Stanford Exploration Project, Report 100, April 20, 1999, pages 315–330

Extremal regularization[1]

William W. Symes[2]

keywords: *inversion, regularization, constrained optimization*

ABSTRACT

Extremal regularization finds a model fitting the data to a specified tolerance, and additionally minimizing an auxiliary criterion. It provides relative model/data space weights when no statistical information about the model or data is available other than an estimate of noise level. A version of the Moré-Hebden algorithm using conjugate gradients to solve the various linear systems implements extremal regularization for large scale inverse problems. A deconvolution application illustrates the possibilities and pitfalls of extremal regularization in the linear case.

INTRODUCTION

Many important inverse problems are ill-posed: precise fit to data is either impossible or produces a model estimate so sensitive to data error as to obscure physically important model features. Since no intrinsic sample-level distinction between signal and noise exists in general, solution of such problems requires specification or estimation of data noise level, or acceptable degree of data misfit. Even so, the set of "feasible" models (fitting the data to within the prescribed tolerance or noise level) is very large. Finding a model representing the information content of the data then requires additional information.

Under some circumstances, Bayesian estimation theory provides a computational prescription for selecting a maximum likelihood model which represents the information inherent in the data and computing its *a posteriori* variance. When the modeling operator is linear, the data statistics are known and Gaussian, and signal and noise are known to be statistically independent, noise variance is the only additional parameter required to set up a linear system for the maximum likelihood estimator. However if these statistical hypotheses are not satisfied or if the modeling operator is nonlinear, Bayesian theory does not give an explict prescription for selecting a representative model.

This paper explores an alternative selection principle, which I call *extremal regularization*. Extremal regularization does not require the extensive statistical assumptions of the Bayesian theory. It selects from the feasible set a model minimizing some auxiliary model

[1] This report will also appear in the TRIP 1999 Annual Report

[2] **email**: symes@caam.rice.edu

property. Use of extremal regularization requires (1) a choice of auxiliary model property to extremize, (2) choices of norms to measure the data misfit and auxiliary model property, (3) knowledge of the data noise level, and (4) an algorithm for finding an extremal solution. Depending on application, (1), (2), and (3) involve greater or lesser degrees of arbitrariness. When the modeling operator is linear and the auxiliary property is quadratic in the model, extremal regularization amounts to minimization of a quadratic function subject to a quadratic constraint. In effect such an algorithm finds the penalty parameter or relative weight between data and model spaces as a function of the prescribed noised level. Note that the noise level has a much more obvious intuitive or physical meaning than the penalty parameter, though it is not always easy to determine from available data.

This notion is not new in geophysics. For example D. D. Jackson proposed similar ideas more than 20 years ago (Jackson, 1973, 1976). From (Jackson, 1979):

> There are some who hold the recalcitrant point of view that the normal Backus-Gilbert resolving kernels tend to present results in too abstract a fashion, but that the use of *priori* data makes any error estimate rather arbitrary. For these, the only satisfactory evidence on which to base physical conclusions is a catalogue of models which fit the data well, are physically plausible, and contain among them models which have the maximum and minimum presence of some hypothetical feature. I must admit to having strong sympathies for this point of view.

Jackson extremized linear functions of the model subject to prescribed data misfit. These extrema represent the ends of model error bars.

This "recalcitrant" point of view is also natural when there is some nonlinear auxiliary quantity that should be minimized - or ideally even zeroed out - by virtue of fundamental model requirements of the model. This is the case for example in Claerbout's proposal for signal extraction *via* Jensen inequalities (Claerbout, 1998, 1992) and for the extended version of differential semblance optimization for velocities (Gockenbach and Symes, 1997). In other settings, for example the deconvolution problem used as an example in this report, extremizing an auxiliary quantity serves merely to pick out a "simplest" solution amongst many.

