalent form:

$$u_t = \frac{1}{3} (u^3)_{xx} \,. \tag{14.44}$$

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

$$\frac{1}{\kappa} \delta_t U_m^n \; = \; \frac{1}{3} \cdot \frac{1}{2h^2} \left(\delta_x^2 (U^3)_m^n + \delta_x^2 (U^3)_m^{n+1} \right) \; ; \tag{14.45}$$

recall the definition (13.3) of the operator δ_x^2. The point we want to make here is that the nonlinear system (14.45) is *different* from the nonlinear system obtained upon substitution of (14.34) 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,

$$(u^3)_x = 3u^2 u_x \,. \tag{14.46}$$

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

$$(u^3)_x \;\to\; \frac{(U_{m+1})^3 - (U_{m-1})^3}{2h} = \frac{(U_{m+1} - U_{m-1})(U_{m+1}^2 + U_{m+1}U_{m-1} + U_{m-1}^2)}{2h} \,. \tag{14.47}$$

Using the same — central-difference — formula to discretize the derivative on the r.h.s. of (14.46), one obtains

$$3u^2 \cdot u_x \;\to\; 3U_m^2 \cdot \frac{U_{m+1} - U_{m-1}}{2h}, \tag{14.48}$$

which, obviously, does not equal the r.h.s. of (14.47), 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 (14.46)). However, these different representations, when discretized *using the same rule*, can still lead to distinct finite-difference equations, as illustrated by (14.47) and (14.48). For Hamiltonian equations, this ambiguity can be utilized to construct methods that preserve specified conserved quantities (like the symplectic Euler and Verlet methods in Lecture 5 almost preserved the Hamiltonian). 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 (14.44) as the model problem. With the discretization error of $O(\kappa^2)$,[46] the u^3 term at the virtual time level $t_{n+\frac{1}{2}}$ can be approximated as follows:

$$u^3 \;\to\; \left(U^{n+\frac{1}{2}}\right)^2 \frac{U^n + U^{n+1}}{2}, \tag{14.49}$$

where the fraction on the r.h.s. is explained by (14.15c). The r.h.s. of (14.49) is now linear with respect to U^{n+1}, but the problem is that we do not yet know $U^{n+\frac{1}{2}}$. To achieve the desired accuracy of $O(\kappa^2)$, that term can be approximated by an explicit method that should have that accuracy. A simple, $O(\kappa^2)$-accurate way to compute $U^{n+\frac{1}{2}}$ is by a multi-step method similar to (3.4):

$$U^{n+\frac{1}{2}} = U^n + \frac{\kappa}{2}\left(\frac{\partial U}{\partial t}\right)^n + O(\kappa^2) = U^n + \frac{\kappa}{2}\,\frac{U^n - U^{n-1}}{\kappa} + O(\kappa^2) \approx \frac{3}{2}U^n - \frac{1}{2}U^{n-1}\,, \tag{14.50}$$

[46] Recall from Section 12.2 that the discretization and global errors have the same order in κ for equations of the form (14.33).

where in the last expression we have dropped the $O(\kappa^2)$-term. The resulting scheme:

$$\delta_t U_m^n = \frac{r}{3}\delta_x^2 \left(\left(U_m^{n+\frac12}\right)^2 \, \frac{U_m^n + U_m^{n+1}}{2} \right) \tag{14.51a}$$

is called *semi-implicit*, for the following reason. The factor $\left(U_m^{n+\frac12}\right)^2$ is known at time level n from formula (14.50). Then, denoting this known factor by $\left(U_m^{n+\frac12}\right)^2 \equiv a_m^{n+\frac12}$, one can rewrite (14.51a) as:

$$\delta_t U_m^n = \frac{r}{3}\left[a_m^{n+\frac12} \left(\frac{U_m^n + U_m^{n+1}}{2} \right)_{xx} + 2 \left(a_m^{n+\frac12}\right)_x \left(\frac{U_m^n + U_m^{n+1}}{2} \right)_x + \left(a_m^{n+\frac12}\right)_{xx} \frac{U_m^n + U_m^{n+1}}{2} \right], \tag{14.51b}$$

where the notation, say, $\left(a_m^{n+\frac12}\right)_x$, stands for the $O(h^2)$-accurate discretization of the first spatial derivative of $a_m^{n+\frac12}$, etc. Details of such discretizations are already worked out in (14.17). In deriving (14.51b) from (14.51a), we also used the identity for the second derivative of the product of two functions f and g:

$$(f\,g)'' = f\,g'' + 2f'\,g' + f''\,g.$$

It is important to stress that while scheme (14.51) does have the unknown variables U_m^{n+1} on the r.h.s., as typical of an implicit scheme, it is *not fully implicit*, because a fully implicit scheme would involve a term $(U_m^{n+1})^3$.

