JOURNAL OF SEISMIC EXPLORATION **30**, 447-453 (2021) 447

TWO FACTORS AFFECTING THE SPEED OF INTERPOLATION WITHIN RAY CELLS

PETR BULANT

Department of Geophysics, Faculty of Mathematics and Physics, Charles University, Ke Karlovu 3, 121 16 Praha 2, Czech Republic. bulant@seis.karlov.mff.cuni.cz

(Received January 2, 2021; accepted June 25, 2021)

ABSTRACT

Bulant, P., 2021. Two factors affecting the speed of interpolation within ray cells. *Journal of Seismic Exploration*, 30: 447-453.

This short study is devoted to further investigation of the interpolation method by Bulant and Klimeš. The method was designed to cover the model volume by prismatic ray cells formed by six points on three rays forming the ray tube, but enabled also the interpolation within degenerate ray cells formed by five or four points on the rays. Some researchers including consortium members were curious about the numerical efficiency of the proposed algorithm based on prismatic ray cells. In the first part of this study we thus compare the CPU time requirements of the interpolation within prismatic cells and within tetrahedral ray cells, and we conclude that computational time is not a criterion, according which one of the two methods is preferable in general. Then the method by Bulant and Klimeš offers bilinear interpolation scheme, and more precise bicubic interpolation scheme. In the second part of this short study we answer the question whether the bicubic interpolation is time consuming or not, and we conclude that this is not the case, and that it should be used whenever possible as it offers much higher accuracy compared to bilinear interpolation.

KEYWORDS: ray theory, traveltime, ray tracing, interpolation of the Green function.

INTRODUCTION

The method of calculating of a set of seismic rays from given source under given initial conditions is called *initial-value ray tracing* (Červený et al., 1988). Once the set of rays is calculated, the traveltime, amplitudes, and

0963-0651/21/$5.00

other quantities are known in the points stored along the rays. If we wish to know these quantities in other points of the model volume, we need to interpolate between the rays.

Several interpolation methods based on decomposition of the model volume into ray tubes, their further decomposition into the ray cells, and on further interpolation within individual ray cells were introduced. In the *controlled initial–value ray tracing* (Bulant, 1999), the model volume is decomposed into ray tubes formed by three rays, and each ray cell corresponds to the space in the ray tube limited by two planes which approximate wave fronts or structural interfaces. Simple bilinear scheme may then be used to interpolate the quantities inside the ray cell from their values stored at the vertices of the ray cell. If the partial derivatives are known in addition to the functional values at the vertices of the ray cell, the bicubic interpolation scheme may be used to increase the accuracy of the interpolation (Bulant and Klimeš, 1999).

The above-mentioned method is thus based on decomposition of ray tubes into prismatic ray cells, i.e., the cells formed by six points on three rays. If the rays interact with structural interfaces, it is sometimes not possible to create such cells, and one to several degenerate cells, which are formed by five or four points, must be generated. As was shown in Bulant and Klimeš (1999), computation of local coordinates within standard prismatic cells formed by six points leads to the solution of a cubic equation, while within degenerate cells formed by five points it leads to a quadratic equation, and finally within cells formed by four points local coordinates may be computed by solving a linear equation. As the solution of cubic equation is more time-consuming than the solution of linear equation, it might be interesting to divide ray tubes into ray cells formed by four points (tetrahedra) and gain from the simplification of the equations. In the first part of this paper we discuss this possibility and compare the required computational time needed for the interpolation within the prismatic ray cells and within the tetrahedral cells.

The accuracy of the bicubic interpolation scheme and its comparison with the bilinear scheme was studied by Bulant and Klimeš (1999). It was shown that the accuracy of the bicubic scheme is much higher than that of the bilinear scheme. On the other hand, it is clear that the bicubic scheme will require more computational time. In the second part of this paper, we compare the two interpolation schemes with respect to the required computational time in order to find out whether the bicubic interpolation is time consuming or not.

