New Algorithms for Solving Large Sparse Systems of Linear Equations and their Application to Nonlinear Optimization*

H. D. Scolnik

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires, (1426) Buenos Aires, Argentina
hugo@nobi.uba.ar

Abstract

Given a system of equations $Ax = b$, where A is a full rank $n \times n$ matrix, usually large and sparse, a starting point x^0, and a direction vector p, a new result is presented which allow us to find the closest point of the form $x^0 + \alpha p$ to the unknown solution x^* in the metric defined by $A^T A$. From that result several algorithms had been developed, initially for symmetric positive definite matrices. The basic idea is to optimize free parameters determining the search directions for approximating Newton's method. The main algorithm is of the Cimmino's type, where the directions given by the projections onto the hyperplanes are combined in an optimal way. The version for positive definite systems is being applied to large scale nonlinear optimization problems. Finally, a couple of numerical experiences are presented.

Keywords: Sparse systems, projection methods, nonlinear optimization.

1 Introduction

Given a linear system $Ax = b$, where $A \in \Re^{n \times n}$ is large, sparse and nonsingular, and a point $x^0 \in \Re^n$, we define the residual vector $r(x^0) = Ax^0 - b$. Let us consider the quadratic function $f(x) = \frac{1}{2}x^T Ax - x^T b$ and its gradient $g(x) = \frac{1}{2}(A + A^T)x - b$. If $A = A^T$, then $g(x^0) = r(x^0)$.

The leading iterative methods for solving systems of linear equations are:

*Work supported by University of Buenos Aires(Project EX.146), Argentina.
Received december 1996, revised version accepted june 1997.

For positive definite symmetric systems

The Conjugate Gradients Algorithm, is in general a very effective method.

However, since the number of required iterations is theoretically given by the number of distinct eigenvalues, the usual approach is to compute a matrix M (called the *preconditioner matrix*) such that the system $MAx = Mb$ is solved, where MA has as many equal eigenvalues as possible. Unfortunately, no universal method exists for computing M.

If the condition number $\kappa(A)$ is close to one, convergence is very fast However, large condition numbers severely affect performance. See [6].

For nonsymmetric systems of equations

Many parameter free iterative methods have been proposed. The main ones are:

1. **CGN.** The Conjugate Gradients Method applied to the normal equations.

   The obvious shortcoming comes from the fact that $\kappa(A^TA) = [\kappa(A)]^2$.

2. **GCR.** (Generalized Conjugate Residuals), Orthomin, Orthodir, which, like CG, generate a Krylov subspace K using only matrix-vector products and enforce some minimizations or orthogonality property on K. They differ primarily in how the basis of K is formed and which inner product is used to define orthogonality or minimality. Elman proved in [3] that restarted versions of these algorithms converge provided that $\frac{A+A^T}{2}$ is positive definite. Obviously this condition not always holds as shown by a matrix
$$A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

   GMRES was developed for overcoming this restriction and it never breaks down. However, the required storage grows with the number of iterations and therefore all these methods are often used in a truncated or restarted form.

   GMRES(k) constructs a sequence $\{x^i\}$ such that
$$x^k \in x^0 + [r^0, Ar^0, \ldots, A^{k-1}r^0]$$
   satisfying that $\|r^k\|$ is minimum. More precisely we have that
$$x^k = x^0 + z^k, \text{ where } z^k = V_k y^k, \; y^k = H_k^{-1}\|r^0\|e_1$$
   and $H_k = V_k^T A V_k$ and $V_k = (\, v^1 \quad \ldots \quad v^k \,)$, where the vectors v^i are obtained by an Arnoldi construction of an orthonormal basis for the Krylov space.

3. **Matrix Splitting Methods.** They include the earliest iterative methods and are based on splitting the coefficient matrix as $A = M - N$. This category includes Jacobi, Gauss-Seidel, and SOR's. Convergence is assured if the spectral radius $\rho(M^{-1}N) < 1$. See [7].

The three categories are restricted in applicability. In general, most nonsymmetric solvers either require storage and computation that grow excessively with iteration number, special properties of the spectrum of A to guarantee convergence, or a symmetrization proccess with potentially disastrous effects on the system.

