Numer. Math. 17, 203–214 (1971)

Projection Method for Solving a Singular System of Linear Equations and its Applications

Kunio Tanabe

Received October 24, 1969

Summary. The iterative method for solving system of linear equations, due to Kaczmarz [2], is investigated. It is shown that the method works well for both singular and non-singular systems and it determines the affine space formed by the solutions if they exist. The method also provides an iterative procedure for computing a generalized inverse of a matrix.

1. Introduction

In this paper solutions of the linear system

$$Ax = b \tag{1}$$

are considered, where $A = (a_{ij})$ is an $m \times n$ complex matrix (possibly $m > n$) and x and b are n- and m-dimensional complex column vectors respectively. Given a matrix A and a vector b, the solution may or may not exist. Even when m is larger than n, the system may have a solution. Then the problem is to tell whether the system has a solution or not and also to find all solution vectors if they exist.

It is well known that when the matrix A is square and nonsingular, the Gaussian elimination process with iterative improvement will give an accurate answer unless the matrix is too ill conditioned. But when the matrix is sparse and of large order, this method may be less efficient than the iterative methods like Gauss-Seidel and Jacobi processes. These iterative methods do not always converge. In any case, efficient algorithms have seldom been reported for singular cases.

However, the projection method devised by Kaczmarz can be proved to converge for any system of linear equations with nonzero rows, even when it is singular and inconsistent and the arithmetic operations required in an iteration of the method are comparatively few.

In the next section we give a formulation of this algorithm, and then the properties of the method are discussed including a computational process for the generalized inverse of a matrix in §3. In §4 some numerical examples are shown including non-singular cases.

2. Algorithm

The linear space of all column vectors of n complex elements will be denoted by $\mathscr{C}^n$, and $\mathscr{S}^\perp$ will denote the orthogonal complement of a linear subspace $\mathscr{S}$

in $\mathscr{C}^n$. The symbols $\mathscr{Im}\, A$ and $\mathscr{Ker}\, A$ will denote the range and null spaces of a mapping (matrix) A, respectively. A^* will denote the conjugate transpose of a matrix A. The letter I will denote the identity matrix of whatever size appropriate to the context. $P_{\mathscr{S}}$ will denote the orthogonal projection onto a linear subspace $\mathscr{S}$ in $\mathscr{C}^n$. $A|_{\mathscr{S}}$ will denote the restriction of the linear mapping A to a subspace $\mathscr{S}$. A matrix will always be identified with the linear mapping that it represents. The complex conjugate of x will be denoted by $\bar{x}$. And $\langle x, y\rangle_m$ and $\|x\|_m$ will denote the inner product of m-dimensional vectors x, y and the norm of an m-dimensional vector x, respectively, defined as

$$\langle x, y\rangle = y^* x \quad \text{and} \quad \|x\|_m = \sqrt{\langle x, x\rangle_m}.$$

The suffix m will be omitted when it is clear from the context. The norm $\|A\|$ of an $m\times n$ matrix is defined as

$$\|A\| = \sup_{x\in\mathscr{C}^n} \frac{\|A\,x\|_m}{\|x\|_n}.$$

The system (1) can be equivalently expressed in the form

$$\langle x, a_i\rangle = b_i \quad \text{for} \quad i=1, \ldots, m, \tag{2}$$

where a_i is the i-th column vector of A^*. Assume $\|a_i\|>0$, for all i. Let the mapping f_i from $\mathscr{C}^n$ to $\mathscr{C}^n$ be defined as

$$f_i(x) = x - \frac{\langle x, a_i\rangle - b_i}{\alpha_i}\, a_i \quad \text{for} \quad i=1, \ldots, m, \tag{3}$$

where $\alpha_i = \langle a_i, a_i\rangle$, and let the mapping F from $\mathscr{C}^{m+n}$ to $\mathscr{C}^n$ be defined as

$$\begin{aligned} F(b, x) &= f_1\circ f_2\circ f_3\circ \cdots \circ f_m(x) \\ &= f_1\big(f_2(f_3(\cdots(f_m(x))\ldots))\big). \end{aligned} \tag{4}$$

Now we introduce the algorithm.

