

Author: Yann Capdeville
email: capdevil@ipgp.jussieu.fr
Version 0.0, Paris, August 2005.

1 Prelude and warning

This is an attempt to provide a (poor) documentation for different programs to compute normal modes and synthetic seismograms with normal mode summation in spherically symmetric Earth models. The present version (0.0) of these programs may not have been used extensively. For example the normal mode summation code (nms) is not the one I use usually to test and validate Spectral Element codes in the Earth, but have been specially developed for the occasion of the spice meeting from a time reversal code I wrote this year. This is to say that a lot of errors and bugs may still be present in the codes. Be warned if you used these codes for your own work (but if you do, a little thank you in the acknowledgment of your corresponding paper would be appreciated) .

1.1 Documentations? About what?

- `minosy` (alias `yannos`): this program computes normal modes (eigenfunction and eigenfrequencies) for given spherically symmetric Earth model for a given frequency band. An eigenfunction for given eigenfrequency ${}_n\omega_\ell$ is written as

$$\mathbf{u}(\mathbf{r}) = [{}_nU_\ell(r)\mathbf{e}_r + {}_nV_\ell(r)\nabla_1 - {}_nW_\ell(r)(\mathbf{e}_r \times \nabla_1)] Y_\ell^m(\theta, \phi)$$

where n is the radial order, ℓ the angular order and m the azimuthal order. Note that, because of the spherical symmetry, eigenfunctions and eigenfrequencies do not depend on m . $\nabla_1 f(\mathbf{r}) = \partial_\theta f(\mathbf{r}) \mathbf{e}_\theta + (\sin \theta)^{-1} \partial_\phi f(\mathbf{r}) \mathbf{e}_\phi$. Y_ℓ^m is the spherical harmonic. The mode normalization is such that

$$\int_0^{r_\Omega} \rho(r) ({}_nU_\ell^2(r) + \ell(\ell+1){}_nV_\ell^2(r)) r^2 dr = \int_0^{r_\Omega} \rho(r) \ell(\ell+1){}_nW_\ell^2(r) r^2 dr = 1 ,$$

where r_Ω is the Earth radius. `minosy` computes ${}_n\omega_\ell$, ${}_nU_\ell(r)$, ${}_nV_\ell(r)$ and ${}_nW_\ell(r)$ for a given Earth model.

- `stock_boule`: Same as `minosy`, but for homogeneous sphere only. The eigenfunctions are computed analytical (with Bessel functions) which can be sometime interesting for accuracy reasons.
- `nms`: this program compute synthetic seismograms for a given list of receivers and sources using the normal mode catalog produced by `minosy`.
- `get_fctf`: little program to extract a given normal mode from the output of `minosy`. It can be useful to visualize the eigenfunctions.
- `generate_prem`: little program to generate different kind of Earth models to be used by `minosy`.

1.2 References

For normal mode computations, it can be useful to read Takeuchi & Saito (1972), Saito (1988), Woodhouse (1988) and for normal mode summation, e.g. Woodhouse & Girnius (1982). An good review of the subject can be found in Dahlen & Tromp (1998).

- Dahlen, F. A. & Tromp, J., 1998. *Theoretical Global Seismology*. Princeton University Press. NJ.
- Saito, M., 1988. DISPER80: A subroutine package for the calculation of seismic normal-mode solutions. In D. J. Doornbos (Ed.), *Seismological algorithms*, pp. 294–319. Academic Press, New York.
- Takeuchi, H. & Saito, M., 1972. Seismic surface waves. *Methods in computational Physics.*, **11**, 217–295.
- Woodhouse, J. H., 1988. The calculation of eigenfrequencies and eigenfunctions of the free oscillations of the earth and the sun. In D. J. Doornbos (Ed.), *Seismological algorithms*, pp. 321–370. Academic Press, New York.
- Woodhouse, J. H. & Girnius, T. P., 1982. Surface waves and free oscillations in a regionalized earth model. *Geophys. J. R. Astron. Soc.*, **78**, 641–660.

2 Compiling the package

You will need a Fortran 90 compiler to compile the package. You need to declare `F90COMPILERNAME` environment variable to

- `intel` for intel compiler (ifort)
- `pgf` for portland compiler
- `xlf90` for IBM compiler (e.g. for mac?)
- `dec` for DEC ALPHA compiler
- `sun` for SUN

and then run the `configure` script which will generate the `flags.mk` to be used by the different makefile. If you need another compiler, you can easily add an new entry in the `configure` script. Run `make` or `gmake` at the top of the package tree and you should get all the executable files in `bin/` directory. If you don't ... well you will have to work a little bit more than expected or send me an email (try the first option first!).

3 Programs use

An example of each input file (“`.dat`”) is given in the `examples` directory.

3.1 generate_prem

`generate_prem` produces some different Earth model ascii files to be used as an input by `minosy`. The output of format `generate_prem` (and input of `minosy`) is the following:

- line 1: name of the model
- line 2: `ifani`, reference frequency for attenuation dispersion, `ifdeck`. It is a good idea to leave `ifani` to one. Only `ifdeck=1` will be consider in the document.

