```
clearvars
format long
```

Research Techniques Project

Milestone 1 (6 October)

Data Exploration of BD+55_441

For this milestone, BD+55_441 is displayed and explored in detail. The other sources are briefly imported in upcoming sections.

```
opts = detectImportOptions("BD+55_441.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
```

"time" is in the units of days and "flux" is "rel_flux_T1" from AstroImageJ outputs.

```
opts.VariableTypes = {'double','double','double','double','double','double'};
preview("BD+55_441.txt",opts)
```

ans = 8×6 table

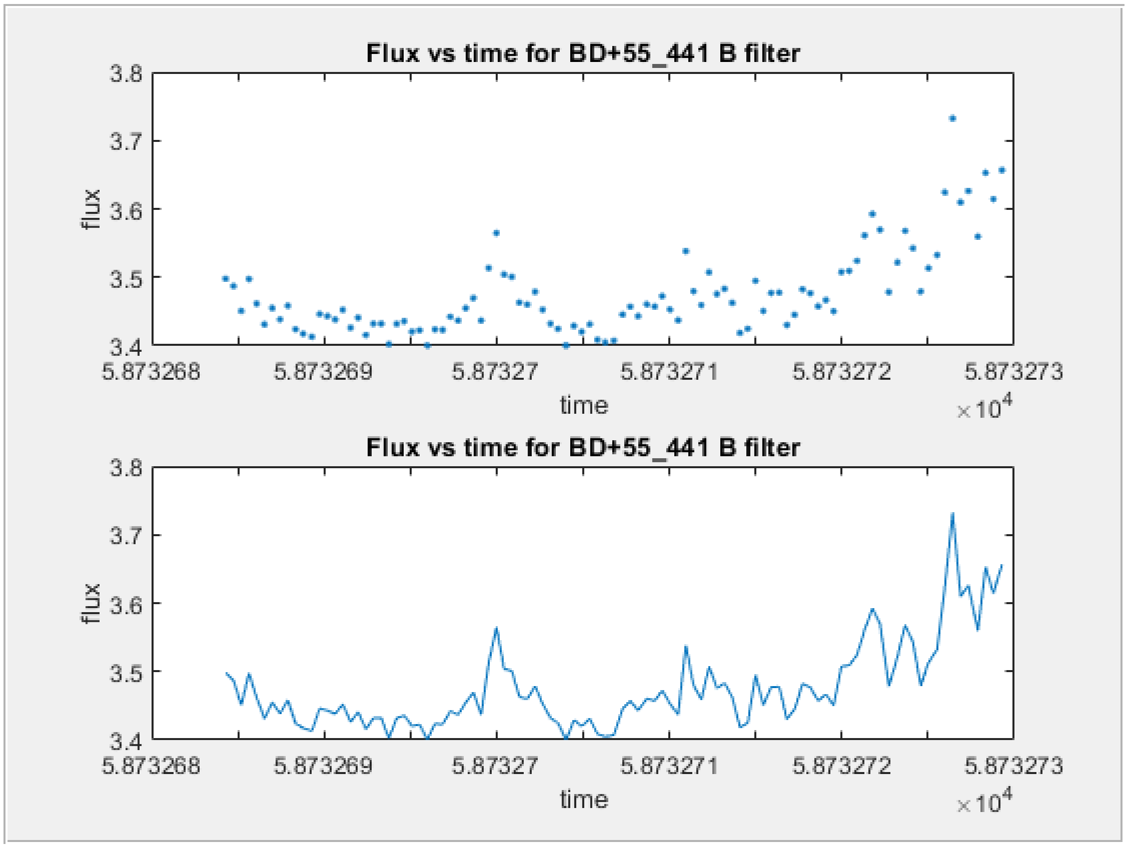
	B_time	B_flux	R_time	R_flux	V_time	V_flux
1	5.873268430...	3.497996...	5.873268459...	1.961830...	5.873268445...	2.352058...
2	5.873268430...	3.497996...	5.873268504...	1.942857...	5.873268490...	2.360458...
3	5.873268475...	3.487289...	5.873268549...	1.947021...	5.873268535...	2.329957...
4	5.873268520...	3.450942...	5.873268594...	1.930594...	5.873268580...	2.344889...
5	5.873268565...	3.497567...	5.873268639...	1.942537...	5.873268625...	2.361048...
6	5.873268610...	3.461555...	5.873268684...	1.935141...	5.873268669...	2.333341...
7	5.873268655...	3.431248...	5.873268729...	1.931290...	5.873268714...	2.331904...
8	5.873268700...	3.455091...	5.873268774...	1.928260...	5.873268759...	2.323782...

```
BD55_441 = readmatrix("BD+55_441.txt",opts);
whos BD55_441
```

```
  Name            Size            Bytes  Class     Attributes

  BD55_441      101x6              4848  double
```

```
hf_sub(1) = figure(1);
hp(1) = uipanel('Parent',hf_sub(1),'Position',[0 0 1 1]);
subplot(2,1,1,'Parent',hp(1));
plot(BD55_441(:,1),BD55_441(:,2),'.')
title('Flux vs time for BD+55\_441 B filter');
xlabel('time'),ylabel('flux');
subplot(2,1,2,'Parent',hp(1));
plot(BD55_441(:,1),BD55_441(:,2)),title('Flux vs time for BD+55\_441 B filter')
xlabel('time');
```

```
ylabel('flux');
```



Noting the graph above - displayed in discrete points and as a line graph - there is some periodicity. There are peaks at 58732.70, 58732.711 and 58732.722.

```
r = abs([0; BD55_441(:,1)] - [BD55_441(:,1);0]);
round(mean(r(3:end-1)),8) %average period between measurements
```

```
ans =
     4.543400000000000e-04
```

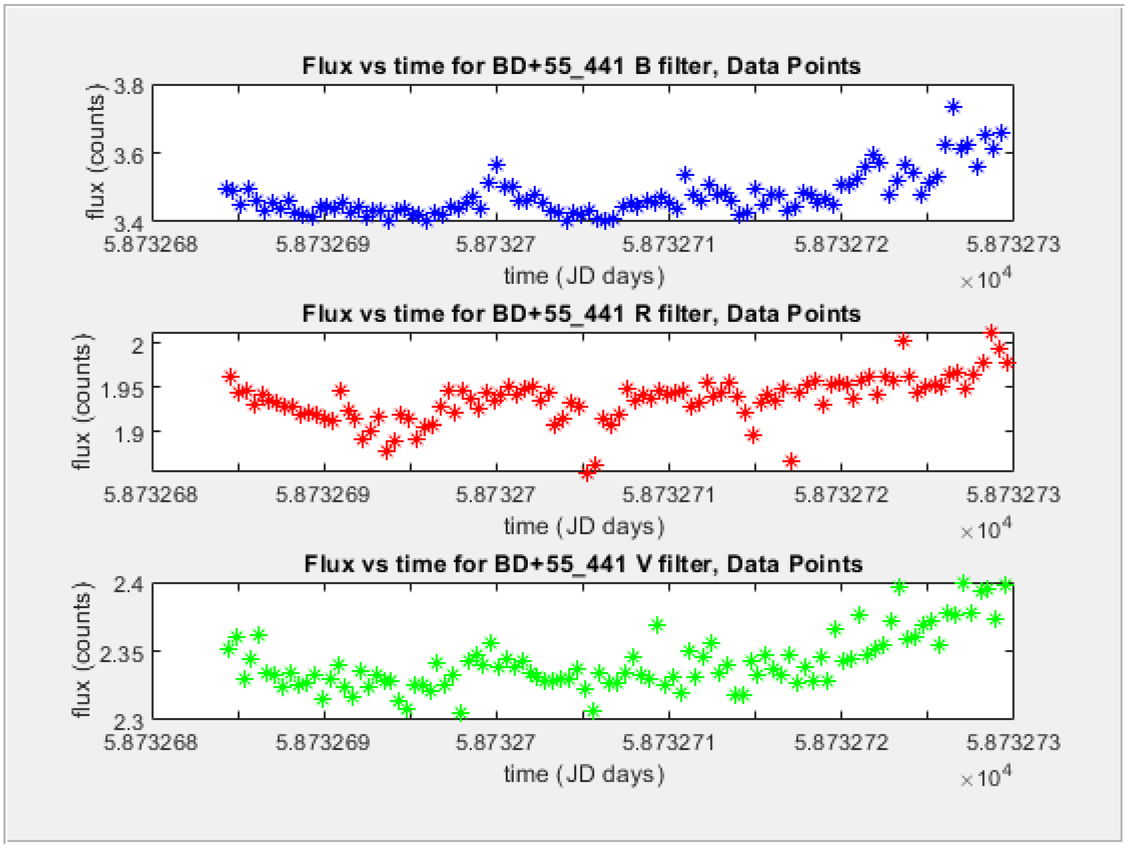
```
round(std(r(3:end-1)),8) %standard deviation - as a measurement of variation in period
```

```
ans =
     1.885000000000000e-05
```

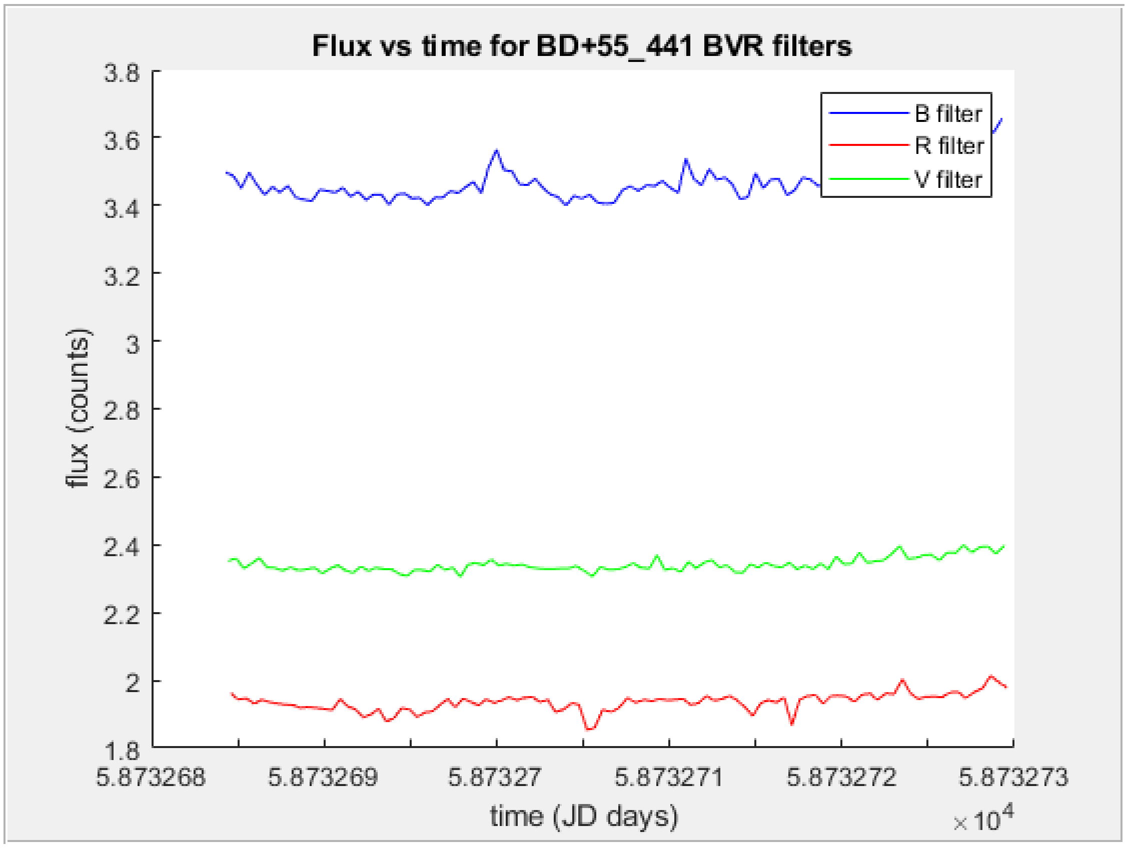
Periodicity between filters

```
hf_sub(2) = figure(2);
hp(2) = uipanel('Parent',hf_sub(2),'Position',[0 0 1 1]);
subplot(3,1,1,'Parent',hp(2));
% plotting in blue because B filter is blue light (~400-500 nm)
plot(BD55_441(:,1),BD55_441(:,2),'*b')
title('Flux vs time for BD+55\_441 B filter, Data Points');
xlabel('time (JD days)');
ylabel('flux (counts)');
```

```
subplot(3,1,2,'Parent',hp(2));
% plotting in red because R filter is red light (~550-800 nm)
plot(BD55_441(:,3),BD55_441(:,4),'*r');
title('Flux vs time for BD+55\_441 R filter, Data Points');
xlabel('time (JD days)');
ylabel('flux (counts)');
subplot(3,1,3,'Parent',hp(2));
% plotting green but V filter is visible light (~500-700 nm)
plot(BD55_441(:,5),BD55_441(:,6),'*g');
title('Flux vs time for BD+55\_441 V filter, Data Points');
xlabel('time (JD days)');
ylabel('flux (counts)');
```



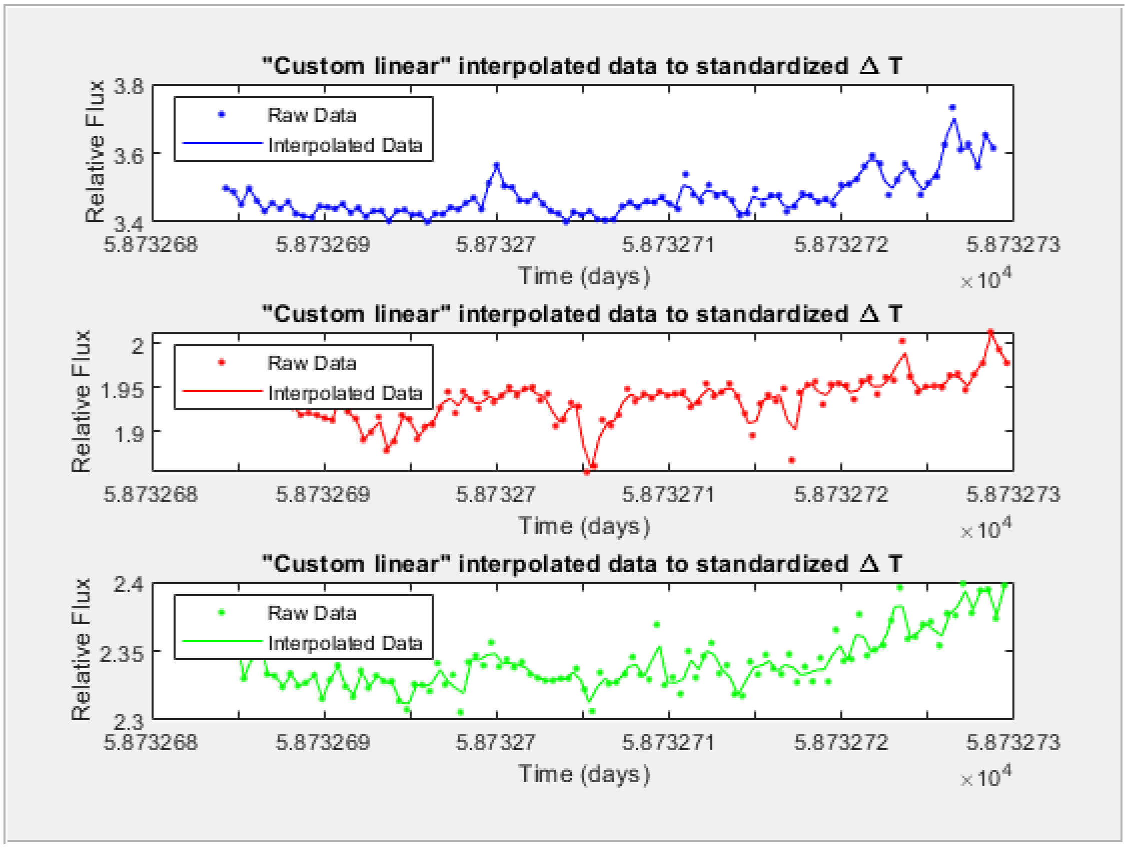

```
hf_sub(3) = figure(3);
hp(3) = uipanel('Parent',hf_sub(3),'Position',[0 0 1 1]);
subplot(1,1,1,'Parent',hp(3))
hold on
plot(BD55_441(:,1),BD55_441(:,2), 'b')
plot(BD55_441(:,3),BD55_441(:,4), 'r')
plot(BD55_441(:,5),BD55_441(:,6), 'g')
hold off
title('Flux vs time for BD+55\_441 BVR filters');
xlabel('time (JD days)');
legend('B filter', 'R filter', 'V filter');
ylabel('flux (counts)');
```



Interpolation investigation

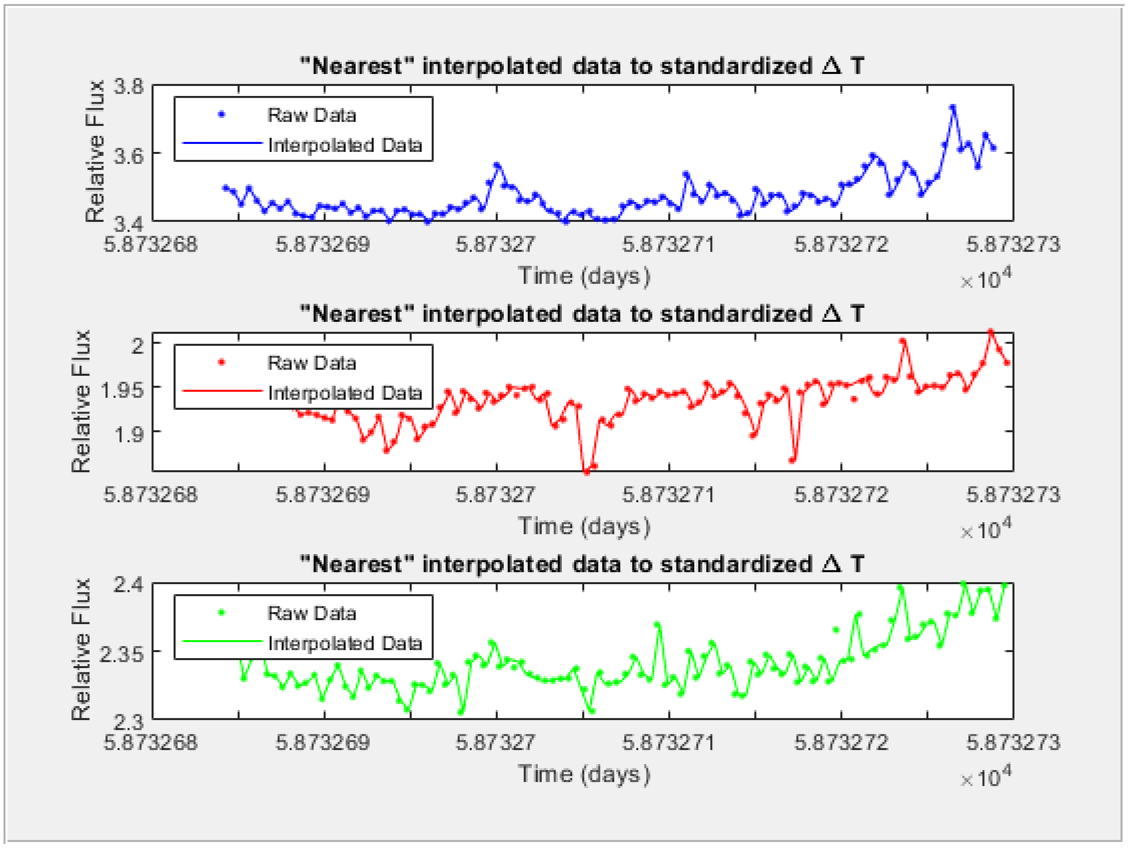
custom linear

```
hf_sub(4) = figure(4);
hp(4) = uipanel('Parent',hf_sub(4),'Position',[0 0 1 1]);
subplot(3,1,1,'Parent',hp(4));
[Tnew,Mnew] = Interp_Lin(BD55_441(1:100,1),BD55_441(1:100,2));
plot(BD55_441(1:100,1),BD55_441(1:100,2),'.b',Tnew,Mnew,'b');
title('"Custom linear" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
subplot(3,1,2,'Parent',hp(4));
[Tnew,Mnew] = Interp_Lin(BD55_441(1:100,3),BD55_441(1:100,4));
plot(BD55_441(1:100,3),BD55_441(1:100,4),'.r',Tnew,Mnew,'r');
title('"Custom linear" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
subplot(3,1,3,'Parent',hp(4));
[Tnew,Mnew] = Interp_Lin(BD55_441(1:100,5),BD55_441(1:100,6));
plot(BD55_441(1:100,5),BD55_441(1:100,6),'.g',Tnew,Mnew,'g');
title('"Custom linear" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
```



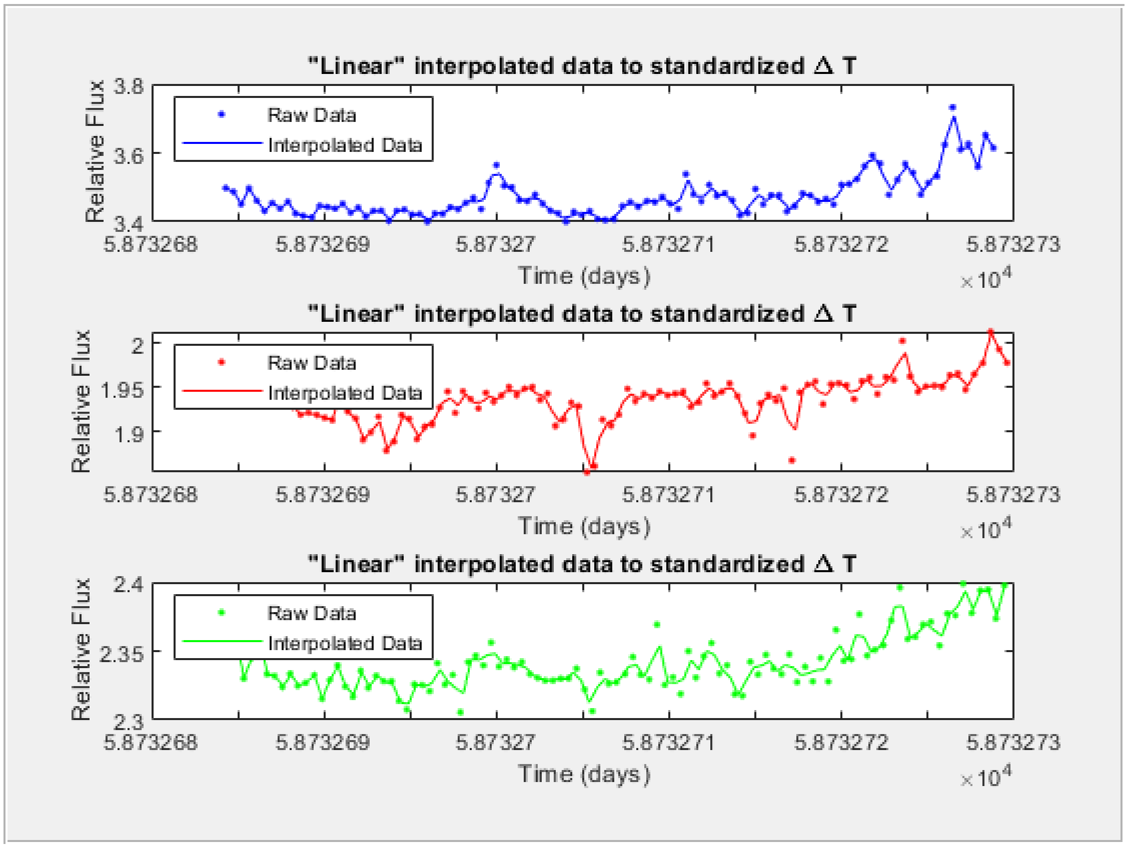
Nearest

```
hf_sub(5) = figure(5);
hp(5) = uipanel('Parent',hf_sub(5),'Position',[0 0 1 1]);
subplot(3,1,1,'Parent',hp(5));
[Tnew,Mnew] = Interp_nearest(BD55_441(2:100,1),BD55_441(2:100,2));
plot(BD55_441(1:100,1),BD55_441(1:100,2),'.b',Tnew,Mnew,'b');
title('"Nearest" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
subplot(3,1,2,'Parent',hp(5));
[Tnew,Mnew] = Interp_nearest(BD55_441(1:100,3),BD55_441(1:100,4));
plot(BD55_441(1:100,3),BD55_441(1:100,4),'.r',Tnew,Mnew,'r');
title('"Nearest" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
subplot(3,1,3,'Parent',hp(5));
[Tnew,Mnew] = Interp_nearest(BD55_441(1:100,5),BD55_441(1:100,6));
plot(BD55_441(1:100,5),BD55_441(1:100,6),'.g',Tnew,Mnew,'g');
title('"Nearest" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
```



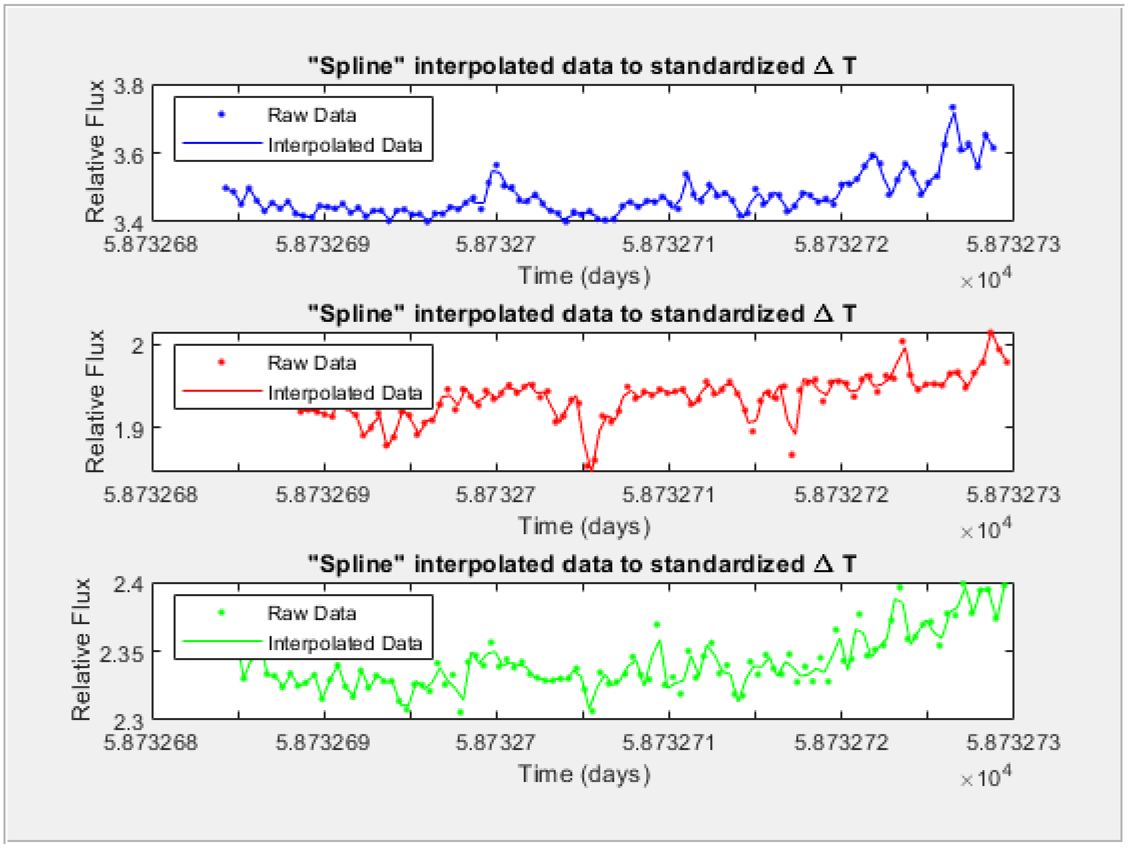
linear

```
hf_sub(6) = figure(6);
hp(6) = uipanel('Parent',hf_sub(6),'Position',[0 0 1 1]);
subplot(3,1,1,'Parent',hp(6));
[Tnew,Mnew] = Interp_linear(BD55_441(2:100,1),BD55_441(2:100,2));
plot(BD55_441(1:100,1),BD55_441(1:100,2),'.b',Tnew,Mnew,'b');
title('"Linear" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
subplot(3,1,2,'Parent',hp(6));
[Tnew,Mnew] = Interp_linear(BD55_441(1:100,3),BD55_441(1:100,4));
plot(BD55_441(1:100,3),BD55_441(1:100,4),'.r',Tnew,Mnew,'r');
title('"Linear" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
subplot(3,1,3,'Parent',hp(6));
[Tnew,Mnew] = Interp_linear(BD55_441(1:100,5),BD55_441(1:100,6));
plot(BD55_441(1:100,5),BD55_441(1:100,6),'.g',Tnew,Mnew,'g');
title('"Linear" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
```



spline

```
hf_sub(7) = figure(7);
hp(7) = uipanel('Parent',hf_sub(7),'Position',[0 0 1 1]);
subplot(3,1,1,'Parent',hp(7));
[Tnew,Mnew] = Interp_spline(BD55_441(2:100,1),BD55_441(2:100,2));
plot(BD55_441(1:100,1),BD55_441(1:100,2),'.b',Tnew,Mnew,'b');
title('"Spline" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
subplot(3,1,2,'Parent',hp(7));
[Tnew,Mnew] = Interp_spline(BD55_441(1:100,3),BD55_441(1:100,4));
plot(BD55_441(1:100,3),BD55_441(1:100,4),'.r',Tnew,Mnew,'r');
title('"Spline" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
subplot(3,1,3,'Parent',hp(7));
[Tnew,Mnew] = Interp_spline(BD55_441(1:100,5),BD55_441(1:100,6));
plot(BD55_441(1:100,5),BD55_441(1:100,6),'.g',Tnew,Mnew,'g');
title('"Spline" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
```



polyfit - needs to be conditioned appropriately.

```
hf_sub(8) = figure(8);
hp(8) = uipanel('Parent',hf_sub(8),'Position',[0 0 1 1]);
subplot(3,1,1,'Parent',hp(8));
[Tnew,Mnew] = Interp_polyfit(BD55_441(2:100,1),BD55_441(2:100,2));
```

```
Warning: Polynomial is badly conditioned. Add points with distinct X values, reduce the degree of the
polynomial, or try centering and scaling as described in HELP POLYFIT.
```

```
plot(BD55_441(1:100,1),BD55_441(1:100,2),'.b',Tnew,Mnew,'b');
title('"Polyfit" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
subplot(3,1,2,'Parent',hp(8));
[Tnew,Mnew] = Interp_polyfit(BD55_441(1:100,3),BD55_441(1:100,4));
```

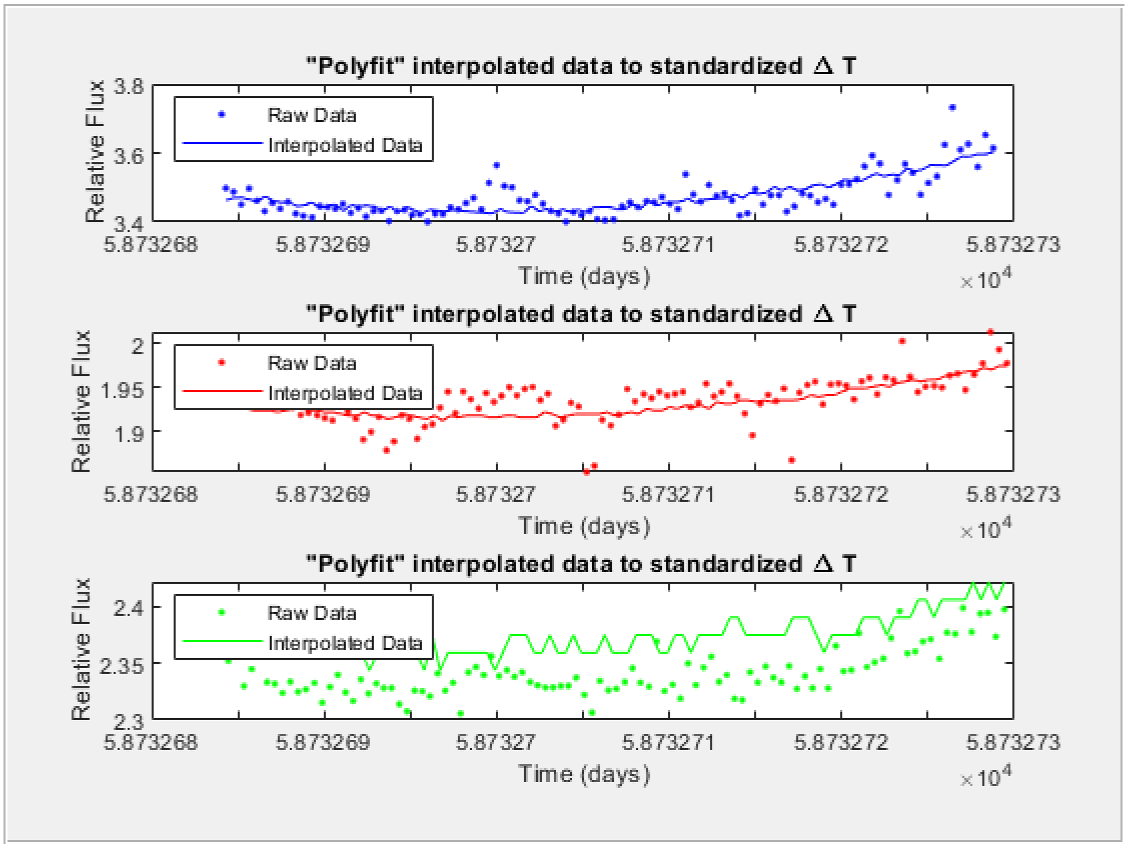
```
Warning: Polynomial is badly conditioned. Add points with distinct X values, reduce the degree of the
polynomial, or try centering and scaling as described in HELP POLYFIT.
```

```
plot(BD55_441(1:100,3),BD55_441(1:100,4),'.r',Tnew,Mnew,'r');
title('"Polyfit" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
```

```
ylabel('Relative Flux');
subplot(3,1,3,'Parent',hp(8));
[Tnew,Mnew] = Interp_polyfit(BD55_441(1:100,5),BD55_441(1:100,6));
```

```
Warning: Polynomial is badly conditioned. Add points with distinct X values, reduce the degree of the
polynomial, or try centering and scaling as described in HELP POLYFIT.
```

```
plot(BD55_441(1:100,5),BD55_441(1:100,6),'.g',Tnew,Mnew,'g');
title('"Polyfit" interpolated data to standardized \Delta T');
legend('Raw Data','Interpolated Data','location','northwest');
xlabel('Time (days)');
ylabel('Relative Flux');
```



Input all data

Text files:

```
%BD+48_1098
opts = detectImportOptions("BD+48_1098.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
BD48_1098 = rmmissing(readmatrix("BD+55_441.txt",opts)); %the matrix data of the text file
```

```
%preview("BD+48_1098.txt",opts)
whos BD48_1098
```

```
  Name            Size            Bytes  Class     Attributes

  BD48_1098      100x6             4800  double
```

```
%HD28497
opts = detectImportOptions("HD28497.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD28497 = rmmissing(readmatrix("HD28497.txt",opts)) %the matrix data of the text file
```

```
HD28497 = 600×6
10^4 ×
   5.844163256000000   0.151884039400000   5.844163279000000   0.047485834000000 ···
   5.844163294000000   0.157713255000000   5.844163317000000   0.047850097100000
   5.844163332000000   0.147648080800000   5.844163355000000   0.050717498800000
   5.844163370000000   0.154105058200000   5.844163393000001   0.048745268800000
   5.844163409000000   0.154416998300000   5.844163431999999   0.046480291500000
   5.844163447000000   0.154034530100000   5.844163470000000   0.052878906200000
   5.844163485000000   0.158524561400000   5.844163508000000   0.050080829500000
   5.844163523000000   0.159731365000000   5.844163546000000   0.053844298900000
   5.844163560999999   0.140623476800000   5.844163584000000   0.050871022200000
   5.844163599000000   0.159835953000000   5.844163622000000   0.053311599500000
      ⋮
```

```
%preview("HD28497.txt",opts)
whos HD28497
```

```
  Name           Size            Bytes  Class     Attributes

  HD28497       600x6            28800  double
```

```
%HD46131
opts = detectImportOptions("HD46131.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD46131 = rmmissing(readmatrix("HD46131.txt",opts)); %the matrix data of the text file
%preview("HD46131.txt",opts)
whos HD46131
```

```
  Name           Size            Bytes  Class     Attributes

  HD46131       250x6            12000  double
```

```
%HD88661
opts = detectImportOptions("HD88661.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD88661 = rmmissing(readmatrix("HD88661.txt",opts)); %the matrix data of the text file
%preview("HD88661.txt",opts)
whos HD88661
```

```
  Name         Size            Bytes  Class     Attributes

  HD88661      35x6             1680  double
```

```
%HD105521
opts = detectImportOptions("HD105521.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD105521 = rmmissing(readmatrix("HD105521.txt",opts)); %the matrix data of the text file
%preview("HD105521.txt",opts)
whos HD105521
```

```
  Name          Size            Bytes  Class     Attributes

  HD105521      180x6            8640  double
```

```
%HD105521
opts = detectImportOptions("HD105521.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD105521 = rmmissing(readmatrix("HD105521.txt",opts)); %the matrix data of the text file
%preview("HD105521.txt",opts)
whos HD105521
```

```
  Name          Size            Bytes  Class     Attributes

  HD105521      180x6            8640  double
```

CSV files

```
%HD106306
%B_filter
opts = detectImportOptions('HD106306_B.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD106306_B.csv",opts)
HD106306_B = readmatrix("HD106306_B.csv",opts);

%R_filter
opts = detectImportOptions('HD106306_R.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD106306_R.csv",opts)
HD106306_R = readmatrix("HD106306_R.csv",opts);

%V_filter
opts = detectImportOptions('HD106306_V.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD106306_V.csv",opts)
HD106306_V = readmatrix("HD106306_V.csv",opts);
```

```
HD106306 = rmmissing([HD106306_B HD106306_R HD106306_V]); %the matrix data of the text file
whos HD106306
```

```
  Name            Size             Bytes  Class     Attributes

  HD106306      100x6               4800  double
```

```
%HD147302
%B_filter
opts = detectImportOptions('HD147302_B.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD147302_B.csv",opts)
HD147302_B = readmatrix("HD147302_B.csv",opts);

%R_filter
opts = detectImportOptions('HD147302_R.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD147302_R.csv",opts)
HD147302_R = readmatrix("HD147302_R.csv",opts);

%V_filter
opts = detectImportOptions('HD147302_V.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD147302_V.csv",opts)
HD147302_V = readmatrix("HD147302_V.csv",opts);

HD147302 = rmmissing([HD147302_B HD147302_R HD147302_V]); %the matrix data of the text file
whos HD147302
```

```
  Name            Size             Bytes  Class     Attributes

  HD147302      100x6               4800  double
```

HD152060 and HD209522 can't be described in the same form as the others.

```
%HD152060
%R_filter
opts = detectImportOptions('HD152060_R.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD147302_R.csv",opts)
HD152060_R = readmatrix("HD152060_R.csv",opts);

%V_filter
opts = detectImportOptions('HD152060_V.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
```

```
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD147302_V.csv",opts)
HD152060_V = readmatrix("HD152060_V.csv",opts);

HD152060 = rmmissing([HD147302_R HD147302_V]); %the matrix data of the text file
whos HD152060
```

```
  Name            Size              Bytes  Class     Attributes

  HD152060      100x4                3200  double
```

```
%HD209522
%B_filter
opts = detectImportOptions('HD209522_B.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD147302_R.csv",opts)
HD209522 = readmatrix("HD209522_B.csv",opts);

%the matrix data of the text file
whos HD209522
```

```
  Name            Size              Bytes  Class     Attributes

  HD209522      408x2                6528  double
```

```
close all   %closes all previous figures from milestone 1 so that the plots in milestone 2 and
```

Milestone 2 (13 October)

https://www.mathworks.com/help/matlab/ref/fft.html?searchHighlight=fft&s_tid=srchtitle

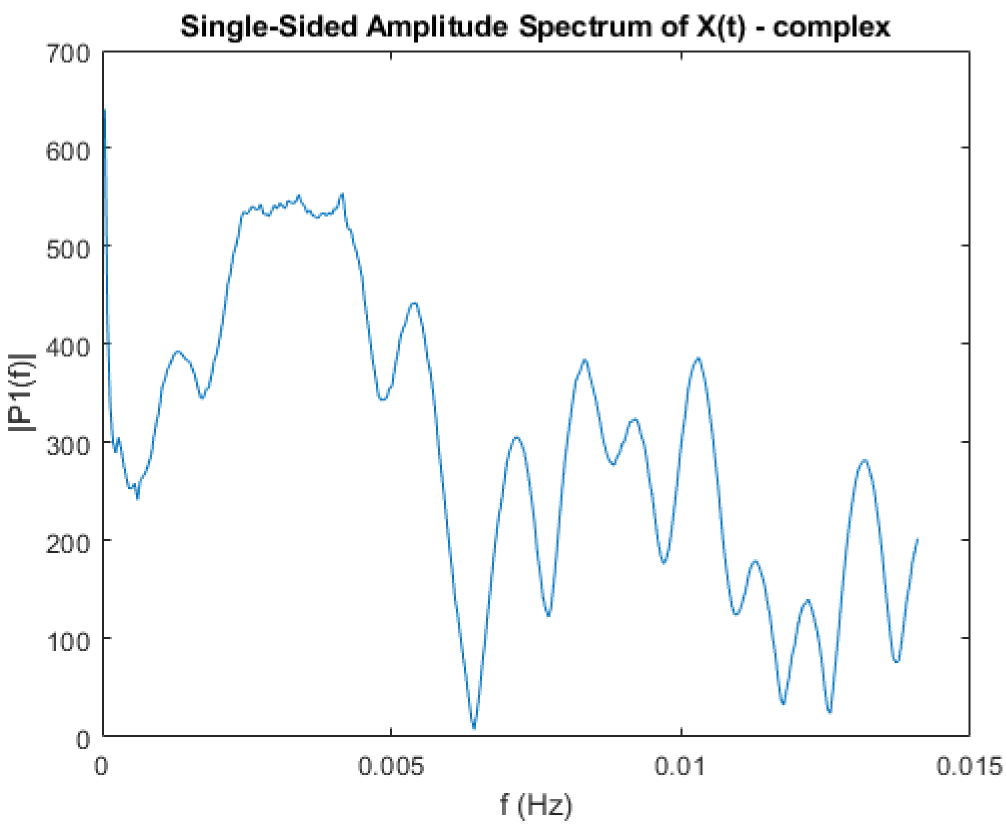
This is the source for implementing Fourier Transform for a Noisy Signal from the MathWorks documentation

