12 The Heat equation in one spatial dimension: Simple explicit method and Stability analysis

12.1 Formulation of the IBVP and the minimax property of its solution

We begin by writing down the Heat equation (in its simplest form) on the interval $x \in [0,\, 1]$ and the corresponding initial and boundary conditions. In fact, this is just a restatement from the end of Lecture 11.

$$u_t = u_{xx} \qquad 0 < x < 1, \quad t > 0\,; \tag{12.1}$$

$$u(x, t=0) = u_0(x) \qquad 0 \le x \le 1\,; \tag{12.2}$$

$$u(0,t) = g_0(t), \quad u(1,t) = g_1(t) \qquad t \ge 0\,. \tag{12.3}$$

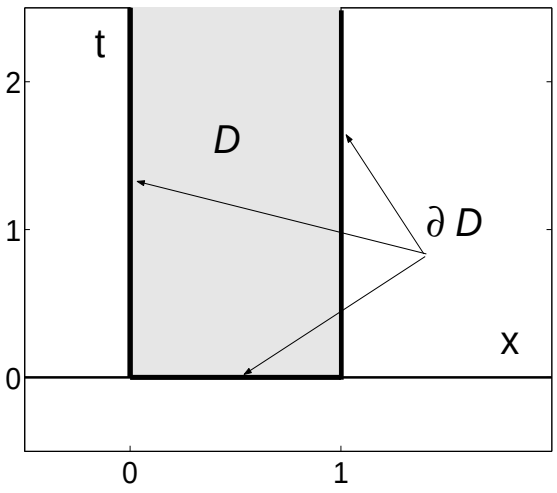
The IBVP (12.1)–(12.3) will be the subject of this and the next lectures. Boundary conditions of a form more general than (12.3) will be considered in Lecture 14. Recall that in order to produce a continuous solution, the boundary and initial conditions must match:

$$u_0(0) = g_0(0) \qquad \text{and} \qquad u_0(1) = g_1(0)\,. \tag{12.4}$$

On physical grounds, in what follows we will always require that the matching conditions (12.4) be satisfied.

It is always useful to know what general properties one may expect of the analytical solution of a given IBVP, so that one could verify that the corresponding numerical solution also has these properties (this is a basic sanity check for the numerical code). Such a property for IBVP (12.1)–(12.3), stated below, is proved in courses on PDEs.

Minimax principle Suppose u_t (and hence u_{xx} and both u_x and u) is continuous in the region $D = [0,\, 1] \times [0,\, \infty)$ (see the figure on the right).[a] Then the solution u of the IBVP (12.1)–(12.3) achieves its maximum and minimum values on ∂D (i.e. either for $t = 0$ or for $x = 0$ or $x = 1$). In other words, u *cannot* achieve its maximum or minimum values *strictly inside* D.



[a] Note that here domain D and its boundary ∂D are defined slightly differently than in the figure at the end of Sec. 11.1.

Note that this, at least partially, agrees with our intuition in "real life". Indeed, suppose one creates some distribution of non-negative temperature in the rod at $t = 0$ while keeping the ends of the rod at zero temperature at all times. Then we expect that the temperature inside the rod at any $t > 0$ will be less than it was at $t = 0$ (because the rod will cool down); that is, the maximum temperature was observed somewhere along the rod at $t = 0$, i.e. at the bottom part of ∂D. On the other hand, we also expect that the temperature in this setup will not drop below zero; that is, the temperature will be minimum at the ends of the rod, i.e. at the sides of ∂D.

12.2 The simplest explicit method for the Heat equation

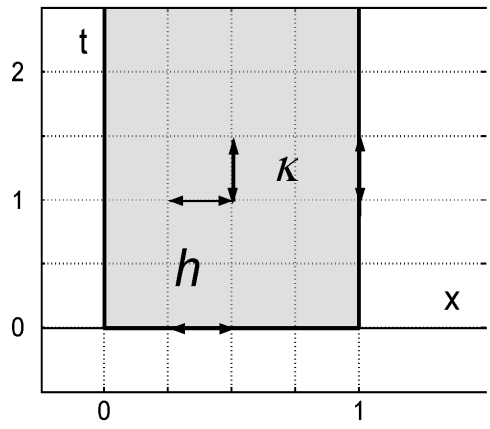
Let us cover the region D with a mesh (or grid), as shown on the right. Denote

$$\begin{aligned} x_m &= mh, \quad m = 0, 1, \ldots, M, \quad (h = \tfrac{1}{M}); \\ t_n &= n\kappa, \quad n = 0, 1, \ldots, N, \quad \left(\kappa = \tfrac{T_{\max}}{N}\right); \end{aligned} \tag{12.5}$$

here $T_{\max}$ is the maximum time until we want to compute the solution. Also, let U_m^n be the solution computed at node (x_m, t_n). For simplicity, in this lecture we will assume that the boundary conditions are homogeneous:

$$g_0(t) = g_1(t) = 0 \qquad \text{for all } \ t\,; \tag{12.6}$$

note that this implies that $u_0(0) = u_0(1) = 0$.



