Math 6630: Numerical Solutions of Partial Differential Equations
Finite difference methods for time-dependent problems, Part II

See LeVeque 2007, Chapter 9,
Langtangen and Linge 2017, Chapter 3,

Kreiss, Oliger, and Gustafsson 2013, Chapters 1, 3, 6

Akil Narayan[1]

[1]Department of Mathematics, and Scientific Computing and Imaging (SCI) Institute
University of Utah

February 22, 2023

FD for parabolic problems

We've considered the problem and FD discretization,

$$u_t = u_{xx}, \qquad u(x,0) = u_0(x) \quad , x \in [0, 2\pi)$$
$$D^+ u_j^n = D_- D_+ u_j^n,$$

with periodic boundary conditions, and

- Equidistant discretization for x and t
- $x_j = \frac{2\pi j}{M}$, $j \in [M]$. Periodic BC's: we identify $x_M \leftrightarrow x_0$.
  $h = \Delta x = x_{j+1} - x_j$
- $t_n = nk$, $k > 0$ for $n = 0, 1, \ldots$
  $k = \Delta t = t_{k+1} - t_k$
- $u_j^n \approx u(x_j, t_n)$, $\boldsymbol{u}^n = (u_0^n, \ldots, u_{M-1}^n)^T$

Today: Stability, accuracy, convergence, etc.

Method of lines

$$D^+ u_j^n = D_- D_+ u_j^n,$$

The scheme above is fully discrete.

A more transparent understanding of algorithmic behavior can be gained from investigating the semi-discrete scheme:

$$u_t = u_{xx} \quad \xrightarrow{\text{Discretize space}} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \boldsymbol{A}\boldsymbol{u}(t), \qquad \boldsymbol{u} = (u_1(t), \ldots, u_M(t))^T$$

With periodic boundary conditions, then $\boldsymbol{A}$ is the matrix,

$$h^2 \boldsymbol{A} = \begin{pmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & & \\ 1 & & & 1 & -2 \end{pmatrix}$$

Method of lines, II

$$u_t = u_{xx} \quad \xrightarrow{\text{Discretize space}} \quad \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \boldsymbol{A}\boldsymbol{u}(t), \qquad \boldsymbol{u} = \left(u_1(t), \ldots, u_M(t)\right)^T$$

This reduction of a *partial* differential equation, to an *ordinary* one through discretization, is called the method of lines.

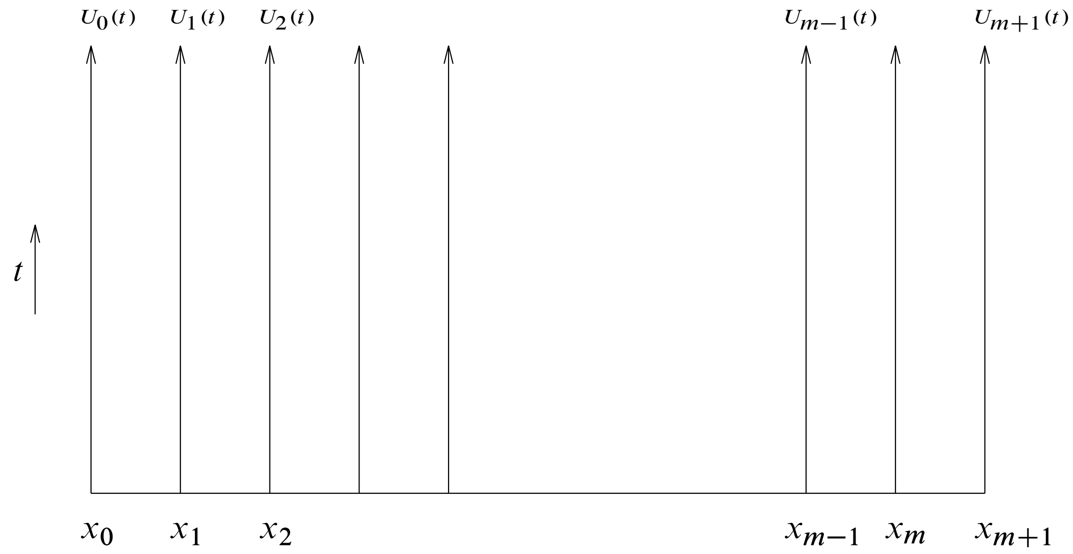


Figure: Method of lines visualization. LeVeque 2007, Figure 9.2

Method of lines, III

$u' = \lambda u$

$$u_t = u_{xx} \quad \xrightarrow{\text{Discretize space}} \quad \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \boldsymbol{A}\boldsymbol{u}(t), \qquad \boldsymbol{u} = (u_1(t), \ldots, u_M(t))^T$$

The semi-discrete form is useful in *decoupling* space and time.

