Normal modes package: modeslib

Smolenice spice meeting, September 2005

Yann Capdeville (capdevil@ipgp.jussieu.fr)

The aim of this tiny package is to provide programs that:

- compute normal modes in spherically symmetric models
- compute synthetic seismograms using normal mode summation.

It provides the following programs:

- `minosy` to compute normal modes
- `stock_boule` to compute normal modes in homogeneous sphere only
- `get_fctp` to extract normal modes
- `generate_prem` to generate Earth models (to be used by `minosy`)
- `nms` to compute synthetic seismograms with normal mode summation.

About `minosy`: it comes from MINOS written by F. Gilbert, J. hoodhouse and G. Master (and others???) with only minor modifications.

Compiling the package

- run `configure compiler` with `compiler` set to
  - `intel` for intel compiler (ifort)
  - `g95` for GNU fortran 95 (warning)
  - `pgf` for portland compiler
  - `xlf90` for IBM compiler (e.g. for mac?)
  - `dec` for DEC ALPHA compiler
  - `sun` for SUN
- run `make` or `gmake`

MINOSY

`minosy` computes normal modes:

- eigenfrequencies: $\omega_{q,n,\ell}$,:
- eigen functions: ${}_nU_\ell(r)$, ${}_nV_\ell(r)$ and ${}_nW_\ell(r)$,

with

$$\mathbf{u}(\mathbf{r}) = \left[{}_nU_\ell(r)\mathbf{e}_r + {}_nV_\ell(r)\boldsymbol{\nabla}_1 - {}_nW_\ell(r)(\mathbf{e}_r \times \boldsymbol{\nabla}_1)\right] Y_\ell^m(\theta,\phi)$$

Input file: `yannos.dat`

Output files:

- for ipgp format: 4 files for eigenfunctions (2 binary files (`*.direct`, 1 speroidal, 1 toroidal) and 2 ascii headers (`*.info`))
- two ascii information files (eigenfrequency, group velocity, attenuation ...)

Input file for minosy: yannos.dat

```
 1 #input earth model file (e.g. created with generate_prem) :
 2 prem
 3 #output information
 4 per_premR
 5 #prefix for output eigunfunction files
 6 fct_premR
 7 #type code (3=spheroidal, 2=toroidal, 0=both)
 8 3
 9 #precision (2) and switch off gravity perturbation frequency (eps1,eps2,wgrav)
10 1.E-10 1.E-10 10.
11 #lmin, lmax, fmin, fmax, nmax:
12 0 1500 0.1 10. 3000
13 #############if you need to output only some radius layers of eigenf
14 #number of layer for output:
15 1
16 #radius layers
17 3480000. 6371000.
18 #### computations flags.
19 # force fmin (usually F):
20 F
21 # Cancel Gravity (usually F)
22 F
23 # never use start level (usually F):
24 F
```

```
25 #use T ref (usually T):
26 T
27 #Check modes (usually F):
28 F
29 # use_remedy  awkward modes (usually T):
30 T
31 # rescue  (try to do something if missing a mode ) (usually T):
32 T
33 #restart (to restart in case of crash during computation
34 F
35 #force_systemic_search
36 F
37 # keep_bad_modes
38 F
39 #  modout_format (ipg, ucb, olm) (ipg is the standard for me)
40 ipg
41 # seuil_ray
42 0.0001
43 # l_startlevel
44 0
```