One group of methods that avoids these difficulties is that of row projection algorithms. The idea goes back to S. Kaczmarz [8] and G. Cimmino [2] who proposed iterative algorithms for solving a linear system of equations by cyclically projecting on the hyperplane defined by one equation. More generally, partition $A \in \Re^{n\times n}$ into m block rows $A^T = (A_1, \ldots, A_m)$ and b accordingly. A row projection algorithm requires the computation of the orthogonal projections $P_i x = A_i(A_i^T A_i)^{-1}A_i^T x$ of a vector x onto Range(A_i), $i = 1, \ldots, n$. Note that the nonsingularity of A implies that each A_i has full rank.

The simplest method of this sort can be derived geometrically. Let

$$H_i = \{x : A_i^T x = b_i\}$$

be the affine set of solutions to the ith block row of equations. The solution x^* to $Ax = b$ is the unique intersection point of those affine sets, and the method of succesive projections gives the iterations

$$x^{k+1} = (I - P_m)(I - P_{m-1})\ldots(I - P_1)x_k + b_u,$$

where

$$b_u = b_m + (I - P_m)b_{m-1} + \ldots + (I - P_m)\ldots(I - P_2)b_1,$$

with

$$\bar{b}_i = A_i(A_i^T A_i)^{-1}b_i.$$

When each block row consists of a single row of A we get Kaczmarz's original algorithm. See [1].

In the Cimmino's sort of algorithms x^{k+1} is obtained from x^k minimizing along a direction which is a convex combination of the directions given by the projections onto the hyperplanes.

In [4], these ideas are extended for solving the convex feasibility problem. In a particular algorithm based upon Cimmino's method, non-
convex combinations appear.

Anyway, how to choose a suitable combination of the directions remains an open question. In the following, we present results aiming at giving an answer to this problem.

2 New Results

The key idea of this paper arises from the following result.

Theorem 2.1: Let us consider the linear system $Ax = b$, where $A \in \Re^{n\times n}$ has full rank, and its unique solution x^*. Given a point $x^0 \in \Re^n$ and a direction $p \in \Re^n$, the closest point of the line $x^0 + \lambda p$ to x^* in the metric defined by the symmetric positive definite matrix $A^T A$ is obtained for

$$\lambda = -\frac{< r^0, Ap >}{\|Ap\|^2} \tag{1}$$

Proof. Let us consider the function

$$\begin{aligned} f(\lambda) &= \|x^0 + \lambda p - x^*\|^2_{A^T A} \\ &= < A(x^0 - x^*) + \lambda Ap, A(x^0 - x^*) + \lambda Ap > \\ &= < r^0 + \lambda Ap, r^0 + \lambda Ap > \\ &= \|r^0\|^2 + 2\lambda < r^0, Ap > + \lambda^2 \|Ap\|^2, \end{aligned}$$

then, from $f'(\lambda) = 0$, it follows that $\lambda^* = -\frac{<r^0,Ap>}{\|Ap\|^2}$.

Note that $f''(\lambda) > 0$. Moreover,

$$\begin{aligned} f(\lambda^*) &= ||r^0||^2(1 - \cos^2(r^0, Ap)) \\ &= ||r^0||^2 sin^2(r^0, Ap). \end{aligned} \tag{2}$$

Lemma 2.1: The merit function (2) is zero if and only if p coincides with the optimal direction $x^0 - x^*$.

Proof. Elementary.

Suppose now we have two directions p_1 and p_2. We want to find α in such a way that $p(\alpha) = \alpha p_1 + (1 - \alpha)p_2$ minimizes the merit function (2). This result is given in the following

Theorem 2.2: Given the system $Ax = b$, where $A \in \Re^{m\times n}$, $m \geq n$, is a full rank matrix, and two directions $p_1, p_2 \in \Re^n$, then the value of α such that $p(\alpha) = \alpha p_1 + (1 - \alpha)p_2$ which minimizes (2) is

$$\alpha = \frac{< A(p_1 - p_2), < g^0, Ap_2 > Ap_2 - \|Ap_2\|^2 g^0 >}{< A(p_1 - p_2), < g^0, Ap_1 > Ap_2 - < g^0, Ap_2 > Ap_1 >}$$

