Stanford Exploration Project, Report 73, November 18, 1997, pages 351–366

Stanford Exploration Project, Report 73, November 18, 1997, pages 351–366

3-D and 2-D phase shift plus interpolation and split-step Fourier modeling

Alexander M. Popovici[1]

keywords: *modeling, algorithm, velocity, Fourier transform*

ABSTRACT

Migration algorithms based on Fourier methods are naturally parallel. The modeling algorithm which is the conjugate transpose of the migration algorithm (or vice-versa) preserves all the parallel features of the direct (migration) algorithm. I implemented two Fourier based modeling algorithms on the Connection Machine. Both methods are the conjugate transpose of two migration algorithms (Phase Shift Plus Interpolation and Split-Step). I tested the algorithms on 3-D and 2-D variable velocity models. I compare both methods considering speed and accuracy.

INTRODUCTION

Modeling algorithms in the frequency-wavenumber domain are special candidates for implementation on parallel computers as the frequency components are independent, allowing the computations to be done in parallel in surface coordinates. The three dimensional case offers a perfect mapping on the parallel computer memory, by storing the data in a serial fashion along the time dimension and in parallel on different processors in the surface coordinates. After Fourier transforming the data along the time axis, we can process each frequency independently, a slice at a time.

I selected two Fourier domain modeling algorithms which handle lateral velocity variations. The modeling algorithms are the conjugate transpose of the migration algorithms (Claerbout, 1992). The Phase Shift Plus Interpolation migration algorithm was presented by Gazdag and Sguazerro (1984) and the Split-step Fourier migration was presented by Stoffa et al. (1990). The two algorithms are extremely similar, and the three dimensional extension of the two dimensional version is quick and painless. I coded the modeling algorithms by transposing and conjugating the migration algorithms that I had at hand (Popovici, 1991). I wrote the parallel FORTRAN 90 version of the code to run on the Connection Machine. I compared the modeling results of the two programs on 2-D and 3-D synthetic models.

[1] **email:** not available

THE PHASE SHIFT PLUS INTERPOLATION MODELING

The derivation of the Phase Shift Plus Interpolation (PSPI) modeling algorithm starts with the scalar wave equation

$$\frac{\partial^2 p}{\partial z^2} + \frac{\partial^2 p}{\partial x^2} = \frac{1}{v^2}\frac{\partial^2 p}{\partial t^2} \tag{1}$$

where $p = p(x, z, t)$ is the pressure field and $v = v(x, z)$ is the earth velocity.

The pressure field $p(x, z, t)$ is a finite function and can be therefore expressed as a double Fourier series

$$p(x, z, t) = \sum_{k_x}\sum_{\omega} P(k_x, z, \omega) e^{i(k_x x - \omega t)}. \tag{2}$$

Substituting equation (2) in equation (1) we obtain

$$\sum_{k_x}\sum_{\omega}[\frac{\partial^2 P(k_x, z, \omega)}{\partial z^2} - k_x^2 P(k_x, z, \omega) + \frac{\omega^2}{v^2(x, z)} P(k_x, z, \omega)] e^{i(k_x x - \omega t)} = 0. \tag{3}$$

Equation (3) should hold for any values of k_x and ω. This is possible only if each term inside the square parenthesis is zero. This reasoning is similar to the condition that if a polynomial is zero for any values of x, the coefficients of the polynomial are zero. Equation (3) becomes

$$\frac{\partial^2 P(k_x, z, \omega)}{\partial z^2} = (k_x^2 - \frac{\omega^2}{v^2(x, z)}) P(k_x, z, \omega) \tag{4}$$

valid for all values of k_x and ω. The problem is that in this form, the x-coordinate in the pressure field is Fourier transformed and there is no direct correspondence between a point (x, z) in the medium, the velocity $v(x, z)$, and the corresponding value of $p(x, z, t)$ at that location.

For a constant velocity we write

$$k_z = \pm[\frac{\omega^2}{v^2} - k_x^2]^{\frac{1}{2}} \tag{5}$$

where k_z is constant for two given values of k_x and ω. Equation (5) is the well known **dispersion** relation. Equation (4) becomes an ordinary differential equation

$$\frac{\partial^2 P}{\partial z^2} = -k_z^2 P. \tag{6}$$

For a constant k_z equation (6) has the analytic solution

$$P(k_x, z_0 + z, \omega) = P(k_x, z_0, \omega) e^{ik_z z}. \tag{7}$$

Equation (7) shows how in constant velocity media we can find the pressure field at a certain depth level $P(k_x, z_0 + z, \omega)$, if we know it at any other depth level $P(k_x, z_0, \omega)$.

However there are several ambiguities in equation (7) that we need to discuss. One is the question of time propagation. If we know the pressure field (or wavefield) at a certain depth we can propagate it forward in time or backward in time. We can also propagate it up (along the z-axis) or down. To understand how we determine the propagation direction we have to analyze the values and sign of k_z. There are several restrictions on the values of k_z. Equation (6) has the solution (7) only for real values of k_z which imposes the condition

$$\frac{\omega^2}{v^2} - k_x^2 \geq 0.$$

The solution represented in equation (7) is the Fourier transform of the wavefield. The general solution in time-space coordinates is obtained by summing all the Fourier coefficients obtained from equation (7)

$$p(x, z, t) = \sum_{k_x} \sum_{\omega} P(k_x, z_0, \omega) e^{ik_z z} e^{i(k_x x - \omega t)}. \tag{8}$$

The function

$$e^{i(k_z z + k_x x - \omega t)}$$

represents a plane wave. Equation (8) sums many plane waves to obtain the general solution. We examine the sign of k_z necessary to upward propagate a wave by examining each plane wave solution. If we ignore $k_x x$, which determines the lateral variation, we can write

$$k_z z - \omega t = phase(z, t).$$

The phase is constant along a plane wave, and we write

$$k_z z = \omega t + const$$

for the phase of a particular plane wave. The plane wave is moving **downward** when k_z has the same sign with ω because z increases with t in order to keep the phase constant. So for the **upward** moving waves we need to have opposite signs of k_z and ω (z is decreasing when t is increasing). We have now figured out that in order to have only **upgoing** waves we have to look at the sign of ω and assign to k_z the opposite sign.

