Math 6620: Analysis of Numerical Methods, II
Finite difference methods for time-dependent problems, Part III

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

FD for hyperbolic problems

We'll focus here on hyperbolic problems, for which the prototypical example is,

$$u_t + a u_x = 0, \qquad u(x,0) = u_0(x), \qquad a \in \mathbb{R}, \qquad x \in [0, 2\pi)$$

Again, we will impose periodic boundary conditions to understand the PDE.

We'll focus here on hyperbolic problems, for which the prototypical example is,

$$u_t + au_x = 0, \qquad u(x,0) = u_0(x), \qquad a \in \mathbb{R}, \qquad x \in [0, 2\pi)$$

Again, we will impose periodic boundary conditions to understand the PDE.

The symbol $\mathcal{P}$ of the operator $p(\partial_x) = -a\partial_x$ is,

$$\mathcal{F}\left[-au_x\right] = -ia\omega\mathcal{F}[u] = \mathcal{P}(\omega)\mathcal{F}[u],$$

so that the exact solution is

$$u(x,t) = \mathcal{F}^{-1}\left[e^{-ia\omega t}U_0(\omega)\right] = u_0(x - at),$$

i.e., this is a *wave* that moves with velocity a. (There is no dissipation.)

We'll focus here on hyperbolic problems, for which the prototypical example is,

$$u_t + au_x = 0, \qquad u(x,0) = u_0(x), \qquad a \in \mathbb{R}, \qquad x \in [0, 2\pi)$$

Again, we will impose periodic boundary conditions to understand the PDE.

The symbol $\mathcal{P}$ of the operator $p(\partial_x) = -a\partial_x$ is,

$$\mathcal{F}\left[-au_x\right] = -ia\omega\mathcal{F}[u] = \mathcal{P}(\omega)\mathcal{F}[u],$$

so that the exact solution is

$$u(x,t) = \mathcal{F}^{-1}\left[e^{-ia\omega t}U_0(\omega)\right] = u_0(x - at),$$

i.e., this is a *wave* that moves with velocity a. (There is no dissipation.)

So *why* we should we try to numerically solve this problem? (The exact solution is so easy!)
The hard versions of this problem to solve have:

- variable wave speed
- nonperiodic boundary conditions
- potential nonlinearities

These problems don't frequently have exact solutions, and most of the useful intuition can be attained by consdering the simple problem.

Energy for wave problems

$$u_t + a u_x = 0, \qquad u(x,0) = u_0(x), \qquad a \in \mathbb{R}, \qquad x \in [0, 2\pi)$$

How does energy behave for this PDE?

Energy for wave problems

$$u_t + a u_x = 0, \qquad u(x,0) = u_0(x), \qquad a \in \mathbb{R}, \qquad x \in [0, 2\pi)$$

How does energy behave for this PDE?

- Multiply by u
- Integrate over $[0, 2\pi]$

$$\frac{\mathrm{d}}{\mathrm{d}t} \|u(\cdot,t)\|^2 = \int_0^{2\pi} a u u_x \mathrm{d}x = \frac{a}{2} \int_0^{2\pi} \frac{\partial}{\partial x} u^2(x,t) \mathrm{d}x \overset{(*)}{=} 0.$$

where $(*)$ uses periodicity of u.

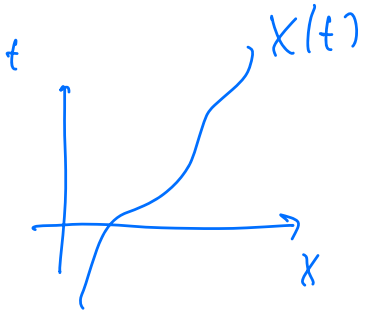
Hence, energy is *not* dissipated by this PDE.

Characteristic tracing, I

One additional consequence of the simplicity of this problem:

Consider a particle moving in space with position $X(t)$. Note that,

$$\frac{\mathrm{d}u(X(t),t)}{\mathrm{d}t} = (u(X(t),t))_t + (u(X(t),t))_x \frac{\mathrm{d}}{\mathrm{d}t} X(t)$$



Characteristic tracing, I

One additional consequence of the simplicity of this problem:

Consider a particle moving in space with position $X(t)$. Note that,

$$\frac{\mathrm{d}u(X(t),t)}{\mathrm{d}t} = \left(u(X(t),t)\right)_t + \left(u(X(t),t)\right)_x \frac{\mathrm{d}}{\mathrm{d}t} X(t)$$

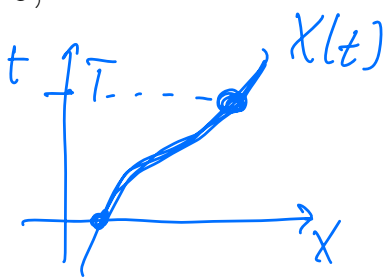
If we choose $X(t)$ to satisfy,

$$X'(t) = a, \qquad X(0) = x_0,$$

then

$$\frac{\mathrm{d}u(X(t),t)}{\mathrm{d}t} = \left(u(X(t),t)\right)_t + a\left(u(X(t),t)\right)_x = 0,$$

i.e., $u(X(t),t) = u(x_0,t)$ is constant for all time.



