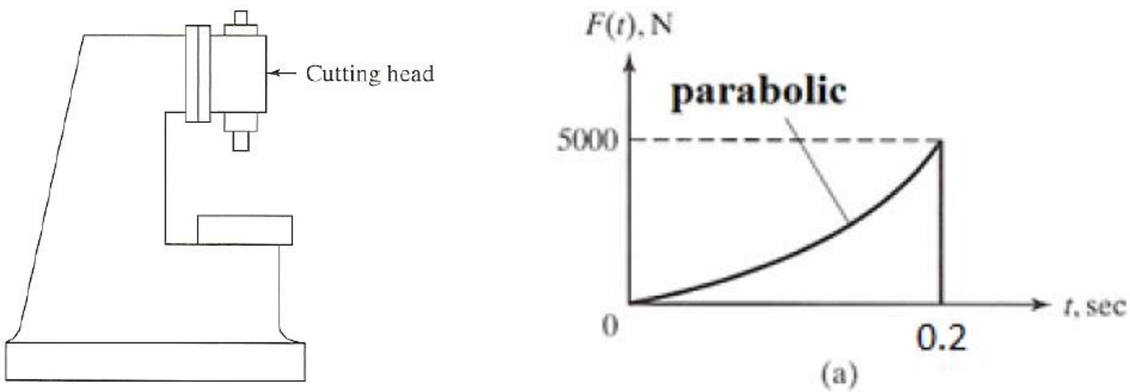
David Gaset

EML 4220 - Vibration Analysis
Fall Semester 2013
(Dr. F. Moslehy)

Computer Project

The cutting force developed during a particular machining operation is shown in figure (a). Model the system as a SDF with equivalent mass of the cutting head = 23 kg. The damping is linear viscous (ς = 12%), and the equivalent spring is nonlinear "hardening" spring such that the spring force is expressed by $F_s = k_1 x + k_2 x^3$,where k_1 = 380 kN/m, and k_2 = 40 kN/m³.
Assume the initial conditions $x(0) = 0$ and. $\dot{x}(0) = 0$.

Compute the inaccuracies (in the vertical direction) in the surface finish due to the cutting force.



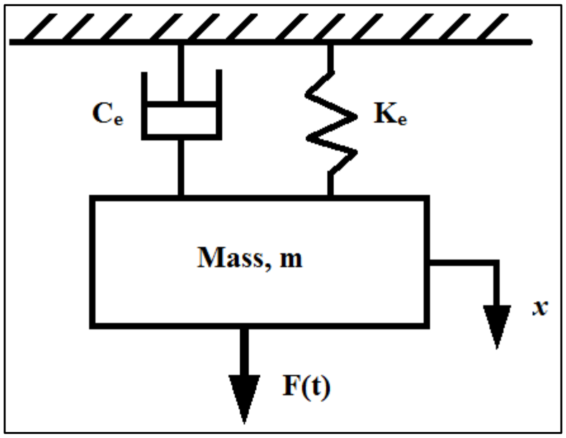
(a)

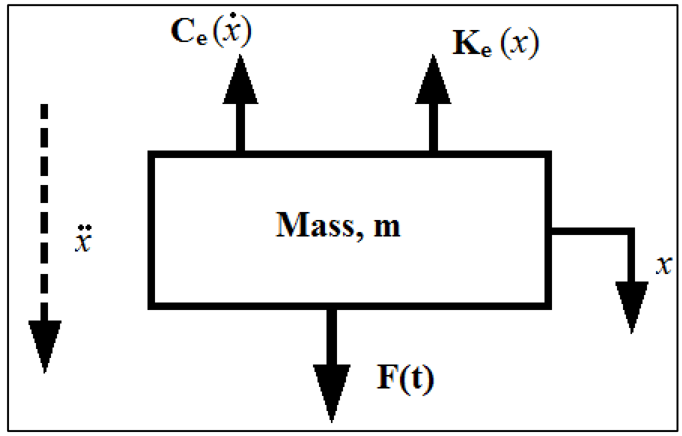
Write a computer program based on the RK-4 method to solve the following *nonlinear* problem.

