

Duplex Waves Migration Processing Guide

Please, also see *Introduction to Duplex Wave Migration -Clients Guide-*

Contents

AboutDWM program.....	1
Data Preparation and QC	2
DWM Computation chart	3
ConfigDWM	4
Tab Model.....	5
Tab Parameters	7
Group(1)	7
Group(2)	8
Tab Input	8
Tab Aperture	10
Group(1)	10
Tab Optimization	11
Tab Other.....	12
TabDirectories	13
Task Passport (Task.ini) Format.....	14
Model specification parameters	15
Output specification parameters	15
Aperture specification parameters	17
Migration parameters	20
Optimization parameters	20
SEG-Y parameters.....	21
Debug parameters.....	22
Bottom Surface Format	22
Running Program.....	22

About DWM program

DWM processing requires paralleling due to massive computations associated with the task.

3D DWM application can run parallel computations on Linux or Windows cluster for virtually any conventional 3D survey layout and reasonable apertures and by its intensity and hardware requirements approximately corresponds to conventional PSDM.

The program implements 3D eikonal based depth pre-stack DWM on both Linux and Windows PC clusters based on MPI. MPICH, MPVPAICH, OpenMPI and LAM MPI implementations were tested.

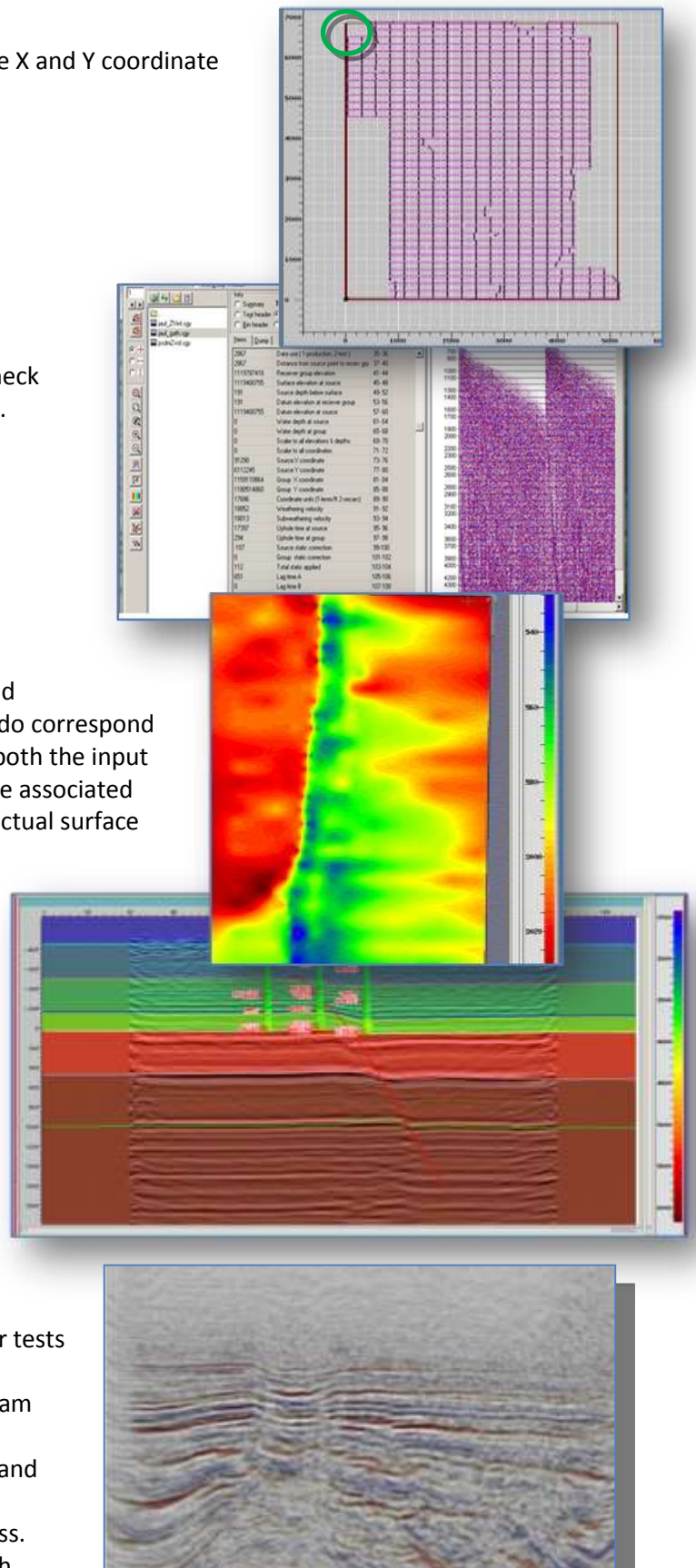
Data Preparation and QC

1. For the area of your project determine X and Y coordinate of its corners.

2. Load the input 3D seismic data and check Input gather headers: coordinates etc.

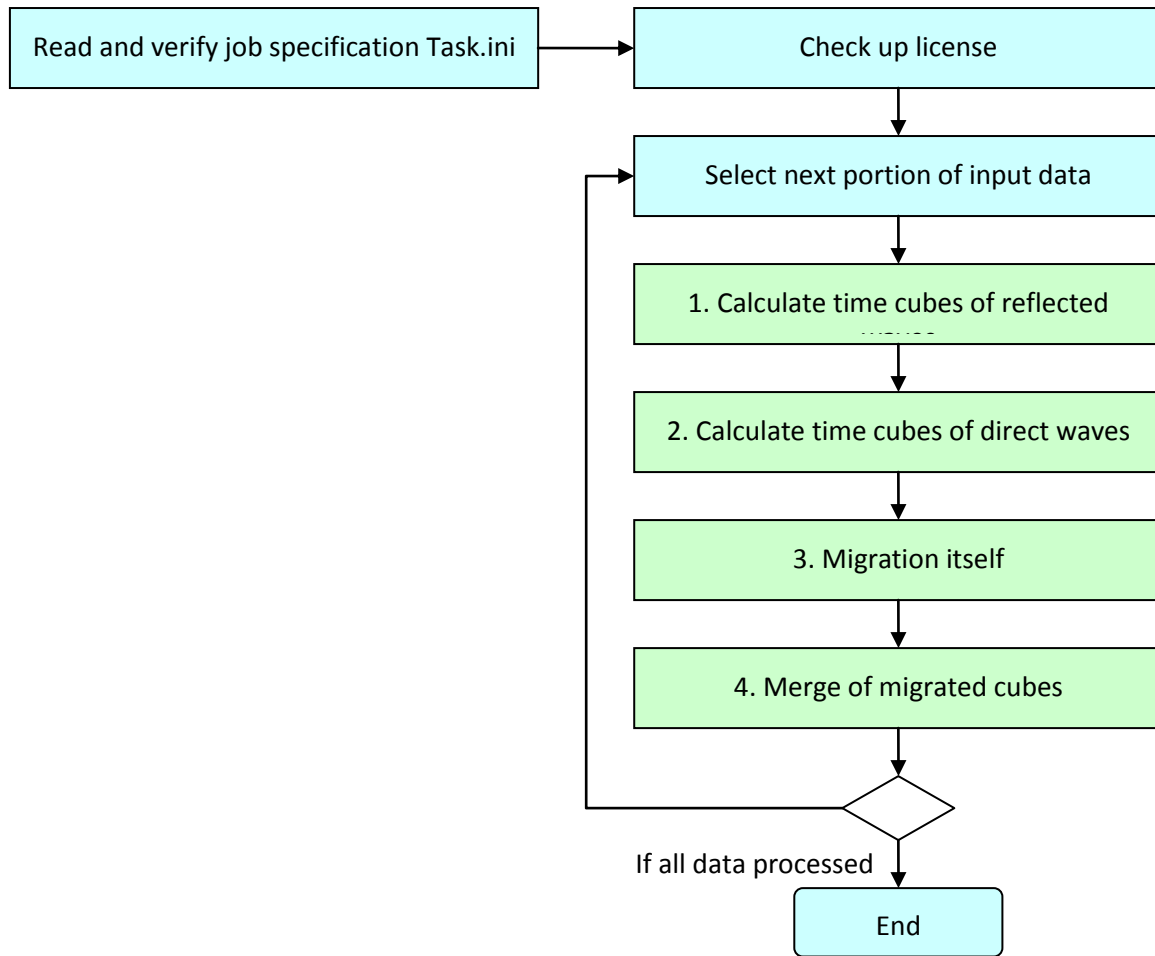
3. Load base boundary (horizon map) and check that the base boundary events do correspond correctly with the images as seen on both the input data and the PSDM data cubes and the associated velocity model. Also ensure that the actual surface elevation, the statics applied, and the processing datum information are well understood so that when the DWM procedure computes it's travel times tables it is referencing the data correctly. This process needs to be done with a great deal of care to make sure that some sort of basic datum referencing error does not contaminate the results.

4. Data QC process may involve applying a basic NMO and stack process to check the input data. Other tests such as targeted Kirchhoff PSDM (procedure supplied with DWM program specifically for data QC) as deemed necessary to ensure that the 3D data and DWM task passport (script) has been properly prepped for the DWM process. PSDM results are over-layered on each other



Previous and QC PSDM results are over-layered on each other to ensure proper data and DWM processing passport preparation

DWM Computation chart



Comments

- ✓ Green rectangles correspond to the main DWM steps executed in parallel
- ✓ Because of hidden in CPU optimization (like cache, pipes, advance calculations, etc) and other technical reasons (e.g., virtual memory mapping) even simple computation times vary in wide range. For typical pure calculation jobs time difference between the shortest and the longest process or thread may reach 50-200%. Data I/O makes such inequity even stronger.
- ✓ For data QC may be used conform PSDM procedure. It is uniformly with DWM procedure treating source data and processing options. Both 3D DWM and *conform PSDM* use two eikonal computed Time Cubes. One time cube is calculated from source points (SP), another – from receiver points (RP). They both are used as input data for migration by traces. They are stored and reused during migration procedure, because it is experimentally proven fact that for the majority of cluster computers such eikonal calculations take longer time then input of ready time cubes from files.