Proof. Let us define

$$H(\alpha) = \frac{< g^0, Ap(\alpha) >^2}{\|Ap(\alpha)\|^2} \equiv \frac{P(\alpha)}{Q(\alpha)}, \tag{3}$$

and we will find its critical points. We have that

$$\begin{aligned} < g^0, Ap(\alpha) > &= < g^0, A(\alpha(p_1 - p_2) + p_2) > \\ &= \alpha < g^0, A(p_1 - p_2) > + < g^0, Ap_2 > . \end{aligned}$$

Then, defining

$$\begin{aligned} a_1 &= < g^0, A(p_1 - p_2) >^2, \\ a_2 &= 2 < g^0, A(p_1 - p_2) >< g^0, Ap_2 >, \\ a_3 &= < g^0, Ap_2 >^2, \end{aligned} \tag{4}$$

we get

$$P(\alpha) =< g^0, Ap(\alpha) >^2= a_1\alpha^2 + a_2\alpha + a_3.$$

Also

$$\begin{aligned} \|Ap(\alpha)\|^2 &= < Ap(\alpha), Ap(\alpha) > \\ &= < \alpha A(p_1 - p_2) + Ap_2, \alpha A(p_1 - p_2) + Ap_2 > \\ &= \|A(p_1 - p_2)\|^2\alpha^2 + 2 < A(p_1 - p_2), Ap_2 > \alpha \\ &\quad + \|Ap_2\|^2. \end{aligned}$$

Therefore, defining

$$\begin{aligned} b_1 &= \|A(p_1 - p_2)\|^2 \\ b_2 &= 2 < A(p_1 - p_2), Ap_2 > \\ b_3 &= \|Ap_2\|^2, \end{aligned} \tag{5}$$

we can write

$$\begin{aligned} Q(\alpha) &= \|Ap(\alpha)\|^2 \\ &= b_1\alpha^2 + b_2\alpha + b_3. \end{aligned}$$

From (3) it follows that

$$H'(\alpha) = \frac{P'(\alpha)Q(\alpha) - P(\alpha)Q'(\alpha)}{Q^2(\alpha)}. \tag{6}$$

Also, we can write

$$P(\alpha) = (< g^0, A(p_1 - p_2) > \alpha + < g^0, Ap_2 >)^2,$$

which can be written as

$$P(\alpha) = (c_1\alpha + c_2)^2, \tag{7}$$

with $a_1 = c_1^2$, $a_2 = 2c_1c_2$, $a_3 = c_2^2$. Thus, from (7) we get

$$P'(\alpha)Q(\alpha) - P(\alpha)Q'(\alpha) =$$

$$= (c_1\alpha + c_2)[(b_2c_1 - 2b_1c_2)\alpha + (2c_1b_3 - c_2b_2)]. \tag{8}$$

Therefore, from (7) and (8) we obtain that one root is $\alpha_1 = -c_2/c_1$, but for this value is $p(\alpha_1) = 0$. In other words, α_1 is a maximizer of the merit function

$$F(\alpha) = \|r^0\|^2(1 - \cos^2(r^0, Ap(\alpha)). \tag{9}$$

The remaining root is

$$\alpha_2 = \frac{c_2b_2 - 2c_1b_3}{b_2c_1 - 2b_1c_2},$$

which after some straightforward calculations can be written as

$$\alpha_2 = \frac{< A(p_1 - p_2), < g^0, Ap_2 > Ap_2 - \|Ap_2\|^2 g^0 >}{< A(p_1 - p_2), < g^0, Ap_1 > Ap_2 - < g^0, Ap_2 > Ap_1 >}$$

3 Algorithms

Given $A \in \Re^{m\times n}$, $1 \le p \le m$

$$\begin{pmatrix} A_1 \\ \vdots \\ A_p \end{pmatrix} x = \begin{pmatrix} b_1 \\ \vdots \\ b_p \end{pmatrix}$$

Algorithm (Block Cimmino Method)

Choose x^0, set $k = 0$.

Repeat until convergence
begin
do in parallel $i = 1, \ldots p$
$\delta_i^k = A_i^+ b_i - P_{\mathrm{R}(A_i^T)} x^k$
$= A_i^T (b_i - A_i x^k)$
end parallel.
$x^{k+1} = x^k + \sum_{i=1}^p w_i \delta_i^k, \; / \; \sum_{i=1}^p w_i = 1.$
set $k = k + 1$
end.