The Moré-Hebden algorithm finds the reelative weight between data and model spaces by applying Newton's method to the so called secular equation. The secular equation requires that the norm of the auxiliary model property be equal to its prescribed value. Since this norm will change as you change the relative weight between data misfit and auxiliary model property, the secular equation determines the weight. This idea is much used in numerical optimization, where quadratically constrained quadratic minimization goes under the name "trust region problem" (Dennis and Schnabel, 1983). The published versions of Moré-Hebden (Moré, 1977; Hebden, 1973), also (Björk, 1997), pp. 205-6, have typically used LU decompositions to solve the linear systems required by Newton's method, so are limited to small and medium scale problems. This report describes a version appropriate for large scale problems, using conjugate gradient iteration. The presence of this "inner" iteration and the necessary lack of precision in the solution of the Newton system has interesitng consequences for convergence of the algorithm.

This report presents extremal regularization of linear inverse problems in the form of the quadratically constrained quadratic minimization problem solved by the Moré-Hebden algorithm. Examples based on ill-posed 1D deconvolution illustrate the extremal regularization concept and the behaviour of the algorithm.

QUADRATICALLY CONSTRAINED QUADRATIC MINIMIZATION

A mathematical statement of the extremal regularization problem is (equivalent to)

$$\min_x (Rx)^T Rx \text{ subject to } (Ax-d)^T(Ax-d) \le \sigma^2 d^T d$$

Here A is the modeling operator, x is the model vector, d is the data, and R is the regularizing operator. The noise level σ is *relative*, as that is usually the most useful way to pose noise estimates. Thus solution of this problem demands quadratically constrained quadratic minimization.

The solution minimizes whatever quality is represented by R, subject to fitting the data to a relative error level σ. The first order necessary conditions of optimality state that the solution satisfies

$$\lambda A^T(Ax-d) + R^T Rx = 0$$

$$\lambda\left[(Ax-d)^T(Ax-d) - \sigma^2 d^T d\right] = 0$$

The first condition states parallelism of the gradients of the constraint and objective functions. The second implies that either the constraint is satisfied as an equality - i.e. the solution is on the boundary of the set of constraint-satisfying models - or else the Lagrange multiplier λ vanishes, which means that the most regular solution actually has a smaller residual than assumed - i.e. σ is larger than the actual noise level.

The first condition is the familiar normal equation of the unconstrained problem

$$\min_x \lambda(Ax-d)^T(Ax-d) + (Rx)^T Rx$$

or

$$\min_x (Ax-d)^T(Ax-d) + \epsilon^2(Rx)^T Rx$$

where $\epsilon = \lambda^{-\frac{1}{2}}$ is the “notoriously elusive” relative weight between model space (really constraint space) and data space.

The point of this paper, and the basis of the Moré-Hebden algorithm, is that the first order conditions make the ϵ a function of the assumed noise level σ. Whenever σ can be estimated directly, this relationship provide a method of estimating ϵ.

ESTIMATING THE REGULARIZATION PARAMETER FROM THE NOISE LEVEL

For arbitrary $\lambda > 0$, denote by $x(\lambda)$ the solution of the normal equations

$$\lambda A^T(Ax(\lambda) - d) + R^T Rx(\lambda) = 0$$

Set

$$\phi(\lambda) = \sqrt{(Ax(\lambda) - d)^T(Ax(\lambda) - d)}$$

The *secular equation* is

$$\phi(\lambda) = \sigma\sqrt{d^T d}$$

and its solution, if it has one, gives the correct value of the Lagrange multiplier λ.

The Moré-Hebden algorithm takes its cue from the simplest possible case: x and d are one-dimensional, and A and R are scalars. In that very special case,

$$x(\lambda) = \frac{\lambda Ad}{\lambda A^2 + R^2}$$

hence

$$\phi(\lambda) = \frac{R^2 d}{\lambda A^2 + R^2}$$

i.e. the reciprocal of ϕ is a linear function of λ. This suggests that Newtons's method is more likely to converge quickly when applied to

$$\psi(\lambda) \equiv \frac{1}{\phi(\lambda)} = \frac{1}{\sigma\sqrt{d^T d}}$$

and that is exactly what the Moré-Hebden algorithm does.

