Stanford Exploration Project, Report 70, December 18, 1997, pages 1–43

3-D depth migration by a predictor-corrector method

Dave Nichols[1]

keywords: *algorithm, approximation, three-dimensional, conjugate gradient*

ABSTRACT

3-D depth migration can be cast as an explicit or implicit finite difference problem. The explicit problem is cheap to solve, but unstable. The stable implicit problem can be solved by direct or iterative methods. Direct methods are expensive and iterative methods are slow to converge at low frequencies. I experiment with the use of a predictor-corrector method, using iterative refinement of an approximate solution to the implicit problem. The approximate solution is obtained by solving the explicit problem. The iterative, conjugate gradient method is still slow to converge at low frequencies. A preconditioned conjugate gradient method would converge faster.

INTRODUCTION

Depth extrapolation of the one-way wave equation in three dimensions can be solved using a finite difference method in the (ω, x, y, z) domain (Claerbout, 1985). The implicit formulation of this problem produces a pentadiagonal system of equations. The direct solution of a pentadiagonal system is an expensive process. If we consider an $n \times n$ grid of data the cost for the simplest methods is $O(n^2)$ operations; whereas, for methods that are efficient on vector or parallel machines the cost is $O(n^2 \log n)$. This is thought to be impractical for seismic applications. A common approximation used in solving this problem is called "splitting," resulting in $2n$ tridiagonal systems. There are several $O(n)$ algorithms for solving tridiagonal systems that can be efficiently implemented on many different computer architectures. Unfortunately, for practical depth step sizes, the splitting approximation results in anisotropic wave propagation, the waves propagating along the axes of the grid are faster than those at 45° to the grid.

There have been several methods proposed to overcome these problems. Hale (1990) uses a stable explicit scheme and a McLellan transform to downward continue the wavefield. However, this method requires long explicit operators to ensure

[1]**email:** not available

stability. These operators are expensive to apply and may have a smoothing effect in the presence of very sharp lateral velocity variations. Cole (1989) used iterative methods to solve the pentadiagonal system of equations. He compared the Gauss-Seidel and Jacobi methods and found that these methods did not converge well at low frequencies, and in some cases became unstable.

In this paper I propose the use of the conjugate-gradient (CG) method (Hestenes & Stiefel, 1952) to overcome the stability problems, however, the CG method suffers from the same problem as the other iterative methods, convergence is slow at low frequencies. Because the cost of running the conjugate gradient method to completion would be prohibitive, I wish to modify the problem so that it converges in a few iterations. This can be done in two ways, either by preconditioning, or starting the algorithm with an estimate that is close to the correct solution. The second method is easy to implement, the initial solution is produced by a cheap explicit algorithm. The potential instability of the explicit method is not a problem because it is only used to generate an initial estimate. The solution at each depth step is generated by solving the implicit equations that are known to be stable.

The CG algorithm requires repeated application of the Laplacian operator. This can be implemented efficiently on mesh connected parallel computers. My program is implemented purely in terms of Laplacian operators and fully parallel multiplications and additions; it will run efficiently when the problem is scaled to a larger size.

DOWNWARD CONTINUATION IN THREE DIMENSIONS

Zero-offset migration is usually implemented as a downward continuation operation. The exploding reflector model is used to form an image of the subsurface. The wavefield at the surface is extrapolated downwards in depth using a one-way wave equation and the zero-time slice is extracted at each depth to form an image of the reflectivity as a function of depth.

The derivation of the method starts with the scalar wave equation for constant velocity media,

$$\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} + \frac{\partial^2 P}{\partial z^2} = \frac{1}{v^2}\frac{\partial^2 P}{\partial t^2}\,.$$

or its equivalent in the Fourier domain

$$k_x^2 + k_y^2 + k_z^2 P = \frac{\omega^2}{v^2} P\,.$$

The one-way wave equation is obtained by solving for k_z or its space-domain equivalent, $\partial/\partial z$:

$$\frac{\partial P}{\partial z} = -i\frac{\omega}{v}\sqrt{1 + \frac{v^2}{\omega^2}\nabla_{xy}^2}P\,. \tag{1}$$

Here, ∇^2_{xy} represents the 2-D Laplacian operator in x and y.

