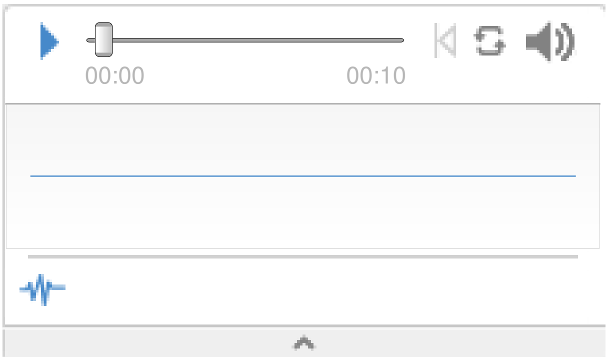
MATH528 Lab09: Sturm-Liouville guitar string example

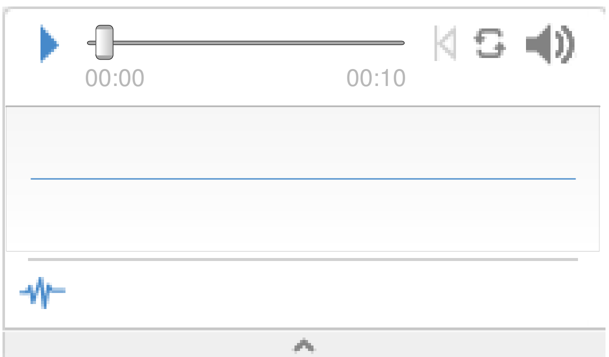
Input sound

This is the recorded sound from vibration of the E_2 guitar string

In[6]:= `m =`



Out[6]=



These are the properties of the sound sample

In[8]:=
```
props = {AudioLength, Duration, AudioChannels, AudioSampleRate, AudioType};
```

In[9]:=
```
Dataset@AssociationThread[props, Through[props[m]]]
```

Out[9]=

AudioLength	457 392 samples
Duration	9.529 s
AudioChannels	2
AudioSampleRate	48 000 Hz
AudioType	SignedInteger16

Find the sampling frquency

In[32]:=
```
fs = AudioSampleRate[m]
```

Out[32]= 48 000 Hz

Transform the sound into data

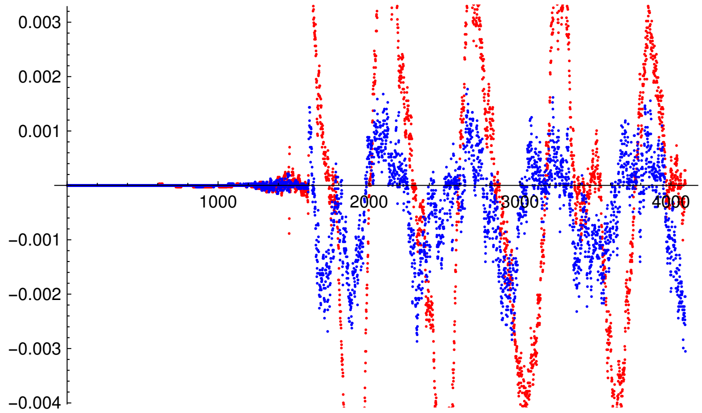
In[11]:=
```
d = AudioData[m];
```

In[∘]:=
```
Dimensions[d]
```

Out[∘]= {2, 457 392}

In[12]:=
```
ListPlot[{d[[1, 1 ;; 4096]], d[[2, 1 ;; 4096]]}, PlotStyle → {Red, Blue} ]
```

