

(Fast) Fourier Transform of Discrete Time Series

Consider a set of experimental data, obtained at M uniformly spaced intervals, $t_n = n \Delta t$, where $n = 0, 1, 2, \dots, M-1$. The total time is

$$T = M \Delta t$$

In what follows we shall set the time interval between measurements, Δt , to be 1, so $T = M$, and fix

$$T = M = 128.$$

We shall obtain the power spectrum by Fourier transforming the data.

Here we assume that the data is a damped oscillation with its maximum at $T/2$. The damping constant will be γ and the frequency ω_0 . We also allow for a phase factor ϕ , though this will make very little difference.

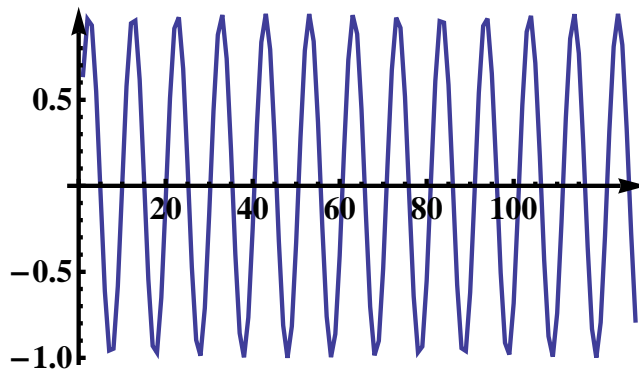
```
timeseries := Table[Exp[- $\gamma$  Abs[t - T/2]] Sin[ $\omega_0$  t +  $\phi$ ], {t, 0, M-1}];
```

Initially we will assume no damping, and put in a frequency *incommensurate* with the total time T (i.e. $\omega_0 T$ is not an integer multiple of 2π):

$$\Delta t = 1; M = 128; T = M \Delta t; \gamma = 0.0; \phi = 0.7; \omega_0 = 0.62;$$

Plotting the data, we see the expected pure sine wave

```
ListPlot[timeseries, Joined  $\rightarrow$  True]
```



Next we compute the power spectrum which is the square of the absolute value of the Fourier transform (the *Mathematica* function **Fourier** does the Fast Fourier Transform (FFT)):

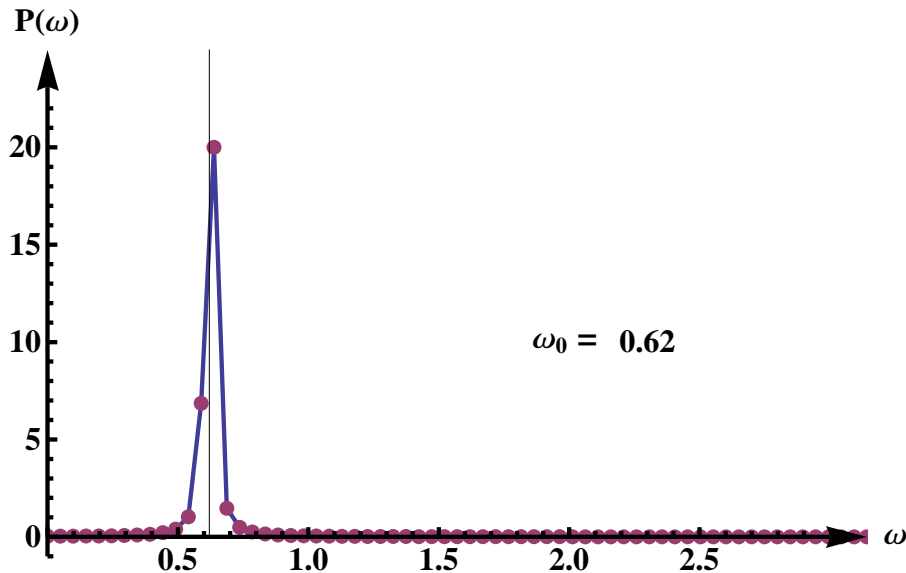
```
powerspectrum = Abs[Fourier[timeseries]]^2;
```

The frequency values are $2\pi n/T$, where n is an integer with $0 \leq n \leq M-1$ (or equivalently any other range of M contiguous values such as $-M/2 < n \leq M/2$):

```
omegaval = Table[ $2\pi$  t / T, {t, 0, M-1}];
```

We next plot the power spectrum. For $\Delta t = 1$, the range of frequencies is from $-\pi$ to π . However, the power spectrum is an even function of ω so we only plot the range from 0 to π .