The iteration proceeds as follows:

- initialize λ somehow
- until convergence do: replace λ by

$$\lambda - \frac{\psi(\lambda) - \frac{1}{\sigma\sqrt{d^T d}}}{\psi'(\lambda)}$$

in which ψ' stands for the deriviative of ψ, which you compute like so:

$$\psi'(\lambda) = -\frac{\phi'(\lambda)}{\phi^2(\lambda)}$$

Now

$$\phi'(\lambda) = \phi^{-1}(\lambda)(Ax'(\lambda))^T(Ax(\lambda) - d)$$

From the normal equations,

$$(\lambda A^T A + R^T R)x'(\lambda) = A^T(d - Ax(\lambda))$$

so

$$\phi'(\lambda) = \phi^{-3}(\lambda)(A^T(Ax(\lambda) - d))^T(\lambda A^T A + R^T)^{-1}(A^T(Ax(\lambda) - d))$$

Putting all of this together, one obtains the following algorithm for updating λ:

- solve the normal equations for x, compute the residual ϕ
- compute the normal residual $g = A^T(Ax - d)$
- solve the auxiliary system $(\lambda A^T A + R^T R)s = g$ for s
- compute $\phi' = g^T s/\phi$
- replace λ by
$$\lambda + \frac{\phi}{\phi'}\left(1 - \frac{\phi}{\sigma\sqrt{d^T d}}\right)$$

The first and third steps involve solution of linear systems, which in geophysical applications may be very large. Therefore, in contrast to conventional implementations of this algorithm, I use *conjugate gradient iteration* (Björk, 1997) to compute the solutions of these systems. As one might expect, the error reduction attained by these inner iterations affects the overall convergence rate of the algorithm.

A final detail: since $\lambda = \epsilon^{-2}$ must remain positive, I have replaced any large decrease implied by the above formula by a bisection strategy. Since $\phi' < 0$, as soon as λ is too small (which forces the weight onto the regularization term and increases the residual), the algorithm produces regular increases in λ and converges very rapidly, usually in one or two steps, so long as the normal equations are solved successfully. This is not always the case, but failure to converge rapidly appears to signal large data components associated with very small eigenvalues and is a sure sign that the noise level estimate σ has been chosen too small.

DECONVOLUTION EXAMPLES

The operator A is 1D convolution of a source pulse w with the input time series x. The data d is this convolution plus the noise series n. The regularization operator R is taken to be the identity I for all examples presented here.

All of the examples in this section will concern the source pulse (w) depicted in Figure 1, which is a 15 Hz Ricker wavelet sampled at 4 ms.

The data space consists of time series of length 1001, sample rate 4 ms. The noise free data is the convolution of the wavelet with a spike located at 1 s, see Figure 2.

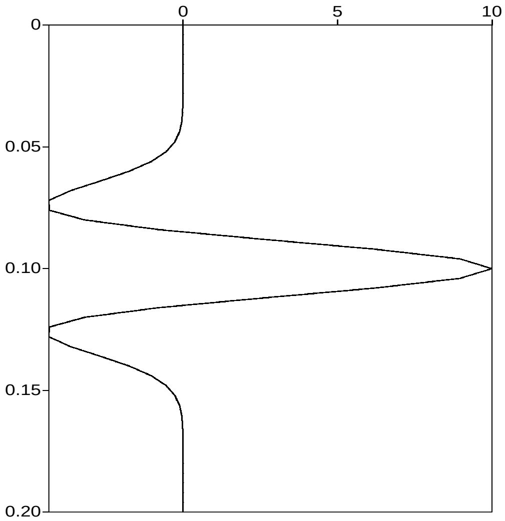

Figure 1: 15 Hz Ricker wavelet used in deconvolution experiments. bill1-fig1 [ER]

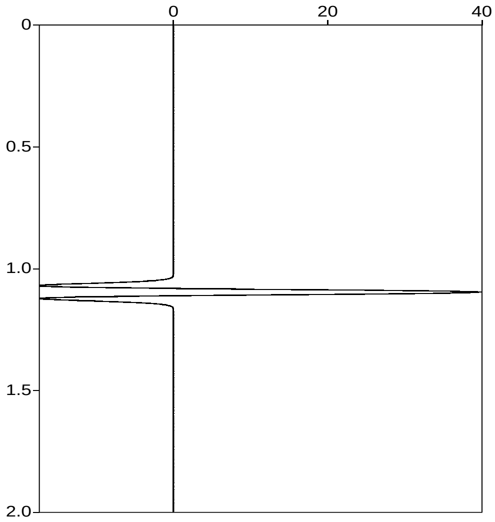

Figure 2: Noise free data. bill1-fig2 [ER]

The noise strength for the first set of experiments is 0.5. The noise is concentrated in the pulse passband, as it is the convolution of a pseudorandom sequence with the pulse, followed by scaling. Thus a signal of reasonable size fits the noisy data (Figure 3) very precisely.

