Stanford Exploration Project, Report 125, January 16, 2007, pages 1–??

Improved flattening with geological constraints

Jesse Lomask

ABSTRACT

A Discrete Cosine Transform (DCT) flattening method is used to precondition a constrained flattening method. In this method, the linear step of a Gauss-Newton flattening method with geological constraints is solved with preconditioned conjugate gradients. The preconditioner utilizes DCTs to invert a Laplacian operator. Memory and computational cost savings from the use of the DCT make this method the most efficient constrained flattenening algorithm to date. A 3D faulted field data example is presented.

INTRODUCTION

As described by Lomask and Guitton (2006), flattening algorithms often need to be constrained. Although a key selling point of flattening without picking (Lomask et al., 2006) is that it requires no picking, it is useful to have the ability to add some geological constraints to restrict the flattening result in areas of poor data quality while allowing it to efficiently tackle other areas where the dips are accurate. Furthermore, constraints can also be used to force the flattening result to conform to data points besides dip alone such as correlations across faults.

In this paper, I present a flattening method with hard constraints that exploits Discrete Cosine Transforms (DCTs) to increase computational efficiency. It is a modification of the constrained Gauss-Newton flattening method (Lomask and Guitton, 2006) using an improved preconditioner. The preconditioner, an unconstrained flattening method that uses DCTs, was presented by Lomask and Fomel (2006). Instead of approximating the inverse of the Laplacian with the helical transform, the DCT is exploited to more accurately invert the Laplacian. The resulting algorithm converges faster than previous constrained flattening methods while the memory usage is similar. Here, I first review the constrained Gauss-Newton flattening method. I then demonstrate its use on a faulted 3D field data set from the Gulf of Mexico.

METHODOLOGY

Constraints were first applied to the flattening problem in Lomask and Guitton (2006). In the 3D Gauss-Newton flattening approach, data is flattened by iterating over the following equations iterate {

$$\mathbf{r} = [\nabla_\epsilon \boldsymbol{\tau}_k(x,y,t) - \mathbf{p}_\epsilon(x,y,\boldsymbol{\tau}_k)] \quad (1)$$

$$\Delta\boldsymbol{\tau} = (\nabla_\epsilon^T \nabla_\epsilon)^{-1} \nabla_\epsilon^T \mathbf{r} \quad (2)$$

$$\boldsymbol{\tau}_{k+1}(x,y,t) = \boldsymbol{\tau}_k(x,y,t) + \Delta\boldsymbol{\tau} \quad (3)$$

} ,
where the subscript k denotes the iteration number, $\boldsymbol{\tau}$ is the estimated time-shift found in the least-squares sense so that its gradient matches the dip $\mathbf{p}$, and ϵ is an adjustable regularization controlling the amount of conformity in the time (or depth) dimension. ∇_ϵ is a 3D gradient operator with an ϵ parameter. The residual update in equation (1) can be rewritten as

$$\mathbf{r} \;=\; \nabla_\epsilon \boldsymbol{\tau} - \mathbf{p}_\epsilon \;=\; \begin{bmatrix} \frac{\partial \boldsymbol{\tau}}{\partial x} \\ \frac{\partial \boldsymbol{\tau}}{\partial y} \\ \epsilon \frac{\partial \boldsymbol{\tau}}{\partial t} \end{bmatrix} - \begin{bmatrix} \mathbf{b} \\ \mathbf{q} \\ \mathbf{0} \end{bmatrix}. \quad (4)$$

I wish to add a model mask $\mathbf{K}$ to prevent changes to specific areas of an initial $\boldsymbol{\tau}_0$ field. This initial $\boldsymbol{\tau}_0$ field can be picks from any source. In general, they may come from a manually picked horizon or group of horizons. These initial constraints do not have to be continuous surfaces but instead could be isolated picks, such as well-to-seismic ties. To apply the mask I follow a similar development to Claerbout (1999) as

$$\mathbf{0} \approx \nabla_\epsilon \boldsymbol{\tau} - \mathbf{p}_\epsilon \quad (5)$$

