Using IWAVE

William. W. Symes [1]

ABSTRACT

IWAVE is a framework for time-domain regular grid finite difference and finite element methods. The current IWAVE package includes source code for component libraries and commands, and examples of typical IWAVE acoustic modeling use cases. The examples are taken from a recent paper on error propagation for heterogeneous medium simulation using finite differences, and allow the user to replicate illustrations of the interface error effect which renders all FD methods effectively first-order accurate. This paper gives a brief tour of the common IWAVE use cases illustrated in the package examples.

INTRODUCTION

Domain-specific simulation such as seismic modeling begs for software re-use via modular design. All applications of this type have the same structure: static fields are initialized, dynamic fields updated, output extracted. A modular approach to code architecture is implicit in this structure, and further specialization leads to even more opportunity for code re-use via modular design.

The software package described in these pages, IWAVE, takes advantage of the aforementioned intrinsic modularity. IWAVE is open source software for finite difference or finite element time-domain simulation on regular rectangular grids, written exclusively in the C99 dialect of ISO C. IWAVE is built around a core framework: that is, a collection of separate software packages which together provide essential services upon which applications may be built. These service components completely define the interfaces to which additional code must be written to formulate a complete application.

Along with the core framework, the current release contains a complete finite difference time-domain acoustic modeling application, featuring

- simple parameter-driven job control;
- modeling in 1D, 2D, and 3D space;
- staggered grid schemes of orders 2 in time and 2k, k = 1, 2,..., in space
- serial, loop-parallel, and task-parallel execution models, scaling to thousands of threads;

- flexible specification of PML absorbing or pressure-release boundary conditions on all faces of the simulation cube;
- arbitrary source and receiver locations, and flexible point and array source specification including simultaneous source modeling (random, plane-wave,...)
- standard input and output data formats (SEGY, RSF)

An isotropic elastic modeling application with similar features, and built around the same core framework, has been developed and will be included in a forthcoming release.

The primary purpose of this short paper is to illustrate the use of IWAVE to calculate synthetic acoustic seismograms. To that end, the paper describes a simple application - 2D synthetic seismogram generation over a simple structural model of the sedimentary column - and provides a set of demonstration examples ("demos") which the reader may reproduce, along with complete annotation of the files needed for job specification and sample graphics derived from the results (as well as commands to produce these graphics).

A secondary purpose is to supply the user with the means to independently verify some of the claims in the paper by Symes and Vdovina (2009), namely the existance of an error component in synthetic data derived from strongly heterogeneous models, in addition to the well-known grid dispersion error. The examples presented here are essentially the same as those presented in that paper. By installing IWAVE and running the demonstrations described here, the reader may reproduce the computational content of (Symes and Vdovina, 2009).

IWAVE is both a standalone application, and a component of the Madagascar software suite (Fomel, 2009) The application package and the examples discussed here (and indeed this paper itself) may be built either independently, or within Madagascar, as explained in detail below.

IWAVE was used in a quality control role in the SEAM Phase I project - see Fehler and Keliher (2011) for an account, including discussion of the many difficulties of large scale numerical simulation of seismograms.

The internal details of IWAVE are not discussed here, except insofar as is necessary to explain the use of the main commands. Symes et al. (2011) briefly describe the structure of the IWAVE framework, with emphasis on its object-oriented design and the resulting mechanisms for coupling modeling with optimization packages to produce inversion applications. The IWAVE project web page (Terentyev et al., 2012) provides extensive reference material, and further information about the design.

The paper begins with a brief review of the system of partial differential equations solved (approximately) by IWAVE's acoustic application, and the choice of finite difference method. The following section presents the examples of Symes and Vdovina (2009), along with some additional examples based on the same distribution of mechanical parameters which shed light on the impact of finite difference order on solution accuracy. Instructions for recreating these examples follow. The paper ends with a brief discussion of the prospects for improvements in performance and accuracy in FD technology, and the evolutionary advantages flowing from the modular, or object, orientation of IWAVE. Two appendices describe the job parameters used in the examples, and download and install instructions.

ACOUSTODYNAMICS

The IWAVE acoustics application is based on the pressure-velocity form of acoustodynamics, consisting of two coupled first-order partial differential equations:

$$\rho \frac{\partial \mathbf{v}}{\partial t} = -\nabla p \tag{1}$$

$$\frac{1}{\kappa}\frac{\partial p}{\partial t} = -\nabla \cdot \mathbf{v} + g \tag{2}$$

In these equations, $p(\mathbf{x}, t)$ is the pressure (excess, relative to an ambient equilibrium pressure), $\mathbf{v}(\mathbf{x}, t)$ is the particle velocity, $\rho(\mathbf{x})$ and $\kappa(\mathbf{x})$ are the density and particle velocity respectively. Bold-faced symbols denote vectors; the above formulation applies in 1, 2, or 3D.

The inhomogeneous term g represents externally supplied energy (a "source"), via a defect in the acoustic constitutive relation. A typical example is the *isotropic point source*

$$g(\mathbf{x}, t) = w(t)\delta(\mathbf{x} - \mathbf{x}_s)$$

at source location $\mathbf{x}_s$.

Virieux (1984) introduced finite difference methods based on this formulation of acoustodynamics to the active source seismic community. Virieux (1986) extended the technique to elastodynamics, and Levander (1988) demonstrated the use of higher (than second) order difference formulas and the consequent improvement in dispersion error. IWAVE's acoustic application uses the principles introduced by these authors to offer a suite of finite difference schemes, all second order in time and of various orders of accuracy in space.

The bulk modulus and buoyancy (reciprocal density) are the natural parameters whose grid samplings appear in the difference formulae. I will display velocity and density instead in the examples below. IWAVE's acoustic application converts velocity and density to bulk modulus and buoyancy as part of the problem setup phase; the user may supply any equivalent combination of parameters.

EXAMPLES BASED ON A 2D DOME MODEL

This simple 2D model embeds an anticline or dome in an otherwise undisturbed package of layers. The velocity and density models are depicted in Figures 1 and

2. These figures display sampled versions of the models with $\Delta x = \Delta z = 5$ m; the model fields are actually given analytically, and can be sampled at any spatial rate.

Symes and Vdovina (2009) use this model to illustrate the *interface error* phenomenon: the tendency, first reported by Brown (1984), of all finite difference schemes for wave propagation to exhibit first order error, regardless of formal order, for models with material parameter discontinuities. Figure 3 exhibits a shot gather, computed with a (2,4) (= 2nd order in time, 4th order in space) staggered grid scheme, $\Delta x = \Delta z = 5$ m and an appropriate near-optimal time step, acquisition geometry as described in caption. The same gather computed at different spatial sample rates seem identical, at first glance, however in fact the sample rate has a considerable effect. Figures 4 and 5 compare traces computed from models sampled at four different spatial rates (20 m to 2.5 m), with proportional time steps. The scheme used is formally 2nd order convergent like the original 2nd order scheme suggested by Virieux (1984), but has better dispersion suppression due to the use of 4th order spatial derivative approximation. Nonetheless, the figures clearly show the first order error, in the form of a grid-dependent time shift, predicted by Brown (1984).

