

Parabolic Peak Interpolation in *Mathematica*

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

15 April 1996

Getting Started

Load the SCMTTheory package:

```
<< SCMTTheory`
```

To see the functions defined in this package type:

```
?SCMTTheory`*
```

To get help on a certain function type:

```
?ParabolaPeak
```

```
ParabolaPeak[s1, s2, s3] returns the parabolic extremum
point of the parabola uniquely described by the three
input samples. Returns a delta sample value for the
location of the peak from the second sample
ParabolaPeak[{x1,y1},{x2,y2},{x3,y3}] will give the x-y
coordinates of the parabolic maximum (or minimum) of an
arbitrary set of three points.
```

Parabolic Fitting

Parabolas are often used to find the spectral peaks in DFTs of sound signals. A spectral peak of a single frequency is the convolution of the delta function (for the frequency) and the spectrum of the analysis window (e.g., the rectangle window has a digital sinc for a transform). The DFT has frequency bins much further apart than the frequency discrimination of the human ear, so just looking for the frequency bin with the largest magnitude is not sufficient for finding a frequency

peak. The typical frequency separation between bins in the DFT is 43 Hz when using a 1024-sample window at a 44.1 kHz sampling rate. At 440 Hz, this bin size is about a minor-second musical interval.

A Parabola of the form $a(x-p)^2+b$ can be described uniquely by three points. So, if you have three samples, you can find the maximum (or minimum) of the parabolic curve that passes through all three points. For the `ParabolaPeak` function, it is assumed that the three input numbers are the amplitudes (y-values) of a discretely sampled function, and the value returned is the deviation of the peak of the parabola from the second (center) point.

```
ParabolaPeak[1,5,3]
```

```
1  
6
```

The $\frac{1}{6}$ in the answer above means that the peak is $\frac{1}{6}$ of a sample after the second sample.

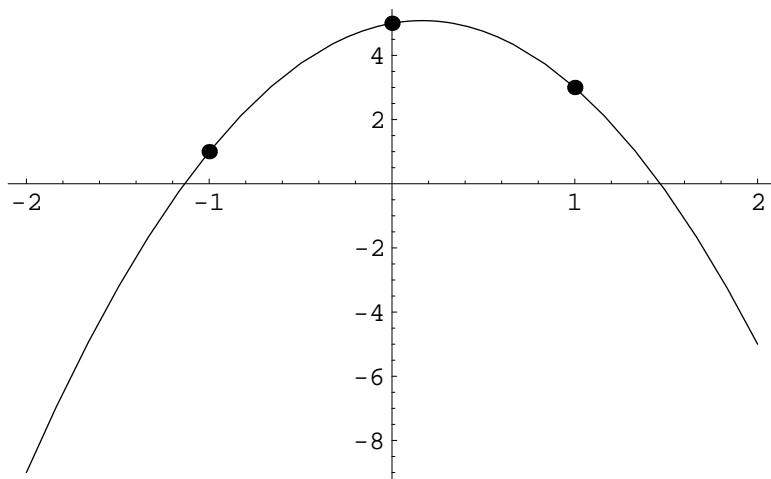
If you need the function defining the fitted parabola, use the `ParabolaFit` function.

```
ParabolaFit[1,5,3]
```

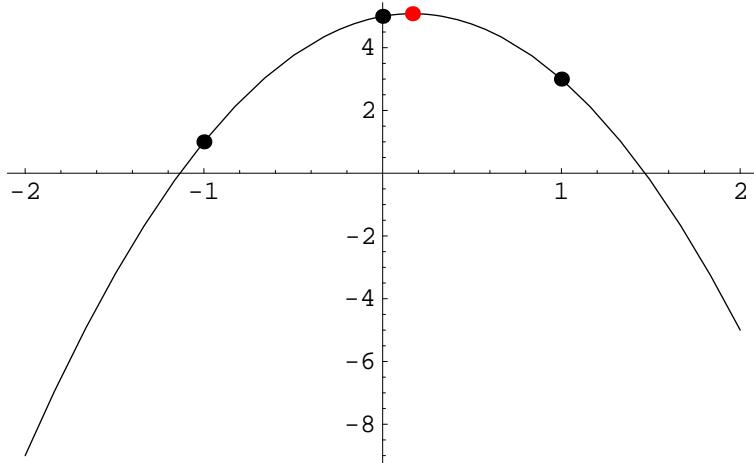
$$9\frac{61}{12} - 3J - \frac{1}{6} + xN^2 =$$

