P. Grohs
W. Wu

Spring Term 2016

Numerical Analysis II

ETH Zürich
D-MATH

Exam Summer 2016

Problem 1 Symplectic RK 1-Step Method for Linear Hamiltonian Differential Equation [29 Marks]

A given *symplectic* matrix $\mathbf{C} \in \mathbb{R}^{2n\times 2n}$ is partitioned into four $(n \times n)$ blocks as follows:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C_{pp}} & \mathbf{C_{pq}} \\ \mathbf{C}_{\mathbf{pq}}^{\top} & \mathbf{C_{qq}} \end{bmatrix}.$$

Furthermore, we set

$$\mathbf{J} = \begin{bmatrix} 0 & \mathbf{I}_n \\ -\mathbf{I}_n & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix},$$

where $\mathbf{I}_n$ stands for $n \times n$ identity matrix.

(1a) ⚀ Derive the Hamiltonian differential equation, which is defined by the Hamilton function [NODE, Def. 1.2.3]

$$H : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, \quad H(\mathbf{p}, \mathbf{q}) = \tfrac{1}{2}\mathbf{y}^{\top}\mathbf{C}\mathbf{y}.$$

□

In the next section we discuss the proof of the following statements:

For an arbitrary Runge-Kutta method the following properties are equivalent.

(i) The stability function $S(z)$ of the method satisfies $S(-z)S(z) = 1$ for all $z \in \mathbb{C}$.

(ii) The method is reversible [NUMODE, Def. 2.1.27] for linear differential equations $\dot{\mathbf{y}} = \mathbf{A}\mathbf{y}$ with $\mathbf{A} \in \mathbb{R}^{2n\times 2n}$, $n \in \mathbb{N}$.

(iii) The method is symplectic [NUMODE, Def. 4.4.18] for the differential equation $\dot{\mathbf{y}} = \mathbf{J}^{-1}\mathbf{C}\mathbf{y}$ with symmetric symplectic matrix $\mathbf{C}$. $\mathbf{C}$ and $\mathbf{J}$ are defined at the beginning of this problem.

(1b) ⚂ Show that (i) ⇒ (ii). □

(1c) ⚁ Show that (ii) ⇒ (i). □

(1d) ⚄ Show that (i) $\Rightarrow$ (iii). ☐

HINT: Use the fact that when applied to matrices, the stability function $S : \mathbb{R}^{2n\times 2n} \to \mathbb{R}^{2n\times 2n}$ satisfies the identities $S(\mathbf{T}\mathbf{M}\mathbf{T}^{-1}) = \mathbf{T}S(\mathbf{M})\mathbf{T}^{-1}$ and $S(\mathbf{M}^\top) = S(\mathbf{M})^\top$ for any matrix $\mathbf{M} \in \mathbb{R}^{2n\times 2n}$ and any invertible matrix $\mathbf{T} \in \mathbb{R}^{2n\times 2n}$.

(1e) ⚄ For simplified case $n = 1$, show that (iii) $\Rightarrow$ (i). (Proof for general n is welcome but not mandatory!) ☐

HINT: Set $\mathbf{C} = \mathbf{I}_2$ and diagonalise the resulting matrix equation by an appropriate basis transformation. Furthermore use the special structure of the stability function of a Runge-Kutta method.

(1f) ⚂ Let $\mathbf{C} = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}$. For the IVP $\dot{\mathbf{y}} = \mathbf{J}^{-1}\mathbf{C}\mathbf{y}$, $\mathbf{y}(0) = (1, 1)^\top$, complete the template `ImpMidSymSolve.m` using implicit midpoint method to approximate $\mathbf{y}(1)$ with $N = 20$ steps. Perform 2 iterations in each Newton method for root-finding problem. Plot the corresponding Hamiltonian of each step. ☐

Problem 2 Singly Diagonal Implicit Runge-Kutta Methods [25 Marks]

For general implicit Runge-Kutta methods, the k_i's cannot be evaluated succesively since they are coupled in the system of implicit equations that is given for their determination. One way to decouple a nonlinear system is to use diagonal implicit Runge-Kutta methods. A special family of those methods are singly diagonal implicit Runge-Kutta methods (SDIRK), in which the matrix of the method is lower triangular and all the diagonal entries are equal. Continuing in this direction we define the following one parameter family of SDIRK-methods

$$\begin{array}{c|cc} \gamma & \gamma & 0 \\ 1-\gamma & 1-2\gamma & \gamma \\ \hline & 1/2 & 1/2 \end{array}$$

Table I: Singly Diagonal Implicit Runge-Kutta

(2a) ⚁ Determine the values of γ for which the corresponding Runge-Kutta method is of at least (consistency) order 3. □

(2b) ⚂ Write down the iterations of the Runge-Kutta method defined in Table I, when applied to the ODE

$$\dot{y} = \lambda y, \qquad y(0) = 1, \qquad \lambda \in \mathbb{C}.$$

□

HINT: Use the stability function and the fact that the ODE is linear.