Generally, even higher order approximation of spatial derivatives yields less dispersive propagation error, which dominates the finite difference error for smoothly varying material models. For discontinuous models, the dispersive component of error is still improved by use of a higher order spatial derivative approximation, but the first order interface error eventually dominates as the grids are refined. Figure 6 shows the same shot gather as displayed earlier, with the same spatial and temporal sampling and acquisition geometry, but computed via the (2,8) (8th order in space) scheme. The two gather figures are difficult to disinguish. The trace details (Figures 7, 8) show clearly that while the coarse grid simulation is more accurate than the (2,4) result, but the convergence rate stalls out to 1st order as the grid as refined, and for fine grids the (2,4) and (2,8) schemes produce very similar results: dispersion error has been suppressed, and what remains is due to the presence of model discontinuities.

See (Symes and Vdovina, 2009) for more examples, analysis, and discussion, also (Fehler and Keliher, 2011) for an account of consequences for quality control in large-scale simulation.

CREATING THE EXAMPLES - RUNNING IWAVE APPLICATIONS

IWAVE builds with SConstruct (`http://www.scons.org`), either as an independent package or as part of Madagascar (Fomel, 2009). For download and install instructions, see Appendix B and sources cited there.

The examples are also scripted with SConstruct. Providing scripts makes the results convenient to rebuild, and I'll first explain how to do that. I also explain how one uses the basic IWAVE commands from the command line, outside the context of these examples.

Scripted Examples

To build the intermediate data and figures for the examples described here,,

- install IWAVE, either within Madagascar or standalone. I will use `$TOP` to denote the path to the top-level IWAVE directory for the standalone version, or to the top-level Madagascar build directory `$RSFSRC`.
- for the standalone version of IWAVE, the examples build with the assistance of SU, which must also be installed. The choice of word order in IWAVE and SU must be compatible: either the XDR option must be set in both, or in neither. Default for both IWAVE and SU is native binary word order. To set the XDR option for SU, follow instructions in `Makefile.config`; for IWAVE, configure compilation with the flag `-DSUXDR`, as described in Appendix B and (Terentyev et al., 2012).
- the Madagascar version builds with entirely with Madagascar commands, so no external package need be supplied.
- build data and figures: in the standalone version of IWAVE,
  - `cd $TOP/demo/data`
  - `scons`

  , or, in the Madagascar version,
  - `cd $TOP/book/trip/iwave/data`
  - `scons`
  - `scons lock -f madfig.sc`

  (the last step is necessary only if you wish to build a copy of this paper from source - it archives the newly created figures and makes them available to the paper build, as described in the next bullet)
- to (re)build this paper, build the figures first. Then in the standalone version,
  - `cd $TOP/papers`
  - `scons`

  or, in the Madagascar version,
  - `cd $TOP/book/iwave`

– scons

Note that the finest (2.5 m) grid consists of roughly 10 million gridpoints. Consequently the modeling runs collectively take a considerable time, from a minutes to a substantial fraction of an hour depending on platform, on a single thred. This example is computationally large enough that parallelism via domain decomposition is worthwhile. IWAVE is designed from the ground up to support parallel computation; a companion report will demonstrate parallel use of IWAVE.

Inspection of the `SConstruct` file in `data` will show that the modeling tool used is `$TOP/asg/main/asg.x`, the IWAVE acoustic modeling command (in Madagascar Flows, this command is referenced simply as `asg`, which is an alias for `$RSFROOT/bin/sfasg`, where `$RSFROOT` is an alias for the Madagascar install directory). Input data is supplied by a parameter list, stored in a file. The model-building tool `standardmodel` builds the velocity and density model files, and works the same way - many of the parameter files in the `data` directory are input for this tool.

Both the IWAVE acoustic modeling command and `standardmodel` self-doc in the style of SU or Madagascar. For modeling command, the self-doc is deprecated in favor of the web documentation mentioned above.

Using IWAVE commands in other contexts

To use the acoustic modeling command outside of the scripted examples, the user needs to create a parameter list. The job parameters for the use case of the scripted examples are described in detail in Appendix A. The html documentation (Terentyev et al., 2012) describes other parameter choices, corresponding the wide variety of use cases accommodated by this application. Key parameters are pathnames to the model data files (velocity and density, or equivalent parameters) and to seismic trace files containing prototype output trace headers and (possibly) source pulse traces, and output traces on normal completion.

The acoustic application currently expect model data files in the RSF format of Madagascar (Fomel, 2009). The scripts use `standardmodel` to store gridded model data in RSF format, and data from other sources will need conversion to this format. An RSF data set consists of two files, an ascii header (grid metadata) file and a flat binary data file. The data set is referenced by the header file name; one of the parameters listed in the header file is the pathname of the binary data file, with key `in`. The header file is small and easily created by hand with an editor, if necessary. Many archival data formats make the grid sample values available as a flat binary file - this is true for instance of the gridded models output by GOCAD (`http://www.gocad.org`), for which the `vo` files contain virtually the same information as (so may easily be translated to) RSF header files in ascii form, and the `vodat` files are flat binary files which may be used unaltered as RSF binary files.

By convention, the dimension of the problem is that of the primary model grid, that associated with the bulk modulus data, if it is given, or failing that, the velocity. This grid is also the primary grid of the simulation: that is, the space steps used in the finite difference method are precisely those of the bulk modulus, or velocity, data. Thus the choice of simulation grid is made externally to IWAVE.

The IWAVE acoustic application uses specific internal scales - m/ms for velocity, g/cm^3 for density, and corresponding units for other parameters. To ensure that data in other (metric) units are properly scaled, the RSF header file should specify a value for the `scale` key, equal to the power of 10 by which the data should be multiplied on being read into the application, to convert to the internal scale. For example, if velocities are given in m/s, the header file should include the line `scale = -3`. In forthcoming releases, this device will be deprecated in favor of explicit unit specifications.

One of IWAVE's design criteria is that acquisition geometry parameters should have no a priori relation to the computational grid geometry: source and receiver locations may be specified anywhere in Euclidean space. The current release accepts a SU (SEGY without reel header) format data file specified by the `hdrfile` keyword: the trace headers in this file are those of the output (pressure) traces. Units of length and time are m and ms respectively, consistent with other internal unit choices. The example scripts use SU or Madagascar commands to create these prototype trace files.

The source pulse may be specified as a Ricker wavelet, or read from another SU file, whose pathname is the value associated with the `source` keyword. Source calibrarion is regulated by several other keywords, as described in Appendix A. In the examples, the Ricker option is used, simply because it avoids some small incompatibilities between SU and Madagascar filter implementations which would otherwise prevent the standalone and Madagascar versions of the examples from generating the same pulses, and therefore prevent the results from matching precisely.

