Spring Term 2016

Numerical Analysis II

Problem Sheet 5

Problem 5.1 Collocation vs. Quadrature rules

Observe the quadrature rule on $[0,1]$ of order p,

$$\sum_{i=0}^{s} b_i f(c_i) \approx \int_0^1 f(\xi)\,\mathrm{d}\xi,$$

with sample points c_i and weights b_i, $i = 1, \ldots, s$. So for all polynomials $f \in \mathcal{P}_p \setminus \mathcal{P}_{p-1}$

$$\sum_{i=0}^{s} b_i f(c_i) \neq \int_0^1 f(\xi) d\xi$$

holds.

The quadrature rule gives us a collocation method for the scalar differential equation $\dot{y} = f(t)$, $y(0) = y_0$ by

$$\begin{aligned} y_h(0) &= y_0 \\ y_h(t_{k+1}) &= y_h(t_k) + h \sum_{i=0}^{s} b_i f(hc_i). \end{aligned}$$

(5.1a) Show that the exact solution of the differential equation at the point h can be written as

$$y(h) = y_0 + h \int_0^1 f(\xi h)\,\mathrm{d}\xi.$$

Solution: We integrate the differential equation up to the point h,

$$\int_0^h \dot{y}(t)\,\mathrm{d}t = \int_0^h f(t)\,\mathrm{d}t,$$

and, using the substitution $t = h\xi$, receive

$$y(t) = y_0 + h \int_0^1 f(\xi h)\,\mathrm{d}\xi.$$

(5.1b) Let f be sufficiently smooth. Show, that the explicit collocation method cannot be of order $p+1$.

HINT: Observe

$$\frac{\partial^n}{\partial h^n}\left(\frac{1}{h}\tau(0,y,h)\right)\Big|_{h=0}$$

where $\tau(0,y,h)$ is the consistency error after the first step

$$\tau(0,y,h) = (\Phi^{0,h} - \Psi^{0,h})y.$$

Solution: With the exact evolution

$$\Phi^{0,h} := y_0 + h\int_0^1 f(\xi h)\,\mathrm{d}\xi$$

and the numeric evolution

$$\Psi^{0,h} := y_h(t_k) + h\sum_{i=0}^{s} b_i f(hc_i),$$

the consistency error of the collocation method is

$$\begin{aligned}\tau(0,y,h) &= (\Phi^{0,h} - \Psi^{0,h})y \\ &= h\left(\sum_{i=0}^{s} b_i f(hc_i) - \int_0^1 f(\xi h)\,\mathrm{d}\xi\right)\end{aligned}$$

If we look at the n^{th} derivative of $\tau(0,y,h)/h$ at the point $h=0$

$$\begin{aligned}\frac{\partial^n}{\partial h^n}\left(\frac{1}{h}\tau(0,y,h)\right)\Big|_{h=0} &= \frac{\partial^n}{\partial h^n}\left(\sum_{i=0}^{s} b_i f(hc_i) - \int_0^1 f(\xi h)\,\mathrm{d}\xi\right)\Big|_{h=0} \\ &= \left(\sum_{i=0}^{s} b_i c_i^n f^{(n)}(hc_i) - \int_0^1 \xi^n f^{(n)}(\xi h)\,\mathrm{d}\xi\right)\Big|_{h=0} \\ &= f^{(n)}(0)\underbrace{\left(\sum_{i=0}^{s} b_i c_i^n - \int_0^1 \xi^n\,\mathrm{d}\xi\right)}_{\substack{=0 \quad \text{for } n\le p \\ \neq 0 \quad \text{for } n>p}},\end{aligned}$$

we see, that $\tau(0,y,h)$ vanishes for $n\le p$, but not for $n>p$ and the collocation method can thus not be of order $p+1$.

Problem 5.2 Linear Independence of Elementary Differentials

Show that the elementary differentials

$$f(\mathbf{y}_0),\quad Df(\mathbf{y}_0)f(\mathbf{y}_0),\quad Df(\mathbf{y}_0)Df(\mathbf{y}_0)f(\mathbf{y}_0),\quad D^2f(\mathbf{y}_0)(f(\mathbf{y}_0),f(\mathbf{y}_0))$$

(cf. example [NUMODE, Ex. 2.3.24]) are linearly independent as $\mathcal{C}^2(\mathrm{U}_\epsilon(\mathbf{y}_0),\mathbb{R}^2)$ mappings. In other words, show that if

$$\alpha_1 f(\mathbf{y}_0) + \alpha_2 Df(\mathbf{y}_0)f(\mathbf{y}_0) + \alpha_3 Df(\mathbf{y}_0)Df(\mathbf{y}_0)f(\mathbf{y}_0) + \alpha_4 D^2f(\mathbf{y}_0)(f(\mathbf{y}_0),f(\mathbf{y}_0)) = 0,$$

holds for all $f \in \mathcal{C}^2(\mathrm{U}_\epsilon(\mathbf{y}_0), \mathbb{R}^2)$, then it follows that $\alpha_i = 0$ for all $i = 1, \ldots, 4$.

