# Performance of 3D Depth Migration Algorithms

Jan Thorbecke and Ben Geesink

*Laboratory of Seismics and Acoustics,*
*Centre for Technical Geoscience,*
*Delft University of Technology,*
*P.O. Box 5046, 2600 GA Delft, The Netherlands*

**Summary**

Recursive Pre-Stack Depth Migration in the frequency domain consist of three steps; *forward extrapolation* of the source wave field, *inverse extrapolation* of the measured wave field and *imaging* at all depth levels. The most compute intensive part is the extrapolation of the wave fields. In this abstract we will look at three different wave field extrapolation techniques and focus on the computational effort. The extrapolation techniques discussed are:

- 2-dimensional convolution, (Blacquière et al., 1989)
- Hale-McClellan expansion, (Hale, 1991)
- series expansion in $\cos(kr)$ (Thorbecke, 1997)

Every method is explained briefly and implementation details of the algorithms are discussed. Numerical examples are given for impulse responses and a synthetic shot record. Finally, we demonstrate how each of the algorithms can benefit from the use of multiple processors.

**Introduction**

Forward wave field extrapolation in the frequency domain can be expressed by the Rayleigh II integral:

$$P^+(\boldsymbol{x},\omega) = \int_{\partial D} G^+(\boldsymbol{x},\boldsymbol{x}',\omega)P^+(\boldsymbol{x}'.\omega)\mathrm{d}^2\boldsymbol{x}', \tag{1}$$

where $P^+(\boldsymbol{x}',\omega)$ represents the measured wave field at the surface $\partial D$ and $G^+(\boldsymbol{x},\boldsymbol{x}',\omega)$ represents the extrapolation operator (Greens function) from the surface $\partial D$ to a point in the subsurface $\boldsymbol{x}$ (Wapenaar and Berkhout, 1989).

In homogeneous media the one-way extrapolation operator in the $\boldsymbol{k}-\omega$ (wavenumber-frequency) domain is a simple analytical function which is given by (Gazdag, 1978):

$$\tilde{G}(k_x,k_y,\omega,\Delta z) = \exp\left(-j\sqrt{\frac{\omega^2}{c^2} - (k_x^2+k_y^2)}\Delta z\right) \tag{2}$$

with $\Delta z$ being a small extrapolation step and $c$ the propagation velocity of the medium. The advantage of using the phase shift operator in the $k_x,k_y-\omega$ domain is that the extrapolation result is obtained by multiplication of the data with the phase shift operator. However, multiplication in the $k_x,k_y-\omega$ domain rules out the possibility of applying a laterally varying operator. To allow laterally varying medium functions a convolution operator in the $x,y-\omega$ (space-frequency) domain should be used. This spatial convolution operator must be designed in such a way that it gives accurate and stable results within a reasonable computational time. To arrive at this goal two steps must be taken; the first step is an optimum *design* of the spatial operator and the second step deals with a fast *implementation* of the spatial convolution. The most efficient algorithms combine these two steps and a spatial operator is designed in such a way that it can be implemented in a fast way. Note that the extrapolation operator is circular symmetric which makes an efficient optimization and implementation possible.

**Design**

Most spatial extrapolation methods can be expressed in the wavenumber domain as an approximation to the phase shift operator of equation (2). Different approximations can be based on a power series or on an expansion with respect to the cosine terms of the Fourier transform. Transforming these expansions, with a limited number of terms and in an optimal way, to the spatial domain gives the spatial convolution operator. Three types of expansions are discussed:

$$\tilde{G}(k_x,k_y) = \exp(-jk_z\Delta z), \tag{3}$$

$$\approx \sum_{m=0}^{M}\sum_{n=0}^{N} G_{mn}\cos(k_x m\Delta x)\cos(k_y n\Delta y), \tag{4}$$

