Finite differences above the MODEL package

Václav Bucha & Luděk Klimeš
Department of Geophysics, Charles University, Ke Karlovu 3, 121 16 Praha 2, Czech Republic, Fax: +420-2-2191 1214,
E-mails: bucha@seis.karlov.mff.cuni.cz, klimes@seis.karlov.mff.cuni.cz

Summary

Fortran code for 2-D P-SV elastic second–order finite differences has been split into program generating gridded "effective" elastic parameters and the finite–difference program. Both programs have been modified to read and write data using the subroutines of package FORMS and are located in program package FD. This enabled to prepare the interface between package MODEL and the finite–difference program and to use the programs of package FORMS to process and display the results of finite–difference package FD. The usage of software package FD is illustrated on two numerical examples. The differences between 2-D finite–difference seismograms and 3-D ray–theory seismograms are demonstrated and discussed in the second example.

Keywords

Seismic models, elastic waves, synthetic seismograms, finite differences, ray theory.

1 Introduction

This paper introduces program package FD. The basis of the FD package is the code for 2-D P-SV elastic second–order finite differences written by Jiří Zahradník (Zahradník 1995, Zahradník & Priolo 1995). The code, however, utilized specific technique to build the models of geological structures. The technique was not compatible with the models specified by means of subroutine package MODEL (Červený, Klimeš & Pšenčík 1988). The Fortran code for 2-D P-SV elastic second–order finite differences has thus been split into program `fdmod.for` generating gridded "effective" elastic parameters and finite–difference program `fd2d.for`. Both programs have been modified to read and write data using the subroutines of package FORMS and are located in program package FD. This enabled to prepare the interface program `modfd.for` between package MODEL and finite–difference program `fd2d.for`, and to use the programs of package FORMS to process and display the results of finite–difference package FD. For example, FD synthetic seismograms may be plotted together with ray–theory synthetic seismograms or travel–time marks by program `sp.for` of package FORMS and the wavefield snapshots can be converted into the series of PostScript figures using package FORMS, which facilitate the creation of the wavefield movies.

The usage of software package FD is illustrated on two numerical examples. The differences between 2-D finite–difference seismograms and 3-D ray–theory seismograms are demonstrated and discussed in the second example. The examples also illustrate the "history files", designed to simplify the computations. The history files contain both the input data and the instructions how to run the programs.

Package FD is included on SW3D-CD-2 and subsequent compact disks, together with the documentation and sample input data.

In: Seismic Waves in Complex 3-D Structures, Report 8 (Department of Geophysics, Charles University, Prague, 1999), pp. 171–192

2 "Effective" elastic parameters

Let us define three elastic parameters,

$$a = \varrho\, v_\mathrm{P}^2 \quad , \tag{1}$$

$$b = \varrho\, v_\mathrm{S}^2 \quad , \tag{2}$$

$$c = \varrho\,(v_\mathrm{P}^2 - 2\, v_\mathrm{S}^2) \quad , \tag{3}$$

where ϱ is the density, v_P and v_S are P and S wave propagation velocities, respectively. In the following equations, parameter m stands for either a, b and c. We introduce parameters m^{11}, m^{13}, m^{31} and m^{33}, $m = a, b, c$. Parameters m^{11} and m^{13} are averaged in the direction of the x^1 axis. Parameters m^{31} and m^{33} are averaged in the direction of the x^3 axis. Parameters m^{11} and m^{33} are averaged along the corresponding grid legs. Parameters m^{13} and m^{31} are averaged along the corresponding inter–legs. The above notation may simply be generalized to 3-D. Note that the former notation was $m^{11} = m^\mathrm{HL}$, $m^{13} = m^\mathrm{HS}$, $m^{31} = m^\mathrm{VS}$ and $m^{33} = m^\mathrm{VL}$.

Assume that the basic 2-D regular rectangular grid has $\mathtt{N1} \times \mathtt{N3}$ gridpoints, denote grid intervals by `D1`, `D3` and the coordinates of the first gridpoint by `O1`, `O3`. In the present version, $\mathtt{D1} = \mathtt{D3}$ is required. Then the following discrete values are calculated and stored in disk files for the 2-D finite–difference program

$$m_{ij}^{11} = \frac{x_{i+1}^1 - x_i^1}{\int_{x_i^1}^{x_{i+1}^1} \left[m(x^1, x_j^3)\right]^{-1} \mathrm{d}x^1} \;, \quad i = 1, 2, ..., \mathtt{N1}{-}1 \;, \quad j = 1, 2, ..., \mathtt{N3} \;, \tag{4}$$

$$m_{ij}^{13} = \frac{x_{i+1}^1 - x_i^1}{\int_{x_i^1}^{x_{i+1}^1} \left[m(x^1, x_{j+\frac{1}{2}}^3)\right]^{-1} \mathrm{d}x^1} \;, \quad i = 1, 2, ..., \mathtt{N1}{-}1 \;, \quad j = 1, 2, ..., \mathtt{N3}{-}1 \;, \tag{5}$$

$$m_{ij}^{31} = \frac{x_{j+1}^3 - x_j^3}{\int_{x_j^3}^{x_{j+1}^3} \left[m(x_{i+\frac{1}{2}}^1, x^3)\right]^{-1} \mathrm{d}x^3} \;, \quad i = 1, 2, ..., \mathtt{N1}{-}1 \;, \quad j = 1, 2, ..., \mathtt{N3}{-}1 \;, \tag{6}$$

$$m_{ij}^{33} = \frac{x_{j+1}^3 - x_j^3}{\int_{x_j^3}^{x_{j+1}^3} \left[m(x_i^1, x^3)\right]^{-1} \mathrm{d}x^3} \;, \quad i = 1, 2, ..., \mathtt{N1} \;, \quad j = 1, \mathtt{N3}{-}1 \;, \tag{7}$$

$$\varrho_{ij} = \varrho(x_i^1, x_j^3) \;, \quad i = 1, 2, ..., \mathtt{N1} \;, \quad j = 1, 2, ..., \mathtt{N3} \;, \tag{8}$$

where

$$x_i^1 = \mathtt{O1} + (i-1)\,\mathtt{D1} \quad , \tag{9}$$

$$x_j^3 = \mathtt{O3} + (j-1)\,\mathtt{D3} \quad . \tag{10}$$

If the gridline locally coincides with an interface, values (4), (5), (6) and (7) should be calculated at both sides of the gridline and arithmetically averaged. If the gridpoint is situated at an interface, values (8) should be calculated at the four corners, created by the crossing gridlines, and arithmetically averaged. Such an averaging is especially important at the free surface.

2.1 Modification of the "effective" elastic parameters

Since the finite–difference scheme by Zahradník (1995) and Zahradník & Priolo (1995) is stable and accurate for the model composed of homogeneous square grid cells, we may modify the above "effective" parameters so that they correspond to homogeneous grid cells as suggested by Opršal & Zahradník (1998 section 5, 1999),

$$\overline{m}^{13}_{ij} = \overline{m}^{31}_{ij} = \{\tfrac{1}{2}[(m^{13}_{ij})^{-1} + (m^{31}_{ij})^{-1}]\}^{-1} \ , \quad i = 1, 2, ..., \mathtt{N1}{-}1, \quad j = 1, 2, ..., \mathtt{N3}{-}1 \ , \tag{11}$$

$$\overline{m}^{11}_{i1} = m^{13}_{i1} \ , \quad \overline{m}^{11}_{i\,\mathrm{N3}} = m^{13}_{i\,\mathrm{N3}} \ ,$$

$$\overline{m}^{11}_{ij} = \tfrac{1}{2}(m^{13}_{i\,j-1} + m^{13}_{ij}) \ , \quad i = 1, 2, ..., \mathtt{N1}-1 \ , \quad j = 2, ..., \mathtt{N3}-1 \ , \tag{12}$$

$$\overline{m}^{33}_{1j} = m^{13}_{1j} \ , \quad \overline{m}^{33}_{\mathrm{N1}\,j} = m^{13}_{\mathrm{N1}\,j} \ ,$$

$$\overline{m}^{33}_{ij} = \tfrac{1}{2}(m^{13}_{i-1\,j} + m^{13}_{ij}) \ , \quad i = 2, ..., \mathtt{N1}-1 \ , \quad j = 1, 2, ..., \mathtt{N3}-1 \ . \tag{13}$$

However, it seems more natural to replace equations (5), (6) and (11) with 2-D averaging

$$\overline{m}^{13}_{ij} = \frac{(x^1_{i+1} - x^1_i)(x^3_{j+1} - x^3_j)}{\int_{x^1_i}^{x^1_{i+1}} \int_{x^3_j}^{x^3_{j+1}} [m(x^1, x^3)]^{-1} \mathrm{d}x^1 \mathrm{d}x^3} \ , \quad i = 1, 2, ..., \mathtt{N1}{-}1 \ , \quad j = 1, 2, ..., \mathtt{N3}{-}1 \ , \tag{14}$$

or similar one. The "effective" elastic parameters are thus worth to be further studied.