As result the general scheme of the 3D DWM program algorithm consists of 4 steps:

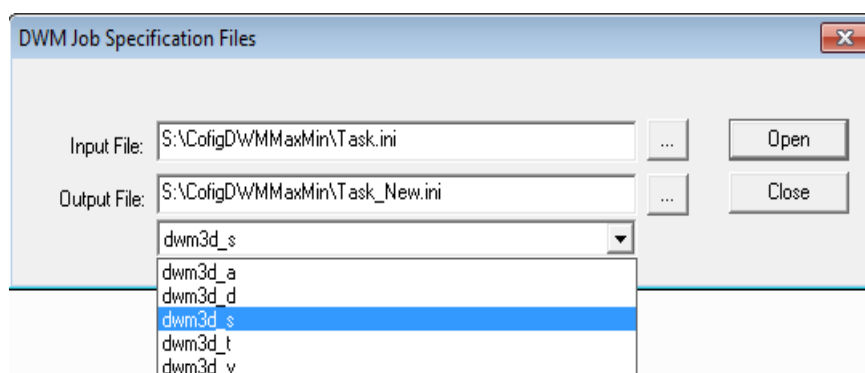
1. Eikonal calculation of reflected wave time cubes (e.g. for *SPs*). Direct time cubes from the same positions are stored if they can be reused at the step 2 (when they coincide with some *RPs*).
2. Eikonal calculation of direct wave time cubes (for *RPs*).
3. Migration with those time cubes.
4. Merge of the migrated cubes.

PSDM algorithm consists of the steps 2, 3 and 4 only.

- ✓ Bigger time cubes mean both longer eikonal calculations and longer I/O operations. To optimize the program size of the time cubes is minimized. To minimize size of an SP time cube the programs account reallocation of its RPs under restriction conditions on their coordinates, inline/crossline numbers and SP/RP offsets defined in the job specification. Similarly they account location of all SP working on RP located in the same position (if such pairs exist).
- ✓ This is why the programs first review all input file trace headers and create relatively small array of coordinates and interfile offsets (for quick random access to the traces). These header arrays are stored in separate files and always reused. The arrays are filtered in memory according to the restriction conditions. Then the array is quickly sorted in memory to allow efficient sequence of processing regardless real gather type.
- ✓ Each trace is loaded one time. Input of traces takes much less time in relation to input of time cubes and migration calculations. Among the last two ones the most important for big data size is input of time cubes. Model is intensively used especially at two first steps. Output cubes of partial sums (of each worker process) are also important.
- ✓ There are multiple program settings to optimize the I/O by intensive use of local folders instead of shared ones. Time cubes, partial migration cubes and model can be copied in local folders for speedy I/O. Time cubes can be compressed. *Note: decompression requires smaller time than reading which itself is less expensive than re-calculation of the time cube.* Multiple shared folders optimize time for directory operations. Some other options are possible for specific hardware.
- ✓ In case of the huge input data size another possible way is to process such data by parts or use special DWM program variant developed for such case.

ConfigDWM

This program helps to create Task.ini file in Linux using Windows environment and to check coordinates alignment between Velocity Model, picked base-boundary surface, Input Gathers and any supplied Anisotropy Cubes. The program has several options for Task.ini file creation depending on what DWM solution is going to be implemented in any particular case. If we want to use Anisotropy DWM solution we need to activate “dwm3d_a” option from down-pull menu or “dwm3d_s” for Isotropy case.



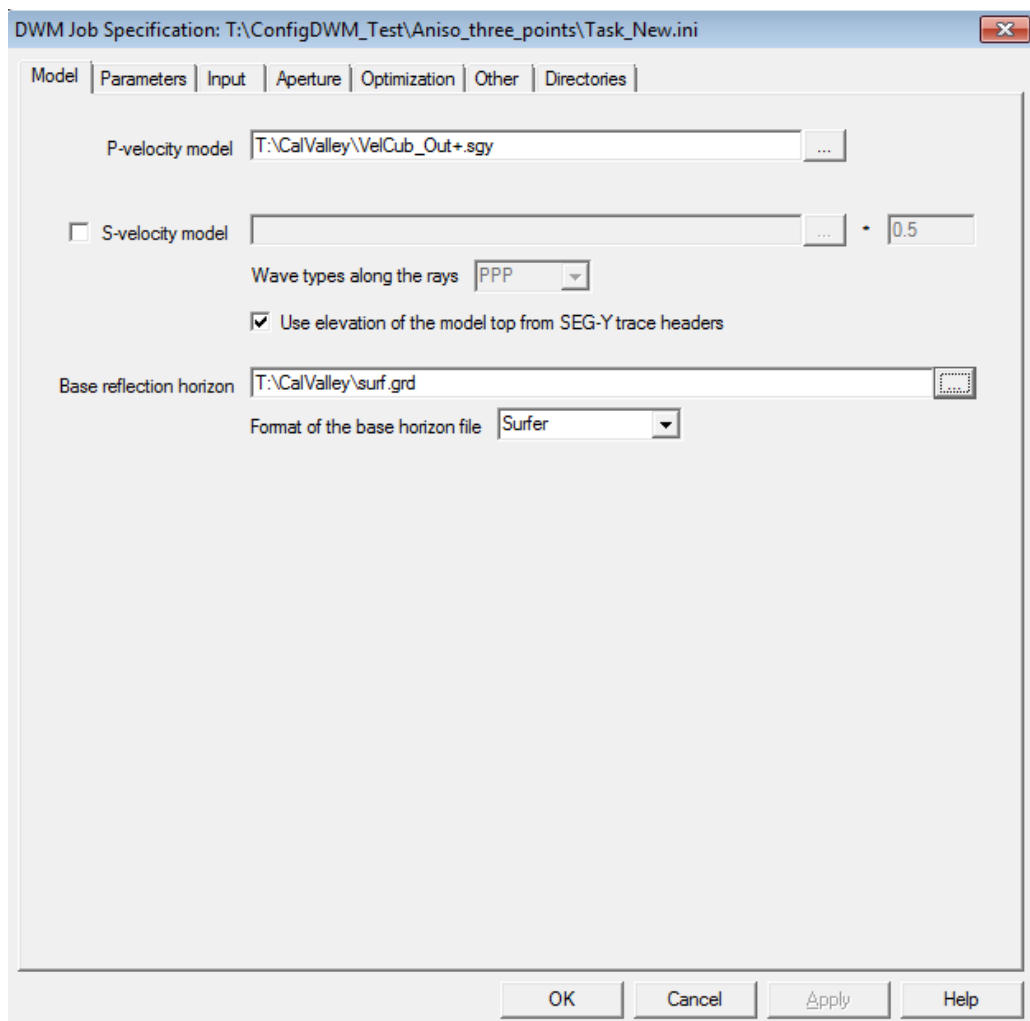
The “dwm3d_*” program execution is controlled by a task authorization. The task authorization is a text file “Task.ini” located in the current work directory. **ConfigDWM.exe** is a Windows dialog-based program which allows to create a new Task.ini file on the base of already existing example or to edit such file.

First, start the program and enter the Input and Output “Task.ini” file paths using buttons “...” to call the standard Windows “Open” dialog and browse through available folders. The paths can be the same in a case of editing the existing job. You may leave one of the paths empty. In such case the file described in non-empty path will be edited.

Then press the button “Open”. The main program dialog appears. It consists of six pages: “Model”, “Parameters”, “Input”, “Aperture”, “Optimization”, “Other” and “Directories”. It’s recommended to fill them in their natural order from left to right (arbitrary editing sequences are enabled as well.)

Tab *Model*

... describes physical parameters of the model including anisotropy parameters, which can be in SGY format or can be described as constant parameters:



1. Enter or select the **P-velocity model** path using the associated button "...". Browsing is useful if the migration data (commonly located on a Linux cluster) are available from the Windows computer. Hereafter you may use both local Windows paths and cluster Linux paths in mixture. They will be automatically converted to the common system using the substitution rules determined in the tab "Directories".
2. Corresponding **shear velocity (S-velocity) model** must be selected too in the case of Converted DWM processing. Its selection enables the choice of "wave type". "PPP" means usual duplex waves, "PSS" means conversion on the first reflection boundary, and "PPS" means conversion on the second one. Other wave types can be interpreted similarly, but they are rarely used.
3. Check the **"Use elevation of the model top from SEG-Y trace headers"** if the model cube does not starts from zero depth.
4. Enter or select the **base reflection horizon** path using the associated button "...". The file represents a strong sub-horizontal reflection boundary by a grid (or other supported by 3D DWM) file. The boundary must be located below the target unknown sub-vertical objects. The grid is in depth scale. Its coordinates and depths must be similarly measured and oriented as the same parameters of the input velocity model.