Algorithm. Choose an arbitrary element $x^0\in\mathscr{C}^n$, and determine the sequence $\{x^i\}$ from the recurrence relation

$$x^{i+1} = F(b, x^i) \qquad i=0, 1, 2, \ldots. \tag{5}$$

Formula (5) has a simple geometrical interpretation. By the mapping f_i a point $x\in\mathscr{C}^n$ is projected on the hyperplane defined as $\langle x, a_i\rangle = b_i$. Then the point $F(b, x)$ is obtained from x, being projected successively on the hyperplanes (2) in the order $i=m, \ldots, 1$. Thus this cycle forms a single iteration of the algorithm (5).

As is seen in the next section, the sequence generated thus will converge for any A, b and x^0, and if the system is consistent then the limit point is a solution of the system. Thus this cyclic procedure is repeated until the approximations agree to the desired accuracy. Note that the generated sequence is contained in the hyperplane $\langle x_1, a_1\rangle = b_1$.

3. Theoretical Consideration and Applications

We can express (3) in matrix terms in the form

$$f_i(x) = P_i\, x + \frac{b_i}{\alpha_i}\, a_i \quad \text{for} \quad i=1, \ldots, m, \tag{6}$$

where

$$P_i = I - \frac{1}{\alpha_i} a_i a_i^* = \left(\delta_{kl} - \frac{a_{ik}\bar{a}_{il}}{\alpha_i}\right).$$

Note that P_i is an orthogonal projection (self-adjoint and idempotent) matrix and that f_i is an affine transformation.

Let $Q_i = P_1 P_2 \ldots P_i$ $(i = 1, \ldots, m)$, where $Q_0 = I$. And let R be a $n \times m$ matrix whose i-th column vector is $\frac{1}{\alpha_i} Q_{i-1} a_i$, i.e.

$$(7) \qquad R\,b = \sum_{i=1}^{m} \frac{b_i}{\alpha_i} Q_{i-1} a_i.$$

Then we have

$$(8) \qquad F(b, x) = Q\,x + R\,b,$$

where the $n \times n$ matrix $Q = Q_m$ and the $n \times m$ matrix R depend only on the matrix A.

Proposition 1. $Q + RA = I$.

Proof. Since the i-th column vector of R and i-th row vector of A are $\frac{1}{\alpha_i} Q_{i-1} a_i$ and a_i^*, respectively, we have

$$\begin{aligned} RA &= \frac{1}{\alpha_1} a_1 a_1^* + \frac{1}{\alpha_2} Q_1 a_2 a_2^* + \frac{1}{\alpha_3} Q_2 a_3 a_3^* + \cdots + \frac{1}{\alpha_m} Q_{m-1} a_m a_m^* \\ &= (I - P_1) + Q_1 (I - P_2) + \cdots + Q_{m-1}(I - P_m) \\ &= I - Q_m, \end{aligned}$$

because $Q_{i-1} P_i = Q_i$ by the definition.

Thus the algorithm (5) is a stationary linear iterative process of the first degree.

We have

$$\mathscr{C}^n = \mathscr{Ker}\, A \oplus \mathscr{Im}\, A^* \quad \text{and} \quad (\mathscr{Ker}\, A)^\perp = \mathscr{Im}\, A^*.$$

In suffixes only, $\mathscr{Ker}\, A$ and $\mathscr{Im}\, A^*$ will also be denoted by $\mathscr{K}$ and $\mathscr{I}$ respectively. It is easily seen that

$$\mathscr{Ker}\, A = \bigcap_{i=1}^{m} \{x \in \mathscr{C}^n;\ P_i x = x\},$$

$$\mathscr{Im}\, A^* = \left[\bigcup_{i=1}^{m} \{x \in \mathscr{C}^n;\ P_i x = 0\}\right],$$

where $[*]$ denotes the linear subspace of $\mathscr{C}^n$ generated by $*$.

Lemma 2. $\|Q\,x\| = \|x\|$ iff $x \in \mathscr{Ker}\, A$.

Proof. If $x \notin \mathscr{Ker}\, A$, then there exists a number $i_0 \leqq m$ such that $P_{i_0} x \neq x$. Let assume that i_0 is the largest of them. Then we have $\|P_{i_0} P_{i_0+1} \ldots P_m x\| = \|P_{i_0} x\| < \|x\|$. Since every projector P_i has unit norm, we have for $i = 1, \ldots, m$

$$\|Q_i\| = \|P_1 P_2 \ldots P_i\| \leqq \|P_1\| \cdot \|P_2\| \cdot \cdots \cdot \|P_i\| = 1.$$

Thus we have

$$\|Q_m x\| = \|Q_{i_0-1}\| \, \|P_{i_0} P_{i_0+1} \dots P_m x\| < 1 \cdot \|x\|.$$

Conversely, if $x \in \mathscr{K}er\, A$, then

$$P_i x = x \quad \text{for} \quad i = 1, \dots, m;$$

thus we have

$$Q x = P_1 P_2 \dots P_m x = x.$$

This completes the proof.