$$\mathbf{0} \approx \nabla_\epsilon(\mathbf{K} + (\mathbf{I} - \mathbf{K}))\boldsymbol{\tau} - \mathbf{p}_\epsilon \quad (6)$$

$$\mathbf{0} \approx \nabla_\epsilon \mathbf{K}\boldsymbol{\tau} + \nabla_\epsilon(\mathbf{I} - \mathbf{K})\boldsymbol{\tau} - \mathbf{p}_\epsilon \quad (7)$$

$$\mathbf{0} \approx \nabla_\epsilon \mathbf{K}\boldsymbol{\tau} + \nabla_\epsilon \boldsymbol{\tau}_0 - \mathbf{p}_\epsilon \quad (8)$$

$$\mathbf{0} \approx \mathbf{r} = \nabla_\epsilon \mathbf{K}\boldsymbol{\tau} + \mathbf{r}_0 - \mathbf{p}_\epsilon. \quad (9)$$

The resulting equations are now iterate {

$$\mathbf{r} = \nabla_\epsilon \mathbf{K}\boldsymbol{\tau}_k - \mathbf{p}_\epsilon(\boldsymbol{\tau}_k) + \mathbf{r}_0 \quad (10)$$

$$\Delta\boldsymbol{\tau} = (\mathbf{K}^T \nabla_\epsilon^T \nabla_\epsilon \mathbf{K})^{-1} \mathbf{K}^T \nabla_\epsilon^T \mathbf{r} \quad (11)$$

$$\boldsymbol{\tau}_{k+1} = \boldsymbol{\tau}_k + \Delta\boldsymbol{\tau} \quad (12)$$

} .

Because the mask $\mathbf{K}$ in equation (11) is non-stationary, I solve it using preconditioned conjugate gradients. The preconditioner efficiently exploits Discrete Cosine Transforms(DCTs) to invert a Laplacian operator (Lomask and Fomel, 2006) as:

$$\Delta\boldsymbol{\tau} \approx \mathrm{DCT_{3D}}^{-1}\left[\frac{\mathrm{DCT_{3D}}\left[\nabla_\epsilon{}^T \mathbf{r}\right]}{J}\right], \quad (13)$$

where $\mathrm{DCT_{3D}}$ is the 3D discrete cosine transform and

$$J = -2(\cos(w\Delta x) + \cos(w\Delta y)) + 2\epsilon^2(1 - \cos(w\Delta t)) + 4. \quad (14)$$

J is the real component of the Z-transform of the 3D finite difference approximation to the Laplacian with adjustable regularization.

Following an approach for weighted 2D phase unwrapping by Ghiglia and Romero (1994), I define $\mathbf{Q} = \mathbf{K}^{\mathrm{T}}\boldsymbol{\nabla}_\epsilon^{\mathrm{T}}\boldsymbol{\nabla}_\epsilon\mathbf{K}$ and $\bar{\mathbf{r}} = \mathbf{K}^{\mathrm{T}}\boldsymbol{\nabla}_\epsilon^{\mathrm{T}}\mathbf{r}$ so that equation (11) becomes

$$\Delta\boldsymbol{\tau} \quad = \quad \mathbf{Q}^{-1}\bar{\mathbf{r}}. \tag{15}$$

For each non-linear Gauss-Newton iteration, the least-squares solution to equation (11) is solved by iterating over the following preconditioned conjugate gradients computation template:

$$
\begin{array}{l}
\text{iterate } \{ \\
\quad \text{solve } \mathbf{z}_k = (\boldsymbol{\nabla}_\epsilon^{\mathrm{T}}\boldsymbol{\nabla}_\epsilon)^{-1}\bar{\mathbf{r}}_k \text{ using equation (13)} \\
\quad \text{subtract off reference trace } \mathbf{z}_k = \mathbf{z}_k - \mathbf{z}_k(ref) \\
\quad \text{if first iteration } \{ \\
\qquad \mathbf{p}_1 \longleftarrow \mathbf{z}_0 \\
\quad \} \text{ else} \{ \\
\qquad \beta_{k+1} \longleftarrow \bar{\mathbf{r}}_k^{\mathrm{T}}\mathbf{z}_k / \bar{\mathbf{r}}_{k-1}^{\mathrm{T}}\mathbf{z}_{k-1} \\
\qquad \mathbf{p}_{k+1} \longleftarrow \mathbf{z}_k + \beta_{k+1}\mathbf{p}_k \\
\qquad \} \\
\quad \alpha_{k+1} \longleftarrow \bar{\mathbf{r}}_k^{\mathrm{T}}\mathbf{z}_k / \mathbf{p}_{k+1}^{\mathrm{T}}\mathbf{Q}\mathbf{p}_{k+1} \\
\quad \Delta\boldsymbol{\tau}_{k+1} \longleftarrow \Delta\boldsymbol{\tau}_k + \alpha_{k+1}\mathbf{p}_{k+1} \\
\quad \bar{\mathbf{r}}_{k+1} \longleftarrow \bar{\mathbf{r}}_k - \alpha_{k+1}\mathbf{Q}\mathbf{p}_{k+1} \\
\quad \}
\end{array}
$$

Here, k is the iteration number starting at *k=0*. It is necessary to subtract off the reference trace at each iteration because the Fourier based solution using equation (13) has no non-stationary information such as the location of the reference trace. Subtracting the reference trace removes a zero frequency shift. Nonetheless, in practice equation (13) makes an adequate preconditioner because $\boldsymbol{\nabla}_\epsilon^{\mathrm{T}}\boldsymbol{\nabla}_\epsilon$ is often close to $\mathbf{K}^{\mathrm{T}}\boldsymbol{\nabla}_\epsilon^{\mathrm{T}}\boldsymbol{\nabla}_\epsilon\mathbf{K}$.

Convergence can be determined using the same criterion described in Lomask et al. (2006). Alternatively, an adequate stopping criterion would be to consider only the norm of the residual $\|\bar{\mathbf{r}}_k\|$. This is essentially the average of the divergence of the remaining dips, i.e. ,

$$\frac{\|\bar{\mathbf{r}}_k\|}{n} \quad < \quad \mu, \tag{16}$$

where $n = n_1 \times n_2 \times n_3$ is the size of the data cube.

Constrained solution with weights

In dealing with noise and certain geological features such as faults and angular unconformities, it is necessary to apply a weight to the flattening method. This weight is applied to the residual to ignore fitting equations that are affected by the bad dips estimated at faults. In the case of angular unconformities, it can be used to disable the vertical regularization in locations where multiple horizons converge. The resulting weighted and constrained Gauss-Newton equations are now iterate {

$$\mathbf{r} = \mathbf{W}\nabla_\epsilon \mathbf{K}\boldsymbol{\tau}_k - \mathbf{W}\mathbf{p}_\epsilon(\boldsymbol{\tau}_k) + \mathbf{W}\mathbf{r}_0 \quad (17)$$

$$\Delta\boldsymbol{\tau} = (\mathbf{K}^{\mathrm{T}}\nabla_\epsilon^{\mathrm{T}}\mathbf{W}_\epsilon^{\mathrm{T}}\mathbf{W}\nabla_\epsilon\mathbf{K})^{-1}\mathbf{K}^{\mathrm{T}}\nabla_\epsilon^{\mathrm{T}}\mathbf{W}_\epsilon^{\mathrm{T}}\mathbf{r} \quad (18)$$

$$\boldsymbol{\tau}_{k+1} = \boldsymbol{\tau}_k + \Delta\boldsymbol{\tau} \quad (19)$$

} .

Again, I solve equation (18) efficiently using preconditioned conjugate gradients with equation (13) as the preconditioner and $\mathbf{Q} = \mathbf{K}^{\mathrm{T}}\nabla_\epsilon^{\mathrm{T}}\mathbf{W}_\epsilon^{\mathrm{T}}\mathbf{W}\nabla_\epsilon\mathbf{K}$ and $\bar{\mathbf{r}} = \mathbf{K}^{\mathrm{T}}\nabla_\epsilon^{\mathrm{T}}\mathbf{W}_\epsilon^{\mathrm{T}}\mathbf{r}$.

CONSTRAINED RESULTS

I conducted tests of this constrained flattening method on a 3D data set from the Gulf of Mexico. This example illustrates how a single pair of traces can be manually correlated and passed to the constrained flattening method to reconstruct across faults.