Present your results as follows:

(a) Draw the analytical model and write the D.E. of motion of the system.
(b) Show how you selected the time increment for the RK-4 program.
(c) "Computer" printout of the program listing.
(d) "Computer" output of results $(t, x, \dot{x})$.
(e) "Computer" plot for $x(t)$ vs. t.
(f) "Computer" plot $\dot{x}(t)$ vs. t.
(g) "Computer" plots for $\dot{x}$ vs. x.
(h) Determine the maximum inaccuracy (x_{max}).

(a) Analytical Model and D.E. of Motion for the System:





The analytical model, seen above to the left, will represent the cutting mechanical operation. Where, F(t), K_e, and C_e, denote the given applied force, equivalent spring constant, and damping coefficient respectively. Using the analytical model, a free-body diagram (F.B.D) is developed, seen above to the right. For both diagrams ***x***, $\dot{x}$, and $\ddot{x}$ denote the positive direction of the system's displacement, velocity, and acceleration.

MathCad was utilized for computation and developing the needed program to solve the problem. Using the given information and assumption, $K_e = K_1$ since $K_1 << K_2$, equations of the cutting, spring, and damping force's with respect to time are formulated. These equations along with other given values are symbolically defined and can be seen below.

Equivalent mass:

$$\text{mass} = m$$
$$m := 23$$

Damping Force:

$$k_e := K_1$$
$$c_e := \zeta \cdot 2 \cdot \sqrt{k_e \cdot m}$$
$$c_e = 709.523784$$
$$F_d(y) := c_e \cdot y$$

Non-linear "hardening" Spring:

$$\text{spring\_force} = F_s$$
$$K_1 := 380000 \quad K_2 := 40000$$
$$F_s(x) := K_1 \cdot (x) + K_2 \cdot (x)^3$$

Cutting Force:

$$A := \frac{5000}{0.2^2}$$
$$A = 125000$$
$$F(t) := A \cdot t^2$$

Applying Newton's 2^{ed} Law to the (F.B.D), where the summation of forces acting on a system is equivalent to the product of the systems mass and acceleration, two sets of first order differential equations of motion are developed as a function of the systems acceleration. The first set, represents the time period in which the cutting force is applied and the latter for the time following the cutting force, both seen to the right. Where x, y, and t denote displacement, velocity and time respectively.

Motion Equation (0< t <0.2):

$$f1(x,y,t) := y$$
$$f2(x,y,t) := \frac{1}{m}\left(F(t) - F_d(y) - F_s(x)\right)$$

Motion Equation (t >0.2):

$$F1(x,y,t) := y$$
$$F2(x,y,t) := \frac{1}{m}\left(-F_d(y) - F_s(x)\right)$$

(b) Selection of Time Increment for Runge Kutta 4th Order Method:

When developing a program utilizing the Runge-Kutta 4th Order Method, it is necessary to define a step increment of time at which the program will find a solution. For the given problem, the step increment is approximated between the natural period (τ_n) divided by 50 and 100. This rage is found, seen to the right (Finding Step), denoted by $Step_{low}$ and $Step_{high}$. Where ω_n, denotes the natural frequency of the system.

Finding Step:

$$\omega_n := \sqrt{\frac{k_e}{m}} \qquad \omega_n = 128.5369$$

$$\tau_n := \frac{2\cdot\pi}{\omega_n} \qquad \tau_n = 0.0489$$

$$Step_{low} := \frac{\tau_n}{50} \qquad Step_{low} = 0.00098$$

$$Step_{high} := \frac{\tau_n}{100} \qquad Step_{high} = 0.00049$$

Set Up for Code:

$$\Delta t := \frac{2}{3000} \qquad \Delta t = 0.000667$$

$$N := 300 \qquad F := 900$$

$$j := 0, 1 .. N \qquad i := N, N+1 .. F$$

$$t_j := j\cdot\Delta t \qquad t_i := i\cdot\Delta t$$

$$t_N = 0.2 \qquad t_F = 0.6$$

The step increment of time (Δt) chosen for the program is defined to the left (Set Up for Code). Also defined, time (t) with respect to the step increments j and i, as well as the increment (N) of j, which correlates to the time at which the cutting force ends (0.2s) and the increment (F) of i which correlates to the time the program will stop (0.6s).

