SPLIB
A Spectral Analysis Subroutine Library
User's Guide and Reference

Remko Scharroo

March 5, 2007

Delft Institute for Earth-Oriented Space Research
Delft University of Technology
Kluyverweg 1, 2629 HS Delft
The Netherlands

Contents

1 Introduction **1**

2 Subroutine Synopsis **2**
2.1 SPCDEM – Complex demodulation of a series of Real data 2
2.2 SPCOMP – Single Discrete Fourier Transform (DFT) component of Real data . . 3
2.3 SPDFTC – Forward/Reverse Discrete Fourier Transform (DFT) of Complex data. 3
2.4 SPDFTR – Forward/Reverse Discrete Fourier Transform (DFT) of Real data. . . . 4
2.5 SPECTR – Determine spectral lines in a non-equally spaced time series. 4
2.6 SPFAMP – Forward/Reverse change between Fourier Transform and Series. 6
2.7 SPFFTC – Forward/Reverse Fast Fourier Transform (FFT) of Complex data. . . . 6
2.8 SPFFTR – Forward/Reverse Fast Fourier Transform (FFT) of Real data. 7
2.9 SPFPER – Compute Fast Lomb-Scargle normalized periodogram. 8
2.10 SPRAND – Generate random number 9
2.11 STATIS – Compute statistics of a series 9

1 Introduction

SPLIB is a subroutine library that contains some handy routines to be used for spectral analyses, Fourier Transforms, etc.

The SPLIB library has been ported to various systems. Presently it is only available on the IBM Risk 6000 workstations of the Section Space Research and Technology and the Convex Supercomputer of the Central Computing Facility of the Delft University.

The library can be linked to any of your FORTRAN programs by specifying including the library in your FORTRAN link step, *e.g.*

```
f77 example.f -o example /user/altim/lib/rssubs.a
```

The SPLIB library is courtesy of Remko Scharroo, Delft University of Technology, Section Space Research & Technology (E-mail: `Remko.Scharroo@lr.tudelft.nl`)

2 Subroutine Synopsis

2.1 SPCDEM – Complex demodulation of a series of Real data

```
    SUBROUTINE SPCDEM (N, X, Y, F, NSMTH)
    INTEGER    N, NSMTH
    REAL*8     X(0:N-1), F
    COMPLEX*16 Y(0:N-1)
 or REAL*8     Y(0:2*N-1)

This routine performs a complex demodulation of a single frequency
component in a series of N equally spaced real data points X(0) through
X(N-1). This means that the varying amplitude and phase of the
signal component with frequency F is determined and subtracted from the
data. On return, Y(0) through Y(N-1) will contain the complex amplitude
of the signal with frequency F at data points X(0) through X(N-1).
When interpreted as REAL*8, Y(0) and Y(1) are the cosine and sine
amplitudes in X(0), etc.
X(0) through X(N-1) will then be replaced by the demodulated signal,
i.e. real values with the modulated signal removed.
The smoothing factor NSMTH determines how many cycles are used for the
signal demodulation. Optimally this should be 1, but higher values
are allowed to increase the smoothing.

Arguments:
  N      (input): Number of data points.
  X      (input): Real values of the N data points.
        (output): Demodulated signal in each of the N data points.
  Y     (output): Modulation (complex amplitude) of the signal component with
                  frequency F.
  F      (input): Frequency associated with the signal to be removed
                  (measured in sampling frequencies).
  NSMTH (input): Smoothing factor (default=1)
```

2.2 SPCOMP – Single Discrete Fourier Transform (DFT) component of Real data

```
COMPLEX*16 FUNCTION SPCOMP (X, N, F)
INTEGER*4  N
REAL*8     X(0:N-1), F
```

```
This routine computes a single DFT component of a series of N equally spaced
real data points X(0) through X(N-1).
The Discrete Fourier Transform of X for one specific frequency F (with 1 being
the sampling frequency) is a COMPLEX*16 value SPCOMP, whose real part is the
cosine component and imaginary part is the sine component.
The number of data points may be even or odd, but must be at least 2.

Arguments:
  X       (input): Real values of the N data points.
  N       (input): Number of data points.
  F       (input): Frequency associated with the DFT (measured in sampling
                   frequencies).
  SPCOMP (output): Discrete Fourier Transform of X for frequency F.
```