Characteristic tracing, I

One additional consequence of the simplicity of this problem:

Consider a particle moving in space with position $X(t)$. Note that,

$$\frac{\mathrm{d}u(X(t),t)}{\mathrm{d}t} = (u(X(t),t))_t + (u(X(t),t))_x \frac{\mathrm{d}}{\mathrm{d}t} X(t)$$

If we choose $X(t)$ to satisfy,

$$X'(t) = a, \qquad X(0) = x_0,$$

then

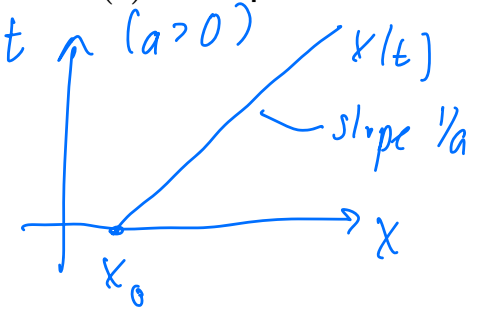
$$\frac{\mathrm{d}u(X(t),t)}{\mathrm{d}t} = (u(X(t),t))_t + a\,(u(X(t),t))_x = 0,$$

i.e., $u(X(t),t) = u(x_0,t)$ is constant for all time. Such trajectories $X(t)$ that preserve the value of u along them are called characteristics.

In this particular case, the characteristics are easy to identify:

$$X(t) = at + x_0,$$

which are lines in the (t,x) plane.



Characteristic tracing, II

$$u_t + a u_x = 0, \qquad u(x,0) = u_0(x),$$
$$u(X(t),t) = u(at + x_0, t) = u(x_0, t) = u_0(x_0).$$

This suggests a(n easy) scheme for computing $u(x,t)$ for some $t > 0$:

- Set $X(t) = x$.
- Compute $X(0) = x - at$
- Compute $u(x,t) = u(X(t),t) = u(x - at, 0) = u_0(x - at)$

$$u_t + a u_x = 0, \qquad u(x,0) = u_0(x),$$
$$u(X(t),t) = u(at + x_0, t) = u(x_0, t) = u_0(x_0).$$

This suggests a(n easy) scheme for computing $u(x,t)$ for some $t > 0$:

- Set $X(t) = x$.
- Compute $X(0) = x - at$
- Compute $u(x,t) = u(X(t),t) = u(x - at, 0) = u_0(x - at)$

This scheme is a called the method of characteristics, which is a *Lagrangian approach*.

This scheme shows another way to reveal something rather unusual about this PDE: the *exact* solution at (x,t) depends only on the initial data at x_0.

$$u_t + a u_x = 0, \qquad u(x,0) = u_0(x),$$
$$u(X(t),t) = u(at + x_0, t) = u(x_0, t) = u_0(x_0).$$

This suggests a(n easy) scheme for computing $u(x,t)$ for some $t > 0$:

- Set $X(t) = x$.
- Compute $X(0) = x - at$
- Compute $u(x,t) = u(X(t),t) = u(x - at, 0) = u_0(x - at)$

This scheme is a called the method of characteristics, which is a *Lagrangian approach*.

This scheme shows another way to reveal something rather unusual about this PDE: the *exact* solution at (x,t) depends only on the initial data at x_0.

A formalization of the above idea is through the domain of dependence:

$$D(x,t) := \{x - at\} \subset \mathbb{R}$$

$D(x,t)$ contains the set of points such that $u(x,t)$ depends *only* on the values $u(D(x,t),0)$.

$$u_t + a u_x = 0, \qquad u(x,0) = u_0(x),$$
$$u(X(t),t) = u(at + x_0, t) = u(x_0, t) = u_0(x_0).$$

This suggests a(n easy) scheme for computing $u(x,t)$ for some $t > 0$:

- Set $X(t) = x$.
- Compute $X(0) = x - at$
- Compute $u(x,t) = u(X(t),t) = u(x - at, 0) = u_0(x - at)$

This scheme is a called the method of characteristics, which is a *Lagrangian approach*.

This scheme shows another way to reveal something rather unusual about this PDE: the *exact* solution at (x,t) depends only on the initial data at x_0.

A formalization of the above idea is through the domain of dependence:

$$D(x,t) := \{x - at\} \subset \mathbb{R}$$

$D(x,t)$ contains the set of points such that $u(x,t)$ depends *only* on the values $u(D(x,t),0)$.

As an alternative example, $D(x,t) = \mathbb{R}$ for the heat equation for every (x,t) with $t > 0$.

