DFTAnalyze[] Examples

Craig Stuart Sapp <craig@ccrma.stanford.edu>

14 April 1997

DFTAnalyze[]

setup

```
Needs@"SCMTheory‘"D
```

```
Options@DFTAnalyzeD
```

```
8SignalOptions Æ 8Causal Æ True<, SpectrumOptions Æ 8Causal Æ False<,
 SignalPlot Æ RealImaginaryPlot, SpectrumPlot Æ MagnitudePhasePlot<
```

The DFTAnalyze function actually uses two other function, (1) RealImaginaryPlot to plot the signal and (2) MagnitudePhasePlot to plot the spectrum

```
Options@RealImaginaryPlotD
```

```
8Axes Æ 8True, None<, LeftOptions Æ 8<, RightOptions Æ 8<<
```

```
Options@MagnitudePhasePlotD
```

```
8LeftOptions Æ 8<, RightOptions Æ 8<<
```

The RealImaginaryPlot in turn, uses the RealPlot and the ImaginaryPlot functions:

```
Options@RealPlotD
```

```
8Frame Æ True, Axes Æ 8Automatic, None<, Interpolate Æ True,
 Stretch Æ 1, Lines Æ False, InterpStyle Æ Thickness@0.D, PlotRange Æ All,
 FrameLabel Æ 8, , FontForm@Real Part, 8Times-Roman, 18<D, <<
```

```
Options@ImaginaryPlotD
```

```
8Frame Æ True, Axes Æ 8Automatic, None<, Interpolate Æ True, InterpStyle Æ Thickness@0.D,
 FrameLabel Æ 8, , FontForm@Imaginary Part, 8Times-Roman, 18<D, <, Stretch Æ 1,
 Lines Æ False<
```

The MagnitudePhasePlot function uses the MagnitudePlot and PhasePlot functions:

```
Options@MagnitudePlotD
```

```
8LogScale Æ True, Frame Æ True, Axes Æ False, Interpolate Æ True,
 InterpStyle Æ Thickness@0.D, FrameLabel Æ 8, , FontForm@Magnitude, 8Times-Roman, 18<D, <,
 Stretch Æ 1, Lines Æ False<
```

```
Options@PhasePlotD
```

```
8Frame Æ True, Axes Æ 8True, None<, PlotRange Æ All, Interpolate Æ True,
 InterpStyle Æ Thickness@0.D, Stretch Æ 1, Wrap Æ True, Frame Æ True,
 Lines Æ False, PointStyle Æ PointSize@0.025D, BreakStyle Æ Dashing@80.005, 0.03<D,
 FrameLabel Æ 8, , FontForm@Phase, 8Times-Roman, 18<D, <<
```

All of these plotting functions are variations on the SeqPlot function, and had to be created in order to create an interpolated line correctly when taking just the real, imaginary, magnitude, or phase of a signal.

```
Options@SeqPlotD
```

```
8Points Æ True, Lines Æ True, PointStyle Æ PointSize@0.03D, LineStyle Æ Thickness@0.012D,
 Repeat Æ 1, Causal Æ True, TickPeriod Æ Automatic, Continuous Æ False,
 PlotRange Æ All, Stretch Æ 1, PadFactor Æ 12, InterpStyle Æ Thickness@0.02D,
 Interpolate Æ False, InterpGraphic Æ Line<
```

Options from the built-in function Plot also work with these plotting functions:

```
Options@PlotD
```

```
9AspectRatio Æ 1/GoldenRatio, Axes Æ Automatic, AxesLabel Æ None, AxesOrigin Æ Automatic,
 AxesStyle Æ Automatic, Background Æ Automatic, ColorOutput Æ Automatic, Compiled Æ True,
 DefaultColor Æ Automatic, Epilog Æ 8<, Frame Æ False, FrameLabel Æ None,
 FrameStyle Æ Automatic, FrameTicks Æ Automatic, GridLines Æ None, MaxBend Æ 10.,
 PlotDivision Æ 20., PlotLabel Æ None, PlotPoints Æ 25, PlotRange Æ Automatic,
 PlotRegion Æ Automatic, PlotStyle Æ Automatic, Prolog Æ 8<, RotateLabel Æ True,
 Ticks Æ Automatic, DefaultFont ¶ $DefaultFont, DisplayFunction ¶ $DisplayFunction=
```

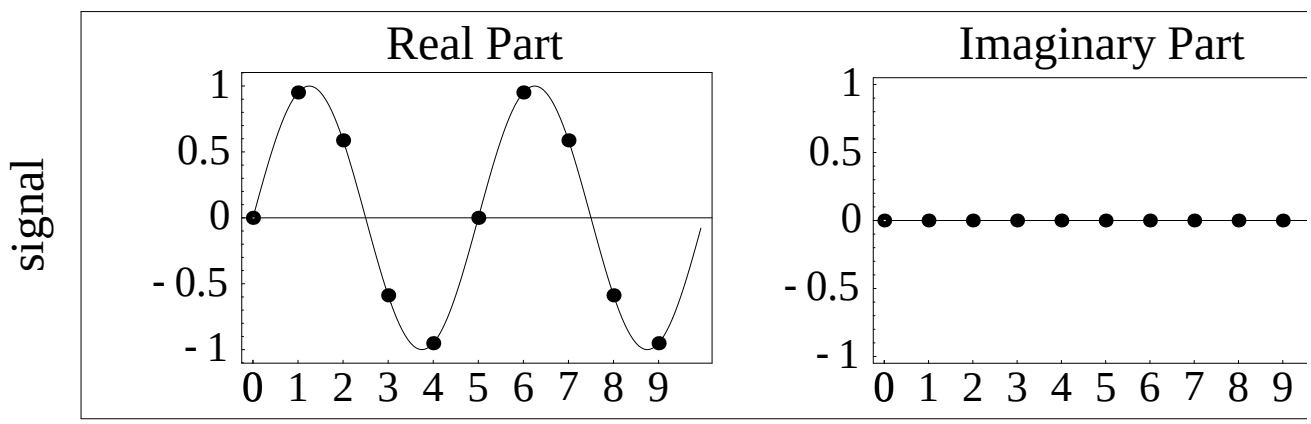
Default Style

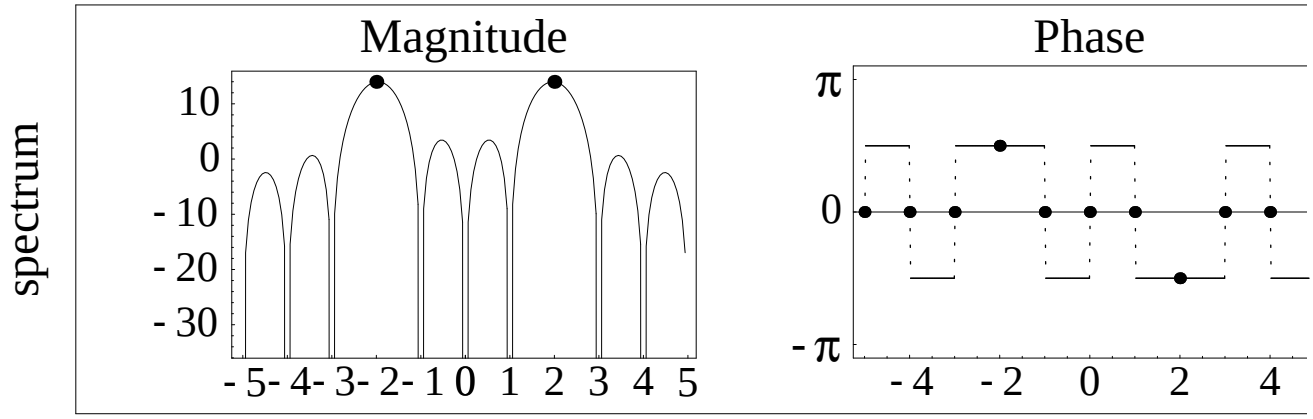
Sample a sine wave for two periods, with a total of 10 samples:

In[10]:=
```
sig = NASampleFunctionASinA 2 π 2 x / 10 E, 8x, 0, 9<EE
```

Out[10]=
```
80, 0.951057, 0.587785, -0.587785, -0.951057, 0, 0.951057, 0.587785, -0.587785,
 -0.951057<
```