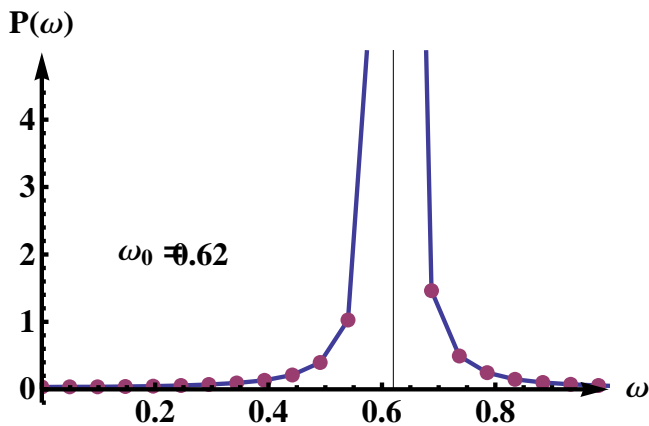
```
ListPlot[Transpose[{omegavals, powerspectrum}], Transpose[{omegavals, powerspectrum}]],
Joined -> {True, False}, PlotRange -> {{0,  $\pi$ }, {-1, 25}}, AxesLabel -> {" $\omega$ ", " $P(\omega)$ "},
Epilog -> {Text[" $\omega_0 =$  ", {2, 10}], Text[" $\omega_0$ ", {2.3, 10}], Line[{{0.62, 0}, {0.62, 25}}]}
```



We see the expected peak at ω_0 but there is also a signal at neighboring frequencies. This is because the time series is not a perfect sine wave since the values at $t = 0$ and T are different and so (bearing in mind that the function represented by the FFT is actually a periodic continuation of the time series) there is a discontinuity when t is an integer multiple of T .

The values of the power spectrum at frequencies neighboring ω_0 are shown in the next figure which is a blowup of the relevant region (the vertical line indicates ω_0)

```
ListPlot[Transpose[{omegavals, powerspectrum}], Transpose[{omegavals, powerspectrum}]],
Joined -> {True, False}, PlotRange -> {{0, 1}, {-0.2, 5}}, AxesLabel -> {" $\omega$ ", " $P(\omega)$ "},
Epilog -> {Text[" $\omega_0 =$  ", {0.2, 2}], Text[" $\omega_0$ ", {0.28, 2}], Line[{{0.62, 0}, {0.62, 25}}]}
```

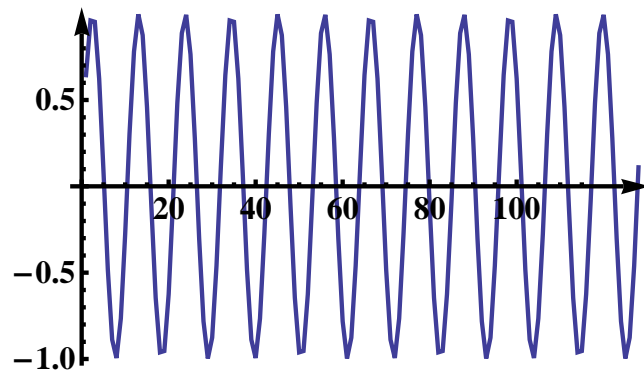


However, if we choose a frequency which is *commensurate* with total time T there is no discontinuity at the boundaries and we get *only* a signal at ω_0 .

```
 $\omega_0 = 12 (2 \pi / T);$ 
```