Let's get to some schemes....

$$u_t + a u_x = 0, \qquad u(x,0) = u_0(x), \qquad x \in [0, 2\pi)$$

with periodic boundary conditions. Let's use our standard setup:

- 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 $n = 0, 1, \ldots$
  $k = \Delta t = t_{k+1} - t_k$
- $u_j^n \approx u(x_j, t_n)$, $\boldsymbol{u}^n = (u_0^n, \ldots, u_{M-1}^n)^T$

Let's get to some schemes....

$$u_t + a u_x = 0, \qquad u(x,0) = u_0(x), \qquad x \in [0, 2\pi)$$

with periodic boundary conditions. Let's use our standard setup:

- 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 $n = 0, 1, \ldots$
  $k = \Delta t = t_{k+1} - t_k$
- $u_j^n \approx u(x_j, t_n)$, $\boldsymbol{u}^n = (u_0^n, \ldots, u_{M-1}^n)^T$

With a little more experience than before, let's first consider a semi-discrete scheme:

$$u_x(x_j, t_n) \longrightarrow D_0 u_j^n = \frac{1}{2h}\left[u_{j+1}^n - u_{j-1}^n\right].$$

Semi-discrete form

This results in the linear ODE:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t) = \boldsymbol{A}\boldsymbol{u}, \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0,$$

where

$$\boldsymbol{A} = -\frac{a}{2h}\begin{pmatrix} 0 & 1 & & & -1 \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & & \\ 1 & & & -1 & 0 \end{pmatrix}$$

$\frac{d}{dt} u_j(t) = -\frac{a}{2h}\left[u_{j+1}(t) - u_{j-1}(t)\right]$

Let's consider Forward Euler for the time discretization:

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

We have several notions of stability to consider. Let's look at most of them. (For this PDE, we'll learn fairly consistent lessons.)

Absolute stability

In order to achieve absolute stability, we require that the spectrum of $\boldsymbol{A}$ lies in the region of stability for Forward Euler.

A direct computation reveals that:

$$\lambda_j(\boldsymbol{A}) = -\frac{ia}{h}\sin(2\pi j/M), \qquad j \in [M]$$

and thus in particular,

$$\Re\lambda_j = 0, \qquad -\frac{|a|}{h} \leqslant \Im\lambda_j \leqslant \frac{|a|}{h}.$$

Note that this operator is *not* stiff! I.e., $\max|\lambda_j| \sim M$.

Absolute stability

In order to achieve absolute stability, we require that the spectrum of $\boldsymbol{A}$ lies in the region of stability for Forward Euler.

A direct computation reveals that:

$$\lambda_j(\boldsymbol{A}) = -\frac{ia}{h}\sin(2\pi j/M), \qquad j \in [M]$$

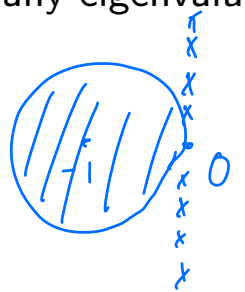
and thus in particular,

$$\Re\lambda_j = 0, \qquad -\frac{|a|}{h} \leqslant \Im\lambda_j \leqslant \frac{|a|}{h}.$$

Note that this operator is *not* stiff! I.e., $\max|\lambda_j| \sim M$.

The stability region for Forward Euler corresponds $z = \lambda k$, for λ any eigenvalue of $\boldsymbol{A}$, satisfying,

$$|1 + z| \leqslant 1.$$



For which values of $k > 0$ is this scheme stable?

Absolute stability

In order to achieve absolute stability, we require that the spectrum of $\boldsymbol{A}$ lies in the region of stability for Forward Euler.

A direct computation reveals that:

$$\lambda_j(\boldsymbol{A}) = -\frac{ia}{h}\sin(2\pi j/M), \qquad j \in [M]$$

and thus in particular,

$$\Re\lambda_j = 0, \qquad -\frac{|a|}{h} \leqslant \Im\lambda_j \leqslant \frac{|a|}{h}.$$

Note that this operator is *not* stiff! I.e., $\max|\lambda_j| \sim M$.

The stability region for Forward Euler corresponds $z = \lambda k$, for λ any eigenvalue of $\boldsymbol{A}$, satisfying,

$$|1 + z| \leqslant 1.$$

For which values of $k > 0$ is this scheme stable?

This scheme is not stable for any choice of $k > 0$. But note that for some explicit methods whose region of stability contains the imaginary axis, we can attain absolute stability with a condition like,

$$|a|k \lesssim h.$$

Von Neumann stability

Von Neumann stability will tell us something similar.

In the domain interior, the scheme reads,

$$u_j^{n+1} = u_j^n - \frac{ak}{2h}\left(u_{j+1}^n - u_{j-1}^n\right).$$

We make the ansatz $u_j^n = e^{ix_j\omega} = e^{ijh\omega}$, and $u_j^{n+1} = g(\omega)e^{ijh\omega}$.

Von Neumann stability

Von Neumann stability will tell us something similar.

In the domain interior, the scheme reads,

$$u_j^{n+1} = u_j^n - \frac{ak}{2h}\left(u_{j+1}^n - u_{j-1}^n\right).$$

We make the ansatz $u_j^n = e^{ix_j\omega} = e^{ijh\omega}$, and $u_j^{n+1} = g(\omega)e^{ijh\omega}$. We obtain,

$$g(\omega) = 1 - i\frac{ak}{h}\sin(\omega h),$$

which for generic ω satisfies,

$$|g(\omega)| > 1,$$

hence this scheme is not stable for any choice of $k > 0$.