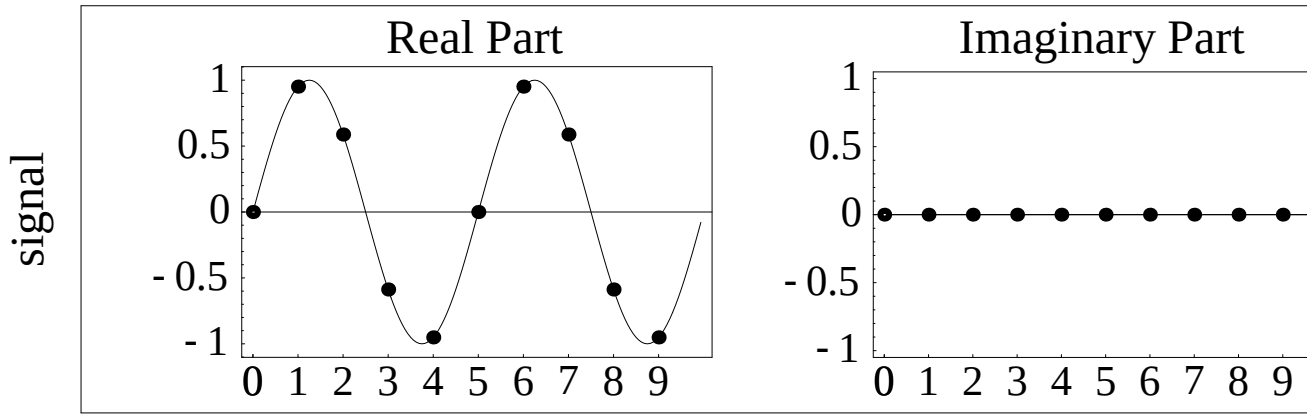
In[11]:=
```
DFTAnalyze@sigD;
```

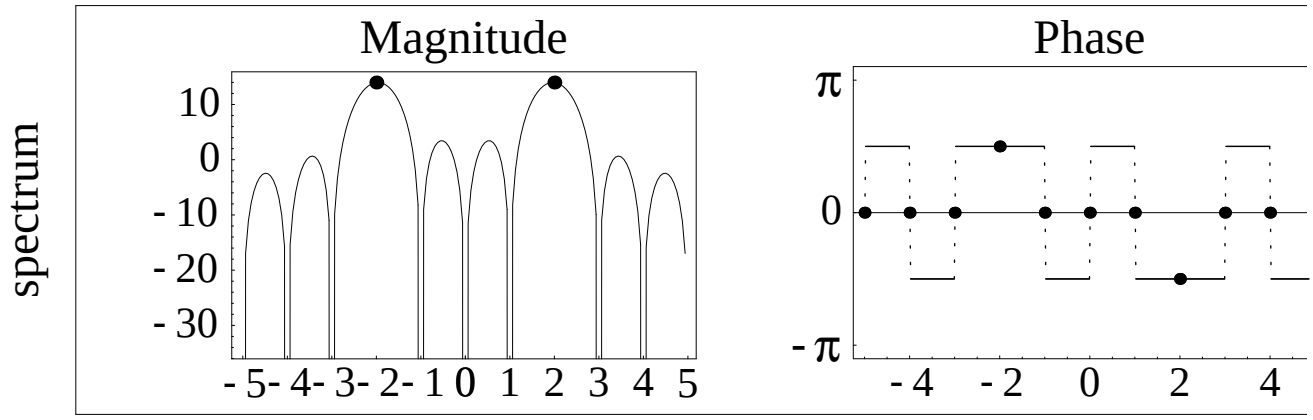




Making the spectrum display only positive frequency

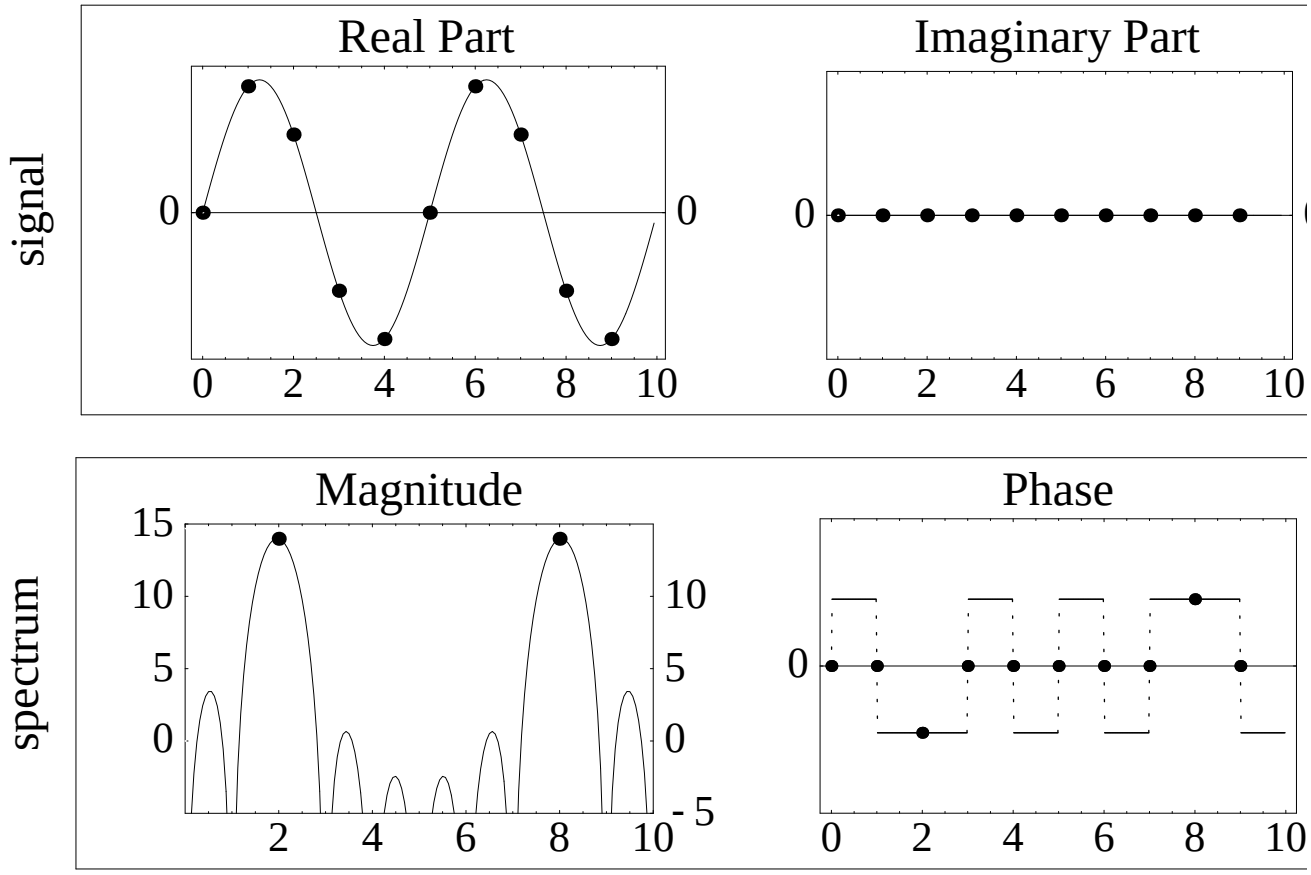
```
DFTAnalyze@sig, Causal → TrueD; H* Not working right now *L
```





