Homework Assignment
Numerical solution of partial differential equations, I (Course 221)
Handed out: Tuesday, September 9, 2003, Due: Tuesday, September 23, 2003

2 Numerical methods for ODE's

2.1 A very high order method

Use the operator identity

$$\frac{d}{dt} = \frac{2}{k} \operatorname{arcsinh} \frac{\delta}{2}$$

with

$$\delta = E^{1/2} - E^{-1/2} \tag{1}$$

$$\delta q(t) = q(t + k/2) - q(t - k/2) \tag{2}$$

to derive an $O(k^8)$ formula for integrating

$$\begin{aligned} q' &= f(t, q) = \sin(t + q) \\ q(t = 0) &= 1 \end{aligned}$$

from $t = 0$ to $t = 1$. Note that nothing has been specified about the time at which you should evaluate f. Try out various choices. Apply the formula with various step sizes $k = 2^{-p}$, $p = 1, 2, \ldots, 10$. Carry out a convergence analysis, i.e. compute the relative error at $t = 1$ and draw a $\lg - \lg$ plot of the relative error versus step size. Analyze the plot. Is the convergence as you would expect from your derivation of the formula?

2.2 But is it stable?

Verify the zero-convergence of the above method for various choices of when to evaluate $f(t, q)$ such as at the least time that appears in the approximation of q', the greatest time or the average of these two.

2.3 Is it stable for practical computations?

Now verify the absolute stability of the above algorithms. Use the boundary locus method and comment on what you obtain. Reinterpret your results from the first problem in light of what you learn from the absolute stability analysis.

2.4 Extension to PDE's

Write down the algorithm obtained by using a second-order, centered finite difference approximation for the spatial derivative and the approximation of the time derivative from problem 1 for the problem

$$\left\{\begin{array}{c} q_t = q_{xx} \\ q(x,0) = \sin(\pi x) \\ q(0,t) = 0,\ q(1,t) = 0 \end{array}\right. \tag{3}$$

on $(x,t) \in [0,1] \times [0,1]$. Draw the stencil of the resulting algorithm. Think of a way to start the computation. Check the stability. Try out a few computations with decreasing step sizes and comment on whether the results converge or not.

2.5 Bonus - Computer generated algorithms

The formal series for deriving finite difference approximations

$$\frac{d}{dt} = \frac{1}{k}\left(\Delta_+ - \frac{1}{2}\Delta_+^2 + \frac{1}{3}\Delta_+^3 - \ldots\right) = \frac{1}{k}\left(\Delta_- + \frac{1}{2}\Delta_-^2 + \frac{1}{3}\Delta_-^3 + \ldots\right) \tag{4}$$

is well suited to computer implementation using a symbolic processing package (Mathematics, Maple). Try to write a program that carries out an $O(k^m)$ in time, $O(h^n)$ in space, finite difference approximation for the IVBP from the previous problem for arbitrary m, n. Most symbolic packages have routines to directly output Fortran or C code; try to do this for your finite difference method. Also, it should be possible to carry out zero-stability and absolute stability analysis in the symbolic program.