Lax-Richtmyer stability

Lax-Richtmyer stability is similar to absolute stability:

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

A sufficient condition is to control the norm of $\boldsymbol{B}$, but

$$\|\boldsymbol{B}\|_2^2 \overset{(*)}{=} \max_j |\lambda_j(\boldsymbol{B})|^2 = 1 + k^2 \max_j |\lambda_j(\boldsymbol{A})|^2 \sim 1 + \left(\frac{ka}{h}\right)^2 > 1$$

where $(*)$ uses the fact that $\boldsymbol{B}$ is a normal matrix.

Thus, in this case it's not clear that we'll get stability.

Lax-Richtmyer stability

Lax-Richtmyer stability is similar to absolute stability:

$$\boldsymbol{u}^{n+1} = (\boldsymbol{I} + k\boldsymbol{A})\,\boldsymbol{u}^n = \boldsymbol{B}\boldsymbol{u}^n. \qquad u^n = (B^n)u^0$$

A sufficient condition is to control the norm of $\boldsymbol{B}$, but

$$\|\boldsymbol{B}\|_2^2 \overset{(*)}{=} \max_j |\lambda_j(\boldsymbol{B})|^2 = 1 + k^2 \max_j |\lambda_j(\boldsymbol{A})|^2 \sim 1 + \left(\frac{ka}{h}\right)^2 > 1$$

where $(*)$ uses the fact that $\boldsymbol{B}$ is a normal matrix.

Thus, in this case it's not clear that we'll get stability.

However, Lax-Richtmyer stability is a little more forgiving than the condition above. In particular, let's choose

$$k = \left(\frac{h}{a}\right)^2 \implies \|\boldsymbol{B}\|_2^2 \leqslant 1 + \left(\frac{ka}{h}\right)^2 = 1 + k$$

and thus,

$$\|\boldsymbol{B}^n\|_2 \leqslant \|\boldsymbol{B}\|_2^n = (1+k)^{n/2} \leqslant e^{T/2}.$$

Thus, we do attain stability, but at an unnecessarily steep timestep restriction.

Class Fri, March 22 Cancel

Alternative schemes

All the above discussions are really meant to convince you that there are better schemes.

One alternative is to change the time discretization: We've already mentioned that choosing a time integration scheme whose region of stability contains the imaginary axis would work.

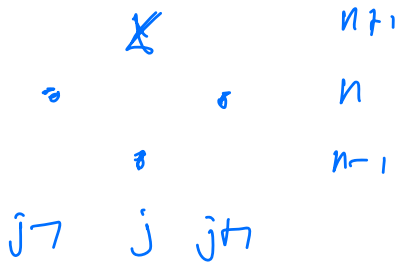
Alternative schemes

All the above discussions are really meant to convince you that there are better schemes.

One alternative is to change the time discretization: We've already mentioned that choosing a time integration scheme whose region of stability contains the imaginary axis would work.

Apart from those we've already discussed, one popular method is the Leapfrog method:

$$D^0 \boldsymbol{u}'(t) = \boldsymbol{A}\boldsymbol{u} \quad \longrightarrow \quad \boldsymbol{u}^{n+1} = \boldsymbol{u}^{n-1} + 2k\boldsymbol{A}\boldsymbol{u}^n.$$



Alternative schemes

All the above discussions are really meant to convince you that there are better schemes.

One alternative is to change the time discretization: We've already mentioned that choosing a time integration scheme whose region of stability contains the imaginary axis would work.

Apart from those we've already discussed, one popular method is the Leapfrog method:

$$D^0 \boldsymbol{u}'(t) = \boldsymbol{A}\boldsymbol{u} \quad \longrightarrow \quad \boldsymbol{u}^{n+1} = \boldsymbol{u}^{n-1} + 2k\boldsymbol{A}\boldsymbol{u}^n.$$

For absolute stability, the region of stability of this method is the imaginary axis, $\Re z = 0$, $-1 \leqslant \Im z \leqslant 1$.

Since the spectrum of $\boldsymbol{A}$ is purely imaginary, with maximum value ia/h, then stability is achieved with

$$k \leqslant h/a,$$

which is reasonable.

The main issue with the leapfrog method is that it is only marginally stable: there is no dissipation. In practice this means that slight deviations (say of the equation or of the data) can cause instabilities.

Adding dissipation, I

Another strategy is a somewhat empirical one: we have determined that the forward Euler discretization for

$$\boldsymbol{u}'(t) = \boldsymbol{A}\boldsymbol{u},$$

is not stable, i.e., the energy and in particular "modes" of the solution (e.g., the projection of $\boldsymbol{u}(t)$ onto the eigenvectors of $\boldsymbol{A}$) grow in time.

Adding dissipation, I

Another strategy is a somewhat empirical one: we have determined that the forward Euler discretization for

$$\boldsymbol{u}'(t) = \boldsymbol{A}\boldsymbol{u},$$

is not stable, i.e., the energy and in particular "modes" of the solution (e.g., the projection of $\boldsymbol{u}(t)$ onto the eigenvectors of $\boldsymbol{A}$) grow in time.

One high-level idea: introduce energy decay through dissipation. Essentially, we could instead try to solve,

$$u_t + au_x = \epsilon u_{xx},$$

for some "small" $\epsilon > 0$.