To perform migration we wish to extrapolate the wavefield at some depth z to another depth $z + \delta z$. There are two main classes of finite difference scheme that are used to do this: Explicit and Implicit schemes.

Explicit schemes

Depth extrapolation is performed using the equation,

$$P(z + \delta z) = P(z) + \left.\frac{\partial P}{\partial z}\right|_z \delta z\,.$$

The square root in equation 1 can be approximated by a truncated series expansion. e.g.

$$\left.\frac{\partial P}{\partial z}\right|_z \quad = \quad -i\frac{\omega}{v}\left(1 + \frac{1}{2}\frac{v^2}{\omega^2}\nabla^2_{xy} - \frac{1}{8}\frac{v^2}{\omega^2}(\nabla^2_{xy})^2 + \cdots\right) P(z)\,. \tag{2}$$

Explicit schemes are very simple to implement. On a regular grid the Laplacian operator can be implemented as a convolutional operator. This is very efficient on a mesh connected multiprocessor such as the Connection Machine.

The problem with explicit schemes is that, in general, they are unstable. Hale (1990) has demonstrated that unconditionally stable explicit operators can be created; however, these operators tend to be long so the convolution is expensive.

Implicit schemes

Most implicit schemes are based on a Crank-Nicholson formulation of the extrapolation problem. This method calculates a centered finite difference approximation to the z-derivative and averages the right hand side of the equation over the two depths:

$$\frac{p_{z+\delta z} - p_z}{\delta z} \quad = \quad -i\frac{\omega}{v}\sqrt{1 + \frac{v^2}{\omega^2}\nabla^2_{xy}}\left(\frac{p_{z+\delta z} + p_z}{2}\right)\,.$$

The square root is usually approximated by a continued fractions approximation. The standard 45° approximation gives the following equations in retarded coordinates:

$$\frac{q_{z+\delta z} - q_z}{\delta z} \quad = \quad \frac{-i\frac{v}{2\omega}\nabla^2_{xy}}{1 + \frac{v}{2\omega}^2\nabla^2_{xy}}\left(\frac{q_{z+\delta z} + q_z}{2}\right)\,.$$

When terms in q_z and $q_{z+\delta z}$ are gathered together, we obtain the following equation.

$$\left[1 + \left(\frac{v}{2\omega}^2 + i\frac{v}{2\omega}\delta z\right)\nabla^2_{xy}\right] q_{z+\delta z} = \left[1 + \left(\frac{v}{2\omega}^2 - i\frac{v}{2\omega}\delta z\right)\nabla^2_{xy}\right] q_z\,.$$

This problem is now in the form of a set of linear equations $Ax = y$. If the second-order-in-space finite difference operator is used to evaluate the Laplacian, the matrix A is pentadiagonal. Direct methods, such as Gaussian elimination, may be used to solve the system. For the particular structure of the system a block-Gaussian-elimination scheme may be used. For an $n \times n$ grid of data in x and y the operation count for the direct solution is $O(n^2)$. However, this method does not vectorize or parallelize well because of recursions in the algorithm. Methods that are more suitable for vectorization, such as block-cyclic-reduction, have a higher operation count, $O(n^2 \log n)$. These costs are usually considered too high for routine application in seismic data processing. If a more accurate Laplacian operator is used the operator A will have a more thickly-banded structure and the cost of the direct solution will be higher.

Because of the high cost, this system is not usually solved in 3-D migration algorithms. Instead, a "splitting" approximation is made and the equations are rewritten in an approximate form that produces two sets of tridiagonal equations (Claerbout, 1985). However, the splitting approximation is not an isotropic approximation. If a time slice is made of the impulse response of the migration algorithm the wavefront is not circular. Waves propagating at 45° to the computational grid travel slower than those along the axes of the grid. Hence, we would prefer to find a method that solves the pentadiagonal system.

