H. Ammari
W. Wu
S. Yu

Spring Term 2017

# Numerical Analysis II

ETH Zürich
D-MATH

## Exam Summer 2017

In all problems, we always assume that all functions $f$ on the right-hand side of Initial Value Problem are smooth with respect to all variables on the domain of definition.

## Problem 1 Linear System [30 Marks]

Consider the system of equations

$$\begin{cases} \dfrac{\mathrm{d}p}{\mathrm{d}t} = -\sin q, & t \geq 0, \\ \dfrac{\mathrm{d}q}{\mathrm{d}t} = p, & \\ p(0) = p_0 \in \mathbb{R},\ q(0) = q_0 \in \mathbb{R}. & \end{cases} \tag{1.1}$$

**(1a)** ⚁ Is (1.1) a Hamiltonian system?

**(1b)** ⚀ Rewrite (1.1) in the form

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = f(x), & \text{where } x = (p,q)^T \\ x(0) = x_0 = (p_0, q_0)^T. & \end{cases} \tag{1.2}$$

**(1c)** ⚄ Consider the midpoint scheme

$$\begin{cases} x^{k+1} = x^k + \Delta t f\Big(\dfrac{x^k + x^{k+1}}{2}\Big), \\ x^0 = x_0. \end{cases} \tag{1.3}$$

for solving (1.2) where $\Delta t$ is the step size. Prove that the method is symplectic.

**(1d)** ⚃ Suppose that there exists a matrix $S \in \mathbb{R}^{2\times 2}$ (symmetric and positive definite) and a positive constant $c$ such that

$$x(t)^T S x(t) = c, \quad \forall t \geq 0, \quad \text{where } x \text{ is the solution to } (1.2). \tag{1.4}$$

Prove that $(x^k)^T S x^k = c$ for $k \geq 0$, where $x^k$ is from (1.3).

**(1e)** ⚁ Let us denote $x^k = (p^k, q^k)$. Does the midpoint scheme (1.3) preserve the quantity

$$\frac{1}{2} p_k^2 - \cos q_k \ ?$$

**(1f)**

- Implement the midpoint scheme (1.3) using templates `ImpMidPointSolve.m` and `ImpMidPointStep.m`. `y1=ImpMidPointStep(f,y0,t0,h,tol)` should return the value of one-step midpoint scheme (1.3), and `y=ImpMidPointSolve(f,y0,T,h,tol)` should return the approximate value of IVP at the end time $T$. In `ImpMidPointStep.m`, we use the fixed point iteration as the method for root-finding problem in implicit method. The parameters `f`, `y0`, `T`, `h`, `tol` stand for the right-hand side of IVP, initial value of IVP, ending time, step size and error tolerance, correspondingly.
- Complete the template `RunPrb1.m` to solve (1.2) on time interval $[0, 8]$ using your codes `ImpMidPointSolve.m` and `ImpMidPointStep.m`. Set the initial value as $x(0) = [\pi/2, 0]^T$. Set the time step `h` as $h = 0.25$ and set the error tolerance `tol` as $tol = 10^{-6}$.
- At the end of `RunPrb1.m`, template code will plot the computed numerical solution.

## Problem 2 Improved Single Step Method [30 Marks]

For Initial value problem

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = f(t,x), & t \in [0,T], \\ x(0) = x_0, & x_0 \in \mathbb{R}, \end{cases} \tag{2.1}$$

consider the following scheme for solving (2.1):

$$\begin{cases} x^{k+1} = x^k + \Delta t(1-\alpha) f(t_k, x^k) + \Delta t\, \alpha\, f\Big(t_k + \frac{\Delta t}{2\alpha}, x^k + \frac{\Delta t}{2\alpha} f(t_k, x^k)\Big), \\ x^0 = x(0), \end{cases} \tag{2.2}$$

with some constant $\alpha \geq 1/4$.