In particular, it's something we know how to understand from a time-integration point of view:

- Stability (A-stabiltiy, 0-stability)
- Accuracy (time discretization)
- Convergence (conditioned on a fixed spatial discretization)

Convergernce to the solution of the original *PDE solution* does require some interaction of space and time.

Stability

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \boldsymbol{A}\boldsymbol{u}(t), \qquad\qquad \boldsymbol{u} = (u_1(t), \ldots, u_M(t))^T$$

We understand how to generate reasonable schemes for this: any 0-stable method could suffice.

To fix some details, one typically initially considers the simplest scheme to understand the system: Forward Euler.

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + k\boldsymbol{A}\boldsymbol{u}^n.$$

This is a *linear* ODE, and so one simple concept to explore is A-stability.

Is it reasonable to expect behavior of the discrete solution corresponding to A-stability?

To determine stability, the eigenvalues/vectors of $\boldsymbol{A}$ are explicitly computable:

$$\lambda_j(\boldsymbol{A}) = -\frac{4}{h^2}\sin^2\left(\frac{\pi\widetilde{j}}{2M}\right), \qquad \widetilde{j} := \begin{cases} j-1, & j \text{ odd} \\ j, & j \text{ even} \end{cases} \qquad j \in [M]$$

Note that the eigenvalues all have negative real parts ... as we hope for.

Stability

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \boldsymbol{A}\boldsymbol{u}(t), \qquad \boldsymbol{u} = (u_1(t), \ldots, u_M(t))^T$$

We understand how to generate reasonable schemes for this: any 0-stable method could suffice.

To fix some details, one typically initially considers the simplest scheme to understand the system: Forward Euler.

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + k\boldsymbol{A}\boldsymbol{u}^n.$$

This is a *linear* ODE, and so one simple concept to explore is A-stability.

Is it reasonable to expect behavior of the discrete solution corresponding to A-stability?

To determine stability, the eigenvalues/vectors of $\boldsymbol{A}$ are explicitly computable:

$$\lambda_j(\boldsymbol{A}) = -\frac{4}{h^2}\sin^2\left(\frac{\pi\widetilde{j}}{2M}\right), \qquad \widetilde{j} := \begin{cases} j-1, & j \text{ odd} \\ j, & j \text{ even} \end{cases} \qquad j \in [M]$$

Note that the eigenvalues all have negative real parts ... as we hope for.

Stability

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \boldsymbol{A}\boldsymbol{u}(t), \qquad \boldsymbol{u} = (u_1(t), \ldots, u_M(t))^T$$

We understand how to generate reasonable schemes for this: any 0-stable method could suffice.

To fix some details, one typically initially considers the simplest scheme to understand the system: Forward Euler.

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + k\boldsymbol{A}\boldsymbol{u}^n.$$

This is a *linear* ODE, and so one simple concept to explore is A-stability.

Is it reasonable to expect behavior of the discrete solution corresponding to A-stability?

To determine stability, the eigenvalues/vectors of $\boldsymbol{A}$ are explicitly computable:

$$\lambda_j(\boldsymbol{A}) = -\frac{4}{h^2}\sin^2\left(\frac{\pi\widetilde{j}}{2M}\right), \qquad \widetilde{j} := \begin{cases} j-1, & j \text{ odd} \\ j, & j \text{ even} \end{cases} \qquad j \in [M]$$

Note that the eigenvalues all have negative real parts ... as we hope for.

Stiffness

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \boldsymbol{A}\boldsymbol{u}(t), \qquad \boldsymbol{u} = (u_1(t), \ldots, u_M(t))^T$$

$$\lambda_j(\boldsymbol{A}) \sim -\frac{4}{h^2}\sin^2\left(\frac{\pi \widetilde{j}}{2M}\right), \quad \widetilde{j} := \begin{cases} j-1, & j \text{ odd} \\ j, & j \text{ even} \end{cases} \quad j \in [M]$$

All these eigenvalues lie in the left half-plane, on the real axis. In particular,

$$\lambda_{\min}(\boldsymbol{A}) = -\frac{4}{h^2} \sim 4M^2 \qquad \lambda_{\max}(\boldsymbol{A}) \sim -1$$

Stiffness

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \boldsymbol{A}\boldsymbol{u}(t), \qquad \boldsymbol{u} = (u_1(t), \ldots, u_M(t))^T$$

$$\lambda_j(\boldsymbol{A}) \sim -\frac{4}{h^2}\sin^2\left(\frac{\pi \widetilde{j}}{2M}\right), \widetilde{j} \qquad := \begin{cases} j-1, & j \text{ odd} \\ j, & j \text{ even} \end{cases} \quad j \qquad \in [M]$$

All these eigenvalues lie in the left half-plane, on the real axis. In particular,

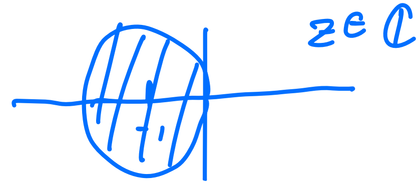
$$\lambda_{\min}(\boldsymbol{A}) = -\frac{4}{h^2} \sim 4M^2 \qquad \lambda_{\max}(\boldsymbol{A}) \sim -1$$

Therefore, there are some parts of the solution that vary slowly (small $|\lambda|$) and other parts of the solution that vary quickly (large $|\lambda|$).

