

MECH309
Projects in Numerical Methods
Numerical solutions to Ordinary Differential Equations

Due date: April 16, 2012

Instructions:

- Up to 3 students working together.
- Write a report (word, LaTeX, LibreOffice...) with comments, equations, figures, and algorithms and send a final pdf file to mathias.legrand@mcgill.ca. It should not exceed 2 Mb. Family names must be part of the file name.
- Always mention "Mech 309 — Projects — Family names" in the subject of your email.
- Internal Matlab commands for ODEs are forbidden unless otherwise stated.
- If you face difficulties, do not hesitate to contact me.

1 One-dimensional Initial Value Problem

1.1 Autonomous ordinary differential equation

Consider the following initial value problem:

$$\begin{cases} \frac{dy}{dt} = \cos y & \text{defined on } D \\ y(0) = -1 \end{cases} \quad (1)$$

where the domain of interest is $D = \{(t, y) \mid 0 \leq t \leq 10; -\infty < y < \infty\}$.

1. Prove existence and uniqueness of a solution $y(t)$ on D . Is the problem well-posed?
2. Implement Euler's method in matlab and solve the ODE numerically. Provide an upper bound of the global truncation error. Implement the Runge-Kutta method of order 4 in matlab and solve the ODE numerically. Plot the two solutions in the vector field.¹
3. Find the closed-form solution to this ODE. Perform calculations (Euler and Runge-Kutta) for decreasing time-steps:
 - Euler: $h = 1, h = 10^{-1}, h = 10^{-2}, h = 10^{-3}, h = 10^{-4}, h = 10^{-5}, h = 10^{-6}, h = 10^{-7}$ (if possible for the last one).
 - Runge-Kutta: $h = 1, h = 10^{-1}, h = 10^{-2}, h = 10^{-3}, h = 10^{-4}, h = 10^{-5}$.

On the same figure, plot on a log – log scale the maximum of the local and global truncation errors on $t \in [0, 10]$ with respect to h for both methods. Comment.

4. Find an upper-bound of the global truncation error based on the exact solution. Compare with the previous upper bound. Comment.

¹The matlab command to plot the vector field of an ODE $y' = f(t, y)$ is:

```
[t,y]=meshgrid(0:.4:9.8,-3:.4:3); % modify to your convenience
dy=f(t,y);
dt=ones(size(dy));
dyu=dy./sqrt(dt.^2+dy.^2);
dtu=dt./sqrt(dt.^2+dy.^2);
quiver(t,y,dtu,dyu);
```

1.2 Time-dependent ordinary differential equation

Consider the following initial value problem:

$$\begin{cases} \frac{dy}{dt} = t \cos y & \text{defined on } D \\ y(0) = -1 \end{cases} \quad (2)$$

where the domain of definition is $D = \{(t, y) \mid 0 \leq t \leq 10; -\infty < y < \infty\}$.

1. Answer questions 1., 2., 3., and 4. from subsection 1.1 for this new Initial Value Problem.
2. Consider $D = \{(t, y) \mid 0 \leq t \leq 100; -\infty < y < \infty\}$. Solve with Runge-Kutta and $h = 0.01$. Solve again with Runge-Kutta and $h = 0.001$. Compare with the previous ODE and comment.
3. Solve with command ode45 in Matlab. Plot the time vector T (when $[T, Y] = \text{ode45}(\dots)$ is used). Comment. Can we numerically solve this ODE with t very large?

2 Harmonic Balance Method

As opposed to the Euler's and Runge-Kutta methods that are *time-marching* techniques, the Harmonic Balance Method (HBM) is able to directly calculate the steady-state response of a possibly non-linear and periodically forced mechanical system. By representing the unknowns (displacement in what follows) in the Fourier space through a chosen number of harmonics, it provides a direct access to a steady-state, periodic solution to the equation of motion. By discarding the potentially long transient response, the harmonic balance method offers a very efficient alternative to time-domain methods for vibration analysis. It is here used to investigate the forced response of the 2-degree-of-freedom system illustrated in Figure 1.

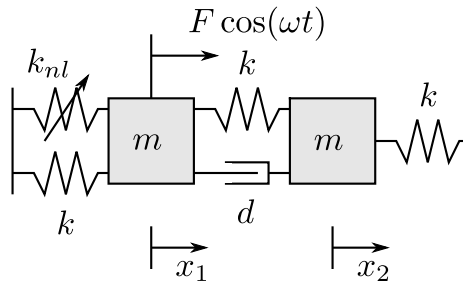


Figure 1: 2 DOF nonlinear oscillator

2.1 Equations of motion

The spring denoted k_{nl} in Figure 1 is a non-linear spring acting in such a way that its displacement-force relationship is:

$$F_{nl}(x_1) = k_{nl}(x_1)x_1 = \epsilon x_1^3 \quad (3)$$

Based on this definition:

1. Derive the two equations of motion (we will name them Eq[1] and Eq[2]) driving the investigated non-linear oscillator.
2. Recast them in a system of first-order ordinary differential equations.
3. Implement a Runge-Kutta method of order 4 to solve these equations.
4. Application: solve with the following parameters: $m = 1$, $d = 0.01$, $k = 1$, $\epsilon = 0.5$, $\omega = 1.1$, and $F = 0.1$ and initial conditions:

- set 1: $(x_1(0), x_2(0)) = (-1, -1)$ and $(\dot{x}_1(0), \dot{x}_2(0)) = (0, 0)$;
- set 2: $(x_1(0), x_2(0)) = (1, 1)$ and $(\dot{x}_1(0), \dot{x}_2(0)) = (0, 0)$;

5. Comment.

2.2 Frequency-domain formulation

In the frequency domain, the two displacements are approximated by the following truncated Fourier series:

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \cos(\omega t) + \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} \sin(\omega t) + \begin{pmatrix} a_3 \\ b_3 \end{pmatrix} \cos(3\omega t) + \begin{pmatrix} a_4 \\ b_4 \end{pmatrix} \sin(3\omega t) \quad (4)$$

Expansion 4 is substituted into the two equations of motion previously derived and then Harmonic Balancing is performed. This consists of performing a *projection* of the two governing equations onto the considered Fourier basis $(\cos(\omega t), \sin(\omega t), \cos(3\omega t), \sin(3\omega t))$, ie for $i = 1, 2$:

$$\begin{aligned} \int_0^{2\pi/\omega} \text{Eq}[i] \cos(\omega t) dt &= 0 \\ \int_0^{2\pi/\omega} \text{Eq}[i] \sin(\omega t) dt &= 0 \\ \int_0^{2\pi/\omega} \text{Eq}[i] \cos(3\omega t) dt &= 0 \\ \int_0^{2\pi/\omega} \text{Eq}[i] \sin(3\omega t) dt &= 0 \end{aligned} \quad (5)$$

yielding a set of eight nonlinear algebraic equations in the eight unknowns $(a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4)$.

1. Establish the system of eight nonlinear equations.
2. Application: consider the same parameters as in the time integration with the following initial guesses and solve with the numerical method of your choice used in project 1 (Basins of attraction):
 - set 1: $(a_i, b_i) = (0.1, 0.1)$ for $i = 1, \dots, 4$
 - set 2: $(a_i, b_i) = (0.5, 0.5)$ for $i = 1, \dots, 4$
3. Plot in time domain. Compare to time domain results.
4. Comment.

Hints Maxima, Maple, or Mathematica may be used to obtain the eight algebraic equations.