Survey of Applied Mathematics Techniques

Brian Wetton [1]

July 8, 2018

[1] www.math.ubc.ca/~wetton, wetton@math.ubc.ca

Lecture 8

Time Stepping Methods.

8.1 Initial Value Problems

We will consider time stepping schemes for general ODE initial value problems for $u(t)$:

$$\frac{du}{dt} = f(u(t), t) \ , \quad u(0) = u_0 \text{ given.} \tag{8.1}$$

We will consider this general form for scalar u when we assess the accuracy of time-stepping schemes, but all the schemes considered will be applicable to the vector $\mathbf{u}$ case. It is the general form since, as discussed in Lecture #5, higher order DEs and higher order systems can always be converted to first order systems. We will also consider the simple scalar problem

$$\frac{du}{dt} = \lambda u \ , \quad u(0) = 1. \tag{8.2}$$

with λ a given complex constant to investigate the stability of time stepping schemes.

We will often use Newton's notation for time derivatives:

$$\frac{du}{dt} := \dot{u}, \quad \frac{d^2u}{dt^2} := \ddot{u}, \text{ etc.}$$

8.2 Basic Time Stepping Schemes and Ideas

Consider approximating $u(t)$ on a regular grid in time with interval size $k = \Delta t$. We will use superscripts for the time level index (since later we will look at PDE problems where we will discretize in space and time).

$$U^n \approx u(nk).$$

The simplest method for approximating solutions to (7.1) is the Explicit Euler (Forward Euler) method:

$$U^{n+1} = U^n + kf(U^n, nk). \tag{8.3}$$

We call this scheme a *one-step method* since the value of the approximation at time level n determines the values at time level $n+1$. This is a convergent scheme with first order convergence: If the solution of (7.1) is well defined in $[0,T]$ and (7.3) is used with N time steps of size $k = T/N$ then

$$|u(T) - U^N| = O(k).$$

Note that if the exact solution is put into (7.3) we get an expression for the truncation error from linear Taylor approximation, since $f(u(nk), nk) = \dot{u}(nk)$:

$$u((n+1)k) - u(nk) - kf(u(nk), nk) = \frac{k^2}{2}\ddot{u}(\xi)$$

for some $\xi \in (nk, (n+1)k)$. Thus, the local error (after one time step) is $O(k^2)$. It makes sense that the error after $O(1/k)$ times steps is first order, $O(k)$. The local error is usually written as $k\tau$, where τ is the truncation error. For the Forward Euler method,

$$\tau = \frac{k}{2}\ddot{u}(\xi). \tag{8.4}$$

Theorem 1 (ODE Theory)**.** *If $f(u,t) \in C_n$ with $n \geq 1$, then the solution $u(t)$ of (7.1) exists, is unique and in C_{n+1} in a neighbourhood of $t = 0$.*

Theorem 2 (Dahlquist)**.** *Any "reasonable" one-step time stepping method converges to the exact solution at times for which it is defined with an order of convergence equal to the order of the truncation error.*

Here, "reasonable" means that the scheme is consistent and has a very basic stability property. Since all of the schemes we will discuss below are "reasonable", this theorem does not help us choose a scheme suitable for a specific problem.

Consider the FE scheme (7.3) applied to the scalar problem (7.2). The exact solution is

$$u(t) = e^{\lambda t}$$

and the discrete solution satisfies $U_0 = 1$, $U^{n+1} = (1 + k\lambda)U^n$, so

$$U^n = (1+z)^n \text{ , with } z = k\lambda.$$

It is clear that the so-called *Growth Factor* $G(z) = 1 + z$ for the scheme should approximate e^z when z is small, as it does.

We consider the set of complex z such that $|G(z)| \leq 1$. This is known as the *stability region* of the method. For FE,

$$|G(z)| = |1 + z| \leq 1$$

is a circle of radius 1 in the complex plane, centred at $z = -1$ as shown in Figure 7.1. Note that if λ is real and negative, the exact solution decays in time, but the FE approximation will only decay if

$$|z| = |k\lambda| < 2 \text{ , that is } k < 2/\lambda. \tag{8.5}$$

This does not violate Dahlquist's theorem above, since as $k \to 0$, (7.5) is eventually satisfied.