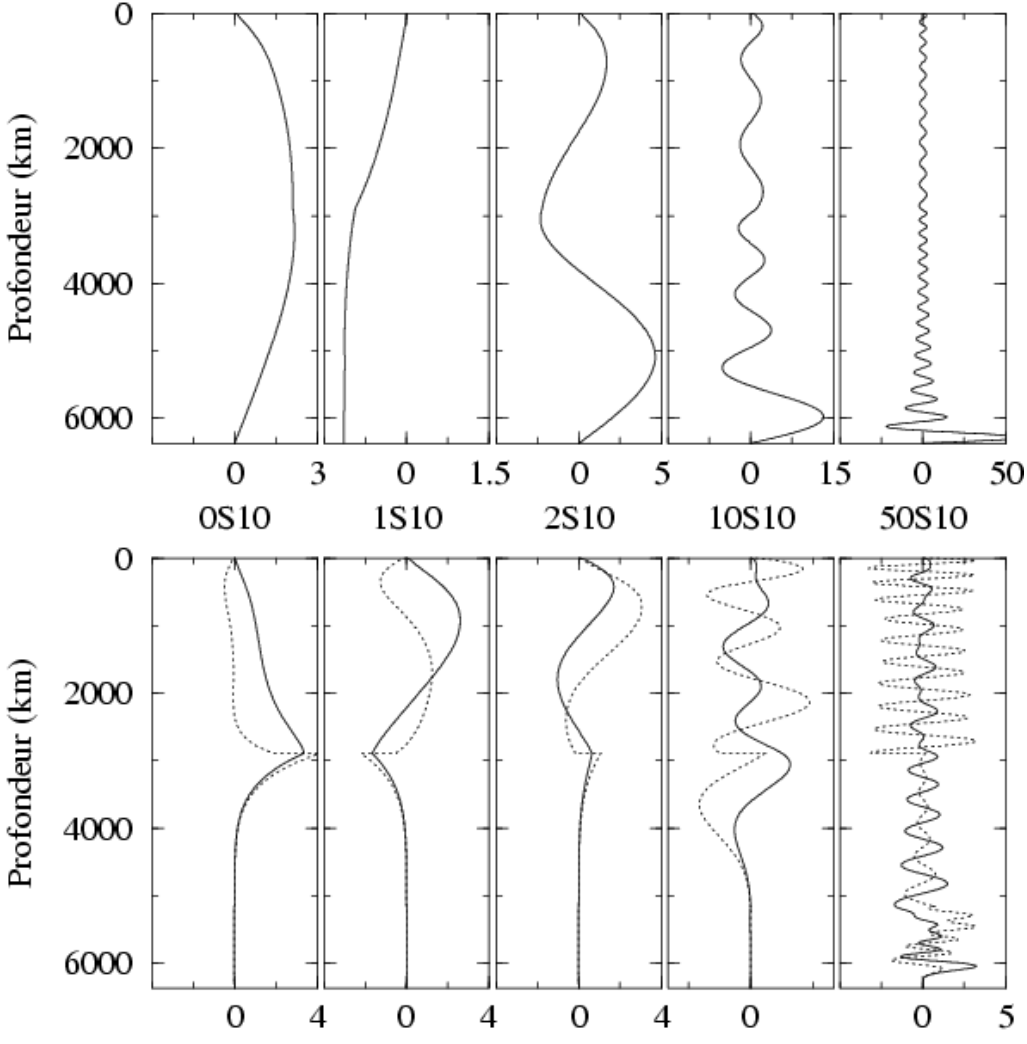
To generate the prem model, use `generate_prem`

An example of information output file (per_prem)

