Theoretical Exercise

Important: answers without explicit mathematical explanations will not be considered.

We aim to integrate the initial value problem

$$u_t = f(t, u),\ u(0) = u_0,$$

using a variation of a 2-step Adams method (not the Adams-Bashforth method seen in the lectures). We first integrate between $t_{n+1} = t_n + h$ and $t_{n+2} = t_n + 2h$ at both sides of the equation:

$$\int_{t_{n+1}}^{t_{n+2}} u_t \, \mathrm{d}t = \int_{t_{n+1}}^{t_{n+2}} f(t, u) \, \mathrm{d}t, \tag{1}$$

to obtain a linear multistep formula of the form

$$\alpha_0 v_n + \alpha_1 v_{n+1} + \alpha_2 v_{n+2} = h \left(\beta_0 f_n + \beta_1 f_{n+1} + \beta_2 f_{n+2}\right), \tag{2}$$

where h is the time-step. The right-hand side of (2) is the result of approximating the integral on the right-hand side of (1) by means of an interpolatory quadrature of maximum exactness, that is:

$$\int_{t_{n+1}}^{t_{n+2}} f(t) \, \mathrm{d}t \approx w_0 f_n + w_1 f_{n+1} + w_2 f_{n+2},$$

where the quantities f_n, f_{n+1} and f_{n+2} are the values of the interpolatory polynomial at the nodes t_n, t_{n+1}, and t_{n+2}, respectively.

a) Determine the system of equations for the quadrature weights w_0, w_1 and w_2.

Note: change to the time variable $\tau = t - t_n$ to simplify the calculation. **(3.0 p.)**

b) Solve the system and write down the resulting linear multistep formula in the usual form

$$\alpha_0 v_n + \alpha_1 v_{n+1} + \alpha_2 v_{n+2} = h\left(\beta_0 f_n + \beta_1 f_{n+1} + \beta_2 f_{n+2}\right).$$

Identify the coefficients α_j and β_j. **(4.0 p.)**

c) Study the stability of the formula in the limit $h \to 0$. **(3.0 p.)**

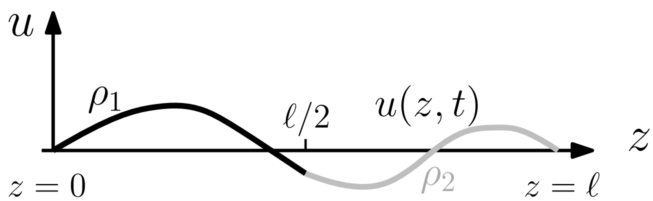
Problem 1

Important: answers without explicit mathematical explanations will not be considered.

Consider the equation corresponding to the small transversal oscillations $u(z,t)$ of a string of length ℓ whose ends $z=0$ and $z=\ell$ are kept fixed, that is:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial z^2},\ u(0,t) = u(\ell,t) = 0.$$

Here, $c^2 = T/\rho$, T being the uniform tension of the string and ρ its density. In this problem we assume that the string is made of two smaller strings of length $\ell/2$ each, with different densities $\rho_1 << \rho_2$ (see figure).



We can model this by considering a single string whose density depends on z through a step function of the form:

$$\rho(z) = \rho_1 + \frac{\rho_2 - \rho_1}{2}\left\{1 + \tanh\left(10\left(z - \frac{\ell}{2}\right)\right)\right\}$$

Henceforth take the numerical values: $T=1$, $\ell=1$, $\rho_1 = 1$ and $\rho_2 = 10$.

a) Sketch by hand the function $\rho(z)$ **(1.0 p.)**

b) Assuming that $u(z,t) = F(z)G(t)$, and using the method of separation of variables, show that $F(z)$ satisfies the boundary value problem

$$\frac{1}{\rho(z)}\frac{\mathrm{d}^2 F}{\mathrm{d}z^2} = \mu F(z),\ F(0) = F(1) = 0, \quad \text{where } \mu \text{ is a constant.}$$

(2.0 p.)

c) Discretize numerically the boundary value problem in (b) and plot the first mode of oscillation $F_1(z)$, and provide its corresponding associated constant μ_1. Detail here the discretization technique that you have used (operators, matrices, nodes, etc.). **(3.0 p.)**

d) Compute the next two fundamental modes, $F_2(z)$, and $F_3(z)$, along with their corrsponding values μ_2 and μ_3, respectively. Plot the modes $F_2(z)$, and $F_3(z)$ and for each mode provide the nodal points z_j, where $F(z_j) = 0$, with at least two exact digits. **(4.0 p.)**