Next we correlate this information with the depth level where we want to find the wavefield. If the known wavefield is at depth z_0 and we want to find the wavefield at depth $z_0 + z$, then we have to propagate the wavefield back in time (toward $t = 0$) because we know that the wavefield travels upward. If the known wavefield is at depth z_0 and we want to find the wavefield at depth $z_0 - z$, then we are propagating the wavefield forward in time. This is the direction we are interested in for modeling.

However for depth varying velocity $v(z)$ we have k_z approximately constant only for small depth intervals (Δz) where we can consider the velocity constant. Therefore equation (7) becomes

$$P(k_x, z_0 + \Delta z, \omega) = P(k_x, z_0, \omega) e^{ik_z \Delta z} \tag{9}$$

and can be used to downward or upward extrapolate the wave field for a small depth interval.

For laterally variant media, Gazdag and Sguazzero (1984) propose to downward extrapolate the wavefield one depth interval at a time with several velocities. We can apply the same idea to upward propagate the wavefield with several velocities. We consider several velocities $(v_1, v_2, ...)$ in the interval $[v_{min}, v_{max}]$ and upward propagate the wavefield $P(k_x, z_0, \omega)$ to $P(k_x, z_0 - \Delta z, \omega)$ with each velocity. We can afterward inverse Fourier transform in x the resulting wavefields

$$(P_1(k_x, z_0 - \Delta z, \omega), P_2(k_x, z_0 - \Delta z, \omega), ...)$$

to obtain

$$(P_1^*(x, z_0 - \Delta z, \omega), P_2^*(x, z_0 - \Delta z, \omega), ...).$$

To obtain a single upward propagated wavefield, at each point $(x, z_0 - \Delta z)$ associated with a velocity $v(x, z_0 - \Delta z)$, the value of the resulting wavefield in this point is interpolated between the two wavefields with closest velocities $(v_i(x, z_0 - \Delta z))$.

In addition we need to implement a technique (Gazdag and Sguazzero, 1984) in the PSPI modeling algorithm to ensure that all the zero dips (corresponding to the case $k_x = 0$) are upward propagated without distortion. The technique consists of multiplying the wavefield $P(x, z_0, \omega)$ with

$$e^{-i \frac{\omega}{v_j} \Delta z} \tag{10}$$

after the Fourier transformation in x and multiplying the upward propagated wavefield $P_j(k_x, z - \Delta z, \omega)$ by

$$e^{i \frac{\omega}{v(x,z)} \Delta z} \tag{11}$$

where the subscript j denotes the index of the constant velocity used in the upward propagation step. For a zero dip plane wave (i.e. $k_x = 0$) we have

$$k_z = \pm \frac{\omega}{v} \tag{12}$$

and the plane wave can be upward propagated with the velocity $v(x, z)$ corresponding to the point coordinates (x, z) instead of the constant velocity v_j.

As it will be seen later when the PSPI results are compared to the Split-step Fourier results, this supplemental phase subtraction and addition improves the accuracy in the migration case but actually produces worse results in the modeling case. Figure **??** shows the flow of the PSPI migration and modeling algorithms.

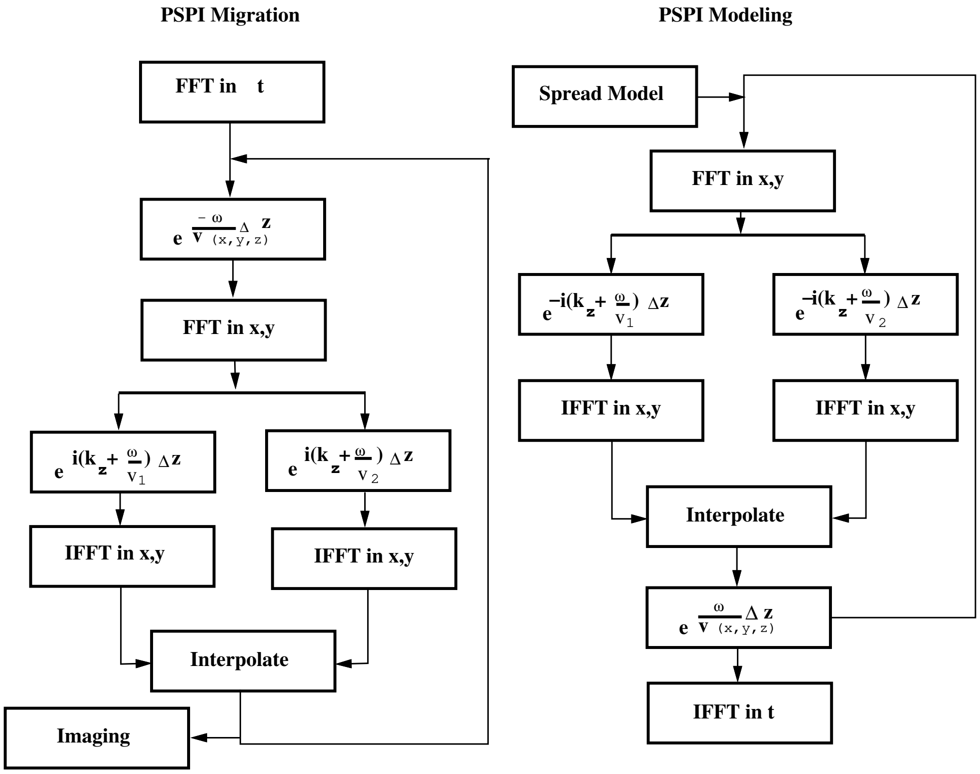

Figure 1: Phase Shift Plus Interpolation (PSPI) migration and the conjugate transpose PSPI modeling algorithm.

