

Stochastic Process Realizations

The objective in this document is to go back-and-forth between the frequency domain and the time domain for stochastic processes. In other words, it is an objective to understand the relationship between a power spectral density function and corresponding time histories, i.e., realizations.

Single Sine Wave

Consider a deterministic sine wave with the following characteristics:

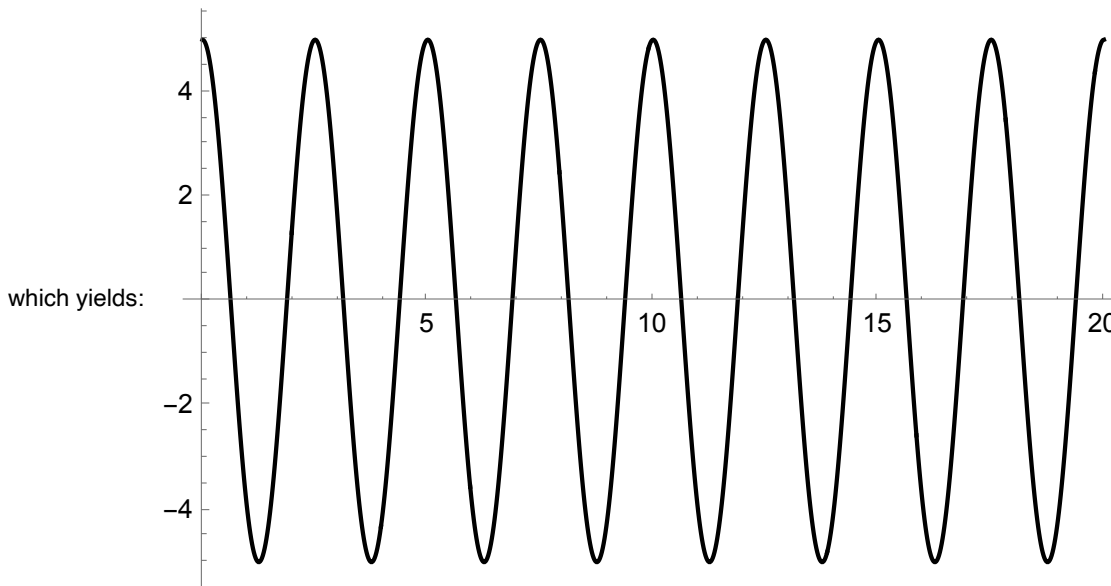
```
amplitude = 5;  
period = 2.5;  
totalTime = 20;
```

Use the following sample rate, i.e., number of data points per second:

```
sampleRate = 50;  
dt =  $\frac{1}{\text{sampleRate}}$ ;  
nintervals = totalTime sampleRate;
```

Create and plot the time history:

```
timeHistory = Table[N[{t, amplitude Cos[ $\frac{2 \pi}{\text{period}} t$ ]}],
  {t, 0, totalTime, dt}];
ListLinePlot[timeHistory, PlotStyle → Black]
```



Conduct a Fourier transform, using certain parameters of the built-in function in Mathematica:

```
spectrum =
  Abs[Fourier[timeHistory[[All, 2]], FourierParameters → {-1, -1}]];
```

Mathematica gives the Fourier transform result as “spectrum along the positive frequency axis” followed by “spectrum along the negative frequency axis.” For that reason, drop the second half here, remembering for later in this spreadsheet that the remainder contains only half of the energy (variance) of the time history:

```
spectrum = Drop[spectrum, - $\frac{\text{nintervals}}{2}$ ];
```

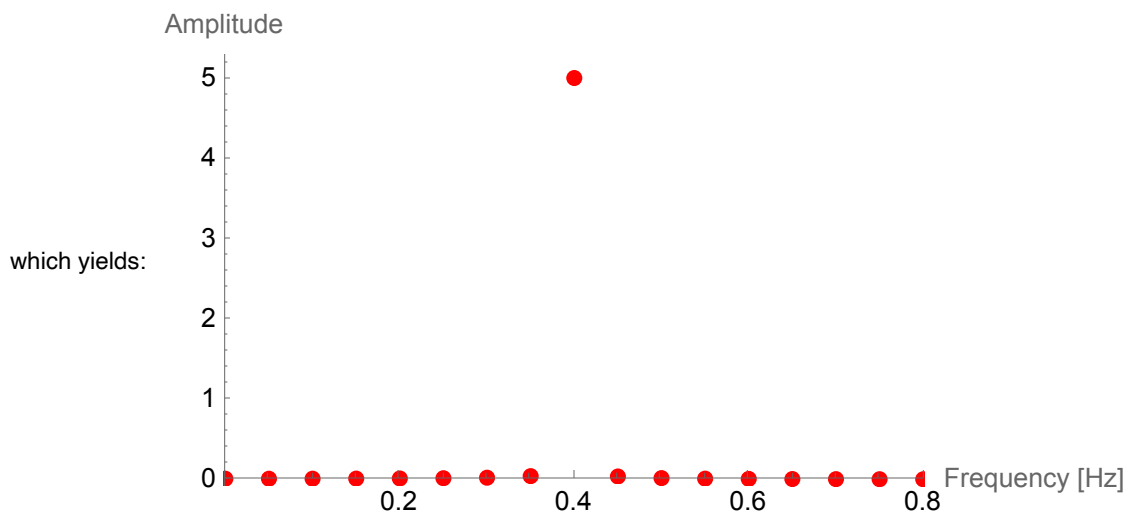
The “delta frequency” value, i.e., the increment along the frequency axis, interestingly:

```
df =  $\frac{\text{sampleRate}}{\text{nintervals} + 1}$ ;
frequencyValues = Range[0, df  $\frac{\text{nintervals}}{2}$ , df];
```

Create the plot along the frequency axis, noticing a spike at the frequency (in Hz) of the single sine

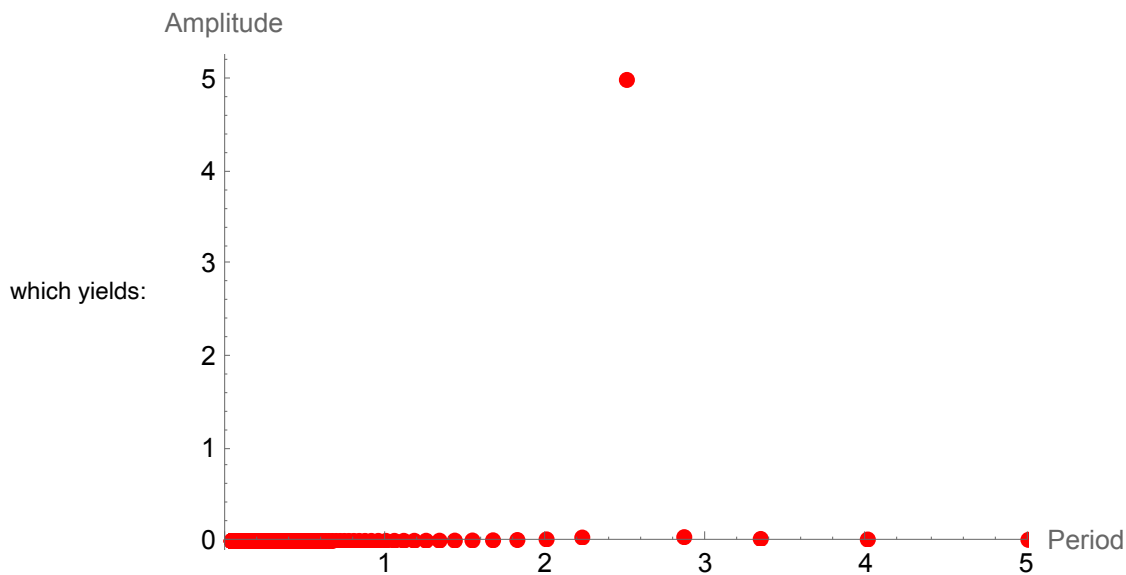
wave:

```
ListPlot[Transpose[{frequencyValues, 2 * spectrum}],
  PlotRange → {{0,  $\frac{2}{\text{period}}$ }, All},
  AxesLabel → {"Frequency [Hz]", "Amplitude"}, PlotMarkers → Automatic,
  PlotStyle → Red]
```



Plot that Fourier transform as a function of period rather than frequency, removing the first zero frequency to avoid divide-by-zero when inverting frequency to get period (notice the energy concentrated at the input period):

```
spectrum = Drop[spectrum, 1];
frequencyValues = Drop[frequencyValues, 1];
periodValues =  $\frac{1}{\text{frequencyValues}}$ ;
ListPlot[Transpose[{periodValues, 2 * spectrum}],
  PlotRange -> {{0, 2 period}, All}, AxesLabel -> {"Period", "Amplitude"},
  PlotMarkers -> Automatic, PlotStyle -> Red]
```



