P. Grohs
Ž. Kereta

Summer Term 2015

Numerical Analysis II

ETH Zürich
D-MATH

Exam Summer 2015

Problem 1 A 2-stage Gauss-Legendre Method [26 Marks]

Consider the Butcher tableau

$$\begin{array}{c|cc} \frac{1}{2}-\gamma & \frac{1}{4} & \frac{1}{4}-\gamma \\ \frac{1}{2}+\gamma & \frac{1}{4}+\gamma & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Table I: Butcher tableau of the 2-stage Gauss-Legendre method

where we take $\gamma = \frac{\sqrt{3}}{6}$. In subproblems (a)-(d) we will apply this Runge-Kutta to the model ODE

$$\dot{y} = \lambda y, \quad y(0) = 1, \qquad \lambda \in \mathbb{C}. \tag{1.1}$$

(1a) ⚀ Write down the equations for the stages k_1 and k_2 of the Runge-Kutta method defined by Table I, when applied to the ODE (1.1) using a step size h at time t and the current iterate being y_n.

(1b) ⚁ Find a linear system $\mathbf{C}\mathbf{k} = \mathbf{b}$, where $\mathbf{k} = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$ are the stages from (1a).

(1c) ⚂ Show that the iterations of the Runge-Kutta method defined by Table I, applied to the ODE (1.1), satisfy

$$y_{n+1} = \frac{1 + \frac{1}{2}z + \frac{1}{12}z^2}{1 - \frac{1}{2}z + \frac{1}{12}z^2} y_n,$$

where $z = \lambda h$.

(1d) ⚃ Show that the Runge-Kutta method defined by Table I is A-stable.

HINT: Apply the maximum modulus principle.

(1e) ⚃ Let us now turn our attention to a different differential equation. Consider the IVP

$$\dot{y} = \frac{1}{t^2} - \frac{y}{t} - y^2, \quad y(1) = -1.$$

Compute its numerical solution using the Runge-Kutta method defined by Table I by completing the template `GaussLegendreRK.m` on the time interval `T = [1 3]`, using `N = 1500` steps. To solve the implicit system use Newton's method and complete the template `Newton.m`.

Problem 2 Shooting Methods [27 Marks]

Consider the two-point boundary value problem (BVP)

$$y'' = f(t, y, y'), \quad y(a) = A,\ y(b) = B \tag{2.1}$$

with $a < b$, and $t \in [a, b]$. Throughout the lectures we have restricted our attention to initial value problems, hence, we cannot apply any of our methods directly to address such a differential equation. The idea behind shooting methods is to instead of trying to solve the two-point boundary value problem as it is, we should rather transform it into a series of initial value problems (which we can solve) whose solutions converge to the solution of the original boundary value problem.

The first step in creating an IVP out of (2.1) is to introduce a condition on $y'(a)$. Let us make an initial guess s for $y'(a)$ and denote by $y(t; s)$ the corresponding solution of the initial value problem

$$y'' = f(t, y, y'), \quad y(a) = A,\ y'(a) = s. \tag{2.2}$$

With $y'(t; s)$ we mean $\frac{\partial y(t;s)}{\partial t}$.

(2a) ⚁ Write the second order IVP (2.2) as a first order IVP system.

The solution $y(t; s)$ of the IVP (2.2) will coincide with the solution of the BVP (2.1) provided that $y(b; s) = B$, that is, provided that we can find an s such that

$$\phi(s) := y(b; s) - B = 0. \tag{2.3}$$

To find the root of this equation we shall consider Newton's method. Newton's method computes a sequence $(s_n)_n$ generated by the formula

$$s_{n+1} = s_n - \frac{\phi(s_n)}{\frac{\partial \phi(s_n)}{\partial s}}, \tag{2.4}$$

which is iterated until we reach a sufficiently good approximation to the root of (2.3).

In terms of implementing (2.4) we need to compute $\frac{\partial \phi(s_n)}{\partial s}$. To do so, we introduce new (dependent) variables

$$\xi(t; s) = \frac{\partial y(t; s)}{\partial s}, \quad \eta(t; s) = \frac{\partial y'(t; s)}{\partial s}.$$

(2b) ⚂ Show that

$$\begin{aligned} \xi'(t; s) &= \eta(t; s), & \xi(a; s) &= 0 \\ \eta'(t; s) &= \frac{\partial f(t, y(t; s), y'(t; s))}{\partial y}\xi(t; s) + \frac{\partial f(t, y(t; s), y'(t; s))}{\partial y'}\eta(t; s), & \eta(a; s) &= 1. \end{aligned} \tag{2.5}$$

If we now assign the value s_n to s, $n \geq 0$, the IVPs from subproblems (2a) and (2b) can be coupled and the resulting system can subsequently be solved using a suitable Runge-Kutta method on the interval $[a, b]$.

(2c) ⚄ Complete the MATLAB template

```
function [u] = ImplicitMidpoint(F, DyF, t, u0)
```

which uses the implicit midpoint rule to find a numerical solution to the problem

$$\mathbf{u} = F(t, \mathbf{u}), \quad \mathbf{u}(t_0) = \mathbf{u}_0.$$

The inputs are a vector valued function `F`, its Jacobian `DyF`, time grid `t`, and the initial value `u0`. The output is the approximated solution `u`. To solve the implicit system use Newton's iterations with the help of the corresponding template `Newton.m`.

HINT: In our case, the Jacobian of the function $F : \mathbb{R} \times \mathbb{R}^4 \to \mathbb{R}^4$ is only computed for the space variables, that is, $D_y F : \mathbb{R} \times \mathbb{R}^4 \to \mathbb{R}^{4\times 4}$.