Using fft for HD28497

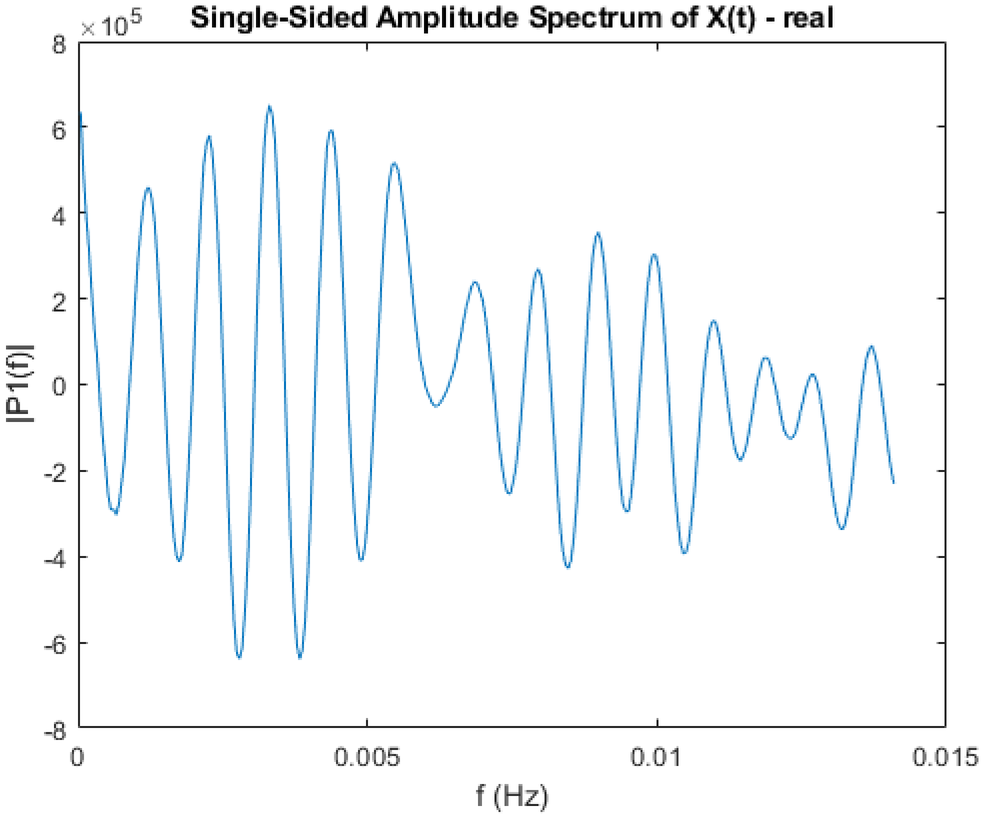
This star had the most data which made it ideal for performing a fourier transform. We chose linear interpolation for the purposes of this experiment at random. This is a feature to be explored later.

```
[Tnew,Mnew] = Interp_linear(HD28497(2:end,1),HD28497(2:end,2));
Tnew = Tnew*86400; %convert time from days to seconds
deltaT = Tnew(2)-Tnew(1);
Fs = 1/deltaT;            % Sampling frequency
T = deltaT;              % Sampling period
t = Tnew;                 % Time vector
L = numel(Tnew);          % Length of signal
X = Mnew;                %signal with noise
%Fourier transformation
Y = fft(X);
P2complex = abs(Y/L); %real and imaginary parts together
%Bk
```

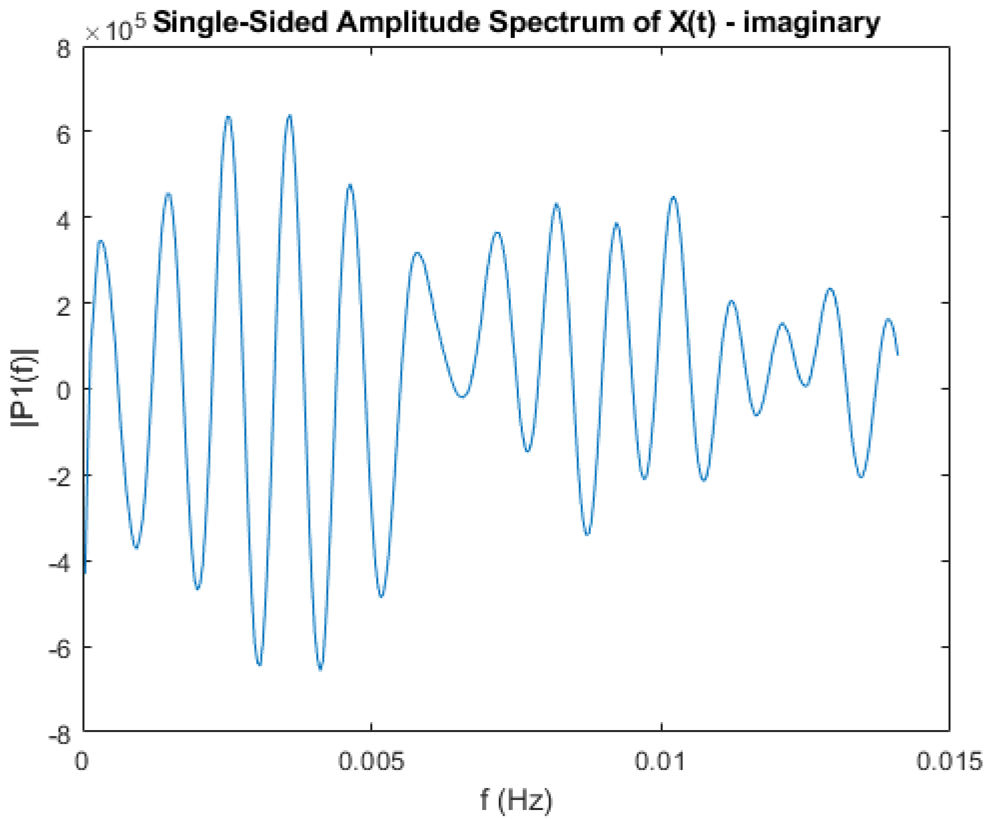
```
P2real = 2*real(Y);
P2imaginary = 2*imag(Y);
P1complex = P2complex(1:floor(L/2)+1);
P1complex(2:end-1) = 2*P1complex(2:end-1);

%Plot

f = Fs*(0:(L/2))/L;
plot(f(2:end-1),P1complex(2:end-1))
title('Single-Sided Amplitude Spectrum of X(t) - complex')
xlabel('f (Hz)')
ylabel('|P1(f)|')
```




```
P1real = P2real(1:floor(L/2)+1);
P1real(2:end-1) = 2*P1real(2:end-1);
%plotting piece
f = Fs*(0:(L/2))/L;
plot(f(2:end-1),P1real(2:end-1))
title('Single-Sided Amplitude Spectrum of X(t) - real')
xlabel('f (Hz)')
ylabel('|P1(f)|')
```




```
P1imaginary = P2imaginary(1:floor(L/2)+1);
P1imaginary(2:end-1) = 2*P1imaginary(2:end-1);
%plotting piece
f = Fs*(0:(L/2))/L;
plot(f(2:end-1),P1imaginary(2:end-1))
title('Single-Sided Amplitude Spectrum of X(t) - imaginary')
xlabel('f (Hz)')
ylabel('|P1(f)|')
```

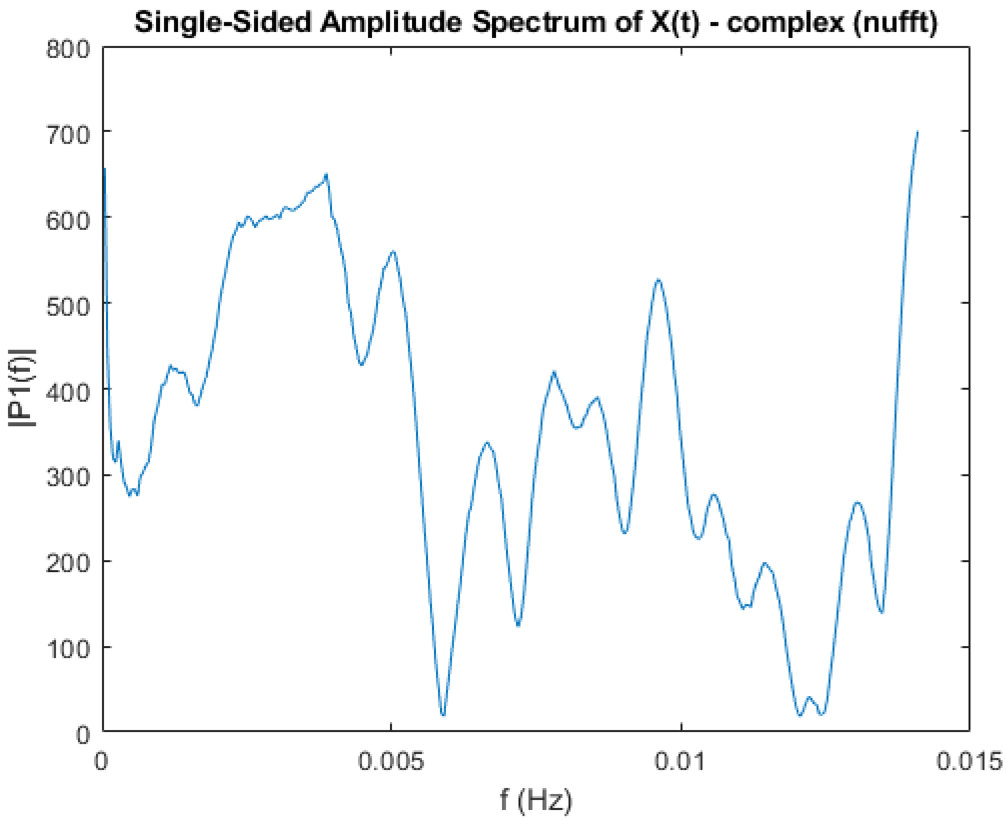


Using NUFFT - used the data without interpolation

Results for this one star using the complex combination was very similar to fft.

```
Tnew = HD28497(2:end,1);
Mnew = HD28497(2:end,2);
Tnew = Tnew*86400;
% --Find averaged time scale--
n = numel(Tnew);
sum = 0;
for l = 1:n-1
    sum = sum + abs(Tnew(l+1) - Tnew(l));
end
deltaT = sum / (n-1); %average time between points
Fs = 1/deltaT;            % Sampling frequency
T = deltaT;             % Sampling period
t = Tnew;                 % Time vector
L = numel(Tnew);          % Length of signal
X = Mnew;               %signal with noise
%Fourier transformation
Y = nufft(X);
P2complex = abs(Y/L); %combination of real and imaginary
P2real = 2*real(Y);
P2imaginary = 2*imag(Y);
P1complex = P2complex(1:floor(L/2)+1);
P1complex(2:end-1) = 2*P1complex(2:end-1);
%plotting piece
f = Fs*(0:(L/2))/L;
```

```
plot(f(2:end-1),P1complex(2:end-1))
title('Single-Sided Amplitude Spectrum of X(t) - complex (nufft)')
xlabel('f (Hz)')
ylabel('|P1(f)|')
```



The investigation of the inverse fourier transform will be explored at a later date to complete milestone 2.

Milestone 3 (including previously unfinished parts of milestone 2)

Getting the Power Spectrum for HD28497

This star had the most data which made it ideal for performing a fourier transform. We chose linear interpolation for the purposes of this experiment at random. This is a feature to be explored late

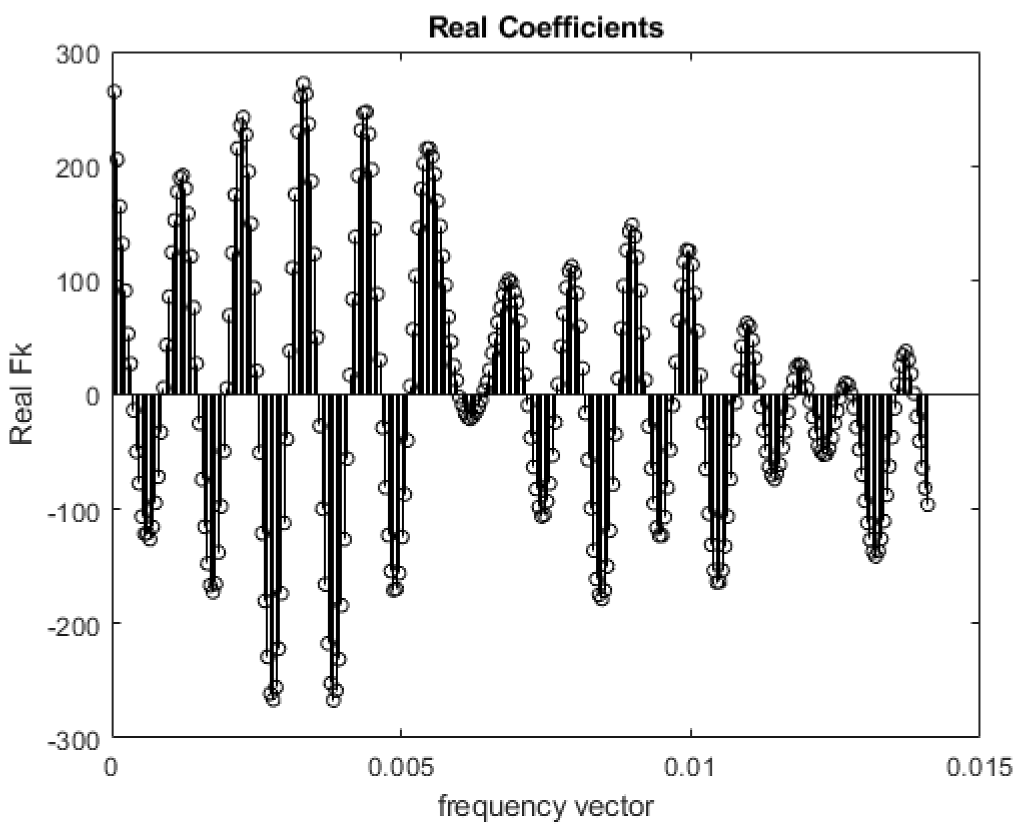
Documentation: We used a combination of the following sources to develop this:

1. https://www.mathworks.com/help/matlab/ref/fft.html?searchHighlight=fft&s_tid=srchtitle (This is the source for implementing Fourier Transform for a Noisy Signal from the MathWorks documentation)
2. Gilat, Example 7-6 (pg. 281)
3. Gilat, Example 7-8 (pg 287)

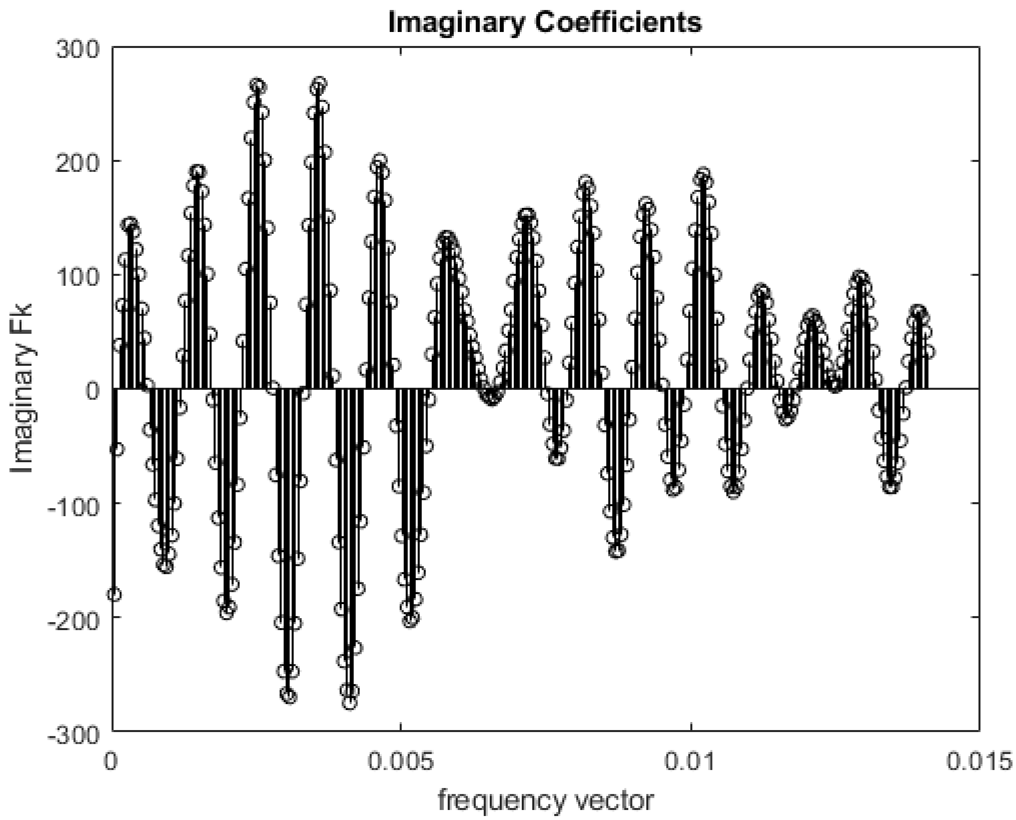
Bd+55_441 and HD 28497 have documented periodicity - citation pending

```
[Tnew,Mnew] = Interp_linear(HD28497(2:end,1),HD28497(2:end,2));
Tnew = Tnew*86400; %convert time from days to seconds
deltaT = Tnew(2)-Tnew(1);
```

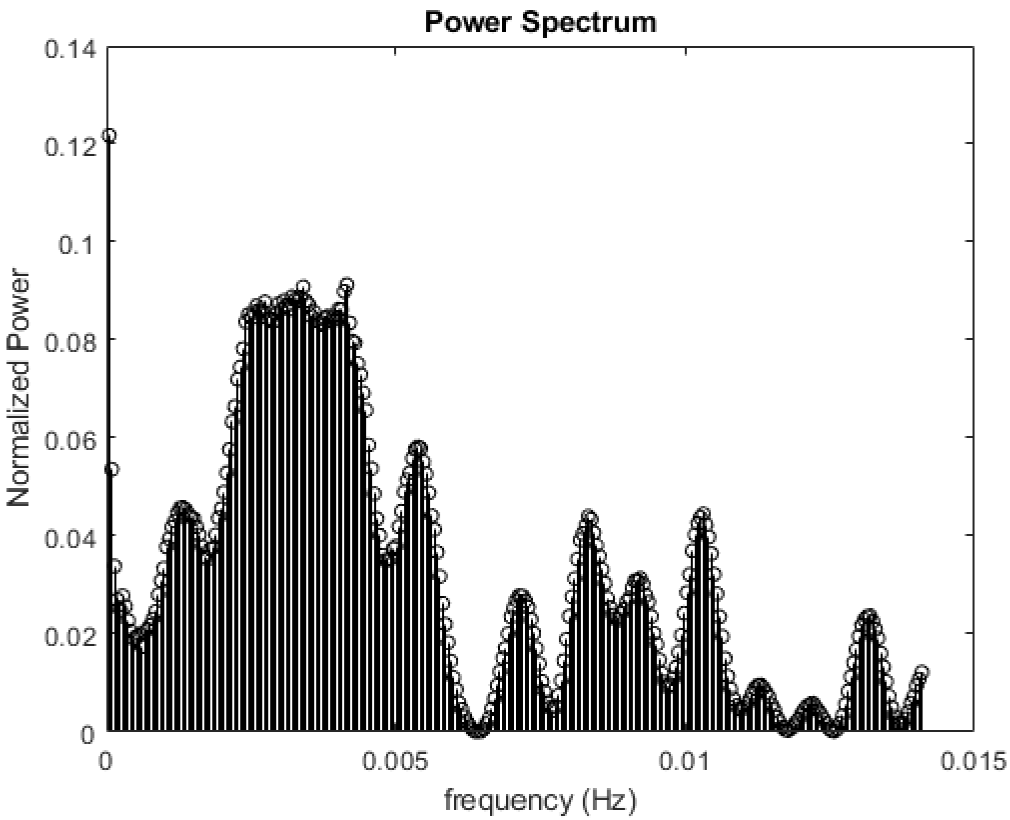
```
Fs = 1/deltaT;           % Sampling frequency
T = deltaT;             % Sampling period
t = Tnew-Tnew(1);                % Time vector
L = numel(Tnew);         % Length of signal
X = Mnew ;              %signal with noise



%Fourier transformation    %Based on Ex 7-6 Gilat
F = fft(X)/L  ;            %fourier transform devided by number of data points
power = F.*conj(F)/L;
powernorm = power/max(power);   %normalized power spectrum
fk = (0:L-1)*(Fs/L);      %frequency vector Gilat 7-6
stem(fk(2:floor(L/2)),real(F(2:floor(L/2))), 'ko','markersize',5)             %Threw out first
xlabel('frequency vector')
ylabel('Real Fk')
title('Real Coefficients')
```



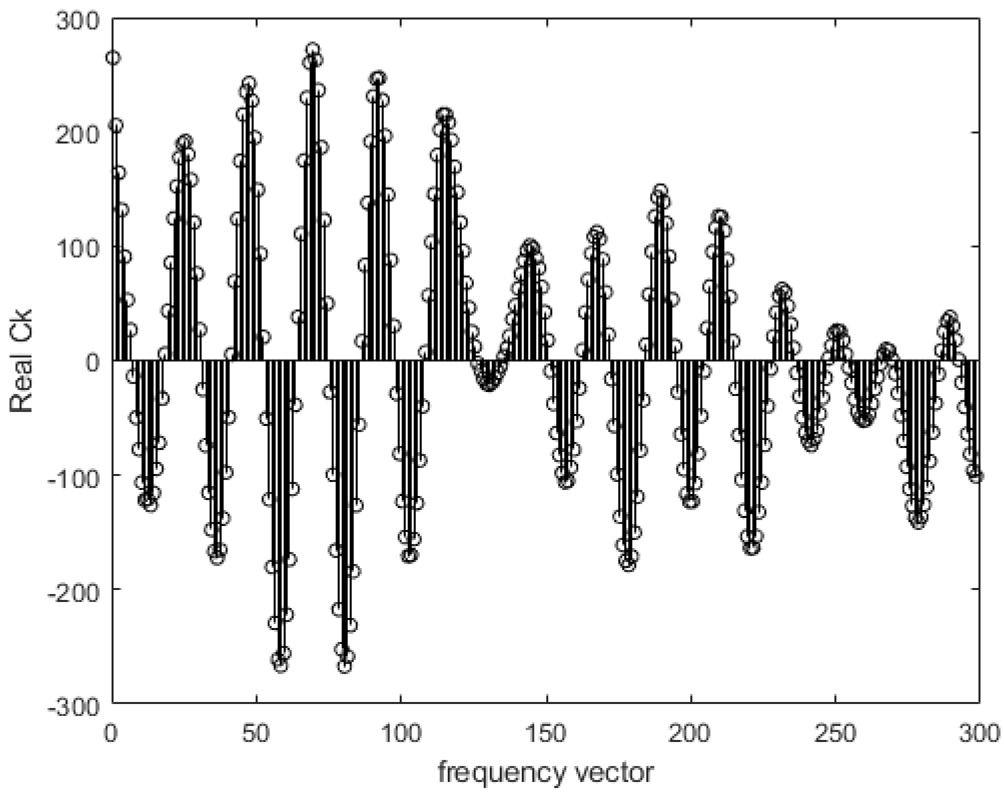

```
figure, stem(fk(2:floor(L/2)),imag(F(2:floor(L/2))),'ko','markersize',5)
xlabel('frequency vector')
ylabel('Imaginary Fk')
title('Imaginary Coefficients')
```



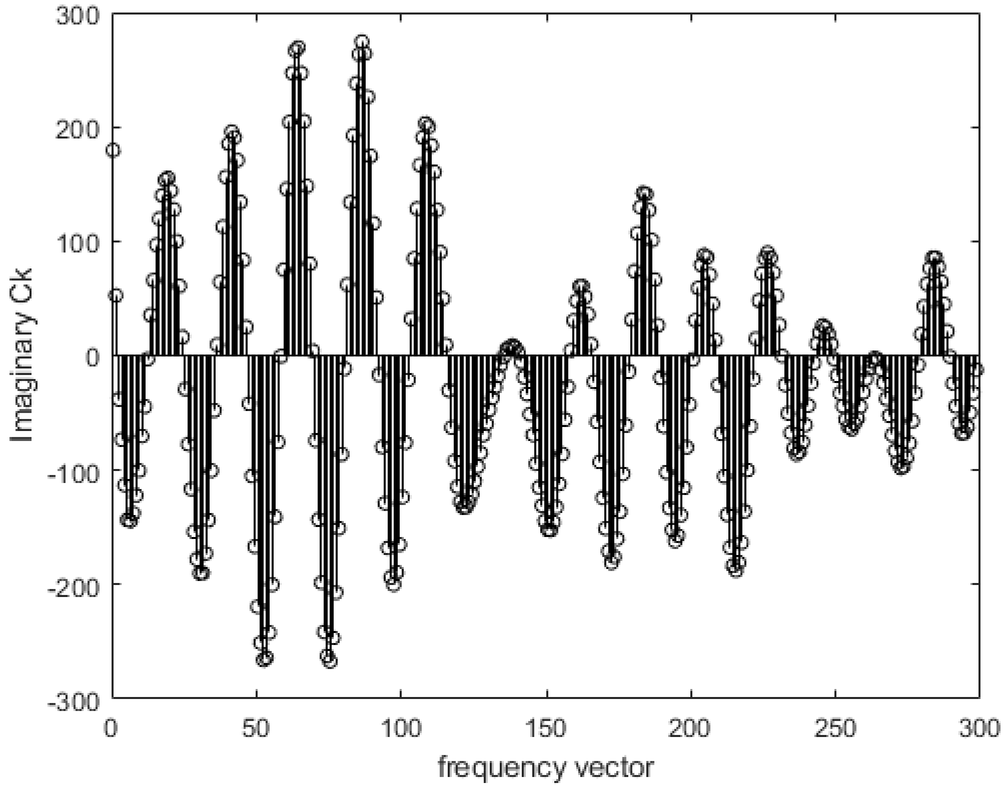

```
figure, stem(fk(2:floor(L/2)),powernorm(2:floor(L/2)),'ko','markersize',5)
xlabel('frequency (Hz)')
ylabel('Normalized Power')
title('Power Spectrum')
```



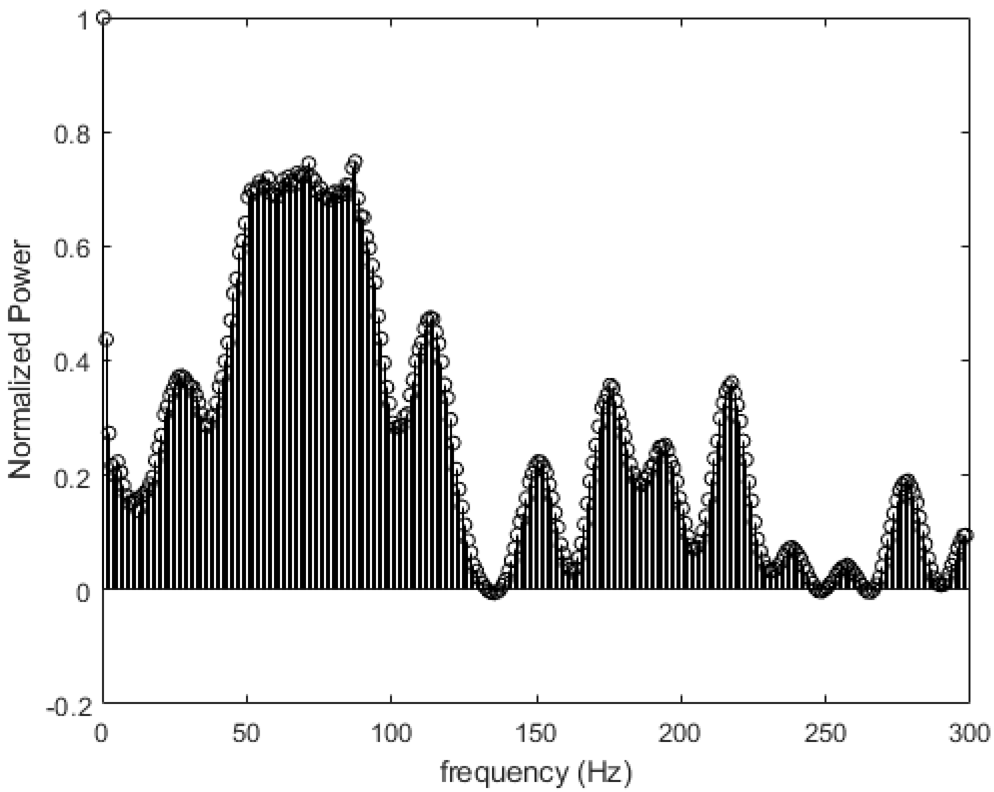

```
%Rearrange the spectrum using fftshift
C0 = fftshift(F(2:L));              %zero frequency in center (threw out first number of F as a
power0 = C0.*conj(C0)/L-1;
powernorm0 = power0/max(power0);   %normalized power spectrum
fk0 = (-L/2+1:L/2-1)*(Fs/L-1);      %frequency vector Gilat 7-6
stem(fk0(1:floor(L/2)),real(C0(1:floor(L/2))), 'ko','markersize',5)
xlabel('frequency vector')
ylabel('Real Ck')
```



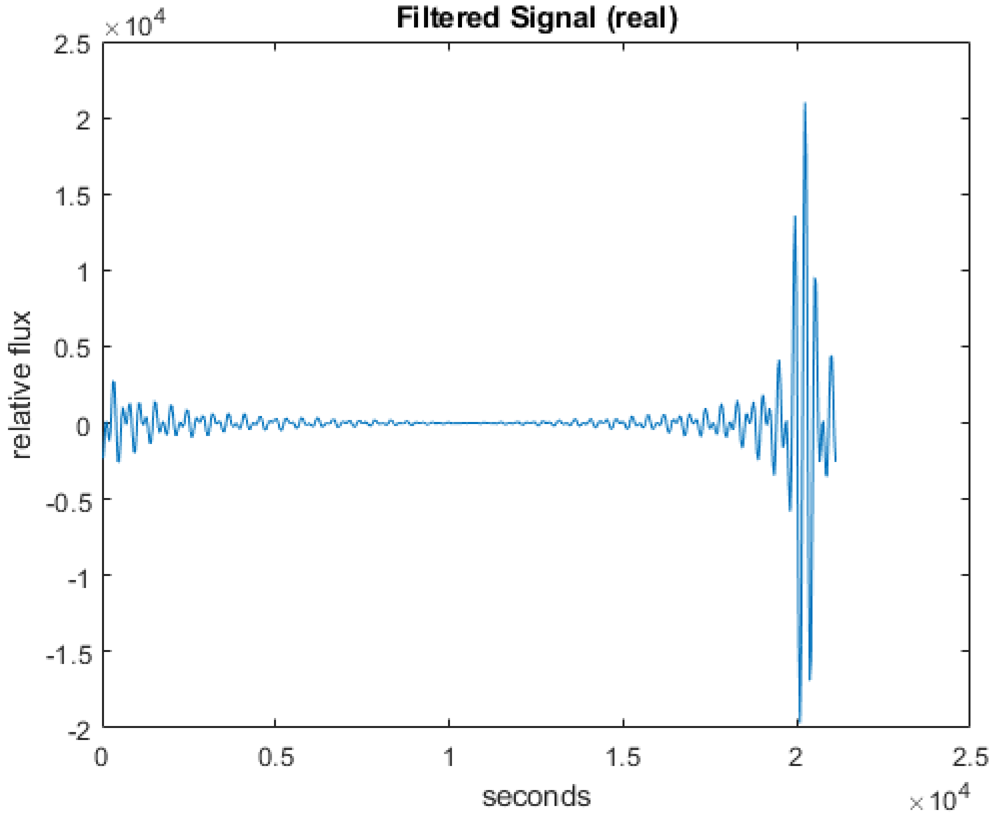

```
figure, stem(fk0(1:floor(L/2)),imag(C0(1:floor(L/2))),'ko','markersize',5)
xlabel('frequency vector')
ylabel('Imaginary Ck')
```



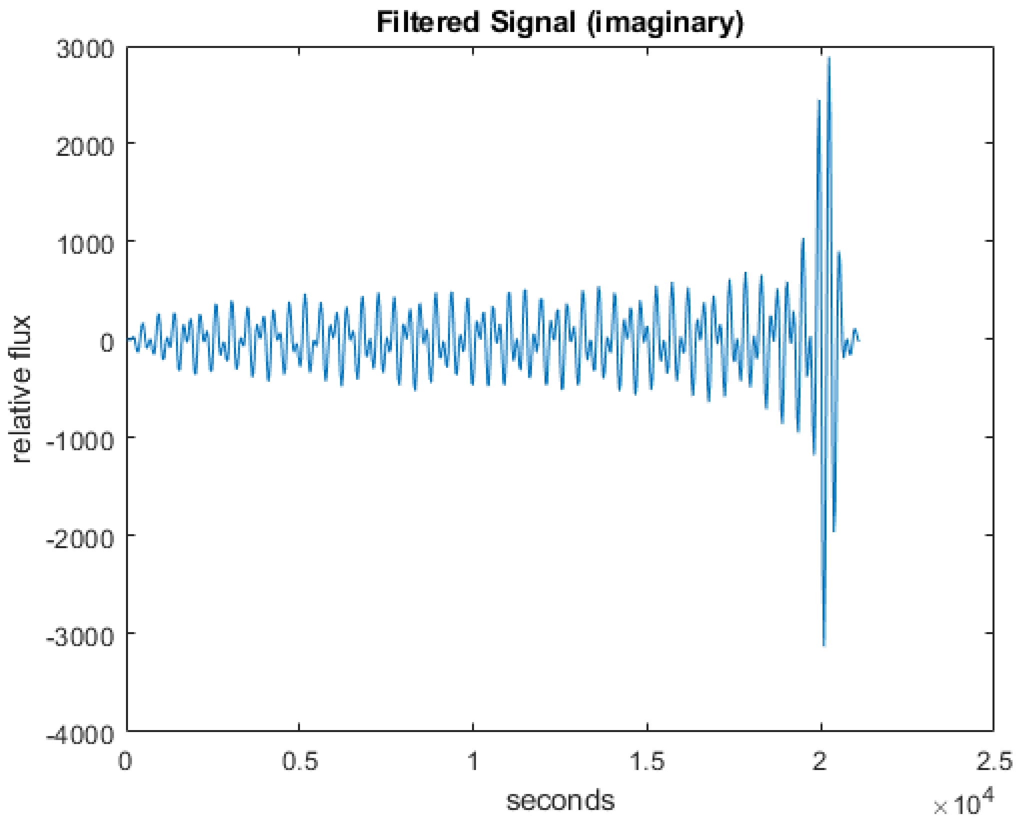

```
figure, stem(fk0(1:floor(L/2)),powernorm0(1:floor(L/2)),'ko','markersize',5)
xlabel('frequency (Hz)')
ylabel('Normalized Power')
```



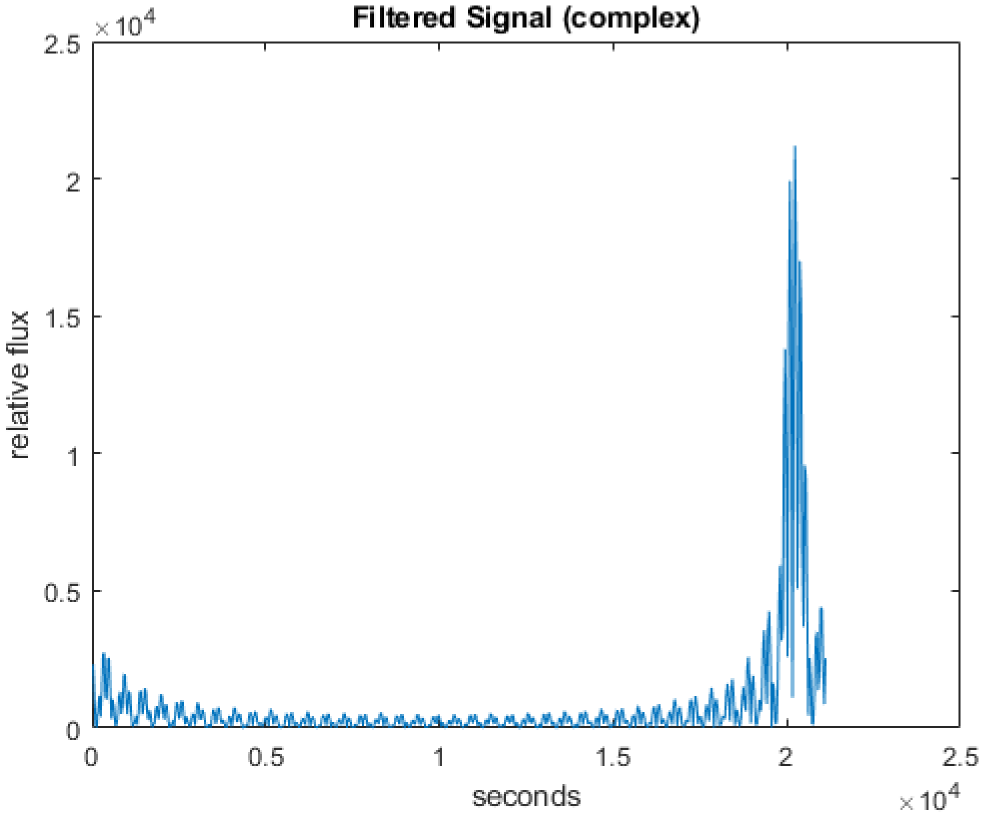

```
%Filtering out the noise            Based on Gilat Example 7-8

for k = 1:L-1
    if abs(fk0(k)) <= 50 || abs(fk0(k)) >= 90
        C0flter(k) = 0+0i;

    else
        C0flter(k) = C0(k);
    end
end


F0i = ifftshift(C0flter) * L-1;
finv = ifft(F0i);
figure, plot(t(2:L),real(finv))
title('Filtered Signal (real)')
xlabel('seconds')
ylabel('relative flux')
```



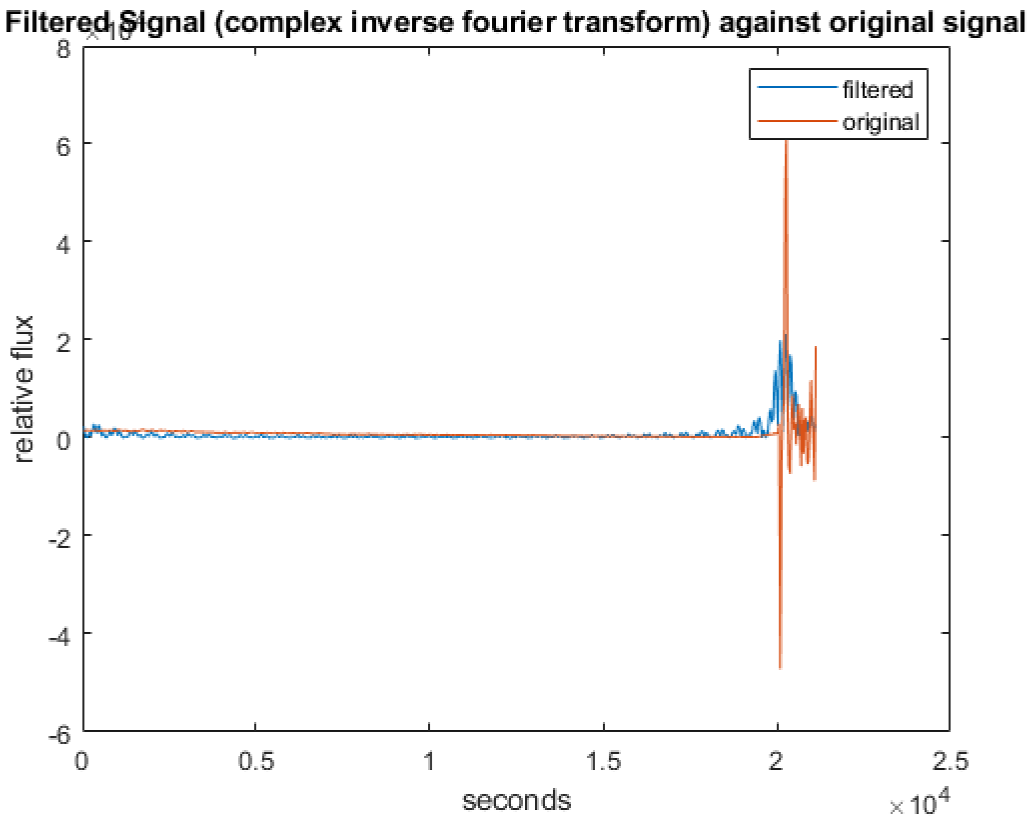

```
F0i = ifftshift(C0flter) * L-1;
finv = ifft(F0i);
figure, plot(t(2:L),imag(finv))
title('Filtered Signal (imaginary)')
xlabel('seconds')
ylabel('relative flux')
```



