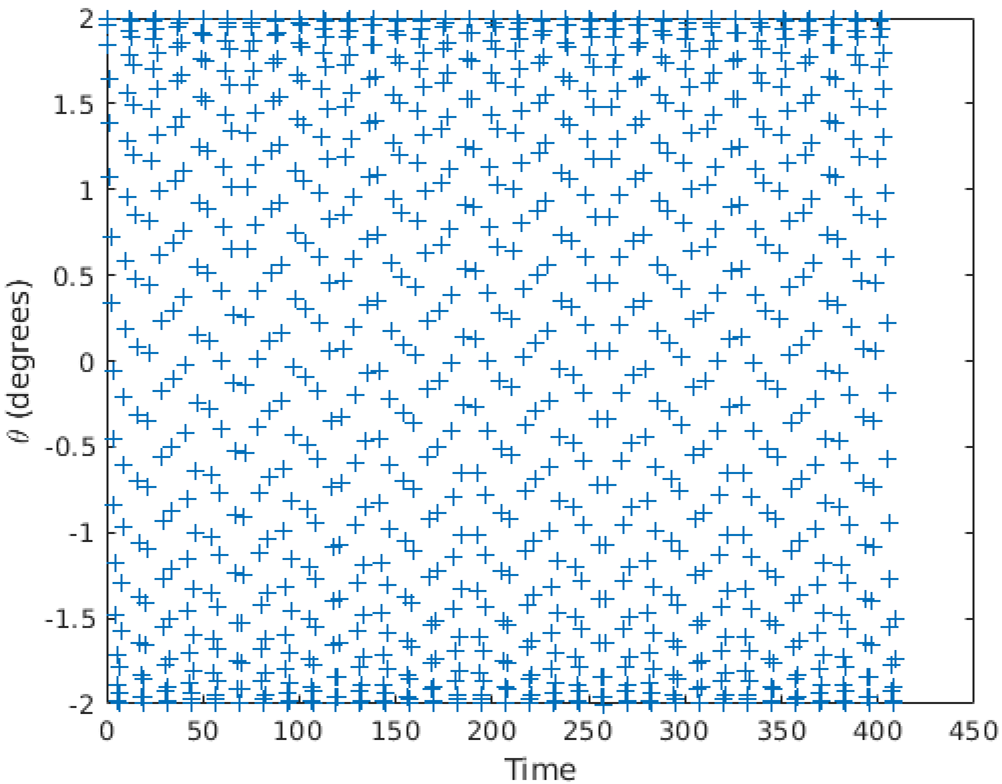
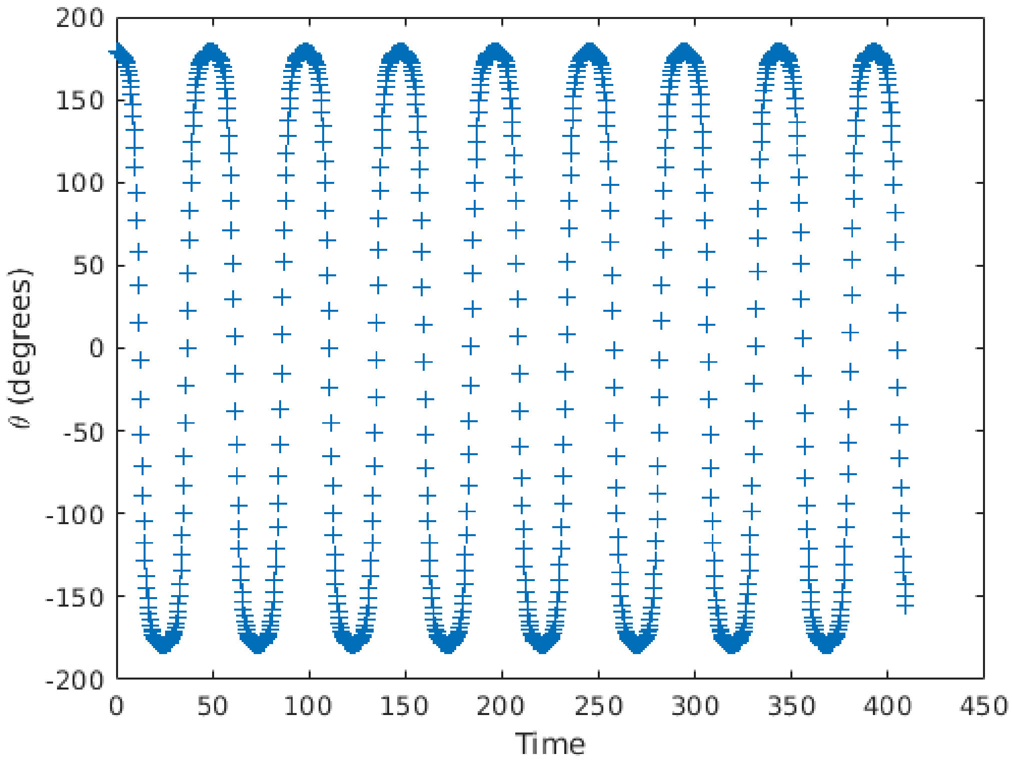
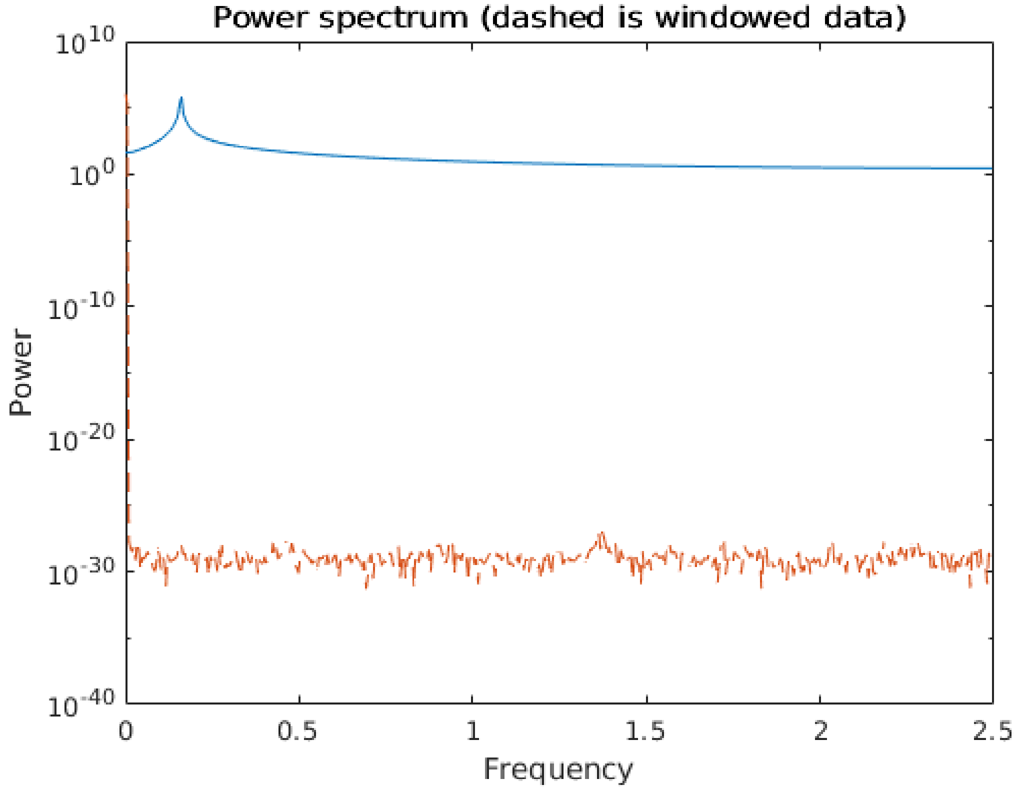
```
% pendul - Program to compute the motion of a simple pendulum
% using the Euler or Verlet method
clear all;

th0 = [2, 179];

for th_idx = 1:2

 theta0 = th0(th_idx)


theta0 =

     2



theta0 =

   179


 %% initial setup

 %* Select the numerical method to use: Euler or Verlet
 NumericalMethod = 2;

 %* Set initial position and velocity of pendulum
 theta = theta0*pi/180;   % Convert angle to radians
 omega = 0;               % Set the initial velocity

 %* Set the physical constants and other variables
 nstep = 1024;
 tau = 0.2;
 g_over_L = 1;            % The constant g/L
 time = (0:(nstep-1))*tau;               % Initial time
 irev = 0;                % Used to count number of reversals

 %% time series soln to motion of mass

 %* Take one backward step to start Verlet
 accel = -g_over_L*sin(theta);    % Gravitational acceleration
 theta_old = theta - omega*tau + 0.5*tau^2*accel;

 %* Loop over desired number of steps with given time step
 %    and numerical method