Having solved the coupled system we obtain an approximation to $y(b; s_n)$ which we can use to calculate $\phi(s_n) = y(b; s_n)$, and an approximation to $\xi(b; s_n) = \frac{\partial}{\partial s}\phi(s_n)$. Therefore, we can perform the next iteration of the Newton's algorithm (2.4). The process is repeated until the iterates are sufficiently close to the desired solution

(2d) ⚃ Implement the previously described shooting method by completing the MATLAB template `ShootingMethod.m`. The function should compute a numerical approximation to the solution of

$$y'' = -\frac{(y')^2}{y}, \qquad y(0) = 1,\ y(1) = 2,$$

by in each step of the Newton's method first computing the solution of the corresponding IVP (using the MATLAB function `ImplicitMidpoint.m`) with the current guess s_n, and then calculating the next approximation s_{n+1} by using (2.4).

Problem 3 Euler Equations for the Motion of a Rigid Body [21 Marks]

In the principle axes coordinates, the motion of a free rigid body is described by the Euler equations

$$\begin{aligned} \dot{y_1} &= a_1 y_2 y_3, \qquad & a_1 &= \frac{I_2 - I_3}{I_2 I_3} \\ \dot{y_2} &= a_2 y_3 y_1, \qquad & a_2 &= \frac{I_3 - I_1}{I_3 I_1} \\ \dot{y_3} &= a_3 y_1 y_2, \qquad & a_3 &= \frac{I_1 - I_2}{I_1 I_2} \end{aligned} \tag{3.1}$$

where $\mathbf{y} = (y_1, y_2, y_3)^\top$ represents angular velocities, and I_1, I_2 and I_3 are the principal moments of inertia. We assume that the equations are nontrivial, that is, that $a_1^2 + a_2^2 + a_3^2 \neq 0$. Equations (3.1) can be written as

$$\begin{pmatrix} \dot{y_1} \\ \dot{y_2} \\ \dot{y_3} \end{pmatrix} = \begin{pmatrix} 0 & y_3/I_3 & -y_2/I_2 \\ -y_3/I_3 & 0 & y_1/I_1 \\ y_2/I_2 & -y_1/I_1 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \tag{3.2}$$

(3a) ⚁ Show that the (squared) length of the angular velocity vector, defined as

$$I(\mathbf{y}) = \frac{1}{2}\left(y_1^2 + y_2^2 + y_3^2\right)$$

is an invariant of the ODE (3.1).

(3b) ⚁ Show that the kinetic energy

$$H(\mathbf{y}) = \frac{1}{2}\left(\frac{y_1^2}{I_1} + \frac{y_2^2}{I_2} + \frac{y_3^2}{I_3}\right)$$

is an invariant of the ODE (3.1).

(3c) ⚃ Show that explicit Euler does not preserve the invariant $I(\mathbf{y})$ and that the error term is of order h^2.

(3d) ⚃ Compute approximate solutions of the ODE (3.1) using implicit midpoint and explicit Euler. Complete the templates `RigidBody.m`, `ExplicitEuler.m` and `ImplicitMidpoint.m`. Take the initial value $\mathbf{y}(0) = (\cos(1.1), 0, \sin(1.1))^\top$ and $I_1 = 2, I_2 = 1, I_3 = 2/3$, and use the step size as `h=0.1` on the time interval `T = [0 30]`. What can you infer from your plots regarding the conservation of the length of angular velocity of the approximate solutions given by these two methods?

Problem 4 A Splitting Scheme for the Lotka-Volterra Model [26 Marks]

Lotka-Volterra equations are one of the earliest models in mathematical biology. They are often used to describe the interactions between populations of predator and prey species in a given biological system. As an example, let $u(t)$ and $v(t)$ represent a scaled measure of population sizes at time t. Changes in population sizes due to the interactions between the two populations can be modeled in terms of non-linear equations

$$\begin{aligned} \dot{u} &= u(v-2), \\ \dot{v} &= v(1-u). \end{aligned} \tag{4.1}$$

(4a) Show that by introducing $p = \ln(u)$ and $q = \ln(v)$, the system (4.1) can be rewritten as a Hamiltonian system.

Let us now partition the equations (4.1) into two systems

$$\begin{aligned} \dot{u} &= u(v-2) \\ \dot{v} &= 0 \end{aligned} \tag{4.2}$$

and

$$\begin{aligned} \dot{u} &= 0 \\ \dot{v} &= v(1-u). \end{aligned} \tag{4.3}$$

Thus, we can rewrite the ODE (4.1) as

$$\dot{\mathbf{y}} = f^{[1]}(\mathbf{y}) + f^{[2]}(\mathbf{y}), \tag{4.4}$$

where $\mathbf{y} = (u, v)^\top$, $f^{[1]}(\mathbf{y}) = \begin{pmatrix} u(v-2) \\ 0 \end{pmatrix}$ and $f^{[2]}(\mathbf{y}) = \begin{pmatrix} 0 \\ v(1-u) \end{pmatrix}$.

(4b) Compute the exact flows of the partitioned systems (4.2) and (4.3). The initial conditions are given by $u(0) = u_0$ and $v(0) = v_0$.

(4c) Apply the Strang splitting scheme to (4.4), and write down the resulting flow $\Psi^h \mathbf{y}$.

(4d) Compute the fixed points of the ODE (4.1) and investigate whether Strang splitting preserves those fixed points.

(4e) Compute a numerical solution of the Lotka-Volterra system (4.1) using the Strang splitting scheme on the interval `T = [0 20]`. Set the initial value $(u_0, v_0)^\top = (6, 2)^\top$ and the step size $h = 0.01$. Furthermore, compute a reference solution using `ode23` and plot both solutions. Complete the template `LotkaVolterraStrang.m`. Is there a qualitative difference between the two solutions ?