2.3 SPDFTC – Forward/Reverse Discrete Fourier Transform (DFT) of Complex data.

```
SUBROUTINE SPDFTC (X, Y, N, ISIGN)
INTEGER    N, ISIGN
COMPLEX*16 X(0:N-1), Y(0:N-1)
```

```
This routine computes the DFT of a series of N equally spaced complex data
points X(0) through X(N-1), or the performs the reverse operation.
The Discrete Fourier Transform of the COMPLEX*16 array X is stored in the
COMPLEX*16 array Y, which may not be the same as array X.
The elements Y(0) through Y(N-1) denote the components of the DFT, from
frequency = 0 up to (N-1)/N.
Use ISIGN=-1 for the forward DFT, ISIGN=+1 for the reverse DFT.
When the reverse DFT is computed, Y is transformed back into N data points X
scaled by a factor N.
The number of data points may be even or odd, but must be at least 2.

Arguments:
  N      (input): Number of data points.
  ISIGN  (input): = -1 for forward DFT.
      X  (input): Complex values of the N data points.
      Y (output): Discrete Fourier Transform of X.
  ISIGN  (input): = +1 for reverse DFT.
      X (output): Complex values of the N data points scaled by a factor N.
      Y  (input): Discrete Fourier Transform of X.
```

2.4 SPDFTR – Forward/Reverse Discrete Fourier Transform (DFT) of Real data.

```
    SUBROUTINE SPDFTR (X, Y, N, ISIGN)
    INTEGER*4  N, ISIGN
    REAL*8     X(0:N-1)
    COMPLEX*16 Y(0:N/2)
 or REAL*8     Y(0:N+1)
```

```
This routine computes the DFT of a series of N equally spaced real data points
X(0) through X(N-1), or the performs the reverse operation.
The Discrete Fourier Transform of X is stored in array Y, which may not be the
same as array X. The real part of the COMPLEX*16 elements Y(0) [frequency = 0]
through Y(N/2) [Nyquist frequency] are the cosine components of the DFT; the
imaginary parts are the sine components. When interpreted as a REAL*8 array,
Y(0) through Y(N+1) are the respective cosine and sine components of the DFT.
Use ISIGN=-1 for the forward DFT, ISIGN=+1 for the reverse DFT.
When the reverse DFT is computed, Y is transformed back into N data points X
scaled by a factor N.
The number of data points may be even or odd, but must be at least 2.

Arguments:
  N      (input): Number of data points.
  ISIGN  (input): = -1 for forward DFT.
      X  (input): Real values of the N data points.
      Y (output): Discrete Fourier Transform of X.
  ISIGN  (input): = +1 for reverse DFT.
      X (output): Real values of the N data points scaled by a factor N.
      Y  (input): Discrete Fourier Transform of X.
```

2.5 SPECTR – Determine spectral lines in a non-equally spaced time series.

```
    SUBROUTINE SPECTR (N, T, X, M, F, A, B, PROB, SIGMA, L, FL, PL,
   .                   OFAC, HIFAC, FALSE, RESID, VERBOSE)
    INTEGER*4 N, M, L
    REAL*8    T(N), X(N), F(M), A(M), B(M), PROB(M), SIGMA(0:M),
   .          FL(L), PL(L), OFAC, HIFAC, FALSE, RESID
    LOGICAL   VERBOSE
```

```
Compute the frequency, sine- and cosine-amplitude of the major spectral
lines in a time series X(t). Lines that do not reach significantly above
the continuum of the background noise will not appear in the spectrum.
To separate the spectral lines from the continuum, the user has to specify
a so-called ‘false alarm probability’, i.e. the probability that lines
from the continuum emerge in the selected spectrum. If the probability is
set too low, physically significant peaks may be missed; if set too high,
peaks due to noise will be mislabeled as signal peaks. Values between 0.05
and 0.01 are suggested for this ‘false alarm probability’, corresponding to
one chance in 20 or one chance in 100 of mistaking a noise peak for signal,
provided the noise is Gaussian.
The spectral lines are isolated one by one using the Lomb-Scargle normalized
```

periodogram. In every iteration the frequency of the largest peak in the periodogram is determined and tested against the false alarm probability. If the peak is significant, the amplitude and phase of the sinusoid at the peak frequency is determined by least squares. Afterwards the peak sinusoid is subtracted from the series point by point. This procedure is repeated until no further peaks meet the false alarm probability. Thus we have

```
          m
  X(t) = sum [ A  cos (2 pi f t) + B  sin (2 pi f t) ] + NOISE
          j    j            j      j            j
```

The frequencies that may be extracted range from zero up to HIFAC times the average Nyquist frequency, being 0.5/(average time interval). The resolution of the frequencies to be extracted can be increased by increasing the oversampling factor OFAC; OFAC=1 refers to a resolution equal to the Fejer frequency, i.e. 1/(total time interval).

