Documentation for
A Fortran 90 library for multitaper spectrum analysis

1   Introduction

The present documentation is for the Fortran 90 Library for multitaper spectral analysis as presented in

> Prieto, G. A., R. L. Parker and F. L. Vernon (2008). A Fortran 90 library for multitaper spectrum analysis. Submitted to Computers and Geosciences, February 2008.

It is intended to be a comprehensive and easy to use collection of subroutines for univariate and multivariate spectral analysis. There are two main methods implemented here, the original adaptive multitaper spectrum (Thomson, 1982) and the sine multitaper method (Riedel and Sidorenko, 1995).

2   Requirements

The first requirement is to have a Fortran 90 or 95 compiler available in your machine. If you want to be able to obtain figures from the programs, *gplot* needs to be installed as well. This library has been tested using the Absoft Compiler in a Mac OS X system, and using the Sun Compiler in the Unix environment. More recently I am using the Intel compiler on an Intel Mac machine. Since we are taking advantage of features in Fortran 90 such as optional arguments and modules not available in F77, a F90/F95 compiler is needed. Most of the things you can do in F77 can be compiled by F90, so there is no reason not to use F90 instead (of course this is a personal opinion, not necessarily shared by my co-authors.).

A second requirement is that you have to edit the Makefile a little bit to adjust to your specific compiler options. There are small differences between compilers, especially the option to look for modules. You edit the Makefile to be able to search the modules used by this library. This release has a set of short programs to reproduce the figures presented in the paper (Prieto *et al.*, 2008).

3 Univariate subroutines

In this documentation the main subroutines calls are explained.

3.1 `mtspec`

This subroutine uses Thomson's spectrum estimation method (Thomson, 1982) and outputs the power spectral density (PSD), and if requested performs F-test for line components, returns such as a jackknife 95% confidence interval, the y_k's, which contain the phase information, the weights of the different eigenspectra, etc.

The basic way to call this subroutine is

```
call mtspec(npts,dt,x,tbp,kspec,nf,freq,spec)
```

A description of all the variables is presented in Table 1. This particular subroutine performs all calculations in double precision, but accepts both single as well as double precision arguments. Note that if double precision is requested, all variables need to be double precision (except integers).

Taking advantage of Fortran 90 optional arguments, the user can request additional outputs or different form of calculations (e.g., F-test line reshaping, different weighting, quadratic multitaper, etc.).

If for example the user would also like to have the jackknife error estimate, the call would be:

```
call mtspec(npts,dt,x,tbp,kspec,nf,freq,spec,err=jack)
```

which will return the appropriate jackknife estimate in the array `jack`, featuring the 5% and the 95% confidence intervals. A more complete description of the optional parameters is presented in Table 2.

Table 1: `mtspec` mandatory arguments

Var	Type	I/O	Dimension	Description
npts	integer	input	1x1	number of points in time series
dt	real	input	1x1	the sampling interval
x	real	input	npts x 1	the real time series
tbp	real	input	1x1	the time-bandwidth product
kspec	integer	input	1x1	the number of tapers to use
nf	integer	input	1x1	number freq. bins
freq	real	output	nf x 1	the real frequency vector
spec	real	output	nf x 1	the spectrum estimate

Table 2: `mtspec` optional arguments

Var	Double	I/O	Dimension	Description
yk	complex	output	npts x kspec	the eigencoefficients
wt	real	output	nf x kspec	weights
err	real	output	nf x 2	95% jackknife c.i.
se	real	output	nf x 1	ndf for each freq bin
sk	real	output	nf x kspec	the eigenspectra ($y_k{}^2$)
verb	integer	input	1 x 1	verbal option
				0 - default, no verbose
				1 - print various steps
qispec	integer	input	npts x kspec	Use quadratic method
				0 - default, standard method
				1 - quadratic method
adapt	integer	input	1 x 1	adaptive multitaper
				0 - default, adaptive MT
				1 - constant weights
rshape	integer	input	1 x 1	F-test periodic components
				1 - remove lines
				2 - remove lines around 60 Hz.
				other - reshape spectrum
fcrit	real	input	1 x 1	F-test threshold probability
				rshape present
fstat	real	output	nf	F statistics (rshape present)

3.2 `mtspec` with zero padding

In many cases it is advantageous to pad the data series with zeros. It that case, it is not recomended to pad the data before tapering, and I prefer to taper and then pad with zeros. This is performed internally inside the subroutine by calling

```
call mtspec(npts,nfft,dt,x,tbp,kspec,nf,freq,spec)
```

where `nfft` $\geq$ `npts`. All other optional arguments can still be added as explained above.

Be aware that the number of frequency points `nf`, has to be either `npts`/2+1 or `nfft`/2+1, depending on the type of call. The output will then be a frequency vector `freq(nf)` containing all the frequency bins, and the spectrum `spec(nf)` with the power spectrum of the time series.

3.3 `sine_psd`

The subroutine is in charge of estimating the adaptive sine multitaper as in Riedel and Sidorenko (1995). If requested it will also calculate the sine multi-

taper with a constant number of tapers. The basic call is:

```
call sine_psd (npts,dt,x,ntap,ntimes,fact,nf,freq,spec)
```

The arguments are explained in Table 3. If the user wants a constant number of tapers, just have the integer `ntap` be greater than 1. This parameter overrides the following two, so if an adaptive sine multitaper is required, `ntap = 0`.