- line 3: total number of model lines, index of the ICB, index of the OCB, number of line for the ocean (can be set to 0).
- line 4 and more: radius, density, V_{pv} , V_{sv} , Q_{kappa} , Q_{shear} , V_{ph} , V_{sh} , η . The format (in fortran) must be: `f8.0, 3f9.2, 2f9.1, 2f9.2, f9.5`

3.2 MINOS

3.2.1 History

`minosy` (I used to call it `yannos`; I know, this is a stupid name that is why I changed that for the spice meeting) is my version of the MINOS program. MINOS is a program with long history which I know only partially. As I understand, at least 3 peoples contribute to its development more than twenty years ago: F. Gilbert, J. Whoodhouse and G. Master, but it is unclear to me who exactly did what. It seems that the first version of the program was `eos` and then MINOS. Another normal modes program with a lot of common thinks with MINOS from J. Woodhouse is OBANI. In 1997 I started working on the coupling of spectral elements with modal solution. To compute the modal solution, I started from MINOS. Because most of MINOS was written with an old Fortran standard (a lot of “goto” and indexed “if”) I had to translate many parts of the program to more standard Fortran because I wasn’t able to work efficiently with the older standard. Finally, after finishing working with the coupling of spectral elements with modal solution, I also started to use my modified MINOS code to compute classical normal mode solution and it appears that this version was more stable for high frequency. Compared to the original version, I didn’t introduce anything new, it is even less efficient than the original version (not for the MPI case). The original version had some little bugs that make it difficult to use to get the full wavefield at relatively high frequency. If you look into `minosy`, you’ll see that it is messy. That is true, but it is a mess that I’m able to understand which wasn’t the case for MINOS. Finally, you should always be careful when using this program as I may have introduced bugs that weren’t in older version (it has been the case many times in the past).

3.2.2 Running the code

There is only one input file, `yannos.dat` that must be in the running directory. Here is an example (where line number have been added and should be removed for an actual run):

```

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)
8 3
9 #preccision (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 4 0.1 4. 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

```

- line 2: input model (e.g. output from `generate_prem`).
- line 4: normal mode catalog information output in ACSII. It will give the eigenfrequency, group velocity, attenuation for each mode. The last two column are the Rayleigh Quotient which give an indication of the accuracy (the smallest is the best) and a logical that is .true. if the mode is an inner core mode (not always accurate). (Rayleigh Quotient: if \mathbf{H} is the elastic operator of the wave equation and if \mathbf{u}_k the eigenmode associated with the eigenfrequency ω_k , therefore $(\mathbf{u}_k, \mathbf{H}\mathbf{u}_k) = \omega_k^2 ((\cdot, \cdot) \text{ is the inner product})$. In this file, the Rayleigh Quotient is $1 - (\mathbf{u}_k, \mathbf{H}\mathbf{u}_k)/\omega_k^2$ and should be equal to a small value if \mathbf{u}_k is indeed a mode associated to the frequency ω_k).
- line 6: prefix of the eigenfunction output. If the “ipg” format is used, on binary file (`.direct`) and one ascii file (`.info`) will be created from this prefix.
- line 10: the two first numbers are respectively integration precision and precision on the eigenfrequency search. The last number is the frequency (in mHz) after which the gravity potential perturbation is not computed anymore. If you set this last value to 0, you will be in the Cowling approximation.
- line 12: the normal modes will be computed between the angular order `lmin` to `lmax` and the frequency `fmin` to `fmax` (in mHz). `nmax` is the maximum number of overtones that will be computed for each angular order. If you need the whole wavefield solution for a given

frequency band, set `lmin` to 0 and `lmax` and `nmax` to large value (e.g. 5000). If you need only the fundamental branch, set `nmax` to 1.

- line 14 to 17: allow to choose one or more layer in depth for which to store the eigenfunctions. Usually one layer for 700km depth up to surface is enough.
- `force_fmin` (line 20): In normal use, the start frequency search increase with the angular degree `l`. But for some situation, it came make sense to deactivate this feature by saying `"true"`. Usual answer is `"false"`
- `cancel_gravity` (line 22): `"true"` means that ALL gravity terms will be set to 0.0 Usual value is `".false."` Warning: if that option is set, the starting frequency search shouldn't be too low because it makes failed the alternate the Bessel subroutine I'm using is that case. The consequence of this is that the 0S1 and 1S1 may not be found. (for prem, $f_{min} < 0.05 \text{mHz}$ is not good)
- `never_use_startlevel` (line 24): `startlevel` is subroutine that determine the integration starting point in depth. Usual value is `".false."` If you have problems with some strange model (e.g. full homogeneous models) it may be a good idea to set this to `".true."`
- `use_tref` (line 26) : Use the reference period of the model (for example: prem 1 s) Usual value is `".true."`
- `check_modes` (line 28) : try to cancel some bad modes (some Stonley and inner core modes) Don't use it unless you know what you are doing. Usual answer is `".false."`
- `use_remedy` (line 30) : `remedy` was build to recompute some modes like inner core modes. The original minos has it, but some problems come from this feature. Usual, I don't use it, but up to you. No Usual value.
- `rescue` (line 32) : Try to do something if the bracketing of modes fails. Use a systematic search for some frequency. It's not very efficient, and it can't heart anyway. Usual value is `".true."`
- `modout_format` (line 34):
 - `ucb`: UC Berkeley
 - `olm`: old minos
 - `ipg`: my format

