Main Phase Modeling

Eficient 2D Finite Difference Modeling

Thomas Mejer Hansen

Bo Holm Jacobsen

Hans Christian Schiøtt

Uni Petersen

Main Phase ModelingEficient 2D Finite Difference Modeling

by Thomas Mejer Hansen Bo Holm Jacobsen
Hans Christian Schiøtt
Uni Petersen

Version 1.3 Edition
Published Wed Dec 10 11:10:53 CET 2008

An F77 program for efficient finite difference waveform modeling of complexity in selected phases.

Revision History

Revision 1.00 11 Sep 2001 Revised by: TMH
Initial release of MPM at Sourceforge
Revision 1.10 xx xxx 2002 Revised by: TMH,UP
First release after extensive code cleanup.

Table of Contents

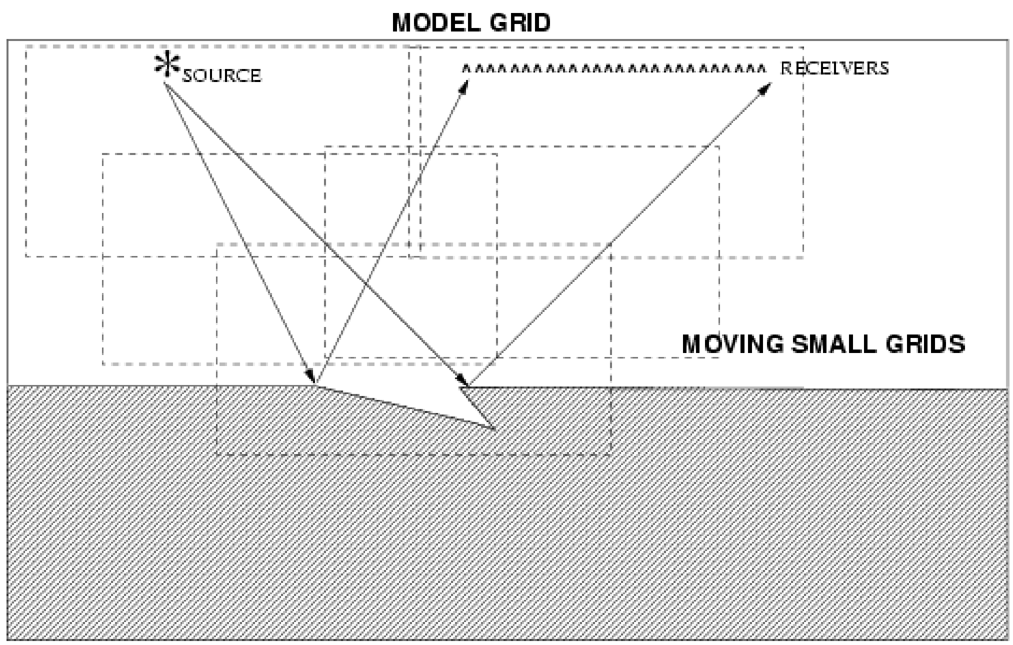
List of Figures

About MPM

Main Phase Modeling, MPM, is an efficient way to model the waveform of individual seismic phases using finite differences. Often large volumes of data are modeled even though only one particular phase is investigated, e.g. in wide angle modeling. Using the ray path obtained from e.g ray tracing, a specific phase or part of the dataset can be followed in the grid, using a moving box. This produces high quality wave field modelings, with significantly lower demands on computer CPU and memory.

This implementation is using a 4th order space 2nd order time 2D finite difference solution of the elastic wave equation (Levander, 1988)

Figure 1. Outline of MPM



MPM enables modeling of certain types of wav-propagation not available using convention finite difference based modeling tools :

Modeling of selected phases. By following one specific wavefront, it can be separated from other waves at the geophone locations. In this way MPM can be used as an analytical tool.

Efficient high frequency modeling of wide angle data. When modeling seismic wide angle data, one most often does only regard a time interval of recorded signals, by looking at the data in reduced time. This means that most if the original data is removed.

For such a case MPM will greatly increase both the modeling time and the memory requirements. Numerical tests show that the quality of the modeled data will match of conventionally modeled data by a factor of less 1e-5.

Another benefit of MPM could be to enable modeling using higher frequencies, than possible using conventional methods.

Chapter 1. Installation

1.1. Get the source

The latest source code is available at *http://mpm.sourceforge.net* along with this document.

