Implementing implicit finite-difference in the time-space domain using spectral factorization and helical deconvolution

Ohad Barak

ABSTRACT

The method of modeling wavefield propagation with an implicit finite-difference approximation to the two-way acoustic isotropic wave equation, using spectral factorization and helical deconvolution, exhibits instability of the propagating wavefield as the time step is increased. In this study, I test several potential sources of the instability problem: the implicit finite-difference scheme itself, the precision of the floating point representation of the filter coefficients, the number of filter coefficients, and the spectral factorization method. None of these issues is the cause for the apparent instability.

INTRODUCTION

Implicit finite-difference methods are inherently more stable than explicit ones. This attribute enables us to increase the time step size (and consequently decrease computation time) while retaining stability of the wavefield. In the previous SEP report (Barak, 2010) I showed that by using spectral factorization and the helix transform, the propagation of a wavefield using an implicit finite-difference approximation of the two-way acoustic wave equation can be achieved by a set of deconvolution operations of filter coefficients applied to the wavefield. Through testing, I have found that despite the theoretical stability advantage of the implicit finite-difference scheme which I used for propagation, the resulting wavefield becomes more dispersive as the time step increases (to the point that the wavefield is no longer useful), and also that beyond a certain time step size - the wavefield diverges.

The increased dispersion of the implicit finite-difference scheme in comparison to an explicit scheme is an attribute of the scheme itself. This is not a fundamental problem, since some of this dispersion can be alleviated simply by using a higher order approximation. However, the causes of the instability of the wavefield beyond a certain time step size remained unclear. In order to understand the reasons behind the instability, I tested several hypotheses for its causes. These were:

1. The implicit finite-difference approximation itself.

2. The precision of the floating point representation of the filter coefficients.
3. The number of filter coefficients.
4. The spectral factorization method.

First I will review the method by which wave propagation can be done by deconvolutions with spectrally factorized filters of a finite-difference approximation, and then I will go over the various tests I carried out to try and determine the causes for the instability problem.

REVIEW OF METHODOLOGY

The two-way acoustic wave equation in one dimension reads:

$$\frac{\partial^2 P}{\partial t^2} = C^2 \frac{\partial^2 P}{\partial x^2}. \tag{1}$$

The central implicit finite-difference approximation I used for the propagation tests was 2nd order in time and 2nd order in space:

$$\begin{aligned} \frac{P_x^{t+1} - 2P_x^t + P_x^{t-1}}{\Delta t^2} &= \frac{C^2}{4\Delta x^2}[\left(P_{x+1}^{t+1} - 2P_x^{t+1} + P_{x-1}^{t+1}\right) \\ &+ 2\left(P_{x+1}^{t} - 2P_x^{t} + P_{x-1}^{t}\right) + \left(P_{x+1}^{t-1} - 2P_x^{t-1} + P_{x-1}^{t-1}\right)], \end{aligned} \tag{2}$$

where P is the pressure wavefield, t and x are the time and space coordinate indices, and Δt and Δx are the temporal and spatial step sizes. Note that this approximation is based on the Crank-Nicolson method, and so the spatial derivative is balanced between the three time steps: $t-1$, t, and $t+1$, where the central time index t has twice the weight of the other two time indices.
In order to propagate the wavefield, the pressure values at time $t+1$ must be equated to the values at times t and $t-1$. The linear system which must then be solved has the form:

$$\begin{aligned} \begin{pmatrix} U_0 & U_1 & 0 & 0 \\ U_1 & U_0 & U_1 & 0 \\ 0 & U_1 & U_0 & U_1 \\ 0 & 0 & U_1 & U_0 \end{pmatrix} \begin{pmatrix} P_1^{t+1} \\ P_2^{t+1} \\ P_3^{t+1} \\ P_4^{t+1} \end{pmatrix} &= \begin{pmatrix} V_0 & V_1 & 0 & 0 \\ V_1 & V_0 & V_1 & 0 \\ 0 & V_1 & V_0 & V_1 \\ 0 & 0 & V_1 & V_0 \end{pmatrix} \begin{pmatrix} P_1^{t} \\ P_2^{t} \\ P_3^{t} \\ P_4^{t} \end{pmatrix} \\ &+ \begin{pmatrix} W_0 & W_1 & 0 & 0 \\ W_1 & W_0 & W_1 & 0 \\ 0 & W_1 & W_0 & W_1 \\ 0 & 0 & W_1 & W_0 \end{pmatrix} \begin{pmatrix} P_1^{t-1} \\ P_2^{t-1} \\ P_3^{t-1} \\ P_4^{t-1} \end{pmatrix}. \end{aligned} \tag{3}$$

For simplicity, we can combine all the constants into one: $\alpha = \frac{C^2\Delta t^2}{4\Delta x^2}$. The matrix coefficients in equation 3 (the finite-difference weights) are then:
$U_0 = 1 + 2\alpha, \qquad U_1 = -\alpha;$
$V_0 = 2 - 4\alpha, \qquad V_1 = 2\alpha;$
$W_0 = -1 - 2\alpha, \qquad W_1 = \alpha.$

In shorter notation, equation 3 reads:

$$UP^{t+1} = VP^t + WP^{t-1}. \tag{4}$$

The solution of this linear system is:

$$P^{t+1} = U^{-1}\left(VP^t + WP^{t-1}\right). \tag{5}$$

To solve this system, we must perform polynomial division. The system is tridiagonal (and easily solvable) only for one dimension. For multiple dimensions, matrix U is block diagonal. Additional non-zero elements appear at a certain offset from the diagonal, making the solution process more complicated. However, using spectral factorization, the finite-difference weights of matrix U (which pertain to time $t+1$) can be factorized into a set of causal filter coefficients u and its time reverse $u^{'}$. Using the helical approach to deconvolution, equation 5 can be recast as:

$$P^{t+1} = (u^{'}u)^{-1}\left(VP^t + WP^{t-1}\right); \tag{6}$$

$$P^{t+1} = u^{-1}(u^{'})^{-1}\left(VP^t + WP^{t-1}\right). \tag{7}$$

Polynomial division is comparable to deconvolution. This means that the polynomial division in equation 5 can be achieved by a set of two deconvolutions of the data by the spectrally factorized coefficients u of matrix U. One deconvolution is done along the data in the reverse direction (application of the adjoint of the filter):

$$y_k = x_k - \sum_{i=1}^{N_u} u_i' y_{k-i}, \tag{8}$$

where u' is the time reversed filter coefficients of u. The other deconvolution is done in the forward direction:

$$x_k = y_k - \sum_{i=1}^{N_u} u_i x_{k-i}. \tag{9}$$

I used the SEPlib module `polydiv`, which uses the helical coordinates to perform the deconvolutions (the polynomial division) in equations 8 and 9.
The wavefield propogation is done by the following sequence:

1. Spectrally factorize the coefficients of matrix U.
2. Multiply the saved wavefield at time $t-1$ by the coefficients of matrix W.
3. Multiply the saved wavefield at time t by the coefficients of matrix V.
4. Sum the results of the previous 2 steps into a result vector.
5. Deconvolve the result vector by the time-reversed factorized filter coefficients u' (eq. 8).
6. Deconvolve the result vector by the factorized filter coefficients u (eq. 9).