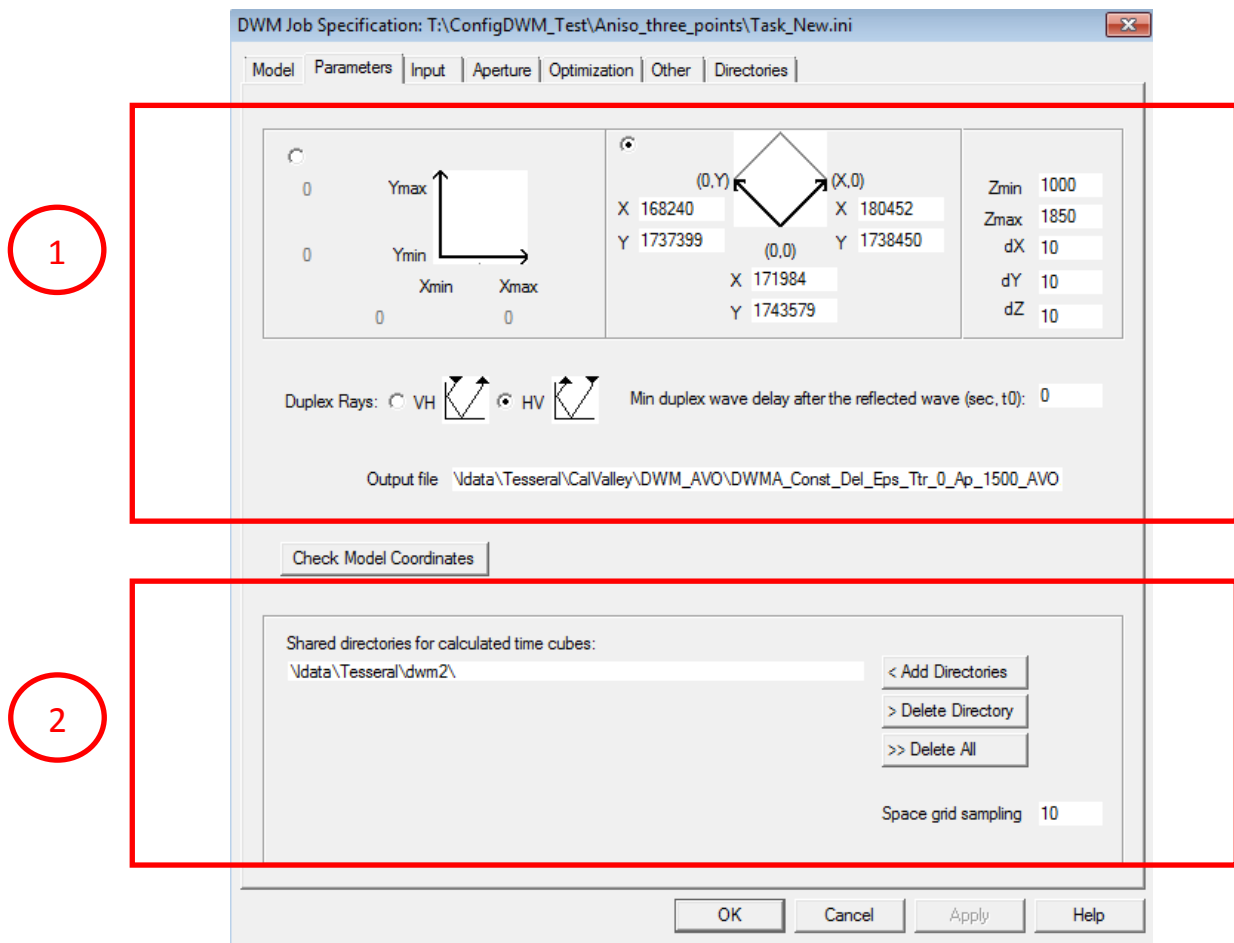
5. *Note: in the case of `dwm3d_a` and `dwm3d_v` program the tab contains also anisotropy parameters:*

The screenshot shows the 'Parameters' tab of the 'DWM Job Specification' dialog box. The title bar indicates the file path: 'T:\ConfigDWM_Test\Aniso_three_points\Task_New.ini'. The 'Model' tab is selected, showing fields for 'P-velocity model' and 'S-velocity model', both pointing to 'T:\CalValley\VelCub_Out+.sgy'. A checkbox 'Use elevation of the model top from SEG-Y trace headers' is checked. The 'Base reflection horizon' is set to 'T:\CalValley\surf.grd' with a 'Format of the base horizon file' dropdown set to 'Surfer'. The 'Anisotropy' section contains several radio button options: 'File of Epsilon', 'Epsilon' (selected), 'Delta model', 'Delta' (selected), 'File of Dip', 'Dip, deg.' (selected), 'File of Azimuth', and 'Azimuth, deg.' (selected). Each selected option has a corresponding input field with a value (0.05, 0.1, 0, 0). At the bottom are 'OK', 'Cancel', 'Apply', and 'Help' buttons.

Parameter	Value
P-velocity model	T:\CalValley\VelCub_Out+.sgy
S-velocity model	T:\CalValley\VelCub_Out+.sgy
Use elevation of the model top from SEG-Y trace headers	Checked
Base reflection horizon	T:\CalValley\surf.grd
Format of the base horizon file	Surfer
Anisotropy: Epsilon	0.05
Anisotropy: Delta	0.1
Anisotropy: Dip, deg.	0
Anisotropy: Azimuth, deg.	0

Anisotropy. Please select the corresponding anisotropy files in a way described above for velocity files.

Tab Parameters



Group(1)

... of parameters describes the migration area, type and output file.

1. Choose the suitable coordinate specification. You may describe the migration (output) area either by coordinate intervals or by coordinates of 3 corner points (left radio-button). Don't forget to **enter output depth range and sampling of the output cube** (dx, dy, dz).
2. Select type of DWM by graphic selection of duplex rays path (VH or HV). The difference here is the duplex wave reflection sequence: first from the horizontal base boundary and then from the unknown sub-vertical boundary (**HV**) or in the opposite direction (**VH**).
3. "**Min duplex wave delay after the reflection wave**" is used to suppress low frequency noise from the much more powerful one-time reflected waves. The value should be approximately equal to T_0 width of the one-time reflected wave together with its ghost signals.
4. Don't forget to specify **the migration output file**.
5. If the model files are accessible from the Windows computer you can check up correspondence of the model coordinates to the specified output area. Use the button "**Check model coordinates**" to run the test. (It can take long enough time for big model or slow connection to the model storage.) The procedure checks up **both the velocity model and base boundary**.
 - The message "**... test succeeded**" means the model completely covers the output region.

- The message “...---FILE NOT FOUND” commonly means the paths aren’t accessible to the model files from the Windows computer. Checkup is impossible in such case. But the paths can be correct for the cluster nodes. So the message does not mean an error.
- If the program reports *partial or absent correspondence between the model and the migration area* you need to verify your data. You may use Tesseral Pro for this purpose.

Group(2)

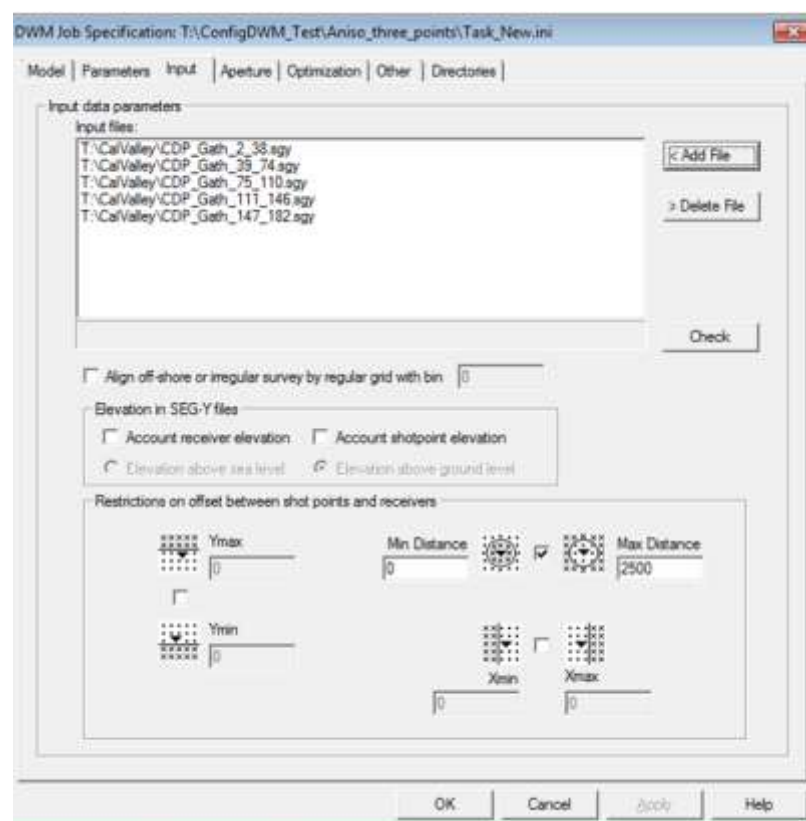
... of parameters describes the path(s) to the shared folder(s) used to store intermediate files of time cubes.

The program needs at least one shared folder to run. Use multiple shared folders to improve performance by load balancing and decreasing directory operations. (The program generates and uses thousands time cube files. The number is usually too big to keep them all in a single directory because of directory operation overheads.)

The “space grid sampling” parameter describes the sampling of temporary grids used for the time cubes calculations by eikonal algorithm. Common values for this parameter are 5-10 m. By increasing this number you will speed-up calculations but the result’s precision will be worse. *The dx, dy and dz values are supposed to be proportional to the “space grid sampling”.*

Tab Input

...describes input files and trace selection for the DWM:



1. Specify the **input dataset files** using the button “Add File”.
2. Specify the **restrictions on shot-receiver offsets for the migrated traces**. In contrast to PSDM the DWM algorithm does not benefit from wide offsets. From performance viewpoint may be better to restrict the migration by the traces with offsets, e.g. not bigger then 1.5-2 km.
 - a. The **offset restrictions** Xmin/Xmax and Ymin/Ymax account sign. Thus they enable specification of “one-side” DWM systems useful for some cases.

