

# Mat1062: Introductory Numerical Methods for PDE

Mary Pugh

January 20, 2009

## 1 Ownership

These notes are the joint property of Rob Almgren and Mary Pugh.

## 2 Stability

The main idea is to look at how the solutions of the discrete scheme behave as  $n \rightarrow \infty$  and  $k \rightarrow 0$ , with  $nk = t$ , *without reference to the true solution*. For a problem which is linear, or locally linear (as most are), this is the same as asking whether small changes in the initial data give rise to small changes in the solution.

**Example:** Consider the difference scheme

$$u^{n+1} = 2u^n - u^{n-1}.$$

The local truncation error is

$$u^{n+1} - (2u^n - u^{n-1}) \approx u_{tt}k^2, \quad k \rightarrow 0.$$

This suggests that the scheme is first-order accurate for *any* problem. But the solution  $u^n$  cannot possibly converge to the solution of  $u_t = f$ , because no knowledge of  $f$  was used in its construction! The reason is that, since  $n+1 = 2n - (n-1)$ , the difference formula has the exact solution  $u^n = n$ . With a given time step  $k$ , the solution at time  $t = nk$  will be  $u^n \approx n = t/n$ , which  $\rightarrow \infty$  as  $n \rightarrow \infty$  for  $t$  fixed. The scheme is not stable.

Stability for discretizations of nonlinear ODEs is an extremely rich subject; there are several definitions of stability, with subtle differences among

them. But since we are really interested in linear PDEs, let us make a tremendous simplification. We shall consider only *linear* problems whose solutions *decay* in time. For our purposes now, let us say that a discrete method is *stable* if no solutions of the difference formula grow in time.

Suppose our system is

$$\frac{d\mathbf{U}}{dt} = \mathbf{A}\mathbf{U} + \mathbf{R},$$

where  $\mathbf{A}$  and  $\mathbf{R}$  are constant in time. Let us suppose that we have an eigenvalue decomposition  $\mathbf{A}\mathbf{P} = \mathbf{P}\mathbf{\Lambda}$ , where  $\mathbf{\Lambda}$  is diagonal with entries  $(\lambda_1, \dots, \lambda_N)$ , and we suppose that  $\mathbf{P}$  is of full rank so that  $\mathbf{\Lambda} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$ . Since we only are interested in problems whose solutions decay in time, we suppose that each  $\lambda_j < 0$ , and hence  $\mathbf{A}$  is invertible.

We can define  $\mathbf{u}$  in terms of a new vector  $\mathbf{Y}$  as  $\mathbf{U} = \mathbf{P}\mathbf{Y} - \mathbf{A}^{-1}\mathbf{R}$ . Substituting into the ODE, we find that  $\mathbf{Y}$  solves  $d\mathbf{Y}/dt = \mathbf{\Lambda}\mathbf{Y}$ , which is a collection of independent scalar equations (because  $\mathbf{\Lambda}$  is diagonal). As a result, solving the ODE  $d\mathbf{U}/dt = \mathbf{A}\mathbf{U} + \mathbf{R}$  is equivalent to solving a family of scalar equation of the form  $dy/dt = \lambda_j y$  where  $\lambda_j$  are the eigenvalues of  $\mathbf{A}$ .

Thus let us consider only the simple linear scalar problem

$$u_t = -\sigma u,$$

(we take  $\sigma = -\lambda$ ) whose exact solution is

$$u(t) = u_0 e^{-\sigma t}.$$

This is an exponential which decays on a time scale of  $1/\sigma$ .

### Forward Euler

$$u^{n+1} = u^n - k\sigma u^n = (1 - k\sigma) u^n,$$

so the solution is

$$u^n = u^0 \eta^n, \quad \text{with } \eta = 1 - k\sigma.$$

To keep you on your toes — the superscript on  $u^n$  is an index, while on  $\eta^n$  it is an exponent. By our definition, the scheme is stable if solutions of the difference formula do not grow in time, that is if and only if  $|\eta| \leq 1$ , which requires

$$k \leq \frac{2}{\sigma}.$$

```

function ode( sigma, dt, tmax )
% ode( sigma, dt, tmax ) Plot discrete solution of ODE

nstep = floor(tmax/dt);
tmax = nstep*dt;          % make tmax be exact multiple of dt

% Uncomment just one of these statements
eta = 1 - sigma*dt;      % Forward Euler
%eta = 1 / ( 1 + sigma*dt );      % Backward Euler
%eta = (1 - 0.5*sigma*dt)/(1 + 0.5*sigma*dt); % Trapezoid

t = linspace( 0, tmax, nstep+1 );
u = zeros(1,nstep+1);    % allocate storage for whole array
u(1) = 1;
for i=1:nstep; u(i+1) = eta*u(i); end

% Exact exponential solution
t0 = linspace( 0, tmax, 101 ); u0 = exp(-sigma*t0);

plot( t, u, '-ok', t0, u0, 'k' ); xlabel('t'); ylabel('u(t)')

```

Note that if  $k < 1/\sigma$  then the solution decreases monotonically to zero. If  $1/\sigma < k < 2/\sigma$  then the sign of the solution alternates and the magnitude decreases to zero.

