TIMEFRQ 4.4

Time-frequency (spectrogram) analysis of unevenly spaced paleoclimatic time series

Manual (August 2018)

 The latest version of the program can be found at the following web site: `www.marum.de/en/Michael-Schulz/Michael-Schulz-Software.html`

The program has been tested, though not rigorously, and is correct to the best of my knowledge. If you find any errors or have any suggestions, I would appreciate if you would let me know:

Michael Schulz
MARUM and Dept. of Geosciences, Univ. Bremen,
Leobener Str. 8, D-28334 Bremen, FRG
mschulz@marum.de

TIMEFRQ was used in the following paper; please consider to cite this paper if you use the program to analyze your data for publications:

Schulz, M., Berger, W. H., Sarnthein, M. and Grootes, P. M. 1999: Amplitude variations of 1470-year climate oscillations during the last 100,000 years linked to fluctuations of continental ice mass. *Geophys. Res. Lett.*, **26**, 3385-3388.

DISCLAIMER OF WARRANTY

1. Program Description

TIMEFRQ estimates temporal amplitude changes of sinusoidal signal components in a given time series over a prescribed frequency range. The program utilizes a modified version of a harmonic-filtering algorithm (Ferraz-Mello, 1981), which fits a sinusoidal wave to a time series by means of least-squares. This method can process unevenly spaced time series directly, that is, without prior interpolation.

To obtain time-dependent amplitude estimates, an input time series is analyzed within a moving window of width `Tseg`:

$$\texttt{Tseg} = \texttt{wfac} \times \texttt{fitper},$$

where `wfac` is a width-factor and `fitper` denotes the signal periodicity of interest. For each frequency (or period) the window is shifted consecutively by one (or more) data point(s) along the time axis of the input time series. Each "window segment" is linearly detrended and can be tapered prior to the amplitude estimation, that is, fitting of the sinusoid to the data within the window. The resulting amplitude of the best-fit sinusoid is saved *vs.* the average of the observation times within the current segment. Note that in case of unevenly spaced time series, segments are of equal duration (`Tseg`) but do not necessarily contain the same number of data points.

A prescribed grid is used for `fitper`, such that `fstart` ≤ `1/fitper` ≤ `fstop`. The spacing of the frequency grid and, hence, the frequency resolution, is set by the user.

The dependence of the window width (`Tseg`) on frequency (`wfac` is constant) leads to a change in temporal resolution with frequency. At low frequencies (long periods) wide windows result in a low temporal resolution, whereas at high frequencies (short periods) narrow windows allow to distinguish short-term variations in signal strength. This "scale dependence" of the temporal resolution is similar to that of wavelet analysis. However, in contrast to existing wavelet programs, TIMEFRQ can process unevenly spaced time series directly.

The selection of `wfac` will be a compromise between minimizing the statistical error (large `wfac` required) and maximizing the temporal resolution (small `wfac` required). Depending on the signal-to-noise ratio, typical values for `wfac` are in the range 3 to 5. Note that due to the finite window width, a step-like increase in signal amplitude appears `wfac` × `fitper` wide (cf. Mudelsee and Schulz, 1997 and example below).

Another effect of the final window width is that signal components with periodicities below and above `fitper` will affect the estimation. A measure of this bias is the half-amplitude bandwidth

(BW), which is approximately given by:

$$BW = \beta \, / \, (\texttt{wfac} \times \texttt{fitper})$$

where the factor β depends on the applied taper:

Taper	β
Rectangular	1.14
Welch	1.54
Hanning	2.09
Triangular	1.70
Blackmann Harris	2.38

TIMEFRQ includes an option to assess the frequency response of the fit-algorithm for a center frequency of 0.5 × (`fstart` + `fstop`).

2. Installation/Contents of the ZIP Archive

Copy the ZIP-archive to an empty directory and unpack it, e.g. by entering the following command at the DOS prompt:

```
unzip timefrq44.zip
```

This should result in the following files in the TIMEFRQ directory:

`timefrq44.exe`	Executable file
`timefrq44.cfg`	Configuration file
`timefrq44.f90`	Fortran source code
`usage44.pdf`	Manual (this file)
`example.cfg`	Configuration file for example
`100k_jmp.dat`	Example data file
`example.plt`	GNUPLOT script for plotting the example results
`mkplot2.bat`	GMT script for plotting the example results

3. Running TIMEFRQ

TIMEFRQ was successfully tested under NT, 2000, XP, and Win10. All program options and parameters are set in a configuration file, which is passed to TIMEFRQ via the command line. An example configuration file is included (`timefrq4.cfg`). To run the program, open a DOS-Box (Start → Windows System → Command Prompt), change to the TIMEFRQ directory and enter a command line of the following structure:

```
timefrq44 myfile.cfg
```