Out[12]=



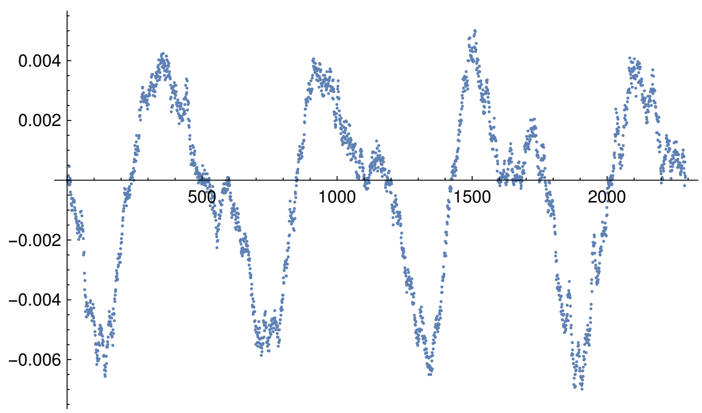
Frequency representation of sound (Fourier transform)

Conformant window

Choose one channel and a window that corresponds to a repeating signal

In[13]:=
```
data = d[[1, 5000 + 270 ;; 5000 + 2559]];
ListPlot[data]
```

Out[14]=



In[15]:=
```
nd = Length[data]
```

Out[15]= 2290

Compute the Fourier coefficients

In[19]:=
```
nMax = 100; dj = 2. π / nd;
a = Table[ 1 / 2. / nd Sum[data[[j]] Cos[j n dj], {j, 1, nd}], {n, 1, nMax}];
b = Table[ 1 / 2. / nd Sum[data[[j]] Sin[j n dj], {j, 1, nd}], {n, 1, nMax}];
c2c = Log10[a^2 + b^2];
```

Find index of largest coefficient

In[26]:=
```
kmx = Position[c2c, Max[c2c]][[1, 1]]
```

Out[26]= 4

Compute the signal period

In[30]:=
```
T = nd / fs
```

Out[30]= $\frac{229}{4800}$/Hz

The dominant frequency is

In[35]:=
```
fmx = N[(T / kmx)^-1]
```

Out[35]= 83.8428 Hz

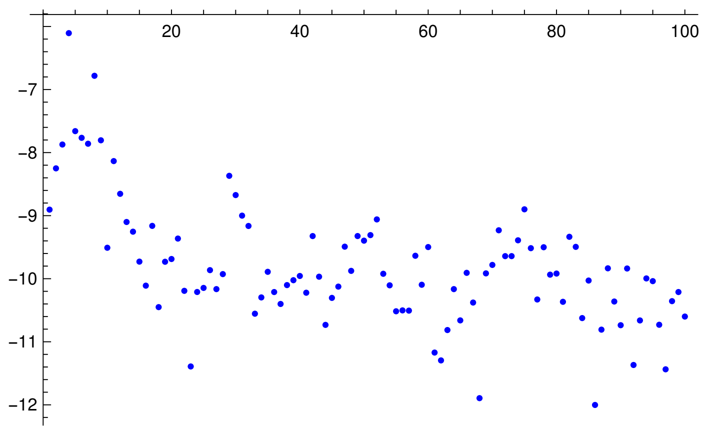
The E_2 note should have a frequency of 82.41 Hz, hence the recorded sound is mistuned by

In[42]:=
```
fE2 = 82.41; fE2G = 83.8428;
err = N[(fE2G - fE2) / fE2]
```

Out[43]= 0.0173862

In[29]:=
```
ConformantWindowSpectrum = ListPlot[c2c, PlotStyle → Blue]
```

Out[29]=

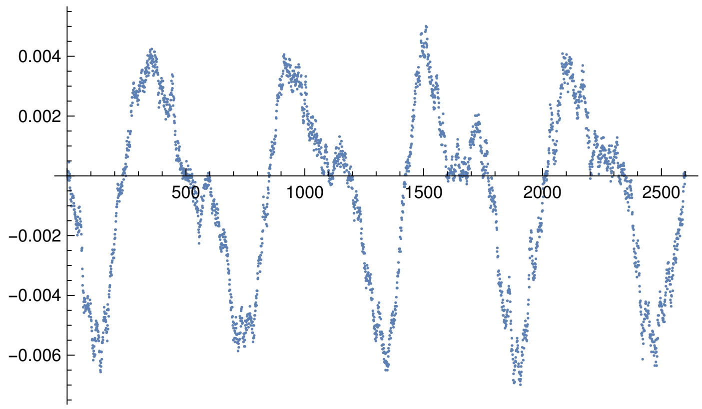


Non-conformant window

If the sound sampling window is chosen correctly, for example induces a discontinuity in the derivative as below, the resulting spectrum is affected by aliasing errors