INTEGRATION PRECISION = 0.1000E-08 ROOT PRECISION = 0.1000E-08 GRAVITY CUT OFF = 0.000 RAD/S

	MODE		W(RAD/S)	W(MHZ)	T(SECS)	GRP VEL(KM/S)	Q	RAYLQUO	
0	S	0	0.5543943E-02	0.8823460	1133.342	0.000000	8912.249	0.1810270E-09	F
1	S	0	0.1062308E-01	1.690715	591.4658	0.000000	2222.735	-0.1849372E-08	F
2	S	0	0.1599140E-01	2.545111	392.9102	0.000000	1782.076	-0.3089906E-08	F
3	S	0	0.2074181E-01	3.301162	302.9236	0.000000	1532.520	-0.5615794E-08	F
4	S	0	0.2595226E-01	4.130430	242.1055	0.000000	1260.282	-0.7692218E-08	F
5	S	0	0.3082609E-01	4.906125	203.8269	0.000000	1134.519	-0.8875676E-08	F
6	S	0	0.3619638E-01	5.760832	173.5860	0.000000	1098.139	-0.1352340E-07	F
7	S	0	0.4146911E-01	6.600014	151.5148	0.000000	1048.585	-0.1683316E-07	F
8	S	0	0.4677034E-01	7.443731	134.3412	0.000000	1030.859	-0.2428890E-07	F
9	S	0	0.5204073E-01	8.282539	120.7359	0.000000	1012.281	-0.3164870E-07	F
10	S	0	0.5703162E-01	9.076864	110.1702	0.000000	986.2369	-0.3709945E-07	F
11	S	0	0.6226241E-01	9.909370	100.9146	0.000000	968.7993	-0.4044274E-07	F
.									
.									
.									
33	S	4	0.8091155E-01	12.87747	77.65498	19.25608	104.0263	-0.7658588E-08	F
34	S	4	0.8094885E-01	12.88341	77.61920	7.210750	159.5799	-0.4402698E-08	F
35	S	4	0.8251678E-01	13.13295	76.14434	15.67589	794.1248	0.4456421E-10	F
36	S	4	0.8752661E-01	13.93029	71.78600	2.760589	267.0168	-0.1949399E-08	F
37	S	4	0.8796525E-01	14.00010	71.42804	13.03259	653.1243	-0.3356146E-09	F
38	S	4	0.9020513E-01	14.35659	69.65442	26.84460	85.15483	-0.2028846E-07	F
39	S	4	0.9310037E-01	14.81738	67.48830	14.44108	885.5094	-0.1058792E-10	F
0	S	5	0.5332214E-02	0.8486483	1178.344	7.756168	389.1457	-0.2778223E-09	F
.									
.									
.									

To extract eigunfunctions: use `get_fctp`.


Profondeur (km)
0
2000
4000
6000
0
3
0
1.5
0
5
0
15
0
50
0S10
1S10
2S10
10S10
50S10
Profondeur (km)
0
2000
4000
6000
0
4
0
4
0
4
0
0
5

Normal Mode Summation: nms

Input files: nms.dat, sources.dat and receivers.dat

- nms.dat

```
 1 #time step
 2 5
 3 #number of time steps
 4 2400
 5 #source t0:
 6 1000.0
 7 #source amplitude
 8 1.e00
 9 #source frequency band (f1,f2,f3,f4):
10 0.10E-2
11 1.00E-2
12 1.20E-2
13 1.50E-2
14 #index min, index max of overtones to be used (-1,-1 for all)
15 -1 -1
16 #geocentric correction?
17 T
18 #component rotation
19 T
20 #S eigenfunction file prefix
21 fct_premR
22 #T eigenfunction file prefix
23 fct_premT
```

- sources.dat

```
1 #number of events
2 1
3 # stacking them (.true.) or not (.false.):
4 .false.
5 #n name      Mrr,     Mtt,       Mpp,     Mrt,      Mrp,     Mtp,      dep,   lat,   phi   tdelay
6 01 M012601A  1.04e+19 -0.43e+19 -0.61e+19 2.98e+19 -2.40e+19 0.43e+19  10.00  03.30 095.94 000.00
```

- `receivers.dat`

```
 1 #number of receivers
 2 11
 3 #name (4 characters), latitude, longitude in degrees
 4 109C   32.889 -117.105
 5 _AAK   42.639   74.494
 6 ACSO   40.232  -82.982
 7 AHID   42.765 -111.100
 8 _AML   42.131   73.694
 9 ANMO   34.946 -106.457
10 ANTO   39.869   32.794
11 _APE   37.069   25.531
12 _AQU   42.354   13.405
13 _ARU   56.430   58.562
14 _ARV   35.127 -118.830
```

Output: there 3 ascii files per receiver, e.g.

- `UZ_ANMO`, `UR_ANMO`, `UT_ANMO` (for vertical, radial and transverse components) for station ANMO if `rotation` is used
- `UZ_ANMO`, `UN_ANMO`, `UE_ANMO` (for vertical, north and est components) if not.

Practical

- Step 1: compile the package;
- Step 2: create a working directory somewhere and copy the `*.dat` files from the `examples` directory in it;
- Step 2: generate a `prem` model file with `generate_prem`. A number of layer of 500 is good enough here.
- Step 3: compute the normal mode catalogs with `minosy`;
- Step 4: take a look at some of the eigenfunction with `get_fctp`;
- Step 5: compute the synthetics with `nms`;
- Step 6: Have a drink ... well, no! wait for the next practical (phase velocity measurement) first ... sorry for that.