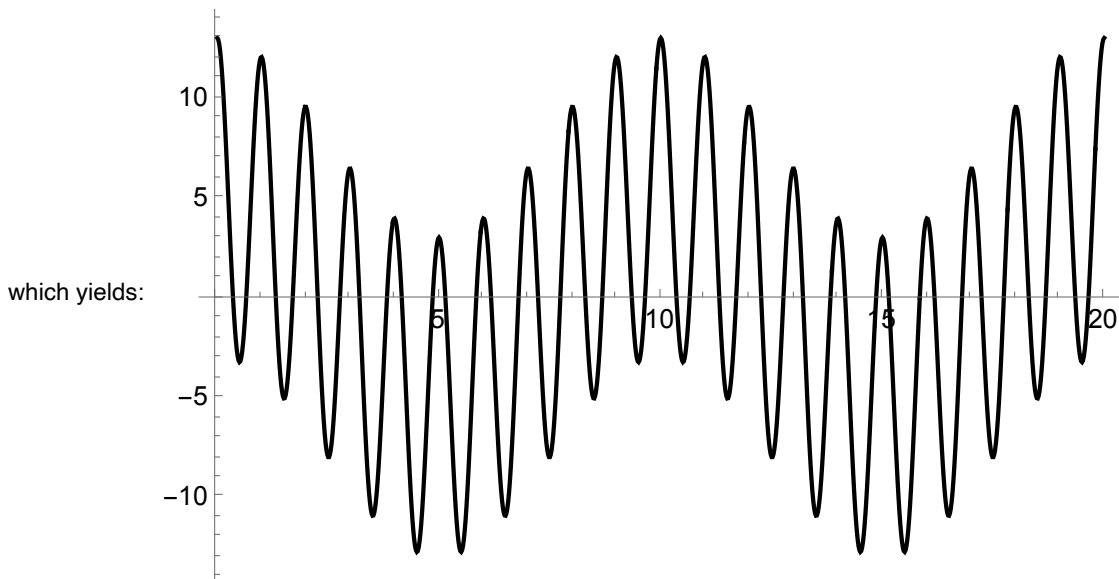
Two Sine Waves

For no particular reason, the exercise above is repeated for two deterministic frequencies. The sample rate and time duration is kept from earlier, and the two sine waves have the following properties:

```
amplitude1 = 5;
period1 = 10;
amplitude2 = 8;
period2 = 1;
```

Create and plot the time history:

```
timeHistory =
  Table[N[{t, amplitude1 Cos[ $\frac{2 \pi}{\text{period1}}$  t] + amplitude2 Cos[ $\frac{2 \pi}{\text{period2}}$  t]}],
    {t, 0, dt nintervals, dt}];
ListLinePlot[timeHistory, PlotStyle → Black]
```



Do the Fourier transform, again dropping the second half mirror image of the spectrum:

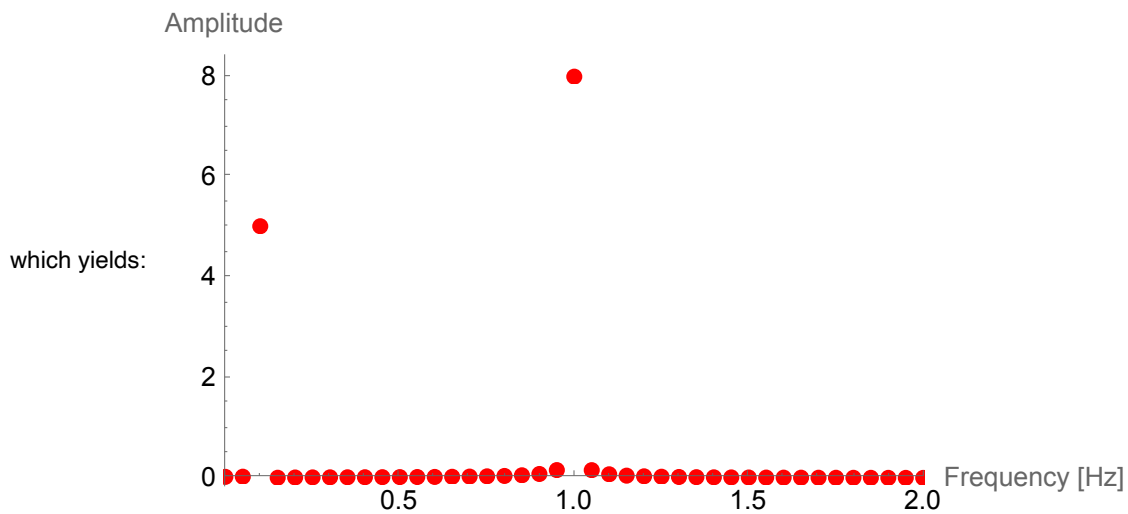
```
spectrum =
  Abs[Fourier[timeHistory[[All, 2]], FourierParameters → {-1, -1}]];
spectrum = Drop[spectrum, - $\frac{\text{nintervals}}{2}$ ];
```

Plot the result:

```

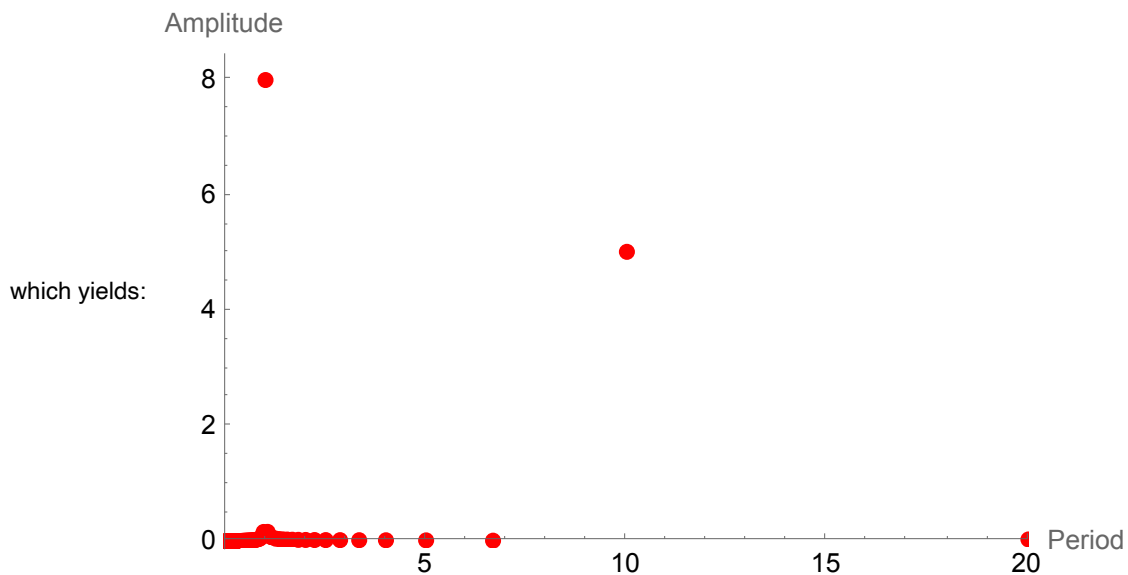
frequencyValues = Range[0, df  $\frac{nintervals}{2}$ , df];
ListPlot[Transpose[{frequencyValues, 2 * spectrum}],
PlotRange -> {{0, 2  $\frac{1}{\text{Min}[\text{period1}, \text{period2}]}$ }, All},
AxesLabel -> {"Frequency [Hz]", "Amplitude"}, PlotMarkers -> Automatic,
PlotStyle -> Red]

```



Plot that same result along the period axis:

```
spectrum = Drop[spectrum, 1];
frequencyValues = Drop[frequencyValues, 1];
periodValues =  $\frac{1}{\text{frequencyValues}}$ ;
ListPlot[Transpose[{periodValues, 2 * spectrum}],
  PlotRange → {{0, 2 Max[period1, period2]}, All},
  AxesLabel → {"Period", "Amplitude"}, PlotMarkers → Automatic,
  PlotStyle → Red]
```



Algorithm to Create Time History

```

createTimeHistory[S_, fmin_, fmax_, nblocks_, sampleRate_,
  totalTime_, randAmpl_] := Module[{ },
  
$$\Delta f = \frac{f_{max} - f_{min}}{nblocks};$$

  timeHistoryFunction = 0;
  For[i = 0, i < nblocks, i++,
    fcentral = fmin + (i + 0.5)  $\Delta f$ ;
    height = S /. f → fcentral;
    intervalStdv =  $\sqrt{\Delta f \text{ height}}$ ;
    If[randAmpl,
      A = RandomVariate[NormalDistribution[0, intervalStdv] ],
      A = intervalStdv;
     $\phi$  = RandomReal[{0, 2  $\pi$ }] ;
    timeHistoryFunction +=  $\sqrt{2}$  Abs[A] Cos[2  $\pi$  fcentral * time +  $\phi$ ]
  ];
  
$$dt = \frac{1}{sampleRate};$$

  timeAxis = Range[0, totalTime, dt];
  timeHistory = timeHistoryFunction /. time → timeAxis;
]
```


