Engineer's Computation Pad
No. 937 811E
STAEDTLER®

16.61 LECTURE #6

NUMERICAL SOLUTION OF NONLINEAR DIFFERENTIAL EQUATIONS.

NUMERICAL SOLUTION

- GIVEN A COMPLEX SET OF DYNAMICS

$$\ddot{x}(t) = F(x, \dot{x})$$

WHERE $F(\cdot)$ COULD BE A NONLINEAR FUNCTION IT CAN BE IMPOSSIBLE TO ACTUALLY SOLVE FOR $x(t)$ EXACTLY.

$\Rightarrow$ DEVELOP A NUMERICAL SOLUTION.

- "CANNED" CODES TO HELP US DO THIS IN MATLAB®, BUT LET US CONSIDER THE BASICS.

- APPROXIMATE THE DERIVATIVES WITH BACKWARD DIFFERENCES:

$$\dot{x}(kT) \simeq \frac{x(kT) - x((k-1)T)}{T} = \frac{x_k - x_{k-1}}{T}$$

T - SMALL FIXED TIME PERIOD

K - INTEGER INDEX

$$\ddot{x}(kT) \simeq \frac{\dot{x}(kT) - \dot{x}((k-1)T)}{T}$$

$$\simeq \frac{\left[x(kT) - x((k-1)T)\right] - \left[x((k-1)T) - x((k-2)T)\right]}{T^2}$$

$$= \frac{x_k - 2x_{k-1} + x_{k-2}}{T^2}$$

- SO, IF WE HAD $\ddot{X} = -3X - 4\dot{X}$

WE COULD APPROXIMATE THIS AS:

$$\frac{X_K - 2X_{K-1} + X_{K-2}}{T^2} = -3X_K - 4\,\frac{X_K - X_{K-1}}{T}$$

$$X_K - 2X_{K-1} + X_{K-2} = -3T^2 X_K - 4T\left(X_K - X_{K-1}\right)$$

$$\left(1 + 4T + 3T^2\right) X_K = \left(2 + 4T\right) X_{K-1} - X_{K-2}$$

$$X_K = \frac{\left(2+4T\right) X_{K-1} - X_{K-2}}{1+3T^2+4T}$$

- CALLED A <u>RECURSION</u> <u>RELATION</u>

- GIVEN X_{K-1}, X_{K-2} WE CAN FIND X_K

  $\Rightarrow$ THEN USE X_{K-1}, X_K TO FIND X_{K+1} ...

- HOW DO WE START?