Adding dissipation, I

Another strategy is a somewhat empirical one: we have determined that the forward Euler discretization for

$$\boldsymbol{u}'(t) = \boldsymbol{A}\boldsymbol{u},$$

is not stable, i.e., the energy and in particular "modes" of the solution (e.g., the projection of $\boldsymbol{u}(t)$ onto the eigenvectors of $\boldsymbol{A}$) grow in time.

One high-level idea: introduce energy decay through dissipation. Essentially, we could instead try to solve,

$$u_t + au_x = \epsilon u_{xx},$$

for some "small" $\epsilon > 0$.

We are indeed making a "mistake" with this strategy, as the solution to the PDE will not match that of the original purely hyperbolic problem.

However, this high-level idea is actually quite effective in general practice and also yields some insight into theoretical considerations.

Adding dissipation, II

$$u_t + a u_x = \epsilon u_{xx},$$

We'll use the same spatial discretization for $a u_x$ (i.e., $a D_0 u_j^n$), and we'll use our standard discretization for ϵu_{xx}, which we know from parabolic problems is effective,

$$\epsilon u_{xx} \quad \longrightarrow \quad \epsilon D_+ D_- u_j^n,$$

resulting in the semi-discrete form,

$$\boldsymbol{u}'(t) = \left(\boldsymbol{A} + \widetilde{\boldsymbol{A}}_\epsilon \right) \boldsymbol{u}(t),$$

where $\boldsymbol{A}$ is as before, and $\widetilde{\boldsymbol{A}}_\epsilon$ corresponds to the ϵu_{xx} term.

Adding dissipation, II

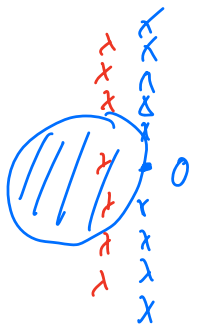
$$u_t + a u_x = \epsilon u_{xx},$$

We'll use the same spatial discretization for $a u_x$ (i.e., $a D_0 u_j^n$), and we'll use our standard discretization for ϵu_{xx}, which we know from parabolic problems is effective,

$$\epsilon u_{xx} \quad \longrightarrow \quad \epsilon D_+ D_- u_j^n,$$

resulting in the semi-discrete form,

$$\boldsymbol{u}'(t) = \left(\boldsymbol{A} + \widetilde{\boldsymbol{A}}_\epsilon \right) \boldsymbol{u}(t),$$



where $\boldsymbol{A}$ is as before, and $\widetilde{\boldsymbol{A}}_\epsilon$ corresponds to the ϵu_{xx} term.

The more important question is how to choose ϵ. Some initial considerations are:

- We'll insist on Forward Euler, so we want the eigenvalues of $\boldsymbol{A} + \widetilde{\boldsymbol{A}}_\epsilon$ to lie within the corresponding region of stability ($|1 + z| \leqslant 1$)
- Choosing ϵ too small will not introduce enough dissipation.
- Choosing ϵ too large will introduce so much dissipation that the operator $\boldsymbol{A} + \widetilde{\boldsymbol{A}}_\epsilon$ will become stiff (like u_{xx}) and will result in a $k \lesssim h^2$ stability restriction

Empirical investigations

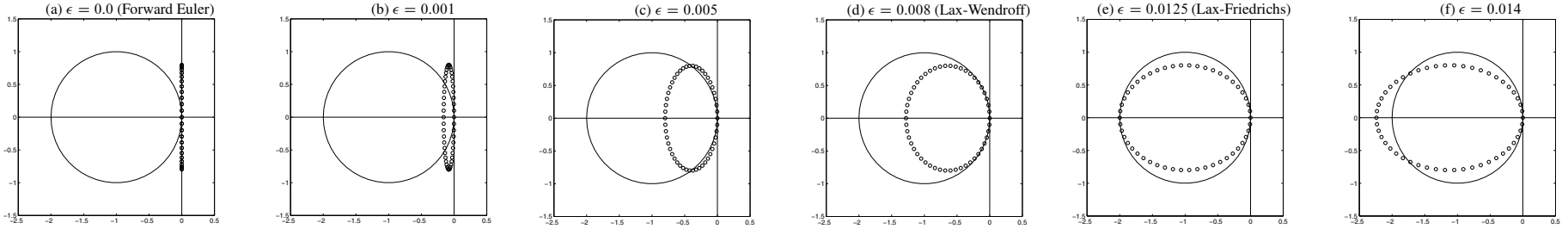


Figure: Eigenvalues of $\boldsymbol{A} + \widetilde{\boldsymbol{A}}_\epsilon$ for $a = 1$, $k = 0.8h$ (open circles), and various ϵ choices (increasing from left to right), versus the Forward Euler region of stability (black line boundary). From LeVeque 2007, Figure 10.1.

The key things to notice:

- The small ϵ and large ϵ regimes behaves as we would expect.
- There are "critical" values of ϵ that are just large/small enough to make the scheme stable.

The Lax-Friedrichs scheme

The spectrum of $\boldsymbol{A} + \widetilde{\boldsymbol{A}}_\epsilon$ is actually explicitly computable, allowing one to show that the eigenvalues actually lie on the boundary of an ellipse whose position + aspect ratio depend on ϵ.

