UW–Madison Math/CS 714

Methods of Computational Mathematics I

Initial value problems III

Instructor: Yue Sun (yue.sun@wisc.edu)

October 7, 2025

Stability

To examine stability of a method, we focus on the test problem

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

with $u(0) = 1$. The behavior of a discretization on this test problem provides a lot of insight into behavior in general.

Solutions will be $u(t) = e^{\lambda t}$. They converges for the region where $\mathrm{Re}(\lambda) \leq 0$, and we would like our numerical method to do the same.

Stability for the Euler method

Applying the forward Euler method $U^{n+1} = U^n + k\lambda U^n$ gives

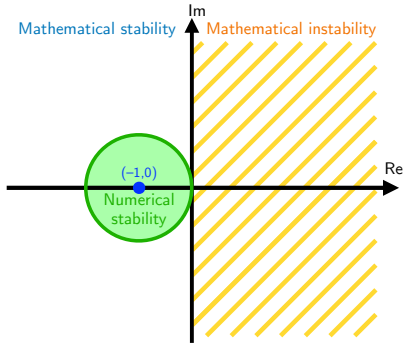
$$U^n = (1 + k\lambda)^n$$

We say that the method is absolutely stable when $|1 + k\lambda| \leq 1$, because the solution will not diverge.

Only the product $z = k\lambda$ matters, and we see that on the real line, the interval of absolute stability where $|1 + z| \leq 1$ corresponds to $z \in [-2, 0]$

Stability for the Euler method

More generally the region of absolute stability in the complex plane is a disc of radius 1, centered on $z = -1$.



As a result we say that the forward Euler method is conditionally stable: when $\mathrm{Re}(\lambda) \leq 0$ we have to restrict k to ensure stability.

Timestep restriction

In order for the Euler method to be stable, we require $-2 \leq z \leq 0$, which is $-2 \leq k\lambda \leq 0$. Consider several different values of λ:

- $\lambda = -1 \quad \Longrightarrow \quad k \leq \frac{2}{-\lambda} = 2$
- $\lambda = -10 \quad \Longrightarrow \quad k \leq \frac{2}{-\lambda} = \frac{1}{5}$
- $\lambda = -1000 \quad \Longrightarrow \quad k \leq \frac{2}{-\lambda} = \frac{1}{500}$

For equations where the solution decays on more rapid timescale, the corresponding timestep must be chosen to be smaller in order to achieve stability.

Stability regions for multistep methods

Now consider applying a linear multistep method to the equation $u' = \lambda u$. Then

$$\sum_{j=0}^{r} \alpha_j U^{n+j} = k \sum_{j=0}^{r} \beta_j \lambda U^{n+j}$$

which becomes

$$\sum_{j=0}^{r} (\alpha_j - z\beta_j) U^{n+j}$$

where $z = k\lambda$. This is a linear difference equation, and we already know the general form of the solution.

Stability regions for multistep methods

Introduce the stability polynomial

$$\pi(\zeta; z) = \rho(\zeta) - z\sigma(\zeta)$$

where ρ and σ are the characteristic polynomials introduced previously. Let the roots, be $\zeta_1, \ldots, \zeta_l$ where $l \leq r$, and let m_j be the multiplicity of ζ_j. As before, the general solution will be

$$U^n = \sum_{j=1}^{l} p_j(n)\zeta_j^n$$

where p_j is a polynomial of degree $m_j - 1$.

Stability regions for multistep methods

For a given z, we want ensure that the numerical solution will not diverge.

We have already seen how to determine this in the case of zero stability, by applying the root condition to $\rho(\zeta) = \pi(\zeta; 0)$.

We now generalize this: the region of absolute stability for a linear multistep method is the set of points $z \in \mathbb{C}$ where $\pi(\zeta; z)$ satisfies the root condition.