3 Visualization of the wavefield snapshots

The 2-D finite–difference program calculates two displacement components u_1 and u_3. We wish to conveniently display the wavefield in a single snapshot per each time level. The authors decided to calculate the divergence and rotation of the displacement, which roughly correspond to P and S waves, respectively, if the heterogeneities are smooth and the waves are split. The divergence has the only component perpendicular to the 2-D grid. The divergence and rotation may than be scaled in colours and displayed within a single snapshot.

Displacement components u_1 and u_3 are numerically recalculated by simple finite differencing into wavefield partial derivatives $u_{1,1}$, $u_{1,3}$, $u_{3,1}$, $u_{3,3}$ that are used for computation of the divergence and rotation. The divergence and rotation are scaled by $\sqrt{v^3 \varrho}$ to preserve the amplitudes when a wave pass through local heterogeneities,

$$DIV = \mathrm{div}(\mathbf{u})\sqrt{v_P^3 \varrho} = (u_{1,1} + u_{1,1})\sqrt{v_P^3 \varrho} \ , \tag{15}$$

$$ROT = \mathrm{rot}(\mathbf{u})\sqrt{v_S^3 \varrho} = (u_{1,3} - u_{3,1})\sqrt{v_S^3 \varrho} \ . \tag{16}$$

Of course, this scaling does not compensate for the geometrical spreading nor scattering attenuation (intrinsic attenuation is not considered in the present version).

Three input parameters `SAMP`, `VSAT` and `VSAT4` are presently used to display the couple of values DIV and ROT in colours. Scaled rotation ROT is multiplied by S–wave scaling factor `SAMP` and vector (DIV, `SAMP` ROT) is projected onto a colour circle of time–variable radius

$$R = \mathtt{VSAT} \exp(-\mathtt{VSAT4}\ t) \ . \tag{17}$$

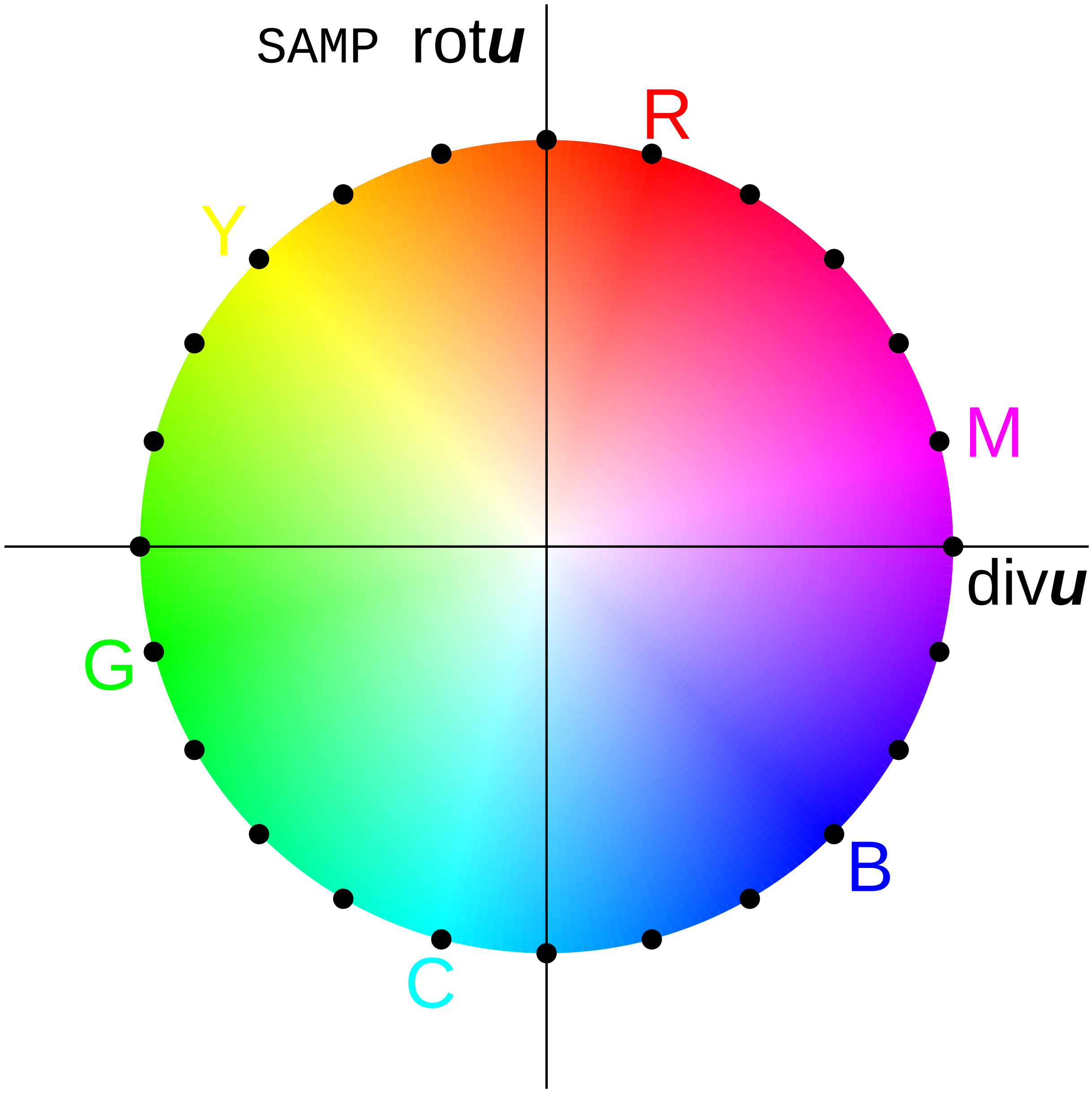

Figure 1. Colour scale to display snapshots. The axis denoted div$u$ represents P waves and the axis `SAMP`rot$u$ represents S waves. The hue corresponds to the type of wave and saturation to the amplitude.

The colour circle used by the authors to display the snapshots is shown in Figure 1. Parameter `VSAT4` has been introduced to compensate for the attenuation, both intrinsic and scattering. However, its compensation for the geometrical spreading corresponding to point or linear sources is poor.

To manually adjust parameters `SAMP`, `VSAT` and `VSAT4`, we plot the time dependence of $\|DIV\|$, `SAMP`$\|ROT\|$ and R in a single diagram like Figure 4 and change parameters `SAMP`, `VSAT` and `VSAT4` until we are satisfied. Recommended norm $\| \bullet \|$ over the grid is the maximum absolute value.

The authors are going to automatize this manual and non–convenient colour–scaling procedure in the future.

4 Finite–difference package FD

The creation of the package started by splitting the former finite differences code written by Jiří Zahradník into two programs `fdmod.for` and `fd2d.for`, followed by the modification of input and output data forms. In many cases, the backward compatibility with the former code has been kept. The package is gradually supplemented with new program codes.

2-D finite–difference package FD consists of the following Fortran 77 programs (for the complete file list see the SW3D-CD-3 disk):

`modfd.for` — Calculation of "effective" elastic parameters (4) to (8) for 2-D elastic finite differences in models specified by means of the MODEL package.

`fdmod.for` — Generation of "effective" elastic parameters (4) to (8) in 2-D models specified in terms of homogeneous convex polygons (former way).

`newpar.for` — Modification (11) of the "effective" elastic parameters for 2-D elastic finite differences. Modifications (12) and (13) are assumed to be performed by program `fd2d.for` on input (designed for future versions).

`fd2d.for` — Calculation of P-SV elastic waves in 2-D models by the finite–difference method of the second order accuracy in both space and time. No absorption is presently included.

4.1 Input data

SEP (Stanford Exploration Project) like *history files* are designed to simultaneously contain input parameters for all programs and to specify the execution of the programs. The source and receiver coordinates are expressed in terms of the model units (formerly by grid indices), their names and coordinates are located in separate files.

4.2 Output seismograms

Computed seismograms are saved in the GSE format that enables efficient processing by the programs of package FORMS. The amplitudes of the output displacement are now independent of the grid step.

4.3 Output snapshots

Snapshots represent displacement distribution in the entire model grid at selected time levels. Snapshots for horizontal and vertical components are saved in two files. The snapshots may be displayed as described in Section 3.

5 Examples

Three history files were prepared to demonstrate the usage of the FD package. History files `fd.h` and `u2d-fd.h` are described in this section. History file `rm-fd.h` has been discussed by Klimeš (1998).

5.1 Demo model for FD (edge of a sedimentary basin)

The simple model of the edge of a sedimentary basin, shown in Figure 2, has been prepared by Jiří Zahradník as a demo model for the 2-D finite–difference code. The former specification of the model is located in data file `fdmod.dat` and may be used by the first part of the former FD program (`fdmod.for`). The model then has been transformed into the format used by program package MODEL, named `modfd.dat` and is ready for program `modfd.for` to calculate the “effective” elastic parameters for program `fd2d.for`.