The Lax-Friedrichs scheme

The spectrum of $\boldsymbol{A} + \widetilde{\boldsymbol{A}}_\epsilon$ is actually explicitly computable, allowing one to show that the eigenvalues actually lie on the boundary of an ellipse whose position + aspect ratio depend on ϵ.

In particular, one can take ϵ as large as possible so that the spectrum lies inside the Forward Euler region of stability. This corresponds to the choice,

$$\epsilon = \frac{h^2}{2k},$$

and results in the Lax-Friedrichs scheme. This scheme as we've derived it reads,

$$D^+ u_j^n = -aD_0 u_j^n + \underbrace{\frac{h^2}{2k}}_{\epsilon} D_+ D_- u_j^n.$$

The Lax-Friedrichs scheme

The spectrum of $\boldsymbol{A} + \widetilde{\boldsymbol{A}}_\epsilon$ is actually explicitly computable, allowing one to show that the eigenvalues actually lie on the boundary of an ellipse whose position + aspect ratio depend on ϵ.

In particular, one can take ϵ as large as possible so that the spectrum lies inside the Forward Euler region of stability. This corresponds to the choice,

$$\epsilon = \frac{h^2}{2k},$$

and results in the Lax-Friedrichs scheme. This scheme as we've derived it reads,

$$D^+ u_j^n = -aD_0 u_j^n + \underbrace{\frac{h^2}{2k}}_{\epsilon} D_+ D_- u_j^n.$$

However, it's more common to combine some terms and write this scheme as,

$$u_j^{n+1} = \frac{1}{2}\left(u_{j-1}^n + u_{j+1}^n\right) - kaD_0 u_j^n$$

Note that this scheme differs from the standard Forward Euler approach *only* by the term highlighted in blue, which in effect just replaced u_j^n with the average at neighboring grid points.

The spectrum of $\boldsymbol{A} + \widetilde{\boldsymbol{A}}_\epsilon$ is actually explicitly computable, allowing one to show that the eigenvalues actually lie on the boundary of an ellipse whose position + aspect ratio depend on ϵ.

In particular, one can take ϵ as large as possible so that the spectrum lies inside the Forward Euler region of stability. This corresponds to the choice,

$$\epsilon = \frac{h^2}{2k},$$

and results in the Lax-Friedrichs scheme. This scheme as we've derived it reads,

$$D^+ u_j^n = -aD_0 u_j^n + \underbrace{\frac{h^2}{2k}}_{\epsilon} D_+ D_- u_j^n.$$

However, it's more common to combine some terms and write this scheme as,

$$u_j^{n+1} = \frac{1}{2}\left(u_{j-1}^n + u_{j+1}^n\right) - kaD_0 u_j^n$$

Note that this scheme differs from the standard Forward Euler approach *only* by the term highlighted in blue, which in effect just replaced u_j^n with the average at neighboring grid points. A computation shows that the Lax-Friedrichs scheme is stable if $|ak/h| \leqslant 1$.

The Lax-Wendroff scheme

Whereas the Lax-Friedrichs uses as much dissipation as possible to ensure stability, an alternative approach is use to *just enough* dissipation, corresponding to $\epsilon = a^2 k/2$:

$$D^+ u_j^n = -aD_0 u_j^n + \underbrace{\frac{a^2 k}{2}}_{\epsilon} D_+ D_- u_j^n .$$

The Lax-Wendroff scheme

Whereas the Lax-Friedrichs uses as much dissipation as possible to ensure stability, an alternative approach is use to *just enough* dissipation, corresponding to $\epsilon = a^2 k/2$:

$$D^+ u_j^n = -aD_0 u_j^n + \underbrace{\frac{a^2 k}{2}}_{\epsilon} D_+ D_- u_j^n.$$

This is the Lax-Wendroff scheme.

Since it uses "just enough" dissipation, it is not too surprising that this is generally more accurate (say in the LTE sense) than Lax-Friedrichs.

The Lax-Wendroff scheme

Whereas the Lax-Friedrichs uses as much dissipation as possible to ensure stability, an alternative approach is use to *just enough* dissipation, corresponding to $\epsilon = a^2 k/2$:

$$D^+ u_j^n = -aD_0 u_j^n + \underbrace{\frac{a^2 k}{2}}_{\epsilon} D_+ D_- u_j^n.$$

This is the Lax-Wendroff scheme.

Since it uses "just enough" dissipation, it is not too surprising that this is generally more accurate (say in the LTE sense) than Lax-Friedrichs.

For stability of Lax-Wendroff, we again require $|ak/h| \leqslant 1$.

$$u_{xx} \sim \frac{1}{h^2}\left(u_{j+1} - 2u_j + u_{j-1}\right)$$

$$u_x \sim \frac{1}{2h}\left(u_{j+1} - u_{j-1}\right)$$

Even more alternatives

Another popular class of methods are upwind schemes.

We started by using D_0 to approximate $\frac{\partial}{\partial x}$, but why not use D_+ or D_-?

Even more alternatives

Another popular class of methods are upwind schemes.

We started by using D_0 to approximate $\frac{\partial}{\partial x}$, but why not use D_+ or D_-?