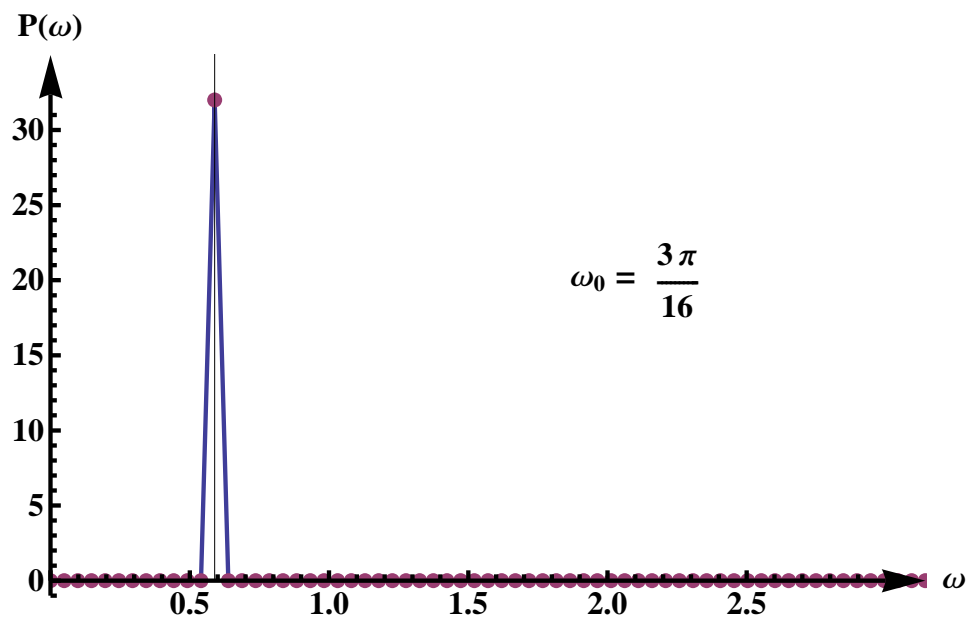
```
timeseries = Table[Exp[- $\gamma$  Abs[t - T/2]] Sin[ $\omega_0 t + \phi$ ], {t, 0, M-1}];
```

```
ListPlot[timeseries, Joined → True]
```

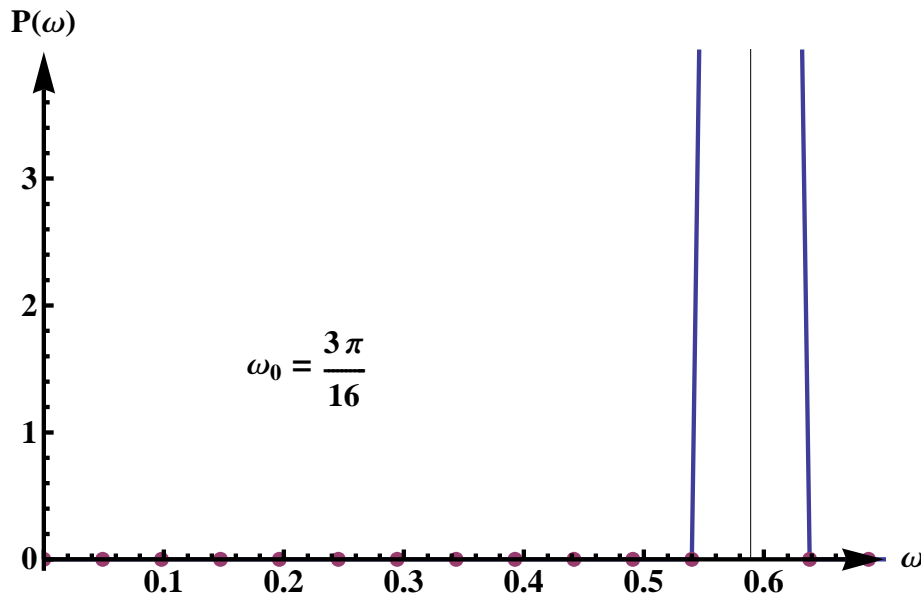


```
powerspectrum = Abs[Fourier[timeseries]]^2;
```

```
ListPlot[{Transpose[{omegavals, powerspectrum}], Transpose[{omegavals, powerspectrum}]},
  Joined → {True, False}, PlotRange → {{0,  $\pi$ }, {-1, 35}}, AxesLabel → {" $\omega$ ", " $P(\omega)$ "},
  Epilog → {Text[" $\omega_0 =$  ", {2, 20}], Text[" $\omega_0$ ", {2.25, 20}], Line[{3  $\pi$  / 16, 0}, {3  $\pi$  / 16, 40}]}}
```



```
ListPlot[Transpose[{omegavals, powerspectrum}], Transpose[{omegavals, powerspectrum}]],
Joined -> {True, False}, PlotRange -> {{0, 0.7}, {-0.09, 4}},
AxesLabel -> {" $\omega$ ", "P( $\omega$ )", Epilog ->
  {Text[" $\omega_0 =$ ", {0.2, 1.5}], Text[" $\omega_0$ ", {0.25, 1.5}], Line[{3  $\pi$ /16, 0}, {3  $\pi$ /16, 5}]}}]
```



We see that all values for the power spectrum are zero except for $\omega = \omega_0$.