(2c) ⚂ Complete the template `StabilityRegion.m` and plot the stability region of the Runge-Kutta method (Table I). Using the given plots, for that values of γ from problem (2a) (the values with which the method is of order 3) can we conjecture the corresponding Runge-Kutta method to be A-stable? □

(2d) ⚃ Investigate whether the method is A-stable for $\gamma = 1$. Is it L-stable? □

HINT: Notice that in order to investigate A-stability it is sufficient to consider only the case $z = \imath y$, for $y \in \mathbb{R}$. Why?

(2e) ⚂ Consider the ODE

$$\dot{y} = \cos(y^2) + \frac{y}{4}, \quad y(0) = \frac{1}{2}.$$

and compute its numerical solution with the Runge-Kutta method table I on `T = [0 3]` and for `N = 300`. To do so, compute the stages k_1 and k_2 by first computing k_1 and then k_2, both by using Newton method. The iteration in Newton method will stop when 100 iterations have been carried out, or the difference between neighboring results is less than `1e-10`. Complete the template

```
function SDIRK(T, N, gamma).
```

□

Problem 3 Extrapolating Implicit Mid-Point and Explicit Euler [24 Marks]

In this problem we will apply the extrapolation method to the implicit midpoint method and the explicit Euler method, and then compare their performances. Consider the logistic ODE

$$\dot{y} = \lambda y(1-y), \quad \lambda > 0, \tag{3.1}$$

with the initial value $y(0) = y_0 > 0$

(3a) ⚀ Find the fixed points of the ODE (3.1) and determine if any of them are attractive. Explain why given $y(0) > 0$ it follows that $y(t) > 0$ for all $t > 0$. ☐

(3b) ⚂ Give the closed form of the discrete evolution $\Psi^h y$ of the implicit mid-point rule when applied to the logistic differential equation (3.1), and argue whether the solution is admissible assuming the initial value satisfies $y(0) > 0$. ☐

HINT: The discrete evolution of the implicit mid-point rule leads to a quadratic equation which admits an explicit solution. Then use (3a) to conclude which of the two expressions makes sense for $y(0) > 0$.

(3c) ⚁ Complete the templates

```
function y = ImplicitMidpoint(y0, lambda, h)
```

and

```
function y = ExplicitEuler(y0, lambda, h),
```

to carry out the implicit midpoint and the explicit euler method for (3.1) where the parameters include a given initial value `y0`, positive parameter `lambda` and step size `h`. Use the result in (3b) directly for implicit midpoint method. ☐

(3d) ⚂ Suppose that for ODE (3.1), we have performed a chosen single step method n times with different step sizes $h = (h_1, \cdots, h_n)$ on time interval $[0, t_0]$, where $h_1 < h_2 < \cdots < h_n$. Let T_i be the approximation of $y(t_0)$ for step size h_i, $i = 1, \cdots, n$. Let $T := (T_1, \cdots, T_n)$. Implement a MATLAB function using the template

```
function y = Extrapolation(T, h).
```

that performs Aitken-Neville extrapolation method to compute the extrapolated value for $y(t_0)$. ☐

(3e) ⚂ Consider again the ODE (3.1), where we take $y(0) = 0.03$, $\lambda = 5$, and complete the template `ExtrapolatedSingleStep.m` which performs a series of the implicit midpoint and explicit Euler methods with different step sizes and calculate the extrapolated result using `Extrapolation(T,h)` in (3d).

In the program we take `2.^(3:8)` subdivisions of the time interval `[0 1]`. Print out the result at the end of program. ☐

Problem 4 Index Reduction and Constraint Stabilization [22 Marks]

(4a) ⚁ Consider the system of differential equations

$$\begin{aligned}\dot{y} &= f(y)\\ \mathbf{0} &= g(y)\end{aligned} \tag{4.1}$$

where $y(t) \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^m$, n and m are fixed constants. Here we assume that $Dg(y)$ has full rank and solution exists in $\mathbb{R}^+$ for all initial value $y_0 \in \mathbb{R}^n$. Prove by differentiation of the constraints that, for initial values satisfying $g(y_0) = \mathbf{0}$, the solution of the differential-algebraic equation (DAE) with a new variable μ of size $\operatorname{rank}(g_y)$

$$\begin{aligned}\dot{y} &= f(y) + Dg(y)^\top \mu\\ \mathbf{0} &= Dg(y)f(y)\\ \mathbf{0} &= g(y)\end{aligned} \tag{4.2}$$

also solves the differential equation (4.1). This trick can be used for index reduction. □