When restricted to the grid, the initial and boundary conditions become:

$$(12.2) \ \Rightarrow \ U_m^0 = u_0(mh), \quad 0 \le m \le M\,; \tag{12.7}$$

$$(12.3) \ \Rightarrow \ \begin{cases} U_0^n = 0, \\ U_M^n = 0, \end{cases} \qquad n \ge 0\,. \tag{12.8}$$

Let us now use the simplest finite-difference approximations to replace the derivatives in the Heat equation:

$$u_t \ \to \ \frac{U_m^{n+1} - U_m^n}{\kappa} + O(\kappa)\,, \tag{12.9}$$

$$u_{xx} \ \to \ \frac{U_{m+1}^n - 2U_m^n + U_{m-1}^n}{h^2} + O(h^2)\,. \tag{12.10}$$

Substituting these formulae into (12.1) yields the simplest explicit method for solving the Heat equation:

$$\frac{U_m^{n+1} - U_m^n}{\kappa} = \frac{U_{m+1}^n - 2U_m^n + U_{m-1}^n}{h^2} + O(\kappa + h^2)\,, \tag{12.11}$$

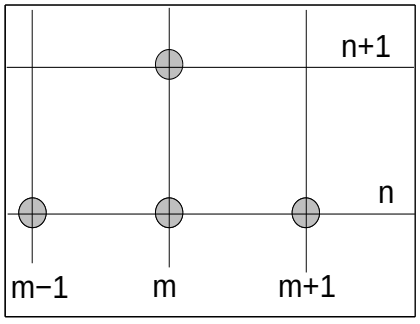
or, equivalently,

$$U_m^{n+1} = rU_{m+1}^n + (1 - 2r)U_m^n + rU_{m-1}^n\,, \tag{12.12}$$

where

$$r = \frac{\kappa}{h^2}\,. \tag{12.13}$$

The numerical solution at node (x_m, t_{n+1}) can thus be found if one knows the solution at nodes (x_m, t_n) and $(x_{m\pm1}, t_n)$. These four nodes form a *stencil* for scheme (12.12), as shown schematically on the right.
Given the initial and boundary conditions (12.7) and (12.8), one can advance the solution U_m^n from time level number n to time level number $(n + 1)$ using scheme (12.12).



Let us point out that the error derived in (12.9)–(12.11) is the *discretization error* of the simple explicit scheme (12.12). It is worth displaying this fact conspicuously:

$$\text{discretization error of (12.12)} = O(\kappa^2 + h^2). \tag{12.14}$$

You may recall that discretization error is one of the three types of error that was defined in Lecture 1 and revisited in Lecture 4. The discretization error is the one caused by the replacement of derivatives by finite differences in the equation. For differential equations that are first-order in the evolution variable (x — for ODEs in Lectures 1 and 4; t — for the Heat equation), it has the same order (for ODEs — in h; for PDEs — in κ and h) as the global error, but, unlike the latter, can be explicitly found. (Please review that material in Lectures 1 and 4 if the above brief review seems insufficient.) Thus, the ***global error of scheme*** (12.12) ***is also given by*** (12.14).

From Eq. (12.11) (or, equivalently, from (12.14)) one can see that the simple explicit method is *consistent* with the PDE (12.1).[37] However, from Lecture 4, we know that consistency alone is not sufficient for the numerical solution to *converge* to the analytical solution of an evolution-type differential equation (i.e., for any of the ODEs considered in Lecture 1 – 5 and for the Heat equation in this Lecture). To assure the convergence, we must also require that the finite-difference scheme be *stable*. We therefore turn to studying stability of scheme (12.12) next.

12.3 Stability analysis

Recall that stability means that small errors made during one step of the computation must not grow at subsequent steps. For ODEs, we stated a theorem that said that "stability + consistency" implied convergence of the numerical solution to the analytical one. For PDEs, a similar result also holds:

Lax Equivalence Theorem, 12.1 For a properly posed (as discussed in Lecture 11) IBVP and for a finite-difference scheme that is consistent with the IBVP, stability is a necessary and sufficient condition for convergence.

As for ODEs, this theorem can be understood from the following simple consideration. Let $u_m^n = u(x_m, t_n)$ be the exact solution of the PDE, $\bar{U}_m^n$ be the exact solution of the finite-difference scheme, and U_m^n be the actually computed solution of that scheme. (It may differ from the exact one because, e.g., of round-off errors.) Then

$$|u_m^n - U_m^n| = \left|\left(u_m^n - \bar{U}_m^n\right) + \left(\bar{U}_m^n - U_m^n\right)\right| \le \left|u_m^n - \bar{U}_m^n\right| + \left|\bar{U}_m^n - U_m^n\right| . \tag{12.15}$$

If the difference scheme is consistent, then the first term on the r.h.s. is small. If the difference scheme is stable, then the second term on the r.h.s. is small for all n (i.e., it does not grow). Thus, if the scheme is both consistent and stable, then the l.h.s. of (12.15) is small for all n, which, in words, means that the numerical solution of the finite-difference scheme closely approximates the analytical solution of the PDE.

Now we will show *how* stability of a finite-difference scheme for a PDE can be studied. We will do this using two alternative methods. Method 1 will show a relation between the stability analysis for PDEs with that for systems of ODEs. Method 2 will be new. It is specific to

[37] Recall from Lecture 4 that consistency means that the solution of the finite-difference scheme approaches the solution of the differential equation as the step size(s), κ and h in this case, tend to zero. In other words, the discretization error τ satisfies $\lim_{\kappa,h\to 0} \tau = 0$.

PDEs and, quite pleasantly, is easier to apply than Method 1. However, nothing is free: this simplicity comes at the price that this method gives less complete information than Method 1. We will provide more details after we will have described both methods.