Algorithm to Create Spectrum

```
createSpectrum[series_, sampleRate_, totalTime_] := Module[{ },
  spectrum = Abs[Fourier[series]]^2;
  spectrum = spectrum  $\frac{2}{sampleRate}$ ; (*factor 2 to make it one-sided*)
  nintervals = totalTime sampleRate;
  df =  $\frac{sampleRate}{nintervals + 1}$ ;
  frequencyAxis = Range[0, df nintervals, df];
  halfPoints = Floor[ $\frac{nintervals}{2}$ ];
  spectrum = Drop[spectrum, -halfPoints];
  frequencyAxis = Drop[frequencyAxis, -halfPoints];
]
```

Constant Spectrum

To apply the two algorithms shown above, start with a power spectral density that is constant between two given frequencies:

```
minHz = 1 ;
maxHz = 2 ;
S = 5 ;
```

Specify the duration of the time history, plus the discretization of the spectrum and the time axis:

```
totalTime = 20;
spectrumBlocks = 150;
sampleRate = 200;
```

Specify if the amplitude of each sine wave (of the generated time history) should be random:

```
randAmpl = True;
```

Create and plot the time series:

```
createTimeHistory[S, minHz, maxHz, spectrumBlocks, sampleRate,
  totalTime, randAmpl];
ListLinePlot[Transpose[{timeAxis, timeHistory}], PlotStyle -> Black,
  AxesLabel -> {"Time [seconds]", "Amplitude"}, PlotRange -> All]
```



To informally verify that the shown time history is reasonable, the standard deviation of the process is computed. That standard deviation is the square root of the variance, which is the area underneath the spectrum:

$$\text{stdv} = \sqrt{(\text{maxHz} - \text{minHz}) S} // N$$

which yields: 2.23607

The average of all peaks should appear to be $\sqrt{\frac{\pi}{2}} = 1.25$ times the standard deviation. Additionally, the average of the extremes should be 3 to 4 times the standard deviation. For reference, the observed extreme in the time history shown above is:

```
Max[Abs[timeHistory]]
```

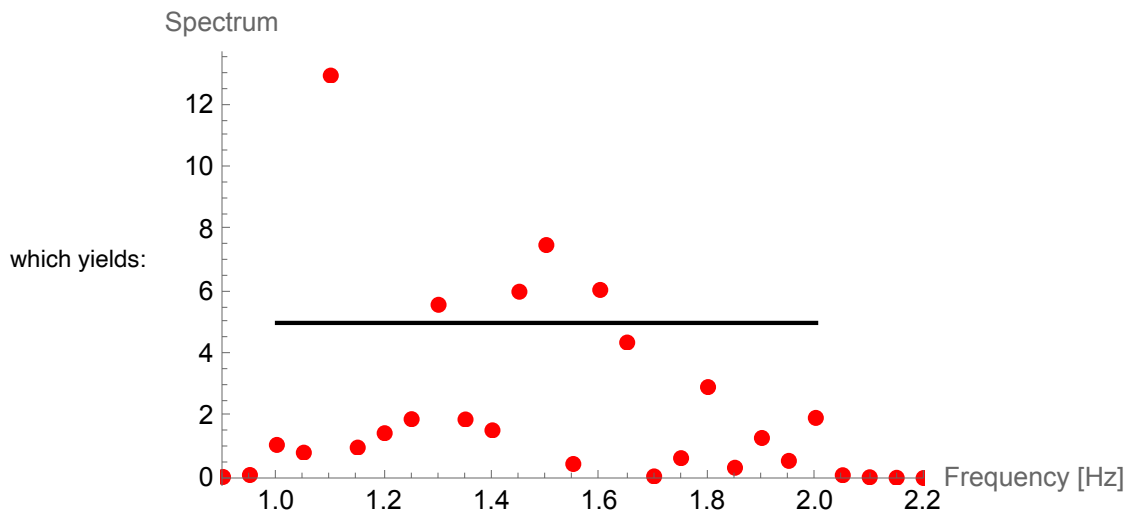
which yields: 5.93361

Apply the Fourier transform algorithm, and plot the resulting spectrum (red dots) together with the input spectrum (black line):

```

createSpectrum[timeHistory, sampleRate, totalTime];
pltIn = ListLinePlot[{{minHz, S}, {maxHz, S}}, PlotStyle → Black];
pltOut = ListPlot[Transpose[{frequencyAxis, spectrum}],
  PlotRange → {{0.9 minHz, 1.1 maxHz}, All},
  AxesLabel → {"Frequency [Hz]", "Spectrum"}, PlotMarkers → Automatic,
  PlotStyle → Red];
Show[pltOut, pltIn]

```



The area of the input spectrum is:

$$S (\text{maxHz} - \text{minHz})$$

which yields: 5

In comparison, the area of the output spectrum is (this number varies significantly between realizations):

$$\text{spectrumArea} = \text{df Total}[\text{spectrum}]$$

which yields: 3.04751

That area, averaged over many realizations, is:

```

summedArea = 0;
For[k = 0, k < 50, k++,
  createTimeHistory[S, minHz, maxHz, spectrumBlocks, sampleRate,
    totalTime, randAmpl];
  createSpectrum[timeHistory, sampleRate, totalTime];
  summedArea += df Total[spectrum];
]
Print[ $\frac{\text{summedArea}}{k}$ ];
4.79149

```

Pierson-Moskowitz Spectrum

Wind velocity at 10m above sea surface:

$$U_{10} = 20;$$

Additional constants:

$$g = 9.81;$$

$$\alpha = 8.1 \times 10^{-3};$$

$$\beta = 0.74;$$

$$\omega_0 = \frac{g}{1.026 U_{10}};$$

Minimum and maximum frequencies, for convenience of plotting and calculations:

$$\text{minHz} = 0.03;$$

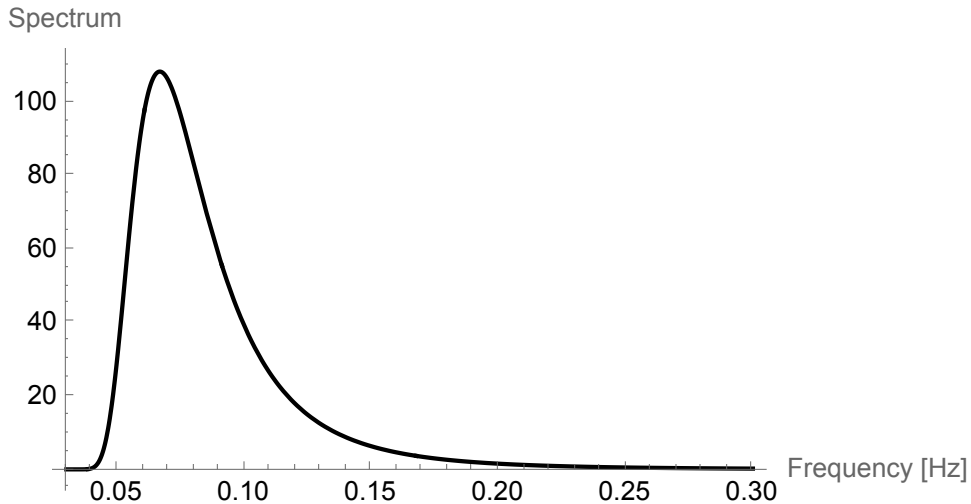
$$\text{maxHz} = 0.3;$$

Expression for the one-sided spectrum (twice the two-sided spectrum):

$$S = 2 \pi \left(\frac{\alpha g^2}{\omega^5} \text{Exp} \left[-\beta \left(\frac{\omega_0}{\omega} \right)^4 \right] \right) / . \omega \rightarrow 2 \pi f;$$

Plot spectrum:

```
pltPM = Plot[S, {f, minHz, maxHz},
  AxesLabel → {"Frequency [Hz]", "Spectrum"}, PlotStyle → Black,
  PlotRange → All]
```



Create a time history from the spectrum:

```
totalTime = 5 × 60;
spectrumBlocks = 100;
sampleRate = 50;
randAmpl = True;
createTimeHistory[S, minHz, maxHz, spectrumBlocks, sampleRate,
  totalTime, randAmpl];
ListLinePlot[Transpose[{timeAxis, timeHistory}], PlotStyle → Black,
  AxesLabel → {"Time [seconds]", "Amplitude"}, PlotRange → All]
```