This is a classic sign of stiffness of an ODE – since even moderate M causes large values of $\lambda_{\min}/\lambda_{\max}$, this is a stiff system for those values of M.

Although we have attempted to separate space and time, our choice of spatial discretization will impact our time discretization.

Stability



$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \boldsymbol{A}\boldsymbol{u}(t), \qquad \boldsymbol{u} = (u_1(t), \ldots, u_M(t))^T$$

What does A-stability tell us about the time discretization? For Forward Euler, recall that the region of stability is defined by,

$$|z+1| \leqslant 1, \qquad z = \lambda k,$$

$u' = \lambda u$

with λ being the eigevalues of $\boldsymbol{A}$.

Stability

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \boldsymbol{A}\boldsymbol{u}(t), \qquad \boldsymbol{u} = \left(u_1(t), \ldots, u_M(t)\right)^T$$

What does A-stability tell us about the time discretization? For Forward Euler, recall that the region of stability is defined by,

$$|z+1| \leqslant 1, \qquad z = \lambda k,$$

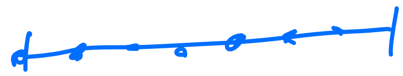
with λ being the eigevalues of $\boldsymbol{A}$.

Since $z = \lambda k$ is real-valued (and negative in this case), we really have the condition,

$$z \geqslant -2 \quad \Longrightarrow \quad k|\lambda_{\min}(\boldsymbol{A})| \leqslant 2 \quad \Longrightarrow \quad k \leqslant \frac{h^2}{2}$$

Note that this is a rather disappointing stability requirement. (Consider, say, $h = 0.01$)

Stability



$u_{xx} \approx \frac{1}{h^2}(\cdots)$

u_{xxxx}

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \boldsymbol{A}\boldsymbol{u}(t), \qquad \boldsymbol{u} = (u_1(t), \ldots, u_M(t))^T$$

What does A-stability tell us about the time discretization? For Forward Euler, recall that the region of stability is defined by,

$$|z+1| \leqslant 1, \qquad z = \lambda k,$$

with λ being the eigevalues of $\boldsymbol{A}$.

Since $z = \lambda k$ is real-valued (and negative in this case), we really have the condition,

$$z \geqslant -2 \quad \Longrightarrow \quad k|\lambda_{\min}(\boldsymbol{A})| \leqslant 2 \quad \Longrightarrow \quad k \leqslant \frac{h^2}{2}$$

Note that this is a rather disappointing stability requirement. (Consider, say, $h = 0.01$)

For this PDE, violating this notion of stability is bad: this PDE dissipates energy. Violating stability causes energy to grow.

Note that changing the type of explicit time-stepping scheme (RK, multi-step, etc) does not really change this stability condition, up to some $\mathcal{O}(1)$ constants.

The only real remedy is an A-stable (implicit) scheme.

Local truncation error

$$u_t = u_{xx}, \qquad\qquad u(x,0) = u_0(x)$$
$$D^+ u_j^n = D_- D_+ u_j^n,$$

For computing the local truncation error, considering the semi-discrete scheme does not provide much benefit.

The LTE is the scheme residual when the exact (smooth) solution is inserted:

$$\mathrm{LTE}^n = D^+ u(x_j, t_n) - D_- D_+ u(x_j, t_n) \sim \mathcal{O}(h^2 + k).$$

As before, we say a scheme is consistent if $\lim_{k,h\downarrow 0} \mathrm{LTE}^n = 0$.

Local truncation error

$$u_t = u_{xx}, \qquad u(x,0) = u_0(x)$$
$$D^+ u_j^n = D_- D_+ u_j^n,$$

For computing the local truncation error, considering the semi-discrete scheme does not provide much benefit.

The LTE is the scheme residual when the exact (smooth) solution is inserted:

$$\mathrm{LTE}^n = D^+ u(x_j, t_n) - D_- D_+ u(x_j, t_n) \sim \mathcal{O}(h^2 + k).$$

As before, we say a scheme is consistent if $\lim_{k,h \downarrow 0} \mathrm{LTE}^n = 0$. Naturally, the temporal order of convergence k^p would change depending on the LTE of the time-stepping scheme.

