Recursive Algorithms for Computing the Cramer-Rao Bound

Mohammad Usman*, Alfred O. Hero*, and Jeffrey A. Fessler**
* Dept. of Electrical Engineering and Computer Science, and ** Dept. of Internal Medicine,
The University of Michigan, Ann Arbor, MI 48109-2122

Abstract

Computation of the Cramer-Rao bound involves inversion of the Fisher information matrix (FIM). The inversion can become computationally intractable when the number of unknown parameters is large. Hero and Fessler [1, 5] have presented a recursive and monotonically convergent algorithm to invert sub-matrices of the FIM corresponding to a few parameters of interest. The convergence rate of this algorithm depends on the choice of a splitting matrix which in [1, 5] was selected based on statistical considerations as a complete-data FIM. In this paper we present methods for the acceleration of the algorithm using several different choices of splitting matrices, over-relaxation, and conjugate gradient. The latter algorithms achieve faster convergence rate at the expense of monotone convergence. For ill-posed problems the FIM is singular and a Cramer-Rao bound is given by the pseudo-inverse of FIM. We extend the recursive algorithms in this paper to efficient computation of the pseudo-inverse. We apply the methods developed in this paper to image reconstruction in emission tomography and to signal deconvolution.

Key Words: Matrix splitting methods, conjugate gradient, pseudo-inverse.

EDICS Number: SP 3.8

Corresponding author:

Professor Alfred O. Hero III

Room 4229, Dept. of EE and CS

University of Michigan

1301 Beal Avenue

Ann Arbor, MI 48109-2122.

Tel: (313) 763-0564.

e-mail: hero@eecs.umich.edu.

[1]This work was supported in part by National Science Foundation under grant BCS-9024370, a Government of Pakistan Postgraduate Fellowship, NIH grant CA-60711, and DOE grant DE-FG02-87ER60561

1 Introduction

The Cramer-Rao (CR) bound determines a lower bound on the covariance of any estimator with a given bias gradient. For an n-dimensional parameter vector, and when the FIM F_Y is non-singular, calculation of the CR bound involves inversion of F_Y. Direct matrix inversion, requiring $\mathcal{O}(n^3)$ flops (floating point operations), is generally computationally intractable if the number of unknown parameters is large. For example, in image reconstruction where the pixel intensities are the parameters to be estimated, a moderate size image of 128×128 pixels has a FIM of dimension $(128)^2 \times (128)^2$ and will require $\mathcal{O}(10^{12})$ flops to compute its inverse. On a 100 MIPS workstation this would require over 12 hours of computation time for each object distribution to be considered.

Often only a few components of the n-dimensional estimator are of interest. For example in image reconstruction one may be primarily interested in a small image region of interest (ROI) corresponding to a tumor or a lesion. The method of sequential partitioning can be used in this case to calculate the bound on $\text{cov}_{\underline{\theta}}(\underline{\widehat{\theta}}^{ROI})$, where $\underline{\widehat{\theta}}^{ROI}$ is an estimator of q parameters $\underline{\theta}^{ROI}$ of interest [8]. This still requires $\mathcal{O}(n^3)$ flops. Hero et al. [1, 5] present a recursive method, based on a linearly convergent geometric series decomposition of the inverse, for calculating a small $q \times q$ sub-matrix of the matrix ${F_Y}^{-1}$ required for computing the CR bound on $\text{cov}_{\underline{\theta}}(\underline{\widehat{\theta}}^{ROI})$. The geometric series algorithm requires only $\mathcal{O}(n^2)$ flops per iteration so that if the convergence rate is fast, a significant computational saving is achieved. Furthermore, if the FIM is of the form $F_Y = A^T \Gamma A$, where Γ is an $m \times m$ diagonal, A is $m \times n$, and $n \ll m$ or when A is sparse, the geometric series algorithm has much reduced storage requirement as compared to the direct inversion methods. As presented in [1], the geometric series algorithm stipulates that we find a $n \times n$ matrix F, called the splitting matrix, such that F^{-1} is easy to compute ($\mathcal{O}(n^2)$ or less) and such that F dominates F_Y in the sense that $F - F_Y \geq 0$. In [1, 5] a method for selecting an appropriate matrix F was introduced based on purely statistical arguments; F is chosen as the FIM of any more informative data set such as a complete-data set used in the EM algorithm [3]. The main advantage of the algorithm of [1, 5] is its monotone convergence which generates a valid and improving lower bound on estimator covariance at each iteration of the algorithm. However, the price paid for monotone convergence is slow linear convergence rate, and its high sensitivity to the condition number of F_Y. For applications where a strict lower bound is not required at each iteration, a non-monotone algorithm may be acceptable. While the statistical approach of [1, 5] provides powerful guiding principle for finding splitting matrices which guarantee monotonic convergence, the approach requires finding a suitable reformulation of the estimation problem. Furthermore, unless the complete-incomplete data space is defined cleverly, the approach of [1, 5] does not guarantee rapid rate of convergence.

In this paper we present methods for acceleration of the recursive algorithm of [1, 5] which are based on purely algebraic considerations. First an expression for an F_Y-dominating splitting matrix is given which optimizes the average convergence rates for regularly circulant FIM and gives a monotonically convergent series. When applied to a non-circulant FIM, our numerical results indicate that this splitting matrix provides very close to optimal average convergence rates. Second, we present an over-relaxed version of the geometric series which can greatly accelerate the convergence rate at the expense of non-monotonic convergence. Third, we present a conjugate gradient type of algorithm tailored for the inversion of positive-definite, symmetric FIM, which has a significantly faster convergence rate and is less sensitive to the condition number of F_Y.

In many signal processing application, e.g. signal deconvolution, the parametric estimation problem is ill-posed which results in a singular FIM. The covariance of the estimates in this case are lower bounded by a function of the Moore-Penrose pseudo-inverse[2] of the FIM [2]. Direct calculation of the pseudo-inverse usually requires a singular value decomposition (SVD) of the FIM, which is an $\mathcal{O}(n^3)$ operation. We present two recursive algorithms to approximate columns of the Moore-Penrose pseudo-inverse of the FIM with $\mathcal{O}(n^2)$ flops per iteration. These algorithms generalize our recursive algorithms to the case of singular FIM. The first algorithm, a perturbation method, converges monotonically to the pseudo-inverse of F_Y when implemented using the geometric series approach. The other algorithm, a differencing method, is non-monotonic for both the geometric series and the conjugate gradient implementations.

We apply the methods developed in this paper to: 1) Single Photon Emission Computed Tomography (SPECT) image reconstruction, and 2) one dimensional discrete deconvolution. In both these applications the FIM F_Y is of the factored form permitting a significant reduction in memory requirements.

2 Unbiased Cramer-Rao Bound

We assume a measurement Y that is a random variable with probability density $f_Y(Y;\underline{\theta})$ dependent on an unknown parameter vector $\underline{\theta} = [\theta_1, ..., \theta_n]^T \in \mathbb{R}^n$. A vector function $\underline{\widehat{\theta}} = \underline{\widehat{\theta}}(y) \in \mathbb{R}^n$ is a parameter estimator based on the observation $Y = y$. The performance of this estimator is measured by its bias:

$$\underline{b}(\underline{\theta}) = E_{\underline{\theta}}(\underline{\widehat{\theta}}) - \underline{\theta},$$

and by its covariance:

$$\text{cov}_{\underline{\theta}}(\underline{\widehat{\theta}}) = E_{\underline{\theta}}\left[(\underline{\widehat{\theta}} - E_{\underline{\theta}}(\underline{\widehat{\theta}}))(\underline{\widehat{\theta}} - E_{\underline{\theta}}(\underline{\widehat{\theta}}))^T\right],$$

[2]In this paper we will use Moore-Penrose pseudo-inverse and pseudo-inverse interchangeably.

where $E_{\underline{\theta}}(Z) = \int z f_Z(z;\underline{\theta})dz$ denotes statistical expectation of a random variable Z. Under suitable regularity conditions [15] the FIM F_Y is given by

$$F_Y(\underline{\theta}) = E_{\underline{\theta}}[\nabla^T_{\underline{\theta}} \ln f_Y(Y;\underline{\theta})\, \nabla_{\underline{\theta}} \ln f_Y(Y;\underline{\theta})],$$

where $\nabla_{\underline{\theta}}$ denotes the (row) gradient vector $[\frac{\partial}{\partial\theta_1}, \frac{\partial}{\partial\theta_2}, ..., \frac{\partial}{\partial\theta_n}]$. When $F_Y = F_Y(\underline{\theta})$ is non-singular the CR lower bound on the covariance of any parameter estimator is specified by the inverse of FIM and the gradient of the bias [2]:

$$\text{cov}_{\underline{\theta}}(\widehat{\underline{\theta}}) \geq [I + \nabla_{\underline{\theta}}\underline{b}(\underline{\theta})]\, F_Y^{-1}(\underline{\theta})\, [I + \nabla_{\underline{\theta}}\underline{b}(\underline{\theta})]^T .$$

For unbiased estimation ($E_{\underline{\theta}}(\widehat{\underline{\theta}}) = \underline{\theta}$), $\nabla_{\underline{\theta}}\underline{b}(\underline{\theta}) = 0$ and we have the so-called unbiased CR bound,

$$\text{cov}_{\underline{\theta}}(\widehat{\underline{\theta}}) \geq F_Y^{-1}(\underline{\theta}). \tag{1}$$

In the case that the FIM is singular the CR bound is [2]:

$$\text{cov}_{\underline{\theta}}(\widehat{\underline{\theta}}) \geq [I + \nabla_{\underline{\theta}}\underline{b}(\underline{\theta})]\, F_Y^{+}(\underline{\theta})\, [I + \nabla_{\underline{\theta}}\underline{b}(\underline{\theta})]^T ,$$

where $F_Y^{+}(\underline{\theta})$ denotes the Moore-Penrose pseudo-inverse of FIM $F_Y(\underline{\theta})$.

In the sequel we will assume unbiased estimation so that $\nabla_{\underline{\theta}}\underline{b}(\underline{\theta}) = 0$. See [6] for a treatment of the biased estimation problem. Since (1) is a matrix inequality, $\text{cov}_{\underline{\theta}}(\widehat{\underline{\theta}}) - F_Y^{-1}(\underline{\theta})$ is positive semi-definite (PSD). In particular the lower bound on the covariance of $\widehat{\theta}_1$, corresponding to the top left (1,1) entry of $\text{cov}_{\underline{\theta}}(\widehat{\underline{\theta}})$, is given by the corresponding entry of $F_Y^{-1}(\underline{\theta})$, denoted $[F_Y^{-1}]_{11}$:

$$\text{var}_{\underline{\theta}}(\widehat{\theta}_1) \geq \underline{e}_1^T F_Y^{-1}(\underline{\theta})\underline{e}_1, \tag{2}$$

where $\underline{e}_1 = [1, 0, ..., 0]^T$. Throughout this paper we will be interested in calculating the (1,1) element of F_Y^{-1} given by the right hand side of (2), or more generally the first column of F_Y^{-1}. The results to be presented are easily extended to q-elements: $q \leq n$.

