UNIVERSITÀ
DEGLI STUDI DI TRIESTE

dipartimento
di matematica
e geoscienze

The DMG Manuals

Pulsyn06

Program for wide-band simulation of a source pulse radiated by a finite earthquake source/fault.

Document Name: DSTXPulsyn.pdf
Document Version: 1.2
Date Issued: 26 September 2017 13:59
Author: Andrea Magrin, Franco Vaccari

University of Trieste
Department of Mathematic and Geosciences
Group of Seismology
via Weiss, 4
34128 Trieste, Italy

Table of Contents

Wide-band simulation of a source pulse radiated by a finite earthquake source/fault

The present manual describes the use of PULSYN program package, version 2008 and should be read with the “Program description” document. The code of PULSYN main module was originally written by Dr. A.A. Gusev in the Matlab programming language and then rewritten in Fortran90 by him, with minor modifications. The following instruction is for the Fortran90 version. This code consists of the driver program PULSYN06 and of a work subroutine PULSYNF06 that performs subsources creation. A number of lower-level modules used by PULSYNF06 are realized as additional subroutines and functions. The compiled executable module is pulsyn06.out.

Required input files

The control and data input to PULSYN06 are performed via disk files only. One example input dataset (**/XDST/Examples/PulsynExamples/Pulsyn/Base** directory) is given. You should copy the required files into a new directory dedicated to your own computations

One file with the fixed name pul.par must be always present in the same directory as pulsyn06.out. This “parameter” file is used essentially instead of the usual “command line” arguments; through it, input file names are passed to PULSYN06. Typically, two input data files are read: source description file (*.src) and site/receiver description file (*.ste). Some other files can be needed to support running PULSYN06.

.src file

To create a single copy/sample of source, PULSYN06 reads a file of parameters from disk, its name follows the pattern XXX999.src Here “XXX999” is the generic pattern, that assumes that XXX is the batch code and 999 is the variant code; however, these six characters can be any alphanumeric ones. A *.src file consists of several data blocks that describe “external” source parameters (EXT block), “internal” source parameters (INT block) and some other blocks. The INT block of parameters is identical to the input parameters of the subroutine PULSYNF06. Parameter blocks will be explained separately.

.ste file

In addition to generation of a source description (via subsources), PULSYN06 also generates the set of subsource-receiver pairs. To do this it reads a file of receiver/station/site parameters. The generic name pattern for this file is is NNN888. Here NNN may denote a site set or a network, and 888 is its particular variant; again, these six characters can be any alphanumeric ones.. A *.ste file consists of two blocks. First block defines a set of numbers for flat-layer structures to be associated with stations. The second block lists station coordinates, names, structure codes and other data. The detailed description of *.ste file is given separately.

Additional input files

Additional information can be needed, and it is read from other files. Names o these files are given in the special block FIL of *.src file. In particular, to work with “Gusev1983” spectral scaling model, PULSYN06 loads the file with corresponding spectral table. Also, other spectral models can be specified by corresponding files. Also, external 2D final slip distribution over a fault can be loaded from file. These files must be present in the current directory. See more detailed information with FIL block description. Also, in a regular calculation mode, PULSYN06 may need the auxillary file pulfav created by PULSYN06 itself in a preliminary run.

Output files

The results of PULSYN06 are saved in the following way.

1. A number of intermediate and output scalar parameters of simulation (to be explained separately) are appended to the initial version of XXX999.src file. If such a data block has already been appended to such a file, the earlier version of this block shall be lost.
2. Matrix results describing the simulated source are saved in two files:
   a) subsource positions (with respect to both local/intrinsic and world reference), subsource seismic moments and time delays; these are put to the file XXX999.xta.
   b) subsource time histories; either in time domain (file XXX999.sbt) or in frequency domain (file XXX999.sbf). These are given in the units of seismic moment rate or its Fourier spectrum.
   c) (optionally) far-field time histories or “source time functions” (STF) representing displacement for a far-field receiver; either in time domain (file XXX999.frt) or in frequency domain (file XXX999.frff). These are formatted like sbt/sbf and given in the units of seismic moment rate [dyne cm/s] or its Fourier spectrum.
3. Also, three output files are generated that contain the results of PULSYN06 that are intended to be used as input parameter files for Green’s function calculator.
   a) XXX999.sut, (subsource data, informally same as in *.xta)
   b) XXX999.obs, (site data, informally same as in *.ste)
   c) XXX999.isg (source-receiver pair data).

   These three files are made compatible with the programs sgr/sgl/sgv and others of the DETHAZ package of DST, University of Trieste.

The file XXX999.xta. contains a few columns, of length nsub=nx ny , that contain coordinates, onset times and amplitudes (seismic moments) of subsources; details to be described later

The files XXX999.sbf/sbt has several variants of structure, to be described later. The basic variants are:

*.sbt: nsub=nx ny columns that contain complete time functions of subsources (real)

*.sbf: table of the same structure that contains spectra of these functions (complex)

For cases when the point source description is sufficient, (typically in the case of a distant receiver) a source time function usable for such receiver generated. Generally, there are two possibilities:

1. To ignore directivity effects related to asymmetric fault rupturing etc. All subsources are stacked in a single time function, to be used for any receiver, independently from ray direction or azimuth.
2. To incorporate directivity effects. Subsource signals are combined using appropriate delays, taking into account ray direction and assumed propagation velocity (for body waves, body wave velocity); results are put to frt/frf files. This approach is more realistic, but still of limited accuracy, in particular because different parts and/or modes and/or harmonics of radiated signals propagate to receiver at different velocities.

When the aim of running PULSYN06 is the creation of a source model only (e.g. in order to generate pulfav file), one can bypass further calculations. This can be done by switching off the calculation of source-receiver pair data. To achieve this, set the name of *.ste file in pul.par file (2nd line) as “#”.

Summary of input and output files

The following list is an example list of files that must be present in the work directory to run PULSYN06:

```
pul.par
GUSEV83.TB5
mod001.src (generally, *.src)
str003.ste (generally, *.ste)
```

After the run of PULSYN06, in this case, a modified version of mod001.src must be seen (the OIS block must appear or be refreshed), and the following additional files must appear (the particular set of *.isg files is an example only):

```
mod001.obs
mod001.sbf
mod001.sut
mod001.xta
mod0010008.isg
mod0010020.isg
```

Input files

pul.par

Input parameter file names are passed to pulsyn06 code through the initial “parameter” file that essentially performs the function of the set of command line arguments. It consists of three lines; it has a standard name pul.par and looks like

```
                        !! path  , now inactive
mod001.src              !! xxx999.src (source) file name
str003.ste              !! yyy888.ste (sites) file name
```

Only 20 characters of each line are read.

Line 1 is empty at present; it is intended to be used for the path info.

Line 2 contains the name of input parameter file *.src that define the synthesis of a particular source model. It must have any 6 alphanumeric characters, following the pattern AAAAAA.src. However, in practice, less general pattern XXX999.src is recommended (with batch-name XXX and variant-number 999. This pattern is not obligatory but convenient, and we shall assume it. The code XXX999 (or, alternatively, AAAAAA) is used systematically as a root to form file names of other files.

Line 3 contains the file name *.ste of site/station data set; it assumed to have 6 characters like yyy888.ste or BBBBBB.ste.

*.src - source description file

General

The input parameters for simulation are read from a disk file, whose example name is "tst001.src". This name is to be passed through the “parameter file” with the standard name pul.par. PULSYN06 should be run from the same directory as the one where “pul.par” file and “xxx999.src” file reside; results are written to the same directory. It is important to note that *.src file is modified by PULSYN06 and additional information is appended to it. Thus for the safe work, one should keep a copy of the initial state of *.src file to prevent problems related to disk or processor error.

The structure of the input file *.src is as follows. It consists of several data blocks each containing logically connected information. Each data block is surrounded by a pair of identical “bracket lines” of the common structure

```
!@@AAA@@
```

where AAA is the code of the block. The order of the blocks is obligatory; however, inactive blocks can be omitted. The following blocks are possible:

- EXT: contains “external” source parameters & control parameters; defines the positioning of the source in the “world” coordinates; obligatory

- INT: contains internal source parameters that control the synthesis; defines the source in its own coordinates; obligatory
- FAR: contains the request for far-field or ray dependent single-pulse calculation, optional
- FIL: contains information on additional files, optional
- OIS: contains output of PULSYN06. During the successful run of pulsyn06 , OIS block is being appended, that keeps output parameters of source synthesis. Initially, this block is absent; it is appended to the initial state of *.src file when is run. This block has the code OIS (means Output related to Intrinsic Source). Its earlier state shall be lost in each run of PULSYN06.

For clarity and convenience, one can freely use in *.src file empty lines and comment lines (must begin with “!”); one can also add comments after a parameter. Empty lines are recognized by leading space, so the first column of data line must always be filled by a parameter value. In EXT, INT and OIS blocks, all parameters are stored one per line, and occupy no more than 15 first positions of the line; other positions can be used for comments.

