NE 155, Class 30, S17
Taylor Series Methods and Runge-Kutta

materials from: Kathryn Huff

April 5, 2017

Introduction

Recall that the initial value problem takes the form:

$$u'(t) = f(u(t), t), \text{ for } t > t_0 \tag{1}$$

where

$$u(t_0) = u_0 \tag{2}$$

In such a problem, we desire to compute $u(t_1)$, $u(t_2)$, $u(t_n)$ and so on.

There are many methods for solving initial value problems numerically. This lesson will introduce a simple one, Forward Euler, which is derived from a Taylor series expansion. We will then use Forward Euler to introduce a two-stage, explicit Runge-Kutta method as well, for higher accuracy.

Taylor Series Derivation of Forward Euler

The simplest method for finding $u(t_n)$ is Forward Euler, which approximates $u(t_n)$. Let's call this approximation, U^n. In this notation, Forward Euler is based on replacing $u'(t_n)$ with $(U^{n+1} - U^n)/k$, where k is the width of the timestep. The Forward Euler method arises from a Taylor series expansion of $u(t_{n+1})$ about $u(t_n)$:

$$u(t_{n+1}) = u(t_n) + ku'(t_n) + \frac{1}{2}k^2u''(t_n) + \cdots \tag{3}$$

With this, the $O(k^2)$ terms can be dropped to give:

$$u(t_{n+1}) \approx u(t_n) + ku'(t_n) \tag{4}$$

And, based on equation (1) we can replace $u'(t_n)$ with $f(u(t_n), t_n)$:

$$u(t_{n+1}) = u(t_n) + kf(u(t_n), t_n) + H.O.T. \tag{5}$$

This expression gives a one-step truncation error of order $O(k^2)$ (that is, this step introduces $O(k^2)$ error) and a global truncation error of $O(k)$ (because you take T/k time steps to get to time T). More accurate schemes can be derived with a Taylor series expansion by retaining higher order terms in equation (3). Since we are only given $u'(t_n) = f(u(t_n), t_n)$, however, the computation of such schemes requires repeated recursive differentiation of this function, and can get quite messy.

Runge-Kutta Methods

LeVeque, Randall J. *Finite Difference Methods for Ordinary and Partial Differential Equations*. Philadelphia, PA: SIAM, 2007.

Runge-Kutta is a method used in practice to get a higher order approximation *without* explicitly calculating higher order derivatives.

Runge-Kutta uses two stages. The first stage is an update using Euler's method, approximating

$u(t_{n+1/2})$.

$$U^{n+1/2} = U^n + \frac{1}{2}kf(U^n) \tag{6}$$

$$\tag{7}$$

The second stage evaluates the function, f, at the midpoint to estimate the slope.

$$U^{n+1} = U^n + kf(U^{n+1/2}) \tag{8}$$

These equations can be combined into a single expression:

$$U^{n+1} = U^n + kf(U^n + \frac{1}{2}kf(U^n)) \tag{9}$$

This approximation, because it uses two points, like a centered approximation, is order $O(k^3)$ one-step accurate (and $O(k^2)$ globally).

A generic r-stage Runge-Kutta method can be expressed as:

$$Y_1 = U^n + k\sum_{j=1}^{r} a_{1j}f(Y_j, t_n + c_jk) \tag{10}$$

$$Y_2 = U^n + k\sum_{j=1}^{r} a_{2j}f(Y_j, t_n + c_jk) \tag{11}$$

$$\vdots$$

$$Y_r = U^n + k\sum_{j=1}^{r} a_{rj}f(Y_j, t_n + c_jk) \tag{12}$$

$$U^{n+1} = U^n + k \sum_{j=1}^{r} b_j f(Y_j, t_n + c_j k) \tag{13}$$

$$\sum_{j=1}^{r} a_{ij} = c_i \,, \qquad i = 1, 2, \ldots, r \tag{14}$$

$$\sum_{j=1}^{r} b_j \,. \tag{15}$$

There are different ways to determine the coefficients that give different orders of accuracy. See textbook referenced for more information.

Application to PRKE

Each of these can be applied to the PRKE. In particular, let's consider the application of a Forward Euler.

To avoid confusion with the multiplication factor, the width of our timestep will be called Δt.

$$n(t_{n+1}) = n(t) + \Delta t \left[\frac{\rho(t_n) - \beta}{\Lambda} n(t_n) + \sum_{j=1}^{j=J} \lambda_j \zeta_j \right] \tag{16}$$