3 Recursive CR Bound Algorithms

3.1 Monotone Convergent Algorithm: Geometric Series

For more details and proof of convergence of this method see [1]. Assume that F_Y is non-singular and that there exists an invertible $n \times n$ matrix F such that F dominates F_Y: $F \geq F_Y > 0$. It can be easily shown that all the eigenvalues of $I - F^{-1}F_Y$ lie in the interval $[0, 1)$. We can therefore apply the matrix form of the geometric series to F_Y^{-1}, also known as the 'Neumann expansion' [23]:

$$F_Y^{-1} = [F - (F - F_Y)]^{-1}$$

$$
\begin{aligned}
&= \left[I - (I - F^{-1}F_Y)\right]^{-1} F^{-1} \\
&= \left[\sum_{k=0}^{\infty}(I - F^{-1}F_Y)^k\right] F^{-1}.
\end{aligned}
$$

An approximation $B^{(m)}$ to F_Y^{-1} can be obtained by truncating the infinite series:

$$B^{(m)} = \left[\sum_{k=0}^{m}(I - F^{-1}F_Y)^k\right] F^{-1}.$$

Using induction on k it is straightforward to establish the following recursive relation:

$$B^{(k+1)} = [I - F^{-1}F_Y]\, B^{(k)} - F^{-1}. \tag{3}$$

The above expression converges monotonically to the CR bound in the sense that $B^{(k)} - B^{(k+1)} = [I - F^{-1}F_Y]^{k+1} F^{-1}$ is positive semi-definite for all k and $F_Y^{-1} - B^{(k)}$ converges to a matrix of zeros as $k \to \infty$. Note that the recursion (3) updates each column of $B^{(k)}$ independently.

Let $\beta^{(k)}$ be the first column of the matrix $B^{(k)}$. Then a recursive algorithm for computing the CR bound $[F_Y^{-1}(\underline{\theta})]_{11}$ on $\mathrm{cov}_{\underline{\theta}}(\widehat{\theta}_1)$, which we call the geometric series (GS) algorithm, is given as follows:

GS Algorithm

INITIALIZE: $\underline{\beta}^{(0)} = \underline{0}$ an $n \times 1$ vector of zeroes.

RECURSION:

$$\underline{\beta}^{(k+1)} = \underline{\beta}^{(k)} - F^{-1}(\ F_Y\, \underline{\beta}^{(k)} + \underline{e}_1\ ). \tag{4}$$

The recursion yields a sequence of vectors $\underline{\beta}^{(k)}$ which converges to the first column of F_Y^{-1}. The first element of $\beta^{(k)}$ is simply $\zeta^{(k)} = \underline{e}_1^T\ \underline{\beta}^{(k)} = \underline{e}_1^T B^{(k)} \underline{e}_1$. $\zeta^{(k)}$ monotonically converges to $[F_Y^{-1}(\underline{\theta})]_{11}$ as $k \to \infty$. This monotone convergence property guarantees that the scalar $\zeta^{(k)}$ will be a valid lower bound for all $k > 0$ and that this bound improves at each iteration [1].

The matrix recursion (3) is a special case of the 'matrix splitting method' [14, 19] for iteratively approximating a matrix inverse. When F is taken as the scaled identity γI, where γ is the Frobenius norm of F_Y, (3) reduces to the Frobenius normalization method for inversion of sparse matrices. When F is taken as the diagonal component of F_Y then (3) reduces to the Jacobi method, which however is generally non-monotone convergent as $F - F_Y$ may not be positive definite.

Since recursion (4) is a vector update, it requires only $\mathcal{O}(n^2)$ flops per iteration. Therefore if the algorithm is stopped for $k \ll n$, we obtain a bound on $\mathrm{cov}_{\underline{\theta}}(\widehat{\theta}_1)$ with significantly fewer flops than required for direct calculation of the (1,1) element of F_Y^{-1}. For this algorithm (4) to have computational advantage over direct inversion and also deliver a tight bound we require F^{-1} to be easy to compute and convergence to be fast.

In [1, 5] Hero et. al. exploited the properties of Fisher information matrices to specify an appropriate splitting matrix F. In this estimation theoretic approach one identifies a complete-incomplete data setting for $f_Y(y;\underline{\theta})$. Assume that we can find a hypothetical set of measurements $X \sim f_X(x;\underline{\theta})$ which is larger and more informative than Y in the sense that the conditional density $f_{Y/X}(y/x;\underline{\theta})$ is independent of $\underline{\theta}$. Let F_X and F_Y be the FIM for the 'complete data' X and 'incomplete data' Y, respectively, the data processing theorem asserts that $F_X - F_Y \geq 0$ [1]. This implies that F_X can be used as a splitting matrix. If X is chosen intelligently such that F_X is easily invertible, then GS can be implemented efficiently. In contrast to this estimation theoretic approach, in this paper we will utilize purely algebraic approaches to specifying a splitting matrix F in order to improve the convergence rate of the GS. In section 7 we will compare the algebraic approach presented in this paper to the estimation theoretic approach of [1, 5].

In the next section we show that the average rate of convergence of the GS algorithm increases as the distance between F and F_Y decreases. The objective will then be to select a simple sparse matrix F which satisfies $F - F_Y \geq 0$ and makes the distance small. It can be easily shown that the splitting matrix F can be viewed as a pre-conditioning matrix for the gradient search algorithm.

4 Convergence Rates for GS Algorithms

Two types of convergence criteria are commonly used in evaluating direct iterative methods: the average rate of convergence and the asymptotic rate of convergence. These are discussed below.

Defining $\Delta\beta^{(k)} = F_Y^{-1}\underline{e}_1 - \beta^{(k)}$, from (4) it is easily shown that

$$\Delta\beta^{(k+1)} = M^k \Delta\beta^{(0)},$$

where $M = I - F^{-1}F_Y$ and $\Delta\beta^{(0)} = F_Y^{-1}\underline{e}_1$. The Euclidean norm $\|\Delta\beta^{(k+1)}\|_2$ of the vector $\Delta\beta^{(k+1)}$ can be related to the matrix-2 and Frobenius norms $|||M|||_2$ and $|||M|||_{\mathcal{F}}$ of the matrix M and the Euclidean norm of $\Delta\underline{\beta}^{(0)}$ by [13]:

$$\begin{aligned} \|\Delta\beta^{(k+1)}\|_2 &\leq |||M^k|||_2\|\Delta\beta^{(0)}\|_2 \\ &\leq |||M^k|||_{\mathcal{F}}\|\Delta\beta^{(0)}\|_2, \end{aligned} \tag{5}$$

where for any real vector $\underline{z}$, $\|\underline{z}\|_2$ is the Euclidean norm:

$$\|\underline{z}\|_2 = \sqrt{\sum_i |z_i|^2},$$

and for any real matrix $A = ((a_{ij}))$, $|||A|||_2$ is the matrix-2 norm and:

$$|||A|||_{\mathcal{F}} = \sqrt{\text{trace}(A^T A)} = \sqrt{\sum_i \sum_j |a_{ij}|^2}.$$

From (5), $|||M^k|||_{\mathcal{F}}$ bounds the relative improvement in the approximation error $\Delta\beta^{(k+1)}$ as compared to the initial approximation error $\Delta\beta^{(0)}$: $\frac{\|\Delta_{\beta^{(k+1)}}\|_2}{\|\Delta_{\beta^{(0)}}\|_2} \leq |||M^k|||_{\mathcal{F}}$. In this context the Frobenius norm is preferable to the matrix-2 norm in applications since computation of $|||M|||_{\mathcal{F}}$ requires no expensive singular value decomposition of M.

We specify an absolute convergence criterion in terms of a threshold $\gamma \in [0, 1]$.

Definition *The GS algorithm is said to have converged after N iterations (to within a factor γ) if*

$$\frac{\|\Delta\beta^{(k+1)}\|_2}{\|\Delta\beta^{(0)}\|_2} < \gamma, \ \ \forall k > N. \tag{6}$$

The minimum integer $N = N(\gamma)$ such that this inequality is satisfied is called 'the required number of iterations' for convergence.

Note that unless $\Delta\beta^{(0)} = 0$, $N(\gamma)$ increases as γ decreases to zero. Relation (5) and the convergence criterion (6) provide the following recipe for specifying an upper bound on the required number of iterations for convergence: *An upper bound on $N(\gamma)$ is any N such that*

$$|||M^k|||_{\mathcal{F}} < \gamma, \ \ \forall k > N. \tag{7}$$

For specified γ the *average rate of convergence* is defined as [22]:

$$R_{ave}(\gamma) = \left(|||M^{N(\gamma)}|||_{\mathcal{F}}\right)^{\frac{1}{N(\gamma)}}. \tag{8}$$

In terms of $R_{ave}(\gamma)$ an upper bound N or $N(\gamma)$ is given by (7) :

$$N = \frac{\log\left(\frac{1}{\gamma}\right)}{\log\left(\frac{1}{R_{ave}(\gamma)}\right)}. \tag{9}$$

Unfortunately $R_{ave}(\gamma)$ is typically difficult to compute and (9) is not implementable. On the other hand, we can obtain a simpler upper bound on $N(\gamma)$ by using the matrix norm inequality $|||M^N|||_{\mathcal{F}} \leq |||M|||_{\mathcal{F}}{}^N$. Using this inequality in (8) and (9) we obtain the following upper bound N on $N(\gamma)$:

$$N(\gamma) \leq N = \frac{\log\left(\frac{1}{\gamma}\right)}{\log\left(\frac{1}{|||M|||_{\mathcal{F}}}\right)}. \tag{10}$$

Therefore, reducing the Frobenius norm $|||M|||_{\mathcal{F}}$ reduces the upper bound on the required number of iterations $N(\gamma)$. If the GS algorithm is stopped after N iterations, where N is given by (10), then convergence to within a factor γ is guaranteed.

Another measure of convergence rate is the asymptotic rate of convergence equal to the root convergence factor $\lim_{N\to\infty} \left(|||M^N|||_{\mathcal{F}}\right)^{\frac{1}{N}}$ [13]. The root convergence factor is associated with the 'worst case' rate of convergence of the matrix splitting methods [14, 22] and is equal to the maximum singular value, i.e. the maximum magnitude eigenvalue $|\lambda^M|_{\max}$ of M, which is called the spectral radius $\rho(M)$.