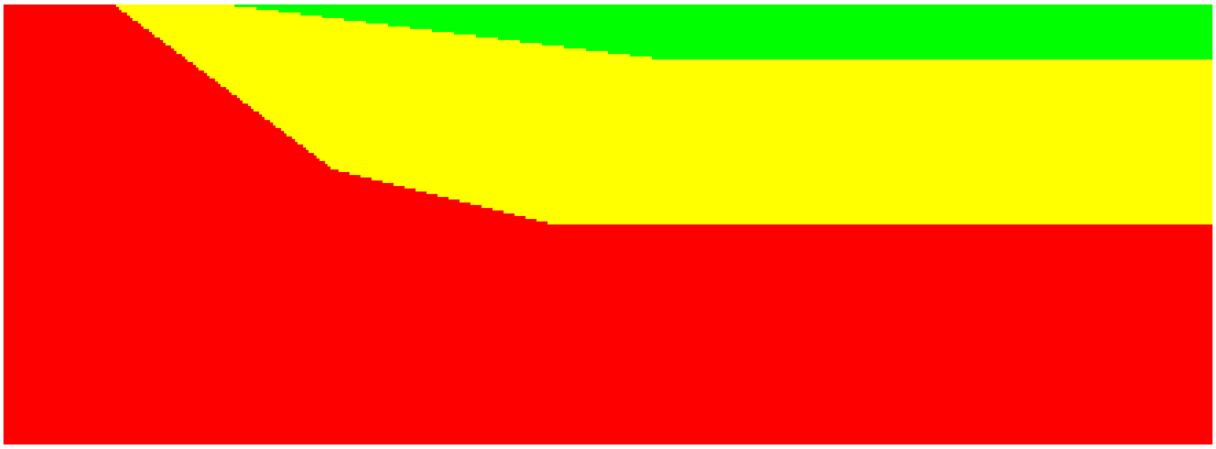

Figure 2. Model `modfd.dat`.

5.1.1 Finite–difference synthetic seismograms

History file `fd.h` is designed to illustrate finite–difference calculation, plotting of synthetic seismograms and the method to display snapshots.

History file `fd.h` begins with comments, which are preceded by “#”.

```
# SEP history file with demo data for 2-D FD
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Input files required
 #chk.pl: "fd/" "modfd.dat"
 #chk.pl: "fd/" "fdmod.dat"
 #chk.pl: "fd/" "fdsrc.dat"
 #chk.pl: "fd/" "fdrec.dat"
 #chk.pl: "fd/" "snapdiv.cal"
 #chk.pl: "fd/" "snaprot.cal"
 #chk.pl: "fd/" "snap.cal"
 #chk.pl: "fd/" "addrec.cal"
 #chk.pl: "fd/" "fdmax.pl"
 #chk.pl: "fd/" "sep.pl"
```

Everything from hash mark "#" to the end of line is assumed to be a comment and is ignored. However, comments containing no equals "=" nor colons ":" need not be preceded by a hash mark, because only equals "=" and colons ":" are interpreted as commands in the history files.

Each "=" (equal sign) immediately preceded by a string is interpreted as the PARAMETER=VALUE couple specifying the value of the parameter named PARAMETER.

Each ":" (colon) immediately preceded by a string is interpreted as the instruction to execute a program. The string between the last space (or the beginning of the line if there are no spaces to the left of ":") and the colon is the name of the program. If the name contains substring ".pl", it is assumed to represent a Perl script. Otherwise, it is assumed to represent executable program, compiled from the Fortran source. Everything to the right from the colon is assumed to represent the data for the program read from the standard input (Fortran) or command line (Perl).

For example, the lines containing "chk.pl:" are the instructions to execute Perl script chk.pl, but disabled by comment sign "#" in the second column. The lines serve as the list of input data files. If the lines are uncommented and Perl script chk.pl is located in the current directory, it checks for the presence of the data files in the current directory. Moreover, if the value of global variable $SW3D in Perl script go.pl is properly edited, Perl script chk.pl (calling subroutine CHK of go.pl) may be employed to copy the data files into the current directory. If the file is not found, go.pl should stop the execution. The string preceding the data filename is the path to the root directory of the SW3D software (corresponding to subdirectory unix/ or dos/ of the compact disk) where the original of the data file is located. However, specification of the path to the root directory of the SW3D-CD-3 software requires to manually edit file go.pl. Moreover, stopping the execution sometimes does not work in the current version (e.g., under MS-DOS). That is why the instructions to run Perl script chk.pl are commented.

```
# Data specifying the model by means of the MODEL package
  MODEL='modfd.dat'

# Alternative (old) model specification for program 'fdmod.for'
  FDMOD='fdmod.dat'
# (this backward compatibility is included just for test purposes)
```

The first PARAMETER=VALUE couple informs the programs that the value of parameter MODEL is string 'modfd.dat', i.e., that the file specifying the model is named 'modfd.dat'.

Parameter FDMOD specifies the file with the former model representation and is included just for debugging and test purposes. Otherwise, it would not be used.

```
# Grid dimensions, FD time steps, snapshots
  N1=111    N3=41     NTFD=1601      N4=321
  D1=1.000  D3=1.000  DTFD=0.000860  D4=0.004300
  O1=0.000  O3=0.000                 O4=0.000
```

Vertical axis x_3 points downwards, free earth surface is situated at the depth of $x_3 = 0\,\mathrm{m}$. Grid interval is 1 m. The number of gridpoints is small in order to perform the demo computations rapidly.

```
# Type and position of the point source
  SRC='fdsrc.dat' # file with the position of the point source
  KSIG=3          # Kuepper signal (with 2 local maxima)
  SIGF=3.333333   # ref. frequency of the Kuepper signal
  SIGA=1.1241E10  # SIGA=SQRT(27)*DEN*DX**3/DT**2, DEN=1600
```

The point explosive source is located at the depth of $x_3 = 2\,\mathrm{m}$, at $x_1 = 60\,\mathrm{m}$. Source coordinates are stored in file `fdsrc.dat`. We selected the Küpper (known also as Müller) signal with 2 local maxima. The reference frequency of 3.333333 Hz corresponds to reference period 0.3 s.

The profile of 12 receivers with spacing 10 m regularly covers the whole grid top, in spite of two leftmost and two rightmost receivers located inside the dumpers. The names and coordinates of the receivers are stored in file `fdrec.dat`.

```
# Files with receiver positions and synthetic seismograms
  REC='fdrec.dat'
  SS='fd.gse'

# Form of files with effective elastic parameters and snapshots
  FORM='formatted'

# Files with snapshots
  SNAP1='fd1.out'
  SNAP3='fd3.out'
```

The output GSE file with synthetic seismograms will be named `fd.gse` by finite-difference program `fd2d.for` and will be read by seismogram plotting program `sp.for`.

Snapshots for the horizontal and vertical components are saved in two files defined by SEP parameters `SNAP1` and `SNAP3` in `fd.h`. The files are not created if the parameters are equal to `' '` (default). The default form is `'formatted'`. Note that programs of package FORMS, processing the snapshots, work just with the formatted files in version 5.30.

```
# Files with effective elastic parameters
  A11='a11.out'  B11='b11.out'  C11='c11.out'
  A13='a13.out'  B13='b13.out'  C13='c13.out'
  A31='a31.out'  B31='b31.out'  C31='c31.out'
  A33='a33.out'  B33='b33.out'  C33='c33.out'
  DEN='den.out'
  # NEWPAR=1  AA13='aa13.out'  BB13='bb13.out'  CC13='cc13.out'
```

The files with "effective" elastic parameters (4) to (8) are written by program `modfd.for` and are read by finite–difference program `fd2d.for`. The parameters commented off are intended for the future extension to modified "effective" elastic parameters (11). Since this feature is not debugged, it is also not documented.

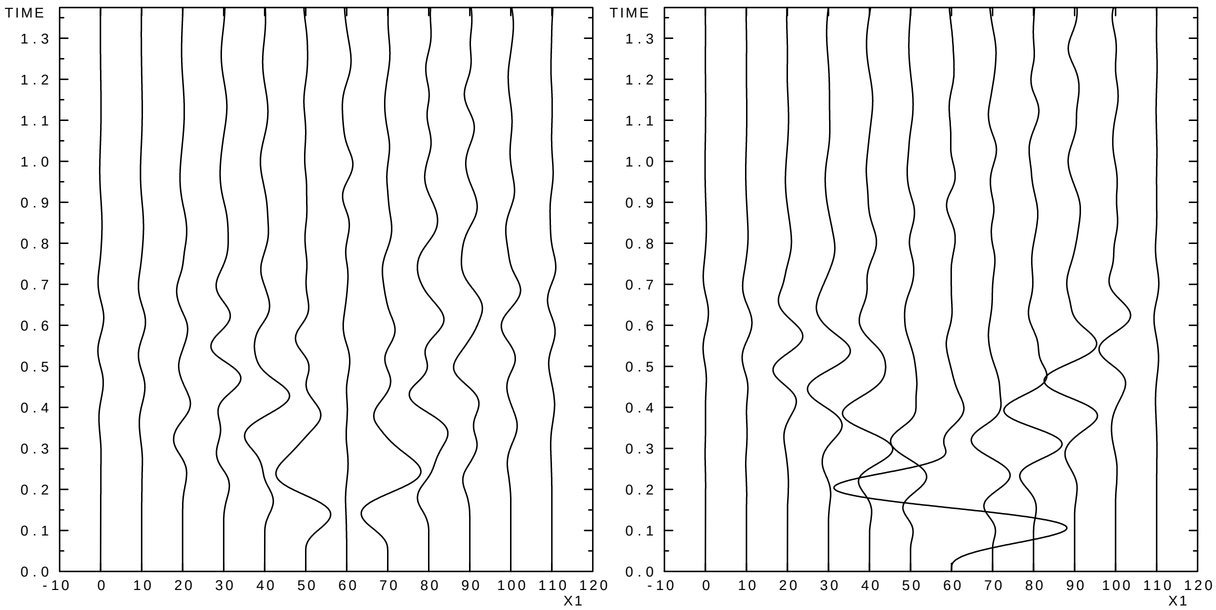

Figure 3. Horizontal (left) and vertical (right) components of the finite–difference seismograms in the demo model. Note that two leftmost and two rightmost receivers are situated inside the dumpers.

```
# Velocities for snapshot processing
  VP='vp.out'
  VS='vs.out'

# Data to control seismogram plotting (program SP)
  SP1='fd1.ps'  SP2=' '  SP3='fd3.ps'
  KODESP=1  SPCHRH=0.25
  SPTMIN=0.000  SPTMAX=1.375  SPTLEN=13.75  SPTDIV=13.75  SPTSUB=2
  SPXMIN=-10    SPXMAX=120    SPXLEN=13.0   SPXDIV=13  SPXSUB=1
  NORMSP=1  SPAMP=0.08
```