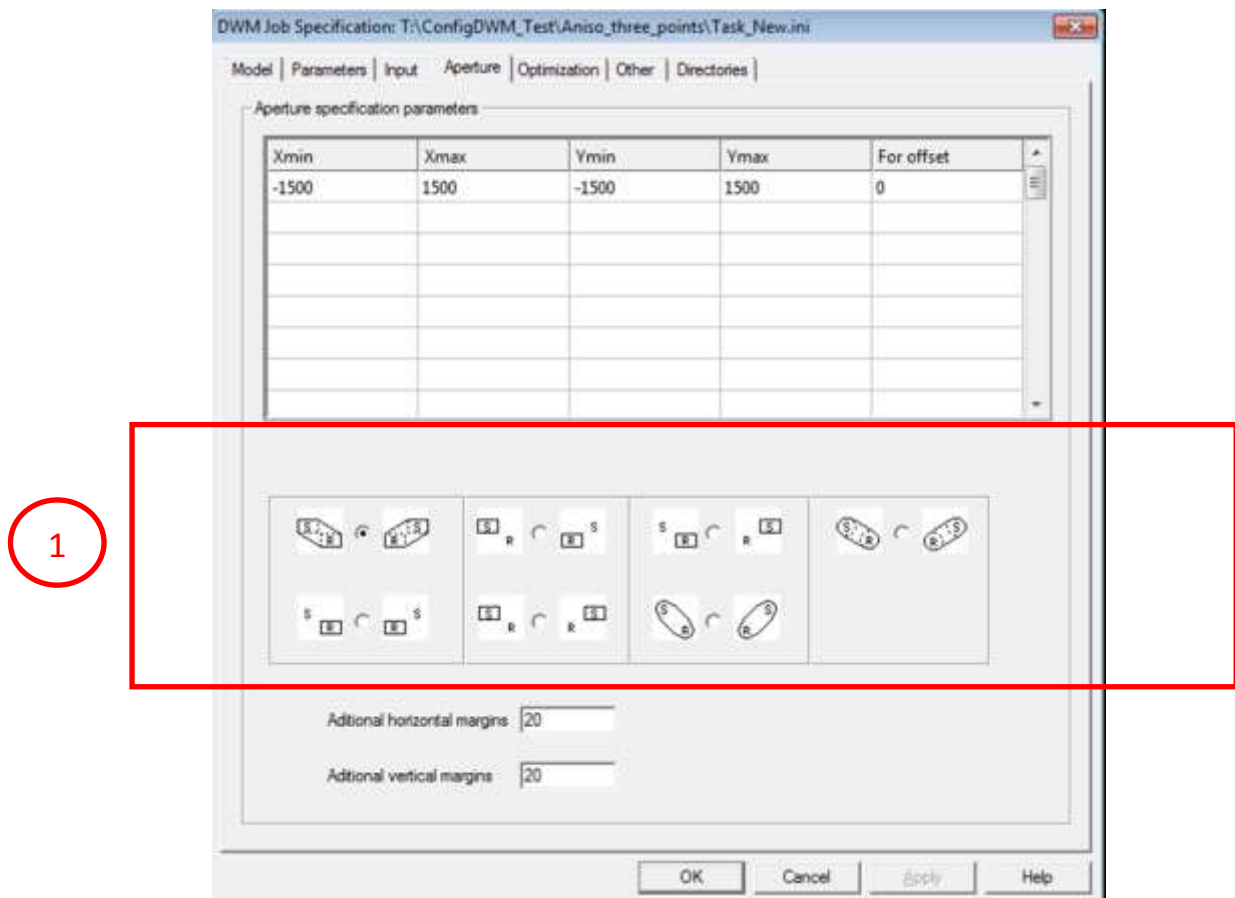
- b. In the ***symmetric case*** X_{max} is positive and $X_{min}=-X_{max}$. The same is right for Y_{min}/Y_{max} .
In the case of “rotated” migration area (specified by 3 corner points) X means in-line and Y means cross-line axes.
3. **Check up the input data** by the button “Check”. The operation is similar to “Check model coordinates” from the “Parameters” tab but it has three specific features:
- The first, it tests the input data, not the model.
 - The second, it tests whether any seismogram traces fit to the output area instead of verification of the output area fitting in the region covered by model data. Here we don’t need to have anything covered, and a partial correspondence commonly means correct specification.
 - The last, you should be ready to wait much longer than for the model verification because of much bigger files to be processed.

Some additional parameters of the tab may be useful in special cases:

1. The “***Align off-shore or irregular survey by regular grid with bin***” parameter is used for off-shore or irregular on-shore surveys for acceleration purpose. The bigger the value the higher performance is. But the quality is proportionally decreased. Typical values of the parameter are 10, 20, 50 and 100 m.
2. Both “***Account receiver elevation***” and “***Account shotpoint elevation***” must be checked:
 - if the input dataset has not been statically corrected for the relief. Select “*Elevation above sea level*” in this case, or
 - if the input dataset has been statically corrected for the relief and aligned on a level different from 0. Select “*Elevation above ground level*” in this case.

Tab Aperture

... describes DWM apertures:



Group(1)

The DWM apertures are quite different from the conventional PSDM/PSTM apertures. Geometrically the DWM apertures look like cylinders when PSDM/PSTM apertures commonly resemble cones. The DWM apertures would be better described by their form in plan if you look at them down from above.

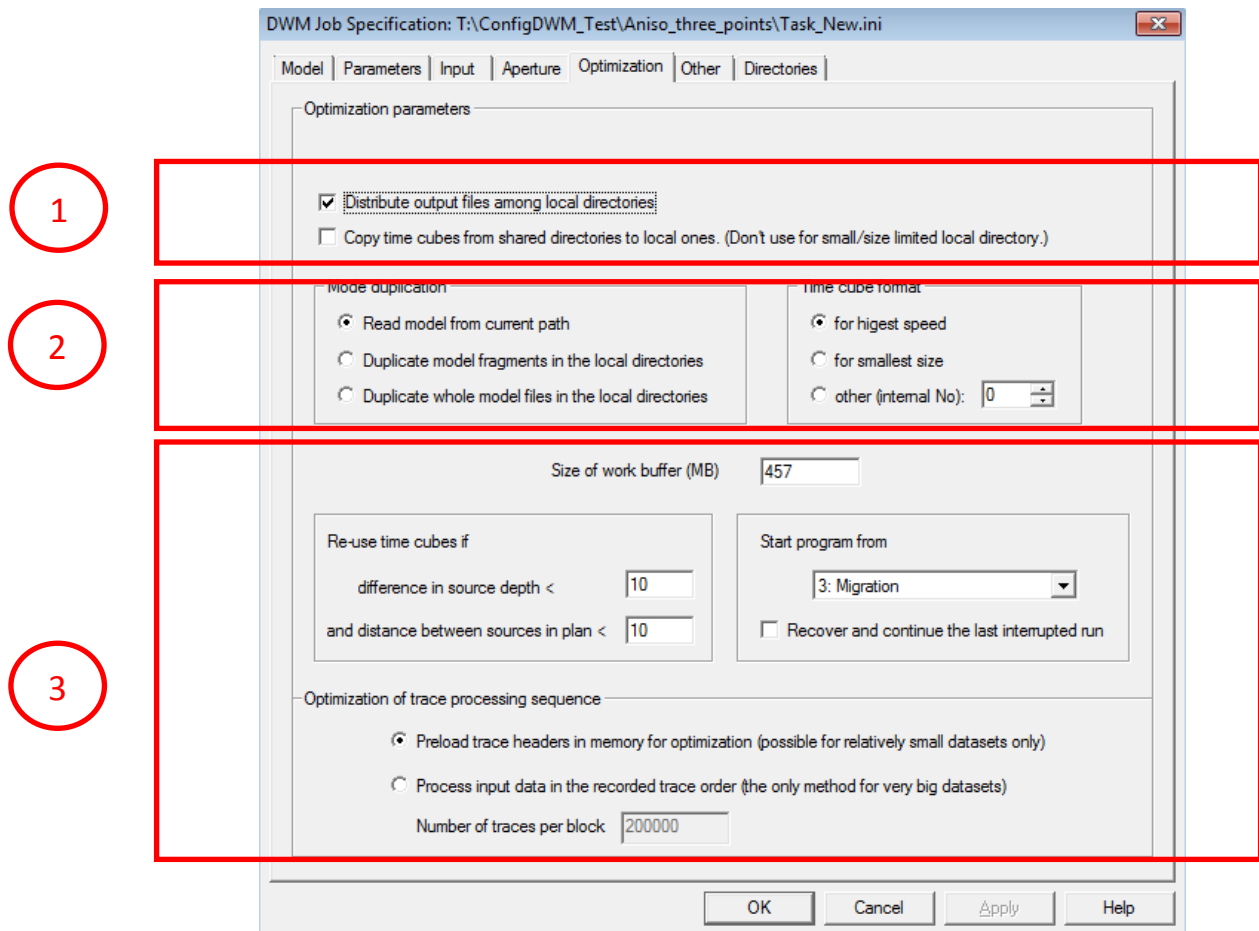
Regardless their geometry the *DWM apertures are formed on the base of a rectangle*. The rectangle edges are positioned relative to either shot or receiver which coordinates are supposed to be (0,0) in the aperture specification dialog. *In the case of "rotated" migration area (specified by 3 corner points) X means in-line and Y means cross-line axes as previously.*

Different basic rectangles can be specified for different offsets. In the case of multiple rows in the table the basic rectangle for other offsets are interpolated.

Both "**Additional horizontal margins**" and "**Additional vertical margins**" improve the time estimation precision near the time cube boundaries. Values of 200-500 m are recommended for the first parameter and of 20-50 m for the second one.

Tab Optimization

...describes the most important additional parameters:



1. Distribution of **intermediate Output Files** is highly recommended for performance reasons. In this case the files are stored in the work directory instead of the current work directory. The same can be done for time cubes (which are usually generated and stored in the shared directory from the tab “Parameters”). However the setting “Copy time cubes...” requires big local folders and does not always increase performance.

2. **Preliminary model duplication** is recommended as well. Here you have a choice between copying the source model file and extraction of the necessary fragments. The last method is better if the processing area is much smaller than the region covered by the velocity model.

Selection of higher compressed **time cube format** sometimes increases performance as well. This is typical for the case when insufficient I/O performance creates a bottleneck for the migration. Other formats are used for internal/debug purposes and aren't recommended for end users.

3. **Other optimization parameters** are oriented on experienced users. One may decrease the size of work buffer or select “**Process input data in the recorded trace order...**” if the program fails because of insufficient memory. Unfortunately both methods decrease performance. (The first one slows down time cube calculations, the second one dramatically decelerates migration itself, especially for small size blocks.)

Tab Other

... describes additional parameters:

DWM Job Specification: T:\ConfigDWM_Test\Aniso_three_points\Task_New.ini

Model | Parameters | Input | Aperture | Optimization | **Other** | Directories

Output SEG-Y parameters

☒ Generate the output cube SEG-Y file in PROMAX format Multiply output by 1

☐ The output cube lines are oriented along Y axis. (Else they are oriented along X axis.)

Inline numbers from 1 Crossline numbers from 1

Inline number increment 1 Crossline number increment 1

☐ Debug parameters

Shot points: X = 0 +/- 0 incremented by 0

Y = 0 +/- 0 incremented by 0

Receivers: X = 0 +/- 0 incremented by 0

Y = 0 +/- 0 incremented by 0

Advanced migration parameters

Signal spreading (not used yet) 0

Footprint removal parameter (rarely used) Do nothing

☐ suppress footprints in X direction

☐ suppress footprints in Y direction

OK Cancel Apply Help

“Generate the output cube SEG-Y file in PROMAX format” may improve import of the DWM output file in 3rd party processing software. Here you can additionally specify **“The output cube lines are oriented along Y axis”** (by default the output cube in-line numbers correspond to X). The parameter **“Multiply output by...”** may be applied if the input data are recorded in too small or too big value scale, in this way the program normalizes the migrated cube values.