For visualization purposes, `PlotParabolaFit` will plot the points and the fitted parabola.

```
ParabolaFitPlot[1, 5, 3, Peak->False];
```



```
ParabolaFitPlot[1,5,3];
```



ParabolaFit, and ParabolaPeak can take any arbitrary three coordinate pairs, not just three consecutive samples.

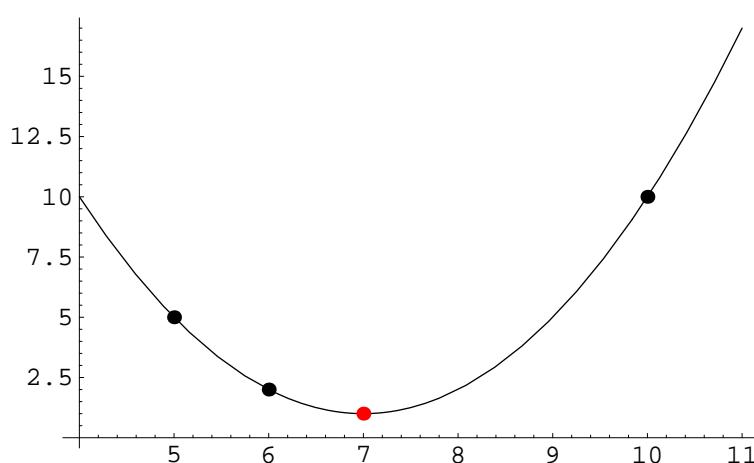
```
ParabolaPeak[{5,5},{6,2},{10,10}]
```

```
87, 1<
```

```
ParabolaFit[{5,5},{6,2},{10,10}]
```

```
81 + H - 7 + xL2<
```

```
ParabolaFitPlot[{5,5},{6,2},{10,10}];
```



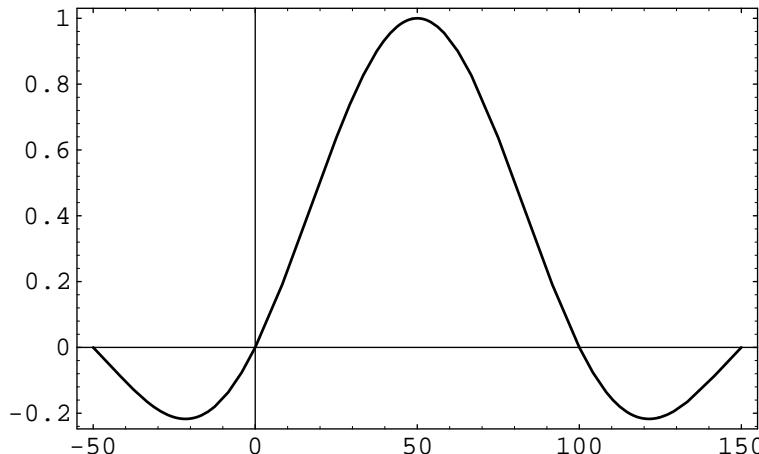
Parabolic interpolation of a Sinc function peak

If the analysis window being used for determining the audio spectrum is a rectangle window, a single frequency will have the form of a (digital) sinc function in the spectral domain:

```
fsinc[x_] := If[x == 0, 1, Sin[Pi x/50]/(Pi x/50)]
```

For example, the plot below could represent the spectrum of a single (complex) frequency component in the analysis window at 50 Hz, with a sampling rate of 200 Hz:

```
Plot[fsinc[x-50], {x,-50,150}, PlotRange->All, Frame->True];
```