**(2a)** ⚄ Show that (2.2) is consistent and that the truncation error $T_k(\Delta t, \alpha)$ can be expressed as

$$T_k(\Delta t, \alpha) = \frac{(\Delta t)^2}{8\alpha}\Big[(\frac{4}{3}\alpha - 1)\frac{\mathrm{d}^3 x}{\mathrm{d}t^3}(t_k) + \frac{\mathrm{d}^2 x}{\mathrm{d}t^2}(t_k)\frac{\partial f}{\partial x}(t_k, x(t_k))\Big] + O((\Delta t)^3). \tag{2.3}$$

**(2b)** Apply the method (2.2) to solve (numerically)

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = -x^p, & t \in [0,1], \\ x(0) = 1, \end{cases} \tag{2.4}$$

where $p \in \mathbb{N}$.

Compare the numerical solution to the explicit analytic solution and give the order of convergence of the global error.

- ⚃ Complete the template `methodPrb2.m` to implement the scheme (2.2). The function `u=methodPrb2(odefun,alp,T,y0,N)` should return the value of $x(1)$. Here `odefun, alp, T, y0, N` means the right-hand-side of the initial value problem, the value of $\alpha$, the end time, initial value $x(0)$ and total step number, respectively.
- ⚂ Now set $p = 3$ and $\alpha = 1/4 + 0.3$ and total step number to be $N = 2^k, k = 1, 2, \cdots, 10$. Complete the template `RunPrb2.m` using your code `methodPrb2.m` to solve (2.4) numerically and then find the convergence order the scheme (2.2).

**(2c)** ⚃ Show (analytically) that, when method (2.2) is applied to (2.4), if $p = 1$, then $T_k(\Delta t, \alpha) = O((\Delta t)^2)$ and, if $p = 2$, then there exists $\alpha \geq 1/4$ such that $T_k(\Delta t, \alpha) = O((\Delta t)^3)$.

## Problem 3 Symmetry of Numerical Method [40 Marks]

Consider the differential equation

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = f(x) \\ x(0) = x_0 \in \mathbb{R}^2 \end{cases} \tag{3.1}$$

We assume that $f$ is Lipschitz continuous in $\mathbb{R}^2$.

Given a numerical method

$$x^{k+1} = \Phi_{\Delta t}(x^k),$$

where $\Delta t$ is the step size. Its adjoint $\tilde{x}^{k+1} = \Phi^*_{\Delta t}(\tilde{x}^k)$ is defined by $\tilde{x}^k = \Phi_{-\Delta t}(\tilde{x}^{k+1})$ or equivalently

$$\tilde{x}^{k+1} = \Phi^{-1}_{-\Delta t}(\tilde{x}^k).$$

We say that a numerical method is symmetric if it satisfies $\Phi_{\Delta t} \circ \Phi_{-\Delta t} = I$.

**(3a)**

1. ⚂ Is explicit Euler method symmetric? Is implicit Euler method symmetric? Prove them.
2. ⚂ Recall that $x^{k+1} = x^k + f((x^{k+1} + x^k)/2)$ is the implicit midpoint rule. Is implicit midpoint rule symmetric? Prove it.
3. ⚂ For $x^{k+1} = \Phi_{\Delta t}(x^k)$ a numerical method, show that

$$x^{k+1} = \Phi_{\Delta t/2} \circ \Phi^*_{\Delta t/2}(x^k) \tag{3.2}$$

   is a symmetric method.
4. ⚃ Denote $x = (p, q)$. Assume that (3.1) is a Hamiltonian system with Hamiltonian function $H = H(p, q)$. Write out the corresponding formulas(may be implicit) for numerical methods that (3.2) yields when $\Phi$ is explicit euler method and implicit euler method.

**(3b)** Consider a Runge-Kutta method that is consistent, i.e. $\sum_{i=1}^m b_i = 1$, and with coefficient such that $\sum_{j=1}^m a_{ij} = c_i$ for $1 \le i \le m$.

1. ⚄ Show that adjoint of the Runge-Kutta method is again a Runge-Kutta method, with coefficient given by

$$a^*_{ij} = b_{m+1-j} - a_{m+1-i,m+1-j},$$
$$b^*_i = b_{m+1-i}$$

   for $1 \le i, j \le m$.
2. ⚂ Deduce from (3b).1 that $a_{ij} = b_j - a_{m+1-i,m+1-j}$ for all $i, j = 1, \cdots, m$, if the method is symmetric.
3. ⚂ Deduce from (3b).2 that, if the Runge-Kutta method is explicit, it cannot be symmetric.