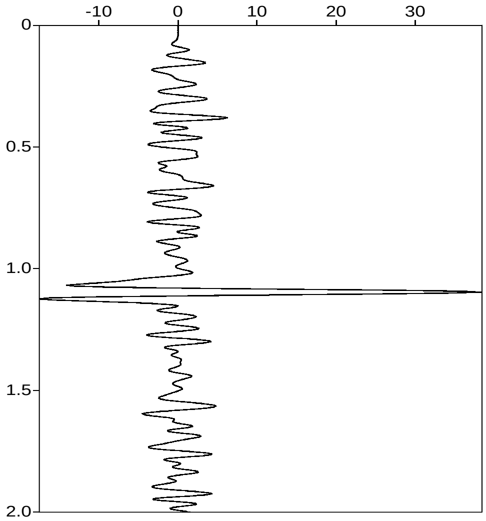

Figure 3: Noisy data: 50% RMS filtered noise. bill1-fig3 [ER]

The deconvolutions (signal estimations) resulting from underestimating, correctly estimating, and overestimating the noise level appear as Figure 4, Figure 5, and Figure 6. The estimated noise levels are 10%, 50%, and 80% respectively. In the notation of the last section, σ =0.1, 0.5, and 0.8 respectively. There is no particular identifiable virtue of one result over the other, which reinforces my contention that in order to solve one of these problems, you must have an independent means of estimating noise level: neither the data nor the results of the signal estimations reveal the signal/noise dichotomy.

Note that even for the correctly estimated noise level, namely $\sigma = 0.5$, you do not recover the isolated spike. The discrepancy is partly due to the less than perfect linear system solves *via* conjugate gradient iteration, but also to the nature of the problem: it is actually possible to achieve the same fit error as that provided by the noise free data with a slightly smaller signal, by fitting the signal less and the noise more. That's because signal and noise are not entirely orthogonal (and they rarely are, so you're going to have to live with this "crosstalk" imperfection!).

The relation between the noise level or fit error and the penalty parameter ϵ really is obscure, as the following results suggest:

- $\sigma = 0.1 \Rightarrow \epsilon = 50.9061$
- $\sigma = 0.5 \Rightarrow \epsilon = 210.593$
- $\sigma = 0.8 \Rightarrow \epsilon = 459.234$

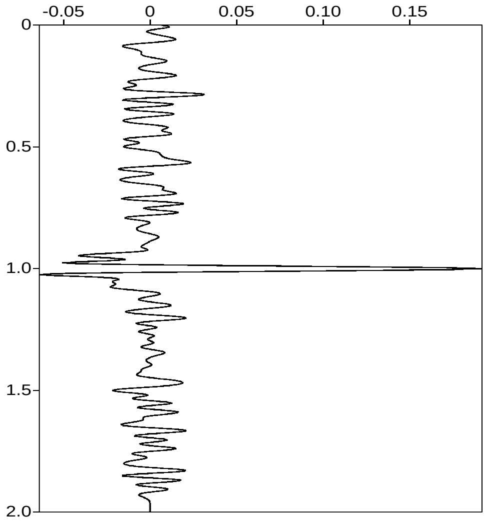

Figure 4: Signal estimate: target noise level 10%, filtered noise. bill1-fig4 [ER]