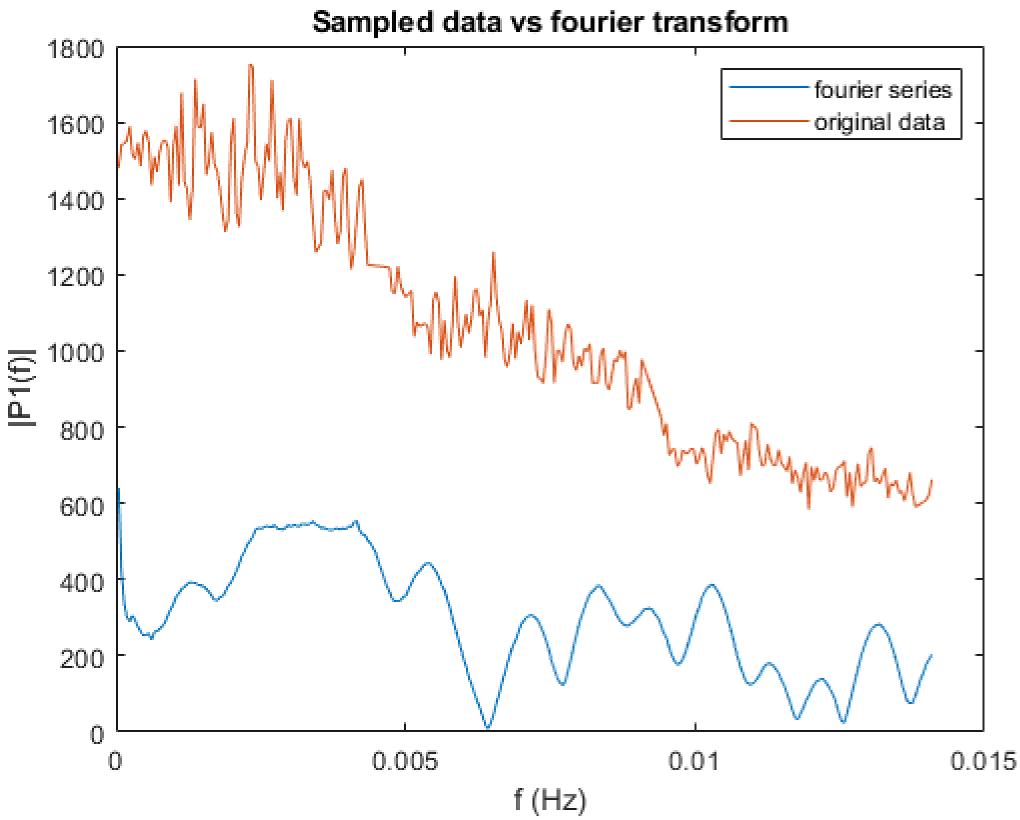

```
F0i = ifftshift(C0flter) * L-1;
finv = ifft(F0i);
figure, plot(t(2:L),abs(finv))
title('Filtered Signal (complex)')
xlabel('seconds')
ylabel('relative flux')
```




```
F0i = ifftshift(C0flter) * L-1;
finv = ifft(F0i);
figure, plot(t(2:L),abs(finv),t(2:L),X(2:L))
legend('filtered','original')
title('Filtered Signal (complex inverse fourier transform) against original signal')
xlabel('seconds')
ylabel('relative flux')
```




```
F =1./t(2:L);

%Fourier transformation
Y = fft(X);
P2complex = abs(Y/L); %real and imaginary parts together
P2real = 2*real(Y);
P2imaginary = 2*imag(Y);
P1complex = P2complex(1:floor(L/2)+1);
P1complex(2:end-1) = 2*P1complex(2:end-1);

%Plot

f = Fs*(0:(L/2))/L;
plot(f(2:end-1),P1complex(2:end-1),f(2:end-1),X(2:floor(l/2)))
title('Sampled data vs fourier transform')
xlabel('f (Hz)')
ylabel('|P1(f)|')
legend('fourier series','original data')
```

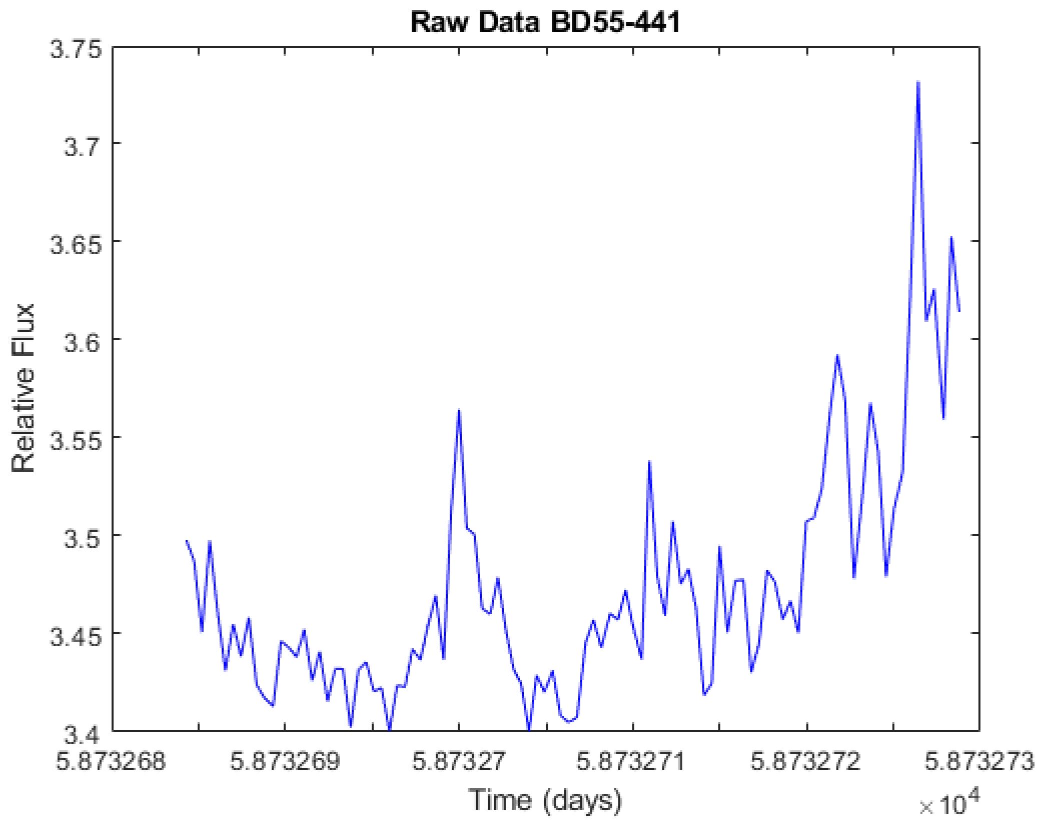


Milestone 4: Linear Combination of non-linear functions

BD55

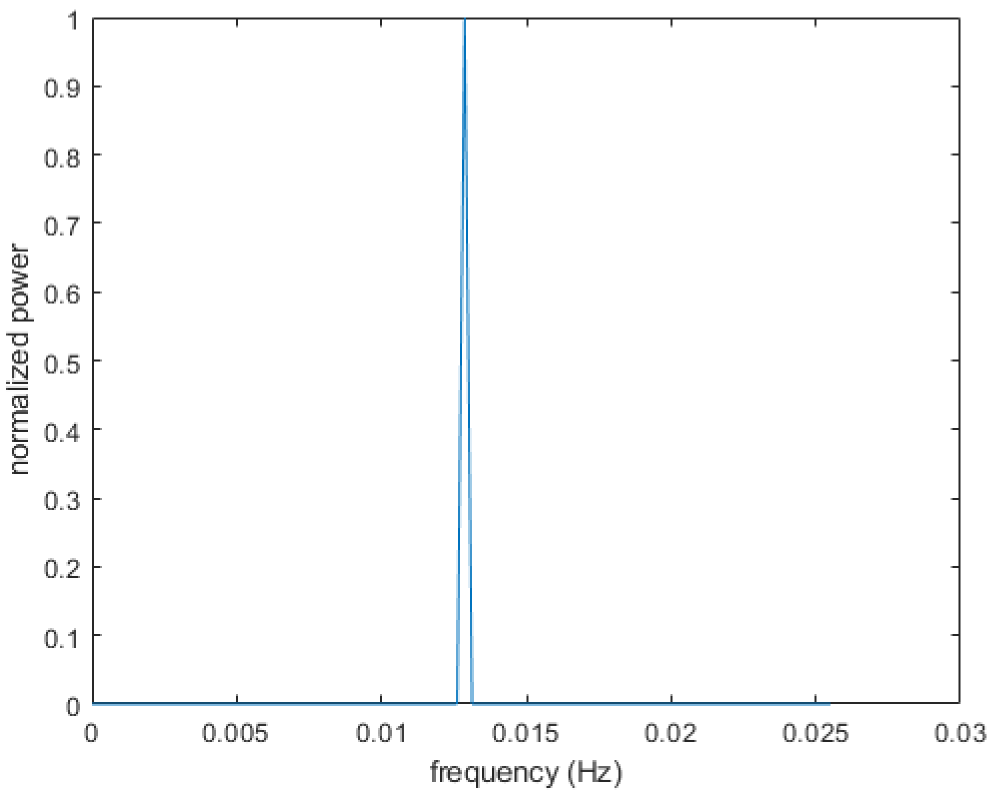
Plot the raw data

```
plot(BD55_441(1:100,1),BD55_441(1:100,2),'b');
title('Raw Data BD55-441');
xlabel('Time (days)');
ylabel('Relative Flux');
```



Use Linear Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_Lin(BD55_441(1:100,1),BD55_441(1:100,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt)
```




```
fk = 1×100
                  0    0.000257606417519    0.000515212835038    0.000772819252558 ···
powerNor = 1×100
   0.000000000470212    0.000000116089656    0.000000017611039    0.000000053274678 ···
```

Functions for linear combination:

(main frequency is 0.0129 HZ, or 1,115 per Day. Use this to choose my non-linear functions)

```
F1 =  @(x) x./x;
F2 =  @(x) cos(2*pi*(1115)).*(x);
F3 =  @(x) sin(2*pi*(1115)).*(x);


%call the NLfit function

c = NLfit(F1,F2,F3,BD55_441(1:100,1),BD55_441(1:100,2))
```

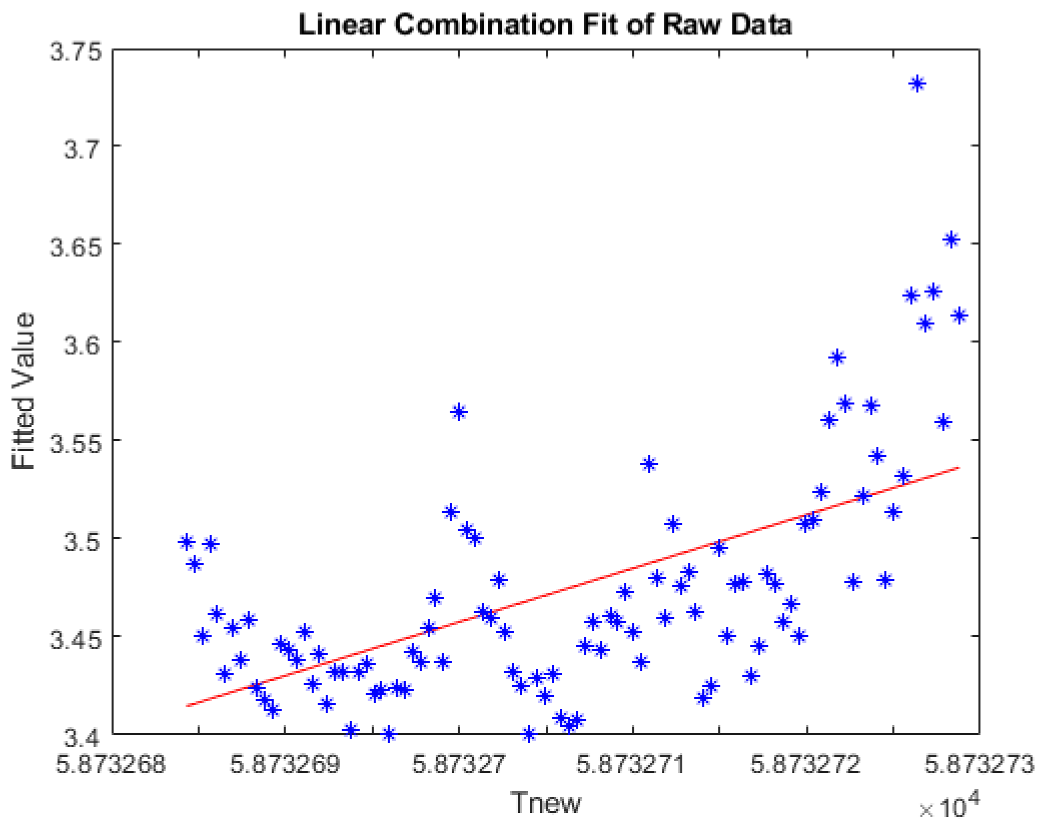
```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  2.980448e-42.
c = 3×1
```

10^{13} ×

```
  -0.000000016070897
   0.000000000000894
   3.724622381729035
```

```



%plot
```

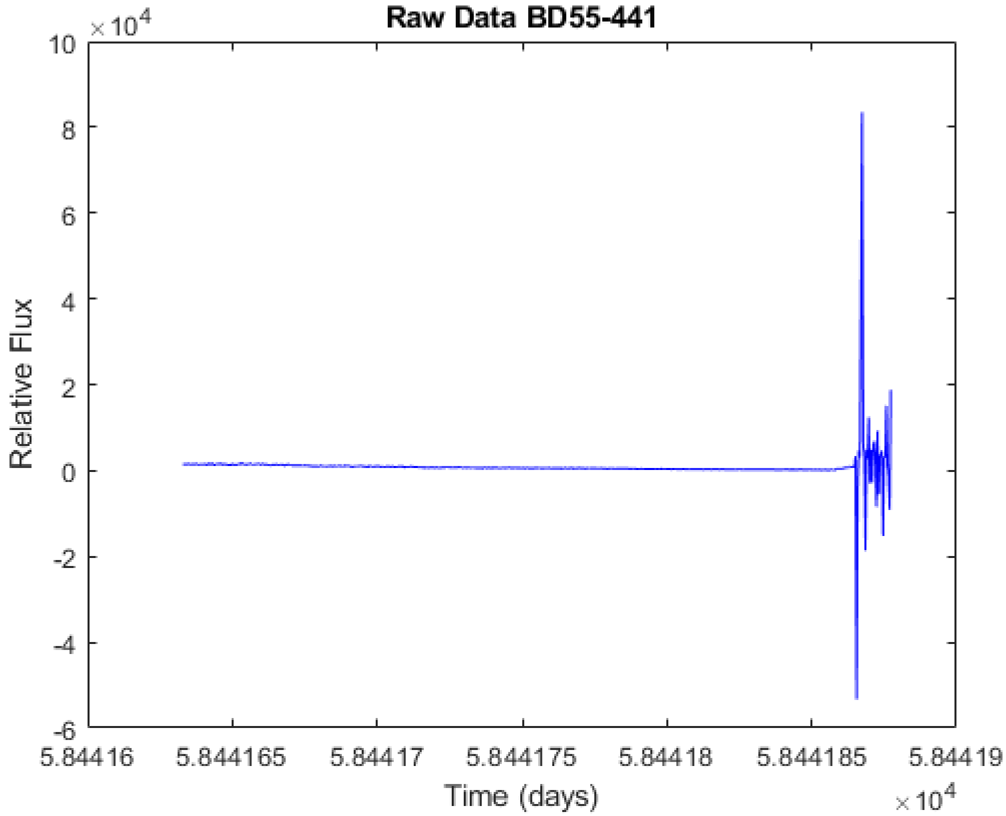
```
yfit =  c(1).*F1(Tnew) +  c(2).*F2(Tnew)  +  c(3).*F3(Tnew) ;
plot(Tnew,yfit,'r',BD55_441(1:100,1),BD55_441(1:100,2),'*b')
title('Linear Combination Fit of Raw Data')
xlabel('Tnew')
ylabel('Fitted Value')
```



HD28497

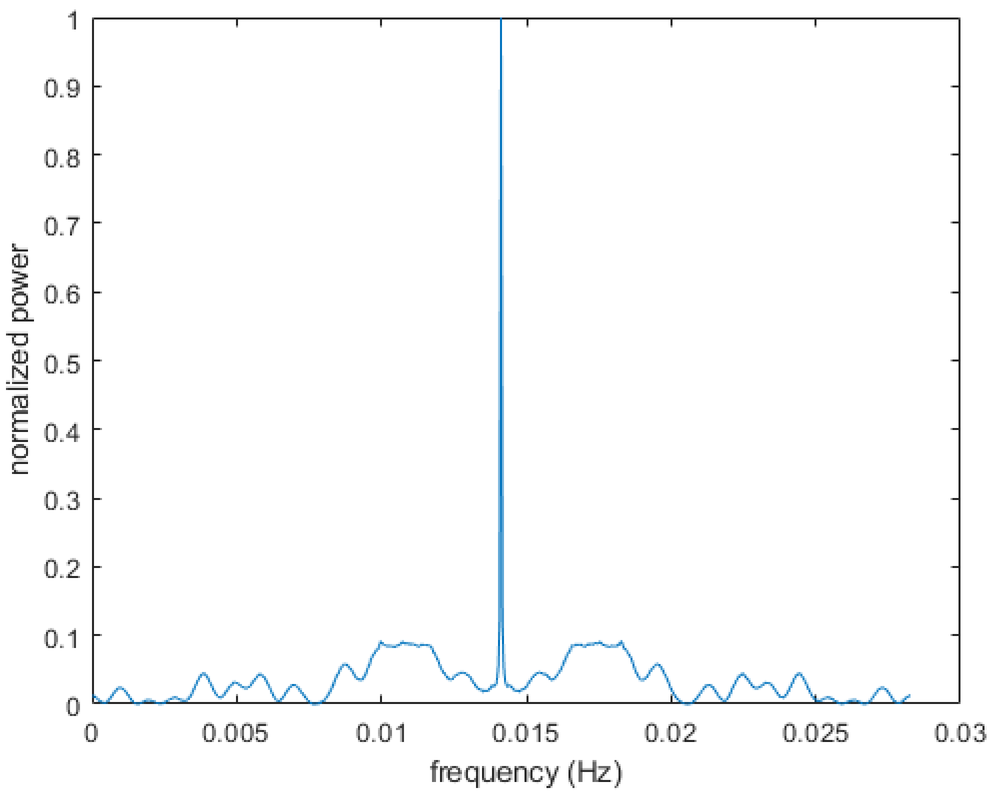
Plot the raw data

```
plot(HD28497(2:end,1),HD28497(2:end,2),'b');
title('Raw Data BD55-441');
xlabel('Time (days)');
ylabel('Relative Flux');
```



Use Linear Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_Lin(HD28497(2:end,1),HD28497(2:end,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Functions for linear combination:

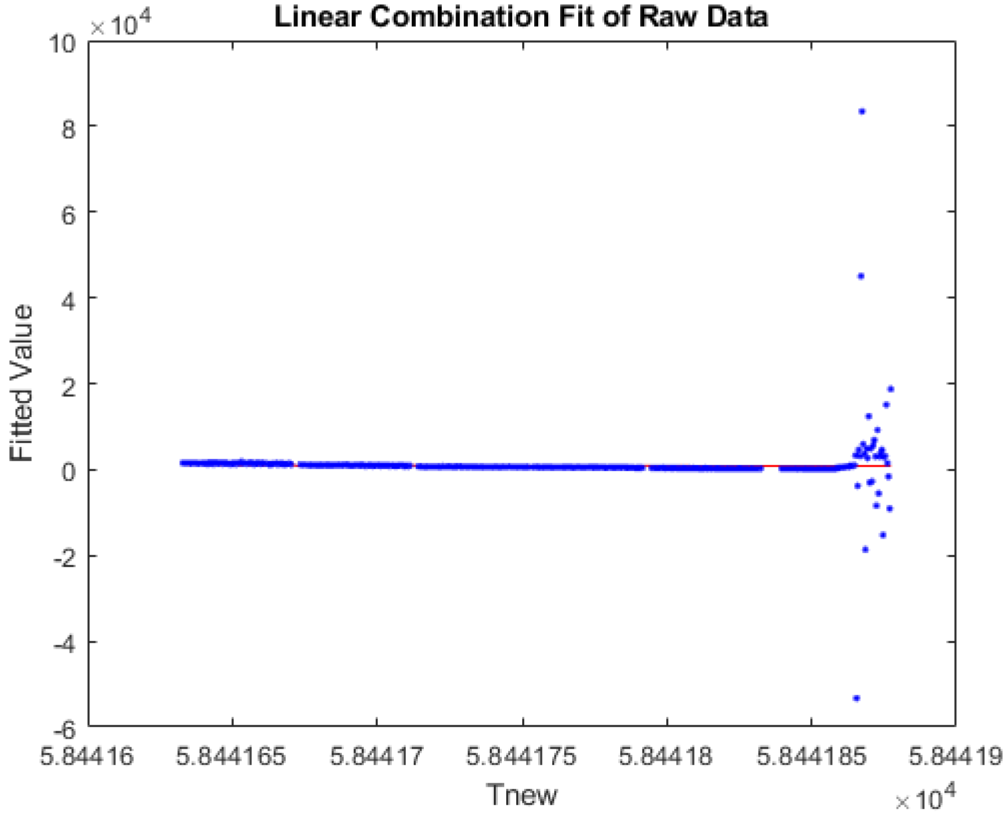
(main frequency at 0.0175 Hz or 1512 / day. use this for non-linear functions)

```
F1 =  @(x) x./x;
F2 =  @(x) cos(2*pi*(1512)).*(x);
F3 =  @(x) sin(2*pi*(1512)).*(x);


%call the NLfit function

c = NLfit(F1,F2,F3,HD28497(2:end,1),HD28497(2:end,2))
```

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  1.465326e-41.
c = 3×1
10^14 ×
   0.000000417432818
  -0.000000000006085
  -7.488964229280974
```

```
%plot

yfit =  c(1).*F1(Tnew) +  c(2).*F2(Tnew)  +  c(3).*F3(Tnew) ;
plot(Tnew,yfit,'r',HD28497(2:end,1),HD28497(2:end,2),'.b')
```

```
title('Linear Combination Fit of Raw Data')
xlabel('Tnew')
ylabel('Fitted Value')
```




```
%This section should be run independently.  This is a copy of what was
%formerly titled Milestone 5v7

clearvars
format long
```

Research Techniques Project Just Milestone 5

Input all data

Text files:

```
opts = detectImportOptions("BD+55_441.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
```

"time" is in the units of days and "flux" is "rel_flux_T1" from AstroImageJ outputs.

```
opts.VariableTypes = {'double','double','double','double','double','double'};
preview("BD+55_441.txt",opts)
```

ans = 8×6 table

	B_time	B_flux	R_time	R_flux	V_time	V_flux
1	5.873268430...	3.497996...	5.873268459...	1.961830...	5.873268445...	2.352058...
2	5.873268430...	3.497996...	5.873268504...	1.942857...	5.873268490...	2.360458...
3	5.873268475...	3.487289...	5.873268549...	1.947021...	5.873268535...	2.329957...
4	5.873268520...	3.450942...	5.873268594...	1.930594...	5.873268580...	2.344889...
5	5.873268565...	3.497567...	5.873268639...	1.942537...	5.873268625...	2.361048...
6	5.873268610...	3.461555...	5.873268684...	1.935141...	5.873268669...	2.333341...
7	5.873268655...	3.431248...	5.873268729...	1.931290...	5.873268714...	2.331904...
8	5.873268700...	3.455091...	5.873268774...	1.928260...	5.873268759...	2.323782...

```
BD55_441 = readmatrix("BD+55_441.txt",opts);
whos BD55_441
```

```
  Name             Size             Bytes  Class     Attributes

  BD55_441       101x6               4848  double
```

```



%BD+48_1098
opts = detectImportOptions("BD+48_1098.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
BD48_1098 = rmmissing(readmatrix("BD+55_441.txt",opts)); %the matrix data of the text file
%preview("BD+48_1098.txt",opts)
whos BD48_1098
```

```
  Name             Size             Bytes  Class     Attributes

  BD48_1098       100x6              4800  double
```

```
%HD28497
opts = detectImportOptions("HD28497.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD28497 = rmmissing(readmatrix("HD28497.txt",opts)) %the matrix data of the text file
```

```
HD28497 = 600×6
10^4 ×
   5.844163256000000   0.151884039400000   5.844163279000000   0.047485834000000 ···
   5.844163294000000   0.157713255000000   5.844163317000000   0.047850097100000
   5.844163332000000   0.147648080800000   5.844163355000000   0.050717498800000
   5.844163370000000   0.154105058200000   5.844163393000001   0.048745268800000
   5.844163409000000   0.154416998300000   5.844163431999999   0.046480291500000
   5.844163447000000   0.154034530100000   5.844163470000000   0.052878906200000
```

```
5.844163485000000   0.158524561400000   5.844163508000000   0.050080829500000
5.844163523000000   0.159731365000000   5.844163546000000   0.053844298900000
5.844163560999999   0.140623476800000   5.844163584000000   0.050871022200000
5.844163599000000   0.159835953000000   5.844163622000000   0.053311599500000
    ⋮
```

```matlab
%preview("HD28497.txt",opts)
whos HD28497
```

```
  Name           Size             Bytes  Class     Attributes

  HD28497      600x6              28800  double
```

```matlab
%HD46131
opts = detectImportOptions("HD46131.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_flux','B_time','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD46131 = rmmissing(readmatrix("HD46131.txt",opts)); %the matrix data of the text file
preview("HD46131.txt",opts)
```

ans = 8×6 table

	B_flux	B_time	R_time	R_flux	V_time	V_flux
1	80.30418...	5.844670852...	29.69434...	5.844670891...	43.24278...	5.844670875...
2	87.01571...	5.844670916...	29.50730...	5.844670956...	43.89599...	5.844670939...
3	86.15075...	5.844670981...	29.59417...	5.844671021...	43.11367...	5.844671004...
4	84.58512...	5.844671046...	29.01948...	5.844671085...	43.53672...	5.844671069...
5	82.72335...	5.844671111...	29.25732...	5.844671150...	44.34534...	5.844671134...
6	79.70069...	5.844671175...	30.09976...	5.844671215...	43.52612...	5.844671199...
7	84.42755...	5.844671240...	29.99023...	5.844671280...	42.63575...	5.844671263...
8	79.15342...	5.844671305...	29.78040...	5.844671344...	43.53322...	5.844671329...

```matlab
whos HD46131
```

```
  Name           Size             Bytes  Class     Attributes

  HD46131      250x6              12000  double
```

```matlab
%HD88661
opts = detectImportOptions("HD88661.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD88661 = rmmissing(readmatrix("HD88661.txt",opts)); %the matrix data of the text file
%preview("HD88661.txt",opts)
whos HD88661
```

```
  Name          Size             Bytes  Class     Attributes

  HD88661      35x6               1680  double
```

```
%HD105521
opts = detectImportOptions("HD105521.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD105521 = rmmissing(readmatrix("HD105521.txt",opts)); %the matrix data of the text file
%preview("HD105521.txt",opts)
whos HD105521
```

```
  Name            Size             Bytes  Class     Attributes

  HD105521       180x6              8640  double
```


```
%HD105521
opts = detectImportOptions("HD105521.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD105521 = rmmissing(readmatrix("HD105521.txt",opts)); %the matrix data of the text file
%preview("HD105521.txt",opts)
whos HD105521
```

```
  Name            Size             Bytes  Class     Attributes

  HD105521       180x6              8640  double
```


```
%KIC_11924311
opts = detectImportOptions("KIC_11924311_Photometry.txt");
opts.DataLines = 3;
%opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableNames = {'Time','Filter','Flux','error'};
opts.VariableTypes = {'double','char','double','double'};
KIC_11924311 = rmmissing(readtable("KIC_11924311_Photometry.txt",opts)); %the matrix data of th
preview("KIC_11924311_Photometry.txt",opts)
```

ans = 8×4 table

	Time	Filter	Flux	error
1	5.79853734...	'B'	0.68590...	0.00300...
2	5.79853742...	'B'	0.68147...	0.00299...
3	5.79853758...	'B'	0.69522...	0.00301...
4	5.79853766...	'B'	0.69789...	0.00303...
5	5.79853775...	'B'	0.69234...	0.00300...
6	5.79853783...	'B'	0.70016...	0.00303...
7	5.79853791...	'B'	0.69389...	0.00299...
8	5.79853799...	'B'	0.70966...	0.00305...

```
filters = unique(KIC_11924311{:,2},'stable');
poscount = [1 0 0 0 0];
[n,~] = size(KIC_11924311);
KIC_11924311temp = table;
l = {0};
```

```
for count=1:4
    poscount(count+1)=poscount(count);
    while (char(filters(count))==char(KIC_11924311{poscount(count+1),2}) && poscount(count+1)<=
        poscount(count+1) = poscount(count+1) + 1;
    end
    KIC = KIC_11924311(poscount(count):poscount(count+1)-1,:)
    KIC = KIC(1:348,:);
    KIC_11924311temp.J_D__2400000 = KIC.Time;
    KIC_11924311temp.rel_flux_T1 = KIC.Flux;
    l{2*count-1} = sprintf('%s_time', char(filters(count)));
    l{2*count} = sprintf('%s_flux', char(filters(count)));
    KIC_11924311temp.Properties.VariableNames = l;
end
```

KIC = 348×4 table

	Time	Filter	Flux	error
1	5.79853734...	'B'	0.68590...	0.00300...
2	5.79853742...	'B'	0.68147...	0.00299...
3	5.79853758...	'B'	0.69522...	0.00301...
4	5.79853766...	'B'	0.69789...	0.00303...
5	5.79853775...	'B'	0.69234...	0.00300...
6	5.79853783...	'B'	0.70016...	0.00303...
7	5.79853791...	'B'	0.69389...	0.00299...
8	5.79853799...	'B'	0.70966...	0.00305...
9	5.79853807...	'B'	0.70262...	0.00302...
10	5.79853815...	'B'	0.70943...	0.00305...
11	5.79853823...	'B'	0.71280...	0.00305...
12	5.79853831...	'B'	0.71538...	0.00306...
13	5.79853840...	'B'	0.71921...	0.00314...
14	5.79853848...	'B'	0.72563...	0.00327...

⋮

KIC = 350×4 table

	Time	Filter	Flux	error
1	5.79853728...	'V'	0.72607...	0.00284...
2	5.79853736...	'V'	0.72676...	0.00284...
3	5.79853744...	'V'	0.72876...	0.00283...
4	5.79853752...	'V'	0.73102...	0.00284...
5	5.79853760...	'V'	0.73476...	0.00285...
6	5.79853769...	'V'	0.74009...	0.00287...
7	5.79853777...	'V'	0.74062...	0.00285...
8	5.79853785...	'V'	0.74477...	0.00287...

	Time	Filter	Flux	error
9	5.79853793...	'V'	0.74259...	0.00285...
10	5.79853801...	'V'	0.74713...	0.00287...
11	5.79853809...	'V'	0.74955...	0.00288...
12	5.79853817...	'V'	0.75450...	0.00289...
13	5.79853826...	'V'	0.76023...	0.00291...
14	5.79853834...	'V'	0.76027...	0.00292...

⋮

KIC = 350×4 table

	Time	Filter	Flux	error
1	5.79853729...	'R'	0.74284...	0.00304...
2	5.79853737...	'R'	0.74455...	0.00306...
3	5.79853746...	'R'	0.75454...	0.00309...
4	5.79853754...	'R'	0.75485...	0.00307...
5	5.79853762...	'R'	0.75807...	0.00307...
6	5.79853770...	'R'	0.76211...	0.00308...
7	5.79853778...	'R'	0.76860...	0.00311...
8	5.79853787...	'R'	0.77020...	0.00313...
9	5.79853795...	'R'	0.76533...	0.00309...
10	5.79853803...	'R'	0.76891...	0.00309...
11	5.79853811...	'R'	0.76780...	0.00309...
12	5.79853819...	'R'	0.77622...	0.00313...
13	5.79853827...	'R'	0.78118...	0.00314...
14	5.79853835...	'R'	0.78604...	0.00316...

⋮

KIC = 350×4 table

	Time	Filter	Flux	error
1	5.79853731...	'I'	0.76493...	0.00514...
2	5.79853739...	'I'	0.78074...	0.00526...
3	5.79853747...	'I'	0.77509...	0.00512...
4	5.79853755...	'I'	0.77435...	0.00518...
5	5.79853763...	'I'	0.78233...	0.00519...
6	5.79853772...	'I'	0.78959...	0.00518...
7	5.79853780...	'I'	0.78702...	0.00513...
8	5.79853788...	'I'	0.78799...	0.00519...

	Time	Filter	Flux	error
9	5.79853796...	'I'	0.79257...	0.00515...
10	5.79853805...	'I'	0.79262...	0.00514...
11	5.79853813...	'I'	0.79891...	0.00518...
12	5.79853821...	'I'	0.80309...	0.00517...
13	5.79853829...	'I'	0.81053...	0.00517...
14	5.79853837...	'I'	0.80186...	0.00522...

⋮

```
poscount
```

```
poscount = 1×5
         1         349         699        1049        1399
```

```
KIC_11924311 = KIC_11924311temp;
whos KIC_11924311
```

```
  Name                 Size              Bytes  Class     Attributes

  KIC_11924311       348x8               24799  table
```

CSV files

```
%HD106306
%B_filter
opts = detectImportOptions('HD106306_B.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD106306_B.csv",opts)
HD106306_B = readmatrix("HD106306_B.csv",opts);

%R_filter
opts = detectImportOptions('HD106306_R.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD106306_R.csv",opts)
HD106306_R = readmatrix("HD106306_R.csv",opts);

%V_filter
opts = detectImportOptions('HD106306_V.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD106306_V.csv",opts)
HD106306_V = readmatrix("HD106306_V.csv",opts);

HD106306 = rmmissing([HD106306_B HD106306_R HD106306_V]); %the matrix data of the text file
whos HD106306
```

```
  Name            Size             Bytes  Class     Attributes

  HD106306      100x6               4800  double
```

```
%HD147302
%B_filter
opts = detectImportOptions('HD147302_B.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD147302_B.csv",opts)
HD147302_B = readmatrix("HD147302_B.csv",opts);

%R_filter
opts = detectImportOptions('HD147302_R.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD147302_R.csv",opts)
HD147302_R = readmatrix("HD147302_R.csv",opts);

%V_filter
opts = detectImportOptions('HD147302_V.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD147302_V.csv",opts)
HD147302_V = readmatrix("HD147302_V.csv",opts);

HD147302 = rmmissing([HD147302_B HD147302_R HD147302_V]); %the matrix data of the text file
whos HD147302
```

```
  Name            Size             Bytes  Class     Attributes

  HD147302      100x6               4800  double
```

HD152060 and HD209522 can't be described in the same form as the others. Will investigate

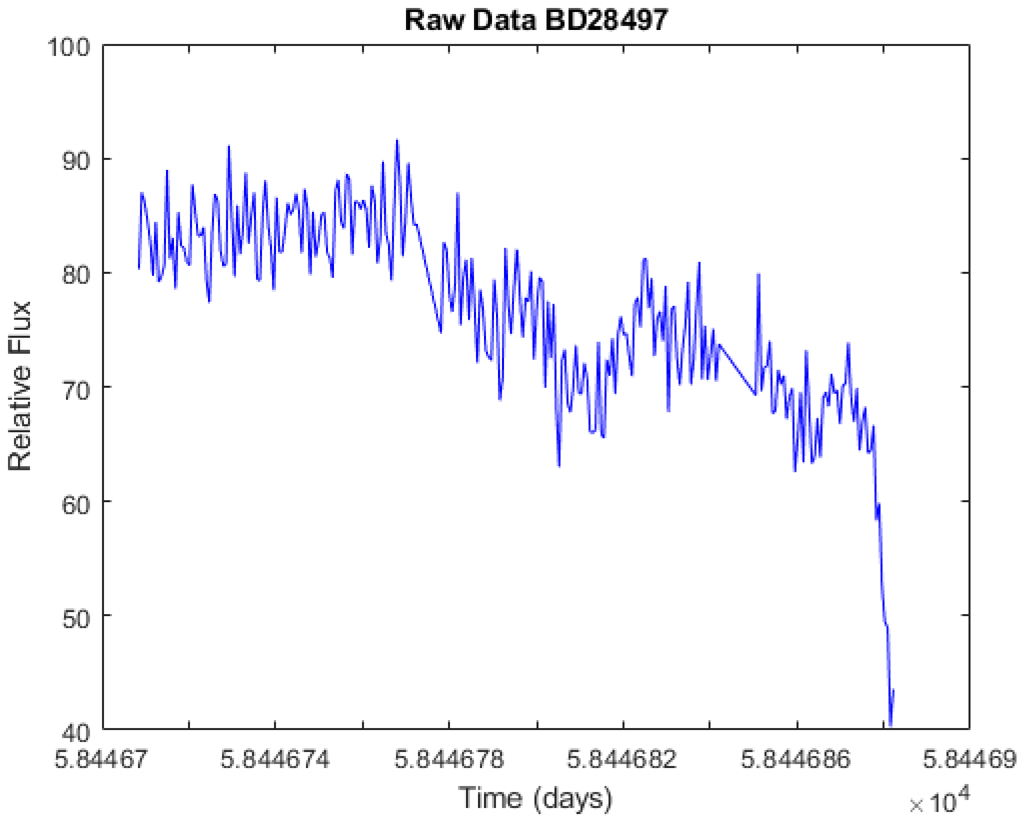
```
close all   %closes all previous figures from milestone 1 so that the plots in milestone 2 and
```

HD46131

Plot the raw data

```
plot(HD46131(:,2),HD46131(:,1),'b');
title('Raw Data BD28497');
```

```
xlabel('Time (days)');
ylabel('Relative Flux');
```

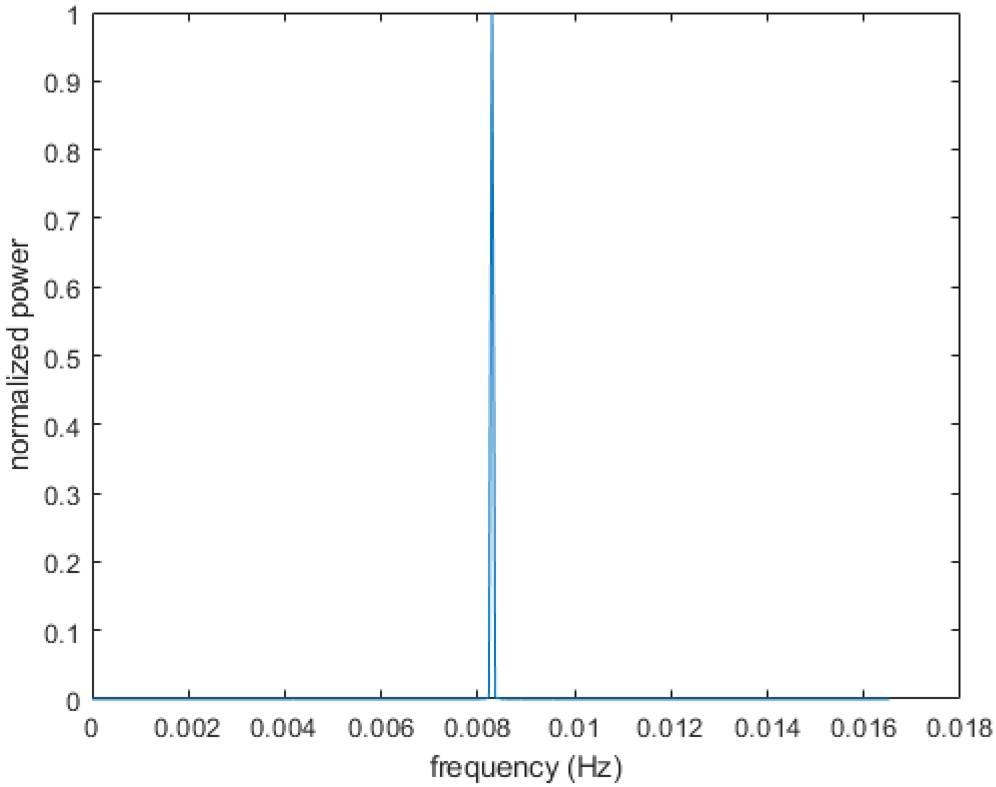



```
avg_freq =1/( mean( diff(HD46131(:,2)) ) *86400 )          %Dr. Riousset showed us how to use
```

```
avg_freq =
   0.016592460385689
```

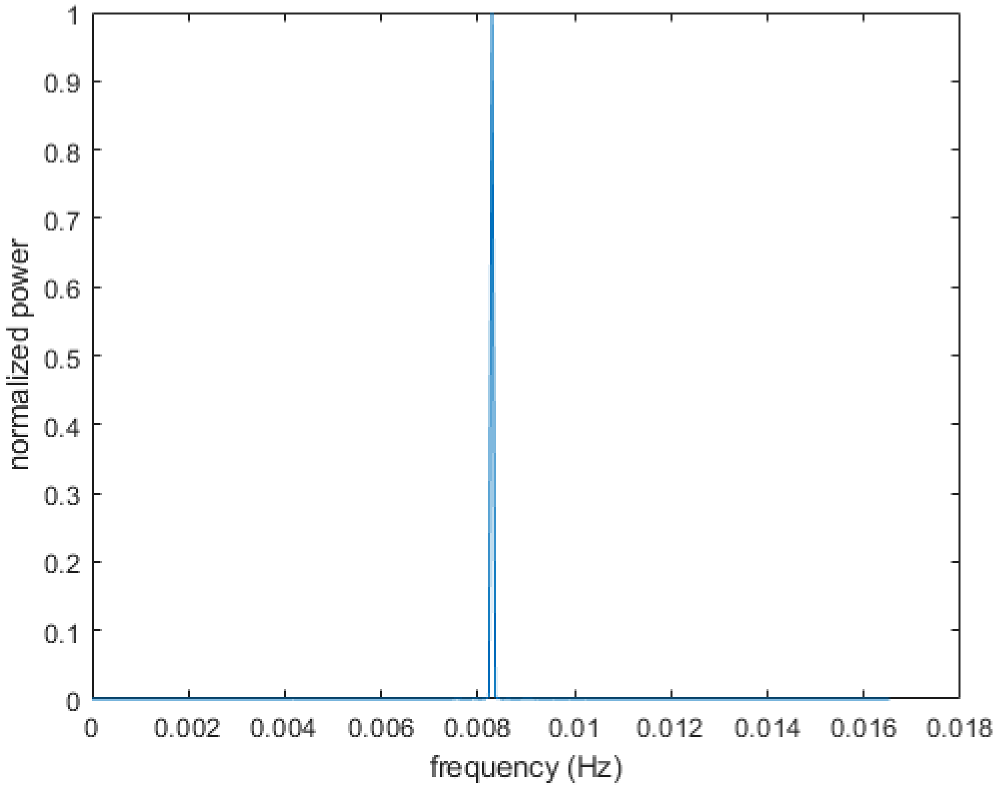
Use Linear Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_Lin(HD46131(:,2),HD46131(:,1));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