Method 1 (Matrix stability analysis)
One can view scheme (12.11) (and hence (12.12)) as the simple explicit Euler method applied to the following coupled system of ODEs:

$$\frac{d}{dt}\begin{pmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ u_{M-1} \end{pmatrix} = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & 0 & \cdot & \cdot & 0 \\ 1 & -2 & 1 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & 1 & -2 & 1 \\ 0 & \cdot & \cdot & 0 & 1 & -2 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ u_{M-1} \end{pmatrix}. \tag{12.16}$$

(In writing out (12.16), we have also used the homogeneous boundary conditions (12.8).) Indeed, Eqs. (12.11) are obtained by discretizing the time derivative in (12.16) according to (12.9). Thus, studying the stability of scheme (12.11) is equivalent to studying the stability of the simple Euler method for system (12.16). You will be asked to do so, using techniques of Lecture 5, in one of the homework problems. Below we will proceed in a slightly different, although, of course, equivalent, way.

We write Eqs. (12.12) in the matrix form:

$$\begin{pmatrix} U_1^{n+1} \\ U_2^{n+1} \\ \cdot \\ \cdot \\ U_{M-1}^{n+1} \end{pmatrix} = \begin{pmatrix} 1-2r & r & 0 & \cdot & \cdot & 0 \\ r & 1-2r & r & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & r & 1-2r & r \\ 0 & \cdot & \cdot & 0 & r & 1-2r \end{pmatrix}\begin{pmatrix} U_1^{n} \\ U_2^{n} \\ \cdot \\ \cdot \\ U_{M-1}^{n} \end{pmatrix}, \tag{12.17}$$

or

$$\vec{\mathbf{U}}^{n+1} = A\vec{\mathbf{U}}^{n}, \tag{12.18}$$

where r is defined by (12.13),

$$\vec{\mathbf{U}}^{n} = \left[U_1^n, U_2^n, \cdots, U_{M-1}^n\right]^T,$$

and A is the matrix on the r.h.s. of (12.17).

Next, to study stability of this finite-difference scheme, we follow the logic of Section 4.2 of Lecture 4. We assume that when computing the solution of (12.18), one makes a small error $\vec{\epsilon}^{\mathbf{n}}$ (defined similarly to $\vec{\mathbf{U}}^n$ above) on top of the exact solution $\vec{\mathbf{U}}^{\mathbf{n}}$.[38] Thus, one has:

$$\vec{\mathbf{U}}^{n} = \vec{\mathbf{U}}^{\mathbf{n}} + \vec{\epsilon}^{\mathbf{n}}. \tag{12.19}$$

Substituting this into (12.18) and using the fact that *both* $\vec{\mathbf{U}}^n$ and $\vec{\mathbf{U}}^{\mathbf{n}}$ satisfy that equation, one concludes that the error also satisfies the same equation, i.e.:

$$\vec{\epsilon}^{\mathbf{n}+1} = A\,\vec{\epsilon}^{\mathbf{n}}. \tag{12.20}$$

Let us stress that the fact that **the exact solution and the error** of the (discretized) Heat equation **satisfy the same equation**, (12.18) and (12.20), respectively, **holds for any linear equation**. This, of course, could also be deduced from Section 4.2 of Lecture 4, where one would set $f(x,y) = a(x)y + b(x)$ for linear equations; see (4.16).

[38] The bar-notation here mimics that in (12.15).

Stability analysis of scheme (12.12) (or, equivalently, (12.18)) amounts to determining when the norm of the error grows or does not grow with the number of time steps in (12.20). As we showed in the Appendix, this norm does not grow only if all the eigenvalues of A do not exceed 1 in magnitude. Therefore, for the stability analysis, we need to know bounds for the eigenvalues of matrix A. In fact, for the matrix of the very special form appearing in (12.17), exact eigenvalues are well known. We present the following result without a proof (which can be found, e.g., in D. Kincaid and W. Cheney, Numerical Analysis: Mathematics of Scientific Computing, 3rd Ed. (Brooks/Cole, 2002); Sec. 9.1; or from the notes posted alongside this Lecture).

Lemma For an arbitrary N, let B be an $N \times N$ tridiagonal matrix of the form

$$B = \begin{pmatrix} b & c & 0 & & \cdot\cdot 0 & \\ a & b & c & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & a & b & c \\ 0 & \cdot & \cdot & 0 & a & b \end{pmatrix}. \tag{12.21}$$

The eigenvalues and the corresponding eigenvectors of B are:

$$\lambda_j = b + 2\sqrt{ac}\cos\frac{\pi j}{N+1}, \qquad \vec{\mathbf{v}}_j = \begin{pmatrix} \left(\frac{a}{c}\right)^{1/2} \sin\frac{1\cdot\pi j}{N+1} \\ \left(\frac{a}{c}\right)^{2/2} \sin\frac{2\cdot\pi j}{N+1} \\ \cdot \\ \cdot \\ \left(\frac{a}{c}\right)^{N/2} \sin\frac{N\cdot\pi j}{N+1} \end{pmatrix}, \qquad j = 1, \ldots, N\,. \tag{12.22}$$

Using this Lemma, we immediately deduce that the eigenvalues of matrix A in (12.18) are

$$\lambda_j \;=\; 1 - 2r + 2r\,\cos\frac{\pi j}{M}\,, \qquad j = 1, \ldots, M-1\,, \tag{12.23}$$

whence

$$\lambda_{\min} = \quad \lambda_{M-1} = \quad 1 - 2r + 2r\,\cos\frac{\pi(M-1)}{M}\,, \tag{12.24}$$

$$\lambda_{\max} = \quad \lambda_1 = \quad 1 - 2r + 2r\,\cos\frac{\pi}{M}\,. \tag{12.25}$$