THE SPLIT-STEP FOURIER MODELING

Stoffa et al. (1990) present an alternative to the PSPI. Stoffa et al. replace the different velocities used in the downward continuation step with a single average velocity. A following phase shift of the wavefield with a perturbation term accounts for lateral velocity variations. The flow of Stoffa's migration is shown in Figure **??**.

The Split-Step algorithm is based on splitting the space variant slowness ($s(x,z) = \frac{1}{v(x,z)}$) into a constant term and a perturbation term,

$$s(x,z) = s_0(z) + \Delta s(x,z),$$

where $s_0(z)$ is a reference slowness defined as the average slowness in a depth interval. The wave equation (1) is Fourier transformed along the time axis to become

$$\frac{\partial^2 P(x,z,\omega)}{\partial z^2} + \frac{\partial^2 P(x,z,\omega)}{\partial x^2} + \omega^2 s^2(x,z) P(x,z,\omega) = 0. \tag{13}$$

After inserting the slowness split into the perturbation term and the average term the equation is transformed into

$$\frac{\partial^2 P(x,z,\omega)}{\partial z^2} + \frac{\partial^2 P(x,z,\omega)}{\partial x^2} + \omega^2 {s_0}^2(z) P(x,z,\omega) = -\omega^2 [2 s_0(z) \Delta s(x,z) + \Delta s^2(x,z)] P(x,z,\omega). \tag{14}$$

Defining the right side of the equation (14)

$$S(x,z,\omega) = \omega^2 (2 s_0(z) \Delta s(x,z) + \Delta s^2(x,z)) P(x,z,\omega)$$

we can write (14) as

$$\frac{\partial^2 P(x,z,\omega)}{\partial z^2} + \frac{\partial^2 P(x,z,\omega)}{\partial x^2} + \omega^2 {s_0}^2(z) P(x,z,\omega) = -S(x,z,\omega) \tag{15}$$

which is an inhomogeneous wave equation with a source term $S(x,z,\omega)$.

Stoffa et al. show that equation (15) can be integrated over a thin depth layer Δz by ignoring the contribution of the $\Delta s^2(x,z)$. This is done by Fourier transforming equation (15) in surface coordinates, dropping the second order term of the slowness perturbation and subsequently integrating over the depth layer Δz. After inverse Fourier transforming into surface coordinates the solution for downward propagating the wavefield has the form

$$P(x, z_0 + \Delta z, \omega) = P_0(x, z_0 + \Delta z, \omega) e^{i\omega \Delta s(x,z) \Delta z}, \tag{16}$$

where $P_0(x, z_0 + \Delta z, \omega)$ represents the wavefield downward continued with the average slowness $s_0(z)$. For upward continuation we just have to change the sign of Δz to have

$$P(x, z_0 - \Delta z, \omega) = P_0(x, z_0 - \Delta z, \omega) e^{-i\omega \Delta s(x,z) \Delta z}, \tag{17}$$

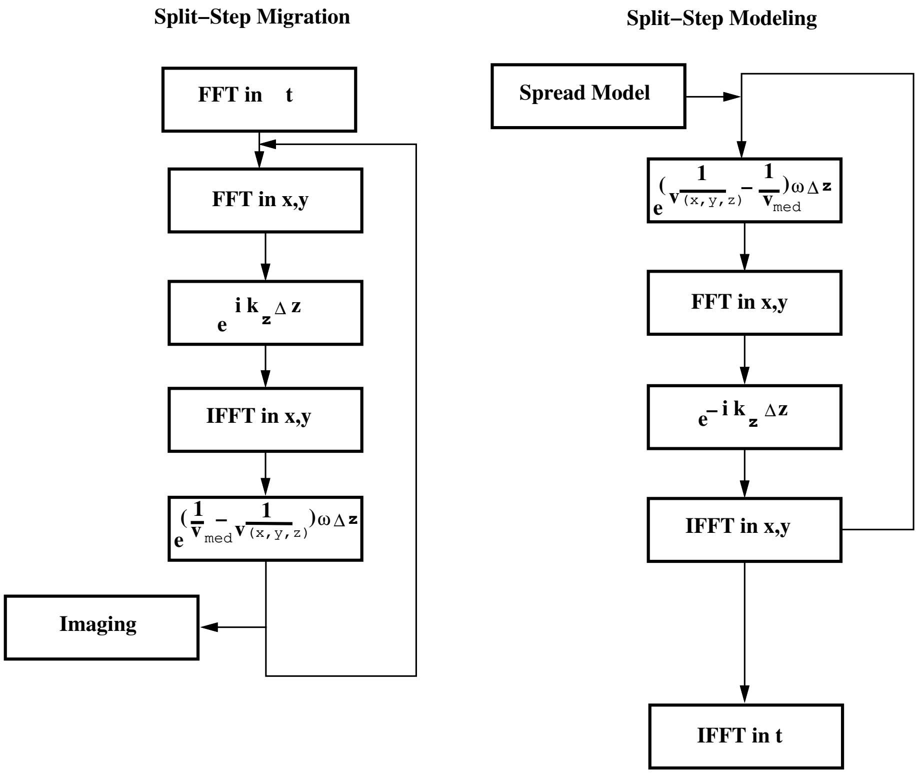

Figure 2: Split-Step Fourier migration and the conjugate transpose Split-Step Fourier modeling algorithm.

