A generalized Broyden's method
for solving simultaneous linear equations

Richard R. Gerber
Franklin T. Luk

TR 80-438

Department of Computer Science
Cornell University
Ithaca, New York 14853

September 1980

A generalized Broyden's method
for solving simultaneous linear equations

Short title: A generalized Broyden's method

Richard R. Gerber

and

Franklin T. Luk

Department of Computer Science

Cornell University

Ithaca, NY 14853

Research supported in part by U.S. Army Research Office Grant
DAAG 29-79-C0124.

Abstract

We present a generalized Broyden's method for solving rectangular systems of linear equations. We show that the method computes a least squares solution to the given simultaneous equations and that it possesses a remarkable finite termination property in exact arithmetic.

1. Introduction

Broyden's 1965 method [1] was designed to solve systems of nonlinear equations

$$f(x) = 0,$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$. Some surprising results have been obtained for the special case where f represents a nonsingular system of linear equations, i.e.

$$f(x) \equiv Ax - b,$$

where $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ and A is nonsingular. Gay [2] showed that in this case Broyden's method terminates in at most $2n$ iterations. He also proved that there exist systems of equations for which a full $2n$ iterations are required.

Gay's results provided the motivation for our work. We generalize Broyden's method so that it can be applied to arbitrary rectangular systems of linear equations. In Section 2 of this paper we show that our algorithm terminates if and only if it has computed a least squares solution to the system of equations. In Section 3 we show that the algorithm guarantees finite termination. Section 4 displays some of our numerical results.

2. The Algorithm

Suppose that $f(x)$ represents a rectangular system of linear equations, i.e.

$$(2.1) \quad f(x) \equiv Ax - b,$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. We choose an $n \times m$ matrix H_0 satisfying following the two conditions:

Assumptions

1. $\mathcal{R}(H_0) = \mathcal{R}(A^T)$ and
2. $\mathcal{N}(H_0) = \mathcal{N}(A^T)$.

Here $\mathcal{R}(\cdot)$ and $\mathcal{N}(\cdot)$ denote the range and null space, respectively, of a matrix. We may, for example, choose H_0 as A^T . If A is square and nonsingular, then H_0 may be any nonsingular matrix.

We now propose the following generalized Broyden's method.

Algorithm (Generalized Broyden's method)

Choose an n -vector x_0 and an $n \times m$ matrix H_0 satisfying Assumptions 1 and 2. Compute $s_0 := -H_0 f(x_0)$ and let $k := 0$.

```
while  $s_k \neq 0$  do
begin
   $x_{k+1} := x_k + s_k$ ;
   $y_k := f(x_{k+1}) - f(x_k)$ ;
   $H_{k+1} := H_k + (s_k - H_k y_k) v_k^T$ ,
    where  $v_k$  satisfies properties i), ii) and iii) below;
   $s_{k+1} := -H_{k+1} f(x_{k+1})$ ;
   $k := k+1$ 
end
```

At each iteration the vector v_k is selected to satisfy the following three properties:

- i) $v_k^T y_k = 1$
- ii) $v_k = H_k^T u_k$ for some vector u_k , with
- iii) $u_k^T s_k \neq 0$.

Property i) establishes the inverse secant equation

$$(2.2) \quad H_{k+1} y_k = s_k.$$

Note that all three properties are satisfied when Broyden's "good" update is used, i.e. $v_k = H_k^T s_k / (s_k^T H_k y_k)$ for $s_k^T H_k y_k \neq 0$. The "bad" Broyden update $v_k = y_k / y_k^T y_k$ clearly satisfies property i), however it may not keep ii) and iii) invariant.

Some preliminary definitions are necessary. We make frequent use of the identities

$$(2.3) \quad \mathbf{R}(A) = (\mathbf{N}(A^T))^{\perp}$$

and

$$(2.4) \quad \mathbf{N}(A) = (\mathbf{R}(A^T))^{\perp}.$$

We can thus split x_0 so that

$$(2.5) \quad x_0 = x_0^N + x_0^R$$

where $x_0^N \in \mathbf{N}(A)$ and $x_0^R \in \mathbf{R}(A^T)$. Similarly,

$$(2.6) \quad b = b^N + b^R$$

where $b^N \in \mathbb{R}(A^T)$ and $b^R \in \mathbb{R}(A)$. Note that if $b^N \neq 0$, then the system (2.1) does not have a root.