In this hyperbolic setting, these two options are not interchangable: the sign of the wavespeed a dictates a "natural" direction of flow:

$$u(x,t) = u_0(x - at).$$

Even more alternatives

Another popular class of methods are upwind schemes.

We started by using D_0 to approximate $\frac{\partial}{\partial x}$, but why not use D_+ or D_-?

In this hyperbolic setting, these two options are not interchangable: the sign of the wavespeed a dictates a "natural" direction of flow:

$$u(x,t) = u_0(x - at).$$

A qualitative argument for using one over the other is as follows:

- If $a > 0$:
  - The solution travels to the right.
  - Hence, to compute a derivative at spatial index j, using data from $j, j-1, j-2, \ldots$ seems consistent with the solution.
  - Use D_-.
- If $a < 0$:
  - The solution travels to the left.
  - Hence, to compute a derivative at spatial index j, using data from $j, j+1, j+2, \ldots$ seems consistent with the solution.
  - Use D_+.

Even more alternatives

Another popular class of methods are upwind schemes.

We started by using D_0 to approximate $\frac{\partial}{\partial x}$, but why not use D_+ or D_-?

In this hyperbolic setting, these two options are not interchangable: the sign of the wavespeed a dictates a "natural" direction of flow:

$$u(x,t) = u_0(x - at).$$

A qualitative argument for using one over the other is as follows:

- If $a > 0$:
  - The solution travels to the right.
  - Hence, to compute a derivative at spatial index j, using data from $j, j-1, j-2, \ldots$ seems consistent with the solution.
  - Use D_-.
- If $a < 0$:
  - The solution travels to the left.
  - Hence, to compute a derivative at spatial index j, using data from $j, j+1, j+2, \ldots$ seems consistent with the solution.
  - Use D_+.

This can be made formal with some stability analysis: the upwind scheme,

$$D^+ u_j^n = -a D_{\pm} u_j^n,$$

is stable iff $|ak/h| \leqslant 1$ and $\pm \operatorname{sign}(a) \leqslant 0$.

Stability

We have seen that the condition,

$$\left|\frac{ak}{h}\right| \leqslant 1,$$

is a necessary stability condition for discretizing $u_t + au_x = 0$ in the schemes we've considered.

Apart from the algebra of stability computations, is there some broader motivation for this condition?

Stability

We have seen that the condition,

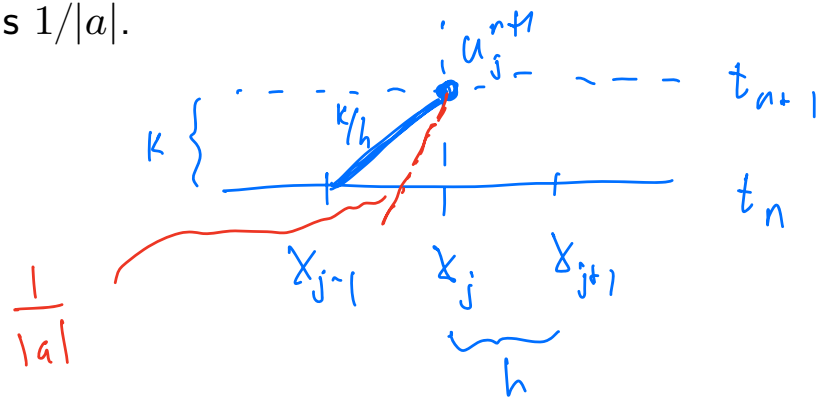
$$\left|\frac{ak}{h}\right| \leqslant 1,$$

is a necessary stability condition for discretizing $u_t + au_x = 0$ in the schemes we've considered.

Apart from the algebra of stability computations, is there some broader motivation for this condition?

In the (x,t) plane, k/h is the slope modulus of a line connecting $u^n_{j\pm 1}$ to u^{n+1}_j.

The *characteristics* of this problem have the form $X(t) = at + x_0$, i.e., in the (x,t) plane these lines have slope modulus $1/|a|$.



We have seen that the condition,

$$\left|\frac{ak}{h}\right| \leqslant 1,$$

is a necessary stability condition for discretizing $u_t + au_x = 0$ in the schemes we've considered.

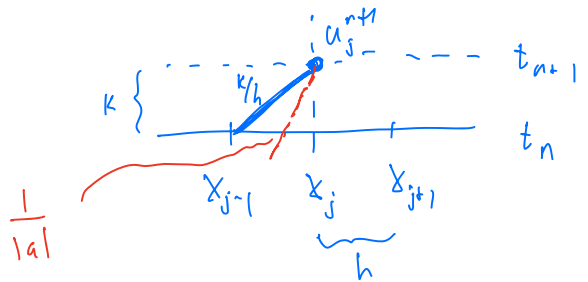
Apart from the algebra of stability computations, is there some broader motivation for this condition?

In the (x,t) plane, k/h is the slope modulus of a line connecting $u_{j\pm 1}^n$ to u_j^{n+1}.

The *characteristics* of this problem have the form $X(t) = at + x_0$, i.e., in the (x,t) plane these lines have slope modulus $1/|a|$.