The seismograms are plotted in the whole finite–difference time domain, between times `SPTMIN`=0 and `SPTMAX`=1.375 (in seconds), the length of the time axis is `SPTLEN`=13.75 (in cm), etc. The amplitude scaling is given explicitly, see Figure 3.

```
# Programs to execute
  modfd:   'fd.h' /           # gridded elastic parameters
 #fdmod:   'fd.h' /           # just for test purposes
  fd2d:    'fd.h' /           # FD synthetic seismograms
  sp:      'fd.h' /           # seismogram plotting

#.......................................................................
```

Program `fdmod.for` is included just for debugging and its execution is disabled by commenting. Program `modfd.for` generates the "effective" elastic parameters. Finite–difference program `fd2d.for` calculates the wavefield and writes synthetic seismograms at the given receivers and snapshots at specified time levels into disk files. Program `sp.for` plots the synthetic seismograms.

5.1.2 Displaying snapshots

```
# Programs to execute to prepare snapshots for plotting
  grdfd:  'fd.h' 'fd1.out' 'fd11.out' ' ' 'fd13.out' /
  grdfd:  'fd.h' 'fd3.out' 'fd31.out' ' ' 'fd33.out' /
  grdcal: 'fd.h' 'snapdiv.cal' 'fd11.out' 'fd33.out' 'vp.out' 'den.out' 'fdp.out' /
  grdcal: 'fd.h' 'snaprot.cal' 'fd13.out' 'fd31.out' 'vs.out' 'den.out' 'fds.out' /
```

Program `grdfd.for` numerically calculates the first partial derivatives of the 2-D displacement by finite differencing. Program `grdcal.for` then transforms the derivatives according to files `snapdiv.cal` and `snaprot.cal` into scaled divergence (15) and scaled rotation (16), respectively. The divergence and rotation are normalized so as to locally preserve the amplitudes during the propagation.

Several parameters may be used to influence the appearance of the snapshots (`VREF`, `CREF`, `VCIRC`, etc., for details see `grdps.for`).

```
# Universal parameters to display snapshots
  UNIT='pt'      YSIGN=-1
  VREF=0.000     CREF=-.208333  # Colour of positive divergence is -5/24
  VCIRC=6.283185
  VWHITE=0.000
  RED=0  GREEN=0  BLUE=0        # Undefined values will be black

# Parameters to display these snapshots
  SAMP=0.500                    # S-wave amplification factor
                                # 2-D equipartition law: SAMP=VS/VP
                                # Seismic force: SAMP=(VS/VP)**1.5
  VSAT=193178    VSAT4=2.000    # Colour saturation (amplitude) control
# VSAT=SQRT(1600*(100/0.35)**3)
  N1NEW=1  N2NEW=1  N3NEW=1  GNORM=999.
  fdmax.pl: fd.h fdp.out fds.out fdmax.ps
# Now display 'fdmax.ps' to check the values of SAMP, VSAT and VSAT4!
```

It is necessary to display and edit file `fdmax.ps` to adjust multiplication factor `SAMP` for the rotation (S wave) to level out the amplitudes of the divergence and rotation and to adjust parameters `VSAT` and `VSAT4` controlling the colour saturation (for details see `fdmax.pl`). The adjusted values then should be entered into the history file. After this correction, the execution of the history file may be resumed by commenting off all previously executed programs, uncommenting the subsequent programs and running the history file again. Here `GNORM=999` specifies that the maximum of the absolute values is used as the norm over the grid in PostScript file `fdmax.ps`, see Section 3. For `GNORM=2`, root mean square would be used, etc., see program `grdnorm.for`.

```
# Denoting receivers by undefined values and making them black
  PTS='fdrec.dat'  PTSGRD='ptsgrd.out'
 #ptsgrd: 'fd.h' /
 #grdcal: 'fd.h' 'addrec.cal' 'fdp.out' 'ptsgrd.out' 'fdprec.out' /
```

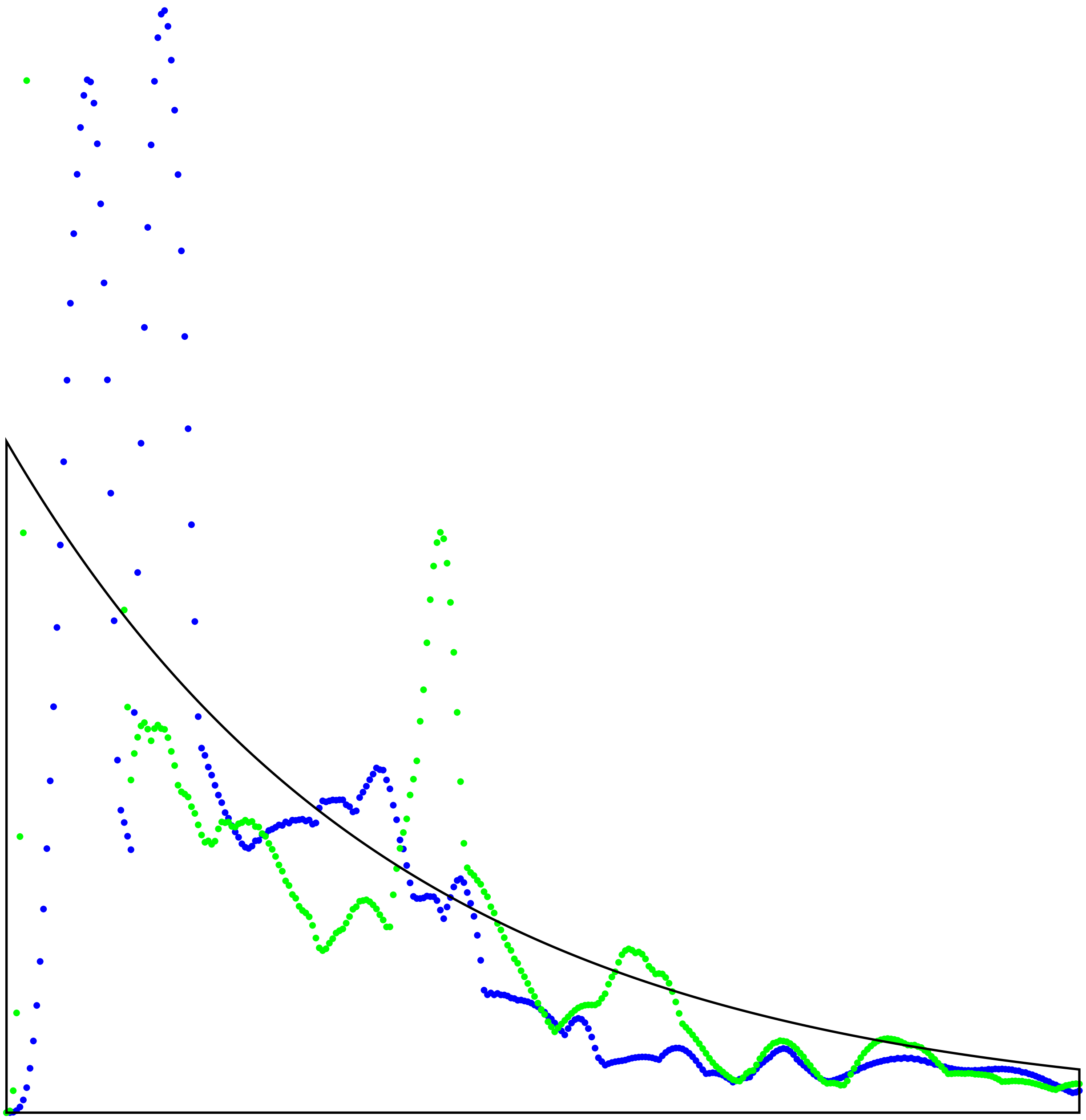

Figure 4. Maximum absolute values $\|DIV\|$ [green] and `SAMP`$\|ROT\|$ [blue], see (15) and (16), plotted with the curve `VSAT` $\exp(-$`VSAT4` $t)$ of the maximum colour saturation. Note that the values at small times, corresponding to the vicinity of the explosive source, are far above the paper.