Solution: The elementary differentials

$$f(\mathbf{y}), \quad Df(\mathbf{y})f(\mathbf{y}), \quad Df(\mathbf{y})Df(\mathbf{y})f(\mathbf{y}), \quad D^2f(\mathbf{y})(f(\mathbf{y}), f(\mathbf{y}))$$

are linearly independent if and only if

$$\alpha_1 f(\mathbf{y}) + \alpha_2 Df(\mathbf{y})f(\mathbf{y}) + \alpha_3 Df(\mathbf{y})Df(\mathbf{y})f(\mathbf{y}) + \alpha_4 D^2f(\mathbf{y})(f(\mathbf{y}), f(\mathbf{y})) = 0 \tag{5.2.1}$$

can be satisfied for all $f \in \mathcal{C}^2(\mathrm{U}_\epsilon(\mathbf{y}_0), \mathbb{R}^2)$, only if $\alpha_i = 0$, $i = 1, \ldots, 4$.

In terms of components, the elementary differentials $Df(\mathbf{y})$ and $D^2f(\mathbf{y})$ have the representations

$$Df(\mathbf{y})(\mathbf{w}) = \begin{pmatrix} \partial_1 f_1(\mathbf{y})w_1 + \partial_2 f_1(\mathbf{y})w_2 \\ \partial_1 f_2(\mathbf{y})w_1 + \partial_2 f_2(\mathbf{y})w_2 \end{pmatrix}$$

and

$$D^2f(\mathbf{y})(\mathbf{w}, \mathbf{z}) = \begin{pmatrix} \sum_{j=1}^2 \sum_{k=1}^2 \partial_j\partial_k f_1(\mathbf{y})w_j z_k \\ \sum_{j=1}^2 \sum_{k=1}^2 \partial_j\partial_k f_2(\mathbf{y})w_j z_k \end{pmatrix}.$$

We will now show that Equation 5.2.1 is only satisfied for all f, if $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$. To this end, we choose concrete functions f, such that higher elementary differentials vanish and conclude successively that $\alpha_i = 0$.

- α_1: If we choose
$$f(\mathbf{y}) = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$
then $f(\mathbf{y}) \neq \mathbf{0}$, $Df(\mathbf{y}) = \mathbf{0}$, $D^2f(\mathbf{y}) = \mathbf{0}$ and therefore $\alpha_1 = 0$.

- α_2: Choose
$$f(\mathbf{y}) = f\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 \\ y_1 \end{pmatrix}, \qquad y_1 \neq 0,$$
to obtain
$$Df(\mathbf{y}) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$
Thus,
$$Df(\mathbf{y})f(\mathbf{y}) = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$
and
$$Df(\mathbf{y})Df(\mathbf{y}) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \qquad D^2f(\mathbf{y}) = \mathbf{0}.$$
We conclude that $\alpha_2 = 0$.

- α_3: The choice $f(\mathbf{y}) = \mathbf{y}$, $\mathbf{y} \neq 0$ yields $Df(\mathbf{y}) = I$, $D^2f(\mathbf{y}) = \mathbf{0}$ and hence $\alpha_3 = 0$.

- α_4: We choose

$$f(\mathbf{y}) = f\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} y_1^2 \\ 0 \end{pmatrix}, \qquad y_1 \neq 0,$$

  to obtain

$$\frac{\partial}{\partial y_1} f(\mathbf{y}) = \begin{pmatrix} 2y_1 \\ 0 \end{pmatrix}, \quad D\Big(\frac{\partial}{\partial y_1} f(\mathbf{y})\Big) = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}, \quad D\Big(\frac{\partial}{\partial y_1} f(\mathbf{y})\Big) f(\mathbf{y}) = \begin{pmatrix} 2y_1^2 \\ 0 \end{pmatrix},$$

  and

$$\frac{\partial}{\partial y_2} f(\mathbf{y}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad D\Big(\frac{\partial}{\partial y_2} f(\mathbf{y})\Big) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad D\Big(\frac{\partial}{\partial y_2} f(\mathbf{y})\Big) f(\mathbf{y}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

  This implies

$$D^2 f(\mathbf{y})(f(\mathbf{y}), f(\mathbf{y})) = \begin{pmatrix} 2y_1^4 \\ 0 \end{pmatrix} \neq \mathbf{0},$$

  and therefore $\alpha_4 = 0$.