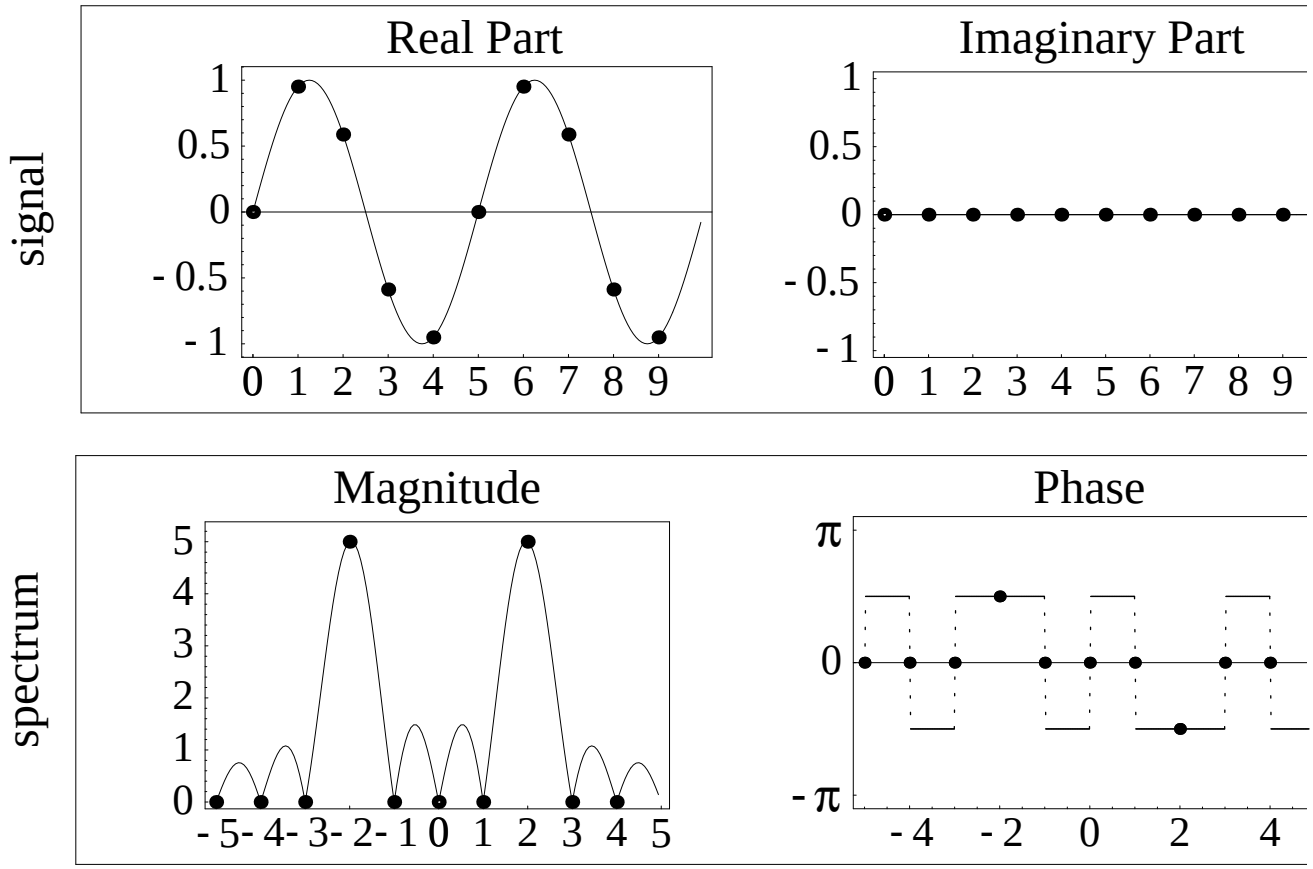
Changing the plotting region of a particular plot

In[13]:=
```
DFTAnalyze@sig, Causal → True,
 SpectrumOptions → 8LeftOptions → 8PlotRange → 880, 10<, 8-5, 15<<<<,
 FrameTicks → 8Automatic, 8-5, 0, 5, 10, 15<<D;
```



Making the spectrum magnitude display on a linear scale

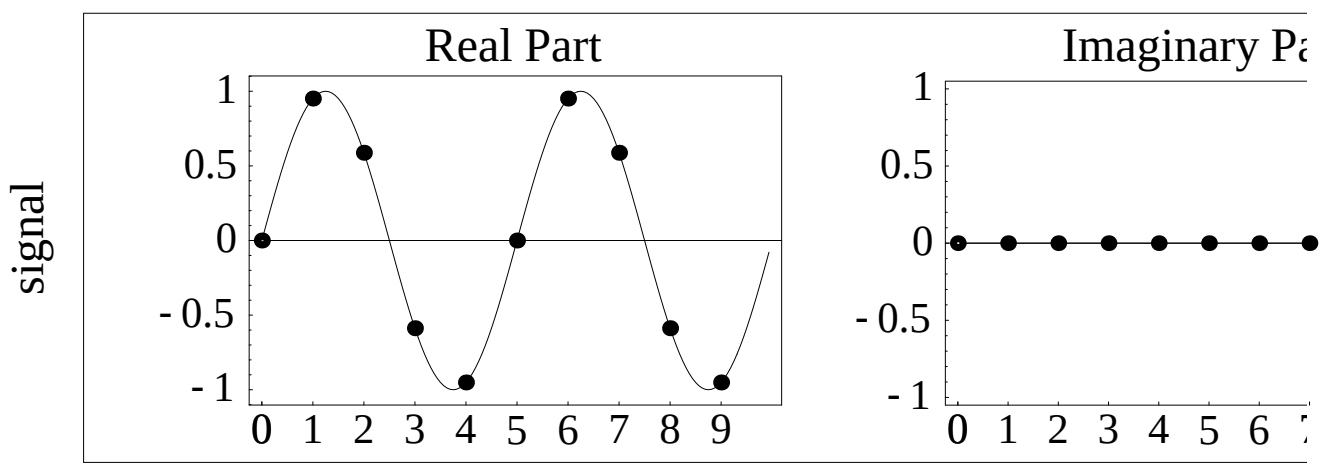
In[14]:=
```
DFTAnalyze@sig, LogScale → FalseD;
```

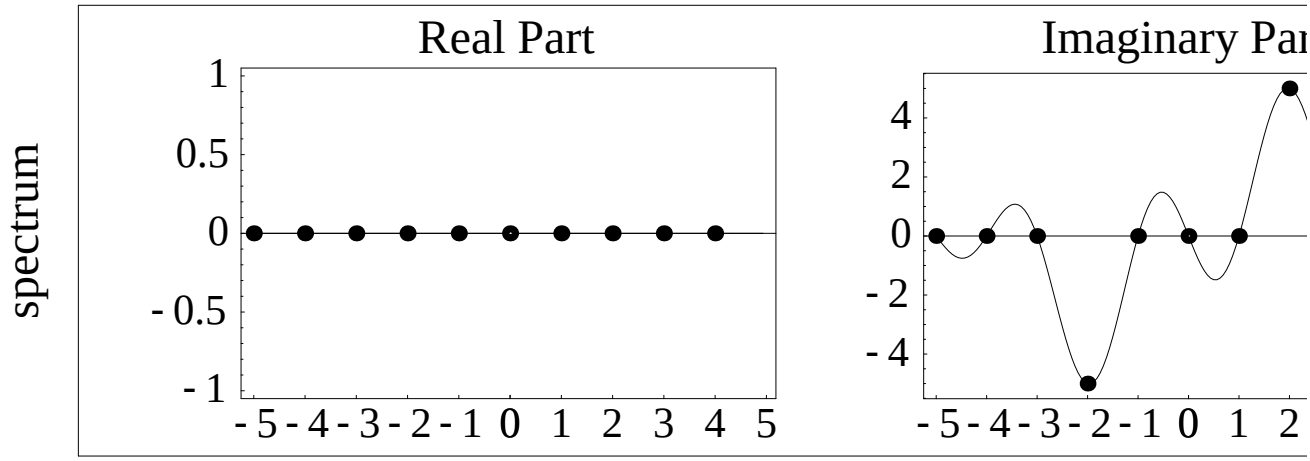


Making the spectrum plots be real/imaginary

Fourier Theorem: A Real/Odd signal gives an Imaginary/Odd spectrum.

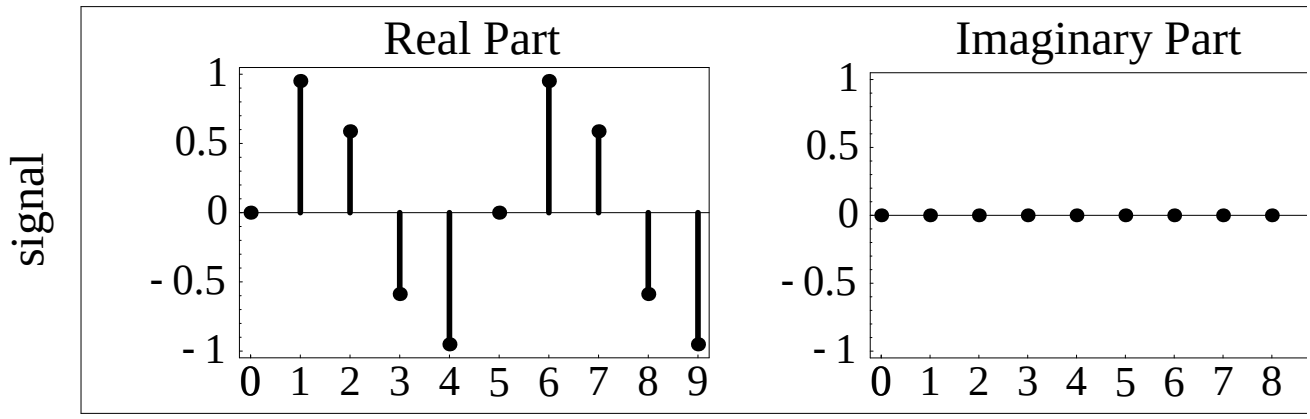
In[15]:=
```
DFTAnalyze@sig, Causal → False, SpectrumPlot → RealImaginaryPlotD;
```