Reading data blocks from *.src file.

To facilitate reading data blocks, if needed, we give here corresponding Fortran statements

```
character lin*30;            !input line buffer
real*8 parex(30),parin(110); !arrays to store input parameters
. . . . . . . . . .
open (1, file=<xxx999.src>, status='old', err=1001);

! ========= reading EXT block =========
iparin=0;
! skip until EXT block:
do i=1,10000; read(1,'(30A)',end=921)lin; if(lin(2:6)=='@@EXT')exit; enddo
do i=1,10000
      read(1,'(30A)',end=922, err=952)lin;                    !next line
      if(lin(2:6)=='@@EXT')exit;                              ! ending block
      if(lin(1:1).ne.'!'.and.lin(1:1).ne.' ')then;           !good line
        iparin=iparin+1;     if(iparin>30)goto 902            !out if excesseve
        read(lin(1:15),'(g15.0)')parex(iparin);               !read and store

    endif;!
   enddo

   ! ========== reading INT block =========
   iparin=0;
   do i=1,10000; read(1,'(30A)',end=901)lin; if(lin(2:6)=='@@INT')exit; enddo !
skip
 do i=1,10000
      read(1,'(30A)',end=902, err=954)lin;                      !next line
      if(lin(2:6)=='@@INT')exit;                                ! ending block
      if(lin(1:1).ne.'!'.and.lin(1:1).ne.' ')then;             !good line
        iparin=iparin+1; if(iparin>110)goto 922                 !out if excesseve
        read(lin(1:15),'(g15.0)')parin(iparin);                 !read&store
    endif;! 1 to 110
   enddo
. . . . . . . . .
901 stop '@@INT block header not found'
902 stop '@@INT block bottom not found'
921 stop '@@EXT block header not found'
922 stop '@@EXT block bottom not found'
952 stop 'error reading EXT block parameters'
954 stop 'error reading INT block parameters'
```

“EXT” block of *.src file

This block contains 30 parameters in 30 lines. An example copy of EXT block follows. A detailed explanation is included within the block as comments. Note that all parameters have non-space in the first column.

```
!@@EXT@@ BLOCK 1 begins: EXTERNAL SOURCE PARAMETERS & CONTROL +++++++++++++++++++++++
+
!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!++ additionally, comments can be after col 16 when after parameter value
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!==========  group 1 general geometry: ALL ANGLES IN degrees with decimals =====
! geographic position of the origin of the intrinsic CARTESIAN COORDINATE SYSTEM
46.3              46.00          !(1)  latitude, for origin of "map" system,
13.0              12.5           !(2)  longitude, for origin of "map" system

! ---reference characteristic point RFC of the simulated fault rectangle is set
22             !(3)  cpcode  code of the characteristic point CP, written as two
digits “ab”
!         a=1/2/3 : CP lays on the linear segment that is:
!                a=1: rectangle's upper side (y=0);
!                a=2: rectangle's long axis of symmetry (y=W/2);
!                a=3: rectangle's lower side (Y=W);
!        b=1/2/3: CP is -x end (x=0)/center (x=L/2) /+x end (x=L) of the segment in
question
!        !! to use the center of the rectangle as the refernce point set cpcode=22
46.253         !(4)  latitude,  for RFC (center if cpcode==22
13.1388        !(5)  longitude, for RFC
5.0            !(6)  depth, km, for RFC

! -- fault plane orientation constraint: ALWAYS: 0<dip<90;
! -- strike direction defines +x axis orientation within fault plane
!            ("long axis", horizontal axis of symmetry)
! -- down-dip direction is +y direction on fault plane
! ---when fault plane is horizontal and strike is 0 deg(North),
!    +x looks North and +y looks East
! -- among two options, select strike direction (and thus +x_fp0) so that
!      looking along strike, +y ray direction moves down, clockwise when dip
!      changes from zero to positive
270.           !(7) strike, defines fault plane +x axis orientation in the World
20.            !(8) dip   defines fault plane +y axis orientation as DOWNDIP]
90.            !(9) rake: from +x counterclockwise (not like azimuth!)
-999           !(10) reflat(1) refrence map quadrangle on a map
-999            !(11) reflat(2)
-999           !(12) reflon(1)
-999           !(13) reflon(2)
-999           !(14)
! ========== group 2 control and settings
-999           !(15)
6              !(16) version no
-3             !(17) mode_output mode of output for subsource&farfield file;
!                    specifies time/ frequency domain and format
!               <0: T domain result, real values,  sum(all)*dt makes M0 [10^20 dyne
cm]
!                  -3, : single file nsub  values in a line, n lines
!               >0: F domain result; complex values (r,i); X(1) equals M0 [10^20 dyne
cm]
!                    1: single file n/2+1 value in a line, nsub lines (economic)
!                    2: nsub files , no more than 99
!                    3: single file nsub  values in a line, n lines (non-economic)
!                    4: single unformatted file like 1

1              !(18) mode_far_world: what reference to be used for ray specification:
!                    (0) fault plane [Az from X azis, alt], (1) world [Az from N,
incid angle
1              !(19) mode_ext_finslip: controls using preset 2D final slip function
!                    (0) regular case==simulated slip, (1) use preset final slip
0              !(20) mode_point: (0) regular case
!                    (1) sum all time-functions, put point source to the rect. center
0              !(21) mode_dethaz == nfqm  control DETHAZ mode
                         (0,-1) regular/ non-DETHAZ mode,
!                        (201) positive integer "DETAHAZ number": no of spectral
points
```

```
!                  the  next 3 parameters   used in DETHAZ mode only
1              !(22) fifr:    basic upper frequency (may set 10 or 1 hz)
1              !(23) npint:   npint+1   == (spectral interpolation multiplicity)
4096           !(24) npts:    no-samples == signal-length;
!!! when using DETHAZ parameters: T=(npint+1)*(nfqm-1)/fifr; dt=T/npts
-999           !(26)
-999           !(25)
-999           !(27)
-999           !(28)
-999           !(29)
-999           !(30)
!@@EXT@@ BLOCK 1 ends  : EXTERNAL SOURCE PARAMETERS  +++++++++++++++++++++++++
```

The EXT block contains input parameters, one parameter per line, free format with 1st column nonspace, to fill up to 15 first columns. For convenience only, each parameter line, in its later, comment part, has the parameter number in parentheses, from 1 to 30. Some positions are reserved and filled by -999. The meaning of parameters is as follows:

Group 1 general geometry

(1) latitude, and

(2) longitude, for the origin point of the "map" system. Select them arbitrarily within 50-100 km from the source hypocenter

Next 4 values define meaning and position of the “reference characteristic point” RFC of the simulated fault rectangle

(3) cpcode: code of the RFC, written as “ab” (like 12 or 21), where a=1/2/3: codes RFC position along fault width; sets RFC to lay on the linear segment that is:

a=1: rectangle's upper side;

a=2: rectangle's long axis of symmetry;

a=3: rectangle's lower side;

similarly, b=1/2/3: codes RFC position along fault length;

sets that RFC lays within the nearest-along-strike (x=–L/2) side of source quadrangle; or at the central shorter axis of symmetry of quadrangle; or within the farthest-along-strike (x=+L/2) side of the source quadrangle.

In particular, to set RFC at the geometric center of the rectangle, set cpcode=22

(4) latitude, and

(5) longitude, and

(6) depth, km, for RFC

“earthquake mechanism” parameters of the source-fault, in degrees:

(7) strike

(8) dip (in [0 90])

(9) rake

Group 2: control and settings

(16)version no   number of version, 06 for PULSYN06

(17)mode_output: defines mode of output for subsource and farfield files; its sign specifies time/frequency domain, and value the particular format:

mode_output <0: T domain result, real values, sum(all)*dt makes M0 [10^20 dyne cm] mode_output = -3, : single file nsub values in a line, n lines

mode_output >0: F domain result; complex values (r,i); X(1) equals M0 [10^20 dyne cm]

mode_output = 1: single file n/2+1 value in a line, nsub lines (economic)

mode_output = 2: nsub files , no more than 99

mode_output = 3: single file nsub values in a line, n lines (non-economic)

mode_output = 4: single unformatted file organized like case 1

(18)mode_far_world: specifies what reference will be used for ray specification for far-field calculation:

mode_far_world= 0 use fault plane [Az from X azis, alt],

mode_far_world= 1 use local world coordinates [Az from N, incid angle]

(19)mode_ext_finslip: controls using preset 2D final slip function: 0 regular case, use simulated slip, 1 use preset final slip

(20)mode_point: controls creation of an isotropic point source model: 0 regular case, nxsubnysub subsources; 1 sum all time-functions, put point source to the center of rectangle.