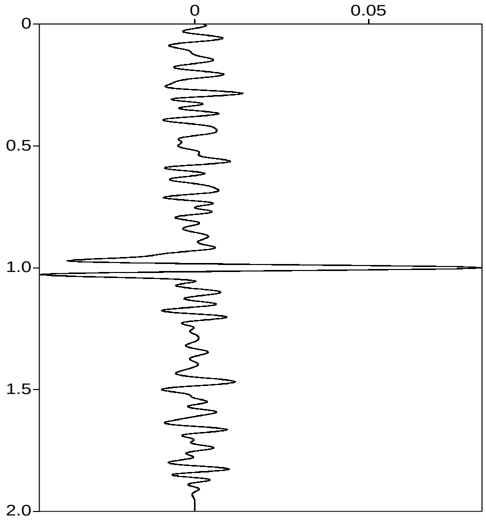

Figure 5: Signal estimate: target noise level 50%, filtered noise. bill1-fig5 [ER]

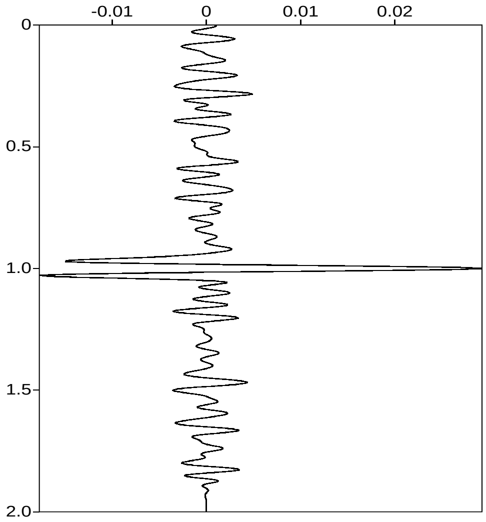

Figure 6: Signal estimate: target noise level 80%, filtered noise. bill1-fig6 [ER]

I would not have guessed the precise values of these ϵs - would you have done? On the other hand the trend is exactly as you would expect: as you permit more misfit, you are able to make the auxiliary quantity (the model L^2 norm in this case) smaller, corresponding to a larger ϵ.

The second set of experiments uses the same noise free data contaminated with unfiltered noise at the 50% level (Figure 7). As the data now contain much out of passband energy, a perfect fit is no longer achievable.

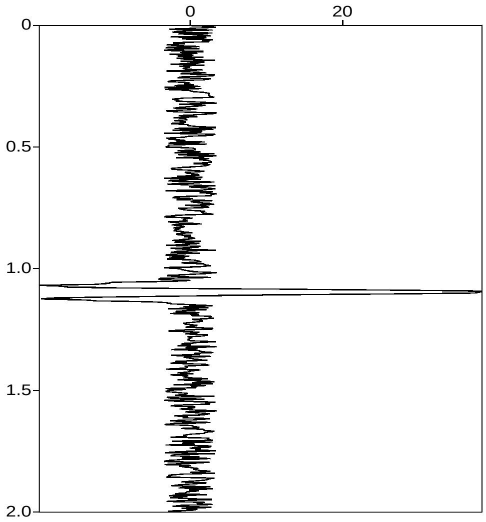

Figure 7: Noisy data: 50% RMS unfiltered noise. bill1-fig7 [ER]

Estimating the noise level at σ =0.1,0.5, and 0.8 as before gives the signals depicted in Figure 8, Figure 9, and Figure 10 respectively. The first of these fit errors is impossible to achieve by means of the conjugate gradient algorithm at least with any reasonable number of iterations. The solution simply grows without bound, as one would expect, and retains almost no character of the target model (Figure 8). The correct estimate $\sigma = 0.5$ on the other hand gives you a reasonable estimate of the signal (Figure 9), with a bandlimited version of the spike dominating the series.