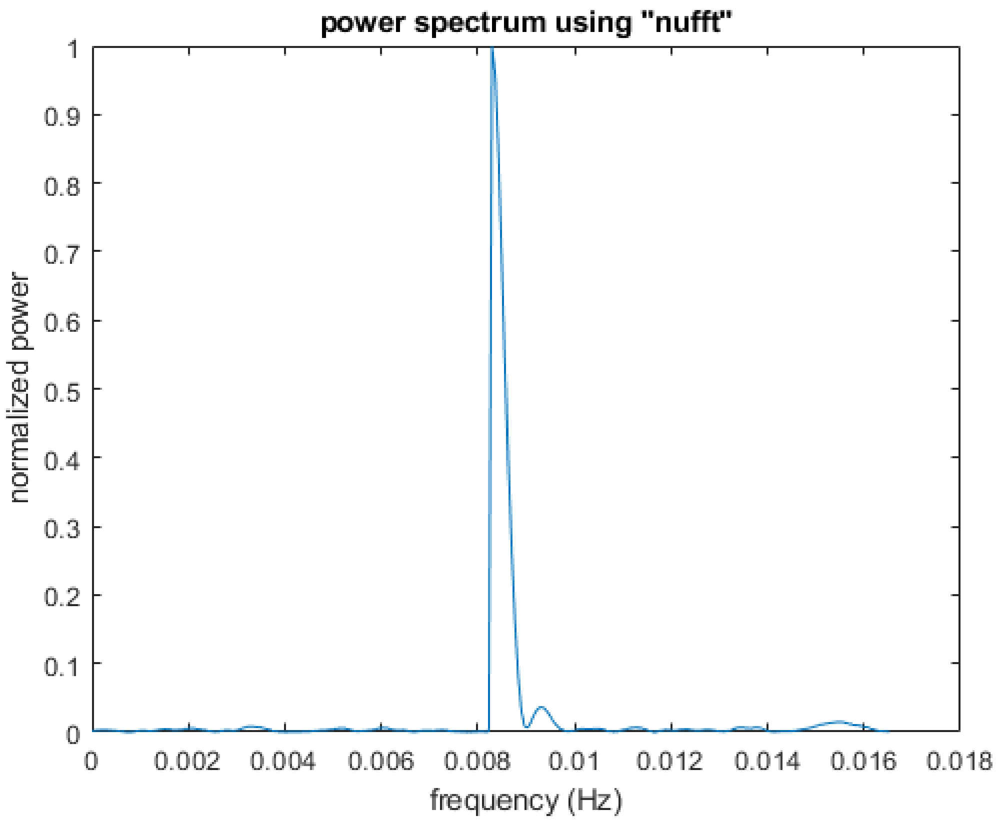
Use Spline Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_spline(HD46131(:,2),HD46131(:,1));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Try non-uniform built in function "nufft"

```
N = length(HD46131(:,2));
F = nufft( HD46131(:,2),HD46131(:,1) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```



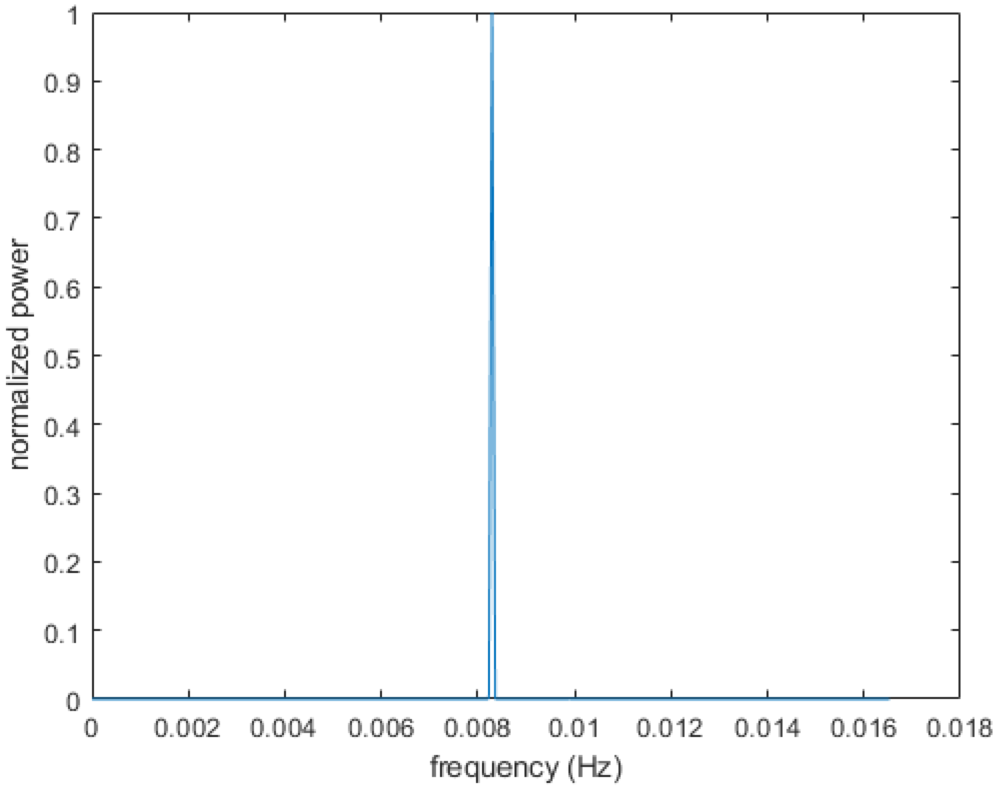

```
key_freq = 0.008296;
avg_freq =1/( mean( diff(HD46131(:,2)) ) *86400 )
```

```
avg_freq =
   0.016592460385689
```

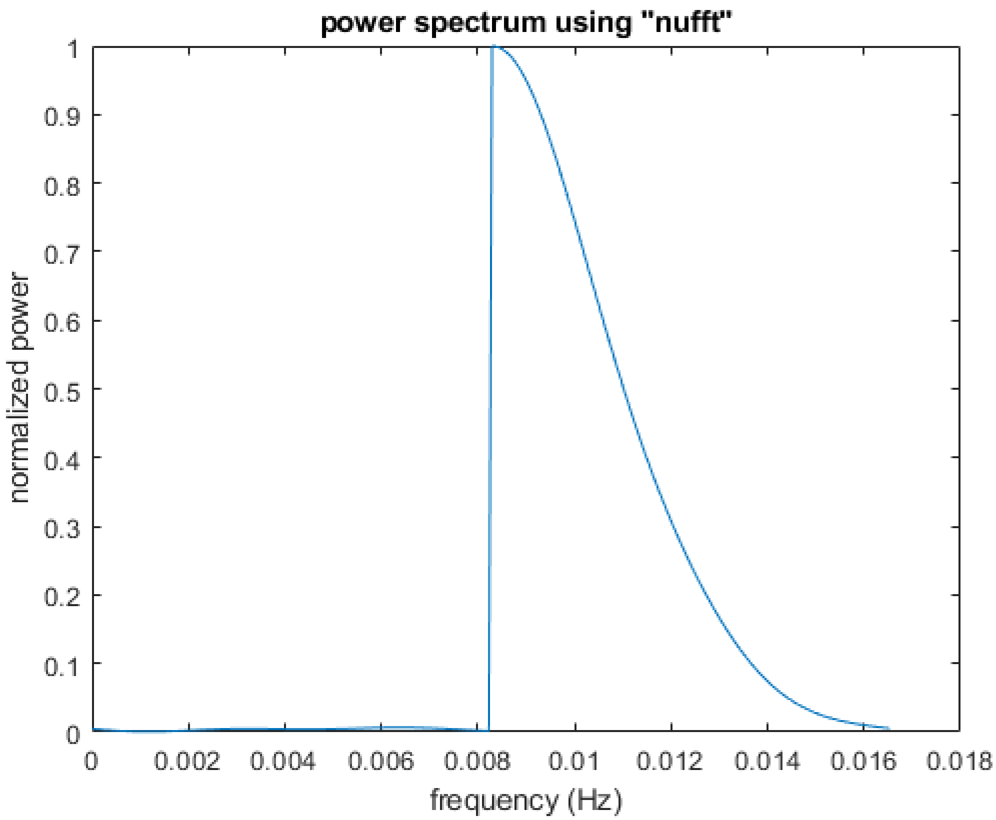
```
avg_freq/key_freq
```

```
ans =
   2.000055494899830
```

```
%R
[Tnew,Mnew] = Interp_spline(HD46131(:,4),HD46131(:,3));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```




```
avg_freq =1/( mean( diff(HD46131(:,4)) ) *86400 );
N = length(HD46131(:,4));
F = nufft( HD46131(:,4),HD46131(:,3) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;
N = length(HD46131(:,4));
F = nufft( HD46131(:,4),HD46131(:,3) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```



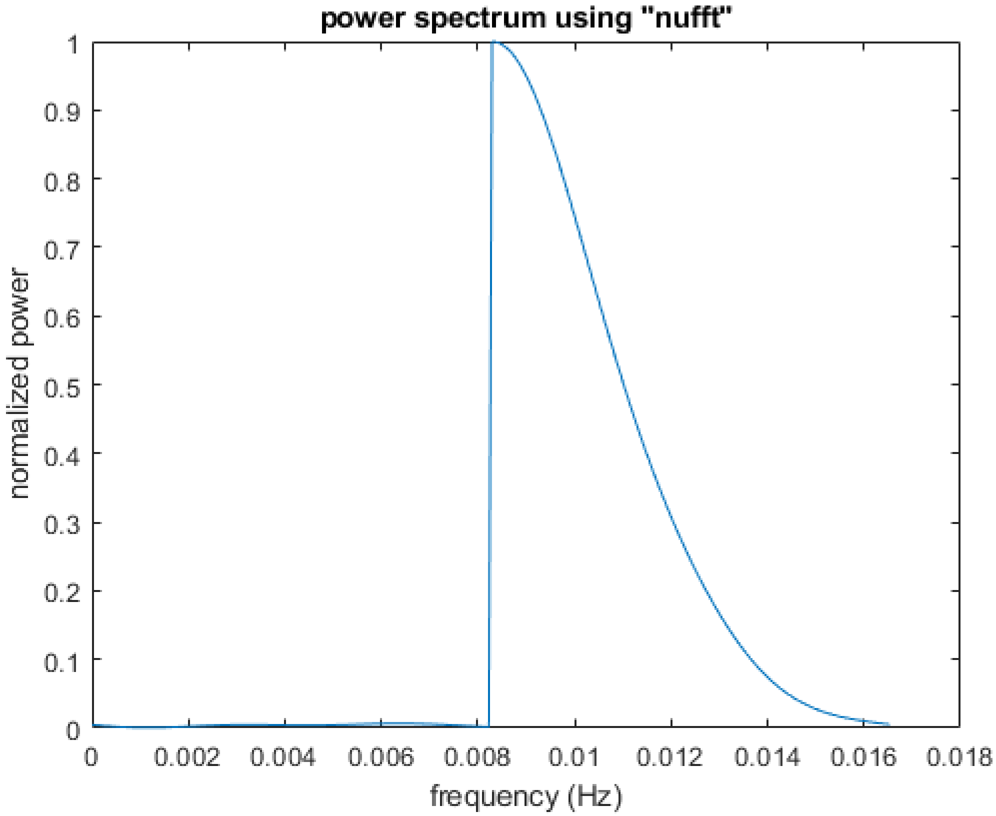

```
key_freq = 0.00829623;
avg_freq =1/( mean( diff(HD46131(:,4)) ) *86400 )
```

```
avg_freq =
   0.016592460386384
```

```
avg_freq/key_freq
```

```
ans =
   2.000000046573450
```

```
powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```



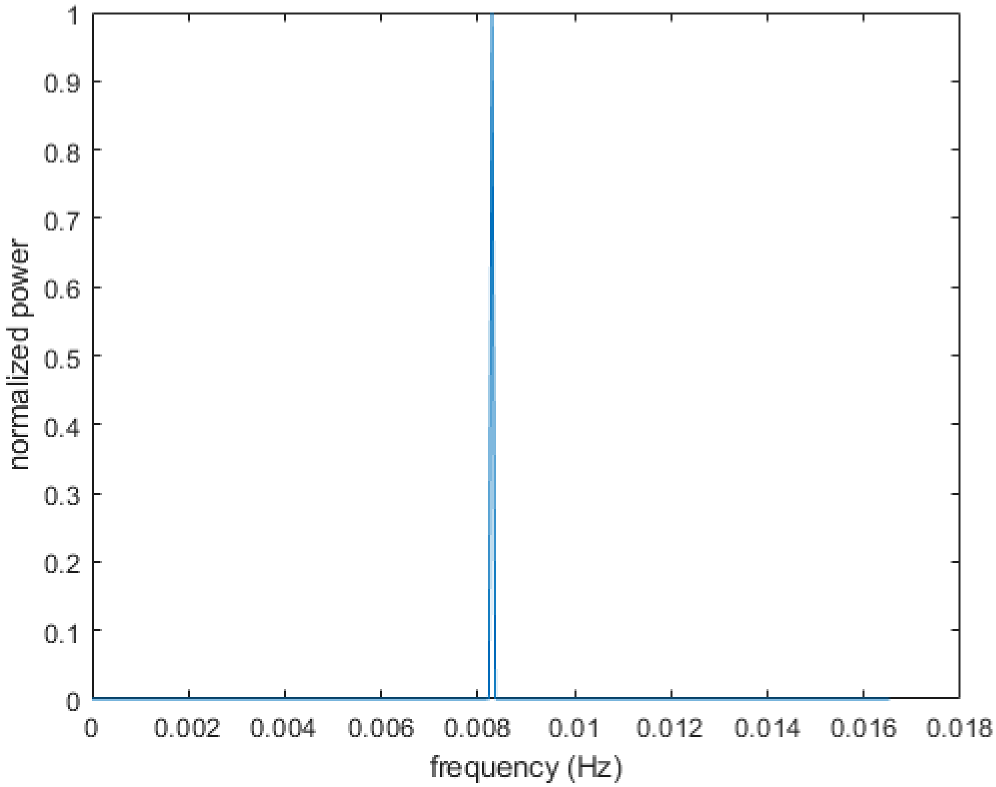

```
key_freq = 0.008296;
avg_freq =1/( mean( diff(HD46131(:,4)) ) *86400 )
```

```
avg_freq =
   0.016592460386384
```

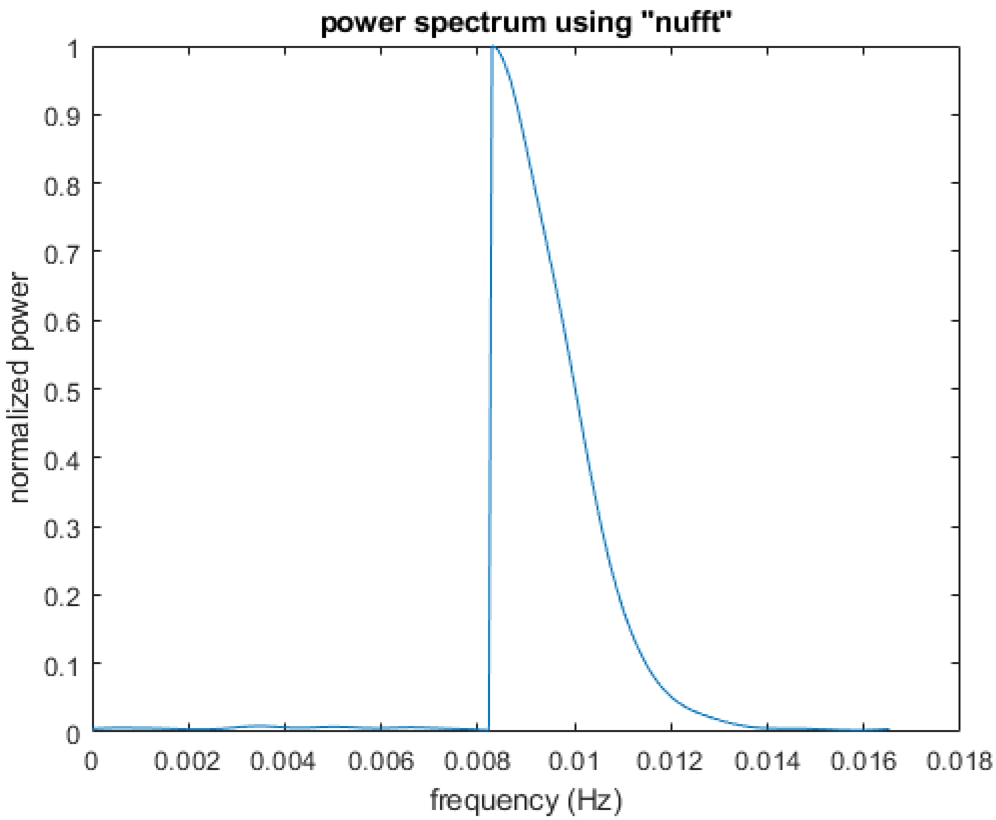
```
avg_freq/key_freq
```

```
ans =
   2.000055494983613
```

```

%v
[Tnew,Mnew] = Interp_spline(HD46131(:,6),HD46131(:,5));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```




```
avg_freq =1/( mean( diff(HD46131(:,6)) ) *86400 );
N = length(HD46131(:,6));
F = nufft( HD46131(:,6),HD46131(:,5) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;
N = length(HD46131(:,6));
F = nufft( HD46131(:,6),HD46131(:,5) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```



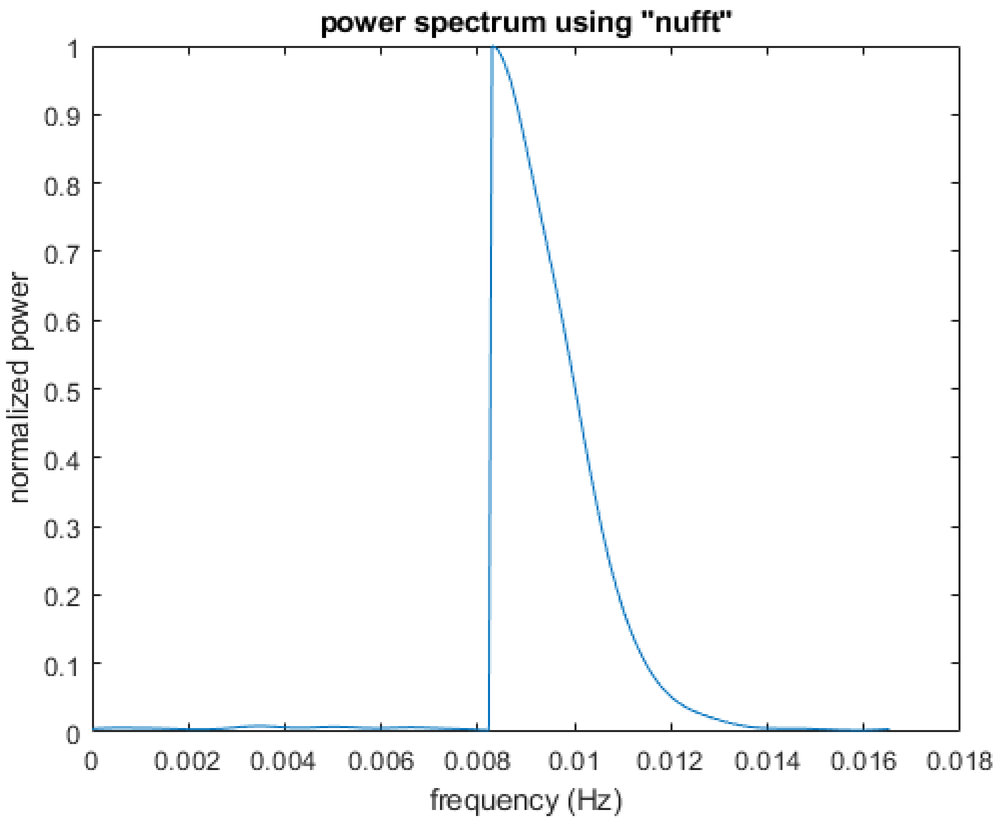

```
key_freq = 0.00829623;
avg_freq =1/( mean( diff(HD46131(:,6)) ) *86400 )
```

```
avg_freq =
   0.016592460385689
```

```
avg_freq/key_freq
```

```
ans =
   2.000000046489669
```

```
powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.008296;
avg_freq =1/( mean( diff(HD46131(:,4)) ) *86400 )
```

```
avg_freq =
   0.016592460386384
```

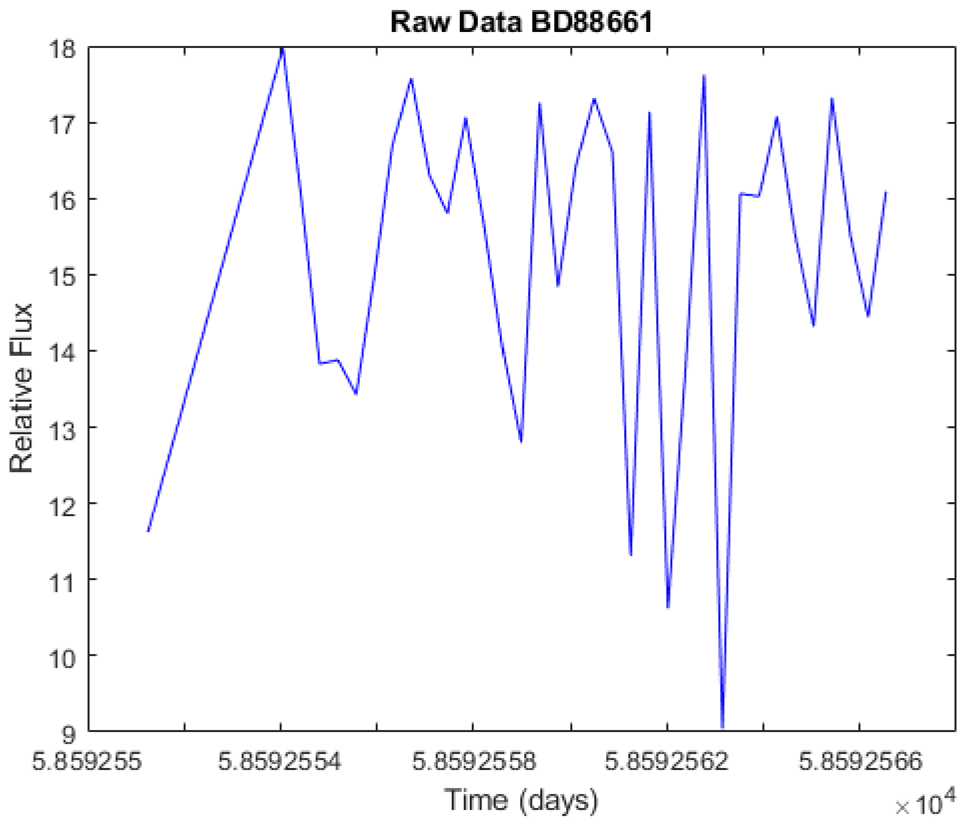
```
avg_freq/key_freq
```

```
ans =
   2.000055494983613
```

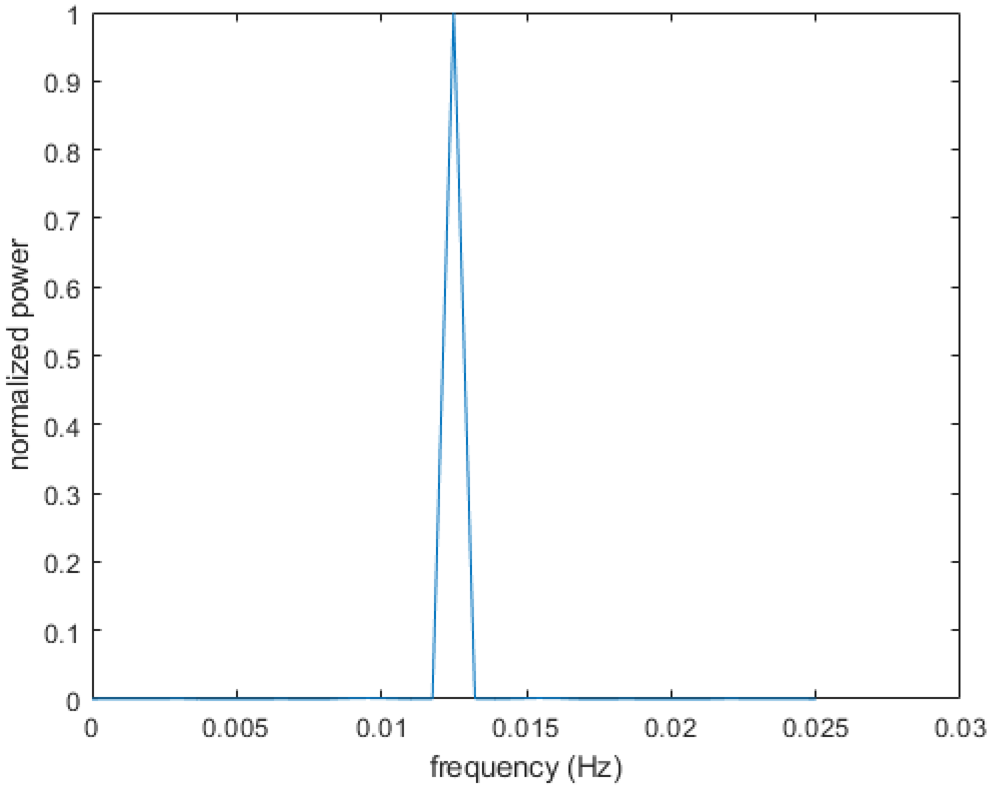
HD88661

Plot the raw data

```
plot(HD88661(:,2),HD88661(:,1),'b');
title('Raw Data BD88661');
xlabel('Time (days)');
ylabel('Relative Flux');
```

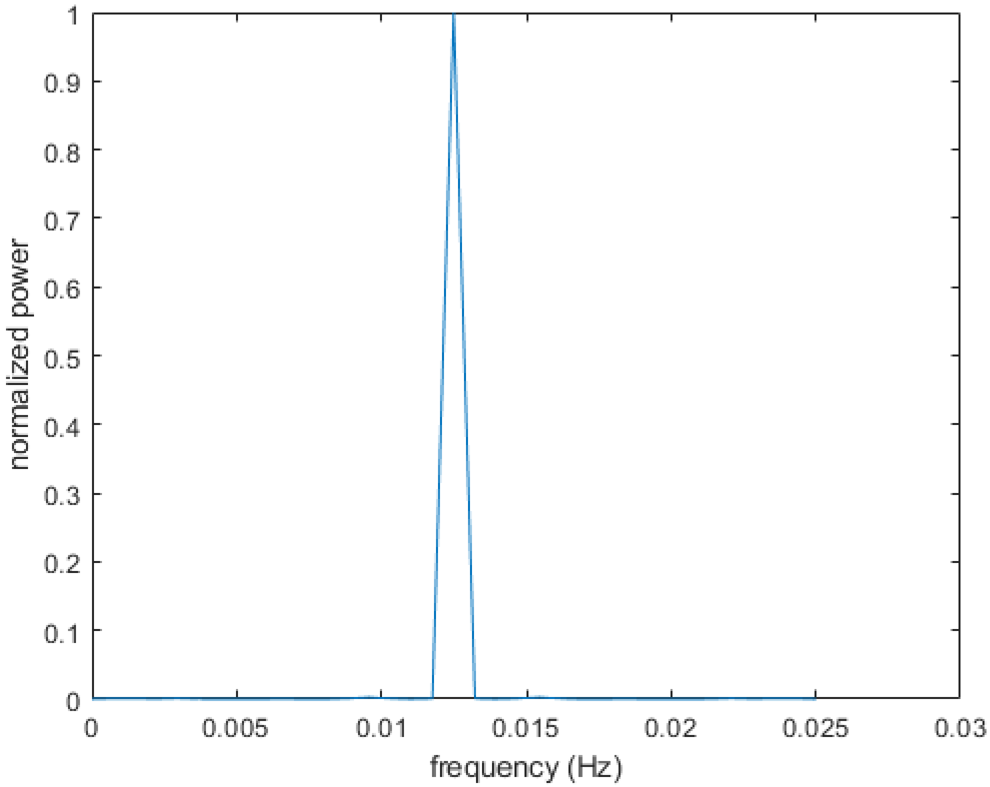



```
%B
avg_freq =1/( mean( diff(HD88661(:,2)) ) *86400 );
[Tnew,Mnew] = Interp_Lin(HD88661(:,2),HD88661(:,1));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



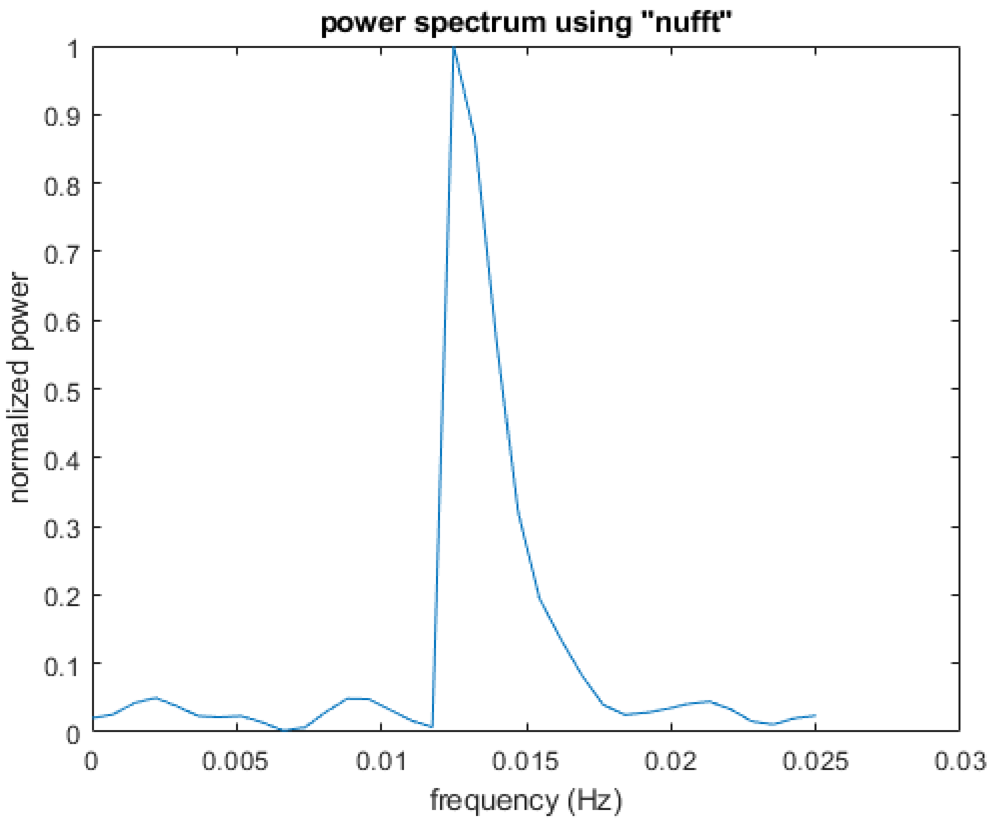
Use Spline Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_spline(HD88661(:,2),HD88661(:,1));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Try non-uniform built in function "nufft"

```
N = length(HD88661(:,2));
F = nufft( HD88661(:,2),HD88661(:,1) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0125008
```

```
key_freq =
   0.012500800000000
```

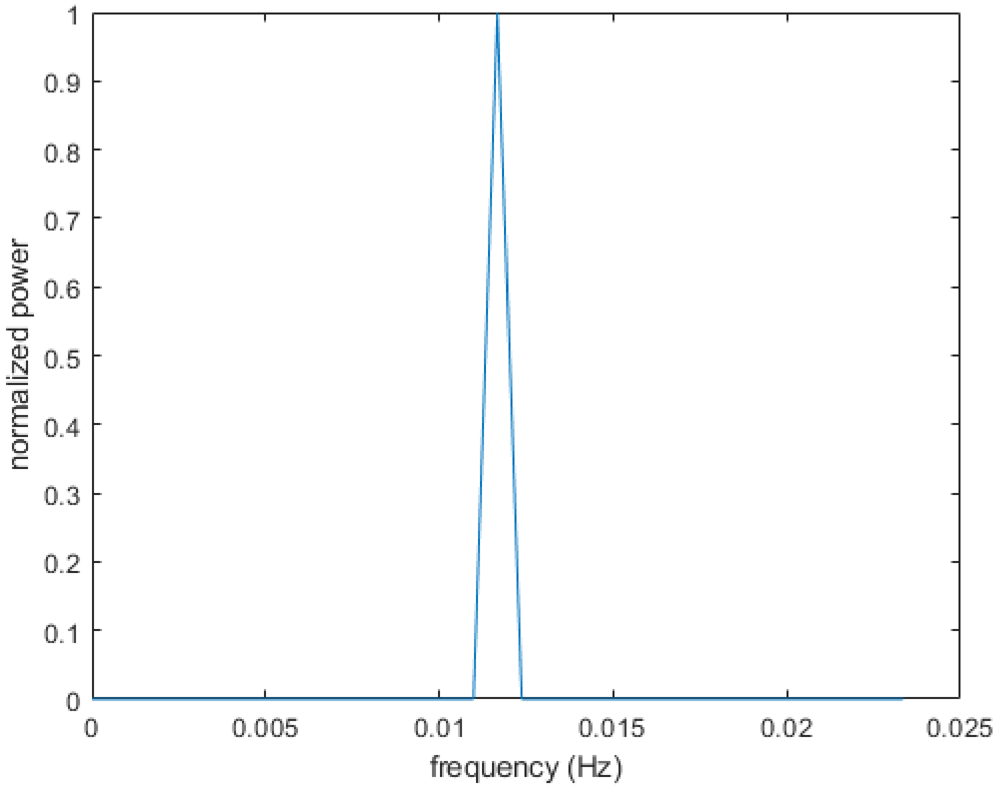
```
avg_freq =1/( mean( diff(HD88661(:,2)) ) *86400 )
```

```
avg_freq =
   0.025736986163581
```

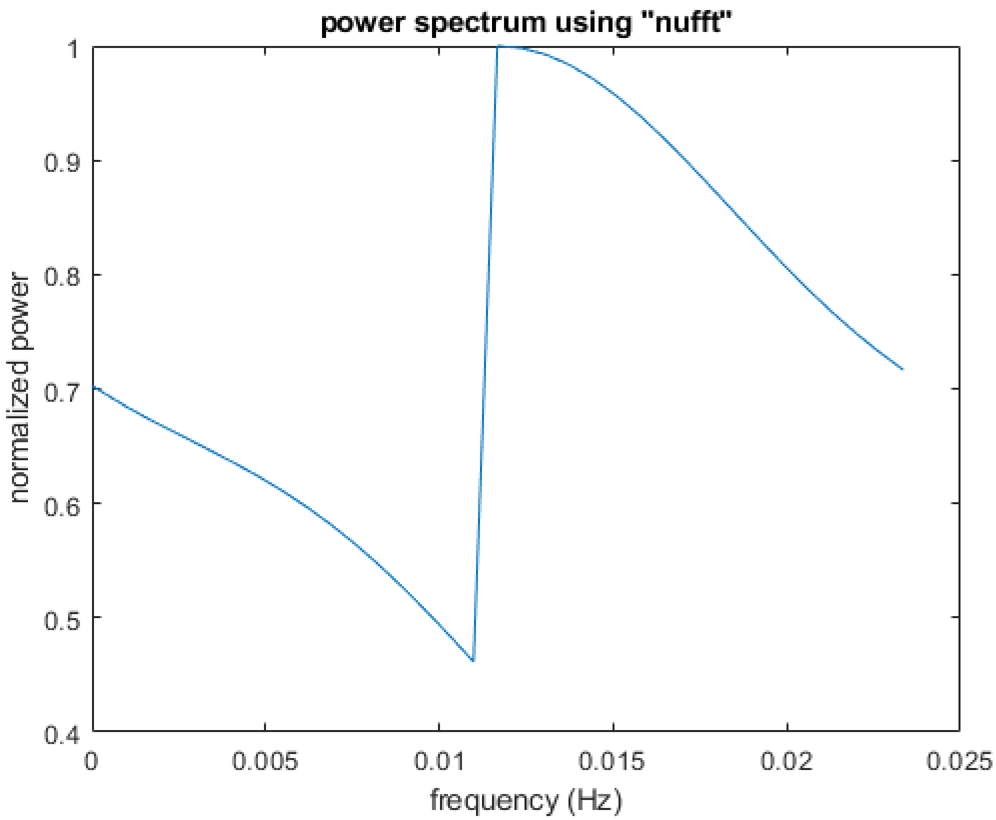
```
avg_freq/key_freq
```

```
ans =
   2.058827128150297
```

```
%R
avg_freq =1/( mean( diff(HD88661(:,4)) ) *86400 );
[Tnew,Mnew] = Interp_Lin(HD88661(:,4),HD88661(:,3));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```




```
N = length(HD88661(:,4));
F = nufft( HD88661(:,4),HD88661(:,3) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0116761
```

```
key_freq =
   0.011676100000000
```

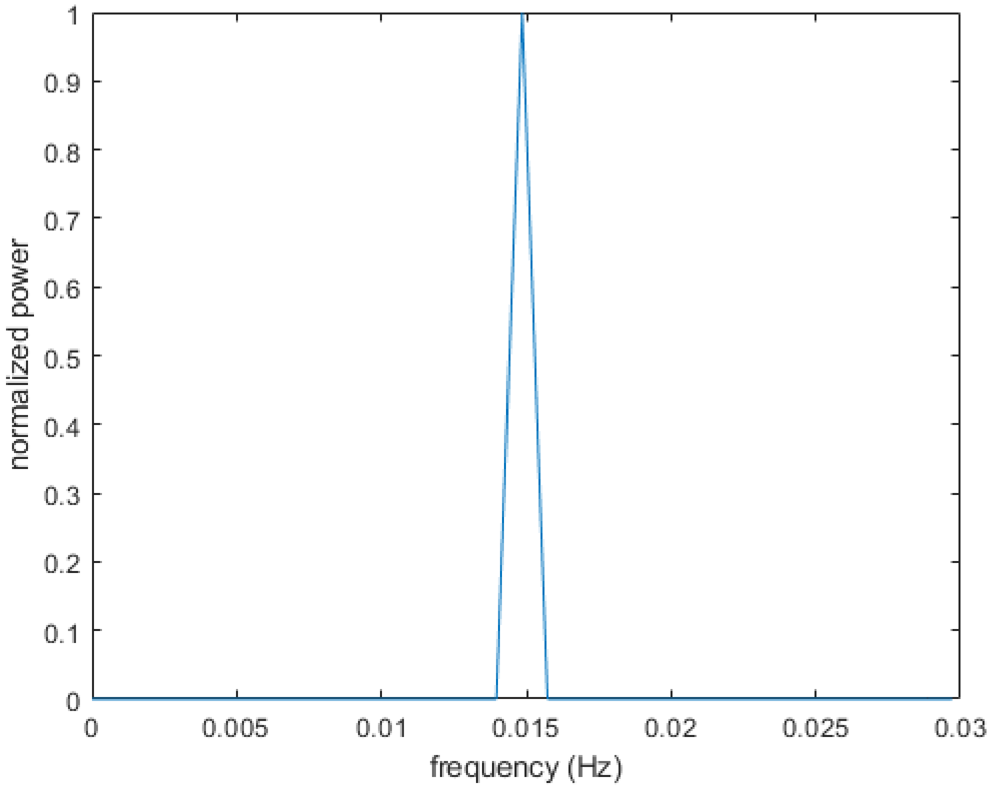
```
avg_freq =1/( mean( diff(HD88661(:,4)) ) *86400 )
```

```
avg_freq =
   0.024039005400274
```

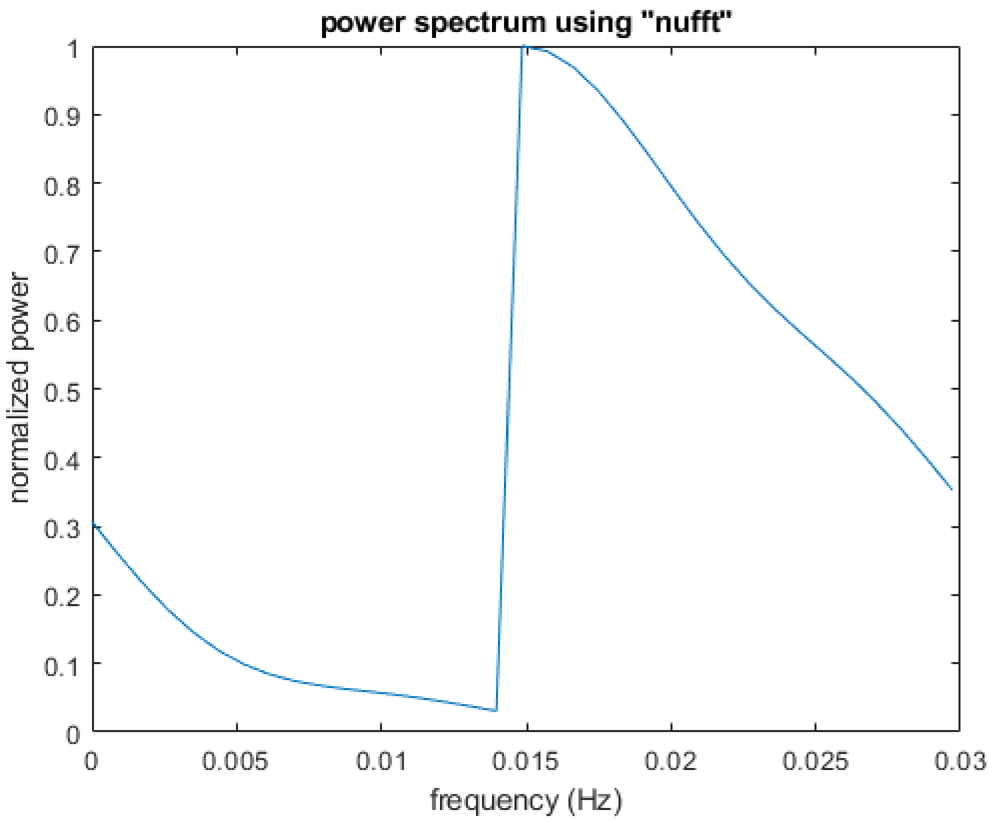
```
avg_freq/key_freq
```

```
ans =
   2.058821472946825
```

```
%V
avg_freq =1/( mean( diff(HD88661(:,6)) ) *86400 );
[Tnew,Mnew] = Interp_Lin(HD88661(:,6),HD88661(:,5));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```




```
N = length(HD88661(:,6));
F = nufft( HD88661(:,6),HD88661(:,5) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0148514;
avg_freq =1/( mean( diff(HD88661(:,6)) ) *86400 )
```