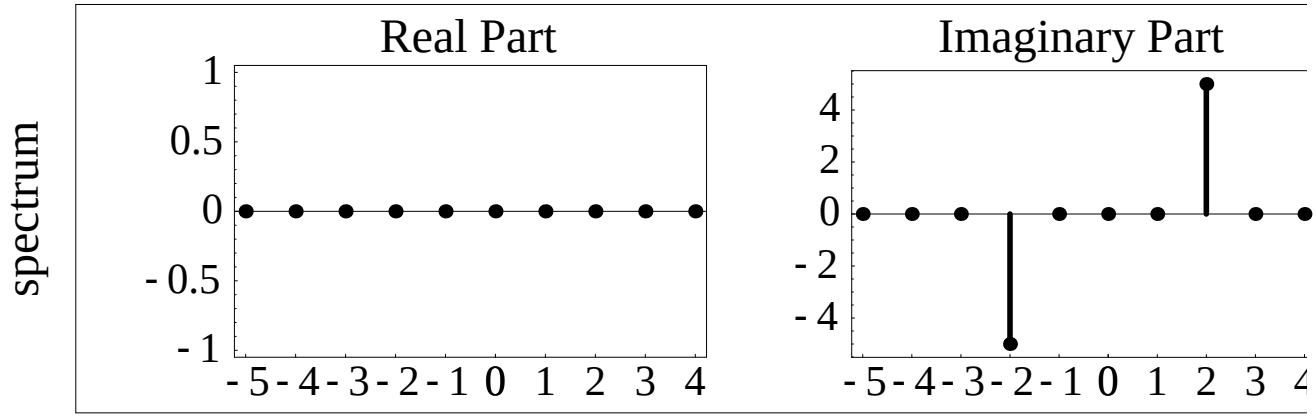




Removing interpolation, adding lines

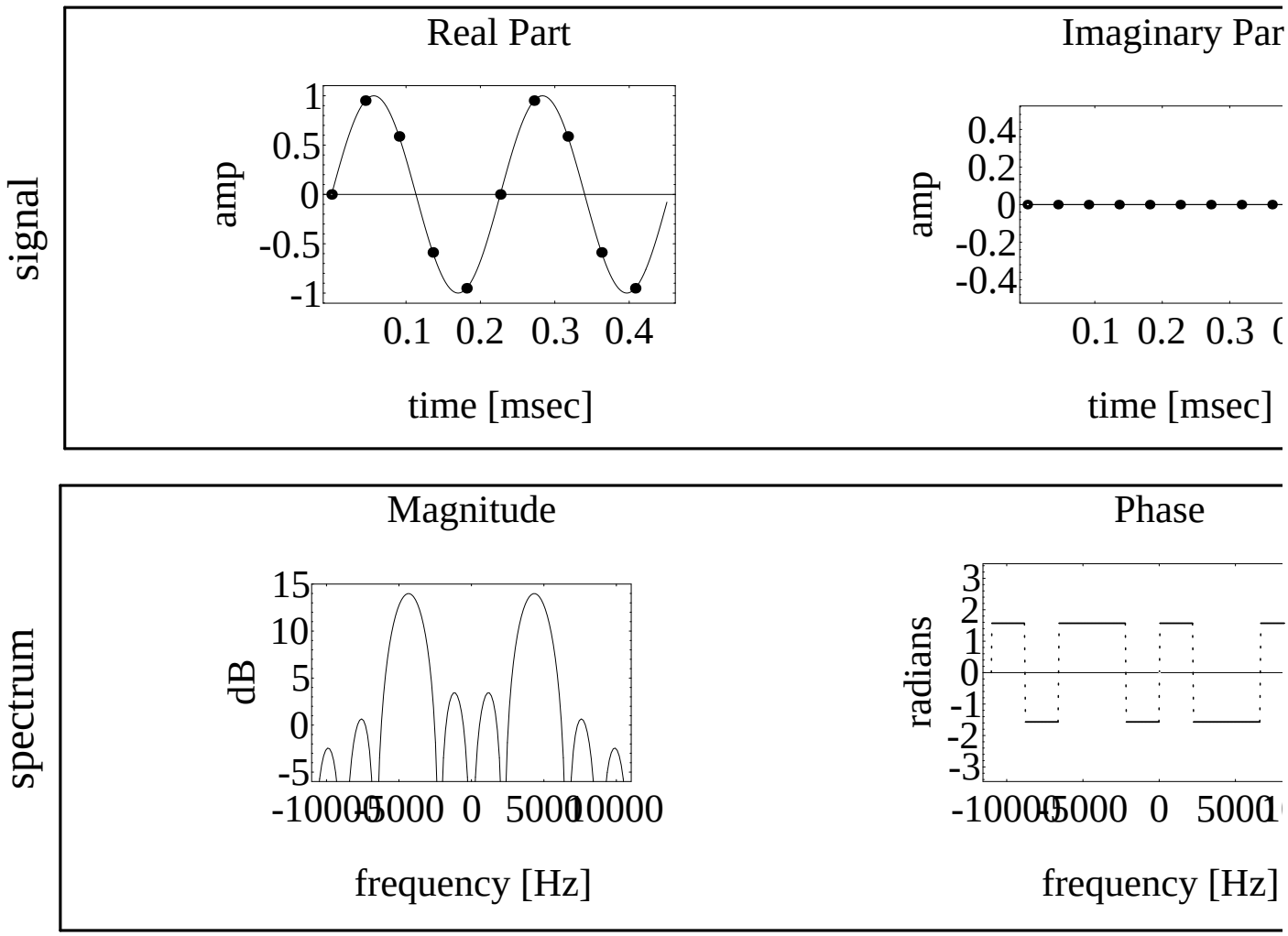
In[16]:=
```
DFTAnalyze@sig, SpectrumPlot → RealImaginaryPlot, Lines → True, Interpolate → FalseD;
```





Removing points, scaling to 22050 Hz sampling rate, adding labels

```
DFTAnalyzeAsig, Lines → False, SignalOptions → 9Stretch → 1.0/22050, FrameTicks →
    8880.0001, "0.1"<, 80.0002, "0.2"<, 80.0003, "0.3"<, 80.0004, "0.4"<<, Automatic<,
   LeftOptions → 8FrameLabel → 8"time @msecD", "amp", "Real Part", ""<<,
   RightOptions → 8FrameLabel → 8"time @msecD", "amp", "Imaginary Part", ""<<=,
 SpectrumOptions → 9Stretch → 22050/Length@sigD, Causal → False,
   Points → False, FrameTicks → 88-10000, -5000, 0, 5000, 10000<, Automatic<,
   LeftOptions → 8FrameLabel → 8"frequency @HzD", "dB", "Magnitude", ""<,
     PlotRange → 88-11025, 11025<, 8-6, 15<<<,
   RightOptions → 8FrameLabel → 8"frequency @HzD", "radians", "Phase", ""<<=E;
```



