Lecture 30: Stability of Time Interation

We seek to understand the apparent time step restriction for the Forward Euler method for the heat equation.

First consider a generic scalar differential equation

$$y'(t) = \lambda y(t)$$

for which the forward Euler method takes the form

$$y_{k+1} = y_k + \Delta t \lambda y_k = (1+\Delta t\lambda) y_k$$

So that $\boxed{y_k = (1+\Delta t\lambda)^k y_0}$

Now the exact solution of the differential equation is $y(t) = e^{\lambda t} y(0)$, which converges, $y(t) \to 0$, provided $Re(\lambda) < 0$, i.e., if $\lambda = \alpha + i\beta$, then

$$e^{\lambda t} = e^{(\alpha+i\beta)t} = e^{\alpha t} e^{i\beta t}$$

$$= e^{\alpha t}(\cos\beta t + i \sin\beta t)$$

So $|e^{\lambda t}| = |e^{\alpha t}(\cos\beta t + i\sin\beta t)|$

$$= e^{\alpha t} |\cos\beta t + i\sin\beta t|$$

$$= e^{\alpha t} \underbrace{\sqrt{\cos^2\beta t + \sin^2\beta t}}_{=1} = e^{\alpha t}.$$

So $e^{\lambda t} \to 0$ if $Re(\lambda) = \alpha < 0$.

Does Forward Euler mimic this behavior?

$$y_k = (1 + \Delta t \lambda)^k y_0$$

Implies that $y_k \to 0$ provided $|1 + \Delta t \lambda| < 1$.

Note that this is a rather different condition than $Re(\lambda) < 0$!

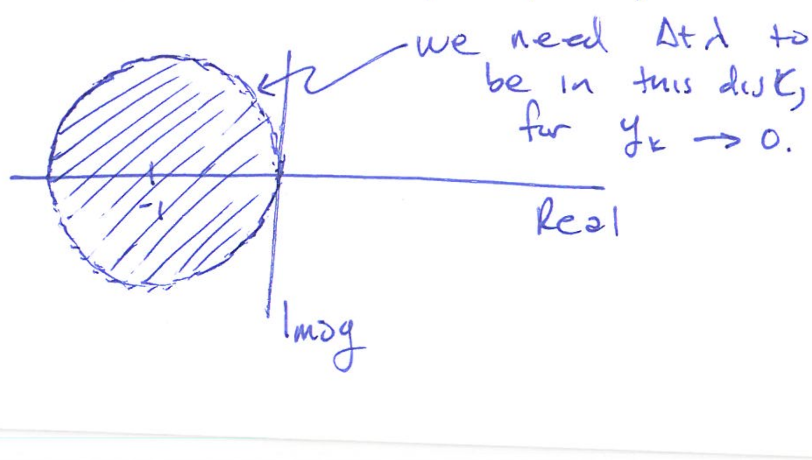
Consider real values of λ, as we have for eigenvalues of the heat discretization. If λ is negative but large in magnitude, we will need Δt to be very small for $|1 + \Delta t \lambda| < 1$.

Here is a helpful interpretation of this condition:

$\{z \in \mathbb{C} : |c - z| < 1\}$ = set of all complex numbers z for which $|c - z| < 1$

= set of all complex numbers z whose distance from c is less than one.

Thus $|1 + \Delta t \lambda| = |-1 - \Delta t \lambda| < 1$ is satisfied if $\Delta t \lambda$ is in the disc of radius 1 in the complex plane, centered at -1.



Now let us interpret this observation in light of the eigenvalue calculations in the last lecture.

The heat equation / Finite Element method gave

$$a'(t) = -M^{-1}K\,a(t)$$

So Forward Euler takes the form

$$a_{j+1} = a_j - \Delta t\, M^{-1}K a_j \Rightarrow \quad a_j = (I - \Delta t M^{-1}K)^j a_0$$

where $a_j \approx a(j\cdot\Delta t)$.

In the last lecture we wrote

$$M^{-1}K = V\Lambda V^T = \sum_{k=1}^{N} \lambda_k v_k v_k^T$$

where $\lambda_k = \dfrac{6}{h^2}\left(\dfrac{2-\gamma_k}{4+\gamma_k}\right)$, $\quad \gamma_k = 2\cos(k\pi/N+1)$,

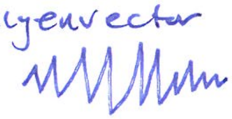
$$h = \frac{1}{N+1}.$$

Forward Euler then becomes

$$\begin{aligned}
a_j &= (I - \Delta t M^{-1}K)^j a_0 \\
&= (I - \Delta t V\Lambda V^T)^j a_0 \\
&= (VV^T - \Delta t V\Lambda V^T)^j a_0 \\
&= V(I - \Delta t \Lambda)^j V^T a_0 \\
&= V \begin{pmatrix} (1-\Delta t\lambda_1)^j & & 0 \\ & \ddots & \\ 0 & & (1-\Delta t\lambda_N)^j \end{pmatrix} V^T a_0 \\
&= \sum_{k=1}^{N} (1-\Delta t\lambda_k)^j v_k v_k^T a_0
\end{aligned}$$

Note:
$VV^T = I$
$V^TV = I$

$\star\star$ The component of this sum that grows fastest is the one for which $|1-\Delta t\lambda_k|$ is largest. This corresponds to λ_N when Δt violates the stability condition $\Rightarrow$ the unstable solution will be dominated by v_N, the "sawtooth" eigenvector



So if we want $a_j \to 0$ as $j \to \infty$,
we need $|1 - \Delta t \lambda_k| < 1$ for <u>all</u> $k = 1, \dots, N$.