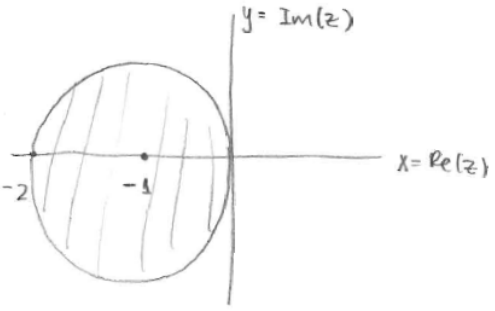

Figure 8.1: Stability region for Forward Euler time stepping.

8.3 Higher order and implicit methods

A second order explicit method called Improved Euler is given below:

$$\begin{aligned} U^* &= U^n + kf(U^n, nk) \\ U^{n+1} &= U^n + \frac{k}{2}\left(f(U^n, nk) + f(U^*, (n+1)k)\right). \end{aligned}$$

This is a *two-stage* method, with two evaluations of the right hand side function f and U^* is the value at the intermediate stage. However, since U^n does determine U^{n+1} it is still a *one-step* method. It is one of a family of second order Runge-Kutta methods. It can be written more compactly:

$$U^{n+1} = U^n + \frac{k}{2}\left(f(U^n, nk) + f(U^n + kf(U^n, nk), (n+1)k)\right). \tag{8.6}$$

From this form, the truncation error can be identified:

$$\tau = \frac{1}{k}\left[u^{n+1} - u^n - \frac{k}{2}(f^n + f(u^n + kf^n, (n+1)k))\right]$$

where $u^n := u(nk)$ and $f^n := f(u^n, nk) = \dot{u}(nk)$. We can expand all variables in Taylor series at $t = nk$, to obtain

$$\tau = \frac{1}{k}\left[k\dot{u} + \frac{k^2}{2}\ddot{u} + \frac{k^3}{6}\frac{d^3u}{dt^3} - \frac{k}{2}\dot{u} - \frac{k}{2}\left(\dot{u} + k(f_u f + f_t) + \frac{k^2}{2}(f_{uu}f^2 + f_{tt} + f_{ut}f)\right) + O(k^4)\right]$$

where all functions are evaluated at $t = nk$ and $u = u(nk)$. Starting with $\dot{u} = f(u, t)$ it can be shown that

$$\ddot{u} = f_u f + f_t$$

and

$$\frac{d^3u}{dt^3} = f_{uu}f^2 + 2f_{ut}f + (f_u)^2 f + f_u f_t + f_{tt}.$$

Using the first of the results above, we see that τ is second order

$$\tau = k^2 \left(\frac{1}{6}\frac{d^3u}{dt^3} - \frac{1}{4}(f_{uu}f^2 + f_{tt} + f_{ut}f) \right) + O(k^3).$$

Note that the dominant error term in the truncation error is not a simple time derivative of u as it was for the FE method. This has implications for error estimation in adaptive methods as we shall see in later discussion.

We can consider the stability of the Improved Euler scheme by considering the form (7.6) with $f(u,t) = \lambda u$, giving

$$U^{n+1} = (1 + z + \frac{z^2}{2})U^n$$

which defines a growth factor $G(z) = 1 + z + z^2/2$. The stability region, $\{z : |G(z)| \leq 1\}$ is shown in Figure 7.2. As with FE, it is seen that IE is not suitable for problems that have λ with a large negative real part. Such problems are called *stiff* problems. We give a more complete definition below.