If $\pi/M \ll 1$ (i.e., if there are sufficiently many grid points on the interval $[0,\,1]$), the preceding expressions reduce to

$$\lambda_{\min} \;\approx\; 1 - 4r + r\left(\frac{\pi}{M}\right)^2, \tag{12.26}$$

$$\lambda_{\max} \;\approx\; 1 - r\left(\frac{\pi}{M}\right)^2, \tag{12.27}$$

where we have used the expansion $\cos\alpha \approx 1 - \frac{1}{2}\alpha^2$ for $\alpha \ll 1$. Then the condition for convergence of the iterations (12.17), which is, as we said before the Lemma,

$$-1 \le \lambda_j \le 1, \qquad j = 1, \ldots, M-1, \tag{12.28}$$

yields

$$\lambda_{\min} \approx 1 - 4r + r\left(\frac{\pi}{M}\right)^2 \geq -1\,;$$
$$\lambda_{\max} \approx 1 - r\left(\frac{\pi}{M}\right)^2 \leq 1\,.$$

The second of these equations is satisfied automatically because $r = \kappa/h^2 > 0$. The first equation yields:

$$r \leq \frac{2}{4 - \left(\frac{\pi}{M}\right)^2} \equiv \frac{2}{4 - (\pi h)^2} \approx \frac{1}{2}\,. \tag{12.29}$$

This condition, in a simplified form

$$r \leq \frac{1}{2}\,, \qquad \text{or} \qquad \kappa \leq \frac{1}{2}h^2\,, \tag{12.30}$$

is usually taken as the **stability condition of the finite-difference scheme (12.12)**. This means that if $\kappa \leq \frac{1}{2}h^2$, then all round-off errors will eventually decay, and the scheme is stable. The corresponding numerical solution will converge to the solution of IBVP (12.1)–(12.3). If, on the other hand, $\kappa > \frac{1}{2}h^2$, then the errors will grow, thereby making the scheme unstable. The corresponding numerical solution, starting at some $t > 0$, will have nothing in common with the exact solution of the IBVP.

Remark 1 Above we said that for stability of iterations (12.18), the eigenvalues of A must be less than 1 in magnitude. Let us stress that this is true only for diagonalizable (e.g., symmetric) matrices. For nondiagonalizable matrices, e.g., for an $(M-1)\times(M-1)$ matrix

$$\mathcal{N} = \begin{pmatrix} 1 & -1 & 0 & & \cdot\cdot & 0 \\ 0 & 1 & -1 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & 0 & 1 & -1 \\ 0 & \cdot & \cdot & 0 & 0 & 1 \end{pmatrix}, \tag{12.31}$$

an eigenvalue-based stability analysis will fail. Indeed, all of $\mathcal{N}$'s eigenvalues equal 1, yet one can show (e.g., using Matlab's command `norm`) that $\|\mathcal{N}^n\| \to \infty$ as n^{M-1}. There is an entire field of matrix analysis that deals with such non-diagonalizable matrices (with the descriptive keyword being "pseudospectra"), but we will not go into its details here.

Condition (12.30) highlights the **main drawback** of the simple explicit scheme (12.12). Namely, in order for this scheme to be stable (and hence converge to the analytical solution of the IBVP), one must take *very small* steps in time, $\kappa \leq \frac{1}{2}h^2$. This will make the code very time-consuming. We will consider alternative approaches, which do not face that problem, in the next lecture.

Now we turn to the second method for stability analysis, announced earlier in this section.

Method 2 (von Neumann stability analysis)

It is rare that eigenvalues of a matrix, like those of matrix A in (12.18), are available. Therefore, we would like to be able to deduce stability of a scheme without finding those eigenvalues. To begin, let us recall that, since the Heat equation and its discrete version (12.12) are linear, the computational errors satisfy the same equations as the solution itself; see (12.20). Let us denote the error at node $(mh, n\kappa)$ as ϵ_m^n. According to the above, it satisfies Eq. (12.12):

$$\epsilon_m^{n+1} = r\epsilon_{m+1}^n + (1-2r)\epsilon_m^n + r\epsilon_{m-1}^n\,. \tag{12.32}$$

At each time level, the error can be expanded as a linear superposition of Fourier harmonics:

$$\epsilon_m^n = \sum_l c_l(n)\, \exp(i\beta_l x_m) \qquad (\text{here } i \equiv \sqrt{-1}). \tag{12.33}$$

The range of values for β_l will be specified as we proceed.

Since Eq. (12.32) is linear, we can substitute in it each individual term of the above expansion. In doing so, we will also let

$$c_l(n) = \rho^n,$$

where ρ is the number to be determined. Thus, substituting $\epsilon_m^n = \rho^n \exp(i\beta m h)$ into (12.32), one obtains

$$\rho^{n+1} e^{i\beta m h} = r\rho^n e^{i\beta(m+1)h} + (1-2r)\rho^n e^{i\beta m h} + r\rho^n e^{i\beta(m-1)h}\,. \tag{12.34}$$

Let us make two remarks about the notations in (12.34). First, the superscript in ϵ_m^n means that the error ϵ is evaluated at the nth time level. On the other hand, the superscript in ρ^n means that the factor ρ is raised to nth power. Second, we have dropped the subscript l of β since we now deal with only one term in expansion (12.33).