COMPARISON OF THE SPEED OF INTERPOLATION WITHIN PRISMATIC VERSUS TETRAHEDRAL RAY CELLS

The method for interpolation of ray-theory travel times in nodes of 3D grids presented by Bulant and Klimeš (1999) is based on three successive steps. Those are decomposition of ray tubes into ray cells, determination of the target gridpoints located within individual cells and calculation of their local coordinates, and the interpolation of traveltime to the determined gridpoints. As was explained in Introduction, computation of local coordinates within standard prismatic cells formed by six points on the rays leads to the solution of a cubic equation, while within degenerate cells formed by four points local coordinates may be computed by solving a linear equation. It might thus be interesting to divide ray tubes into tetrahedral ray cells and gain from the simplification of the equations. This may be done very easy by splitting each cell with five or six vertices into two or three tetrahedra, see Fig. 1. The sides of tetrahedra for two neighbouring ray tubes must coincide. This condition is satisfied, e.g., when we create first tetrahedron from the bottom of the ray cell along the ray with the highest index, and the third tetrahedron from the top of the ray cell along the ray with the lowest index from the indices of the three rays forming the original prismatic cell.

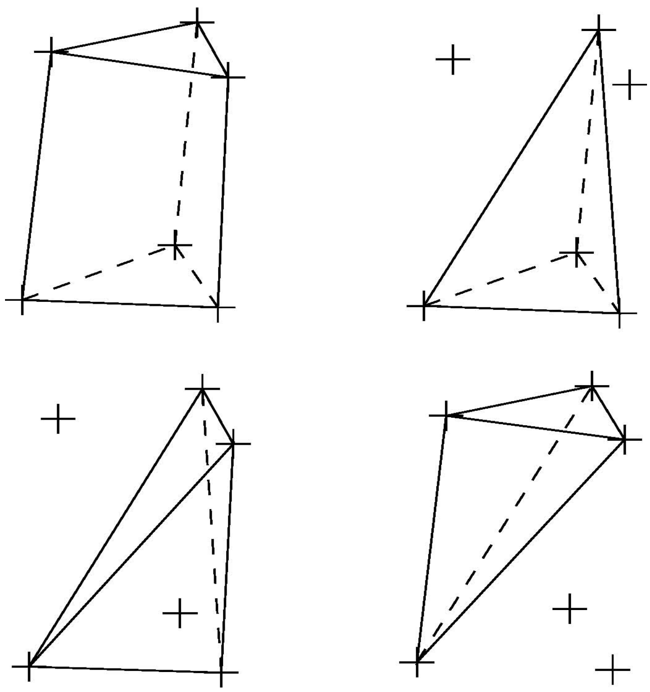

Fig. 1. Prismatic ray cell decomposed into three tetrahedral ray cells.

Due to the simplification of the equations for local coordinates we could expect, that the interpolation within tetrahedra will be faster.

On the other hand, we know that all the receivers situated inside the smallest box containing the whole ray cell enter into the computation of the local coordinates. The above mentioned boxes are about the same for all the three tetrahedra and for the original prismatic ray cell containing the tetrahedra. Thus, if we use tetrahedra, we will have to solve the equations for local coordinates for about three times more points of the target grid than in the case of prismatic cells. This will slow down the tetrahedra computation more in case of irregular (e.g., long and thin) ray cell, and less in case of regular cell, when more of the receivers from the above mentioned box are located within the cell.

Moreover, we know that special care must be taken of the receivers located at the sides of ray cells. The use of tetrahedra means also increased number of sides of the cells and thus more receivers located at the sides.

As the final effect of the use of tetrahedra instead of prismatic cells is not clear, we decided to compare the methods numerically. We took the code for interpolation within prismatic cells, changed the decomposition of ray tubes into tetrahedra instead of former decomposition into prismatic cells, and removed all operations connected with quadratic and cubic equations for computation of local coordinates within prismatic ray cells. The code for final interpolation of travel times and other quantities in tetrahedra remained the same as in prismatic cells, the interpolation is thus again trilinear, but with two pairs of coinciding points. The results of the interpolation are thus within numerical errors the same for both kinds of ray cells. Then we tested both the methods in model with lenticular inclusion and in model “98” of the package DATA (Bucha and Bulant, 2019).

The computational time of interpolation in model 98 was 8 minutes 34 seconds for tetrahedra method and 6 minutes 11 seconds for prismatic cells method.

The computational time in model with lenticular inclusion was 1 minute 52 seconds for tetrahedra and 2 minutes 36 seconds for prismatic cells.