If $p = m$ we get the original Cimmino's Algorithm.

One possible Cimmino's type (Algorithm 1)

Choose x^0, $\epsilon_1, \epsilon_2, \epsilon_3$, **set** $k = 0$, $r^0 = Ax^0 - b$.
Repeat until convergence
begin
do in parallel $i = 1, \ldots n$
$p_1 = -r^k$
$p_2 = a_i^T$
if $|\cos(p_1, p_2)| < \epsilon_2$ **skip** a_i (if all a_i are rejected, $p^k = -r^k$)
compute α_i **which minimizes**
$1 - \cos^2(r^k, A(\alpha p_1 + (1 - \alpha p_2))$
if $F(\alpha) < \epsilon_3$ **proceed to line search**
end parallel

choose j **such that** $F(\alpha_j) = \min_i F(\alpha_i)$
$p^k = \alpha_j(-r^k) + (1 - \alpha_j) a_j^T$

Line search procedure
if $A = A^T > 0$ **then**
$\lambda^k = -\frac{<r^k, p>}{<p^k, Ap^k>}$
else
$\lambda^k = -\frac{<Ap^k, r^k>}{\|Ap^k\|^2}$
end if
end

$x^{k+1} = x^k + \lambda^k p^k$
$r^{k+1} = Ax^{k+1} - b$
if $\|r^{k+1} < \epsilon_1 \|r^0\|$ **stop**

set $k = k + 1$
end

Now, let us prove that Algorithm 1 is globally convergent for symmetric matrices.

Theorem 3.1: Given the system $Ax = b$, $A \in \Re^{n\times n}$, $A = A^T > 0$, then for any starting point $x^0 \in \Re^n$, Algorithm 1 is globally convergent.

Proof. In the following, $\| \, . \, \|$ will denote the l_2 norm. Let us consider the function

$$f(x) = \frac{1}{2} x^T A x - b^T x \tag{10}$$

and its gradient

$$g^k = g(x^k) = Ax^k - b \tag{11}$$

which is Lipschitz continuous because

$$\|g(x) - g(y)\| = \|A(x-y)\| \leq \sigma_1 \|x - y\|, \tag{12}$$

where $\sigma_1 = \|A\|$ is the largest singular value of A.

We define

$$\cos\theta^k = -\frac{< g^k, p^k >}{\|g^k\| \|p^k\|} \geq \epsilon > 0$$

because of the way directions are chosen in Algorithm 1.

Due to the exact line search, we have that

$$< g(x^k) + \lambda^k p^k, p^k > = < g^{k+1}, p^k > = 0 \tag{13}$$

Hence, from (12) we get

$$< g^{k+1} - g^k, p^k > = - < g^k, p^k > = \lambda^k < Ap^k, p^k > \tag{14}$$

On the other hand, from (12)

$$< g^{k+1} - g^k, p^k > \leq \sigma_1 \lambda^k \|p^k\|^2 \tag{15}$$

Then, using (14) and (15) we obtain

$$\lambda^k \geq -\frac{< g^k, p^k >}{\sigma_1 \|p^k\|^2} \tag{16}$$

Also

$$\begin{aligned} f(x^{k+1}) &= f(x^k) + \lambda^k < g^k, p^k > + \frac{(\lambda^k)^2}{2} < p^k, Ap^k > \\ &= f(x^k) - \lambda^k < g^k, p^k > - \frac{(\lambda^k)^2}{2} < p^k, Ap^k > \end{aligned}$$

because of (14).

Defining

$$c = \frac{\|g^k\|^2 < p^k, Ap^k >}{2\sigma_1^2}$$

We finally get

$$f(x^{k+1}) \leq f(x^k) + c\|g^k\|^2 \cos^2 \theta^k \qquad (17)$$

Since $f(x)$ is bounded below, it follows from (17) that

$$\sum_{k=1}^{\infty} \|g^k\|^2 \cos^2 \theta^k < \infty \quad \text{(The Zoutendijk condition [9])}$$

Taking into account that $\cos^2 \theta^k \geq \epsilon_2^0 > 0$, we conclude that

$$\lim_{k \to \infty} \|g^k\| = 0.$$

4 Numerical Experiments

In order to test Algorithm 1, an experimental Fortran program was written. The following result were obtained using 486 DX2-66 Mhz PC and the Microsoft Fortran 5.1 compiler. We compare Algorithm 1 with PCG (Preconditioned Conjugate Gradients Method as implemented in the IMSL library) and GMRES(n).