Problem 5.3 The Implicit Midpoint Method for Non-Expansive Evolutions

The implicit midpoint method is a RK method and was previously introduced in [NODE, Ex. 2.1.5]. For the following autonomous initial value problem

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}), \qquad \mathbf{y}(0) = \mathbf{y}_0. \tag{5.3.1}$$

the defining equations of the midpoint method are

$$\mathbf{y}_1 = \mathbf{y}_0 + h\mathbf{f}\big(\frac{1}{2}(\mathbf{y}_1 + \mathbf{y}_0)\big). \tag{5.3.2}$$

By [NODE, Thm. 2.2.19] we know that (5.3.2) solvable with respect to $\mathbf{y}_1$ given that h is small enough and $\mathbf{f}$ is locally Lipschitz continuous.

We assume that $\mathbf{f} : \mathbb{R}^d \mapsto \mathbb{R}^d$ is a *dissipative vector field* ($\rightarrow$ [NUMODE, Def. 3.3.5], also [NODE, Rem. 2.2.16]) which satisfies a *global* Lipschitz condition.

The aim of this question is to show that the discrete evolution corresponding to the implicit midpoint method applied to (5.3.1) exists for every $h > 0$.

(5.3a) Convert (5.3.2) into the problem of finding the zeros of $\mathbf{G}(\mathbf{y}_1) = 0$ for an appropriate function $G : \mathbb{R}^d \mapsto \mathbb{R}^d$.

Solution: Let $\mathbf{G} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be defined by

$$\mathbf{G}(\mathbf{x}) := \mathbf{x} - \mathbf{y}_0 - h\mathbf{f}\big(\frac{1}{2}(\mathbf{x} + \mathbf{y}_0)\big).$$

Then the converted problem is: find $\mathbf{x} \in \mathbb{R}^d$ such that $\mathbf{G}(\mathbf{x}) = 0$.

(5.3b) Show that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$

$$\big(\mathbf{G}(\mathbf{x}) - \mathbf{G}(\mathbf{y})\big)(\mathbf{x} - \mathbf{y}) \geq \|\mathbf{x} - \mathbf{y}\|^2,$$

in which $\mathbf{G}$ is the function from subproblem (5.3a).

Solution:

$$\begin{aligned}
\big(\mathbf{G}(\mathbf{x}) - \mathbf{G}(\mathbf{y})\big)(\mathbf{x} - \mathbf{y}) &= \big(\mathbf{x} - \mathbf{y}_0 - h\mathbf{f}\big(\tfrac{1}{2}(\mathbf{x} + \mathbf{y}_0)\big) - \mathbf{y} + \mathbf{y}_0 + h\mathbf{f}\big(\tfrac{1}{2}(\mathbf{y} + \mathbf{y}_0)\big)\big)(\mathbf{x} - \mathbf{y}) \\
&= \big(\mathbf{x} - h\mathbf{f}\big(\tfrac{1}{2}(\mathbf{x} + \mathbf{y}_0)\big) - \mathbf{y} + h\mathbf{f}\big(\tfrac{1}{2}(\mathbf{y} + \mathbf{y}_0)\big)\big)\big)(\mathbf{x} - \mathbf{y}) \\
&= \|\mathbf{x} - \mathbf{y}\|^2 - 2h \underbrace{\big(\mathbf{f}\big(\tfrac{1}{2}(\mathbf{x} + \mathbf{y}_0)\big) - \mathbf{f}\big(\tfrac{1}{2}(\mathbf{y} + \mathbf{y}_0)\big)\big)\big(\tfrac{1}{2}(\mathbf{x} + \mathbf{y}_0) - \tfrac{1}{2}(\mathbf{y} + \mathbf{y}_0)\big)\big)}_{<0,\text{ because } \mathbf{f} \text{ dissipativ}} \\
&\geq \|\mathbf{x} - \mathbf{y}\|^2.
\end{aligned}$$

(5.3c) Show that $\mathbf{G}$ is globally Lipschitz continuous on $\mathbb{R}^d$.

Solution: Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, then

$$\begin{aligned}
\big\|\mathbf{G}(\mathbf{x}) - \mathbf{G}(\mathbf{y})\big\| &= \Big\|\mathbf{x} - \mathbf{y}_0 - h\mathbf{f}\big(\tfrac{1}{2}(\mathbf{x} + \mathbf{y}_0)\big) - \mathbf{y} + \mathbf{y}_0 + h\mathbf{f}\big(\tfrac{1}{2}(\mathbf{y} + \mathbf{y}_0)\big)\Big\| \\
&= \Big\|\mathbf{x} - h\mathbf{f}\big(\tfrac{1}{2}(\mathbf{x} + \mathbf{y}_0)\big) - \mathbf{y} + h\mathbf{f}\big(\tfrac{1}{2}(\mathbf{y} + \mathbf{y}_0)\big)\Big\| \\
&\leq \big\|\mathbf{x} - \mathbf{y}\big\| + h \underbrace{\Big\|\mathbf{f}\big(\tfrac{1}{2}(\mathbf{x} + \mathbf{y}_0)\big) - \mathbf{f}\big(\tfrac{1}{2}(\mathbf{y} + \mathbf{y}_0)\big)\Big\|}_{\leq \frac{L}{2}\|\mathbf{x}-\mathbf{y}\|,\text{ because } \mathbf{f} \text{ is globally Lipschitz continuous}} \\
&\leq (1 + \frac{hL}{2})\big\|\mathbf{x} - \mathbf{y}\big\|.
\end{aligned}$$

(5.3d) Prove the following Lemma, which guarantees the solvability of (5.3.2) with respect to $\mathbf{y}_1$.