Because the number of parameters describing a simulation task is reasonably large (roughly 15 in a simple case), the job parameters for IWAVE's acoustic application are most conveniently stored in a file, passed to the application via a command line parameter. Denoting by `$ASG` either `$TOP/asg/main/asg.x` for the standalone implementation, or `$RSFROOT/bin/sfasg` for the Madagascar install, the proper command takes the form

```
[prefix] $ASG par=[parfile].
```

Here `[prefix]` is any necessary command prefix, eg. `mpirun ...`, and `[parfile]` is the pathname of the parameter file. On successful completion, the output data will be stored in a file (SU format) indicated by the key `datafile` in the parameter file.

DISCUSSION AND CONCLUSION

The rather large and only slowly disappearing error revealed by the examples from Symes and Vdovina (2009) suggests strong limits for the accuracy of regular grid finite difference methods. Finite element methods suffer from the same limitations: accurate solution of acoustodynamic or elastodynamic problems appears to demand interface-fitted meshed (Cohen, 2002), with the attendant increase in code and computational complexity.

The situation may not be so bleak, however. For one special case, namely constant density acoustics, Terentyev and Symes (2009) show that a regular grid finite difference method, derived from a regular grid Galerkin finite element method, has accuracy properties one would expect in homogeneous media (second order convergence, reduction of grid dispersion through higher order space differencing) even for discontinuous models: the interface error effect is attenuated. This type of result actually goes quite far back in computational geophysics (see for example Muir et al. (1992)), though theoretical support has been slower in coming.

Pure regular grid methods cannot take advantage of changes in average velocity across the model, and concommitant changes in wavelength. Coupling of local regular grids is possible, however, and can yield substantial computational efficiency through grid coarsening in higher velocity zones - see Moczo et al. (2006). IWAVE already accommodates multiple grids (in domain decomposition parallelism), and extension to incommensurable multiple grids would be a significant change, but in principle straightforward. The use of logically rectangular but geometrically irregular ("stretched") grids is completely straightforward, on the other hand.

These and other extensions, both past and future, are eased by the reusability designed into the IWAVE core framework. This design has produced reasonably well-performing and easy-to-use applications, and has proven extensible to new models and schemes. Moreover, as explained by Symes et al. (2011), the object-oriented design of IWAVE dovetails with similarly designed optimization software to support the construction of waveform inversion software. The inversion applications resulting from this marriage inherit the features of IWAVE - parallel execution, high-order stencils, efficient boundary conditions, simple job control - without requiring that these aspects be reworked in the code extensions.

The IWAVE acoustic application supports many use cases beyond those of the scripted examples, such as various modes of parallel execution, array sources, movie output, 3D modeling, and many others described in the documentation. It is hoped that the brief overview above, the detailed description of the example parameters given in Appendix A, and the much more extensive description of use cases in (Terentyev et al., 2012) will enable the reader to constuct a wide variety of synthetic data sets with relative ease.

ACKNOWLEDGEMENTS

IWAVE has been a team effort: the original design of the core framework is due to Igor Terentyev, and Tanya Vdovina, Dong Sun, Marco Enriquez, and Xin Wang have each made major contributions. Development of IWAVE was supported by the SEG Advanced Modeling (SEAM) project, by the National Science Foundation under awards 0620821 and 0714193, and by the sponsors of The Rice Inversion Project. The IWAVE project owes a great deal to several open source seismic software packages (Seismic Un*x, SEPlib, Madagascar), debts which we gratefully acknowledge. The author wishes to record his special gratitude to Sergey Fomel, the architect of Madagascar, for his inspiring ideas and his generous and crucial help in the integration of IWAVE into Madagascar.

REFERENCES

Brown, D., 1984, A note on the numerical solution of the wave equation with piecewise smooth coefficients: Mathematics of Computation, **42**, 369–391.

Cohen, G. C., 2002, Higher order numerical methods for transient wave equations: Springer.

Fehler, M., and P. J. Keliher, 2011, SEAM Phase I: Challenges of subsalt imaging in tertiary basins, with emphasis on deepwater gulf of mexico: Society of Exploration Geophysicists. (eISBN=9781560802884, eBook catalog number 114E).

Fomel, S., 2009, Madagascar web portal: `http://www.reproducibility.org`, accessed 5 April 2009.

Levander, A., 1988, Fourth-order finite-difference P-SV seismograms: Geophysics, **53**, 1425–1436.

Moczo, P., J. O. A. Robertsson, and L. Eisner, 2006, The finite-difference time-domain method for modeling of seismic wave propagation: Advances in Geophysics, **48**, 421–516.

Muir, F., J. Dellinger, J. Etgen, and D. Nichols, 1992, Modeling elastic fields across irregular boundaries: Geophysics, **57**, 1189–1196.

Symes, W. W., D. Sun, and M. Enriquez, 2011, From modelling to inversion: designing a well-adapted simulator: Geophysical Prospecting, **59**, 814–833. (DOI:10.1111/j.1365-2478.2011.00977.x).

Symes, W. W., and T. Vdovina, 2009, Interface error analysis for numerical wave propagation: Computational Geosciences, **13**, 363–370.

Terentyev, I., and W. W. Symes, 2009, Subgrid modeling via mass lumping in constant density acoustics: Technical Report 09-06, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, USA.

Terentyev, I., T. Vdovina, X. Wang, and W. W. Symes, 2012, IWAVE: a framework for wave simulation: http://www.trip.caam.rice.edu/software/iwave/doc/html/index.html, accessed 21 Sept 2012.

Virieux, J., 1984, SH-wave propagation in heterogeneous media: Velocity stress finite-difference method: Geophysics, **49**, 1933–1957.

———, 1986, P-SV wave propagation in heterogeneous media: Velocity stress finite-difference method: Geophysics, **51**, 889–901.