Conjugate gradient implicit schemes

An alternative to using a direct method of solution of the pentadiagonal system is to use an iterative scheme. Cole (1989) investigated the use of Jacobi and Gauss-Seidel methods to solve the system. He found stability problems with these methods and that rates of convergence were very low for the low frequencies. The pentadiagonal system is less diagonally dominant at low frequencies. I decided to use the Conjugate Gradient method as it is guaranteed to converge for positive-definite operators. In order to use the CG method I must be able to apply the operator A and its conjugate. The operator and its conjugate are given by

$$A = \left[1 + \left(\frac{v}{2\omega}^2 + i \frac{v}{2\omega} \delta z \right) \nabla^2_{xy} \right] ,$$

and

$$A^* = \left[1 + \left(\frac{v}{2\omega}^2 - i \frac{v}{2\omega} \delta z \right) \nabla^2_{xy} \right] .$$

Application of these operators involves multiplication, addition, and use of the Laplacian operator. On a mesh-connected parallel computer all these operations can be implemented efficiently, making full use of all the processors. The algorithm is unchanged if I choose to use a more accurate Laplacian operator. The only difference is that a higher order Laplacian will produce a less diagonally dominant operator so the method will converge more slowly. The cost of applying the Laplacian operator

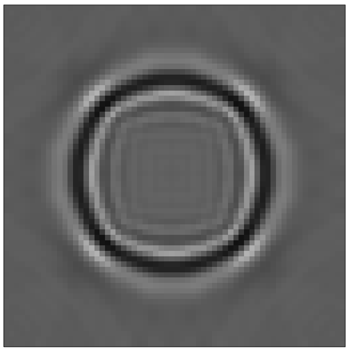
	Iterations	
frequency/Hz	First Order	Second Order
15	200	200
20	167	200
27	64	90
35	29	43
42	15	23
54	8	11
66	7	7
82	6	6
110	5	5

Figure 1: Number of iterations required for simple CG method to converge.

is $O(n^2)$. If the CG method is run to completion it will take n^2 iterations so the total cost will be $O(n^4)$. However, the aim is to find a set of equations that will converge to sufficient accuracy in less than the full number of iterations.

I implemented a simple CG method on the Connection Machine. I use the L_2 norm of the residuals as the indicator of convergence of the algorithm. If the ratio of the norm of the residual to the norm of the RHS of the equation was less than 10^{-5}, I stopped the algorithm. The test problem was a small 3-D grid. The input was 64 points in x and y with 64 time samples. The output had 50 depth samples. The second column in Figure 1 shows the number of iterations required to meet this criterion for a range of frequencies when using a Laplacian operator with three points in X and Y. The third column shows the results when using a Laplacian operator with five points in X and Y. An arbitrary limit of 200 iterations was imposed. The method converged poorly at low frequencies in both cases, with the higher order Laplacian producing a more thickly banded, less diagonally dominant, operator the convergence is even slower.

Figure **??** shows a time slice of the migration impulse response from the CG method. The impulse response is circular and does not suffer from the numerical anisotropy introduced by the splitting method.



Impulse response time slice

Figure 2: Impulse response of 45° implicit migration. $\boxed{\text{dave2-filtdata.slice}}$ [NR]

Predictor-corrector method

There are two methods that can be used to speed up a CG algorithm, the use of a preconditioning matrix and estimation of a better starting solution. The second method is the simplest to use. If some other method is used to estimate the solution of the linear equations, the CG algorithm can be used to refine the solution. The explicit method is an obvious candidate to produce an initial estimate of the solution. It is cheap and simple to implement. It does not matter that the explicit method can become unstable because it is only used to generate the initial solution. The final equations solved are those of the implicit method and these equations are known to be stable. Using an explicit solution as the input to an iterative solution of the implicit equations, is known as the "predictor-corrector" method. The table in Figure 5 shows the number of iteration required for the predictor-corrector method to converge. The explicit method used in this implementation is the one given by equation 2. The method converges in fewer iterations for most frequencies, but for the low frequencies it actually takes more iterations than the simple CG method. This implies that the explicit solution gave a worse estimate of the correct solution than a zero vector. This is a disappointing result as I had hoped that the predictor-corrector method would improve the convergence of the CG algorithm at low frequencies whereas the reverse seems to be true.