Program `ptsgrd.for` reads points from file given by `PTS` and writes spatial grid containing undefined values at the points and zeros elsewhere into file given by `PTSGRD`. File `'fdprec.out'` then differs from `'fdp.out'` just by undefined values at the receivers.

```
# Programs to execute to plot snapshots
 #grdcal: 'fd.h' 'snap.cal' 'fdprec.out' 'fds.out' 'fda.out' 'fdn.out' /
 #grdps:  'fd.h' 'fda.out' 'fd000001.ps' ' ' 'fdn.out' /
```

Programs `grdcal.for` and `grdps.for` create the PostScript figures of individual snapshots, named `fd000001.ps`, `fd000002.ps`, etc. For selected snapshots (`fd000011.ps`, `fd000021.ps`, ..., `fd000321.ps`) refer to Figure 5.

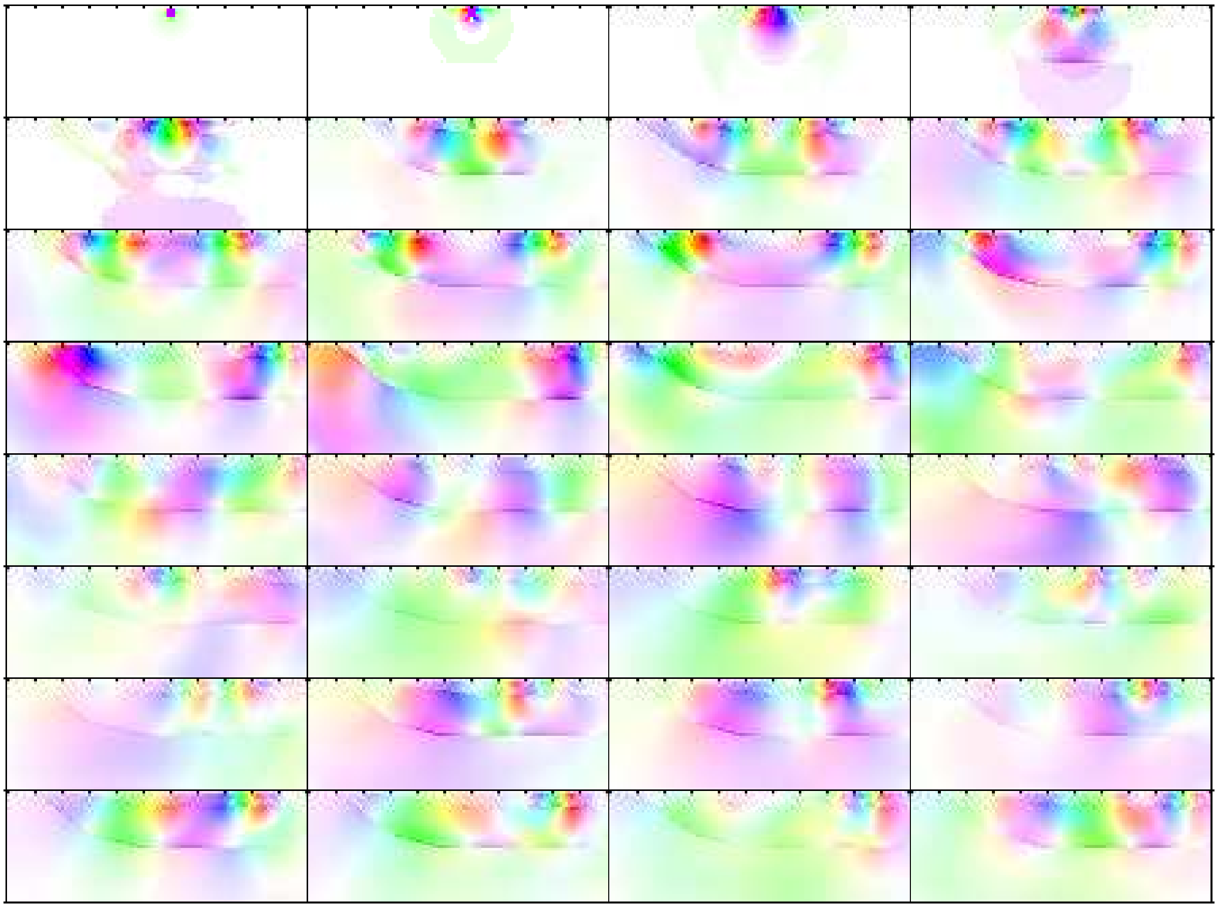

Figure 5. Snapshots at time levels 0.043 s, 0.086 s, ..., 1.376 s in the demo model.

```
# Mask for snapshot PS files and the name of the animated GIF file
  SNAPS='fd00*.ps'  MOVIE='fd.gif'

# Conversion of PS files to animated GIF (special software needed)
 #fdpcx.pl:    fd.h            # conversion to PCX
 #fdgiflux.pl: fd.h            # conversion to GIF (Linux)
 #fdgifdos.pl: fd.h            # conversion to GIF (MS-DOS)
```

Finally, Perl script `fdpcx.pl` may be used to convert PostScript figures into the PCX format. Other two scripts (`fdgiflux.pl`—Linux, `fdgifdos.pl`—DOS) may convert the PCX files to the GIF format and merge them to a single animated GIF file. These scripts should be edited before use to check whether the required third–party programs like `gs`, `display`, `gifmerge`, etc., are available. As an example, you may find the finite–difference GIF animation created by the `fd.h` history file on SW3D-CD-2 and subsequent compact disks.

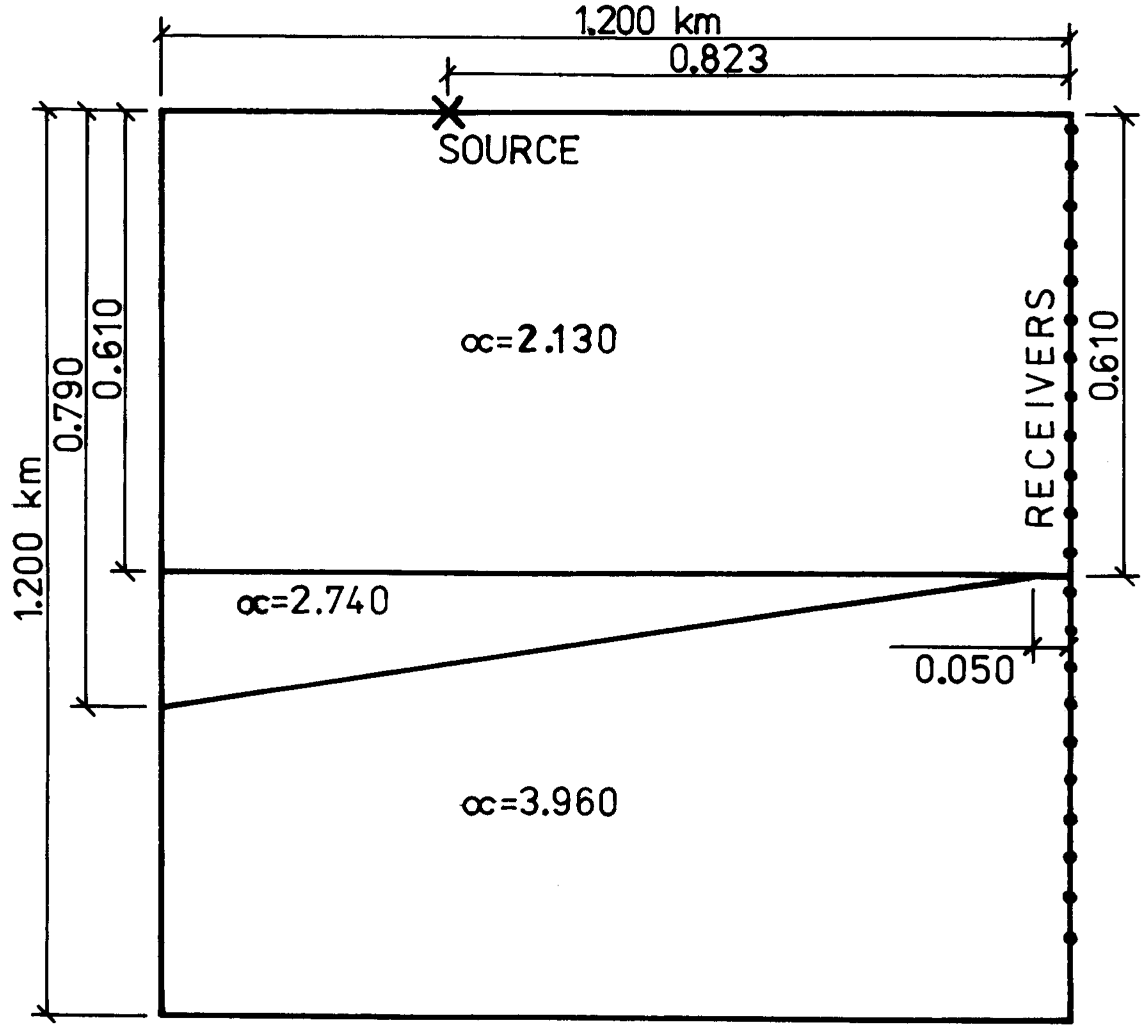

Figure 6. Two-dimensional model UNCONFORMITY by Cormier & Mellen (1984) composed of 3 homogeneous blocks. P-wave velocities are denoted by α.