Lemma: *Let $\mathbf{G} : \mathbb{R}^d \to \mathbb{R}^d$ be a globally Lipschitz continuous function with the following property: there exists a $\gamma > 0$, such that*

$$\big(\mathbf{G}(\mathbf{x}) - \mathbf{G}(\mathbf{y})\big)(\mathbf{x} - \mathbf{y}) \geq \gamma\|\mathbf{x} - \mathbf{y}\|^2, \tag{5.3.3}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. Then $\mathbf{G}(\mathbf{x}) = 0$ has a unique solution.

HINT: Let $\boldsymbol{\Phi}(\mathbf{x}) := \mathbf{x} - \alpha\mathbf{G}(\mathbf{x})$, for some $\alpha > 0$. First show that $\mathbf{x} \in \mathbb{R}^d$ is a unique solution of $\mathbf{G}(\mathbf{x}) = 0$ if and only if $\boldsymbol{\Phi}(\mathbf{x}) = \mathbf{x}$. Then show that for α small enough $\boldsymbol{\Phi}$ is a contraction on $\mathbb{R}^d$ and apply the Banach fixed-point theorem.

Solution: If $\mathbf{G}(\mathbf{x}) = 0$, then $\boldsymbol{\Phi}(\mathbf{x}) = \mathbf{x} - \alpha\mathbf{G}(\mathbf{x}) = \mathbf{x}$.
If $\boldsymbol{\Phi}(\mathbf{x}) = \mathbf{x}$, then $\mathbf{x} - \alpha\mathbf{G}(\mathbf{x}) = \mathbf{x}$. Notice $\alpha > 0$ implies $\mathbf{G}(\mathbf{x}) = 0$.

Now we show that for α small enough $\boldsymbol{\Phi}$ is a contraction on $\mathbb{R}^d$. Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, then

$$\begin{aligned}
\|\boldsymbol{\Phi}(\mathbf{x}) - \boldsymbol{\Phi}(\mathbf{y})\|^2 &= \|\mathbf{x} - \alpha\mathbf{G}(\mathbf{x}) - \mathbf{y} + \alpha\mathbf{G}(\mathbf{y})\|^2 \\
&= \|\mathbf{x}-\mathbf{y}\|^2 - 2\alpha\big(\mathbf{G}(\mathbf{x}) - \mathbf{G}(\mathbf{y})\big)(\mathbf{x}-\mathbf{y}) + \alpha^2 \underbrace{\|\mathbf{G}(\mathbf{x}) - \mathbf{G}(\mathbf{y})\|^2}_{\leq L^2\|\mathbf{x}-\mathbf{y}\|^2} \\
&\overset{(5.3.3)}{\leq} \|\mathbf{x}-\mathbf{y}\|^2 - 2\alpha\gamma\|\mathbf{x}-\mathbf{y}\|^2 + \alpha^2 L^2\|\mathbf{x}-\mathbf{y}\|^2 \\
&= (1 - 2\alpha\gamma + \alpha^2 L^2)\|\mathbf{x}-\mathbf{y}\|^2.
\end{aligned}$$

Since

$$\begin{aligned}
1 - 2\alpha\gamma + \alpha^2 L^2 < 1 &\iff \alpha^2 L^2 - 2\alpha\gamma < 0 \\
&\iff \alpha(\alpha L^2 - 2\gamma) < 0 \\
&\iff \alpha < \frac{2\gamma}{L^2},
\end{aligned}$$

it follows that for α sufficiently small $\boldsymbol{\Phi}$ is a contraction on $\mathbb{R}^d$. Now because $\mathbb{R}^d$ is a complete metric space $\boldsymbol{\Phi}$ satisfies all requirements of the Banach fixed-point theorem and we therefore get a unique fixed-point.

Problem 5.4 Implicit Trapezoidal Rule

The implicit trapezoidal rule for solving the autonomous differential equation $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$ is given by

$$\mathbf{y}_1 = \Psi^{t_0,t_0+h} := \mathbf{y}_0 + \tfrac{1}{2}h[\mathbf{f}(\mathbf{y}_0) + \mathbf{f}(\mathbf{y}_1)]. \tag{5.4.1}$$

(5.4a) The method (5.4.1) can be interpreted as a Runge-Kutta-method. Produce the corresponding Butcher-tableau.