In[44]:=
```
data = d[[1, 5000 + 270 ;; 5000 + 2870]];
ListPlot[data]
```

Out[45]=



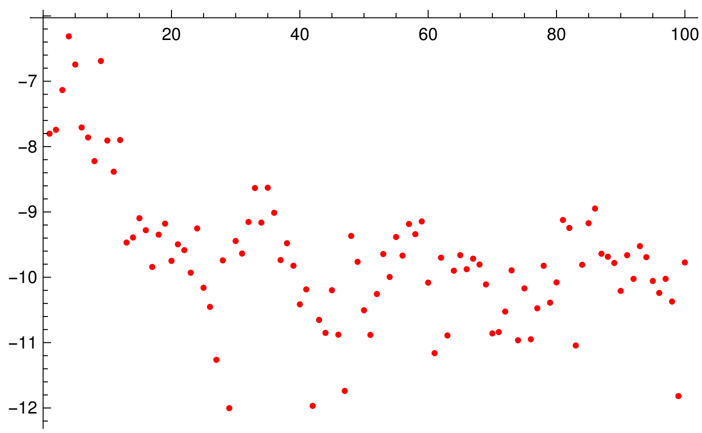
In[46]:=
```
nd = Length[data]
```

Out[46]= 2601

In[47]:=
```
nMax = 100; dj = 2. π / nd;
a = Table[ 1 / 2. / nd Sum[data[[j]] Cos[j n dj], {j, 1, nd}], {n, 1, nMax}];
b = Table[ 1 / 2. / nd Sum[data[[j]] Sin[j n dj], {j, 1, nd}], {n, 1, nMax}];
c2c = Log10[a^2 + b^2];
```

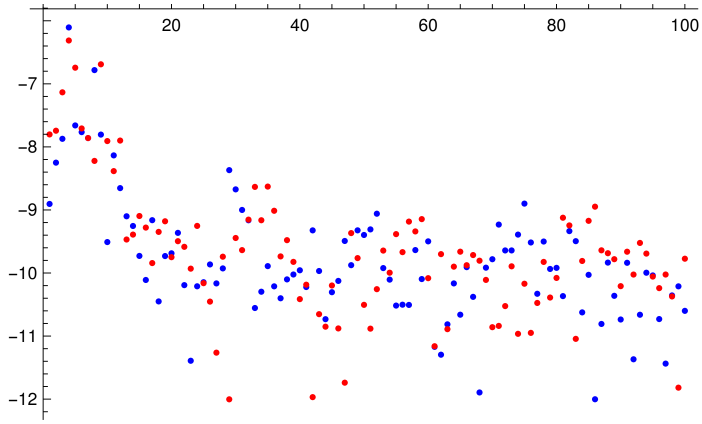
In[51]:=
```
NonConformantWindowSpectrum = ListPlot[c2c, PlotStyle → Red]
```

Out[51]=



In[52]:=
```
Show[ConformantWindowSpectrum, NonConformantWindowSpectrum]
```

Out[52]=



Notice that the dominant frequency is still at mode k=4, but the low-frequency amplitudes are higher than in the case of the conformant sample window. This is due to the discontinuity in the boundary condition inducing high frequency components that are not resolved by the sampling rate, but are captured as false low-frequency modes. We say that the unresolved high-frequency modes are “aliased” onto low-frequency modes, similar to the apparent slow motion of a helicopter rotor filmed at some fixed frame rate lower than the rotor angular velocity.