Though the mathematical path is very different from the one Gazdag and Sguazzero (1984) followed, the solution is very similar if you consider the PSPI algorithm with a single velocity.

The phase addition and subtraction trick in the PSPI algorithm is replaced by multiplication with

$$e^{i(\frac{1}{v(x,z)} - \frac{1}{v_{med}})\omega\Delta z} \tag{18}$$

after the downward extrapolation. Compare the phase shift in equation (18) with the trick in PSPI modeling to phase shift with

$$e^{-i\frac{\omega}{v_j}\Delta z},$$

followed by

$$e^{i\frac{\omega}{v(x,z)}\Delta z}.$$

The only difference is that in PSPI the phase shift is done after the inverse Fourier transform while in Split-Step it is done before the Fourier transform. The two modeling algorithms are compared in Figure **??**.

A 2-D example

The 2-D velocity model in Figure **??** is composed of three regions with variable velocity separated by a dipping linear interface and a curved interface. Above the first interface, the velocity has a horizontal and a vertical gradient. Between the first dipping interface and the curved interface the horizontal gradient is reversed. Under the curved interface there is only a vertical gradient. There is a velocity discontinuity at the location of the dipping event and another velocity jump under the curved boundary. The structural events are a dipping interface at the first velocity boundary and nine diffractors positioned at equal depth intervals. Both modeling algorithms are implemented using absorbing boundaries (Cerjan et al., 1985). The size of the absorbing region is a 20 gridpoint strip on each side of the grid.

While in the migration case the PSPI appears to perform better than Split-step in the presence of lateral velocity discontinuities, in the modeling case I found the Split-Step algorithm to achieve better results. Figure **??** presents the results of modeling with the two algorithms using the velocity model shown in Figure **??**. The PSPI modeling in this figure actually was implemented without the additional phase shift subtraction and addition introduced by Gazdag in the original PSPI algorithm. When using PSPI modeling, the amplitudes of the diffractions appear weaker than in the Split-Step algorithm.

Figure **??** shows the results of running PSPI modeling with the additional phase shift trick. The difference between the two figures is that in Figure **??**a the extra phase-shift term from equation (11) is applied **after** the upward propagation step which corresponds to the conjugate transpose of the original algorithm. In Figure **??**b the extra phase-shift term is applied **before** the upward propagation step, which is