Definition 1 (Stiff ODE systems)**.** *Solutions of a well-behaved ODE system can be multiplied by $e^{-\beta t}$ with some $\beta > 0$ so that they are bounded. The value of β can always be chosen and time scaled so that the resulting transformed system, when linearized around desired solutions, has $O(1)$ eigenvalues (components that evolve on an $O(1)$ time scale). A system is called* stiff *if the linearization of this transformed system also has eigenvalues of large size.*

Theorem 3 (Dahlquist)**.** *The stability region of every explicit scheme is bounded.*

From this, it can be seen that no explicit scheme is suitable for stiff problems.

There are two properties we would like to have for a time-stepping scheme for stiff problems, summarized in the following definitions.

Definition 2 (L-stability)**.** *A time-stepping scheme is called L-stable if $|G(z)| < 1$ for all z with $\Re(z) < 0$. That is, the stability region contains the left half plane.*

Definition 3 (A-stability)**.** *A time-stepping scheme is called A-stable if*

$$G(z) \to 0 \quad as \quad \Re(z) \to -\infty.$$

This matches the property that $G(z)$ should approximate e^z.

Important Note: In some literature, the "A" and "L" definitions are reversed! There were two groups that could not agree on notation. I like "L" for left half-plane and "A" for asymptotic.

From Theorem 3 we see that no explicit scheme is L-stable or A-stable. Thus we turn to implicit schemes. The simplest scheme of this type is the Backward Euler scheme:

$$U^{n+1} = U^n + kf(U^{n+1}, (n+1)k).$$

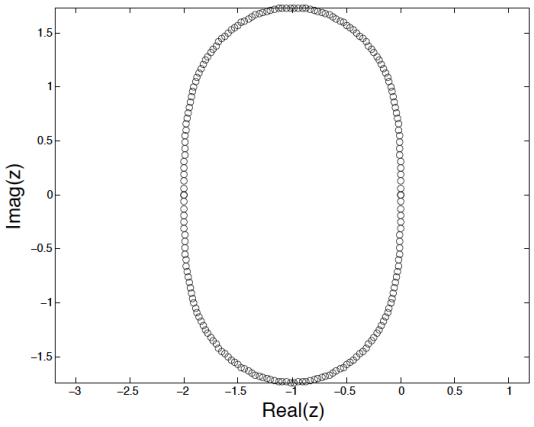

Figure 8.2: Stability region for Improved Euler time stepping (inside the curve shown). This is computed by setting $G(z) = e^{i\theta}$ for a grid of θ points and plotting the roots of the quadratic.

It is called an implicit method because U^{n+1} is specified by an implicit relationship above. A linear or nonlinear system must be solved for U^{n+1} at every time step. It is a first order method with truncation error

$$\tau = -\frac{k}{2}\ddot{u}(\xi) \tag{8.7}$$

for some $\xi \in (nk, (n+1)k)$. It has growth factor

$$G(z) = \frac{1}{1-z}$$

and so has a stability region that is outside the unit circle centred at $z = 1$ as shown in Figure 7.3. From the form of $G(z)$ above and the shape of the stability region, it is clear that BE is both L-stable and A-stable. Thus, it is suitable for application to stiff problems.

8.4 Higher Order Implicit Schemes

We consider four higher order implicit schemes, each with some advantages and some disadvantages.

8.4.1 Trapezoidal Rule

We consider the Trapezoidal Rule

$$U^{n+1} = U^n + \frac{k}{2}\left(f(U^n, nk) + f(U^{n+1}, (n+1)k)\right)$$

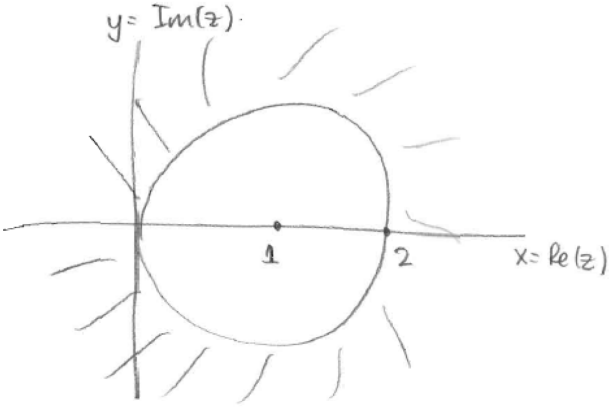

Figure 8.3: Stability region for Backward Euler time stepping.

