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

# Interval velocity estimation through convex optimization

*Ben Witten and Michael Grant*

**ABSTRACT**

Convex optimization is an optimization technique which maximizes efficiency by fully harnessing the convex nature of certain problems. Here we test a convex optimization solver on a least-squares formulations of the Dix equation. Convex optimization has many useful traits including the ability to set bounds on the solution which are explored here. As well, this example serves as a test for the feasibility of convex optimization for future, more expensive tomography problems.

## INTRODUCTION

Interval velocity estimation is a fundamental problem in seismology. The simplest technique for finding interval velocity is the Dix equation (Dix, 1952) which analytically inverts the root-mean-square (RMS) velocity for interval velocity. The Dix equation has many flaws including the assumption of a vertically stratified earth and numerical problems that can cause the inversion to become unstable for rapidly varying velocities. To better constrain the solution, the Dix equation is often cast as a least-squares problem, which is regularized in time with a differential operator that penalizes rapid variations to produce a smooth result (Clapp et al., 1998).

Valenciano et al. (2003) expanded on this work to use both $\ell_2$ and $\ell_1$ regularization. The $\ell_2$ regularization is justified when the expected velocity model is smooth. When geological expectations dictate abrupt changes in interval velocity $\ell_1$ regularization can be utilized to preserve sharp boundaries when they are present, yet allows for smooth velocity elsewhere.

Since least-squares problems are already convex, this is a perfect problem to test the utility of convex optimization. Here we utilize convex optimization to solve the problem of interval velocity estimation using the same examples as those present by Valenciano et al. (2003) for the $\ell_2$ and $\ell_1$ regularization, which used conjugate gradient methods. As well, bounds will be enforced on the solution to further constrain the Dix inversion to a more geologically sensible answer.

## LEAST-SQUARES DIX EQUATION

The Dix equation is a nonlinear relationship between RMS velocity and interval velocity. It is, however, linear in the square of the interval velocity. This linearized formulation of the Dix equation was solved by Clapp et al. (1998) by using a preconditioned least-squares optimization with spatial smoothness constraints. In this approach or data fitting goal is to minimize the residual of

$$\mathbf{W}(\mathbf{C}\mathbf{u}-\mathbf{d}) \approx \mathbf{0} \tag{1}$$

where $\mathbf{u}$ is a vector whose components range over vertical traveltime depth $\tau$ and whose values are the interval velocity squared $v_{int}^2$. $\mathbf{d}$ is the data vector which has the same range as $\mathbf{u}$, but whose values are the scaled RMS velocity squared $\tau v_{RMS}^2/\Delta\tau$ where $\tau/\Delta\tau$ is the index on the time axis. $\mathbf{C}$ is the casual integration operator. And $\mathbf{W}$ is a weight matrix which is proportional to our confidence in RMS velocities.

Since the fitting goal, equation 1, is unstable when there are high frequency variations in RMS velocity, a regularization term is added to penalize this erratic behavior. As done by Valenciano et al. (2003), first order derivatives are used. This system of equations is

$$\begin{aligned} \mathbf{W}(\mathbf{C}\mathbf{u}-\mathbf{d}) &\approx \mathbf{0} \\ \epsilon_\tau \mathbf{D}_\tau \mathbf{u} &\approx \mathbf{0} \\ \epsilon_x \mathbf{D}_\mathbf{x} \mathbf{u} &\approx \mathbf{0} \end{aligned} \tag{2}$$

where $\mathbf{D}_\tau$ and $\mathbf{D}_\mathbf{x}$ are first-order finite differences derivatives in traveltime and midpoint, respectively. $\epsilon_\tau$ and $\epsilon_x$ balance the relative importance of the two model residuals, effectively controlling the smoothness.

The approach taken towards the regularization terms will determine whether a smooth or discontinuous model is found. $\ell_2$ regularization will produce a smooth result. If a discontinuous velocity is geologically expected, such as for carbonates, $\ell_1$ regularization can be used to produce a blocky model (Valenciano et al., 2003).