Parameters 21-24 are used to activate special simulation mode to match the results to the standards of the DETHAZ package

(21)mode_dethaz: (0) regular/ non-DETHAZ mode, (201) DETHAZ mode on; in DETHAZ mode, then the value of nfqm parameter (=201) is taken from mode_dethaz

the next 3 parameters are used in DETHAZ mode only, otherwise ignored:

(22)fifr: basic upper frequency (may be set as 10 Hz or 1 Hz)

(23)npint: (npint+1) = (spectral interpolation multiplicity)

(24)npts: number of samples = signal-size, same as n below;

Note: as the result of using DETHAZ parameters:

duration: T=(npint+1)*(nfqm-1)/fifr;

time step: dt=T/n;

“INT” block of *.src file

This block contains 110 parameters in 110 lines. An example copy of INT block follows:

```
!@@INT@@ BLOCK 2 begins: INTERNAL SOURCE PARAMETERS +++++++++++++++++++++++++++++++++++
+++++++
!++++ comments after leading ! or after col 15 when parameter value present+++++++++++
+++++++
!+++++lines with leading sace skipped ++++++++++++++++++++++++++++++++++++++++++++++++
+++++++
!=========== group 1 general info may be common to many sources
3.3            !(1)  cS , cS to scale rupture velocity
4.1            !(2)  CMS const to convert Mw to source area
-1             !(3)  cSS, cS to scale mu to scale source area
0.1            !(4)  C_heaton
-999           !(5)  ##inactive
```

```
1.5            !(6)  ARlow
4              !(7)  ARhi
5.5            !(8)  MARlow
8.             !(9)  MARhi
100.           !(10) Wbig
2              !(11) pos_acc: 1; (+)accel.pulses to positive time,
!                            2: (+)posit and neg times; 3: (+)and(-)tricky
1.2            !(12) CVAXY    CV - scatter in slip amplitudes over x,y;  default 0.5
1.5            !(13) spowxy exponent of the power-law spectrum over x,y;default 1.75
1.5            !(14) cappowxy exponent for the "cap"/2D taper over x,y; default 0.3
!                        cappowxy<=0 makes boxcar envelope along x and y
!  deterministic boxcar case: set CVAXY=0.05; spowxy=0; cappowxy=-1
!  deterministic cap case: set:   CVAXY=0.05; spowxy=0; cappowxy=1

1.5            !(15) CVAT CV-scatter in pulse/count amplitudes over t; default 0.5
0.5            !(16) spowt  exponent of the power-law spectrum over t; default 0.5
0.7            !(17) shappowt     exponent for the shape/taper fun. of t;
!                                 default 0.5 for exp, 1 for hyp, 0.7 for cap
2              !(18) typshapet    defines shape/taper fun:
!                                 (1)box/ (2)cap/ (3)exp/ (4)hyp
-999           !(19) ##inactive
1.0            !(20) ctwind    const defines local time history duration  default 2
0.             !(21) relscax   relative scatter of x(subsource) within a cell $$
0.             !(22) relscay   relative scatter of y(subsource) within a cell $$

0000           !(23) modefreeb code indicates where free boundary is positioned:$$
!                               0: none; 1000: -x, 0100: +x; 0010: -y; 0001: +y
!               +y looks downdip, so 0010 is the right mode for a surface fault
-999           !(24) ##inactive

! ============ group 2 specify a source
6.4            !(25) Mw     MOMENT MAGNITUDE
-1             !(26) dlsd   delta(log10(stress_drop)), ignored when L and W are set
!                    defines mean slip to mean radius ratio,from L and Mw
-1             !(27) AR         aspect ratio L/W;   -1: formula is used !
0.85           !(28) Mach          Mach = v_rupture_average/cS,
-999           !(29) ## inactive
1              !(30) isclaw        spectral scaling law, 1/2  means G83/Brune
18             !(31) L  set source length L; 0 means: do calclate -1 special use
9              !(32) W  set source width W;  0 means: do calculate -1 special use
1.2            !(33) dlsdhf  delta(log10(stress_drop)) for HF specral part
!              defines HF spectral level

! set rupture nucleation point position (x,y) in relative (L,W) units, taking the
! faultcenter as the origin/reference point (-0.5<xdlstart<0.5; -0.5<ydlstart<0.5)
!          case (-0.5 0):purely unilateral rupture to positive x, symmetric on y
-0.35          !(34)  xdlstart
0.2            !(35)  ydlstart
-99            !(36) ##reserve
-99            !(37) ##reserve
-99            !(38) ##reserve
-99            !(39) ##reserve
-99            !(40) ##reserve

! ===============group 3 numerical details
! defining n/T/dt/df combination.
! for DETHAZ case see mode_dethaz parameter in EXT block above
0.04            !(41) dt  time step  , or  -1: will calculate it from T and n
81.92           !(42) T duration of simulated signal; or -1: will calculate as dt*n
2048            !(43) n number of points, -1: auto mode, will find T from Mw; dt must
be set
20              !(44) lgMoshift subsource Modot values are in 10^lgMoshift units
-999            !(45) # reserved [see EXT block]
11             !(46) nxsub  no of subsources along x; (-1)auto;
3              !(47) nysub  no of subs. along y; (-1)==auto;  (1) line_source
-999           !(48) ## reserve
1.5            !(49) cdurrat    if n automatic, controls padding zeros; never <0.7!
!                               set 2-4  more for graphically nice spectrum;
0              !(50) crop      cropping in time of 90 deg pulse:(0)off,(1)on - do not
use, makes art
-999           !(51) #reserve [see FAR block]
-999           !(52) #reserve [see FAR block]]
0              !(53) ##internal use - Trupul
-999           !(54) ##reserve
-999           !(55) ##reserve
0.13           !(56) cpulsedur const used in smoothing HF spectrum, empirical choice,
-999           !(57) ##reserve
-999           !(58) ##reserve
-999           !(59) ##reserve
!================== defining simulation modes ================================
-999           !(60) ##reserve
1              !(61) mode_noaver:
```

```
!                      (1) simple mode, single run makes and uses unit pulse
!                      (0) advanced mode; uses average unit pulse shape
1              !(62) mode_makeaver
!                    (1) determine average pulse shape in this run;
!                        calculate it from multiple "maver" realizations
!                    (0) load prefabricated pulse shape from "pulsav"  file
0              !(63) mode_avertodisk
!                    (1) save just generated average pulse shape to file "pulfav"
!                    (0) do not save
1              !(64) mode_useaver
!                    (1) use average pulse (new or from disk) to make simulation
!                    (0) no simulation
25             !(65) maver  no of realizations for constructing average pulse
-888           !(66) @@@internal use:  modew_makepulse
-888           !(67) @@@internal use:  modew_usepulse
-999           !(68) ##reserve

!====== setting / controlling random seeds====================================
! the meaning for values of "common_nstate" and/or of individual "nstate"
! is as follows
!                   (-1) use fixed seed or do not perturb
!                   (0) use seed from clock;
!                   (any positive integer) use this number as a seed
1               !(69) mode_seedscommon:
!                     (0) use individual "nstate" value for each seed
!                     (1) use identical, same "nstate" values for all seeds,
!                     equal to "common_nstate";
!                     and ignore preset individual "nstate" values
0               !(70) common_nstate common "nstate" value;
-1              !(71) nstatet;     Mosub(t) seed     (-1|0|n) see above
-1              !(72) nstatex;     slip(x,y) seed    (-1|0|n)
-1              !(73) nstatev;     velocity(r) seed  (-1|0|n)
-1              !(74) nstatem;     Mach value seed   (-1|0|n)
-1              !(75) nstaten;     nucl point seed   (-1|0|n)
-1              !(76) nstates;     stress drop seed  (-1|0|n)
-1              !(77) nstateh;     HF stress drop seed (-1|0|n)
-1              !(78) nstatez;     reserve seed      (-1|0|n)
-1              !(79) nstatez2;     reserve seed      (-1|0|n)
-999            !(80)

! ============  setting boundaries for random parameters
! ============  within bounds, ass random parametrs are assumed to have uniform pdf
=========
-0.01          !(81) Mach_wind1, Mach+Mach_wind1 is lower limit for random Mach
+0.01          !(82) Mach_wind2; Mach+Mach_wind2 is upper limit for random Mach
-0.01          !(83) nuc_wind(1); lower limit for random increment of xdlstart
+0.01          !(84) nuc_wind(2); upper limit for random increment of xdlstart
-0.001         !(85) nuc_wind(3); lower limit for random increment of ydlstart
+0.001         !(86) nuc_wind(4); upper limit for random increment of ydlstart
0.01           !(87) dlsd_wind   (+/-) bounds for random increment of dlsd
0.01           !(88) dlsdhf_wind (+/-) bounds for random increment of dlsdhf
-999           !(89)
-999           !(90)


!=========== control parameters for rupture front propagation history,
!                                                   [copied to tfrpar]
1              !(91)  mode_circ; (1) set circular rupture front,
!                                (2) mode v_rupt(x,y)
-888           !(92)  @@@internal use
0.05           !(93)  vlow;     if circ front then    Vmin,     else   vlow
0.8            !(94)  vhigh;    if circ front then    Cvrange,  else   vhigh
1.3            !(95)  svpow;    spectral exponent for v-rupt field
0.8            !(96)  cvsmu;    spectral boundary for v-rupt field
3              !(97)  pmode;    type of v_rupt distribution
0              !(98)  ##ccorner;  const controls blind corners inoperative
0              !(99)  ##mode_show;  show propagation pictures inoperative
2000           !(100)  NM;       const controls propagation grid size
-999           !(101)
-999           !(102)
-999           !(103)
-999           !(104)
-999           !(105)
-999           !(106)
-999           !(107)
-999           !(108)
-999           !(109)
-999           !(110)
! =======finishes parameter list of 110 parameters==================================
!@@INT@@ BLOCK 2 ends  : INTERNAL SOURCE PARAMETERS  +++++++++++++++++++++++++++++++
```

The INT block contains input parameters, one parameter per line, free format with 1st column nonspace, to fill up to 15 first columns. For convenience only, each parameter line, in its later, comment part, has the parameter number in parentheses, from 1 to 110. Some positions are reserved and filled by -999.