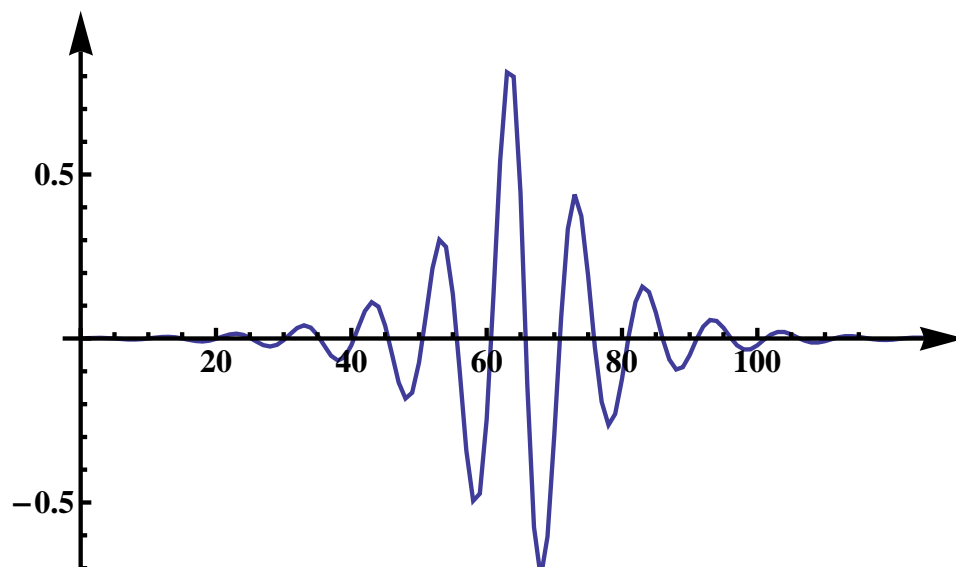
Now let's put in some damping, and go back to the original (incommensurate) frequency

```
 $\gamma = 0.1$ ;  $\omega_0 = 0.62$ ;
```

```
timeseries = Table[Exp[- $\gamma$  Abs[t - T/2]] Sin[ $\omega_0$  t +  $\phi$ ], {t, 0, M-1}];
```

The time series shows oscillations decaying about the middle of the range

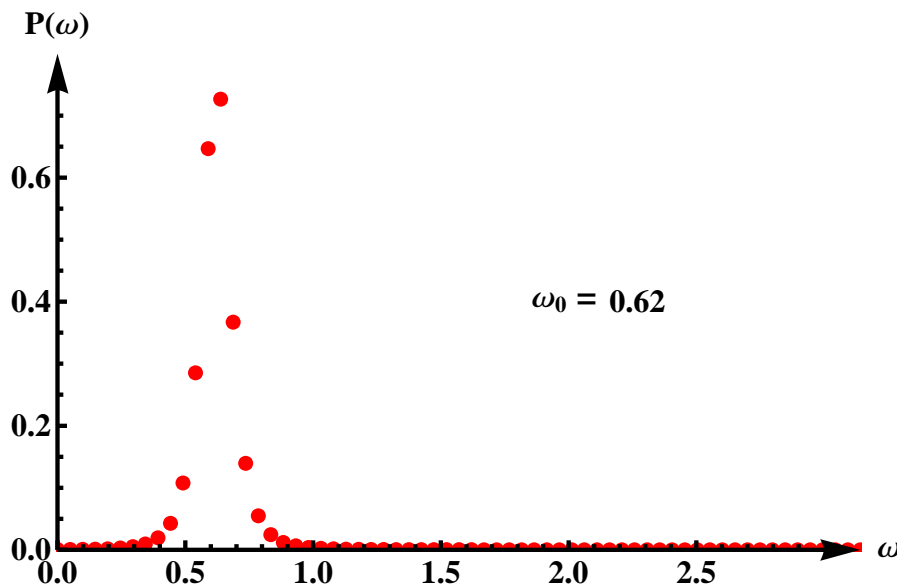
```
ListPlot[timeseries, PlotRange -> {-0.7, 1}, Joined -> True]
```



```
powerspectrum = Abs[Fourier[timeseries]]^2;
```