You can additionally control in-line/cross-line numbers in the output SEG-Y trace headers.

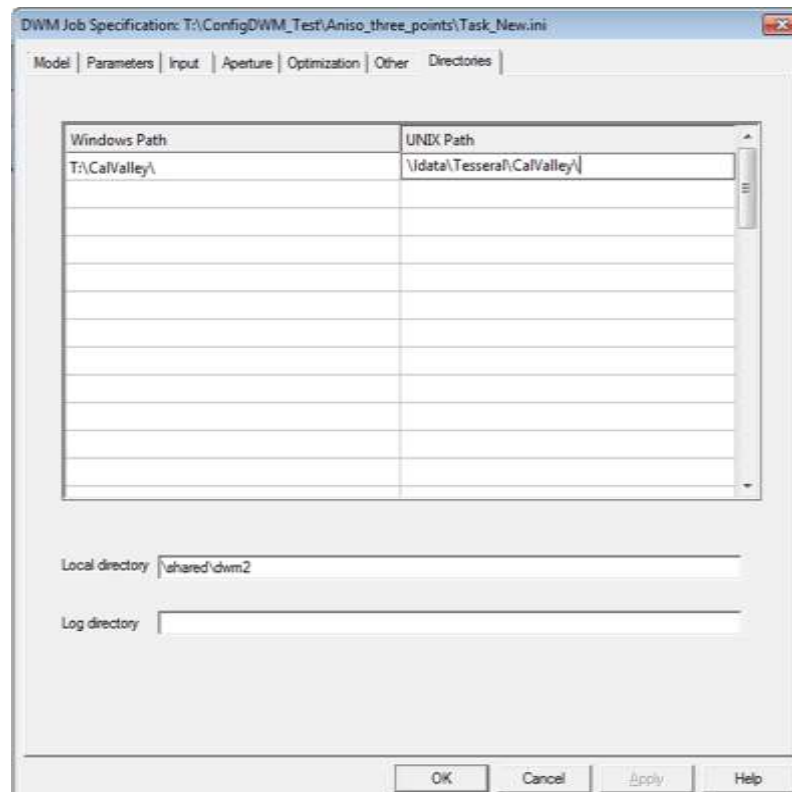
The **debug parameter group** can be used to build one trace response.

The parameters X and Y of **“Shot points”** and **“Receivers”** restrict used traces by limiting shot and/or receiver coordinates so that response from single trace, single shot or receiver, single shot or receiver line can be calculated as well as a response from a combination of a shot line with a receiver line. Supplementary parameters **“+/-”** specify necessary precision for the case of approximated coordinates. Additional parameters **“incremented by”** allow to select traces with regular spacing. Any of the section parameters may be omitted.

The **“Advanced migration parameters”** group contains couple options which were not studied enough. Among them **“Signal spreading”** resembles a PSDM parameter used to contrast images. The value is commonly set to the default 0 that means **“don’t apply”**. The same is true for the **“Footprint removal parameter”**.

TabDirectories

...describes correspondence between Windows paths visible by the ConfigDWM program and the Linux paths which the DWM3D * program uses on cluster.



This option allows substitution of any path parts during both import and export of the DWM job directories between WINDOWS and UNIX. This needs to be done only once for main directory and all sub-directories will be automatically arranged.

Press OK to save the edited job specification in a new **Task.ini** file specified in the first dialog.

Task Passport (Task.ini) Format

A task passport contains of four, five or six sections. The first section is started from the string [DWM3D]. The second one is started after [Apertures]. The third one begins from [InputFiles]. The fourth one is anticipated by [TempDirs]. Other sections [SEG-Y], [PathSubst] and [Trace] are optional and may be omitted. Section names are put in square brackets. Each section contains a set of parameters in the format "name = value". The order of the sections is arbitrary as well as the parameter sequence in each the section. Semicolon ";" starts comment part of a string. The task example is shown below:

```
[DWM3D]
FastOpen=0
FileModelVs=/1data/Tesseral/CalValley/VelCub_Out+.sgy
K_Vs=0.50
Epsilon=0.05
Delta=0.10
Alpha=0
Phi=0
Xx0=180452
Yx0=1738450
X00=171984
Y00=1743579
X0y=168240
Y0y=1737399
HorzBeforeVert=1
FileModelVp=/1data/Tesseral/CalValley/VelCub_Out+.sgy
K_Vp=1
FileBottomFormat=Surfer
FileBottom=/1data/Tesseral/CalValley/surf.grd
FileOutput=/1data/Tesseral/CalValley/DWM_AVO/DWMA_Const_De1_Eps_Ttr_0_Ap_1500_AVO.sgy
dX=10
dY=10
Zbeg=1000
Zend=1850
dZ=10
EikonalStepXY=10
SimilarityXY=10
SimilarityZ=10
NumTempDirs=1
workDir=/shared/dwm2
LogDir=
NumInputFiles=1
MindistXY=0
MaxDistXY=2500
ReliefForShots=0
ReliefForReceivers=0
BasType=0
BasMargins=20
BottomMargin=20
MaxTmpLen=119799808
DuplicateModel=0
DuplicateTimes=0
runStep=3
DistributeOutput=1
Recovery=0
Grad=0
CleanEdges=0
Kmul=1
Ttr=0
NumApertures=1
EikonalStepZ=10.00
MaxDistY=0
RectType=3
FreeBoss=1
OldFileNames=0
rangeFrom=0
rangeTo=1
[SEG-Y]
ZERO_TOP=1
ELEV_TO_Z=1
PROMAX=1
LINES_ALONG_Y=0
INLINE_START=1
INLINE_STEP=1
CROSSLINE_START=1
CROSSLINE_STEP=1
[InputFiles]
nObservationGridStep=0
FileInput1=/1data/Tesseral/CalValley/CDP_Gath_2_38.sgy
FileInput2=/1data/Tesseral/CalValley/CDP_Gath_39_74.sgy
FileInput3=/1data/Tesseral/CalValley/CDP_Gath_75_110.sgy
FileInput4=/1data/Tesseral/CalValley/CDP_Gath_111_146.sgy
FileInput5=/1data/Tesseral/CalValley/CDP_Gath_147_182.sgy
[PathSubst]
NumSubst=0
[FastOpen]
NumTraceInBlock=0
[TempDirs]
TempDir1=/1data/Tesseral/dwm2/
[Apertures]
Xmin1=-1500
Xmax1=1500
Ymin1=-1500
Ymax1=1500
Dist1=0
```

The task passport parameter values are supposed to be changed. But the parameters and section names

are fixed case sensitive words. File paths have to be printed in the format native for the operation system where the program runs.

Model specification parameters

... are specified in the dialog's **General/Model** group of parameters.

FileModel – obligatory, ASCII string: the P-wave velocity model file path. The model should be in the depth SEG-Y format. The file name extension either **'.sgy'** or **'.SGY'** is required. The file must be located in a shared directory. Contrary to standard SEG-Y the depth SEG-Y traces are recorded not along the time axis, but along the depth one. The depth axis is oriented downward. Depths are automatically converted to the units of X and Y coordinates by multiplying to corresponded integer power of 10. Commonly **m** or **ft** units are used for X/Y and logically substitute milliseconds of the SEG-Y traces. The values recorded the traces are supposed to be in the corresponded velocity scale. This means **m/sec** or **ft/sec** scale respectively. The model data must cover all the computation area and depths from Zbeg to Zend.

FileModelSW – optional, ASCII string: the S-wave velocity model file path similar to the P-wave velocity model file. It is used only if WaveTypes contains 'S' characters.

WaveTypes – optional, ASCII string: a sequence of 3 letters P or S. Each the letter represents a wave type for a segment of a ray path. The first letter corresponds to a segment from a source two first reflecting boundary. The first boundary can be either vertical or basic horizontal one depending on the parameter HorzBeforeVert. The second letter corresponds to a ray path segment between the reflecting boundaries. The third one corresponds to a segment from the last of two boundaries to a receiver. WaveTypes=PSP, WaveTypes=PSS, WaveTypes=PPS are typical values for the converted duplex waves. Default WaveTypes=PPP describes ordinary duplex waves. In the cases of absent WaveTypes or WaveTypes=PPP the parameter FileModelSW is not used and can be discarded.

ZERO_TOP (optional, boolean) parameter in the section [SEG-Y] controls interpretation of the recorded model elevations. Default nonzero means model traces always start from 0 depth. Else the model elevation is stored in trace headers. False (0) is default value despite true (non-zero) is frequently used.

FileBottom – obligatory, ASCII string: the bottom refracting boundary path. The file can be in the Surfer's grid format described bellow. Charisma, GeoQuest, Landmark, Paradigm, Schlumberger, Triple, and Zmap text grid formats are supported as well. The file must be located in a shared directory. The boundary must cover all the computation area from Xbeg to Xend and from Ybeg to Yend. Besides, all its depth values must be between Zbeg and Zend.

FileBottomFormat – optional, ASCII string: the refracting boundary format name. The name is case sensitive. Default value is FileBottomFormat=Surfer. The next key words are enabled: "Charisma", "GeoQuest", "Landmark", "Paradigm", "Schlumberger", "Surfer", "Triple", "Zmap".

HorzBeforeVert – obligatory, boolean: the flag controls path type of the duplex waves to process. Non-zero value means "Shot → Horizontal boundary → Vertical boundary → Receiver" way of rays. "0" means "Shot → Vertical boundary → Horizontal boundary → Receiver".

Output specification parameters

... are specified in the dialog's ...

FileOutput – obligatory, ASCII string: the output file path in SEG-Y format. The file must be located in a shared directory. The file name extension either **'.sgy'** is added automatically if absent.