When trying to solve a DAE or a ODE using a single step method, one usually cannot ensure that the constraints are preserved during each step. In practice, projection method is often used to remedy this defect. For differential equation system (4.1), suppose that the initial value $y_n \in \mathcal{M} := \{y; g(y) = \mathbf{0}\}$. We want to ensure that y_{n+1} is also in $\mathcal{M}$. One step of projection method $y_n \to y_{n+1}$ is defined as follows:

- Step 1: Compute $\tilde{y}_{n+1} = \Psi_h(y_n)$, where Ψ_h could be any single step method applied to $\dot{y} = f(y)$.
- Step 2: Project the value $\tilde{y}_{n+1}$ onto the manifold $\mathcal{M}$ to obtain $y_{n+1} \in \mathcal{M}$.

For the computation of Step 2, one has to solve the constrained minimization problem

$$\|y_{n+1} - \tilde{y}_{n+1}\| \to \min \qquad \text{subject to} \quad g(y_{n+1}) = \mathbf{0}$$

To solve this minimization problem, a standard approach is to introduce Lagrange multipliers $\boldsymbol{\lambda} = \boldsymbol{\lambda}(n)$ which depends only on n, and to consider the Lagrange function

$$\mathcal{L}(y_{n+1}, \lambda) = \|y_{n+1} - \tilde{y}_{n+1}\|^2 - g(y_{n+1})^\top \boldsymbol{\lambda}(n+1).$$

$\partial\mathcal{L}/\partial y_{n+1} = 0$ is a necessary condition for $\|y_{n+1} - \tilde{y}_{n+1}\|$ to reach the minimum. This leads to the system

$$y_{n+1} = \tilde{y}_{n+1} + Dg(\tilde{y}_{n+1})^\top \boldsymbol{\lambda}(n+1) \tag{4.3}$$

$$\mathbf{0} = g(y_{n+1}) \tag{4.4}$$

Notice that y_{n+1} has been replaced by $\tilde{y}_{n+1}$ in first equation to simplify the calculation of $Dg(y)$. We have to calculate $\boldsymbol{\lambda}$ for y_{n+1}. Inserting (4.3) into (4.4) gives a nonlinear equation for λ:

$$g(\tilde{y}_{n+1} + Dg(\tilde{y}_{n+1})^\top \boldsymbol{\lambda}(n+1)) = \mathbf{0} \tag{4.5}$$

To solve $\boldsymbol{\lambda}$ out of (4.5), we use Newton method with *one* iteration. The initial estimation for $\boldsymbol{\lambda}(n+1)$ is $\boldsymbol{\lambda}(n)$, and $\Delta\lambda_n := \boldsymbol{\lambda}(n+1) - \boldsymbol{\lambda}(n)$ can be approximately given as

$$\Delta\lambda_n = -(Dg(\tilde{y}_{n+1})Dg(\tilde{y}_{n+1})^\top)^{-1} g(\tilde{y}_{n+1} + Dg(\tilde{y}_{n+1})^\top \lambda_i),$$

Now we consider a practical example.

(4b) ⚁ Consider the perturbed Kepler problem with Hamiltonian function

$$H(\mathbf{p},\mathbf{q}) = \frac{1}{2}(p_1^2+p_2^2) - \frac{1}{\sqrt{q_1^2+q_2^2}} - \frac{0.005}{2\sqrt{(q_1^2+q_2^2)^3}}. \tag{4.6}$$

Write out the Hamiltonian system given Hamiltonian in (4.6).

Prove that the angular momentum, $L(\mathbf{p},\mathbf{q}) := q_1p_2 - q_2p_1$, is an invariant([NODE, Def. 1.2.2]) to the Hamiltonian system above. □

(4c) ⚁ Given initial values $q_1(0) = 1-e$, $q_2(0) = 0$, $p_1(0) = 0$, $p_2(0) = \sqrt{(1+e)/(1-e)}$, where $e = 0.6$ stands for the eccentricity. Use explicit Euler method with step size $h = 0.03$ on time interval $[0, 201]$. Complete the template `KeplerWithoutProjection.m` and plot the trajectory of solution and the evolution in the value of H and L with time. Are H and L conserved? □

(4d) ⚄ Given the same initial values as in (4c), conduct the explicit Euler method *with the projection method* introduced before (4b). Complete the template `KeplerWithProjection.m` and plot trajectory of solution and the evolution of the value of H and L with time. What conclusion can you draw by comparing the outcome of `KeplerWithoutProjection.m` and `KeplerWithProjection.m`? □

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 # 63606.