Without directly considering cost of space vs time discretization, one would logically want to balance the LTE by choosing $k \sim h^2$, which is similar to the stability condition.

However, we've already seen that this is not really an attractive strategy for choosing k, motivating that this scheme is not really a good one.

Convergence, I

As usual, the holy grail is convergence. The idea for how to proceed is similar to what we've seen before:

Suppose numerical solution satisfies the scheme exactly:

$$\boldsymbol{u}^{n+1} = \boldsymbol{B}\boldsymbol{u}^{n} + \boldsymbol{f}^{n},$$

where

- $\boldsymbol{B}$ is a matrix such $\boldsymbol{k}\boldsymbol{A}$ for the Forward Euler method
- $\boldsymbol{f}^n$ is any inhomogeneity in the equation (e.g., the term f in $u_t = u_{xx} + f(x,t)$)

Convergence, I

As usual, the holy grail is convergence. The idea for how to proceed is similar to what we've seen before:

Suppose numerical solution satisfies the scheme exactly:

$$\boldsymbol{u}^{n+1} = \boldsymbol{B}\boldsymbol{u}^n + \boldsymbol{f}^n,$$

where

- $\boldsymbol{B}$ is a matrix such $\boldsymbol{k}\boldsymbol{A}$ for the Forward Euler method
- $\boldsymbol{f}^n$ is any inhomogeneity in the equation (e.g., the term f in $u_t = u_{xx} + f(x,t)$)

The exact solution $u(x,t)$ at the grids points $\boldsymbol{U}(t)$ satisfies the scheme with an LTE correction $\boldsymbol{\tau}_n$:

$$\boldsymbol{U}(t_{n+1}) = \boldsymbol{B}\boldsymbol{U}(t_n) + \boldsymbol{f}^n + k\boldsymbol{\tau}^n,$$

Convergence, I

As usual, the holy grail is convergence. The idea for how to proceed is similar to what we've seen before:

Suppose numerical solution satisfies the scheme exactly:

$$\boldsymbol{u}^{n+1} = \boldsymbol{B}\boldsymbol{u}^n + \boldsymbol{f}^n,$$

where

- $\boldsymbol{B}$ is a matrix such $\boldsymbol{k}\boldsymbol{A}$ for the Forward Euler method
- $\boldsymbol{f}^n$ is any inhomogeneity in the equation (e.g., the term f in $u_t = u_{xx} + f(x,t)$)

The exact solution $u(x,t)$ at the grids points $\boldsymbol{U}(t)$ satisfies the scheme with an LTE correction $\boldsymbol{\tau}_n$:

$$\boldsymbol{U}(t_{n+1}) = \boldsymbol{B}\boldsymbol{U}(t_n) + \boldsymbol{f}^n + k\boldsymbol{\tau}^n,$$

Subtracting these two, the error $\boldsymbol{e}_n := \boldsymbol{U}(t_n) - \boldsymbol{u}^n$ satisfies,

$$\boldsymbol{e}_{n+1} = \boldsymbol{B}\boldsymbol{e}_n + k\boldsymbol{\tau}^n,$$

Convergence, II

$$\begin{aligned} \boldsymbol{u}^{n+1} &= \boldsymbol{B}\boldsymbol{u}^n + \boldsymbol{f}^n, \\ \boldsymbol{U}(t_{n+1}) &= \boldsymbol{B}\boldsymbol{U}(t_n) + \boldsymbol{f}^n + k\boldsymbol{\tau}^n, \\ \boldsymbol{e}_{n+1} &= \boldsymbol{B}\boldsymbol{e}_n + k\boldsymbol{\tau}^n, \end{aligned}$$

Iterating the error equation, we conclude,

$$\boldsymbol{e}_n = \boldsymbol{B}^n \boldsymbol{e}_0 + k \sum_{j=1}^{n} \boldsymbol{B}^{n-j} \boldsymbol{\tau}^{j-1}$$

NB: the superscripts n and $n-j$ on $\boldsymbol{B}$ *are* exponents.

Convergence, III

Therefore,

$$\|\boldsymbol{e}_n\| = \|\boldsymbol{B}^n\| \|\boldsymbol{e}_0\| + k \sum_{j=1}^{n} \|\boldsymbol{B}^{n-j}\| \|\boldsymbol{\tau}^{j-1}\|$$

Convergence, III

Therefore,

$$\|\boldsymbol{e}_n\| = \|\boldsymbol{B}^n\| \|\boldsymbol{e}_0\| + k \sum_{j=1}^{n} \|\boldsymbol{B}^{n-j}\| \|\boldsymbol{\tau}^{j-1}\|$$

This reveals that we need to control $\boldsymbol{B}^n$, motivating a new definition.