The output (imaging) area is rectangular. There are two methods to define the rectangle:

– In the general case the output rectangle is defined by coordinates of 3 sequential corner points: Xx0,Yx0, X00,Y00, X0y,Y0y – optional, numeric parameters. These values are in the same units as model is. Commonly **m** or **ft** units are in use. Approximate values may be used. The program adjusts the coordinates

to form a true rectangle.

– In the case the output rectangle boundaries parallel to coordinate axes it can be alternatively defined by the next parameters: Xbeg,Xend, Ybeg,Yend – optional, numeric: minimal and maximal coordinates. These values are in the same units as model is. Commonly **m** or **ftunits** are in use. The program supports output of vertical sections, so combinations like Xbeg=Xend and Ybeg=Yend are enabled.

The task passport must include either Xx0,Yx0, X00,Y00, X0y,Y0y or Xbeg,Xend, Ybeg,Yend.

Zbeg,Zend – obligatory, numeric: depth interval of the output array. These values are in the same units as model is. Commonly **m** or **ftunits** are in use. Zbeg must be upper (smaller) then Zend. Zend must be below (bigger) then the basic boundary depths in all the output area.

dX,dY,dZ – obligatory, numeric: grid steps of the output array. These values are in the same units as model is. Commonly **m** or **ftunits** are in use. dX, dY and dZ must be positive multiples to EikonalStepXY and EikonalStepZ. Each the parameter must get positive value. Intervals between traces of input files and models don't affect dX, dY and dZ as well as orientation of observation lines or density of basic boundary grids.

Input specification parameters

NumInputFiles – obligatory, integer: a number of the input seismic files (3D seismograms). All the seismograms are listed in the section [InputFiles] as ASCII string parameters FileInput1, FileInput2, and so on. All the files must be located in share directories. The file name extensions either **'sgy'** or **'SGY'** are required.

ReliefForShots – optional, boolean: the flag controls use of the shot elevation values from input seismograms. Non-zero value means each shot point logically moved to its elevation height before calculating the wave arrival times. Default 0 means input seismograms were initially corrected for the relief. In such case the elevation values are ignored and shot points are placed on the upper boundary of the calculation cube.

ReliefForReceivers – optional, boolean: the flag controls use of the receiver elevation values from input seismograms. Non-zero value means each receiver logically moved to its elevation height before calculating the wave arrival times. Default 0 means input seismograms were initially corrected for the relief. In such case the elevation values are ignored and receivers are placed on the upper boundary of the calculation cube.

ELEV_TO_Z (optional, boolean) parameter in the section [SEG-Y] controls interpretation of the recorded trace elevations. Nonzero means trace elevations are measured relatively to the top model level. Else the trace elevations are measured relatively to zero (sea) level. False (0) is default value despite true (non-zero) is frequently used. This option has no affect if ReliefForReceivers/ReliefForShots are set to false (0).

MinDistXY, MaxDistXY – optional, numeric: minimal and maximal enabled planar distance between receiver and shot for a trace to be used in the migration. This value is in the same units as model is. Commonly **m** or **ftunits** are in use. Both the values set to the default 0 value mean "don't restrict".

MaxDistY – optional, numeric: maximal absolute difference between shot and receiver Y values of a trace to be used in the migration. This value is in the same units as model is. Commonly **m** or **ftunits** are in use. The default 0 value means "don't restrict".

MinRcvrXMinusShotX, MaxRcvrXMinusShotX – optional, numeric: minimal and maximal enabled shifts of receiver from shot in X direction for a trace to be used in the migration. This value is in the same units as model is. Commonly **m** or **ftunits** are in use. Both the values set to the default 0 value mean "don't restrict".

MinRcvrYMinusShotY, MaxRcvrYMinusShotY – optional, numeric: minimal and maximal enabled shifts of receiver from shot in Y direction for a trace to be used in the migration. This value is in the same units as model is. Commonly **m** or **ftunits** are in use. Both the values set to the default 0 value mean "don't restrict".

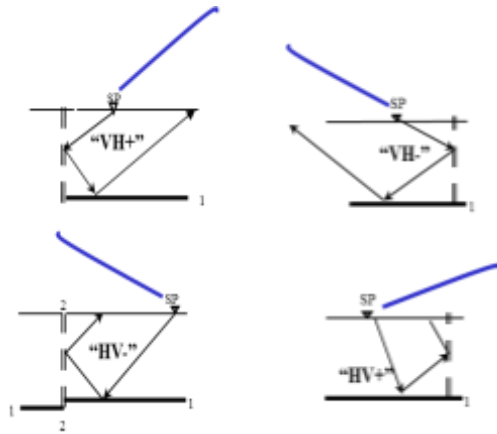
nObservationGridStep parameter in the section [InputFiles] is used to align all trace coordinate to a regular

grid with inter-node step equal `nObservationGridStep`. Such operation is useful to decrease number of different time cubes for non-regular observations, e.g. for offshore seismic.

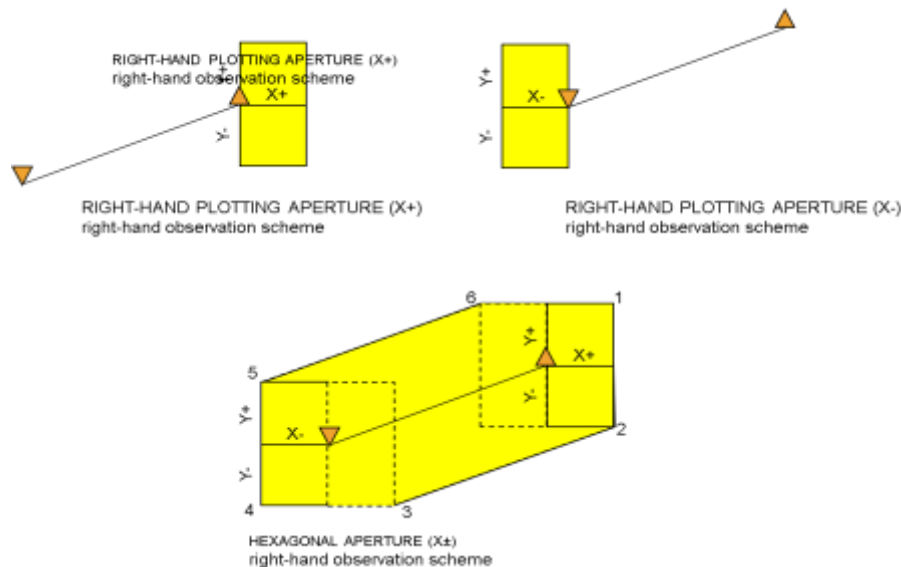
Aperture specification parameters

... are specified in the dialog's ...

DWM Provides 4 Views:

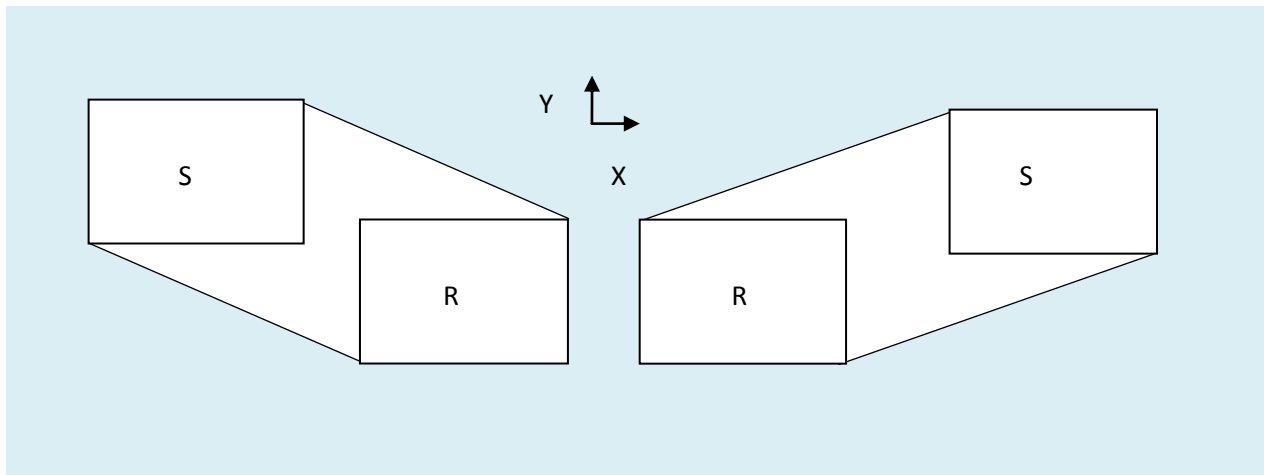


MIGRATION 3D MAPPINGS APERTURES

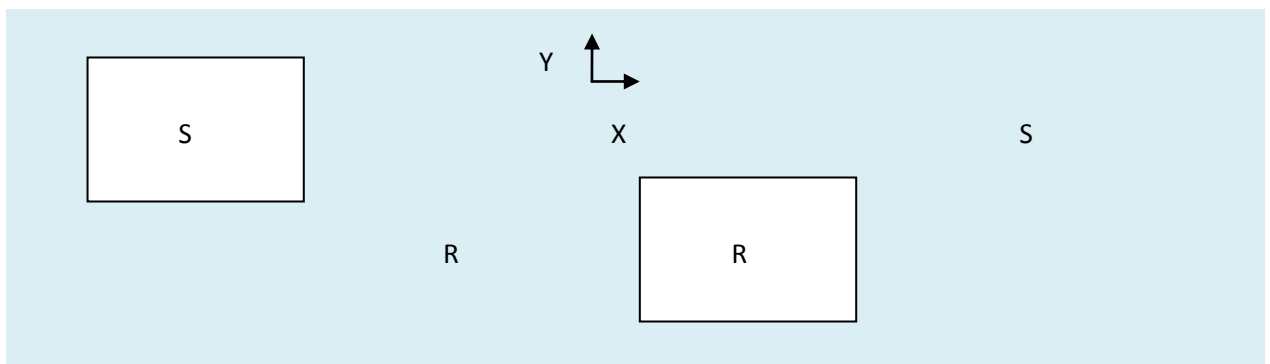


BasType – optional, integer from 0 to 6: the type of the migration apertures.