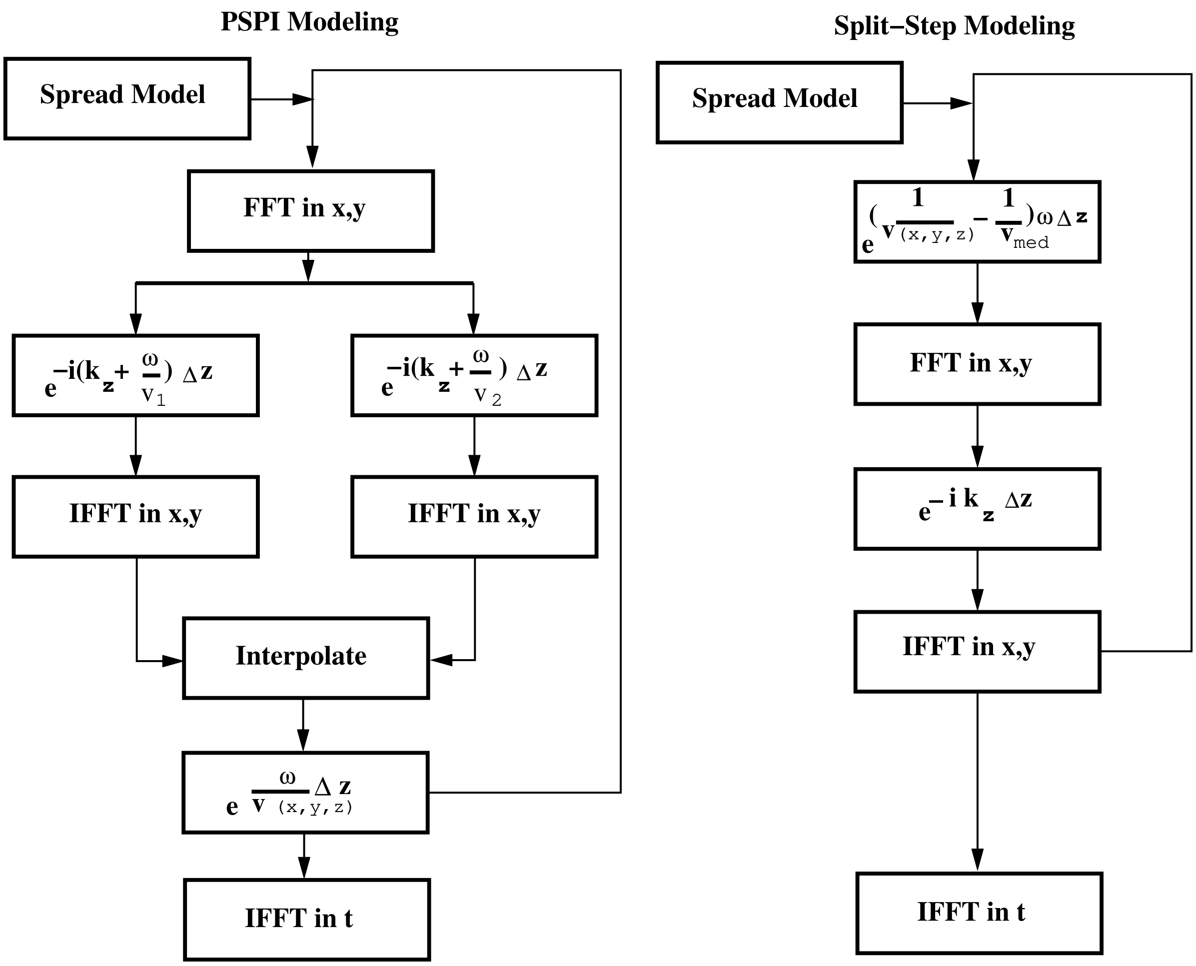

Figure 3: PSPI modeling and Split-Step Fourier modeling.

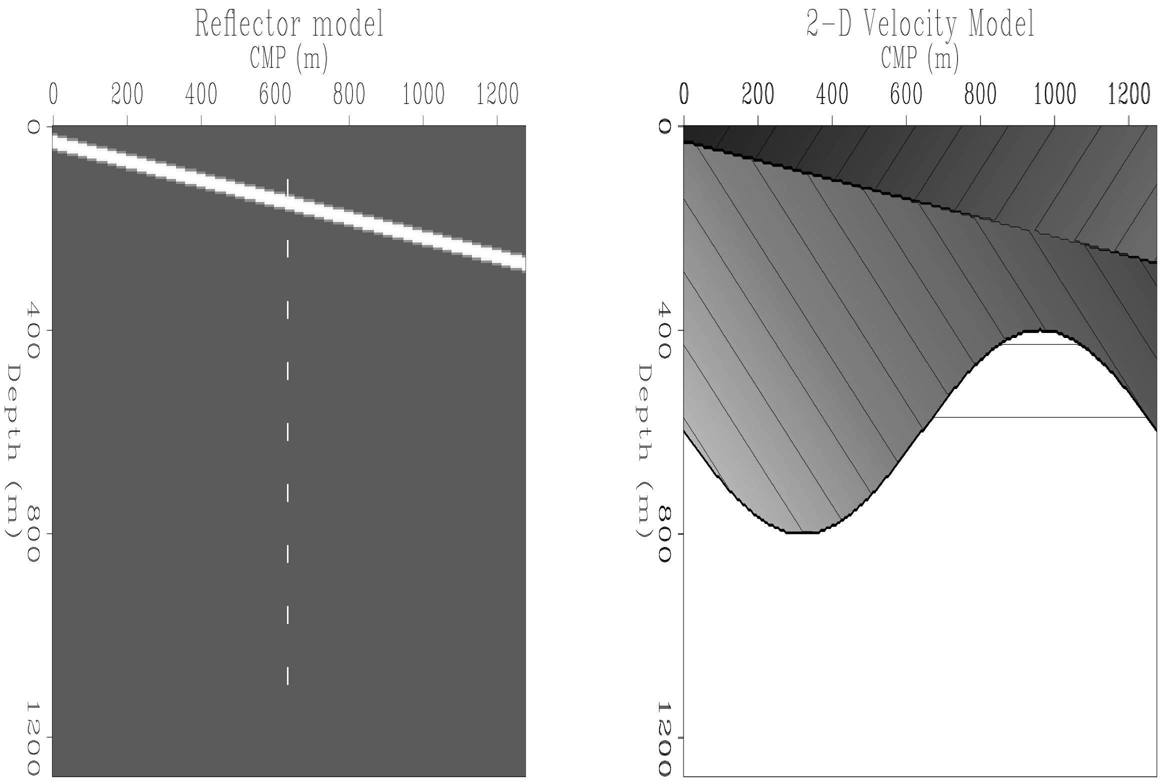

Figure 4: The 2-D reflection model and the velocity model. Velocity in the upper layer $v(x, z) = 1500{+}0.8x{+}0.7z$ (m/s), in the intermediate layer $v(x, z) = 3000{-}0.8x{+}0.7z$ (m/s) and in the lower layer $v(x, z) = 4000 + 0.7z$ (m/s).