Definition

A numerical scheme of the form $\boldsymbol{u}^{n+1} = \boldsymbol{B}\boldsymbol{u}^n + \boldsymbol{f}^n$ for computing a solution up to terminal time T is Lax-Richtmyer stable if

$$\|\boldsymbol{B}^n\| \leqslant C(T),$$

for all k sufficiently small and all time indices n satisfying $nk \leqslant T$.

Convergence, III

Therefore,

$$\|\boldsymbol{e}_n\| = \|\boldsymbol{B}^n\|\|\boldsymbol{e}_0\| + k\sum_{j=1}^{n} \|\boldsymbol{B}^{n-j}\|\|\boldsymbol{\tau}^{j-1}\|$$

This reveals that we need to control $\boldsymbol{B}^n$, motivating a new definition.

Definition

A numerical scheme of the form $\boldsymbol{u}^{n+1} = \boldsymbol{B}\boldsymbol{u}^n + \boldsymbol{f}^n$ for computing a solution up to terminal time T is Lax-Richtmyer stable if

$$\|\boldsymbol{B}^n\| \leqslant C(T), \qquad \|B\| \leq 1$$

for all k sufficiently small and all time indices n satisfying $nk \leqslant T$.

In practice, showing $\|\boldsymbol{B}\| \leqslant 1 + Ck$ for some constant C independent of k is enough.

$$\|B^n\| \leq \|B\|^n \leq \left(1 + \frac{T}{N} \cdot \frac{1}{T} C\right)^n \sim e^{C/T}$$

Lax-Richtmyer, redux

Convergence of the scheme, under consistency and (Lax-Richtmyer) stability follows:

$$
\begin{aligned}
\|\boldsymbol{e}_n\| &= \|\boldsymbol{B}^n\| \|\boldsymbol{e}_0\| + k \sum_{j=1}^{n} \|\boldsymbol{B}^{n-j}\| \|\boldsymbol{\tau}^{j-1}\| \\
&\stackrel{\text{stability}}{\leqslant} C(T) \left[\|\boldsymbol{e}_0\| + kn \max_{j \in [n]} \|\boldsymbol{\tau}^{j-1}\| \right], \\
&\leqslant C(T) \left[\|\boldsymbol{e}_0\| + T \max_{j \in [n]} \|\boldsymbol{\tau}^{j-1}\| \right], \\
&\xrightarrow{k,h \downarrow 0+ \text{ consistency}} 0,
\end{aligned}
$$

where we additionally need $\boldsymbol{e}^0 \to 0$ as $k \downarrow 0$.

Lax-Richtmyer, redux

Convergence of the scheme, under consistency and (Lax-Richtmyer) stability follows:

$$\begin{aligned}
\|\boldsymbol{e}_n\| &= \|\boldsymbol{B}^n\|\|\boldsymbol{e}_0\| + k\sum_{j=1}^{n}\|\boldsymbol{B}^{n-j}\|\|\boldsymbol{\tau}^{j-1}\| \\
&\stackrel{\text{stability}}{\leqslant} C(T)\left[\|\boldsymbol{e}_0\| + kn\max_{j\in[n]}\|\boldsymbol{\tau}^{j-1}\|\right], \\
&\leqslant C(T)\left[\|\boldsymbol{e}_0\| + T\max_{j\in[n]}\|\boldsymbol{\tau}^{j-1}\|\right], \\
&\xrightarrow{k,h\downarrow 0+\ \text{consistency}} 0,
\end{aligned}$$

where we additionally need $\boldsymbol{e}^0 \to 0$ as $k \downarrow 0$.
We have just shown part of the following result:

Theorem (Lax-Richtmyer Equivalence)

A linear scheme is convergent if and only if it is consistent and (Lax-Richtmyer) stable.

I.e.,

$$\text{Stability} + \text{Consistency} = \text{Convergence}$$

Achieving stability

How would we achieve (Lax-Richtmyer) stability? The general form is,

$$\boldsymbol{u}^{n+1} = \boldsymbol{B}\boldsymbol{u}^n + \boldsymbol{f}^n,$$

and our Forward Euler in time, central difference in space approximation is,

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + k\boldsymbol{A}\boldsymbol{u}^n = \left(\boldsymbol{I} + k\boldsymbol{A}\right)\boldsymbol{u}^n,$$

so for stability, say in the 2-norm, we require,

$$\left\|\left(\boldsymbol{I} + k\boldsymbol{A}\right)^n\right\|_2 \leqslant 1.$$