Group 1: general info may be common to many sources

The first subgroup (parameters 1-24) contains general information, typically common to many simulated sources. The meaning of parameters is as follows:

(1) cs: c_S S-wave velocity of the medium, mainly used to set rupture velocity

(2) CMS: C_{MS} constant in the M_w-S relationship

(3) Css: c_S value used to modify mu in scaling source area

(4) C_heaton: C_{heaton} see text, default value 0.125

(6) ARlow: AR_l next six parameters define the M_w-AR relationship

(7) ARhi: AR_h

(8) MARlow: M_{wl}

(9) MARhi : M_{wh}

(10)Wbig: W_{big}

(11)pos_acc controls the use of Sort 1, Sort 2 and Sort 3 pulses (see text for details):

(1): all pulses of Sort 1 (tails to positive time)

(2): first half rupture duration Sort 1, then Sort 3 (tails to negative time).

(3): (default) first 13% of rupture duration Sort 1, last 13% of rupture duration Sort 3, in the middle the 50% to 50% random mix of Sort 1 and Sort 2 (symmetric in time, negative acceleration spike..

(12)CVAXY: CV for the scatter in slip amplitudes of subsources over x,y; (default 0.5)

(13)spowxy: exponent *p* of the power-law spectrum for slip over x,y; (default 1.75)

(14)cappowxy: exponent *g* for the "cap" or 2D taper function over x,y; (default 0.3)

(15)CVAT: CV of the scatter in pulse amplitudes over t; default 0.5

(16)spowt: exponent p of the power-law spectrum over t; default 0.5 see Eq. 12

(17)shappowt: exponent *g* for the 1D "shape" or envelope function over t; see Eq 13 for typeshapet=2,3,4 use defaults 0.7, 1, 0.5

(18)typeshapet: defines the type of 1D "shape" or envelope function over t:

(1) boxcar

(2) symmetric cap

(3) like x exp(-x)

(4) hyperbolic

(20)ctwind: local time history duration in units of Trise+Tgap; default value 1

(21)relscax: relative scatter of subsource position within a cell of the grid, along x, as a fraction of dsub; default value 0.65

(22)relscay: similar parameter with respect to y direction

(23)modefreeb: code that indicates where the free boundary is positioned: 0: none; 1000: -x, 0100: +x; 0010: -y; 0001: +y; 0010 is recommended for a surface fault, with +y looking downdip

NOTE: to emulate non-random slip with boxcar shape over x and y (the original Haskell case) use:

CVAXY=0; spowxy=0; cappowxy=-1;

to emulate non-random slip with smooth “parabolic hill” shape over x and y use

CVAXY=0; spowxy=0; cappowxy=1;

Group 2: specify a source

The second subgroup (parameters 25-40) specifies a particular source

(25)Mw: target moment magnitude M_w

(26)dlsd: δ or deviation of $\log_{10} \Delta\sigma$ from its typical/mean value. (default 0.0)

(27)AR: target *AR*; -1 calls default formula giving magnitude-dependent value

(28)Mach: target Mach value (average-v-rupture/c_S),

(30)isclaw: number of spectral scaling law, (1 means G83, default),(2 means Brune scaling law)

(31)L: *L* (default: 0 : do calculate it)

(32)W: *W* (default : 0 : do calculate it)

(33)dlsdhf: delta(log10(stress_drop)) for HF specral part; defines HF spectral level at a given true fc (Brune's fc depends on dlsdhf!)

(34)xdlstart *x*-position of rupture-starting point; in *L* units; the center of the rectangular source area, with the coordinates 0L, 0W, is the origin.

(35)ydlstart similarly, *y*-position of rupture-starting point; in *W* units.

NOTE: To set symmetric bilateral rupture set (xdlstart , ydlstart)=(0,0)

To set completely asymmetric along *x*, purely unilateral rupture, propagating “forward”, mostly to positive *x* direction, set (-0.5,0)

To set most asymmetric along *x* and *y*, purely unilateral,”diagonal” rupture, propagating “backward”, mostly to negative *x* direction, set (0.5,0.5) or (0.5,-0.5)

recommended default, mostly (“90%”) unilateral rupture: (xdlstart , ydlstart)=(-0.4,-0.2):

Group 3: numerical details

The third subgroup (parameters 41-55) specifies mostly the numerical details of simulation.

(41)dt: time step in s; may be set as -1, then found automatically from T and n

(42)T: duration of the simulated signal (modeling time window) in s; may be set as -1, and found automatically as dt*n

(43)n: time history length in counts: preset (n>0), or automatic (set as -1); when n>0, n=2^K is recommended; if not 2^K will be replaced by the next power of 2.
The case n=-1 is the case of automatic selection of duration. T is determined from the expected source duration based on Mw and dlsd. In this case, a definite time step dt must be set.
The automatic n value is found in the following way. At first the unilateral-propagation source duration Trupul is roughly estimated from the source length *L* (that typically is determined, in its turn, from M_w and δ), as, 1.25L/c_S.
(Later, more accurate estimate of the source duration are determined, denoted Tprop. Trupul is a preliminary value that is expected to be larger than or near Tprop.)
Then the modeling time window size T is determined as:
T=2*cdurrat*Trupul;

if T>10s, T=10s.
(For the “cdurrat” constant see comments below; default value cdurrat=0.7.)
Then n is determined as n=T/dt; then the final value n = nextpow2(n) is found , and T is redefined as T=dt*n.

(44)lgMoshift: output subsource Mo(t) and final Mo values are scaled down by $10^{lgMoshift}$ (default: 20)

(46)nxsub: n_x, no of subsources along *x,* if even will be increased by 1; Do not set nxsub less than 1/Cheaton. if nxsub=-1, then selected automatically .

(47)nysub: n_y, no of subsources along *y*; if even will be increased by 1; -1: automatic mode. Set 1 to define a line-source along x.

(49)cdurrat: controls padding zeros, that is adding a trail of zeros to the tail of the source time function. Operates only when work time window size T is not preset, see comments to dt/n parameters above. Default value 1. Use 2-3 to have nice-looking spectra

(50)crop: controls cropping (see text). 0: no cropping; 1: time cropping of 90 deg pulse

(56)cpulsedur: const used in smoothing HF spectrum, empirical choice 0.13

(57)mode_pict: show pictures? (1)=yes (0) no, inoperative in Fortran

Simulation modes

(61)mode_noaver;
(1) initial mode: simulation in a single run; no construction of the average unit pulse
(0) new mode use average pulse shape from many realizations

(62)mode_makeaver
(1)determine pulse shape by “maver” realizations, see “maver” below
(0) load it from "pulsav" file

(63)mode_avertodisk
(1) save generated average pulse shape to file "pulfav "
(0) do not save

(64)mode_useaver
(1) use average pulse (freshly simulated or from disk) to make real simulation
(0) no actual simulation with average pulse

(65)maver: number of realizations for constructing average pulse (default =25)

The recommended combinations of these parameters (mode_noaver mode_makeaver mode_avertodisk mode_useaver maver) are:

- (1 0 0 0 0) to be set for a simple simulation, not attempting to emulate realistically the scatter of amplitudes over realizations.
- (0 1 0 1 25) to be set for an advanced single simulation emulating scatter of amplitudes over realizations. This mode is slow, can be repeated 2-5 times to feel the realistic scatter.
- (0 1 1 0 25) to be used once, than many times use (0 0 0 1 0). This mode is an advanced multiple simulation, emulating scatter of amplitudes and sufficiently fast. In the first run, the pulfav file is created. The second step uses ready pulse from pulfav, it can be repeated as many as 10-20 times to estimate dispersion of results accurately

Parameters for setting / controlling/ random seeds and switching randomization on/off.

To control randomization and seeds, six “nstate” parameters with the generic names nstate# are used, where the # symbol is substituted by one of six letters indicating particular seed, giving nstatet, nstatex etc.

For these parameters, the possible values are integers -1, 0, +n, with the following meanings

(0) use seed from clock;

(n) use seed=n

(-1) use fixed seed or fixed parameter value

Similar meaning has the value of the common_nstate parameter.

