Stanford Exploration Project, Report 125, January 16, 2007, pages 209–217

How much precision do we need in wave-equation based migration?

Robert G. Clapp

INTRODUCTION

Seismic processing has always been computationally intensive. Data problems are growing in size, there is a continual push for faster processing turn around times, and a need for more sophisticated (computationally demanding) methodologies. As a result the industry is always interested in new technologies that promise significant speedups.

Three technologies that are currently look promising are Floating Point Gate Arrays (FPGAs)(He et al., 2004), Graphical Processing Units (GPUs), and the Cell processor. All of these technologies, at least in theory, promise at least an order of magnitude speed up compared to a traditional processor, either today or in the near future. A primary bottleneck with each technology is the time involved in transfering data from main memory, through the processor, to the hardware, and back. There are two different ways to address this problem, a faster interconnect or transfering less data. Currently there is significant work being done to provide a faster interconnect[1], but these are generally theoretical at this stage and only reduce, but don't eliminate the problem. Therefore it is worth attacking the problem of reducing the amount of data that needs to be transfered.

Donoho and Polzer (1999) discuss methodologies for compressing sesimic data. A more basic approach is to build on the sine-bit recording experience(Cochran, 1973). It was shown that by just recording whether the earth was moving up or down, a good representation of the earth could be generated (Houston and Richard, 2004).

In this paper I attempt to test the degradation in image quality of reducing the precsion of a downward continuation based migration(Gazdag and Sguazzero, 1985) . I show that nearly identical results can be achieved by using $1/4$ of the number of bits to describe the input. In addition I demonstrate that the migration result is only marginally affected by reducing the precision within the migration algorithm. I conclude by commenting on what these results mean the viability of FPGAs for migration.

[1] http://www.gridtoday.com/03/0609/101530.htm

INPUT DATA PRECISION

To understand why we can reduce the precision of our input data without meaningful loss in final image quality it is important to remember that migraton is summing along a surface in multi-dimensional space. Imagine the process of forming an image m at a given ix,iy, and iz. To form this one point in image space involves a sumation over a five-dimensional (t,h_x,h_y,m_x,m_y) input space of the data $\mathbf{d}$ multiplied by the Green's function $\mathbf{G}$,

$$m(ix,iy,iz) = \sum_{m=1}^{ndhx} \sum_{l=1}^{ndhy} \sum_{k=1}^{ndmx} \sum_{j=1}^{ndmy} \sum_{i=1}^{nt} G(i,j,k,l,m,ix,iy,iz) d(i,j,k,l,m), \tag{1}$$

where $ndhx$, $ndhy$, $ndmx$, $ndmy$, and nt are the maximum number of samples of the data in all five dimensions. In reality $\mathbf{G}$ is limited by aperature range in space and only has a few non-zero elements along the time axis, but still we are summing over a very large number of points to form a single output location.

When we reduce the precision of our data what we are really doing is introducing an error in each data sample, as a result Equation(1) becomes

$$m(ix,iy,iz) = \sum_{=1}^{ndhx} \sum_{j=1}^{ndhy} \sum_{k=1}^{ndmx} \sum_{l=1}^{ndmy} \sum_{m=1}^{nt} G(i,j,k,l,m,ix,iy,iz)(d(i,j,k,l,m)+e(i,j,k,l,m)), \tag{2}$$

where e is the error associated with reducing the data precision. When we reduce the precision we are quantizing our data. The quantization process is zero mean and has a standard deviation of $\frac{q^2}{12}$ where q is our quantization interval. $\mathbf{G}$ has relatively low amplitude variation so should not emphasize the quantization error in any coherent manner. For this analysis we can think of rewriting Equation (2) as

$$m(ix,iy,iz) = \sum_{i=1}^{ndhx} \sum_{j=1}^{ndhy} \sum_{k=1}^{ndmx} \sum_{l=1}^{ndmy} \sum_{m=1}^{nt} G(i,j,k,l,m,ix,iy,iz) d(i,j,k,l,m) + G_{\text{approx}} \sum_{i}^{n} e(i), \tag{3}$$

where n is the number of non-zero elements of $\mathbf{G}$ and G_{approx} is a scalar with the mean non-zero value of $\mathbf{G}$. Most error analysis theory assumes that our errors have a normal rather than an uniform distribution, so we won't get quite the same level of error reduction, but generally the error $\mathbf{e}_{\text{mig}}$ in our migration result by quantizing are data should be a little higher than

$$\begin{aligned} e_{\text{mig}} &= \frac{\sigma}{\sqrt{n}} \\ &= \frac{q^2}{12\sqrt{n}} \end{aligned} \tag{4}$$

The size of n is going to depend on what type of problem we are doing. If we want velocity information, our image space has an offset or angle axis n will be smaller and we will have to have a smaller quantization step to obtain an equivalent image.

