APMA 4301: Problem Set 5

Brian Dawes

November 25, 2014

1.

(a) Butcher Tableaus

Forward Euler

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

Backward Euler

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

Mid-Point

$$\begin{array}{c|cc} 0 & 0 & \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array} \tag{3}$$

Improved Euler (RK2)

$$\begin{array}{c|cc} 0 & 0 & \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{4}$$

Trapezoidal

$$\begin{array}{c|cc} 0 & 0 & \\ 1 & 0 & 1 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{5}$$

Classical 4th order Runge-Kutta (RK4)

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

TR-BDF2

(b) Explicit Form

Forward Euler

$$u_1(k) = (1+z)u_0 \tag{7}$$

Backward Euler

$$\begin{aligned} u_1(k) &= u_0 + zu_1(k) \\ u_1(k) &= \frac{1}{1-z}u_0 \end{aligned} \tag{8}$$

Mid-Point

$$u_1(k) = (1+z+z^2/2)u_0 \tag{9}$$

Improved Euler (RK2)

$$u_1(k) = (1+z+z^2/2)u_0 \tag{10}$$

Trapezoidal

$$\begin{aligned} u_1(k) &= u_0 + z/2(u_0 + u_1(k)) \\ u_1(k) &= \frac{1+z/2}{1-z/2}u_0 \end{aligned} \tag{11}$$

Classical 4th order Runge-Kutta (RK4)

$$\begin{aligned} u_1(k) &= (1/6 + 1/3 + z/6 + 1/3 + z/6 + z^2/12 + 1/6 + z/6 + z^2/12 + z^3/12)u_0 \\ &= (1 + z/2 + z^2/6 + z^3/12)u_0 \end{aligned} \tag{12}$$

TR-BDF2

(c) Step Error

Forward Euler

$$e^z - R(z) = z^2/2 + O(z^3) \tag{13}$$

Backward Euler

$$\begin{aligned} e^z - R(z) &= (1 + z + z^2/2 + \dots) - (1 + z + z^2 + \dots) \\ &= -z^2/2 + O(z^3) \end{aligned} \tag{14}$$

Mid-Point

$$e^z - R(z) = z^3/6 + O(z^4) \tag{15}$$

Improved Euler (RK2)

$$e^z - R(z) = z^3/6 + O(z^4) \tag{16}$$

Trapezoidal

$$\begin{aligned} e^z - R(z) &= (1 + z + z^2/2 + z^3/6 \dots) - (1 + z + z^2/2 + z^3/4 \dots) \\ &= -z^3/12 + O(z^4) \end{aligned} \tag{17}$$

Classical 4th order Runge-Kutta (RK4)

$$e^z - R(z) = z^5/5! + O(z^6) \tag{18}$$

TR-BDF2

(d) Stability Plots

Forward Euler

Backward Euler

Mid-Point

Improved Euler (RK2)

Trapezoidal

Classical 4th order Runge-Kutta (RK4)

TR-BDF2

(e) Stable Time Steps

Forward Euler

$$|1 + \lambda k_{max}| = 1 \implies k_{max} = -\frac{2}{\lambda} \tag{19}$$

Backward Euler

$$|\frac{1}{1 - \lambda k_{max}}| = 1 \implies k_{maz} = \infty \tag{20}$$

Mid-Point

$$|1 + \lambda k_{max} + (\lambda k_{max})^2/2| = 1 \implies k_{max} = -\frac{1}{\lambda} \tag{21}$$

Improved Euler (RK2)

$$|1 + \lambda k_{max} + (\lambda k_{max})^2/2| = 1 \implies k_{max} = -\frac{1}{\lambda} \tag{22}$$

Trapezoidal

$$|\frac{1 + \lambda k_{max}/2}{1 - \lambda k_{max}/2}| = 1 \implies k_{maz} = \infty \tag{23}$$

Classical 4th order Runge-Kutta (RK4)

TR-BDF2

2.

$$\frac{\partial T}{\partial t} = \nabla^2 T; \quad T(\mathbf{x}, 0) = A \exp\left[\frac{-\mathbf{x}^T\mathbf{x}}{\sigma^2}\right] \tag{24}$$

(a)

$$T(\mathbf{x},t) = \frac{A}{1+2t/\sigma^2} \exp\left[\frac{\mathbf{x}^T\mathbf{x}}{\sigma^2+4t}\right] \tag{25}$$

This obviously satisfies the initial condition at $t = 0$.

$$\frac{\partial T}{\partial t} = \frac{A}{1+2t/\sigma^2}\left[\frac{4\mathbf{x}^T\mathbf{x}}{(\sigma^2+4t)^2} - \frac{2}{\sigma^2+4t}\right] \exp\left[\frac{\mathbf{x}^T\mathbf{x}}{\sigma^2+4t}\right] = \nabla^2 T \tag{26}$$

(b)

i.

$$\begin{aligned} F(u_i) &= \int_\Omega u_t[u_i - u_n + (1-\theta)\nabla^2 u_n + \theta\nabla u_i^2]\, d\mathbf{x} \\ &= \int_\Omega u_t(u_i - u_n) + k\nabla u_t \cdot [(1-\theta)\nabla u_n + \theta\nabla u_i]\, d\mathbf{x} \end{aligned} \tag{27}$$

Note that the surface terms are 0 due to the Neumann boundary condition.

In UFL, this can be written as:

```
F = (k*inner(grad(u_t),theta*grad(u_i)
    +(1-theta)*grad(u_n))+inner(u_t,(u_i-u_n)))*dx
```

ii.

Differentiating F, we get the Jacobian:

$$J(u_i) = u_t - k\theta\nabla^2 u_t \tag{28}$$

after using the identity $\frac{\partial}{\partial u}(f \cdot \nabla u) = -\nabla f$.

In UFL this can be done implicitly with:

```
J = derivative(F,us_i,us_a)
```

iii.

The file `diffusion.tfml` solves this problem with $\theta = 0$. The file `diffusion.shml` loops over the problem for $\theta = 0, 0.5, 1$. The L1 error functional is included in the diagnostics.

iv.

v.

For $\theta = 0.1$, we have an L1 error of 8.7694862558e-05 at $t = 0.012$ and $k = 0.0002$. For $\theta = 0.5$, we have an L1 error of 3.3144604514e-05 at $t = 0.012$ and $k = 0.0002$. Since the step error of the backwards Euler method is second order, we should get the same error when running with $k = 0.0000285$. Using this timestep, we get an error of 3.1004511669e-05, which is actually slightly better than the trapezoidal scheme.