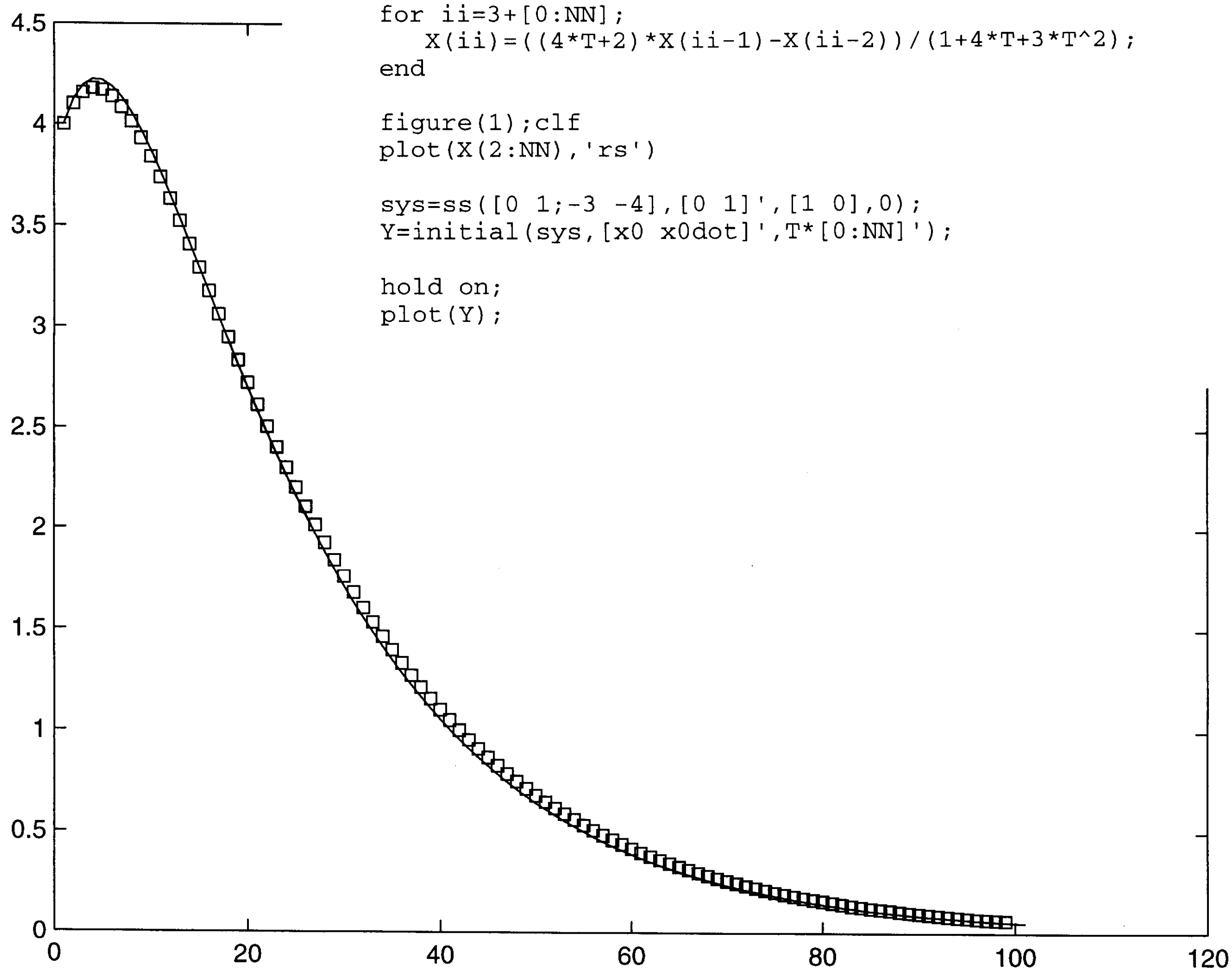
  IF $x(0) = 4$; $\dot{x}(0) = 3$

  $\Rightarrow$ $x(0) = 4$ AND $X_0 - X_{-1} = 3T$; $X_{-1} = 4 - 3T$

- SIMPLE APPROACH, BUT LIMITED ACCURACY

  - KEEP T SMALL.

```
% 16.61 - Numerical example for \ddot x + 4 \dot x + 3 x = 0
% Prof. How

clear all
T=0.05;
% actual IC
x0=4; x0dot=3;
% start numerics
xm1=x0-x0dot*T;

NN=100;
X=[xm1 x0];
for ii=3+[0:NN];
   X(ii)=((4*T+2)*X(ii-1)-X(ii-2))/(1+4*T+3*T^2);
end

figure(1);clf
plot(X(2:NN),'rs')

sys=ss([0 1;-3 -4],[0 1]',[1 0],0);
Y=initial(sys,[x0 x0dot]',T*[0:NN]');

hold on;
plot(Y);
```