Examples

For the forward Euler method

$$\pi(\zeta; z) = \zeta - (1 + z)$$

which has a single root at $\zeta_1 = 1 + z$. The region of absolute stability is the disk $|1 + z| \leq 1$ that we found previously.

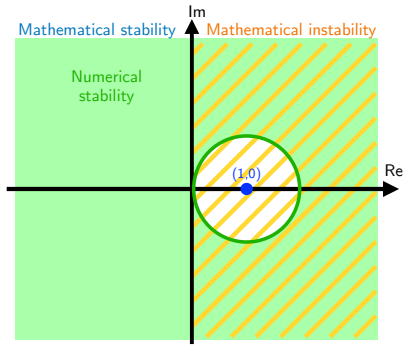
For the backward Euler method

$$\pi(\zeta; z) = (1 - z)\zeta - 1$$

which has a single root at $\zeta_1 = (1 - z)^{-1}$. The region of absolute stability is given by $|1 - z| \geq 1$.

Stability of backward Euler

In this case the region of stability contains the entire left half plane



As a result we say that the backward Euler method is unconditionally stable or A-stable, and there is no restriction on k for stability.

Example: the trapezoid method

Recall that the trapezoid method is $U^{n+1} - U^n = \frac{k}{2}(f(U^{n+1}) + f(U^n))$. The stability polynomial is

$$\pi(\zeta; z) = (1 - \tfrac{z}{2})\zeta - (1 + \tfrac{z}{2})$$

so there is a single root at

$$\zeta_1 = \frac{2 + z}{2 - z}$$

The condition $|\zeta_1| \leq 1$ is satisfied when $\mathrm{Re}(z) \leq 0$. In this case the stability region contains exactly the left half plane and the method is unconditionally stable.

Stability calculations for other methods

The program *stab_region.py* computes the stability region for several multi-step methods in the BDF (backward differentiation formula) family:

- $11U^{n+3} - 18U^{n+2} + 9U^{n+1} - 2U^n = 6kf(U^{n+3})$
- $25U^{n+4} - 48U^{n+3} + 36U^{n+2} - 16U^{n+1} + 3U^n = 12kf(U^{n+4})$
- $137U^{n+5} - 300U^{n+4} + 300U^{n+3} - 200U^{n+2} + 75U^{n+1} - 12U^n = 60kf(U^{n+5})$

For each position z in the complex plane, the program computes the maximum modulus of a root of $\pi(\zeta; z)$.

Boundary locus method

A point $z \in \mathbb{C}$ is in the stability region if $\pi(\zeta; z)$ satisfies the root condition for this value of z. If a point z is on the boundary of the stability region, then $\pi(\zeta; z)$ must have at least one root ζ_j where $|\zeta_j| = 1$, *i.e.*

$$\zeta_j = e^{i\theta}$$

for some real θ. Then $\pi(e^{i\theta}; z) = 0$, so

$$\rho(e^{i\theta}) - z\sigma(e^{i\theta}) = 0$$

and therefore

$$z(\theta) = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})}$$

Tracing out all points like this gives the locus of all points *potentially* on the boundary of the stability region.

Example: boundary locus method

For the forward Euler method

$$\rho(\zeta) = \zeta - 1, \qquad \sigma(\zeta) = 1$$

Hence

$$z(\theta) = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})} = e^{i\theta} - 1.$$

This will trace out the unit disk centered on $z = -1$, which matches our previous analyses. To determine which side is stable, we can evaluated the root condition at a single point.