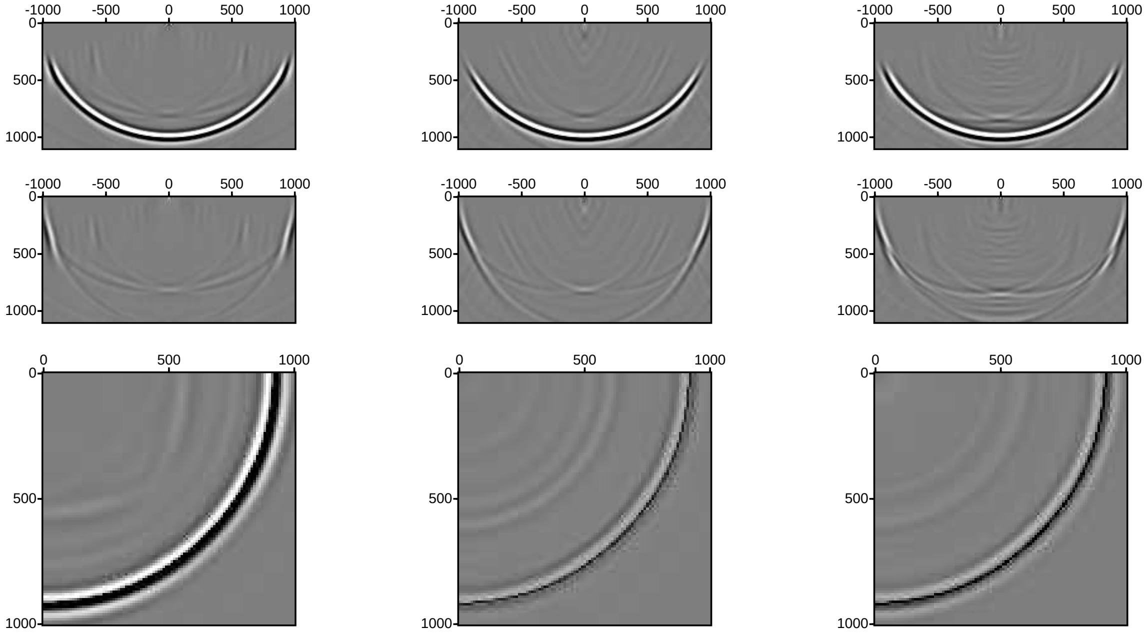

Figure 1: *Impulse responses for the three extrapolation methods; on the left the 2D convolution, in the middle the Hale McClellan method and on the right the series expansion. The top picture shows a cross-sections along $x = 0$, the middle picture is the difference with the analytical answer and the bottom shows a depth-slices at 400 m ($\approx 67^o$).*

$$\approx \sum_{l=0}^{L} G_l T_l(\cos(k_r \Delta x)), \quad (5)$$

$$\approx \sum_{j=0}^{J} Y_j [\cos(k_r \Delta x)]^j, \quad (6)$$

with $k_r = \sqrt{(k_x^2 + k_y^2)}$ and $k_z = \sqrt{k^2 - k_r^2}$. Note that there are many more expansions possible like the expansions in Laplacian $(k_x^2 + k_y^2)$(Soubaras, 1996) or $k_z$ terms.

Equation (4) represents the inverse Fourier transform of a symmetric (in $x$ and $y$) spatially limited operator. $G_{mn}$ are the coefficients of the 2-D spatial convolution operator. The coefficients $G_{mn}$ used in this paper are obtained by a Weighted Least Squares optimization method. The spatial dimensions of the convolution operator are $(2M - 1)^2$).

In equation (5) the 2-dimensional problem is reduced to a 1-dimensional filter problem using the circular symmetry of the 2-D phase shift operator (Hale, 1991). This method is represented by an Chebychev polynomial ($T_l$) in 1-dimensional cosine terms. The coefficients $G_l$ represent the 1-D phase-shift operator and are obtained with any preferred 1-D optimization method. The cosine terms $\cos(k_r \Delta x)$ are approximated by small ($SxS$) 2-D convolution filters. The spatial size of the operator is $(L(S - 1) - S + 2)^2$.

Equation (6) is a series expansions in $\cos(k_r \Delta x)$. The cosine terms are approximated by small (SxS) 2-D convolution filters. The coefficients $Y_j$ in the series expansions are obtained by numerically optimizing the coefficients given the approximation of the cosine terms. Note that the number of expansion terms is more limited by the accuracy of the floating point implementation than the Chebychev recursion. The spatial size of the operator is also $(J(S - 1) - S + 2)^2$.

**Accuracy**

To check the accuracy of the extrapolation method an impulse response is calculated with $\Delta x = \Delta y = \Delta z = 10$ m, $c = 1000 m/s$ and a Ricker wavelet with a time delay of 1.0 seconds and a frequency peak at 10 Hz. The impulse response shows us three things; the circularity of the operator, the numerical artifacts and the result at high angles.

To have a fair comparison between the different methods the footprint of the spatial operator is chosen to be the same. For the 2D convolution we have used a 33x33 convolution operator and for the Chebychev recursion and series expansion we used a 3x3 operator for the cosine operator and 17 terms in the expansion. In Figure 1 the impulse responses are shown for the three spatial convolution methods. Note that all methods give comparable results; the depth slice at $67^o$ of the 2D convolution shows a higher (more accurate) amplitude than the other methods. Note that a higher accuracy for the cosine methods can be achieved by optimizing the cosine terms for every wavenumber $k = \frac{\omega}{c}$. However, the performance will suffer from the fact that the cos(kr) stencil depends on the local velocity. Using a larger stencil (5x5) and less terms (9,) to have the same footprint, gives non-accurate results. A 13x(5x5) operator gives comparable results but has a much larger footprint.

