P. Grohs
S. Hosseini
Ž. Kereta

Numerical Analysis II

ETH Zürich
D-MATH

Problem Sheet 3

Problem 3.1 Implementation of Semi-Explicit Methods

In implicit methods, the numerical approximation of the IVP is computed by solving a generally non-linear system of equations. If this system is solved using only one step of the Newton's method the underlying numerical method is called semi-implicit.

(3.1a) Let $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n$ be a function. Create a MATLAB function `newton(x0, F, DF)`, which carries out one step of Newton's method for the solution of $\mathbf{F}(x) = 0$. The inputs of the function are the starting value `x0`, and the function fandles `F` and `DF` representing the function $\mathbf{F}$ and its Jacobian $\mathbf{D_y F}$.

(3.1b) Implement the semi-implicit Euler method as follows: transform the implicit Euler method [NUMODE, Eq. (1.4.13)] into a root-finding problem and apply your implementation of the Newton method from subproblem (3.1a). The inputs of this MATLAB function are initial value `y0`, the right hand side `f`, its Jacobian `Df`, the end time `T`, and the number of discretisation steps `Nh`. Use the template `impEul.m`.

(3.1c) Implement the semi-implicit mid-point rule as follows: transform the implicit mid-point rule [NUMODE, Eq. (1.4.19)] into a root-finding problem and apply your implementation of the Newton method from subproblem (3.1a). The inputs of this MATLAB function are initial value `y0`, the right hand side `f`, its Jacobian `Df`, the end time `T`, and the number of discretisation steps `Nh`. Use the template `impEul.m`. Use the template `impMpr.m`.

(3.1d) Consider the IVP

$$\dot{\mathbf{y}} = \exp(\mathbf{y}) \sin(\mathbf{y}); \quad \mathbf{y}(0) = \pi/4.$$

Determine the absolute errors of both the semi-implicit Euler method and the semi-implicit mid-point rule at time `T=0.5` for $N_h = 2^i$ steps, where $i = 4, \ldots, 9$. Plot the two error curves versus the number of steps in a suitable axis scale and determine the algebraic convergence rate using the MATLAB function `polyfit`. Use the template `impConv.m`.

HINT: You can determine the reference solution using `ode45`. Set the relative und absolute tolerances to 10^{-12}.

Problem 3.2 Differential of a Consistent Evolution

Let $\boldsymbol{\Phi}^h$ be the exact and $\boldsymbol{\Psi}^h$ the consistent discrete evolution for the autonomous ODE $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$. Furthermore, we assume that all maps are sufficiently smooth.

(3.2a) Show that

$$\boldsymbol{\Phi}^h(\mathbf{y}_0+\mathbf{z}) = \boldsymbol{\Phi}^h\mathbf{y}_0+\mathbf{z}+D_{\mathbf{y}}\mathbf{f}(\mathbf{y}_0)h\mathbf{z}+\mathbf{r}(h,\mathbf{z}),$$

where

$$\|\mathbf{r}(h,\mathbf{z})\| \leq C(h^2\|\mathbf{z}\|+\|\mathbf{z}\|^2),$$

with a constant $C>0$ that is independent of h and $\mathbf{z}$.

HINT: Use the variational equation [NUMODE, Thm. 1.3.29] for the Wronski matrix.

(3.2b) Show that

$$\boldsymbol{\Psi}^h(\mathbf{y}_0+\mathbf{z}) = \boldsymbol{\Psi}^h\mathbf{y}_0+\mathbf{z}+D_{\mathbf{y}}\mathbf{f}(\mathbf{y}_0)h\mathbf{z}+\tilde{\mathbf{r}}(h,\mathbf{z}),$$

where

$$\|\tilde{\mathbf{r}}(h,\mathbf{z})\| \leq C(h^2\|\mathbf{z}\|+\|\mathbf{z}\|^2),$$

with a constant $C>0$ independent of h and $\mathbf{z}$.

HINT: Use (3.2a) and consistency.

Problem 3.3 Convergence Order of the Explicit Mid-Point Rule

We want to numerically estimate the convergence order of the explicit mid-point rule using the IVP

$$\dot{y} = ty+t^3, \quad y(0)=0, \tag{3.3.1}$$

for which the exact solution is given by

$$y(t) = 2\mathrm{e}^{t^2/2}-t^2-2. \tag{3.3.2}$$

(3.3a) Complete the implementation of the explicit mid-point rule

$$y_{k+1} = y_k + hf\left(t_k+\frac{h}{2}, y_k+\frac{h}{2}f(t_k,y_k)\right), \quad k\geq 0.$$

using the MATLAB template `ExpMidPoint.m`.

(3.3b) Implement the right hand side and the solution of the initial value problem (3.3.1) in the MATLAB files `rhs.m` and `sol.m`. Plot the exact solution (3.3.2) as well as the approximation from the explicit mid-point rule and step size $h=0.2$ up to end point $T=1$ in the same figure.

(3.3c) Complete the function `ExpMidPointConv.m` and plot the global error of the explicit mid-point rule at the end-point $T=1$ as a function of the step size $h=2^{-k}$, $k=2,3,\ldots,8$ on a double-logarithmic scale. Compute the convergence order of the method using the MATLAB function `polyfit`. Save the figure as `a1conv.eps`. What is the convergence order of the method?

(3.3d) Explain how can one read the convergence order of the method straight off the plot obtained in subproblem (3.3c).

Published on 3 March 2015.

To be submitted by 10 March 2015.

References

[NUMODE] Lecture Slides for the course "Numerical Methods for Ordinary Differential Equations", SVN revision # 63606.

Last modified on March 4, 2015