Standard deviation of the process:

$$\text{stdv} = \sqrt{\int_{\text{minHz}}^{\text{maxHz}} S \, df}$$

which yields: 2.24191

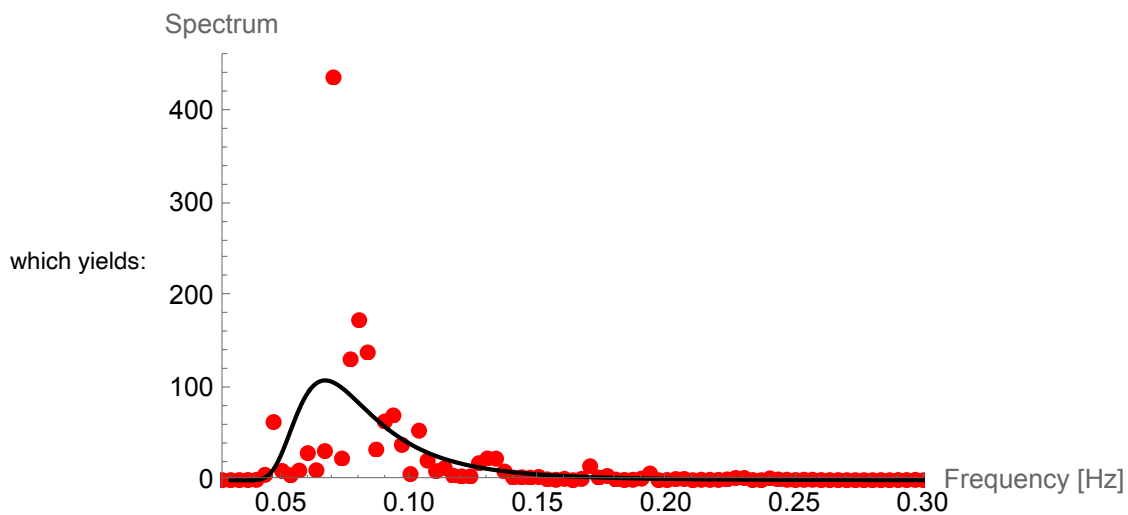
Observed extreme of the generated time history:

`Max[timeHistory]`

which yields: 6.78558

Create a spectrum from the time history:

```
createSpectrum[timeHistory, sampleRate, totalTime];
pltPMout = ListPlot[Transpose[{frequencyAxis, spectrum}],
  PlotRange → {{0.9 minHz, maxHz}, All},
  AxesLabel → {"Frequency [Hz]", "Spectrum"}, PlotMarkers → Automatic,
  PlotStyle → Red];
Show[pltPMout, pltPM]
```



Compare the area of the input spectrum...

$$\int_{\min\text{Hz}}^{\max\text{Hz}} S \, df$$

which yields: 5.02614

... with the area of the spectrum from the Fourier transform of many generated time histories:

```
summedArea = 0;
For[k = 0, k < 20, k++,
  createTimeHistory[S, minHz, maxHz, spectrumBlocks, sampleRate,
    totalTime, randAmpl];
  createSpectrum[timeHistory, sampleRate, totalTime];
  summedArea += df Total[spectrum];
]
Print[ $\frac{\text{summedArea}}{k}$ ];
5.12331
```

Kanai-Tajimi Spectrum

Expression for the one-sided spectrum (twice the two-sided spectrum):

$$S_0 = \frac{\sigma^2 \, 2 \, \xi g}{\pi \, \omega g \, (4 \, \xi g^2 + 1)};$$

$$K = 2 \, \xi g \, \omega g \, \omega;$$

$$S_{\text{gen}} = 2 \, \pi \left(S_0 \frac{(\omega g^4 + K^2)}{(\omega g^2 - \omega^2)^2 + K^2} \right) / . \, \omega \rightarrow 2 \, \pi \, f;$$

Make a parameter choice here:

$$S = S_{\text{gen}} / . \{ \sigma \rightarrow 100, \, \omega g \rightarrow 2 \, \pi \, 5, \, \xi g \rightarrow 0.4 \};$$

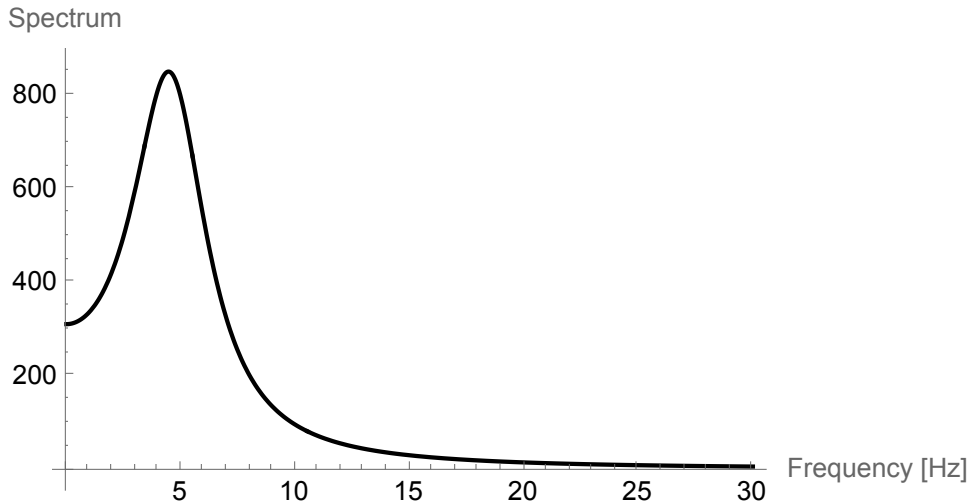
Frequency bounds, for convenience of spectrum discretization and plotting:

$$\min\text{Hz} = 0;$$

$$\max\text{Hz} = 30;$$

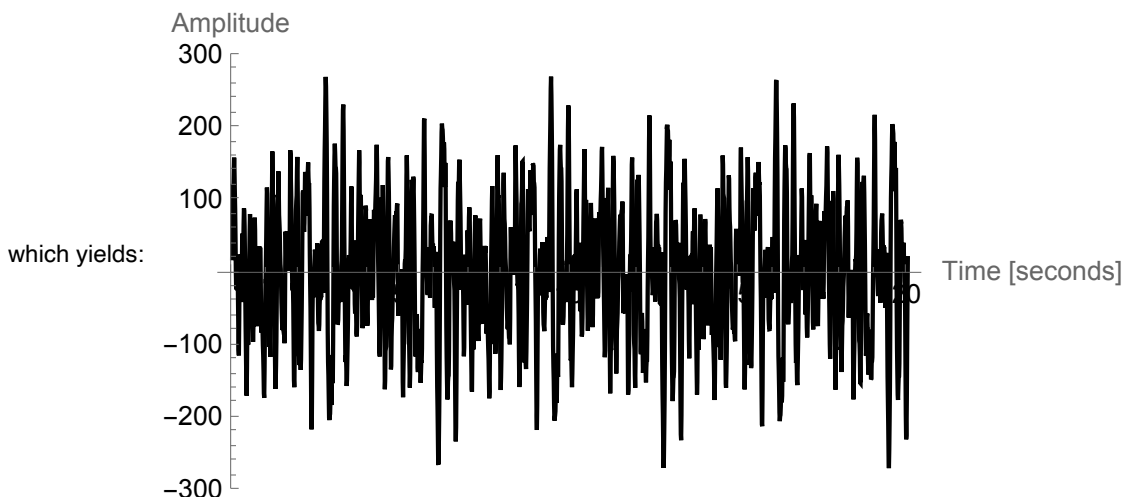
Plot spectrum:

```
pltKT = Plot[S, {f, minHz, maxHz},
  AxesLabel → {"Frequency [Hz]", "Spectrum"}, PlotStyle → Black,
  PlotRange → All]
```



Create a time history from the spectrum:

```
totalTime = 20;
spectrumBlocks = 100;
sampleRate = 100;
randAmpl = True;
createTimeHistory[S, minHz, maxHz, spectrumBlocks, sampleRate,
  totalTime, randAmpl];
ListLinePlot[Transpose[{timeAxis, timeHistory}], PlotStyle → Black,
  AxesLabel → {"Time [seconds]", "Amplitude"}, PlotRange → All]
```



Standard deviation of the process:

$$\text{stdv} = \sqrt{\text{Abs} \left[\int_{\text{minHz}}^{\text{maxHz}} S \, df \right]}$$

which yields: 69.4969

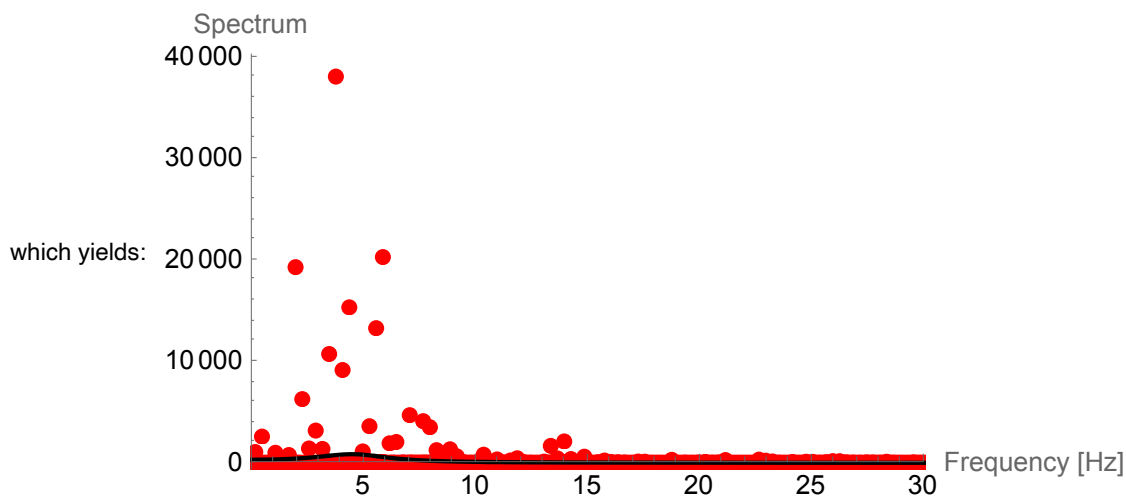
Extreme of the process:

`Max[timeHistory]`

which yields: 270.102

Create a spectrum from the time history:

```
createSpectrum[timeHistory, sampleRate, totalTime];
pltKTout = ListPlot[Transpose[{frequencyAxis, spectrum}],
  PlotRange → {{0, maxHz}, All},
  AxesLabel → {"Frequency [Hz]", "Spectrum"}, PlotMarkers → Automatic,
  PlotStyle → Red];
Show[pltKTout, pltKT]
```



Area of the ~~Kanai-Tajimi~~ spectrum:

$$\text{Abs} \left[\int_{\text{minHz}}^{\text{maxHz}} S \, df \right]$$

which yields: 4829.82

Area of the spectrum obtained by many Fourier transformations:

```
summedArea = 0;
For[k = 0, k < 50, k++,
  createTimeHistory[S, minHz, maxHz, spectrumBlocks, sampleRate,
    totalTime, randAmpl];
  createSpectrum[timeHistory, sampleRate, totalTime];
  summedArea += dfTotal[spectrum];
]
Print[ $\frac{\text{summedArea}}{k}$ ];
4838.98
```

Recorded Ground Motion (Part I)

Read time history from file, giving an acceleration time history with cm/s^2 as unit:

```
timeHistory =
  100 × 9.81
  Flatten[Import["Terjes Mathematica Files/GroundAccelerationX.txt",
    "Data"]];
```

Sampling rate for that ground motion record:

```
dt = 0.01;
sampleRate =  $\frac{1}{dt}$ ;
```

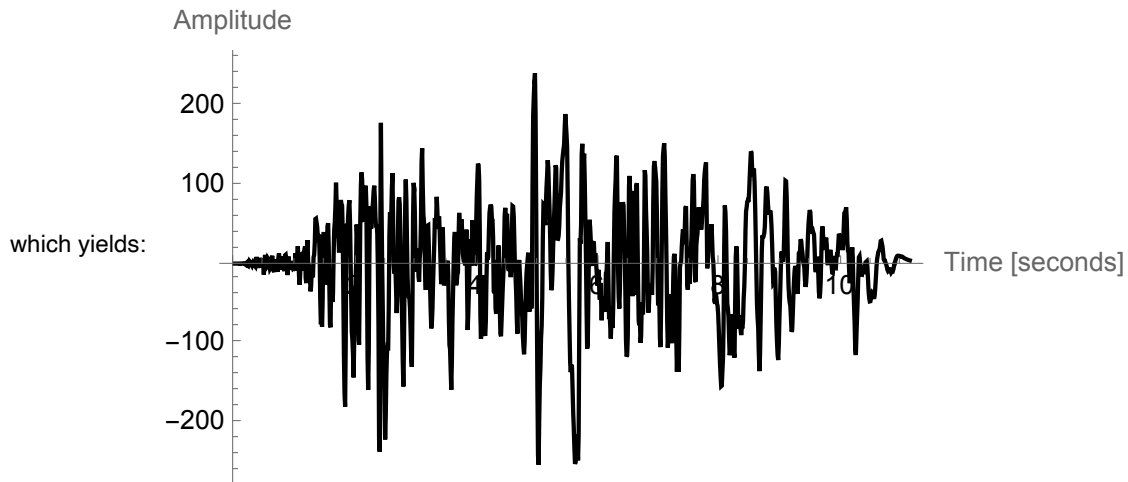
Total duration:

```
totalTime = dt (Length[timeHistory] - 1)
```

which yields: 11.14

Plot the original ground motion:

```
timeAxis = Range[0, totalTime, dt];  
ListLinePlot[Transpose[{timeAxis, timeHistory}], PlotStyle -> Black,  
  AxesLabel -> {"Time [seconds]", "Amplitude"}, PlotRange -> All]
```



Remove first and last part of the ground motion and plot it again:

```

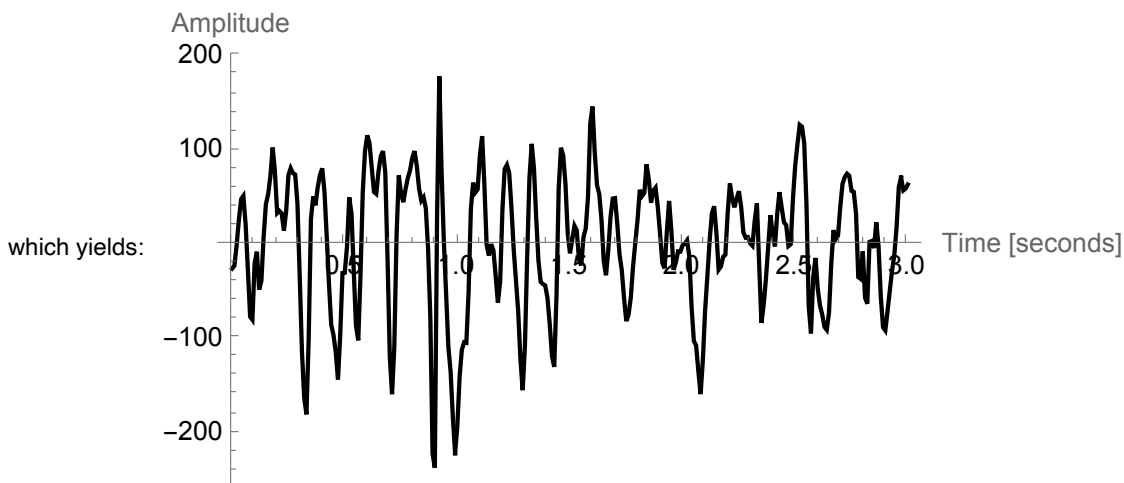
minTime = 1.5;
maxTime = 4.5;

earlyPoints = Floor[ $\frac{\text{minTime} - 0}{dt}$ ];

latePoints = Floor[ $\frac{\text{totalTime} - \text{maxTime}}{dt}$ ];

timeHistory = Drop[timeHistory, earlyPoints];
timeHistory = Drop[timeHistory, -latePoints];
totalTime = dt (Length[timeHistory] - 1);
timeAxis = Range[0, totalTime, dt];
ListLinePlot[Transpose[{timeAxis, timeHistory}], PlotStyle -> Black,
  AxesLabel -> {"Time [seconds]", "Amplitude"}, PlotRange -> All]

```



Create a spectrum from the time history:

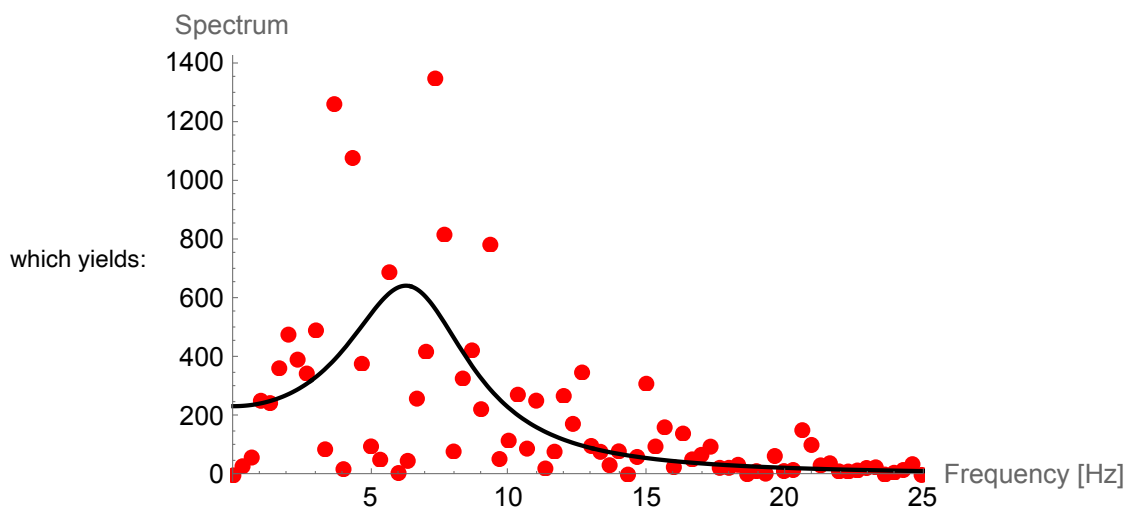
```
createSpectrum[timeHistory, sampleRate, totalTime];
```

