3D Block Travel Time Inversion

Victor Pereyra
Weidlinger Assoc. Los Altos, CA

1 Abstract

We consider the solution of large scale travel time inversion problems by a domain decomposition approach. Both data and parameters are partitioned, giving rise to a number of sub-problems, which can be chosen small enough to be amenable to SVD based Marquardt type algorithms. A nonlinear block Jacobi or Gauss-Seidel iteration is employed to resolve the (weak) couplings between blocks. The key to the success of the approach is to use model parametrizations that are concise and local in nature (B-splines). Another important feature of the approach is that it is easily parallelizable. The performance of an implementation is demonstrated on some synthetic examples.

2 Introduction

We consider large scale nonlinear least squares problems whose objective functions are very expensive to evaluate, and whose Jacobian matrices are generally ill-conditioned and have an almost block diagonal structure.
For this class of problems, conventional iterative methods that require a large number of iterations to converge are impractical. Because of the ill-conditioning, we would like to use Singular Value Decompositions techniques as a useful tool for dynamic regularization. However, due to the large problem size, direct SVD based methods are also impractical.
On the positive side, the almost block diagonal structure facilitates the application of asynchronous block iterative methods that parallelize naturally. If the block sizes are chosen sufficiently small, then direct SVD methods are also feasible at the block level, and if the partitioning is done wisely it will provide an *a priori* regularization mechanism. Regularization can also be introduced at the block level, if necessary, producing a stable, robust, and efficient algorithm.
We show how some three-dimensional seismic travel time inversion calculations of significant practical importance belong to the class of problems that we are considering, and demonstrate the feasibility of our approach by implementing a complete algorithm which uses PVM to distribute the computation on various multi-processor configurations. The performance on several medium to large scale problems is also exemplified.
We have not found in the literature much previous work specifically for this type of problems, with the exception of the excellent research of Diniz-Ehrhardt and J.M. Martinez [1], which discusses a nonlinear version of a Cimmino type algorithm. This leads to a fairly different method from ours; note also that that research is concurrent with ours, first reported in [8] (see also [6, 9, 10]).
In the seismic inversion literature, large scale problems of this kind arise even in two-dimensions, because of the type of model parametrizations used. Typically, the methods used to solve the linearized problems are of the conjugate gradient type, and simplifications are made to avoid the large cost of recomputing full ray traced data. This often leads to artifacts and has resulted on generalized mistrust for the technique.
In the general minimization arena, the recent work of R. B. Schnabel and his collaborators [11], in the area of molecular dynamic problems shares also some of the flavour of our approach.

3 Block Gauss-Seidel iteration

We consider now the minimization of nonlinear functionals of special type by the Levenberg-Marquardt iterative method. Let $F : R^m \longrightarrow R^n$, and let $f(\mathbf{x}) = \frac{1}{2} \parallel F(\mathbf{x}) \parallel_2^2$ be twice continuously differentiable in an open convex set $D \subset R^m$. We also assume that the problem is ill-conditioned and large.
Let us consider the Singular Value Decomposition of $J(\mathbf{x})$:

$$J(\mathbf{x}) = USV^T,$$

and introduce the mollified pseudoinverse of J as:

$$\tilde{J}(\mathbf{x}, \mu) = VDS^{\dagger}U^T,$$

Here the diagonal matrix D is defined as:

$$D = diag\{d_i\} = diag\{\frac{k_i^2}{k_i^2 + \mu^2}\},$$

where $k_i = s_i/s_1$ are the normalized singular values of J. Thus, $\mu > 0$ provides a mollified cutoff point for the normalized singular values (see [4]). With this we can define the Levenberg-Marquardt iteration as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \tilde{J}(\mathbf{x}_k, \mu)^{\dagger} F(\mathbf{x}_k). \tag{1}$$

If n is large (> 100 say), direct Singular Value Decomposition methods will not be practical, and we will loose the advantage that the information on singular values gives to assess and handle the ill-conditioning inherent to these problems. In order to preserve this advantage, and also to open the door to distributed computing methods, we consider a domain decomposition approach for problems of this type.
The idea is to partition the data space so that for each block of observations only a small part of parameter space is relevant. We have exemplified in Pereyra (1995) how such models occur in geophysical applications. Schnabel (1995) has used similar ideas in molecular dynamics applications.
Our hypothesis regarding the essentially local character of the model parametrization, together with the judicious choice of the subsets of observations, imply a priori that there will be many zero columns in the Jacobian matrix, corresponding to parameters that do not influence the value of the residual norm at all for that subset of observations. This has the effect of bringing the sub-problems down to a manageable size.

4 SVD subspace selection

Jupp and Vozoff (1975) introduced the idea of relevant and irrelevant parameters, based on Singular Value Decomposition linearized analysis. They consider the *rotated parameters* (in tangent space):

$$\delta \mathbf{p} = s_1 \mathbf{V}^{\mathbf{T}} \delta \mathbf{x},$$

and show their influence on the variation of the misfit functional:

$$\| \delta \mathbf{F} \|_2^2 = \sum_{i=1}^{r} k_i^2 {\delta p_i}^2.$$

This equation implies that the parameters δp_i that are associated with singular values that are small relative to s_1, contribute little to variations in the misfit functional.

