P. Grohs
Ž. Kereta
C. Schillings

Spring Term 2014

Numerical Analysis II

ETH Zürich
D-MATH

Exam Summer 2014

Problem 1 Discontinuous Collocation Methods [25 Marks]

Consider the following non-autonomous ODE

$$\dot{y} = f(t, y)\,, \qquad y(t_0) = y_0\,.$$

Let $c_2, \ldots, c_{s-1} \in \mathbb{R}$ be distinct real numbers with $0 \leq c_j \leq 1,\ j = 2, \ldots, s-1$ and let $b_1, b_s \in \mathbb{R}$ be two arbitrary real numbers. The corresponding *discontinuous collocation method* is then defined via a polynomial $u(t)$ of degree $s-2$ satisfying

$$u(t_0) = y_0 - hb_1(\dot{u}(t_0) - f(t_0, u(t_0))) \tag{1.1}$$

$$\dot{u}(t_0 + c_i h) = f(t_0 + c_i h, u(t_0 + c_i h))\,, \quad i = 2, \ldots, s-1 \tag{1.2}$$

$$y_1 = u(t_1) - hb_s(\dot{u}(t_1) - f(t_1, u(t_1))) \tag{1.3}$$

with $t_1 = t_0 + h$.

(1a) Let $u(t)$ be the interpolation polynomial of degree $s-2$ and define

$$k_i := \dot{u}(t_0 + c_i h) \quad i = 2, \ldots, s-1\,,$$

$k_1 := f(t_0, u(t_0))$ and $k_s := f(t_1, u(t_1))$. We have

$$\dot{u}(t_0 + \tau h) = \sum_{j=2}^{s-1} k_j l_j(\tau)$$

by the Lagrange interpolation formula, where $l_j(\tau)$ is the Lagrange polynomial

$$l_j(\tau) = \prod_{l=2, l\neq j}^{s-1} (\tau - c_l)/(c_j - c_l)\,.$$

Show (by integration) that

$$k_i = f(t_0 + c_i h, y_0 + h\sum_{j=1}^{s} k_j a_{ij})\,, \quad i = 1, \ldots, s\,, \tag{1.4}$$

where the coefficients $a_{ij},\ i,j=1,\dots,s$ are given by

$$a_{i1} = b_1\,, \qquad a_{is} = 0\,, \quad i=1,\dots,s\,, \tag{1.5}$$

$$a_{ij} = \int_0^{c_i} l_j(\tau)\,\mathrm{d}\tau - b_1 l_j(0)\,, \quad i=1,\dots,s\,,\ j=2,\dots,s-1\,, \tag{1.6}$$

$$b_j = \int_0^1 l_j(\tau)d\tau - b_1 l_j(0) - b_s l_j(1)\,, \quad j=2,\dots,s-1 \tag{1.7}$$

with $c_1 = 0$, $c_s = 1$.

(1b) Using $k_i,\ i=1,\dots,s$ given by (1.4), show that

$$y_1 = y_0 + h\sum_{j=1}^{s} k_j b_j\,, \tag{1.8}$$

with $b_j,\ j=2,\dots,s-1$ defined by (1.7).

(1c) Let the coefficients $a_{ij},\ i,j=1,\dots,s$ be given by (1.5) - (1.6) with $c_1 = 0$, $c_s = 1$. Show that

$$\sum_{j=1}^{s} a_{ij} = c_i\,. \tag{1.9}$$

Thus, the discontinuous collocation method (1.1) - (1.3) is equivalent to an s-stage Runge-Kutta method with coefficients (1.5) - (1.7) and $c_1 = 0$, $c_s = 1$.

(1d) Investigate the link to standard collocation methods in the case where $b_1 = b_s = 0$.

(1e) Write a MATLAB function

```
function [A,b] = CollCoeff(c,b1,bs)
```

which takes the collocation points $c_i \in [0,1]$ as a vector $\mathbf{c} \in \mathbb{R}^s$ and the two real numbers $b_1,\ b_s$ as input and returns the matrix $\mathfrak{A} \in \mathbb{R}^{s\times s}$ and the vector $\mathbf{b} \in \mathbb{R}^s$ with $(\mathfrak{A})_{ij} = a_{ij}$ and $(\mathbf{b})_i = b_i$ (see (1.5) - (1.7)) of the corresponding Runge-Kutta method.

HINT: The MATLAB functions `polyint`, `polyval` and `vander` may be of use.