Achieving stability

How would we achieve (Lax-Richtmyer) stability? The general form is,

$$\boldsymbol{u}^{n+1} = \boldsymbol{B}\boldsymbol{u}^n + \boldsymbol{f}^n,$$

and our Forward Euler in time, central difference in space approximation is,

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + k\boldsymbol{A}\boldsymbol{u}^n = (\boldsymbol{I} + k\boldsymbol{A})\,\boldsymbol{u}^n,$$

so for stability, say in the 2-norm, we require,

$$\|(\boldsymbol{I} + k\boldsymbol{A})^n\|_2 \leqslant 1.$$

Using submutiplicativity of the norm, this is ensured with

$$\|\boldsymbol{I} + k\boldsymbol{A}\|_2 \leqslant 1,$$

which, in turn due to symmetry of $\boldsymbol{I}$, $\boldsymbol{A}$ requires,

$$|1 + k\lambda_j(\boldsymbol{A})| \leqslant 1.$$

Achieving stability

How would we achieve (Lax-Richtmyer) stability? The general form is,

$$\boldsymbol{u}^{n+1} = \boldsymbol{B}\boldsymbol{u}^n + \boldsymbol{f}^n,$$

and our Forward Euler in time, central difference in space approximation is,

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + k\boldsymbol{A}\boldsymbol{u}^n = \left(\boldsymbol{I} + k\boldsymbol{A}\right)\boldsymbol{u}^n,$$

so for stability, say in the 2-norm, we require,

$$\left\|\left(\boldsymbol{I} + k\boldsymbol{A}\right)^n\right\|_2 \leqslant 1.$$

Using submutiplicativity of the norm, this is ensured with

$$\|\boldsymbol{I} + k\boldsymbol{A}\|_2 \leqslant 1,$$

which, in turn due to symmetry of $\boldsymbol{I}$, $\boldsymbol{A}$ requires,

$$|1 + k\lambda_j(\boldsymbol{A})| \leqslant 1.$$

Since all eigenvalues of $\boldsymbol{A}$ are real and negative, this is ensured via,

$$k|\lambda_{\min}(\boldsymbol{A})| \leqslant 2 \quad \Longrightarrow \quad k \leqslant \frac{h^2}{2}$$

which is *exactly* the same requirement we obtained from A-stability.

Scheme convergence

Thus, we have that

$$u_t = u_{xx} \quad \longrightarrow \quad D^+ u_j^n = D_+ D_- u_j^n$$

has an LTE and stability criterion:

$$\mathrm{LTE}_n = \mathcal{O}(k^2 + h)$$
$$k \leqslant \frac{h^2}{2}$$

Thus, under these conditions, we expect the scheme error to behave like $k + h^2$.

Scheme convergence

Thus, we have that

$$u_t = u_{xx} \quad \longrightarrow \quad D^+ u_j^n = D_+ D_- u_j^n$$

has an LTE and stability criterion:

$$\text{LTE}_n = \mathcal{O}(k^2 + h)$$
$$k \leqslant \frac{h^2}{2}$$

Thus, under these conditions, we expect the scheme error to behave like $k + h^2$.

This explain many "weird" issues we observed when naively trying to ascertain convergence of this method:

- Things are unstable if we don't satisfy $k \lesssim h^2$. In particular $k \sim h$ is not useful.
- How would we numerically verify h convergence? We'd need to
  - Pick a smallest h, say $h_{\min}$
  - Fix $k \leqslant h_{\min}^2/2$
  - Compare errors for $h = h_{\min}, 2h_{\min}, 4h_{\min}, 8h_{\min}, \ldots$
- How would we numerically verify k convergence?
  - $k \gg h^2$ is not possible, $k \ll h^2$ is not possible.
  - When refining k, must correspondingly refine h to satisfy, $h \sim \sqrt{2k}$.

Scheme convergence

Thus, we have that

$$u_t = u_{xx} \quad \longrightarrow \quad D^+ u_j^n = D_+ D_- u_j^n$$

has an LTE and stability criterion:

$$\mathrm{LTE}_n = \mathcal{O}(k^2 + h)$$
$$k \leqslant \frac{h^2}{2}$$

Thus, under these conditions, we expect the scheme error to behave like $k + h^2$.

This explain many "weird" issues we observed when naively trying to ascertain convergence of this method:

- Things are unstable if we don't satisfy $k \lesssim h^2$. In particular $k \sim h$ is not useful.
- How would we numerically verify h convergence? We'd need to
  - Pick a smallest h, say $h_{\min}$
  - Fix $k \leqslant h_{\min}^2/2$
  - Compare errors for $h = h_{\min}, 2h_{\min}, 4h_{\min}, 8h_{\min}, \ldots$
- How would we numerically verify k convergence?
  - $k \gg h^2$ is not possible, $k \ll h^2$ is not possible.
  - When refining k, must correspondingly refine h to satisfy, $h \sim \sqrt{2k}$.