Continuing with our derivation, we divide all terms in (12.34) by $\rho^n \exp(i\beta m h)$ and obtain:

$$\rho \;=\; re^{i\beta h} + (1-2r) + re^{-i\beta h} \;=\; 1 - 2r + 2r\cos(\beta h)\,. \tag{12.35}$$

Condition $|\rho| \le 1$, which would guarantee that the errors do not grow, yields:

$$-1 \;\le\; 1 - 2r + 2r\cos(\beta h) \;\le\; 1\,. \tag{12.36}$$

To obtain a condition on r from this double inequality, we need to know what values the parameter β can take. Even though periodic boundary conditions, which are tacitly implied by the use of the Fourier expansion (12.33) (as shown in graduate courses on Fourier analysis), yield certain discrete values for β, we will follow an alternative — and simplified — approach. Namely, we will assume that the cosine in (12.36) can take its full range of values:

$$-1 \le \cos(\beta h) \le 1 \qquad \Rightarrow \qquad 0 \le \beta h \le \pi\,. \tag{12.37}$$

Using now the half-angle formula, valid for any α:

$$1 - \cos\alpha \;=\; 2\sin^2\left(\frac{\alpha}{2}\right),$$

one rewrites (12.36) as

$$-1 \;\le\; 1 - 4r\,\sin^2\left(\frac{\beta h}{2}\right) \;\le\; 1. \tag{12.38}$$

The right-hand inequality in (12.38) holds automatically, while the left-hand one implies:

$$r\,\sin^2\left(\frac{\beta h}{2}\right) \;\le\; 1/2. \tag{12.39}$$

To guarantee stability of the method, this inequality must hold for all values of βh from (12.37). In particular, it must hold for the "worst"-case value that yields the largest value of $\sin^2(\beta h/2)$. The latter value is 1, occurring for $\beta h = \pi$. Then, the stability condition is

$$r \cdot 1 \le \frac{1}{2}\,, \tag{12.30}$$

which is the simplified form of the stability condition obtained in Method 1 above.

A few remarks are now in order.

Remark 2 The reason why the condition obtained by the von Neumann analysis is slightly different from the *exact* condition (12.29) is that the latter, based on the eigenvalues of matrix A in (12.17), takes into account the boundary conditions (QSA: how?), while the von Neumann analysis, based on expansion (12.33), ignores those conditions.

Remark 3, related to Remark 2. A condition on r obtained via the von Neumann analysis is, in many cases, a necessary, but not sufficient, condition for stability of a finite-difference scheme. That is, a scheme may be found to be stable according to the von Neumann analysis, *but taking into account the information about the boundary conditions* may reveal that there still is an instability. The latter can come from a mode that "hinges" on the boundary but decays towards the inside of the x-domain. An example of such a boundary mode that is unstable analytically (i.e., according to the IBVP for the PDE), will be given in HW 14. An example of a boundary mode that is unstable due to the numerical scheme used, can be found in R.D. Richtmyer and K.W. Morton, Difference methods for initial-value problems, 2nd Ed. (Interscience/John Wiley, New York, 1967); pp. 154–156. A simple, yet practical approach to extend the results of the von Neumann analysis to an IBVP with non-periodic boundary conditions is to follow these two steps: (i) Apply the von Neumann analysis to a given scheme, find the condition (usually on r) that is required for the scheme to be stable, and then test the scheme on the problem of interest while monitoring if any modes localized near the boundaries tend to become unstable.

Let us note that in some — admittedly, much rarer — cases (specifically, in one case known to this instructor), the von Neumann analysis may do the opposite of what is said in the previous paragraph. Namely, a scheme can be found to be von Neumann-unstable, but an analysis taking into account non-periodic boundary conditions can find the instability to be suppressed. A paper (by this instructor) illustrating this situation for an evolution equation of a hyperbolic type (i.e., one describing wave propagation), is posted next to this Lecture.

Let us also remind the reader that Method 1 provides a sufficient condition for stability of the numerical scheme,[39] because it takes into account the boundary conditions when setting up matrix A. However, that method is difficult to apply in practice since it requires the knowledge of the eigenvalues of A.

Remark 4 Note, however, that in finite-difference discretization of hyperbolic equations, where the counterpart of matrix A may turn out to be nondiagonalizable, the von Neumann analysis would provide more information about the stability of the numerical scheme than Method 1. An extreme example is that of matrix $\mathcal{N}$ in (12.31), for which the information about its eigenvalues is useless for the stability analysis (see above). Yet, the von Neumann analysis in this case can be shown to correctly predict stability or instability of the numerical scheme.

An important feature of the von Neumann analysis is that it tells the user *which harmonics* (or modes) of the numerical solution will first become unstable if the stability condition is slightly violated. For example, it follows from (12.36) and (12.39) that if r just slightly exceeds the critical value of $1/2$, then modes with $\beta \approx \pi/h$ will have the amplification factor ρ that will be slightly less than -1:

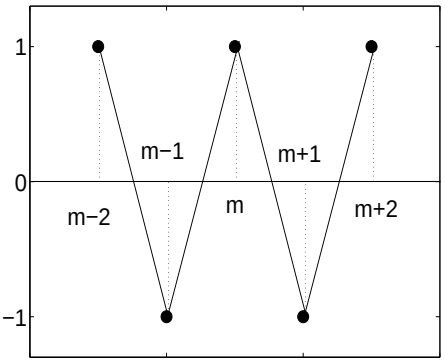
$$r > \frac{1}{2} \qquad \Rightarrow \qquad \rho\left(\beta \approx \frac{\pi}{h}\right) < -1\,. \tag{12.40}$$

[39] We refer to the case of the Heat equation, where matrix A is diagonalizable and hence has a basis of eigenvectors over which any initial condition $\vec{\mathbf{U}}^0$ can be expanded.