Now, we define

$$(2.7) \quad f_k \equiv f(x_k) = Ax_k - b$$

and

$$(2.8) \quad g_k \equiv Ax_k - b^R = f_k + b^N.$$

First we show that $y_k \neq 0$ throughout the execution of the algorithm. From the equation

$$(2.9) \quad H_{k+1} = H_k(I - f_{k+1} v_k^T),$$

we see that

$$(2.10) \quad \mathbb{R}(H_{k+1}) \subseteq \mathbb{R}(H_k).$$

Assumption 1 now gives

$$(2.11) \quad \mathbb{R}(H_k) \subseteq \mathbb{R}(A^T)$$

for $k = 0, 1, \dots$. As

$$s_k = -H_k f_k \quad \text{and}$$

$$y_k = As_k,$$

we can use (2.11) and (2.4) to deduce

$$(2.12) \quad s_k \neq 0 \quad \Rightarrow \quad y_k \neq 0.$$

Now we will show that $s_k = 0$ if and only if $A^T f_k = 0$. By property ii) for v_k equation (2.9) can be rewritten as

$$(2.13) \quad H_{k+1} = [I - H_k f_{k+1} u_k^T] H_k.$$

But the matrix inside the brackets is nonsingular because

$$\begin{aligned} \det [I - H_k f_{k+1} u_k^T] &= 1 - u_k^T H_k f_{k+1} \\ &= 1 - u_k^T H_k f_{k+1} / (u_k^T H_k y_k) \\ &= -u_k^T H_k f_k \\ &= u_k^T s_k \neq 0 \quad \text{by property iii).} \end{aligned}$$

Hence,

$$\mathbb{N}(H_{k+1}) = \mathbb{N}(H_k).$$

From Assumption 2, we get that

$$(2.14) \quad \mathbb{N}(H_k) = \mathbb{N}(A^T)$$

for $k = 0, 1, \dots$. It follows that

$$(2.15) \quad s_k = -H_k f_k = 0 \quad \Leftrightarrow \quad A^T f_k = 0,$$

i.e. x_k is a least squares solution to the simultaneous linear equations

$$(2.16) \quad Ax = b.$$

Now assume that x_k satisfies $A^T f_k = 0$. From

$$x_k = x_0 - H_0 f_0 - H_1 f_1 - \dots - H_{k-1} f_{k-1}$$

and relation (2.11), we get that

$$x_k = x_0^N + w,$$

where $w \in \mathbb{R}(A^T)$. It follows that

$$(2.17) \quad x_k = x_0^N + A^+ b,$$

where M^+ denotes the pseudo-inverse of a given matrix M . Therefore, x_k would be the minimum norm least squares solution to (2.16) if x_0 were the null vector.

3. Finite Termination

As mentioned in Section 1, Gay [2] has shown that Broyden's method, when applied to a nonsingular system of linear equations, will terminate in at most $2n$ steps. Here we show how the exact number of iterations can be determined for our generalized Broyden's method. Our results therefore include Gay's as a special case.

Our ensuing work uses the fact established by (2.12) that y_k is a nonzero vector throughout the execution of the generalized Broyden's method. We also need the relations

$$(3.1) \quad y_k = -AH_k f_k = -AH_k g_k$$

and

$$(3.2) \quad (I - AH_{k+1})y_k = 0.$$

Now, let us define

$$(3.3) \quad Z_k \equiv \text{Sp}\{g_k, AH_k g_k, (AH_k)^2 g_k, \dots\}$$

for $k = 0, 1, \dots$; i.e., Z_k is the subspace spanned by the Krylov sequence $g_k, AH_k g_k, (AH_k)^2 g_k, \dots$. In Section 2 we have proved that our algorithm terminates at the k^{th} step if and only if $A^T f_k = 0$, i.e. $g_k = 0$. Thus

$$(3.4) \quad s_k = 0 \quad \Leftrightarrow \quad \dim(Z_k) = 0.$$

We proceed to show how the Z_k 's decrease in dimension. In so doing we need several lemmas, the first of which is due to Powell [4].