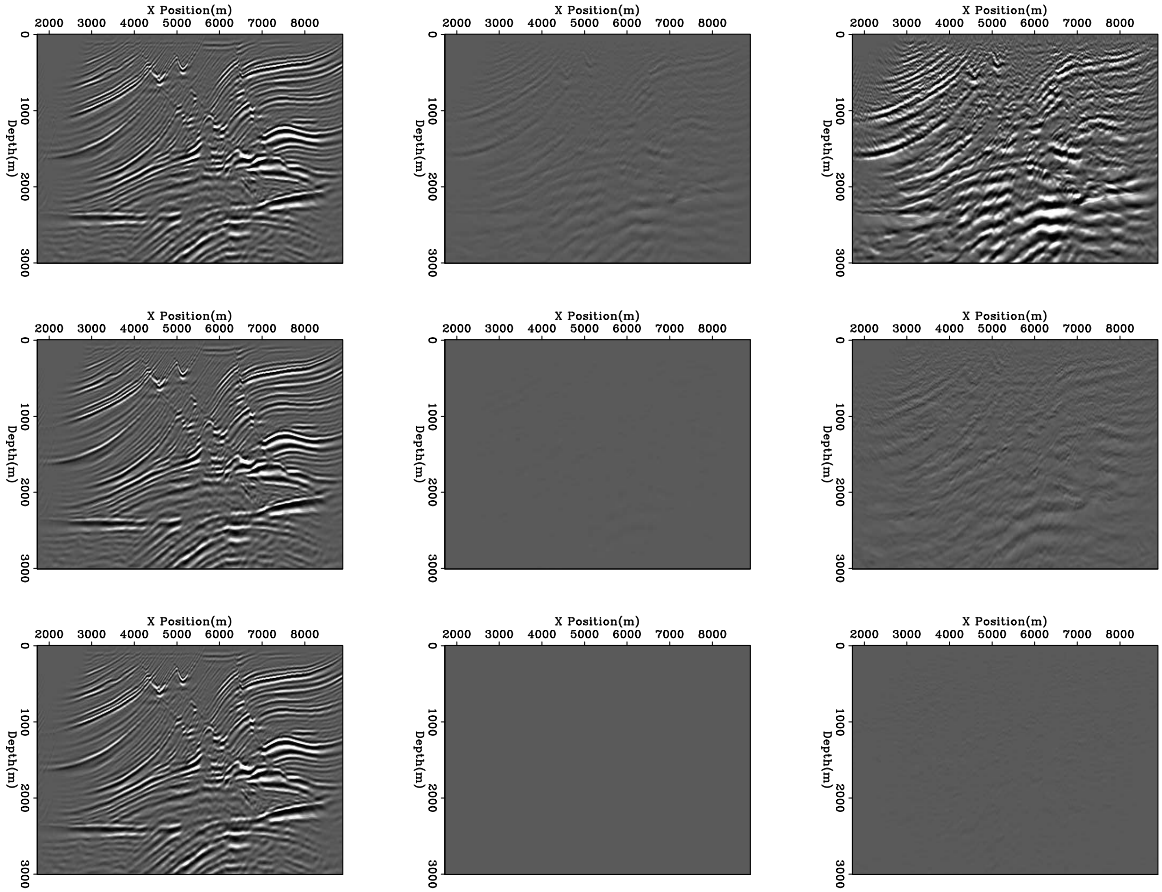

Figure 1: The result of migrating with a reduced precision input. The left collumn is the migration result. The center collumn is the difference between the full precission migration and the reduced precission image clipped at the same level is the migration. The right collumn is the same as the center panel, clipped at 1/10 the value. The top row is quantizated at .25 (two bits)., the center row at .0625 (four bits), and the bottom row at .015625 (six bits). bob2-bitinput [ER,M]