- IN THE CASE THAT $F(x,\dot{x})$ IS LINEAR, WE CAN SOLVE THE EOM IN MATLAB® USING <u>LSIM</u>

  - OFTEN FIND THAT LINEAR DYNAMICS COUPLE MORE THAN ONE VARIABLE

  ⇒ CAN ALWAYS WRITE THE DYNAMICS AS

$$\dot{X} = A X + B u$$

  WHERE X IS A VECTOR OF VARIABLES

  ⇒ CALLED THE <u>STATE</u>

<u>EXAMPLE</u>: HILL'S EQUATIONS FOR TWO CLOSELY SPACED SPACECRAFT: $n \sim 2\pi / 90$ mins

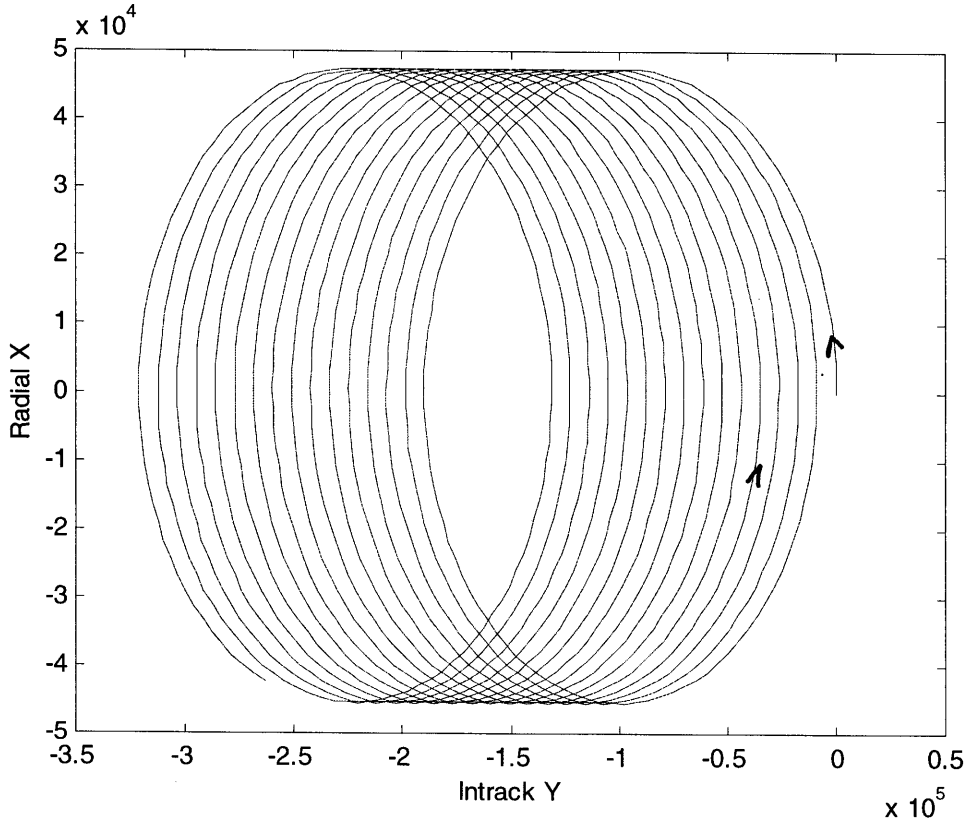
$$\ddot{x} = 2n\dot{y} + 3n^2 x + f_x$$
$$\ddot{y} = -2n\dot{x} + f_y$$

y - INTRACK

x - RADIAL

LET $\mathbf{X} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} \Rightarrow \dot{\mathbf{X}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ +3n^2 & 0 & 0 & +2n \\ 0 & 0 & 0 & 1 \\ 0 & -2n & 0 & 0 \end{bmatrix} \mathbf{X} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix}$

- LINEAR MATRIX FORM THAT CAN BE DIRECTLY SOLVED IN MATLAB®.

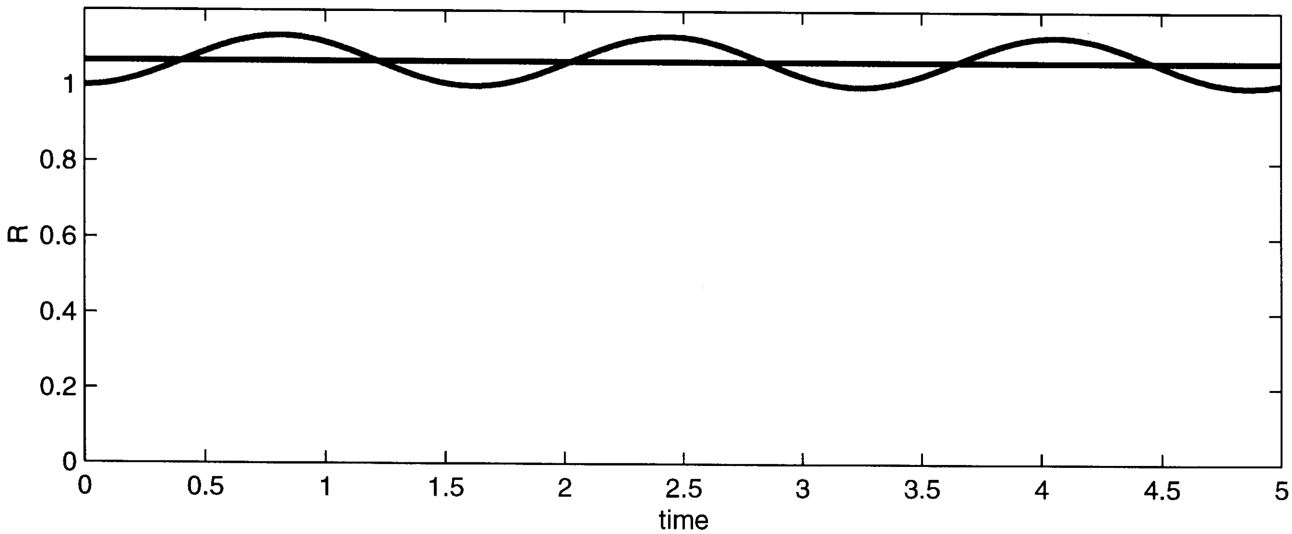



```
% LSIM model propagated using LSIM
% 16,61 Spr 02
% Jonathan How
%
n=2*pi/(90*60);
A=[0 1 0 0;3*n^2 0 0 2*n;0 0 0 1;0 -2*n 0 0];
B=[0 0;1 0;0 0;0 1];
C=[1 0 0 0;0 0 1 0];
D=zeros(2,2);
Hills=ss(A,B,C,D);

% time
t=[0:.01:5*pi]*90*60;
% control inputs
U=[ones(length(t),2)*0];U(1,1)=1;U(1,2)=.01;
% initial conditions
X0=[0 0 100 0]';
% simulations
[Y,T] = lsim(Hills,U,t,X0);

figure(1)
plot(Y(:,2),Y(:,1));
xlabel('Intrack Y')
ylabel('Radial X')
```