Thus, the condition $|ak/h| \leqslant 1$, rewritten, says,

$$\frac{k}{h} \leqslant \frac{1}{|a|}.$$



Under this condition, the characteristics connecting (x_j, t_{n+1}) to time level t_n lie *between* the grid points x_{j-1} and x_{j+1}.

$$\frac{k}{h} \leqslant \frac{1}{|a|}.$$

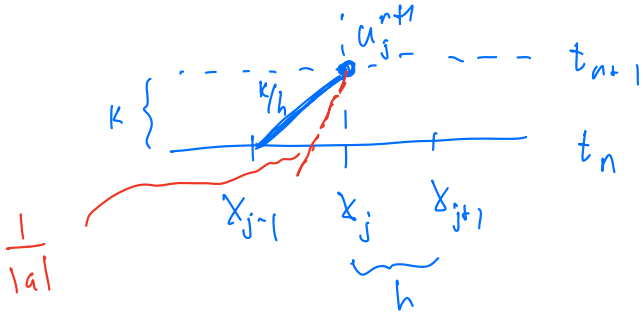
Recall that $D(x,t)$ is the *domain of dependence* for the exact solution.

If we simply consider domains of dependence from time t_{n+1} back to time t_n, then, the above equation can be interpreted as,

(For D_j's): The interval $[x_{j-1}, x_{j+1}]$ contains $D(x_j, t_{n+1})$ for time t_n

Domains of dependence

$$\frac{k}{h} \leqslant \frac{1}{|a|}.$$



Recall that $D(x,t)$ is the *domain of dependence* for the exact solution.

If we simply consider domains of dependence from time t_{n+1} back to time t_n, then, the above equation can be interpreted as,

$$\text{The interval } [x_{j-1}, x_{j+1}] \underline{\text{ contains }} D(x_j, t_{n+1}) \text{ for time } t_n$$

For many of the (single-step) schemes we've considered, $[x_{j-1}, x_{j+1}]$ is the numerical domain of dependence of the scheme, i.e.,

$$\widetilde{D}(x_j, t_{n+1}) = I,$$

For D, u_j^n : $I = [x_{j-1}, x_{j+1}]$

where I is the smallest closed interval containing the time t_n stencil for the scheme.

$$\frac{k}{h} \leqslant \frac{1}{|a|}.$$

Recall that $D(x,t)$ is the *domain of dependence* for the exact solution.

If we simply consider domains of dependence from time t_{n+1} back to time t_n, then, the above equation can be interpreted as,

$$\text{The interval } [x_{j-1}, x_{j+1}] \text{ contains } D(x_j, t_{n+1}) \text{ for time } t_n$$

For many of the (single-step) schemes we've considered, $[x_{j-1}, x_{j+1}]$ is the numerical domain of dependence of the scheme, i.e.,

$$\widetilde{D}(x_j, t_{n+1}) = I,$$

where I is the smallest closed interval containing the time t_n stencil for the scheme.

Then $k/h \leqslant 1/|a|$, reinterpreted again, states that,

$$\widetilde{D}(x_j, t_{n+1}) \supseteq D(x_j, t_{n+1})$$

$$\text{The numerical domain of dependence contains the analytical domain of dependence}$$

This condition requires the numerical scheme to use data that can encode the exact solution.

The CFL condition

The general condition that

> The numerical domain of dependence contains the analytical domain of dependence

is called the Courant-Friedrichs-Lewy (CFL) condition.

The CFL condition

The general condition that

> The numerical domain of dependence contains the analytical domain of dependence

is called the Courant-Friedrichs-Lewy (CFL) condition.

This condition is actually a rigorous requirement: A necessary condition for convergence of a numerical scheme is that as $k, h \downarrow 0$, the numerical domain of dependence contains the (analytical) domain of dependence.

The CFL condition

$u_t = \frac{\partial^p u}{\partial x^p} \Rightarrow k \sim h^p$

The general condition that

> The numerical domain of dependence contains the analytical domain of dependence

is called the Courant-Friedrichs-Lewy (CFL) condition.

This condition is actually a rigorous requirement: A necessary condition for convergence of a numerical scheme is that as $k, h \downarrow 0$, the numerical domain of dependence contains the (analytical) domain of dependence.

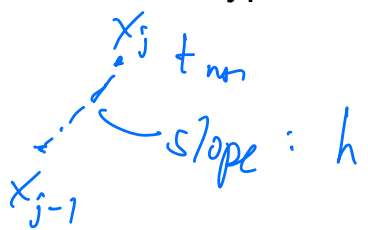
For wave-like problems with wavespeed a, this condition is essentially always of the form,

$$k \leqslant |a| h \qquad \frac{h}{|a|}$$

The CFL condition provides an easy (+ transparent) rule-of-thumb for time-step restrictions in hyperbolic problems.

claim: $k \ll h^2$ is the CFL condition for $u_t = u_{xx}$.

$k, h \to 0$



 Courant, R., K. Friedrichs, and H. Lewy (1967). "On the Partial Difference Equations of Mathematical Physics". In: *IBM Journal of Research and Development* 11.2, pp. 215–234. ISSN: 0018-8646. DOI: 10.1147/rd.112.0215.

 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.