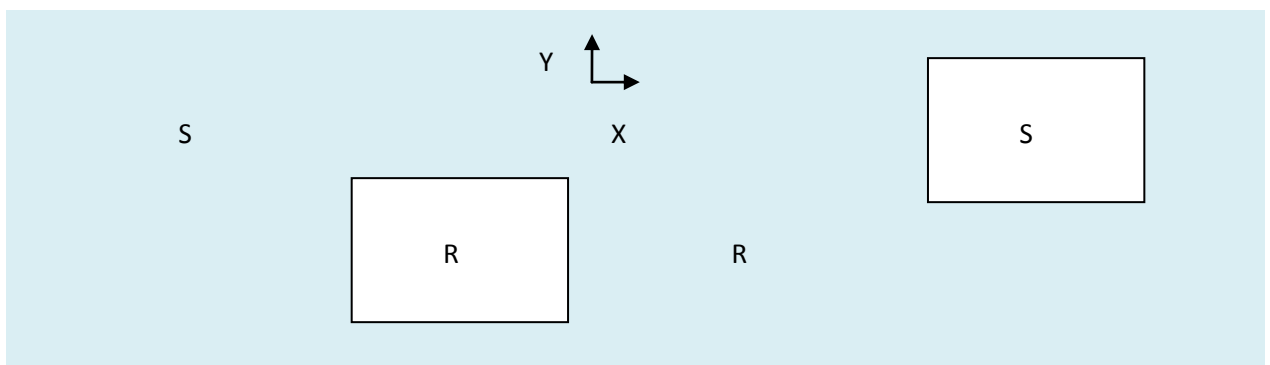
– Default 0 means area that is hexagonal in plan and not changed with depth. Each the hexagon can be imagined as a track of horizontal movement of a rectangle from the trace's shot to its receiver. The rectangle size is determined by the distance between the shot and receiver points. The dependency is described in the section [Apertures].



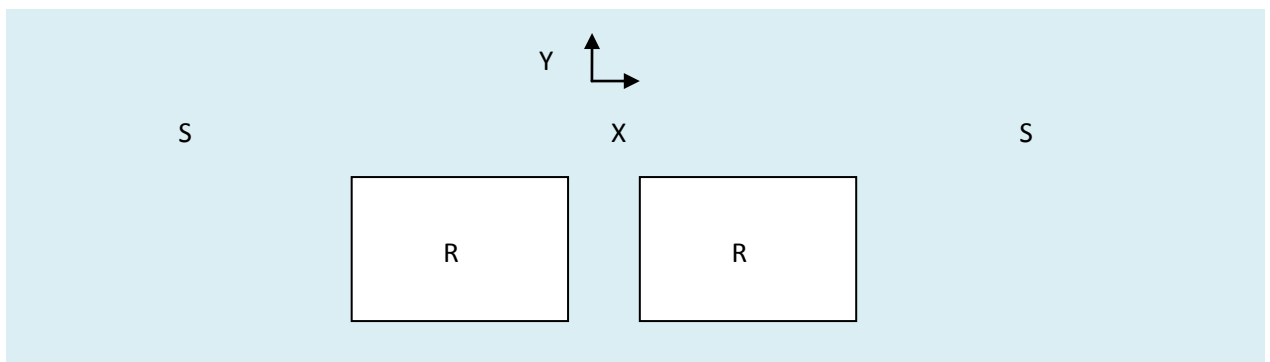
– 1 means area formed by the same rectangle fixed to that position from the trace's shot and receiver, where X is smaller. In other word it is the most left position.



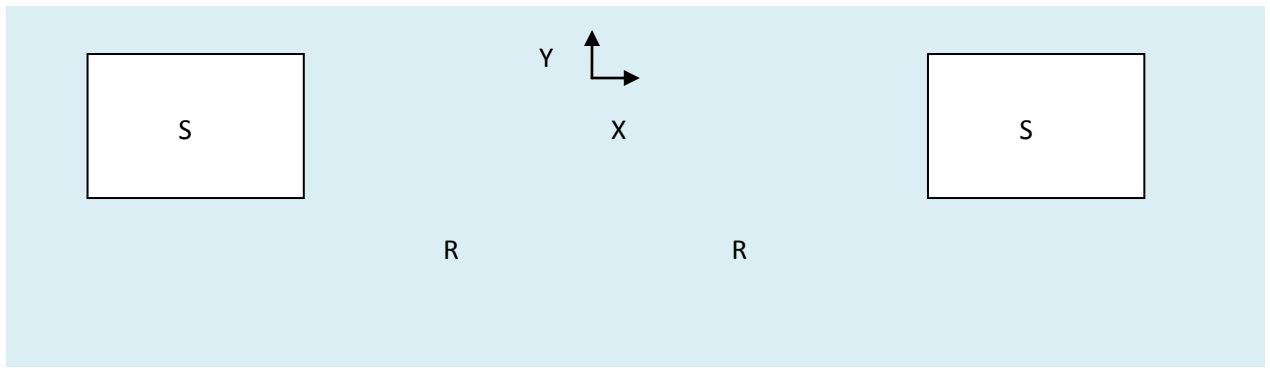
– 2 means the same rectangle at the most right position (where X is maximal).



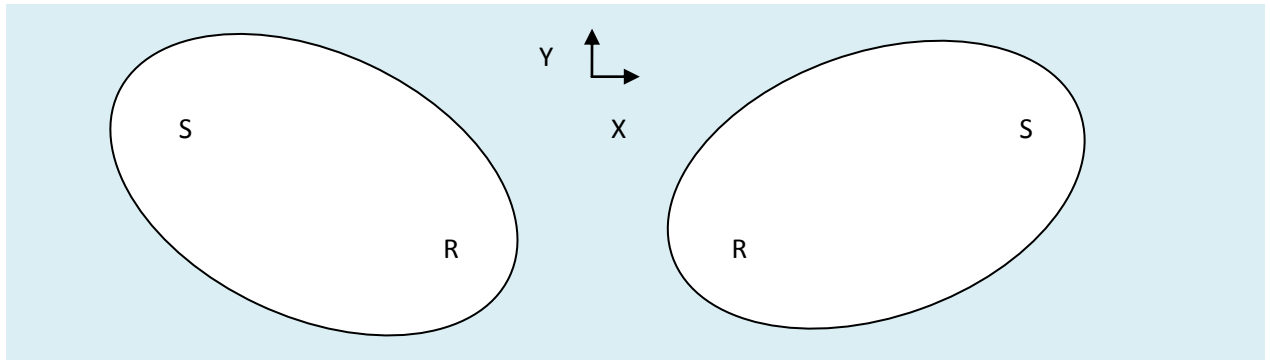
– 3 means the same rectangle where Y is minimal.



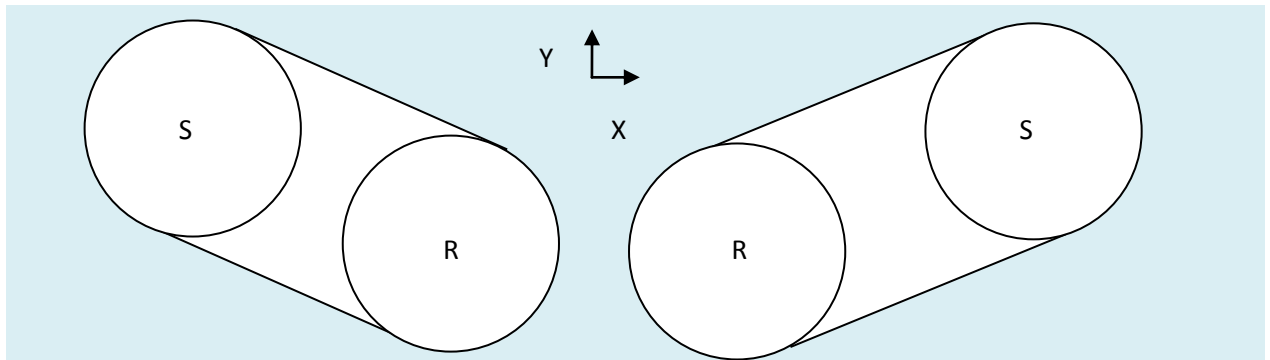
– 4 means the same rectangle where Y is maximal.



– 5 means ellipse with poles in the shot and receiver points and the major radius equal half distance between the poles plus aperture.



– 6 means two boundary connected circles with centers in the shot and receiver points and the radiuses equal to aperture.



BasType=0, 5 or 6 can be used for any orientation of input files. Other values require the input files, models and bottom refracting boundary to be preliminary rotated so that either X or Y axis corresponds to the estimated orientation of important sub-vertical boundaries. Values 1 and 2 are the best to image the sub-vertical boundaries approximately parallel to the Y axis (or a line from (X00,Y00) to (X0y,Y0y) in the case of rotated output rectangle). Values 3 and 4 are the best to image the sub-vertical boundaries approximately parallel to the X axis (or a line from (X00,Y00) to (Xx0,Yx0) in the case of rotated output rectangle).

NumApertures – obligatory, integer: a number of aperture sets. All the aperture sets are listed in the section [Apertures] in two forms:

– as numeric parameters Xmin1, Xmin2,..., Xmax1, Xmax2,..., Ymin1, Ymin2,..., Ymax1, Ymax2,..., and so on. Xmin<i>, Xmax<i>, Ymin<i>, and Ymax<i> together with early described BasType determine position of vertical boundaries for the area where signals from the trace can be added to. Dist<i> determines the shot-receiver distance associated with <i>-th set of the apertures. This association is a base for the linear interpolation of the apertures. There is recommended relation between BasType and apertures found upon the processed ray paths. BasType=0, 5, or 6 commonly correspond to symmetric apertures: Xmin<i>=-Xmax<i> and Ymin<i>=-Ymax<i>. BasType=1 commonly corresponds to Xmax<i>=0. BasType=2 commonly corresponds to Xmin<i>=0. BasType=3 commonly corresponds to Ymax<i>=0. And BasType=4 commonly corresponds to Ymin<i>=0.

– as numeric parameters Ap1, Ap2,..., and so on. This form is a simplified copy of the previous one for the case of Xmin<math>i>=-Ap<math>i>, Xmax<math>i>=Ap<math>i>, Ymin<math>i>=-

Ap<math>i>, and Ymax<math>i>=Ap<math>i>. Dist<math>i> is used similarly to the previous form.