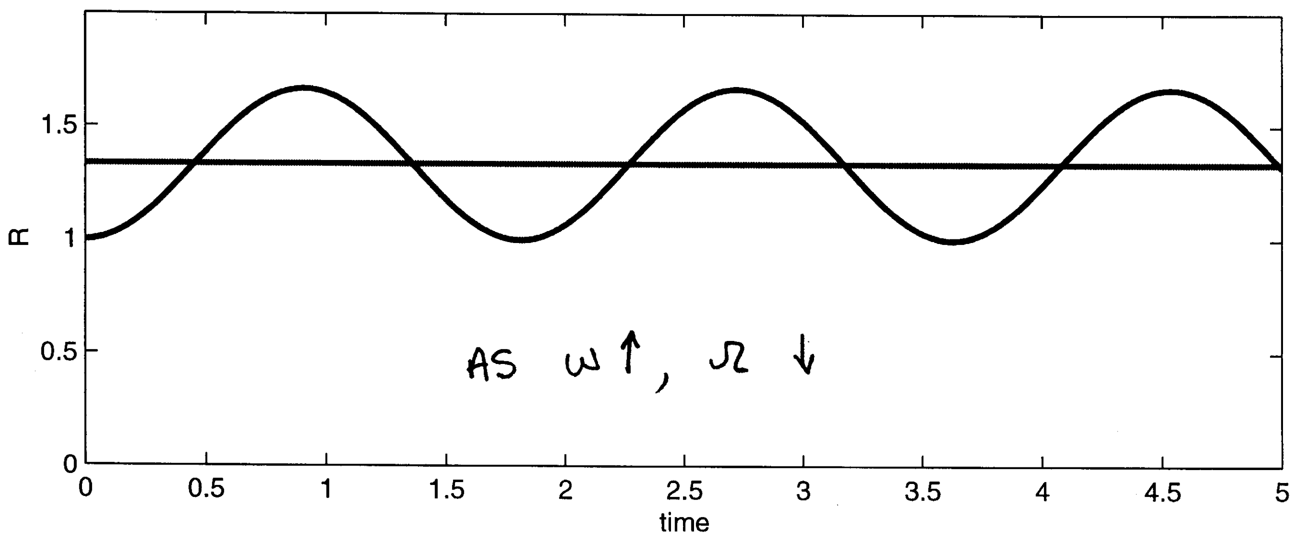
```
function y=mgr1(w);
% 16.61 - MGR case with radial MSD onboard
% Page 3-4 of notes
% Prof. How
% Call using lines at the bottom after the "return"
%
if nargin < 1;w=2;end

R0=1; % radial offset
K=8;M=1; % random model parameters
% state space model of the dynamics
% state is [R; \dot R]
A=[0 1;-(2*K/M-w^2) 0];
B=[0 R0*w^2]';
C=eye(2);
D=[0 0]';
% weird matlab form
sys=ss(A,B,C,D);
T=[0:.01:5];
% Linear model SIMulation
RR=lsim(sys,ones(size(T)),T);
% basic offset is R0, so y_actual=R0+R

figure(1);%clf
plot(T,R0+RR(:,1),[min(T) max(T)],R0+R0/(2*K/M/w^2-1)*[1 1],'LineWidth',2);
title(['Freq = ',num2str(w)])
axis([min(T) max(T) 0 1.5*max(RR(:,1))+R0])
xlabel('time')
ylabel('R')

return
subplot(311);mgr1(1);subplot(312);mgr1(2);subplot(313);mgr1(3.8);
```

