CS3220 Lecture Notes: Backward Euler Method

Steve Marschner
Cornell University

22 April 2009

These notes are to provide a reference on Backward Euler, which we discussed in class but is not covered in the textbook. The notation has been cleaned up some from what I used in lecture, to make the derivations easier to read, but I've tried to indicate the connection to my earlier notation.

1 One equation

Given a one-variable ODE in the usual form:

$$x'(t) = f(t, x(t))$$

the forward Euler step (as discussed in the textbook) from time t to $t+h$ (which I called t^+ in lecture) is

$$x^+ = x + hf(t, x)$$

where x and x^+ are the computed values for $x(t)$ and $x(t+h)$, respectively. Or, if I write it a little more like a Runge-Kutta method,

$$\begin{aligned} k &= hf(t, x) \\ x^+ &= x + k. \end{aligned}$$

Forward Euler takes a step along the derivative *at the current time and position.*

The backward Euler method uses almost the same time stepping equation:

$$k = hf(t + h, x + k)$$

Backward Euler chooses the step, k, so that the derivative *at the new time and position* is consistent with k. Doing this requires solving this equation for k, which amounts to a root finding problem if f is nonlinear, but we know how to solve those. The forward Euler step $k = hf(t, x)$ is a reasonable place to start the root finding iteration.

2 More than one equation

Backward Euler also works for a system of n ODEs:

$$\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t)),$$

where $\mathbf{x} : I\!R \to I\!R^n$. The forward Euler step from time t to time $t + h$ is simply

$$\mathbf{k} = h\mathbf{f}(t, \mathbf{x}).$$

where $\mathbf{k} = \mathbf{x}^+ - \mathbf{x}$. Backward Euler uses the same step equation but evaluates the derivative at the ending time, $t + h$, and position, $\mathbf{x} + \mathbf{k}$:

$$\mathbf{k} = h\mathbf{f}(t + h, \mathbf{x} + \mathbf{k}).$$

This is a system of n nonlinear equations in n variables, which we can solve for $\mathbf{k}$ using the multivariable Newton's method, which we studied earlier in the course. In this case it's worth looking at how this will work out.

First, package the problem up in the standard form:

$$\mathbf{g}(\mathbf{k}) = \mathbf{k} - h\mathbf{f}(t + h, \mathbf{x} + \mathbf{k}) = 0.$$

The Netwon step for this system is the solution to the matrix equation:

$$\nabla\mathbf{g}(\mathbf{k})\Delta\mathbf{k} = -\mathbf{g}(\mathbf{k})$$

Where $\mathbf{k}$ is the current iterate and $\mathbf{k} + \Delta\mathbf{k}$ is the next one. Recall that $\nabla\mathbf{g}(\mathbf{k})$ is the n-by-n matrix of partial derivatives of $\mathbf{g}$, evaluated at $\mathbf{k}$. This derivative is

$$\nabla\mathbf{g}(\mathbf{k}) = \mathbf{I} - h\nabla_{\mathbf{x}}\mathbf{f}(t + h, \mathbf{x} + \mathbf{k})$$

so the system to be solved for the Newton step is

$$\left(\mathbf{I} - h\nabla_{\mathbf{x}}\mathbf{f}(t + h, \mathbf{x} + \mathbf{k})\right)\Delta\mathbf{k} = h\mathbf{f}(t + h, \mathbf{x} + \mathbf{k}) - \mathbf{k}.$$

Of course, one needs a starting value for $\mathbf{k}$. One could use the forward Euler step; or another popular starting value is zero.

This derivation can be made slicker by converting the problem to an autonomous one, one with no explicit time dependence, as discussed in the book. This leads to a system

$$\mathbf{x}'(t) = \mathbf{f}(\mathbf{x}(t))$$

where one of the entries in $\mathbf{x}$ plays the role of t if needed. Starting from this equation, the step equation is

$$\mathbf{k} = h\mathbf{f}(\mathbf{x} + \mathbf{k})$$

and the linear system for the Newton iteration is

$$\left(\mathbf{I} - h\nabla\mathbf{f}(\mathbf{x} + \mathbf{k})\right)\Delta\mathbf{k} = h\mathbf{f}(\mathbf{x} + \mathbf{k}) - \mathbf{k},$$

or, if we just use one Newton iteration and use zero for the starting value of $\mathbf{k}$,

$$\left(\mathbf{I} - h\nabla\mathbf{f}(\mathbf{x})\right)\mathbf{k} = h\mathbf{f}(\mathbf{x}).$$