A semi-implicit scheme, such as (14.51), has a *strong advantage* over a fully implicit one in that it is *linear* in the unknown variables U_m^{n+1} and hence can be solved as a linear system of equations. On the other hand, since it is not fully implicit, it cannot be unconditionally stable (see Theorem 4.2 at the end of Lecture 4). However, one can *reasonably expect* that ***stability properties of a semi-explicit scheme should be better than those of a fully explicit scheme*** (although they may be worse than those of a fully implicit scheme). In a Bonus homework problem, you will be given the opportunity to explore this matter further. Thus, on balance, a semi-implicit scheme may be a good one to try since: (i) ***it is much easier to implement than the Newton–Raphson method*** and (ii) ***it is expected to have better stability properties than an explicit scheme***.

Remark 3: Method (14.51), (14.50) is a member of a large class of semi-implicit methods. It can be straightforwardly generalized to the following class of equations:

$$u_t = a(u, u_x, x, t)u_{xx} + b(u, u_x, x, t)u_x\,, \tag{14.52}$$

where, as stated above, the coefficients a and b may depend on the solution u and its derivative u_x.[47] An extension of scheme (14.51), (14.50) for (14.52) is:

$$\begin{aligned} \frac{\delta_t U_m^n}{\kappa} \;=\; & a\left(U_m^{n+\frac12}, (U_m^{n+\frac12})_x, x_m, t_{n+\frac12}\right) \frac{(U_m^n)_{xx} + (U_m^{n+1})_{xx}}{2} \;+ \\ & b\left(U_m^{n+\frac12}, (U_m^{n+\frac12})_x, x_m, t_{n+\frac12}\right) \frac{(U_m^n)_x + (U_m^{n+1})_x}{2}\,, \end{aligned} \tag{14.53}$$

where $U_m^{n+\frac12}$ is given by (14.50), $(U_m^n)_{xx}$ denotes the second-order accurate finite-difference approximation of $u_{xx}(x_m, t_n)$ (i.e., $\delta_x^2 U_m^n/h^2$), and similarly for $(U_m^n)_x$.

[47] Further generalizations of this form are possible, but for the purpose of our brief discussion, form (14.52) is sufficient.

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

$$u_t = F\left(u(t),t\right) \equiv F_0(u(t),t) + F_1(u(t),t)\,, \tag{14.54}$$

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 (14.52), 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 (14.54) 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. (14.54) 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 (14.54) is:

$$\frac{U^{n+1}-U^n}{\kappa} = F_0(U^n,t_n) + (1-\theta)F_1(U^n,t_n) + \theta F_1(U^{n+1},t_{n+1})\,, \tag{14.55}$$

where θ is a parameter, as in Lecture 13. Note that since, by design, F_1 depends on U^{n+1} linearly, scheme (14.55) 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

$$u_t = \lambda_0 u + \lambda_1 u\,, \tag{14.56}$$

where λ_0 and λ_1 correspond to F_0 and F_1. Substituting $U^n = \rho^n$ into scheme (14.55) applied to Eq. (14.56), one finds that

$$\rho \equiv \rho(z_0,z_1) = \frac{1+z_0+(1-\theta)z_1}{1-\theta z_1}\,, \tag{14.57}$$

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

$$|\rho(z_0,z_1)| < 1 \tag{14.58}$$

for stability. Inequality (14.58) 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 interpreted in two different ways, depending on what one knows about the family of equations that one wants to solve.

First interpretation of (14.58)

Suppose one has to design a method (14.55) that should be applicable to equations of the form (14.54) 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:

$$\mathcal{D}_0 : \qquad |1 + z_0| < 1\,, \qquad \text{where} \quad z_0 \equiv \lambda_0 \kappa, \tag{14.59}$$