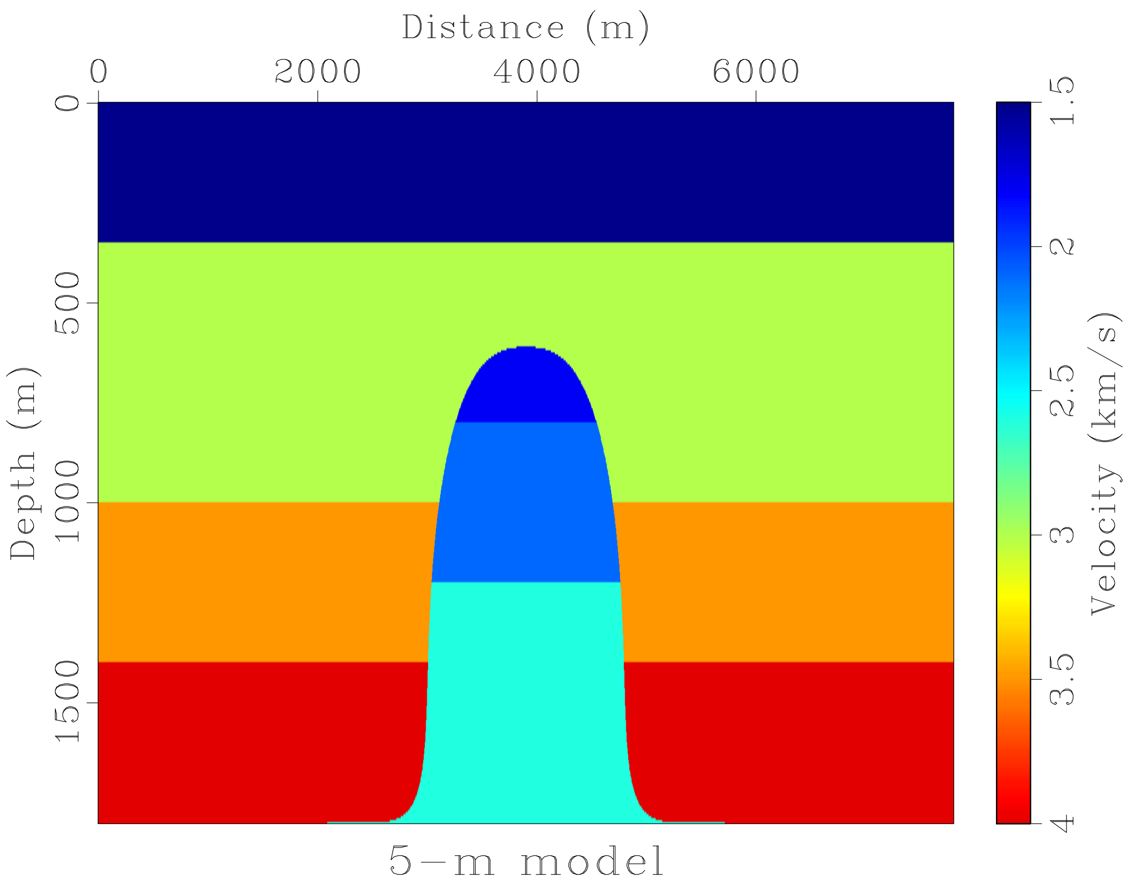

Figure 1: Dome velocity model

ANNOTATED PARAMETER FILES

All IWAVE applications are parameter-driven: that is, they accept as input a *map* or associative array, defined by a list of `key = value` pairs. These parameter specifications can be included on the command line. However, because the number of such parameter specifications is rather large, it's convenient to store them in a parameter file ("par file"). The use of a par file has the added advantage that the file may include annotations and white space to improve readability.

The examples displayed in this paper are created in the directory `$TOP/demo/data`. The par file `parfile` is a by-product of data creation - the SConstruct script text-processes it from prototype files including macros, which are resolved when the scripts are run. Four such prototype par files are present in `data`, each one defining a