(c) Printout of Program Listing:

The program developed utilizing the Runge-Kutta 4th Order Method, can be found on the following page. The program consist of two for loops, the first for the time period in which the cutting force is applied and the second for the time following the cutting force. The output of the developed program yields the displacement and velocity at each step increment of time. These output values are then defined, Displacement and Velocity respectively, also seen on the following page

$$
\text{Solution} := \left|
\begin{array}{l}
x_0 \leftarrow 0 \\
y_0 \leftarrow 0 \\
\text{for } i \in 0,1..N \\
\quad \left|
\begin{array}{l}
K11 \leftarrow \Delta t \cdot f1(x_i, y_i, t_i) \\
K12 \leftarrow \Delta t \cdot f2(x_i, y_i, t_i) \\
K21 \leftarrow \Delta t \cdot f1\left(x_i + \frac{1}{2} \cdot K11, y_i + \frac{1}{2} \cdot K12, t_i + \frac{1}{2} \cdot \Delta t\right) \\
K22 \leftarrow \Delta t \cdot f2\left(x_i + \frac{1}{2} \cdot K11, y_i + \frac{1}{2} \cdot K12, t_i + \frac{1}{2} \cdot \Delta t\right) \\
K31 \leftarrow \Delta t \cdot f1\left(x_i + \frac{1}{2} \cdot K21, y_i + \frac{1}{2} \cdot K22, t_i + \frac{1}{2} \cdot \Delta t\right) \\
K32 \leftarrow \Delta t \cdot f2\left(x_i + \frac{1}{2} \cdot K21, y_i + \frac{1}{2} \cdot K22, t_i + \frac{1}{2} \cdot \Delta t\right) \\
K41 \leftarrow \Delta t \cdot f1(x_i + K31, y_i + K32, t_i + \Delta t) \\
K42 \leftarrow \Delta t \cdot f2(x_i + K31, y_i + K32, t_i + \Delta t) \\
x_{i+1} \leftarrow x_i + \frac{1}{6}(K11 + 2 \cdot K21 + 2 \cdot K31 + K41) \\
y_{i+1} \leftarrow y_i + \frac{1}{6}(K12 + 2 \cdot K22 + 2 \cdot K32 + K42)
\end{array}
\right. \\
x_{N+1} \leftarrow x_N \\
y_{N+1} \leftarrow y_N \\
\text{for } i \in N+1, N+2..F \\
\quad \left|
\begin{array}{l}
K11_2 \leftarrow \Delta t \cdot F1(x_i, y_i, t_i) \\
K12_2 \leftarrow \Delta t \cdot F2(x_i, y_i, t_i) \\
K21_2 \leftarrow \Delta t \cdot F1\left(x_i + \frac{1}{2} \cdot K11_2, y_i + \frac{1}{2} \cdot K12_2, t_i + \frac{1}{2} \cdot \Delta t\right) \\
K22_2 \leftarrow \Delta t \cdot F2\left(x_i + \frac{1}{2} \cdot K11_2, y_i + \frac{1}{2} \cdot K12_2, t_i + \frac{1}{2} \cdot \Delta t\right) \\
K31_2 \leftarrow \Delta t \cdot F1\left(x_i + \frac{1}{2} \cdot K21_2, y_i + \frac{1}{2} \cdot K22_2, t_i + \frac{1}{2} \cdot \Delta t\right) \\
K32_2 \leftarrow \Delta t \cdot F2\left(x_i + \frac{1}{2} \cdot K21_2, y_i + \frac{1}{2} \cdot K22_2, t_i + \frac{1}{2} \cdot \Delta t\right) \\
K41_2 \leftarrow \Delta t \cdot F1(x_i + K31_2, y_i + K32_2, t_i + \Delta t) \\
K42_2 \leftarrow \Delta t \cdot F2(x_i + K31_2, y_i + K32_2, t_i + \Delta t) \\
x_{i+1} \leftarrow x_i + \frac{1}{6}(K11_2 + 2 \cdot K21_2 + 2 \cdot K31_2 + K41_2) \\
y_{i+1} \leftarrow y_i + \frac{1}{6}(K12_2 + 2 \cdot K22_2 + 2 \cdot K32_2 + K42_2)
\end{array}
\right. \\
\begin{pmatrix} x \\ y \end{pmatrix}
\end{array}
\right.
= \begin{pmatrix} \{902,1\} \\ \{902,1\} \end{pmatrix}
$$