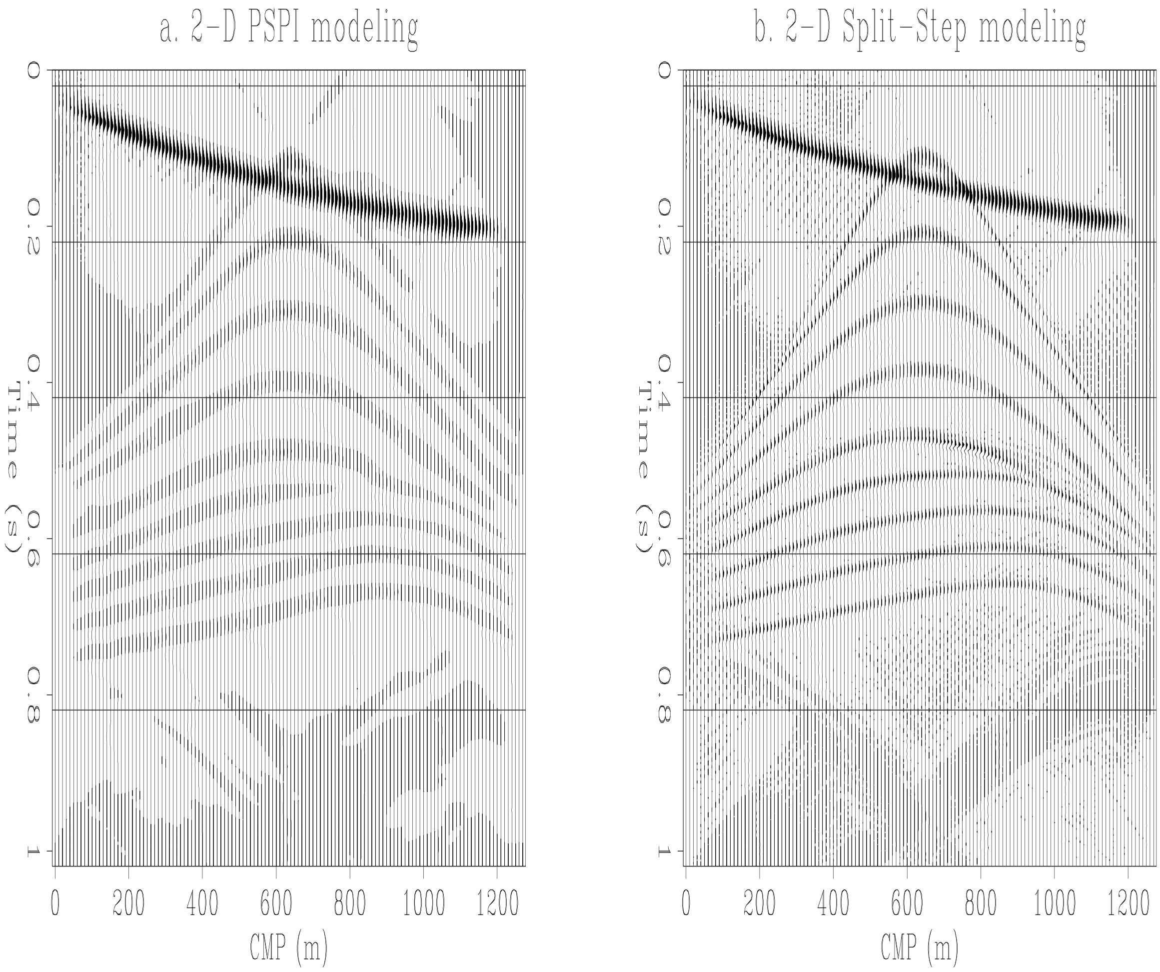

Figure 5: 2-D comparison of the two modeling algorithms:
a. PSPI modeling. The individual diffractors appear less defined compared to the Split-Step image.
b. Split-Step Fourier modeling.

equivalent to applying the Gazdag trick after downward propagation in the migration algorithm, in the same manner as it is done in the Split-Step migration. As seen here the two figures look almost identical. However it appears that while the individual diffractions are better delineated than in the simple PSPI algorithm (without the phase-shift trick), the dipping reflector is not. Overall the faster Split-Step appears to give better results for this particular velocity model.