```
avg_freq =
   0.030576419468500
```

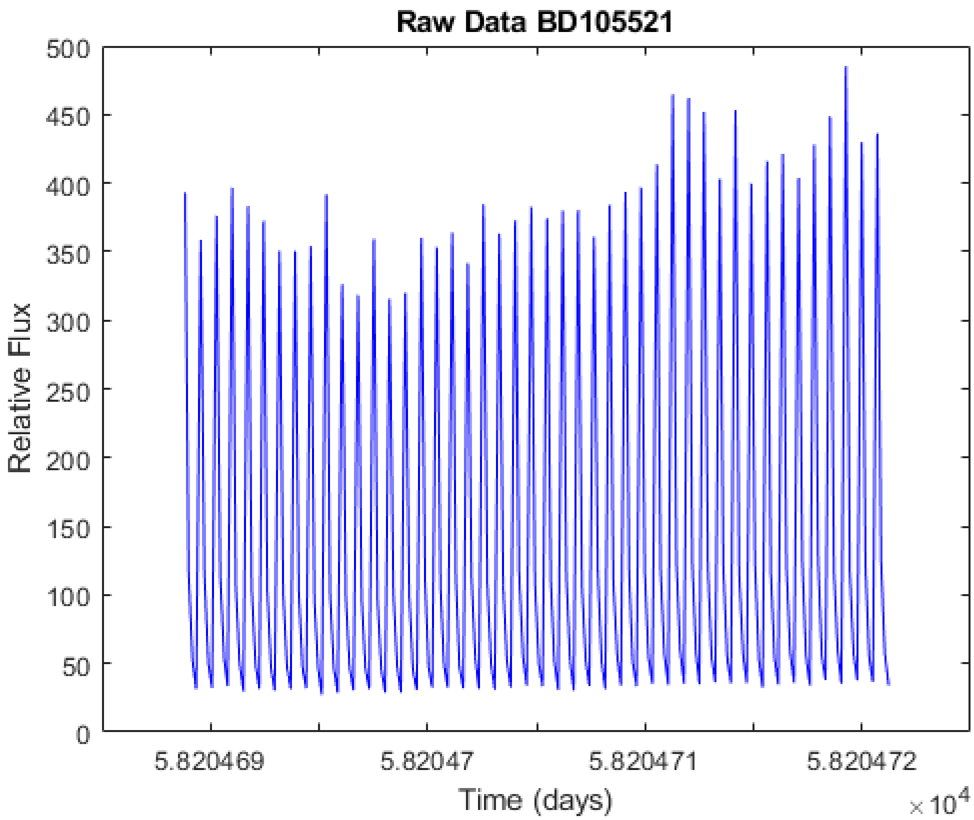
```
avg_freq/key_freq
```

```
ans =
   2.058824048136881
```

HD105521

Plot the raw data

```
plot(HD105521(:,1),HD105521(:,2),'b');
title('Raw Data BD105521');
xlabel('Time (days)');
ylabel('Relative Flux');
```

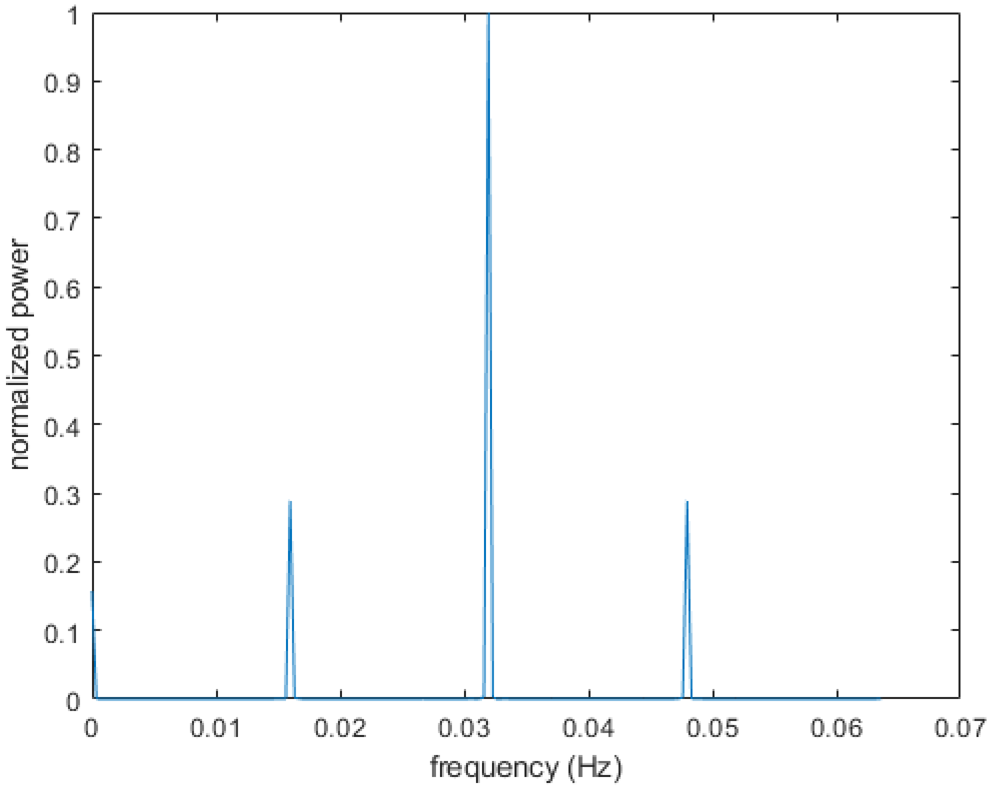



```
%B
avg_freq =1/( mean( diff(HD105521(:,1)) ) *86400 )
```

```
avg_freq =
   0.063943187006892
```

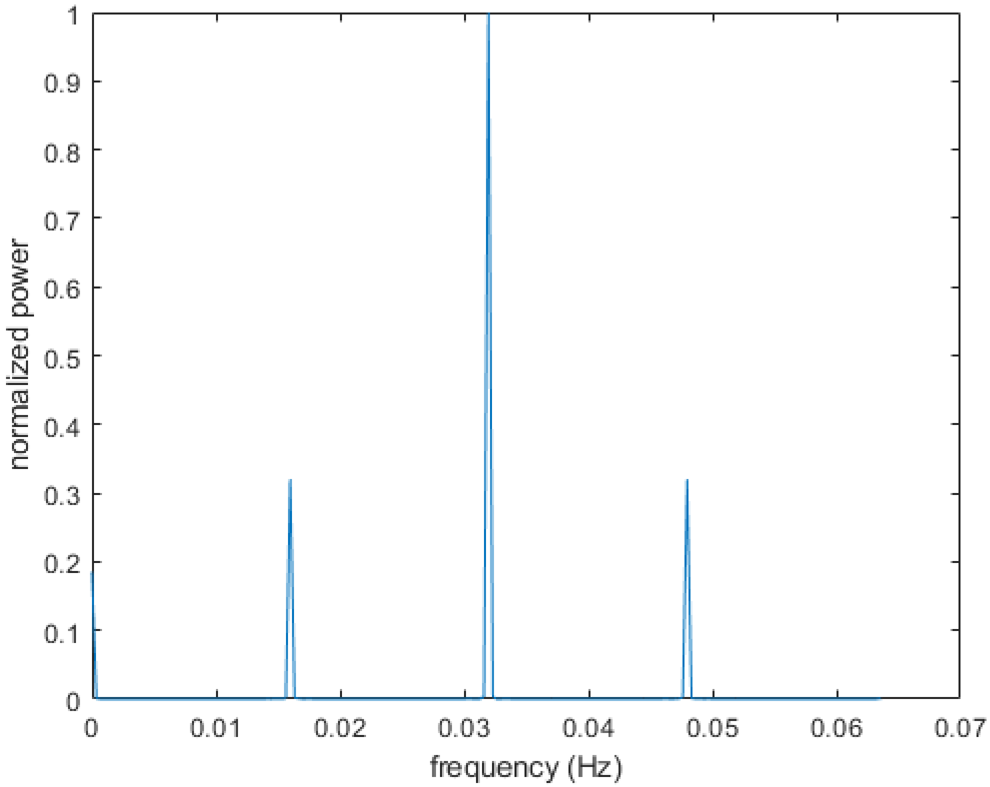
Use Linear Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_Lin(HD105521(:,1),HD105521(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



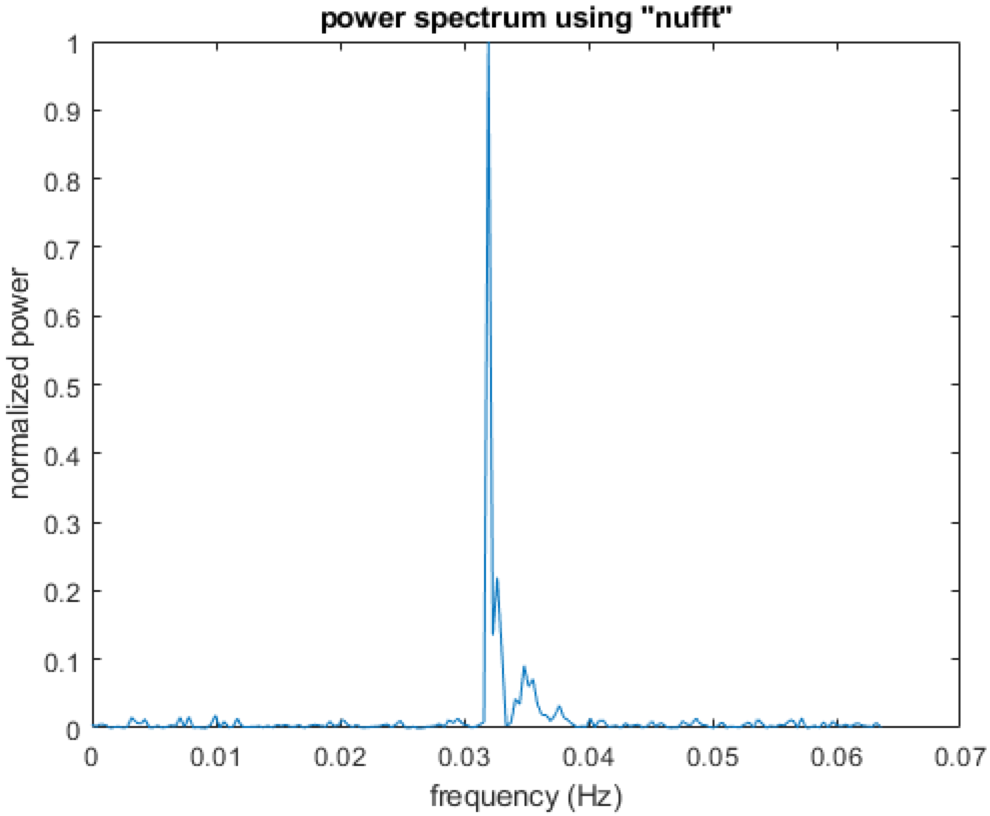
Use Spline Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_spline(HD105521(:,1),HD105521(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Try non-uniform built in function "nufft"

```
N = length(HD105521(:,1));
F = nufft( HD105521(:,1),HD105521(:,2) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0319716
```

```
key_freq =
   0.031971600000000
```

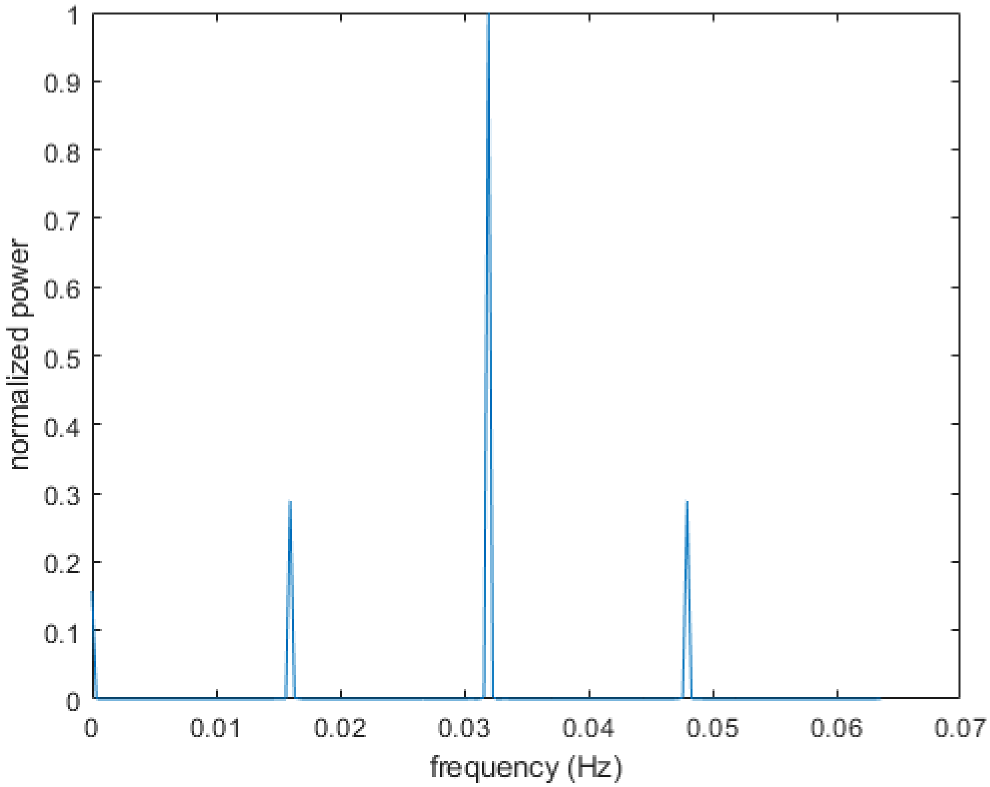
```
avg_freq =1/( mean( diff(HD105521(:,1)) ) *86400 )
```

```
avg_freq =
   0.063943187006892
```

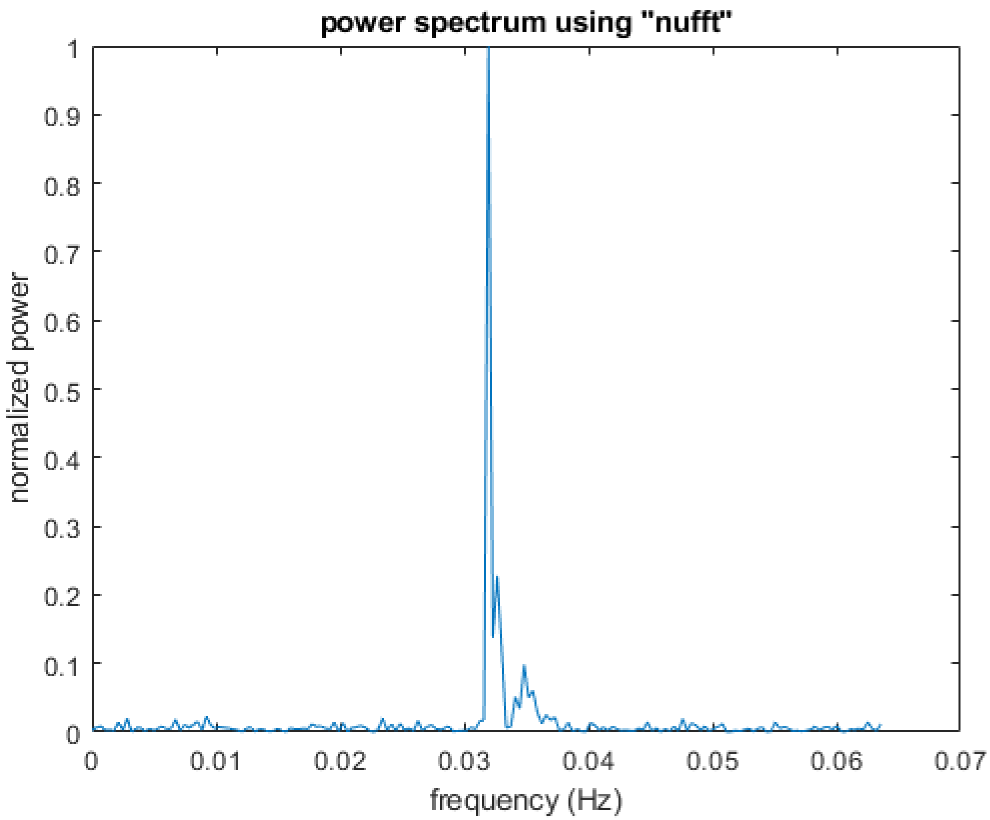
```
avg_freq/key_freq
```

```
ans =
   1.999999593604684
```

```
%R
avg_freq =1/( mean( diff(HD105521(:,3)) ) *86400 );
[Tnew,Mnew] = Interp_Lin(HD105521(:,3),HD105521(:,4));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```




```
N = length(HD105521(:,3));
F = nufft( HD105521(:,3),HD105521(:,4) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0319716
```

```
key_freq =
   0.031971600000000
```

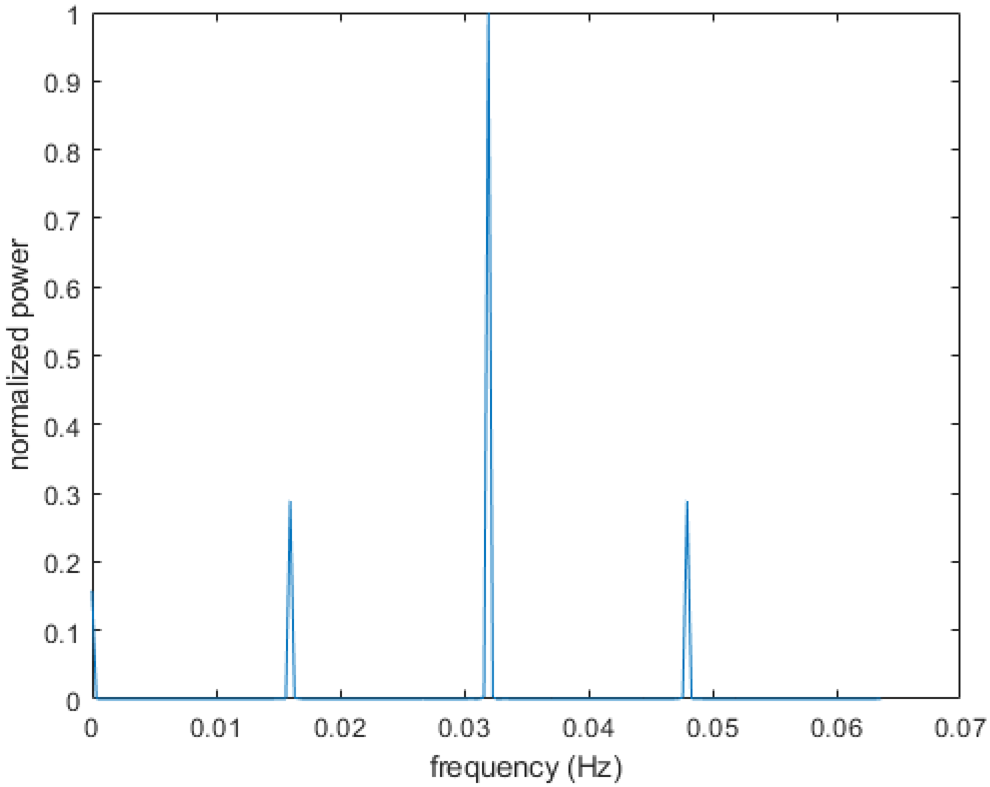
```
avg_freq =1/( mean( diff(HD105521(:,3)) ) *86400 )
```

```
avg_freq =
   0.063943187006892
```

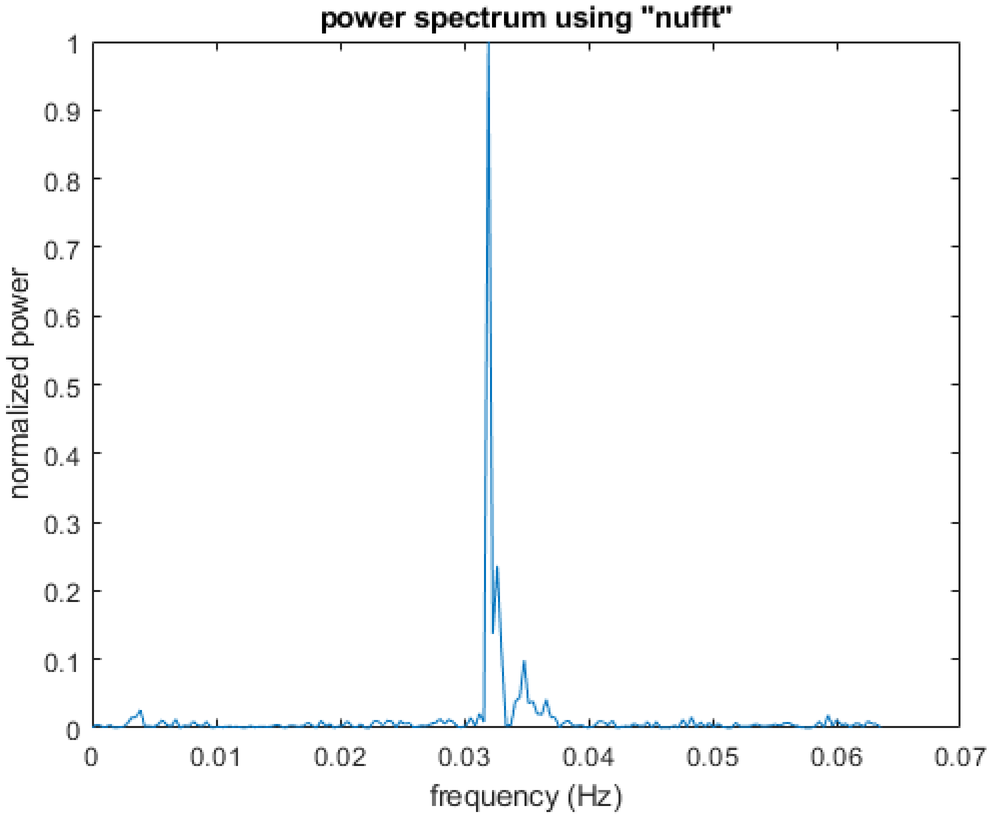
```
avg_freq/key_freq
```

```
ans =
   1.999999593604684
```

```
%V
avg_freq =1/( mean( diff(HD105521(:,5)) ) *86400 );
[Tnew,Mnew] = Interp_Lin(HD105521(:,5),HD105521(:,6));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```




```
N = length(HD105521(:,5));
F = nufft( HD105521(:,5),HD105521(:,6) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0319716
```

```
key_freq =
   0.031971600000000
```

```
avg_freq =1/( mean( diff(HD105521(:,5)) ) *86400 )
```

```
avg_freq =
   0.063943187006892
```

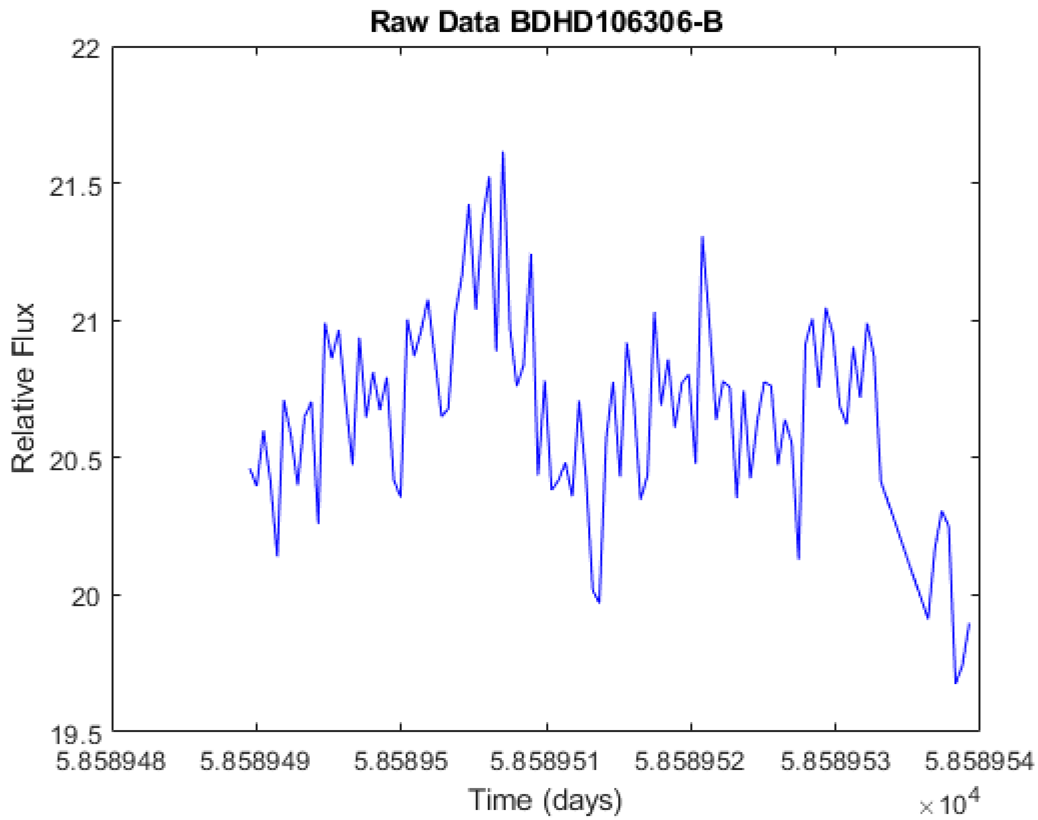
```
avg_freq/key_freq
```

```
ans =
   1.999999593604684
```

HD106306 B

Plot the raw data

```
plot(HD106306_B(:,1),HD106306_B(:,2),'b');
title('Raw Data BDHD106306-B');
xlabel('Time (days)');
ylabel('Relative Flux');
```

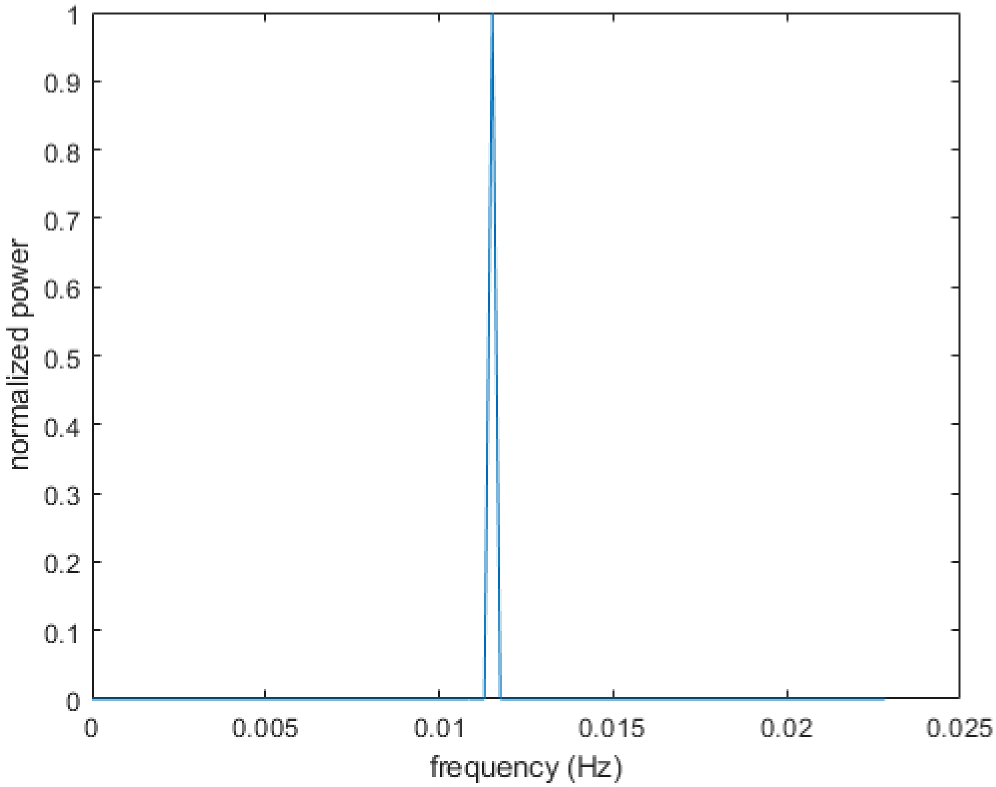



```
avg_freq =1/( mean( diff(HD106306_B(:,1)) ) *86400 )
```

```
avg_freq =
   0.023064278044436
```

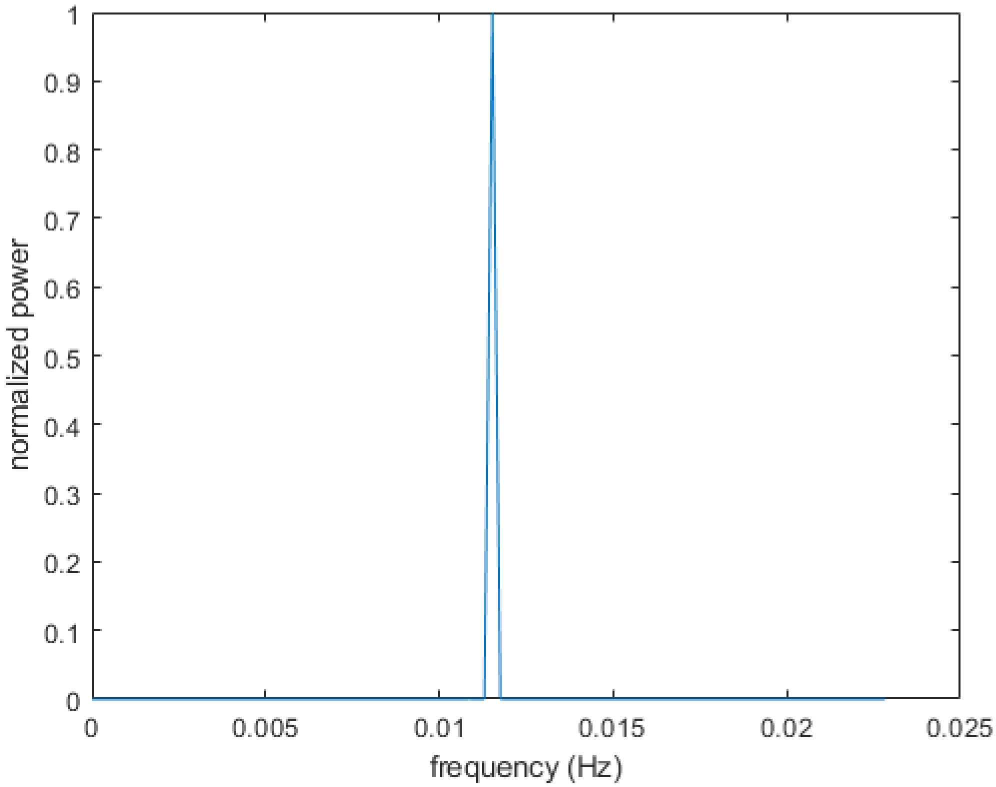
Use Linear Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_Lin(HD106306_B(:,1),HD106306_B(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



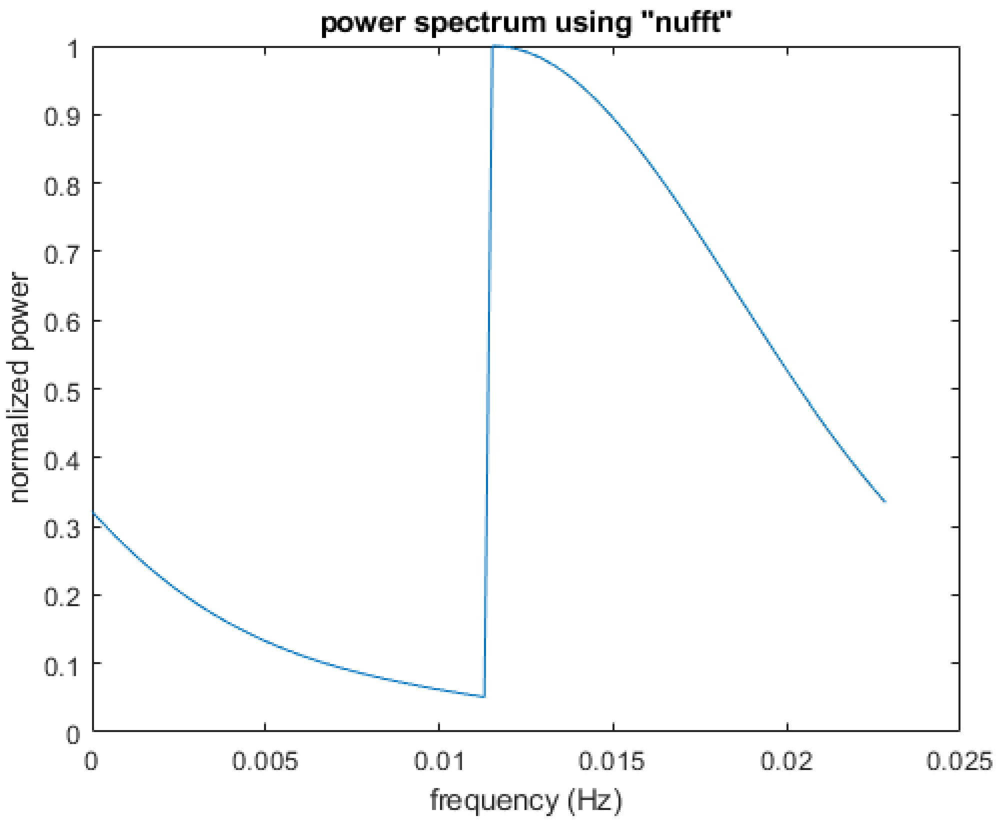
Use Spline Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_spline(HD106306_B(:,1),HD106306_B(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Try non-uniform built in function "nufft"

```
N = length(HD106306_B(:,1));
F = nufft( HD106306_B(:,1),HD106306_B(:,2) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0115321
```

```
key_freq =
   0.011532100000000
```

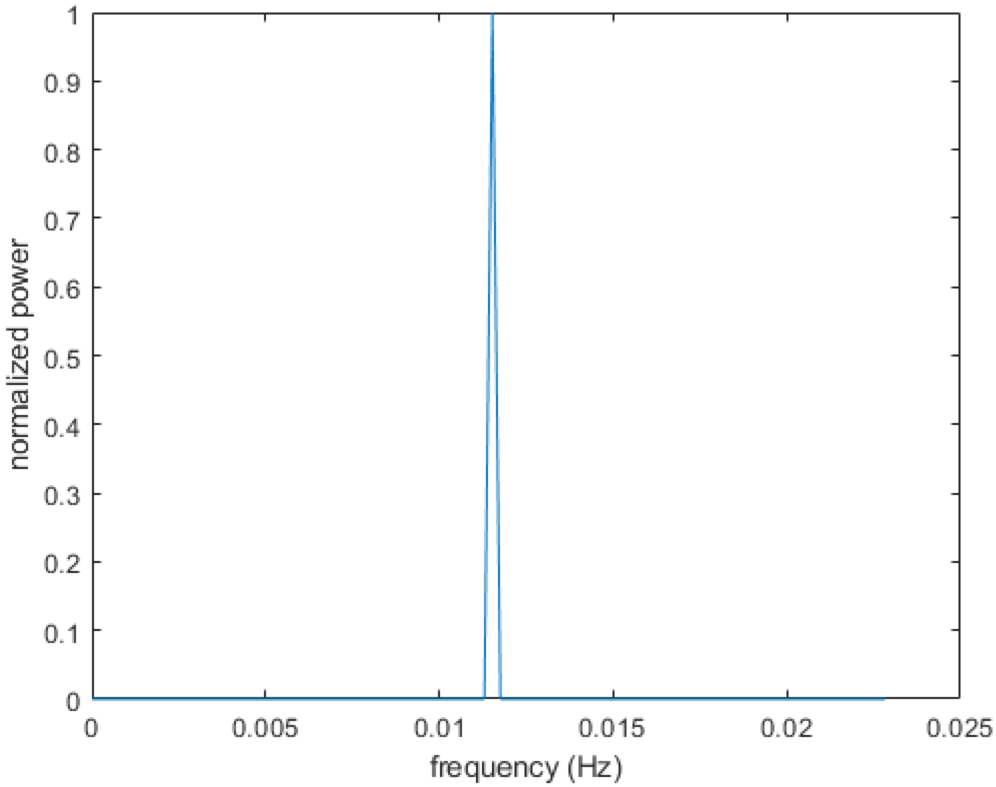
```
avg_freq =1/( mean( diff(HD106306_B(:,1)) ) *86400 )
```

```
avg_freq =
   0.023064278044436
```

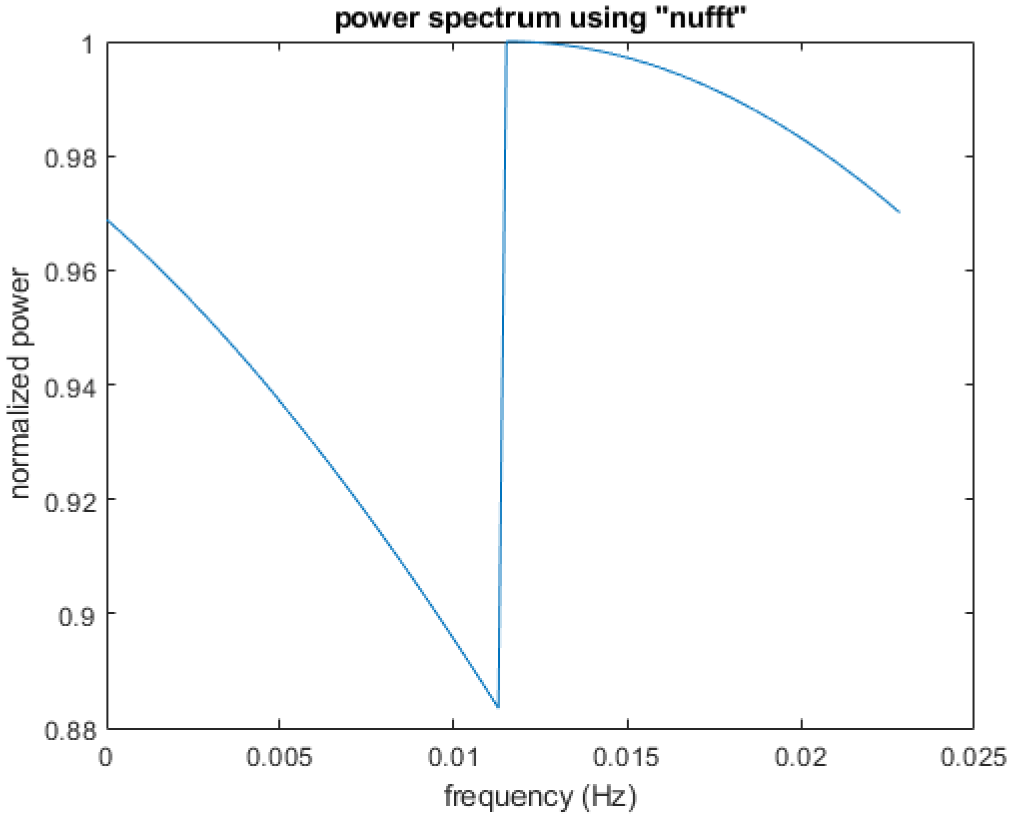
```
avg_freq/key_freq
```

```
ans =
   2.000006767582301
```

```
%R
avg_freq =1/( mean( diff(HD106306_R(:,1)) ) *86400 );
[Tnew,Mnew] = Interp_Lin(HD106306_R(:,1),HD106306_R(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```




```
N = length(HD106306_R(:,1));
F = nufft( HD106306_R(:,1),HD106306_R(:,2) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0115326
```

```
key_freq =
   0.011532600000000
```

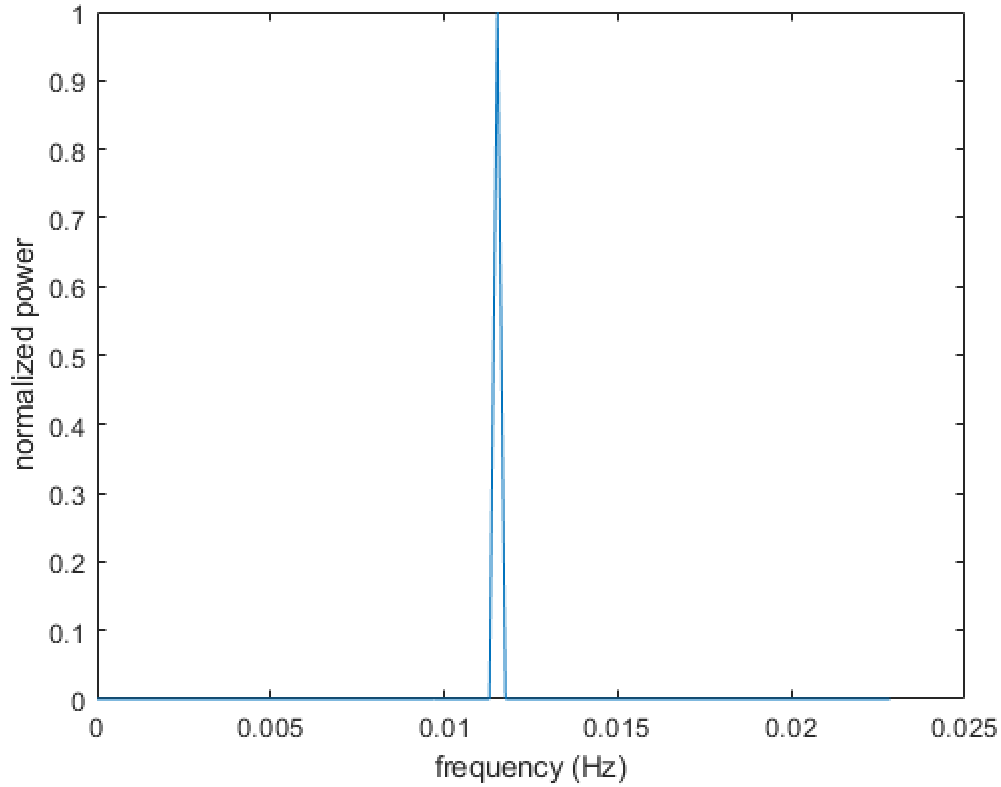
```
avg_freq =1/( mean( diff(HD106306_R(:,1)) ) *86400 )
```

```
avg_freq =
   0.023065206595740
```