Since $\rho(M) = \lim_{N\to\infty} R_{ave}(\gamma)$, in view of relation (9), the spectral radius is sometimes used to approximate the required number of iterations $N(\gamma)$:

$$N(\gamma) \approx \frac{\log\left(\frac{1}{\gamma}\right)}{\log\left(\frac{1}{\rho(M)}\right)}. \tag{11}$$

However, $\rho(M)$ can be significantly less than $R_{ave}(\gamma)$, and therefore, unless γ is very small, use of the expression (11) can result in severely underestimating the required number of iterations. Therefore, as seen from the bound (10), and as pointed out in [22], while the spectral radius is a good predictor of the asymptotic convergence rate of the GS algorithm, the Frobenius norm $|||M|||_{\mathcal{F}}$ is more reliable as a predictor of the actual number of iterations required for convergence. This will be quantitatively illustrated in Section 7.

4.1 Design of Splitting Matrix

To ensure monotone convergence and optimize the convergence rates, two worthwhile objectives would be to select the splitting matrix F such that F is simple to invert, $F - F_Y$ is non-negative definite and such that either: i) $\rho(M)$ is small; or ii) $|||M|||_{\mathcal{F}}$ is small, where as before $M = I - F^{-1}F_Y$. For large dimension n, computation of $\rho(M)$ is intractable since it requires performing an SVD or power iterations on M to find the maximum eigenvalue [13]. Optimization of the Frobenius norm $|||M|||_{\mathcal{F}}$ is easier, but we consider an even simpler but related objective: find a symmetric p-diagonal $n \times n$ matrix F_p which is simply inverted, satisfies $F_p - F_Y \geq 0$, and makes $|||F - F_Y|||_{\mathcal{F}}$ small. A p-diagonal matrix F_p is defined as any banded matrix of the form

$$F_p = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1p} & & \cdots & 0 \\ d_{21} & d_{22} & & & \ddots & & \vdots \\ \vdots & & \ddots & & & \ddots & \\ d_{p1} & & & \ddots & & & d_{n-p,n} \\ & \ddots & & & \ddots & & \vdots \\ \vdots & & \ddots & & & \ddots & d_{n-1,n} \\ 0 & \cdots & & d_{n,n-p} & \cdots & d_{n,n-1} & d_{nn} \end{bmatrix}. \tag{12}$$

Making $|||F - F_Y|||_{\mathcal{F}}$ small is a reasonable objective since the following inequalities relate the minimization of $|||F - F_Y|||_{\mathcal{F}}$ to minimization of $\rho(M) = \lim_{N\to\infty} |||(I - F^{-1}F_Y)^N|||^{\frac{1}{N}}$ and of $|||M|||_{\mathcal{F}}$, where $|||.|||$ is any matrix norm.

$$\begin{aligned} |||(I - F^{-1}F_Y)^N|||^{\frac{1}{N}} &\leq |||(I - F^{-1}F_Y)^N||| \\ &\leq |||F^{-1}|||\,|||F - F_Y||| \end{aligned}$$

$$
\begin{aligned}
&\leq |||F_Y^{-1}|||\;|||F - F_Y||| \\
&\leq |||F_Y^{-1}|||_{\mathcal{F}}\;|||F - F_Y|||_{\mathcal{F}}\, C,
\end{aligned}
$$

for some constant C depending upon the $|||.|||$. For the spectral norm $|||.||| = |||.|||_2$ the constant C is equal to 1. The first inequality is well known [13], the second is the Cauchy-Schwartz inequality for matrix norms, and the third inequality follows from the restriction $F - F_Y \geq 0$.

Consider the following p-diagonal matrix, D_p, constructed from the elements a_{ij} of the matrix A:

$$
D_p = \begin{bmatrix}
\sum\limits_{j\neq 2,\ldots,p} |a_{1j}| & a_{12} & \cdots & a_{1p} & & \cdots & 0 \\
a_{21} & \sum\limits_{j\neq 1,3,\ldots,p+1} |a_{2j}| & & & \ddots & & \vdots \\
\vdots & & \ddots & & & \ddots & \\
 & & & \sum\limits_{\substack{j\neq k-p,\ldots,k-1 \\ j\neq k+1,\ldots,k+p-1}} |a_{kj}| & & \ddots & \\
a_{p1} & & & & \ddots & & a_{n-p,n} \\
 & \ddots & & & & \ddots & \vdots \\
\vdots & & \ddots & & & \ddots & a_{n-1,n} \\
0 & \cdots & & a_{n,n-p} & \cdots & a_{n,n-1} & \sum\limits_{j\neq n-p,\ldots,n-1} |a_{nj}|
\end{bmatrix}. \tag{13}
$$

In particular D_1 and D_2 are diagonal and tridiagonal matrices:

$$
D_1 = \begin{bmatrix}
\sum_{j=1}^{n} |a_{1j}| & & \cdots & 0 \\
 & \sum_{j=1}^{n} |a_{2j}| & & \vdots \\
\vdots & & \ddots & \\
0 & \cdots & & \sum_{j=1}^{n} |a_{nj}|
\end{bmatrix}. \tag{14}
$$

$$
D_2 = \begin{bmatrix}
\sum\limits_{j\neq 2} |a_{1j}| & a_{12} & & & & \cdots & 0 \\
a_{21} & \sum\limits_{j\neq 1,3} |a_{2j}| & \ddots & & & & \vdots \\
 & \ddots & \ddots & \ddots & & & \\
 & & \ddots & \sum\limits_{j\neq k-1,k+1} |a_{kj}| & \ddots & & \\
\vdots & & & \ddots & \ddots & a_{n-1,n} \\
0 & \cdots & & & a_{n,n-1} & \sum\limits_{j\neq n-1} |a_{nj}|
\end{bmatrix}.
$$

The following lemma is proven in Appendix A. The lemma could be alternatively proven by using Gershgorin circle theorem [14].

Lemma 1 *Assume that A is an $n \times n$ symmetric matrix. Then $D_p - A$ is PSD. Furthermore, if A has only non-negative entries then $D_p - A$ has rank at most $n - 1$.*

One can show that a necessary condition for optimality of F_p is that $F_p - A$ is singular, i.e. the minimum of $F_p - A$ over $\{F_p : F_p - A \geq 0\}$ occurs at the boundary. Note that while this condition is not sufficient for optimality, Lemma 1 asserts that D_p satisfies this necessary condition.

An $n \times n$ matrix A is regular circulant [13] if it has the form:

$$\begin{bmatrix} a_0 & a_1 & & \cdots & a_{n-1} \\ a_{n-1} & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & a_1 \\ a_1 & \cdots & & a_{n-1} & a_0 \end{bmatrix},$$

and is symmetric regular circulant if it is regular circulant and it is symmetric about the diagonal, i.e. $a_{n-k} = a_k,\ k = 1, ..., n-1$. For a symmetric regular circulant matrix the vector $\frac{1}{\sqrt{n}}\underline{1}_+$ is an eigenvector, and $\sum_{i=1}^{n-1} a_i$ is the corresponding eigenvalue.

The following lemma is proven in Appendix B.

Lemma 2 *When $A = ((a_{ij}))$ is a regular symmetric circulant matrix and $a_{ij} \geq 0, \forall i, j$, the diagonal matrix D_1 minimizes the Frobenius norm $|||F - A|||_{\mathcal{F}}$ over all diagonal F which satisfy $F - A \geq 0$.*

While D_p may not always be optimal for non-circulant matrices A for $p \geq 1$, we can easily show that if $m > q$, then $|||D_m - A|||_{\mathcal{F}} = |||D_q - A|||_{\mathcal{F}} - 2\sum_{j=q+1}^{m}\sum_{k=1}^{n-q}(a_{k,k+q})^2$, so that $|||D_m - A|||_{\mathcal{F}} \leq |||D_q - A|||_{\mathcal{F}}$. Thus use of D_p for $p > 1$ should have as good or better average convergence rate than the diagonal matrix D_1.

4.1.1 Geometric Series with Over-Relaxation

The speed of convergence of the GS algorithm can be improved significantly by introducing an over-relaxation parameter ψ [22]. Assume that the splitting matrix F satisfies $F - A \geq 0$. The GS algorithm with over-relaxation (GSOR) is specified by:

$$\begin{aligned} \underline{\beta}_{GSOR}^{(k+1)} &= \underline{\beta}_{GSOR}^{(k)} - \psi F^{-1}(\ A\,\underline{\beta}_{GSOR}^{(k)} + \underline{e}_1\ ) \\ &= (I - \psi F^{-1}A)\,\underline{\beta}_{GSOR}^{(k)} - \psi F^{-1}\underline{e}_1. \end{aligned} \tag{15}$$

The parameter $\psi \in (0, 2]$ can be chosen to optimize the asymptotic convergence rate by minimizing $\rho(I - \psi F^{-1}A)$. The optimal choice of ψ that results in the fastest asymptotic rate of convergence depends on the maximum magnitude eigenvalue $|\lambda^M|_{\max}$ of $M = I - F^{-1}A$:

$$\psi_{asymptotic} = \frac{2}{2 - |\lambda^M|_{\max}}. \tag{16}$$

Note that $\psi_{asy} \in [1, 2)$ so that the eigenvalues of $I - \psi F^{-1}A$ are spread over the interval (-1,1) resulting in a non-monotonic convergent GS algorithm. Since $|\lambda^M|_{\max}$ is expensive to compute, an alternative strategy for optimizing the GSOR (15) is to optimize the average convergence rate by minimizing $|||F - \psi A|||_{\mathcal{F}}$. This results in the easily computed value of ψ:

$$\psi_{average} = \frac{\sum_{i,j} f_{ij} a_{ij}}{\sum_{i,j} (a_{ij})^2}, \tag{17}$$

where f_{ij} and a_{ij} denote elements of the matrices F and A.

4.1.2 Computational Issues

The most expensive computations of the GS algorithm are the two matrix-vector multiplications in (4). We can rewrite (4) as $\underline{\beta}^{(k+1)} = \underline{\beta}^{(k)} + \underline{w}$, where $\underline{w} = F^{-1}(F_Y \underline{\beta}^{(k)} + \underline{e}_1)$. While computing $\underline{t} = F_Y \underline{\beta}^{(k)} + \underline{e}_1$ requires $2n^2 + n$ flops, computing $\underline{w} = F^{-1} \underline{t}$ requires only n flops if F is a diagonal splitting matrix. The use of banded diagonal splitting matrices $F = D_p$ does not result in a significant overhead in flops per iteration if we use the Cholesky LU decomposition [14, 20] for D_p:

$$D_p = L_p L_p^T,$$

where L_p denotes a lower triangular matrix. Since D_p is a banded diagonal, L_p has only p non-zero lower diagonals. In particular for $p \ll n$ the Cholesky decomposition requires only $\mathcal{O}(p^2 n)$ flops [20]. If we let $\underline{t} = F_Y \underline{\beta}^{(k)} + \underline{e}_1$ in (4), and $\underline{w} = D_p^{-1} \underline{t}$, then $\underline{w}$ can be computed in $\mathcal{O}(p^2 n)$ flops by solving

$$L_p L_p^T \underline{w} = \underline{t}$$

using back-substitution.