Plot the resulting spectrum, together with a tailored ~~Kanai-Tajimi~~ spectrum:

```

minHz = 0;
maxHz = 25;
S = Sgen /. {σ → 103, ωg → 2 π 7, ξg → 0.4};
plt1 = ListPlot[Transpose[{frequencyAxis, spectrum}],
  PlotRange → {{minHz, maxHz}, All},
  AxesLabel → {"Frequency [Hz]", "Spectrum"}, PlotMarkers → Automatic,
  PlotStyle → Red];
plt2 = Plot[S, {f, minHz, maxHz},
  AxesLabel → {"Frequency [Hz]", "Spectrum"}, PlotStyle → Black];
Show[plt1, plt2]

```



Area of the Kanai-Tajimi spectrum:

$$\text{Abs} \left[\int_{\text{minHz}}^{\text{maxHz}} S \, df \right]$$

which yields: 4985.66

Area of the spectrum obtained by Fourier transformation:

$$\text{df Total[spectrum]}$$

which yields: 5026.56

Recorded Ground Motion (Part II, El Centro)

Read time history from file, giving an acceleration time history with cm/s^2 as unit:

```
timeHistory = Import["Terjes Mathematica Files/ElCentroNS.txt",
  "Data"];
timeHistory = 100 × 9.81 timeHistory[[All, 2]];
```

Parameters for that ground motion:

```
dt = 0.02;
sampleRate =  $\frac{1}{dt}$ ;
```

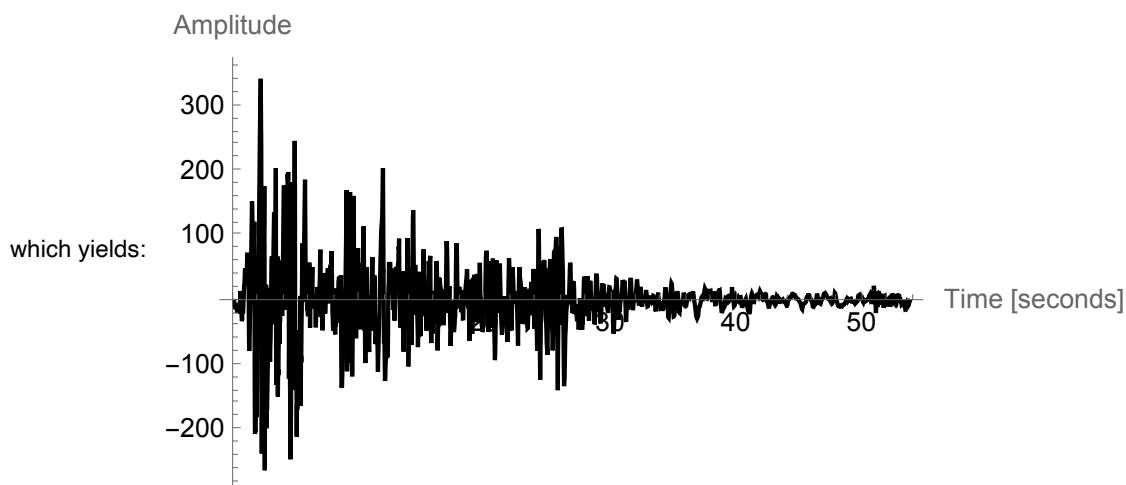
Total duration:

```
totalTime = dt (Length[timeHistory] - 1)
```

which yields: 53.74

Plot the original ground motion:

```
timeAxis = Range[0, totalTime, dt];
ListLinePlot[Transpose[{timeAxis, timeHistory}], PlotStyle → Black,
  AxesLabel → {"Time [seconds]", "Amplitude"}, PlotRange → All]
```



Remove first and last part of the ground motion and plot it again:

```

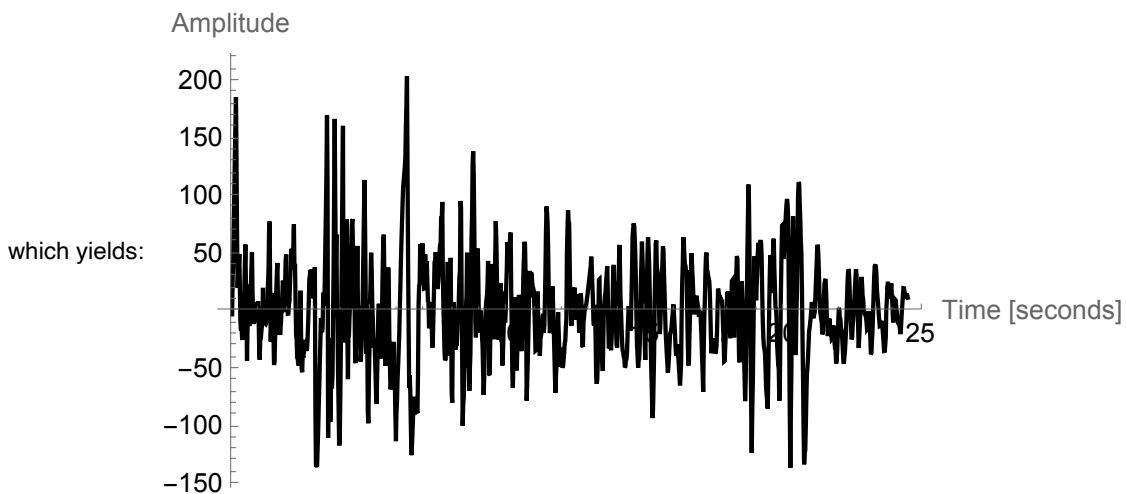
minTime = 5.5;
maxTime = 30;

earlyPoints = Floor[ $\frac{\text{minTime} - 0}{dt}$ ];

latePoints = Floor[ $\frac{\text{totalTime} - \text{maxTime}}{dt}$ ];

timeHistory = Drop[timeHistory, earlyPoints];
timeHistory = Drop[timeHistory, -latePoints];
totalTime = dt (Length[timeHistory] - 1);
timeAxis = Range[0, totalTime, dt];
ListLinePlot[Transpose[{timeAxis, timeHistory}], PlotStyle -> Black,
  AxesLabel -> {"Time [seconds]", "Amplitude"}, PlotRange -> All]

```



Create a spectrum from the time history:

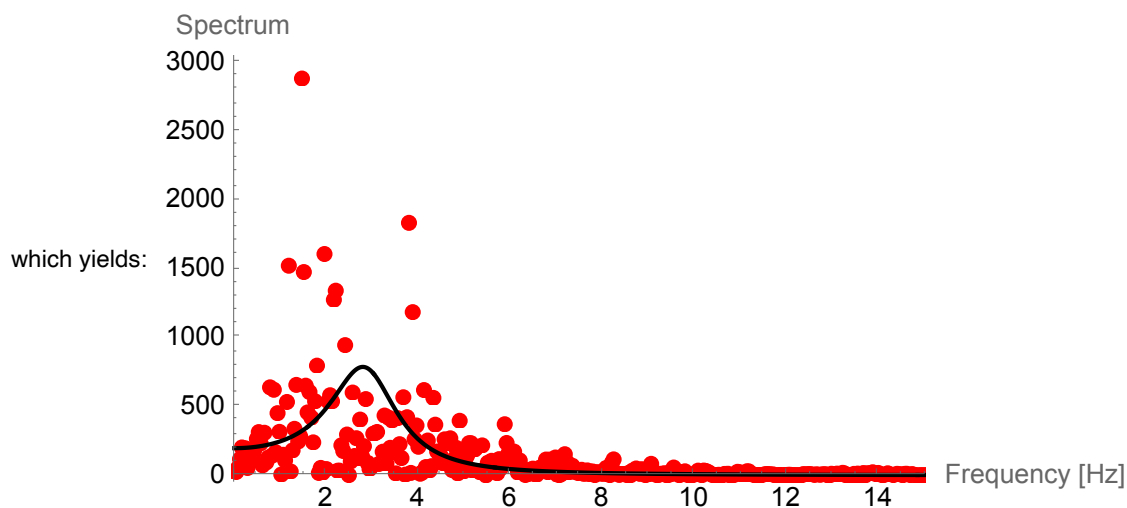
```
createSpectrum[timeHistory, sampleRate, totalTime];
```

Plot the resulting spectrum, together with a tailored ~~Kanai-Tajimi~~ Kanai-Tajimi spectrum:

```

minHz = 0;
maxHz = 15;
S = Sgen /. {σ → 65, ωg → 2 π 3, ξg → 0.3};
plt1 = ListPlot[Transpose[{frequencyAxis, spectrum}],
  PlotRange → {{minHz, maxHz}, All},
  AxesLabel → {"Frequency [Hz]", "Spectrum"}, PlotMarkers → Automatic,
  PlotStyle → Red];
plt2 = Plot[S, {f, minHz, maxHz},
  AxesLabel → {"Frequency [Hz]", "Spectrum"}, PlotStyle → Black];
Show[plt1, plt2]