Steps 2 - 6 are repeated for each time step. The inputs of the spectral factorization are the finite-difference weights of the matrix U (in Eq. 3), and the outputs are coefficients of a minumum phase filter u. Since I used a constant velocity in all propagation tests, the finite-difference weights are constant, and the filter is stationary.

IMPLEMENTATION OF METHODOLOGY WITH INCREASING TIME STEP SIZE

Figures 1(a)-1(f) show how wave propagation in one dimension using the implicit scheme from Eq. 2 and spectral factorization fails when the time step size is increased beyond a certain limit. The horizontal axis is time, and the vertical is distance. On the left the propagation is done using a linear equation system solver, and on the right is the result of deconvolving the wavefield with the filter coefficients obtained from spectral factorization. At smaller time steps, the two solutions are similar. The increasing dispersion with increasing time step size is apparent in both solutions. However, once the time step exceeds $5ms$, the wavefield propagated by deconvolution diverges, whereas the wavefield propagated by the "standard" linear system solver exhibits additional dispersion, yet remains stable.

Figures 2(a)-2(f) show the same kind of comparison as Figures 1(a)-1(f), except that here a small ϵ value was added to the central finite-difference weight (U_0) which was sent as an input to the spectral factorizer:

$$U_0 = 1 + 2\alpha + \epsilon, \qquad U_1 = -\alpha.$$

This results in a filter with slightly different coefficients, and with this filter the propagation is stable (with added dispersion). The value of ϵ required to maintain stability increases as the time step size increases. So far I've been unable to determine the relation between the value of the time step and the value of ϵ, but I know it is not arbitrary. If ϵ is too large, the result is a low-frequency dispersion which seems to initially precede the wavefield, as shown in Figure 3. Afterwards, the wavefield loses amplitude until eventually it disappears altogether.

While the addition of some ϵ value does stabilize the wavefield, the flip side is that it causes wrong propagation kinematics. This is a direct result of the artificial increase of the central finite weight. The incorrect kinematics can be seen in Figure 2(d) when looking at the wavelet as it reaches the edge at the 4 second mark. The arrival time of the wavelet is retarded as ϵ increases.

A similar phenomena occurs in 2D. In Figure 4 the effect of increasing the time step size from $\Delta t = 5ms$ to $10ms$ is shown. The increase causes the wavefield to diverge. Adding $\epsilon = 0.005$ to the central finite-difference weight, as in Figure 5, alters the filter coefficients obtained by spectral factorization, and enables stable propagation, with a slight time retardation of the wavefront. If ϵ is too large, then an unusual dispersion pattern appears. As the time step is increased further (Figure 6), the value of ϵ required for stable propagation increases as well, as does the time retardation of the wavefront. With too large an ϵ value the unusual dispersion pattern appears.

Summary of current implementation

An increase in the time step size causes the wavefield to diverge after a certain number of propagation steps. This divergence can be avoided - by artificially increasing the value of the central finite-difference weight. This correspondingly increases the zero-lag coefficient of the factorized filter, making it more dominant in comparison to the other filter coefficients. The result is stable propagation, albeit with much dispersion owing to the finite-difference approximation itself. The exact minimum value required for ϵ which ensures stable propagation is difficult to ascertain. Too large a value and an odd dispersion pattern unlike that of standard numerical dispersion begins to appear. The addition of ϵ to the central finite-difference weight also has the rather unfortunate effect of ruining the propagation kinematics.

EFFECT OF FLOATING POINT PRECISION

As a result of the tests shown in the previous section, I concluded that the reason for the unstable propagation at large time step sizes had to do with the spectral factorization, and not with the finite-difference approximation. One of the characteristics of the filter coefficients produced by the spectral factorization algorithm I used (SEPlib module `wilson`) is that they are very small. The smaller ones can reach 10^{-20}. Biondi and Clapp (pers. comm., 2010) suggested that I attempt to use double precision variables instead of single precision, in order to see whether the precision of the representation of the filter coefficients is indeed an issue. Ronen (pers. comm.,

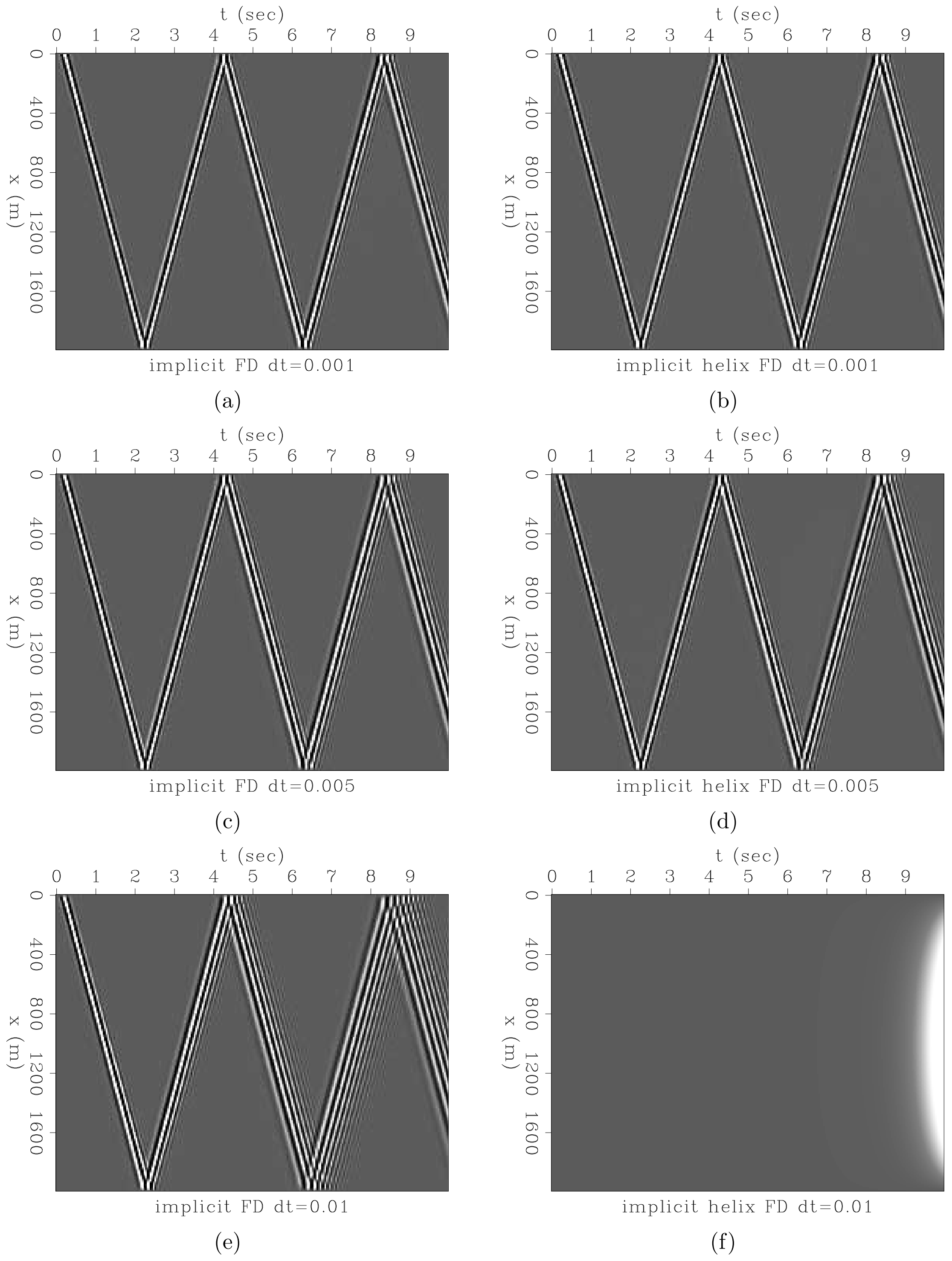
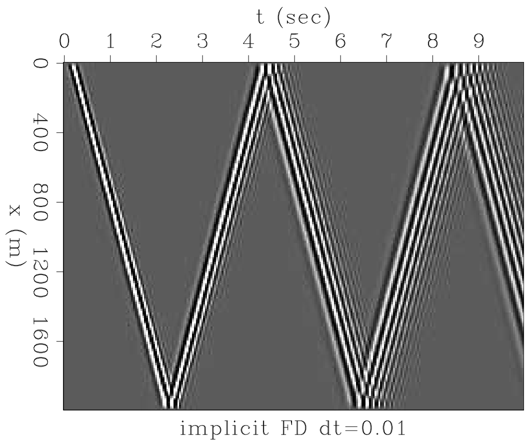
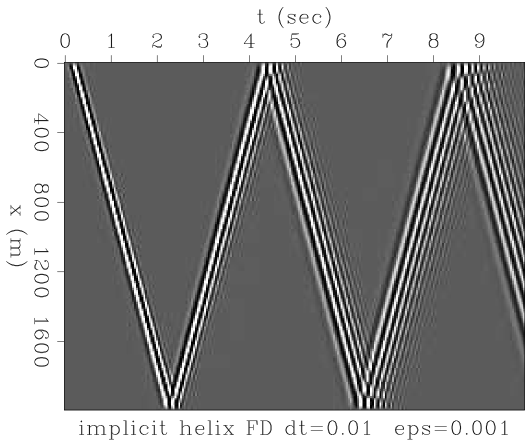