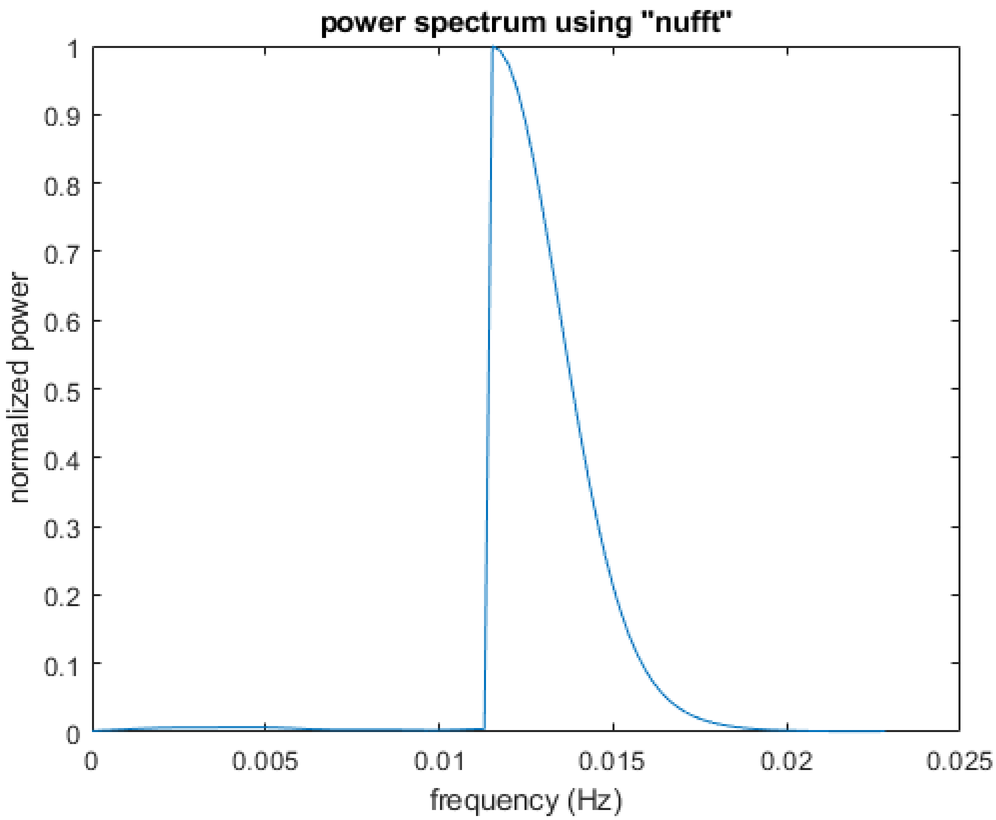
```
avg_freq/key_freq
```

```
ans =
   2.000000571921362
```

```
%V
avg_freq =1/( mean( diff(HD106306_V(:,1)) ) *86400 );
[Tnew,Mnew] = Interp_Lin(HD106306_V(:,1),HD106306_V(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```




```
N = length(HD106306_V(:,1));
F = nufft( HD106306_V(:,1),HD106306_V(:,2) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0115335
```

```
key_freq =
   0.011533500000000
```

```
avg_freq =1/( mean( diff(HD106306_V(:,1)) ) *86400 )
```

```
avg_freq =
   0.023067063922664
```

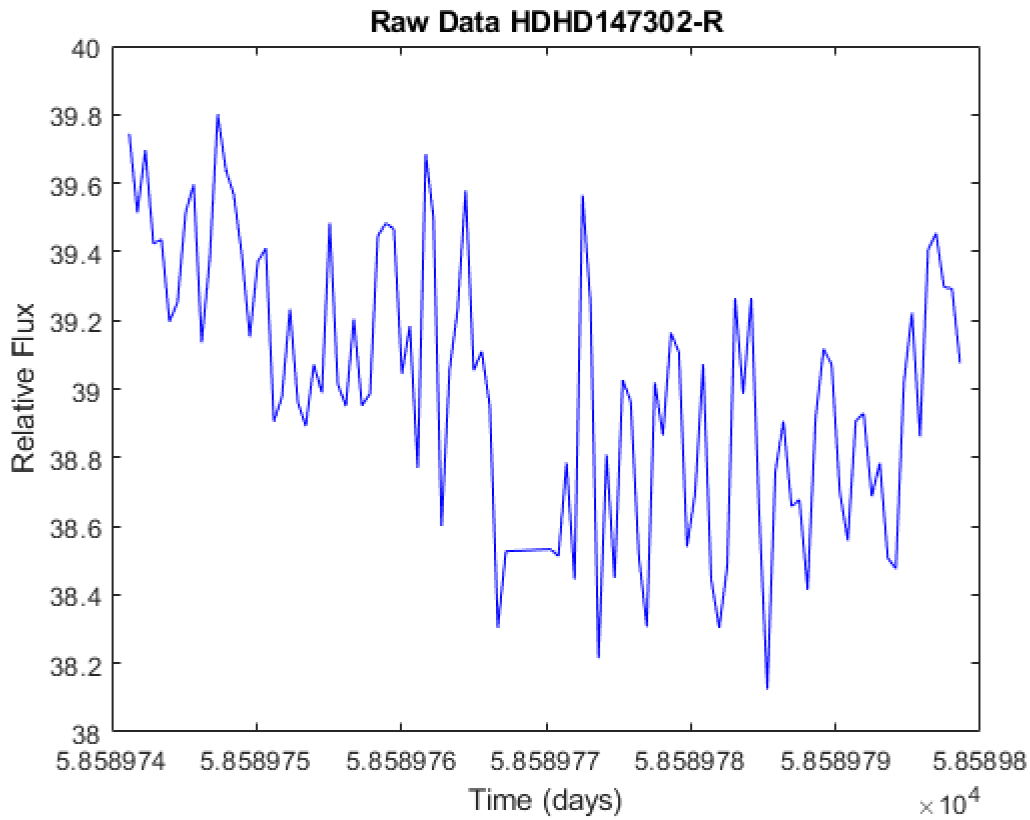
```
avg_freq/key_freq
```

```
ans =
   2.000005542347433
```

HD147302_R

Plot the raw data

```
plot(HD147302_R(:,1),HD147302_R(:,2),'b');
title('Raw Data HDHD147302-R');
xlabel('Time (days)');
ylabel('Relative Flux');
```

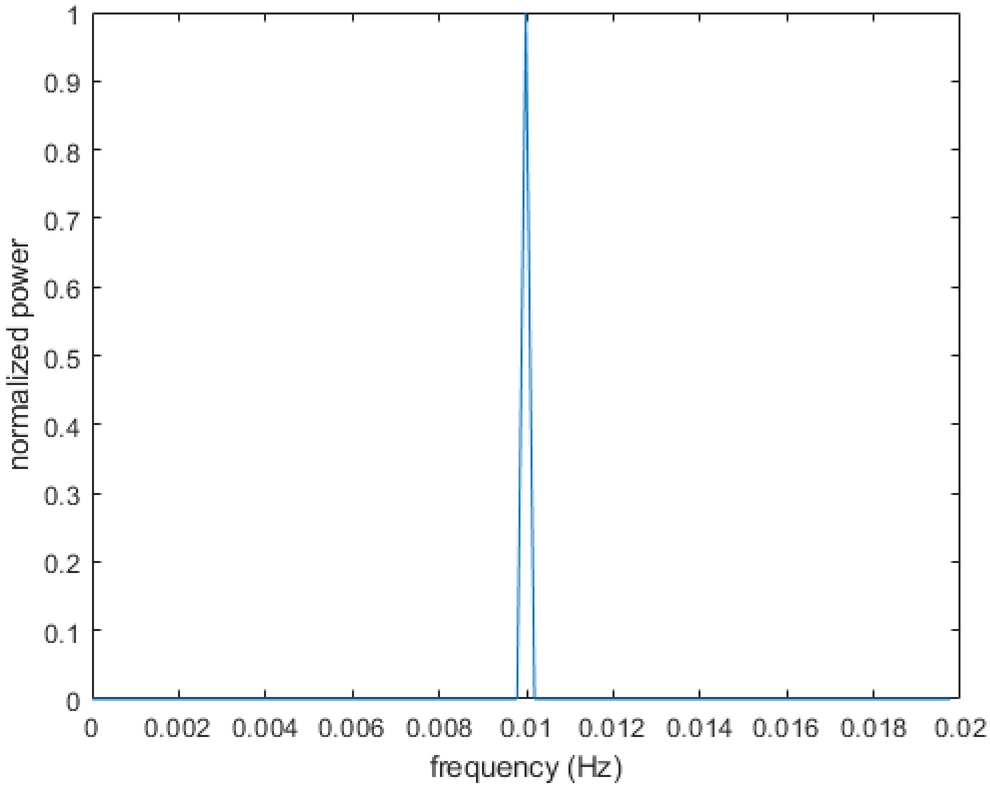



```
avg_freq =1/( mean( diff(HD147302_R(:,1)) ) *86400 )
```

```
avg_freq =
   0.019970951344655
```

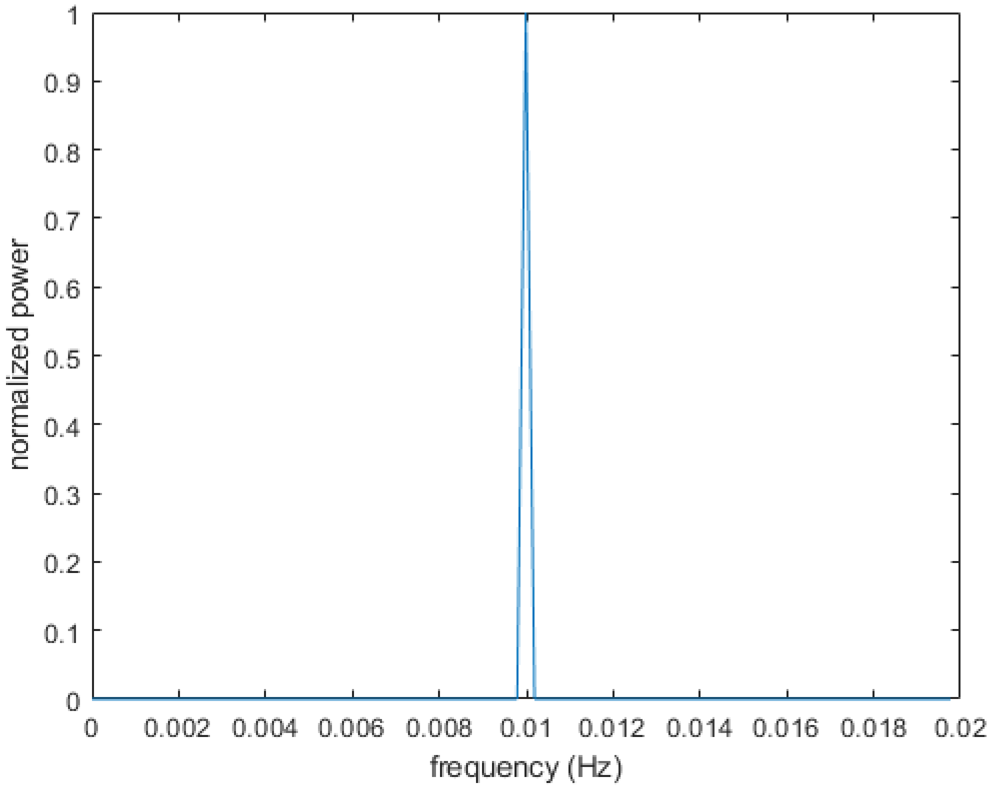
Use Linear Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_Lin(HD147302_R(:,1),HD147302_R(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



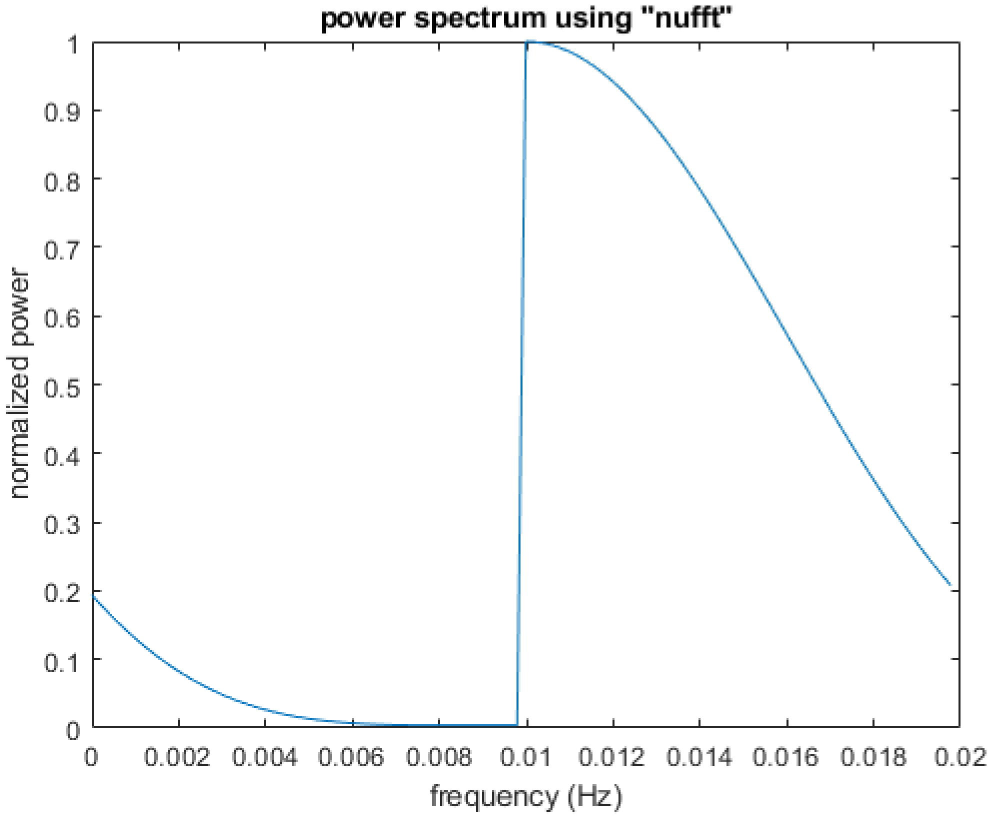
Use Spline Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_spline(HD147302_R(:,1),HD147302_R(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Try non-uniform built in function "nufft"

```
N = length(HD147302_R(:,1));
F = nufft( HD147302_R(:,1),HD147302_R(:,2) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.00998548
```

```
key_freq =
   0.009985480000000
```

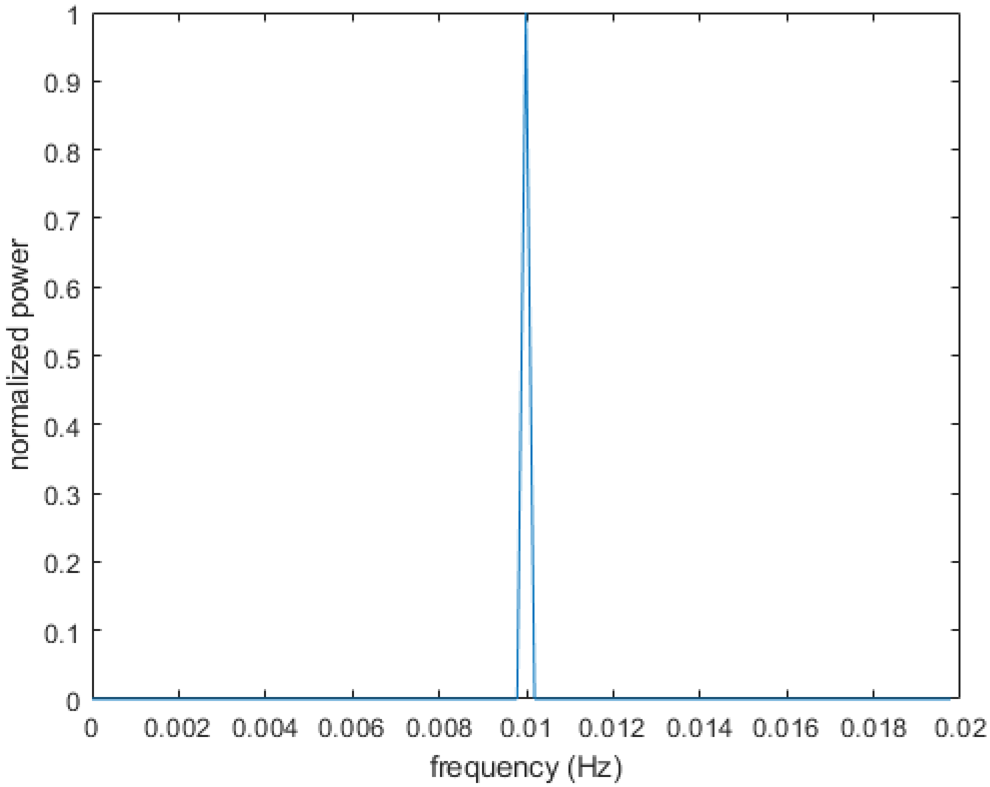
```
avg_freq =1/( mean( diff(HD147302_R(:,1)) ) *86400 )
```

```
avg_freq =
   0.019970951344655
```

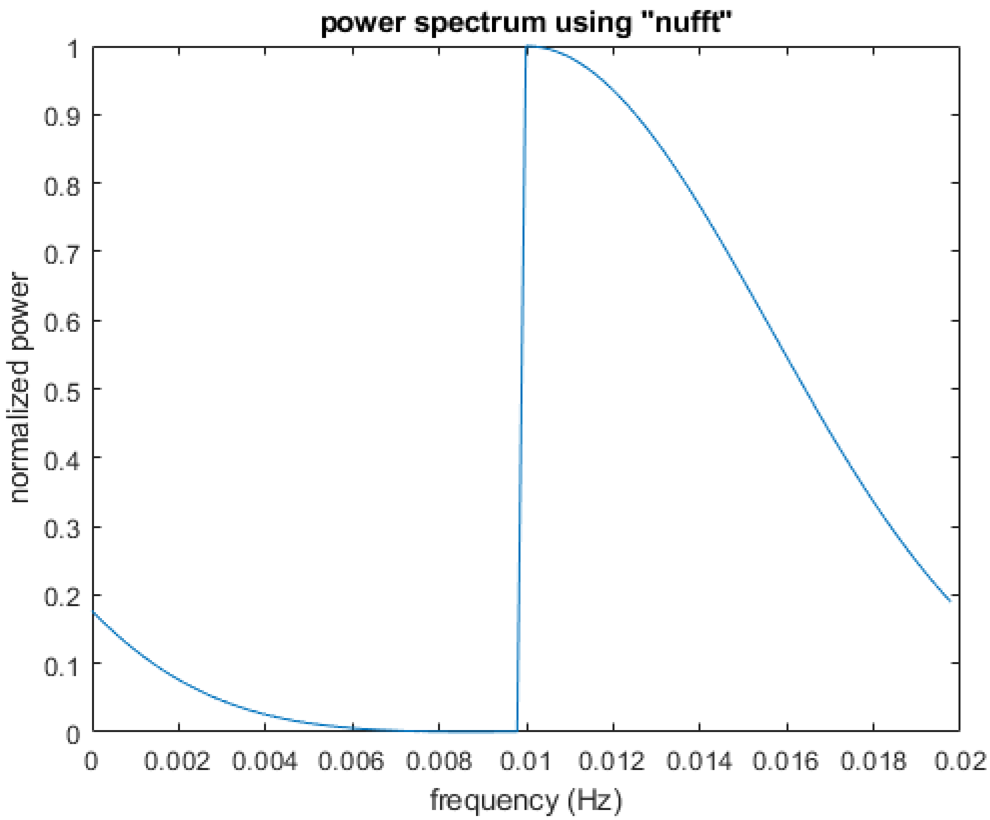
```
avg_freq/key_freq
```

```
ans =
   1.999999133206939
```

```
%V
avg_freq =1/( mean( diff(HD147302_V(:,1)) ) *86400 );
[Tnew,Mnew] = Interp_Lin(HD147302_V(:,1),HD147302_V(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```




```
N = length(HD147302_V(:,1));
F = nufft( HD147302_V(:,1),HD147302_V(:,2) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.00998635
```

```
key_freq =
   0.009986350000000
```

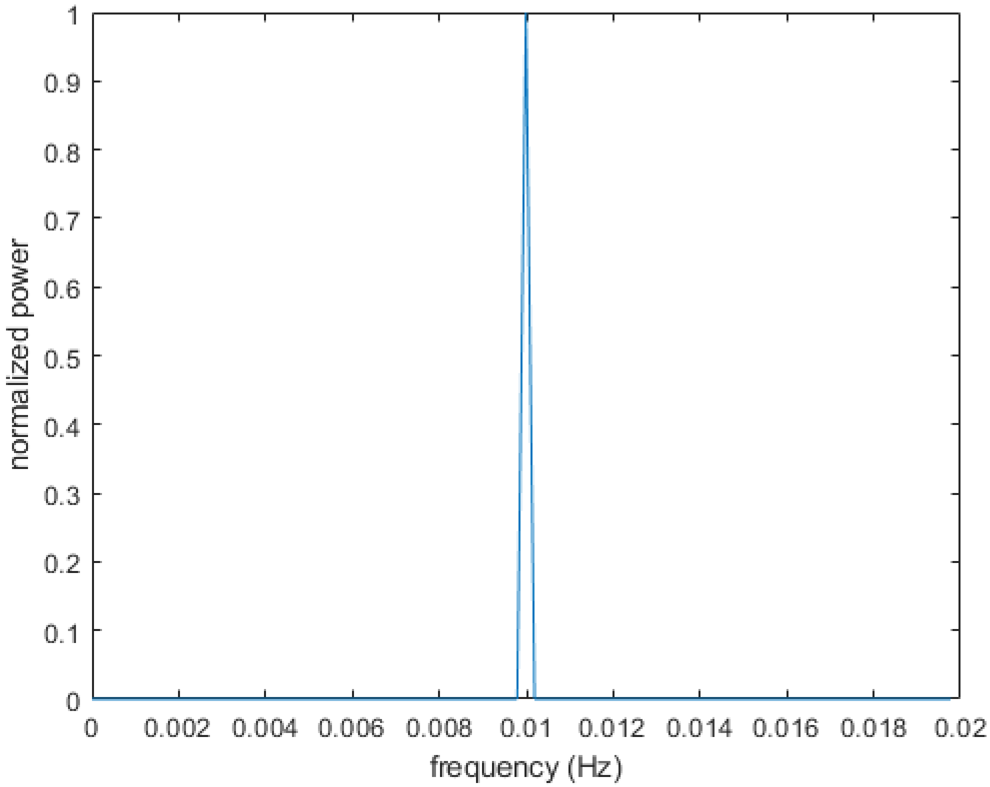
```
avg_freq =1/( mean( diff(HD147302_V(:,1)) ) *86400 )
```

```
avg_freq =
   0.019972691882309
```

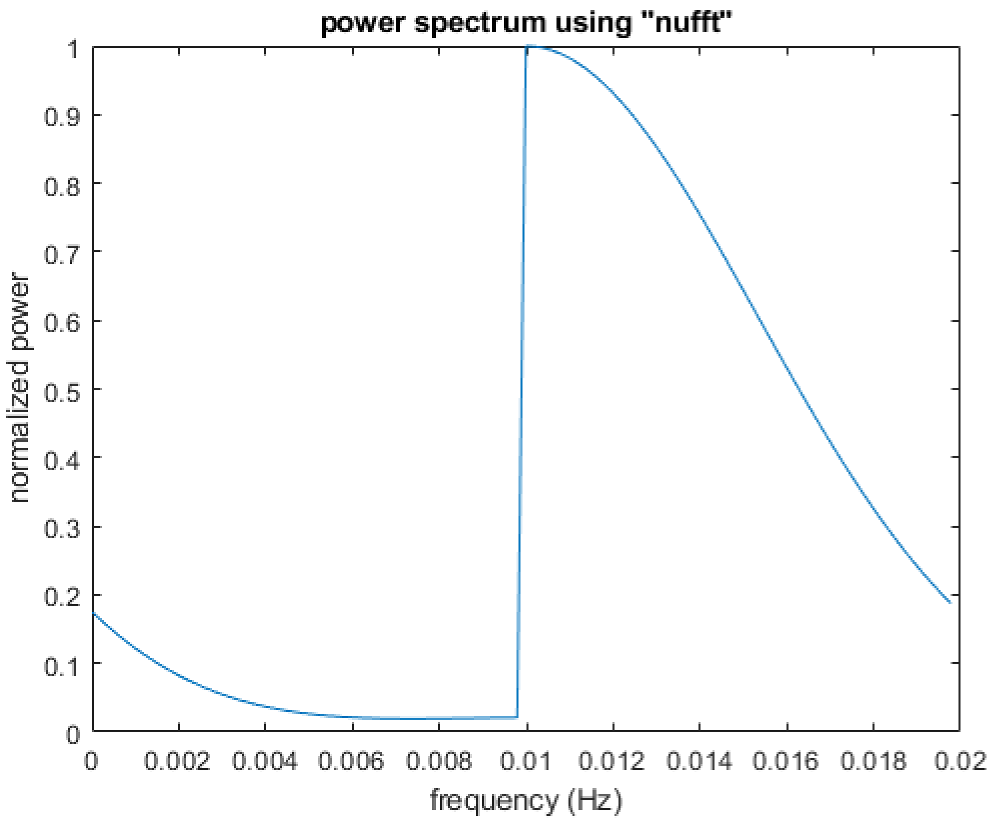
```
avg_freq/key_freq
```

```
ans =
   1.999999187121354
```

```
%B
avg_freq =1/( mean( diff(HD147302_B(:,1)) ) *86400 );
[Tnew,Mnew] = Interp_Lin(HD147302_B(:,1),HD147302_B(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```




```
N = length(HD147302_B(:,1));
F = nufft( HD147302_B(:,1),HD147302_B(:,2) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.00998669
```

```
key_freq =
   0.009986690000000
```

```
avg_freq =1/( mean( diff(HD147302_B(:,1)) ) *86400 )
```

```
avg_freq =
   0.019973388183336
```

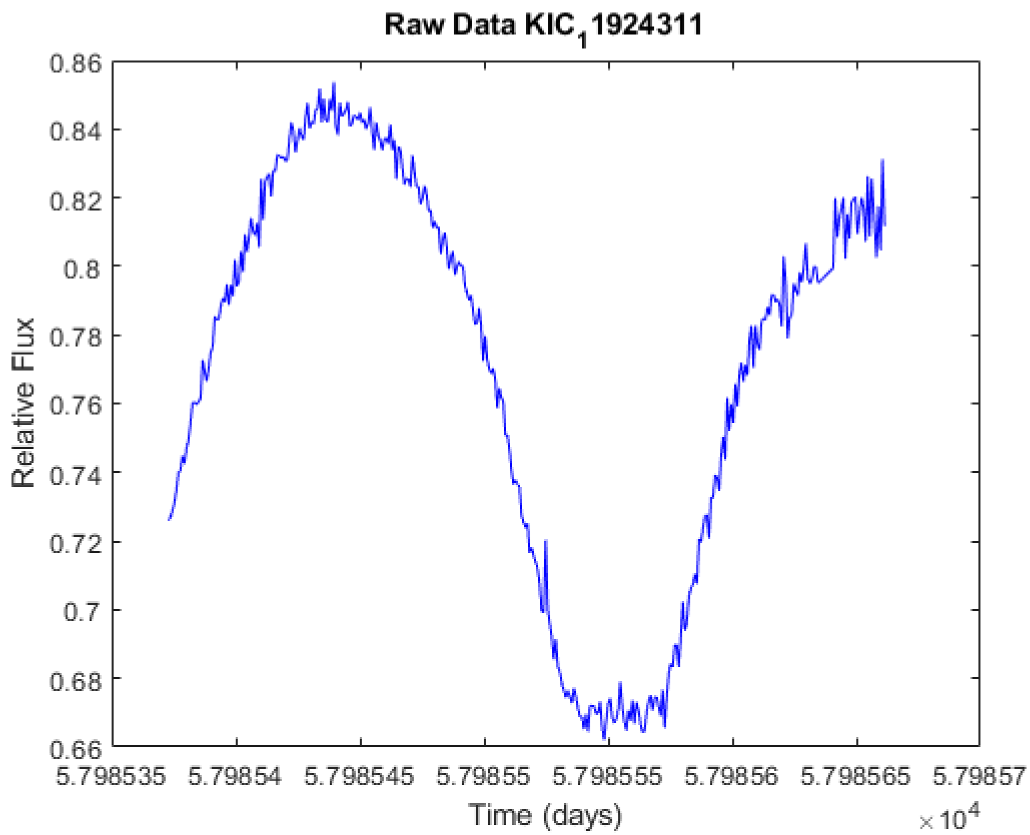
```
avg_freq/key_freq
```

```
ans =
   2.000000819424234
```

KIC_11924311

Plot the raw data

```
plot(KIC_11924311.V_time,KIC_11924311.V_flux,'b');
title('Raw Data KIC_11924311');
xlabel('Time (days)');
ylabel('Relative Flux');
```

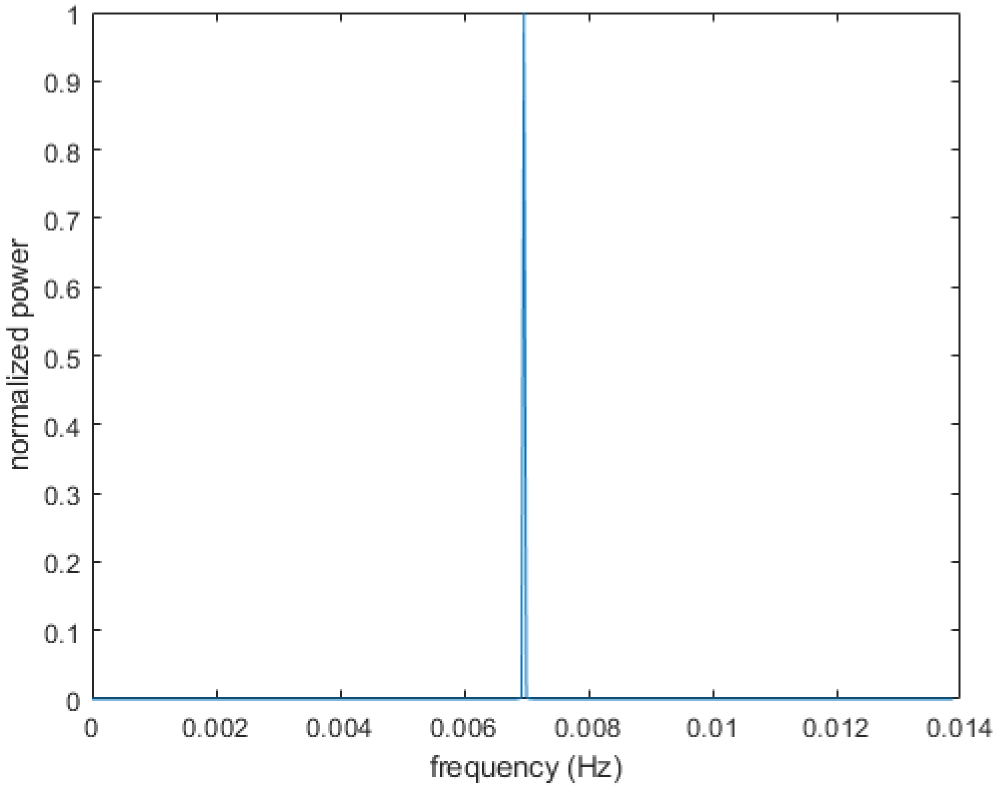



```
avg_freq =1/( mean( diff(KIC_11924311.V_time) ) *86400 )
```

```
avg_freq =
   0.013917606486280
```

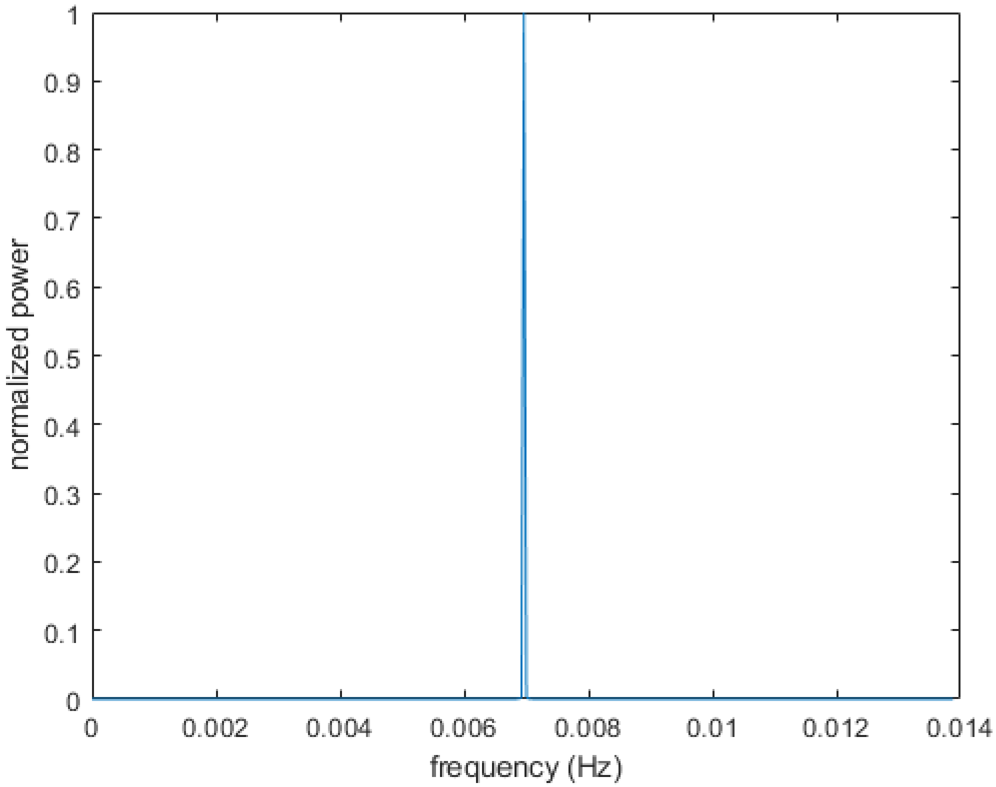
Use Linear Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_Lin(KIC_11924311.V_time,KIC_11924311.V_flux);
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



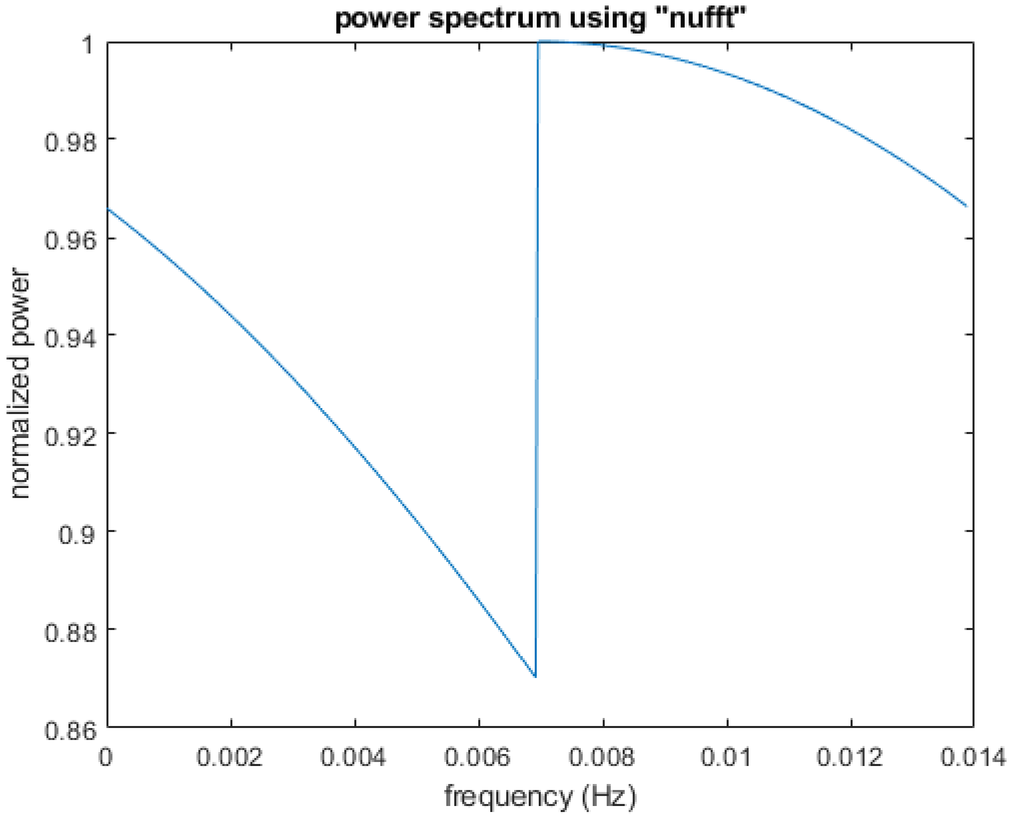
Use Spline Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_spline(KIC_11924311.V_time,KIC_11924311.V_flux);
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Try non-uniform built in function "nufft"

```
N = length(KIC_11924311.V_time);
F = nufft( KIC_11924311.V_time,KIC_11924311.V_flux )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0069588
```

```
key_freq =
   0.006958800000000
```

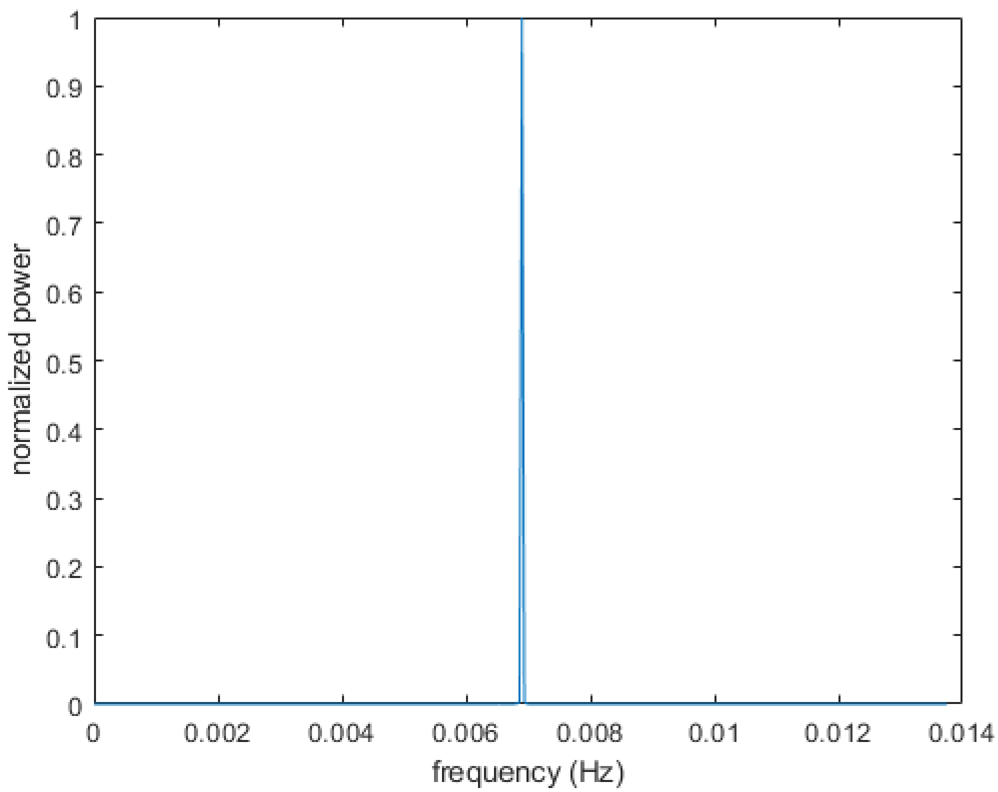
```
avg_freq =1/( mean( diff(KIC_11924311.V_time) ) *86400 )
```

```
avg_freq =
   0.013917606486280
```

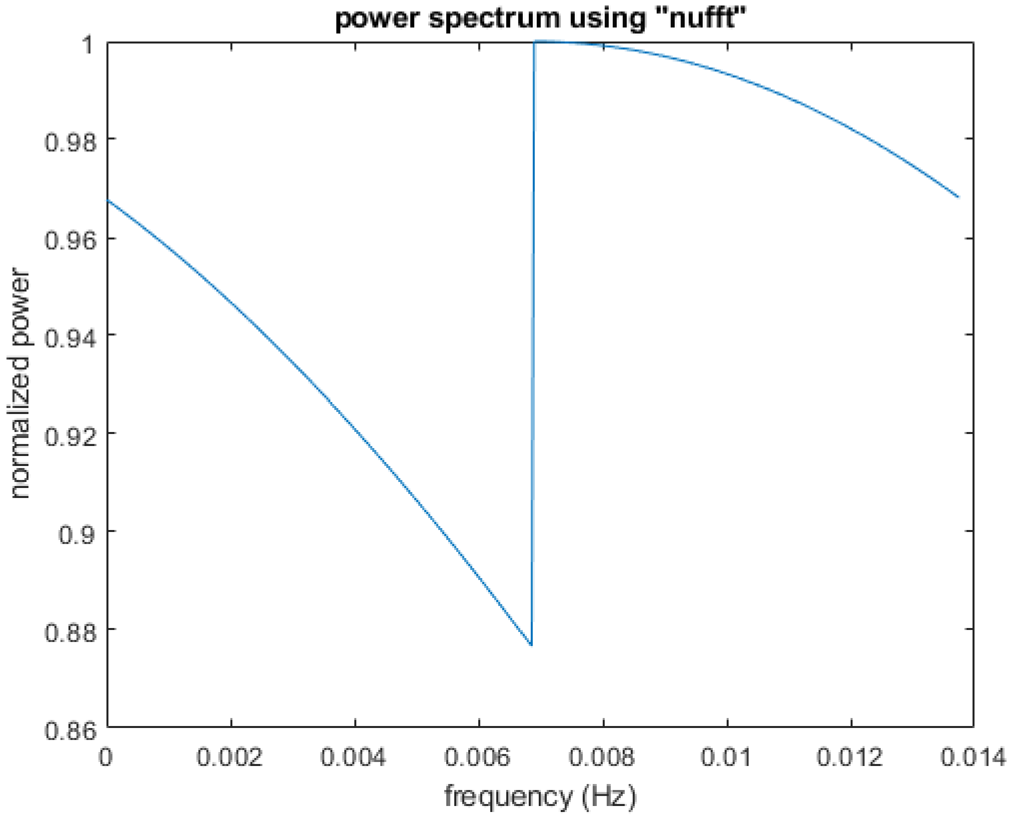
```
avg_freq/key_freq
```

```
ans =
   2.000000932097445
```

```
%B
avg_freq =1/( mean( diff(KIC_11924311.B_time) ) *86400 );
[Tnew,Mnew] = Interp_spline(KIC_11924311.B_time,KIC_11924311.B_flux);
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(KIC_11924311.B_time,KIC_11924311.B_flux,dt);
```




```
N = length(KIC_11924311.B_time);
F = nufft( KIC_11924311.B_time,KIC_11924311.B_flux )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.00689501
```