Lemma 3.1 Let z_{k+1} be any vector in Z_{k+1} . Then there exists a vector z_k in Z_k such that

$$z_{k+1} = (I - AH_k)z_k.$$

Proof It suffices to show that for $j = 0, 1, \dots$, there is a vector t_j in Z_k such that

$$(AH_{k+1})^j g_{k+1} = (I - AH_k)t_j.$$

We prove this by induction. For the case where $j = 0$, we have $t_0 = g_k$ because

$$\begin{aligned} g_{k+1} &= f_{k+1} + b^N \\ &= (I - AH_k)f_k + b^N \\ &= (I - AH_k)f_k + (I - AH_k)b^N \\ &= (I - AH_k)g_k. \end{aligned}$$

Now, assume there is a vector t_j in Z_k such that

$$(AH_{k+1})^j g_{k+1} = (I - AH_k)t_j.$$

By the definition of H_{k+1} , we get

$$\begin{aligned} (AH_{k+1})^{j+1} g_{k+1} &= (AH_{k+1})(AH_{k+1})^j g_{k+1} \\ &= [AH_k + A(s_k - H_k y_k)v_k^T](I - AH_k)t_j \\ &= (I - AH_k)AH_k t_j + c(I - AH_k)y_k, \end{aligned}$$

where $c = v_k^T(I - AH_k)t_j$.

So, $(AH_{k+1})^{j+1} g_{k+1} = (I - AH_k)t_{j+1}$, where $t_{j+1} = AH_k t_j + cy_k$. And by (3.1) $t_{j+1} \in Z_k$. $\square$

We immediately see that

$$(3.5) \quad \mathbf{Z}_{k+1} \subset \mathbf{Z}_k.$$

Another straightforward implication of Lemma 3.1 is stated in the following lemma.

Lemma 3.2 Let the vectors

$$t_1, t_2, \dots, t_d$$

span $\mathbf{Z}_k$. Then the vectors

$$(I-AH_k)t_1, (I-AH_k)t_2, \dots, (I-AH_k)t_d$$

span $\mathbf{Z}_{k+1}$.

Lemma 3.3 Let $\dim(\mathbf{Z}_k) = d+1$. If there is a nonzero vector w_k in $\mathbf{Z}_k \cap \mathbb{N}(I-AH_k)$ then

$$w_k = \sum_{i=0}^d \alpha_i (AH_k)^i g_k,$$

with $\alpha_d \neq 0$.

Proof We have

$$\begin{aligned} (I-AH_k)w_k &= \alpha_0 g_k + \sum_{i=1}^d (\alpha_i - \alpha_{i-1})(AH_k)^i g_k - \alpha_d (AH_k)^{d+1} g_k \\ &= 0. \end{aligned}$$

Suppose that $\alpha_d = 0$. As the vectors $(AH_k)^i g_k$ for $i = 0, 1, \dots, d$ are linearly independent, we deduce that $\alpha_i = 0$ for $i = 0, 1, \dots, d-1$, which contradicts the assumption of a nonzero w_k . $\square$

A consequence of Lemma 3.3 is that w_k is unique up to a scalar multiple, i.e., if there is a nonzero vector w_k in $Z_k \cap \mathbb{N}(I-AH_k)$, then w_k spans $Z_k \cap \mathbb{N}(I-AH_k)$. Thus, with Lemma 3.2 we obtain the inequality

$$(3.6) \quad \dim(Z_k) - 1 \leq \dim(Z_{k+1}) \leq \dim(Z_k).$$

We wish to prove the following theorems.