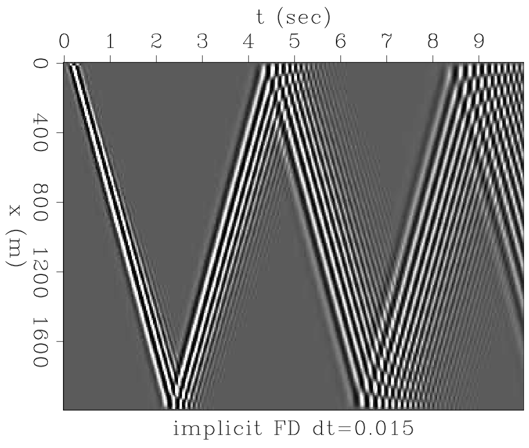
Figure 1: 1D Implicit (left) vs. Helical Implicit (right) finite-difference with constant velocity $= 1000m/s$. Horizontal axis is time, and the vertical axis is distance. Source is a Ricker wavelet with central frequency $= 12.5Hz$. The time step size is $\Delta t = 1ms$ for the top Figures, $5ms$ for the center Figures, and $10ms$ for the bottom Figures. $\Delta x = 10m$.[**ER**]



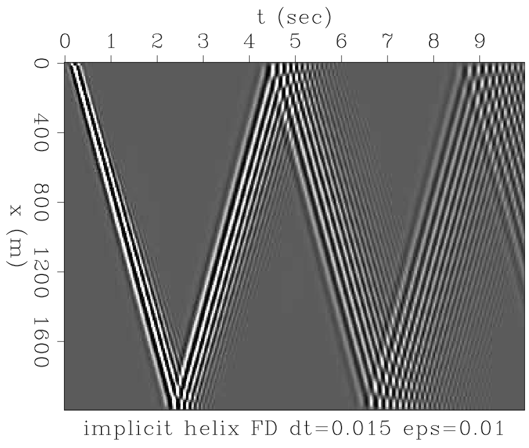
(a)



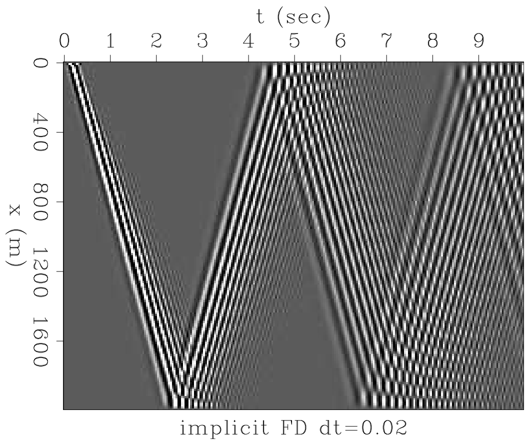
(b)



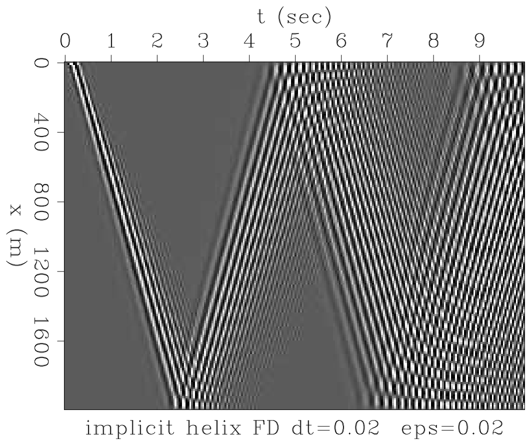
(c)



(d)



(e)



(f)

Figure 2: 1D Implicit (left) vs. Helical Implicit (right) finite-difference with constant velocity = $1000m/s$. Horizontal axis is time, and the vertical axis is distance. Source is a Ricker wavelet with central frequency = $12.5Hz$. The time step size is $\Delta t = 10ms$ for the top Figures, $15ms$ for the center Figures, and $20ms$ for the bottom Figures. Top right $\epsilon = 0.001$; center right $\epsilon = 0.01$; bottom right $\epsilon = 0.02$. $\Delta x = 10m$.[**ER**]