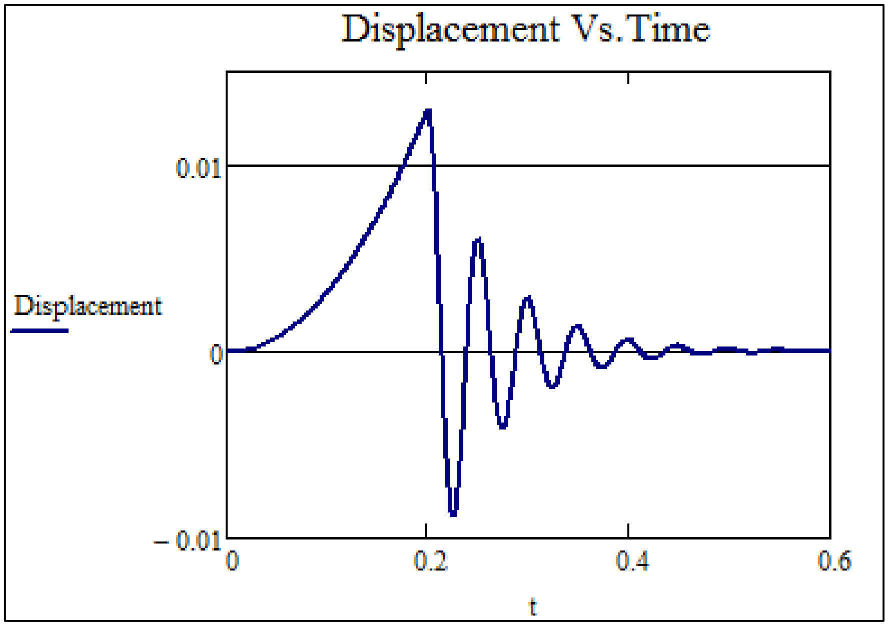
$$
\begin{pmatrix} \text{Displacement} \\ \text{Velocity} \end{pmatrix} := \text{Solution}
$$

(d) Results:

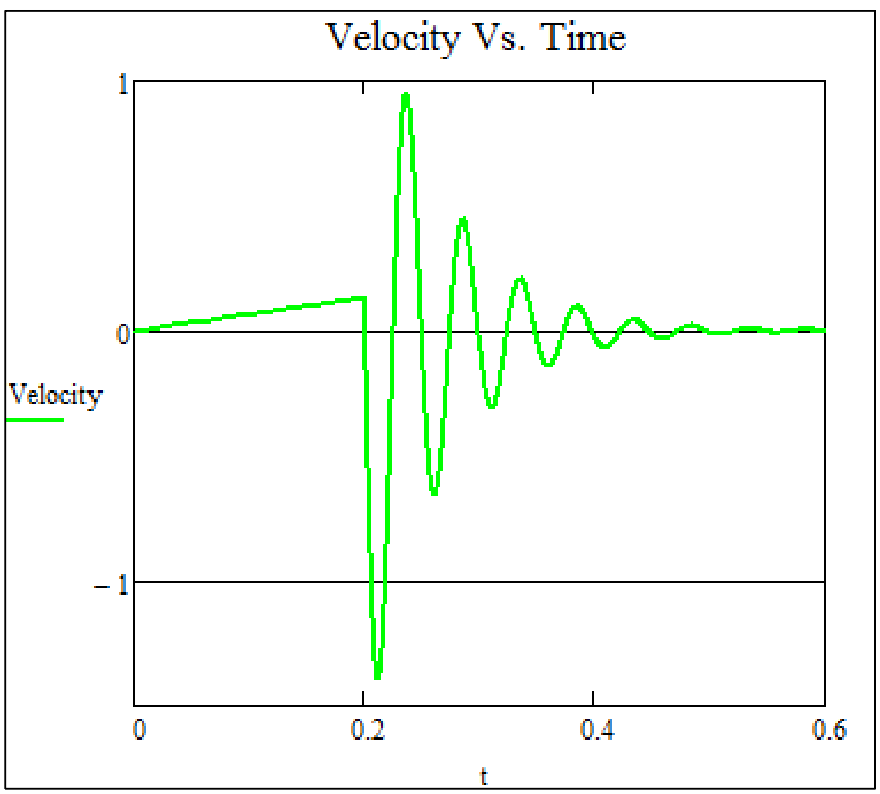
The Table seen below, contains the found results of displacement (*m*) and velocity ($^{m}/_{s}$) with respect to time (s). For practicality, since the developed program yields 900 results, every 30th solution is displayed within the graph.