Theorem 3.1 If $\dim(Z_{k+1}) = \dim(Z_k)$ then $\dim(Z_{k+2}) = \dim(Z_{k+1}) - 1$.

Theorem 3.2 If $\dim(Z_{k+1}) = \dim(Z_k) - 1$, and if $v_k^T w_k \neq 0$ where w_k spans $\mathbb{N}(I-AH_k) \cap Z_k$, then $\dim(Z_{k+2}) = \dim(Z_{k+1})$.

A few remarks are in order on the vector w_k in Theorem 3.2. Since $\dim(Z_{k+1}) = \dim(Z_k) - 1$, Lemma 3.2 shows that a nonzero w_k must exist. In fact, it can be shown that $w_k = \lambda y_{k-1}$ for some scalar λ , with the exception of the case where there exists a nonzero w_0 .

Theorems 3.1 and 3.2 yield the result of finite termination. Let $\dim(Z_0) = d_0$ and $\dim(Z_1) = d_1$. From (3.6) we see that $d_0 - 1 \leq d_1 \leq d_0$. Applying the two theorems, we conclude that the algorithm must terminate in exactly $d_0 + d_1$ steps.

Proof of Theorem 3.1 Since $\dim(Z_{k+1}) = \dim(Z_k)$, $Z_{k+1} = Z_k$ by (3.5), and so $y_k \in Z_{k+1}$. By (3.1-2) and Lemma 3.3, y_k spans $\mathbb{N}(I-AH_{k+1}) \cap Z_{k+1}$. Applying Lemma 3.2 completes the proof. $\square$

An additional lemma is needed to prove Theorem 3.2.

Lemma 3.4 If there is a nonzero vector w_k in $\mathcal{N}(I-AH_k) \cap \mathcal{Z}_k$ and if $v_k^T w_k \neq 0$, then $\mathcal{N}(I-AH_{k+1}) \cap \mathcal{Z}_{k+1} = \{0\}$.

Proof Suppose there is a nonzero vector w_{k+1} in $\mathcal{N}(I-AH_{k+1}) \cap \mathcal{Z}_{k+1}$. First we show that $w_{k+1} = \lambda y_k$ for some scalar λ . Then we will prove that y_k is not in $\mathcal{Z}_{k+1}$.

Assume $w_{k+1} \neq \lambda y_k$ for all nonzero scalars λ . Since $w_{k+1} \in \mathcal{Z}_k$ and from the equation

$$(3.7) \quad (I-AH_{k+1}) = (I-AH_k)(I-y_k v_k^T),$$

we deduce that

$$(I-y_k v_k^T)w_{k+1} = \alpha w_k$$

for some nonzero scalar α . But then

$$\alpha v_k^T w_k = v_k^T w_{k+1} - v_k^T y_k v_k^T w_{k+1} = 0,$$

contradicting the assumption that $v_k^T w_k \neq 0$. So $w_{k+1} = \lambda y_k$ for some nonzero scalar λ .

However, y_k is not in $\mathcal{Z}_{k+1}$. Let $\dim(\mathcal{Z}_k) = d+1$ where $d \geq 1$. By Lemma 3.3 the set of vectors

$$\{g_k, AH_k g_k, \dots, (AH_k)^{d-1} g_k, w_k\}$$

is a basis for $\mathcal{Z}_k$. Assuming that $y_k \in \mathcal{Z}_{k+1}$, Lemma 3.2 implies that

$$\begin{aligned} y_k &= (I-AH_k) \left[\sum_{i=0}^{d-1} \beta_i (AH_k)^i g_k + \beta_d w_k \right] \\ &= (I-AH_k) \left[\sum_{i=0}^{d-1} \beta_i (AH_k)^i g_k \right]. \end{aligned}$$

So,

$$\beta_0 g_k + (\beta_1 - \beta_0 + 1) A H_k g_k + \dots + (\beta_{d-1} - \beta_{d-2}) (A H_k)^{d-1} g_k + \beta_{d-1} (A H_k)^d g_k = 0,$$

which is impossible as the vectors $(A H_k)^i g_k$, $i = 0, \dots, d$, are linearly independent. So $y_k \notin Z_{k+1}$ and hence $\mathbb{N}(I - A H_{k+1}) \cap Z_{k+1} = \{ 0 \}$. $\square$