3.1 Configuration File Format

The configuration file is in ASCII format and can be edited with any text editor. It is recommended that you copy the original timefrq44.cfg file to a working file in order to have a backup. The configuration file contains a Fortran 90 namelist, e.g.:

```
&cfg
       fnin = 'c:/data/myfile.dat',
      fnout = 'c:/data/myfile.tf',
     fstart = 0.001,
      fstop = 0.02,
         df = 0.0002,
       wfac = 4.0,
      dtout = 5.0,
     bwtest = .false.,
       gplt = .true.,
       iwin = 1
    /
```

(If you are unfamiliar with namelists, please note the following:

- a string `&cfg` in the first line and a single slash in the last line
- each data line, except the last, ends with a comma
- filenames must be enclosed in `'...'` or ″...″
- directories are marked by a normal slash and <u>not</u> by the usual DOS backslash
- namelist entries can be in lower or upper case
- comment lines are marked by a leading “!”)
- logical values can be entered as `T`, `.true.`, `F` or `.false.`
- use point (not a comma) in decimal numbers

The parameters in the namelist have the following meaning:

`fnin` Input filename (full path!) with time series data.

`fnout` Results are written to this file (ASCII format).

`fstart` Lower bound of investigated frequency range (unit = 1/(time unit of input data)).

`fstop` Upper bound of investigated frequency range.

`df` Frequency spacing in the interval [`fstart`, `fstop`].

`wfac` Window-width factor (typical values 3-5; fractional values, e.g. 3.5, are allowed).

`dtout` Time-interval by which the sliding window is moved along the time axis; if set to zero (= default) window is shifted by one data point per step (in units of input data).

`bwtest` Toggle bandwidth determination:

- If set to `T`: Estimate frequency response of the harmonic filter (useful for empirical determination of the bandwidth). Create 200 sinusoidal test signals (using the time axis of input data) with varying frequency (f_t; symmetric around 0.5 × (`fstart` + `fstop`)). For each test signal, the average amplitude of the harmonic filter at frequency f_t is written to the file `bw.plt` as function of f_t.
- If set to `F`: Skip bandwidth determination (default).

`iwin` Taper-type identifier:

- 0: Rectangular (= no taper = default)
- 1: Welch
- 2: Hanning
- 3: Triangular
- 4: Blackman-Harris.

`gplt` Controls output format:

- If set to `T`: Blocks with identical frequencies are separated by a blank line, for compatibility with GNUPLOT.
- If set to `F`: Continuous output, for compatibility with GMT.

Except `dtout`, `bwtest`, `iwin` and `gplt` all parameters must be specified.

3.2 Input Data Format

Time series data are read from space- or tab-delimited ASCII files of the following format:

```
# comment lines
    # .
    t(1)   x(1)
    t(2)   x(2)
     .       .
     .       .
    t(N)   x(N)
```

where $t(1) < t(2) < ... < t(N)$ denote the sampling times, which can be geological ages or physical times. The sampling times may be evenly or unevenly spaced. In decimal numbers, decimal digits MUST be separated by a point (not a comma even if your Windows environment supports this, e.g. 1.23 NOT 1,23). The maximum number of data points N is limited to 25,000. Program versions for larger values of N can be produced by changing the source code and re-compiling the program.

TIMEFRQ checks if the ages are in increasing order. If the program encounters decreasing ages, that is, $t(i+1) < t(i)$, it will stop. Identical age entries and the corresponding data values are automatically averaged. The averaged time series is written to the file `TimeSeries.avg`.

The input file must not contain more than two data columns. Make sure that the file contains no blank lines at the end of the file and within the data section. Comment lines are indicated by a leading # and are only allowed at the beginning of the file. The number of comment lines is unlimited.

4. Output

- Estimated parameters are written to the file defined by `FNOUT`.
- Warning messages are written to `NYQ_ERR.DAT`.
- Automatically averaged time series, generated from input files with duplicate age entries, are saved to `TimeSeries.avg`.

The main output file is ASCII-formatted an can be used plotted using GNUPLOT or GMT. Alternatively, it can be imported into most spreadsheets for plotting. The contents of the file should be largely self explanatory:

1. The header section lists the main settings from the configuration file. In addition, the number of frequencies being investigated (`NFrq`) is shown.

2. The line Occurrences of `fstop > local f_Nyq` indicates how often the spacing of the input time was insufficient to resolve the highest frequency of interest. That is, whether `fstop` exceeds the Nyquist frequency between any two sampling times. Times at which the Nyquist criterion is violated are listed in the file `NYQ_ERR.DAT`.

3. The three-column data section lists the Time-Frequency-Amplitude triplets. Here `Time` is the central time within a sliding window for which the instantaneous amplitude is given. `Amplitude` values have the same physical unit as the input data.