Only `"ipg"` is implemented in the MPI version, but easy to do yourself. The `nms` program uses the `ipg` format.

- `l_startlevel` (line 44): If you use `"startlevel"`, specify after which `"l"` it should do so. 0 is the standard value.
- `seuil_ray` (line 38) : cancel the mode if it's Rayleigh quotient is greater than this value. 10^{-4} can be a good choice, but it really depends of what you are doing. It's really up to you. Information about the discarded modes will be found in the `.lost` file.
- `restart`: allow to restart after a crash and avoids to restart from start each time. For debug only.
- `force_systemic_search`: an alternate frequency search: don't use it.

3.3 yannos_MPI version

For large frequency band, the computation can take time. In that case and if you have access to a parallel computer, the MPI version of minosy, yannos_MPI can be useful. To compile this version, you will need to set the MF90 variable in the flag.mk to the proper mpi f90 script (e.g. mpif90 for mpich) and execute `make yannos_MPI` in the `yannos/src` directory.

3.4 stock_boule

`stock_boule.dat` does the same job has minosy but for homogeneous sphere only using Bessel functions to compute the solutions (see Takeuchi & Saito 1972). Is a single input file: `stock_boule.dat`:

```
1          Lmax= 0500
2 number of layers  = 0600
3 maximum of overtone= 0401
4      rho (kg m^-3) = 03000.0
5          vp (m/s) = 08000.0
6          vs (m/s) = 06000.0
7 rayon dela boule(m)= 6.371E+06
8  frequency minimum = 1.000E-08
9  frequency maximum = 0.300E-02
10 prefix file Rayleig= fctS
11 prefix file Love    = fctT
```

where the format must be respected (the first 15 character of each lines won't be used as an input).

- line 1 is the maximum angular number.
- line 2 is the number of radial sample that will be used to output the eigenfunctions.
- maximum number of overtones to be computed
- line 4 to 6: density, vp and vs
- line 8 and 9 frequency band in Hz
- line 10 and 11: prefix for spheroidal and toroidal eigenfunction files.

3.5 To read a single mode to ASCII format: get_fctp

`get_fctp` allows you to extract a particular eigenmode form the binary output of minosy. The output file form `get_fctp` is a two columns (depth , amplitude) ASCII file.

3.6 Normal mode summation program: NMS

There are 3 input files for nms: `nms.dat` `receivers.dat` `sources.dat`
`nms.dat`:

```
1 #time step
2 5
3 #number of time steps
```

```

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

```

with

- line 2: time step in second
- line 4: number of time step
- line 6: central time of the source wavelet in second
- line 8 to 11: definition of the source wavelet in the frequency domain. The spectrum is flat between f_2 and f_3 and a cosine taper between f_1 and f_2 and between f_3 and f_4 .
- line 13: overtone selection. e.g. 0 0 will select only the fundamental mode. (-1 -1 will take all the overtone)
- line 15: do we apply geocentric correction to source and receiver coordinate (see Dahlen and Tromp, p)
- line 17: do we apply component rotation.
- line 19 and 21: eigenmode catalog prefix output of minosy for spheroidal and toroidal modes respectively.

`sources.dat` contains all the informations about the source(s):

```

1 #nombre d'evenements
2 1
3 # stacking them (.true.) or not (.false.):
4 .false.
5 #n name      Mrr,      Mtt,      Mpp,      Mrt,      Mrp,      Mtp,
dep,  lat,  phi
6 01 M012601A  1.04e+19 -0.43e+19 -0.61e+19  2.98e+19 -2.40e+19
      0.43e+19  10.00  03.30 095.94 100.00

```

Note line 5 and 6 have been cut to fit in the page.

- line 2: number of events to be computed
- line 4: in the case of several sources, should the be stacked in a single trace per receiver and component?
- line 6: number, name, moment tensor, depth and location of the event.

`receivers.dat` contains all the informations about the receivers:

```

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

```

- : line2 : number of receivers
- line 4 to number of receivers +3: name (4 characters), latitude, longitude in degrees of each receiver.

The outputs are 3 ASCII files for each couple source–receiver. The two first characters of each file will be UZ, UR and UT (for vertical, radial and transverse components) if the rotation is used and UZ, UN and UE (for vertical, North and Est components) if not.