6-7

Freq = 1 $= \omega$

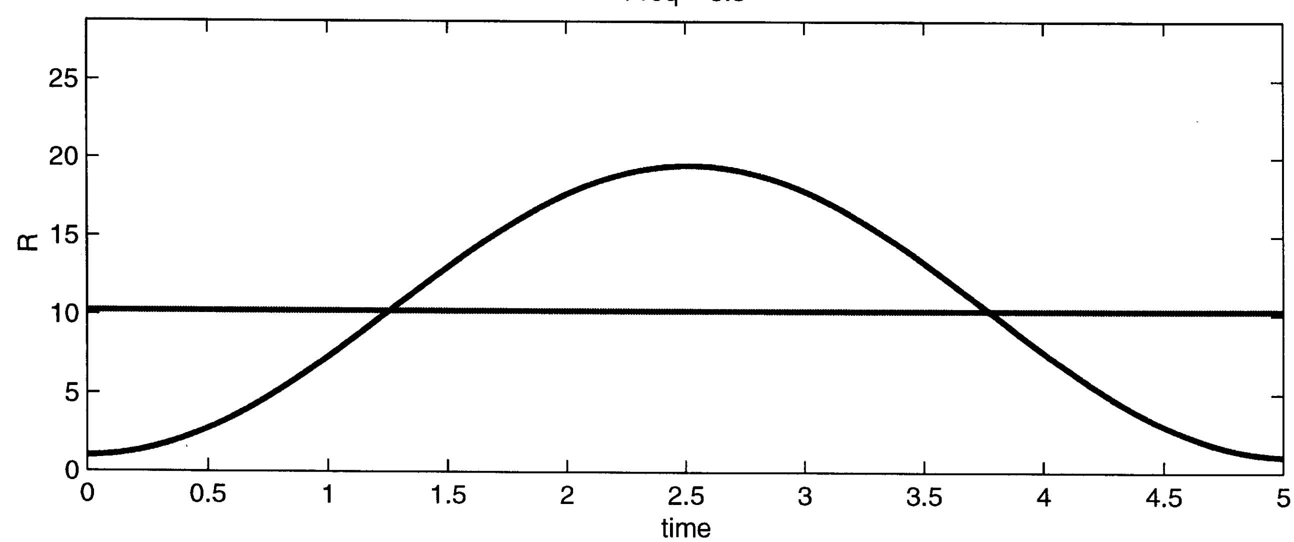


Freq = 2



AS $\omega \uparrow$, $\Omega \downarrow$

Freq = 3.8



- IF $F(x,\dot{x})$ IS NONLINEAR THEN WE HAVE TO WORK JUST A BIT HARDER
  - APPROXIMATE DIFFERENTIAL EQUATION, AS BEFORE, BUT WE LET MATLAB® DO IT.
    - ⇒ ALL WE HAVE TO SUPPLY IS A PROGRAM THAT CAN COMPUTE $\ddot{x}$ $(= F(x,\dot{x}))$ GIVEN THE CURRENT VALUES OF $x, \dot{x}$
- THEN CALL <u>ODE23</u>
- **EXAMPLE:** $\ddot{x} + 3x^3 = 0$ CUBIC SPRING

$$\dot{\begin{bmatrix} x \\ \dot{x} \end{bmatrix}} = \begin{bmatrix} \dot{x} \\ -3x^3 \end{bmatrix}$$