Enginyeria Física
Numerical and Computational Methods II (2020-2021)

Dept. Physics UPC)
Name:

Problem 2

The motion of a charged particle in the field of a magnetic dipole, restricted to the equatorial plane, is given by the ODE system

$$\ddot{x} = -\frac{b}{r^3}\dot{y}, \quad \ddot{y} = \frac{b}{r^3}\dot{x}, \quad r = \sqrt{x^2 + y^2}. \tag{1}$$

The constant b depends on the dipole strength, and the mass and charge of the particle. We will take $b = 1$. There are two conserved magnitudes in this problem:

$$\dot{x}^2 + \dot{y}^2 = e \quad \text{and} \quad \frac{1}{r} + x\dot{y} - y\dot{x} = \ell. \tag{2}$$

1. Exercise (1 p.): Write the ODE in the form $\dot{X} = F(X)$:

2. Exercise (4 p.): Using the four-order Runge-Kutta scheme, integrate the ODE system from the initial condition $x = 1$, $y = 0$, $\dot{x} = 0.22$, $\dot{y} = 0$, for 1000 time steps. Set the time step so that the absolute error on the conserved magnitudes is less than 2×10^{-6}. ($|e - e_0|$, $|\ell - \ell_0| \leq 2 \times 10^{-6}$).

3. Exercise (1 p.): Plot the trajectory on the (x, y) plane.

4. Exercise (3 p.): Obtain the time series for $x(t)$, using the same time step than in 2, but with 3000 time steps. Compute and plot its Fourier transform (DFT).

5. Exercise (1 p.): What are the two frequencies with larger amplitudes?

Theoretical Exercise

Important: answers without explicit mathematical explanations will not be considered.

We aim to integrate the initial value problem

$$u_t = f(t,u),\ u(0) = u_0,$$

using a variation of a 2-step Adams method (not the Adams-Bashforth method seen in the lectures). We first integrate between $t_{n+1} = t_n + h$ and $t_{n+2} = t_n + 2h$ at both sides of the equation:

$$\int_{t_{n+1}}^{t_{n+2}} u_t dt = \int_{t_{n+1}}^{t_{n+2}} f(t,u)dt, \tag{1}$$

to obtain a linear multistep formula of the form

$$\alpha_0 v_n + \alpha_1 v_{n+1} + \alpha_2 v_{n+2} = h(\beta_0 f_n + \beta_1 f_{n+1} + \beta_2 f_{n+2}), \tag{2}$$

where h is the time-step. The right-hand side of (2) is the result of approximating the integral on the right-hand side of (1) by means of an interpolatory quadrature of maximum exactness, that is:

$$\int_{t_{n+1}}^{t_{n+2}} f(t)dt \approx w_0 f_n + w_1 f_{n+1} + w_2 f_{n+2}, \tag{3}$$

where the quantities f_n, f_{n+1} and f_{n+2} are the values of the interpolatory polynomial at the nodes t_n, t_{n+1}, and t_{n+2}, respectively.

a) Determine the system of equations for the quadrature weights w_0, w_1 and w_2.

Note: change to the time variable $\tau = t - t_n$ to simplify the calculation. **(3.0 p.)**

The change of variables reduces (3) to

$$\int_{h}^{2h} f(\tau)d\tau \approx w_0 f_n + w_1 f_{n+1} + w_2 f_{n+2}, \tag{4}$$

where f_{n+j} is $f(\tau = jh)$. By requiring exactness of the previous equation for polynomials up to degree 2 in τ (1, τ and τ^2) we obtain

$$\begin{aligned} 1w_0 + 1w_1 + 1w_2 &= \textstyle\int_h^{2h} 1 d\tau = h, \\ 0w_0 + hw_1 + 2hw_2 &= \textstyle\int_h^{2h} \tau d\tau = 3h^2/2, \\ 0w_0 + h^2 w_1 + (2h)^2 w_2 &= \textstyle\int_h^{2h} \tau^2 d\tau = 7h^3/3, \end{aligned} \quad \Rightarrow \quad \begin{aligned} w_0 + w_1 + w_2 &= h, \\ w_1 + 2w_2 &= 3h/2, \\ w_1 + 4w_2 &= 7h/3. \end{aligned} \tag{5}$$