## CONVEX OPTIMIZATION

A problem is a convex optimization problem if it has the form

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } f_i(x) \le b_i,\ i = 1,\ldots,m \end{aligned} \tag{3}$$

where $f_0,\ldots,f_m$ and $b_1,\ldots,b_m$ are convex functions and $x$ exists in a convex set, $\mathbf{S}$ (Boyd and Vandenberghe, 2004). Convex optimization problems have many attractive features including the guarantee of local minima to be global and strong optimality, feasibility, and sensitivity information. As well, there are reliable and efficient numerical algorithms to solve these problems. Since this Dix formulation is a least-squares problem, which is already convex, it seems natural to try convex optimization techniques to find the solution.

The data fitting goal, equation (1), can be rewritten in optimization notation as

$$\text{minimize } \|\mathbf{W}(\mathbf{C}\mathbf{u}-\mathbf{d})\|_2 \tag{4}$$

where $\|\cdot\|_2$ means the least-squares norm. When necessary regularization terms are added the full set of goals, equation (2) becomes

$$\text{minimize } \|\mathbf{W}(\mathbf{Cu}-\mathbf{d})\|_\mathbf{2} + \|\epsilon_\tau \mathbf{D}_\tau \mathbf{u}\|_\mathbf{i} + \|\epsilon_\mathbf{x} \mathbf{D}_\tau \mathbf{u}\|_\mathbf{i} \tag{5}$$

in optimization notation. If $i$ is 2 then an $\ell_2$ regularization is used and a smooth model is obtained. If $i$ is 1 then an $\ell_1$ regularization is used and the result is a blocky model, instead.

Even after inversion, there may be points in the model space which do not make geological sense, usually due to picking errors caused by poor resolution. Convex optimization allows for bound constraints to be imposed on the solution, which can correct for such inconsistencies. If we constrain the solution then we have

$$\begin{aligned} \text{minimize } \|\mathbf{W}(\mathbf{Cu}-\mathbf{d})\|_\mathbf{2} + \|\epsilon_\tau \mathbf{D}_\tau \mathbf{u}\|_\mathrm{i} + \|\epsilon_\mathbf{x} \mathbf{D}_\tau \mathbf{u}\|_\mathrm{i} \\ u \leq v_{\max} \\ u \geq v_{\min}, \end{aligned} \tag{6}$$

where $v_{\max}$ and $v_{\min}$ are the square of the maximum and minimum allowable velocity models, respectively.

To do the convex optimization, **cvx** (Grant et al., 2006) will be used. **cvx** is a MATLAB based system for solving convex optimization problems. It allows constraints and objectives to be specified with common MATLAB syntax.

## REAL DATA EXAMPLES

125 CMP's from a 2-D prestack data set that was acquired in the Gulf of Mexico were used. Since the Gulf of Mexico usually exhibits flat reflectors, this is suitable for the Dix equation. The region is also faulted which implies discontinuous velocities.

The approach taken for obtaining the RMS velocities is the same as that in (Valenciano et al., 2003), but will briefly be recreated here. First, velocity analysis was preformed on each CMP. Then an auto-picker was used to pick the maximum stacking power corresponding to the best RMS velocity at each CMP position. An example velocity analysis with picks for a single CMP is shown on the top of Figure 1. The velocity values from all the CMPs can be combined to form a complete RMS velocity model space. This is shown on the bottom of Figure 1. Please note that the velocities picked in Figure 1 are in slowness rather than velocity, while the complete RMS model space is in velocity. This is because the conjugate gradient method used a slowness model, but the convex optimization failed with slowness and velocity had to be used. This limitation may be because the values were all close to 0 forcing the solution down or due to the narrow range of slowness values. Thus the images computed with conjugate gradients were computed in slowness and inverted to velocity, while the convex optimization images were computed directly with velocity values. Figure 2 shows the stacked section which displays the faulting mentioned above. The middle panel has the faults highlighted and the bottom panels shows the same faults on the raw RMS velocity. It is interesting to note that the raw RMS velocity shows the faults clearly.