That is the key to our algorithm for partitioning the parameter set in blocks, given a partition of the data set.

1. Given a block of data, $\mathbf{d^k}$ with m^k elements, and a base point $\mathbf{x}$, calculate $\mathbf{J}(\mathbf{x})$ for this data set (i.e., $\mathbf{J}$ has only m^k rows).
2. Delete all zero columns of $\mathbf{J}$ and calculate its Singular Value Decomposition.
3. Inspect the columns of $\mathbf{V}$, and select the largest components in absolute value. Choose the indices of the variables in parameter space corresponding to these entries to form the subset IC_k of parameters that are most influenced by the data set $\mathbf{d}^k$.

The different subsets of parameters may in general overlap. Observe also that it is possible for the union of all the subsets to be smaller than the entire set of parameters. This will indicate that there are parameters that cannot be resolved by the given data, at least in a neighborhood of the base point $\mathbf{x}$. Since this analysis is local, it should be periodically revised as the optimization process advances. Once this partition has been effected, we use an outer block nonlinear Gauss-Seidel iteration in order to obtain the solution of the full problem.

To make this more precise, let us partition the index sets $\bar{M} = [1, M], \bar{N} = [1, N]$, into the subsets $\{IR_k\}, \{IC_k\}$, i.e., the index sets that describe the partition of our problem into blocks. The sub-problems can now be written as:

$$\min_{\mathbf{x}^k \in \Re^k} \| \mathbf{F^k}(\mathbf{x^k}) - \mathbf{d^k} \|, \qquad (2)$$

where, $\mathbf{x}^k = \{x_j^k\}_{j \in IC_k}$, $\mathbf{d}^k = \{d_i^k\}_{i \in IR_k}$,

$$f_i^k(\mathbf{x^k}) = f_i(\mathbf{x}), i \in IC_k,$$

and $x_i = x_i^k, i \in \bar{N} - IC_k$. In other words, we fix the values of the parameters that are not in the vector $\mathbf{x^k}$. Observe that the dimension of the subproblems is then $m^k \times n^k$, where $n^k = |(IC_k)|$, $m^k = |(IR_k)|$ ($|\ .\ |$ stands for the number of elements in a set).

5 Inverting synthetic data

5.1 Recovery of variable velocity from reflection and VSP data

We illustrate now how the ideas above work by inverting some synthetic data. We consider a three-dimensional block limited by a planar free surface and a planar reflector at depth $z = 4\ kms$, for a domain with dimensions $10 \times 10 \times 5\ kms^3$. The P-velocity in this layer is given by a tri-cubic tensor product of B-splines, with $5 \times 5 \times 5$ control vertices all equal to $1.6\ kms/sec$, except for the plane $k = 3$ which is given by:

1.6	1.6	1.6	1.6	1.6
1.6	1.7	1.7	1.7	1.6
1.6	1.7	1.9	1.7	1.6
1.6	1.7	1.7	1.7	1.6
1.6	1.6	1.6	1.6	1.6

Thus we have a high velocity anomaly in the center of the region. As an starting guess we set all the control vertices $V_{ijk} = 1.4$. Thus, we have the wrong constant background and no hint of the anomaly. The data consists of travel times calculated by ray tracing on the target model for both a reflection and a VSP survey.

The reflection data corresponds to 36 shots in a square 6×6 array, with a 14×14 moving square array of receivers centered at the shots, for a total of 7100 arrivals. For the VSP data, we have a straight well located at $x = 5,\ y = 5$, where we position 11 equally spaced sources. Receivers are in the free surface in a 21×21 array uniformly distributed in the square $[3, 7] \times [3, 7]$, for a total of 4850 aditional arrivals.

We define the blocks by subdividing the data into individual shots and running our pre-processor code. The maximum rms error on a block for the initial guess is 0.863 sec. After a few block Gauss-Seidel sweeps we stop with a maximum rms of 1.6 msec. Inspecting the resulting velocity cube we see that the recovery is imperfect, although the rms indicates a good fit of the data. This is connected to the fact that the vertical components of velocity are not easily recoverable when the ray paths are mostly vertical.

To prove this and also to verify that the codes are correct, we restrict now the velocity to a $2\ 1/2D$ configuration, in which we keep the lateral variations of plane $k = 3$ but make it constant in z, effectively considering a laterally varying interval velocity (see [2] for a discussion and similar approach).

Since now we have only 25 parameters instead of 125, we coarsen our data set, taking less shots and receivers, for a total of 3300 data points. After just one sweep of block Gauss-Seidel we obtain a perfect recovery of the target model.

All the calculations have been performed in a single Sun 5 workstation. The first example took several hours of computing time, while the second problem took only 937 sec. of computing time. Obviously, this performance can

be greatly enhanced by using multiple workstations in a distributed version.