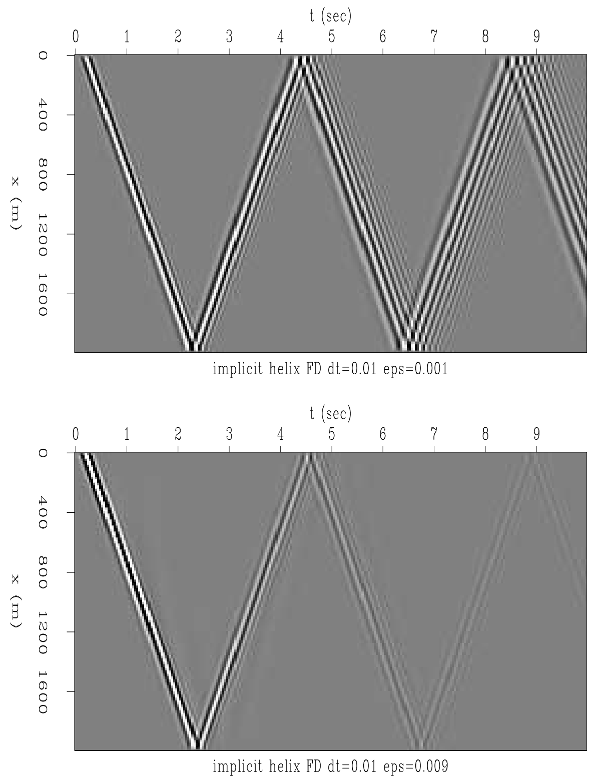


Figure 3: 1D Helical implicit finite-difference propagation with constant velocity $= 1000m/s$. The time step size is $\Delta t = 10ms$. $\epsilon = 0.001$ for the top Figure, $\epsilon = 0.009$ for the bottom Figure.[**ER**]

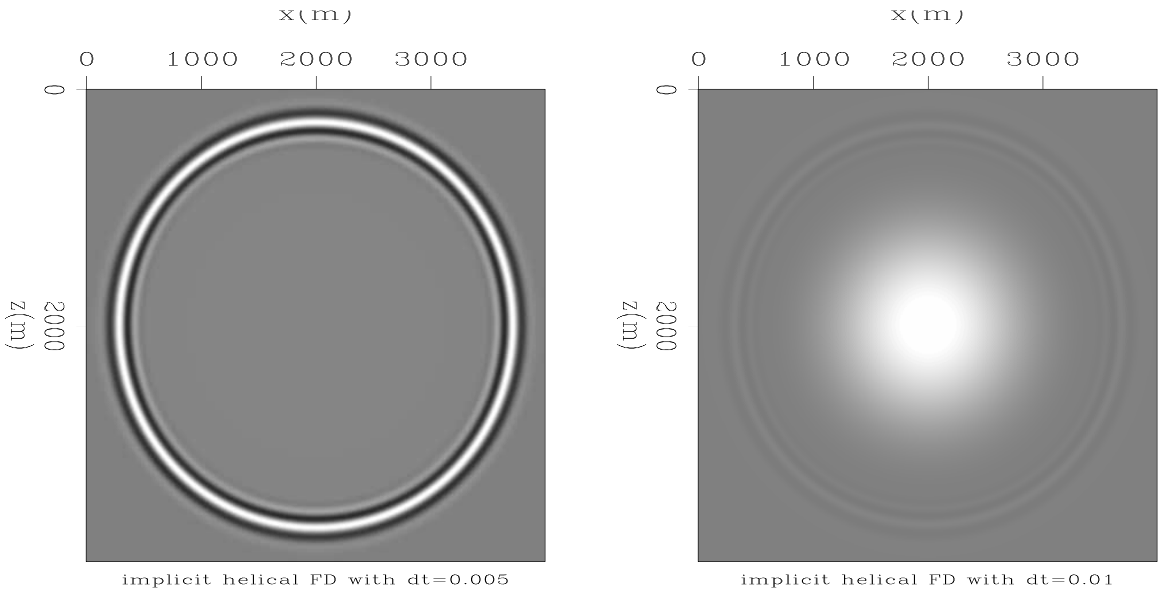


Figure 4: 2D helical implicit finite-difference with constant velocity $= 1000m/s$. Wavefields are after 2 seconds of propagation. Source is a Ricker wavelet with central frequency $= 12.5Hz$. The time step size is $\Delta t = 5ms$ for the left Figure, and $10ms$ for the right Figure. $\Delta x = \Delta z = 10m$.[**ER**]

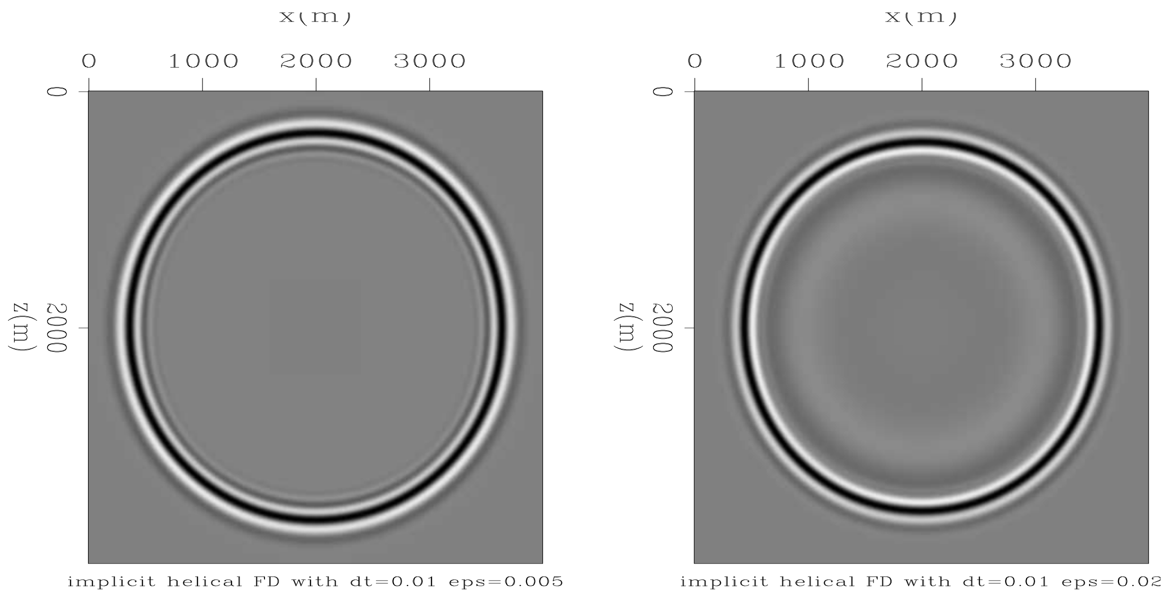


Figure 5: 2D helical implicit finite-difference with constant velocity = $1000 m/s$. Wavefields are after 2 seconds of propagation. Source is a Ricker wavelet with central frequency = $12.5 Hz$. The time step size is $\Delta t = 10 ms$. $\epsilon = 0.005$ for the left Figure, and $\epsilon = 0.02$ for the right Figure. $\Delta x = \Delta z = 10 m$.[**ER**]

