Projections

```
With[{context = "projections`"}, If[Context[] ≠ context, Begin[context]]];
Dynamic[Refresh[Context[], UpdateInterval → 1]]
```

```
Global`
```

```
c1 = {1, 1} / Sqrt[2]
c2 = c1 * {1, -1}
```

$\left\{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right\}$

$\left\{\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right\}$

```
v = {2, 3}
```

```
{2, 3}
```

```
{v.c1, v.c2}
```

$\left\{\frac{3}{\sqrt{2}} + \sqrt{2}, -\frac{3}{\sqrt{2}} + \sqrt{2}\right\}$

```
Simplify[(v.c1) c1 + (v.c2) c2]
```

```
{2, 3}
```

```
With[{context = "projections`"}, If[Context[] == context, End[], "Not in context"]]
```

```
projections`
```

DFT

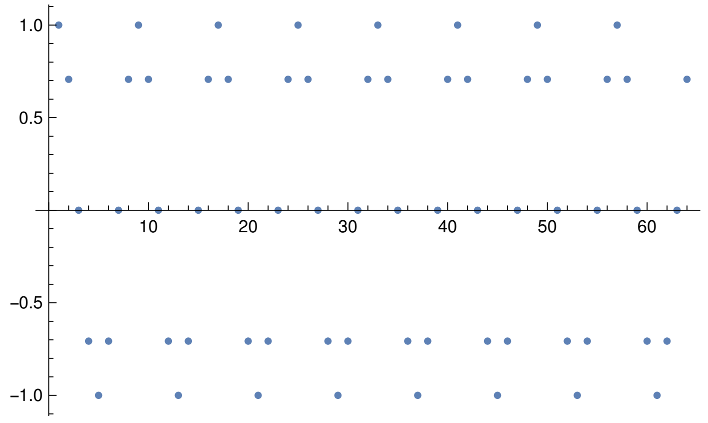
```
In[15]:= With[{context = "dft`"}, If[Context[] ≠ context, Begin[context]]];
         Dynamic[Refresh[Context[], UpdateInterval → 1]]
Out[15]= Global`
```

```
In[16]:= data = Cos[Pi / 4 * Range[0, 63]]
```

$$\text{Out[16]=}\ \{1, \tfrac{1}{\sqrt{2}}, 0, -\tfrac{1}{\sqrt{2}}, -1, -\tfrac{1}{\sqrt{2}}, 0, \tfrac{1}{\sqrt{2}}, 1, \tfrac{1}{\sqrt{2}}, 0, -\tfrac{1}{\sqrt{2}}, -1, -\tfrac{1}{\sqrt{2}}, 0, \tfrac{1}{\sqrt{2}}, 1, \tfrac{1}{\sqrt{2}}, 0, -\tfrac{1}{\sqrt{2}}, -1, -\tfrac{1}{\sqrt{2}}, 0, \tfrac{1}{\sqrt{2}}, 1, \tfrac{1}{\sqrt{2}}, 0, -\tfrac{1}{\sqrt{2}}, -1, -\tfrac{1}{\sqrt{2}}, 0, \tfrac{1}{\sqrt{2}}, 1, \tfrac{1}{\sqrt{2}}, 0, -\tfrac{1}{\sqrt{2}}, -1, -\tfrac{1}{\sqrt{2}}, 0, \tfrac{1}{\sqrt{2}}, 1, \tfrac{1}{\sqrt{2}}, 0, -\tfrac{1}{\sqrt{2}}, -1, -\tfrac{1}{\sqrt{2}}, 0, \tfrac{1}{\sqrt{2}}, 1, \tfrac{1}{\sqrt{2}}, 0, -\tfrac{1}{\sqrt{2}}, -1, -\tfrac{1}{\sqrt{2}}, 0, \tfrac{1}{\sqrt{2}}, 1, \tfrac{1}{\sqrt{2}}, 0, -\tfrac{1}{\sqrt{2}}, -1, -\tfrac{1}{\sqrt{2}}, 0, \tfrac{1}{\sqrt{2}}\}$$

```
In[17]:= ListPlot[data]
```