The iteration performed to select peaks in the periodogram is terminated when either:

```
1. A peak does not reach the ‘false alarm probability’ as explained above.
2. The residual noise after removing the selected signals has a std.dev.
   which is less than RESID times the std.dev. of the a priori signal.
   (RESID has to be specified by the user.)
3. The maximum number of frequencies to be isolated (M) is reached.
```

```
Arguments:
 N        (input) : Number of points in the time series.
 T        (input) : The independent variable (time).
 X        (input) : Values in each point.
         (output) : Residual noise after removal of all significant signals.
 M        (input) : Maximum number of frequencies to be isolated.
         (output) : Number of isolated frequencies that meet the false alarm
                    probability.
 F       (output) : Frequencies of the spectral lines in 1/(same units as T).
 A       (output) : Cosine amplitude for each spectral line in same units as X.
 B       (output) : Sine amplitude for each spectral line in same units as X.
 PROB    (output) : False alarm probability of each spectral line.
 SIGMA   (output) : Standard deviation of the signal remaining after removal
                    of each spectral line, in the same units as X.
                    At return SIGMA(0) will contain the A PRIORI sigma.
 L        (input) : Dimension of the working spaces FL and PL.
         (output) : Number of samples in the last computed periodogram.
 FL      (output) : Frequency of each sample in this periodogram.
 PL      (output) : Spectral density in this periodogram (dB).
 OFAC     (input) : Oversampling factor (typical value: 4 or larger).
 HIFAC    (input) : Maximum frequency measured in Nyquist frequencies.
 FALSE    (input) : Limit of false alarm probability (see above).
 RESID    (input) : Maximum fraction of noise remaining (see above).
 VERBOSE (input) : If .TRUE., type out period, amplitude and phase (deg)
                    during iteration.
```

2.6 SPFAMP – Forward/Reverse change between Fourier Transform and Series.

```
    SUBROUTINE SPFAMP (X, N, ISIGN)
    INTEGER*4  N, ISIGN
    COMPLEX*16 X(0:N/2)
 or REAL*8     X(0:N+1)
```

```
This routine computes the cosine and sine amplitudes for each frequency
component of a Fourier series out of the Fourier Transform, or vice versa.
The array X is either COMPLEX*16 or REAL*8. When interpreted as REAL*8
X(0) through X(N+1) are the respective cosine and sine components of the
Fourier Transform, or the cosine and sine amplitudes of the Fourier Series.

Arguments:
  N      (input): Number of data points.
  ISIGN  (input): = -1 for forward transformation.
      X  (input): Fourier Transform components.
        (output): Fourier Series amplitudes.
  ISIGN  (input): = +1 for reverse transformation.
      X  (input): Fourier Series amplitudes.
        (output): Fourier Transform components.
```

2.7 SPFFTC – Forward/Reverse Fast Fourier Transform (FFT) of Complex data.

```
    SUBROUTINE SPFFTC (X, N, ISIGN)
    INTEGER*4  N, ISIGN
    COMPLEX*16 X(0:N-1)
```

```
This routine computes the FFT of a series of N equally spaced complex data
points X(0) through X(N-1), or the performs the reverse operation.
The Fast Fourier Transform of the COMPLEX*16 array X is stored back into the
array X itself. The elements X(0) through X(N-1) denote the components of the
FFT, from frequency = 0 up to (N-1)/N.
Use ISIGN=-1 for the forward FFT, ISIGN=+1 for the reverse FFT.
When the reverse FFT is computed, the Fourier transform of X is transformed
back into N data points X scaled by a factor N.
The number of data points must be a power of 2, i.e. 2, 4, 8, etc.

Arguments:
  N      (input): Number of data points.
  ISIGN  (input): = -1 for forward FFT.
      X  (input): Complex values of the N data points.
        (output): Fast Fourier Transform of X.
  ISIGN  (input): = +1 for reverse FFT.
      X  (input): Fast Fourier Transform.
        (output): Complex values of the N data points scaled by a factor N.
```