Solution: The rule can be written as

$$\mathbf{y}_1 = \mathbf{y}_0 + \tfrac{1}{2}h(\mathbf{k}_1 + \mathbf{k}_2)$$

with steps

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{f}(t_0, \mathbf{y}_0) \\
\mathbf{k}_2 &= \mathbf{f}\big(t_0 + h, \mathbf{y}_0 + \tfrac{1}{2}h(\mathbf{k}_1 + \mathbf{k}_2)\big).
\end{aligned}$$

The corresponding Butcher-tableau is thereby

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

(5.4b) The method (5.4.1) can also be interpreted as a polynomial collocation method. What are the corresponding collocation points

Solution:

The collocation points are $c_1 = 0$ and $c_2 = 1$. In effect the entries of above Butcher-tableau follow from these points by the known formulas:

$$a_{11} = \int_0^0 L_1(\tau)\,\mathrm{d}\tau = 0, \qquad a_{12} = \int_0^0 L_2(\tau)\,\mathrm{d}\tau = 0,$$
$$b_1 = a_{21} = \int_0^1 L_1(\tau)\,\mathrm{d}\tau = \tfrac{1}{2}, \qquad b_2 = a_{22} = \int_0^1 L_2(\tau)\,\mathrm{d}\tau = \tfrac{1}{2}.$$

Here $L_1(\tau) = 1 - \tau$ and $L_2(\tau) = \tau$ are the Langrangian polynomials to the collocation points c_1 and c_2 respectively.

Problem 5.5 Construction of One-Step Methods

This exercise studies the so called *Taylor series method* to construct one-step methods. For a given autonomous system $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$, we define a family of discrete evolutions $\Psi_n^{t,s}$, $n \in \mathbb{N}$ by

$$\Psi_n^{t,t+h}\mathbf{z} := \mathbf{z} + \sum_{i=1}^{n} \frac{h^i}{i!}\Delta_i \mathbf{z}, \qquad t, t+h \in J(\mathbf{z}),$$

where

$$\Delta_i \mathbf{z} := \frac{\mathrm{d}^{i-1}}{\mathrm{d}s^{i-1}}\mathbf{f}(\mathbf{y}(s))\Big|_{s=t} \qquad \text{with} \qquad \frac{\mathrm{d}}{\mathrm{d}s}\mathbf{y}(s) = \mathbf{f}(\mathbf{y}(s)), \quad \mathbf{y}(t) = \mathbf{z}.$$

(5.5a) Show that the discrete evolutions $\Psi_n^{t,s}$ are consistent with the autonomous system $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$.
HINT: Apply [NODE, Lemma. 2.1.10].

Solution: By [NODE, Lemma. 2.1.10], these evolutions are consistent with the autonomous system $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$, as the increment functions

$$h \mapsto \psi_n(\mathbf{z}, h) := \sum_{i=1}^{n} \frac{h^{i-1}}{i!}\Delta_i \mathbf{z}$$

are continuous and $\psi_n(\mathbf{z}, 0) = \mathbf{f}(\mathbf{z})$.

(5.5b) Express Δ_1, Δ_2 and Δ_3 symbolically in relation to the (sufficiently smooth) right hand side $\mathbf{f}$ and its derivatives.

Solution: By use of the chain rule we receive

$$\begin{aligned}
\Delta_1 \mathbf{z} &= \mathbf{f}(\mathbf{z}),\\
\Delta_2 \mathbf{z} &= (\mathrm{D}_{\mathbf{y}}\mathbf{f})(\mathbf{z})\,\mathbf{f}(\mathbf{z}),\\
\Delta_3 \mathbf{z} &= (\mathrm{D}^2_{\mathbf{y}}\mathbf{f})(\mathbf{z})\big(\mathbf{f}(\mathbf{z}), \mathbf{f}(\mathbf{z})\big) + \big((\mathrm{D}_{\mathbf{y}}\mathbf{f})(\mathbf{z})\big)^2\,\mathbf{f}(\mathbf{z}).
\end{aligned}$$

(5.5c) Find the order of consistency $\boldsymbol{\tau}(t, \mathbf{y}, h)$ of these methods for autonomous systems.
Solution: These methods are constructed upon the Taylor series of the analytic solution. As $\Psi_n^{t,t+h}\mathbf{z}$ contains the first n terms of the Taylor series, the methods have consistency order n.

