Chapter 7
Eigenvalue Stability

As we have seen, while numerical methods can be convergent, they can still exhibit instabilities as the number of timesteps n increases for finite Δt. For example, when applying the midpoint method to the ice particle problem, instabilities were seen as n increased. The key to understanding these results is to analyze the stability for finite Δt.

25 Self-Assessment

Before reading this chapter, you may wish to review...

- zero stability
- eigenvalues
- linearization
- Jacobian

After reading this chapter you should be able to...

- determine the amplification factor for a given multi-step method
- plot the eigenvalue stability region for any method
- determine the maximum allowable time step for a method and problem to maintain eigenvalue stability
- proficiently utilize the eigenvalue stability applet
- describe the representation of a solution as an expansion in eigenspace

26 Eigenvalue Stability for a Linear ODE

Eigenvalue stability analysis differs from our previous analysis tools in that we will *not* consider the limit $\Delta t \to 0$. Instead, we will assume that Δt is a finite number. This is important because when we implement numerical methods, we can never achieve the limit $\Delta t \to 0$; in the end, we must fix some (small) positive number.

Exercise 1. What is the behavior associated with eigenvalues with nonzero imaginary part?

(a) decay
(b) growth
(c) oscillations
(d) none of the above

Suppose we are interested in solving the linear ODE,

$$u_t = \lambda u. \tag{51}$$

Consider the Forward Euler method applied to this problem,

$$v^{n+1} = v^n + \lambda \Delta t v^n. \tag{52}$$

Similar to the zero stability analysis, we will assume that the solution has the following form,

$$v^n = g^n v^0, \tag{53}$$

where g is the amplification factor (and the superscript n acting on g is again raising to a power).

Exercise 2. Under what conditions on g will v^n grow unbounded as $n \to \infty$?

(a) $|g| < 1$
(b) $|g| > 1$
(c) $|g| < 0$
(d) $|g| > 0$

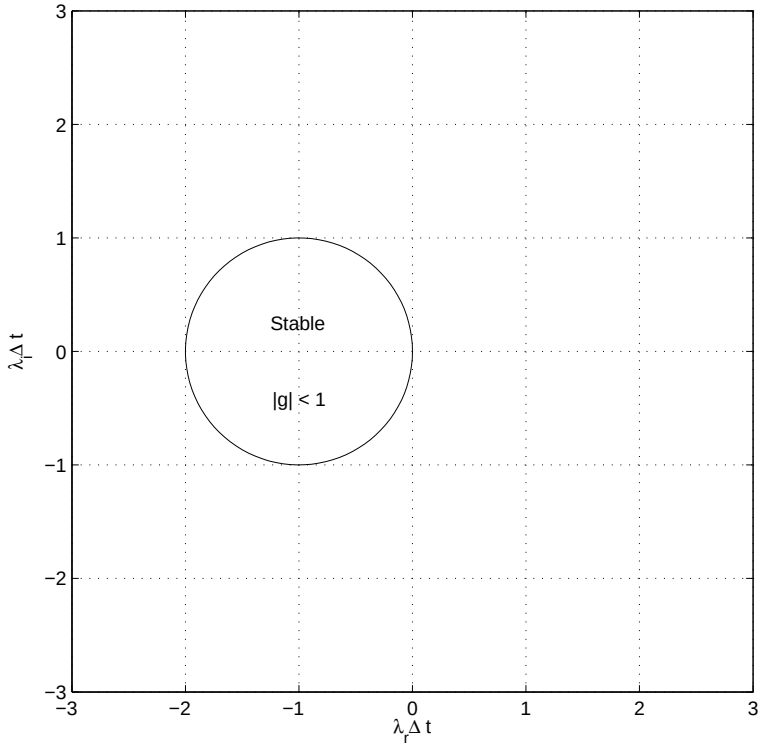

Fig. 5 Forward Euler stability region

Substituting Equation 53 into Equation 52 gives,

$$g^{n+1} = (1 + \lambda \Delta t) g^n. \tag{54}$$

Thus, the only non-zero root of this equation gives,

$$g = 1 + \lambda \Delta t, \tag{55}$$

which is the amplification factor for the forward Euler method.

Exercise 3. Derive the amplification factor $g(\lambda \Delta t)$ for the midpoint method

$$v^{n+1} = v^{n-1} + 2\Delta t f(v^n, t^n). \tag{56}$$

Submit your evaluation of the amplification factor for $g(i)$.