Now recall that the modes are proportional to $\exp(i\beta mh)$, hence the unstable modes mentioned above are

$$\exp(i\beta mh) \;=\; \exp\left(i\frac{\pi}{h}\cdot mh\right) \;=\; \exp(i\pi m)\,. \quad (12.41)$$

Therefore, with the account of $e^{i\pi} = -1$, the mode changes its sign from one node to the next, as shown on the right. In other words, it is *modes with the highest frequency* that can cause numerical instability of the simple explicit method for the Heat equation.



12.4 Explicit methods of higher order

As it follows from (12.11), scheme (12.12) has the first order of consistency in t and the second order of consistency in x (i.e., the global error is $O(\kappa + h^2)$). Note, however, that since the stability condition (12.30),

$$\kappa \le \frac{1}{2}h^2\,, \quad (12.30)$$

must hold, then one always has $O(\kappa) = O(h^2)$ for a stable scheme. In other words, it would *not* make sense to derive a method with the discretization (or, equivalently, global — see the end of Section 12.2) error of $O(\kappa^2 + h^2)$ *while keeping* $\kappa \le \frac{1}{2}h^2$. However, it will still be of value to derive a method with the discretization error $O(\kappa^2 + h^4)$, which we will now do.

Remembering how we derived higher-order methods for ODEs, we start off by writing out the Taylor expansions for the finite differences appearing in (12.9) and (12.10):

$$\frac{U_m^{n+1} - U_m^n}{\kappa} \;=\; \frac{\partial}{\partial t}U_m^n + \frac{\kappa}{2}\frac{\partial^2}{\partial t^2}U_m^n + O(\kappa^2)\,, \quad (12.42)$$

$$\frac{U_{m+1}^n - 2U_m^n + U_{m-1}^n}{h^2} \;=\; \frac{\partial^2}{\partial x^2}U_m^n + \frac{h^2}{12}\frac{\partial^4}{\partial x^4}U_m^n + O(h^4)\,. \quad (12.43)$$

Equation (12.42) is the counterpart of Eq. (8.35), and Eq. (12.43) was obtained in Problem 3 of HW 5. Substituting (12.42) and (12.43) into the Heat equation (12.1), we obtain:

$$\frac{U_m^{n+1} - U_m^n}{\kappa} - \frac{U_{m+1}^n - 2U_m^n + U_{m-1}^n}{h^2} = \left(\frac{\partial}{\partial t}U_m^n - \frac{\partial^2}{\partial x^2}U_m^n\right) + \left(\frac{\kappa}{2}\frac{\partial^2}{\partial t^2}U_m^n - \frac{h^2}{12}\frac{\partial^4}{\partial x^4}U_m^n\right) + O(\kappa^2 + h^4)\,. \quad (12.44)$$

If the term in second parentheses on the r.h.s. does not vanish, then this equation merely repeats the previously stated fact, (12.14), that the simple explicit scheme has the accuracy $O(\kappa^2 + h^2)$. Therefore, we need to find condition(s) when this second-parentheses term vanishes; then the discretization error will be given by the last term.

To find such a condition, we note that the first term on the r.h.s. of (12.44) vanishes, because U_m^n is assumed to satisfy the Heat equation. Next, we will handle the second term by relating $(U_m^n)_{tt}$ with $(U_m^n)_{xxxx}$. To that end, differentiating both sides of the Heat equation with respect to t and then using the Heat equation again, we obtain:

$$\frac{\partial}{\partial t}(u_t - u_{xx}) \;=\; u_{tt} - \frac{\partial^2}{\partial x^2}u_t \;=\; u_{tt} - u_{xx\,xx}\,, \qquad \Rightarrow \quad u_{tt} = u_{xxxx}\,. \quad (12.45)$$

Note that in the middle part of the first equation above, we have used that $u_{xxt} = u_{txx}$, which implies that the solution has to be differentiable sufficiently many times with respect x and t.[40]

Continuing with the derivation of a higher-order scheme, we use (12.45) to write the second term on the r.h.s. of (12.44) as

$$\left(\frac{\kappa}{2}u_{tt} - \frac{h^2}{12}u_{xxxx}\right) = \left(\frac{\kappa}{2} - \frac{h^2}{12}\right)u_{xxxx}\,. \tag{12.46}$$

Thus, if one chooses

$$\kappa = \frac{1}{6}h^2, \qquad \text{or} \qquad r = \frac{1}{6}\,, \tag{12.47}$$

then the term (12.46) vanishes identically. Then the r.h.s. of (12.44) becomes $O(\kappa^2 + h^4) = O(h^4)$ (or $O(\kappa^2)$), since κ and h^2 are related by (12.47). Thus, scheme (12.12) with $r = 1/6$ has the error $O(\kappa^2) = O(h^4)$; it is sometimes called the Douglas method.