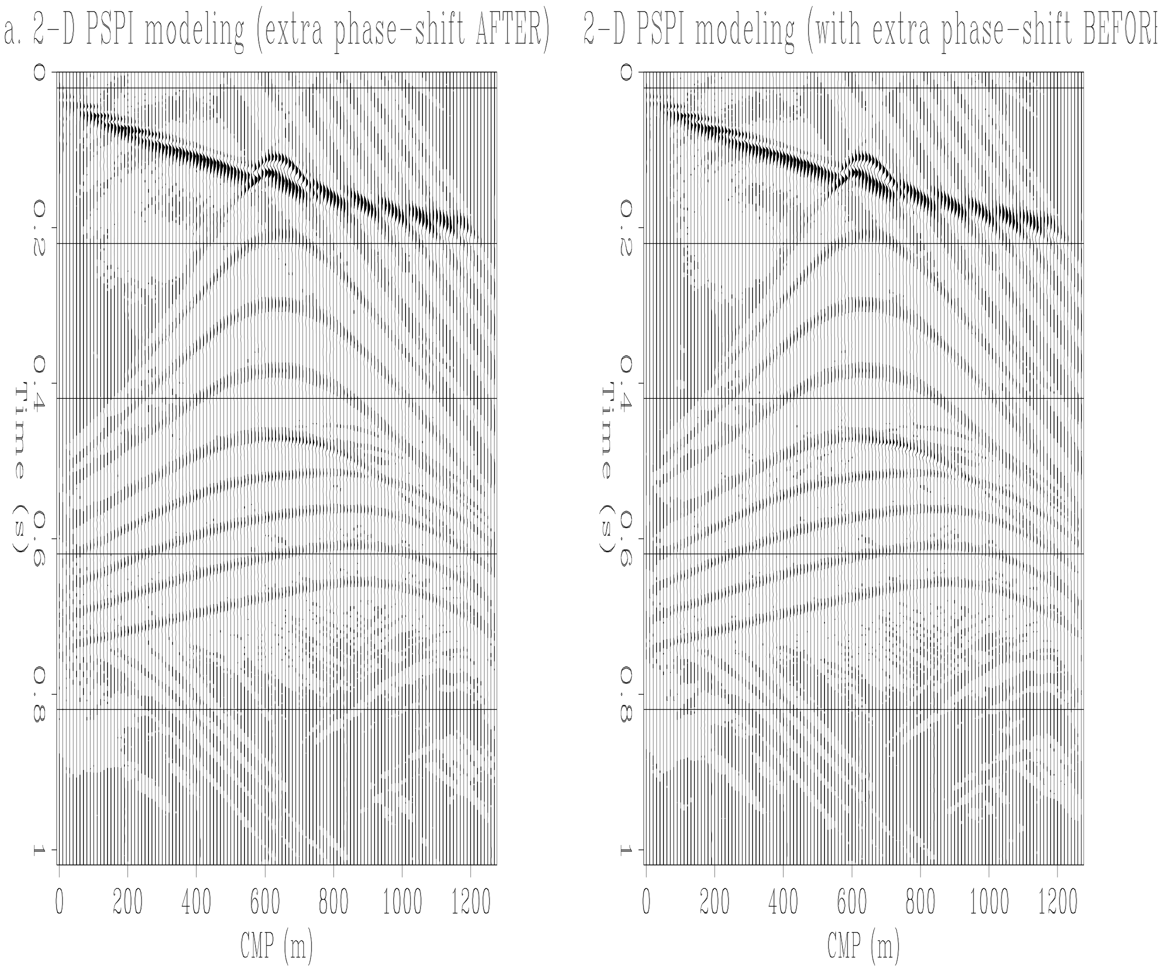

Figure 6: 2-D comparison of the two modeling algorithms:
a. PSPI modeling with extra phase-shift **after** upward propagation.
b. PSPI modeling with extra phase-shift **before** upward propagation.

A 3-D example

The three dimensional model is built for a linearly varying velocity in all three coordinates: $v(x, y, z) = v_0 + ax + by + cz$. The reflection model is just a diffractor situated inside the velocity cube. The size of the modeling baby-grid (64x64x256) was imposed by the amount of memory available on the SEP Connection Machine. The size of the available memory is also the reason why I have not implemented absorbing boundaries for the 3-D algorithm. With 20 gridpoints on each side of the 64 points mesh the active area will be only 24 points! Figure **??** displays time slices through the 3-D cube of diffraction data obtained using PSPI modeling while Figure **??** displays time slices through the 3-D cube of diffraction data obtained using Split-Step modeling.

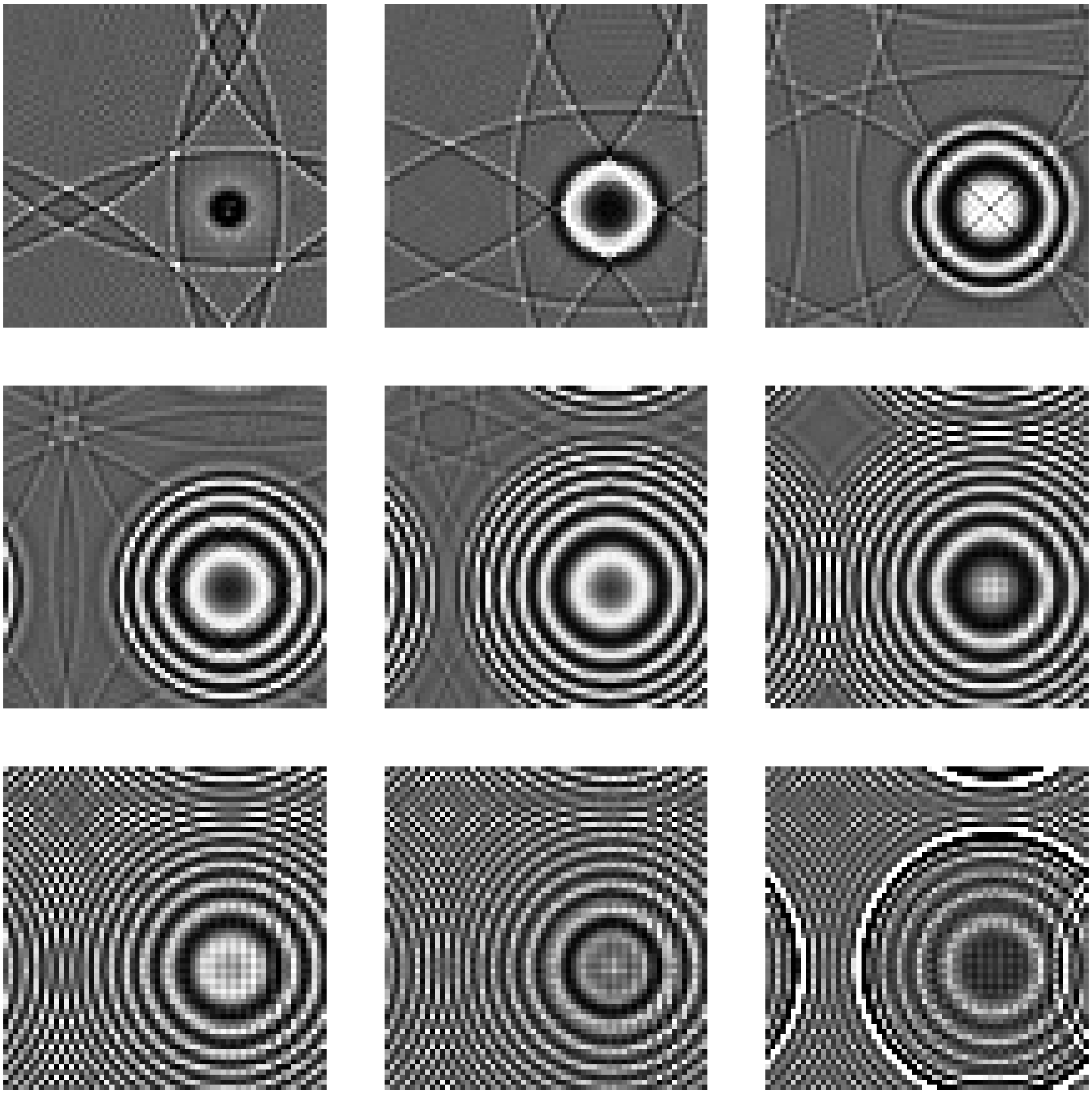

Figure 7: Time slices through the 3-D diffraction cube (64x64x256) output of the **PSPI** modeling program. The wraparound is due to non-absorbing boundaries.