```



Area of the Kanai-Tajimi spectrum:

$$\text{Abs} \left[\int_{\text{minHz}}^{\text{maxHz}} S \, df \right]$$

which yields: 2067.18

Area of the spectrum obtained by Fourier transformation:

```
df Total[spectrum]
```

which yields: 1998.13

Filtered White Noise

First define the filter associated with each pulse along the time axis. This is essentially the impulse

response function for a single-degree-of-freedom problem; not the displacement but the acceleration. That is the reason for the double-derivative:

```
dispIRF = Sin[ $\omega_n$  Sqrt[1 -  $\xi^2$ ] tt] Exp[- $\xi \omega_n$  tt];
velIRF = D[dispIRF, tt];
accelIRF = D[velIRF, tt];
Filter[t_,  $\omega_n$ _,  $\xi$ _] := If[t <= 0, 0, accelIRF /. tt -> t];
```

Specify natural frequency and damping:

```
 $\omega_n$  = 2  $\pi$  7;
 $\xi$  = 0.3;
```

Set frequency limits, for convenience of later plotting, etc.:

```
minHz = 0;
maxHz = 25;
```

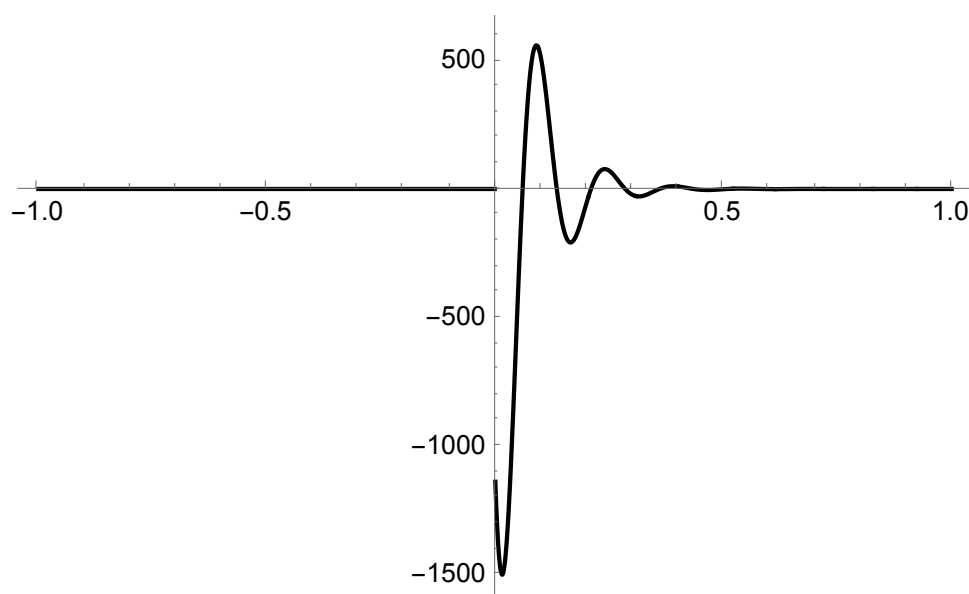
Give a time duration for some preliminary plots:

```
tempMaxTime = 1;
```

Plot the filter:

```
Plot[Filter[time,  $\omega_n$ ,  $\xi$ ], {time, -tempMaxTime, tempMaxTime},
PlotRange -> All, PlotStyle -> Black]
```

which yields:



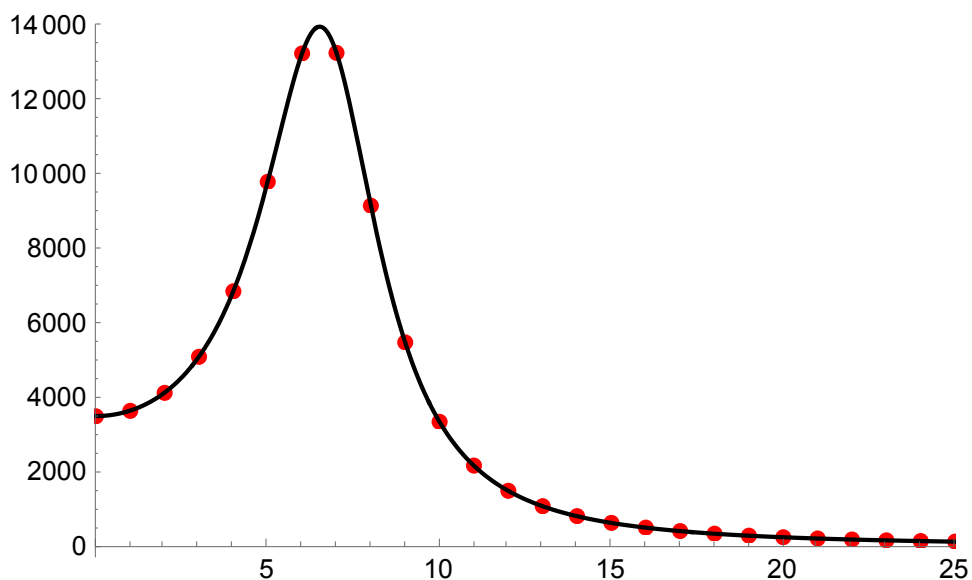
Do a Fourier transform of the filter, and plot the result on top of the ~~Kanai-Tajimi~~ spectrum:

```

sampleRate = 10 000;
dt =  $\frac{1}{\text{sampleRate}}$ ;
timeHistory = Table[N[{t, Filter[t,  $\omega_n$ ,  $\xi$ ]}], {t, 0, tempMaxTime, dt}];
createSpectrum[timeHistory[[All, 2]], sampleRate, tempMaxTime];
pltFourier = ListPlot[Transpose[{frequencyAxis, spectrum}],
  PlotRange → {{minHz, maxHz}, All}, PlotMarkers → Automatic,
  PlotStyle → Red];
spectrumMax = Max[spectrum];
locSpectrumMax = Position[spectrum, Max[spectrum]];
S = Sgen /. { $\sigma \rightarrow 1$ ,  $\omega_g \rightarrow \omega_n$ ,  $\xi_g \rightarrow \xi$ };
scale =  $\frac{\text{spectrumMax}}{S /. f \rightarrow \text{frequencyAxis}[[\text{locSpectrumMax}[[1, 1]]]]}$ ;
pltKT = Plot[scale S, {f, minHz, maxHz},
  AxesLabel → {"Frequency [Hz]", "Spectrum"}, PlotRange → All,
  PlotStyle → Black];
Show[pltFourier, pltKT]

```

which yields:



Area of those two curves, just as a check:

$$\text{Abs} \left[\int_{\text{minHz}}^{\text{5 maxHz}} \text{scale S df} \right]$$

which yields: 87 178.4

$$\text{df Total}[\text{spectrum}]$$

which yields: 89 432.3

As a side note, compare the expression of the transfer function for displacements:

$$\frac{\omega_n^4}{(\omega_n^2 - \omega^2)^2 + (2 \xi \omega_n \omega)^2}$$

... to the ~~Kanai~~ Kanai-Tajimi filter for the acceleration:

$$\frac{\omega_n^4 + (2 \xi \omega_n \omega)^2}{(\omega_n^2 - \omega^2)^2 + (2 \xi \omega_n \omega)^2};$$

... noticing the added term in the numerator for the spectrum of the acceleration. Now, set the total time duration of the ground motion that will soon be generated:

$$\text{totalTime} = 20;$$

A trapezoidal modulating function:

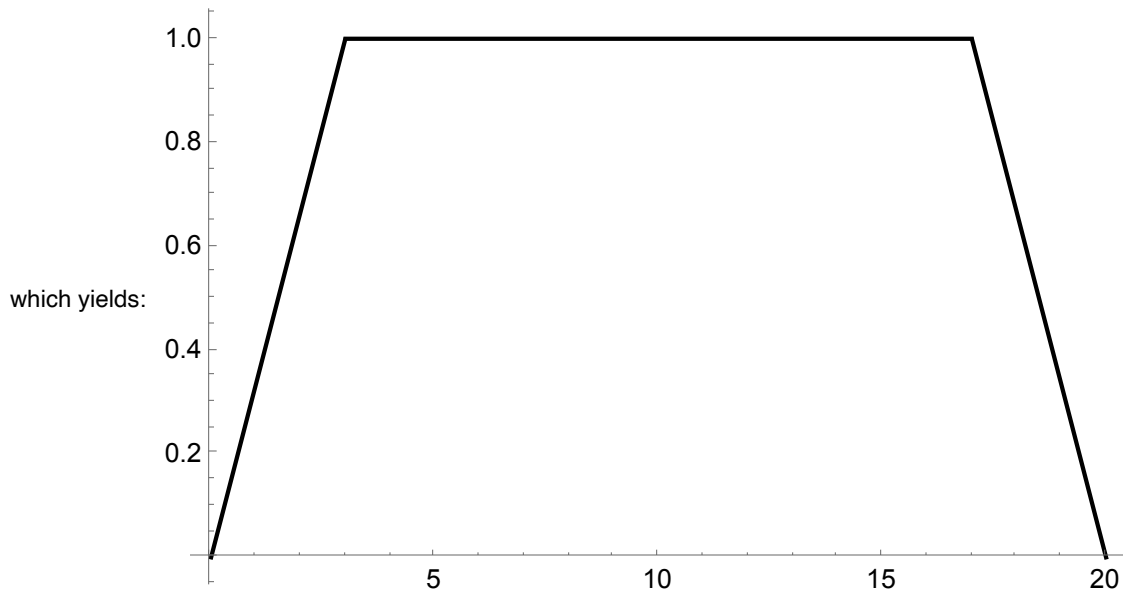
$$\begin{aligned} \text{Modulator}[t_ , c_ , t1_ , t2_ , t3_ , t4_] := \\ \text{If} \left[t < t1, 0, \text{If} \left[t < t2, \frac{c (t - t1)}{t2 - t1}, \right. \right. \\ \left. \left. \text{If} \left[t < t3, c, \text{If} \left[t < t4, c - \frac{c (t - t3)}{t4 - t3}, 0 \right] \right] \right] \right]; \end{aligned}$$

Properties of the modulating function:

$$\begin{aligned} \text{rampTime} &= 0.15 \text{ totalTime}; \\ t1 &= 0; \\ t2 &= \text{rampTime}; \\ t3 &= \text{totalTime} - \text{rampTime}; \\ t4 &= \text{totalTime}; \end{aligned}$$

Plot the modulating function:

```
Plot[Modulator[time, 1, t1, t2, t3, t4], {time, 0, totalTime},
PlotRange → All, PlotStyle → Black]
```



Specify the sampling rate to be used in the following example:

```
sampleRate = 100;
dt =  $\frac{1}{\text{sampleRate}}$ ;
```

Specify the pulse rate:

```
pulseRate = 50;
pulseSpacing =  $\frac{1}{\text{pulseRate}}$ ;
```

Set the target variance:

```
targetVariance = 5000;
```

Calculate the variance of the unfactored and deterministic time history:

```

totalNumPulses = totalTime pulseRate;
summedVariance = 0;
For [ i = 0, i < totalNumPulses, i ++,
    kickTime = i pulseSpacing;
    summedVariance +=  $\left( \text{Filter} \left[ \frac{\text{totalTime}}{2} - \text{kickTime}, \omega_n, \xi \right] \right)^2$ ;
];

```

The scaling factor to get the target variance is therefore:

$$\text{varianceScaling} = \frac{\text{targetVariance}}{\text{summedVariance}};$$

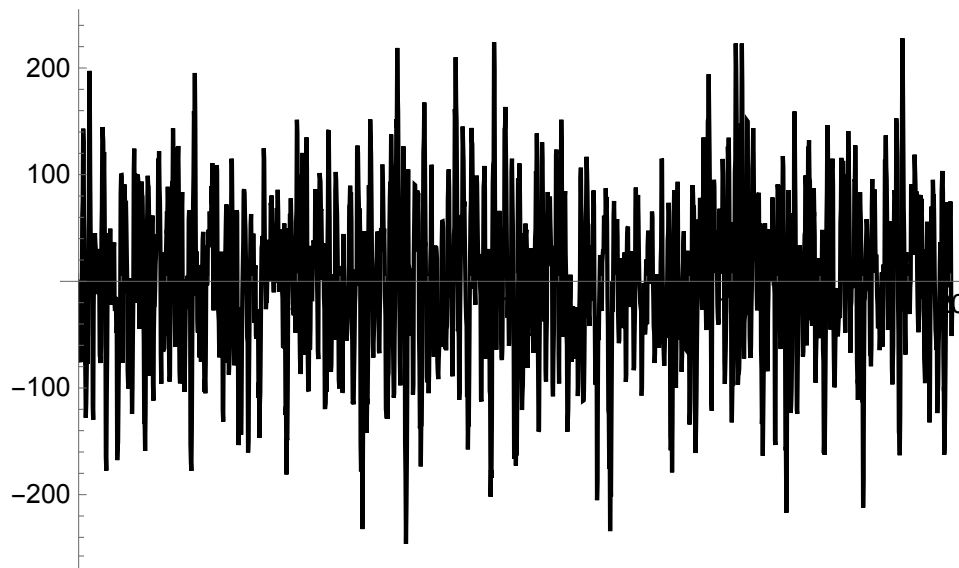
Then the generation of the actual time history:

```

nintervals = totalTime sampleRate;
timeHistory = ConstantArray[0, nintervals + 1];
For[i = 0, i < totalNumPulses, i++,
  pulse = RandomVariate[NormalDistribution[0,  $\sqrt{\text{varianceScaling}}$ ]];
  kickTime = i pulseSpacing;
  For[j = 0, j <= nintervals, j++,
    t = j dt;
    timeHistory[[j + 1]] += pulse Filter[t - kickTime,  $\omega_n$ ,  $\xi$ ];
  ];
];
timeAxis = Range[0, totalTime, dt];
ListLinePlot[Transpose[{timeAxis, timeHistory}], PlotRange → All,
  PlotStyle → Black]

```

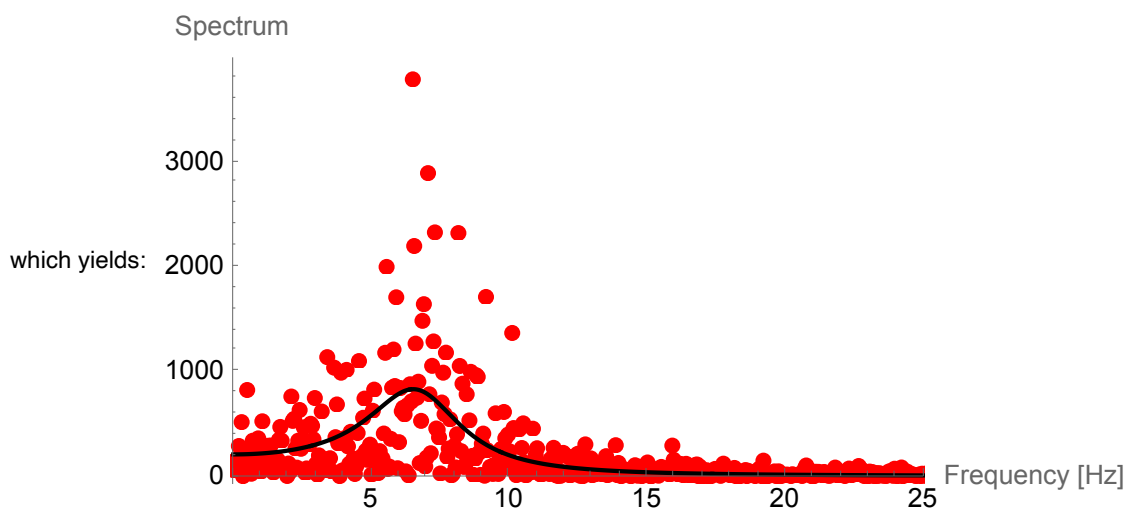
which yields:



```

createSpectrum[timeHistory, sampleRate, totalTime];
plt1 = ListPlot[Transpose[{frequencyAxis, spectrum}],
  PlotRange → {{minHz, maxHz}, All},
  AxesLabel → {"Frequency [Hz]", "Spectrum"}, PlotMarkers → Automatic,
  PlotStyle → Red];
S = Sgen /. {σ → 102, ωg → ωn, ξg → ξ};
plt2 = Plot[S, {f, minHz, maxHz},
  AxesLabel → {"Frequency [Hz]", "Spectrum"}, PlotStyle → Black];
Show[plt1, plt2]

```



Area of that ~~Kanai~~-Tajimi spectrum:

$$\text{Abs} \left[\int_{\min\text{Hz}}^{\max\text{Hz}} S \, df \right]$$

which yields: 5036.53

Area of the spectrum obtained by Fourier transformation, using several realizations:

```

summedArea = 0;
For[k = 0, k < 20, k++,
  timeHistory = ConstantArray[0, nintervals + 1];
  For[i = 0, i < totalNumPulses, i++,
    pulse = RandomVariate[NormalDistribution[0,  $\sqrt{\text{varianceScaling}}$ ]];
    kickTime = i pulseSpacing;
    For[j = 0, j <= nintervals, j++,
      t = j dt;
      timeHistory[[j + 1]] += pulse Filter[t - kickTime,  $\omega_n$ ,  $\xi$ ];
    ];
  ];
createSpectrum[timeHistory, sampleRate, totalTime];
summedArea += df Total[spectrum];
]
Print[ $\frac{\text{summedArea}}{k}$ ];

5612.71

```