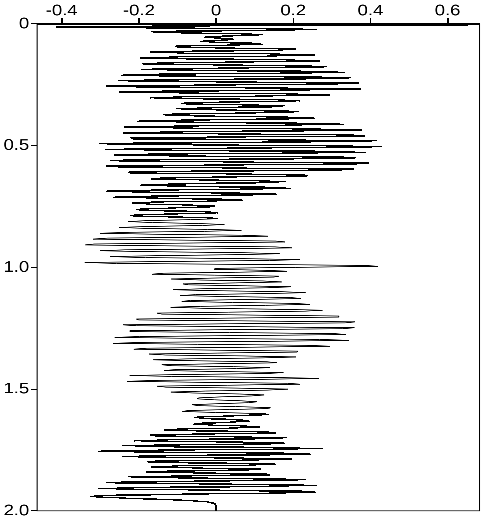

Figure 8: Signal estimate: target noise level 10%, unfiltered noise. bill1-fig8 [ER]

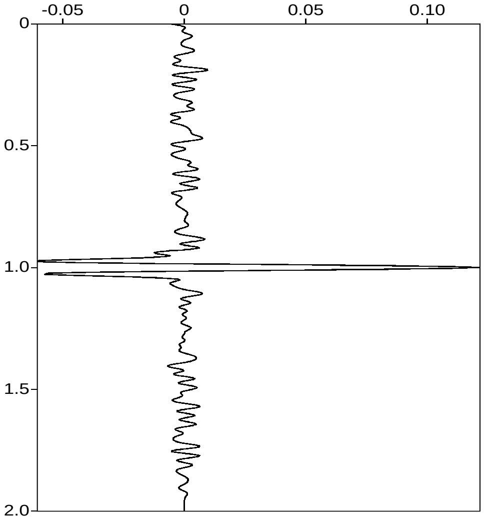

Figure 9: Signal estimate: target noise level 50%, unfiltered noise. bill1-fig9 [ER]

Again, the precise values of ϵ are inscrutable:

- $\sigma = 0.1 \Rightarrow \epsilon = 8.87808\text{e-}11$
- $\sigma = 0.5 \Rightarrow \epsilon = 144.445$

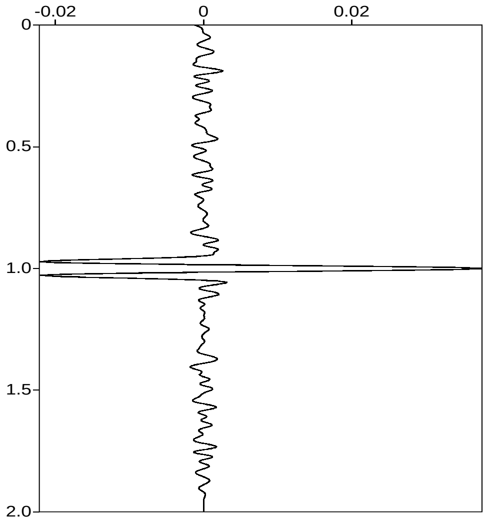

Figure 10: Signal estimate: target noise level 80%, unfiltered noise. bill1-fig10 [ER]

- $\sigma = 0.8 \Rightarrow \epsilon = 405.233$

The trend is even more marked here. The large out-of-band components in the data are essentially impossible to fit. So when you ask for a rather precise fit - 10% error - the weight on the model space decreases throughout the iteration, apparently with no end in sight. The value in the table above was the result of 10 Moré-Hebden iterations, and ϵ diminished by an order of magnitude or so each iteration. As soon as the level of fit permits you to discard the out of band components (that's what happened at $\sigma = 0.5$), the desired fit actually occurs and a reasonable value of ϵ results.

Clearly, prior knowledge of a reasonable model size would enable you to guess σ in this example. However then you have to know the size of the model! This may be no more obvious than the size of the noise. This observation reinforces my contention that solution of problems like these demands that you know *something* in addition to the data samples - there is no "born yesterday" bootstrapping into a signal - noise distinction.

CONCLUSION

Extremal regularization appears to be practical for large scale problems, as the Moré-Hebden algorithm with conjugate gradient inner solves either converges in a reasonable number of steps or doesn't converge when the constraint (target noise level) forces too many small singular values into the act. All of these terms are relative - small, doesn't converge, etc. Modulo floating point arithmetic, the algorithm will *always* work if enough effort is expended. The issue of course is reasonable level of effort, and that is in some sense a translation of the concept of "noise level" - it's the misfit between the data and what you can achieve with an easily computable model, no more.