12.5 Effect of smoothness of initial condition (12.2) on accuracy of scheme (12.12)

As has been noted after Eq. (12.45), the order of the truncation error of the numerical scheme depends on the smoothness of the solution, which, in its turn, is determined by the smoothness of the initial and boundary data. Below we give a corresponding result, whose proof may be found in Sec. 1.7 of the book by Richtmyer and Morton, mentioned a couple of pages back.

Consider the IBVP (12.1)–(12.3) with constant boundary conditions ($g_0(t) = \text{const}$ and $g_1(t) = \text{const}$) which satisfy the consistency conditions (12.4). Let the initial condition $u_0(x)$ have $(p-1)$ continuous derivatives, while its pth derivative is discontinuous but bounded. Then for scheme (12.12) with $r \le 1/2$ and $r \ne 1/6$, there hold the following *conservative* estimates for the error of the numerical solution:

$$||\epsilon^n|| = \begin{cases} O(\kappa^{p/4}) &= O(h^{p/2}), & \text{for } 1 \le p \le 3; \\ O(\kappa|\ln\kappa|) &= O(h^2|\ln h|), & \text{for } p = 4; \\ O(\kappa) &= O(h^2), & \text{for } p > 4\,. \end{cases} \tag{12.48}$$

For the Douglas method (i.e. scheme (12.12) with $r = 1/6$), the analogous error estimates are:

$$||\epsilon^n|| = \begin{cases} O(\kappa^{p/3}) &= O(h^{2p/3}), & \text{for } 1 \le p \le 5; \\ O(\kappa^2\ln\kappa) &= O(h^4|\ln h|), & \text{for } p = 6; \\ O(\kappa^2) &= O(h^4), & \text{for } p > 6\,. \end{cases} \tag{12.49}$$

An intuitive understanding of why an insufficiently smooth initial condition causes the accuracy of the numerical solution to decrease comes from the expressions for the discretization error stated in Section 12.4. Namely, the r.h.s. of Eq. (12.43) shows that in order for the spatial discretization error to be $O(h^2)$, it suffices that u_{xxxx} be continuous. More generally, **all accuracy estimates obtained above (and also in Lecture 13) hold only under the assumption that the corresponding Taylor expansions are valid**. It is this requirement for which sufficient smoothness of the solution (and hence of the initial condition) is needed.

[40] We will state some results of the effect of smoothness of the solution on the order of the error in the next section.

Let us emphasize that estimates (12.48) and (12.49) are very conservative and, according to Richtmyer and Morton, more precise estimates can be obtained, which would show that the error tends to zero with κ and h faster than predicted by (12.48) and (12.49). These estimates, however, do show two important trends, namely:
(i) If the initial condition is not sufficiently smooth, the numerical error will tend to zero slower than for a smooth initial condition. In other words, the "full potential" of a scheme in regards to its accuracy can be utilized only for sufficiently smooth initial data; see the last lines in (12.48) and (12.49).
(ii) The higher the (formally derived) order of the truncation error, the smoother the initial condition needs to be for the numerical solution to actually achieve that order of accuracy.

Finally, let us mention that there is *one more* important trend in regards to the accuracy of numerical schemes, which estimates (12.48) and (12.49) do *not* illustrate. Namely, the accuracy of a scheme depends also on how close the parameter r is to the stability threshold (which is $1/2$ for scheme (12.12)). Intuitively, the reason for this dependency can be understood as follows. Note that when r is at the stability threshold, there is a mode that does not decay, because for it, the amplification factor satisfies: $|\rho| = 1$ (ρ was introduced before Eq. (12.34)). According to the end of Sec. 12.3, such a mode for scheme (12.12) is the highest-frequency mode with $\beta = \pi/h = \pi M$. It is intuitively clear that any discontinuous or jagged initial condition will contain such a mode and modes with similar values of β (i.e. $\beta = \pi(M-1)$, $\pi(M-2)$, etc.). For those modes, $|\rho|$ will be just slightly less than 1, and hence they will decay very slowly, thereby lowering the accuracy of the scheme. On the contrary, when r is, say, 0.4, i.e. less than the threshold by a finite amount, then *all* modes will decay at a finite rate, and the accuracy of the scheme is expected to be higher than for $r = 0.5$. In a homework problem, you will be asked to use a model initial datum to explore the effect of its smoothness, as well as the effect of the proximity of r to the stability threshold, on the accuracy of scheme (12.12).

12.6 Appendix: Role of eigenvalues in stability analysis

Discretization schemes that we encounter for PDEs in this and subsequent lectures have the following matrix form:

$$\vec{\mathbf{U}}^{n+1} = \mathcal{M}\vec{\mathbf{U}}^n, \tag{12.50}$$

where $\vec{\mathbf{U}}^n$ is defined after Eq. (12.18) and $\mathcal{M}$ is some (square) matrix; e.g., in (12.18) $\mathcal{M} \equiv A$. The **question** that we will want to answer is whether the "size" of vector $\vec{\mathbf{U}}^n$ grows or decays with time step n. 'Size' above is a layman term for the norm of a vector. This question can be answered in terms of the norm of matrix $\mathcal{M}$, similarly to our considerations of Method 1 (Picard iterations) in Sec. 8.6 of Lecture 8. However, for a large class of matrices (see below), this question can be answered in terms of eigenvalues of $\mathcal{M}$.