BasMargins – optional, numeric: the width of additional margins assigned to the time cubes to decrease the influence of their borders on the time fields. The margin width should be in the same scale as coordinates are. Commonly **m** or **ftunits** are in use. The default value is 0.

BottomMargin – optional, numeric: the height of additional margins assigned to the time cubes to decrease the influence of their borders on the time fields. The margin height should be in the same scale as coordinates are. Commonly **m** or **ftunits** are in use. The default value is 0.

Time calculation parameters

EikonalStepXY, EikonalStepZ – obligatory, numeric: horizontal and vertical steps of a temporary grids used for the time cubes calculations. These values are in the same units as model is. Commonly **m** or **ftunits** are in use. EikonalStepXY and EikonalStepZ must be equal.

SimilarityXY, SimilarityZ – obligatory, numeric: horizontal and vertical distances between two receivers or shot points small enough to substitute the time cube instead if its recalculation. These values are in the same units as model is. Commonly **m** or **ftunits** are in use.

NumTempDirs – obligatory, integer: a number of other temporary directories reserved for the time files. These directories must be shared for uniform accessible from each of the task's cluster nodes. The directories are listed in the section [TempDirs] as ASCII string parameters TempDir1, TempDir2, and so on. The directories can be created automatically, but their parent directories must exist when the program starts.

WorkDir – optional, ASCII string: a path to a work directory reserved for the temporary files. It is recommended to be located so to maximize access speed, e.g. locally at each cluster node involved in the computation. Current work directory is used by default. The directory can be created automatically, but its parent directory must exist when the program starts.

Migration parameters

... are specified in the dialog's ...

Ttr – obligatory, numeric: threshold value (in seconds) used to suppress one-time refracted waves. Summing is executed only for the points where complete time of duplex wave is more then complete time of one time refracted wave plus Ttr. 0.05 and 0.1 are typical Ttr values.

Grad – optional, numeric: gradient used to suppress noised waves. The value is commonly set to the default 0 that means “don't apply”.

CleanEdges – optional, integer from -1 to +1.

– +1 means decreasing the migration sums at the edges of apertures via multiplication by cos.

– -1 means decreasing the migration sums in the centers of apertures via multiplication by (1-cos).

– The default value 0 meaning “do nothing” is used in most cases. Kmul – optional, numeric: scaling factor for the output file. It uniformly decreases amplitudes to protect results from abnormally high values. This coefficient is entered for convenience of interpreting only. It does not affect essence of the migration results. The default value is 1.

Optimization parameters

... are specified in the dialog's ...

Recovery – obligatory, boolean: the flag controls resuming of interrupted computations. Nonzero value

means “to continue and reuse already done calculations”.

MaxTmpLen – obligatory, integer: size of a temporary buffer for computation of both direct and refracted time cubes by the eikonal algorithm. The size is measured in 4-byte words. Too big size increases the computation time because of memory swapping. Too small size decreases precision and also increases the computation time because of extra scanning the intermediate files. Optimal value depends on available memory, number of processors per node and sizes of the intermediate files. It should be found posteriori.

DuplicateModel – optional, boolean: non-zero means “to distribute both P- and S-wave velocity model copies over work directories WorkDir”. The default value “0” means “don't copy and use the original model files”. Model duplication can improve productivity for paralleling on 10 and more processes.

DuplicateTimes – optional, boolean: non-zero means “to distribute time cube copies over work directories WorkDir”. The default value 0 means “don't copy and use the original time files”. Time duplication can improve productivity for paralleling on 10 and more processes but requires a lot of local disk memory.

RecType – optional, integer from 0 to 2: determines precision of the time files. It is recommended to be 2 to maximize access speed. (Default 0 provides highest precision but four times slows access and increases necessary storage size. 1 is an intermediate format two times slower than 2.)

The optional section [PathSubst] supports disk storages with asymmetry access speed. The section contains couples of correspondent paths: for ordinary I/O RdWrDir<i> and for accelerated read only access to the same files ReadDir<i>. Number of such couples is stored in NumSubst parameter of the same section.

FreeBoss – optional, boolean: a method of controlling the parallel computation. Non-zero value means on-line job dividing (that is a best choice for most cases). Default 0 means initial job dividing. (It is recommended for small level of paralleling – up to 5 processes).

runStep – optional, integer from 0 to 4: a number of the computation step to start from.

– Both 0 and 1 mean main computation scheme (first calculate all refracted times with associated direct times, then rest direct times, then migration).

– 2 means a scheme for quicker transiting to the migration step (first prepare all direct times, then migration together with the refracted times calculation). This scheme repeatedly calculates direct times for the points duplicating the refracted time sources and so is slower.

– 3 means migration.

– 4 means migration results joining only.

This scheme is used for sequential runs with ready to use time cubes. The parameter default value 0 is suitable for all cases.

OldFileNames – optional, boolean. In the last version temporary time cubes have got new names more convenient for typing the shell commands. Use OldFileNames=1 to restore previous naming. Generally, default value 0 is in use.

SEG-Y parameters

The optional section [SEG-Y] is used for adjustment of the program for specific input and output file parameters.

ELEV_TO_Z – optional, boolean. Nonzero means trace elevations are measured relatively to the top model level. Else the trace elevations are measured relatively to zero (sea) level. False (0) is default value despite true (non-zero) is frequently used. This option has no affect if ReliefForReceivers/ReliefForShots are set to false (0).

ZERO_TOP – optional, boolean. Default nonzero means model traces always start from 0 depth. Else the model elevation is stored in trace headers. False (0) is default value despite true (nonzero) is more frequently used.

PROMAX – optional, boolean. Nonzero means output file and trace headers will be filled according to ProMax rules. Default false (0) means more general form of the headers.

© 2015 TetraSoft LLC. All rights reserved. optional, boolean. Nonzero means output file lines will correspond to the

Y axis (or a line from (X00,Y00) to (X0y,Y0y). Default false (0) means they correspond to the X axis (or a line from (X00,Y00) to (Xx0,Yx0) in the case of rotated output rectangle).

INLINE_START, INLINE_STEP, CROSSLINE_START, CROSSLINE_STEP – optional, integer. These parameters control receiver group and shot parameters in the output cube trace headers. Each the parameter default value is 1.

Debug parameters

... are specified in the dialog's ...

rangeFrom, rangeTo – optional, numeric between 0 and 1: range of the input seismogram traces to process in portions of the whole set of traces of all input files together. Generally, default rangeFrom=0 and rangeTo=1 are in use. Other values can be used for debugging, for finishing of interrupted calculations or for calculations in parts.

The optional section [Trace] is used for the testing and debug purposes. Its optional numeric parameters XS/YS and XR/YR restrict used traces by limiting shot and/or receiver coordinates so that response from single trace, single shot or receiver, single shot or receiver line can be calculated as well as a response from a combination of a shot line with a receiver line. Supplementary parameters dXS/dYS and dXR/dYR specify necessary precision for the case of approximate coordinates. Additional parameters stepXS/stepYS and stepXR/stepYR allow to select traces with regular spacing. Any of the section parameters may be omitted.

Bottom Surface Format

... are specified in the dialog's ...

Bottom refracting boundary is presented in a grid format. Various grid formats are currently supported. Simplest of them, the Surfer GRD is a text format shown below.

DSAA

8	4						
-2100	2100						
-300	300						
1500	1501						
1500	1500	1500	1500	1500	1500	1500	1500
1500	1500	1500	1500	1500	1500	1500	1500
1500	1500	1500	1500	1500	1500	1500	1500
1500	1500	1500	1500	1500	1500	1500	1500

In the Surfer ASCII grid format 1st row contains a signature "DSAA". 2nd row contains the grid size in X and Y dimensions respectively. 3rd row contains Xmin and Xmax, and 4th one contains Ymin and Ymax. 5th row contains min and max value of the grid. The grid itself is printed from 6th row to the end of file. Grid nodes are ordered so that columns represent changing Y with fixed X.

The above sample presents horizontal surface of depth 1500. Its X varies from -2100 to 2100 with step 600, and Y varies from -300 to 300 with step 200.

Running Program

To start the "dwm3d_s" program under LAM MPI use command "mpirun". Detailed description of this command can be obtained in MPI documentation. Some useful variants are listed bellow:

- Run the "dwm3d_s" program on each processor of the cluster:


```
mpirun C dwm3d_s
```

- Run the “**dwm3d_s**” program on each node of the cluster:

```
mpirun N dwm3d_s
```

- Run 12 entities of the “**dwm3d_s**” program:

```
mpirun -np 12 dwm3d_s
```

- Run 6 entities of the “**dwm3d_s**” program so that 2 entities are run on node 0, 1 entity on node 1 and all others are executed on node 4:

```
mpirun n0,0,1,4,4 dwm3d_s
```

Other MPI implementations provide other options of mpirun. Another command can control MPI program execution in middleware such as SLURM (sbatch). GUI based cluster management systems are available for some clusters.

To run two or more tasks concurrently run “**dwm3d_s**” from different current directories and make sure that neither `WorkDir` nor `[TempDirs]` of different tasks coincides.

Being run the “**dwm3d_s**” program prints run time information both in standard output and ASCII text log files. Each the process has its own log file. Their names are “**DWM3D-0.log**”, “**DWM3D-1.log**”, “**DWM3D-2.log**”, and so on. Just the main process (number 0) prints its messages in the standard output. Read and analyze the log files in the cases of faults to understand what’s happened.

Temporary time cubes stored in the directories listed in `[TempDirs]` section of “**Task.ini**” can be re-used. So delete them before start of another task using the same directories.

Files “1.rec”, “2.rec” and so on located in the same place with the output SEG-Y file contain partial migration results of each of the parallel processes. They must be deleted after obtaining the migration results as well.

The log files “**DWM3D-*.log**” are appended by each the program run in the same current directory. So they are recommended to be also erased.

In the fault cases the program can left some garbage files in the `WorkDir` local directories. You can left them, if you re-use the `WorkDir` directory name. The program use the same temporary file names each time it runs.