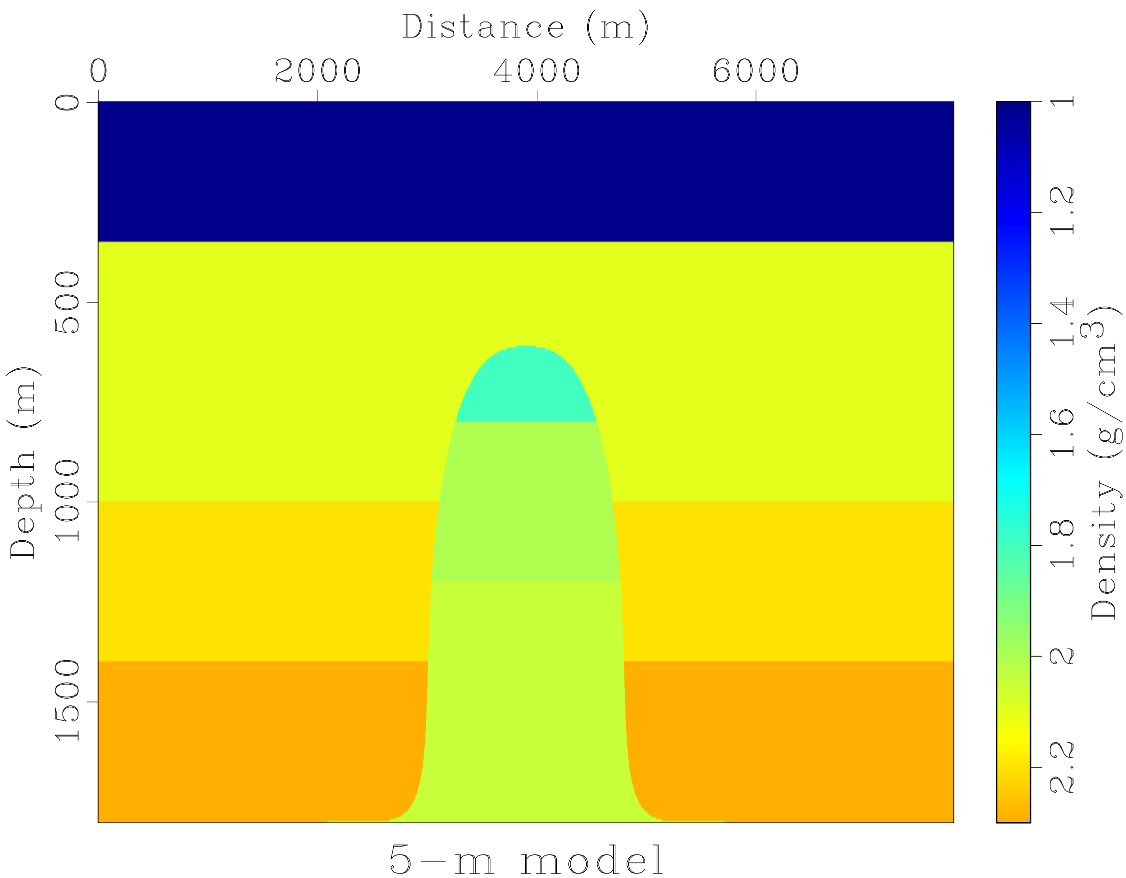

Figure 2: Dome density model

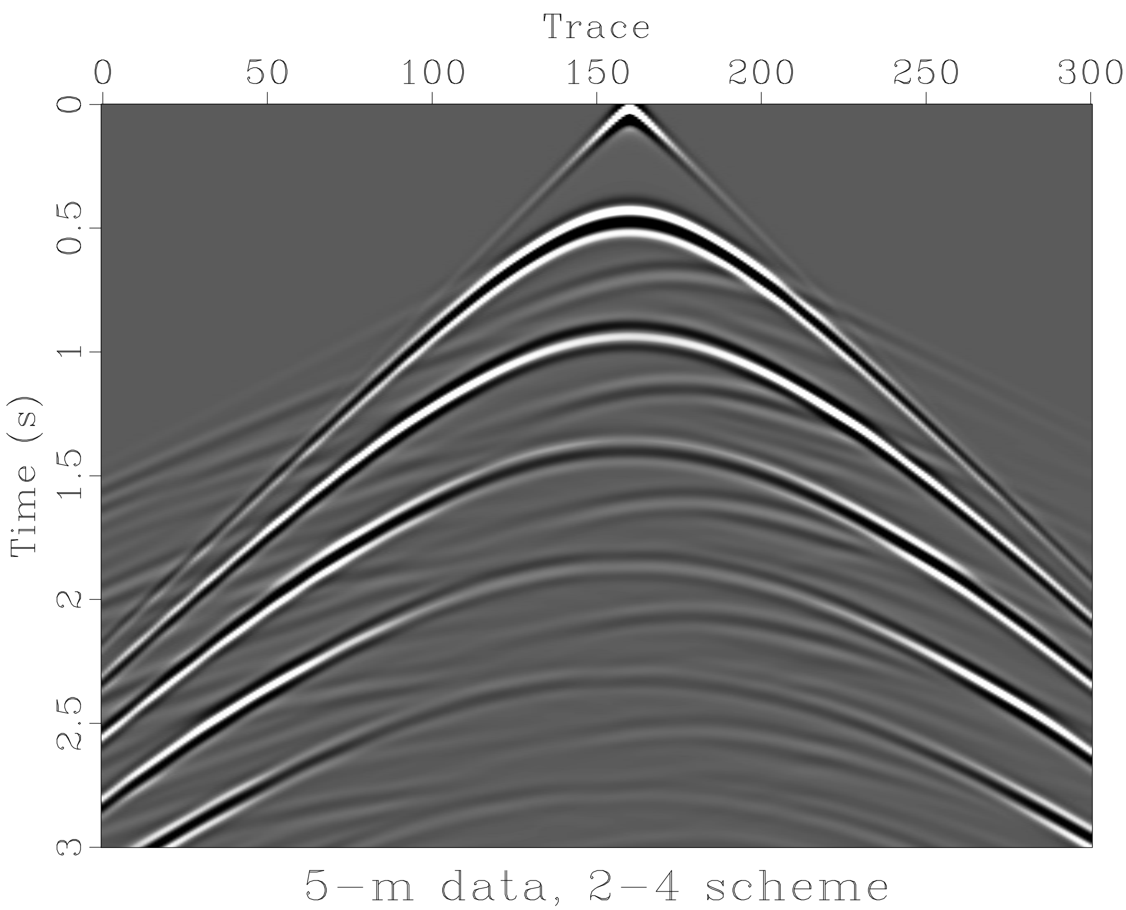

Figure 3: 2D shot record, (2,4) staggered grid scheme, $\Delta x = \Delta z = 5$ m, appropriate Δt, 301 traces: shot x = 3300 m, shot z = 40 m, receiver x = 100 - 6100 m, receiver z = 20 m, number of time samples = 1501, time sample interval = 2 ms. Source pulse = zero phase trapezoidal [0.0, 2.4, 15.0, 20.0] Hz bandpass filter.

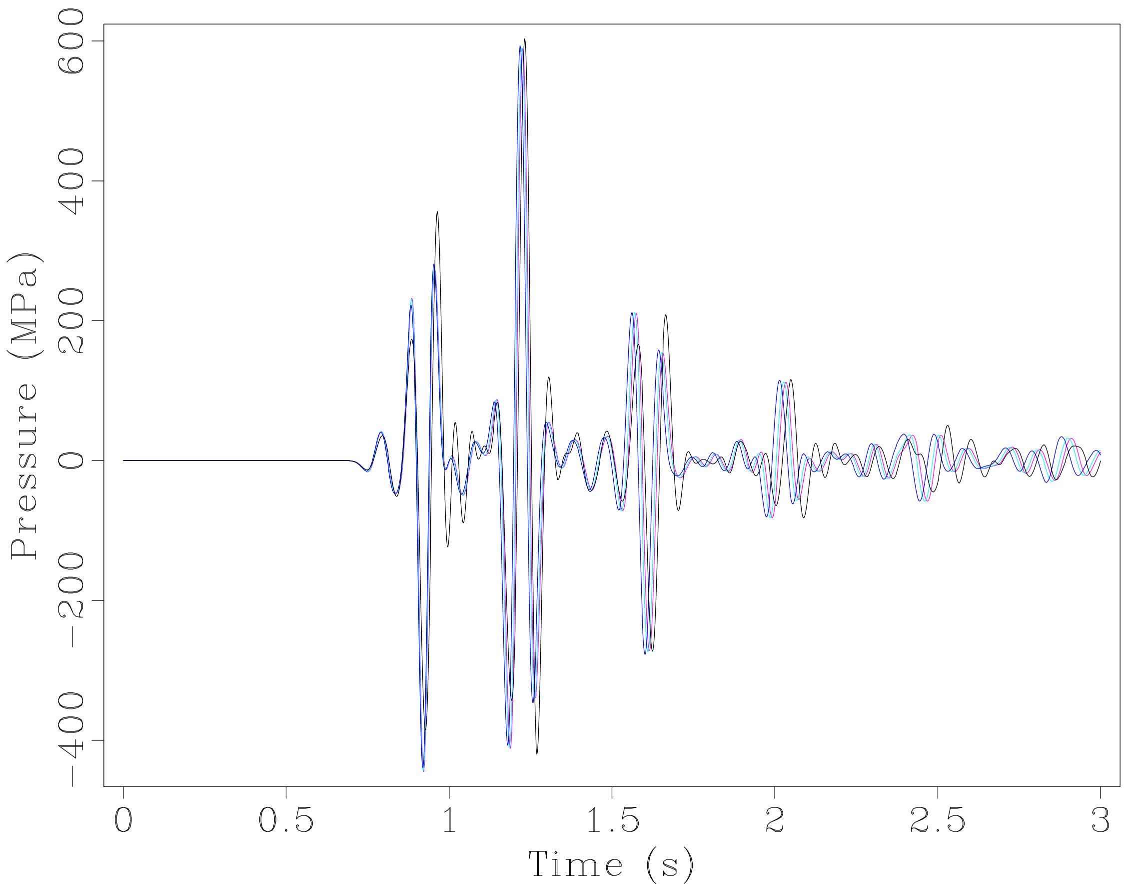

Figure 4: Trace 100 (receiver x = 2100 m) for $\Delta x = \Delta z = 20$ m (black), 10 m (blue), 5 m (green), and 2.5 m (red). Note arrival time discrepancy after 1 s: this is the interface error discussed in (Symes and Vdovina, 2009). Except for the 20 m result, grid dispersion error is minimal.