5.2 Recovery of a reflector. Marine survey

We want to consider now a larger problem that is not amenable to a direct full inversion. We consider model OILMID, which is a constant velocity two layers model, with a horizontal reflector at depth 4.0 kms. This reflector has some meandering channels with an average depth of 100 m. We use a representation for this reflector in terms of a 21×21 basis of B-splines, giving an initial parameter space dimension of $n = 441$. This is somewhat diminished by constraining the two most external rings of parameters to their correct values of 4.0 in order to avoid edge of model effects. This leaves 289 free parameters to be determined. The synthetic data is generated by ray tracing on the target model which has constant properties:

Region	v_p	v_s	ρ
1	3.0	1.8	2.0
2	4.5	2.1	2.3

The reflector is defined by:

$$R_2(x,y) = \sum_{i,j=1}^{21} V_{ij} B_i(x) B_j(y), \qquad (3)$$

where the coefficients V_{ij} are to be determined.
The 3D survey consists of 121 shots, with the moving receiver array containing 21×21 receivers for each shot, for a total of $56,600$ arrivals. After generating the synthetic data with ray tracing, and using BLKINV to define the 121 blocks, we start our Block Gauss-Seidel code from the flat background $\alpha^0 \equiv 3.8$, (with the exception of the fixed boundary values and an additional intermediary ring that is set to 3.9). After four full sweeps we obtain a fairly good recovery, with the exception of a localized feature where no correction was effected. It is quite remarkable that the block approach keeps that unresolved spot localized; we feel that a more global approach would tend to polute the solution away from the unresolved spot.
In Figure 1 we see an evolution of the inverse process in the form of contour plots of the target reflector for various iterations From an initial maximum residual of 0.16 in block #37, and after approximately six hours on a Sun SPARC 10 workstation, the final residual is reduced to 2.58×10^{-5}.

6 Distributed block Jacobi algorithm

We have written a preliminary version in PVM of a distributed block Jabobi iteration. Essentially, it assigns different blocks to different processors, updating a central version of the parameters as the block processes are completed.
We have run the inversion of model OILMID on a number of Hewlet-Packard 9000-735 workstations connected by a giga-switch at Sandia National Laboratory in Livermore, California. The Table below shows the wall-clock times for various configurations.

Processors	Time(mins.)
1	61.6
2	30.9
4	18.7
8	9.0

The results are encouraging, showing a good parallel efficiency, and clearly displaying how the combination of faster workstations and multiple processors can bring the time from 3,600 mins. on a SUN 10 to just 9 mins., for an improvement in wall-clock time by a factor of 400, with a minimal investment in code development. This code has also been ported to a Convex Parallel computer. We thank Dr. Juan Meza for his help with this part of the project.

7 Conclusions

We have described a travel time inversion system that can deal with complex geological structures. The system has been designed to handle large scale problems by using a distributed approach. A novel decomposition method is our tool to attack large scale problems in parallel.
The system is fully implemented and we have demonstrated its performance in some synthetic examples, showing its capabilities for integrating heterogeneous data sets coming from a variety of surveying approaches and also showing how it performs in a distributed environment.

References

[1] Diniz-Ehrhardt, M.A., and Martinez, J.M., (1993), *A Parallel Projection Method for Overdetermined Nonlinear Systems of Equations,* Numerical Algorithms 4, 241-262.

[2] Farra, V., and Madariaga, R. (1988), *Non-Linear Reflection Tomography*, Geophysical Journal 95, 135-147.

[3] Feng, D., and Schnabel, R.B. (1993), *Globally Convergent Parallel Algorithms for Solving Block Bordered Systems of Nonlinear Equations,* Optimization Methods and Software, 2, 269-295.

[4] Jupp, D.L.B., and Vozoff, K. (1975), *Stable Iterative Methods for the Inversion of Geophysical Data,* Geophys. J. R. astr. Soc. 42, 957-976.

[5] Koshy, M., Pereyra, V., and Meza, J.C., *Distributed Computing Applications in Forward and Inverse Geophysical Modeling.* In *Expanded Abstracts,* SEG 61st Annual Meeting, Houston, TX., 1991, 349-352.

[6] Koshy, M., Meza J., and Pereyra, V., 1995, *Asynchronous Global Optimization Techniques for Medium and Large Inversion Problems,* SEG 65 Annual Meeting, Houston. Extended Abstracts, 1091-1094.

[7] Pereyra, V. (1992), *Two Point Ray Tracing in General 3D Media*, Geophysical Prospecting, 40, 267-287.

[8] Pereyra, V. (1993), *Parallel Block Inversion of Geophysical Data*, Proceedings of Mathematical Methods in Geophysical Imaging, SPIE Intern. Symp. on Optical Applied Science and Eng., San Diego, CA, 2033, pp. 192-198.

[9] Pereyra, V. (1995), *Modeling, Ray Tracing, and Block Nonlinear Travel Time Inversion in 3D.* To appear in Pure and Applied Geophysics.

[10] Pereyra, V., (1995b) *Asynchronous Global Optimization Techniques for Medium and Large Inversion Problems. Extended Abstracts* SEG Annual Meeting, Houston, TX, pp. 1091-1094.

[11] Schnabel, R.B. (1995), *View of the Limitations, Opportunities, and Challenges in Parallel Nonlinear Optimization,* Parallel Computing, 21, 875-905.