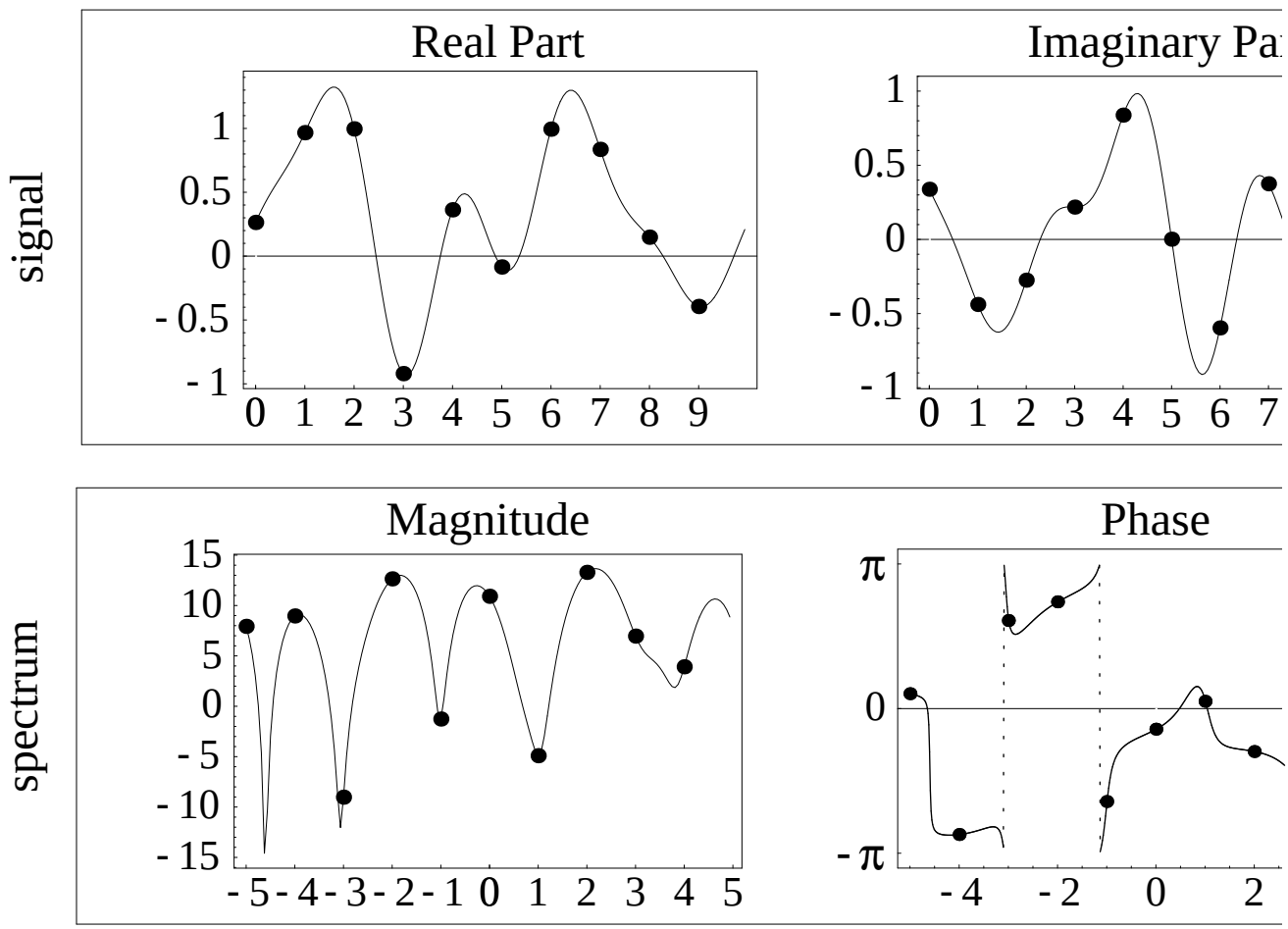
Unwrapping Phase

In[17]:=
```
sig = Table@Random@Complex, 8-1 - I, 1 + I<D, 8x, 0, 9<D
```

Out[17]=
```
80.263887 - 0.338227 I, 0.966811 + 0.438621 I, 0.99635 + 0.274774 I,
 -0.920187 - 0.217933 I, 0.36278 - 0.838223 I, -0.085088 - 0.00140039 I,
 0.994521 + 0.59654 I, 0.83579 - 0.375222 I, 0.148216 - 0.0572662 I, -0.394292 - 0.99999 I<
```