(1f) Implement the discontinuous collocation method (1.1) - (1.3) based on Lobatto quadrature formulas (note that $c_1 = 0$ and $c_s = 1$ for Lobatto methods) to solve autonomous differential equations of the form

$$\dot{y} = f(y)\,, \qquad y(t_0) = y_0\,.$$

Compute the Lobatto points with the function `LobattoRoots.m`, find the coefficients with your implementation from subproblem (1e) and rephrase the method as a root finding problem. Apply the implementation of Newton's method `Newton.m` to it.

Complete the template `DiscCollLobatto.m`. Rephrase the method as a root-finding problem by completing the templates `fNewton.m` and `DfNewton.m`.

HINT: In case of problems with rephrasing the method as a root-finding problem, you can use the corresponding `pcodes`. Bear in mind though that you will not be awarded full marks for the subproblem unless you complete all the templates.

(1g) Consider the initial value problem

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

Find the absolute error of the discontinuous collocation method at the point `T=0.5` for a variation of the number of steps $N_h = 2^i$, $i = 2, \ldots, 6$ and of the stages $s = 3, b_1 = 1/6, b_s = 1/6$ and $s = 4, b_1 = 1/12, b_s = 1/12$. Use a fixed number of Newton iterations `nNewton`$= 3$. Plot the error curves against the number of steps on logarithmic scale and find the algebraic convergence order using the MATLAB function `polyfit`. Use the template `DiscCollLobattoConv.m`.

HINT: You can find a reference solution with `ode45`. Set the relative and absolute tolerances to 10^{-12}.

Problem 2 Partitioned Runge-Kutta Methods [30 Marks]

We consider differential equations in the partitioned form

$$\dot{p} = f(p,q)\,, \qquad \dot{q} = g(p,q)\,, \tag{2.1}$$

where $p(t) \in \mathbb{R}^n$ and $q(t) \in \mathbb{R}^m$ with $m, n \in \mathbb{N}$. The initial data is given by $p(t_0) = p_0$ and $q(t_0) = q_0$.

Let b_i, a_{ij} and $\hat{b}_i, \hat{a}_{ij}$ be the coefficients of two s-stage Runge-Kutta methods

$$\begin{array}{c|c} \mathbf{c} & \mathfrak{A} \\ \hline & \mathbf{b}^\top \end{array} \qquad \begin{array}{c|c} \hat{\mathbf{c}} & \hat{\mathfrak{A}} \\ \hline & \hat{\mathbf{b}}^\top \end{array}\,.$$

A *partitioned Runge-Kutta method* with s stages for the solution of (2.1) is defined by

$$k_i = f\big(p_0 + h\sum_{j=1}^{s} a_{ij}k_j, q_0 + h\sum_{j=1}^{s} \hat{a}_{ij}l_j\big) \tag{2.2}$$

$$l_i = g\big(p_0 + h\sum_{j=1}^{s} a_{ij}k_j, q_0 + h\sum_{j=1}^{s} \hat{a}_{ij}l_j\big) \tag{2.3}$$

$$p_1 = p_0 + h\sum_{i=1}^{s} b_i k_i\,, \quad q_1 = q_0 + h\sum_{i=1}^{s} \hat{b}_i l_i\,. \tag{2.4}$$

(2a) Show that the partitioned Runge-Kutta method (2.2) - (2.4) is of order 2, if the coupling conditions

$$\sum_{i=1}^{s} b_i\hat{c}_i = \frac{1}{2}\,, \qquad \sum_{i=1}^{s} \hat{b}_i c_i = \frac{1}{2} \tag{2.5}$$

are satisfied, in addition to the usual Runge-Kutta conditions for order 2.

(2b) Show that Störmer-Verlet method defined for systems of the form (2.1) by

$$p_{1/2} = p_0 + \frac{h}{2}f(p_{1/2}, q_0) \tag{2.6}$$

$$q_1 = q_0 + \frac{h}{2}(g(p_{1/2}, q_0) + g(p_{1/2}, q_1)) \tag{2.7}$$

$$p_1 = p_{1/2} + \frac{h}{2}f(p_{1/2}, q_1) \tag{2.8}$$

can be interpreted as a partitioned Runge-Kutta method.

(2c) Consider the following ODE

$$\begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} 0 & \omega \\ -\omega & 0 \end{pmatrix}\begin{pmatrix} p \\ q \end{pmatrix}. \tag{2.9}$$

Compute the exact solution of (2.9) and find an invariant of (2.9).