In Figure 1 is a 3D Gulf of Mexico data set provided by Chevron. The manually interpreted fault model is displayed in Figure 2. Two faults are identified in the figure. Fault 1 has part of its tip-line encased within the cube as can be observed by its termination in the time slice. Fault 2, on the other hand, does not terminate within the data cube. Because Fault 1 terminates within the data cube, no constraints need to be provided to flatten across it, however, Fault 2 requires some picking. In this case, I picked one vertical pair of traces across Fault 2. Then I applied the weighted Gauss-Newton method with the weight **W** being the picked fault model. The binary mask **K** is ones for unconstrained model locations and zeros for constrained model locations. The initial model $\boldsymbol{\tau}_0$ is the picked pair of traces correlating across Fault 2.

Figure 3 is flattened volume of the data in Figure 1. Notice the horizons are reconstructed across both faults. Notice the faint outline of a channel that is annotated on the figure. Another view of the same cube is displayed in Figure 4. Several stratigraphic features are reconstructed across both faults. In addition to a fault model, the only picks required were from a single trace of correlations across Fault 2. Also, the $\boldsymbol{\tau}$ field used to flatten this data is displayed in Figure 5.

In Figure 6 every 25th tracked horizon of the Gauss-Newton constrained flattening method in displayed. Notice the overlain horizons track their respective events across both faults. In short, I reconstructed this 3D volume by correlating a single pair of traces across a fault. It should be pointed out that an automatic fault indicator could substitute for the fault model, reducing the amount of manual interpretation even further.

../jesse1/Figs/shoal_flt.ps

Figure 1: Gulf of Mexico data. The black lines superimposed onto the orthogonal sections identify the location of these sections: a time slice at time=1.584 s, an in-line section at y=3203 m, and a cross-line section at x=3116 m. The reference trace is located at x=2828 m and y=1775 m.

../jesse1/Figs/shoal_flt_wt.ps

Figure 2: As Figure 1 displaying the manually picked fault model used for flattening.

../jesse1/Figs/shoal_flt_flat.ps

Figure 3: As Figure 1 only after flattening. Notice that reflectors on both sides of both faults are properly reconstructed. Also, notice a sinusoidal channel that is annotated on the horizon slice.

../jesse1/Figs/shoal_flt_flat2.ps

Figure 4: As Figure 3 except displaying a different horizon slice, time=1.504 s. Several stratigraphic channel features are visible.

../jesse1/Figs/shoal_flt_tau.ps

Figure 5: As Figure 1 showing the τ field used for flattening.

../jesse1/Figs/shoal_flt_pck.ps

Figure 6: As Figure 1 only displaying every 25th tracked horizon of the Gauss-Newton constrained flattening method. The fault model (solid black) was manually picked.

CONCLUSIONS

In this paper, a preconditioned constrained flattening method is introduced that exploits Discrete Cosine Transforms to invert a Laplacian operator. This method is efficient in both its memory use and computational time.

As demonstrated here, the ability to incorporate some picking allows the reconstruction of horizons across faults that cut across the entire data cube. An interpreter can pick a few points on a 2D line and then flatten the entire 3D cube. With computational improvements in both the algorithm and hardware, this method could be applied on the fly, as the interpreter adds new picks.

ACKNOWLEDGMENT

I would like to thank Brian Bergquist and Chevron for supplying the data used in this paper.

REFERENCES

Claerbout, J., 1999, Geophysical estimation by example: Environmental soundings image enhancement: Stanford Exploration Project, `http://sepwww.stanford.edu/sep/prof/`.

Ghiglia, D. C. and L. A. Romero, 1994, Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods: Optical Society of America, **11**, no. 1, 107–117.

Lomask, J. and S. Fomel, 2006, Flattening with cosine transforms: SEP–**124**.

Lomask, J. and A. Guitton, 2006, Flattening with geological constraints: SEP–**124**.

Lomask, J., A. Guitton, S. Fomel, and J. Claerbout, 2006, Flattening without picking: Geophysics, **71**, no. 4, 13–20.