In the next numerical test carried out again in the model with lenticular inclusion we tested the influence of the shape of ray cells on the computational time. The length of the cells was managed by parameter which describes the time interval for storing points along the rays. Note that typical velocity in the model is 5 km/s. The width of the cells was influenced by parameter which describes the maximum width of the cells in the ray-tube metric. Three computations were realized, see Table 1.

Table 1. Computational times in model with lenticular inclusion in dependence on the shape of the ray cells. We can see that the computational times for long and narrow cells (first line) and for short and wide cells (third line) are longer in comparisson with optimal ray cells (second line).

Length of the cells:	Maximum width of the cells:	Computational time for prismatic cells method:	Computational time for tetrahedra cells method:
1.000 sec	10. km	156 sec	112 sec
0.175 sec	20. km	105 sec	101 sec
0.084 sec	40. km	122 sec	139 sec

We can see that the interpolation within tetrahedral cells may be sometimes slower and sometimes faster than the interpolation within prismatic cells. The computational time is mostly affected by the shape of the ray cells.

COMPARISON OF THE SPEED OF BICUBIC INTERPOLATION VERSUS BILINEAR INTERPOLATION

As we already mentioned, the method by Bulant and Klimeš (1999) is based on three successive steps: the decomposition of ray tubes into ray cells, the determination of the gridpoints located within individual cells, and the interpolation to the determined gridpoints. As the partial derivatives of travel times are known at the vertices of the ray cell, two interpolation schemes may be used for the interpolation of traveltimes. Those are the bilinear interpolation scheme and the bicubic interpolation scheme.

When we look at the equations for the determination of the gridpoints located within individual ray cells (Bulant and Klimeš, 1999, Chapter 3), and at the equations for the interpolation (Bulant and Klimeš, 1999, Chapter 4), the interpolation does not seem to require many numerical operations compared to the determination. Moreover, the determination must be done for more gridpoints than the interpolation, and some additional computational time is also required by the decomposition of ray tubes into ray cells. We could thus assume that the bicubic interpolation will not influence the computational time significantly.

To prove these considerations, we have measured the computational time required for the two interpolation schemes. We used an example of the interpolation in model '98', described by Bulant and Klimeš (1998). The results are summarized in Table 2.

Table 2. Computational times in model '98' for bicubic and bilinear interpolation schemes. The computational time for bicubic interpolation is slighly longer, but the difference is negligible compared to the computational time of the whole code.

Kind of interpolation scheme:	Bicubic	Bilinear
Execution time to run the whole code:	4 min 33.98 s	4 min 33.79 s
Execution time of the determination of gridpoints located within the cells:	2 min 42.04 s	2 min 41.85 s
Execution time of the interpolation:	3.87 s	3.68 s

We can see that the difference between the computational time required for the bicubic interpolation and the computational time required for the bilinear interpolation is negligible with respect to the computational time required to run the whole code. This conclusion has been numerically verified only for the presented computation, but as it is in accordance with our theoretical estimates, we believe that it holds in general.

CONCLUSIONS

The bicubic interpolation scheme used in Bulant and Klimeš (1999) is not time consuming in comparison with the bilinear interpolation scheme. As it is much more accurate (Bulant and Klimeš, 1999, Fig. 8), it should be used whenever possible.

Using the tetrahedral ray cells may be sometimes slower and sometimes faster than using the prismatic ray cells. Computational time is not a criterion, according which one of the two methods is preferable in general. It is much more affected by the shape of the ray cells.

ACKNOWLEDGEMENTS

The research has been supported by the Czech Science Foundation under contract 20-06887S, and by the members of the consortium "Seismic Waves in Complex 3-D Structures" (see "http://sw3d.cz").

REFERENCES

Bucha, V. and Bulant, P. (Eds.), 2019. SW3D-CD-23 (DVD-ROM). Seismic Waves in Complex 3-D Structures, 29: 71-72, online at "http://sw3d.cz".

Bulant, P., 1999. Two-point ray-tracing and controlled initial-value ray-tracing in 3-D heterogeneous block structures. J. Seismic Explor., 8: 57-75.

Bulant, P. and Klimeš, L., 1998. Computations in the model composed during the 1998 consortium meeting. Seismic Waves in Complex 3-D Structures, 7: 33-56, online at "http://sw3d.cz".

Bulant, P. and Klimeš, L., 1999. Interpolation of ray theory traveltimes within ray cells. Geophys. J. Internat., 139: 273-282.

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