When F_Y is of the factored form $F_Y = A^T \Gamma A$ with Γ an $m \times m$ diagonal, and A is an $m \times n$ matrix, implementation of the recursion (4) requires $\mathcal{O}(n \times m)$ bytes of storage as compared to $\mathcal{O}(n^2)$ required for general F_Y. When $n \ll m$ or when A is sparse this corresponds to a significant memory savings.

5 Conjugate Gradient

The often slow convergence of the GS algorithm led us to consider using over-relaxation, which improves the convergence rate at the expense of monotonicity. Having sacrificed monotonicity, it is natural to ask whether there are other iterative methods with still faster convergence. The conjugate gradient (CG) method can be used to solve the linear system of equations:

$$A\, \underline{u} = \underline{b}, \tag{18}$$

for a non-singular matrix A. The CG algorithm is an iterative method, but for solving linear equations it converges to the exact solution in a finite number of iterations when run with infinite precision.

If we substitute $F_Y = A$, and $\underline{b} = \underline{e}_1 = [1,0,0,...,0]^T$ in (18), then the solution to the system of equations (18) will be the first column of the inverse of F_Y^{-1}. The following CG algorithm is tailored to symmetric, positive definite matrices F_Y [20].

CG Algorithm

INITIALIZE: $\underline{u}^{(0)} = \underline{0}$; $\underline{r}^{(0)} = \underline{e}_1$

RECURSION:

$$\begin{aligned}
\alpha^{(i)} &= \begin{cases} 0 & i = 0 \\ \frac{<\underline{r}^{(i)}, \underline{r}^{(i)}>}{<\underline{r}^{(i-1)}, \underline{r}^{(i-1)}>} & i > 0 \end{cases} \\
\underline{p}^{(i)} &= \underline{r}^{(i)} + \alpha^{(i)} \, \underline{p}^{(i-1)} \\
\lambda^{(i)} &= \frac{< \underline{r}^{(i)}, \underline{r}^{(i)} >}{< \underline{p}^{(i)}, F_Y \, \underline{p}^{(i)} >} \\
\underline{u}^{(i+1)} &= \underline{u}^{(i)} + \lambda^{(i)} \, \underline{p}^{(i)} \\
\underline{r}^{(i+1)} &= \underline{r}^{(i)} - \lambda^{(i)} \, F_Y \, \underline{p}^{(i)}
\end{aligned}$$

BOUND APPROXIMATION:

$$\eta^{(i)} = \underline{e}_1^T \, \underline{u}^{(i)}. \tag{19}$$

The iterations are terminated when the residual error $\|r^{(i)}\|_2 = \|\underline{b} - F_Y \underline{u}^{(i)}\|_2$ is less than a user specified tolerance.

In the CG algorithm the direction vectors $\{\underline{p}^{(i)}\}_{i=1}^n$ are updated in such a way that $\underline{r}^i$ form an orthogonal basis for $I\!R^n$. With infinite precision arithmetic, this condition guarantees that $F_Y \underline{u}^{(n)} = \underline{b}$, where $\underline{u}^{(n)}$ is obtained at the n-th and final iteration in the CG algorithm. Each iteration of the algorithm involves a matrix-vector multiplication $F_Y \, \underline{p}^{(i)}$, requiring $\mathcal{O}(n^2)$ flops and therefore if the algorithm is stopped well before n iterations, an $\mathcal{O}(n^2)$ approximation to the CR bound is obtained. The convergence rate of the CG algorithm depends on the condition number of F_Y and its eigenvalue spread [14]. The speed of convergence of the CG algorithm can be further increased by pre-conditioning [20].

While the CG algorithm is guaranteed to give exact solution in a maximum of n iterations, the quantity $\eta^{(i)}$ (19) does not converge monotonically to the bound $[F_Y^{-1}]_{11}$. This means that if stopped at some iteration $k < n$ then the quantity $\eta^{(i)}$ calculated by the CG algorithm might not be a valid lower bound on the variance of an unbiased estimator $\widehat{\theta}_1$.

6 Recursive Methods to Approximate the Pseudo-Inverse of a Matrix

A square non-negative definite symmetric matrix F_Y of rank $r \leq n$ has an eigen-decomposition:

$$F_Y = \sum_{i=1}^{r} \sigma_i \, \underline{u}_i \underline{u}_i^T, \tag{20}$$

where $\sigma_1 \geq \sigma_2, ..., \geq \sigma_r > 0$ are the positive eigenvalues of F_Y arranged in decreasing order, and $\{\underline{u}_i\}$ is an orthonormal set of eigenvectors of F_Y. The Moore-Penrose pseudo-inverse, denoted by F_Y^+, is given by [14, 20]:

$$F_Y^+ = \sum_{i=1}^{r} \frac{1}{\sigma_i} \underline{u}_i \underline{u}_i^T. \tag{21}$$

The Moore-Penrose pseudo-inverse provides an unbiased CR bound for the case of singular FIM F_Y [2]. Direct methods for calculation of the pseudo-inverse require performing an eigen-decomposition of F_Y. Recursive techniques for the calculation of the pseudo-inverse [9, 10, 11, 12] are computationally expensive: $\mathcal{O}(n^3)$ flops. In this section we give two iterative methods of approximation to the Moore-Penrose pseudo-inverse for use in approximating a 1×1 sub-block of F_Y^+ requiring only $\mathcal{O}(n^2)$ flops per iteration.

6.1 Method 1: Perturbation

This method is based on adding a full rank matrix perturbation to F_Y to make it invertible. The objective is to keep the perturbation parameter ϵ as small as possible so that the approximation does not deviate too much from the actual solution. The approximation is based on the following [14]:

Theorem 1 *Let F_Y be an $n \times n$, non-negative definite symmetric matrix with rank $r \leq n$. Then the pseudo-inverse of F_Y given in (21), has the representation:*

$$F_Y^+ = \lim_{\epsilon \to 0^+} G_1(\epsilon),$$

where

$$G_1(\epsilon) \stackrel{def}{=} (F_Y^T F_Y + \epsilon I)^{-1} F_Y^T, \tag{22}$$

$\epsilon > 0$ is a scalar, and I is the $n \times n$ identity matrix [14]. Furthermore for all $\epsilon > 0$,

$$F_Y^+ \geq G_1(\epsilon). \tag{23}$$

Proof: We have,

$$\begin{aligned} F_Y^+ - G_1(\epsilon) &= \sum_{i=1}^{r} \frac{1}{\sigma_i} \underline{u}_i \underline{u}_i^T - \sum_{i=1}^{r} \frac{\sigma_i}{(\sigma_i{}^2 + \epsilon)} \underline{u}_i \underline{u}_i^T & (24) \\ &= \sum_{i=1}^{r} \frac{\epsilon}{\sigma_i(\sigma_i{}^2 + \epsilon)} \underline{u}_i \underline{u}_i^T & (25) \\ &\geq 0. \end{aligned}$$

Clearly, $\lim_{\epsilon\to 0} G_1(\epsilon) = F_Y^+$, and the theorem follows. ∎

From Theorem 1, $G_1(\epsilon)$ can be viewed as an approximation to the pseudo-inverse of F_Y based on Tikhonov regularization which has the property (23) that $G_1(\epsilon)$ is a valid lower bound on $\mathrm{cov}_{\underline{\theta}}(\underline{\widehat{\theta}})$ for all $\epsilon > 0$.

The choice of ϵ in Theorem 1 affects the error of approximation $F_Y^+ - G_1(\epsilon)$. For example if the trace norm $|||F_Y^+ - G_1(\epsilon)|||_{\mathcal{T}} = \mathrm{trace}(F_Y^+ - G_1(\epsilon))$ is small the approximation error will be small due to non-negative definiteness of $F_Y^+ - G_1(\epsilon)$. In the following discussion we will study the choice of ϵ in terms of the trade-off between the error of approximation, and the condition number of $F_Y^T F_Y + \epsilon I$.

We want ϵ to be as small as possible for best approximation error $F_Y^+ - G_1(\epsilon)$. On the other hand for singular F_Y, we can not make ϵ too small, or else the matrix inversion $(F_Y^T F_Y + \epsilon I)^{-1}$ in (22) becomes ill-conditioned. Therefore, the choice of ϵ, and hence the error of approximation, must be driven by the maximum tolerable condition number c of $F_Y^T F_Y + \epsilon I$, where the condition number of a symmetric positive semi-definite matrix A is defined as the ratio of the maximum and minimum eigenvalues of A. The condition number of $F_Y^T F_Y + \epsilon I$ must satisfy:

$$\frac{\sigma_1^2 + \epsilon}{\epsilon} \approx \frac{\sigma_1^2}{\epsilon} \leq c. \tag{26}$$

One can find σ_1 using the power method [14, 20]. Thus choosing $\epsilon \geq \frac{\sigma_1^2}{c}$ guarantees that the condition number of $F_Y^T F_Y + \epsilon I$ lies below the maximum tolerable condition number c, and that the matrix can be inverted without running into precision problems.

Since in many cases the maximum eigenvalue σ_1 is not available, ϵ can be chosen more conservatively as:

$$\epsilon \geq \frac{\sum_{i=1}^r \sigma_i^2}{c} = \frac{\mathrm{trace}(F_Y^T F_Y)}{c}. \tag{27}$$

The above choice of ϵ satisfies (26).

For $\epsilon \geq \frac{\sigma_1^2}{c}$, the trace error of approximation, denoted δ_p, can be easily computed.

$$\begin{aligned}
\delta_p &\overset{def}{=} |||F_Y^+ - G_1(\epsilon)|||_{\mathcal{T}} \\
&= \sum_{i=1}^r \frac{\epsilon}{\sigma_i({\sigma_i}^2 + \epsilon)} \\
&= \sum_{i=1}^r \frac{\frac{\sigma_1^2}{c}}{\sigma_i({\sigma_i}^2 + \frac{\sigma_1^2}{c})}.
\end{aligned}$$

We can upper bound δ_p by $\delta_p^{\max}$:

$$\delta_p \leq \delta_p^{\max} = r\,\frac{\frac{\sigma_1^2}{c}}{\sigma_r({\sigma_r}^2 + \frac{\sigma_1^2}{c})}.$$

6.2 Recursive Algorithm for Perturbation Method

Both the GS and the CG algorithms can be readily adapted to the perturbation method by the following steps:

1. Compute the column $\underline{\zeta}_1 = (F_Y^T F_Y + \epsilon I)^{-1} \underline{e}_1$ using either the GS or the CG method of section 3.

