Amath-Math 586/Atm S 581 Spring 2005

Homework 3 Solutions

1a) Assume A(t) is rapidly driven into a balance between its source 1 - cos(t) and its sink -λA . Then $A(t) \approx A_1\lambda^{-1}$, where $A_1(t)$ = 1 - cos(t) . This dominant balance is not exact, but by writing A in a perturbation series $A = A_1(t)\lambda^{-1} + A_2(t)\lambda^{-2} + \ldots$, substituting into

$$dA/dt = -\lambda A + 1 - \cos(t),$$

and matching terms in powers of λ^{-1}, we obtain $A_2(t) = - dA_1/dt = \sin(t)$, etc. We now substitute this perturbation series into the second ODE

$$dB/dt + B = \lambda A = A_1(t) + A_2(t)\lambda^{-1} + \ldots = 1 - \cos(t) + O(\lambda^{-1}) .$$

Solving this ODE subject to the initial condition $B(0) = 0$ using the integrating factor e^t:

$$B(t)e^t = B(0) + \int_0^t e^{t'}\left[1-\cos(t')\right]dt' + O(\lambda^{-1})$$

$$= \mathrm{Re}\int_0^t \left[e^{t'} - e^{(1+i)t'}\right]dt' + O(\lambda^{-1})$$

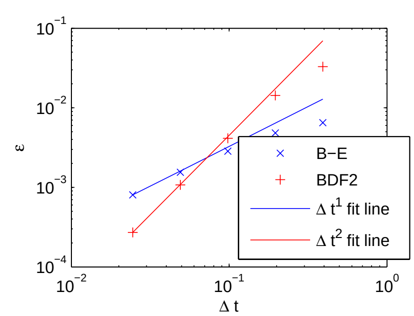
$$= e^t - 1 - \mathrm{Re}\left[\frac{e^{(1+i)t}-1}{1+i}\right] + O(\lambda^{-1})$$

Now,

$$\mathrm{Re}\left[\frac{e^{(1+i)t}-1}{1+i}\right] = \frac{1}{2}\mathrm{Re}\left[(1-i)(e^t\cos t - 1 + ie^t\sin t)\right] = \frac{1}{2}\left[e^t(\cos t + \sin t) - 1\right]$$

so $$B(t) = 1 - e^{-t} - \left[\cos t + \sin t - e^{-t}\right]/2 + O(\lambda^{-1}) = 1 - 0.5e^{-t} - 0.5\left(\cos t + \sin t\right) + O(\lambda^{-1}) .$$

1b-c) The script hw3p1bc.m generated the desired log-log plot (below) of solution increments ε vs. Δt for backward Euler method and BDF2 methods. As expected, ε is O(Δt) for BE and O(Δt^2) for BDF2. For $\Delta t >$, the BE method is more accurate, but the BDF2 method becomes superior for smaller Δt..



Both methods were coded by writing the system in the form:

$$d\boldsymbol{\varphi}/dt = \mathbf{M}\boldsymbol{\varphi} + \mathbf{s}(t),$$

where $\boldsymbol{\varphi} = \begin{bmatrix} A(t) \\ B(t) \end{bmatrix}$, $\mathbf{M} = \begin{bmatrix} -\lambda & 0 \\ \lambda & -1 \end{bmatrix}$, and $\mathbf{s}(t) = \begin{bmatrix} 1-\cos(t) \\ 0 \end{bmatrix}$.

In this format, the methods are:

BE: $$\left(\mathbf{I} - \mathbf{M}\Delta t\right)\boldsymbol{\varphi}^{n+1} = \boldsymbol{\varphi}^n + \Delta t\mathbf{s}(t^{n+1}),$$

BDF2: $$\left(3\mathbf{I} - 2\mathbf{M}\Delta t\right)\boldsymbol{\varphi}^{n+1} = 4\boldsymbol{\varphi}^n - \boldsymbol{\varphi}^{n-1} + 2\Delta t\mathbf{s}(t^{n+1}) .$$

In this problem **M** is tridiagonal, so one can sequentially solve for A^{n+1} then B^{n+1}, but I just used the Matlab Gaussian elimination function M\.

2a) The problem can be written in characteristic form as $du/dt = 0$ (u = constant) along characteristics $dx/dt = 1+9x$. Integrating the characteristic equation from a arbitrary starting time$(0,t_0)$ at which the characteristic passes through $x = 0$, we have

$$t - t_0 = \int_0^x dx'/(1+9x') = \log(1+9x)/9\,.$$

Substituting $x = 1$, and setting $\tau = \log(10)/9 = 0.256$ we conclude that

$$u(1, t_0) = u(0, t_0+\tau) = \begin{cases} 0, & t < \tau, \\ 1, & t > \tau. \end{cases}$$

Only one BC is required per characteristic, and this has been specified at $x = 0$, so no additional BC is required at $x = 1$.

2b) The stencil for the leapfrog centered-in-space method (right) includes the analytical domain of dependence (the characteristic dt/dx = c-1 through (xj,tn+1)) within the numerical domain of dependence if the Courant number $c\Delta t/\Delta x < 1$. The maximum c across the domain is c_{max} = $\max_{0\le x\le 1}(1+9x) = 10$, so the CFL stability limit is $\Delta t \le \Delta x/c_{max} = 0.005$.

2c) A second-order accurate one-sided approximation to u_x at the maximum gridpoint $j = N$ is

$$u_x(1,t_n) \approx 2\delta_{2x}^B u_N^n - \delta_{2x}^B u_N^n = \left(1.5u_N^n - 2u_N^{n-1} + 0.5u_N^{n-2}\right)/\Delta x$$

This can be seen by Taylor expanding one-sided differences of span Δx and $2\Delta x$ and finding a linear combination of these that cancels the $O(\Delta x)$ part of their truncation errors:

$$\delta_x^B u(x) = \left(u(x) - u(x-\Delta x)\right)/\Delta x = u_x - \Delta x\left(u_{xx}/2\right) + O(\Delta x^2)$$

$$\delta_{2x}^B u(x) = \left(u(x) - u(x-2\Delta x)\right)/(2\Delta x) = u_x - \Delta x\left(u_{xx}\right) + O(\Delta x^2)$$

so $$2\delta_{2x}^B u(x) - \delta_{2x}^B u(x) = u_x + O(\Delta x^2)\,.$$

2d) The Matlab script hw3p2d.m implements the leapfrog-centered in space method with a midpoint RK2 starting step, the one-sided difference at the right boundary, and optional Asselin filtering, using $\Delta x/c_{max} = 0.05$ and $\Delta t = 0.005$, producing the plots below. Asselin filtering is required to control the $2\Delta t$ errors that otherwise dominate $u(1, t)$. With Asselin filtering there is a slight overshoot (dispersion error) in the solution, which is what we would expect from the even-order spatial differencing, but the main error is diffusion of the exact step solution due to the temporal smoothing of high frequencies from the Asselin time-filtering scheme.