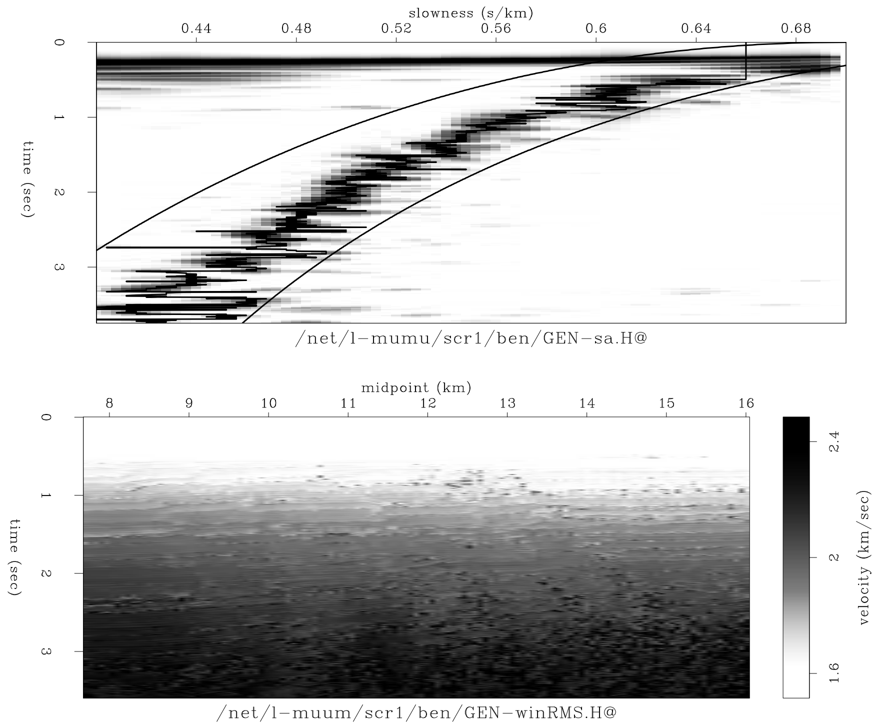



Figure 1: Top: Auto-picked RMS velocity of one CMP from a 2-D prestack dataset. Bottom: Raw RMS velocity map for all 125 CMPs ben1-vel-vrms [ER]

The top of Figure 3 shows the interval velocity resulting from solving equations (2) with $\ell_2$ regularization. The bottom of Figure 3 shows the interval velocity when equation (5) is solved with $\ell_2$ regularization. Note that all interval velocities are clipped at their respective maximums. Both images in Figure 3 are very similar showing that convex optimization is at least equivalent to conjugate gradients in terms of quality of solution.

If we now look at the solutions to equations (2) and (5) solved with a $\ell_1$ regularization, shown in Figure 4, we can see that, as expected, a much blockier solution is found. As in the previous figure, the top panel of Figure 4, created by conjugate gradients, is very similar to the bottom image, solved with convex optimization.

In the $\ell_1$ regularization image, the faults do show up faintly. If we overlay the same lines shown in Figure 2 onto Figure 4, this becomes more obvious as shown in Figure 5. The faults may be slightly more obvious in the problem solved with **cvx**, which is blockier than with conjugate gradients. The difference in “blockiness” is due to the different $\epsilon$’s and how they are applied in each case.