```
key_freq =
   0.006895010000000
```

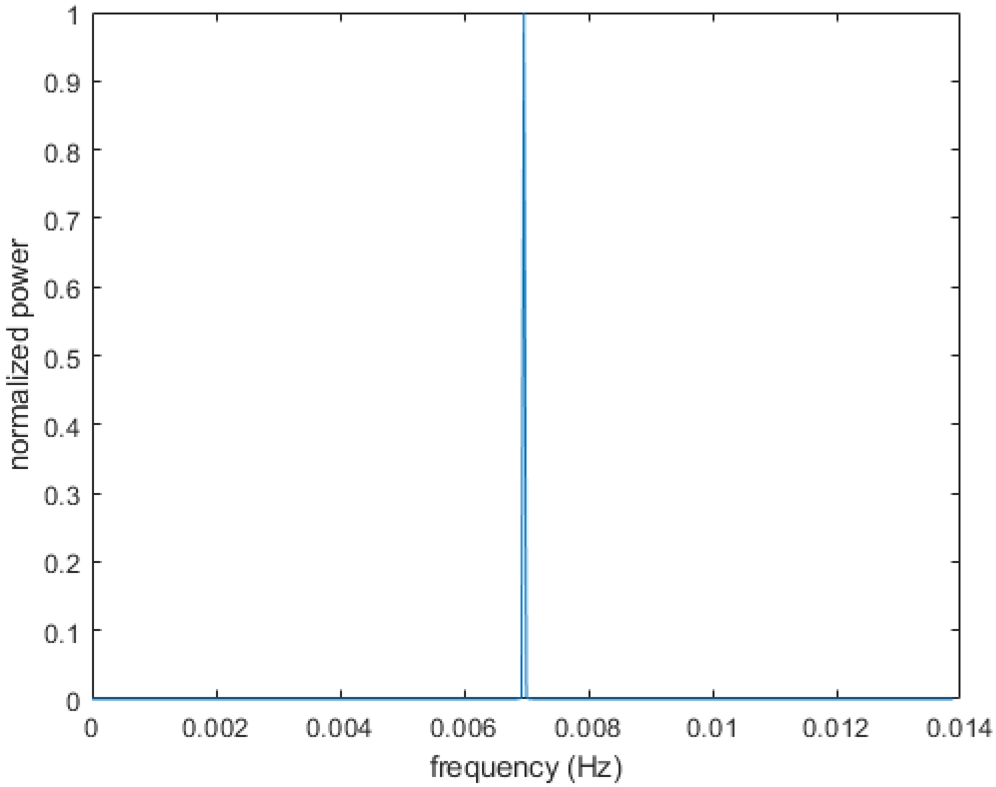
```
avg_freq =1/( mean( diff(KIC_11924311.B_time) ) *86400 )
```

```
avg_freq =
   0.013790014090173
```

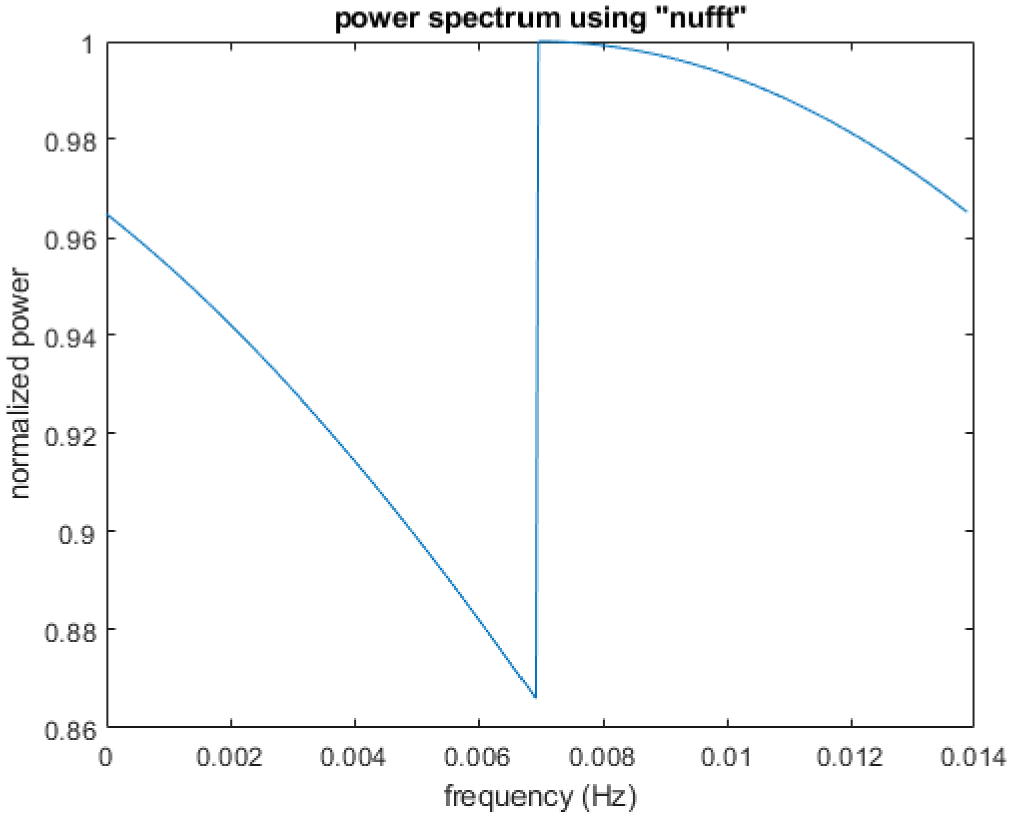
```
avg_freq/key_freq
```

```
ans =
   1.999999142883464
```

```
%R
avg_freq =1/( mean( diff(KIC_11924311.R_time) ) *86400 );
[Tnew,Mnew] = Interp_spline(KIC_11924311.R_time,KIC_11924311.R_flux);
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(KIC_11924311.R_time,KIC_11924311.R_flux,dt);
```




```
N = length(KIC_11924311.R_time);
F = nufft( KIC_11924311.R_time,KIC_11924311.R_flux )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0069588
```

```
key_freq =
   0.006958800000000
```

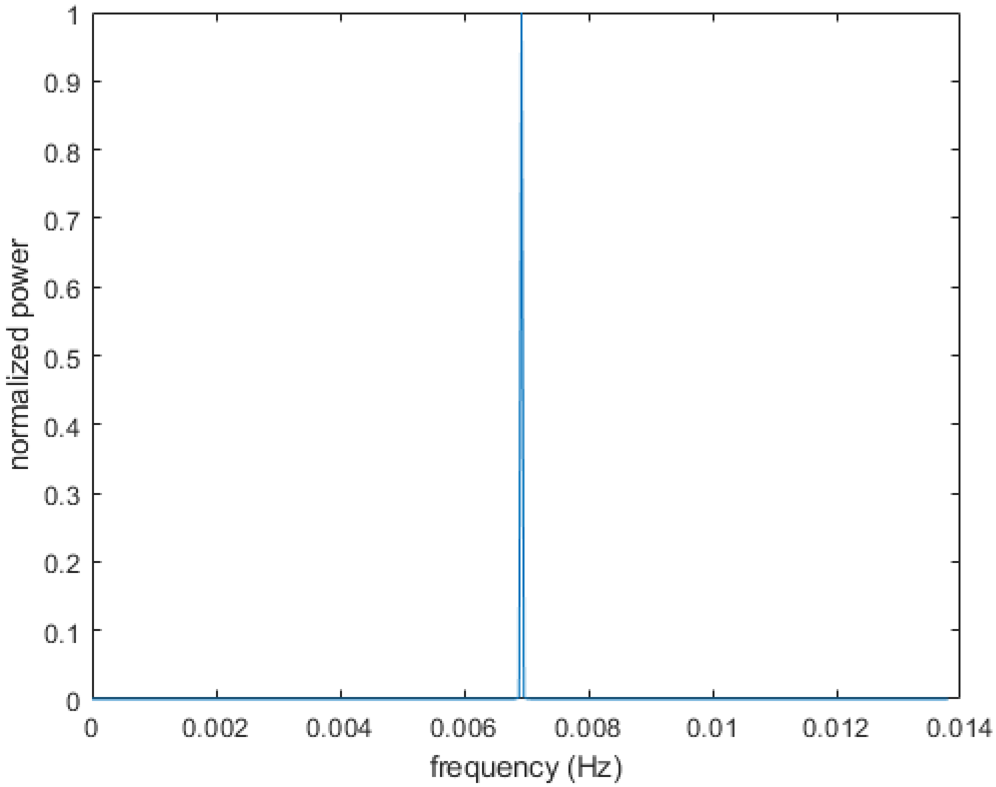
```
avg_freq =1/( mean( diff(KIC_11924311.R_time) ) *86400 )
```

```
avg_freq =
   0.013917606486280
```

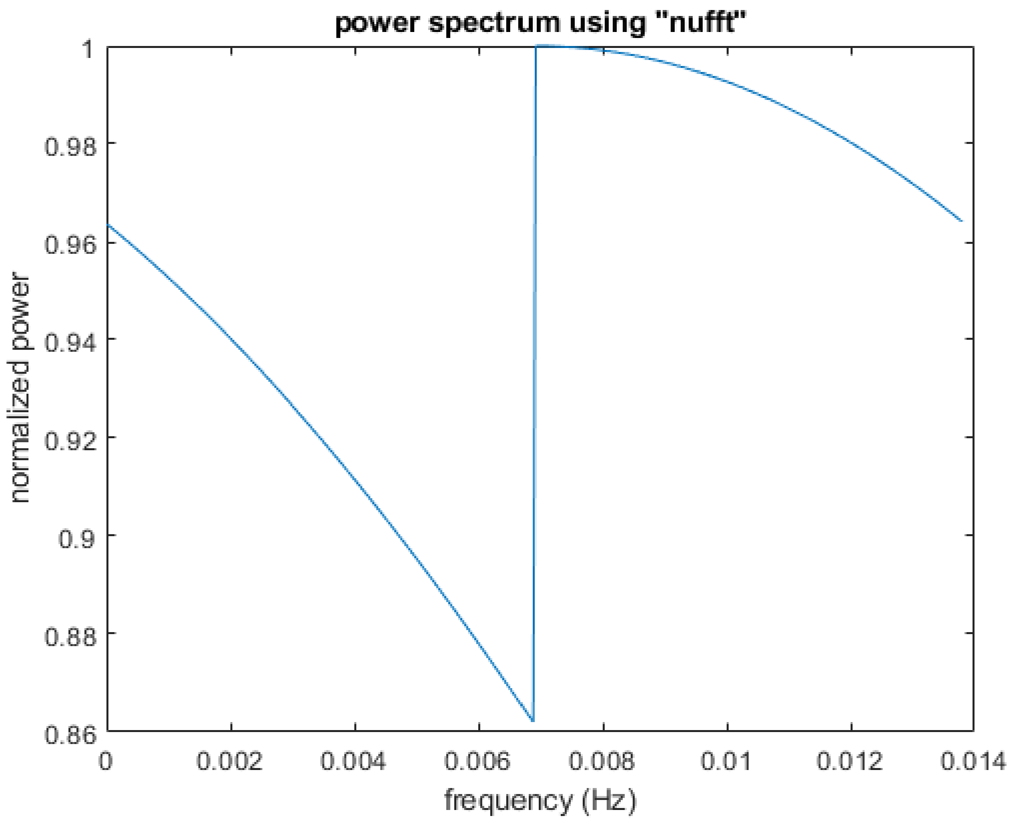
```
avg_freq/key_freq
```

```
ans =
   2.000000932097445
```

```
%I
avg_freq =1/( mean( diff(KIC_11924311.I_time) ) *86400 );
[Tnew,Mnew] = Interp_spline(KIC_11924311.I_time,KIC_11924311.I_flux);
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(KIC_11924311.I_time,KIC_11924311.I_flux,dt);
```




```
N = length(KIC_11924311.I_time);
F = nufft( KIC_11924311.I_time,KIC_11924311.I_flux )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.00691972
```

```
key_freq =
   0.006919720000000
```

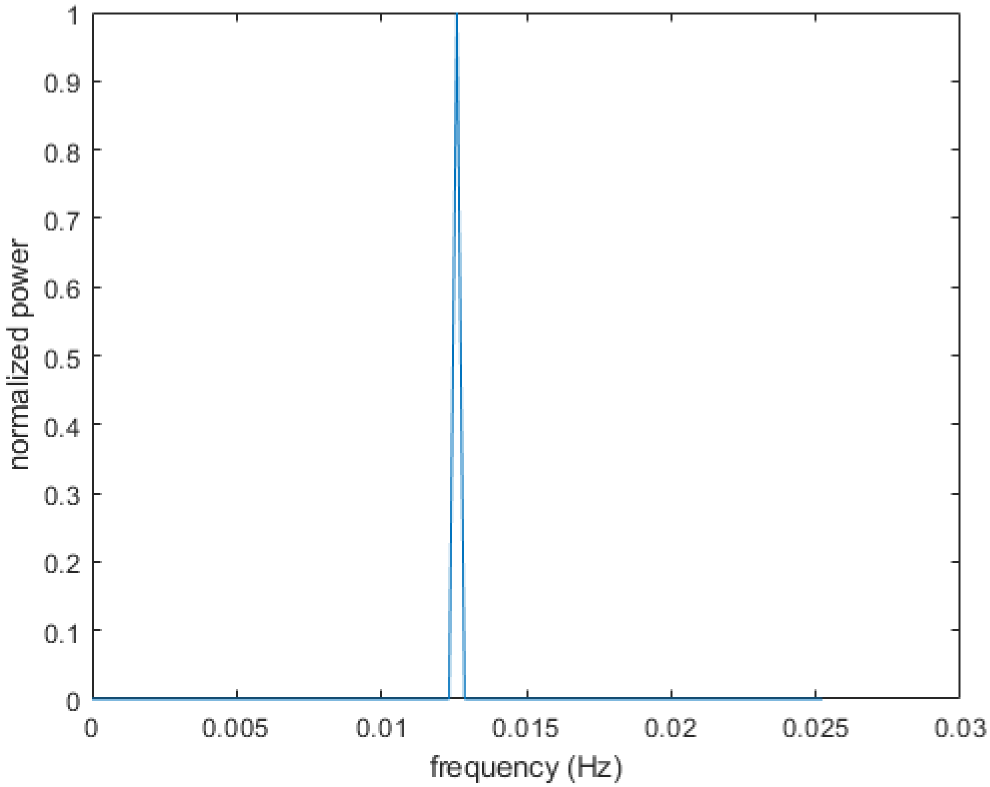
```
avg_freq =1/( mean( diff(KIC_11924311.I_time) ) *86400 )
```

```
avg_freq =
   0.013839433851694
```

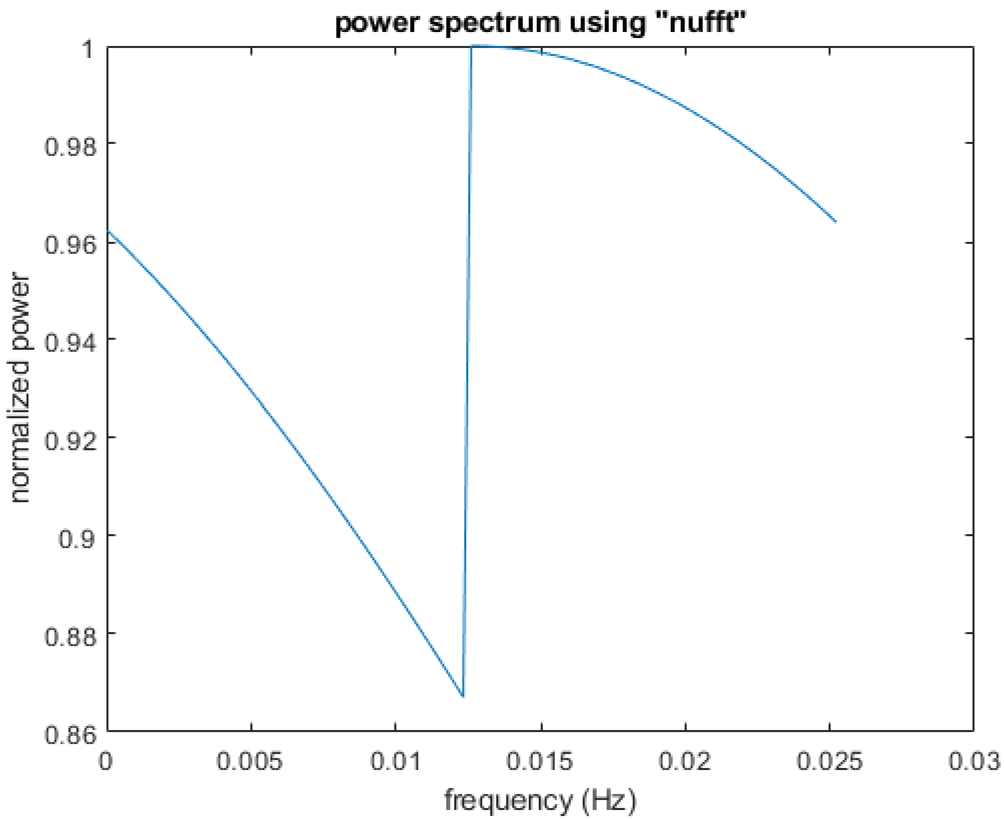
```
avg_freq/key_freq
```

```
ans =
   1.999999111480495
```

```
%BD48_1098
BD48_1098 = BD48_1098(2:end,:);
%B
avg_freq =1/( mean( diff(BD48_1098(:,1)) ) *86400 );

[Tnew,Mnew] = Interp_spline(BD48_1098(:,1),BD48_1098(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(BD48_1098(:,1),BD48_1098(:,2),dt);
```




```
N = length(BD48_1098(:,1));
F = nufft(BD48_1098(:,1),BD48_1098(:,2) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.01275
```

```
key_freq =
   0.012750000000000
```

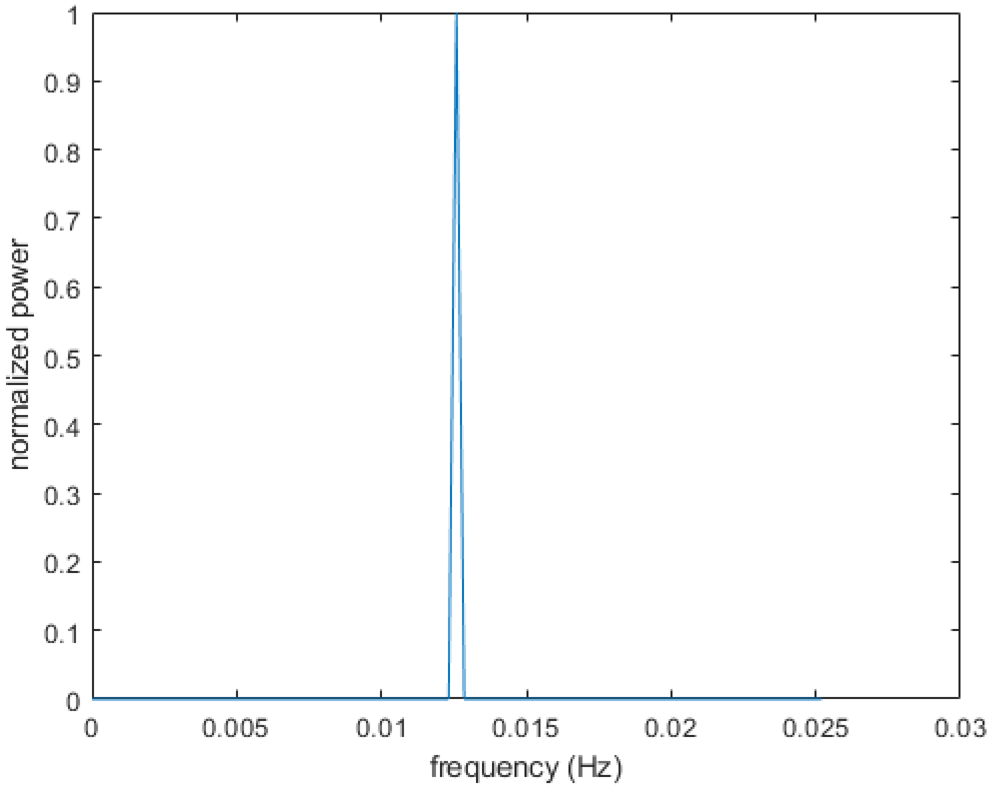
```
avg_freq =1/( mean( diff(BD48_1098(:,1)) ) *86400 )
```

```
avg_freq =
   0.025491165955887
```

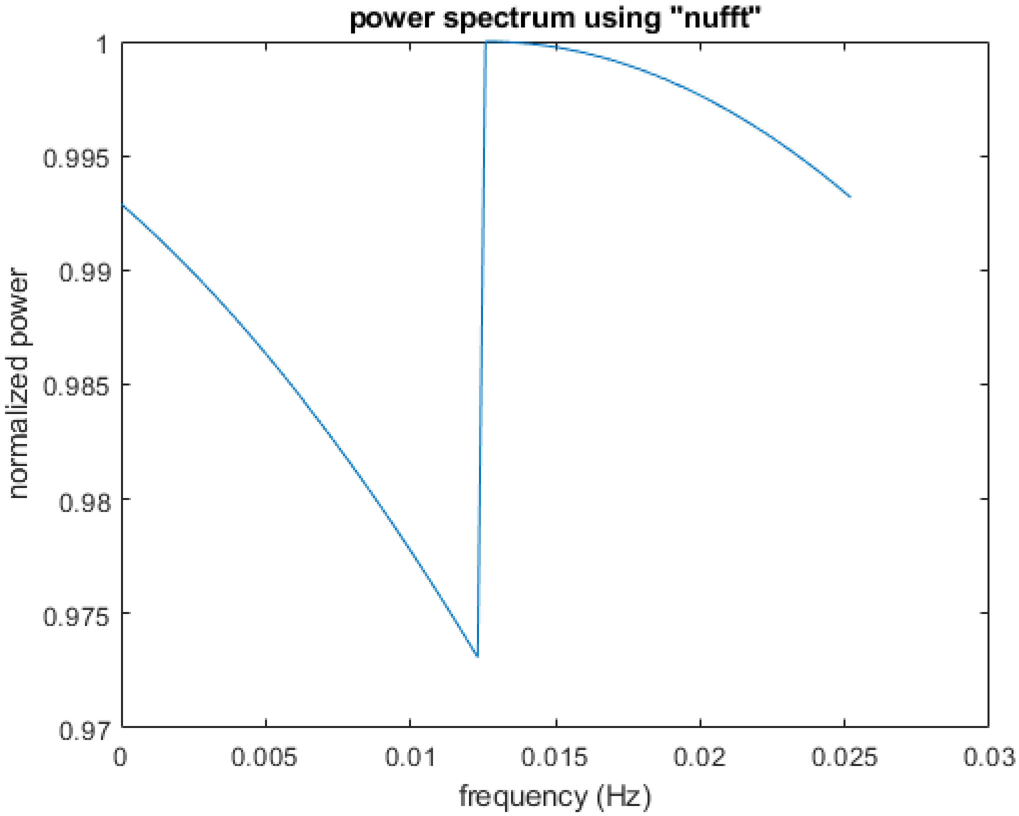
```
avg_freq/key_freq
```

```
ans =
   1.999307133795079
```

```
%R
avg_freq =1/( mean( diff(BD48_1098(:,3)) ) *86400 );

[Tnew,Mnew] = Interp_spline(BD48_1098(:,3),BD48_1098(:,4));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(BD48_1098(:,3),BD48_1098(:,4),dt);
```




```
N = length(BD48_1098(:,3));
F = nufft(BD48_1098(:,3),BD48_1098(:,4) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0125967
```

```
key_freq =
   0.012596700000000
```

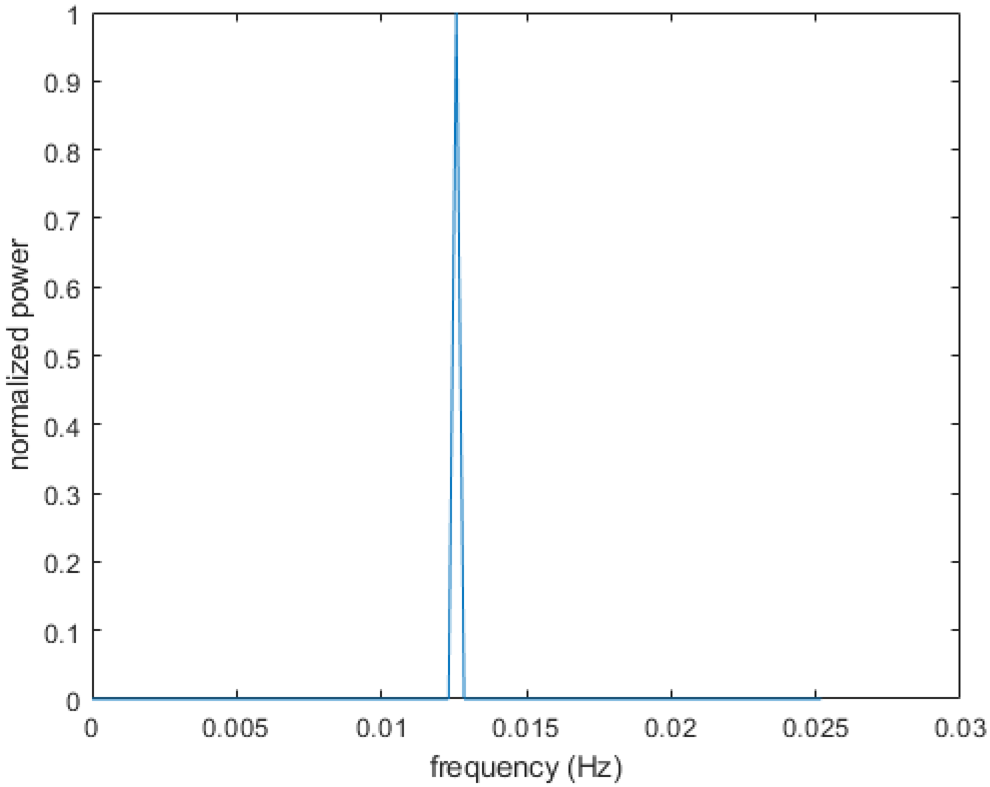
```
avg_freq =1/( mean( diff(BD48_1098(:,3)) ) *86400 )
```

```
avg_freq =
   0.025461337770577
```

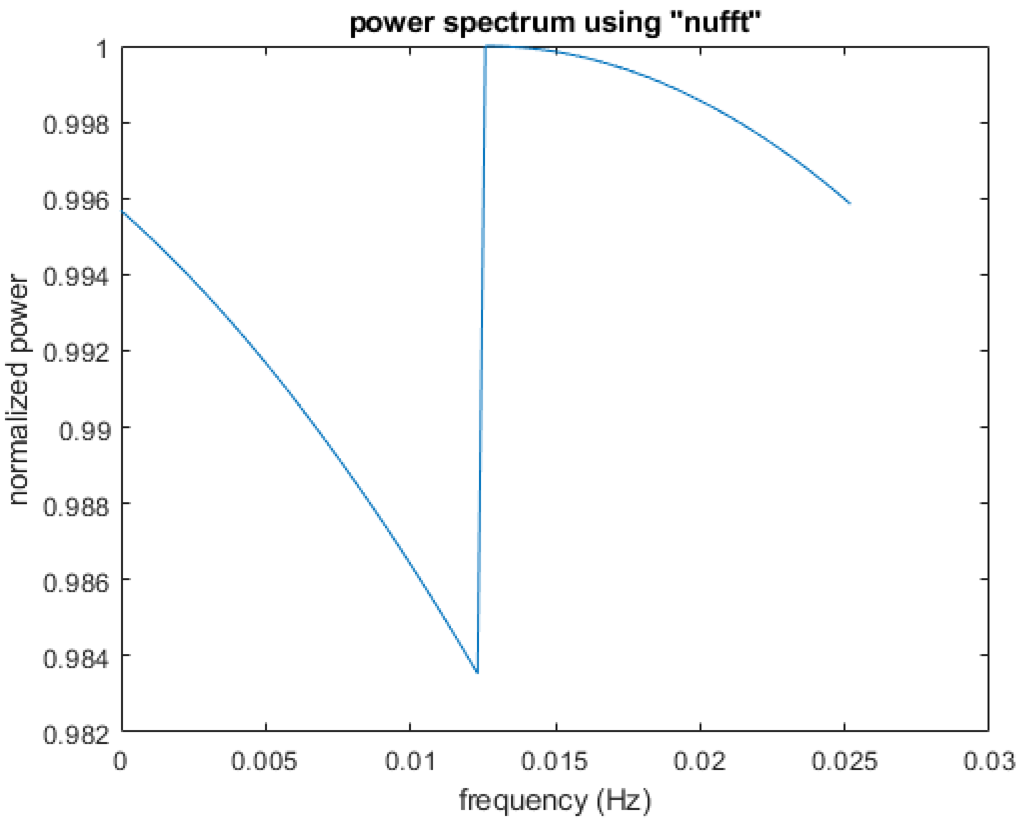
```
avg_freq/key_freq
```

```
ans =
   2.021270473264992
```

```
%V
avg_freq =1/( mean( diff(BD48_1098(:,5)) ) *86400 );

[Tnew,Mnew] = Interp_spline(BD48_1098(:,5),BD48_1098(:,6));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(BD48_1098(:,5),BD48_1098(:,6),dt);
```




```
N = length(BD48_1098(:,5));
F = nufft(BD48_1098(:,5),BD48_1098(:,6) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0127234
```

```
key_freq =
   0.012723400000000
```

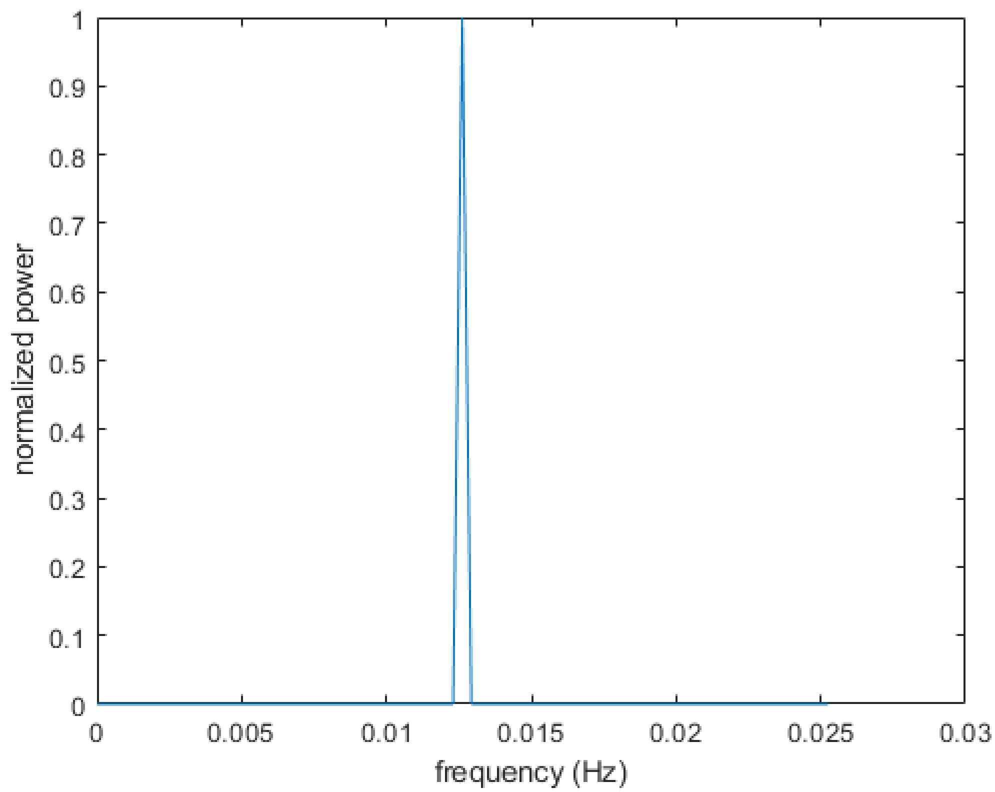
```
avg_freq =1/( mean( diff(BD48_1098(:,5)) ) *86400 )
```

```
avg_freq =
   0.025455380510899
```

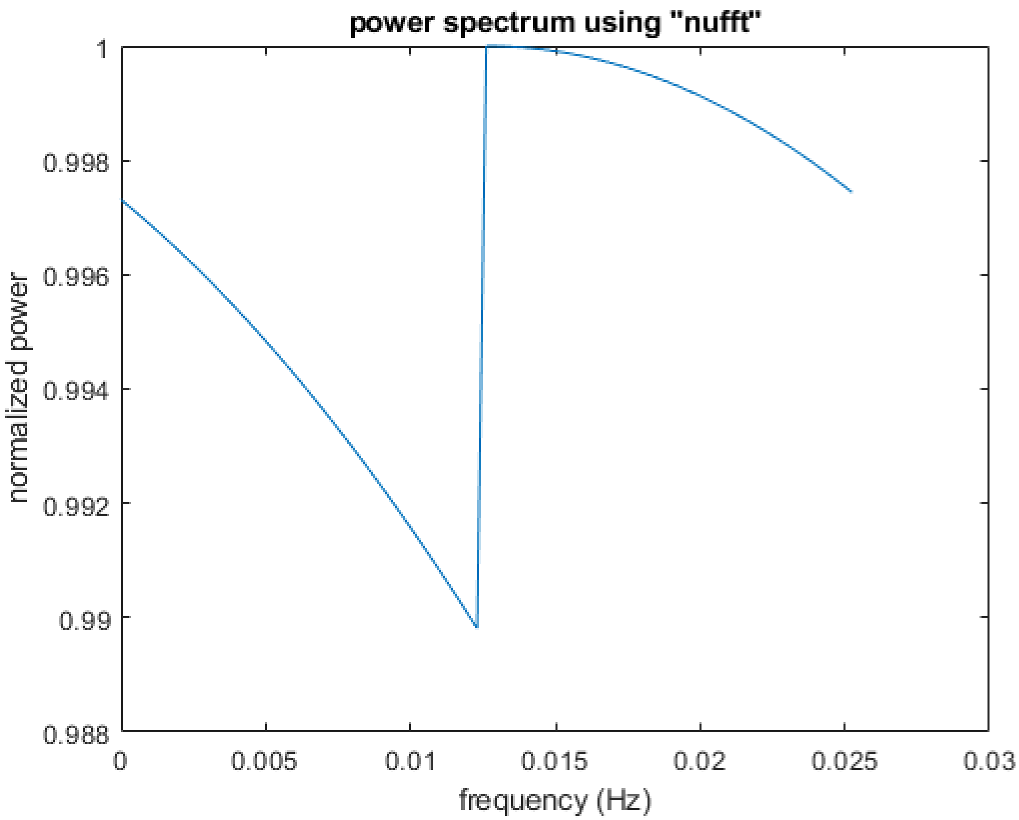
```
avg_freq/key_freq
```

```
ans =
   2.000674388205868
```

```
%BD5_441
BD55_441 = BD55_441(2:end-1,:);
%V
avg_freq =1/( mean( diff(BD55_441(:,5)) ) *86400 );

[Tnew,Mnew] = Interp_spline(BD55_441(:,5),BD55_441(:,6));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(BD55_441(:,5),BD55_441(:,6),dt);
```




```
N = length(BD55_441(:,5));
F = nufft(BD55_441(:,5),BD55_441(:,6) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0127232
```

```
key_freq =
   0.012723200000000
```

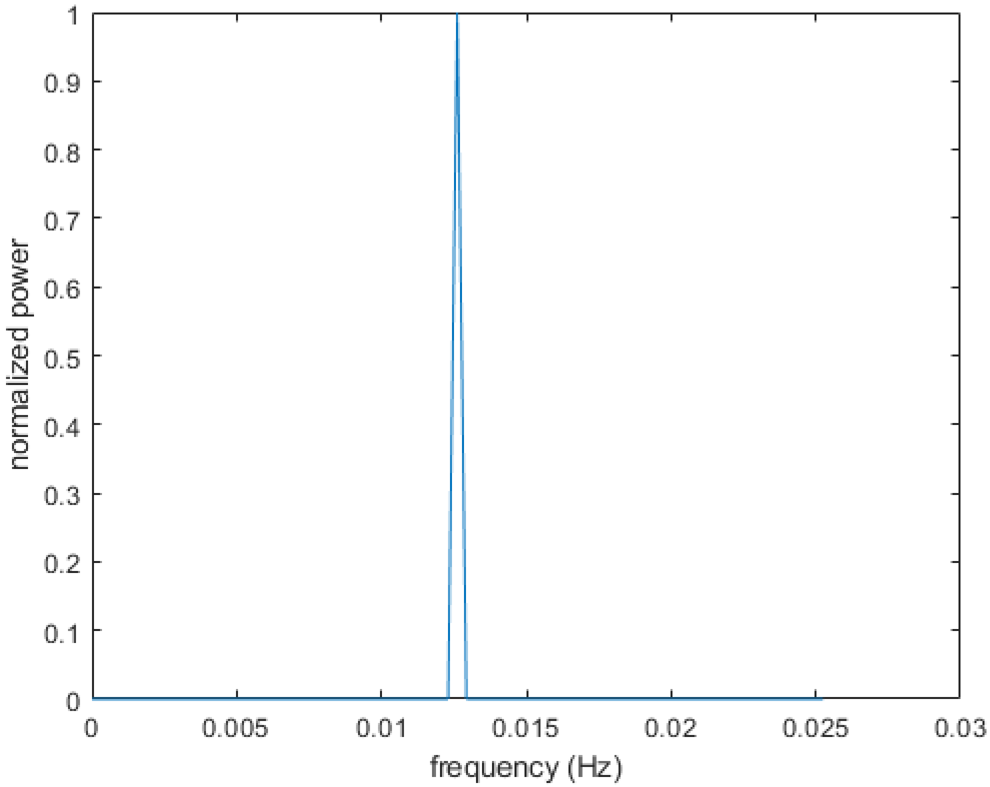
```
avg_freq =1/( mean( diff(BD55_441(:,5)) ) *86400 )
```

```
avg_freq =
   0.025563940526795
```

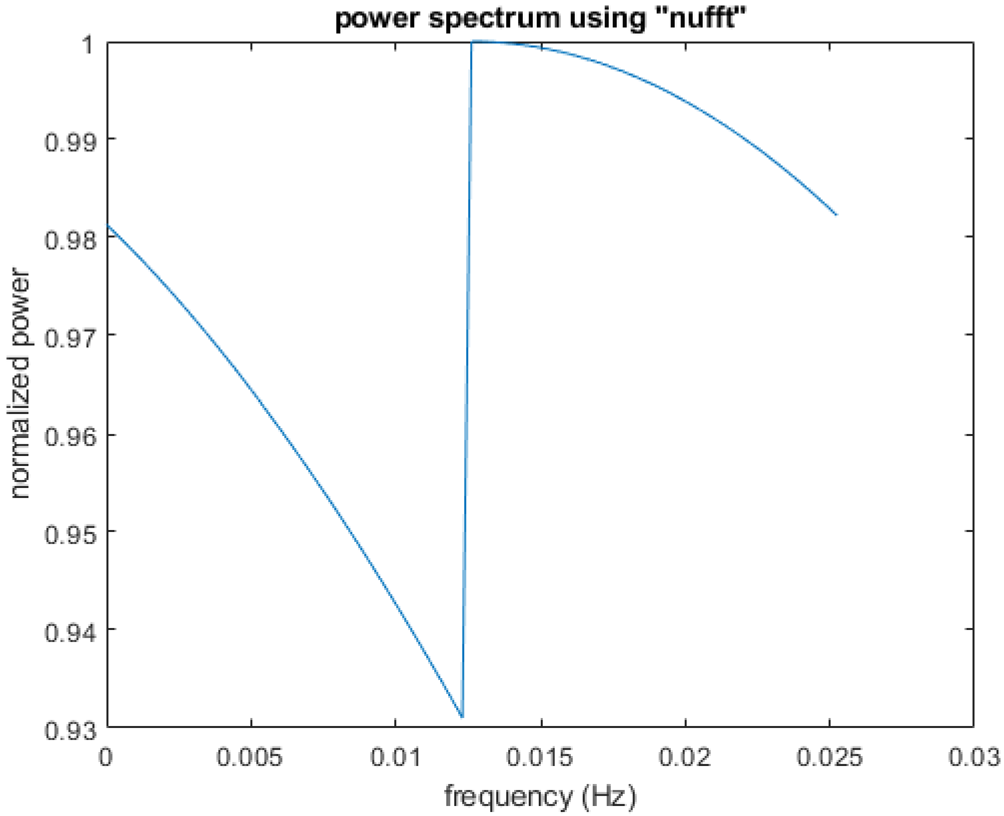
```
avg_freq/key_freq
```

```
ans =
   2.009238283355988
```

```
%B
avg_freq =1/( mean( diff(BD55_441(:,1)) ) *86400 );

[Tnew,Mnew] = Interp_spline(BD55_441(:,1),BD55_441(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(BD55_441(:,1),BD55_441(:,2),dt);
```




```
N = length(BD55_441(:,1));
F = nufft(BD55_441(:,1),BD55_441(:,2) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0127699
```

```
key_freq =
   0.012769900000000
```

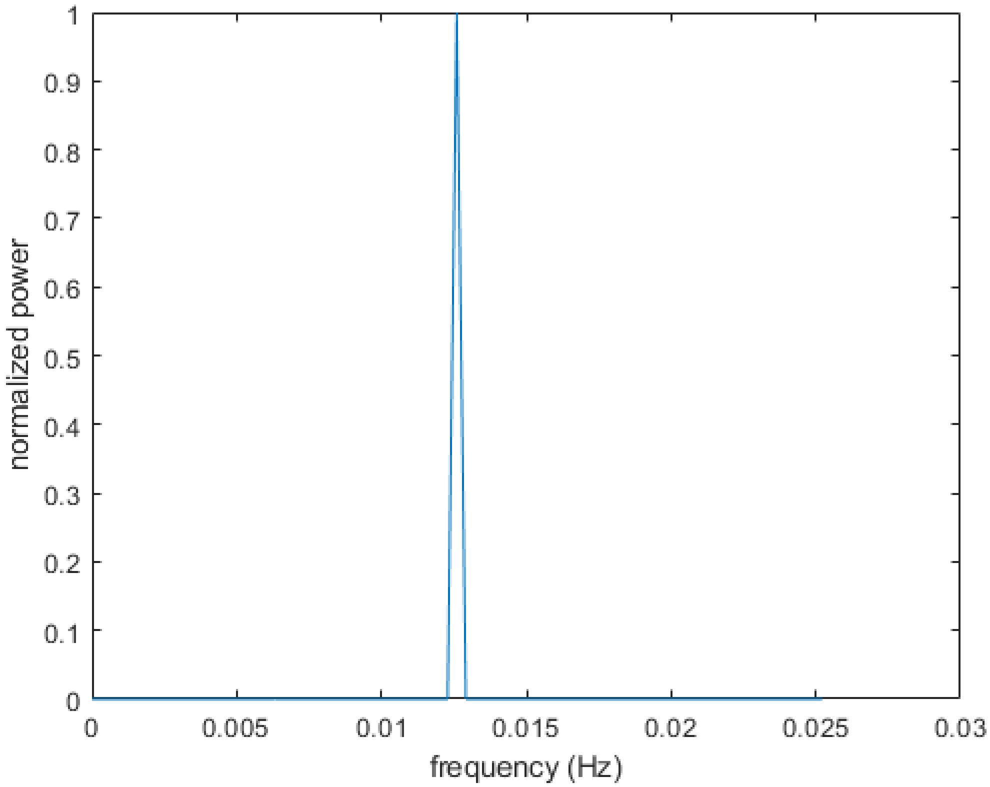
```
avg_freq =1/( mean( diff(BD55_441(:,1)) ) *86400 )
```