(2d) Consider the s-stage Runge-Kutta methods given by

$$\begin{array}{c|c} \mathbf{c} & \mathfrak{A} \\ \hline & \mathbf{b}^\top \end{array} \qquad \begin{array}{c|c} \hat{\mathbf{c}} & \hat{\mathfrak{A}} \\ \hline & \hat{\mathbf{b}}^\top \end{array}.$$

Provide an explicit expression for the numerical evolution of the corresponding partitioned Runge-Kutta method applied to (2.9).

(2e) We consider the initial value problem

$$\begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} 0 & \pi \\ -\pi & 0 \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix}, \quad \begin{pmatrix} p(0) \\ q(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{2.10}$$

Write a MATLAB function

```
function yend = SVMeth(T,N)
```

which solves (2.10) with N uniform steps on the time interval $[0, T]$ of the Störmer-Verlet method (2.6) - (2.8), interpreted as a partitioned Runge-Kutta method, and returns an approximation of $y(T)$.

Problem 3 The Rattle Method [29 Marks]

We consider a mechanical system with coordinates $q \in \mathbb{R}^d$ that are subject to constraints $g(q) = 0$. The equations of motion are given by

$$\begin{aligned} \dot{p} &= -\nabla_q H(p,q) - \nabla_q g(q)\lambda(p,q) && (3.1)\\ \dot{q} &= \nabla_p H(p,q) && (3.2)\\ 0 &= g(q)\,, && (3.3) \end{aligned}$$

where the Hamiltonian is defined as $H(p,q) = \frac{1}{2}p^\top M^{-1}p + U(q)$ with a positive definite mass matrix M, potential $U(q)$, Lagrange multiplier $\lambda(p,q) \in \mathbb{R}^m$, constraint $g(q) = (g_1(q), \ldots, g_m(q))^\top$, where $\nabla_q g = (\nabla_q g_1, \ldots, \nabla_q g_m)$ denotes the transposed Jacobian matrix, and vectors $p,\ q \in \mathbb{R}^d$.

Differentiating the constraint $g(q(t)) = 0$ with respect to time yields the hidden constraint

$$0 = \nabla_q g(q)^\top \nabla_p H(p,q)\,.$$

To solve the system (3.1) - (3.3), we will apply an adaption of the Störmer-Verlet method, the so-called *Rattle method*, defined as follows

$$\begin{aligned} p_{1/2} &= p_0 - \frac{h}{2}(\nabla_q H(p_{1/2}, q_0) + \nabla_q g(q_0)\lambda_0) && (3.4)\\ q_1 &= q_0 + \frac{h}{2}(\nabla_p H(p_{1/2}, q_0) + \nabla_p H(p_{1/2}, q_1)) && (3.5)\\ 0 &= g(q_1) && (3.6)\\ p_1 &= p_{1/2} - \frac{h}{2}(\nabla_q H(p_{1/2}, q_1) + \nabla_q g(q_1)\mu_0) && (3.7)\\ 0 &= \nabla_q g(q_1)^\top \nabla_p H(p_1, q_1)\,. && (3.8) \end{aligned}$$

The vector $\mu_0 \in \mathbb{R}^m$ results from an additional projection step (3.7) - (3.8), therefore one has to additionally determine the unknowns λ_0 and μ_0 in order to compute p_1 and q_1.

(3a) Show that the Rattle method (3.4) - (3.8) is reversible.

HINT: A single step method is said to be reversible with respect to an autonomous ODE, if the corresponding discrete evolution Ψ satisfies $\Psi^{-h} \circ \Psi^h = Id$, cp. [NUMODE, Def. 2.1.27].

(3b) Let us now consider a particle moving on the unit sphere and being attracted by a fixed point $a \in \mathbb{R}^3$ on the sphere. Formulate the Rattle method (3.4) - (3.8) for the Kepler problem on the sphere, which can be described by the equations of motion (3.1) - (3.3) with the Hamiltonian of the form

$$H(p,q) = \frac{1}{2}p^\top p + U(q,a) \qquad (3.9)$$

and the constraint

$$g(q) = q^\top q - 1\,, \qquad (3.10)$$

where the potential is given by $U(q,a) = -\frac{q^\top a}{\sqrt{1-(q^\top a)^2}}$ and $q, p \in \mathbb{R}^3$. Provide explicit expressions for the unknowns p_1, q_1, μ_0 and λ_0.

HINT: Due to the separability of the Hamiltonian and the quadratic constraint g, the Rattle method applied to the Kepler problem (3.9) - (3.10) will be explicit except for the computation of λ_0, for which a quadratic equation needs to be solved.