CONCLUSIONS

I selected two Fourier domain modeling algorithms, based on the conjugate transpose of the Phase Shift Plus Interpolation and Split-step migration algorithms, and

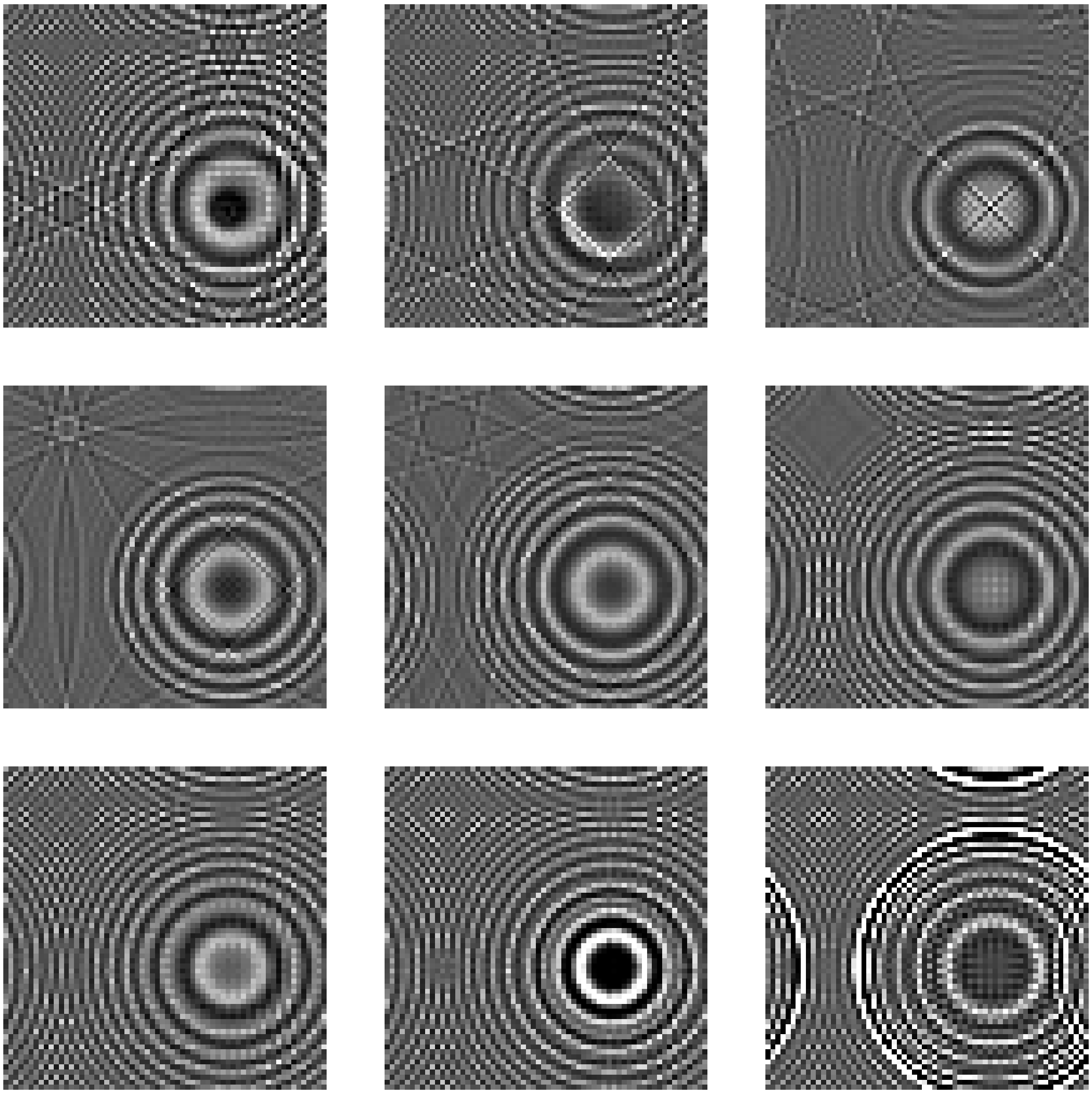

Figure 8: Time slices through the 3-D diffraction cube (64x64x256) output of the **Split-Step** modeling program. The wraparound is due to non-absorbing boundaries.

implemented them on a parallel computer (Connection Machine 2). I compared the modeling results of the two methods on 2-D and 3-D laterally varying velocity models. Even though the PSPI migration algorithm handles lateral velocity discontinuities better than the Split-Step migration algorithm (Popovici, 1991), it appears that in the modeling case the image quality performance is reversed.

REFERENCES

Cerjan, C., Kosloff, D., Kosloff, R., and Reshef, M., 1985, A nonreflecting boundary condition for discrete acoustic and elastic wave equations: Geophysics, **50**, 705–708.

Claerbout, J. F., 1992, Earth Soundings Analysis: Blackwell Scientific Publications.

Gazdag, J., 1978, Wave equation migration with the phase shift method: Geophysics, **43**, 1342–1351.

Gazdag, J., and Sguazzero, P., 1984, Migration of seismic data by phase shift plus interpolation: Geophysics, **49**, 124–131.

Popovici, A. M., 1991, Phase shift plus interpolation and split-step Fourier migration: SEP–**72**, 139–152.

Stoffa, P. L., Fokkema, J. T., de Luna Freire, R. M., and Kessinger, W. P., 1990, Split-step Fourier migration: Geophysics, **55**, 410–421.