ODE Stability

Our understanding of the stability of $u' = \lambda u$ extends directly to the case $u' = Au$, where $u \in \mathbb{R}^s, A \in \mathbb{R}^{s \times s}$

Suppose that A is diagonalizable, so that we have the eigenvalue decomposition $A = V \Lambda V^{-1}$, where

- $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_s)$, where the λ_j are eigenvalues
- V is matrix with eigenvectors as columns, $v_1, v_2, \ldots, v_s$

Then,

$$u' = Au = V \Lambda V^{-1} u \implies V^{-1} u' = \Lambda V^{-1} u \implies z' = \Lambda z$$

where $z \equiv V^{-1} u$ and $z_0 \equiv V^{-1} u_0$

ODE Stability

Hence we have s decoupled ODEs for z, and stability of z_i is determined by λ_i for each i

Since z and u are related by the matrix V, then (roughly speaking) if all z_i are stable then all u_i will also be stable

Hence we have $\text{Re}(\lambda_i) \leq 0$ for $i = 1, \ldots, s \implies u' = Au$ is a stable ODE

Stiff systems

You may have heard of "stiffness" in the context of ODEs. This is an important concept although not one with a general mathematical definition.

One definition of stiffness for a linear ODE system $y' = Ay$ is that A has eigenvalues that differ greatly in magnitude.

The eigenvalues determine the time scales, and hence large differences in λ's $\implies$ resolve disparate timescales simultaneously.

Stiff systems

Suppose we're primarily interested in the long timescale. Then:

- We'd like to take large time steps and resolve the long timescale accurately
- But we may be forced to take extremely small timesteps to avoid instabilities due to the fast timescale

In this context it can be highly beneficial to use an implicit method since that enforces stability regardless of timestep size

Stiff systems

From a practical point of view, an ODE is stiff if there is a significant benefit in using an implicit instead of explicit method

e.g. this occurs if the time-step size required for stability is much smaller than size required for the accuracy level we want

Example [*stiff.py*/*stiff2.py*] : Consider $u' = Au$, $u_0 = (1, 0)$ where

$$A = \begin{bmatrix} 998 & 1998 \\ -999 & -1999 \end{bmatrix}$$

which has $\lambda_1 = -1$, $\lambda_2 = -1000$ and exact solution

$$u(t) = \begin{bmatrix} 2e^{-t} - e^{-1000t} \\ -e^{-t} + e^{-1000t} \end{bmatrix}$$

Different stability definitions

The previous example demonstrates the benefits of using the backward Euler method, which we showed previously was unconditionally stable or A-stable, *i.e.* the region of absolute stability contains the entire left half plane, $\mathrm{Re}(z) \leq 0$.

For linear multistep methods, A-stability is restrictive. Dahlquist's second barrier theorem states that any A-stable linear multistep methods is at most second-order accurate.

Furthermore, the trapezoidal rule is the A-stable method with smallest truncation error.

A(α)-stability

For many stiff problems, the eigenvalues are large and negative, but do not have large imaginary components.

Hence it is useful to introduce A(α)-stability, when a method that is stable for all $z \in \mathbb{C}$ in the wedge such that $|\pi - \arg(z)| \leq \alpha$.[1]

A method is A(0)-stable if it is stable for the negative real axis.

[1] Here $\arg(z)$ is measured over the range $[0, 2\pi)$.

L-stability

Even though backward Euler and the trapezoid method are A-stable, the way that they handle large negative eigenvalues is different. For $u' = \lambda u$, and $z = k\lambda$, then $U^{n+1} = R(z)U^n$.

For backward Euler,

$$R(z) = \frac{1}{1-z} \qquad \Longrightarrow \ |R(z)| \to 0 \text{ as } z \to \infty.$$

For the trapezoid method,

$$R(z) = \frac{2+z}{2-z} \qquad \Longrightarrow \ |R(z)| \to 1 \text{ as } z \to \infty.$$

L-stability

It is preferable to have rapid decay of terms with large negative λ. See *stiff3.py*, which demonstrates this for the trapezoid rule applied to the example stiff ODE.

Even though the trapezoid rule gives stable results, there are large transient oscillations from the $\lambda_2 = -2000$ eigenvalue. This motivates an additional definition.

A method is defined to be L-stable if it is A-stable and the stability function satisfies $\lim_{z\to\infty} |R(z)| = 0$.

The backward Euler method is L-stable.