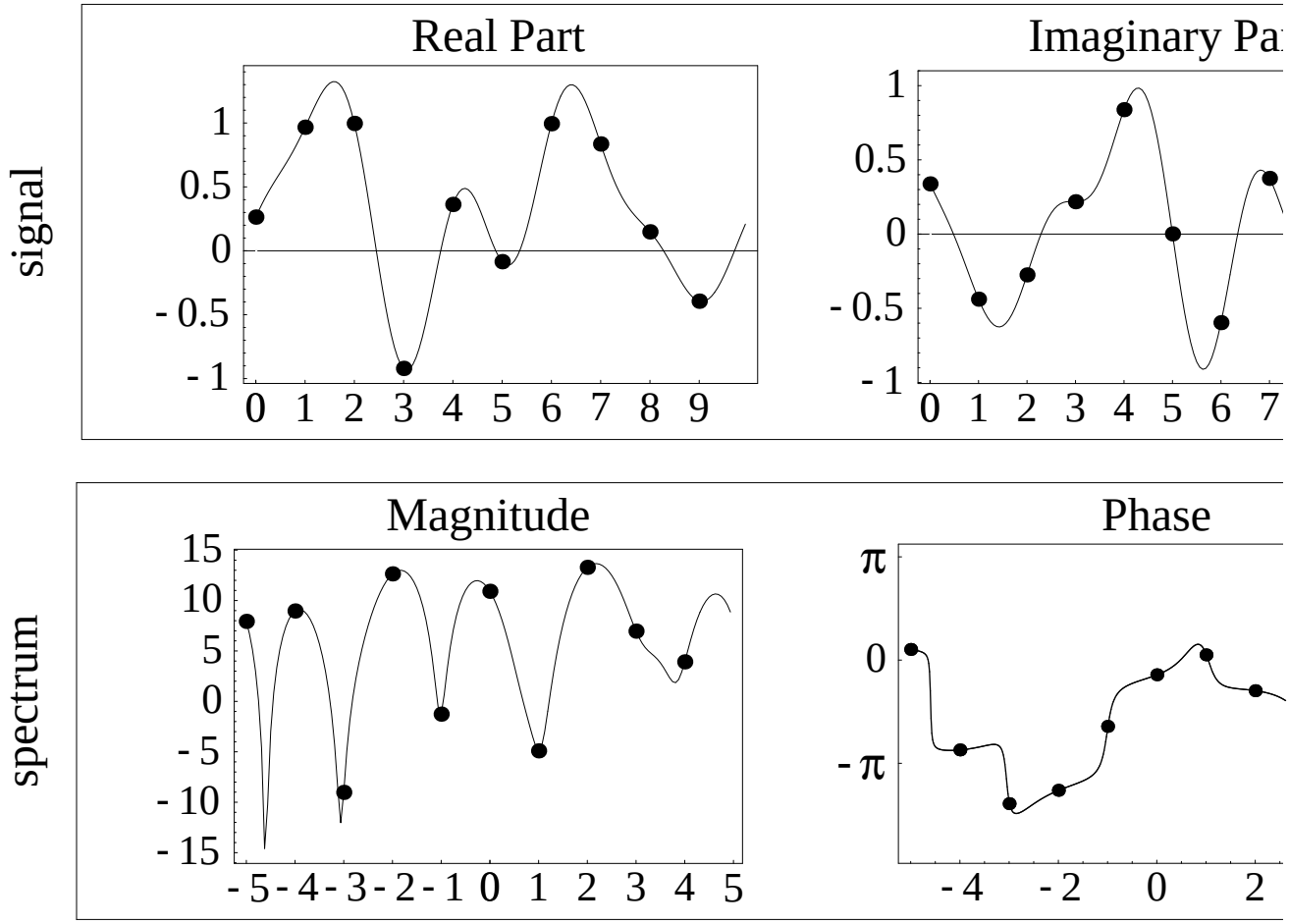
‡ Wrapped phase

In[18]:=
```
DFTAnalyze@sigD;
```



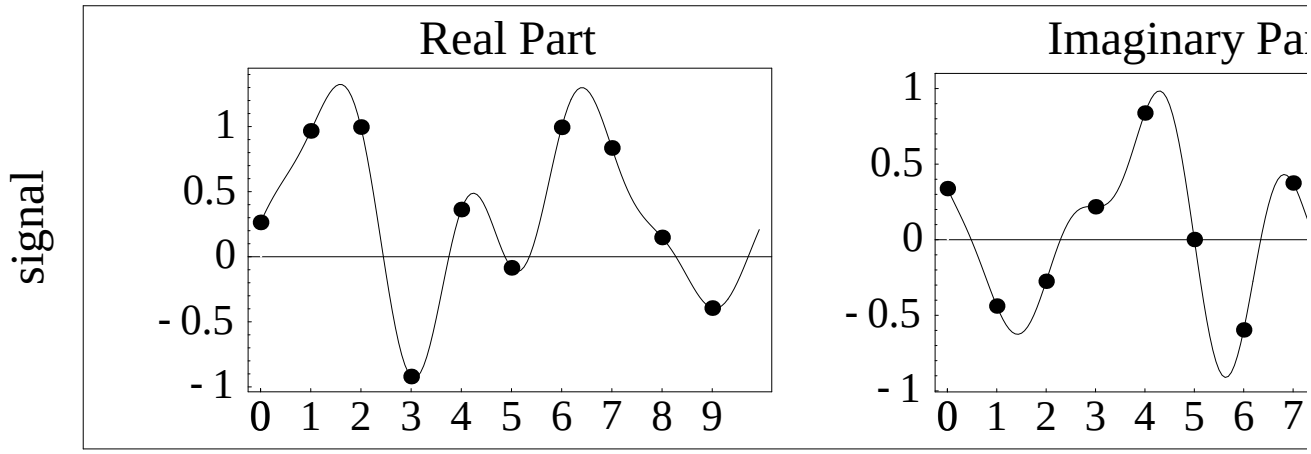
‡ Unwrapped phase

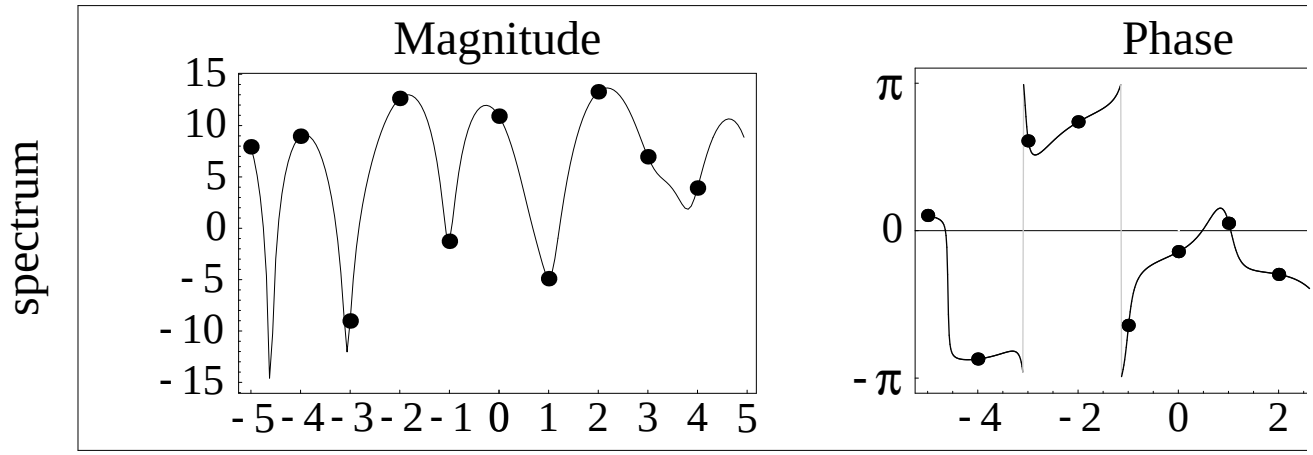
In[19]:=
```
DFTAnalyze@sig, Wrap → FalseD;
```



Changing Phase discontinuity style

In[20]:=
```
DFTAnalyze@sig, BreakStyle → GrayLevel@.8DD;
```





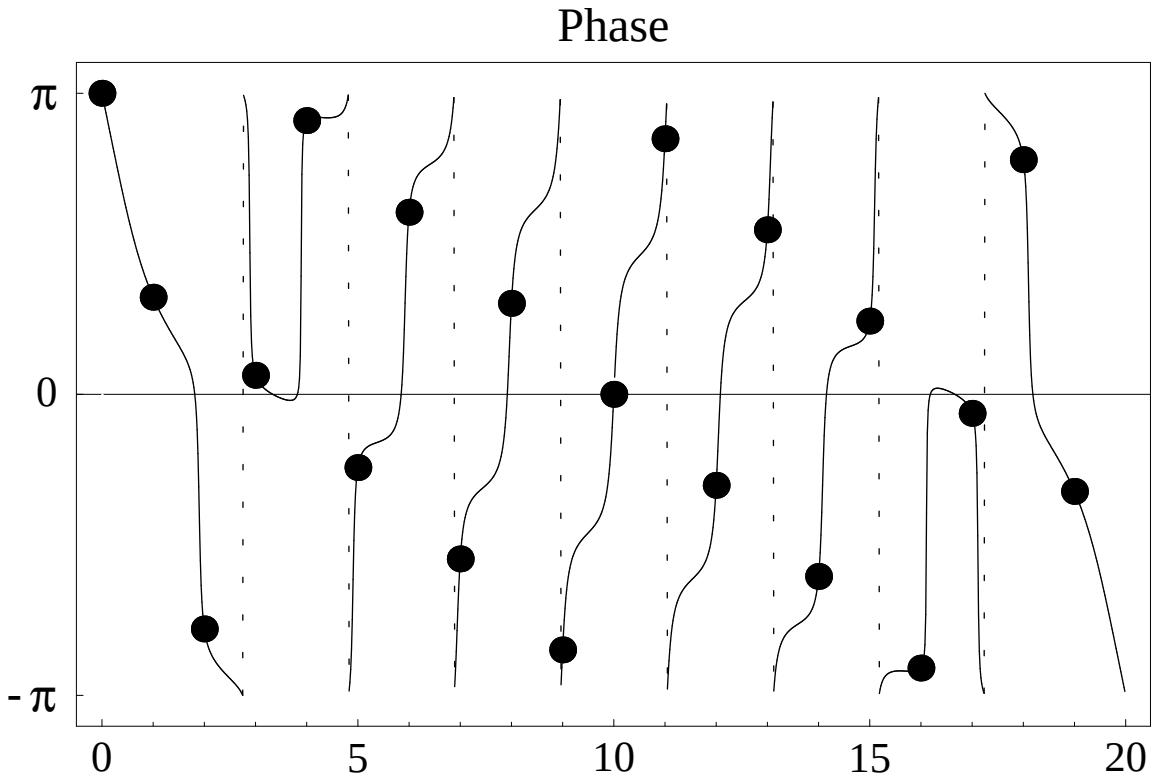
Component Plotting functions

PhasePlot[]

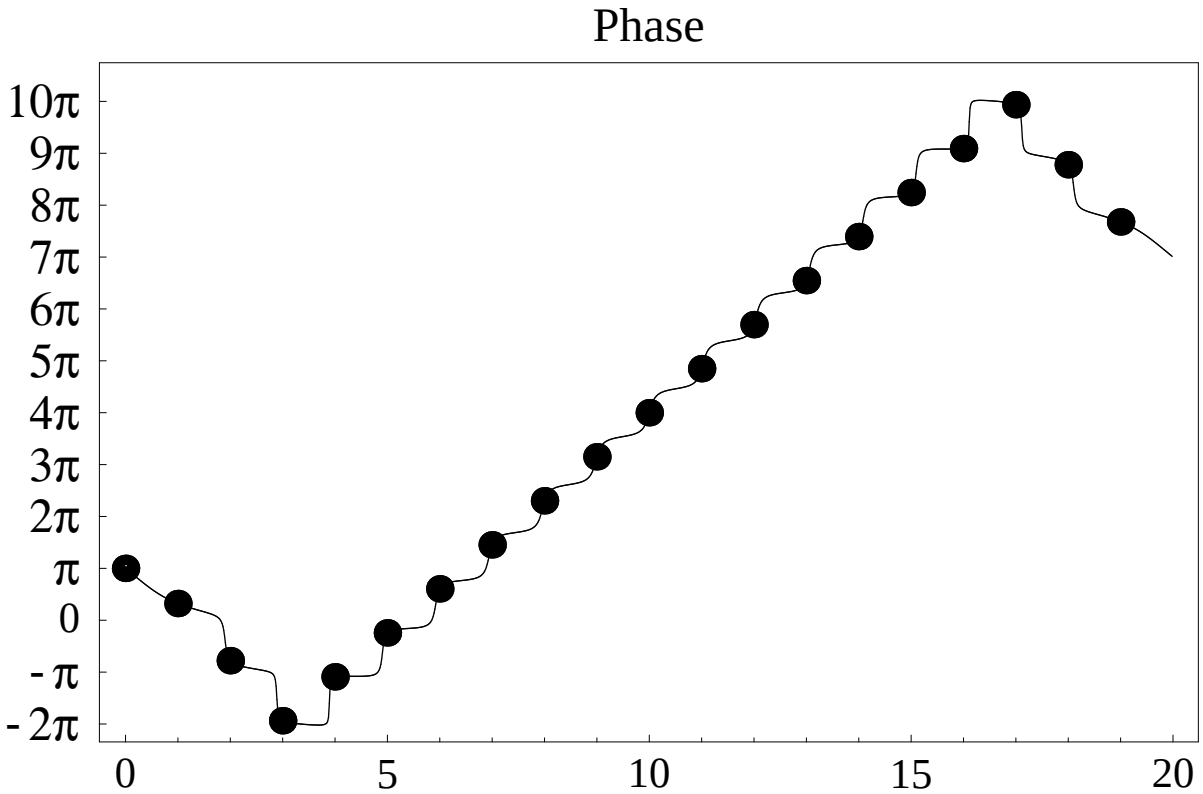
In[21]:=
```
Options@PhasePlotD
```

Out[21]=
```
8Frame Æ True, Axes Æ 8True, None<, PlotRange Æ All, Interpolate Æ True,
 InterpStyle Æ Thickness@0.D, Stretch Æ 1, Wrap Æ True, Frame Æ True,
 Lines Æ False, PointStyle Æ PointSize@0.025D, BreakStyle Æ Dashing@80.005, 0.03<D,
 FrameLabel Æ 8, , FontForm@Phase, 8Times-Roman, 18<D, <<
```