Out[17]=



```
In[18]:= MatrixForm@{Fourier[data]}
Out[18]//MatrixForm=
  ( 0.  0.  0.  0.  0.  0.  0.  0.  4.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  (

In[19]:= Simplify[Cos[θ] - 1 / 2 (Exp[I θ] + Exp[- I θ])]
Out[19]= 0

In[20]:=

In[21]:= With[{context = "dft`"}, If[Context[] == context, End[], "Not in context"]]
Out[21]= dft`
```

Sound Alteration

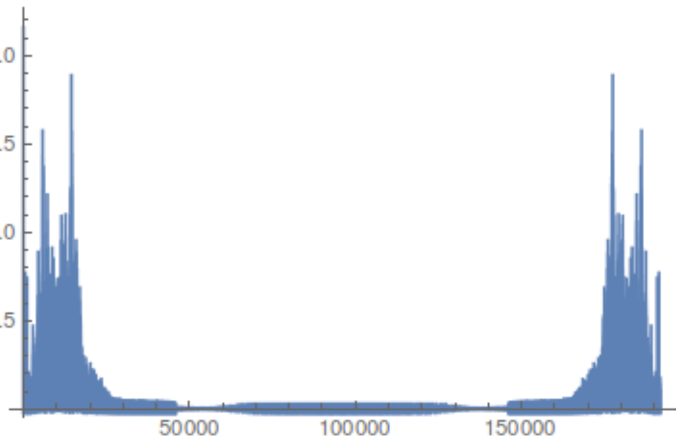
```
In[22]:= With[{context = "alter`"}, If[Context[] ≠ context, Begin[context]]];
         Dynamic[Refresh[Context[], UpdateInterval → 1]]
Out[22]= Global`

In[23]:= SetDirectory@NotebookDirectory[]
Out[23]= /home/eric/Documents/School/QEA2/Acoustic Modem/Bset 1
```

```
In[51]:= playSound[data_] := Module[{path = "/tmp/sound.wav"}, Export[path, data];
           SystemOpen[path]]
         playSound[data_, fs_] := playSound[Sound@SampledSoundList[data, fs]]

In[31]:= ExampleData["Audio"]

Out[31]= {{Audio, Apollo11ReturnSafely}, {Audio, Apollo11SmallStep}, {Audio, BalloonPop},
          {Audio, Bee}, {Audio, Bird}, {Audio, BlackcapWarbler}, {Audio, Cat}, {Audio, Cello},
          {Audio, CelloScale}, {Audio, ChurchBell}, {Audio, Clapping}, {Audio, CreakyDoor},
          {Audio, Crowd}, {Audio, DogBark}, {Audio, Drums}, {Audio, FemaleVoice},
          {Audio, Flute}, {Audio, FluteScale}, {Audio, FogHorn}, {Audio, IRMaesHowe},
          {Audio, IRRailwayTunnel}, {Audio, IRSportsCenter}, {Audio, IRStairway},
          {Audio, IRStAndrewsChurch}, {Audio, IRStMarysChurch}, {Audio, IRYorkMinsterChurch},
          {Audio, Laughing}, {Audio, MaleVoice}, {Audio, NoisyTalk}, {Audio, Piano},
          {Audio, PianoScale}, {Audio, PowerSupply}, {Audio, Scream}, {Audio, SubwayTrain},
          {Audio, Sword}, {Audio, ThaiBells}, {Audio, Water}, {Audio, Wind}}
```

One Small Step

```
In[84]:= fs = ExampleData[{"Audio", "Apollo11SmallStep"}, "SampleRate"]

Out[84]= 22 050

In[85]:= data = ExampleData[{"Audio", "Apollo11SmallStep"}, "Data"];

In[212]:= playSound[data, fs]

In[146]:= fft = Fourier[data];

In[213]:= Rasterize@ListLinePlot[Abs@fft, PlotRange → Full]
```

