Technische Universität Berlin **Summer Term 2014**
Fakultät II - Mathematik und Naturwissenschaften
Institut für Mathematik
Andreas Steinbrecher, Jeroen Stolwijk

Numerical Analysis II

Homework Sheet 9

Exercises

Tutorial on June 23

1. Problem
Solve the homogeneous linear difference equation $u_{m+2} - u_{m+1} - u_m = 0$, $u_0 = 0$, $u_1 = 1$. How is this sequence called?

2. Problem
Determine the parameter α in the multi-step method

$$u_{m+2} - (1+\alpha)u_{m+1} + \alpha u_m = \frac{h}{12}\left((5+\alpha)f_{m+2} + 8(1-\alpha)f_{m+1} - (1+5\alpha)f_m\right)$$

such that the method is zero-stable. What is the maximal order of consistency that we can achieve?

3. Problem
By the first Dahlquist barrier, an explicit, zero-stable linear k-step method is at most consistent of order k. Consider the explicit two-step method

$$\sum_{l=0}^{2} \alpha_l u_{m+l} = h \sum_{l=0}^{1} \beta_l f_{m+l}$$

and show that if this method is zero-stable, its order cannot exceed 2, i.e., prove that the first Dahlquist barrier holds here.

Theoretical Homework

Due: July 1, during the lecture

1. Problem (9 Points)
Determine the largest possible interval $I \subseteq \mathbb{R}$ such that the explicit, linear three-step method

$$u_{m+3} + \alpha(u_{m+2} - u_{m+1}) - u_m = \frac{h}{2}(3+\alpha)(f_{m+2} + f_{m+1})$$

is zero-stable for all $\alpha \in I$. Further show that there is an α, for which the above method is consistent of order $p = 4$, but that a zero-stable method of the above type is at most of order $p = 2$.

2. Problem (8 Points)
Consider the difference equation

$$u_{m+2} - 2zu_{m+1} - u_m = 0$$

with $z \in \mathbb{C}$, that corresponds to the explicit midpoint rule. Compute the region $G \subseteq \mathbb{C}$ such that the solution of the difference equation is bounded for all $z \in G$. Although a MATLAB solution (code and plot) will be accepted, full points will only be awarded for an algebraic solution of this problem.

3. Problem (8 Points)
Is a consistent linear two-step method with $\alpha_0 = \beta_0$ A-stable? Explain why. The use of MATLAB is allowed if the code and the plots are provided.

Total Points: 25

Programming Homework

Due: June 30 (first chance) or July 7 (second chance)

Attention: You can only submit your program on July 7 if you presented a programming approach on June 30!

Write a program that solves an ordinary differential equation $\dot{y}(t) = f(t, y(t)),\ y(t_0) = y_0$ on an interval $[t_0, t_0 + a]$ using a predictor-corrector method.

Idea: The aim of a predictor-corrector method is to combine an explicit with an implicit multi-step method without solving implicit equations. Hence, for each step an explicit method 'predictor' is used to compute a first approximation, e.g.

$$u_{m+3}^{(0)} = u_{m+2} + \frac{h}{12}\Big(23f_{m+2} - 16f_{m+1} + 5f_m\Big) \qquad \textit{(Adams-Bashforth)}$$

and then an additional M approximations are computed with an implicit method 'corrector' using the 'old' value in the right hand side of the method, e.g.

$$u_{m+3}^{(j)} = u_{m+2} + \frac{h}{24}\Big(9f(t_{m+3}, u_{m+3}^{(j-1)}) + 19f_{m+2} - 5f_{m+1} + f_m\Big) \qquad \textit{(Adams-Moulton)}$$

for $j = 1, \ldots, M$. Before starting with the next step one would let $u_{m+3} := u_{m+3}^{(M)}$ be the final approximation and evaluate $f_{m+3} := f(t_{m+3}, u_{m+3}^{(M)})$.

Write a routine that implements the above described Adams-Bashforth/Adams-Moulton method

$$[\mathtt{h}, \mathtt{t}, \mathtt{u}] = \mathtt{adbaadmo}(\mathtt{fun}, \mathtt{t0}, \mathtt{y0}, \mathtt{N}, \mathtt{a}, \mathtt{i})$$

as well as another routine that uses the Nyström/Milne-Simpson method

$$[\mathtt{h}, \mathtt{t}, \mathtt{u}] = \mathtt{nymisi}(\mathtt{fun}, \mathtt{t0}, \mathtt{y0}, \mathtt{N}, \mathtt{a}, \mathtt{i})$$

given by

$$u_{m+3} = u_{m+1} + \frac{h}{3}\Big(7f_{m+2} - 2f_{m+1} + f_m\Big) \qquad \text{(Nyström)}$$

and

$$u_{m+3} = u_{m+1} + \frac{h}{3}\Big(f_{m+3} + 4f_{m+2} + f_{m+1}\Big) \qquad \text{(Milne-Simpson).}$$

Here, `fun` should be a MATLAB function handle corresponding to the right hand side $f(t, y)$ of the differential equation. It should also be possible for `y` and `f` to be vectors of $\mathbb{R}^n$. The parameter $\mathtt{t0} = t_0$ is the lower interval bound, $\mathtt{y0} = y_0 \in \mathbb{R}^n$ is the initial value, `N` is the number of steps, $\mathtt{a} = a$ is the interval length, and `i` indicates the method used to compute a sufficient number of start-up steps. For $\mathtt{i} = 1$, the Forward Euler method should be used and for $\mathtt{i} = 2$, you should take the classical Runge-Kutta method.
Further, the number of corrector steps is fixed to be $M = 2$ in both routines. The routine should return the step size $\mathtt{h} = \mathtt{a}/\mathtt{N}$, the vector of grid points $\mathtt{t} = [t_0, t_1, \ldots, t_N]$, and the corresponding approximated solution $\mathtt{u} = [u_0, u_1, \ldots, u_N]$.
The methods are to be implemented such that the computational effort, in particular the number of evaluations of $f(t, y)$, is minimized.
To test your program, use the initial value problems

1. $\dot{y}(t) = 2y(t) - e^t, \quad y(0) = 2, \quad t \in [0, 1],$
2. $\dot{y}(t) = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} y(t), \quad y(0) = [1\ 0]^T, \quad t \in [0, 1]\ ,$
3. $\dot{y}(t) = -\tan(t)y(t), \quad y(0) = 1, \quad t \in [0, 3],$

and compare both routines for $\mathtt{i} = 1, 2$ in terms of accuracy and computational complexity. In particular, make a log-lot plot of the error for the first test problem for all **four** methods and the classical Runge-Kutta method using the values $N = 10, 20, 40, 80, 160, 320$. Which methods show the best trade-off between complexity and accuracy?