2. Approximate the 1-1 element of the pseudo-inverse of F_Y: $[G_1(\epsilon)]_{11} = \underline{e}_1^T F_Y \underline{\zeta}_1$.

If GS is to be used in step 1, a splitting matrix F must be chosen such that $F \geq (F_Y^T F_Y + \epsilon\, I)$. The resultant un-relaxed GS algorithm has monotonic convergence since $\underline{e}_1^T (\underline{\zeta}_1^{(k)} - \underline{\zeta}_1^{(k+1)})$ $= \underline{e}_1^T \left([I - F^{-1}(F_Y^T F_Y + \epsilon I)]^{k+1} F^{-1} \right) \underline{e}_1 \geq 0$. Therefore, the algorithm generates a sequence of approximations $\underline{\zeta}_1^{(k)}$ which monotonically converge to $\underline{e}_1^T G_1(\epsilon) \underline{e}_1 \leq \underline{e}_1^T F_Y^+ \underline{e}_1$ as $k \rightarrow \infty$.

The GS recursions for the perturbation method are given by:

$$\underline{\beta}^{(k+1)} = \underline{\beta}^{(k)} - F^{-1}(\ F_Y^T F_Y\, \underline{\beta}^{(k)} + \epsilon \underline{\beta}^{(k)} + \underline{e}_1\ ).$$

If we denote $\underline{t} = F_Y\, \underline{\beta}^{(k)}$, then we have an additional matrix-vector multiplication ($2n^2$ flops), $F_Y^T \underline{t}$, when compared with (4) for the computation of F_Y^{-1}. Similarly, the CG algorithm has an additional $2n^2$ flops per iteration when applied to matrix perturbations via computation of the solutions $\underline{\zeta} = \underline{\zeta_1}$ to $(F_Y^T F_Y + \epsilon I)\underline{\zeta} = \underline{e}_1$.

The choice of ϵ directly affects the asymptotic rate of convergence of the GS and the CG algorithms. For the GS algorithm the rate of convergence depends on the matrix $M = M^* = [I - F^{-1}(F_Y^T F_Y + \epsilon I)]$. Since F_Y is rank deficient the spectral radius $\rho(M^*)$ is equal to $1 - O(\epsilon)$, which is close to unity for small ϵ resulting in slow asymptotic rate of convergence. Thus we see that there is a trade-off between the asymptotic error of approximation $|||F_Y^+ - G_1(\epsilon)|||_{\mathcal{T}}$ and the asymptotic rate of convergence determined by $\rho(M^*)$. In particular we want ϵ to be small for good error of approximation, while ϵ should be large for fast asymptotic rate of convergence. Ideally, we would like the rate of convergence to be rapid for small ϵ. The following method reduces the effect of perturbation by differencing.

6.3 Method 2: Differencing Method

Theorem 2 *Let F_Y be an $n \times n$, non-negative definite symmetric matrix with rank $r \leq n$. Then the pseudo-inverse of F_Y given in (21), has the representation:*

$$F_Y^+ = \lim_{\epsilon \rightarrow 0+} G_2(\epsilon),$$

where,

$$G_2(\epsilon) \stackrel{def}{=} \frac{a}{a-b}(F_Y + a\,\epsilon I)^{-1} - \frac{b}{a-b}(F_Y + b\,\epsilon I)^{-1}, \tag{28}$$

$\epsilon > 0$ *is a scalar,* $a, b \in (0, \infty)$*,* $a \neq b$*, and* I *is the* $n \times n$ *identity matrix. Furthermore for all* $\epsilon > 0$*,*

$$F_Y^+ \geq G_2(\epsilon). \tag{29}$$

Note that due to (29) the CR bound approximated using Theorem 2 will be a valid lower bound.

Proof: Similar to the proof of Theorem 1, one can show:

$$\begin{aligned} F_Y^+ - G_2(\epsilon) &= \sum_{i=1}^{r} \frac{1}{\sigma_i} \underline{u}_i \underline{u}_i^T - \sum_{i=1}^{r} \frac{a}{(a-b)(\sigma_i + a\epsilon)} \underline{u}_i \underline{u}_i^T + \sum_{i=1}^{r} \frac{b}{(a-b)(\sigma_i + b\epsilon)} \underline{u}_i \underline{u}_i^T \\ &= \sum_{i=1}^{r} \frac{\epsilon\, (a\, \sigma_i + b\, \sigma_i + a\, b\, \epsilon)}{\sigma_i\, (\sigma_i + a\, \epsilon)\, (\sigma_i + b\, \epsilon)} \underline{u}_i \underline{u}_i^T \qquad (30) \\ &\geq 0. \qquad (31) \end{aligned}$$

Clearly $\lim_{\epsilon \to 0} G_2(\epsilon) = F_Y^+$, and the theorem follows. ∎

Following the approach outlined for the perturbation method in section 6.1, we want to choose ϵ such that the condition numbers of $(F_Y + a\epsilon I)$ and $(F_Y + b\epsilon I)$ are less than the maximum tolerable condition number c. The condition numbers of $(F_Y + \epsilon I)$ and $(F_Y + b\epsilon I)$ must satisfy:

$$\frac{\sigma_1 + a\epsilon}{\epsilon}, \frac{\sigma_1 + b\epsilon}{\epsilon} \approx \frac{\sigma_1}{\epsilon} \leq c,$$

for $\frac{\sigma_1}{\epsilon} \gg a, b$, a situation satisfied in most practical applications. Hence by choosing $\epsilon \geq \frac{\sigma_1}{c}$, the matrix inversions $(F_Y + a\epsilon I)^{-1}$ and $(F_Y + b\epsilon I)^{-1}$ will be well conditioned. For $\epsilon = \frac{\sigma_1}{c}$, the asymptotic error of approximation δ_d associated with the differencing method is:

$$\begin{aligned} \delta_d &= |||F_Y^+ - G_2(\epsilon)|||_T \\ &= \sum_{i=1}^{r} \frac{\epsilon\, ((a+b)\, \sigma_i + a\, b\, \epsilon)}{\sigma_i\, (\sigma_i + a\, \epsilon)\, (\sigma_i + b\, \epsilon)}, \end{aligned}$$

and since δ_d is monotone in ϵ,

$$\delta_d \leq \delta_d^{\max} = r\, \frac{\frac{\sigma_1}{c}\, ((a+b)\, \sigma_r + a\, b\, \frac{\sigma_1}{c})}{\sigma_r\, (\sigma_r + a\, \frac{\sigma_1}{c})\, (\sigma_r + b\, \frac{\sigma_1}{c})}. \tag{32}$$

Since in many cases σ_1 is not available ϵ can be chosen conservatively as:

$$\epsilon \geq \frac{\mathrm{trace}(F_Y)}{c} = \frac{\sum_{i=1}^{r} \sigma_i}{c}. \tag{33}$$

6.3.1 Recursive Algorithm for Differencing Method

Using the GS or the CG algorithms, $[G_2(\epsilon)]_{11}$ can be computed by the following steps.

1. Approximate columns $\underline{\gamma}_1 = (F_Y + a\epsilon I)^{-1} \underline{e}_1$ and $\underline{\gamma}_2 = (F_Y + b\epsilon I)^{-1} \underline{e}_1$ using either the GS or the CG algorithms of section 3.

2. Use approximation $[G_2(\epsilon)]_{11} \approx \underline{e}_1^T \left(\frac{a}{a-b}\underline{\gamma}_1 - \frac{b}{a-b}\underline{\gamma}_2\right)$ to approximate $[F_Y^+]_{11}$.

If the GS algorithm is to be used in step 1, two splitting matrices F_1 and F_2 must be chosen such that $F_1 > (F_Y + a\epsilon I)$, and $F_2 > (F_Y + b\epsilon I)$ in step 1. The asymptotic rate of convergence depends on the choice of ϵ in $M_1^* = I - F_1^{-1}(F_Y + \epsilon I)$, and $M_2^* = I - F_2^{-1}(F_Y + \epsilon I)$.

6.4 Comparison of Perturbation and Differencing Methods

If for a fixed condition number c we choose $\epsilon_d = \frac{\sigma_1}{c}$ for the differencing method, and $\epsilon_p = \frac{\sigma_1^2}{c}$ for the perturbation method, then the difference between the error bounds $\delta_p^{\max}$ and $\delta_d^{\max}$ is:

$$\begin{aligned} \delta_p^{\max} - \delta_d^{\max} &= r\,\frac{\frac{\sigma_1^2}{c}}{\sigma_r({\sigma_r}^2 + \frac{\sigma_1^2}{c})} - r\,\frac{\frac{\sigma_1}{c}\left((a+b)\,\sigma_r + a\,b\,\frac{\sigma_1}{c}\right)}{\sigma_i\,(\sigma_r + a\,\frac{\sigma_1}{c})\,(\sigma_r + b\,\frac{\sigma_1}{c})} \\ &= \frac{r\sigma_1}{c}\left[\frac{\sigma_1}{{\sigma_r}^2 + \frac{\sigma_1^2}{c}} - \frac{\sigma_1}{\frac{{\sigma_r}^2}{(a+b)\frac{\sigma_r}{\sigma_1} + \frac{ab}{c}} + \frac{\sigma_1^2}{c}}\right]. \end{aligned}$$

From which it follows that if $(a+b)\frac{\sigma_r}{\sigma_1} + \frac{ab}{c} < 1$ then the asymptotic error of approximation of the differencing method is expected to be better than that of the perturbation method. In particular this will hold when the maximum tolerable condition number $c \gg 1$ and the ratio of the largest to smallest non-zero eigenvalues of F_Y satisfy $\frac{\sigma_1}{\sigma_r} \gg 1$.

While the GS implementation of the perturbation method gives a monotonically convergent algorithm, the GS implementation of the differencing method does not converge monotonically due to the differencing of the two monotone series. Thus unlike the other recursive algorithms, the GS implementation of the perturbation method can be terminated at any iteration to obtain a valid lower bound on estimator variance.

7 Applications

In this section we apply the recursive algorithms described in the previous section to: 1) Single Photon Emission Computed Tomography (SPECT), and 2) one-dimensional discrete deconvolution.