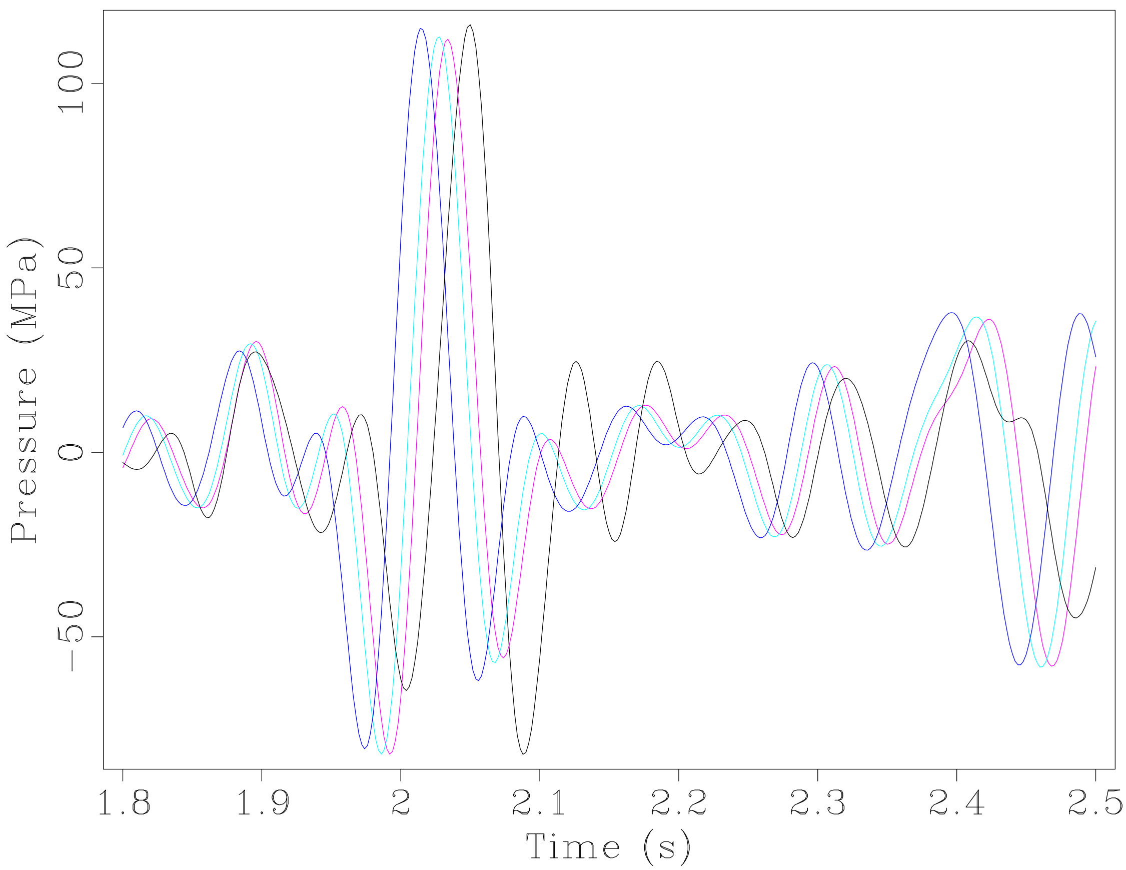

Figure 5: Trace 100 detail, 1.8-2.5 s, showing more clearly the first-order interface error: the time shift between computed events and the truth (the 2.5 m result, more or less) is proportional to Δt, or equivalently to Δz.

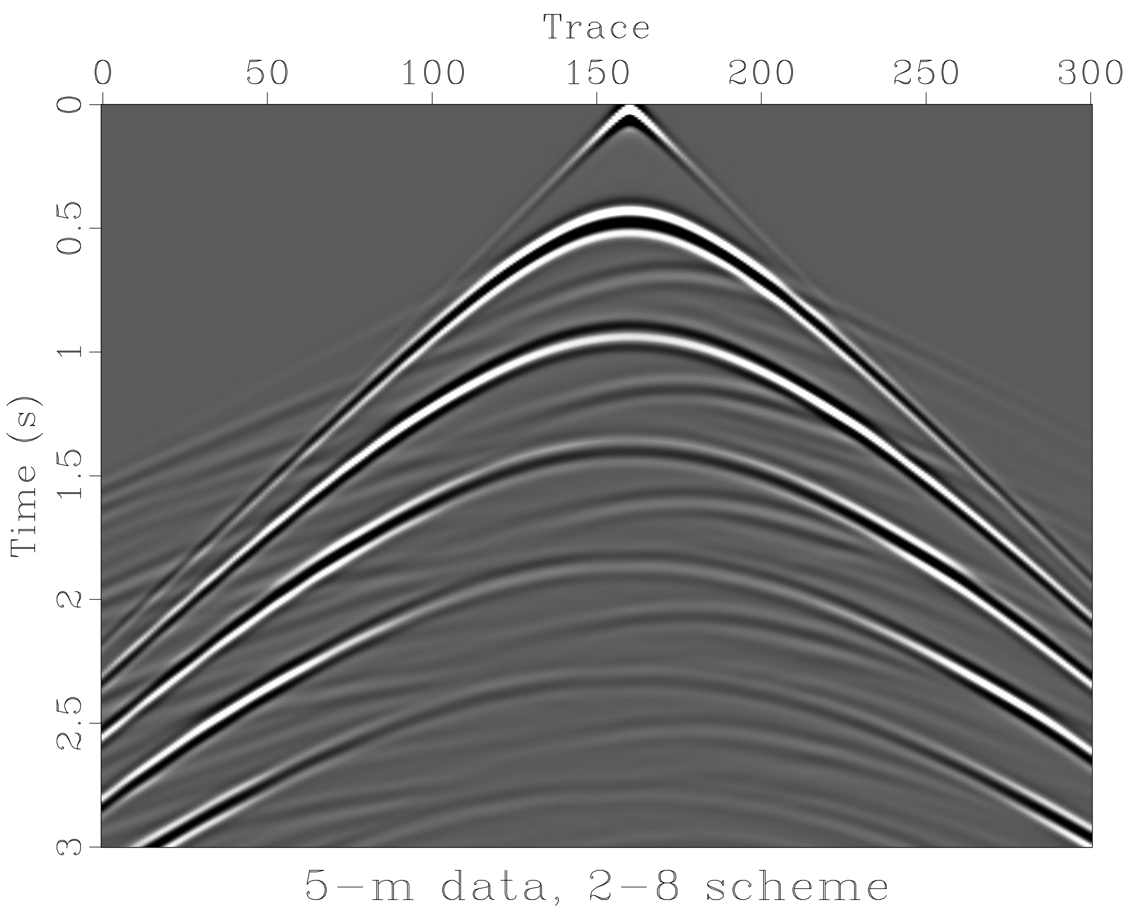

Figure 6: 2D shot record, (2,8) scheme, other parameters as in Figure 3.