We have wrote, by abuse of language, $f(t,u(t)) = f(t_n + \tau, u(t_n + \tau)) = g(\tau) = f(\tau)$.

b) Solve the system and write down the resulting linear multistep formula in the usual form

$$\alpha_0 v_n + \alpha_1 v_{n+1} + \alpha_2 v_{n+2} = h\left(\beta_0 f_n + \beta_1 f_{n+1} + \beta_2 f_{n+2}\right). \tag{6}$$

Identify the coefficients α_j and β_j. **(4.0 p.)**

Solving the previous linear system for w_j we obtain

$$w_0 = -\frac{h}{12}, \quad w_1 = \frac{2h}{3}, \quad w_3 = \frac{5h}{12}.$$

By using the fundamental theorem of calculus, from (1) we obtain

$$\int_{t_{n+1}}^{t_{n+2}} u_t dt = u(t_{n+2}) - u(t_{n+1}) = u_{n+2} - u_{n+1}.$$

Substitution into (3) produces the linear multistep formula

$$v_{n+2} - v_{n+1} = w_0 f_n + w_1 f_{n+1} + w_2 f_{n+2} = h(\beta_0 f_n + \beta_1 f_{n+1} + \beta_2 f_{n+2}),$$

where v is the numerical approximation of u given by the algorithm considered. Comparing with (2) and using $\beta_j = w_j/h$ we have

$$\alpha_0 = 0, \quad \alpha_1 = -1, \quad \alpha_2 = 1, \quad \beta_0 = -1/12, \quad \beta_1 = 2/3, \quad \beta_2 = 5/12.$$

c) Study the stability of the formula in the limit $h \to 0$. **(3.0 p.)**

In the limit of $h \to 0$, the polynomial associated to the linear multistep formula

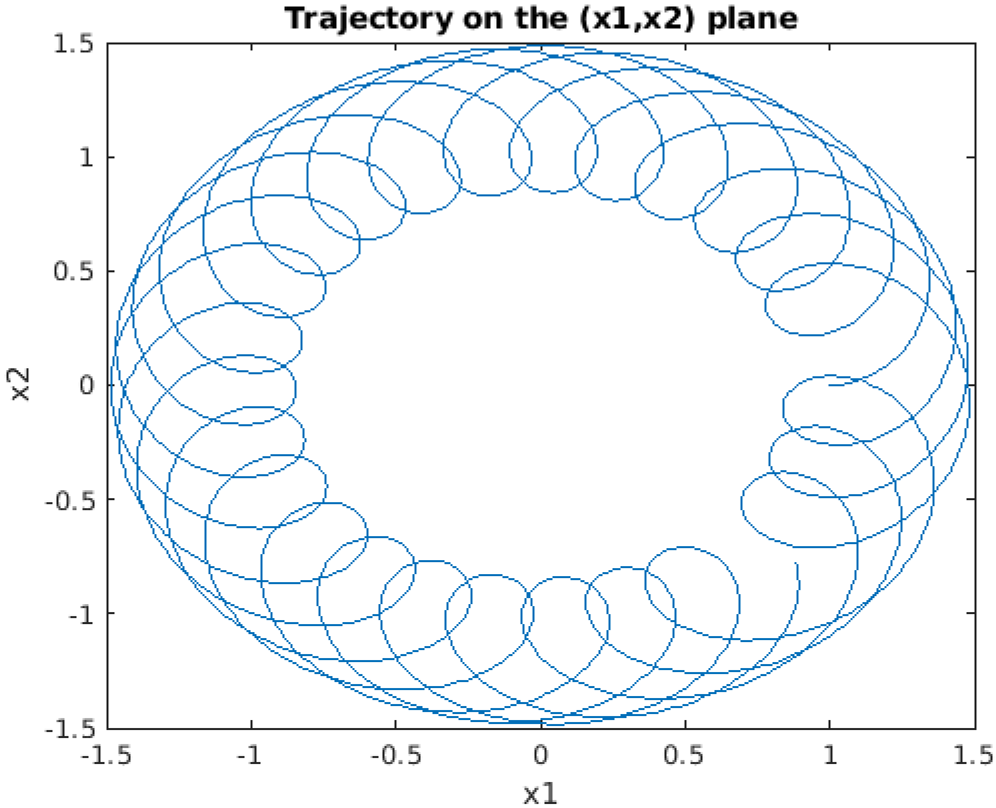
$$v_{n+2} - v_{n+1} = h(-f_n/12 + 2f_{n+1}/3 + 5f_{n+2}/12)$$