Proof of Theorem 3.2 The proof follows directly from Lemmas 3.2 and 3.4. $\square$

4. Test Results

We ran our tests on the IBM 370/168 computer at Cornell University. Our algorithm uses double precision arithmetic, and it stops iterating when $s_k^T s_k < 10^{-16}$. For the choice of v_k we used Broyden's "good" update, i.e. $v_k = H_k^T s_k / (s_k^T H_k s_k)$. In each case our starting values were $x_0 = 0$ and $H_0 = A^T$.

Example 1

We have chosen the overdetermined systems of linear equations from [3]:

$$(4.1) \quad Ax^{(i)} = b^{(i)},$$

for $i = 1, 2, 3$, where

$$A = \begin{bmatrix} 22 & 10 & 2 & 3 & 7 \\ 14 & 7 & 10 & 0 & 8 \\ -1 & 13 & -1 & -11 & 3 \\ -3 & -2 & 13 & -2 & 4 \\ 9 & 8 & 1 & -2 & 4 \\ 9 & 1 & -7 & 5 & -1 \\ 2 & -6 & 6 & 5 & 1 \\ 4 & 5 & 0 & -2 & 2 \end{bmatrix}$$

and

$$(b^{(1)}, b^{(2)}, b^{(3)}) = \begin{bmatrix} -1 & 1 & 0 \\ 2 & -1 & 1 \\ 1 & 10 & 11 \\ 4 & 0 & 4 \\ 0 & -6 & -6 \\ -3 & 6 & 3 \\ 1 & 11 & 12 \\ 0 & -5 & -5 \end{bmatrix}$$

The matrix A is of rank 3 as its singular values are $\sqrt{1248}$, 20, $\sqrt{384}$, 0 and 0. The minimum norm least squares solutions to (4.1) are

$$(x^{(1)}, x^{(2)}, x^{(3)}) = \begin{pmatrix} -\frac{1}{12} & 0 & -\frac{1}{12} \\ 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \\ -\frac{1}{12} & 0 & -\frac{1}{12} \\ \frac{1}{12} & 0 & \frac{1}{12} \end{pmatrix}$$

Our generalized Broyden's method computed these solutions to machine accuracy in 2, 0 and 2 iterations, for $i = 1, 2$ and 3, respectively. It is easy to check that the vectors $b^{(1)}$ and $b^{(2)}$ are left singular vectors of A corresponding to the singular values $\sqrt{384}$ and 0. In addition, $b^{(3)}$ is a sum of $b^{(1)}$ and $b^{(2)}$. For $b^{(4)} = (1, \dots, 1)^T$ and $b^{(5)} = (1, 2, \dots, 8)^T$, our algorithm required 4 and 6 iterations, respectively.

Example 2

We take the matrix A and the vectors $x^{(1)}$ and $x^{(2)}$ from Example 1. Let $B = A^T$, $d^{(1)} = x^{(1)}$ and $d^{(2)} = x^{(2)}$. We construct the underdetermined system

$$(4.2) \quad B y^{(i)} = d^{(i)},$$

for $i = 1$ and 2. Our algorithm computed the minimum norm solutions $y^{(1)} = b^{(1)} / \sqrt{384}$ and $y^{(2)} = 0$ in 2 and 0 iterations, respectively. For the right hand vectors $(1, 1, 1, 1, 1)^T$ and $(1, 2, 3, 4, 5)^T$, 6 iterations were required for convergence.