In[22]:=
```
PhasePlot[
 NumericDFT[ShiftA[SampleFunction[Sin[-2/201 π 2 x] + Sin[2 π x/20], {x, 0, 19}], 12]],
 Wrap → True, Points → True];
```



In[23]:=
```
PhasePlotA
 NumericDFTAShiftASampleFunctionASinA- 2/201 π 2 xE + SinA 2 π x/20 E, 8x, 0, 19<E, 12EE,
 Wrap → False, Points → TrueE;
```



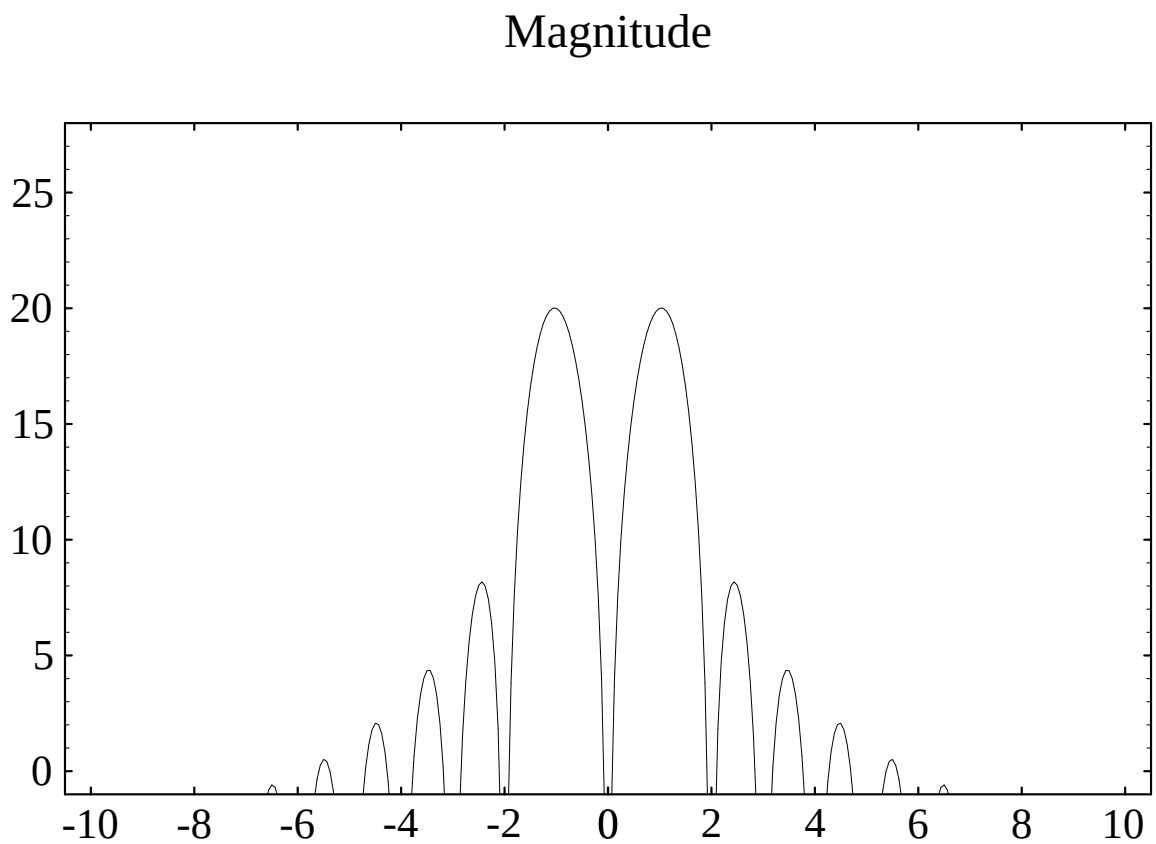
MagnitudePlot[]

```
Options@MagnitudePlotD
```

```
8LogScale Æ True, Frame Æ True, Axes Æ False, Interpolate Æ True,
 InterpStyle Æ Thickness@0.D, FrameLabel Æ 8, , FontForm@Magnitude, 8Times-Roman, 18<D, <,
 Stretch Æ 1, Lines Æ False<
```

Here is an animation of frequency sweeping from bin +/-1 to bin +/- 2 under a rectantular window, double-click on the picture below to start the animation.

```
Table@MagnitudePlot@NumericDFT@SampleFunction@Sin@-2/20 π H0 + ffL x + π/3 E, 8x, 0, 19<EE,
  Points → False, Causal → False, LogScale → True,
  PlotRange → 88-10.5, 10.5<, 8-1, 28<<, Repeat → 1E, 8ff, 1, 2, 0.1<E;
H* Currently a bug here *L
```

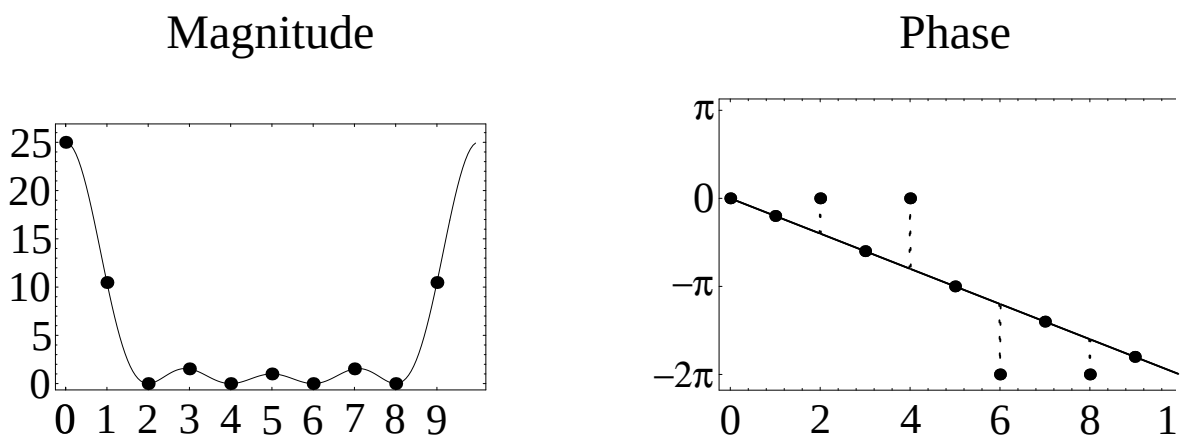


MagnitudePhasePlot[]

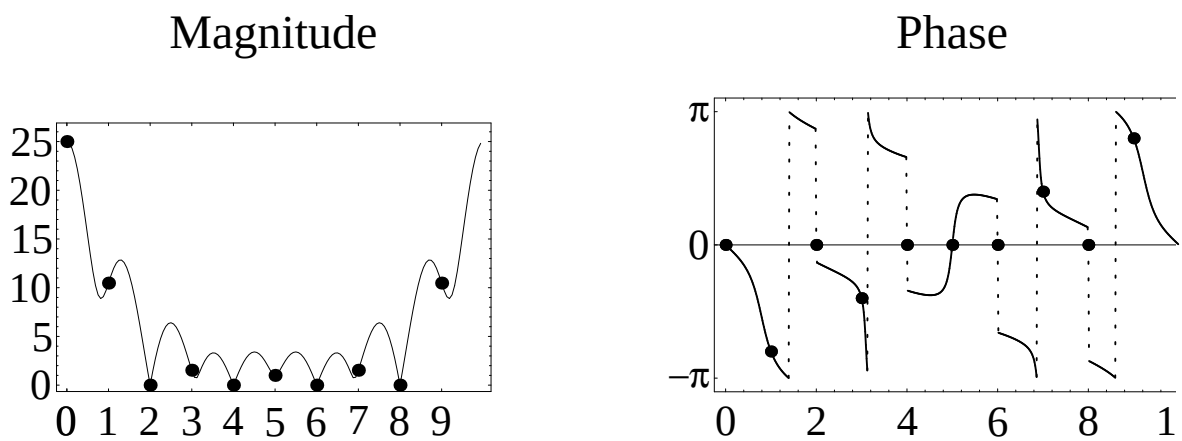
```
Options@MagnitudePhasePlotD
```

```
8LeftOptions Æ 8<, RightOptions Æ 8<<
```

```
MagnitudePhasePlot@NumericDFT@84, 5, 4, 3, 2, 1, 0, 1, 2, 3<D, Wrap → False,
 LogScale → FalseD;
```




```
MagnitudePhasePlot@NumericDFT@81, 2, 3, 4, 5, 4, 3, 2, 1, 0<D, Wrap → True,
 LogScale → FalseD;
```