The power spectrum (shown by the points in the figure below) is centered on ω and has a finite width.

```
discrete = ListPlot[Transpose[{omegavals, powerspectrum}], Joined -> False,
  PlotRange -> {{0,  $\pi$ }, {0, 0.8}}, PlotStyle -> Directive[Red, PointSize[Large]],
  AxesLabel -> {" $\omega$ ", "P( $\omega$ )"}, Epilog -> {Text[" $\omega_0 =$ ", {2, 0.4}], Text[" $\omega_0$ ", {2.27, 0.4}]}]
```

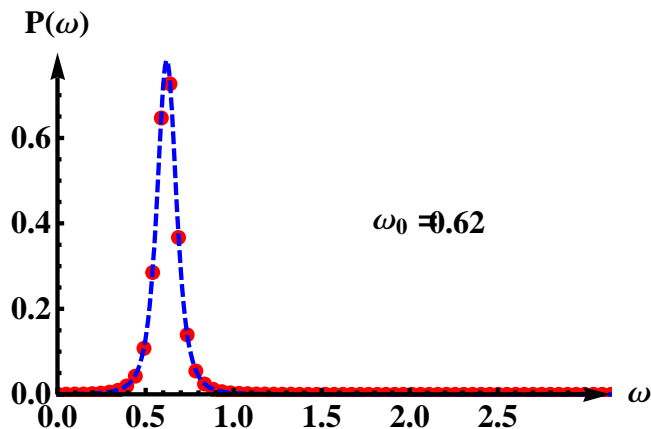


However, unlike the above case with no damping, where the width was an artifact of the finite length of the time series, here the width is *intrinsic* and due to the damping. To see this we compute the power spectrum for a damped oscillator in continuous time extending up to $T = \infty$ (a Lorentzian)

```
continuous = Plot[( $\gamma / ((\omega - \omega_0)^2 + \gamma^2)$ )^2 / M,
  { $\omega$ , 0,  $\pi$ }, PlotRange -> {0, 0.8}, PlotStyle -> {Blue, Dashed, Thick}];
```

In order to compare with the power spectrum of the discrete FT we divide by M (the number of points) because *Mathematica* puts in a factor of $1/\sqrt{M}$ in the definition of both the direct and indirect FT (i.e. making the two equivalent) rather than a factor of 1 in the direct FT and $1/N$ in the inverse FT as discussed in class. We show the power spectrum for continuous time by the dashed line on the same plot as the discrete FFT :

```
Show[discrete, continuous]
```



The two agree extremely well.

I conclude with some words of caution on Fourier Transforms of discrete time series.

1. The first difficulty comes because the data is at *discrete times* $m \Delta t$. This means that the range of frequencies is limited to $-\pi / \Delta t < \omega \leq \pi / \Delta t$. If the data has frequencies outside this range they will be mapped on to a frequency inside this range by a shift of an integer multiple of $2 \pi / \Delta t$. (Note that, at the discrete times $m \Delta t$, data for frequencies ω and $\omega + 2 \pi n / \Delta t$ are identical). This effect is called *aliasing*. Clearly the Fourier Transform is misleading if the physical phenomena take at higher frequencies than those which can be represented by data at the discrete times.

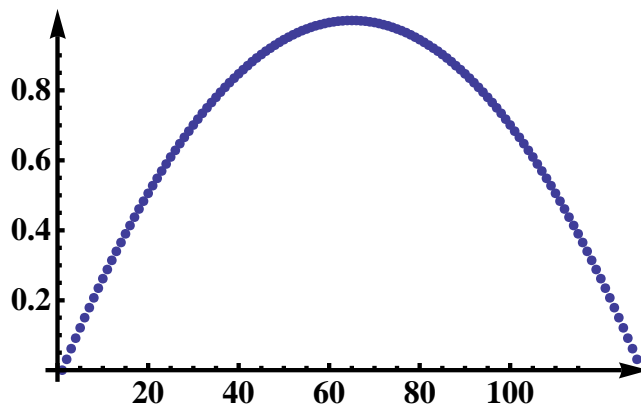
2. The second difficulty comes because the data is only over a *finite time* T , and the data at one end point will not, in general, match up with data at the other end point. The FFT actually represents a periodic continuation of the data, and so there is a discontinuity at multiples of T which, as we have seen, puts weight in neighboring Fourier components.