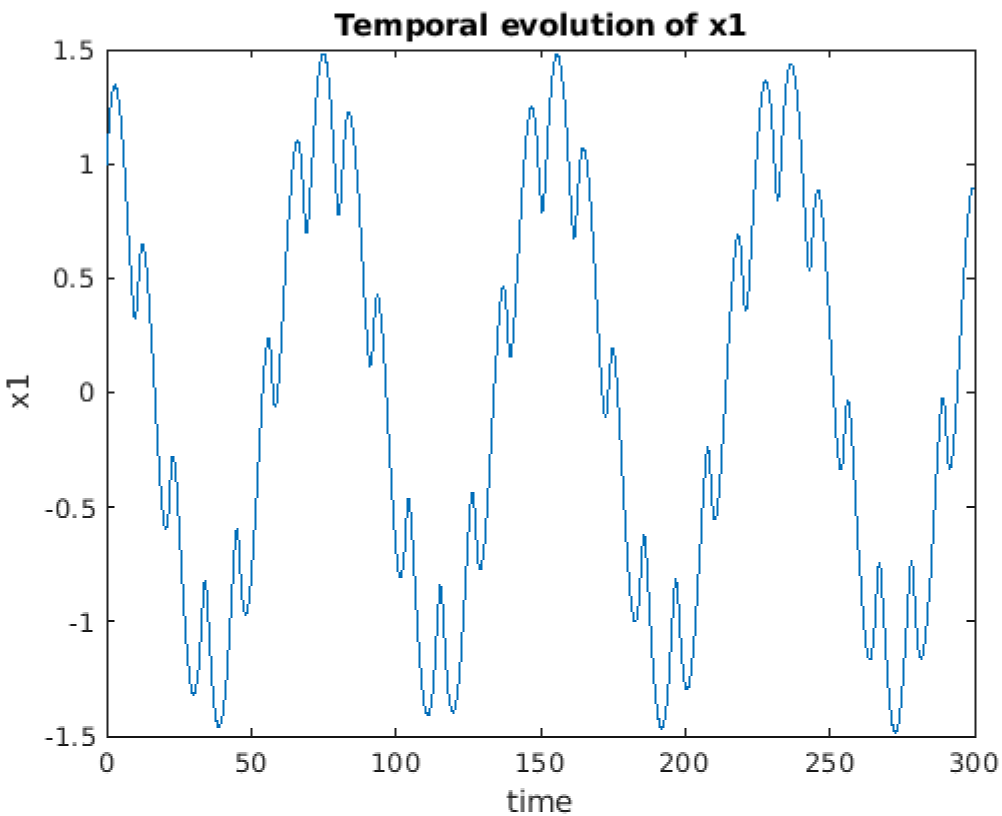
involves only the left-hand-syde and is $\rho(\mu) = \mu^2 - \mu$, with roots 0 and 1. The method is stable, because all roots satisfy $|\mu| \leq 1$, and the roots of modulus one ($\mu = 1$) are simple.