Scheme convergence

Thus, we have that

$$u_t = u_{xx} \quad \longrightarrow \quad D^+ u_j^n = D_+ D_- u_j^n$$

has an LTE and stability criterion:

$$\mathrm{LTE}_n = \mathcal{O}(k^2 + h)$$
$$k \leqslant \frac{h^2}{2}$$

Thus, under these conditions, we expect the scheme error to behave like $k + h^2$.

This explain many "weird" issues we observed when naively trying to ascertain convergence of this method:

- Things are unstable if we don't satisfy $k \lesssim h^2$. In particular $k \sim h$ is not useful.
- How would we numerically verify h convergence? We'd need to
  - Pick a smallest h, say h_{min}
  - Fix $k \leqslant h_{\mathrm{min}}^2/2$
  - Compare errors for $h = h_{\mathrm{min}}, 2h_{\mathrm{min}}, 4h_{\mathrm{min}}, 8h_{\mathrm{min}}, \ldots$
- How would we numerically verify k convergence?
  - $k \gg h^2$ is not possible, $k \ll h^2$ is not possible.
  - When refining k, must correspondingly refine h to satisfy, $h \sim \sqrt{2k}$.

Higher-order schemes?

What about "better" schemes?

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \boldsymbol{A}\boldsymbol{u}.$$

Suppose we choose a higher order explicit time-stepping method, say Runge-Kutta 4.

Higher-order schemes?

What about "better" schemes?

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \boldsymbol{A}\boldsymbol{u}.$$

Suppose we choose a higher order explicit time-stepping method, say Runge-Kutta 4.

- Stability requires $k \lesssim h^2$
- The LTE is $k^4 + h^2$.
- Fix k, and varying h to satisfy $k \leqslant h^2/2$ would allow us to detect h-convergence
- To detect k convergence, we require $h^2 \lesssim k^4$, which contradicts the stability condition

I.e., in this case there is little benefit to using RK4 – we won't see any benefit.

Higher-order schemes?

What about "better" schemes?

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \boldsymbol{A}\boldsymbol{u}.$$

Suppose we choose a higher order explicit time-stepping method, say Runge-Kutta 4.

- Stability requires $k \lesssim h^2$
- The LTE is $k^4 + h^2$.
- Fix k, and varying h to satisfy $k \leqslant h^2/2$ would allow us to detect h-convergence
- To detect k convergence, we require $h^2 \lesssim k^4$, which contradicts the stability condition

I.e., in this case there is little benefit to using RK4 – we won't see any benefit.

If we alternatively use Crank-Nicholson:

- Stability is unconditional ($\|\boldsymbol{B}^n\| \leqslant 1$ is automatic)
- The LTE is $k^2 + h^2$.
- Fix k, varying h to satisfy $k \leqslant h$ would allow us to detect h-convergence
- Fix h, varying k to satisfy $h \leqslant k$ would allow us to detect k-convergence

Von Neumann stability, I

We've seen that it's possible to directly verify Lax-Richtmyer stability.

But in even slightly more complicated scenarios, a similar analysis is quite difficult.

An alternative notion, Von Nuemann stability, is a much easier necessary (not sufficient) stability requirement.

Von Neumann stability, I

We've seen that it's possible to directly verify Lax-Richtmyer stability.

But in even slightly more complicated scenarios, a similar analysis is quite difficult.

An alternative notion, Von Nuemann stability, is a much easier necessary (not sufficient) stability requirement.

Von Neumann stability proceeds by ignoring boundary conditions, and realizing that for linear differential equations, *complex exponentials* are eigenfunctions.

E.g.,

$$\left(e^{i\omega x}\right)_{xx} = C(\omega)e^{i\omega x}.$$

A similar computation is true for reasonable spatial discretizations of derivatives.

Von Neumann stability, I

We've seen that it's possible to directly verify Lax-Richtmyer stability.

But in even slightly more complicated scenarios, a similar analysis is quite difficult.

An alternative notion, Von Nuemann stability, is a much easier necessary (not sufficient) stability requirement.

Von Neumann stability proceeds by ignoring boundary conditions, and realizing that for linear differential equations, *complex exponentials* are eigenfunctions.

E.g.,

$$\left(e^{i\omega x}\right)_{xx} = C(\omega) e^{i\omega x}.$$

A similar computation is true for reasonable spatial discretizations of derivatives.

Then a reasonable (somewhat empirical) notion of (Von Neumann) stability for a scheme would assert that the scheme does *not amplify* eigenfunctions in time.