When  $\sigma$  is large, we need a small time step for stability. This is true for *all* explicit methods. The reason for the instability is clear (Figure 1, p. 5): When  $\sigma$  is large, the solution heads toward zero, but overshoots.

Fix a time  $t$  and use it to set the time-step  $k$  via  $k = t/n$ . We can then write

$$u^n = u^0 \eta^n = u^0 \eta^{t/k} = u^0 e^{-\sigma_k t},$$

and so the (discrete) decay rate for a solution computed with time step  $k$  is

$$\sigma_k = -\frac{1}{k} \ln \eta = -\frac{1}{k} \ln(1 - k\sigma) = \sigma \left(1 + \frac{1}{2}k\sigma + \dots\right) = \sigma + \sigma \mathcal{O}(k\sigma), \quad \sigma k \rightarrow 0,$$

This shows that solutions of the discrete problem approximate solutions of the continuous problem, as long as  $\sigma k$  is small; that is, the time step must be small relative to the intrinsic time scale of the solution itself. This is always true for explicit methods.

**Backward Euler**

$$u^{n+1} = u^n - k\sigma u^{n+1}, \quad \text{so} \quad u^{n+1} = \frac{1}{1+k\sigma} u^n$$

Again  $u^n = u^0 \eta^n$ , but now

$$\eta = \frac{1}{1+k\sigma}$$

(see Figure 4). Since  $\eta < 1$  for *all*  $k > 0$  (for  $\sigma > 0$ ), the scheme is *stable for all timesteps*  $k$ . This is a characteristic of well-constructed implicit methods, and is the reason for their use. The discrete decay rate is

$$\sigma_k = -\frac{1}{k} \log \frac{1}{1+k\sigma} = \sigma \left(1 - \frac{1}{2} k\sigma + \dots\right) = \sigma + \mathcal{O}(\sigma k), \quad \sigma k \rightarrow 0.$$

Since  $-\sigma < -\sigma_k < 0$ , solutions to the backward Euler formula always decay in time, though not as fast as the continuous solution.

When  $k$  is large for a given  $\sigma$ ,  $\eta$  is very near zero, so the discrete solution decays very rapidly. The discrete solution “fails gracefully.”

**Trapezoid**

$$u^{n+1} = u^n - \frac{1}{2}k\sigma(u^{n+1} + u^n),$$

so  $u^n = u^0 \eta^n$  with

$$\eta = \frac{1 - \frac{1}{2}k\sigma}{1 + \frac{1}{2}k\sigma}.$$

Since  $-1 < \eta < 1$  for all  $k$  the method is stable for *all*  $k > 0$  (Figure 4). The discrete decay rate is

$$\sigma_k = \sigma \left(1 + \frac{1}{12}\sigma^2 k^2 + \dots\right) = \sigma + \mathcal{O}((\sigma k)^2), \quad \sigma k \rightarrow 0.$$

In this case,  $-\sigma_k < -\sigma < 0$  and so solutions to the trapezoid rule always decay in time, slightly faster than the continuous solution. Note that for the trapezoid rule the discrete decay rate is significantly closer to the continuous decay rate than it was for Forward or Backward Euler. (The approximation is one higher order in  $k$  near  $k = 0$  than with Euler, since this is a second-order method.)