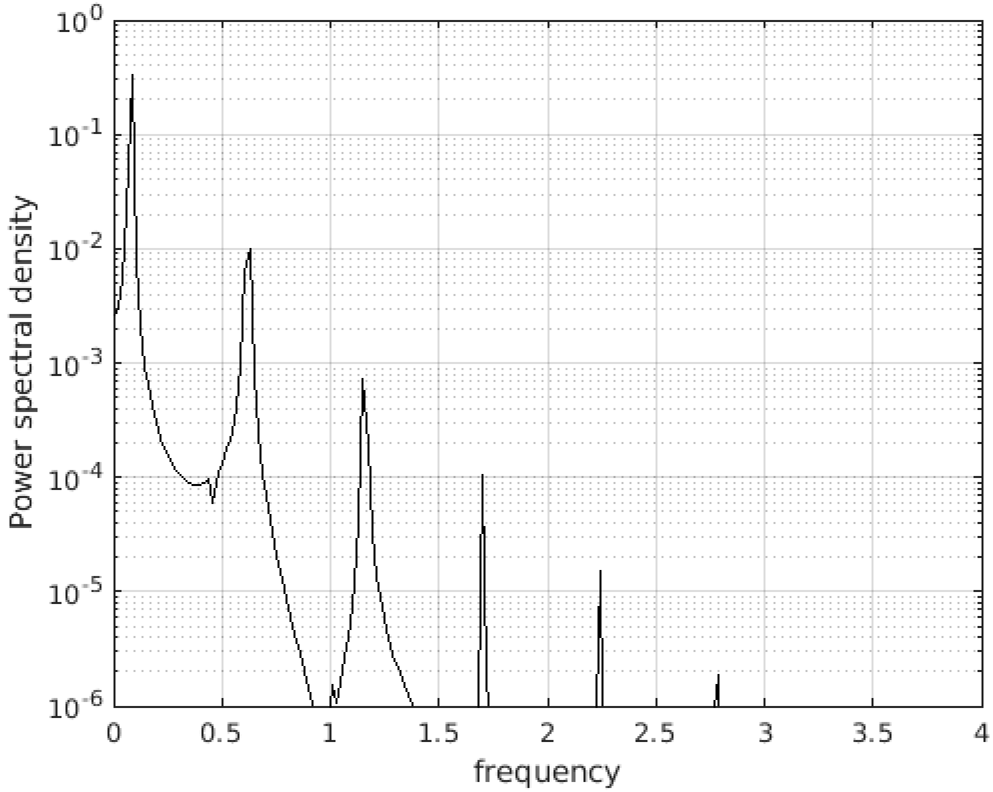
```
%%Second problem final exam

clear all
close all
clc;

format long

%System of ODEs to be solved

% x1'= x3
% x2'= x4
% x3'= -x4/(x1^2+x2^2)^*3/2
% x4'= x3/(x1^2+x2^2)^*3/2

%Calculate initial energy and momentum
x1 = 1; x2 = 0; x3 = 0.22; x4=0; r = sqrt(x1^2+x2^2);
e0 = x3^2+x4^2;
l0 = 1/r+x1*x4-x2*x3;

%set initial condition vector
initial_v = [x1;x2;x3;x4]';

%%Find time step size dt to satistfy the abs(e-e0) < 2e-6 and abs(l-
l0) <
%%2e-6 (with two digits of accuracy)

%%set loop condition as true
i = 0;
%%set initial time step size
dt = 1;

while i == 0
  %%Set initial time to zero and number of integration steps
  t0 = 0; Nint = 1000;
  [X,T] = RK4(@rhs,t0,dt,initial_v,Nint-1);
  %%Calculate energy and angular momentum
  x1 = X(1,end); x2 = X(2,end); x3 = X(3,end); x4=X(4,end); r =
 sqrt(x1^2+x2^2);
  e = x3^2+x4^2;
  l = 1/r+x1*x4-x2*x3;
  ne = abs(e-e0);
  nl = abs(l-l0);
  if (ne > 2e-6) | (nl > 2e-6)
      dt = dt - 0.01;
  else
      i = 1;
  end
end

Y = sprintf('The time step size required to satisfy both conditions is
 %d.',dt);
```

```
disp(Y)

%%Integrate over 3000 steps

Nint = 3000;
[X,T]= RK4(@rhs,t0,dt,initial_v,Nint-1);

%%Plot the trajectory on the x,y plane
figure(1)
plot(X(1,:),X(2,:))
title('Trajectory on the (x1,x2) plane')
xlabel('x1')
ylabel('x2')

%%Calculate energy as a function of time

%%Time series for x1(t)
figure(2)
plot(T,X(1,:))
title('Temporal evolution of x1')
xlabel('time')
ylabel('x1')
%%Calculate discrete Fourier transform

fk = dftmat(X(1,:)');

%%Plot power spectral density vs frequency

N = Nint;
Tm = dt*N ;
wk = 2*pi*[0:N/2-1]/Tm;
figure(3)
semilogy(wk,abs(fk(N/2+1:N)).^2,'-k') ;
axis([0, 4, 1.e-6, 1]); grid on
xlabel('frequency')
ylabel('Power spectral density')

%%The two frequencies with the largest amplitude (obtained from the
 figure (3)) are
w1 = 0.084;
w2 = 0.628;
```

The time step size required to satisfy both conditions is 1.000000e-01.


Trajectory on the (x1,x2) plane
1.5
1
0.5
0
-0.5
-1
-1.5
x2
-1.5
-1
-0.5
0
0.5
1
1.5
x1


Temporal evolution of x1
1.5
1
0.5
0
-0.5
-1
-1.5
x1
0
50
100
150
200
250
300
time



Published with MATLAB® R2020b