Table 3: sine_psd mandatory arguments

Var	Type	I/O	Dimension	Description
npts	integer	input	1x1	number of points in time series
dt	real	input	1x1	the sampling interval
x	real	input	npts x 1	the real data series
ntap	integer	input	1x1	number of tapers to avg
				0 - default, adaptive method
ntimes	integer	input	1x1	number of iterations
fact	real	input	1x1	degree of smoothing
				1.0 - default
				range 0.0 to 1.0
nf	integer	input	1x1	number freq. bins
freq	real	output	nf x 1	the real frequency vector
spec	real	output	nf x 1	the spectrum estimate

The subroutine also has some optional arguments, including the number of tapers used at each frequency bin (useful for uncertainties), and the approximate 95% confidence intervals. The call would then be:

```
call sine_psd (npts,dt,x,ntap,ntimes,fact,nf,freq,spec,kopt,err)
```

Table 4 briefly provides information about these two arguments.

Table 4: sine_psd optional arguments

Var	Type	I/O	Dimension	Description
kopt	integer	output	nf x 1	number of tapers used
err	real	output	nf x 2	95% confidence intervals

4   Multivariate subroutines

Various multivariate subroutines are available, for coherence and transfer function estimation and deconvolution. Table 5 provides a complete description

of the arguments for all multivariate subroutines. To keep the documentation short, I only present examples of simple calls for all multivariate subroutines.

Using Thomson's multitaper method the user can find three subroutines with corresponding calls:

```
call mt_cohe( npts,dt,xi,xj,tbp,kspec,nf,p,freq,cohe)

call mt_transfer(npts,nfft,dt,xi,xj,tbp,kspec,nf,    &
                      freq,cohe,trf)

call mt_deconv ( npts,nfft,dt,xi,xj,tbp,kspec,nf,tfun)
```

To use the sine multitaper method for coherence analysis type:

```
call sine_cohe (npts,dt,xi,xj,ntap,ntimes,fact,nf,p,freq,cohe)
```

For a complete description of all mandatory and optional arguments, consult Table 5.

Germán A. Prieto
Stanford, California
March 25, 2008

References

Prieto, G. A., Vernon, F. L., Masters, G., and Thomson, D. J. (2005). Multitaper Wigner-Ville spectrum for detecting dispersive signals from earthquake records. In *Proc. of the Thirty-nine Asilomar Conf. on Signals, Systems and Computers*, pages 938–941. IEEE.

Prieto, G. A., Parker, R. L., and Vernon, F. L. (2008). A fortran 90 library for multitaper spectrum analysis. *Submitted to Computers and Geosciences*.

Riedel, K. S. and Sidorenko, A. (1995). Minimum bias multiple taper spectral estimation. *IEEE Trans. on Signal Processing*, **43**, 188–195.

Thomson, D. J. (1982). Spectrum estimation and harmonic analysis. In *Proceedings of the IEEE*, volume 70, pages 1055–1096.

Thomson, D. J. (1990). Quadratic-Inverse spectrum estimates: applications to paleoclimatology. *Phys. Trans. R. Soc. London A*, **332**, 539–597.

Thomson, D. J. (1994). An overview of multiple-window and quadratic-inverse spectrum estimation methods. In *ICASSP - 94*, volume VI, pages 185–194.

Table 5: Multivariate subroutines arguments

Var	Type	I/O	Dimension	Subroutines	Description
npts	integer	input	1 x 1	all	number of points in xi, xj
nfft	integer	input	1 x 1	mt_transfer, mt_deconv	allows for zero padding
dt	real	input	1 x 1	all	sampling interval
xi	real	input	npts x 1	all	real data series 1
xj	real	input	npts x 1	all	real data series 2
tbp	real	input	1 x 1	all, except sine_cohe	time-bandwidth product
kspec	integer	input	1 x 1	all, except sine_cohe	number of tapers
nf	integer	input	1 x 1	all	number of frequency bins
p	real	input	nf x 1	all	prob. of zero coherence
ntap	integer	input	1 x 1	only sine_cohe	number of tapers to avg
ntimes	integer	input	1 x 1	only sine_cohe	number of iterations
fact	real	input	1 x 1	only sine_cohe	degree of smoothing
freq	real	output, optional	nf x 1	all	real frequency vector
cohe	real	output, optional	nf x 1	all	coherence output
trf	complex	output, optional	nfft x 1	all	transfer function output
tfun	real	output, optional	nfft x 1	all	time domain deconv result
phase	real	output, optional	nf x 1	mt_cohe, sine_cohe	phase between two signals
speci	real	output, optional	nf x 1	all	spectrum of signal xi
specj	real	output, optional	nf x 1	all	spectrum of signal xj
conf	real	output, optiona;	nf x 1	mt_cohe, sine_cohe	bound for null hypothesis of no coherence
cohe_ci	real	output, optional	nf x 2	mt_cohe, sine_cohe	confidence interval for coherence
phase_ci	real	output,optional	nf x 2	mt_cohe, sine_cohe	confidence interval for phase
cspec	complex	output, optional	nf x 1	mt_transfer	complex cross-spectrum, positive frequencies only
iadapt	integer	input, optional	1 x 1	mt_transfer, mt_deconv	choose either adaptive or constant wieghts
demean	integer	input, optional	1 x 1	mt_transfer, mt_deconv	force zero mean in complex transfer function
spec_ratio	real	output, optional	nf x 1	only mt_deconv	spectral ratio of two series
fmax	real	input, optional	1 x 1	only mt_deconv	apply cosine filter with fmax
kopt	integer	output, optional	nf x 1	only sine_cohe	number of tapers used

6