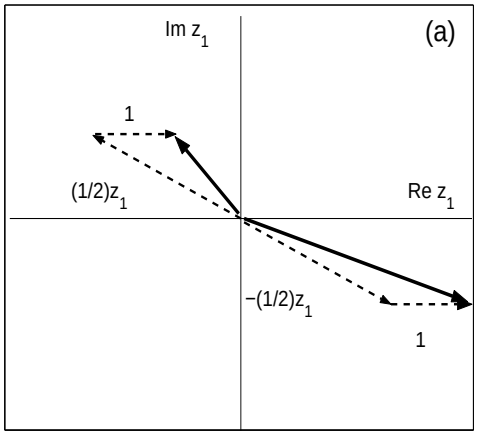
and where λ_0 has been described in the previous sentence. An example of such a situation is when F_0 contains terms describing nonlinear, but non-stiff, reaction or advection, while F_1 contains the simple diffusion term u_{xx}, for which all λ_1's lie on the negative real axis (see Problem 2 in HW 12). It turns out that enforcing both conditions, (14.58) and (14.59) imposes a restriction on the values of z_1, which in the absence of (14.59) (or, equivalently, if $z_0 = 0$ in (14.58)) would not have occurred. Let us now explain *why* this restriction on the values of z_1 occurs.

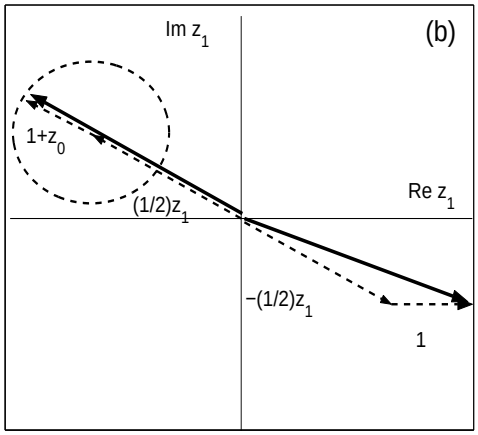
For the sake of argument, consider the value $\theta = 1/2$ in (14.55), which would lead to the Crank–Nicolson scheme if F_0 were absent. Since that scheme (again, still with $F_0 \equiv 0$ for now)

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 6 in HW 4). That is,

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

which holds 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 (14.60) are depicted by the solid-line 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.[48]





[48] Pause to study what this Figure panel shows, and how that relates to the previous sentence. Do not proceed without this, as you will then be lost on the rest of this section.

On the other hand, when $F_0 \neq 0$, the stability condition (14.58) with $\theta = 1/2$ is

$$\frac{\left|(1+z_0)+\frac{1}{2}z_1\right|}{\left|1-\frac{1}{2}z_1\right|} \leq 1, \tag{14.61}$$

which must hold for *all* z_0 such that $|1+z_0| \leq 1$. As illustrated in Figure (b) above, condition (14.61) can be violated for some of such z_0 *unless* $\mathrm{Im}\,(z_1) = 0$.[49] Thus, if one insists on having the full stability region for the explicit part of the IMEX method (14.55), the stability region of this method *with respect to its implicit part* is necessarily *less* than the corresponding stability region of (14.55) with $F_0 \equiv 0$. Specifically, we have just shown that for $\theta = 1/2$, the stability region of z_1 *has shrunk* from being the entire left half of the complex plane (i.e., $\mathrm{Re} z_1 \leq 0$) for $z_0 = 0$ — to being only the negative real axis (i.e., $z_1 \leq 0$) when z_0 is allowed to be anywhere within the region (14.59). (Often, accuracy — as opposed to stability — considerations dictate that $\mathcal{D}_0$ be smaller than in (14.59); say, $|a+z_0| < a$ for some $0 < a < 1$. Then, the stability region for z_1 expands into some sector with vertex at the origin that contains the negative real z_1-axis.)

In general, in this case one can show[50] that the stability condition (14.58) yields the inequality

$$\mathcal{D}_1: \qquad 1 + |(1-\theta)z_1| < |1-\theta z_1|\,, \qquad \text{where} \quad z_1 \equiv \lambda_1 \kappa\,. \tag{14.62}$$

This is the second condition in the set. That is, (14.59) and (14.62) together are equivalent to (14.58).