(69)mode_seedscommon:
  (0) use individual, specific nstate value for each “nstate” variables
  (1) use the same "nstate" variables equal to common_nstate

(70)common_nstate: common "nstate" value, used only in case mode_seedscommon==1;

(71)nstatet: Modot-sub(t) seed

(72)nstatex: slip(x,y) seed

(73)nstatev: velocity(r) seed

(74)nstatem: Mach value seed

(75)nstaten: nucl point seed

(76)nstates: stress drop (dlsd) seed

(77)nstateh: HF stress drop (dlsdhf) seed

Parameters for setting boundaries for random parameters

(81) Mach_wind1: Mach+Mach_wind1 is lower limit for random Mach

(82) Mach_wind2; Mach+Mach_wind1is upper limit for random Mach

(83) nuc_wind(1): lower limit for random increment of xdlstart

(84) nuc_wind(2): upper limit for random increment of xdlstart

(85) nuc_wind(3): lower limit for random increment of ydlstart

(86) nuc_wind(4): upper limit for random increment of ydlstart

(87) dlsd_wind: (+/-) bounds for random increment of dlsd

(88) dlsdhf_wind: (+/-) bounds for random increment of dlsdhf

Within thus defined limits, all random parameters are assumed to have uniform probability density function.

Parameters to define rupture propagation history

(copied internally to tfrpar vector)

(91)mode_circ;
  (1) simulate circular rupture front
  (2) simulate 2D rupture propagation (inactive)

(92)vlow: if circular front then Vmin, else vlow

(93)vhigh: if circular front then Cvrange, else vhigh

(94)svpow: spectral exponent for v-rupt field; if circular front not used

(95)cvsmu: spectral boundary for v-rupt field; if circular front not used

(96)pmode: type of v_rupt distribution; if circular front not used

(97)NM: grid size for rupture front history simulation; if circular front not used

FAR block of *.src file

This block may be empty. FAR block can contain information on requested far-field STF traces. Two modes are possible to define rays: using as reference the fault plane, or world coordinates. FAR block consists of lines with 3 parameters each: two angles and wave propagation velocity (P,S; or the preferred one for surface waves) , free format, occupying first 30 columns. Angle data are understood in only one of these two ways, depending on the parameter mode_far_world: (No. 18 in the EXT block) In case mode_far_world==1; world reference is used for ray specification, and the angle pair is : (Azimuth from North, incident angle from zenith). In case mode_far_world==0, fault plane reference is used, and the angle pair is : (Azimuth from X axis, altitude above fault plane). Altitude of the ray to the far-field point is measured from fault plane (like latitude from equator); in other words, the distance of the ray from the fault normal is (90-altitude).All angles in degrees. The wave propagation velocity (km/s) defines relative delay between subsource time histories before summation. Setting this velocity very big (like 10^{16}) results in point source with isotropic STF or Mdot(t), same as one obtained in mode_point=1 case. In this case, STF is the same for any mode_world and any ray.

Here is an example:

```
!@@FAR@@ BLOCK  begins:  FAR-FIELD STF (SourceTimeFunction, often BODY WAVE pulse)
REQUEST +++++++++++++
!  to specify rays to far-field receivers use up to 60 angle pair + velocity triples;
!   a triple can fill 30 columns, can add comment in col. 30+.  Each triple consists
of:
!  either azumuth (to "X")-altitude pairs (mode_far_world=0) ;azimuth from X within
fault plane and
!  or     azimuth(from N)- angle-of-incidence pairs (mode_far_world=1) ;
!  3rd value in a line is wave velocity (P,S; or the preferred one for surface waves)
!  useful combinations
!0.   0.      1e16                !isotropic STF or Mdot(t) point soucce at any
mode_world and any ray
!0.   0.      3.5                 !at mode_world==0,  forward, along +x
!90.  90.     3.5                 !at mode_world==0,  normal to the fault, isotropic
STF Mdot(t) aga
!180. 0.      3.5                 !at mode_world==0,  backward, along -x

0.    0.     1e16                  !isotropic STF Mdot(t) point soucce at any
mode_world and any ray
90. 135.   3.5                     ! S wave to E at 45 deg down
270. 135.  3.5                     !
270. 45.   3.5                     !
!@@FAR@@ BLOCK  ends  :  FAR-FIELD STF REQUEST   +++++++++++++++++++++++++++++++++++++
++++++++++++++++++
```

FIL block of *.src file

This block may be empty. FIL block can contains information on additional files needed to run PULSYN06. They include:

- files that describe spectral models, in lines 1-5. In the current version, only model No.1 (defined by the value isclaw=1 in INT block, entry 30) is systematically needed. If no file is present, some dummy names must be present.

- files that specify preset source properties, in lines 6-9. In the present version, only final 2D slip distribution over a fault can be set in line 6.
- *.ste file in line 10. It can be set here as a preferred alternative to its setting in pul.par file. If line 10 contains “#”, the value from pul.par file is used.

Structure of data files.

File GUSEV83.TB5 (or its possible substitition) contains table with 12 lines of length 21. Line 1 is frequency grid. Line 2 is “rock ground-to-average crust” log spectral corrections from Gusev1983. Lines 3-12 contain the values of $\log_{10}(f^{2}M_0\mathrm{dot}(f))$ from Gusev 1983 for values of $\log M_0$(dyne cm) from 30 to 21.

File with final slip contains a table of final slip of size nysub nxsub organized in the following way.

each line/row contains a group size nysub organized along +y, or left to right along width in a slip picture; line/row sequence of size nxsub, top to bottom, is along –x (down along length in slip picture). This organization is visually transparent: data table can be put on the fault with rows ordered down dip and columns ordered back to strike. Note however that in the xyta and sbt/sbf files, another ordering scheme is assumed: within data group, data are ordered along +x, and data groups are ordered along +y.

Example:

```
!@@FIL@@ BLOCK  begins:  EXTERNAL FILE LIST  ++++++++++++++++++++++++++++
!                         can add comment in col. 30+
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!============== begins file name list ==========================
GUSEV83.TB5                   !(1) spectral scaling law tabulated as in Gusev 1983,
used if isclaw=1
X2.mof                        !(2) dummy spectral scaling data
used if isclaw=2
X3.mof                        !(3) dummy spectral scaling data
used if isclaw=3
X4.mof                        !(4) dummy spectral scaling data
used if isclaw=4
M0dot.dat                     !(5) spectral table adjusted by mk_external_sp.m, used
if isclaw=5
test.fsl                      !(6) preset final slip  data, to substitute simulated
one,
X7.tfr                        !(7) preset front time data, to substitute simulated one
X8.xxx                        !(8) dummy
X9.xxx                        !(9) dummy
#                             !(10) ste file name like X10.ste; use # when *.ste is
set by pul.par f

!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!@@FIL@@ BLOCK  ends  :  FILE LIST   ++++++++++++++++++++++++++++
```

OIS block of *.src file

In contrast to all previous blocks of *.src file, that contained input data, OIS block contains output parameters of source creation. Also modified/calculated versions of input parameters like n, L,W, AR, etc, are given here. In the end, there is a time stamp. Note that if a non-initial version of *.src file is present before the run of pulsyn06, old OIS block is lost and replaced by a new one in the each new run.