Time	Displacement	Velocity
0	0	0
0.02	5.24E-05	9.14E-03
0.04	4.40E-04	0.028
0.06	1.08E-03	0.036
0.08	1.96E-03	0.052
0.1	3.14E-03	0.065
0.12	4.55E-03	0.077
0.14	6.24E-03	0.091
0.16	8.19E-03	0.104
0.18	0.01	0.117
0.2	0.013	0.13
0.22	-6.27E-03	-0.856
0.24	7.81E-04	0.896
0.26	2.43E-03	-0.633
0.28	-3.39E-03	0.289
0.3	2.83E-03	-0.012
0.32	-1.62E-03	-0.142
0.34	4.58E-04	0.179
0.36	3.17E-04	-0.143
0.38	-6.34E-04	0.077
0.4	6.03E-04	-0.018
0.42	-3.95E-04	-0.02
0.44	1.56E-04	0.034
0.46	2.19E-05	-0.031
0.48	-1.11E-04	0.019
0.5	1.24E-04	-6.82E-03
0.52	-9.14E-05	-2.05E-03
0.54	4.47E-05	6.19E-03
0.56	-5.30E-06	-6.45E-03
0.58	-1.77E-05	4.54E-03
0.6	2.44E-05	-2.06E-03

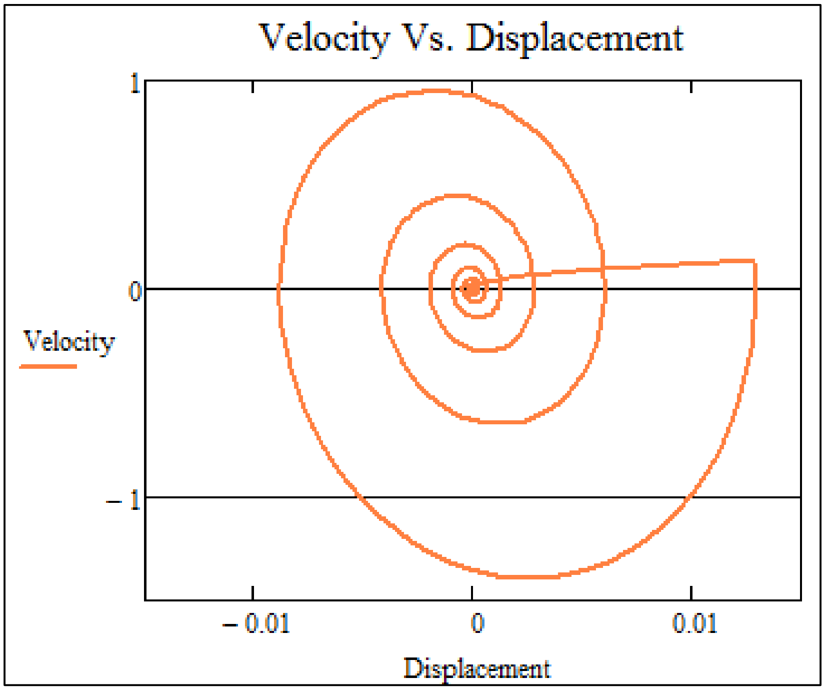
(e)


Displacement Vs.Time
Displacement
0.01
0
− 0.01
0
0.2
0.4
0.6
t

(f)


Velocity Vs. Time
Velocity
1
0
− 1
0
0.2
0.4
0.6
t

(g)



(h) <u>Maximum Inaccuracy:</u>

The maximum inaccuracy, symbolically defined to and found on the right, is equivalent to the maximum displacement of the system. Since the maximum displacement occurs when the cutting force is at its maximum with respect to time. The maximum is found to be the displacement at step time increment N.

<u>Maximum Inaccuracy:</u>

$$Max_{inaccuracy} := Displacement_N$$

$$Max_{inaccuracy} = 0.012876$$