(5.5d) We will now look at the scalar logistic differential equation

$$\dot{y} = 2y(1-y). \tag{5.5.1}$$

Write three MATLAB functions

```
function y = trv_1(h,T,y0)
function y = trv_2(h,T,y0)
function y = trv_3(h,T,y0)
```

in which, for a given step size `h`, end point `T` and initial value `y0`, you calculate the solution to (5.5.1) using Ψ_1, Ψ_2 and Ψ_3 respectively.

Solution: For the logistic differential equation with right hand side $f(y) = 2(y - y^2)$, we have

$$\begin{aligned}
\Delta_1 z &= f(z) \\
&= 2z - 2z^2 \\
\Delta_2 z &= f'(z)f(z) \\
&= (2-4z)(2z-2z^2) \\
\Delta_3 z &= f''(z)f(z)^2 + f'(z)^2 f(z) \\
&= -4(2z-2z^2)^2 + (2-4z)^2(2z-2z^2).
\end{aligned}$$

For the implementation of the Taylor series method see Listing 5.1, Listing 5.2 and Listing 5.3 respectively.

Listing 5.1: `trv_1`

```
function y = trv_1(h,T,y0)

N = round(T/h);
% or with %%%%%%%%%%%%%%%%%%%%%%%%%
%   N = ceil(T/h)
%   h = T/N
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%preallocation
y = zeros(1,N+1);

% initial value y0
y(1) = y0;

%define the functions f, Df, DDf
f=@(y) 2*(y-y^2);
Df=@(y) 2*(1-2*y);
DDf=@(y) -4;

% for-loop for the Taylor series method
```

```
for k=1:N
    y(k+1) = y(k) + h*f(y(k));
end

%save approximated solution
y=y(end);

end
```

Listing 5.2: `trv_2`

```
function y = trv_2(h,T,y0)

N = round(T/h);

%preallocation
y = zeros(1,N+1);

% initial value y0
y(1) = y0;

%define the functions f, Df, DDf
f=@(y) 2*(y-y^2);
Df=@(y) 2*(1-2*y);
DDf=@(y) -4;

% for-loop for the Taylor series method
for k=1:N
    y(k+1) = y(k) + h*f(y(k)) + ...
        h^2/2*Df(y(k))*f(y(k));
end

%save approximated solution
y=y(end);

end
```

Listing 5.3: `trv_3`

```
function y = trv_3(h,T,y0)

N = round(T/h);

%preallocation
y = zeros(1,N+1);

% initial value y0
y(1) = y0;
```

```

%define the functions f, Df, DDf
f=@(y) 2*(y-y^2);
Df=@(y) 2*(1-2*y);
DDf=@(y) -4;

% for-loop for the Taylor series method
for k=1:N
    y(k+1) = y(k) + h*f(y(k)) + ...
        h^2/2*Df(y(k))*f(y(k)) + ...
        h^3/6*(DDf(y(k))*(f(y(k)))^2 + Df(y(k))^2*f(y(k)));
end

%save approximated solution
y=y(end);
end
```

(5.5e) Write a MATLAB function

`function p = trvconv`

in which you calculate the convergence order of Ψ_1, Ψ_2 and Ψ_3 for (5.5.1).
HINT: Choose $y(0) = 2$ as initial value and calculate the error at the end point $T = 2$ for different step sizes $h = 2^{-4}, 2^{-5}, \dots, 2^{-10}$. Then, apply a linear interpolation of the logarithm of the error vs. the logarithm of the stepsize with the MATLAB function `polyfit`.
HINT: The exact solution of (5.5.1) with initial value y_0 is $y(t) = y_0(y_0 + (1 - y_0)e^{-2t})^{-1}$.
Solution: The convergence order of these methods are determined in the implementation in Listing 5.4. The code outputs the following values which coincide with the theoretical values:

- convergence rate of Ψ_1: 0.9847,
- convergence rate of Ψ_2: 2.0475,
- convergence rate of Ψ_3: 3.0628.

Figure 5.1 depicts the results.

Listing 5.4: `trvconv.m`

```
function p = trvconv

% end point
T=2;
%vector of stepsizes
h=0.5.^[4:10];
% initial value
y0=2;

```

```
% exact solution of y'(t)=2(y-y^2), y(0)=y0
sol=@(t,y0)  y0./(y0+(1-y0)*exp(-2*t));

% Preallocate storage area
err1 = zeros(1,length(h));
err2 = zeros(1,length(h));
err3 = zeros(1,length(h));

% find method errorVerfahrensfehler ermitteln
for j=1:length(h)
    % compute the absolute error for the different methods
     err1(j) = abs( trv_1(h(j),T,y0)-sol(T,y0));
     err2(j) = abs( trv_2(h(j),T,y0)-sol(T,y0));
     err3(j) = abs( trv_3(h(j),T,y0)-sol(T,y0));
end

% plot results
figure;
loglog(h, err1, 'bx-', h, err2, 'rx-', h,err3, 'gx-', 'LineWidth', 2);
title('convergence rate for the Taylor series method for n=1,2,3');
xlabel('h');
ylabel('error');
set(gca, 'XDir', 'reverse');
grid on;
print -depsc trvconv

% compute rate of convergence
fit = polyfit(log(h),log(err1),1);
p(1) = fit(1);
fprintf('experimental convergence rate for n=1: %f\n', p(1));
fit = polyfit(log(h),log(err2),1);
p(2) = fit(1);
fprintf('experimental convergence rate for n=2: %f\n', p(2));
fit = polyfit(log(h),log(err3),1);
p(3) = fit(1);
fprintf('experimental convergence rate for n=3: %f\n', p(3));
```