Now, we must determine what values of $\lambda \Delta t$ lead to instability (or stability). A simple way to do this for multi-step methods is to solve for the stability boundary for which $|g| = 1$. To do this, let $g = e^{i\theta}$ (since $|e^{i\theta}| = 1$) where $\theta = [0, 2\pi]$. Making this substitution into the amplification factor (55) for forward Euler,

$$e^{i\theta} = 1 + \lambda \Delta t \quad \Rightarrow \quad \lambda \Delta t = e^{i\theta} - 1. \tag{57}$$

Thus, the stability boundary for the forward Euler method lies on a circle of radius one centered at -1 along the real axis and is shown in Figure 5. We determine that the region *inside* the circle is stable by testing a point in the interior. For example, take $\lambda \Delta t = -1.5$; then $|g| = |1 + \lambda \Delta t| = |1 - 1.5| = 0.5 < 1$.

Thought Experiment. Determine the eigenvalue stability region for the midpoint method.

For a given problem, i.e. with a given λ, the timestep must be chosen so that the algorithm remains stable for $n \to \infty$. Let's consider an example.

Example 4.2 *Falling Particle*
Let's return to the previous example,

$$u_t = f(u) = -u^2 \tag{58}$$

with initial condition $u(0) = 1$. To determine the timestep restrictions, we must estimate the eigenvalue for this problem. As described before, linearizing this problem about a known state gives the eigenvalue as

$$\lambda(u) = \frac{\partial f}{\partial u} = -2u. \tag{59}$$

Since the forcing function $f(u)$ is nonlinear, the eigenvalue depends on the position u.

We will need to choose a time step for the simulation to maintain eigenvalue stability for the entire simulation period. Therefore, we must reason the possible velocities of the particle (i.e., an interval for u in this case) in order to find the range of eigenvalues that will occur. Since the solution will decay from the initial condition because $u_t = -u^2 < 0$, we can reason that the velocity u will take on values in the interval $[0, 1]$. The particle will start with initial velocity $u(0) = 1$ and slow down due to the drag until it finally comes to rest where $u = 0$.

It follows that the eigenvalue $\lambda(u) = -2u$ will take on values in the interval $[-2, 0]$. The largest magnitude of the eigenvalue occurs at the initial condition when $u(0) = 1$ where $\lambda(u) = -2$. Since this eigenvalue is a negative real number, the maximum allowable Δt to maintain eigenvalue stability will occur at the maximum extent of the stability region along the negative real axis. This occurs when $\lambda \Delta t = -2$, which implies that we require $\Delta t < 1$ for eigenvalue stability.

To test the validity of this analysis, you will experiment with the numerical integration of the falling particle.

Exercise 4. In this exercise you will modify a generic forward Euler code and test our eigenvalue stability analysis.

(a) Download the Matlab code

```
N = 10;                              % 10 forward Euler steps
dt = 0.1;                            % time step size
u(1) = 1;                            % initial condition
for n=1:N                            % loop over forward Euler steps
    dudt = -2 * u(n)                 % compute time derivative
    u(n+1) = u(n) + dt * dudt        % forward Euler step
end
```

.

(b) Modify the code to specify $f(u)$ for the falling particle problem.
(c) From the MATLAB command line, run the forward Euler integrator for a specified time step `dt` and plot the results.
(d) Change `dt` and rerun the code. At what value of `dt` does the simulation go unstable?

Eigenvalue stability applet video tutorial

Experiment with the Eigenvalue Stability Mathlet.

(a) Select the trapezoidal scheme from the drop down menu
(b) Derive the amplification factor g for that scheme
(c) Verify your findings by clicking the radio button *Formula*
(d) Let $g = e^{i\theta}$ and determine z for $\theta = 0, \pi/3, \pi/2, \pi, 3\pi/2$ to sketch the stability region
(e) Drag the θ sliding bar on the upper right of the mathlet to trace out the stability region in the z-plane
(f) In the z-plane in the upper left, click within the blue shaded region in the left-half plane
(g) Then, click anywhere in the $|\lambda|$ vs. Δt window in the lower right
(h) Using the Δt slider, determine the range of time steps for which the solution is eigenvalue stable
(i) Choose a different scheme from the drop down menu and repeat steps (d)–(h)

In the next section, we will show that a linear system of ODEs (e.g. arising from the linearization of a nonlinear system) can be decoupled by a change of coordinates. In the transformed space, the principles from our eigenvalue stability analysis of scalar ODEs above can be applied to analyze the eigenvalue stability of the original system.