To plot the data matrix, a graphic program is required that can handle 3D-data sets (such as GMT or Surfer). Note that an unevenly spaced input time series leads to an unevenly spaced time vector in the output (in contrast, frequencies are always evenly spaced with step `df`). Since most graphics packages will display only gridded data sets, these programs will interpolate data matrix. If possible, it is recommended to use triangulation instead of kriging for this purpose.

5. Example Computation

The test data set (`100k_jmp.dat`) consists of a 100-ky sinusoidal wave, the amplitude of which jumps from zero to one at 1 My. To estimate this evolution of the signal amplitude in time-frequency space for `wfac = 4`, open a DOS-Box, change to the TIMEFRQ directory and enter the following command:

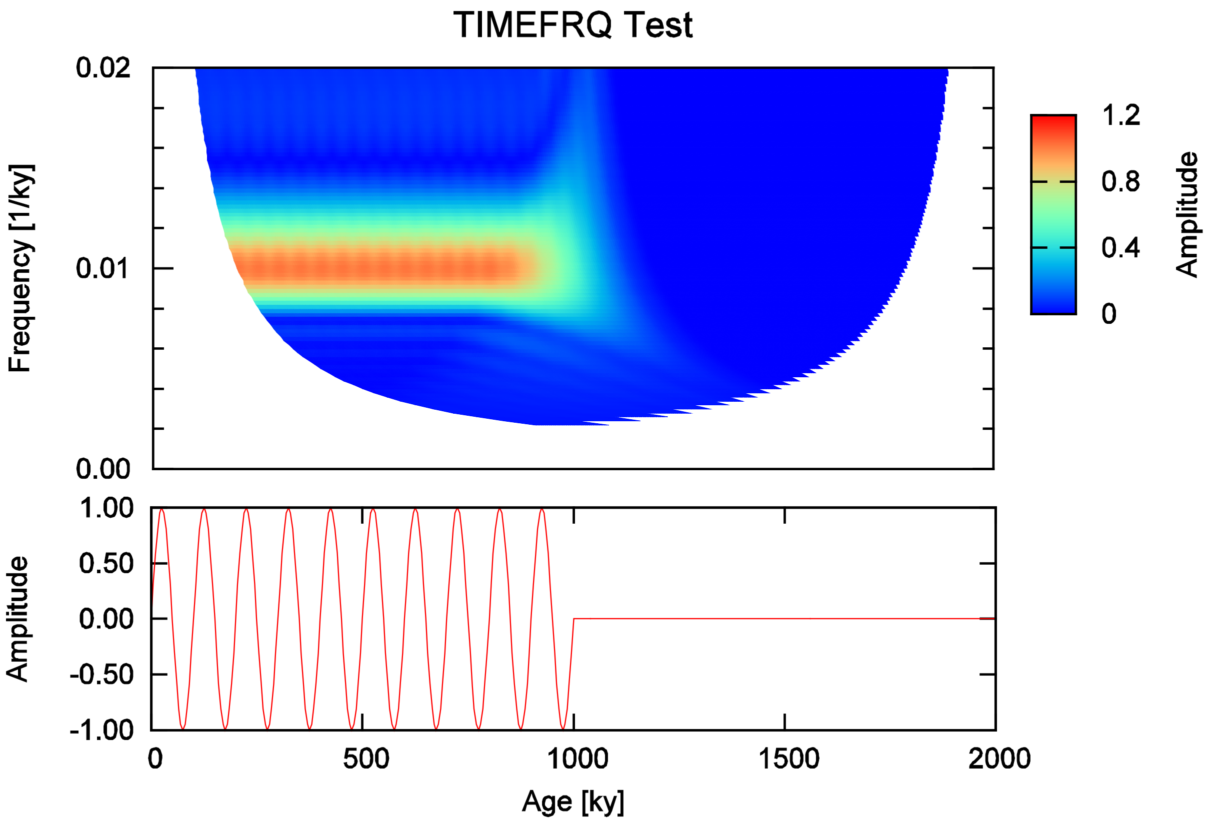
```
timefrq44 example.cfg
```

The results of the computation will be written to the file `100k_jmp.tf`. If the GNUPLOT plotting package is installed, the script file `example.plt` can be used to produce the attached figure of the results.

6. References

Ferraz-Mello, S. 1981: Estimation of periods from unequally spaced observations. Astron. J., **86**, 619-624.

Mudelsee, M. and Schulz, M. 1997: The Mid-Pleistocene climate transition: onset of 100 ka cycle lags ice volume build-up by 280 ka. Earth Planet. Sci. Lett., **151**, 117-123.


TIMEFRQ Test
0.02
0.01
0.00
Frequency [1/ky]
1.2
0.8
0.4
0
Amplitude
1.00
0.50
0.00
-0.50
-1.00
Amplitude
0
500
1000
1500
2000
Age [ky]