This is a sinc function with a peak centered at 50, with a main-lobe width of 100. The following code samples this sinc function at various offsets. The example, as set up, has the equivalent of a zero padding factor of 2 (interp = 2), and an offset of 44 (offset of 50 means that frequency bin is exactly at peak of sinc.)

```
interp = 2; offset = 44;
bin = 100; width = bin/(2*interp);
samples =
  Chop[N[SampleFunction[fsinc[x-bin/2],
    {x,0-offset, 3bin/2, width}]]];
pp = Transpose[
  {(width Range[0,Length[samples]-1]) - offset,
   samples}];
points = Map[Point, pp]; Print["Predicted Peak:"]
maxpos = First[First[Position[samples,Max[samples]]]];
guess = First[ParabolaPeak[pp[[maxpos-1]],
  pp[[maxpos]], pp[[maxpos+1]]]];
Print["Actual Peak is 50, Relative error in percent follows:"]
error = 100.0 * Abs[guess - bin/2]/(bin/2.0)
Block[{$DisplayFunction = Identity},
  plot = Show[{
```

```

    Plot[fsinc[x-50], {x,-bin/2,3bin/2},
        PlotRange->All, Frame->True],
    Graphics[{PointSize[0.04], points}],
    ParabolaFitPlot[pp[[maxpos-1]],
        pp[[maxpos]], pp[[maxpos+1]],
        PlotStyle->{RGBColor[0,1,0]}]
  }];
Show[plot];

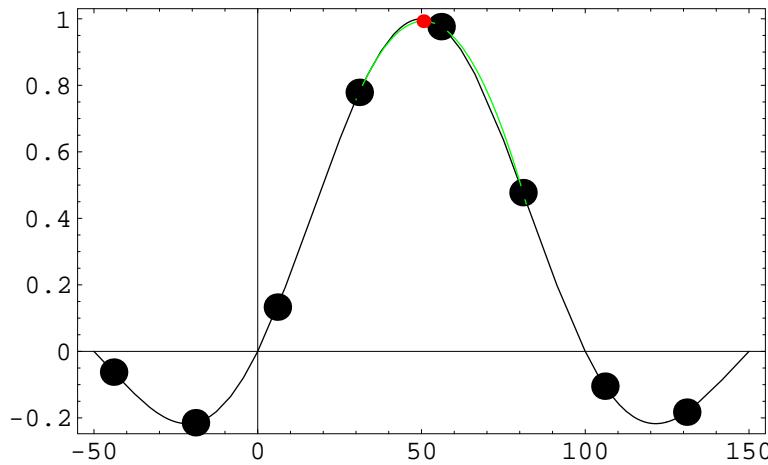
```

Predicted Peak:

50.5914

Actual Peak is 50, Relative error in percent follows:

1.1829



Zeropadding less than a factor of two gives visibly poor results:

```

interp = 1;
offset = 33;
bin = 100;
width = bin/(2*interp);
samples =
  Chop[N[SampleFunction[fsinc[x-bin/2],
    {x,0-offset, 3bin/2, width}]]];
pp = Transpose[
  {(width Range[0,Length[samples]-1]) - offset,
   samples}];
points = Map[Point, pp];
Print["Predicted Peak:"]
maxpos = First[First[Position[samples,Max[samples]]]];
guess = First[ParabolaPeak[pp[[maxpos-1]],
  pp[[maxpos]], pp[[maxpos+1]]]]
Print["Actual Peak is 50, Relative error in percent follows:"]
error = 100.0 * Abs[guess - bin/2]/(bin/2.0)

Block[{$DisplayFunction = Identity},
  plot = Show[{