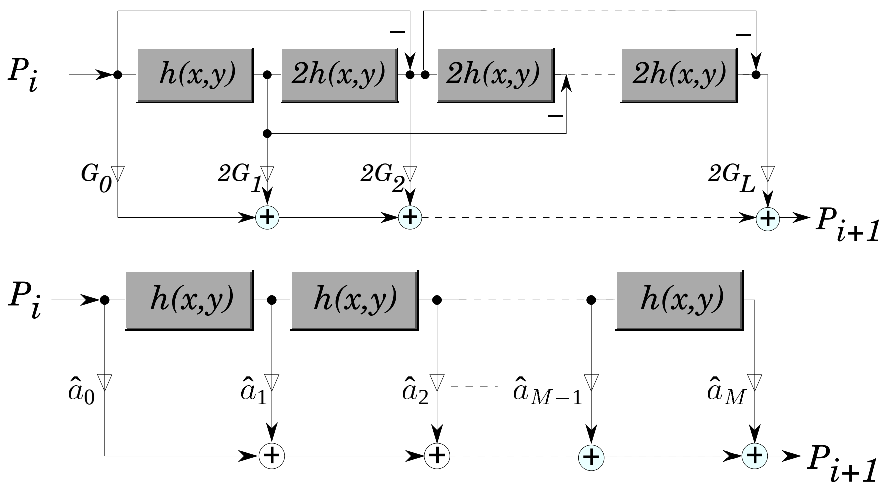


Figure 2: *The Chebychev recursion scheme (top) and the series expansion (bottom) in* $\cos(k_r \Delta x)$*. The* $h(x,y)$ *boxes represent the 2-D McClellan transformation of* $\cos(k_r)$*,* $G_l$ *represents the coefficients of the 1-D convolution operator. Note the simple structure of the series expansion in comparison with the Chebychev recursion scheme.*

### Implementation

In recursive depth extrapolation the (complex) computations are carried out from the $xy$ plane at depth $z$ to the $xy$ plane at depth $z + \Delta z$. The secondary cache of the MIPS R10k, which is used in the Origin 2000, has a size of 4 MB and can contain one $xy$ plane (of complex numbers) of $724 \times 724$ samples or 2 planes of $512 \times 512$. Larger or more depth planes will cause cache misses and introduces a higher latency on load/store operations. The R10K has 32 64-bit floating point registers and can do one multiply-add and one load/store operations at every clock-cycle. Taking these hardware constraints into account the implementation of the different extrapolation methods was carried out:

- The implementation of the 2D spatial convolution operators uses the symmetry in $x$ and $y$ explicitly. To reduce misses from the cache the data is rearranged in such a way that the convolution can be done within the cache.

```
for (iy = 0; iy < ny; iy++) {
    for (ix = 0; ix < nx; ix++) {
        for (j = 0; j < opersize; j++) {
            data[iy*nx+ix] += (tmp3[index3+j] + tmp4[index4+j])*hopx[j];
}   }   }
```

- The Hale McClellan method is implemented as a Chebychev recursion scheme as shown in Figure 2. The recursion scheme is not very cache friendly because three copies of the $xy$ planes are needed to calculate the plane at $z + \Delta z$. The basic scheme is given by:

```
for (o = 0; o < order; o++)
    for (iy = 0; iy < ny; iy++)
        for (ix = 0; ix < nx; ix++)
            term3[iy*nx+ix]  = 2.0*term2[iy*nx+ix] - term1[iy*nx+ix];
            data[iy*nx+ix] += op[o]*term3[iy*nx+ix];
}   }   }
```

  Note that all arrays in these calculations are complex. The term[1,2,3] arrays contain the pre-computed results of the small 2D convolutions of $\cos(k_r \Delta x)$ at different orders.

- The series expansion in $\cos(k_r \Delta x)$, also shown in Figure 2, is a straightforward implementation of the small 2D convolutions without the extra storage as needed in the Chebychev recursion scheme.

```
for (o = 0; o < order; o++) {
    for (iy = 0; iy < nyo; iy++) {
        for (ix = 0; ix < nxo; ix++) {
            data[iy*nx+ix] += a_m[o]*term1[iy*nx+ix];
}   }   }
```

| Method | load | store | add | mult | madd | max. perf. |
|---|---|---|---|---|---|---|
| Conv | $\frac{6}{4}(M+1)^2$ | $\frac{2}{4}(M+1)^2$ | $\frac{2}{4}(M+1)^2$ | | $\frac{4}{4}(M+1)^2$ | $\frac{10}{16} = 62\%$ |
| McC (3x3) | $26(L-2)+24$ | $8(L-1)$ | $16(L-1)$ | $6(L-1)+2$ | $6(L-1)$ | $\frac{34L-32}{68L-72} \lessapprox 50\%$ |
| Series (3x3) | $20(J-1)+4$ | $4J$ | $16(J-1)$ | $6(J-1)+2$ | $4(J-1)+2$ | $\frac{30J-26}{52J-48} \lessapprox 58\%$ |