MAT1: Let $A_n = (a_{ij})$ be the non symmetric matrix defined by

$$a_{ij} = \begin{cases} 1, & j \geq i \\ a_j, & j < i. \end{cases}$$

The determinant of A_n is given by

$$\det(A_n) = (1 - a_1)(1 - a_2) \ldots (1 - a_{n-1})$$

We chose $n = 50$ and $a_j = t$ for $j = 1 \ldots n-1$.

The system has $x_i^* = 1$, and $x_i^0 = 0.5$ for $j = 1 \ldots n$.

Method	t	CPU	$\|r\|$	$\|x - x^*\|$
New	0.8	0.17	$0.130\,10^{-9}$	$0.991\,10^{-10}$
PCG	0.8	2.09	$0.134\,10^{-9}$	$0.251\,10^{-9}$
GMRES	0.8	0.33	$0.772\,10^{-8}$	$0.672\,10^{-7}$
New	0.9	0.16	$0.501\,10^{-9}$	$0.787\,10^{-9}$
PCG	0.9	1.53	$0.108\,10^{-8}$	$0.130\,10^{-6}$
GMRES	0.9	0.28	$0.765\,10^{-9}$	$0.186\,10^{-6}$

MAT2: Let us consider the Hilbert matrix

$$A_n = (a_{ij}) = (\frac{1}{i+j-1}) \quad i, j = 1, \ldots, n$$

and the linear system $A_n x = b$ for $n = 7$ and the same starting point and solution as before.

The condition number is approximately $e^{3.5n}$, and $\det(A_7)$=$4.8358\,10^{-25}$

Method	CPU	$\|r\|$	$\|x - x^*\|$
New	0.03	$0.533\,10^{-6}$	$0.381\,10^{-2}$
PCG	0.04	$0.307\,10^{-5}$	$0.175\,10^{-2}$
GMRES	0.03	$0.268\,10^{-6}$	$0.746\,10^{-2}$

Remark 4.1: PCG cannot reach lower values of $\|r\|$ because of numerical failure. Note that closer iterates to x^* not necessarily imply lower values of $\|r\|$.

Reference: [5].

Conclusion: The results presented in this paper can be applied to a variety of possible row action algorithms. Here we just compared an experimental program implementing one possible algorithm against two professionally developed products. These first results are very encouraging because they show the new algorithms can deal with ill conditioned symmetric and non symmetric matrices efficiently.

5 Applications to Nonlinear Optimization

Let us consider the unconstrained minimization problem

$$\min_{x \in \Re^n} f(x), \tag{18}$$

and the variable metric methods for solving it:

$$p = -Hg \tag{19}$$

$$x_+ = x + \alpha p, \tag{20}$$

where x is the current estimator of the optimum, α is a steplength and $g \equiv \nabla f(x)$.

The approximation to the inverse of the Hessian H can be updated taking a matrix from the *Broyden family :*

$$H^+ = H - \frac{H\gamma\gamma^T H}{\gamma^T H\gamma} + \frac{\delta\delta^T}{\delta^T\gamma} + \phi\gamma^T H\gamma\left(\frac{\delta}{\delta^T\gamma} - \frac{H\gamma}{\gamma^T H\gamma}\right)\left(\frac{\delta}{\delta^T\gamma} - \frac{H\gamma}{\gamma^T H\gamma}\right)^T,$$

with $\delta \equiv x_+ - x$ and $\gamma \equiv g_+ - g$.

By choosing different values for the parameter ϕ, we obtain different updates whose performances differ greately. (e.g. $\phi = 0$ corresponds to DFP method and $\phi = 1$ to BFGS method).

Let us suppose now that we are at x_+, and that we want to choose the search direction

$$p(\phi) \equiv -H^+ g_+$$

that best resembles (in certain sense) Newton's direction $p_{\mathrm{N}} \equiv -A_+^{-1} g_+$, where $A_+ \equiv \nabla^2 f(x_+)$.

