Norwegian University of Science and Technology
Department of Mathematical Sciences

Semester project TMA4215

Deadline November 15th 2010

You can work in groups of maximum 3 persons. This project is divided in two parts: the first part is a set of exercises relevant for the final exam. The solution of this part can be **handwritten**; the second part is a project on the approximation of elliptic functions and integrals.Your reports for this part should **not exceed 4 pages** (reduce the size of the figures to stay within the limits of pages). We appreciate if you try to write well documented Matlab code.

Part 1

Problem 1 Show that the polynomial $p(x)$ of degree less than or equal to 2 satisfying

$$p(0) = y_1, \quad p(1) = y_2, \quad \int_0^1 p(t)\,dt = y_3,$$

exists and is unique.

Problem 2 Show that if $\phi(x) = f(x)g(x)$, then the following formula holds for the divided differences

$$\phi[x_0, x_1, \ldots, x_n] = \sum_{r=0}^{n} f[x_0, x_1, \ldots, x_r]g[x_r, \ldots, x_n].$$

Problem 3 We shall study and construct splines of degree 2 on the knots $a = x_0 < x_1 < \cdots < x_n = b$.

a) Consider the construction of the quadratic spline interpolating the data

x	x_0	$\cdots$	x_n
$f(x)$	f_0	$\cdots$	f_n

and with the extra condition $s'(x_0) = 0$. Derive the linear system of equations to be solved to compute the s'_i, $i = 0, \ldots, n$ by means of the f_i, $i = 0, \ldots, n$. Here $s'_i := s'(x_i)$.

Consider then the data

x	0	0.5	1
$f(x)$	1	2	0.5

.

Solve the linear system you obtained to construct the quadratic spline, and give an expression for the spline as a polynomial of degree 2 on each subinterval I_0 and I_1[1].

b) Show the error bound

$$|f(x) - s(x)| \leq 2h\|f'\|_\infty, \forall x \in [a, b],$$

where $f \in C^1([a, b])$ and $s(x)$ is the quadratic spline interpolating the data $f_i := f(x_i)$, $i = 0, \dots, n$ and the knots are assumed equidistant ($x_i - x_{i-1} = h$ for $i = 1, \dots, n$).

Problem 4 The function

$$R(z) = \frac{1 + z/2}{1 - z/2} = 1 + \sum_{n=1}^{\infty} \frac{z^n}{2^{n-1}}, \tag{1}$$

is defined for all $-2 < z < 2$.

a) Prove that $\mathrm{e}^z - R(z) = -\frac{z^3}{12} + \mathcal{O}(z^4)$.

b) Find a unit diagonal, lower triangular matrix L and an upper triangular matrix U such that $LU = A$ when

$$A = \begin{bmatrix} 1.10 & -0.05 & 0.00 \\ -0.05 & 1.10 & -0.05 \\ 0.00 & -0.05 & 1.10 \end{bmatrix}.$$

Hint: Pivoting is not needed.

c) The *matrix exponential* e^{hX} where X is a square matrix and $h \in \mathbf{R}$, is defined by the series expansion

$$\mathrm{e}^{hX} = I + \sum_{n=1}^{\infty} \frac{(hX)^n}{n!}. \tag{2}$$

Here, I is the identity matrix of the same size as X. We remark that this series converges for all square matrices X.

In this problem we will use *rational approximations* to e^{hX}. Specifically, we will evaluate $R(hX)$ with $R(z)$ defined by the power series in (1).

Assume $\|\frac{h}{2}X\| < 1$, use the Neumann series[2] for $(I - \frac{h}{2}X)^{-1}$ to obtain a closed form expression for $R(hX)$.

[1] The polynomial on each of the two intervals is a function of the computed s'_i and of the interpolated values $s_i = f_i$, $i = 0, 1, 2$.

[2] Ch. 4.5 Kincaid and Cheney.

d) We want to compute ***the second column*** of $R(hX)$, which we denote by $\mathbf{x}$. Show that this amounts at solving a linear system of equations $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$. Give an expression for A and $\mathbf{b}$. Assume then

$$h = 0.1, \quad X = \begin{bmatrix} -2 & 1 & \\ 1 & -2 & 1 \\ & 1 & -2 \end{bmatrix},$$

and compute $\mathbf{x}$.
Hint: The constant 1 must be replaced by the identity matrix when evaluating $R(z)$ at matricial arguments.

Part 2

Romberg quadrature

Problem 5 Given $0 \leq k < 1$, the function

$$\varphi \mapsto F(\varphi, k) \ := \int_0^{\varphi} \frac{d\theta}{\sqrt{1 - k^2 \sin^2 \theta}} \tag{3}$$

is called (incomplete) *Jacobi elliptic integral of the first kind* with modulus k and is an invertible smooth map. Its inverse is called *amplitude* of modulus k and is an odd function

$$\mathrm{am}(\,\cdot\,, k) : \mathbf{R} \to \Big(-\frac{\pi}{2}, \frac{\pi}{2} \Big),$$

and $\varphi = \mathrm{am}(u, k)$ is simply the solution of the scalar nonlinear equation

$$G(\varphi) = 0, \quad G(\varphi) := F(\varphi, k) - u. \tag{4}$$

The *Jacobi elliptic functions* sn and cn of modulus k are the functions $\mathbf{R} \to [-1, 1]$ defined as

$$\mathrm{sn}(u, k) = \sin(\mathrm{am}(u, k))\,, \qquad \mathrm{cn}(u, k) = \cos(\mathrm{am}(u, k)) \tag{5}$$

and are periodic of period $4K(k)$, where $K(k) = F(\frac{\pi}{2}, k)$ (the so called complete elliptic integral of the first type of modulus k). We can also define the functions

$$\mathrm{dn}(u, k) = \sqrt{1 - k^2 \mathrm{sn}(u, k)^2}\,, \qquad \mathrm{sd}(u, k) = \frac{\mathrm{sn}(u, k)}{\mathrm{dn}(u, k)}. \tag{6}$$

The Jacobi elliptic functions can be computed using the built in Matlab routine `ellipj` based on the use of the so called Arithmetic Geometric Mean method.