There appears to be some low velocity anomalies near the bottom of the interval velocity solution. This is predominately seen in the blocky models, but there are uncharacteristically low velocities at late times in all the models. As we can see from the stack in Figure 2, there is no evidence to support such velocities. To correct this we can constrain the solution further by adding bounds when solving the convex optimization problem. If we assume that the interval

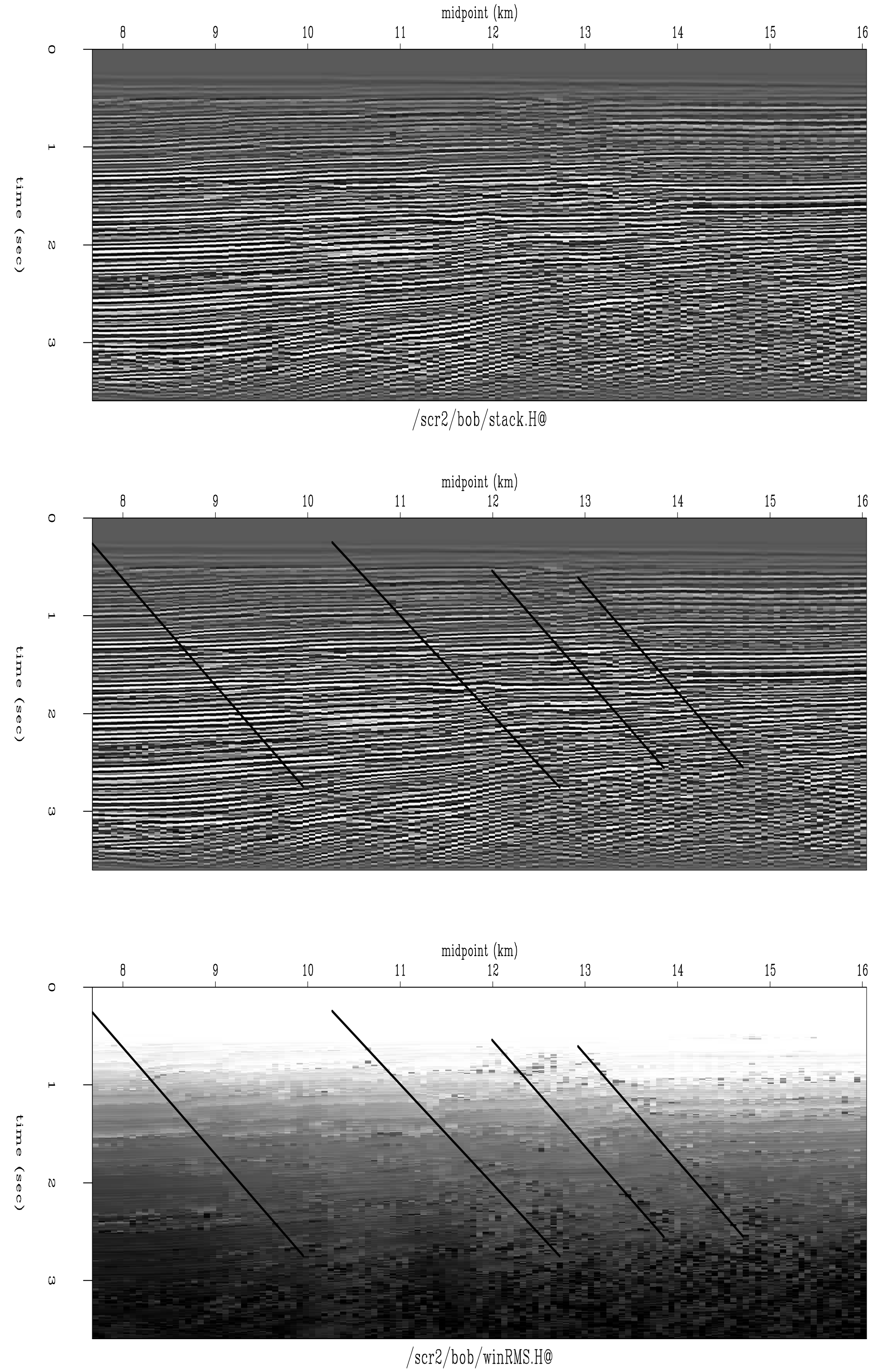



Figure 2: Top: Stacked data using the raw RMS velocity. Middle: Stack data with lines showing the faults. Bottom: RMS velocity with same lines showing the faults. ben1-stack1 [ER]