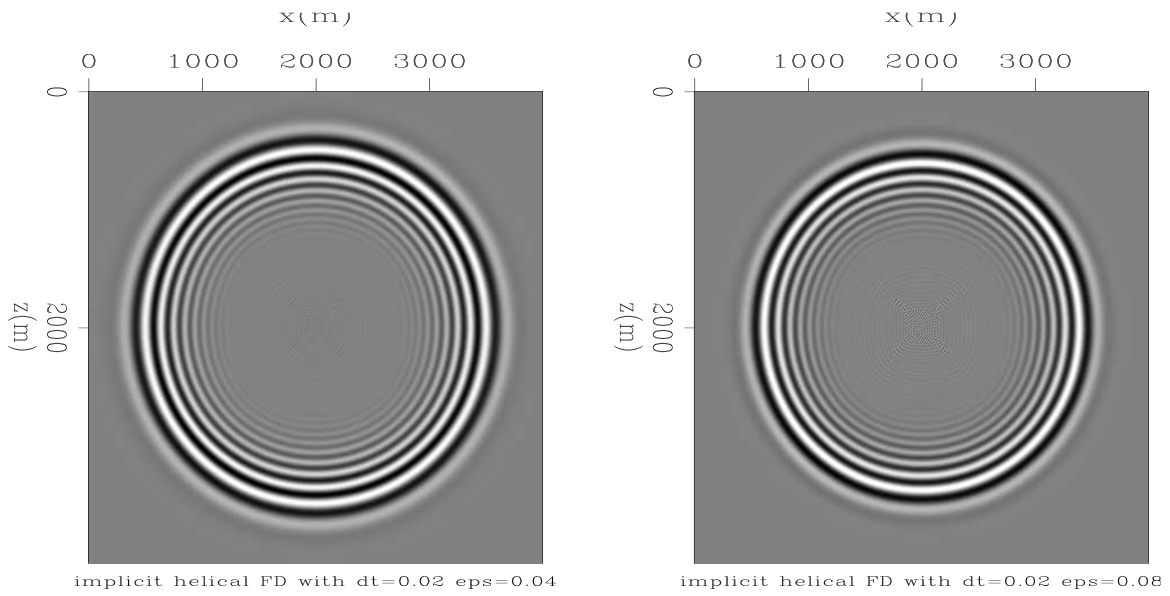


Figure 6: 2D helical implicit finite-difference with constant velocity = $1000 m/s$. Wavefields are after 2 seconds of propagation. Source is a Ricker wavelet with central frequency = $12.5 Hz$. The time step size is $\Delta t = 20 ms$. $\epsilon = 0.04$ for the left Figure, and $\epsilon = 0.08$ for the right Figure. $\Delta x = \Delta z = 10 m$.[**ER**]

2010) also suggested trying to do the opposite - reduce the precision and see whether that would have a degrading effect on the propagation.

To achieve double precision, both the spectral factorization algorithm and the helical deconvolution module had to be rewritten to include double precision variables. The wavefield itself was also composed of double precision variables. Furthermore, I used a 4th order in space, 2nd order in time approximation for this test. Figures 7(a)-7(d) show the comparison between propagation with single precision (left) and double precision (right). For the top Figures I used $\epsilon = 0$, and for the bottom ones $\epsilon = 0.01$. The time step Δt was $10ms$ for all Figures. The results for single and double precision are identical for this time step, and from other tests with many different time step sizes I can say that the behaviour is always identical, and so is the response to varying value of ϵ. The similarity in the wavefield values extends to the statistics of the wavefields - the mean, average, RMS and min/max values are nearly identical as well. In summary - I could not find a set of parameters for which propagation with double precision variables is better (or at all different) than propagation with single precision.

The next step was to attempt to reduce the precision of the spectrally factorized coefficients one decimal point at a time, and see when propagation with a certain set of parameters is destroyed as a result of this loss of precision. This should give an indication as to how important the floating point precision actually is for stable propagation. The precision reduction was done by running the regular `wilson` spectral factorization subroutine, and then reducing precision by the following two lines of code:

noindent `IntFilter = CutFactor * FloatFilter`
```
FloatFilter = IntFilter / CutFactor
```

`CutFactor` is a power of 10. Multiplying by this factor and then casting to integer effectiveley removes decimal precision from the filter coefficients. Example:
```
10000 * 1.23456 = 12345
12345 / 10000 = 1.2345
```

The purpose of this test was to see how many decimal precision digits can be removed from the filter coefficients before wavefield propagation using those coefficients is altered, in comparison to propagation with standard floating point precision. Results can be seen in Figures 8(a)-8(c). On the left is the result of propagation with single precision coefficients, with parameters which have shown stability ($\Delta t = 5ms, \epsilon = 0$). The center Figure shows propagation with coefficients which have had their precision truncated to 3 decimal points only. The wavefield exhibits a phase shift in comparison to the single precision wavefield, and yet it remains stable. Only when precision is truncated to 2 decimal points (right) is propagation severely affected.
The wavefields in Figures 7(a)-7(d) and 8(a)-8(c) were generated using factorization of the finite-difference weights of the 4th spatial order approximation (A-1). The values of these weights when using the specific set of propagation parameters were:
$U_0 = 1.3125, \qquad U_1 = -8.3333343E-02, \qquad U_2 = 5.2083340E-03.$

Since $\Delta x = \Delta z$, the weights are identical for both dimensions. These weights are fed to the spectral factorization routine, which is supposed to produce a causal set of filter coefficients, such that their cross-correlation will reproduce the finite-difference weights (Claerbout, 1997). This suggests that one way of testing the sensitivity of propagation to the floating point precision of the filter coefficients is to correlate the filter coefficients and compare the result to the finite-difference weights.

I used 21 filter coefficients to produce Figures 8(a)-8(c). For single precision propagation, the values of the correlation of the filter coefficients were (Only the first four values of the correlation are displayed. The rest are in A-2 to A-4):

$$1.312500 \quad -8.3333343E-02 \quad 5.2083335E-03 \quad 2.9154580E-12$$

For the propagation where precision was reduced to 3 decimal points only, the correlation was:

$$1.312500 \quad -8.3329208E-02 \quad 5.2077500E-03 \quad 0.0000000E+00$$

For the propagation where precision was reduced to 2 decimal points only, the correlation was:

$$1.312500 \quad -7.8187048E-02 \quad 0.0000000E+00 \quad 0.0000000E+00$$

Note that the correlation products are arranged in order of lags, so that the first coefficient corresponds to the central finite-difference weight U_0, the second to U_1, and the third to U_2. Note also that only after reducing precision to the 2nd decimal point, the weight U_2 is effectively erased, and the weight U_1 is considerably altered.

This comparison proves that the wavefield divergence shown in the previous sections is not the result of inadequate representation of the filter coefficient's floating point values when using single precision. If it were, then propagation with reduced precision would not have been possible. However, this result raises another question: If propagation is stable with so little precision, how come a small value of ϵ added to the central finite-difference weight (and by that also to the zero-lag filter coefficient) causes the wavefield to stabilize, when the effect that this slight addition has on the filter's correlation is so much less pronounced than the precision reduction?