```
avg_freq =
   0.025563940526795
```

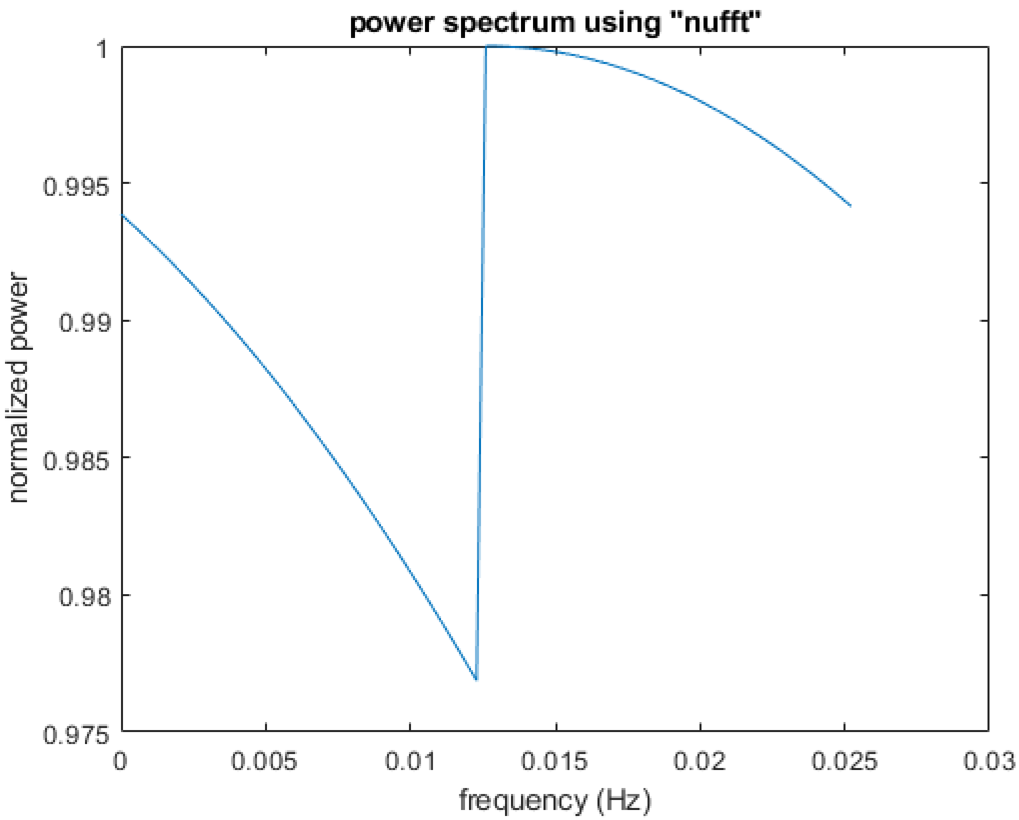
```
avg_freq/key_freq
```

```
ans =
   2.001890424106290
```

```
%R
avg_freq =1/( mean( diff(BD55_441(:,3)) ) *86400 );

[Tnew,Mnew] = Interp_spline(BD55_441(:,3),BD55_441(:,4));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(BD55_441(:,3),BD55_441(:,4),dt);
```




```
N = length(BD55_441(:,3));
F = nufft(BD55_441(:,3),BD55_441(:,4) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0127829
```

```
key_freq =
   0.012782900000000
```

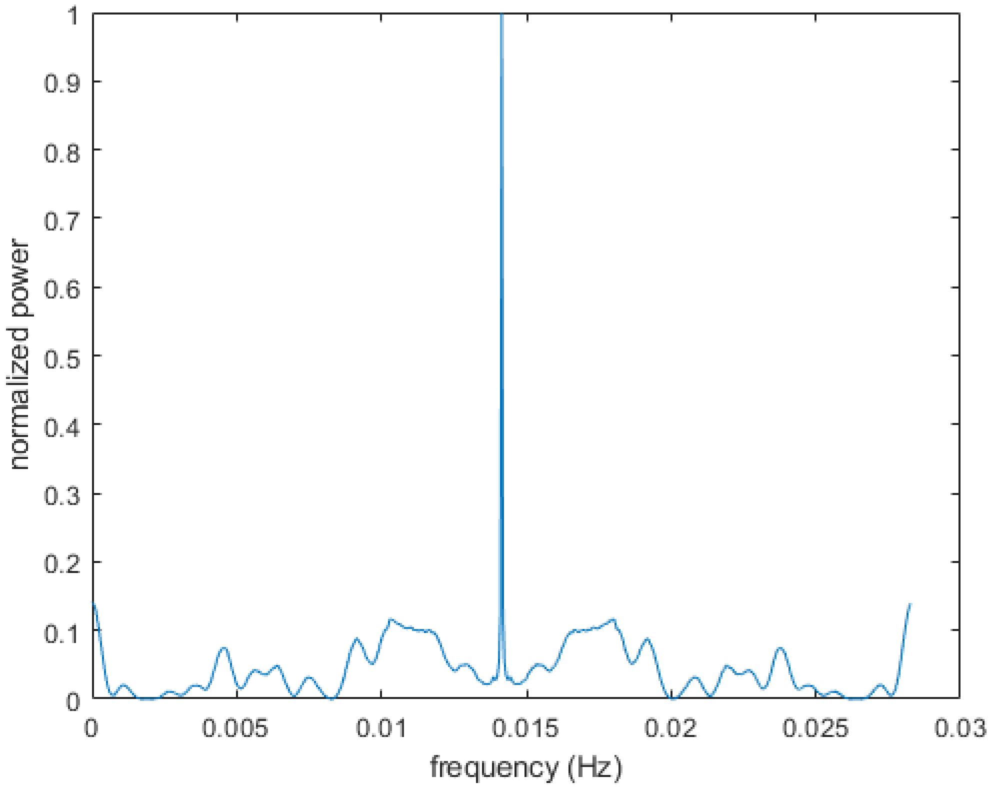
```
avg_freq =1/( mean( diff(BD55_441(:,3)) ) *86400 )
```

```
avg_freq =
   0.025528699366024
```

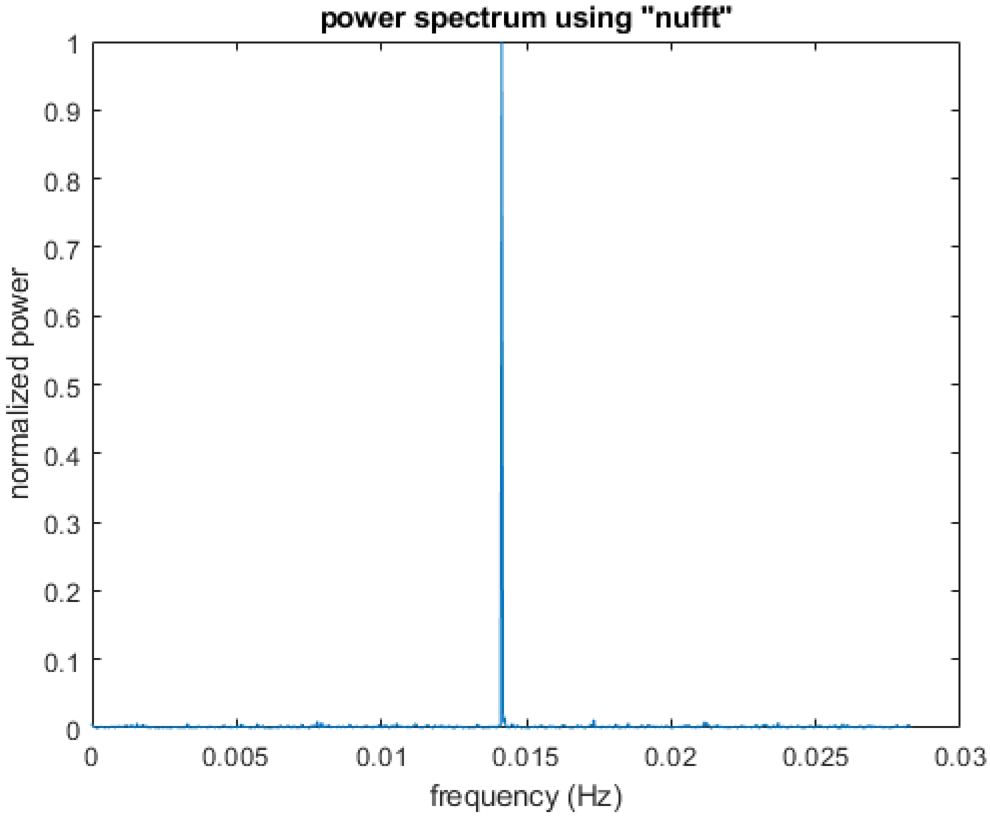
```
avg_freq/key_freq
```

```
ans =
   1.997097635593180
```

```
%HD28497
%B
avg_freq =1/( mean( diff(HD28497(:,1)) ) *86400 );

[Tnew,Mnew] = Interp_spline(HD28497(:,1),HD28497(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(HD28497(:,1),HD28497(:,2),dt);
```




```
N = length(HD28497(:,1));
F = nufft(HD28497(:,1),HD28497(:,2) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0141626
```

```
key_freq =
   0.014162600000000
```

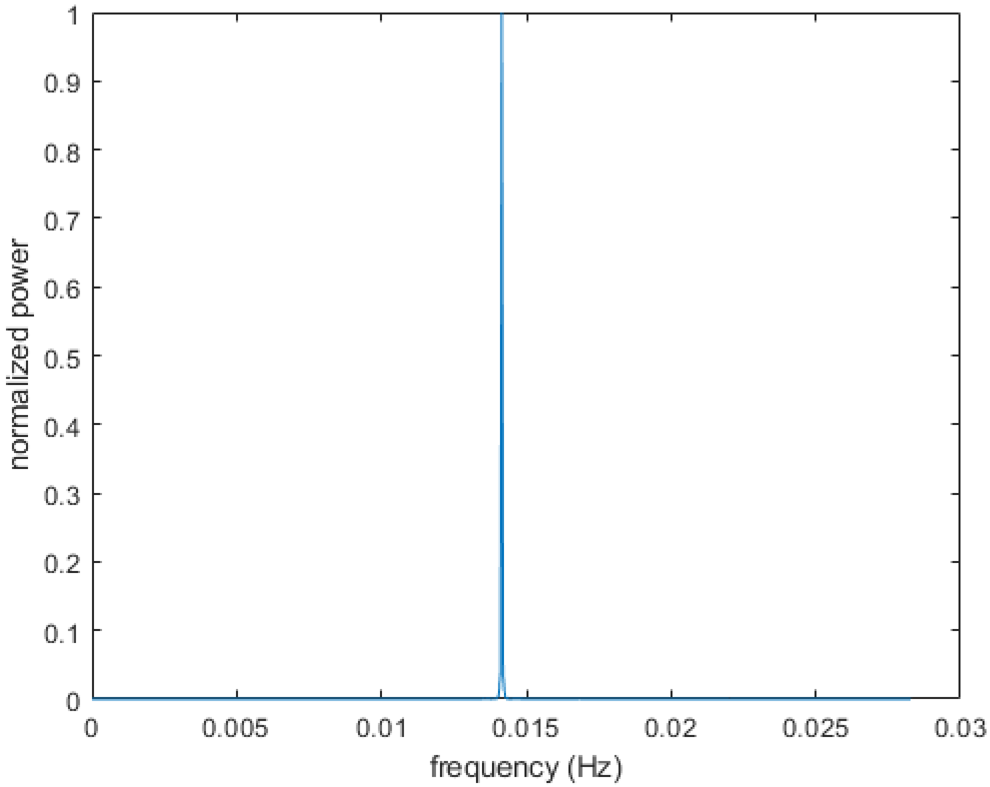
```
avg_freq =1/( mean( diff(HD28497(:,1)) ) *86400 )
```

```
avg_freq =
   0.028325177195336
```

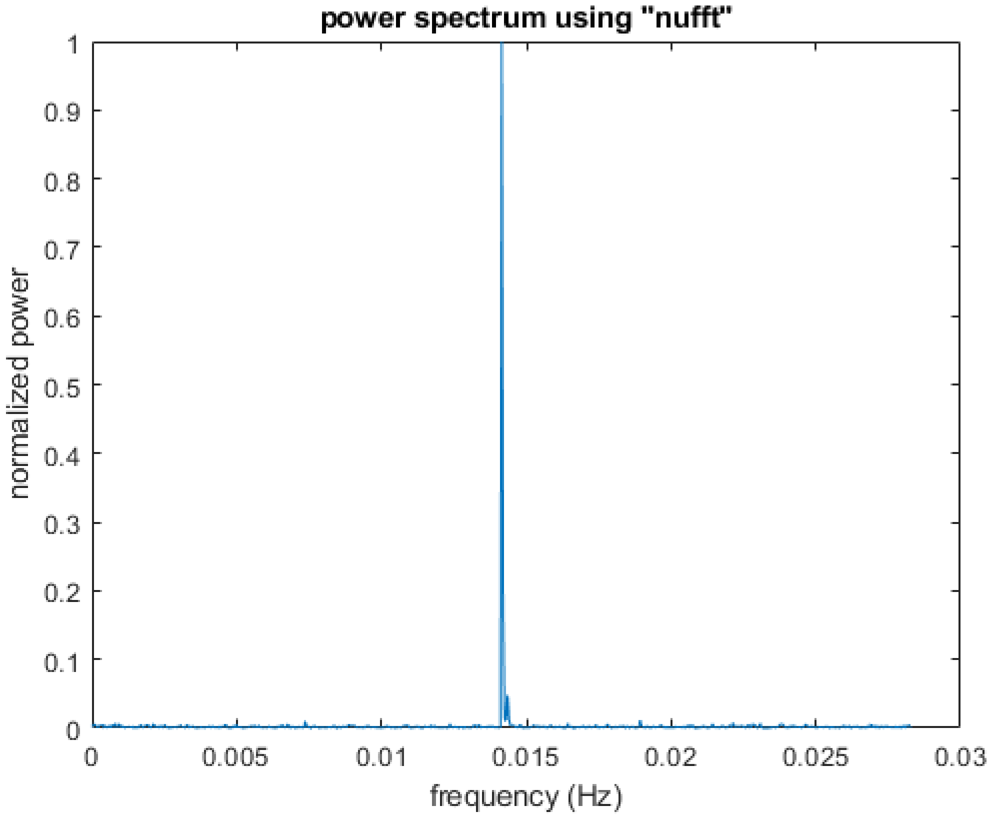
```
avg_freq/key_freq
```

```
ans =
   1.999998389796817
```

```
%R
avg_freq =1/( mean( diff(HD28497(:,3)) ) *86400 );

[Tnew,Mnew] = Interp_spline(HD28497(:,3),HD28497(:,4));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(HD28497(:,3),HD28497(:,4),dt);
```




```
N = length(HD28497(:,3));
F = nufft(HD28497(:,3),HD28497(:,4) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0141626
```

```
key_freq =
   0.014162600000000
```

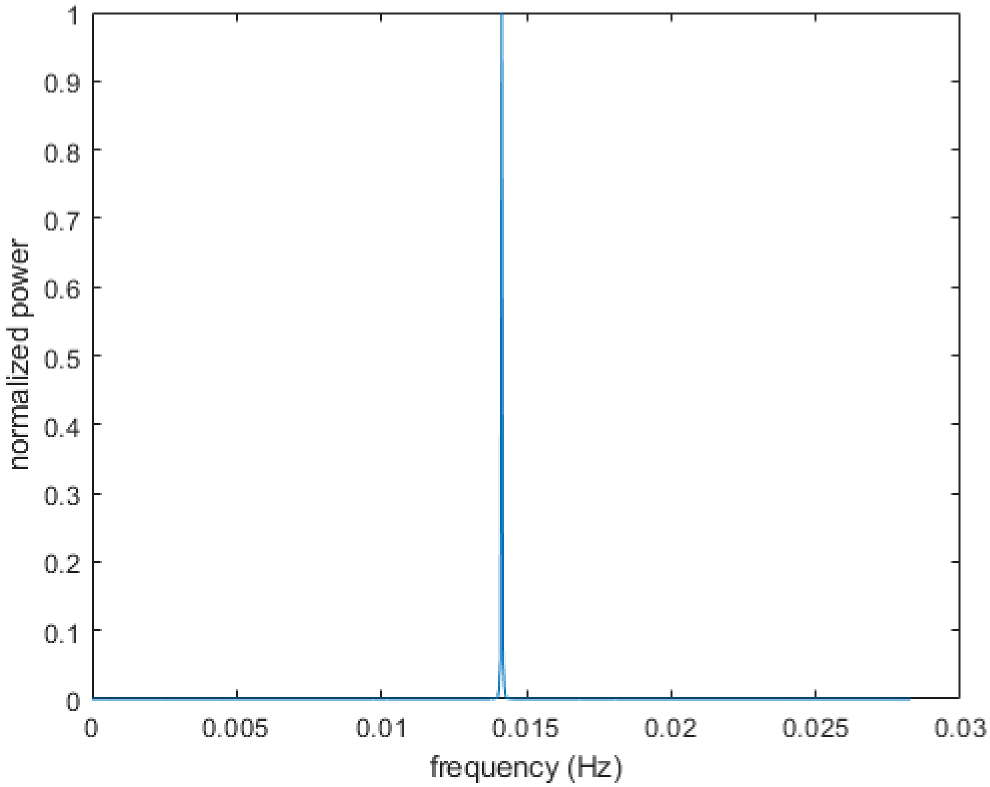
```
avg_freq =1/( mean( diff(HD28497(:,3)) ) *86400 )
```

```
avg_freq =
   0.028325177196178
```

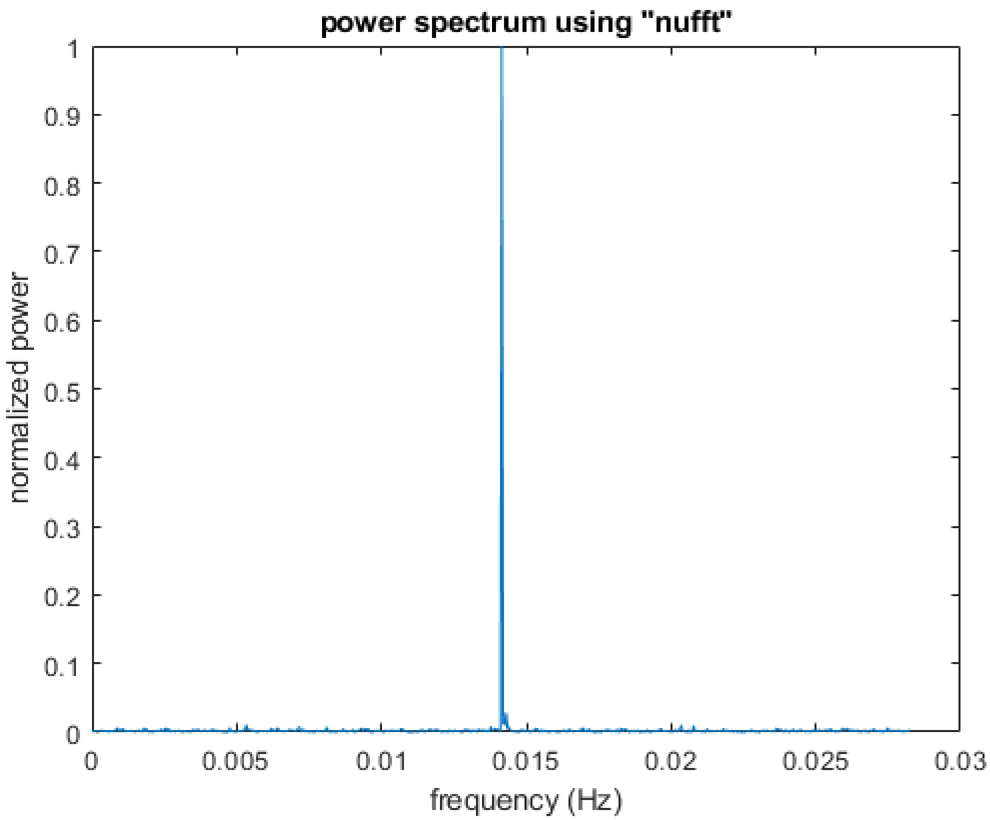
```
avg_freq/key_freq
```

```
ans =
   1.999998389856271
```

```
%V
avg_freq =1/( mean( diff(HD28497(:,5)) ) *86400 );

[Tnew,Mnew] = Interp_spline(HD28497(:,5),HD28497(:,6));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(HD28497(:,5),HD28497(:,6),dt);
```




```
N = length(HD28497(:,5));
F = nufft(HD28497(:,5),HD28497(:,6) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.0141626
```

```
key_freq =
   0.014162600000000
```

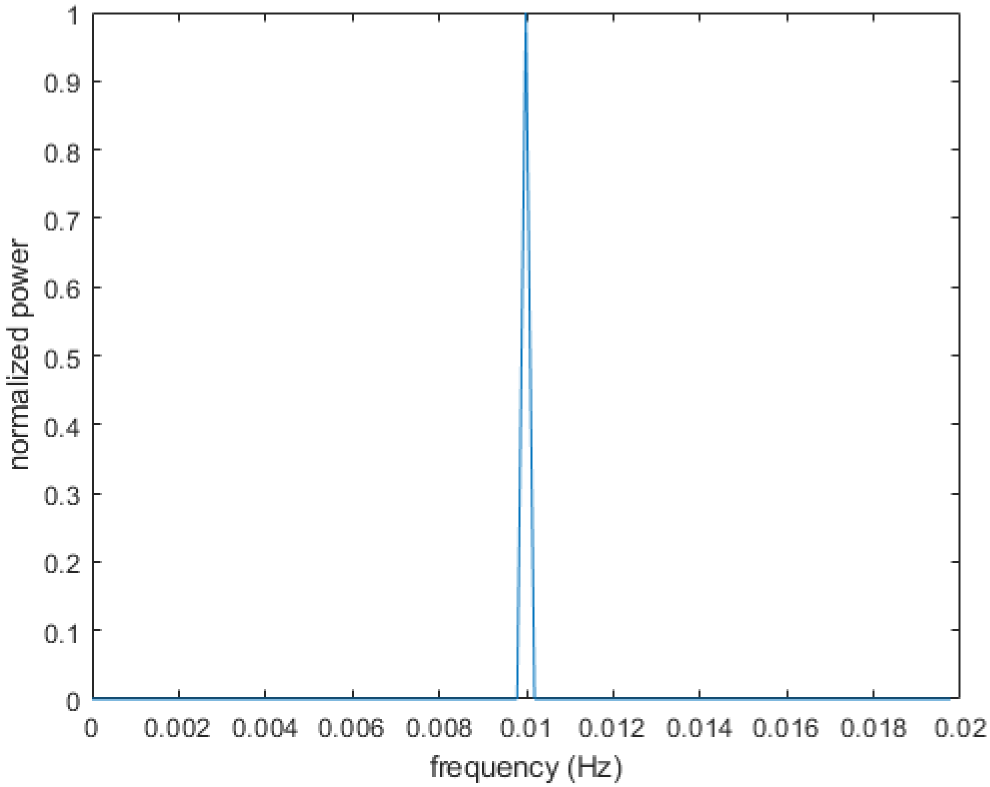
```
avg_freq =1/( mean( diff(HD28497(:,5)) ) *86400 )
```

```
avg_freq =
   0.028325177195336
```

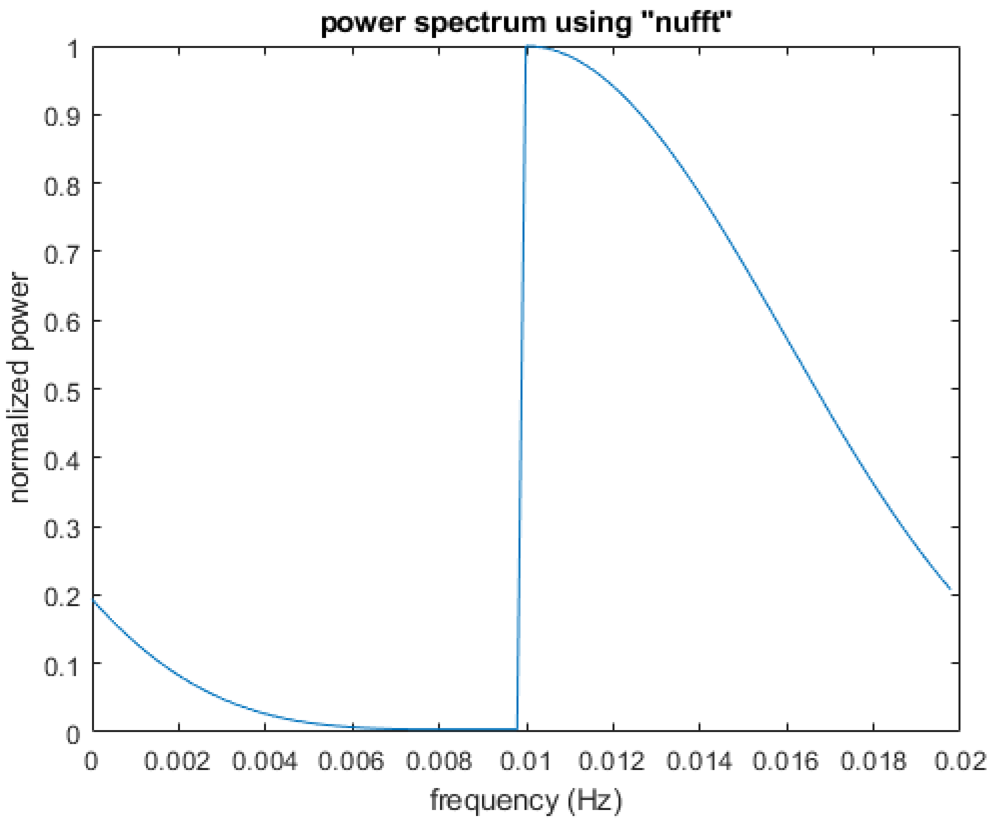
```
avg_freq/key_freq
```

```
ans =
   1.999998389796817
```

```
%HD152060
%R
avg_freq =1/( mean( diff(HD152060(:,1)) ) *86400 );

[Tnew,Mnew] = Interp_spline(HD152060(:,1),HD152060(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(HD152060(:,1),HD152060(:,2),dt);
```




```
N = length(HD152060(:,1));
F = nufft(HD152060(:,1),HD152060(:,2) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.00998548
```

```
key_freq =
   0.009985480000000
```

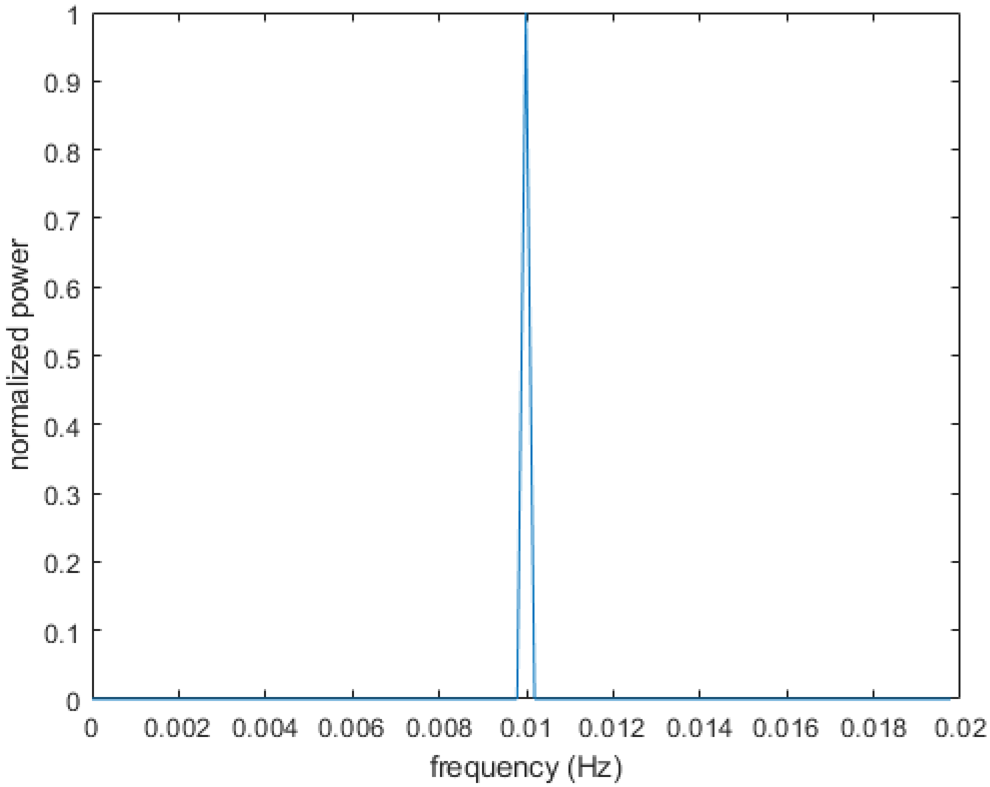
```
avg_freq =1/( mean( diff(HD152060(:,1)) ) *86400 )
```

```
avg_freq =
   0.019970951344655
```

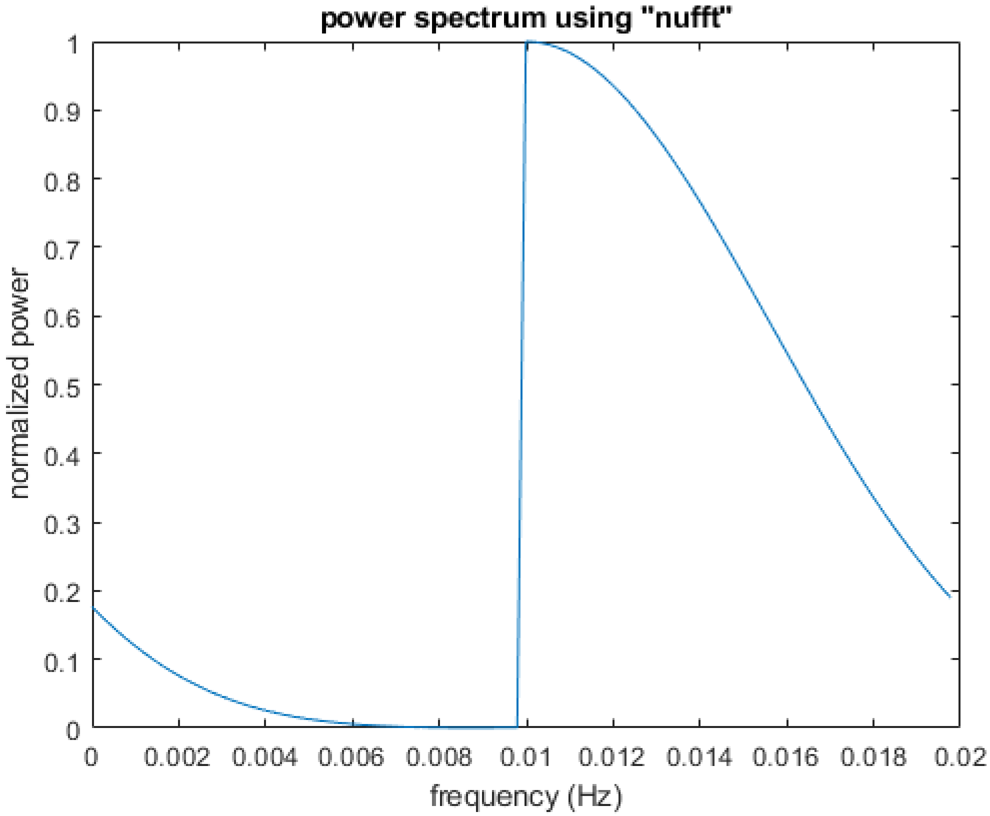
```
avg_freq/key_freq
```

```
ans =
   1.999999133206939
```

```
%V
avg_freq =1/( mean( diff(HD152060(:,3)) ) *86400 );

[Tnew,Mnew] = Interp_spline(HD152060(:,3),HD152060(:,4));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(HD152060(:,3),HD152060(:,4),dt);
```




```
N = length(HD152060(:,3));
F = nufft(HD152060(:,3),HD152060(:,4) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.00998635
```

```
key_freq =
   0.009986350000000
```

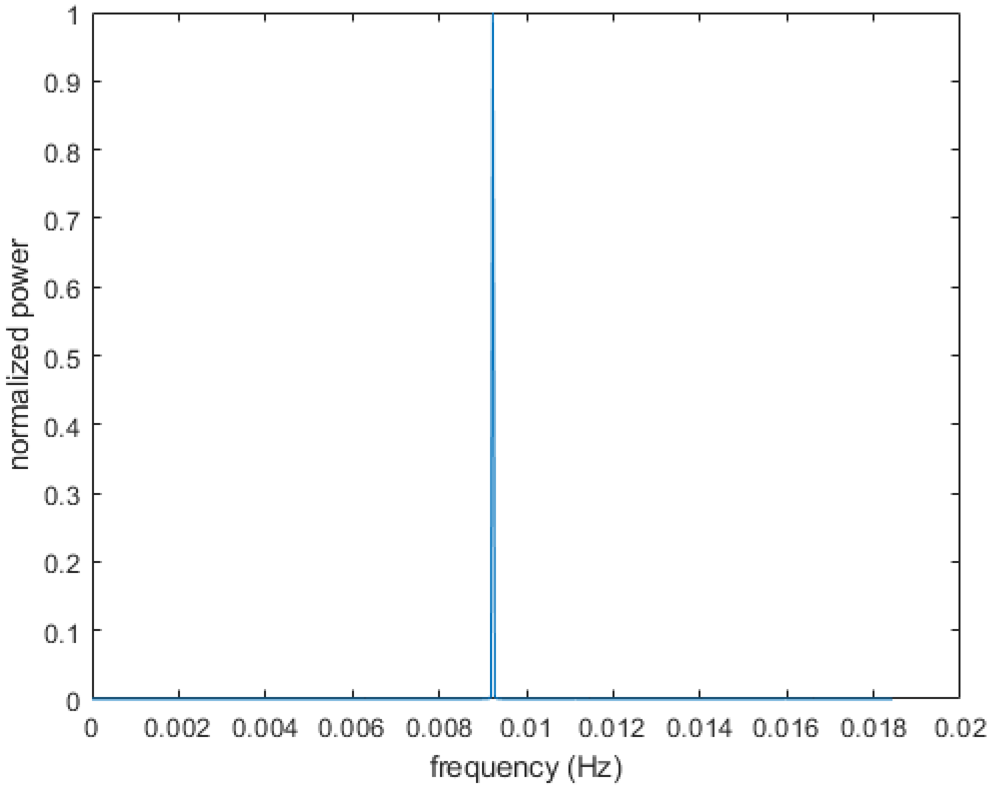
```
avg_freq =1/( mean( diff(HD152060(:,3)) ) *86400 )
```

```
avg_freq =
   0.019972691882309
```

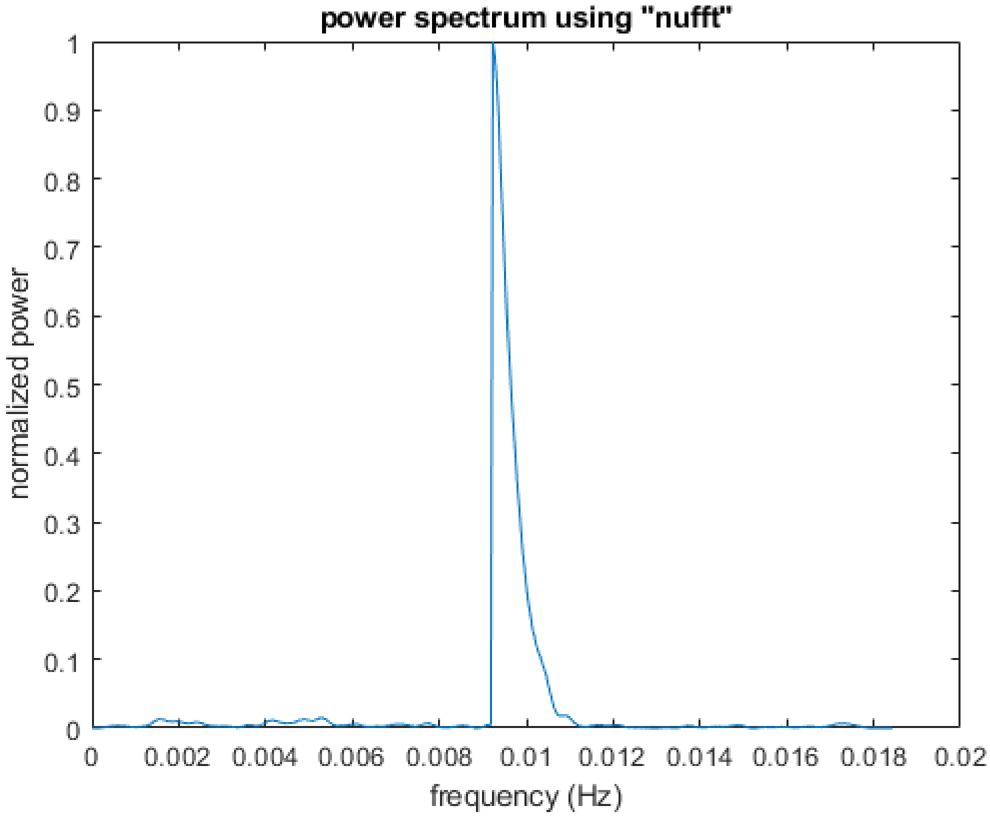
```
avg_freq/key_freq
```

```
ans =
   1.999999187121354
```

```
%HD209522
avg_freq =1/( mean( diff(HD209522(:,1)) ) *86400 );

[Tnew,Mnew] = Interp_spline(HD209522(:,1),HD209522(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(HD209522(:,1),HD209522(:,2),dt);
```




```
N = length(HD209522(:,1));
F = nufft(HD209522(:,1),HD209522(:,2) )/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);

%Plot power Spectrum
fs = avg_freq;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')
title('power spectrum using "nufft"')
```




```
key_freq = 0.00923606
```

```
key_freq =
   0.009236060000000
```

```
avg_freq =1/( mean( diff(HD209522(:,1)) ) *86400 )
```

```
avg_freq =
   0.018472115837205
```

```
avg_freq/key_freq
```

```
ans =
   1.999999549288869
```

Interpolation Functions

There are two different linear interpolation functions - the first is without built-in functions and the other is built-in. The results from both funcitons are similar.

Other interpolation methods are displayed.

Custom functions

```
function [Tnew,Mnew] = Interp_Lin(T,M)
%This is Joe's custom linear interpolation:
%----
```

```
% --Sum all of the time differences between measurements--
n = numel(T);
sum = 0;
for l = 1:n-1
    sum = sum + abs(T(l+1) - T(l));
end

% --Find averaged time scale--
avg_dT = sum / (n-1);
Tnew = T(1):avg_dT:T(n);

% --Calculate Mnew values--
m = numel(Tnew);
Mnew = zeros(1,m);

Mnew(1) = M(1);
Mnew(m) = M(n);

for l = 2:m-1
    for k = 1:n
        if T(k) <= Tnew(l)
            if Tnew(l) <= T(k+1)
                Mnew(l) = (M(k+1) - M(k))./(T(k+1) - T(k)).*(Tnew(l)-T(k)) + M(k);
                %eq for a line. i.e. y = mx + b
            end
        end
    end
end

end
```

Various built-in Matlab Functions

```
function [Tnew,Mnew] = Interp_nearest(T,M)
%This uses the built-in function 'interp1' with method 'nearest'
%----

% --Sum all of the time differences between measurements--
n = numel(T);
sum = 0;
for l = 1:n-1
    sum = sum + abs(T(l+1) - T(l));
end

% --Find averaged time scale--
avg_dT = sum / (n-1);
Tnew = T(1):avg_dT:T(n);

% --Calculate Mnew values--
Mnew = interp1(T,M,Tnew,'nearest');
end
```

```
function [Tnew,Mnew] = Interp_linear(T,M)
%This uses the built-in function 'interp1' with method 'linear'
%----

% --Sum all of the time differences between measurements--
n = numel(T);
sum = 0;
for l = 1:n-1
    sum = sum + abs(T(l+1) - T(l));
end

% --Find averaged time scale--
avg_dT = sum / (n-1);
Tnew = T(1):avg_dT:T(n);

%--Calculate Mnew values--
Mnew = interp1(T,M,Tnew,'linear');
end
```

```
function [Tnew,Mnew] = Interp_spline(T,M)
%This uses the built-in function 'interp1' with method 'spline'
%----

% --Sum all of the time differences between measurements--
n = numel(T);
sum = 0;
for l = 1:n-1
    sum = sum + abs(T(l+1) - T(l));
end

% --Find averaged time scale--
avg_dT = sum / (n-1);
Tnew = T(1):avg_dT:T(n);

% --Calculate Mnew values--
Mnew = interp1(T,M,Tnew,'spline');
end
```

```
function [Tnew,Mnew] = Interp_polyfit(T,M)
%This uses the built-in function 'interp1' with method 'polyfit'
%----

% --Sum all of the time differences between measurements--
n = numel(T);
sum = 0;
for l = 1:n-1
    sum = sum + abs(T(l+1) - T(l));
end
```

```
% --Find averaged time scale--
avg_dT = sum / (n-1);
Tnew = T(1):avg_dT:T(n);

% --Calculate Mnew values--
n = numel(T);
p = polyfit(T,M,5);
Mnew = polyval(p,Tnew);
end
```

Joe's linear combo of non-linear functions function

```
function C = NLfit(F1,F2,F3,x,y)
%documentation:  This program was developed from Gilat example 6-9

F1 = F1(x);
F2 = F2(x);
F3 = F3(x);

A(1,1) = sum(F1 .* F1);
A(1,2) = sum(F1 .* F2);
A(1,3) = sum(F1 .* F3);
A(2,2) = sum(F2 .* F2);
A(2,3) = sum(F2 .* F3);
A(3,3) = sum(F3 .* F3);

A(2,1) = A(1,2);    % A is symmetric
A(3,1) = A(1,3);
A(3,2) = A(2,3);

B(1,1) = sum(y .* F1);
B(2,1) = sum(y .* F2);
B(3,1) = sum(y .* F3);

%form is A*C=B, Therefore,

C = A\B;

end
```

Joe's power spectrum function

```
function [fk,powerNor] = EnergySpec(t,f,dt)
%this function is developed with refeernce to Gilat Program 7-4 and Example
%7-6

N = length(f);
F = fft(f)/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);
```

```
%Plot power Spectrum
fs = 1/dt;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')


end
```