```

```

        Plot[fsinc[x-50], {x,-bin/2,3bin/2},
          PlotRange->All, Frame->True],
        Graphics[{PointSize[0.04], points}],
        ParabolaFitPlot[pp[[maxpos-1]],
          pp[[maxpos]], pp[[maxpos+1]],
          PlotStyle->{RGBColor[0,1,0]}]
      }
];
Show[plot];

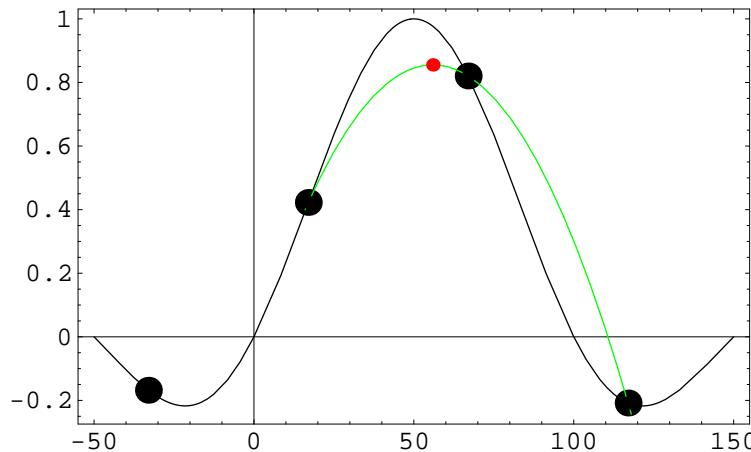
```

Predicted Peak:

55.9438

Actual Peak is 50, Relative error in percent follows:

11.8876



Zero padding a windowed signal will interpolate frequency bins in the spectrum. The more zero padding, the better the estimation of the peak when fitting the peak with a parabola. Shown below are tables of typical relative errors with zeropad factors. Frequency discrimination for humans is about 0.5 percent, so the detection of frequency peaks should be below that value. Zero padding by a factor of 4 would be good for this case.

Typical parabolic interpolation errors (Relative errors as percentages)

Zero Padding	Offset	Error
1	50	0.
1	49	0.999
1	48	1.99
1	47	2.98
1	46	3.95
1	45	4.9
1	44	5.82
1	43	6.71
1	42	7.57
1	41	8.38

1	40	9.13
1	39	9.82
1	38	10.4
1	37	11.
1	36	11.4
1	35	11.7
1	34	11.9
1	33	11.9
1	32	11.7
1	31	11.3
1	30	10.6
1	29	9.55
1	28	8.1
1	27	6.12
1	26	3.49

Zero Padding	Offset	Error
2	50	0.
2	49	0.247
2	48	0.485
2	47	0.707
2	46	0.903
2	45	1.07
2	44	1.18
2	43	1.25
2	42	1.24
2	41	1.16
2	40	0.979
2	39	0.689
2	38	0.267
2	37	0.267
2	36	0.689
2	35	0.979
2	34	1.16
2	33	1.24
2	32	1.25
2	31	1.18
2	30	1.07
2	29	0.903
2	28	0.707
2	27	0.485
2	26	0.247

Zero Padding	Offset	Error
3	50	0.
3	49	0.108
3	48	0.208
3	47	0.289
3	46	0.343
3	45	0.36
3	44	0.328
3	43	0.237
3	42	0.0731
3	41	0.137
3	40	0.275
3	39	0.345
3	38	0.359
3	37	0.329
3	36	0.265
3	35	0.176
3	34	0.0729
3	33	0.0366
3	32	0.143

3	31	0.238	
3	30	0.311	
3	29	0.354	
3	28	0.355	
3	27	0.305	
3	26	0.191	
4	Zero Padding	Offset	Error
4		50	0.
4		49	0.0602
4		48	0.111
4		47	0.144
4		46	0.148
4		45	0.114
4		44	0.03
4		43	0.0788
4		42	0.136
4		41	0.15
4		40	0.13
4		39	0.0875
4		38	0.0307
4		37	0.0307
4		36	0.0875
4		35	0.13
4		34	0.15
4		33	0.136
4		32	0.0788
4		31	0.03
4		30	0.114
4		29	0.148
4		28	0.144
4		27	0.111
4		26	0.0602
5	Zero Padding	Offset	Error
5		50	0.
5		49	0.038
5		48	0.0666
5		47	0.0765
5		46	0.0577
5		45	0.
5		44	0.0577
5		43	0.0765
5		42	0.0666
5		41	0.038
5		40	0.
5		39	0.038
5		38	0.0666
5		37	0.0765
5		36	0.0577
5		35	0.
5		34	0.0577
5		33	0.0765
5		32	0.0666
5		31	0.038
5		30	0.
5		29	0.038
5		28	0.0666
5		27	0.0765
5		26	0.0577

