Users guide for the program yspec

Introduction

The program yspec calculates synthetic seismograms in spherically symmetric earth models using the direct radial integration method introduced by Friedrich & Dalkolmo (1995), and extended in Al-Attar & Woodhouse (2008) to incorporate the effects of self-gravitation.

Compilation

A makefile is included, so that compilation 'should' be straightforward. Various changes may, however, be needed to get things working depending on the compilers available ...

Running the code

The program is run using the command:

```
>> yspec yspec.in
```

where yspec.in is the name of the parameter file.

An example of this parameter file is shown below. Note that the line spacing in the parameter file does matter and so should not be changed. All read statements are, however, unformatted.

```
# This file contains the parameters needed to run the
# program yspec

# prefix for output files
  yspec.out

# Earth model
  prem.200

# attenuation switch: 1 = on, 0 = off
1

# gravitation: 0 = none, 1 = cowling, 2 = self
2

# output: 0 = displacement, 1 = velocity, 2 = acceleration
2

# potential and tilt corrections: 0 = no, 1 = yes
1

# lmin
  0

# lmax
  1500
```

```
# fmin (mHz) (this should always be greater than 0)
  0.2

# fmax (mHz)
  100

#  length of  time series (min)
  60

# time step (sec)
  1.0

# f11 filter (mHz)
  1

# f12 filter (mHz)
  2

# f21 filter (mHz)
  98

# f22 filter (mHz)
  100

# source depth (km)
  50.0

# source latitude (deg)
  0

# source longitude (deg)
  0

# M_{r,r} (Nm)
  1.0e+18

# M_{r,theta} (Nm)
  1.0e+18

# M_{r,phi} (Nm)
  1.0e+18

# M_{theta,theta} (Nm)
 -1.0e+18

# M_{theta,phi} (Nm)
  2.0e+18

# M_{phi,phi} (Nm)
  0.0e+18

# receiver depth
  3

# number of receivers
  18

# receiver latitudes and longitudes
```

```
0    10
0    20
0    30
0    40
0    50
0    60
0    70
0    80
0    90
0   100
0   110
0   120
0   130
0   140
0   150
0   160
0   170
0   180
```

A summary of the input parameters is as follows:

1. `prefix for output files` – a string used for labelling the outputs of the program. For example, if this string is `yspec.out`, then the output files will be `yspec.out.1`, `yspec.out.2`, ..., `yspec.out.n`, where n denotes the number of receivers, and the ordering of these files matches the input order of the receivers (see below).

2. `earth model` – name of the file listing the earth model to be used in the calculations. The input model must be in the 'deck format' used in the program minos. The model can have a fluid inner core, and the program takes this into account without any explicit input.

3. `attenuation switch` – set to one to include attenuation in the calculations, and to zero if attenuation should not be included. Note that the form of attenuation is the logarithmic dispersion relation used in the parameterization of the PREM model. Furthermore, the code implements this attenuation exactly, and does not depend upon first-order perturbation theory (as is, usually, done in the case of normal mode summation).

4. `gravitation` – The code can incorporate the effects of gravitation in a number of ways. Set parameter to zero to ignore gravitation completely, equal to one to employ the Cowling approximation, and to two to enable full self-gravitation. Using either no gravitation or the Cowling approximation will be a little faster, but not substantially so.

5. `output` – Determines the form of the output seismograms. Set to zero to get displacements, equal to one for velocities, and equal to two for accelerations. In all cases the outputs are in SI units.

6. `potential and tilt corrections` – set equal to one to make potential and tilt corrections to the sphierioidal solutions, and zero otherwise. Note, if the previous option must be set equal to two (i.e. to output accelerations) for these corrections to be applied.

7. `lmin` – minimum value of spherical harmonic degree l used in the calculations – probably should always be set equal to one.

8. `lmax` – Maximum spherical harmonic degree for the calculations. The appropriate choice of this parameter depends on the maximum frequency and also source-receiver geometry. If `lmax` is too small the seismograms will be ringy prior to the arival of the direct p-wave – if in doubt make this parameter bigger and see if anything changes.

9. `fmin` – minimum frequency for the calculations in mHz. Must be greater than zero – the equations are singular in the zero-frequency limit, and this code cannot calculate static displacements.

10. `fmax` – maximum frequency for teh calculations in mHz.

11. `length of time series` – length of the desired time series in minutes as measured from the origin time of the event.

12. `time step` – the constant time-step in seconds desired for the output seismograms.

13. `f11, f12, f21, f22` – parameters for cosine high and low pass filters applied prior to calculation of the inverse Fourier transform. A high-pass filter is applied between f_{11} and f_{12}, and a low-pass filter between f_{21} and f_{22}. **If all these parameters are set to zero, no filtering will be done**. Note, however, that some form of low-pass filtering must be done to remover the effects of the sharp cut-off of the calculations at the frequency f_{max}.

14. `source depth, latitude and longitude` – location of point source (depth in km, angles in degrees).

15. $M_{rr}, \ldots, M_{pp}$ – moment tensor components $M_{rr}, \ldots, M_{\phi\phi}$ in Nm refered to the standard spherical polar co-ordinate system (r, θ, ϕ) defined relative to the pole of the model. So, locally the r direction is vertically upwards, the θ direction is south, and the ϕ direction is east.

16. `receiver depth` – depth of **all** the different receivers in km. Note that the receiver can be placed below the source.

17. `number of receivers` – the total number of different receiver positions. Can be as large as you like, but if too big the program can run out of memory and will crash. Computations for multiple receivers involves only slightly more work than those for one receiver, so having lots of different receivers is fine.

18. `receiver latitudes and longitudes` – list of the different receiver latitudes and longitudes in the form

    ```
    lat1 lon1
    lat2 lon2
    lat3 lon3
    ```

    with all angles being in degrees. The ordering of the output files corresponds to that of this list.

Outputs

For each receiver is output an ascii file containing the three desired three component seismogram. The format is:

```
time1   vertical-component1   north-component1  east-component1
time2   vertical-component2   north-component2  east-component2
time3   vertical-component3   north-component3  east-component3
```

where all outputs are in SI units, and the form of the seismogram (displacement, velocity, or acceleration) is determined by parameter 5 described above.

Running the code as a batch job

For high frequency calculations `yspec` can be quite slow. If you have access to a cluster computer, things can be greatly speeded up by running the code in parallel as a ‘batch job’ – I’m sure cleverer things could be done, but I don’t really understant parallel computing!

To run the code as a batch job you need to run three programs:

```
yspec_pre
```

```
yspec_cal
```

```
yspec_pro
```

The first of these programs reads in the usual parameter file, and works out how to distribute the calculations between the available processors (it is, itself, run in serial). The second program is run in parallel on the different processors, and does the main calculations. Finally, the third program (which runs in serial) puts together the results and output the desired seismograms. To use these programs you can use the script `run_yspec_batch` which is as follows:

```
yspec_pre $1 $2

qsub -cwd -sync y -t 1-$2 run_yspec_cal

yspec_pro $2
```

The first argument to the script is the name of the yspec parameter file, and the second argument is the number of processors you want to split the calculations between. The script `run_yspec_cal` is simply

```
../binc/yspec_cal $SGE_TASK_ID
```

which runs the program `yspec_cal` for each of the jobs identified by the variable `SGE_TASK_ID` which is set by the program `qsub` – I imagine the details of this will likely vary between clusters. As an example, at the command line you might type:

```
>> run_yspec_batch yspec.in 100
```

which would produce equivalent results to typing

```
>> yspec yspec.in
```

but would split the calculations between 100 processors, and so would be significantly faster.