(5.5f) Formulate the discrete evolutions Ψ_n for the linear problem

$$\dot{\mathbf{y}} = \boldsymbol{A}\mathbf{y}, \qquad \mathbf{y} \in \mathbb{R}^d, \quad \boldsymbol{A} \in \mathbb{R}^{d \times d},$$

and general $n \in \mathbb{N}$. Implement Ψ_n, $n = 1, 2, 3$ for $\boldsymbol{A} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ and test which of these methods is length-preserving.

HINT: First show, that $\Delta_i \mathbf{z} = \boldsymbol{A}^i \mathbf{z}$.

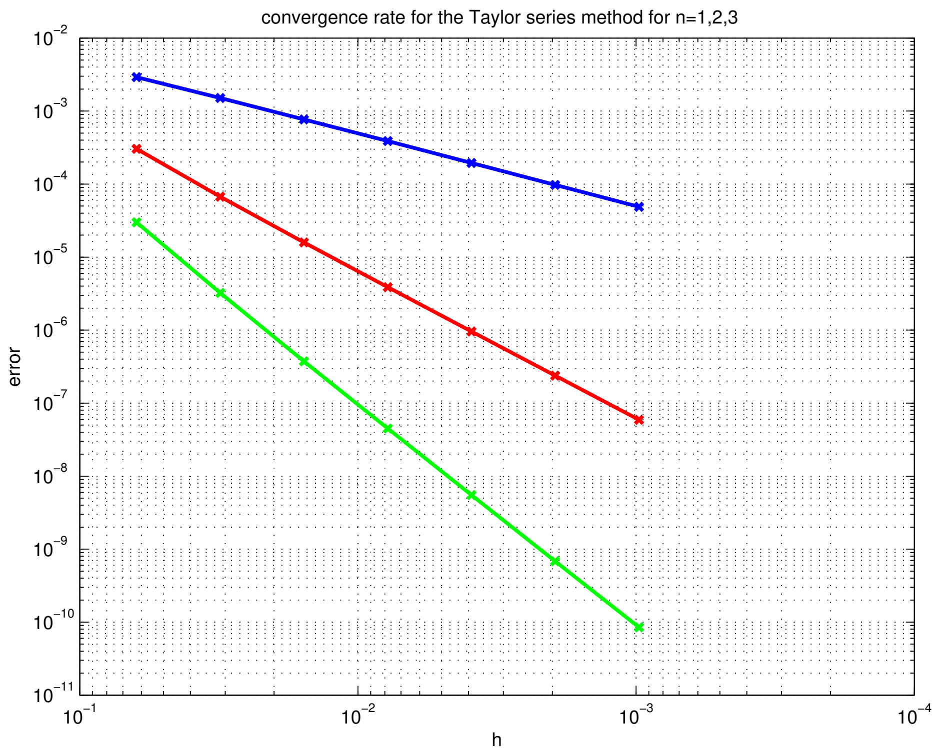

Figure 5.1: Convergence rate of the Taylor series method Ψ_1, Ψ_2, Ψ_3 for the logistic differential equation.

Solution: For the given linear problem, we have

$$\frac{\mathrm{d}^{i-1}}{\mathrm{d}s^{i-1}}\mathbf{f}(\mathbf{y}(s)) = \frac{\mathrm{d}^{i-1}}{\mathrm{d}s^{i-1}}\boldsymbol{A}\mathbf{y}(s) = \frac{\mathrm{d}^{i-2}}{\mathrm{d}s^{i-2}}\boldsymbol{A}\frac{\mathrm{d}}{\mathrm{d}s}\mathbf{y}(s) = \frac{\mathrm{d}^{i-2}}{\mathrm{d}s^{i-2}}\boldsymbol{A}^2\mathbf{y}(s) = \cdots = \boldsymbol{A}^i\mathbf{y}(s)$$

Thereby we have

$$\Psi_n^{t,t+h}\mathbf{z} = \sum_{i=0}^{n}\frac{1}{i!}(h\mathbf{A})^i\mathbf{z}.$$

For the implementation see 5.5. The quality of length-preservation of the individual methods is displayed in Figure 5.2. While the deviation from the initial value's norm sinks with rising index n, none of the methods looked at here preserve length up to machine accuracy.