With some effort, one can further show from (14.62) that the unconditional stability of the IMEX method (14.55) is attained only for $\theta = 1$. For $\theta < 1/2$, scheme (14.55) is unstable. (This should be contrasted with the situation when $F_0 \equiv 0$, for which method (14.55) 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 (14.62), collapses onto the negative real axis: $z_1 < 0$, as mentioned before (14.62). 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, $\alpha \approx 25^\circ$ and $\alpha > 50^\circ$ 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 (14.58)

Alternatively, suppose that (14.54) is a *system of coupled equations* for variables $u^{(1)}$, $u^{(2)}, \ldots$, and suppose that $F_1 \equiv [F_1^{(1)}, F_1^{(2)}, \ldots]$ 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 $\partial(F_1^{(1)}, F_1^{(2)}, \ldots)/\partial(u^{(1)}, u^{(2)}, \ldots)$ (see Sec. 5.4 in Lecture 5) can be found anywhere in the left half of the complex plane, i.e. $\mathrm{Re}\, z_1^{(j)} \leq 0$ for all j. Thus, one may want to know for which complex z_0 one can fulfill condition (14.58) given that z_1 can be allowed anywhere in the left-half complex plane. The next paragraph contains a summary of results from the book by Hundsdorfer and Verwer for this case.

Similarly to the previous case, the corresponding nonempty region $\mathcal{D}_0$ exists only for $\theta \geq 1/2$. That is, if $\theta < 1/2$, then the IMEX method (14.55) where z_1 can be found anywhere in the left-half plane, *is unstable for any* $z_0 \neq 0$ *with* $\mathrm{Re}\,(z_0) \leq 0$! (This should be contrasted with the situation when $F_1 \equiv 0$, for which method (14.55) — i.e., the simple Euler method —

[49] Again, pause, then sketch the vector $z_1/2$ for $z_1 < 0$ (i.e., with $\mathrm{Im}\,(z_1) = 0$ being satisfied) and verify that the previous statement is true.

[50] Please take this on faith.

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 (14.54). 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 (14.55). 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:

$$\begin{aligned} \frac{U^{n+1} - U^n}{\kappa} &= \frac{3}{2}F_0(U^n, t_n) - \frac{1}{2}F_0(U^{n-1}, t_{n-1}) \; + \\ &\theta F_1(U^{n+1}, t_{n+1}) + \Big(\frac{3}{2} - 2\theta\Big) F_1(U^n, t_n) + \Big(\theta - \frac{1}{2}\Big)F_1(U^{n-1}, t_{n-1})\,. \end{aligned} \tag{14.63}$$

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 (14.58)). 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 (similar to that for the modified explicit Euler method; see Lecture 4) such that its 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:

$$\frac{U^{n+1} - U^{n-1}}{2\kappa} = F_0(U^n, t_n) + \frac{1}{2}\Big(F_1(U^{n+1}, t_{n+1}) + F_1(U^{n-1}, t_{n-1})\Big)\,. \tag{14.64}$$

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 (14.52) where a is independent of u can be solved by this method. Another example is the Nonlinear Schrödinger equation (14.65) 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. (14.52) where now the coefficient a *does* depend on u. Then one can replace the implicit integration in (14.55) with an analogue of the semi-implicit method (14.53). 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*:

$$iu_t + u_{xx} + 2|u|^2 u = 0, \qquad (\text{note the } i = \sqrt{-1} \text{ in front of } u_t) \tag{14.65}$$

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:

$$\begin{aligned}
&\text{Given } \; U^n(x) \equiv u(x,t_n), \\
&\text{Solve } \; iu_t + u_{xx} = 0 \;\; \text{from } t_n \text{ to } t_{n+1}; \qquad \Rightarrow \;\; \text{get } \; U^{\rm aux}; \\
&\text{Using } \; U^{\rm aux} \;\; \text{as the initial condition,} \\
&\text{Solve } \; iu_t + 2u|u|^2 = 0 \;\; \text{from } t_n \text{ to } t_{n+1}; \;\; \Rightarrow \;\; \text{get } \; U^{n+1}.
\end{aligned} \tag{14.66}$$

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:

$$u = A\, e^{2iA^2 t}, \qquad \text{where } A \text{ is a real constant,} \tag{14.67}$$

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:

$$u = A\,\mathrm{sech}(Ax)\; e^{iA^2 t}, \tag{14.68}$$

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 (14.68) can be made based on the knowledge of numerical stability or instability of the plane wave solution (14.67).