DATA QUANTIZATION RESULTS

To test the validity of reducing the data precission I simulated a reduced byte representation by first normalizing the dataset so its maximum value is one and then using various quantization intervals. Figure 1 shows three different quantization levels. The left collumn is the migration result. The center collumn is the difference between the full precission migration and the reduced precission image clipped at the same level as the migration. The right collumn is the same as the center panel but clipped at 1/10 of the value. The top row is quantizated at .25 (two bits)., the center row at .0625 (four bits), and the bottom row at .015625 (six bits). At two bits you can see some very minor differences in structure and a significant difference in amplitude the quanitized image. At four bits the error is only noticable at 1/10th the clip, and at six bits the error has almost completely disapeared.

If we assume that our migration aperture is 200 samples with 50 offsets and 2 non-zero elements in **G** in time we get an error reduction of about 140 through the migration process. The image on the screen is an eight bit representation of the migration result. The data is clipped at 40% of the maximum value. As a result we should need $\frac{256}{.4} = 640$ quantization intervals without the migration process and a little more than $\frac{640}{140} \approx 5$ after. Which is confirmed by our test. Two bits wasn't sufficient four bits was. If we decrease our clip (right panel) by a factor of ten we should need a little more than $\frac{640*10}{140} \approx 46$ intervals which explains why the error is elimiated using 6 bits.

If we form angle gathers rather than just the zero offset image we are summing over very few points in the offset space to form each angle. If we assume each angle is on averge formed from two offsets we get an error reduction of $\sqrt{20*2*2} = 30$. So an average should take two more bits to form a good angle domain image. Figure 2 confirms this hypothesis. The collumns follow the same structure as Figure 1. The top row is equivalent to 4 bit representation, the center row 6 bit, and to bottom row 8 bit. As predicted two more bits are needed. The top panel shows significant error at the migration clip, the center panel only shows error at $\frac{1}{10}$ the clip and the bottom panel shows no meaningful error.

For 3-D problems the number of bits that are needed should further reduce. Instead of summing over 15000 points to form a single image space in this small 2-D synthetic the number quickly jumps to more than 10000000, increasing the error reduction factor to more than 3000 for post-stack images and 300 for prestack images. As a result even less precision is needed to achieve the same result. A significant caveat is where to begin the quantization intervals. In areas with very strong noise (ground roll) or a strong reflector (salt) a clip before introducing the quantization intervals is a good idea.

INTERNAL PRECISION

The above sections assumed that the data was automatically converted to a floating point representation throughout the migration process. If you can and want to completely implement your migration on an external hardware device than the initial reduced bit precision is acceptable. The more likely scenario is that part of the migration process will be implemented on the processor and part on an external hardware device. FPGAs, GPUs, and the cell processor excel at certain tasks, but perform quite poorly at others. For example, implementing something that is 'if' statetement intensive on an FPGA is not efficient. As a result quantization of the input data alone is not sufficient.

Downward continuing a wavefield in a $v(x,z)$ medium has three basic computational parts that are constantly repeated: multiplication by a correction factor, Fast Fourier Transform (FFT), and multiplication by a complex exponential. At least two, and potentially all three, would be good candidates for an FPGA. Other poritions of the downward continuation process such as managing wavefields to handle multiple reference velocities would be inappropriate. As a result what we need to limit are internal data representation so we can reduce the number of bits that need to be sent to and from the external hardware device. This makes the error analysis of the last section much more complicated.