1.2. System requirements

MPM should compile on any system with an ANSI Fortran 77 compiler. We have tested two compilers :

GNU fortran 77 (G77) is available for many platforms. MPM has successfully been installed on Linux (Gnu) and Windows(cygwin-tools). Thanks to *Sourceforge* (http://sourceforge.net/) MPM can be compiled for Intel/x86 - Linux, Intel/x86 - FreeBSD, Compaq/Alpha - Linux, Compaq/Alpha - Tru64, Sun/Sparc - Linux, and Sun/Sparc - Solaris.

Write to tmh@gfy.ku.dk if you want MPM compiled for any of these platforms.

Compilers from *PGI* (http://www.pgigroup.com/) significantly speeds up the code compared to GNU's compiler. Up to 70% increase in speed has been obtained compared to G77 on a single CPU Linux machine. The auto parallelization gave an increase of 57% on a DUAL Pentium Linux machine using both CPUS.

Several m-files are distributed with the source code. They have been testes using *Matlab 5* (http://www.mathworks.com/)

1.3. Install

As of version 1.1 you can install both using RPM files and from basic tar.gz files.

1.3.1. RPM file

To install the binary verions do

```
> su
> rpm -Uvh mpm-1.3
-1.i386.rpm
```

This will install mpm in /usr/bin/mpm, and the examples and m-files will be in /usr/doc/mpm-1.3 .

1.3.2. Install from source files

Assuming that GNU Make is installed on your system : Unpack the sourcefiles using one of the two following commands :

```
> gunzip -c mpm.tar.gz | tar tvf '-'
> tar xvfz mpm.tar.gz
```

This creates a directory called `mpm_1.3`.

The Fortran source code is found in `mpm_1.3 /src`, examples in `mpm_1.3 /examples` and help m-files and scripts in `mpm_1.3 /misc`. Documentation is located in `mpm_1.3 /doc`.

Go to the directory containing the sourcecode and compile the source :

```
> cd mpm_1.3
/src
> make
```

You may have to edit the `Makefile` before compiling the code, to select compiler and compiler options.

If GNU Make is not installted on your system, you can try to run the scripts `CompileMPM` :

```
> cd mpm_1.3
/src
> ./CompileMPM
```

An executable called `mpm`, should now be present in `mpm_1.3 /mpm`

1.3.3. Set path to Matlab files

If Matlab will be used to run the examples, add the path to the matlab files to Matlab using the `pathtool` from Matlab.

Using the RPM file the path will be `/usr/doc/mpm-1.3 /misc`.

Chapter 2. Running MPM

2.1. Input Files

MPM requires that several files are present at execution time. At least 5 binary model grids and one ascii file must be present.

2.1.1. l.mod, mu.mod, l2mu.mod, denu.mod, denw.mod [REQUIRED]

The staggered grid implementation can be defined in five grids, or five binary model files :

`l.mod1`, `mu.mod1`, `l2mu.mod1`, `denu.mod1`, `denw.mod1`, corresponding to `lambda`, `mu`, `lambda + 2 mu`, `rhox` and `rhoz`, \cite{Levander}.

Each file must be of size `bignx,bignz` and must be written in using direct access (if using 'fortran').

A model file must be written column by column :

```
for ix=1:nx
for iz=1:nz
  fwrite(fid,rhox(iy,ix),'float32');
end
end
```

or equivalently :

```
fwrite(fid,rhox,'float32');
```

`save_mpm.m` and `save_mpm_el.m` saves model files created in Matlab in the proper format.

2.1.2. mpm.par [REQUIRED]

This ascii file contains all parameters necessary to do FD modeling. Below is an example (see the examples for heavily commented examples). Variable names used in the source-code is listed in boldface :

```
# MODEL ON DISK (2 lines)
     6252,1667  bignx_disk,bignz_disk [samples]
     240        dx [meter]
```

```
    1          autopad
# MOVING BOX (3 lines)
    100000, 100000, 50000 wbox,hbox,bufferwidth [meter]
    0          movflag
    48240 1200 0 0  boxx0 boxz0 boxvpx boxvpz [meter,meter,meter/s,meter/s]
# TIME SAMPLING (2 lines)
    5          tmax [seconds]
    0.016      dt [seconds]
# SOURCE (4 lines)
    3 0        sourcetype[0;1;2;3],rotation[0;1]
    201 5      xs,zs [meters]
    4          maxf [Hz]
    50         pulsedelay [samples]
# RECEIVERS
    1          geotype[1;2;3]
    1200          geodepth [meter]
# BOUNDARIES (4 lines)
    1          freeupper
    2          absmode
    0.92, 20, 2, 2  edgefactor, dampingwidth,dampingexponent,dampingtype
    0.8, 6000, 4, 0.1 edegfactorf,vdamp,maxfdamp,expbas
# AUTOSAVE (2 lines)
    0          restoreautosave
    500        dautosave
# IO (6 lines)
    1 1 1 0    usnapflag,wsnapflag,divsnapflag,rotsnapflag
    2          snapsize
    25, 25     beginsnap,dsnap
    2          xyskip
    1          traceskip
    5          tskip
    1          verbose
```