2.8 SPFFTR – Forward/Reverse Fast Fourier Transform (FFT) of Real data.

```
    SUBROUTINE SPFFTR (X, N, ISIGN)
    INTEGER*4  N, ISIGN
    REAL*8     X(0:N+1)

This routine computes the FFT of a series of N [=2**k] equally spaced real
data points X(0) through X(N-1), or it performs the reverse operation.
The Fast Fourier Transform of X is stored back into array X. The elements
X(0) through X(N+1) are the respective cosine and sine components of the FFT,
starting with the zero-frequency pair [X(0),X(1)], up to the Nyquist frequency
components [X(N),X(N+1)].
Use ISIGN=-1 for the forward FFT; ISIGN=+1 for the reverse FFT.
When the reverse FFT is computed, the Fourier transform is transformed back
into N data points X scaled by a factor N.
The number of data points must be a power of 2, i.e. 2, 4, 8, etc.

Arguments:
  N      (input): Number of data points.
  ISIGN  (input): = -1 for forward FFT.
      X  (input): Real values of the N data points.
        (output): Fast Fourier Transform of X.
  ISIGN  (input): = +1 for reverse FFT.
      X  (input): Fast Fourier Transform.
        (output): Real values of the N data points scaled by a factor N.
```

2.9 SPFPER – Compute Fast Lomb-Scargle normalized periodogram.

```
    SUBROUTINE SPFPER (X, Y, N, YMEAN, YSIGMA, OFAC, HIFAC,
   |                   WK1, WK2, NWK, NOUT, JMAX, PROB)
    INTEGER*4 N, NWK, NOUT, JMAX
    REAL*8    X(N), Y(N), YMEAN, YSIGMA, OFAC, HIFAC,
   |          WK1(0:NWK-1), WK2(0:NWK-1), PROB
```

```
Given N data points with abscissas X (which need not be equally spaced,
but must be sequential) and ordinates Y (with mean YMEAN and standard dev
YSIGMA), and given a desired oversampling factor OFAC (a typical value
being 4 or larger), this routine computes the normalized periodogram
of the data points using the Lomb-Scargle method (see reference below).

The routine fills array WK1 with a sequence of NOUT increasing frequencies
(not angular frequencies but in units of 1 over the unit of X) up to HIFAC
times the ‘average’ Nyquist frequency, and fills array WK2 with the values
of the Lomb-Scargle normalized periodogram at those frequencies. The
arrays X and Y are not altered.

NWK, the dimension of WK1 and WK2, must be large enough for intermediate
work space, or an error (pause) results. The routine also returns JMAX,
such that WK2(JMAX) is the maximum element in WK2, and PROB, an estimate
of the significance of that maximum against the hypothesis of random noise.
A small value of PROB indicates that a significant periodic signal is
present.

In order to infer the amplitudes AMP from the normalized periodogram WK2,
compute AMP(K) = 2 * YSIGMA * SQRT(WK2(K)/N).

Arguments:
  X      (input): abscissas of the data points (e.g. time).
  Y      (input): N real data points, need not be equally spaced.
  N      (input): Number of data points.
  YMEAN  (input): Average of the N data values.
  YSIGMA (input): Standard deviation of the N data values.
  OFAC   (input): Oversampling factor.
  HIFAC  (input): Determines the highest frequency in the periodogram (in
                  average Nyquist units).
  WK1   (output): Frequencies (NOT angular, in 1/units of X)
  WK2   (output): Normalized periodogram.
  NWK    (input): Size of the working spaces as defined in the calling
                  (sub)program.
  NOUT  (output): Number of peaks in periodogram.
  JMAX  (output): Index of highest peak in the periodogram.
  PROB  (output): Significance of the highest peak in the periodogram.

Ref: Press W.H. and George B. Rybicki.  Fast algorithm for spectral analysis
of unevenly sampled data, The Astronomical Journal, 338, 227-280, 1989.
```

2.10 SPRAND – Generate random number

```
REAL*8 FUNCTION SPRAND()
```

```
This routine generates a random number in the range [0,1]. The expected
distribution is uniform.

Arguments:
  SPRAND (output): The random number in the range [0,1]
```

2.11 STATIS – Compute statistics of a series

```
SUBROUTINE STATIS (N, X, XMEAN, XRMS, XSIGMA)
INTEGER N
REAL*8  X(N), XMEAN, XRMS, XSIGMA
```

```
This routine computes the average (XMEAN), root-mean-squate (XRMS) and
standard deviation (XSIGMA) of a series of N values (X).

Arguments:
 N      (input) : Number of values in the series.
 X      (input) : Series of values.
 XMEAN (output) : Average of the series.
 XRMS  (output) : RMS of the series.
 XSIGMA (output) : Standard deviation of the series.
```