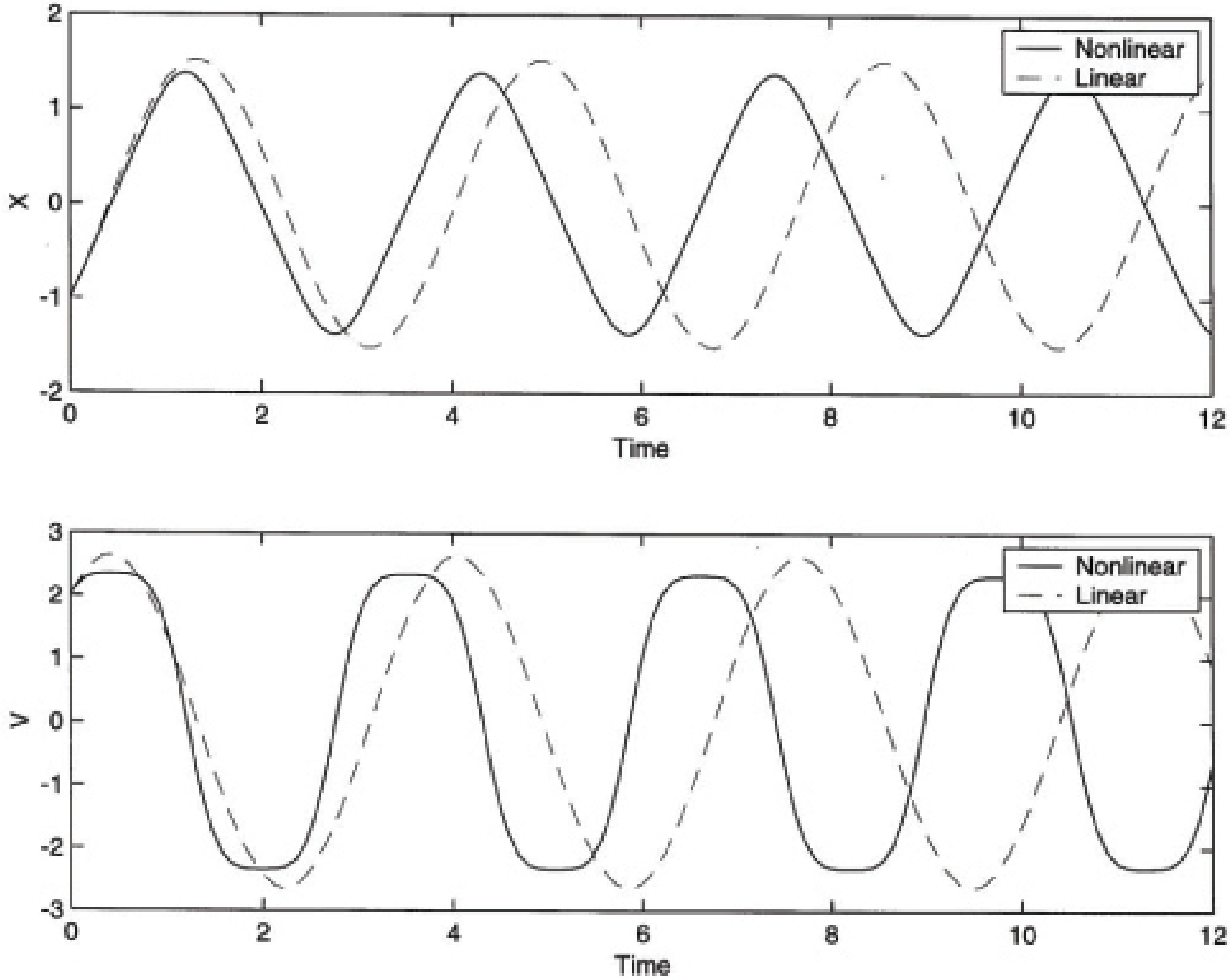
$$\dot{\underline{X}} = f(\underline{X})$$

$$\underline{X} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

```
% plant.m
function [xdot] = plant(t,x)
global n
xdot(1) = x(2);
xdot(2) = -3*(x(1))^n;
xdot = xdot';

return
```

THIS PART CALLS THE TOP PART.

```
% call_plant.m
global n
n=3;                NL
x0 = [-1 2];
[T,x]=ode23('plant', [0 12], x0);

n=1;                LINEAR
[T1,x1]=ode23('plant', [0 12], x0);

subplot(211)
plot(T,x(:,1),T1,x1(:,1),'--');
legend('Nonlinear','Linear')
ylabel('X')
xlabel('Time')
subplot(212)
plot(T,x(:,2),T1,x1(:,2),'--');
legend('Nonlinear','Linear')
ylabel('V')
xlabel('Time')
```



MATLAB® is a trademark of The MathWorks, Inc.