Here is the example of OIS block, with comments on its right in oblique-bold.

```
!@@FIN0@@## this line is the final line of raw, non-appended *.src file #########
```

```
!@@OIS@@ BLOCK 6 begins: INTERNAL SOURCE OUTPUT PARAMETER TABLE  ++++
!+++++++++110 data lines, integer: 30; real: 80  +++++++++++++++++++
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

        4096    ( 1)    n              no of time/frequency counts in result
          27    ( 2)    nsub           number of subsources, nxsub*nysub
          30    ( 3)    nw             number of time histories/spectra in sbf/sbt file
           9    ( 4)    nxsub          actual nxsub
           3    ( 5)    nysub          actual nysub
           2    ( 6)    ixnuc          number of the subsource taken as the starting
                                       one, along x
           2    ( 7)    iynuc          number of the subsource taken as the starting
                                       one, along y
          80    ( 8)    ntshift        time shift: for all subsources, time count
starts                                      ntshift*dt s after the beginning of the
time                                   sample sequence
          20    ( 9)    lgMoshift      copy of input value
        -999    (10)
          13    (11)    jstatet        actual nstate of the random number generator,
for                                    nstatet seed defines  Mo-sub(t)
          13    (12)    jstatex        actual nstate of the random number generator,
for                                    nstatex seed defines slip(x,y)
          13    (13)    jstatev        actual nstate of the random number generator,
for                                    nstatev seed defines rupture velocity(x,y)
          13    (14)    jstatem        actual nstate of the random number generator,
for                                    nstatem seed defines Mach value
          13    (15)    jstaten        actual nstate of the random number generator,
for                                    nstaten seed defines nucleation point
          13    (16)    jstates        actual nstate of the random number generator,
for                                    nstates seed defines stress drop seed
The above “nstate” values can be used to generate repeatedly any particular “random”
realization (e.g. for demonstration)

          -1    (17)
        -999    (18)
..............................................................
        -999    (30)
    20.0000     (31)    L              actual L
    10.0000     (32)    W              actual W
    71.0000     (33)    S              source rectangle area, S=LW
    6.25000     (34)    Trupul         explained above
    .781250     (35)    Trise          explained above
    .488281E-02 (36)    dt             actual time step
    .500000E-01 (37)    df             actual frequency step, work window duration is
                                       T=1/df
    2.22222     (38)    dxsub          actual average subsource grid cell size along x
    3.33333     (39)    dysub          actual average subsource grid cell size along y
    3.20000     (40)    vrupmean       actual mean rupture velocity
    5.17894     (41)    Tprop          propagation time rough estimate
    5.96019     (42)    Trup2          duration of the skeleton source
    .113013     (43)    fcgro          1/Trup2
    .952851E-01 (44)    fc0            1/Trupul
    4.06591     (45)    PF             Peak factor of  ( “acceleration”) time function
    .868311     (46)    Cvacc          coefficient of variation for  function
    1.26964     (47)    RatPF          the ratio of actual PF to the value expected
                                       for a segment of Gaussian process.
    125893.     (48)    Mo_sc          scaled Mo value; true value is
Mo_sc*10^lgM0shift
    25.1000     (49)    algMo          true log10(Mo) value, dyne cm
   -.601545     (50)    dlsdnew        perturbed, actual dlsd value
    2.00000     (51)    ARnew          new AR
    .800000     (52)    Machnew        new Mach
   -.380000     (53)    xdlstartnew    new xdlstart
    .000000     (54)    ydlstartnew    new ydlstart
    .710000     (55)    Chill          as is
    1.62844     (56)    dsub0          as is
    1.73611     (57)    Tgap           as is
    2.51736     (58)    Trisew         as is
    5.03472     (59)    Twind          as is
    3.33333     (60)    xnuc           accepted nucl. point x-coord
    5.00000     (61)    ynuc           accepted nucl. point y-coord
   -9999.00     (62)    vrep1          inactive
    .000000     (63)    v2             inactive
    .000000     (64)    v3             inactive
    .000000     (65)    v4             inactive
    5.17894     (66)    Tprop          actual propagation time
    5.96019     (67)    Trup2          internal parameter
    10.2197     (68)    Trup3          internal parameter
    .167780     (69)    fc2            internal parameter
    .113013     (70)    fcgro          internal parameter
    .952851E-01 (71)    fc0            internal parameter
    .774825     (72)    Tacfwind       internal parameter
    .372942     (73)   fl            internal parameter
    1.35615     (74)   fh            internal parameter
    1.00003     (75)   corrlf        internal parameter
    1.03939     (76)   corrhf        internal parameter
    125889.     (77)   aMo0          internal parameter
    .178826     (78)   am1puls       internal parameter
    11.0967     (79)   am1pulsv      internal parameter
    1.25868     (80)   am1envt       internal parameter
    119685.     (81)   aMo1          internal parameter
    .900000     (82)   dlhf
    141.254     (83)   SM
-999.000        (84)
-999.000        (85)
```

TIME STAMP follows:

```
!++++   executed on 2003-11-27   at 19h41m59s ++++++++++++++++++++++
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!@@OIS@@ BLOCK 6 ends  :  INTERNAL SOURCE OUTPUT PARAMETER TABLE ++++
```

*.ste – site/station file

Site/ station data file *.ste consists of 2 data blocks; also comments (!xxxx...) and empty lines are permitted. Generally, it is assumed that around each receiver the flat Earth surface and its boundary has z coordinate equal to zero; the receiver is located on the surface or maybe deeper (option not active). Under a receiver, there is a flat-layered structure and wave propagation is simulated according to this structure model. (Of course, this is not a fully consistent description of real wave propagation through a real 3D Earth structure; but this is a reasonable approximation in many cases.) Absractly speaking, we permit a unique structure to be associated with each receiver. For practical aims however, we assume that there is a small number of possible structures and much larger number of receivers. To make description of the situation systematic, a 4-digit number (like 0001, 0002, … 0010, … 9999) is ascribed to each structure. This “structure number ” is used by DETHAZ package as a root of the name of the data file that contains specification of this structure. However we consider structure numbers as less convenient and introduce additionally mnemonic structure codes that are arbitrary 3-character mnemonic codes. These codes are used exclusively within a *.ste data block.

Correspondingly, a *.ste file has an initial data block that describes how the structure numbers are related to 3-character mnemonic codes. The sequence of structure numbers in this block must be increasing. In the second, main data block, that is in the site table, structures associated with each site are specified by their mnemonic codes. An option is provided to limit further calculation of pulsyn06 by a particular subset of structures and therefore of stations.

Here is a sample inital block:

```
+0001 R00  0.0010    rock or type A(US)
+0003 S00  0.0000    mix of: deep-soil-basin, and deep-soil-non-basin
 0004 I00  0.0020    shallow-soil
```

Each line contains (Fortran-style format in parentheses):

- Structure usage byte (A1) ; it contains + or space, meaning that calculation is on or off for this structure. Write + to use all stations with a particular structure, or space to switch them off
- ground/structure number code (I4.4)
- mnemonic structure code (1X,A3), columns 7-9
- kappa value for the stucture (2X,G8.5), columns 12-21 , (optional)

- arbitrary comments starting col 22 (optional)

The end of the first block is marked by ### in the beginning of the line.

Then the second, main data block follows; it consists of station/site lines, one line per station.

Sample lines follow:

```
+001 ARL   34.236 -118.439 D    B    dsb  S00 Arleta-NordhoffFireSta
+002 CNP   34.212 -118.606 D    C    dsb  S00 Canoga Park-Topanga Can
+003 JEN   34.313 -118.499 D    B    dsb  S00 Jensen Filter Plant
+004 LA0   34.106 -118.454 A    A    ro   R00 LA 00
```

Each line contains (format in parentheses):

- Station usage byte (A1): must be non-space and non-exclamation mark for the station to be included in the calculation;
- Number (I3,1X) ordinal number in this list, not used in further calculations (optional)
- Name-code(A4): 4-characters station/site name-code, recommended letters small;
- Latitude(1X,F7.3): latitude in degrees and decimals of degree
- Longitude(1X,F8.3): longitude in degrees and decimals of degree
- Ground type1(1X,A3): code of ground type according to classification scheme 1 (e.g. International Building Code(IBC) with ground classes A/B/C/D/E) (optional)
- Ground type2(1X,A3): code of ground type in classification scheme 2 (e.g. ROK/MED/SFT) (optional)
- Ground type3(1X,A3): code of ground type in classification scheme 3 (e.g. non-basin - basin) (optional)
- Mnemonic ground/structure code(1X,A3): mnemonic code of ground/stucture, i.e. of the velocity-density vertical profile to be accociated with a given site/station, obligatory.
- Full station name(1X,A24): as is in the data source (optional)

Output files

xxx999.xta – subsource parameter table

Output file XXX999.xta has 9 columns with nsub values in each column. Each row specifies a subsource and contains: x(along L), y(alongW), rupture front arrival time(s), seismic moment, x, y and z=depth coordinates in the Cartesian reference specified in the EXT block of *.src file, and latitude-longitude pair.

x and y should be understood as positions in fault-intrinsic coordinate system (x_fp along strike, y_fp down-dip), in km. Arrival time in s. Subsource moment values are in (dyne*cm)*10^lgMoshift units, typically in (dyne*cm)*10^{20} units.

xxx999.sbt/.sbf - subsource time history

As a standard version, the subsource time history description (as time counts or spectra) is contained in a single *.sbt or *.sbf file. (However there is a possibility to write each individual subsource spectrum to an individual file (up to 99 files). In this special case, individual files are named xxx999_01.sbf, xxx999_02.sbf, etc.)

The output file XXX999.sbt in the basic version has nw or nw2 columns with n real values in each column. Here nw =nsub or na for the regular case (subsources) or far-field case, and nw2=nw+3 when additional three time functions are requested. When nw=nsub, columns represent the values of discrete moment rate function of each subsource, time step dt. The true time scale is assumed to start at the moment of the onset of the "hypocenteral" subsource. Within the working time window, this moment is shifted nshift counts later than the first count . When nw=na, columns contain "point source time functions" for a far-field observers located on different rays. Rays are defined by the FAR block in the *.src file. In the case of nw2=nw+3 data columns, far field time function for three particular observers are added. These three observers are assumed to be located: (1) at the normal to fault plane, (2) at the +x ray and (3) at the –x ray. Subsource moment rate function values are in (dyne*cm/s)*10^lgMoshift units, typically in (dyne*cm/s)*10^{20} units.

Alternatively, the output file XXX999.sbf in the basic version has nw columns of the same structure with n complex values in each column that represent the discrete values of the Fourier transform of the moment rate function of each subsource, frequency step df=1/T. The time scale is modified in the same way as in previous paragraph in connection with ntshift value.