and the Implicit Midpoint Rule

$$U^{n+1} = U^n + kf((U^{n+1} + U^n)/2, (n + 1/2)k).$$

These are different schemes but become identical when applied to constant coefficient, linear, autonomous problems. They have the same dominant error term and the same growth factor and stability regions. When the Trapezoidal Rule is applied to diffusion problems, it is also known as the Crank-Nicholson method, and this name is sometimes used for the approach applied to other problems.

The growth factor is

$$G(z) = \frac{1 + z/2}{1 - z/2}$$

Here, the stability region is *exactly* the left half plane, and the method is thus L-stable. However,

$$G(z) \to -1 \quad \text{as} \quad \Re(z) \to -\infty.$$

so the method is not A-stable. Care should be used in the application of the method to stiff problems since components that should almost decay to zero in one time step instead just oscillate in sign. On the other hand, it does have desirable properties: L-stable, second order with a small error constant, one-step and one-stage.

8.4.2 Second Order Backward Differentiation Formula (BDF2)

This is a multi-step method, involving the values of two previous time steps, U^n and U^{n-1}. The first step, U^1, can be computed with BE without loss of overall

accuracy.

$$U^{n+1} = \frac{4}{3}U^n - \frac{1}{3}U^{n-1} + \frac{2k}{3}f(U^{n+1}, (n+1)k)$$

It is based on the second order, one sided difference formula we derived earlier in the course

$$\dot{u}((n+1)k) \approx \frac{3U^{n+1} - 4U^n + U^{n-1}}{2k}.$$

It has truncation error

$$\tau = \frac{2}{9}k^2\frac{d^3u}{dt^3}(\xi)$$

with $\xi \in ((n-1)k, (n+1)k)$. To analyze the stability, consider the method applied to (7.2):

$$U^{n+1} = \frac{4}{3}U^n - \frac{1}{3}U^{n-1} + \frac{2}{3}zU^{n+1}$$

or

$$(1 - \frac{2z}{3})U^{n+1} - \frac{4}{3}U^n + \frac{1}{3}U^{n-1} = 0.$$

For a given z, this is a second order constant coefficient difference equation with solution

$$U^n = AG_1^n + BG_2^n \tag{8.8}$$

for some constants A and B and $G_1(z)$, $G_2(z)$ roots of

$$(1 - \frac{2z}{3})G^2 - \frac{4}{3}G + \frac{1}{3} = 0.$$

It is important to clarify that the superscript n for U in (7.8) is the time level, but for G_1 and G_2 it is an exponent. The stability region for two-step schemes is

$$\{z : |G_1(z)| < 1 \text{ and } \quad |G_2(z)| < 1\}$$

The stability region for BDF-2 is shown in Figure 7.4. It is clear that it is L-stable. Since

$$G_{1,2} = \frac{\frac{2}{3} \pm \sqrt{\frac{4}{9} - \frac{1}{3}(1 - 2z/3)}}{1 - 2z/3} \tag{8.9}$$

it is also clear that the scheme is A-stable ($|G_{1,2}| = O(|z|^{-1/2})$ in the limit $|z| \to \infty$). Considering (7.9) in the limit as $z \to 0$ (think of fixed λ, $k \to 0$), we have

$$G_1 \approx 1\ , \quad G_2 \approx \frac{1}{3}.$$

The term G_2^n appears as an initial layer that accounts for the error from the initialization procedure for U^1.