 for istep=1:nstep

   %* Record angle and time for plotting
   t_plot(istep) = time(istep);
   th_plot(istep) = theta*180/pi;   % Convert angle to degrees
   time = time + tau;
```

```
  %* Compute new position and velocity using
  %    Euler or Verlet method
  accel = -g_over_L*sin(theta);    % Gravitational acceleration
  if( NumericalMethod == 1 )
    theta_old = theta;              % Save previous angle
    theta = theta + tau*omega;      % Euler method
    omega = omega + tau*accel;
  else
    theta_new = 2*theta - theta_old + tau^2*accel;
    theta_old = theta;       % Verlet method
    theta = theta_new;
  end
end

%% plot of time series

%* Graph the oscillations as theta versus time
clf;  figure(gcf);         % Clear and forward figure window
plot(t_plot,th_plot,'+');
xlabel('Time');  ylabel('\theta (degrees)');
```

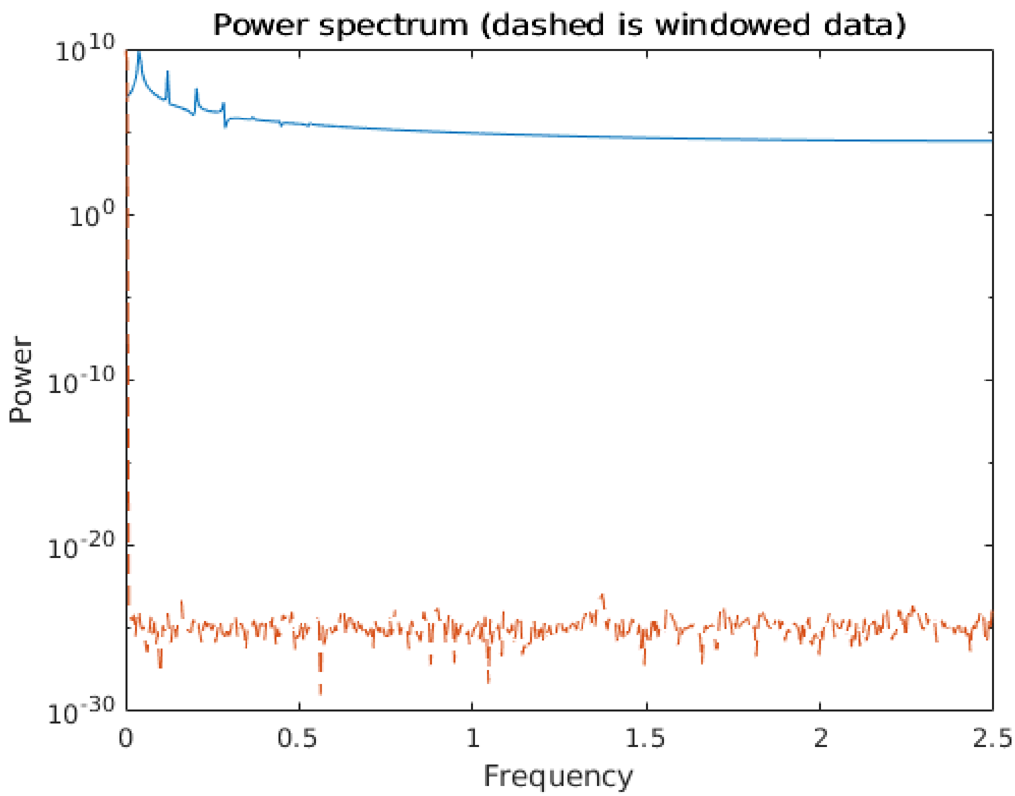





```
%% spectral analysis

%* Calculate the power spectrum of the time series for mass #1
f(1:nstep) = (0:(nstep-1))/(tau*nstep);      % Frequency
x1 = th_plot;                     % Displacement of mass 1
x1fft = fft(x1);                     % Fourier transform of displacement
spect = abs(x1fft).^2;               % Power spectrum of displacement

%* Apply the Hanning window to the time series and calculate
%  the resulting power spectrum
window = 0.5*(1-cos(2*pi*((1:nstep)-1)/nstep)); % Hanning window
x1w = x1 .* window';                 % Windowed time series
x1wfft = fft(x1w);                   % Fourier transf. (windowed data)
spectw = abs(x1wfft).^2;             % Power spectrum (windowed data)

%* Graph the power spectra for original and windowed data
figure(2); clf;  % Clear figure 2 window and bring forward
semilogy(f(1:(nstep/2)),spect(1:(nstep/2)),'-',...
  f(1:(nstep/2)),spectw(1:(nstep/2)),'--');
title('Power spectrum (dashed is windowed data)');
xlabel('Frequency'); ylabel('Power');
```



Power spectrum (dashed is windowed data)
Power
Frequency
10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
10^{-40}
0
0.5
1
1.5
2
2.5



Power spectrum (dashed is windowed data)
Power
Frequency
10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
0
0.5
1
1.5
2
2.5


```
end
```

Published with MATLAB® R2019a