The same data can be also packed differently. In addition to the basic case, the following options are present:

1. each line is filled by the spectrum, and there are nw or nw2 lines; in this case only n/2+1 complex values are keeps within a line.
2. Alternatively, the literally same structure can be produced as unformatted file.
3. One more option is to put each spectrum into a separate file, with naming convention like XXX999_01.sbf, XXX999_02.sbf ….

These options are controlled by the `modeout` variable, see its description above for more info.

To fix details, write statements follow for different modeout values (nw2 can be nsub or nsub+3):

```
complex*16 subdo(nmax,nsubmax); real*8 subd(nmax,nsubmax);
integer n    !actual time/spectral array size, counts
integer nw2  !no of subsources, or (no of subsources) + 3
. . . . . .
nfd=n/2+1; !no points for compact files in F domain
selectcase (modeout)
case (0,-13,-3)  ! T-domain case, all traces in one REAL matrix,   n  lines
   wname=trim(iopath)//batch//var//'.sbt'
   open(1,FILE=wname);!
     subd=real(subdo);!
     do  j=1,n; write(1,'(500(1X,g15.9))') (subd(j,isub),isub=1,nw2 );  enddo;!!
   close(1);
case (1,11)    ! F-domain case 1: all subevent/farfield spectra in one COMPLEX
               !                 matrix, f along line n/2+1 values, nw lines
   open(1,FILE=trim(iopath)//batch//var//'.sbf');!
      write(1,'(2i8,f8.4,i8)')nw2,nfd,T,n
     do isub=1,nw2; write(1,'(8193(1X,g16.9))')(subdo(j,isub), j=1,nfd ); enddo;!
   close(1)
case (2,12)    ! F-domain case 2: subevent/farfield spectra each in separate
file,
               !                up to nw2=99 files
   do itrace=1,nw2
     write(word,'(i2.2)')itrace ;
     wname=trim(iopath)//batch//var//'_'//word//'.sbf'
     open(1,FILE=wname);!
    write(1,'(''     frequency       Modot-real       Modot-imag'')')
       do j=1,n; ffi=(j-1)*df; write(1,'(3(1X,g16.9))')ffi,subdo(j,itrace);
enddo;!
     close(1)!
   enddo
case (3,13)     ! F domain case 3, spectra in one COMPLEX matrix, susources
                !                 along line,  n lines
     open(1,FILE=trim(iopath)//batch//var//'.sbf');!
     do  j=1,n ; write(1,'(500(1X,g16.9))')  (subdo(j,isub),isub=1,nw2 );
enddo;!
     close(1)!
case (4,14)     ! F domain case 4, spectra in one COMPLEX matrix f along line,
!                 n/2+1 values in a line, nw lines
     open(1,FILE=trim(iopath)//batch//var//'.sbf', FORM='UNFORMATTED');!
    write(1)nw2,nfd,T,n
       do  isub=1,nw2; write(1)  (subdo(j,isub), j=1,nfd );  enddo;!
     close(1)!
endselect
```

xxx999.frt/.frf - far-field time histories

The far-field or STF time histories are packed identically to the sbt/sbf case just above. The order follows the order in the FAR block of src file.

Auxiliary file pulfav

In the special mode (mode_avertodisk=1), pulsyn06 creates the file pulfav. It contains the average source spectrum obtained from the requested realisations. It can be used by later runs of pulsyn06 instead of the target curve table (e.g. GUSEV83.TB5 or similar) when mode_useaver=1 and mode_makeaver=0. A user is advised to erase any existing pulfav file before a run of pulsyn06 with new source data. Here, “new” means almost any changes of fault/source parameters. The only significant parameter that can be changed keeping the same pulfav is nstatet. However, when working in “mode_noaver=1” mode, no problems can arise.

Output files for the interface with the DETHAZ package

xxx999.sut

Contains subsource data in external coordinates for DETHAZ package and conforms to its requirements. Sample file:

```
     label         slon      slat    depth   strike    dip   rake     mag   isub    weight       tshift
 0001aaXXXXYYYY -118.6273  34.4439  10.357 122.000  40.000 101.000 -9.00    1   .1000E+01   .0000E+00
 0001aaXXXXYYYY -118.6067  34.4333  10.357 122.000  40.000 101.000 -9.00    2   .1000E+01   .0000E+00
 0001aaXXXXYYYY -118.5862  34.4227  10.357 122.000  40.000 101.000 -9.00    3   .1000E+01   .0000E+00
 0001aaXXXXYYYY -118.5657  34.4121  10.357 122.000  40.000 101.000 -9.00    4   .1000E+01   .0000E+00
 0001aaXXXXYYYY -118.5452  34.4014  10.357 122.000  40.000 101.000 -9.00    5   .1000E+01   .0000E+00
 0001aaXXXXYYYY -118.5247  34.3908  10.357 122.000  40.000 101.000 -9.00    6   .1000E+01   .0000E+00
 0001aaXXXXYYYY -118.5042  34.3802  10.357 122.000  40.000 101.000 -9.00    7   .1000E+01   .0000E+00
 0001aaXXXXYYYY -118.4837  34.3695  10.357 122.000  40.000 101.000 -9.00    8   .1000E+01   .0000E+00
 0001aaXXXXYYYY -118.4632  34.3589  10.357 122.000  40.000 101.000 -9.00    9   .1000E+01   .0000E+00
 0001aaXXXXYYYY -118.6421  34.4245  12.500 122.000  40.000 101.000 -9.00   10   .1000E+01   .0000E+00
 0001aaXXXXYYYY -118.6215  34.4139  12.500 122.000  40.000 101.000 -9.00   11   .1000E+01   .0000E+00
. . . . . . . . . . .
 0001aaXXXXYYYY -118.5338  34.3413  14.643 122.000  40.000 101.000 -9.00   25   .1000E+01   .0000E+00
 0001aaXXXXYYYY -118.5133  34.3307  14.643 122.000  40.000 101.000 -9.00   26   .1000E+01   .0000E+00
 0001aaXXXXYYYY -118.4928  34.3200  14.643 122.000  40.000 101.000 -9.00   27   .1000E+01   .0000E+00
```

xxx999.obs

Contains site/station data for DETHAZ package and conforms to its requirements. Sample file:

```
       lon      Lat     struc    rdep
  -118.4390  34.2360   0003     .000
  -118.6060  34.2120   0003     .000
  -118.4990  34.3130   0003     .000
  -118.4540  34.1060   0001     .000
  -118.4830  34.2810   0001     .000
```

xxx999kkkk.isg

Contains subsource-station pair data and conforms to requirements for DETHAZ package. “kkkk” pattern component in the file name is the 4-digit “ground number”- numeric code of a structure or velocity profile associated with all the stations/sites included in the particular file (see description of *.ste file). The number of *.isg files generated by Pulsyn06 depends on the number of “active” structures in the yyy888.ste file; here “active” means there is at least one “active” station with such a structure.

Sample file:

```
  INPUT ".isg" to GF calculation, npair=   81      ground structure no/code =0003/S00  .ste file: n94002.ste
 ARL 00055 2  1  0    .000   0    0 0 0  .1000E+02  .5000E-01  201 4096  28.870  10.357 338.931  40.000 101.000 GENERAL
 1 1  .0000E+00 0 0  .0000E+00  .0000E+00  .0000E+00 FORCE SYSTEM AND DURATION
 0 0  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00 INSTRUMENT
 0  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00 122.000 FILTER AND EXTRA
    1AAXXXXXXXX -118.6273  34.4439  10.357 122.000  40.000 101.000 -9.00    1   .1000E+01   .0000E+00 SOURCE
  999AAYYYYYYYY -9.00     0 -118.4390  34.2360  003  28.870 143.069  28.870   .9987E+00    1    .000 RECEIVER
 ARL 00056 2  1  0    .000   0    0 0 1  .1000E+02  .5000E-01  201 4096  26.808  10.357 337.223  40.000 101.000 GENERAL
 1 1  .0000E+00 0 0  .0000E+00  .0000E+00  .0000E+00 FORCE SYSTEM AND DURATION
 0 0  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00 INSTRUMENT
 0  .0000E+00  .0000E+00  .0000E+00  .0000E+00  .0000E+00 122.000 FILTER AND EXTRA
    1AAXXXXXXXX -118.6067  34.4333  10.357 122.000  40.000 101.000 -9.00    2   .1000E+01   .0000E+00 SOURCE
  999AAYYYYYYYY -9.00     0 -118.4390  34.2360  003  26.808 144.777  26.808   .9986E+00    2    .000 RECEIVER
 ARL 00057 2  1  0    .000   0    0 0 1  .1000E+02  .5000E-01  201 4096  24.774  10.357 335.233  40.000 101.000 GENERAL
```

. . . .

