Math 6316, Spring 2016
Due 22 April 2016

Homework 7 – Initial Value Problems

1. Under the standard assumptions

$$\sum_{j=1}^{s} b_j = 1 \qquad \text{and} \qquad \sum_{j=1}^{s} a_{i,j} = c_i, \quad i = 1, \ldots, s,$$

derive the equations that must be satisfied by the coefficients $\{c_i\}_{i=1}^3$, $\{b_j\}_{j=1}^3$ and $\{a_{i,j}\}_{i,j=1}^3$ for a 3-stage, $O(h^3)$-accurate, explicit Runge Kutta method. Show all relevant steps in this derivation.

Verify that the 3-stage explicit Runge-Kutta method given by the Butcher table

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & -1 & 2 & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

satisfies these equations.

2. Derive the Adams-Bashforth method that uses $\{f_n, f_{n-1}, f_{n-2}\}$ to compute u_{n+1}. Show all relevant steps in this derivation.

3. Matlab: create a function of the form

```
function unew = AB3_step(f, t, u, u1, u2, h)
```

that implements your Adams-Bashforth method above. The inputs to this function are:

- `f` is the function handle or name to be evolved,
- `t` is the current time, t_n,
- `u`, `u1` and `u2` are the current and two previous time step solutions, u_n, u_{n-1} and u_{n-2},
- `h` is the time step size, $t_{n+1} - t_n$.

The output is the approximation of the new time-step solution, u_{n+1}.

Also create a function of the form

```
function unew = RK3_step(f, t, u, h)
```

that implements the 3-stage Runge-Kutta method given by the Butcher table in problem 1. The inputs and output for this function are identical to the similarly-named inputs to `AB3_step`.

Write a script named `hw7.m` that uses these functions to approximate solutions to the initial value problem

$$y'(t) = -y(t) + 2\cos(t), \quad y(0) = 1, \quad t \in [0, 10],$$

For the first two steps, use the 3-stage ERK method given by the Butcher table in problem 1. Use your Adams-Bashforth method for the remaining steps.

Use this combination RK3/AB3 method with time steps of size $h = \frac{10}{n_j}$ for $n_j = 10 * 2^j$ for $j = 1, \ldots, 9$. For each n output the maximum error in the solution over the time, $\|y - u\|_\infty$, and for $j > 0$, compute the convergence rate of your approximations.