5.2 Synthetic seismograms in 2-D model UNCONFORMITY

The 2-D model UNCONFORMITY by Cormier & Mellen (1984), situated in a vertical square 1.2km×1.2km, is composed of 3 homogeneous blocks separated by planar interfaces. The geometry of the model, P-wave velocities α and the positions of the point source and of 22 receivers are depicted in Figure 6. The origin of Cartesian coordinate system was selected in the left top corner, with horizontal axis x_1 pointing to the right and vertical axis x_3 pointing upwards. The model and the calculation of the ray–theory synthetic seismograms by the programs of package CRT were described by Klimeš (1996b).

The density in all 3 blocks is given by

$$\frac{\varrho}{\mathrm{g\ cm^{-3}}} = 1.7 + 0.2\,\frac{\alpha}{\mathrm{km\ s^{-1}}} \quad . \tag{18}$$

The S-wave velocities are not specified in the input data file (named `u2d-mod.dat`) for the `MODEL` package. Default S-wave velocities

$$\beta = 0.57735\,\alpha \tag{19}$$

are thus assumed by the subroutines of the `MODEL` package.

History file `u2d-fd.h` is designed to compare 3-D ray–theory and 2-D finite–difference synthetic seismograms in the 2-D model UNCONFORMITY. The data are located in subdirectory `u2d/` of the CRT package.

History file `u2d-fd.h` requires programs of packages FORMS, MODEL, CRT and FD. It can be executed on systems supporting Perl scripting language (e.g., Unix or MS-DOS) by typing a command like "`perl go.pl u2d-fd.h`". Perl program `go.pl` of package FORMS copies input history file `u2d-fd.h` line by line to output history file `u2d-fd.out`. When a "`:`" (colon) instruction to execute the program is encountered, the program is executed with the name of the output history file substituted for the name of the input history file in order to hide the PARAMETER=VALUE couples between the "`:`" (colon) instruction and end of file to the program. If you wish to name the output history file, e.g., `new.out`, the history file may be executed by command "`perl go.pl u2d-fd.h new.out`".

History file `u2d-fd.h` begins with some comments and the list of input data files.

```
# Comparing ray and FD seismograms in 2-D model UNCONFORMITY
# ============================================================
# Input files required
 #chk.pl: "model/u2d/" "u2d-mod.dat"
 #chk.pl: "crt/u2d/"   "u2d-crt.dat"
 #chk.pl: "crt/u2d/"   "u2d-rpa.dat"
 #chk.pl: "crt/u2d/"   "u2d-wri.dat"
 #chk.pl: "crt/u2d/"   "u2d-src.dat"
 #chk.pl: "crt/u2d/"   "u2d-rec.dat"
 #chk.pl: "crt/u2d/"   "u2d-hil.dat"
 #chk.pl: "crt/"       "sourcex.dat"
```

If the value of global variable `$SW3D` in Perl script `go.pl` is properly edited, Perl script `chk.pl` may be employed to copy the data files into the current directory by uncommenting the corresponding lines.

```
# Model UNCONFORMITY (Cormier and Mellen 1984)
  MODEL='u2d-mod.dat'

# Source and receivers
  SRC='u2d-src.dat'  REC='u2d-rec.dat'
  SOURCE='sourcex.dat' # explosive source
```

Model specification `u2d-mod.dat` is used both for ray tracing and finite differences. Source coordinates are stored in file `u2d-src.dat` and the names and coordinates of the receivers are stored in file `u2d-rec.dat`. File `sourcex.dat` contains the data describing the unit seismic moment of an explosive source, because finite difference program `fd2d.for` versions 5.20 or 5.30 considers only the explosive source. The rotation centre used by Klimeš (1996b) cannot be considered.

```
# Source time function (line source)
  KSIG=3   SIGW=2  # Kuepper signal with 2 local maxima
  SIGF=10.         # reference frequency
  SIGA=1.          # maximum amplitude
  DER=0.5          # halfth derivative for the point source
```

We replaced the former Gabor signal by the Küpper (known also as Müller) signal with 2 local maxima because finite difference program `fd2d.for` does not support another source time function now. The reference frequency has been decreased from 100 Hz to 10 Hz because of finite differences, as discussed later on.

We wish to approximate the point source by the line source excited by the Küpper signal in 2-D program `fd2d.for`. We thus take the halfth derivative of the Küpper signal as the source time function describing the time dependence of the seismic moment of the explosive point source. Then the point and line sources will generate wavelets of the same form in the far–field approximation.

```
# Data to control seismogram plotting (program SP)
  KODESP=3  SPCHRH=0.23
  SPTMIN=0.300  SPTMAX=1.000   SPTLEN=11.20  SPTDIV=7  SPTSUB=1
  SPXMIN=0.136  SPXMAX=-1.241  SPXLEN=13.77  SPXDIV=9  SPXSUB=3
  NORMSP=1  SPAMP=-15.00  # amplitude scaling for point source in 3-D
  SPHILI='u2d-hil.dat'  SPHIWI=0.51
```

The seismograms are plotted between times SPTMIN=0.300 and SPTMAX=1.000 (in seconds), with the length of the time axis of SPTLEN=11.2 (in cm), etc. The amplitude scale corresponds to a point source in 3-D.

To be able to plot also the 2-D FD seismograms, corresponding to the line source, in a comparable scale, we choose the reference distance SPDIST for the amplitude power scaling during the plotting equal to the hypocentral distance (in a homogeneous medium), at which the wavefields generated by the point and line sources have the same amplitudes.

```
# Hypocentral distance where the point and line sources have comparable
# amplitudes (for amplitude power scaling roughly converting line
# source to point source)
  SPDIST=0.07472   # SPDIST=1/(2*pi*VP), VP=2.13
```

When plotting, we will use no amplitude power scaling for the point source but the power scaling, proportional to the hypocentral distance powered to $-\frac{1}{2}$, for the line source. In this way, seismograms corresponding to the point and line sources will coincide in the neighbourhood of the source point. They, of course, may considerably differ for reflected waves and at large distances in a heterogeneous medium.

5.2.1 Ray–theory synthetic seismograms

The data specific to the ray method do not require many comments.

```
# Ray synthetic seismograms (point source)
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Data for ray tracing
  DCRT='u2d-crt.dat'
  CODE='u2d-crt.dat'
  RPAR='u2d-rpa.dat'
  WRIT='u2d-wri.dat'
  CRTLOG='u2d-log.out'
  INIPAR=2  # ray parameters: geographic-like spherical coordinates
```

The selection of 22 elementary waves for ray tracing has been performed according to Cormier & Mellen (1984). The codes of the elementary waves used for 2-D two–point ray tracing are specified in the second part of data file `u2d-crt.dat`. Note that ray tracing program `crt.for` reads the input parameters from a history file since version 5.30.

```
# No attenuation for comparison with FD
  TIMUL=0
```

Since the finite differences do not consider attenuation, we discard the attenuation from the ray computation by multiplying the imaginary part of travel time by zero.

```
# Data describing the filtration of the source time function
  FMIN=0.  FLOW=1.  FHIGH=55.  FMAX=60. # (cosine filter)

# Time step and time interval for the Fast Fourier Transform
  DT=0.001  NFFT=1024

# GSE file with synthetic seismograms
  SS='u2d-ss.gse'

# Data to control optional plotting of program SS
  SIGPLOT='u2d-sig.ps'  MPTS=640
```

Parameters `FMIN`, `FLOW`, `FHIGH`, `FMAX` specify the cosine window to filter the source time function in order to remove low or negative frequencies for the ray method. Since finite difference program `fd2d.for` does not filter the signal, the filtering should not distort the signal. The time step and time interval for the Fast Fourier Transform should always be specified. However, `DT` could be chosen considerably longer in this case.

```
# Programs to execute
  crt:     'u2d-fd.h' /       # two-point ray tracing
                              # (do not link 'crt.for' with 'scropc.for'!)
  green:   'u2d-fd.h' /       # Green function
  greenss: 'u2d-fd.h' /       # response function
  ss:      'u2d-fd.h' /       # synthetic seismograms
```

Programs crt.for, green.for, greenss.for and ss.for take the data from history file u2d-fd.h. When executing the history file, only data preceding these lines are submitted to the programs. Plotting the seismograms is located at the end of the history file.

