```
% ### EXduffing.m ###       09.19.14 CB (revised
2020.01.30)

% Purpose: Numerically integrate the driven damped duffing
oscillator
%   m*x''+ b*x' + k*x + c*x^3 = A*sin(wt)

% Notes
% o Default params taken from wikipedia page re "Duffing
Equation"
% (presumably as they point to a chaotic regime)

clear
% -------------------------------------------------------
% User input (Note: All paramters are stored in a
structure)
P.y0(1) = -1.0;   % initial position [m]
P.y0(2) = 0.0;   % initial velocity [m/s]
P.b= 0.02;  % damping coefficient [kg/s]
P.k= 1.0;   % stiffness [N m]
P.m= 1;    % mass [kg]
P.c= 5;       % nonlinear stiffness term
% --- (sinusoidal) driving term
P.A= 8.0;   % amplitude [N] (set to zero to turn off) {8}
fD= 0.5/(2*pi);  % freq. (Hz) [expressed as fraction of
resonant freq.] {0.5}
% --- Integration limits
P.t0 = 0.0;   % Start value
P.tf = 100.0;   % Finish value
P.dt = 0.01;  % time step
%
-----------------------------------------------------------
-----------
% +++ bookeeping + display some basic derived quantities
P.wr= 2*pi*fD;  % convert to angular freq.
disp(sprintf('Resonant frequency ~%g [Hz]', sqrt(P.k/P.m)/
(2*pi)));
Q = (sqrt(P.k/P.m))/(P.b/P.m);  % quality factor
disp(sprintf('Q-value = %g', Q));
% +++ use built-in ode45 to solve
[t y] = ode45('DUFFfunction', [P.t0:P.dt:P.tf],P.y0,[],P);
```

```
% ----------------------------------------------------------
% --- visualize timecourse
figure(1); clf; plot(t,y(:,1)); hold on; grid on;
xlabel('t [s]');    ylabel('x(t) [m]')
% --- Phase plane
figure(2); clf;
plot(y(:,1), y(:,2)); hold on; grid on;
xlabel('x [m]');    ylabel('dx/dt [m/s]')
```