To help alleviate this problem one often multiplies the data by a *window function* $w(t_n)$. This is designed to be relatively flat in the central region and to fall to zero for $t \rightarrow 0$ and $t \rightarrow T$. Many window functions have been proposed. Perhaps the simplest is the Welch window

$$w(t_n) = 1 - \left(\frac{(n - M/2)}{M/2} \right)^2$$

which is just a parabola which vanishes at $n = 0$ and M , i.e. $t_n = 0$ and T .

```
welch = Table[1 - ((n - M/2) / (M/2))^2, {n, 0, M-1}];
ListPlot[welch, PlotStyle -> PointSize[Medium]]
```

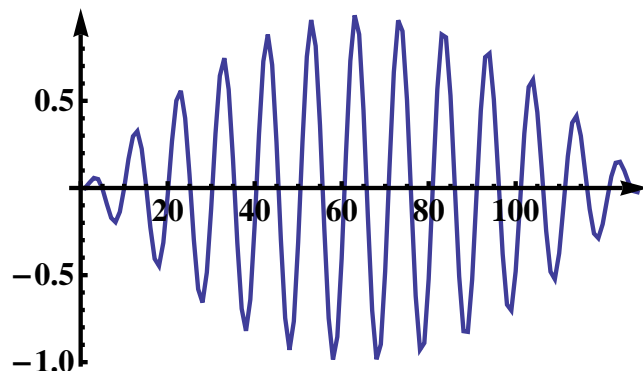


Let's go back to the undamped sine wave, and multiply the data by the Welch window.

```
Δt = 1; M = 128; T = MΔt; γ = 0.0; φ = 0.7; ω0 = 0.62;
timeseries = Table[Exp[-γ Abs[t - T/2]] Sin[ω0 t + φ], {t, 0, M-1}];
timeseries = timeseries * welch;
```

We plot the resulting time series.

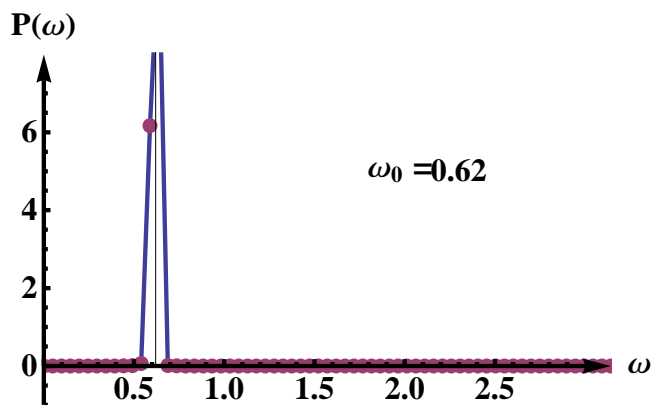
```
ListPlot[timeseries, Joined → True]
```



As expected it tends to zero at the two ends. Computing the power spectrum gives

```
powerspectrum = Abs[Fourier[timeseries]]^2;

ListPlot[{Transpose[{omegavals, powerspectrum}], Transpose[{omegavals, powerspectrum}]},
  Joined → {True, False}, PlotRange → {{0,  $\pi$ }, {-1, 8}}, AxesLabel → {" $\omega$ ", "P( $\omega$ )"},
  Epilog → {Text[" $\omega_0 =$ ", {2, 5}], Text[" $\omega_0$ ", {2.3, 5}], Line[{{0.62, 0}, {0.62, 25}}]}}
```



Blowing up the region near the peak we see that much less spectral weight spreads out onto neighboring frequencies than without the window function.

```
ListPlot[{Transpose[{omegavals, powerspectrum}], Transpose[{omegavals, powerspectrum}]},
  Joined → {True, False}, PlotRange → {{0, 1}, {-0.2, 1.2}}, AxesLabel → {" $\omega$ ", "P( $\omega$ )"},
  Epilog → {Text[" $\omega_0 =$ ", {0.2, 0.5}], Text[" $\omega_0$ ", {0.32, 0.5}], Line[{{0.62, 0}, {0.62, 25}}]}}
```