5.2.2 Finite–difference synthetic seismograms

Finite differences are often limited by the memory requirements or by the computational costs. This is also the case of the following grid dimensions:

```
# FD synthetic seismograms (line source)
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Grid dimensions, FD time steps, snapshots
  N1=316     N3=646     NTFD=2501     N4=101
  D1=0.003   D3=-.003   DTFD=0.0004   D4=0.010
  O1=0.315   O3=0.786                 O4=0.000
```

Dimension MRAM of array RAM in file ram.inc must be

$$\mathtt{MRAM} \geq 19\,N_1 N_3 + 10\,N_1 + 12\,N_3 + (2\,\mathtt{NTFD} + 5)\max(1, \mathtt{NREC}) \quad , \tag{20}$$

where NTFD is the number of time levels and NREC is the number of receivers, here $\mathtt{NREC} = 22$. We wish to keep the value of $\mathtt{MRAM} = 4\,000\,000$, which is set in the distributed version of ram.inc and should be suitable for 16MB of available RAM. The top boundary of the finite–difference grid is a free earth surface in program fd2d.for. Since we considered no free surface in the ray–theory calculations and wish to have the same model, the reflections from the top of the grid should arrive at the uppermost receiver, situated 0.017 km below the source, at the least, after the P wave reflected from the horizontal interface 0.610 km below the source. We thus wish the top of the grid higher than $d_3' = 0.610\,\mathrm{km} - 0.017\,\mathrm{km} = 0.593\,\mathrm{km}$ above the source. The horizontal source–receiver distance is $d_1 = 0.823\,\mathrm{km}$ and the lowermost receiver is situated $d_3'' = 1.088\,\mathrm{km}$ below the source. The grid thus should cover region d_1 times $d_3 = d_3' + d_3'' = 1.681\,\mathrm{km}$. For grid interval D, it requires $N_1 = d_1/D + 41$ times $N_3 = d_3/D + 21$ gridpoints, where 20 gridpoints are reserved for the dumpers at the left–hand, bottom and right–hand grid margins. The grid interval then has to be longer than 0.0026 km. We choose $D = 0.003\,\mathrm{km}$, see parameters D1 and D3. Note that D3 is negative because the grid origin must be situated at the earth surface and the x_3 axis is pointing upwards in this model.

The time step must satisfy stability condition

$$\mathtt{DTFD} < \frac{D}{1.65 v_{\max}} \quad , \tag{21}$$

where $v_{\max}$ is the maximum P–wave velocity in the model, here $v_{\max} = 3.960\,\mathrm{km\,s^{-1}}$, which results in $\mathtt{DTFD} < 0.000459\,\mathrm{s}$. We choose $\mathtt{DTFD} = 0.0004\,\mathrm{s}$. The number of time levels from 0 s to 1 s is then $\mathtt{NTFD} = 2\,501$.

Then

$$19\,N_1 N_3 + 10\,N_1 + 12\,N_3 \leq 4\,000\,000 - 110\,154 \quad , \tag{22}$$

i.e.,

$$N_3 \leq (3\,889\,846 - 10\,N_1)/(19\,N_1 + 12) \quad . \tag{23}$$

We choose the minimum horizontal number of gridpoints $N_1 = 316$, the grid begins 0.062 km to the left from the source and ends 0.060 km to the right from the receivers. The maximum vertical number of gridpoints is then $N_3 = 646$. To position the bottom of the grid 0.061 km below the lowermost receiver, we have put the grid origin 0.786 km above the source.

The relative wavefield error due to the grid dispersion of the second–order finite differences, accumulated along distance s in a homogeneous medium with $v_P^2 \leq 3v_S^2$, is (Klimeš 1996a, equations 24, 33, 108, 110)

$$\Delta_u \approx \left(\frac{2\pi f}{v}\right)^3 \frac{D^2\, s}{24} \quad . \tag{24}$$

The considerable part of the error is accumulated by the S wave travelling from the interface 0.610 km below the source to the uppermost receiver. The distance is about $s \approx 0.6$ km and the velocity is $2.13\,\mathrm{km\,s^{-1}}/\sqrt{3}$. Then the relative error of the harmonic S wave is roughly

$$\Delta_u \approx (f\ 0.03\,\mathrm{s})^3 \quad . \tag{25}$$

It is obvious that we cannot use 100 Hz source time function as Klimeš (1996b) did for the ray method. Choosing the Küpper signal of reference frequency `SIGF` = 10 Hz, we probably keep the relative error below something like 4%.

```
# GSE file with synthetic seismograms
  SS='u2d-fd.gse'

# Form of files with effective elastic parameters and snapshots
  FORM='formatted'

# Files with snapshots (not generated by default)
 #SNAP1='u2d-fd1.out'
 #SNAP3='u2d-fd3.out'
```

If the names of the output files with snapshots are not specified, the snapshots are not generated. Uncomment these lines if you wish to store `N4` snapshots on your disk. The default form is `'formatted'`. Note that programs of package FORMS, processing the snapshots, work just with formatted files in version 5.30.

```
# Files with effective elastic parameters
  A11='a11.out'  B11='b11.out'  C11='c11.out'
  A13='a13.out'  B13='b13.out'  C13='c13.out'
  A31='a31.out'  B31='b31.out'  C31='c31.out'
  A33='a33.out'  B33='b33.out'  C33='c33.out'
  DEN='den.out'
```

The files with "effective" elastic parameters are written by program `modfd.for` and read by finite–difference program `fd2d.for`.

```
# Programs to execute
  modfd:   'u2d-fd.h' /          # gridded elastic parameters
  fd2d:    'u2d-fd.h' /          # FD synthetic seismograms
```

Since the "nonreflecting" boundary conditions of finite differences may, to some extent, reflect the waves, it is highly recommended to change the finite–difference grid roughly by a quarter of the prevailing wavelength and to run the finite differences once again. The reflections from the boundaries are then shifted roughly by half the prevailing period and are clearly visible in the common plots of the seismograms from both the computations. Moreover, in this example we have also very strong unwanted reflections from the free surface at the top of the finite–difference grid.

The reference P–wave wavelength is 71 to 132 grid intervals, S–wave wavelength is 41 to 76 grid intervals. We enlarge the grid by 20 grid intervals at all 4 sides.

```
# FD synthetic seismograms on a larger grid to test boundary conditions
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Increasing the box by 20*D1=0.060 at all 4 sides
  N1=356     N3=686
  O1=0.255   O3=0.846

# GSE file with synthetic seismograms
  SS='u2d-fda.gse'

# No snapshots for this test
  SNAP1=' '  SNAP3=' '
```

Parameters `SNAP1` and `SNAP3` are cleared to avoid the snapshot generation even if they were enabled for the previous FD computation.

```
# Programs to execute
 #modfd:   'u2d-fd.h' /          # gridded elastic parameters
 #fd2d:    'u2d-fd.h' /          # FD synthetic seismograms
# Do not forget to increase default value of MRAM in ram.inc!
```

Execution of the programs is disabled by commenting the lines because this calculation cannot be carried out with the distributed version 5.30 of `ram.inc`. You may increase `MRAM` in `ram.inc` accordingly, compile program `fd2d.for`, and enable this computation.

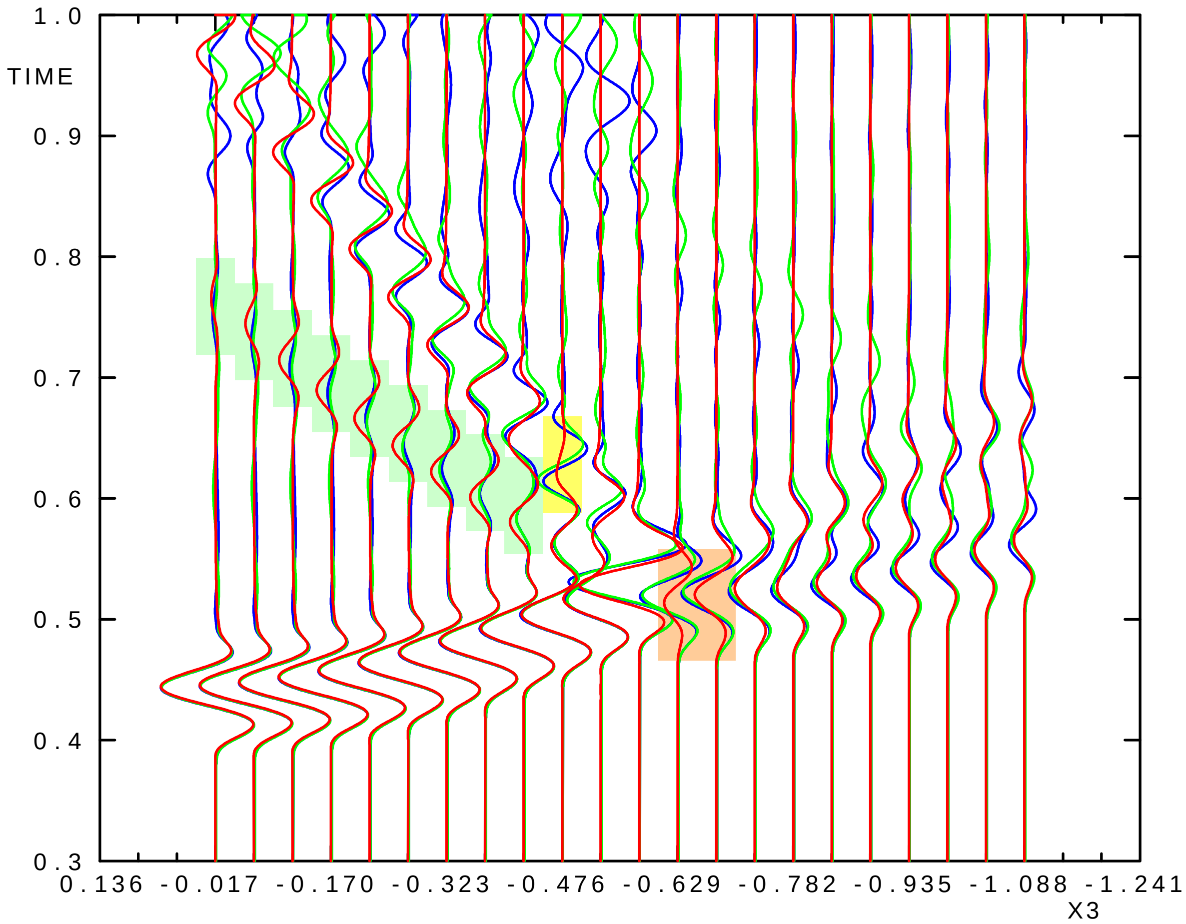

Figure 7. Horizontal component of the synthetic seismograms in the UNCONFORMITY model. 2-D finite differences are *blue* and *green*, 3-D ray method is *red*. For a more detailed description refer to the text.