Thus extremal regularization as implemented in this report appears to give a reasonable approach to relative weighting in model and data space when an independent estimate of noise level is somehow available. This is the case for example in the examples mentioned in the introduction. Maybe quiet parts of seismic traces furnish pure noise series which might give a usable estimate of noise level - provided that the modeling operator is sophisticated enough to fit the rest!

ACKNOWLEDGEMENT

The author performed part of the work reported here while a guest of the Stanford Exploration Project. The author thanks the Director of SEP, Prof. Jon Claerbout, and the other project members for their hospitality and for the stimulating intellectual atmosphere of SEP.

This work was partially supported by the National Science Foundation under grant numbers DMS 9404283 and DMS 9627355; the Office of Naval Research under grant number N00014-96-1-0156; the U. S. Department of Energy under grant number DE-FG07-97 ER14827; and The Rice Inversion Project. TRIP Sponsors for 1998 are Amerada Hess, Amoco Production Co., ARCO Oil and Gas, Conoco Inc., Discovery Bay, Exxon Production Research Co., Mobil Research and Development Corp., and Shell International Research.

REFERENCES

Björk, A., 1997, Numerical methods for least squares problems: Society for Industrial and Applied Mathematics, Philadelphia.

Claerbout, J. F., 1992, Earth sounding analysis: Processing versus inversion: Blackwell, Boston.

Claerbout, J., S/n segregation: The best ratio between data fitting and model regularization: Gain control:, Technical report, Stanford Exploration Project, 1998.

DennisJr., J., and Schnabel, R., 1983, Numerical methods for unconstrained optimization and nonlinear equations: Prentice-Hall, Englewood Cliffs.

Gockenbach, M., and Symes, W. W., 1997, Duality for inverse problems in wave propagation *in* Biegler, L., Coleman, T., Santosa, F., and Conn, A., Eds., Large Scale Optimization:: Springer Verlag.

Hebden, M. D., An algorithm for minimization using exact second derivatives:, Technical Report TP515, A.E.R.E., Harwell, 1973.

Jackson, D. D., 1973, Marginal solutions to quasi-linear inverse problems in geophysics: the edgehog method: Geophysical Journal of the Royal Astronomical Society, **35**, 121–136.

Jackson, D. D., 1976, Most squares inversion: J. Geophys. Research, **81**, 1027–1030.

Jackson, D. D., 1979, The use of *a priori* data to resolve nonuniqueness in linear inversion: Geophysical Journal of the Royal Astronomical Society, **57**, 137–157.

Moré, J. J., 1977, The levenberg-marquardt algorithm: implementation and theory *in* Watson, G. A., Ed., Numerical Analysis:: Springer-Verlag, 630 ff.

APPENDIX: WORKING WITH THE EXAMPLES IN THIS REPORT

This report, together with its associated files, constitutes a reproducible research document. Makefiles tie together the various components - text, code source, data, and postscript figures. The principal make rules are the SEP standards: build, view, clean, burn. In this section, I will assume familiarity with the Stanford Exploration Project reproducible research concept, which guides the structure of this document. This code behind this document is an application of the Hilbert Class Library. So the first thing you need to do is to make HCL available. If HCL is already installed on your system, you do this by adding a line to your `.cshrc` file or appropriate component file. Otherwise you must install HCL first. The easiest way to do this is to download it from the TRIP web page:

```
http://www.trip.caam.rice.edu
```

and follow the installation instructions. The code also depends on the SU/SEGY vector class package `sVector`, which therefore must also be installed. It will be part of the next HCL release. **NB:** At Rice/TRIP and Stanford/SEP, no installation is necessary: the packages are already installed. Simply add the following lines to your shell environment files:

- at TRIP, add to your `.cshrc`:

  ```
  setenv HCLROOT /import/masc39c/symes/hclr0.9
  setenv KBDAROOT /import/masc39c/symes/kbda
  setenv QPROOT <path to the root directory of this package>
  ```

- at SEP, add to your `Setup/cshrc.generic`:

  ```
  setenv HCLROOT /jon/symes/hclr0.9
  setenv KBDAROOT /jon/symes/kbda
  setenv QPROOT <path to the root directory of this package>
  ```