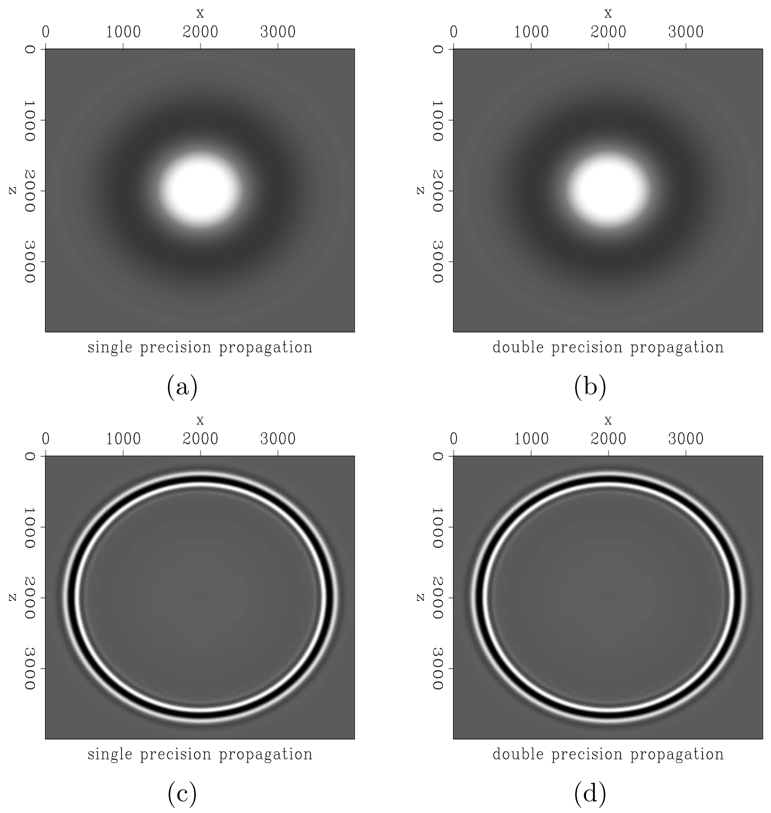


Figure 7: 2D helical implicit finite-difference using single (left) and double (right) precision. Velocity $= 1000m/s$. Wavefields are after 2 seconds of propagation. Source is a Ricker wavelet with central frequency $= 12.5Hz$. The time step size is $\Delta t = 10ms$. $\epsilon = 0$ for the top Figures, and $\epsilon = 0.01$ for the bottom Figures. $\Delta x = \Delta z = 10m$.[**ER**]

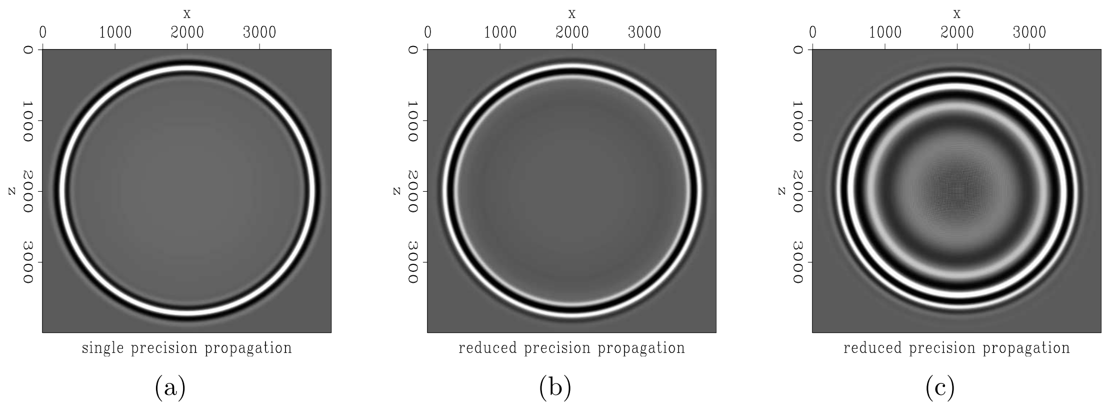


Figure 8: 2D helical implicit finite-difference using single precision (left), precision reduced to 3 decimal points (center), and precision reduced to 2 decimal points (right). Velocity $= 1000m/s$, $\Delta t = 5ms$, $\epsilon = 0$, $\Delta x = \Delta z = 10m$.[**ER**]

EFFECT OF NUMBER OF SPECTRALLY FACTORIZED COEFFICIENTS

I had initially assumed that the number of filter coefficients would be the most dominant factor in determining the accuracy of the propagation. I had supposed that the more filter coefficients used in the spectral factorization, the closer would be the value of their correlation to the finite-difference weights. Indeed, the instinctive response I had to the divergence problem was to increase the number of coefficients in the spectral factorization parameters. This, unfortunately, had no effect. Furthermore, the correlation of the filter coefficients created by the spectral factorizer (the Wilson-Burg algorithm) produced accurate finite-difference weights even when very few filter coefficients were present.

An example of the lack of the effect of number of coefficients on the propagation is shown in Figure 9. This 1D example shows how propagation using 2 spectrally factorized filter coefficients is basically identical to propagation when using 50 filter coefficients. Another indication comes from observing the filter coefficients themselves. This example was produced by a 2nd order scheme, which means that there are only 2 finite-difference weights. When factorizing using only 2 filter coefficients, the Wilson-Burg algorithm (for the propagation parameters used in Figure 9) yielded:

$$1.000000 \quad -5.5728100E-02.$$

The coefficients are displayed in order of lags, so the 1.0 is the zero-lag filter coefficient. Correlating these coefficients, we get:

$$1.125 \quad -6.2500007E-02$$

at lag 0 and lag 1, which are equal to the floating point representations of the finite-difference weights for Figure 9.

Factorizing using 50 filter coefficients produced (only the first four coefficients are shown, the rest are in A-5 and A-6):

$$1.000000 \quad -5.5728100E-02 \quad -2.7755576E-17 \quad 1.7347235E-18.$$

After lag 20, the coefficients are all zeros. Note that the first two coefficients are identical to the ones produced by the factorizer when requesting only two coefficients.The correlation of this filter is:

$$1.125000 \quad -6.2500007E-02 \quad -3.1236769E-17 \quad 1.9455218E-18$$

This correlation again shows the accurate representation of the finite-difference coefficients at lag 0 and lag 1. In addition, the correlation produces a set of other values at later lags, which are much smaller than the weights themselves.

The fact that two filter coefficients were sufficient to produce the finite-difference weights by correlation was interesting, but what is more important is to test what effect the change in the number of coefficients might have on the deconvolution process. Correlating the coefficients is like convolving them over a spike, once in the forward direction and once in reverse order. In order to test the exact effect that a change in the number of coefficients had on the deconvolution, I tested the result of deconvolving the coefficients over a spike. Here as well, the result was identical. I shall spare displaying the numbers themselves for this case.

In summary, I could not a find a combination of parameters (of propagation or of factorization) for which wavefield propagation was more stable if more filter coefficients were used.