In all situations that we will encounter in this course this matrix will be symmetric (or, perhaps, differ from such only in a small number of its entries). From your undergraduate Linear Algebra you know that symmetric matrices are always diagonalizable. This means not only that the first formula in Sec. 5.4.1 of Lecture 5 applies to them, but also that they have *as many* linearly independent eigenvectors as their dimension. That is:

$$\mathcal{M}\vec{\mathbf{m}}_{\mathbf{j}} = \lambda_j \vec{\mathbf{m}}_j, \qquad j = 1, 2, \ldots, M-1. \tag{12.51}$$

In other words, these eigenvectors for a *basis* in the same space $\mathbb{R}^{M-1}$ where $\vec{\mathbf{U}}^n$ "lives". Then, any initial condition $\vec{\mathbf{U}}^0$ can be expanded over this basis:

$$\vec{\mathbf{U}}^0 = c_1^{(0)}\vec{\mathbf{m}}_1 + c_2^{(0)}\vec{\mathbf{m}}_2 + \cdots + c_{M-1}^{(0)}\vec{\mathbf{m}}_{M-1} \tag{12.52}$$

for some $c_j^{(0)}$. Substituting (12.52) into (12.50) with $n = 0$ and using (12.51), one obtains:

$$\vec{\mathbf{U}}^1 = \lambda_1 c_1^{(0)} \vec{\mathbf{m}}_1 + \lambda_2 c_2^{(0)} \vec{\mathbf{m}}_2 + \cdots + \lambda_{M-1} c_{M-1}^{(0)} \vec{\mathbf{m}}_{M-1}. \tag{12.53}$$

Now note that $\vec{\mathbf{U}}^1$ can be expanded over the same basis using a counterpart of (12.52):

$$\vec{\mathbf{U}}^1 = c_1^{(1)} \vec{\mathbf{m}}_1 + c_2^{(1)} \vec{\mathbf{m}}_2 + \cdots + c_{M-1}^{(1)} \vec{\mathbf{m}}_{M-1} \tag{12.54}$$

for some $c_j^{(1)}$. Comparing the last two equations we conclude that

$$c_j^{(1)} = \lambda_j c_j^{(0)}, \qquad j = 1, 2, \ldots, M-1. \tag{12.55a}$$

Similarly:

$$c_j^{(n)} = \lambda_j^n c_j^{(0)}, \qquad j = 1, 2, \ldots, M-1. \tag{12.55b}$$

(Note that while the superscript 'n' in $\vec{\mathbf{U}}^n$ denotes the time level t_n, on the r.h.s. of (12.55b) the same superscript denotes the power exponent.) From (12.55b) one sees that *coordinates* $c_j^{(n)}$ (and hence the norm) of vector $\vec{\mathbf{U}}^n$ grow or decay depending on whether

$$|\lambda_j| > 1 \qquad \text{or} \qquad |\lambda_j| < 1. \tag{12.56}$$

respectively.

Let us now clarify how the above applies to the **stability** of a given discretization scheme. Since the Heat equation, which we study in this Lecture, is linear, then the numerical error satisfies the same evolution equation as the solution; the boxed statement after Eq. (12.20). Therefore:

- If for at least one j, one has $|\lambda_j| > 1$, then the discretization scheme is unstable;
- Otherwise, i.e. if *all* $|\lambda_j| \le 1$, then the scheme is stable.

Remark 5: In advanced texts on Numerical Analysis one also considers a somewhat rare subcase of the second case where there are repeated eigenvalues with absolute value of 1. For general matrices $\mathcal{M}$, this may lead to a slow (sub-exponential) growth of the error. However, for diagonalizable matrices, this never occurs. (This is shown in graduate courses on Linear Algebra or Differential Equations.)

Remark 6: You may notice that formulas and conclusions of this Appendix, starting with Eq. (12.55b), are very similar to those in the stability analysis of general-purpose methods in Section 5.4.1 of Lecture 5.[41] This is not a coincidence. Indeed, as we pointed out after Eq. (12.16), all we do in this Lecture is apply the simple Euler method (which is one of the general-purpose methods) to the *system* of ODEs (12.16).

12.7 Questions for self-assessment

1. State the minimax principle and provide its intuitive interpretation. When can this principle be useful?
2. Obtain (12.12).
3. State the Lax Equivalence Theorem and provide a justification for it, based on (12.15).

[41] Quantities c_j here are direct counterparts of z_j there.

4. Make sure you can obtain (12.16) as explained in the text below that equation. Where are the boundary conditions (12.8) used in this derivation?
5. Make sure you can obtain (12.17) from (12.12).
6. Describe the idea behind Method 1 of stability analysis of the Heat equation.
7. Make sure you can obtain Eq. (12.20) as explained in the text.
8. Obtain Eq. (12.53) as explained in the text.
9. What will happen to the solution of scheme (12.12) if condition (12.30) is not satisfied?
10. Describe the idea behind the von Neumann stability analysis.
11. Make sure you can obtain Eqs. (12.34) and (12.35).
12. Answer the QSA posed in Remark 2 after the description of the von Neumann stability analysis.
13. Describe advantages and disadvantages of the von Neumann method relative to Method 1.
14. What piece of information would be required to turn a von Neumann-like analysis from a necessary to a sufficient condition of stability?
15. Which harmonics are "most dangerous" from the point of view of making scheme (12.12) unstable? How would you proceed answering this question for an arbitrary numerical scheme?
16. Make sure you can follow the derivation of (12.45).
17. Can you recall a counterpart of the Douglas method for ODEs?
18. Which factors affect the accuracy of a numerical scheme?