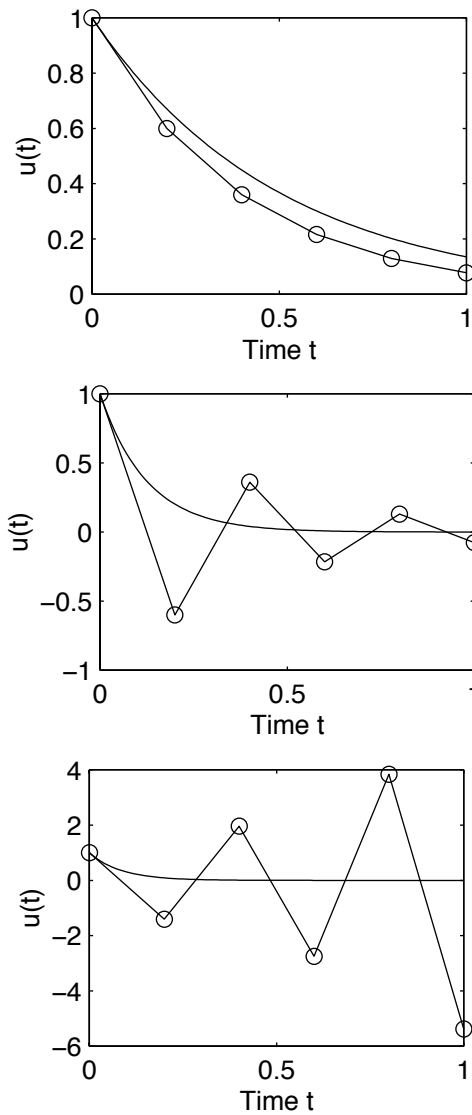


Figure 1: Forward Euler method with  $\sigma = 2, 8, 12$ ,  $k = 0.2$ . For  $\sigma k < 1$ , the discrete solution is well-behaved. For  $1 < \sigma k < 2$ , the discrete solution has  $-1 < \eta < 0$  (see Figure 4, so it decays with oscillations. For  $\sigma k > 2$ ,  $\eta < -1$ , so the discrete solution oscillates and grows.

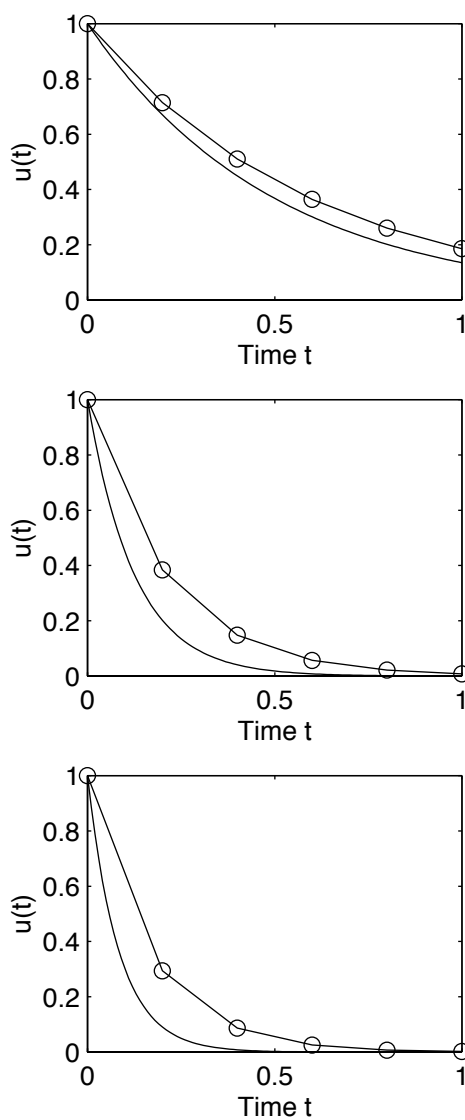


Figure 2: Backward Euler method with  $\sigma = 2, 8, 12$ ,  $k = 0.2$ . For all values of  $\sigma k$ , we have  $0 < \eta < 1$ , so the solution always decays without oscillating. Since  $\eta > \eta_{\text{true}}$ , the discrete solution decays more slowly than the true solution.