This file MUST contain at least the lines presented above. The lines starting with # are for information only, and separates the parameters into sections :

2.1.2.1. MODEL ON DISK

bignx_disk,bignz_disk [samples]. `bignx_disk,bignz_disk` is the size of the model in samples, of the model to be used for modeling, i.e. the model on the disk.

dx [meters]. `dx` is the spatial sampling interval in meters

autopad. `autopad` determines whether to pad the disk-model with the size of the absorbin borders.

if this is set to '1', then a padded model will be written to disk as `l2mu_pad.mod`,`l_pad.mod`, and so forth. The model will be padded `dampingwidth` samples at all borders. Thus `bignx` will be changed to `bignx+2*dampingwidth`, and `bignz` will be changed to `bignz+2*dampingwidth`.

If a `freeupper=1` the top of the model is padded an additional 3 samples, the depth of the frea surface.

`autopad=1` will cause the positions of source and receivers to be shifted, internally in MPM. Thus a position given in `mpm.par` will relate to the position in the unpadded grid.

`bignx,bignz` is set to `bignx_disk,bignz_disk`. Thus if an absorbing boundary is used, then the region of absorption is within the model area. If a free surface is used, the top three samples of the loaded model will be used for modeling the free surface.

2.1.2.2. MOVING BOX

wbox,hbox,bufferwidth [meters]. `wbox,hbox]` is the size of the moving box in meters.

`bufferwidth` is the width of the bufferzone in meters. If this is set to 0, the the box will not be moving.

movflag. `movflag` movflag controls how the box is moving. [0], the positions of the center of the box is loaded from the file \textitwavepos.asc. [1] the next line defines the movement of the box :\newline

boxx0,boxz0,boxvpx,boxvpz.

boxx0,boxz0 is the startposition of the center of the box in meters.

boxvpx,boxvpz is the velocity of the moving box, \textit[m/s], in the x- and z-direction.

2.1.2.3. TIME SAMPLING

tmax [seconds]. `tmax` is time to model in seconds.

dt [seconds]. `dt` is the time sample interval in seconds.

2.1.2.4. SOURCE

sourcetype. `sourcetype` defines the type of source wavelet: [1] Ricker [2] cosine, [3] Gaussian. If chosen to [0] the source pulse is defined in the ascii file `source.asc`.

rotation. Determines whether the source is inserted as a [1] rotational S-wave or [0] a compressional P wave.

xs,zs [meters]. `xs,zs` defines the center of the source in meters.

maxf. `maxf` sets the dominant frequency of the sourcepulse.

pulsedelay. `pulsedelay` sets delay in time samples for the peak of the wavelet. [0] calculates the delay from the dominant frequency and the time sample interval.

2.1.2.5. RECEIVERS

RECEIVERS geotype : Determines the type of geophone [1]=velocity , [2]=(div/rot), [3]=('both'). \newline geopdepth is the depth in samples at which the geophones are positioned.[0] assumes the receivers are defined in the file `geopos.asc`. The positions of geophones in `geopos.asc` are in meters.

2.1.2.5.1. geopos.asc

`geopos.asc` contain a list of positions (x,z) in meters for geophones, one per line. The file must end with a line starting with the number '999999' :

```
 11.00  10.00   FIRST GEOPHONE
 10.00   9.00   SECOND GEOPHONE
  9.00  10.00   THIRD GEOPHONE
 10.00  11.00   FOURTH G.....
...
999999
```

2.1.2.6. BOUNDARIES

BOUNDARIES freeupper [1] for a free surface, [0] otherwise\newline absmode [0] Clayton and Enqvist, \citeClayeng. [1] Cerjan type exponential damping \shortciteCerjan). [2] Wide angle Fresnel Zone based, [3] Deep and narrow Fresnel Zone based. Type [2] and [3] are experimental absorbing boundaries, not currently suitable for production modeling. The next line is only used if \varabsmode was chosen to [1]. The edgefactor sets the damping value at the border. dampingwidth sets the with of the damping zone in samples. At \vardampingwidth samples from the border the damping will be 1, i.e. no damping. dampingexponent is the exponent in the exponential formula. dampingtype determines wether only the stress field, [1], ot both the stress and velocity fields [2] are damped. The last line in the boundary section is only used of Fresnel zone base boundaries have been chosen (\varabsmode=[2] or [3]) edgefactorf sets the damping value at the border. vdamp the the velocity, and maxfdamp sets the maximum frequency to be used to determine the width of the Fresnel zone. expbase sets the exponent of the damping expression.