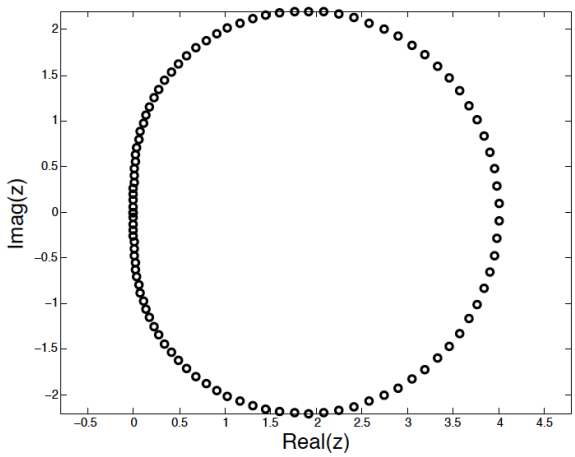

Figure 8.4: Stability region for BDF-2 (outside the curve shown). This is computed by setting $G(z) = e^{i\theta}$ for a grid of θ points and plotting $z = 3/2 - (2G - 1/2)/G^2$.

8.4.3 Second Order Diagonally Implicit Runge-Kutta (DIRK2)

This is a one-step, two-stage implicit method:

$$\begin{aligned} U^* &= U^n + k\alpha f(U^*, (n+\alpha)k) \\ U^{n+1} &= U^n + k\left(\alpha f(U^{n+1}, (n+1)k) + (1-\alpha) f(U^*, (n+\alpha)k)\right). \end{aligned}$$

with $\alpha = 1 - \sqrt{2}/2$. The term "diagonal" applies since each stage is only implicit in the value at that stage. It is both A-stable and L-stable. Since it is a one-step method, adaptive methods for time step size are more easily implemented than for BDF-2. Over one time step, it is more accurate than BDF-2 under certain assumptions (the error terms are not directly comparable in general). However, taking into account the fact that BDF-2 has one implicit solve per time step and DIRK-2 has two, BDF-2 is more efficient for fixed time step computations.

8.4.4 Radau II-A

This is a third order, two stage method that is both L-stable and A-stable.

$$\begin{aligned} U^* &= U^n + k\left(\frac{5}{12} f(U^*, (n+1/3)k) - \frac{1}{12} f(U^{n+1}, (n+1)k)\right) \\ U^{n+1} &= U^n + k\left(\frac{3}{4} f(U^*, (n+1/3)k) + \frac{1}{4} f(U^{n+1}, (n+1)k)\right). \end{aligned}$$

Note that this scheme is implicit in both U^* and U^{n+1} simultaneously.

8.5 Adaptive Time Stepping

We discuss adaptive time stepping with a particular, simple example. Consider applying BE to a problem and wanting to make the local error at each time step smaller than a user defined tolerance δ. The local error is (7.7):

$$k\tau = -\frac{k^2}{2}\ddot{u}(\xi_1) \approx -\frac{k^2}{2}\ddot{u}(nk)$$

The local error for FE (7.4) is

$$k\tau = \frac{k^2}{2}\ddot{u}(\xi_2) \approx \frac{k^2}{2}\ddot{u}(nk)$$

So, if we compute the FE solution U^{FE} as well as the BE solution U^{n+1}, the local error E in U^{n+1} is approximately

$$E \approx \frac{1}{2}|U^{n+1} - U^{FE}|.$$

Note that U^{FE} is cheap to compute (an explicit method) and may be useful as a good initial guess for iterative solvers for U^{n+1}. The value of U^{FE} is not used in subsequent calculations, so there is no stability problem.

From the value of E computed above, we would make some decisions about the step from t_n to $t_{n+1} = t_n + k$:

If $E > \delta$: We would fail the time step and recompute U^{n+1} with a reduced time step k.

If $E \leq \delta$: We would accept the time step and would compute a new time step k so that $E \leq \delta$ would be likely to be satisfied again. Since

$$E \approx Ck_{old}^2$$

(with $C \approx |\ddot{u}(t_n)|/2$) and we want $E \leq \delta$ we could take k_{new} with

$$Ck_{new}^2 \approx \frac{E}{k_{old}^2}k_{new}^2 \leq \delta.$$

In practice, the formula

$$k_{new} = \theta k_{old}\sqrt{\delta/E}$$

is used, with $\theta < 1$ a computational parameter, a "safety" factor. I typically use $\theta = 0.8$. There is also typically a limit on how much k is allowed to increase. On a failed step for which E is not that much bigger than δ, the formula above can also be used. On a "bad" failure, k is typically reduced drastically (i.e. by a factor of two).