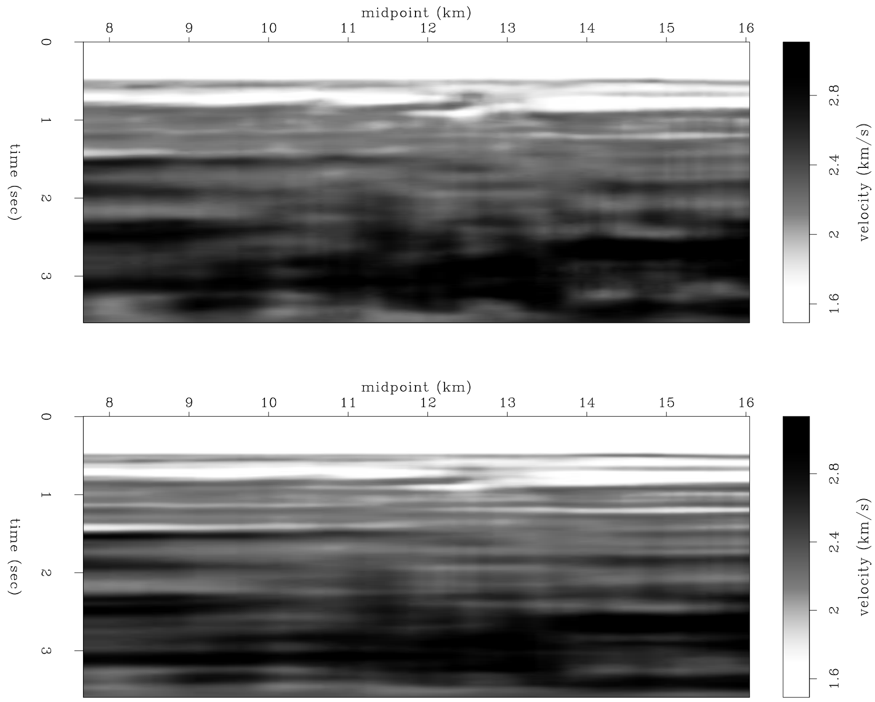



Figure 3: Top: Interval velocity computed using conjugate gradients with $\ell_2$ regularization. Bottom: Interval velocity computed using convex optimization with $\ell_2$ regularization. ben1-L2vint [CR]