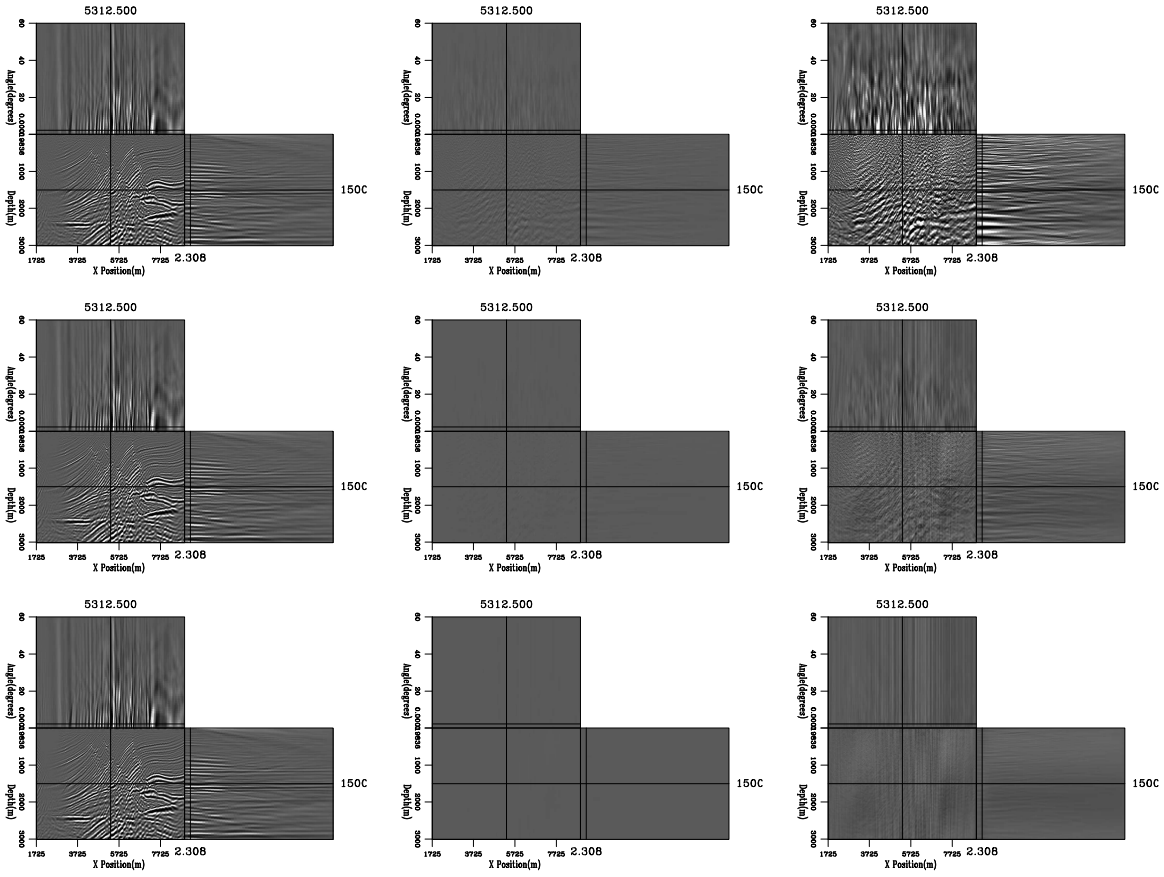

Figure 2: The result of migrating with a reduced precision input. The left collumn is the migration result. The center collumn is the difference between the full precission migration and the reduced precission image clipped at the same level is the migration. The right collumn is the same as the center panel, clipped at 1/10 the value. The top row is equivalent to 4 bit representation, the center row 6 bit, and to bottom row 8 bit. bob2-angle [ER,M]

The fact that we only need to produce an image that is accurate to about 1000 quantization intervals gives us some hope. The FFT operation is a series of intergals, so the same error analysis can be performed. We are not summing over a large number of points at every step in the downward continuation process so we will need more precision than before. Figure 3 is the same design as Figures **??** and 2. The top row simulates six bits, the center 8 bits, and the bottom 9 bits. In this case it takes 9 bits to get the error down to less than 1%.

To produce acceptable angle gathers requires significantly more bits. Figure 4 is in the same form as the previous figures. The top row simulates nine bits, the center 11 bits, and the bottom row 12 bits. To get the error below two percent requires a significant bit representation, but for almost all applications the nine bit representation is more than acceptable.