We could try to do this by choosing the direction $p(\phi)$ that minimizes the merit function

$$F(p) = \|g_+\|^2\big(1 - \cos^2(g_+, A_+ p)\big) \geq 0,$$

which attains its least value precisely at the Newton's direction p_{N}.

Lemma 5.1: The value of ϕ which minimizes $F(p(\alpha))$ is

$$\phi = \frac{\alpha_3\alpha_5 - \alpha_1\alpha_4}{\alpha_1\alpha_2\alpha_6 - \alpha_2\alpha_4\alpha_5},$$

where

$$\begin{array}{ll} \alpha_1 \equiv < A_+ g_+, Z g_+ >, & \alpha_2 \equiv a < v, g_+ >, \\ \alpha_3 \equiv \|A_+ Z g_+\|^2, & \alpha_4 \equiv < A_+ Z g_+, A_+ v >, \\ \alpha_5 \equiv < g_+, A_+ v >, & \alpha_6 \equiv \|A_+ v\|^2, \end{array}$$

and

$$H^+ = \overbrace{H - \frac{H\gamma\gamma^T H}{\gamma^T H\gamma} + \frac{\delta\delta^T}{\delta^T\gamma}}^{Z} + \phi \underbrace{\gamma^T H\gamma}_{a} \overbrace{\left(\frac{\delta}{\delta^T\gamma} - \frac{H\gamma}{\gamma^T H\gamma}\right)}^{v}\left(\frac{\delta}{\delta^T\gamma} - \frac{H\gamma}{\gamma^T H\gamma}\right)^T.$$

Another way of trying to approximate Newton's direction in some sense (due to M.F. Marazzi) could be obtained by solving

$$\min_{\phi} \|H^{+}\nabla^2 f(x) - I\| \tag{21}$$

for some matrix norm. If the Hessian of f is not available, we may replace it by the current approximation $B \equiv H^{-1}$, which is in hand, and solve

$$\min_{\phi} \|H^{+}B - I\|, \tag{22}$$

instead of solving (21).

Lemma 5.2: The unique solution to (22) in the weighted Frobenius norm $\| \, . \, \|_H \equiv \|H^{-1/2}(\, . \,)H^{1/2}\|_{\text{F}}$ is

$$\phi = -\frac{\delta^T\gamma}{\gamma^T H\gamma}.$$

Remark 5.1: These results can be used with automatic differentiation.

Acknowledgment: The author acknowledges very insightful discussions with N. Echebest, M.T. Guardarucci, M. Marazzi, C. Ruscitti and M.C. Vacchino.

References

[1] R. Bramley and A. Sameh, "Row Projection Methods for Large Nonsymmetric Linear Systems". *SIAM J. Sci. Sts. Comp.*, 13, 1992, pp.168-193.

[2] G. Cimmino, "Calcolo approssimate per le soluzioni dei sistemi di equazioni lineari". *La Ricerca scientifica ed il Progresso tecnico nell' Economia nazionale (Roma)*, Consiglio Nazionale delle Ricerche, Ministero dell' Educazione nazionale, 9,1, 1938, pp.326-333.

[3] H. Elman, *Iterative methods for Large, Sparse, Nonsymmetric Systems of Linear Equations.* Ph.D. Thesis and Research Report n°229, Dept.of Comp.Sc., Yale University, 1982.

[4] U. García Palomares, "Parallel Projected Aggregation Methods for Solving the Convex feasibility Problem". *SIAM J.Opt.*, 3, 1993, pp.882-900.

[5] R. Gregory and D. Karney, *A Collection of Matrices for Testing Computational Algorithms.* J.Wiley, 1969.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Third Edition, The Johns Hopkins University Press, 1996.

[7] L. Hageman and D. Young, *Applied Iterative Methods.* Academic Press, 1981.

[8] S. Kaczmarz, "Angenaherte Auflosung von Systemen linearer Gleichungen". *Bulletin internationel de l'academie Polonaise des Sciences et des lettres.* Classe des Sciences mathématiques et naturelles. Séries A: Sciences mathématiques.

[9] G. Zoutendijk, "Nonlinear Programming, Computational Methods in Integerand Nonlinear Programming". *J. Abadie*, ed., North Holland (Amsterdam), 1970, pp. 37-86.