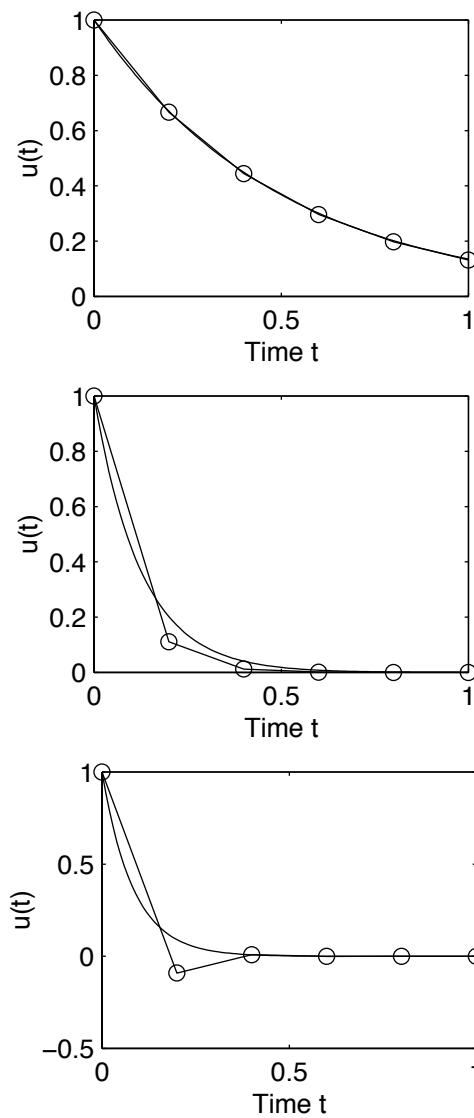


Figure 3: Trapezoid method with  $\sigma = 2, 8, 12$  and  $k = 0.2$ . The discrete solution is well-behaved for all values of  $\sigma k$  since  $|\eta| < 1$ , though it exhibits mild oscillations for  $\sigma k > 2$ , when  $\eta < 0$ . For small  $\sigma k$  it gives an extremely good approximation (second-order accurate).

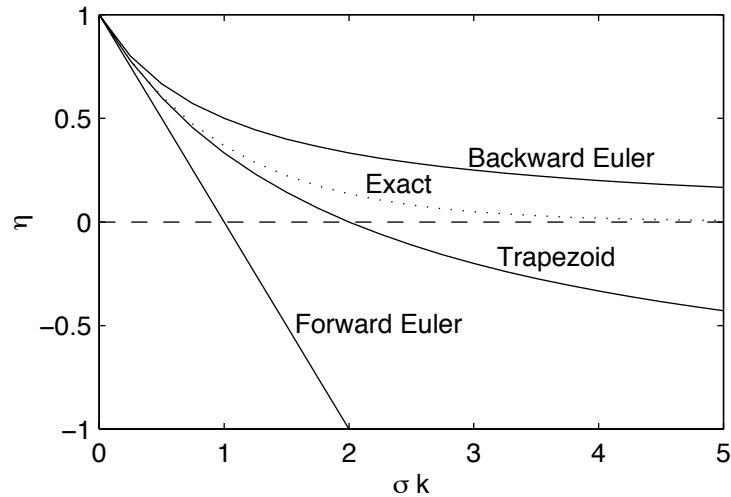


Figure 4: Amplification factors  $\eta$  as functions of  $\sigma k$ , and  $\eta_{\text{true}} = e^{-\sigma k}$ . We must have  $|\eta| \leq 1$  for stability. The forward Euler method loses stability when  $\sigma k > 2$ . The trapezoid (Crank-Nicolson) method has a negative amplification factor for  $\sigma k > 2$ , but it is stable for all  $\sigma k > 0$ . It has a second-order accurate match to the true solution near  $\sigma k = 0$ .

### Leapfrog

$$u^{n+1} = u^{n-1} - 2k\sigma u^n$$

Because this is a two-level formula, the discrete system has *two* exponential solutions. Looking for solutions in the form  $u^n = \eta^n$ , we get the quadratic

$$\eta^2 + 2k\sigma\eta - 1 = 0,$$

whose solutions are

$$\eta = -k\sigma \pm \sqrt{(k\sigma)^2 + 1}.$$

The general solution to the difference formula is

$$u^n = C_+\eta_+^n + C_-\eta_-^n \quad (1)$$

where  $C_{\pm}$  are determined by the necessary two levels of initial data, and

$$\begin{aligned} \eta_+ &= -k\sigma + \sqrt{(k\sigma)^2 + 1} \approx 1 - k\sigma + \dots, & \sigma k \rightarrow 0 \\ \eta_- &= -k\sigma - \sqrt{(k\sigma)^2 + 1} \approx -1 - k\sigma + \dots, & \sigma k \rightarrow 0. \end{aligned}$$

For all  $k > 0$  (and  $\sigma > 0$ ),  $0 < \eta_+ < 1$  and  $\eta_- < -1$  and so  $|\eta_+| < 1$  and  $|\eta_-| > 1$ . The solution  $\eta_+^n$  is a good approximation to a solution of the ODE. The solution  $\eta_-^n$  is an artificial, unstable, solution of the difference formula. The general solution (1) includes both modes and so even if the initial data were such that  $C_- = 0$  (meaning that on paper all is well and good) then round-off error would cause  $C_+ \neq 0$  and so small perturbations would grow exponentially fast. And so, the leapfrog scheme is *unstable for all k*.

All two-level schemes have two discrete solutions, one physical and one unphysical. But for a well-constructed method, the unphysical solution *decays* in time and does not affect the accuracy of the solution.