(3c) Implement the Rattle method (3.4) - (3.8) to solve the Kepler problem on the sphere (3.9) - (3.10) with $a = (0.1/\sqrt{1.05}, 1/\sqrt{1.05}, 0.2/\sqrt{1.05})^\top$ and initial values $q_0 = (1,0,0)^\top$, $p_0 = (0,0,-1)^\top$, step size $h = 0.01$ and time $T = 10.0$. Complete the template `RattleKepler.m`.

Problem 4 Linearly Implicit Runge-Kutta Method [16 Marks]

Implicit Runge-Kutta methods applied to nonlinear initial value problems lead to a nonlinear system of equations. Their solutions can in general only be obtained by iteration, which adds the question of convergence of the iterations to the already existing problem of stability. In this problem we will take a look at an alternative approach.

We consider an autonomous differential equation

$$\dot{y} = f(y), \qquad f \in C^1(\Omega_0, \mathbb{R}^d). \tag{4.1}$$

For the computation of the discrete flow $\Psi^h y$ at $y \in \Omega_0$ let us rewrite the ODE (4.1) as

$$\dot{y}(t) = Jy(t) + (f(y(t)) - Jy(t))$$

where $J = Df(y)$. Now we discretise implicitly only the first, linear, term. The second, nonlinear, term is discretised explicitly. Hence, we can use a Runge-Kutta method and consider discrete evolutions of the form

$$\Psi^h y = y + h \sum_{j=1}^{s} b_j k_j \tag{4.2}$$

where

$$k_i = J\left(y + h\sum_{j=1}^{i} d_{ij}k_j\right) + \left(f\left(y + h\sum_{j=1}^{i-1} a_{ij}k_j\right) - J\left(y + h\sum_{j=1}^{i-1} a_{ij}k_j\right)\right). \tag{4.3}$$

Let us define matrices $\mathfrak{A} = (a_{ij})_{i,j=1}^s$ and $\mathfrak{D} = (d_{ij})_{i,j=1}^s$, filled with zeros where needed. Note that $\mathfrak{A}$ corresponds to an explicit Runge-Kutta method, hence it is strictly lower triangular, while $\mathfrak{D}$ corresponds to a diagonally implicit Runge-Kutta method, hence it is lower triangular.

(4a) Determine the system of successive linear equations for each of the coefficients k_i.

The question of the existence of solutions of linear systems obtained in (4a) is easier than it is for general implicit Runge-Kutta methods.

(4b) Let $\beta \geq 0$ and take a matrix $J \in \mathbb{R}^{d\times d}$. Show that the matrix $I - h\beta J$ is invertible for $0 \leq h < h_*$ where h_* depends on the spectral abscissa $\nu(J)$ as follows

$$h_* = \infty \text{ for } \nu(J) \leq 0 \text{ and } h_* = \frac{1}{\beta\nu(J)} \text{ for } \nu(J) > 0.$$

Hence, we conclude that for stiff problems with $\nu(J) \leq 0$ there are no restrictions on the step size.

HINT: Spectral abscissa is defined as $\nu(J) = \max_{\lambda\in\sigma(J)}\{\operatorname{Re}\lambda_i\}$, where $\sigma(J)$ denotes the spectrum of J.

(4c) Show that the stability function of (4.2) - (4.3) is given by

$$S(z) = 1 + z\mathbf{b}^\top(I - z\mathfrak{D})^{-1}\mathbf{1}$$

where $\mathbf{1} = (1, 1, \ldots, 1)^\top$ and $\mathbf{b} = (b_1, \ldots, b_s)^\top$.

(4d) Consider the logistic equation

$$\dot{y} = f(y) = \lambda y(1-y), \; y(0) = \frac{1}{100},$$

for $\lambda = 500$. Find an approximate solution of this ODE by applying a linearly implicit method consisting of a Runge-Kutta method of order 3, given by the Butcher tableau

$$\begin{array}{c|cc} \frac{1}{2}+\frac{1}{2\sqrt{3}} & \frac{1}{2}+\frac{1}{2\sqrt{3}} & 0 \\ \frac{1}{2}-\frac{1}{2\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{2}+\frac{1}{2\sqrt{3}} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array},$$

and used for the linear term, and the explicit trapezoidal rule [NUMODE, Eq. (2.3.3)], which we use to treat the nonlinear term. Write the equations for the coefficients (as in (4a)), implement the resulting method in MATLAB with `T=1` and study the convergence order using the MATLAB function `polyfit` for $N_h = 2^i$, $i = 9, \ldots, 17$. Complete the templates `LinearlyImplicit.m` and `ConvergenceOrder.m`.

References

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