Note that higher order single step methods have more complicated error structure. For these methods, typically higher order methods are used to assess accuracy of lower order methods. For example, MATLAB's `ode45` explicit solver uses a fifth order Runge-Kutta method to assess the accuracy of a fourth order one. It uses the Dormand-Prince pair, which is a six stage method carefully chosen so that the same function evaluations are used for both methods.

8.6 Butcher Tables

The coefficients of one-step methods can be represented in Butcher Tables. The general s stage method (s function evaluations) is given by

$$\begin{aligned} U^{n+1} &= U^n + \sum_{i=1}^{s} b_i K_i \\ K_i &= kf\left(U^n + \sum_{j=1}^{s} a_{ij} K_j, t_n + c_i k \right) \end{aligned}$$

The constants in the general method above can be entered into a table:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \ddots & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

An explicit method has zeros in the diagonal of $\mathbf{A}$ and above. A diagonally implicit method has non-zeros on the diagonal but zeros above. The tables for some of the schemes discussed in this section are given below:

Improved Euler:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

DIRK-2: with $\alpha = 1 - \sqrt{2}/2$

$$\begin{array}{c|cc} \alpha & \alpha & 0 \\ 1 & 1-\alpha & \alpha \\ \hline & 1-\alpha & \alpha \end{array}$$

Radau IIA:

$$\begin{array}{c|cc} 1/3 & 5/12 & -1/12 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4 \end{array}$$

8.7 Lecture #8 Problems

Problem 1. *Show that the DIRK-2 method in the class notes is second order accurate and A-stable.*

Problem 2. *Implement the Forward Euler scheme on the problem $u(t)$ with*

$$\ddot{u} + u = 0$$

with initial conditions $u(0) = 1$, $\dot{u}(0) = 0$, written as a first order system. The exact solution is $u(t) = \cos t$. Observe the convergence of the scheme at $t = 8\pi$. The exact solution satisfies

$$\ddot{u}^2 + u^2 \equiv 1$$

for all time (this can be shown analytically). How does this quantity vary in time in your scheme from A4? Based on the stability region for FE and the eigenvalues of this problem written as a system, why would you expect this behaviour? Do any of the schemes considered in the lecture notes preserve this identity exactly? (either discuss why they don't or find one that does).

Problem 3. *Show that the Radau IIA method in the class notes is indeed third order order, L-stable and A-stable.*

Problem 4. *Consider the Lax Wendroff scheme applied to the one-way wave equation $u_t = u_x$:*

$$U^{n+1} = U^n + kD_1U^n + \frac{k^2}{2}D_2U^n$$

where solutions are 1-periodic in x. Consider the scheme with $k = Ch$ with $0 < C < 1$. Show that it is second order accurate. Note that this is not a MOL scheme. Hint: *You will have to differentiate the partial differential equation with time to match some of the terms. Show that the method is stable as long as $k < h$.* Hint: *use von-Neumann analysis.*

Problem 5. *Consider the following scheme for $\ddot{u} = f(u)$:*

$$\begin{aligned} U^{n+1} &= U^n + k\left(V^n + \frac{k}{2}f(U^n)\right) \\ V^{n+1} &= V^n + \frac{k}{2}\left(f(U^n) + f(U^{n+1})\right). \end{aligned}$$

Show that the method is second order accurate. Note that it is a non-standard explicit scheme, specialized to this particular problem structure.

Problem 6. *Consider the following scheme for the 1D wave equation $u_{tt} = u_{xx}$:*

$$\frac{1}{k^2}(U_j^{n+1} - 2U_j^n + U_j^{n-1}) = \frac{1}{h^2}(U_{j+1}^n - 2U_j^n + U_{j-1}^n).$$

What time step restrictions are needed for the scheme to be stable?