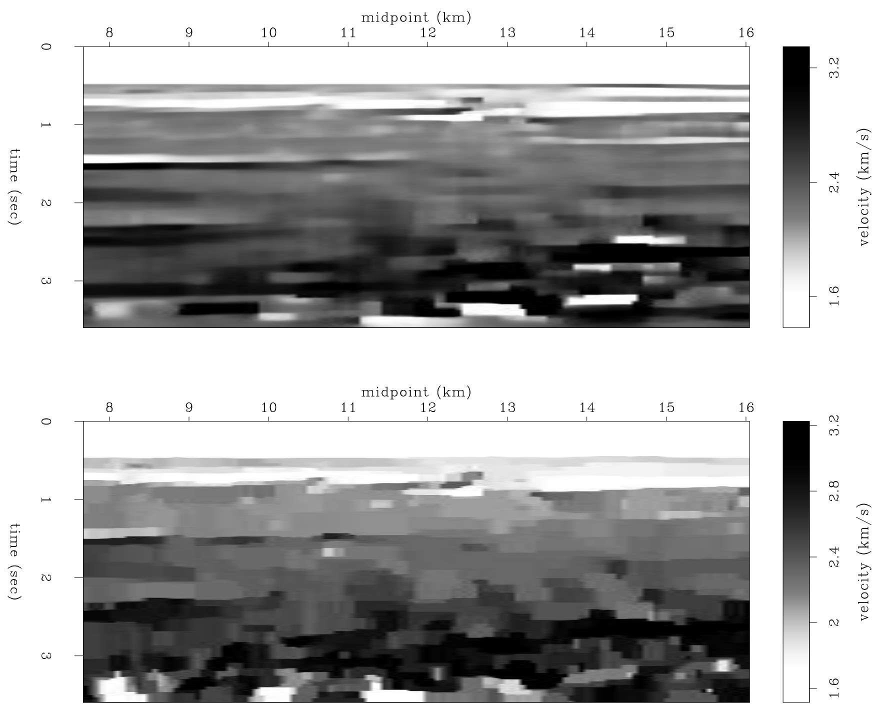
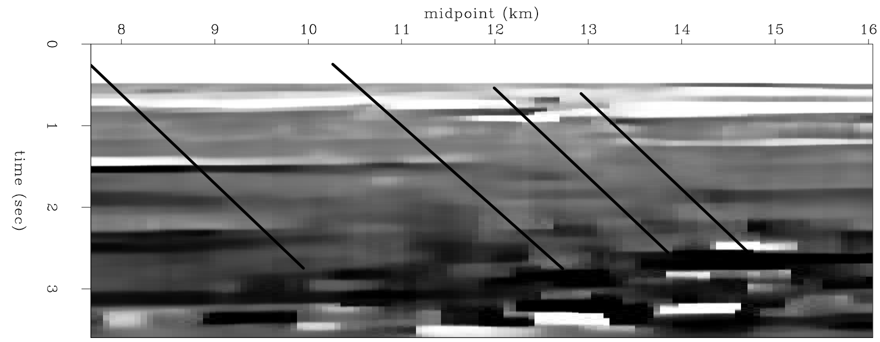



Figure 4: Top: Interval velocity computed using conjugate gradients with $\ell_1$ regularization. Bottom: Interval velocity computed using convex optimization with $\ell_1$ regularization. ben1-L1vint [CR]



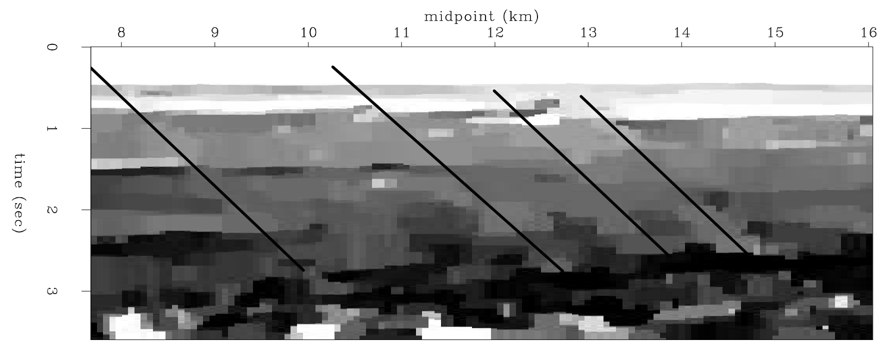




Figure 5: Same as Figure 4 except with lines marking fault locations. ben1-L1-interp [CR]

velocity $v(z)$ increases linearly with depth:

$$v(z) = v_0 + \alpha z, \tag{7}$$

then we can get a general estimate of the interval velocity. This is shown in Figure 6. Velocities 20 percent above and below this model are used as the maximum and minimum constraints in equation (6). Figures 7 and 8 show the bounded constrained $\ell_2$ and $\ell_1$ solutions, respectively. As we can see, the low velocities occurring at late times have been attenuated.

## CONCLUSIONS

Convex optimization methods show promise for solving least-squares problems. As exemplified by the least-squares Dix equation, convex optimization can yield similar results to those obtained through conjugate gradient methods. Yet convex optimization also has the advantage of imposing geologically constrained bounds to further enhance the solution. The $\ell_1$ **cvx** solution also shows a slightly better ability to pick up the faults than the conjugate gradient method. As stated above, this could simply be a function of $\epsilon$ choice.