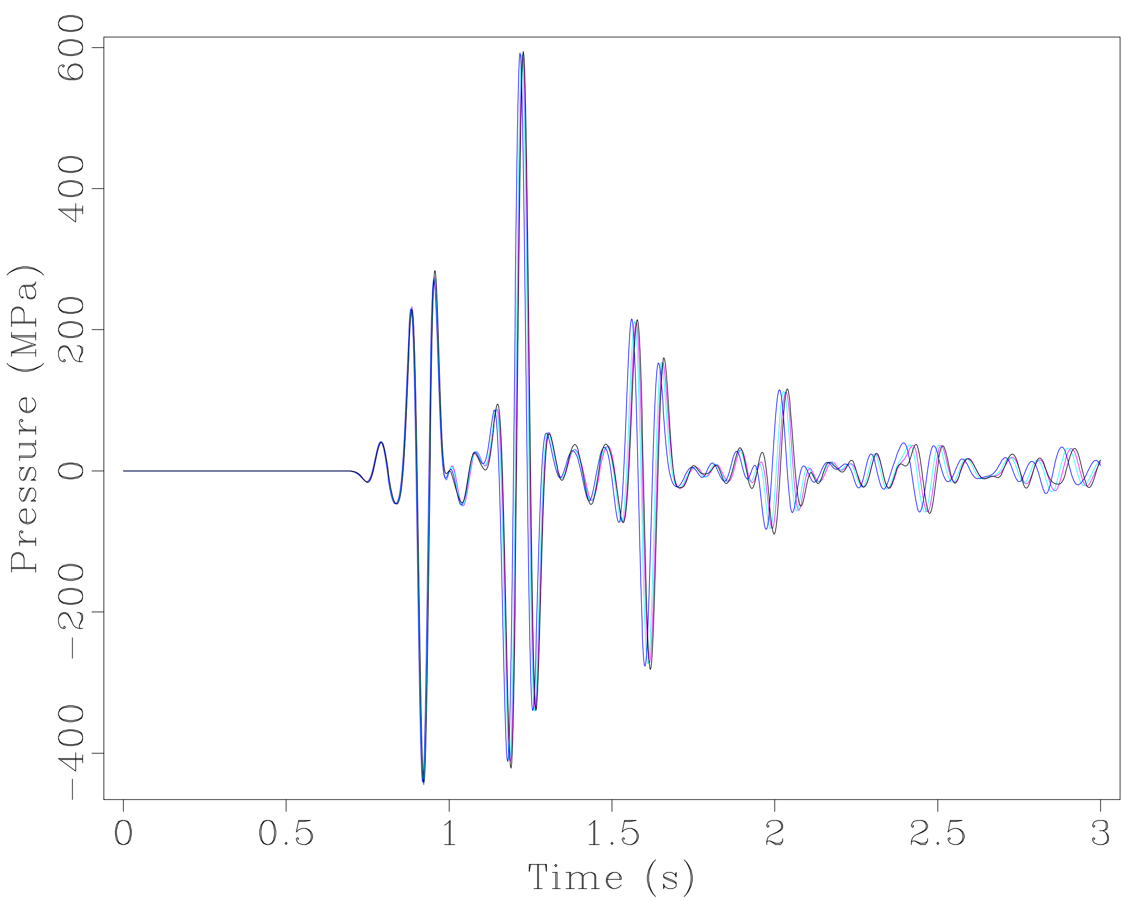

Figure 7: Trace 100 computed with the (2,8) scheme, other parameters as described in the captions of Figures 3 and 4.

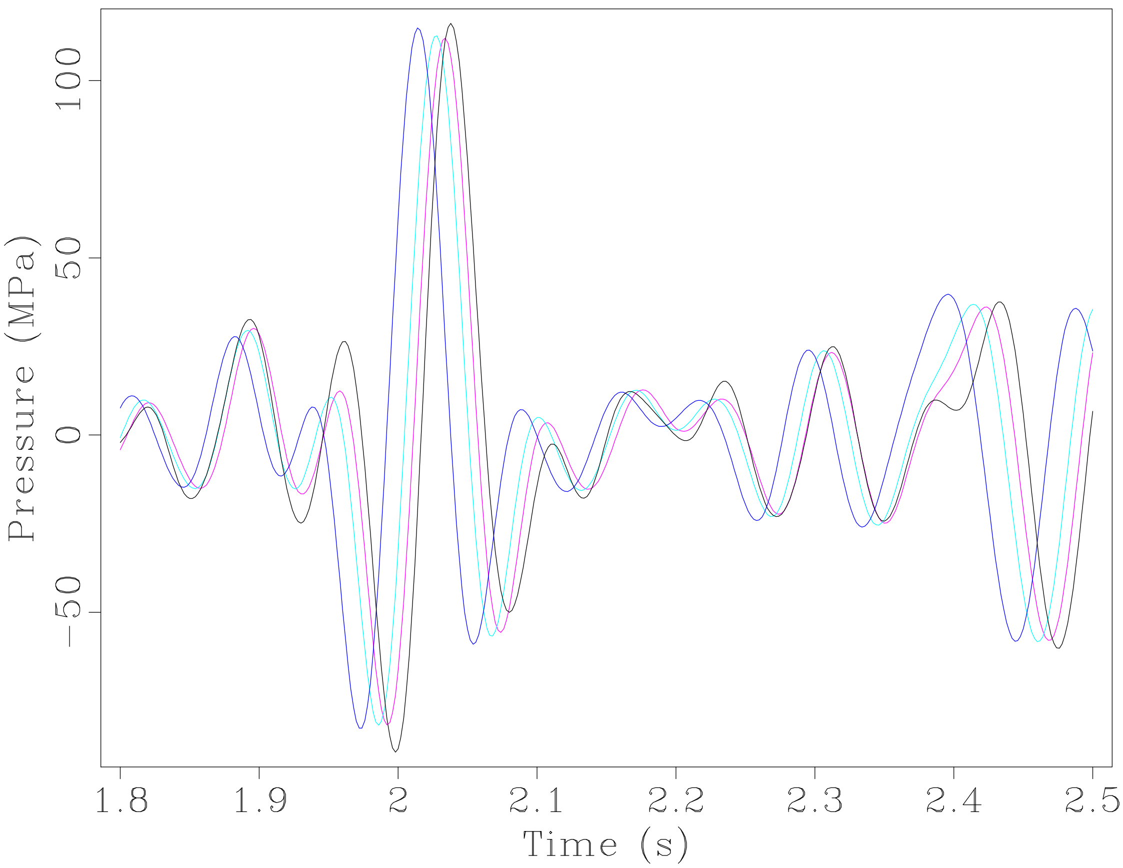

Figure 8: Trace 100 detail, 1.8-2.5 s, (2,8) scheme.. Comparing to Figure 5, notice that the dispersion error for the 20 m grid is considerably smaller, but the results for finer grids are nearly identical to those produced by the (2,4) grids - almost all of the remaining error is due to the presence of discontinuities in the model.

modeling task corresponding to a given level of grid refinement. The actual input to the modeling command is `parfile`.

The meaning of each parameter in the par file is described in the IWAVE web documentation (Terentyev et al., 2012). This appendix gives a brief description of the parameter assignments appearing in the `parfile` generated for the 20 m grid example. To run this example, and coincidentally generate its parameter file,

- `cd $TOP/demo/data`
- `scons demo20m`

The file `parfile` groups job parameters into blocks. The first block looks like this:

```
INPUT DATA FOR iwave

------------------------------------------------------------------------
FD:

         order = 2           spatial half-order
           cfl = 0.4         proportion of max dt/dx
          cmin = 1.0
          cmax = 4.5
          dmin = 0.5
          dmax = 5.0
         fpeak = 0.010       central frequency
```

Note that comments, block labels, and typographical separators are all accommodated. The IWAVE parameter parser identifies parameter specifications by strings of the form

```
          key = value
```

consisting of a string with no embedded whitespace, followed by an `=` sign surrounded by any amount of whitespace on either side, followed by another string with no embedded whitespace. Strings with embedded whitespace are also allowed, provided that they are double-quoted - thus `"this is a value"` is a legitimate value expression. Other capabilities of the parser are described in its html documentation. All values are first read as strings, then converted to other types as required.