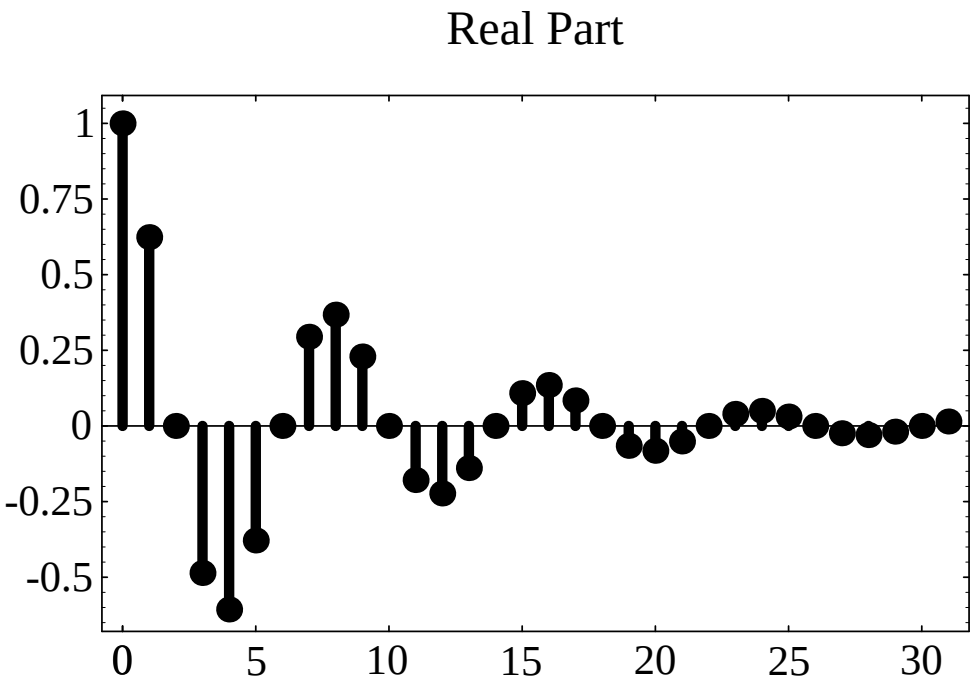
RealPlot[]

```
Options@RealPlotD
```

```
8Frame Æ True, Axes Æ 8Automatic, None<, Interpolate Æ True,
 Stretch Æ 1, Lines Æ False, InterpStyle Æ Thickness@0.D, PlotRange Æ All,
 FrameLabel Æ 8, , FontForm@Real Part, 8Times-Roman, 18<D, <<
```

```
seq = SampleFunctionAE^(-x/8) E^(1/8 I 2 π x), 8x, 0, 31<E;
```

```
RealPlot@seq, Lines → True, Interpolate → False D;
```

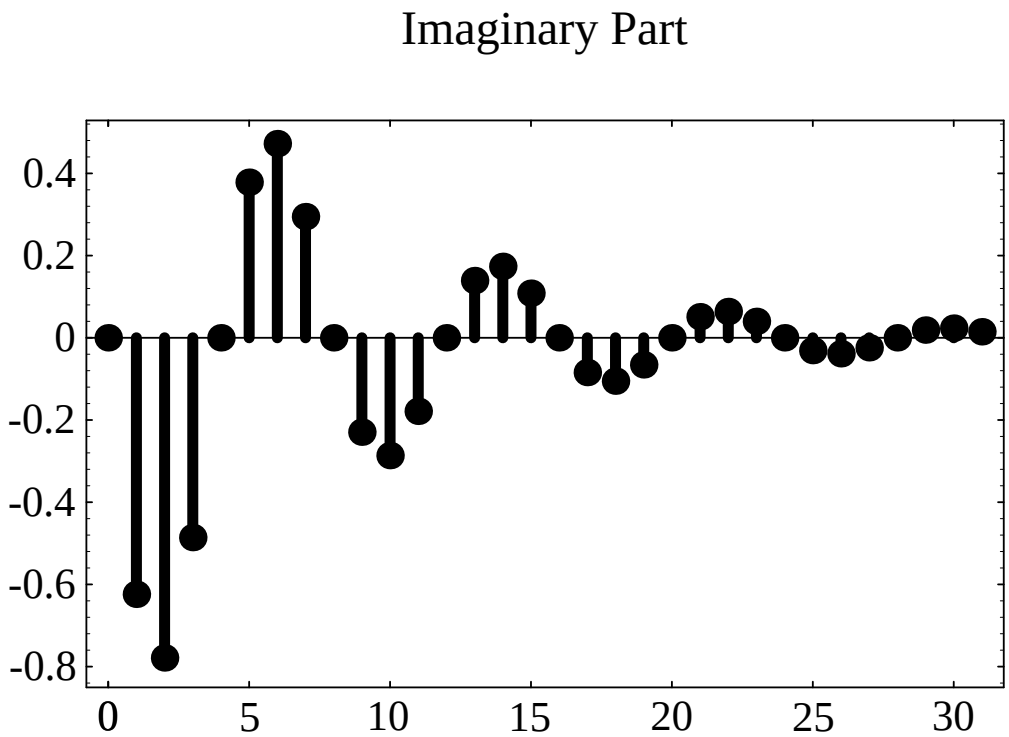


ImaginaryPlot[]

```
Options@ImaginaryPlot D
```

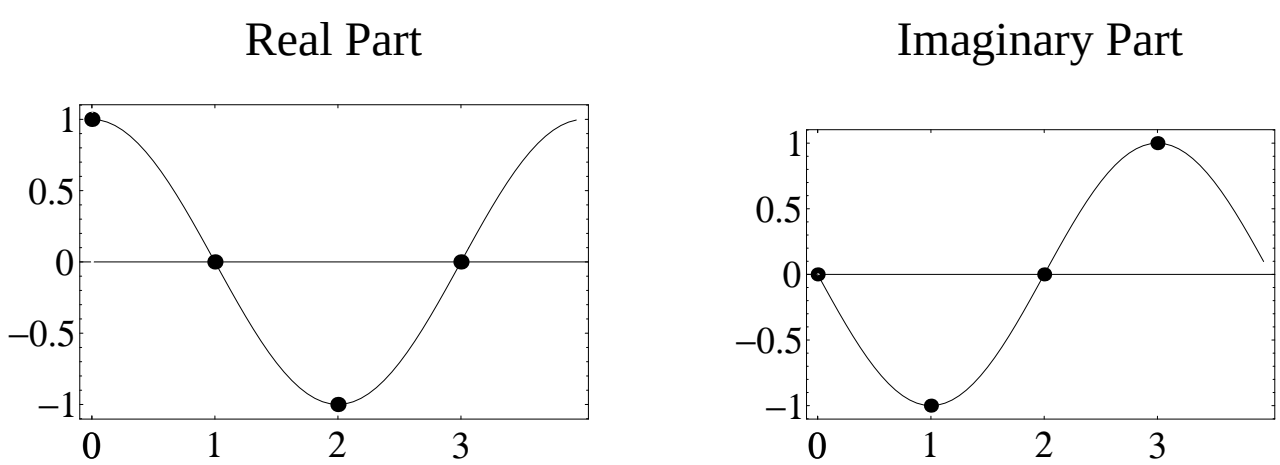
```
8Frame Æ True, Axes Æ 8Automatic, None<, Interpolate Æ True, InterpStyle Æ Thickness@0.D,
 FrameLabel Æ 8, , FontForm@Imaginary Part, 8Times-Roman, 18<D, <, Stretch Æ 1,
 Lines Æ False<
```

```
ImaginaryPlot@seq, Lines → True, Interpolate → FalseD;
```



RealImaginaryPlot[]

```
Options@RealImaginaryPlotD
```

```
8Axes Æ 8True, None<, LeftOptions Æ 8<, RightOptions Æ 8<<
```

```
RealImaginaryPlot@81, I, -1, -I<D;
```