The convex optimization solver may not be as fast as conjugate gradient methods, but the solution obtained is guaranteed to be correct. The conjugate gradient solution is usually obtained by iterating until we are tired. The convex solver, on the other hand, works until a preset accuracy is achieved. In these problems, this precision was set at $10^{-9}$. The efficiency

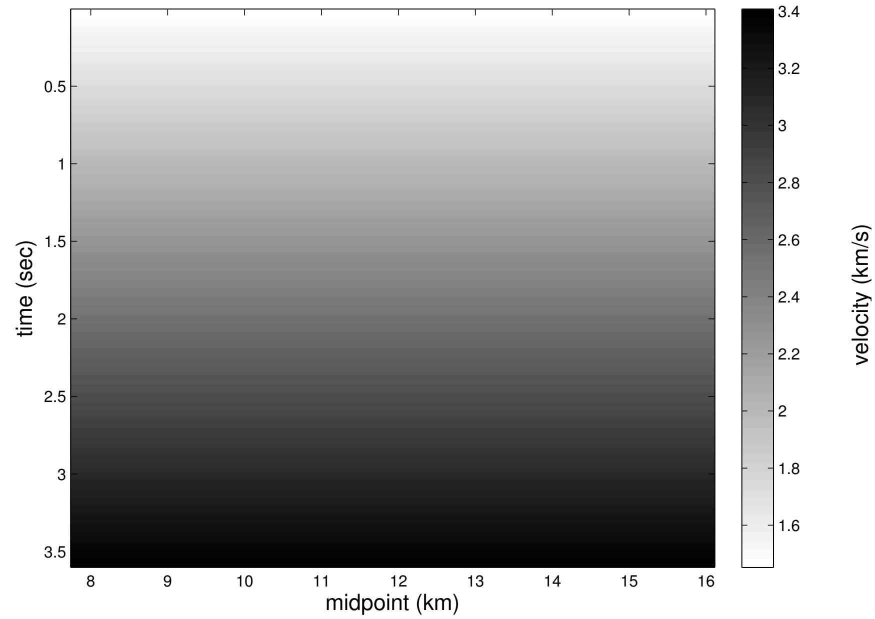



Figure 6: Model from which the upper and lower velocity constraints are formed. Upper is 20% greater than this everywhere and the lower is 20% less. ben1-bounds [CR]

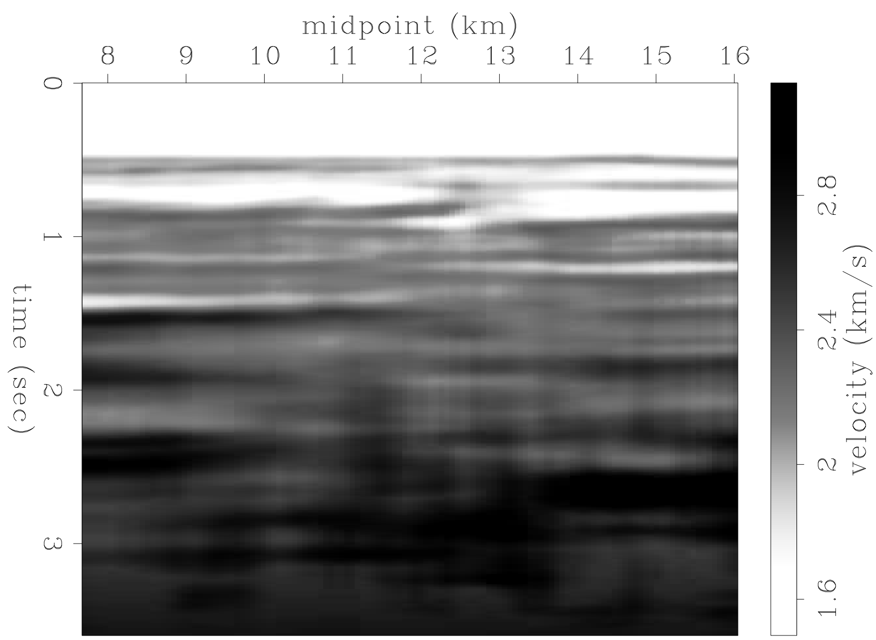



Figure 7: Solution by convex optimization with $\ell_2$ regularization and bound constraints. ben1-L2vint-cvx-bounded [CR]