2.1.2.7. AUTOSAVE

restoreautosave. `restoreautosave` [1] Restart FD-modeling from auto saved data, [0] Start from beginning of modeling.

dautosave. `dautosave` set the interval in timesamples at which the whole state of the modelling is saved.

2.1.2.8. IO

usnapflag,wsnapflag,divsnapflag,rotsnapflag. determines whether snapshots of velocities (u_t, w_t) or 'Divergence' and 'Rotational' fields are written to disk : [1] snapshots, [0] no snapshots.

snapsize. `snapsize` determines the size of the snapshot region : [1] The small moving grid (nx,nz). [2] The buffer grid (nx+buffer,nz+buffer). [3] The whole model (bignx,bignz). Option [2] is under consideration at the moment and not ment to be used.\newline The snapfiles can be inspected in Seismic Unix by \varxmovie < ut.snap n1=nz n2=nx regarding the values of xyskip and snapsize.

CURRENTLY ONLY OPTION [3] IS WORKING FOR ALL TYPES.

beginsnap,dsnap. beginsnap is the time sample from which snapshots are printed for every dsnap timesteps.\newline

xyskip. xyskip skips every \varxyskip sample for snapshots.\newline

traceskip. `traceskip` skips every `traceskip` geophone.

tskip. `tskip` skips every `tskip` time sample for geophones.

verbose. `verbose` determines the amount of information printed to the terminal window : [-1] None, [0] Very brief, [1] Normal, [5] Huge amount of information.

2.1.3. wavepos.asc

If movflag was set to [0] in \verb1mpm.par1, then \verb1wavepos.asc1 must be defined. This ascii file \textitwavepos.asc gives the position of the center of the box at a specific time step. below is an example :

```
 0     1000   1000
10    70000  70000
20     1000   1000
```

Each line determines the center (in meters) of the moving box, defined by column 2 and 3 (x and z) at a specific time, defined by column 1 (in seconds)

2.1.4. receiver.asc

2.1.5. source.asc

2.2. Output from MPM

\textitwaveposout.asc contains the position of the top left corner of the moving box in samples for every timestep.\newline \textitsourceout.asc contains the source wavelet at all timesteps.\newline

Files with the extension \file.autosave are auto saved snapshots of the stress- and velocity-fields and the time step at which auto save occurred

2.2.1. Text Information

Several files with `.asc` extension, contains information about the modeling progression.

2.2.1.1. waveposout.asc

This file contains the position of the moving box at each times step, [time, x, y].

```
    0.00706999982  10535.  7035.
    0.0141399996  10535.  7035.
    0.0212099999  10570.  7035.
    0.0282799993  10570.  7035.
    0.0353499986  10605.  7035.
    0.0424199998  10640.  7035.
  .
  .
  .
```

2.2.1.2. modelpadding.asc

This file conatins information on the amount of modelpadding generated by MPM.

[padleft,padtop,padright,padbottom] :

```
50 53 50 50
```

2.2.1.3. geophoneposout.asc

This file contains the position of each geophone in the grid :

```
  1   51.   54.
  2   52.   54.
  3   53.   54.
  4   54.   54.
  5   55.   54.
  6   56.   54.
  7   57.   54.
  8   58.   54.
.
.
.
```

2.2.1.4. sourceout.asc

This file conatins the inserted sourcem for all modeled timesteps

```
   1.55307494E-12
   2.23443638E-12
   3.20685817E-12
   4.59121586E-12
   6.55710078E-12
   9.34183546E-12
   1.32766576E-11
   1.88226813E-11
   2.66201436E-11
   3.75556704E-11
   5.28538741E-11
.
.
```

2.2.2. Seismograms

\filegeou.f77,\filegeou.f77,\filegeodiv.f77,\filegeorot.f77 contains the seismograms if specified in the parameter file. The files can be read using \varf77strip.m. \newline The same seismograms as in the *.f77 files are kept in the *.bin files. The *.bin files are ment to be used directly in Seismic Unix by e.g. \varxwigb < geou.bin n1=nt