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 (14.65). 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 Appendix: General form of the Newton–Raphson method for (14.33)

For convenience of the reader, we restate Eq. (14.33) here:

$$u_t = f(u, u_x, \ldots). \tag{(14.33)}$$

Everywhere below in this Appendix, '...' denote possible dependence of f on higher spatial derivatives of u (i.e., u_{xx} etc.). The CN scheme for it is (see Eqs. (14.17) and, earlier, (13.1) in Lecture 13):

$$\frac{\delta_t U_m^n}{\kappa} = \frac{1}{2}\left[f(\tilde{U}_m^n, (\tilde{U}_x)_m^n, \ldots) + f(\tilde{U}_m^{n+1}, (\tilde{U}_x)_m^{n+1}, \ldots)\right]. \tag{14.69}$$

Here the tilde in the notation $\tilde{U}_m^n$ signifies the fact that the value $u(x_m, t_n)$ may need to be evaluated using not only its value at node x_m, but also adjacent nodes, $x_{m\pm1}$; see the very first term on the r.h.s. of (14.19). The notation $(\tilde{U}_x)_m^n$ has the analogous meaning. As mentioned before, the challenge in solving (14.69) is that, for a nonlinear function f, those equations (which have to be solved for all m at ones) constitute a system of *nonlinear* equations.

This challenge can be addressed by noticing that solutions at adjacent time levels should differ by a small amount (proportional to κ), and therefore it is reasonable to write:

$$U_m^{n+1} = U_m^n + \varepsilon_m, \qquad \text{where} \quad |\varepsilon_m| \ll |U_m^n|. \tag{14.70a}$$

It is this ε_m — a small change of the solution from time level n to $(n+1)$, — that we want to determine. Since $\tilde{U}_m^n$ is some combination of the solution at nodes x_m and $x_{m\pm1}$, one can also write

$$\tilde{U}_m^{n+1} = \tilde{U}_m^n + \tilde{\varepsilon}_m, \qquad \text{where} \quad |\tilde{\varepsilon}_m| \ll |\tilde{U}_m^n|. \tag{14.70b}$$

Substituting (14.70b) into the second term on the r.h.s. of (14.69) and using two leading terms in the Taylor series to expand it, one obtains:

$$\begin{gathered} f(\tilde{U}_m^{n+1}, (\tilde{U}_x)_m^{n+1}, \ldots) \;=\; f(\tilde{U}_m^n + \tilde{\varepsilon}_m,\; (\tilde{U}_x)_m^n + (\tilde{\varepsilon}_x)_m,\; \ldots) \;\approx \\ f(\tilde{U}_m^n, (\tilde{U}_x)_m^n, \ldots) \;+\; \tilde{\varepsilon}_m\, f_u(\tilde{U}_m^n, (\tilde{U}_x)_m^n, \ldots) \;+\; (\tilde{\varepsilon}_x)_m\, f_{u_x}(\tilde{U}_m^n, (\tilde{U}_x)_m^n, \ldots) + \ldots; \end{gathered} \tag{14.71}$$

where subscripts of f denote the corresponding partial derivatives. Compare this with (14.29a). To slightly simplify notations, in what follows we will drop the '...'. Substituting (14.71) for the second term on the r.h.s. of (14.69) and moving terms with the unknowns ε_m (and, consequently, also with $\tilde{\varepsilon}_m$) to the l.h.s., we obtain:

$$\frac{\varepsilon_m}{\kappa} - \frac{1}{2}\left(\tilde{\varepsilon}_m\, f_u(\tilde{U}_m^n, (\tilde{U}_x)_m^n) \;+\; (\tilde{\varepsilon}_x)_m\, f_{u_x}(\tilde{U}_m^n, (\tilde{U}_x)_m^n)\right) \;=\; f(\tilde{U}_m^n, (\tilde{U}_x)_m^n). \tag{14.72}$$

To solve this system of equations (for all m), we first note that, unlike system (14.69), system (14.72) is linear. Moreover, since $\tilde{\varepsilon}_m$ is composed of ε_m and ε_{m+1}, it is tridiagonal (just like system (13.7), equivalently written as (13.9), is in Lecture 13). Therefore, it can be solved time-efficiently by the Thomas algorithm. This will, in the first nontrivial approximation, determine the solution at the next time level via (14.70a). If a higher accuracy is desired, one can determine the next-order correction in the same way as (8.87) and (8.88) in Lecture 3. A specific example is worked out in Section 14.4.2.

14.7 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 (14.15c). *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 (14.34) 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. Following the Appendix, explain the idea behind the Newton–Raphson method when applied to nonlinear PDEs.

16. Describe the issue about discretization of nonlinear terms, pointed out in Remarks 2 and 2+.

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

18. Explain why the r.h.s. of (14.50) approximates the l.h.s. of that equation.

19. Obtain (14.57).

20. What are two possible interpretations of (14.58)?

21. Make sure you can follow the argument made around condition (14.61).

22. Why does method (14.63) have the name 'Adams' in it?

23. When can method (14.64) be used?