Listing 5.5: `trvlin.m`

```
function trvlin(n)
% Solves the initial valtrvlin(1:5)ue problem y'=Ay, y(0)=y0
   up to the end point T
% with the explicit Taylor series method and constant
   stepsize h=T/N
% and plots the development of the relative error in the
   iterations Norm
%
% Input: n: indexes of the observed method as row vector
%
```

```
% Output: none
%
% Example call-up:
% trvlin(1:5)

% initialisation
A = [0, -1; 1, 0];
y0 = [1; 1] / sqrt(2);
T = 1;
N = 20;
h = T / N;

% save initial value's norm
norm_y0 = norm(y0);

% preallocate storage space
Y = repmat(y0, 1, length(n));
error = zeros(N, length(n));

% loop over steps
for k = 1:N

    % loop over methods
    for i = 1:length(n)

        % compute next iteration
        Y(:, i) = taylorstep(Y(:, i), A, h, n(i));

        % compute relative error in norm
        error(k, i) = abs(norm(Y(:, i)) - norm_y0) /
           norm_y0;

    end
end

% plot solutions
figure
semilogy(1:N, error, 'Linewidth', 2.0)

% label plot
xlabel('Iterations')
ylabel('$\vert \Vert y_k \Vert_2 - \Vert y_0 \Vert_2 \vert /
   \Vert y_0 \Vert_2$', 'Interpreter', 'latex', 'Fontsize', 12)
labels = cell(length(n), 1);
for i = 1:length(n)

    labels{i} = sprintf('Taylor series method n=%u', n(i));
```

```
end
legend(labels, 'Location', 'BestOutside')

function y = taylorstep(y, A, h, j)
% Completes a step of the Taylor series method Psi_j for the
% linear diff equation y'=Ay.
%
% input:    y: current iteration,
%           A: matrix in differential equation,
%           h: stepsize,
%           j: index of applied method.
%
% output : y: next iteration
%
% example call-up:
% y = taylorstep([1; 1]/sqrt(2), [0, -1; 1, 0], 1/20, 1)

z = y;
for i = 1:j

     z = (h/i) * A * z;
     y = y + z;
end
```

Published on 21 March 2016.

To be submitted by 5 April 2016.

References

[NODE] Lecture Notes for the course "Numerical Methods for Ordinary Differential Equations".

[NUMODE] Lecture Slides for the course "Numerical Methods for Ordinary Differential Equations", SVN revision # 52913.

Last modified on April 22, 2016

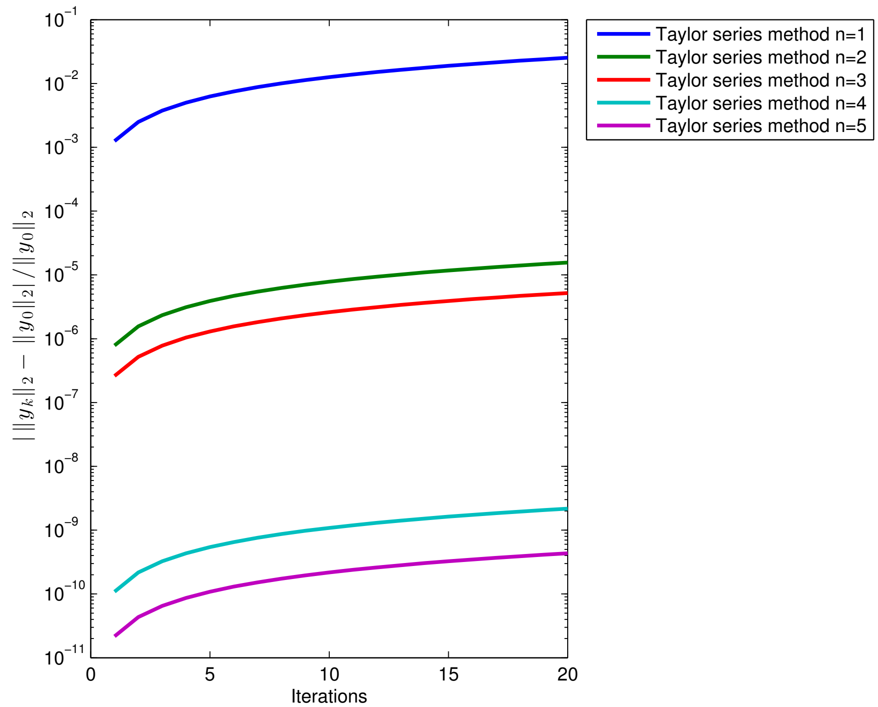

Figure 5.2: Development of the relative error in the iterations norm for the Taylor series methods $\Psi_n, n = 1, \dots, 5$.