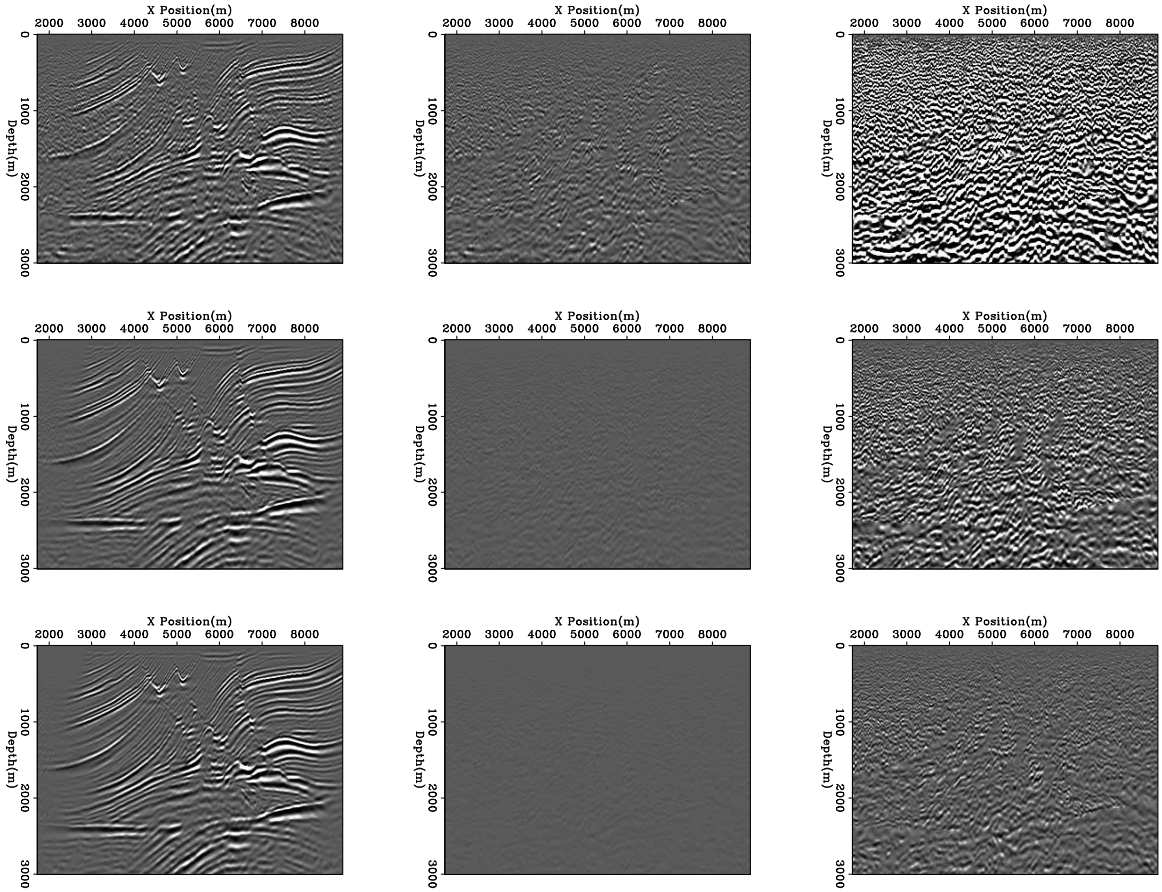

Figure 3: The result of migrating with a reduced precision input. The left collumn is the migration result. The center collumn is the difference between the full precission migration and the reduced precission image clipped at the same level is the migration. The right collumn is the same as the center panel, clipped at 1/10 the value. The top row simulates six bits, the center 8 bits, and the bottom 9 bits. bob2-ints [ER,M]

FPGA

The quantization approach used in this paper has an added advantage when dealng with current FPGA hardware. FPGAs, like early processors, are much more efficient on integer operations, than floating point operations. In addition they provide much more precision flexiblity than traditional processors. Floating point arrithmtic can be replaced by fixed-point integer math. Early work suggests that FFT portion of migration can be sped up by a factor of 3-10 given current hardware.