Zero Padding	Offset	Error
6	50	0.
6	49	0.0259
6	48	0.0424
6	47	0.04
6	46	0.00873
6	45	0.0333
6	44	0.0442
6	43	0.0329
6	42	0.00909
6	41	0.0178
6	40	0.0385
6	39	0.0435
6	38	0.023
6	37	0.023
6	36	0.0435
6	35	0.0385
6	34	0.0178
6	33	0.00909
6	32	0.0329
6	31	0.0442
6	30	0.0333
6	29	0.00873
6	28	0.04
6	27	0.0424
6	26	0.0259

Zero Padding	Offset	Error
7	50	0.
7	49	0.0186
7	48	0.0278
7	47	0.0179
7	46	0.0144
7	45	0.0278
7	44	0.0207
7	43	0.00287
7	42	0.0163
7	41	0.0274
7	40	0.0209
7	39	0.0103
7	38	0.0273
7	37	0.0226
7	36	0.00572
7	35	0.0138
7	34	0.0267
7	33	0.0233
7	32	0.00547
7	31	0.0265
7	30	0.0242
7	29	0.00851
7	28	0.0112
7	27	0.0256
7	26	0.0251

Zero Padding	Offset	Error
8	50	0.
8	49	0.0139
8	48	0.0183
8	47	0.00366
8	46	0.0168
8	45	0.0162
8	44	0.00383

8	43	0.0109
8	42	0.0186
8	41	0.00964
8	40	0.014
8	39	0.0178
8	38	0.00752
8	37	0.00752
8	36	0.0178
8	35	0.014
8	34	0.00964
8	33	0.0186
8	32	0.0109
8	31	0.00383
8	30	0.0162
8	29	0.0168
8	28	0.00366
8	27	0.0183
8	26	0.0139

Zero Padding	Offset	Error
9	50	0.
9	49	0.0106
9	48	0.0118
9	47	0.00481
9	46	0.013
9	45	0.0065
9	44	0.00528
9	43	0.0129
9	42	0.00676
9	41	0.0109
9	40	0.0114
9	39	0.00135
9	38	0.00973
9	37	0.0124
9	36	0.00256
9	35	0.013
9	34	0.00766
9	33	0.004
9	32	0.0125
9	31	0.00841
9	30	0.00979
9	29	0.012
9	28	0.00269
9	27	0.00874
9	26	0.0128

Zero Padding	Offset	Error
10	50	0.
10	49	0.0083
10	48	0.00713
10	47	0.00713
10	46	0.0083
10	45	0.
10	44	0.0083
10	43	0.00713
10	42	0.00713
10	41	0.0083
10	40	0.
10	39	0.0083
10	38	0.00713
10	37	0.00713
10	36	0.0083
10	35	0.

10	34	0.0083
10	33	0.00713
10	32	0.00713
10	31	0.0083
10	30	0.
10	29	0.0083
10	28	0.00713
10	27	0.00713
10	26	0.0083

‡ Code for generating the previous tables

```

interp = 1;
ans = {};
For[offset = 50, offset > 25, offset--,
  bin = 100;
  width = bin/(2*interp);
  samples =
    Chop[N[SampleFunction[fsinc[x-bin/2],
      {x,0-offset, 3bin/2, width}]]];
  pp = Transpose[
    {(width Range[0,Length[samples]-1]) - offset,
     samples}];
  points = Map[Point, pp];
  maxpos = First[First[Position[samples,Max[samples]]]];
  guess = First[ParabolaPeak[pp[[maxpos-1]],
    pp[[maxpos]], pp[[maxpos+1]]]];
  AppendTo[ans, {offset,
    100.0 * Abs[guess - bin/2]/(bin/2.0) }]
]
{a, b} = Transpose[ans];
b = N[b, 3];
TableForm[Transpose[{Table[interp, {i,1,Length[b]}], a, b}],
  TableSpacing->{0,9}, TableHeadings->{{""},{ "Zero Padding", "Offset", "Error "}}]

```