27 Eigenvalue Stability for a Linear System of ODEs

In order to analyze the eigenvalue stability in the context of simulating a nonlinear system of ODEs by a given numerical scheme, we must first linearize the system as described above. Let $\mathbf{A}$ be a d-by-d matrix arising from linearization of a nonlinear system, and consider the resulting first-order linear system of ODEs

$$\mathbf{u}_t = \mathbf{A}\mathbf{u}. \tag{60}$$

In general, the matrix $\mathbf{A}$ will not be diagonal and therefore will couple components of $\mathbf{u}$; e.g., du_1/dt may depend on u_1, u_3, and u_{19}. In what follows, we will make a change of coordinates to decouple the system (60). We can then apply the eigenvalue stability analysis for scalar ODEs to each component individually. If eigenvalue stability is established for each component individually, we can conclude that the original (untransformed) system will also be eigenvalue stable.

Recall from 18.03 that an eigenvalue λ and eigenvector $\mathbf{r}$ of the matrix $\mathbf{A}$ satisfy the equation $\mathbf{A}\mathbf{r} = \lambda\mathbf{r}$. If the matrix $\mathbf{A}$ has a complete set of eigenvectors (aka is diagonalizable), then we can write

$$\mathbf{A}\begin{bmatrix} | & | & & | \\ \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_d \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_d \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d \end{bmatrix} \tag{61}$$

$$\mathbf{A} \qquad \mathbf{R} \qquad = \qquad \mathbf{R} \qquad \Lambda \tag{62}$$

where we have introduced the square matrices $\mathbf{R}$ and Λ.

If we multiply on the right by $\mathbf{R}^{-1}$, we obtain

$$\mathbf{A} = \mathbf{A}\mathbf{R}\mathbf{R}^{-1} = \mathbf{R}\Lambda\mathbf{R}^{-1}. \tag{63}$$

This is the eigendecomposition of the matrix $\mathbf{A}$.

The solution to (60) can be derived then as follows

$$\mathbf{u}_t = \mathbf{A}\mathbf{u} = \mathbf{R}\Lambda\mathbf{R}^{-1}\mathbf{u} \quad \Rightarrow \quad \mathbf{R}^{-1}\mathbf{u}_t = \Lambda\mathbf{R}^{-1}\mathbf{u}. \tag{64}$$

Now make the change of coordinates $\mathbf{w} = \mathbf{R}^{-1}\mathbf{u}$ to get

$$\mathbf{w}_t = \Lambda\mathbf{w}. \tag{65}$$

Since Λ is a diagonal matrix, this system of equations is actually decoupled. That is,

$$\frac{dw_j}{dt} = \lambda_j w_j, \qquad j = 1,2,\ldots,d. \tag{66}$$

We know the solution to this scalar ODE; it is $w_j(t) = w_j(0)e^{\lambda_j t}$. Thus, if we solve for each w_j, we can reconstruct the solution to our original problem. We change back to $\mathbf{u}$ coordinates by $\mathbf{u} = \mathbf{R}\mathbf{w}$,

$$\mathbf{u}(t) = \begin{bmatrix} | & | & & | \\ \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_d \\ | & | & & | \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \\ \vdots \\ w_d(t) \end{bmatrix} = \sum_{j=1}^{d} w_j(0)e^{\lambda_j t}\mathbf{r}_j. \tag{67}$$

For the numerical integration of a linear system of ODEs to be eigenvalue stable, all of the scaled eigenvalues $\lambda_j \Delta t$ for $j = 1,2,\ldots,d$ must fall within the stability boundary of the proposed numerical scheme. If they do not, one or more of the coefficients $w_j(0)e^{\lambda_j t}$ in the expansion (67) will blow up, and so then will $\mathbf{u}$ itself.

Thought Experiment The Jacobian of a nonlinear system will change depending on the point $\mathbf{u}^*$ about which linearization is performed. Therefore, the associated eigenvalues will change as the state $\mathbf{u}$ changes. How can you choose a timestep Δt for the simulation to be sure eigenvalue stability will be maintained?