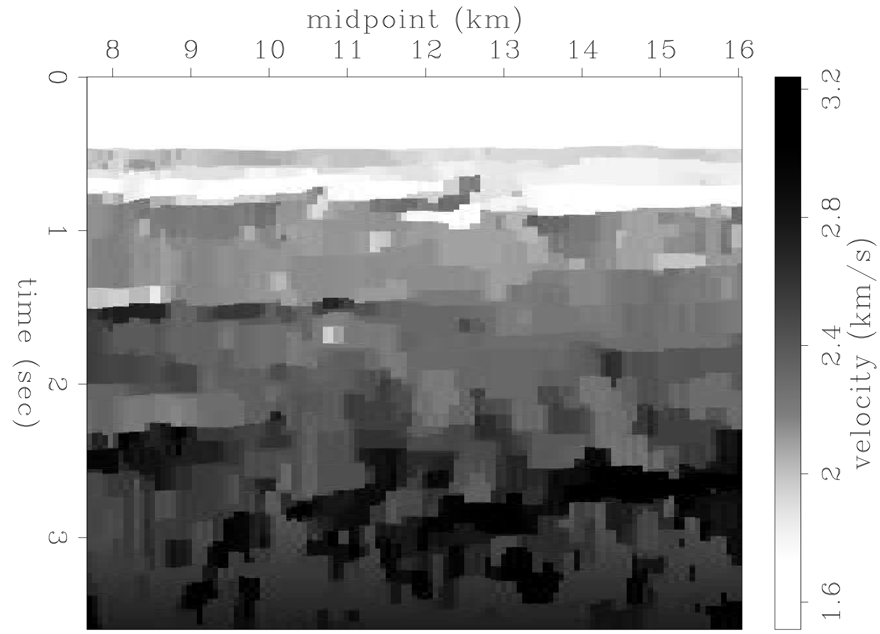



Figure 8: Solution by convex optimization with $\ell_1$ regularization and bound constraints. ben1-L1vint-cvx-bounded [CR]

of $\ell_2$ versus $\ell_1$ regularization is quite striking. It takes 8 times more iterations to do the $\ell_1$ than the $\ell_2$, but only 3 times as many when both are bounded. This is difference comparable to that of conjugate gradients.

To apply convex optimization larger problems and more complex operators, a convex optimization solver that does not rely on MATLAB is needed. While the **cvx** software is efficient and easy to use, it is limited by MATLAB's efficiency and lack of memory. If a new solver can be created convex optimization could be successful for future endeavors.

## ACKNOWLEDGMENTS

We would like to thank Bob Clapp for his discussion and advice on this project and Biondo Biondi for bringing us together to attempt convex optimization on SEP problems.

## REFERENCES

Boyd, S. and L. Vandenberghe, 2004, Convex optimization: Cambridge University Press.

Clapp, R. G., P. Sava, and J. F. Claerbout, 1998, Interval velocity estimation with a null-space: SEP–**97**, 147–156.

Dix, C. H., 1952, Seismic prospecting for oil: Harper and Brothers.

Grant, M., S. Boyd, and Y. Ye, 2006, cvx: Matlab software for disciplined convex programming: http://www.stanford.edu/ boyd/cvx/.

Valenciano, A. A., M. Brown, M. D. Sacchi, and A. Guitton, 2003, Interval velocity estimation using edge-preserving regularization: SEP–**114**, 136–150.