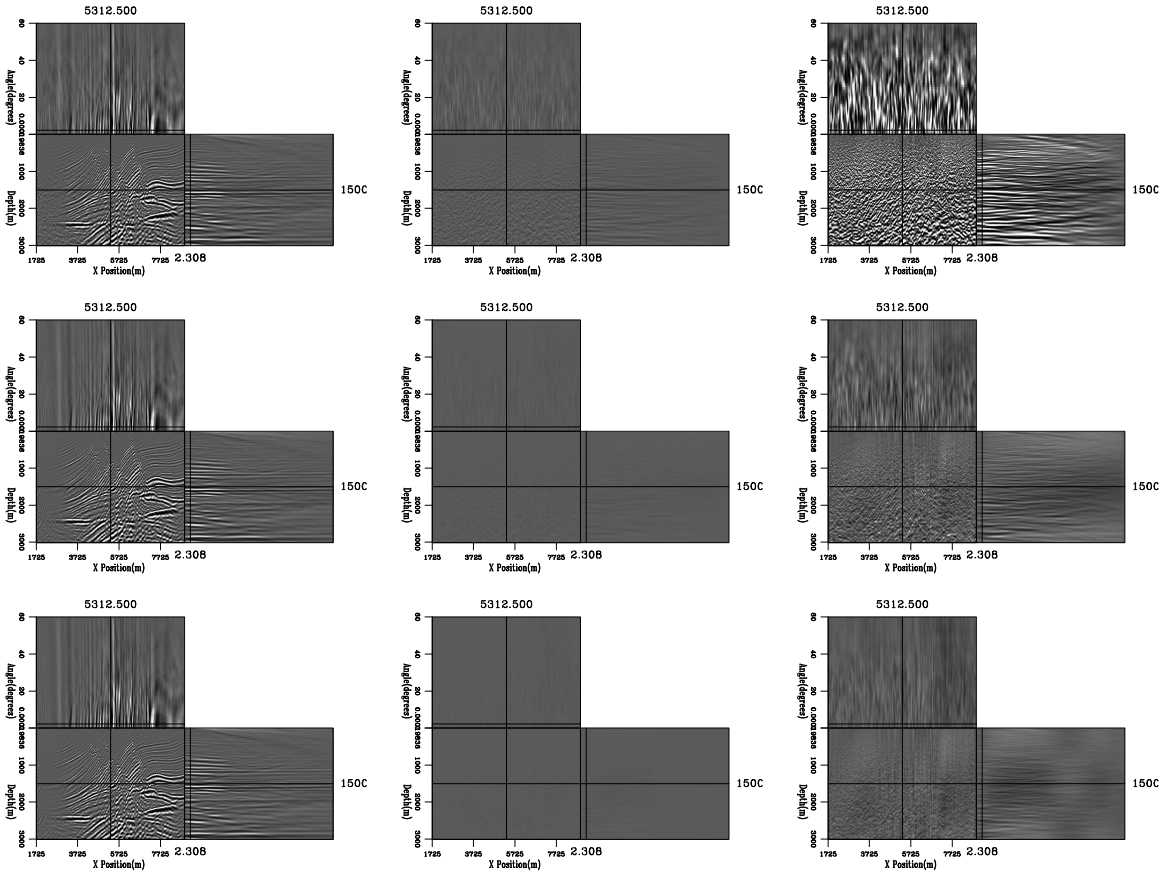

Figure 4: The result of migrating with a reduced precision input. The left collumn is the migration result. The center collumn is the difference between the full precission migration and the reduced precission image clipped at the same level is the migration. The right collumn is the same as the center panel, clipped at 1/10 the value. The top row simulates nine bits, the center 11 bits, and the bottom row 12 bits. bob2-ang [ER,M]

CONCLUSIONS

In this paper I showed that data at 32 bit floating bit precision is not needed to form an accurate migration image. It takes only four bits to form an acceptable zero offset image and six bits for a prestack image. The representation of the wavefield within the migration can also be reduced by at least a factor of two. These results offer promise in making external hardware processing solutions, such as FPGAs, GPUs, and the cell processor, effective in the near term.

REFERENCES

Cochran, M. D., 1973, Seismic signal detection using sign bits: Seismic signal detection using sign bits:, Soc. of Expl. Geophys., Geophysics, 1042–1052.

Donoho, R. E., P. and R. Polzer, 1999, Development of seismic data compression methods

for reliable, low-noise performance:, *in* 69th Annual International Meeting Soc. of Expl. Geophys., 1903–1906.

Gazdag, J. and P. Sguazzero, 1985, Migration of seismic data by phase shift plus interpolation, *in* Gardner, G. H. F., Ed., Migration of seismic data: Society Of Exploration Geophysicists, 323–330.

He, C., M. Lu, and C. Sun, 2004, Accelerating seismic migration using FPGA-based coprocessor platform: 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, IEEE, Expanded Abstracts, 207–216.

Houston, L. M. and B. A. Richard, 2004, The helmholtzâĂŞkirchoff 2.5d integral theorem for sign-bit data: The helmholtzâĂŞkirchoff 2.5d integral theorem for sign-bit data:, Journal of Geophysics and Engineering, 84–87.