We want here to use Romberg integration to compute the Jacobi elliptic integrals of the first kind, solve (4) with Newton method, and use (5) and (6) to compute the elliptic functions.

a) Implement a Matlab function performing Romberg integration to evaluate elliptic integrals of the first kind. Give in input $[0, \varphi]$ and n to the Romberg routine: 2^n is the number of intervals used in the row $n+1$ of the Romberg matrix (see Kincaid and Cheney chapter 7).

Compare the results you obtain and the results given by one of the built in Matlab routines for quadrature.

You should provide numerical evidence of the convergence of Romberg algorithm as $n \to \infty$. Keep φ and k fixed and use a sufficiently small tolerance in the built-in Matlab routine. Consider $E(n,0) := |F(\varphi,k) - R(n,0)|$, $E(n,n) := |F(\varphi,k) - R(n,n)|$, where $R(n,m)$ for $n = 0,\dots,$ $m = 0,\dots,n$ are the entries of the Romberg matrix.

Consider then $F(\frac{\pi}{2},k)$, the *complete* elliptic integral of the first kind. This integral can be efficiently computed in Matlab by the routine `ellipke`. Investigate numerically the performance of the Romberg algorithm: what do you observe. Explain the behavior using the Euler-Mclaurin formula.[3]

b) Modify your Romberg routine so that you can give a prescribed tolerance *TOL* as input instead of the value n. Use the following estimate for the error

$$Est(n,m) := \frac{1}{4^m - 1}[R(n,m-1) - R(n-1,m-1)].$$

If $|Est(n,m)| < TOL$ give $R(n,m)$ as output approximation to the integral. Compare the results you get with what you obtain using built-in Matalb functions for quadrature: do you get numerical approximations within the accuracy prescribed by your tolerance? Show this with a numerical experiment.

c) Implement Newton's method for (4) to obtain $\varphi = \text{am}(t,k)$ and evaluate the Jacobi elliptic functions by (5) and (6). Observe that $G'(\varphi) = \frac{1}{\sqrt{1-k^2\sin^2\varphi}}$. Implement an appropriate stopping criterion for the Newton method.

How should you choose the tolerance for stopping the Newton iteration compared to the tolerance of the Romberg algorithm?

The accuracy of the quadrature will influence the solution of the nonlinear equation by the Newton method. Compare the results you obtain and the results given by `ellipj` in Matlab. If you use $TOL = 1e-16$ and your program is implemented properly, your results should differ from the values produced by `ellipj` no more than $C \cdot TOL$ with C a constant and $C \le 10$.

Test your routines first with positive and negative input values u of moderate absolute value $|u|$. Then increase $|u|$, if the performance of your code deteriorates, try to suggest and implement strategies to improve the code.

[3]Note that a similar behavior can be observed when using Romberg method and the trapezoid rule to approximate the following integrals: $\int_\alpha^{\alpha+2\pi} \cos(x)\,dx$ or $\int_\alpha^{\alpha+1} \sin(2\pi x)\,dx$.

Rigid body simulation

Problem 6

Consider the free rigid body Euler equations

$$\dot{\mathbf{m}} = \mathbf{m} \times (T^{-1}\mathbf{m}), \quad \mathbf{m}(0) = \mathbf{m}_0,$$

where T is the diagonal inertia tensor, $T = \mathrm{diag}(I_1, I_2, I_3)$. We assume that the principal moments of inertia (the diagonal entries of the inertia tensor) are distinct and in increasing order, i.e. $I_1 < I_2 < I_3$.

The 2-norm of the angular momentum γ and the energy function E

$$\gamma = \mathbf{m}(t)^T \mathbf{m}(t), \quad E = \frac{1}{2}\mathbf{m}(t)^T \,(T^{-1}\mathbf{m}(t)),$$

are constant along the solution $\mathbf{m}(t)$.

By using the constants of motion it is possible to find the solutions of these equations by means of the Jacobi elliptic functions. These solutions are implemented in the code `freewR`.

a) Implement the mid-point Runge–Kutta implicit integration method to solve numerically the free rigid body Euler equations.

Find an explicit expression for the Jacobian of the nonlinear system of equations to be solved at each time-step, and make a Matlab function implementing this Jacobian. Use the Jacobian in a Newton iteration method to solve the nonlinear system. Provide numerical evidence that the mid-point method has order 2: compare the solution given by the mid-point method for different values of h and the solution obtained using the Jacobi elliptic functions. Choose $[0, 1]$ as time interval.

b) Use now one of the built-in Matlab routines for the numerical solution of ordinary differential equations to solve the free rigid body Euler equations. Integrate on relatively large time intervals. Compute and plot the error in γ and E as a function of time for the midpoint method, for the Matlab ode-routine, and for the solution computed using Jacobi elliptic functions. What do you observe? How is this error affected by the tolerances used in the methods?

c) Perform finally some tests for different values of the principal moments of inertia. The inertia values dictate the shape of the rigid body. How do these values influence the performance of the methods?