Below follows the copy of the original description of *.isg file of the DETHAZ package. There is a slight visual difference related to the fact that within the label (“sislab” variable) in the “GENERAL” line, station code name is present in the output of PULSYN

***.isg**: input parameters for the computation of synthetic seismograms. Each .isg file contains the input for all the GreenFunctions associated with a single structural model.

```
 I N P U T   P A R A M E T E R S   F O R   13  S E I S M O G R A M (S)
      13 1  1  0 0.0000E+00   0    0 0 0 0.1000E+01 0.5000E-02  201 4096 0.4105E+02 0.1000E+02 0.7840E+02
0.5100E+02-0.1070E+03 GEN.
 1 1 0.0000E+00 0 0 0.0000E+00 0.0000E+00 0.0000E+00 FORCE SYSTEM AND DURATION
 0 0 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 INSTRUMENT
 0 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.1140E+03 FILTER AND EXTRA
 002aa026041    11.100  44.100   7.0 114 51-107  6.3    1  0.1000E+01  0.0000E+00: SOURCE
 002aa026041  6.3   0   11.400  44.400   1  41.05  35.6  41.05  0.1000E+01: RECEIVER
      23 1  1  0 0.0000E+00   0    0 0 1 0.1000E+01 0.5000E-02  201 4096 0.1367E+02 0.1000E+02 0.3157E+03 0.5900E+02
0.2340E+03 GEN.
 1 1 0.0000E+00 0 0 0.0000E+00 0.0000E+00 0.0000E+00 FORCE SYSTEM AND DURATION 0 0 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 INSTRUMENT
 0 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.1000E+03 FILTER AND EXTRA
 001aa027043    11.300  44.500  18.0 100 59 234  6.2    1  0.1000E+01  0.0000E+00: SOURCE
 001aa027043  6.2   1   11.400  44.400   1  13.67 144.3  13.67  0.1000E+01: RECEIVER
```

FORTRAN statements to read the file:

```
      read(3,1)sislab,itype,npint,iflg,thkr,modest,modetr,igrdmo,
     1             marflg,fifr,delfrq,nfqm,npts,dist,thk,phideg,deldeg,xlmdeg
    1 format(1x,a7,i2,2i3,e11.4,2i4,2i2,2e11.4,2i5,5e11.4)
      read(3,2)iptf,itystf,durat,istd,ifinit,frbsou,falen1, falen2
    2 format(2i2,e11.4,2i2,3e11.4)
      read(3,3)insres,itycal,xmag,t0,dampra,t1,t2,h1,h2,sigsq,tg,geodam
    3 format(2i2,10e11.4)
      read(3,4)igauss,cmaxf,pfocf,cutamp,extra1,extra2,extra3
    4 format(i2,6e11.4)
      read(3,5)iev,lev,elon,elat,xdepth,iangl,idip,irake,xma,nsub,weight,tshift
    5 format(1x,i3,a8,f9.3,f8.3,f6.1,i4,i3,i4,f5.1,i5,2e12.4)
      read(3,6)ne,lev1,xma1,ipsou,slon,slat,istr,dist1,azim,distr1,weight
    6 format(1x,i3,a8,f5.1,i4,f9.3,f8.3,i4,f7.2,f6.1,f7.2,e12.4)
```

EXPLANATION OF DATA

value in example

```
record 0 (title)
   number of seismogram to be created by program                    13
record 1 (“gen”)
  sislab          name (or num) of seismogram [seismogram label]
  itype           type (1-displ., 2-vel., 3-accel)                   1
  npint           interpolation (0, 1-9) - from box                 1
  iflg            ground motion component =0                        0
  thkr            receiver depth -  from box                        0.E0
  modest,modetr   first,last used modes (0-all)                     0 0
  igrdmo          input ground motion (0-calc, 1-read in)           0
  marflg          input(1)/calculate(0) eigenfunctions – computed   0

  fifr            final frequency - from box                        0.1E1
  delfrq          spectrum frequency “interval” (step) - from box   0.5E-2
  nfqm            number of frequency points (==201) - from box     201
  npts    number of time series points (2**n)=1024,..8192 - from box  4096
  dist            distance source-receiver (km) – computed          0.41E2

  thk             source depth (km)                                 0.1E2
  phideg          angke (strike-receiver) with strike               0.78E2
  deldeg          dip  - (see idip in box) = idip in rec.5          0.51E2
  xlmdeg          rake - (see irake in box) = irake in rec.5        -0.107E2
```

```
1 1 0.0000E+00 0 0 0.0000E+00 0.0000E+00 0.0000E+00 FORCE SYSTEM AND DURATION
      read(3,2)iptf,itystf,durat,istd,ifinit,frbsou,falen1, falen2
    2 format(2i2,e11.4,2i2,3e11.4)

record 2 ("force system and duration")
  iptf           force type (1 = double couple)
  itystf         time function (1=step)
  durat          source duration (=0)
  istd           shape of time function
  ifinit         source fitiness (0-point)
  frbsou         rupture velocity not used
  falen1         fault length along strike not used
  falen2         fault length anti-strike not used
record 3 ("instrument") – actually not used
record 4 ("filter and extra")
  igauss         flag for computation of low-pass gaussian filter
  cmaxf          cutoff frequency (Hz)
  pfocf          percentage of cutoff with unit response
  cutamp         amplitude at cutoff
  extra1,extra2,extra3,extra4,extra5  not used (extra5=iangl - see record5)
record 5("source")
  iev            index of source in *.fps not used
  lev            aa-run name, ix,iy - coordinates on grid, not used
  elon,elat      coordinates of source (longitude,latitude)
  xdepth         depth of source in used fps (info only)
  iangl          fps: strike
  idip,irake     fps: dip and rake (= deldeg and xlmdeg in rec.1)
  xma            magnitude (== xma1 in rec.6)
  nsub           number of the subsource
  weight         weight for the subsources
  tshift         shift time for subsources not used,=0
record 6("receiver")
  ne             index of seismogenic zone not used
  lev1           2 symbols- name of run, xxxyyy - two int value (see file *.sut)
  xma1           magnitude (==xma in rec.5) not used
  ipsou          index poligon source not used
  slon,slat      coordinates of site (longitude,latitude)
  istr           index of structure for receiver (# by order in file por)
  dist1          == dist (rec.1)
  azim           azimut from source
  distr1         dist in use (in the case *.obs may differ from dist in record1)
  weight         correction factor for distance when dist.ne.distl
```

FAQ

1Q. How to make pulsyn06.out to generate identical output in several runs

1A. In the *.src file, (block INT) set

mode_seedscommon(#69) =1 and   common_nstate(#70)=k

where k is any natural number

Computation of source spectra in the far field approximation (makegus.out)

PULSYN06 can be used to compute source spectra in far field approximation as described in section about FAR block of src file. *makegus.out* uses those features of PULSYN06 to compute a set of source spectra in far field approximation in a simplified way.

Required input files

`makegus.par`	file with input parameters
`GUSEV83.TB5`	spectral scaling table

Steps to perform the parametric tests

The example input files can be found in

/XDST/Examples/PulsynExamples/Makegus/Base

1) Preparation of the parameter file

File `makegus.par`, shown below, contains all the parameters that can be changed by the user to configure the set of source spectra. Name of spectral scaling table is defined in the parameter file, so instead of `GUSEV83.TB5,` another can be used.

```
Parameter file for program makegus
---------------------------------------------
bil                            root filename (max 3 char)
GUSEV83.TB5                    spectral scaling tabulated as in Gusev 1983
10           maximum frequency
9            interpolation
2            rupture type (1=unilateral, 2=bilateral, 3=random)
2            curve type (1=old,2=type guphas090)
2            number of realizations
1            random seed of first realizations
0 180 90     directivity angle (start, stop, step)
5 6 1.0      magnitude (start, stop, step)
30  5        fault size (length x width, in km) if magnitude is fixed
```

2) Computation of the source spectra

The command that will execute the computation is:

```
makegus.out
```

The program calls the programs *`pulsyn06.out`* to compute the source spectra and the script *`plotslip.sh`* to plot the slip maps.

3) Main output files

Parameters defined in example `makegus.par` will produce several files, the main are:

- files with source spectra:

  ```
  bilr01000.50
  bilr01000.60
  bilr01090.50
  bilr01090.60
  bilr01180.50
  bilr01180.60
  bilr02000.50
  bilr02000.60
  bilr02090.50
  bilr02090.60
  bilr02180.50
  bilr02180.60
  ```

- plot of slip maps (in png and ps formats):

  ```
  bilr01.50.slip.png
  bilr01.60.slip.png
  bilr02.50.slip.png
  bilr02.60.slip.png
  bilr01.50.slip.ps
  bilr01.60.slip.ps
  bilr02.50.slip.ps
  bilr02.60.slip.ps
  ```

- src files with input for PULSYN that produced source spectra:

  ```
  bilr01.src.50
  bilr01.src.60
  bilr02.src.50
  bilr02.src.60
  ```

Names of source spectra follow this scheme:

`nnnrssddd.mm`

where:

`nnn`	three digit for root filename
`r`	fixed
ss	two digit for random seed
`ddd`	directivity angle (e.g.: 000=forward, 090=neutral, 180=backward)
`mm`	magnitude*10