

NMR NOTES #13

Using Linear Prediction

Linear prediction is a software processing tool that examines a periodic function, such as an NMR free induction decay, extracts a set of coefficients, and extrapolates, either forward or backward to predict what the data would have done had it been collected. This is a potentially valuable technique for extracting useful information from marginal data, but it is also easily subject to abuse as well. It is not possible to collect a very small data set and then extrapolate that data set in order to get high resolution, just as if you had collected the data. If the data is not sampled sufficiently long to determine the difference in two frequencies, they will extrapolate as a single frequency with no improvement in resolution.

There are two cases where linear prediction is very useful. The first is to backward calculate the first several data points of a data set that are severely distorted by either receiver recovery problems or shaped excitation pulses. In this case somewhere between 2 and 6 points are calculated to replace the faulty data. If the first several data points are bad, the fourier transformed spectrum will have a large baseline offset or roll, making it difficult to phase and integrate spectra. The second case is to forward calculate data at the end of the fid in 2D data sets where the acquisition time is arbitrarily short. The parameters in either case are almost identical. Linear prediction parameters are added to a data set with the `parlp` command. `Parlp` is equivalent to `parlp(1)` and adds parameters for the first dimension. `Parlp(2)` adds parameters for the second dimension, etc. The `dglp` command will display all of the defined linear prediction parameters in a tabular form.

In order to perform a linear prediction calculation, it is necessary to calculate a set of coefficients describing the frequencies present. `lpfilt` defines the number of frequency components to analyze for and `lpnupts` defines the number of data points to use to calculate these coefficients. `lpnupts` must be at least twice `lpfilt`, and should probably be 4-8 times `lpfilt`, depending on the noise level in the data.

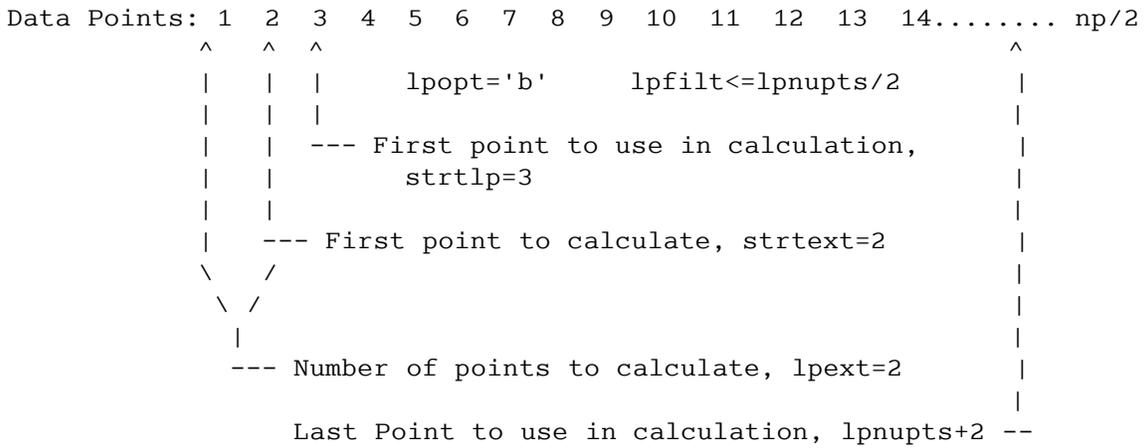
`lpopt` is the option parameter that defines whether you are doing forward or backward calculations. It can take only 'f' or 'b' as values. The value of `lpopt` defines the sense of the `strtlp` and `strtext` parameters. `strtlp` is the data point number to use to start calculating the necessary coefficients. This will be the first data point used in the case of a backward calculation, and the last data point used in the case of a forward calculation. The `strtext` parameter defines the starting data point number for the extrapolated data. In the case of a forward calculation, this will be the first data point added, and in the case of a backward calculation it will be the last data point calculated.

The `lpext` parameter defines the number of data points to be calculated and included in the data set. This number should be no more than `lpnupts`, the number of points used to calculate the coefficients. The `lpalg` parameter defines the algorithm used to calculate the coefficients and should always be 'lpfft'. The only other valid value is 'lparfft' which can handle noisier data and broader lines, but cannot model narrow lines.

Finally, the `proc` parameter defines the processing used by the `wft` command. A value of 'ft' does the normal fourier transform, and a value of 'lp' will do the linear prediction instead.

The following diagrams are a visual representation of the inter-relationship of these various parameters. All of the parameters are measured in complex data points. When performing 2D hypercomplex data acquisitions, the ni parameter counts the number of complex points collected in the t1/f1 dimension. In both 1D and 2D spectroscopy, the np parameter indicates the number of computer data points collected, real and imaginary, not complex points. Consequently there are np/2 complex points collected. The first example demonstrates doing a backward prediction on a 1D data set to correct the first 2 data points in order to improve the baseline. The second example shows the parameters necessary to forward predict a 2D data set in order to more fully use the increments collected.

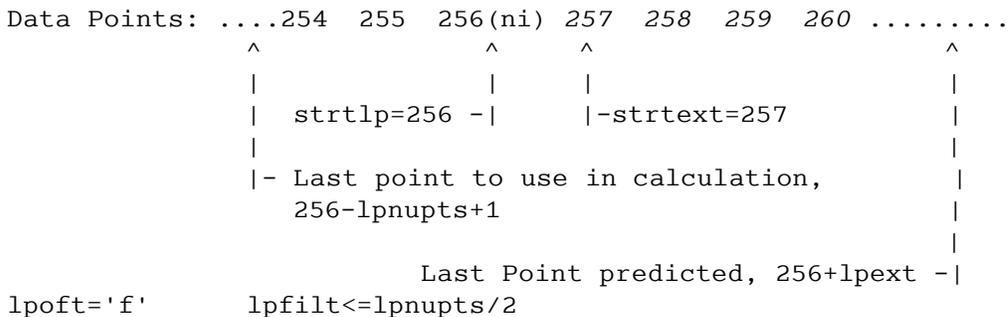
Example 1: 1D backward Linear Prediction:



```

Use:  lpopt='b'      lpalg='lpfft'
      lpfilt=16
      strtlp=3      lpnupts=256
      strttext=2    lpext=2
  
```

Example 2: 2D Forward Linear Prediction:



```

Use:  lpopt='f'      lpalg='lpfft'
      lpfilt=8
      strtlp=256     lpnupts=128
      strttext=257   lpext=128 or 256
  
```