The parameters appearing in `parfile` are as follows:

- `order = 2`: half-order of the spatial difference scheme - `asg` implements schemes of order 2 in time, and 2k in space, for certain values of k, the spatial half-order, which is the value associated to the key `order`. Permissible values of `order` in the current release are 1, 2, and 4.

- `cfl = 0.4`: max time step is computed using one osf several criteria - see html docs for details. This number is the fraction of the max step used. Must lie between 0.0 and 1.0.

- `cmin, cmax, dmin, dmax`: sanity checks on density and velocity values. The max permitted velocity `cmax` also figures in two of the max time step criteria. Violation of these bounds causes an informative error message with traceback information to be written to the output file `cout[rk].txt`, where `rk` is the MPI global rank (= 0 for serial execution), and the program to exit. IWAVE handles all trappable fatal errors in this way.

- `fpeak = 0.010`: nominal central frequency, in kHz. Used in two ways: (1) to set the width of absorbing boundary layers by defining a wavelength at max velocity `cmax`, and (2) as the center frequency of a Ricker wavelet in case the point source Ricker option is chosen for source generation. Plays no other role.

- `nl1, nr1, nl2, nr2`: specify 2D PML layer thicknesses: `nl1` describes the layer thickness, in wavelengths determined as described in the preceding bullet, of the *left* boundary (with lesser coordinate) in the axis 1 direction, etc. Set = 0.0 for no layer, in which case the free (pressure-release) boundary condition is applied.

- `srctype = point`: this application implements two source representations, a point source with amplitude options and a very flexible array source option.

- `"source = ...."`: a quoted parameter spec is just a string, from the IWAVE parser's point of view, so does not define anything: this parameter is commented out. If it were not quoted, it would define the pathname to an SU file containing source data - either a wavelet (first trace) for point source, or an array source specified by a number of traces (if `srctype=array`). If a source wavelet is not specified (as it is not for the scripted examples), the application creates a Ricker wavelet of central frequency `fpeak`.

- `sampord = 0`: order of spatial interpolation. Legal values are 0 and 1. 0 signifies rounding down the source coordinates to the nearest gridpoint with smaller coordinates. 1 signifies piecewise multilinear interpolation (or adjoint interpolation, for the source), so that a point source at $\mathbf{x}_s$ is represented as a convex linear combination of point sources at the corners of the grid cube in which $\mathbf{x}_s$ lies, and receiver values are similar convex combinations of nearby grid function values. The first option is appropriate for synthetic examples in which sources and receivers lie on the grid. The second permits arbitrary placement of sources and receivers, and is compatible with the overall second-order accuracy of `asg`.

- `refdist = 1000.0, refamp = 1.0`: point source calibration rule developed for the SEAM project - the wavelet is adjusted to produce the pulse shape read from the file specified by the `source` parameter, or a Ricker wavelet of central frequency `fpeak` if the `source` parameter is not assigned a value, with amplitude (in GPa) given by `refamp` at the prescribed distance (in m) given by `refdist`, assuming a homogenous medium with parameters the same as those at the source point, and absorbing boundary conditions. If `refdist` is set to 0.0, then source pulse (either read from a file, if `source` is set, or a Ricker of peak frequency `fpeak` otherwise) is simply used as the time function in the discrete point source radiator.

- `hdrfile = ...`: IWAVE specifies acquisition parameters such as source and receiver locations, time sample rates and delays, and so on, by supplying trace headers in a file: the traces produced in simulation have the same headers. At present, the only implemented option for specifying headers is via a path to an SU file, that is, a SEGY-formatted file with reel header stripped off. Other options are planned for future releases.

- `datafile = ...`: pathname for output data file; on normal completion of run, contains traces with same headers as in `hdrfile`, computed trace samples. Note that sample rate of output traces is whatever is specified in `hdrfile`, and generally is not the same as the time step used in the simulation, the trace samples being resampled on output. Note also that pathnames may be either fully qualified (as in the `hdrfile` entry) or relative.

- `velocity = ..., density = ...`: pathnames to rsf header files for velocity and density. Other combinations of physical parameters are admissible, such as bulk modulus and density, bulk modulus and bouyancy (reciprocal density), velocity and buoyancy. Data stored in RSF disk format, described in Madagascar web documentation. Current proxy for unit conversions: scaling during read/write by power of 10, given by `scale` keyword (extension to standard RSF). Must be chosen so that output is in m/ms or km/s for velocity, g/cm^3 for density, or compatible units for other parameters (eg. GPa for bulk modulus).

- `mpi_np1 = ..., partask = 1`: parallelism parameters - `mpi_np1` gives the number of domains along axis 1, etc. (loop or domain decomposition), and `partask` gives the number of shots to load simultaneously (task parallelization over shots). Domain decomposition and task parallelization may be used alone or in combination. Setting the value = 1 for all of these parameters signifies serial execution, even if the code is compiled with MPI. To execute in parallel, compilation with MPI is a precondition - see installation instructions.

- `dump_pi = ...`: dump parameters regulate verbosity, with output being sent to text files `cout0.txt` (serial) or `cout[rk].txt`, `rk` = MPI global rank encoded with uniform field width for parallel execution. Individual parameters described in the html documentation. If all dump parameters are set to zero, `asg` is silent, i.e. all `cout...` files will be empty on completion.

The parameters described here represent one common use case of IWAVE's acoustic application. The web documentation describes a number of other use cases.

DOWNLOADING AND INSTALLING IWAVE

Download and installation instructions are available on the IWAVE web site (Terentyev et al., 2012). In brief,

- The primary source for IWAVE is the SourceForge Subversion repository for Madagascar. To download IWAVE alone,

  ```
  svn co http://svn.code.sf.net/p/rsf/code/trunk/iwave $TOP
  ```

  where `$TOP` is the full or relative pathname under which you wish IWAVE source to be installed. To download the entire Madagascar package (development version), simply leave off `/iwave` in the above URL.

- if a repository download is not possible for some reason, gzipped tar files are available. For the latest development version of IWAVE only, download via the link

  ```
  http://www.trip.caam.rice.edu/software/iwave-dev.tar.gz
  ```

  The Madagascar web site includes a link for download of the latest stable release of Madagascar as a gzipped tar archive. At some point this stable release will include IWAVE.

- to install with default options,
  - `cd $TOP`
  - `scons`

- to install with more agressively optimized compilation, create a configuration file per instructions in (Terentyev et al., 2012), and recompile.