The "root directory of this package" referenced in these instructions is the directory you create by downloading the `tar` file containing this report. In so doing, you create a directory tree with root named `qcqm`. This is the "root" in question. All pathnames in the following discussion are relative to this root. HCL includes a set of make rules which evolved from the SEP rule set as it was about two years ago. I am sure that SEP's rules have also evolved, and differently. The makefiles for this report are all output of the HCL makefile autowriter, `maw`, and use HCL's rules. If you get as far as modifying makefiles by hand, bear the possible HCL/SEP incompatibility in mind. You can rebuild the entire package simply by entering `make build` in the root directory. You will construct all of the executables and final results (`.ps` files in the `Fig`) directories, including this postscript version of this report. You can make individual results in the usual way. Note that all figures in this report except the first two will vary from build to build, as you choose a new pseudorandom seed each time you execute the commands. The command for deconvolution is `sfilter/decon.x`. You execute it by following it with a parameter file name: `decon.x par` - it reads all of its parameters from the file. Regrettably the "getpar" device used in curren HCL programs is not flexible enough to permit specification of parameters on the command line. Probably I should just steal SEP's getpar! You can use the executable `sfilter/decon.x` with other data by altering its input parameters, and so explore the capabilities of the algorithm using data other than that supplied with this report. Parameter control requires you to edit the parameter files manually. HCL parameter files are `keyword=value` lists; the values can be integers, floating point numbers with any size of mantissa, and strings. Parameters to be read only by one part of the program (typically a class constructor) get an identifying string prepended, with a double colon. Thus the parameter `Tol` for the conjugate gradient algorithm becomes `CG::Tol`. That is, the parameter file can specify many variables named `Tol`, so ong as they have been equipped with qualifiers which allow each program unit to choose a unique value. Files should be in either SU (stripped SEGY) or SEGY formats. `sVector` currently supports only native binary floating point representation, so if you port data from Linux to SGI etc. you will have to byte-swap it. Here is the parameter file structure for the deconvolution example:

```
Sigma=0.5
Lambda=0.0001
Wavelet="rick15.su"
DataTimeSeries="fnd.su"
CG::Tol=1.e-4
CG::MaxItn=50
CG::DispFlag=3
QCQM::Tol=0.01
QCQM::MaxItn=10
QCQM::DispFlag=1
```

The parameters are

- `Sigma`: target noise level
- `Lambda`: initial estimate of λ. It may be worth thinking about sensible defaults or crude estimates for this, and the scalar model actually suggests such an estimate. For the moment, set by hand.
- `Wavelet`: name of wavelet data file,in either SU or SEGY format.
- `DataTimeSeries`: filename for data time series

- `CG::Tol`: convergence tolerance for HCL conjugate code - iteration terminates if *normal* residual, i.e. in this case $\lambda A^T(Ax - d) + R^T Rx$
- `CG::MaxItn`: maximum conjugate gradient iterations
- `CG::DispFlag`: controls verbosity of CG output, as explained in HCL reference manual (through TRIP web page: point your browser at

  `http://www.trip.caam.rice.edu`

  and follow the links to the HCL reference manual, page on the conjugate gradient algorithms). Level 3 is max output, including a summary of the progress at each iteration. Note that the residual reported is the *normal* residual for this application!
- `QCQM::Tol`: the Moré-Hebden algorithm considers itself converged when the relative error in the constraint is within this amount
- `QCQM::MaxItn`: maximum Moré-Hebden iterations
- `QCQM::DispFlag`: controls verbosity of Moré-Hebden code: 0 = silent, any other value prints diagnostic information about run

Amongst the diagnostics printed out at the end of a run when `QCQM::DispFlag` is set you will find “Lagrange cosine”. This is the cosine of the angle between the constraint gradient and the objective gradient. It diagnoses the success of the constrained optimization: if it is very close to 1, then the two gradients are parallel and the first order necessary condition has been satisfied. This occurs in the deconvolution examples in all cases except that depicted in Figure 8, in which nothing converges and you can’t fit the data. Apparently failure to get the Lagrange cosine close to 1 in a reasonable number of CG and Moré-Hebden iterations implies that the noise level has been set too small and you are trying to match data components associated with very small singular values.