Example 3

We choose the $n \times n$ upper triangular matrix

$$C = \begin{bmatrix} 1 & . & . & . & 1 \\ & . & & & . \\ & & . & & . \\ 0 & & & . & . \\ & & & & 1 \end{bmatrix}$$

and consider the system

$$(4.3) \quad Cz = p,$$

where $p = (n, n-1, \dots, 1)^T$, so that the exact solution is

$z = e \equiv (1, 1, \dots, 1)^T$. The table below gives the number of iterations required for convergence.

n	# iterations	$5.2\sqrt{n}$
5	10	12
10	18	16
20	26	23
30	30	28
40	33	33
50	36	37
60	40	40

The third column exhibits the interesting phenomenon that the number of iterations is roughly proportional to $\sqrt{n}$. As each step of Algorithm 2 requires $O(mn)$ arithmetic operations, the total work appears to be only $O(mn^{1.5})$. Also note that except for $n = 5$, our algorithm requires fewer than $2n$ iterations to compute a solution to machine accuracy.

The next table gives some details for the case where $n = 60$. The rapid rate of convergence is particularly interesting. Our numbers are accurate to 3 significant digits, and $a \pm b$ represents $a \times 10^{\pm b}$.

k	$\ x_k - e\ _2$	$\ f(x_k)\ _2$
5	1.39 ⁺ 5	1.07 ⁺ 6
10	1.75 ⁺ 4	6.12 ⁺ 4
15	2.36 ⁺ 4	6.13 ⁺ 4
20	2.17 ⁺ 1	4.04 ⁺ 1
25	9.40 ⁻ 2	1.48 ⁻ 1
30	1.06 ⁻ 4	9.79 ⁻ 5
35	1.14 ⁻ 6	7.97 ⁻ 7
40	8.51 ⁻ 9	4.83 ⁻ 9

Example 4

We select a famous nxn matrix

$$C = \begin{bmatrix} 1 & -1 & . & . & . & -1 \\ & . & . & . & . & . \\ & & . & . & . & . \\ & & & . & . & . \\ & & & & . & . \\ \tau & & & & & -1 \\ & & & & & 1 \end{bmatrix}$$

which is upper triangular except for the (n,1) entry $\tau = -2^{-n+2}$. Now consider the simultaneous equations

$$(4.4) \quad Cz = e_n,$$

where $e_n = (0, \dots, 0, 1)^T$. It is not difficult to verify that the matrix C is of rank n-1 and that the system (4.4) is inconsistent.

The two tables here are similar to those in Example 3.

n	# iterations	$5.9\sqrt{n}$
5	8	13
10	17	19
20	25	26
30	31	32
40	36	37
50	42	42
60	47	46

$n = 60$

k	$\ x_k - x_{47}\ _2$	$\ A^T f(x_k)\ _2$
5	$1.24 \cdot 10^1$	$9.62 \cdot 10^2$
10	$3.52 \cdot 10^1$	$9.40 \cdot 10^2$
15	$1.32 \cdot 10^1$	$2.32 \cdot 10^2$
20	$1.13 \cdot 10^1$	$1.66 \cdot 10^2$
25	$2.42 \cdot 10^{-1}$	$3.57 \cdot 10^0$
30	$3.76 \cdot 10^{-2}$	$3.77 \cdot 10^{-1}$
35	$1.70 \cdot 10^{-4}$	$1.16 \cdot 10^{-3}$
40	$8.39 \cdot 10^{-6}$	$3.41 \cdot 10^{-5}$
45	$1.34 \cdot 10^{-8}$	$3.71 \cdot 10^{-8}$
47	0	$4.16 \cdot 10^{-9}$

References

- [1] Broyden, C.G., "A class of methods for solving nonlinear simultaneous equations," Math. Comput. 19 (1965), 577-593.
- [2] Gay, D.M., "Some convergence properties of Broyden's method," SIAM J. Numer. Anal. 16 (1979), 623-630.
- [3] Golub, G.H., and Reinsch, C., "Singular value decomposition and least squares solutions," Numer. Math. 14 (1970), 403-420.
- [4] Powell, M.J.D., "A new proof of Gay's (1977) theorem on the convergence of Broyden's method," unpublished manuscript.

Acknowledgements

The authors would like to acknowledge their helpful discussions with Professors John E. Dennis, David M. Gay and Charles F. Van Loan.