The table above shows the kind and number of operations for every $(x, y)$ point in the $xy$ plane of the most inner-loop in the convolution. From this table it can be concluded that the 2D convolution is bound by the number of load/store operations. Note that it is possible to implement the 2D convolution in such a way (by not using the symmetry in the operator) that it will be floating point bound. Although this code will run closer to maximum performance, it will also use more operations and it will take more time for the same task. The Hale McClellan scheme is also bound by the number of load/store operations and can only be run at 50% of the floating point peak performance. The series expansion is the only scheme which is bound by the number of floating point operations.

**Performance and scalability**

| Method | operator (s) | migration (s) | IO+comm (s) | total (s) | Mflop/s | max. perf. | |
|---|---|---|---|---|---|---|---|
| Conv | 115.3 | 1610.9 | 1.0 | 1727.2 | 302 | 61 % | - |
| McC | 0.1 | 1017.0 | 1.2 | 1018.3 | 181 | 36 % | - |
| Series | 33.5 | 586.7 | 0.8 | 621.0 | 226 | 45 % | - |
| CPU's | operator (s) | migration (s) | misc. (s) | total (s) | Mflop/s | max. perf. | Speedup |
| 1 | 115.3 | 1610.9 | 1.0 | 1727.2 | 302 | 61 % | 1.00 |
| 2 | 58.1 | 800.5 | 1.4 | 859.9 | 607 | 61 % | 2.00 |
| 4 | 29.9 | 403.1 | 2.3 | 435.3 | 1200 | 61 % | 3.97 |
| 8 | 16.8 | 200.4 | 2.3 | 219.5 | 2380 | 60 % | 7.87 |
| 16 | 9.7 | 100.3 | 2.7 | 112.7 | 4634 | 58 % | 15.4 |
| 32 | 6.2 | 50.3 | 3.7 | 60.2 | 8674 | 54 % | 28.7 |

In the above table the performance of the discussed methods is shown. Comparing the max. perf. column with the theoretical max. perf. shows that the Series and McC method can still be improved. From the scaling results, using frequency parallelism with the 2D convolution method (the bottom part of the table), it is observed that up to 16 processors the algorithm scales good. The relative poor scaling for 32 processors is due to the small computational time for every processor compared to the time the IO (non-parallel; $0.6$ s) and communication takes ($4^{th}$ column). The operator calculation time has also a negative effect on the scaling, because the operator tabels are computed for all the frequencies the CPU handles. In this way the frequency boundaries between the processors are calculated twice. This means that for 1 CPU 131 operators are calculated, but for 32 CPU's 205 operators are calculated.

In the table below a simplified summary is given. The columns in this table have the following meaning:
• *circular*: the circularity of the impulse response.
• *operator*: the amount of cycles needed to compute all the operator coefficients needed in the convolution scheme. In the table ++ means a minimum time.
• *implementation*: the simplicity of the implementation.
• *performance*: the performance of the scheme on the Origin 2000.

| **method** | **accuracy** | **circular** | **operator** | **implementation** | **performance** |
|---|---|---|---|---|---|
| Conv (33x33) | ++ | + | - | ++ | □ |
| Conv (25x25) | + | + | □ | ++ | + |
| McC 17x(3x3) | □/+ | □ | + | □ | □ |
| McC 9x(5x5) | - | □ | ++ | □ | □/+ |
| Series 17x(3x3) | □/+ | □/+ | □ | ++ | ++ |
| Series 9x(5x5) | - | □ | + | ++ | ++ |

## Conclusion

From the three methods discussed in this abstract the 2D convolution requires the most multiplications and additions but also gives the highest accuracy and makes best use of the system hardware. The performance of the Hale McClellan method suffers from the fact that three copies of the depth stencil are needed for one extrapolation step. The series expansion overcomes this problem and can be implemented more efficiently.

### References


Blacquière, G., Debeye, H. W. J., Wapenaar, C. P. A., and Berkhout, A. J., 1989, 3D table-driven migration: Geophysical Prospecting, **37**, no. 8, 925–958.

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

Hale, D., 1991, 3-D depth migration via McClellan transformations: Geophysics, **56**, 1778–1785.

Soubaras, R., 1996, Explicit 3-D migration using equiripple polynomial expansion and Laplacian synthesis: Geophysics, **61**, 1386–1393.

Thorbecke, J. W., 1997, Common focus point technology: Ph.D. thesis, Delft University of Technology.

Wapenaar, C. P. A., and Berkhout, A. J., 1989, Elastic wave field extrapolation: Elsevier, Amsterdam.