*Wave Inversion Technology, Report No. 1, pages 173-**??***

Strategies for 3D travel time computation

A. Leidenfrost and D. Gajewski[1]

keywords: *travel time*

ABSTRACT

Based on an already made examination of a number of 2D methods we choose to investigate the applicability of an FD eikonal solver to 3D problems. The properties that we focus on comprise not only accuracy but also memory consumption and - very important for large problems - computational time. A gradient model and a two-layers model serve as test cases and results are compared to results from the graph method. The comparison shows that the already in 2D notable gap in terms of computational time between the FD eikonal solver and the graph method is much bigger in 3D. Further, the accuracy of the FD eikonal solver is good. The only drawback with respect to the method is the in 3D smaller maximum velocity contrast that can be handled.

INTRODUCTION

The share of 3D problems in today's applied seismics is growing fast. This is due to rising computational power on the one hand and generally higher expressiveness of 3D data sets in comparison to 2D ones on the other hand. Consequently, we extended our investigations on 2D travel time computation as presented by (Leidenfrost et al., 1996) to 3D.

Despite of faster CPUs and growing storage capacities 3D problems are still a challenge mainly because of the sheer amount of data that has to be handled. As an illustration, consider the Marmousi data set (Versteeg and Grau, 1991) sampled with 10 m grid spacing giving 736×300 data points or about 0.8 MB (assuming 4-byte-floats). Extruding the flat model into a box with the second horizontal edge length matching the depth would yield a data set of about 250 MB! Please note that this is only the memory required for holding the input data, the computed travel times fill another array of the same size.

METHOD SELECTION

Consequently, for a 3D travel time calculation algorithm it is not sufficient to be only fast and accurate. In fact, it is also worth examining a method's capabilities of working

[1]**email:** leidenfrost@dkrz.de

the data in portions to save core memory and/or to divide the whole task into several independent threads which can then be distributed among a number of machines, e.g., using the MPI-protocol (Uni, 1995)

When reviewing the methods investigated by (Leidenfrost et al., 1996) with respect to their applicability to 3D problems, one should take into account the aforementioned qualities. The methods can be divided into two groups: One containing methods which follow a rigid scheme in which each point is timed exactly once and the other group containing more flexible methods that take into account the model structure. The FD eikonal solvers both in Cartesian (Vidale, 1988, 1990) and in polar (Leidenfrost et al., 1996) coordinates belong to the former group, while the latter consists of an extended and parallelized version of Vidale's algorithm (Podvin and Lecomte, 1991), the graph method of (Moser, 1991) in an improved and accelerated version by (Klimes and Kvasnicka, 1994) and wavefront construction (Vinje et al., 1993) as implemented by (Ettrich and Gajewski, 1995).

As a consequence of their rigid calculation scheme, the first group's methods can relative easily be divided into several tasks. Further, the model can be portioned such that only the region where the calculation actually takes place must be held in core memory. However, the, e.g., in (Vidale, 1988) described restrictions, i.e., velocity contrasts must not exceed $1 : \sqrt{2}$, intensify to a maximum contrast of $1 : \sqrt{3}/\sqrt{2} \approx 1.22$ in 3D. Finally, the FD eikonal solvers promise to be comparably fast as well in 3D as they are in 2D, although the algorithms in 2D and 3D differ in many aspects. The methods of the second group can neither be (easily) threaded nor are they well suited for holding only parts of the model or time array in core memory. On the other hand, they do not have any restrictions concerning model complexity, which makes them interesting mainly for complicated subsurface structures.

RESULTS

As in 3D much weight has to be put on speed and economy in terms of memory requirements, we here investigate the FD eikonal solver as described in (Vidale, 1990) in our own implementation. We compare it against one method of the other group, the graph method as published by (Klimes and Kvasnicka, 1994), as it has proven to be reliable and because it contains an error estimation for the calculated travel times.

Two 3D models serve as test cases, both consisting of $101 \times 101 \times 101$ samples with a grid spacing of 10 m in each direction, thus giving a cube with an edge length of 1 km. The first model is a gradient model of the form: $v(z) = v(0) + b \cdot z$, $v(0) = 2$ km/s, $b = 1$, z [km], i.e., a velocity of 2 km/s at the top and 3 km/s at the model bottom. The second is a two-layers model with a horizontal interface at $z = 310$ m and velocities $v_1 = 3000$ m/s above and $v_2 = 3600$ m/s below the interface. In both cases, the source is located at position $(0, 0, 0)$.

The FD eikonal solver is with 13.7 s CPU time (taken as u-time from the time command's output) about two orders of magnitude faster than the graph method. Accuracy is also quite excellent. The absolute value of the maximum relative error to the respective analytical solution of the FD eikonal solver is for both models slightly above 6%. But this is restricted to a very limited region near the source and is due to the rather small initialization zone of $3 \times 3 \times 3$ grid points around the source. The average relative error of 0.03%, however, indicates that the method is generally very accurate. Although in the models used here the average error of the FD eikonal solver is smaller than the one of the graph method, the former is expected to be outperformed by the latter in terms of accuracy when switching to more complex models with stronger velocity contrasts, where the eikonal solver's algorithm breaks down and replacement formulas have to be used. All numbers were taken on a Pentium 133 MHz machine.

ACKNOWLEDGEMENTS

The authors are indebted to Torsten Graaf, Hamburg University, Germany for providing the 3D analytical solution for the gradient model.

REFERENCES

Ettrich, N., and Gajewski, D., 1995, Efficient prestack Kirchhoff migration using wavefront construction: 57th EAGE Conference & Technical Exhibition, Extended Abstracts.

Klimes, L., and Kvasnicka, M., 1994, 3-D network ray tracing: Geophysical Journal International, **116**, 726–738.

Leidenfrost, A., Gajewski, D., and Ettrich, N., 1996, Strategies for 3D traveltime computation: 58th EAGE Conference & Technical Exhibition, Extended Abstracts.

Moser, T. J., 1991, Shortest path calculation of seismic rays: Geophysics, **56**, 59–67.

Podvin, P., and Lecomte, I., 1991, Finite difference computation of traveltimes in very contrasted velocity models: A massively parallel approach and its associated tools: Geophysical Journal International, **105**, 271–284.

University of Tennessee. **MPI: A MESSAGE-PASSING INTERFACE STANDARD**, 1995. Message Passing Interface Forum.

Versteeg, R., and Grau, G., Eds. **THE MARMOUSI EXPERIENCE**, Proceedings of 1990 EAEG Workshop on Practical Aspects of Seismic Data Inversion, 1991.

Vidale, J., 1988, Finite-difference calculation of travel times: Bulletin of the Seismological Society of America, **78**, 2026–2076.

Vidale, J., 1990, Finite-difference calculation of traveltimes in three dimensions: Geophysics, **55**, 521–526.

Vinje, V., Iversen, E., and Gjøstdal, H., 1993, Traveltime and amplitude estimation using wavefront construction: Geophysics, **58**, 1157–1166.