7.1 Single Photon Emission Computed Tomography

A SPECT system consists of three basic components: 1) a source of γ-ray photons, 2) a photon sensitive γ-ray detector and, 3) a γ-ray collimator. The function of the collimator is to reduce the uncertainty associated with the emission location of a detected γ-ray to a line or a strip in the field of view (Figure 3). During the image scan, the γ-ray detector is rotated through small steps 180^o around the source. A γ-ray photon passing through one of the collimator slits at one of the rotation angles is counted as an event acquired in one 'detector bin'. For reconstruction the source domain is divided into n small regions, called pixels. The detection process is governed by Poisson statistics:

$$f_{\underline{Y}}(\underline{y};\underline{\theta}) \;=\; \prod_{j=1}^{d} \frac{\mu_j^{y_j}}{y_j!} e^{\mu_j}, \tag{34}$$

where θ_i is the average γ-ray intensity at the i-th pixel; $i = 1, \ldots n$, and μ_j is the average γ-ray intensity at the j-th detector bin; $j = 1, \ldots, d$:

$$\underline{\mu} = A\,\underline{\theta}, \tag{35}$$

where A is the $d \times n$ matrix that depends on the tomographic system [4].

The objective is to estimate the object intensity at each pixel $\underline{\theta} = [\theta_1, \ldots, \theta_n]^T$ given the set of observations Y. We make the standard assumption that $\theta_i \geq 0$ and $\mu_j \geq 0$, $i = 1, \ldots, n$, $j = 1, \ldots, d$. From (34) it can be easily shown that the FIM is of the factored form:

$$F_Y(\underline{\theta}) \;=\; A^T\,[\mathrm{diag}(\mu_i)]^{-1}\,A, \tag{36}$$

which ensures an $\mathcal{O}(n \times d)$ memory requirement for the recursive CR bound algorithms. Since A is sparse this corresponds to a significant memory savings.

The FIM, denoted F_{EM}, for the complete-data set in emission tomography satisfies $F_{EM} - F_Y \geq 0$ and has the form [1]:

$$F_{EM} = \mathrm{diag}(\underline{1}^T\,A)\,[\mathrm{diag}(\underline{\theta})]^{-1}, \tag{37}$$

where $\mathrm{diag}(\underline{z})$ denotes a diagonal $n \times n$ matrix with the elements of the n-vector $\underline{z}$ along the diagonal. Note that since F_{EM} is a diagonal matrix, it is well suited for use as a splitting matrix in the GS algorithm.

7.2 Numerical Comparisons

All the simulations were done using MATLAB version 4.1 on an HP-755 computer running under the $HP-UX$ operating system.

7.2.1 GS Implemented with p-Diagonal Matrices Compared to CG

Several two-dimensional objects were investigated (Figure 4). The high intensity pixels have a normalized intensity value of 2, the dark grey and the light grey pixels have normalized intensity 1 and 0.5 respectively. The black pixels are of zero intensity. The object lies in a field of view that is a disk of diameter 32 pixels. The total number of unknown parameters θ_i is $n = 716$ pixels.

Figure 5 and 6 display the effect of increasing the number of diagonals in the splitting matrix for the GS algorithm, and a comparison to CG. Figure 5 and 6 correspond to the brain image with two different pixels of interest, labeled 1 and 2 in Figure 4. In Table 1 the total number of iterations to converge, the Frobenius norm $|||\Delta|||_{\mathcal{F}} = |||F - F_Y|||_{\mathcal{F}}$ between the splitting matrix F and the FIM F_Y, the spectral norm $\rho(M) = \lambda_{\max}^M$ of $M = I - F^{-1}F_Y$, and the total number of Mflops to converge are given for the three image intensities and five different choices of the splitting matrices F for the GS algorithm. The splitting matrix D_λ is a diagonal matrix determined by numerically minimizing the spectral norm $\rho(M) = |\lambda^M|_{\max} = \lambda_{\max}^M$ for each F_Y corresponding to each image. Also shown is the convergence behavior of the CG algorithm. In all cases, the algorithms were initialized by the all zero vector corresponding to a zero image. For the results shown in Table 1 we used the convergence criterion (6) with a threshold $\gamma = 10^{-3}$, i.e. the algorithm was declared to have converged in N iterations when:

$$\frac{\|\underline{\beta}^{(k)} - F_Y^{-1}\underline{e}_1\|_2}{\|F_Y^{-1}\underline{e}_1\|_2} \leq \gamma = 10^{-3},$$

for more than 5 consecutive iterations $k = N, N+1, ..., N+4$, where $\underline{\beta}^{(k)}$ is the approximation to $F_Y^{-1}\underline{e}_1$ at the k-th iteration of either the GS or the CG algorithm. For the GS algorithm the total number of Mflops, shown in Table 1, was calculated taking account of a Cholesky LU decomposition of the corresponding splitting matrix F. The following comments on Table 1 are of interest:

- In each case the Frobenius norm $|||\Delta|||_{\mathcal{F}}$ is a much better predictor of the actual number of iterations required for GS convergence than is the root convergence factor $\lambda_{\max}^M$, which only predicts the asymptotic convergence rate. In particular, observe that while the use of $F = D_\lambda$ minimizes each $\lambda_{\max}^M$ for each of the three objects, our choice of diagonal matrix $F = D_1$ introduced in (14) converges in fewer iterations and has smaller Frobenius error norm.

- For the complicated brain image the diagonal matrices D_1 and D_λ were found to be identical within the machine precision. On the other hand, for the simple uniform image, the diagonal matrices D_1 and F_{EM} are analytically identical, while D_1 and D_λ are quite different. For these cases F_{EM} and D_λ are not competitive with D_1 in terms of average convergence rate.

- As expected, since D_p gives smaller Frobenius norm $|||D_p - F_Y|||_{\mathcal{F}}$, for increasing p number of non-zero diagonals the number of iterations to converge decreases as p is increased. Unexpectedly, increasing p also decreases the root convergence factor $\lambda_{\max}^M$ resulting in more rapid asymptotic convergence rate. The use of 50 diagonals in the splitting matrix brings the required number of iterations of GS to within a factor of approximately 2 of the CG.

- The CG algorithm converges most rapidly, in under 35 iterations for all cases, and is approximately 2 to 3 times less computationally costly to implement as compared to the most rapid GS algorithm, indicated by $F = D_{50}$.

- The CG algorithm can have significant overshoots. In Figures 5 and 6 only 5% overshoot was observed. The magnitude of the overshoot was observed to depend on the initial condition. Thus if a monotone algorithm is desired, guaranteeing a valid bound at each iteration, the GS algorithm with $F = D_{50}$ is preferable to the CG algorithm.

7.3 Comparison of CG, GS and GS with Over-Relaxation

Figure 1 shows a comparison of the CG, the GS and the GSOR algorithms, the latter two implemented with the diagonal splitting matrix D_1 of (14), for pixel **1** in the brain phantom. To establish a benchmark for GSOR the over-relaxation parameter $\psi = 1.99$ was chosen using (16) for fastest asymptotic convergence. Since in general the use of (16) requires the impractical computation of the maximum eigenvalue of M, the asymptotic convergence of the implementable GSOR algorithm using (17) may be somewhat slower. The GSOR exhibits a much faster initial rate of convergence when compared to the GS, however it oscillates around the point of convergence and fails to meet our convergence criterion of $\gamma = 10^{-3}$ even after the GS achieves convergence.

7.4 One-Dimensional Discrete Deconvolution

We illustrate the ill-conditioned CR bound algorithms described in Section 6 for the problem of discrete deconvolution, specified by the model:

$$\underline{Y} = A\underline{\theta} + \underline{q}, \tag{38}$$

where $\underline{Y}$ is the blurred data, A corresponds to a convolution kernel, $\underline{\theta}$ is the parameter we want to deconvolve or estimate and $\underline{q}$ is additive white Gaussian noise with covariance Σ. The blurring in the data can be due to several factors, e.g. imperfect measuring devices. We chose a discrete Gaussian kernel $a_{ij} = \frac{1}{\sqrt{2\pi}} e^{\frac{-(i-j)^2}{2w^2}}$ in these simulations. Philip [21] has remarked that for the discrete deconvolution problem the most ill-conditioned kernels have the form of discrete Gaussian kernels. The FIM F_Y is given by:

$$F_Y = A^T \mathrm{cov}(\underline{Y})^{-1} A.$$

By choosing the width of the Gaussian kernel A we can vary the condition number of the FIM F_Y.

7.5 Dependence of the Rate of Convergence on the Condition Number

First we consider a uniform source as shown in Figure 2 a). We used the same convergence criterion (6) as in the tomography example, i.e. $\gamma = 10^{-3}$, to terminate the iterations. The matrix A in (38) had dimension 300×300. Figure 7 indicates that the number of iterations required for convergence is more sensitive to the condition number for GS than for CG.

7.6 Performance of Pseudo-inverse Algorithm

In this section we consider the problem of super-resolution with an up-sampling factor of 2. This refers to a situation when the number of available independent observations are only a half the number of parameters we want to estimate. The A matrix has dimension 150×300. This problem is obviously under-determined. The width, w, of the Gaussian convolution kernel was varied from 5 to 32. The region of interest was the center pixel θ_{150} for the point source displayed in Figure 2. The value of ϵ was chosen using (27) and (33). The maximum allowable condition number used was $c = 10^{12}$. For the differencing method we chose a=1 and b=2 in (28). Figure 8 clearly indicates that differencing method is more accurate for all resolutions as measured by the normalized error of approximation $\frac{\mathrm{trace}(F_Y^+ - G(\epsilon))}{\mathrm{trace}(F_Y^+)}$, where $G(\epsilon) = G_1(\epsilon)$ for the perturbation method and $G(\epsilon) = G_2(\epsilon)$ for the differencing method. The accuracy for both the methods degrades for wide convolution kernels, but the degradation is more pronounced in the perturbation method. Thus for ill-conditioned problems corresponding to wider convolution kernels and resulting in large but finite condition numbers, the differencing method is more accurate than the perturbation method.

To study the effect of ϵ on the normalized error of approximation $\frac{\mathrm{trace}(F_Y^+ - G(\epsilon))}{\mathrm{trace}(F_Y^+)}$ of the differencing and the perturbation method we fixed the width of the Gaussian kernel to 20 points. The accuracy of both the algorithms degrades gracefully over a wide range of ϵ. In both cases the value of ϵ chosen using (27) and (33), indicated by vertical bars in Figure 9, is close to the knee of the curves. The differencing method is clearly more accurate than the perturbation method over the whole range of ϵ. The perturbation method fails to achieve the desired value of $\frac{\mathrm{trace}(F_Y^+ - G(\epsilon))}{\mathrm{trace}(F_Y^+)} \approx 0$ for all values of ϵ.

7.6.1 Non-monotone Convergence of Differencing Method

Figure 10 shows the non-monotone convergence curves of the differencing method. The width of the Gaussian kernel was 3 points.