Von Neumann stability, II

The general strategy for von Neumann stability on linear problems is to consider the scheme,

$$u_j^{n+1} = B(\boldsymbol{u}^n)$$

for a linear operator B acting on the degrees of freedom at time step j. If we make the ansatz,

$$\boldsymbol{u}^n = e^{i\omega \boldsymbol{x}} \quad \longrightarrow \quad u_j^n = e^{i\omega x_j} = e^{i\omega j h},$$

then we expect that plugging this into scheme will yield the expression,

$$u_j^{n+1} = g(\omega) e^{i\omega j h},$$

for some constant $g(\omega)$.[1]

[1] In principle g can depend on j, but it will not if the discretization is spatially homogeneous.

Von Neumann stability, II

The general strategy for von Neumann stability on linear problems is to consider the scheme,

$$u_j^{n+1} = B(\boldsymbol{u}^n)$$

for a linear operator B acting on the degrees of freedom at time step j. If we make the ansatz,

$$\boldsymbol{u}^n = e^{i\omega \boldsymbol{x}} \quad \longrightarrow \quad u_j^n = e^{i\omega x_j} = e^{i\omega j h},$$

then we expect that plugging this into scheme will yield the expression,

$$u_j^{n+1} = g(\omega) e^{i\omega j h},$$

for some constant $g(\omega)$.[1]

The function g is called the (Von Neumann) *amplification factor* of the scheme.

The scheme will be (Von Neumann) stable if $|g(\omega)| \leqslant 1$.[2]

[1] In principle g can depend on j, but it will not if the discretization is spatially homogeneous.
[2] Like for Lax-Richtmyer stability, we'll actually just need $|g(\omega)| \leqslant 1 + Ck$.

Examples

Example

Compute the Von Neumann stability condition for

$$D^+ u_j^n = D_0 u_j^n$$

Setting $u_j^n = e^{i\omega jh}$, and $u_j^{n+1} = g(\omega)e^{i\omega jh}$, then we have,

$$u_j^{n+1} = u_j^n + \frac{k}{h^2}\left[u_{j+1}^n - 2u_j^n + u_{j-1}^n\right]$$

$$\downarrow$$

$$g(\omega)e^{i\omega jh} = e^{i\omega jh} + \frac{k}{h^2}e^{i\omega jh}\left[e^{i\omega h} - 2 + e^{-i\omega h}\right],$$

i.e.,

$$g(\omega) = 1 + \frac{2k}{h^2}\left(\cos\omega h - 1\right).$$

Examples

Example

Compute the Von Neumann stability condition for

$$D^+ u_j^n = D_0 u_j^n$$

Setting $u_j^n = e^{i\omega jh}$, and $u_j^{n+1} = g(\omega) e^{i\omega jh}$, then we have,

$$u_j^{n+1} = u_j^n + \frac{k}{h^2}\left[u_{j+1}^n - 2u_j^n + u_{j-1}^n\right]$$

$$\downarrow$$

$$g(\omega) e^{i\omega jh} = e^{i\omega jh} + \frac{k}{h^2} e^{i\omega jh}\left[e^{i\omega h} - 2 + e^{-i\omega h}\right],$$

i.e.,

$$g(\omega) = 1 + \frac{2k}{h^2}\left(\cos\omega h - 1\right).$$

Since $-2 \leqslant (\cos\omega h - 1) \leqslant 0$, then $|g(\omega)| \leqslant 1$ if

$$\frac{2k}{h^2} \leqslant 1 \quad \Longrightarrow \quad k \leqslant \frac{h^2}{2}$$

Examples

Example

Compute the Von Neumann stability condition for

$$D^+ u_j^n = D_0 u_j^n$$

Example

Compute the Von Neumann stability condition for

$$D^+ u_j^n = \frac{1}{2} D_0 u_j^n + \frac{1}{2} D_0 u_j^{n+1}$$

Examples

Example

Compute the Von Neumann stability condition for

$$D^+ u_j^n = D_0 u_j^n$$

Example

Compute the Von Neumann stability condition for

$$D^+ u_j^n = \frac{1}{2} D_0 u_j^n + \frac{1}{2} D_0 u_j^{n+1}$$

Example

Compute the Von Neumann stability condition for

$$D^+ u_j^n = D_+ D_+ u_j^n$$

References I

Kreiss, Heinz-Otto, Joseph Oliger, and Bertil Gustafsson (2013). *Time-Dependent Problems and Difference Methods*. John Wiley & Sons. ISBN: 978-1-118-54852-3.

Langtangen, Hans Petter and Svein Linge (2017). *Finite Difference Computing with PDEs: A Modern Software Approach*. Springer. ISBN: 978-3-319-55456-3.

LeVeque, Randall J. (2007). *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM. ISBN: 978-0-89871-783-9.

Richtmyer, Robert D. and K. W. Morton (1994). *Difference Methods for Initial-Value Problems*. Malabar, Fla. ISBN: 978-0-89464-763-5.