One possible reason for this can be seen by considering the impulse response of the migration in Figure **??**. Many of the lowest frequencies in this figure are in the top of the heart, the highest dips having the lowest frequency content. Since we know that this algorithm is inaccurate for dips above about 45°, we should not put a lot of effort into solving the implicit equation for dips where the result is known to be wrong. One simple way of avoiding this effect is to dip filter the input so that high dips are not present in the data. The data should at least be filtered so that no evanescent energy is present in the input section. I chose to filter the input spike to attenuate dips above 50°. The impulse response produced by migrating the filtered spike is shown in Figure **??**. The table in Figure 6 shows the number of iterations required for convergence when migrating the filtered spike. The predictor-corrector method is now uniformly better than the simple CG method. However, the decrease in the number of iterations is small. The main reason is that the explicit scheme is

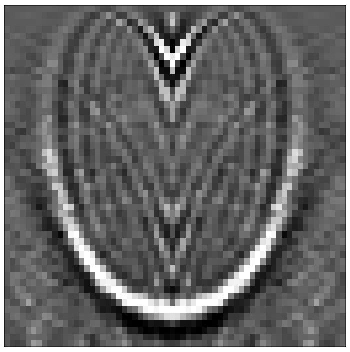


Figure 3: Impulse response of 45° implicit migration dave2-data.cut [NR]

inaccurate at low frequencies. I would prefer a predictor step that is accurate at low frequencies at the expense of high frequency accuracy. The high frequencies converge rapidly without a predictor step, it is the low frequencies that require acceleration.

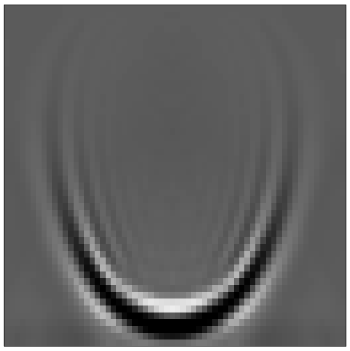


Figure 4: 45° implicit migration of a filtered spike dave2-filtdata.cut [NR]

Other options

In the future I intend to try other methods of accelerating the convergence of the CG method. The most common way of speeding up the CG algorithm is to use a preconditioning matrix. This is an operator, M, that is in some way approximately equivalent to the operator A but has an inverse that is easy to compute, (find M such that $AM^{-1} \approx I$). All that is required of AM^{-1} is that it has a better condition number than A. There are two forms for the operator that I intend to experiment with, the split form of the 3-D operator and an incomplete Cholesky decomposition of A.

The Cholesky decomposition of A is closely related to the work done by Cole (1991) on Given's rotation of the 3-D operator. The incomplete Cholesky decomposition is an approximate decomposition that has the same sparsity structure as the original operator and is thus cheap to compute and apply.

	Iterations	
frequency/Hz	First Order	Second Order
15	200	200
20	181	200
27	70	98
35	32	47
42	16	25
54	7	12
66	5	7
82	4	4
110	3	3

Figure 5: Number of iterations required for Predictor-Corrector method to converge.

frequency/Hz	Iterations: Conjugate Gradient	Iterations: Predictor Corrector
15	200	200
20	132	131
27	45	44
35	20	18
42	11	10
54	7	6
66	6	5
82	6	4
110	5	4

Figure 6: Number of iterations required for simple CG and predictor-corrector methods to converge when input is a dip-filtered spike. A first order Laplacian was used.

The preconditioned conjugate gradient method involves solving the equation $Mz = r$ at every iteration of the algorithm. However, the operator M does not change between iterations so that though solving $Mz = r$ may be expensive (e.g., by using the split form of the operator), the decomposed operator can be reused for every iteration.

CONCLUSIONS

The conjugate gradient method can be used to perform 3-D migration using thickly-banded operators. Convergence at low frequencies is poor and an attempt to use a predictor-corrector method to accelerate convergence was unsuccessful. I propose to next try the use of preconditioning schemes to accelerate convergence.

REFERENCES

Claerbout, J., 1985, Imaging the Earth's Interior, Blackwell Scientific Publications.

Cole, S., 1989, Iterative methods for three dimensional finite difference migration and modeling: SEP–**60**, 149–163.

Cole, S., 1991, Simplifying 3-D migration operators using Givens rotation: SEP–**70**, ???–???.

Hale, D., 1990, 3-D depth migration via McClellan transformations, SEG Expanded Abstracts, 1325-1328.

Hestenes, M.R., and Stiefel, E., 1952, Method of conjugate gradients for solving linear systems: J. Res. Nat. Bur. Stand. **49**, 409–436.