A Proof of Lemma 1

The matrix $D_p - A$ has the form:

$$\begin{bmatrix}
\sum\limits_{j\neq 2,\ldots,p} |a_{1j}| & 0 & \cdots & 0 & -a_{1,p+1} & \cdots & -a_{1n} \\
0 & \sum\limits_{j\neq 1,3,\ldots,p+1} |a_{2j}| & & & \ddots & & \vdots \\
\vdots & \ddots & & & & \ddots & -a_{n-p-1,n} \\
 & & \ddots & & & \ddots & \\
0 & & & \sum\limits_{\substack{j\neq k-p,\ldots,k-1 \\ j\neq k+1,\ldots,k+p-1}} |a_{kj}| & & & 0 \\
-a_{p+1,1} & \ddots & & & \ddots & & \\
 & & \ddots & & & \ddots & \vdots \\
\vdots & & \ddots & & & \ddots & 0 \\
-a_{n1} & \cdots & -a_{n,n-p-1} & 0 & \cdots & 0 & \sum\limits_{j\neq n-p,\ldots,n-1} |a_{nj}|
\end{bmatrix}.$$

Since $D_p - A$ is a special case of $D_1 - A$ for $a_{ij} = 0, |i-j| \geq p$, it suffices to prove that the following matrix is Positive Semi-Definite (PSD):

$$D_1 - A = \begin{bmatrix} \sum_{j\neq 1} \mid a_{1j} \mid & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & \sum_{j\neq 2} \mid -a_{2j} \mid & \cdots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & \sum_{j\neq n} \mid -a_{nj} \mid \end{bmatrix} \tag{39}$$

Define:

$$M^{(q)} = \begin{bmatrix} N & \underline{b} \\ \underline{b}^T & a \end{bmatrix},$$

where,

$$\begin{aligned} \underline{b} &= [-a_{q1}, -a_{q2}, \ldots, -a_{q,q-1}]^T, \\ a &= \sum_{i=1}^{q-1} |b_i|, \\ N &= M^{(q-1)} + \mathrm{diag}_i(|b_i|). \end{aligned}$$

Since $D_1 - A$ is symmetric $M^{(q)}$ is the $q \times q$ upper left hand block of the $n \times n$ matrix $D_1 - A$. We proceed by induction. Assume that $M^{(q-1)}$ is positive semi-definite. Since $a = 0$ implies $\underline{b} = \underline{0}$, when $a = 0$, $M^{(q)}$ is obviously positive semi-definite. We therefore assume $a > 0$, in which case $\underline{b}$ is not identically zero. Consider the following factorization of $M^{(q)}$.

$$M^{(q)} = \begin{bmatrix} I & \frac{1}{a}\underline{b} \\ \underline{0}^T & 1 \end{bmatrix} \begin{bmatrix} N - \frac{1}{a}\underline{b}\underline{b}^T & \underline{0} \\ \underline{0}^T & a \end{bmatrix} \begin{bmatrix} I & \underline{0} \\ \frac{1}{a}\underline{b}^T & 1 \end{bmatrix}.$$

We only need to show that the $(q-1) \times (q-1)$ matrix,

$$N - \frac{1}{a}\underline{b}\underline{b}^T = M^{q-1} + \mathrm{diag}_i(|b_i|) - \frac{1}{a}\underline{b}\underline{b}^T$$

is PSD. Since the sum of PSD matrices is PSD, it is sufficient to show that

$$\text{diag}_i(|b_i|) - \frac{1}{a}\underline{b}\underline{b}^T \geq 0.$$

Consider for any $\underline{x} = [x_1, x_2, ..., x_{q-1}]^T$:

$$\begin{aligned}
\underline{x}^T \left(\text{diag}_i(|b_i|) - \frac{1}{a}\underline{b}\underline{b}^T\right) \underline{x} &= \sum_{i=1}^{q-1} x_i^2 |b_i| - \frac{1}{a}\left(\sum_{i=1}^{q-1} x_i b_i\right)^2 \\
&= \sum_{j=1}^{q-1} |b_j| \left(\sum_{i=1}^{q-1} x_i^2 \frac{|b_i|}{\sum_{j=1}^{q-1}|b_j|} - \left(\sum_{i=1}^{q-1} x_i \frac{b_i}{\sum_{j=1}^{q-1}|b_j|}\right)^2\right) \\
&\geq a\left(\sum_{i=1}^{q-1} x_i^2 \frac{|b_i|}{\sum_{j=1}^{q-1}|b_j|} - \left(\sum_{i=1}^{q-1} |x_i| \frac{|b_i|}{\sum_{j=1}^{q-1}|b_j|}\right)^2\right) \\
&= a\left(\sum_{i=1}^{q-1} p_i |x_i|^2 - \left(\sum_{i=1}^{q-1} p_i |x_i|\right)^2\right) \qquad (40) \\
&= a\sum_{i=1}^{q-1} p_i \left(|x_i| - \sum_{j=1}^{q-1} p_j |x_j|\right)^2 \\
&\geq 0.
\end{aligned}$$

Where in (40) we have defined $p_i = \frac{|b_i|}{\sum_{j=1}^{q-1}|b_j|}$, $p_i \in [0,1]$, $\sum_{i=1}^{q-1} p_i = 1$. Furthermore, for $\underline{x} = \underline{1}$, $(D_p - A)\underline{1} = \underline{0}$, where $\underline{1} = [1, ..., 1]^T$. This implies that $D_p - A$ has at least one zero eigenvalue, therefore $D_p - A$ is rank deficient with rank at most $n-1$.

■

B Proof of Lemma 2

Define $S(A)$ as a set of diagonal F such that $F - A \geq 0$. Let us denote $F_{\min} \in S(A)$ as the matrix F that minimizes $|||F - A|||_{\mathcal{F}}$, and $F_{\min}(\underline{x})$ as the matrix that minimizes $|||F - A|||_{\mathcal{F}}$ over the set of diagonal F such that $\underline{x}^T(F - A)\underline{x} \geq 0$. It can be easily shown using Lagrange multiplier theory that $F_{\min}(\underline{x})$ is given by:

$$F_{\min}(\underline{x}) = \text{diag}(A) - \frac{\underline{x}^T\left(\text{diag}(A) - A\right)\underline{x}}{\underline{x}^T \,\text{diag}\left(\underline{x}\underline{x}^T\right)\, \underline{x}} \text{diag}\left(\underline{x}\underline{x}^T\right).$$

Clearly $\mathcal{S}(A) \subset \mathcal{S}_{\underline{x}}(A)$, since the condition $F - A \geq 0$ implies $\underline{x}^T(F - A)\underline{x} \geq 0$ for all $\underline{x}$, so by the definition of $F_{\min}$:

$$|||F_{\min}(\underline{x}) - A|||_{\mathcal{F}} \leq |||F_{\min} - A|||_{\mathcal{F}}, \ \ \forall \underline{x}.$$

For $\underline{x} = \underline{1}$ and A symmetric regular circulant with non-negative entries one can verify that $F_{\min}(\underline{1}) = (a_{11} + ... + a_{1n})I = D_1$. By Lemma 1, $D_1 - A \geq 0$. Therefore, $F_{\min}(\underline{1}) \in S(A)$. Thus by the definition of $F_{\min}$ we must also have,

$$|||F_{\min} - A|||_{\mathcal{F}} \leq |||F_{\min}(\underline{1}) - A|||_{\mathcal{F}}$$

The above two inequalities together imply that

$$F_{\min}(\underline{1}) = F_{\min} = D_1.$$

∎

References

[1] A.O. Hero and J.A. Fessler
A Recursive Algorithm for Computing CR-Type Bounds on Estimator Covariance, to appear in IEEE Transactions on Information Theory, 1994.

[2] J. D. Gorman and Alfred O. Hero
Lower bounds on parametric estimators with constraints. *IEEE transactions on Information Theory*, 36:6, pp. 1285-1301, 1991.

[3] A.P. Dempster, N.M. Laird and D.B. Rubin
Maximum Likelihood From Incomplete Data via the EM Algorithm; *Journal of the Royal Statistical Society*, Ser. B, Vol. 39, pp 1-37, 1977

[4] J. A. Fessler
Penalized Weighted Least-Squares Image Reconstruction for Positron Emission Tomography, IEEE Transactions on Medical Imaging, 13:2, June 1994, In Press.

[5] A.O. Hero, J.A. Fessler and W.L. Rogers
A Fast Recursive Algorithm for Computing CR-Type Bounds for Image Reconstruction Problems. *IEEE Nuclear Science Symposium and Medical Imaging Conference*, Orlando, 1993

[6] A. O. Hero
A Cramer-Rao Type Lower Bound for Essentially Unbiased parameter Estimation, Technical Report 890, Lincoln Laboratory, MIT 1992.

[7] M. Usman and A.O. Hero
The Analysis of a Single Photon Emission Computed Tomography (SPECT) System Using Mutual Information and Cramer-Rao Bound as Performance Criteria, CSPL Technical Report, The University of Michigan, 1993.

[8] A. Kuruc
Lower bounds on Multiple-source Direction Finding in the Presence of Direction-dependent Antenna-array-calibration errors, Technical Report 799, MIT Lincoln Laboratory, Lexington MA, 1989.

[9] T. N. E. Greville
Some Applications of the Pseudo-Inverse of a Matrix. *SIAM Review*, vol. 2, pp. 15-22, 1960

[10] P. A.Janakiraman and S. Renganathan
Recursive Computation of Pseudo-Inverse of Matrices. *Automatica*, vol. 18, pp. 631-633, 1982.

[11] W. Guo-rong and C. Yong-lin
A Recursive Algorithm for Computing the Weighted Moore-Penrose Inverse A^{+}_{MN}. *Journal of Computational Mathematics*, vol. 4:1, pp. 74-85, 1986

[12] R.C. K. Lee
Optimal Estimation, Identification, and Control, Research Monograph no. 28, The MIT Press, Cambridge Massachusetts 1964.

[13] R.A. Horn and C.R. Johnson
Matrix Analysis, Cambridge, 1985.

[14] G.H. Golub and C.F. van Loan
Matrix Computations (Second Edition), The Johns Hopkins University Press, 1989.

[15] I.A. Ibragimov and R.Z. Has'minskii
Statistical Estimation: Asymptotic Theory, Springer Verlag, New York, 1981.

[16] D.G. Luenberger
Optimization by Vector Space Methods, pp 290-299, John Wiley and Sons, 1969.

[17] D. Rose and R.A. Willoughby
Sparse Matrices and their Applications, Plenum Press, 1972.

[18] A.N. Tikhonov and V.Y. Arsenin
Solutions of Ill-Posed Problems, Halsted Press, 1977.

[19] L.A. Hageman and D.M. Young
Applied Recursive Methods, pp 18-22, Academic Press, 1981.

[20] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B.P. Flannery
Numerical Recipes in FORTRAN, The Art of Scientific Computing, (Second Edition), Cambridge University Press, 1992.