5.2.3 Comparison of the synthetic seismograms

```
# Plotting the seismograms
# ~~~~~~~~~~~~~~~~~~~~~~~~
  SS=' '
  SS='u2d-fd.gse'   KOLOR=4   SPOWER=-.5   # FD seismograms (blue)
 #SS1='u2d-fda.gse' KOLOR1=3  SPOWER1=-.5  # test FD seismograms (green)
  SS2='u2d-ss.gse'  KOLOR2=2  SPOWER2=0.0  # ray-theory seismograms (red)
  SP1='u2d-1.ps'  SP2=' '  SP3='u2d-3.ps'
  sp:      'u2d-fd.h' /          # seismogram plotting
```

We plot the calculated synthetic seismograms together, in the colour PostScript plots of the horizontal in–plane (`u2d-1.ps`) and vertical (`u2d-3.ps`) components. The transversal component is zero. The colour indices correspond to the distributed version of file `calcops.rgb`. Parameter `SS` is cleared to be sure that commenting off the plotting of the first FD seismograms stored in file `u2d-fd.gse` cannot cause problems. Plotting the second FD seismograms stored in file `u2d-fda.gse` is disabled. Enable the plotting if you are calculating them.

The calculated seismograms are shown in Figures 7 and 8, where the reflections from the nonreflecting boundaries can clearly be identified. The blue reflections (the

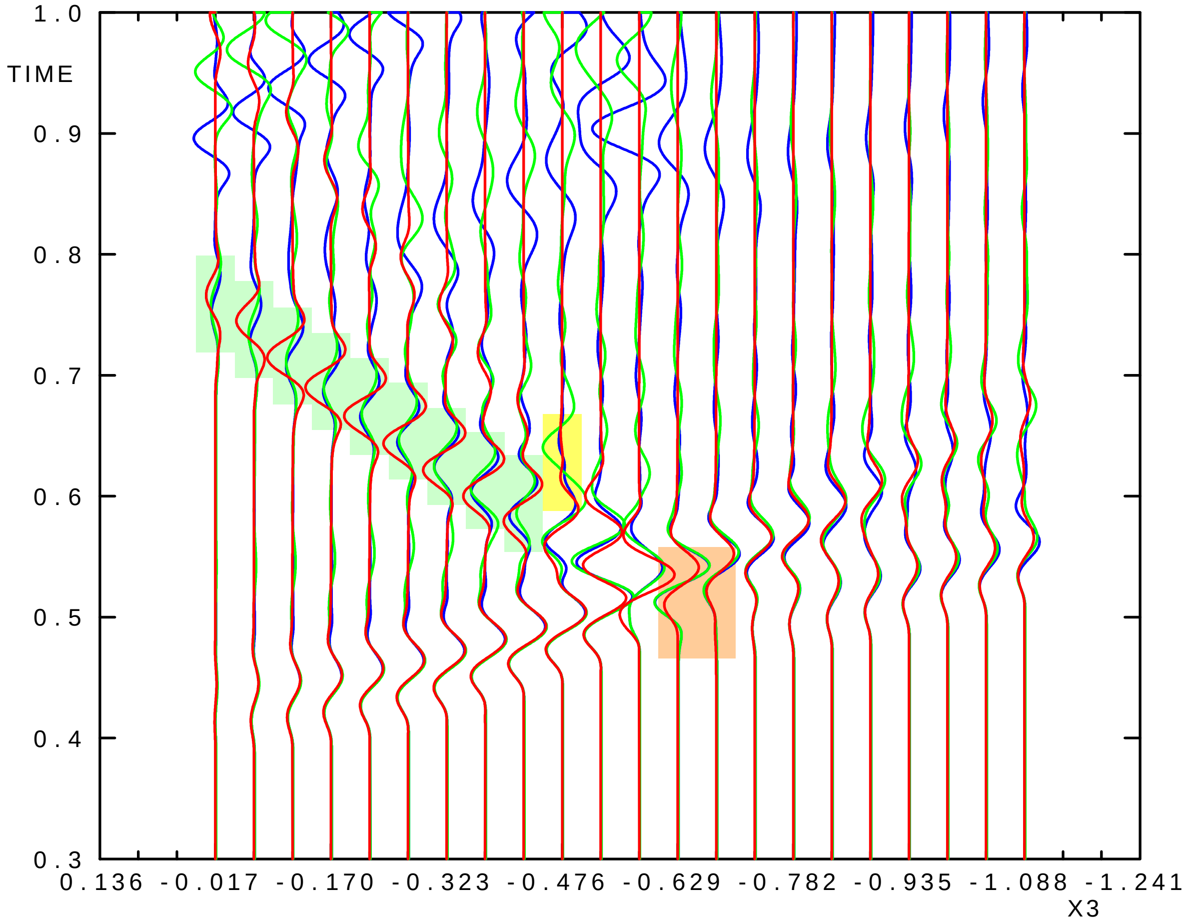

Figure 8. Vertical component of the synthetic seismograms in the UNCONFORMITY model. 2-D finite differences are *blue* and *green*, 3-D ray–matrix method is *red*. For a more detailed description refer to the text.

first FD calculation) come before the green ones (the second FD calculation with the enlarged grid). The 2-D finite–difference seismograms are overlaid by the red 3-D ray–theory synthetic seismograms.

Where blue and green seismograms coincide, the finite–difference seismograms are not distorted by reflections from the non–reflecting boundaries.

The most pronounced drawbacks of the ray–theory are highlighted in pastel colours. From the left, the ghost green indicates the increased ray amplitudes in the vicinity of the P wave critical angle, which is somewhere around the second receiver from the left. The light yellow colour indicates the gap in the reflected P–S wave due to the missing diffracted wave which should fill the ray–theory shadow zone between reflections from the two interfaces. The light orange (sand) indicates the distorted amplitudes just below the interface at the depth of 0.610 km due to the missing first–order ray–theory approximations of refracted waves.

Except for the above mentioned problems, the finite–difference and ray–theory seismograms are in a good agreement, especially if we consider the difference between the point and line sources.

6 Problems

Up to date test computations showed that program `fd2d.for` should not be used to calculate elastic wavefields in fluids. Spurious waves, other than P waves, and instabilities have been observed in fluids. Another problem is that the absorbing boundary conditions and dumpers do not work perfectly. The reflections from the boundaries may be identified in the seismograms by calculating the seismograms with two different positions of the boundaries and comparing the results (see, e.g., Section 5.2 of this paper or Klimeš 1998).

Acknowledgements

We are indebted to Jiří Zahradník, the author of the finite–difference code, for his kind and patient assistance in preparing package FD and for the suggestions enabling to improve this paper.

References

Červený, V., Klimeš, L. & Pšenčík, I. (1988): Complete seismic–ray tracing in three–dimensional structures. In: Doornbos, D.J. (ed.), Seismological Algorithms, pp. 89–168, Academic Press, New York.

Cormier, V.F. & Mellen, M.H. (1984): Application of asymptotic ray theory to vertical seismic profiling. In: Toksöz, M.N. & Stewart, R.R. (eds.), Vertical Seismic Profiling: Advanced Concepts, pp. 28–44, Geophysical Press, London.

Klimeš, L. (1996a): Accuracy of finite differences in smooth media. PAGEOPH, **148**, 39–76.

Klimeš, L. (1996b): Synthetic seismograms in 2-D model UNCONFORMITY. In: Seismic Waves in Complex 3-D Structures, Report 4, pp. 77–89, Dep. Geophys., Charles Univ., Prague.

Klimeš, L. (1998): Comparison of ray–matrix and finite–difference methods in a simple 1-D model. In: Seismic Waves in Complex 3-D Structures, Report 7, Dep. Geophys., Charles Univ., Prague, pp. 169–180.

Opršal, I. & Zahradník, J. (1998): Stable finite–difference modelling of sedimentary basins. In: Seismic Waves in Complex 3-D Structures, Report 7, Dep. Geophys., Charles Univ., Prague, pp. 349–374.

Opršal, I. & Zahradník, J. (1999): From unstable to stable seismic modeling by finite–difference method. J. Phys. and Chem. of the Earth, **24**(part A), 247–252.

Zahradník, J. (1995): Simple elastic finite-difference scheme. Bull. Seism. Soc. Am., **85**, 1879–1887.

Zahradník, J. & Priolo, E. (1995): Heterogeneous formulations of elastodynamic equations and finite–difference schemes. Geophys. J. Int., **120**, 663–676.