Since $\lambda_k \in \left(0, \frac{12}{h^2}\right)$ for all $k = 1, \dots, N$,
we should pick Δt sufficiently small that

$$\left|1 - \underbrace{\Delta t \frac{12}{h^2}}_{>0}\right| < 1$$

Since $\Delta t \frac{12}{h^2} > 0$, for any Δt, $1 - \Delta t \frac{12}{h^2} < 1$.
Hence, Δt will be determined by the condition

$$-1 < 1 - \Delta t \frac{12}{h^2}$$

$$\Rightarrow \Delta t \frac{12}{h^2} < 2 \Rightarrow \boxed{\Delta t < \frac{h^2}{6}}$$

Notice the profound implication of this calculation:

\# If you want to double N (to get better accuracy in space, you reduce h by $\frac{1}{2}$, and you must then reduce <u>Δt by $\frac{1}{4}$</u>.

This severe stability constraint is called the $\boxed{\text{CFL condition}}$ after its discoverers: Richard Courant, Kurt Friedrichs, and Hans Lewy.

EACH DIFFERENT TIME INTEGRATOR LEADS TO A DIFFERENT STABILITY CONSTRAINT.

AS AN EXAMPLE, CONSIDER THE BACKWARD EULER METHOD. For the generic equation $y'(t) = \lambda y(t)$, note that

$$y'(t) = \lim_{\Delta t \to 0} \frac{y(t+\Delta t) - y(t)}{\Delta t}$$

also suggests that

$$0 = \lim_{\Delta t \to 0} y'(t+\Delta t) - \left(\frac{y(t+\Delta t) - y(t)}{\Delta t} \right)$$

Hence, for finite Δt, approximate

$$y(t+\Delta t) \simeq y(t) + \Delta t \, y'(t+\Delta t)$$

Writing $y_j \simeq y(j\Delta t)$, this gives

$$y_{j+1} = y_j + \Delta t \lambda y_{j+1} \qquad (y' = \lambda y)$$

$$\Rightarrow \quad (1 - \Delta t \lambda) y_{j+1} = y_j$$

$$\Rightarrow \quad \boxed{y_{j+1} = \frac{1}{1-\Delta t \lambda} y_j}$$

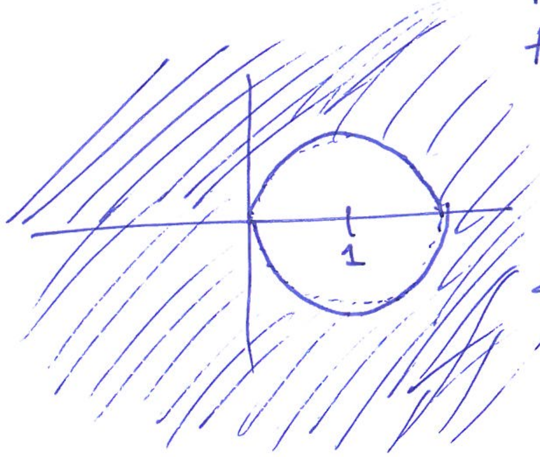
and so we can accumulate j steps as

$$\boxed{y_j = \left(\frac{1}{1-\Delta t \lambda} \right)^j y_0}$$

Now if $Re(\lambda) < 0$, $y(t) \to 0$. Is the same true for backward Euler?

$y_j \to 0$ if $\left| \frac{1}{1-\Delta t \lambda} \right| < 1$, which implies

$1 < |1 - \Delta t \lambda|$ = set of all $\Delta t \lambda$ that are a distance greater than 1 from 1 in the complex plane



So $y_j \to 0$ provided $\Delta t \lambda$ is in the shaded region....

If $Re(\lambda) < 0$, then for any Δt, $Re(\Delta t \lambda) < 0$, and so $\Delta t \lambda$ is in the shaded region.

$\Rightarrow$ BACKWARD EULER IS STABLE FOR ANY Δt.

(You will still adjust Δt to give an accurate solution, i.e., to make the approximation
$$y' \approx \frac{y(t+\Delta t) - y(t)}{\Delta t}$$
accurate.)

For the heat equation, $a'(t) = -M^{-1} K a(t)$, Backward Euler gives

$$a_j = \left((I + \Delta t M^{-1} K)^{-1} \right)^j a_0 = \sum_{k=1}^{N} \Big(\underbrace{\frac{1}{1 + \Delta t \lambda_k}}_{\lambda_k \in \left(0, \frac{12}{h^2}\right)} \Big)^j v_k v_k^T a_0$$

Since $\lambda_k > 0$, $\left| \frac{1}{1 + \Delta t \lambda_k} \right| < 1$ for all Δt.

Hence Backward Euler is stable for the heat equation for any $\Delta t > 0$.

<u>CODA</u>: A WORD ABOUT TIME-STEPPING IMPLEMENTATIONS (See Lecture 35 for related detail.)

FORWARD EULER:

$$a_{j+1} = a_j - \Delta t\, M^{-1} K a_j$$

BACKWARD EULER:

$$\begin{aligned} a_{j+1} &= (I + \Delta t M^{-1} K)^{-1} a_j \\ &= (M^{-1}M + \Delta t M^{-1} K)^{-1} a_j \\ &= (M^{-1}(M + \Delta t K))^{-1} a_j \\ &= (M + \Delta t K)^{-1} {M^{-1}}^{-1} a_j \\ &= (M + \Delta t K)^{-1} M a_j \end{aligned}$$

Both equations involve the inverse of one matrix, which can be implemented via LU factorization ⇒ Similar work per step!

For more complicated equations (especially nonlinear equations) Backward Euler is rather more expensive per step than Forward Euler.