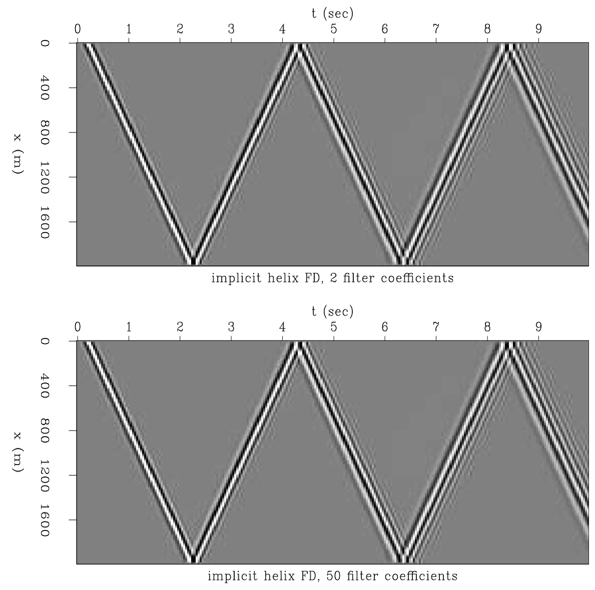


Figure 9: 1D helical implicit finite-difference with 2 spectrally factorized filter coefficients (top), and 50 coefficients (bottom). Velocity $= 1000m/s$, $\Delta t = 5ms$, $\epsilon = 0$, $\Delta x = 10m$.[**ER**]

WILSON-BURG VS. KOLMOGOROFF SPECTRAL FACTORIZATION

Last on the checklist was the spectral factorization algorithm itself. In Rickett (2001) the Kolmogoroff spectral factorization method is shown to be successful for modeling seismic activity on the surface of the Sun. I used the SEPlib `ccrosskolmog` module, and compared wavefield propagation when the spectral factorization was done by the Kolmogoroff method vs. the Wilson-Burg method. The comparison is shown in Figure 10. On the left are wavefields propagated with Wilson factorization, and on the right - Kolmogoroff. The time step is $\Delta t = .5ms$ in the top Figures. When the time step is increased to $1ms$, the propagation with Kolmogoroff coefficients diverges. However, if $\epsilon = 10^{-4}$ is added to the central finite-difference coefficient prior to factorization (bottom right), propagation is successful and appears similar to propagation by Wilson factorized coefficients.

This result indicates that the Kolmogoroff factorization method is even less suitable than the Wilson method for this finite-difference scheme, since the addition of a small value to the central FD coefficient when using Wilson is necessary only at greater time step sizes.

The ten Wilson filter coefficients used to create the center panels in Figure 10 were:

$$\begin{array}{cccc}
1.0 & -2.4875777E-03 & 0.0000000E+00 & -6.7762636E-21 \\
-2.6469780E-23 & 5.1698788E-26 & 0.0000000E+00 & -3.9443045E-31 \\
0.0000000E+00 & -1.5046328E-36 & & .
\end{array}$$

The Kolmogoroff coefficients were:

$$\begin{array}{cccc}
1.002494 & -2.4938183E-03 & 4.7695384E-08 & 1.3291222E-08 \\
-3.6223135E-08 & 4.7327675E-09 & 8.5023713E-09 & 8.8970848E-09 \\
-9.8089106E-12 & 4.9380566E-09 & & .
\end{array}$$

Other than the zero-lag coefficient not being equal to 1, a striking difference is that the Kolmogoroff coefficients do not drop off quickly as do the Wilson coefficients. This has a degrading effect on the filter correlation. The Wilson filter's correlation is:

$$\begin{array}{cccc}
1.005 & -2.5000004E-03 & 1.6940662E-23 & -6.8100374E-21 \\
-2.6602097E-23 & 5.1956968E-26 & 9.8607629E-34 & -3.9640020E-31 \\
0.0000000E+00 & -1.5121468E-36 & &
\end{array}$$

The Kolmogoroff's filter correlation is:

$$\begin{array}{cccc}
1.005 & -2.4999434E-03 & 8.8091141E-08 & 7.9071558E-09 \\
-2.9466412E-08 & -1.4710333E-10 & 3.0997090E-11 & 3.9199342E-09 \\
-6.6459863E-12 & -2.8234270E-09 & &
\end{array}$$

The finite-difference coefficients for the parameter set of the wavefields in Figure 10 are $U_0 = 1.005, U_1 = -2.5E-03$. The Wilson filter's correlation recreates these weights precisely, while the Kolmogoroff filter's correlation does not. Also, the drop-off in the magnitude of the filter correlation at lags which do not represent finite-difference weights (i.e. not lag 0 or lag 1) is much better for the Wilson filter.

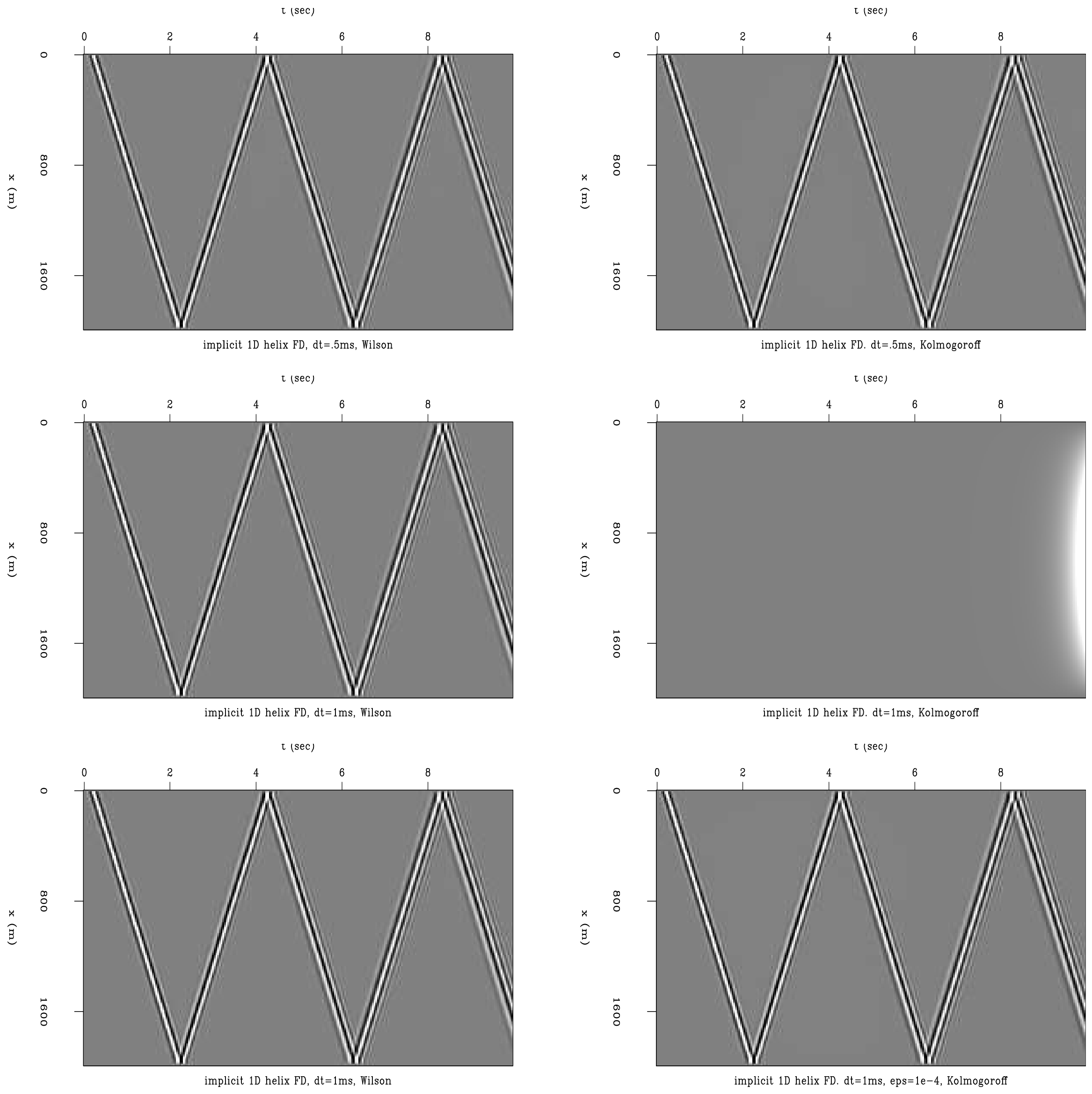


Figure 10: 1D helical implicit finite-difference with Wilson-Burg spectral factorization (left), and Kolmogoroff spectral factorization (right). $\Delta t = .5ms$ (top), $1ms$ (center and bottom). $\epsilon = 1e^{-4}$ only on the bottom right Figure, otherwise $\epsilon = 0$. Velocity $= 1000m/s$, $\Delta x = 10m$.[**ER**]