Out[213]=

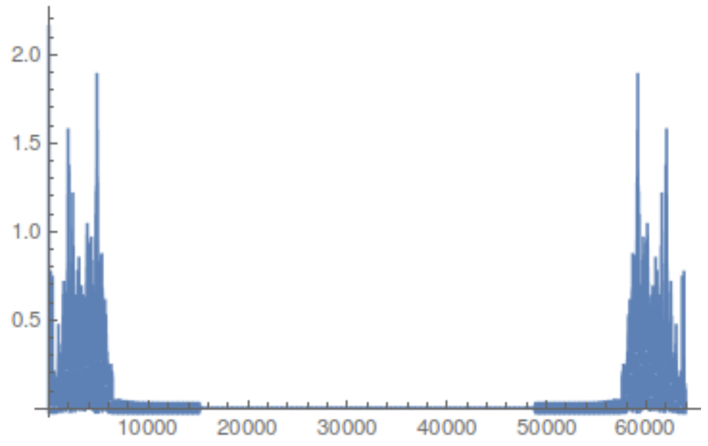


Perform filtering: Note that I am cutting out a higher frequency than the Bset recommends because my audio sample is different.

```
In[204]:= filteredFourier = fft;
          cutoff = Round[Length@fft / 10];
          filteredFourier[[cutoff ;; -cutoff]] = fft[[cutoff ;; -cutoff]] / 10;
```

```
In[215]:= Rasterize@ListLinePlot[Abs[filteredFourier[[ ;; ;; 3]]], PlotRange → Full]
```

Out[215]=



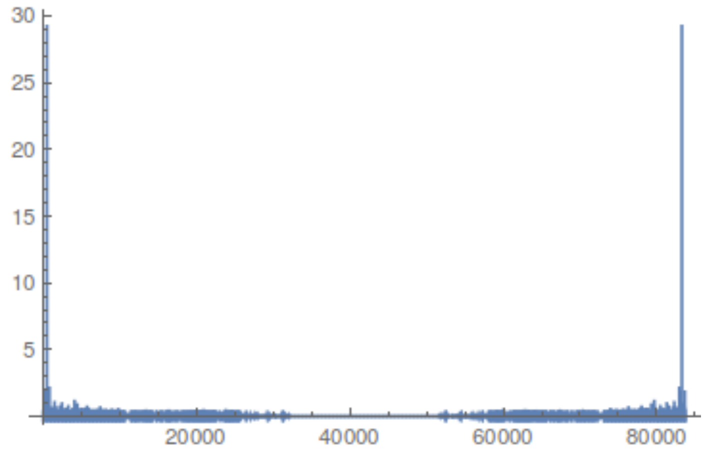
Convert back to sound

```
In[235]:= data2 = Re@InverseFourier[filteredFourier];
          playSound[data2, fs]
```

Guitar

```
In[220]:= fs = Import["provided files/guitar_riff_hum.wav", "SampleRate"]
          data = Import["provided files/guitar_riff_hum.wav", "Data"];

Out[220]= 44 100

In[237]:= playSound[data, fs]

In[223]:= fft = Fourier[data];

In[224]:= Rasterize@ListLinePlot[Abs@fft, PlotRange → Full]
```

Out[224]=

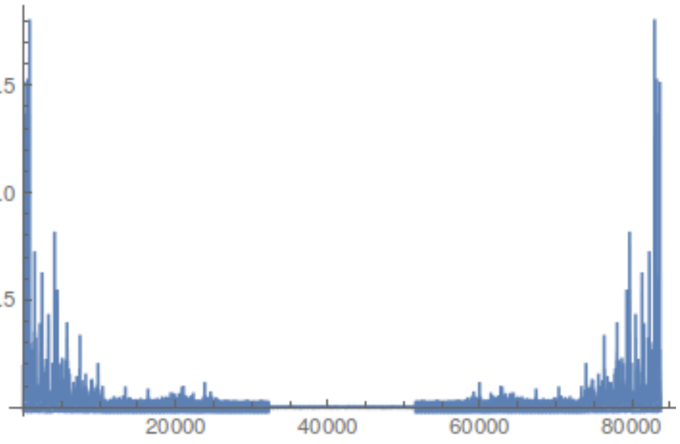


Perform filtering: Looking at the graph above, I decided to neuter all overpowered freqs.

```
badFreq = Max
```

```
In[246]:= cutoff = 3;
          filteredFourier = Map[Function[val, If[Abs@val > cutoff, 0, val]], fft];

In[250]:= Rasterize@ListLinePlot[Abs[filteredFourier], PlotRange → Full]
```

Out[250]=



Convert back to sound

```
In[248]:= data2 = Re@InverseFourier[filteredFourier];
          playSound[data2, fs]
```

Wrap up

```
In[251]:= With[{context = "alter`"}, If[Context[] == context, End[], "Not in context"]]

Out[251]= alter`
```

Template

Scratch work

```
In[252]:= Export["Mathematica Scratch.pdf", EvaluationNotebook[]]
```