2.2.3. Snapshots

\fileut.snap and \filewt.snap contains velocity snapshots, and \filediv.snap and \filerot.snap contains the snapshots for the divergent (P-wave) and rotational (S-wave) fields.\newline

Can be read using \varf77strip.m. and can be plotted using \filempmmov.m

2.2.4. Snapshots

\fileut.snap and \filewt.snap contains velocity snapshots, and \filediv.snap and \filerot.snap contains the snapshots for the divergent (P-wave) and rotational (S-wave) fields.\newline

Can be read using \varf77strip.m. and can be plotted using \filempmmov.m

2.3. Matlab m-files

Several m-files are included to help set up the model and process the data. All M-files work equally well with \hrefhttp://www.mathworks.comMatlab 5 or using the freely available Octave. f77strip.m Reads the Fortran style binary file into memory f77unstrip.m Saves a matrix into a Fortran style binary float-file. save_mpm.m Creates the proper elastic grids from Vp, and Vp/Vs and Vp/Density ratios. save_mpm_e,.m Creates the proper elastic grids from Vp, Vs and Density vp2den.m Computes velocity profile from density profile. From Christensen and Mooney, JGR, 1995,VOL 100, No. B7 den2vp.m Computes density profile from velocity profile. From Christensen and Mooney, JGR, 1995,VOL 100, No. B7 mpmmov.m [Matlab 5 only] Shows a movie of binary snapshot files. vonk2d Creates 2D Random fields with both a Gaussian and multimodal PDF, using Gaussian, Exponential and von Karman distributions.

2.3.1. mpmmov.m

Shows a movie of moviefile.

```
>> mpmmov('div.snap',1e-4)
```

Show a movie starting at snapshot 10, ending at snapshot 14, in steps of 2 snapshots :

```
>> [movnx,movnz]=get_movie_size;
>> mstart=10; mend=14; mstep=2;
>> mpmmov('div.snap',movnx,movnz,mstart,mend,mstep,1e-4)
```

You can choose to return a matrix suitable for movie creation as :

```
>> M=mpmmov('div.snap',movnx,movnz,mstart,mend,mstep,1e-4)
>> movie(M,4)
```

Fianlly you can save the movie `M` as an AVI file using :

```
>> movie2avi(M,'movie.avi')
```

2.3.2. read_mpm_par.m

Gets information from `mpm.par`.

```
>> [wbox,bignx]=read_mpm_par('wbox','bignx')
```

2.3.3. write_mpm_par.m

Write the file `mpm.par`. The first eight parameters must be present.

```
>> write_mpm_par(bignx,bignz,dx,xs,zs,tmax,Vmax,Vmin,'wbox',wbox,'hbox',hbox)
```

Chapter 3. Examples

Examples of the use of MPM can be found in \cmd{mpm-1.3 /examples/}, or /usr/doc/mpm-1.3 /examples.

Make sure mpm is located in the path.

Be sure to add 1.3 /misc/ to the Matlab path if using the matlab examples.

All examples can be run both as a matlab script and as a shell script. e.g. the first benchmark test can be called from matlab using `bench1.m`, and from a shell using `bench1.sh`.

3.1. Benchmark

3.1.1. Bench I

A short benchmark.

```
COMPUTER          COMPILER         TIME(s)
----------------------------------------
Linux 1Ghz          pgf77             28
Linux 1Ghz         g77 -Wall          44
Linux 1Ghz          g77 -O3           42
Linux 800Mhz     g77 -O3 -Wall        78
Linux 2.4, 800Mhz ifc -O3 -W0         39
Linux 2.4, 800Mhz  ifc -W0            47
----------------------------------------
```

3.1.2. Bench II

A longer benchmark, with a moving box.

```
COMPUTER          COMPILER         TIME(s)
----------------------------------------
----------------------------------------
```

3.2. moving_box

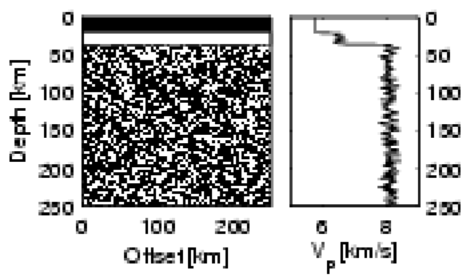
the moving box example, consists of 2 examples, using the same model.

3.2.1. No Moving Box

3.2.2. Steady Moving Box

3.3. Wide Angle Wavefield Modeling

Figure 3-1. Model for wide angle modeling example



3.3.1. Steady Box

3.3.2. Moving Box

3.3.3. Comparison