[21] J. Philip
The Most Ill-posed Non-negative Kernels in Discrete Deconvolution, Inverse Problems, 3:2, pp 309-328, May 1987.

[22] D.M. Young
Iterative Solution of Large Linear Systems, Academic Press, New York, 1971.

[23] R. Barret
Templates for the Solutions of Linear Systems Building Blocks for Iterative Methods, SIAM Press, Philadephia, 1994.

Algorithm	Object											
	Uniform				Checker Board				Brain Image			
	$\|\|\|\Delta\|\|\|_{\mathcal{F}}$	$\lambda_{\max}^{M}$	No. of iter.	Mflops	$\|\|\|\Delta\|\|\|_{\mathcal{F}}$	$\lambda_{\max}^{M}$	No. of iter.	Mflops	$\|\|\|\Delta\|\|\|_{\mathcal{F}}$	$\lambda_{\max}^{M}$	No. of iter.	Mflops
$F = F_{EM}$	119.0	.9857	281	50.8	259.9	.9905	455	82.3	18634	.9999	583	105.4
$F = D_{\lambda}$	137.8	.9828	366	65.9	173.9	.9829	327	58.9	665.5	.9971	245	44.2
$F = D_1$	119.0	.9857	281	50.8	159.0	.9857	269	48.6	665.5	.9971	245	44.2
$F = D_2$	111.1	.9840	264	48.8	148.9	.9841	258	47.7	489.3	.9934	226	41.8
$F = D_{50}$	33.9	.9453	56	15.5	45.32	.9452	53	14.7	73.6	.9499	47	13.0
Conjugate Gradient			22	4.1			22	4.1			35	6.58

Table 1: Summary of convergence properties. $|||\Delta|||_{\mathcal{F}} = |||F - F_Y|||_{\mathcal{F}}$, $M = I - F^{-1}F_Y$.

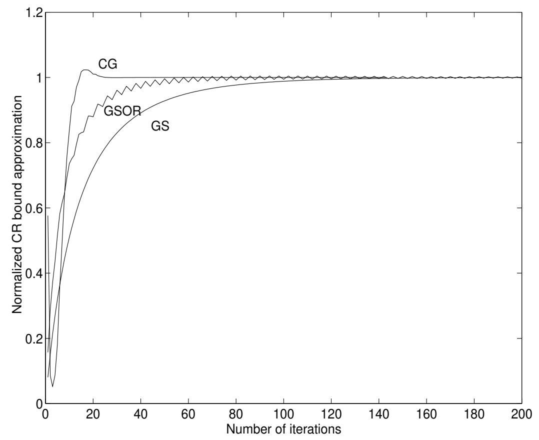

Figure 1: GS with over-relaxation, denoted GSOR, shows oscillatory effect.

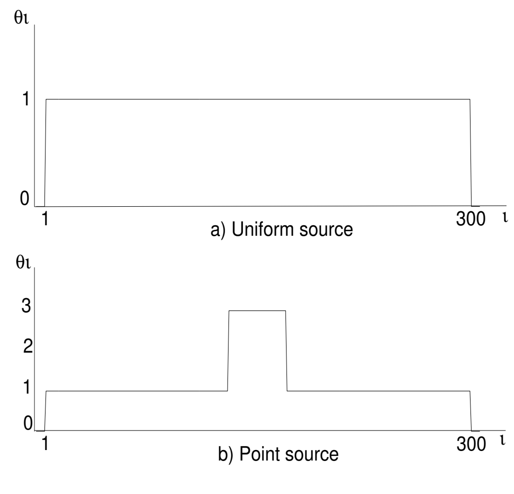

Figure 2: One dimensional sources used in discrete deconvolution problem.

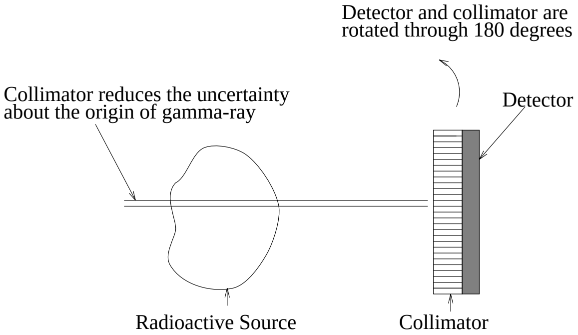

Figure 3: A SPECT system

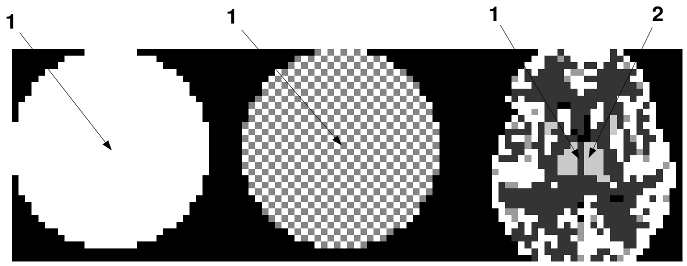

Figure 4: Objects used for numerical comparisons. The pixels of interest are the center pixel denoted by **1** and a neighboring pixel denoted by **2** in the brain image. The intensity ranges from 0 (black) to 2 (white). Objects are: Left: uniform, Middle: Checker board and, Right: Brain image.

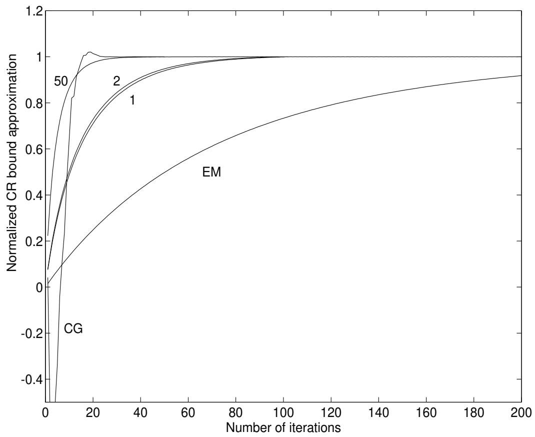

Figure 5: Effect of changing the number of diagonals for GS implemented with p-diagonal F for the low intensity pixel **1** for the brain phantom Figure 4. Numbers labeling each curve indicate the number of diagonals utilized in the splitting matrix D_p.

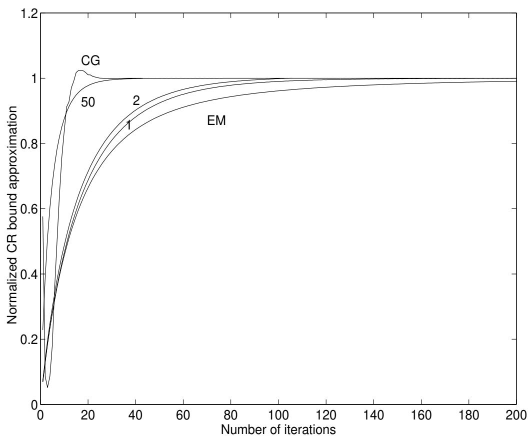

Figure 6: Same as Figure 5, except the pixel of interest was the high intensity pixel **2**.

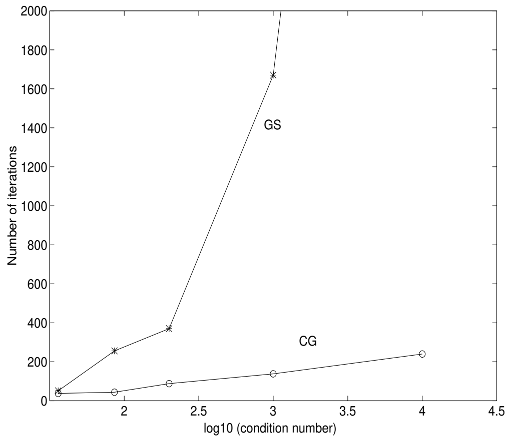

Figure 7: Effect of the condition number on the convergence rate for the GS and the CG algorithms for the uniform source. Matrices of dimension 300×300 were used.

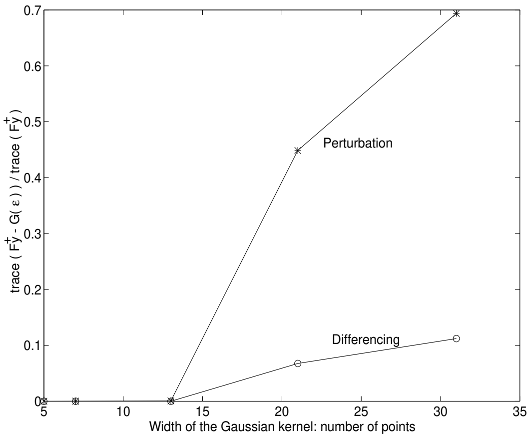

Figure 8: Accuracy of the perturbation method and differencing method for a singular FIM F_Y for the point source.

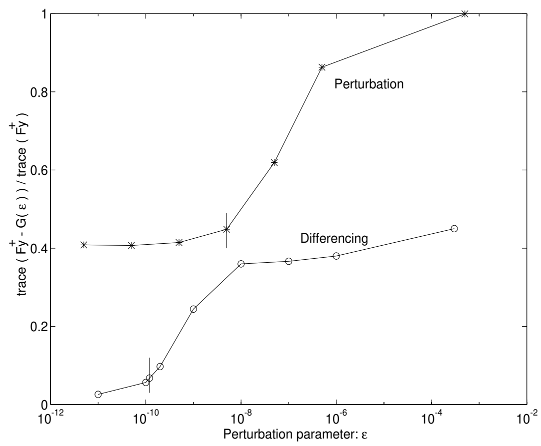

Figure 9: Performance comparison of the perturbation method and differencing method for varying ϵ for the point source. Vertical bar on both the plots corresponds to the choice of ϵ using (27) and (33).

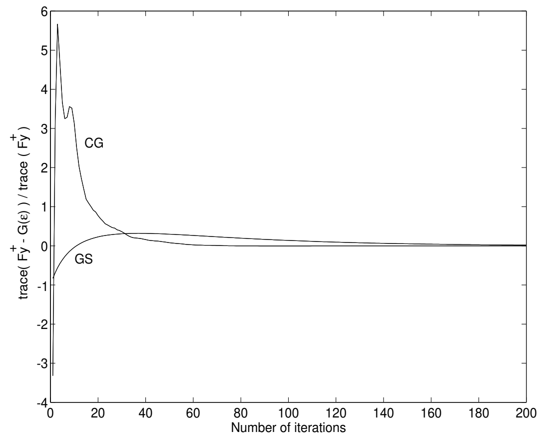

Figure 10: Differencing method has non-monotone convergence for both the GS and the CG.

List of Figures

List of Tables