CONCLUSION AND FUTURE WORK

After conducting the aforementioned tests, I still cannot say why wavefield propagation by the proposed methodology does not function beyond a certain time step size. I can only conclude that for some reason the spectral factorization fails when the finite-difference weights, which I wish to factorize, are not dominated by the central finite-difference weight. Since the entire purpose of attempting to use the combination of implicit finite-difference and spectral factorization for propagation was to increase the time step size (thereby decreasing the total computation time, but also decreasing dominance of the central finite-difference weight), this failure makes the method unuseful. At the time step sizes for which this method does work, explicit methods will function better and faster.

There is one possible avenue in which to continue research of this method. The central weight of the finite-difference scheme which I used does decrease in dominance as the time step size is increased, but I am not bound to use this scheme only. It is possible that an alternate implicit finite-difference scheme will not have this attribute, and will thus be more amenable to factorization when the time step size is increased.

One source of such a scheme could be the pseudo-Laplacian discussed in Etgen and Bransdsberg-Dahl (2009).

REFERENCES

Barak, O., 2010, Implicit finite difference in time-space domain with the helix transform: Stanford Exploration Project, **Report 140**, 103–118.

Claerbout, J. F., 1997, Multidimensional recursive filters via a helix: Stanford Exploration Project, **Report 95**, 1–13.

Etgen, J. T. and S. Bransdsberg-Dahl, 2009, The pseudo-analytical method: application of pseudo-laplacians to acoustic and acoustic anisotropic wave propagation: SEG Expanded Abstracts, 2552–2556.

Rickett, J., 2001, Spectral factorization of wavefields and wave operators: PhD thesis, Stanford University.

APPENDIX A

The 4th order in space and 2nd order in time implicit finite difference scheme used to create Figures 7(a) and 8(a) was:

$$\begin{aligned}\frac{P_x^{t+1}-2P_x^t+P_x^{t-1}}{\Delta t^2} &= \frac{C^2}{4\Delta x^2}\Big[\left(\frac{-1}{12}(P_{x+2}^{t+1}+P_{x-2}^{t+1})-\frac{16}{12}(P_{x+1}^{t+1}+P_{x-1}^{t+1})-\frac{30}{12}P_x^{t+1}\right)\\ &+ 2\left(\frac{-1}{12}(P_{x+2}^{t}+P_{x-2}^{t})-\frac{16}{12}(P_{x+1}^{t}+P_{x-1}^{t})-\frac{30}{12}P_x^{t}\right)\\ &+ \left(\frac{-1}{12}(P_{x+2}^{t-1}+P_{x-2}^{t-1})-\frac{16}{12}(P_{x+1}^{t-1}+P_{x-1}^{t-1})-\frac{30}{12}P_x^{t-1}\right)\Big]. \quad \text{(A-1)}\end{aligned}$$

The correlation of the 21 filter coefficients used to create Figure 8(a) for single precision propagation was (I apologize for having the temerity to show raw numbers, but I couldn't find a suitable graphic representation):

$$\begin{array}{cccc} 1.312500 & -8.3333343E-02 & 5.2083335E-03 & 2.9154580E-12 \\ 5.4817577E-09 & -4.2199644E-09 & -2.5573333E-12 & 2.8602476E-13 \\ 3.4990573E-11 & -4.1297177E-10 & -8.3333343E-02 & -2.3679786E-12 \\ 2.3124791E-15 & -1.7574841E-13 & -3.4801828E-10 & 3.0937783E-10 \\ 5.6366680E-15 & -1.0001472E-13 & 0.0000000E+00 & -3.7887658E-11 \\ 5.2083340E-03 & & & . \end{array} \quad \text{(A-2)}$$

For propagation where precision was reduced to 3 decimal points only, the correlation was:

$$\begin{array}{cccc} 1.312500 & -8.3329208E-02 & 5.2077500E-03 & 0.0000000E+00 \\ 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 & -2.0831001E-05 \\ 5.2077517E-06 & 4.1662315E-05 & -8.2350150E-02 & -2.0831001E-05 \\ 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 \\ 0.0000000E+00 & 0.0000000E+00 & 2.0831001E-05 & -3.3329602E-04 \\ 5.2077500E-03 & & & . \end{array} \quad \text{(A-3)}$$

For propagation where precision was reduced to 2 decimal points only, the correlation was:

$$\begin{array}{cccc} 1.312500 & -7.8187048E-02 & 0.0000000E+00 & 0.0000000E+00 \\ 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 \\ 0.0000000E+00 & 4.6912231E-03 & -7.8187048E-02 & 0.0000000E+00 \\ 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 \\ 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 & 0.0000000E+00 \\ 0.0000000E+00 & & & . \end{array} \quad \text{(A-4)}$$

The correlation products are arranged in order of lags. Since the finite-difference operator is two dimensional, the weights for U_1 and U_2 reappear at lags corresponding to the wrap-around of the 1D filter around the edges of the 2D grid (in helical coordinates). Therefore the 11th coefficient is equal to the 2nd coefficient, and the 21st is equal to the 3rd.

The 50 filter coefficients used to produce Figure 9:

$$\begin{array}{cccc}
1.000000 & -5.5728100E-02 & -2.7755576E-17 & 1.7347235E-18 \\
0.0000000E+00 & -6.7497938E-21 & 8.2718061E-25 & 7.7548182E-26 \\
8.2718061E-25 & 2.6315262E-28 & 1.4603365E-29 & 2.0273725E-28 \\
6.7288147E-32 & 3.7618776E-33 & -2.4442825E-32 & 1.1651518E-35 \\
6.4931696E-37 & 3.5264831E-38 & 0.0000000E+00 & -2.3509887E-38 \\
0.0000000E+00 & .. & .. & ..
\end{array} \tag{A-5}$$

After lag 20, the coefficients were all zeros. The correlation of these coefficients was:

$$\begin{array}{cccc}
1.125000 & -6.2500007E-02 & -3.1236769E-17 & 1.9455218E-18 \\
4.2186216E-22 & -7.5700597E-21 & 9.2285031E-25 & 3.5272806E-26 \\
9.2768060E-25 & 2.9421741E-28 & 3.7068419E-30 & 2.2736905E-28 \\
7.5229681E-32 & 5.7466863E-33 & -2.7413771E-32 & 1.3026792E-35 \\
7.2821997E-37 & 3.9550105E-38 & 0.0000000E+00 & -2.6366737E-38 \\
0.0000000E+00 & .. & .. & ..
\end{array} \tag{A-6}$$