Corollary 3. $\|Q\| \leqq 1$. If rank $A < n$ then $\|Q\| = 1$.

Corollary 4. $Q x = x$ iff $x \in \mathscr{K}er\, A$.

Theorem 5. 1. $\mathscr{K}er\, A$ and $\mathscr{I}m\, A^*$ are invariant subspaces of the linear mapping Q and further, $Q|_{\mathscr{K}} = I_{\mathscr{K}}$, i.e.

$$Q = P_{\mathscr{K}} \oplus \tilde{Q} \quad \text{and} \quad P_{\mathscr{K}} \tilde{Q} = \tilde{Q} P_{\mathscr{K}} = 0,$$

where $\tilde{Q} = Q P_{\mathscr{I}}$.

2. $\|\tilde{Q}\| = \sup\limits_{\substack{x \in \mathscr{I}m A^* \\ \|x\| = 1}} \|Q x\| < 1$.

Proof. It is easily seen from the proof of the lemma that $Q|_{\mathscr{K}} = I_{\mathscr{K}}$. Similarly we have $Q^*|_{\mathscr{K}} = I_{\mathscr{K}}$ since $Q^* = P_m P_{m-1} \dots P_1$. Then for any $x \in \mathscr{K}er\, A$ and any $y \in \mathscr{I}m\, A^*$ we have

$$\langle x, Q y \rangle = \langle Q^* x, y \rangle = \langle x, y \rangle = 0.$$

Thus we have $Q y \in (\mathscr{K}er\, A)^{\perp} = \mathscr{I}m\, A^*$. Since $P_{\mathscr{K}} + P_{\mathscr{I}} = I$ and $P_{\mathscr{K}} P_{\mathscr{I}} = P_{\mathscr{I}} P_{\mathscr{K}} = 0$, we have the first half of the theorem.

Next it follows from Corollary 3 that

$$\|\tilde{Q}\| \leqq \|Q\| \leqq 1.$$

If $\|\tilde{Q}\| = 1$, then there exists a nonzero vector $x_0 \in \mathscr{I}m\, A^*$ such that $\|Q x_0\| = \|x_0\|$, since $\|Q x\|$ is continuous on the compact set $\{x \in \mathscr{I}m\, A^*;\ \|x\| = 1\}$. Then from Lemma 2 we have $x_0 \in \mathscr{K}er\, A$, thus $x_0 = 0$. This contradicts the hypothesis. Thus we have the desired result.

Note that if rank $A = n$, then $Q = \tilde{Q}$ and $\|Q\| < 1$.

Since $Q = P_{\mathscr{K}} \oplus \tilde{Q}$, we have the following corollary.

Corollary 6. $\lim\limits_{i \to \infty} Q^i = P_{\mathscr{K}}$.

Given nonzero vectors x, $a_1, a_2, \dots, a_m$ (possibly dependent), we are often asked to calculate the component $P_{\mathscr{K}} x$ which is orthogonal to every vector $a_1, a_2, \dots, a_m$. For example, see [6, 7]. An answer is provided by the next corollary.

Corollary 7. For $b = 0$, the algorithm (5)

$$x^{i+1} = F(0, x^i) = Q x^i, \qquad i = 0, 1, 2, 3, \dots \tag{9}$$

generates a sequence $\{x^i\}$ which converges to $P_{\mathscr{K}} x^0$, where x^0 is an initial vector.

Applying the above process (9) to the n initial vectors $e_1=(1, 0, \dots, 0)^t$, $e_2=(0, 1, 0, \dots, 0)^t, \dots, e_n=(0, \dots, 0, 1)^t$, we can also calculate the projection matrix $P_{\mathscr{K}}$, since $P_{\mathscr{K}} e_i$ forms the i-th column vector of $P_{\mathscr{K}}$.

Theorem 8. 1. $\lim_{i\to\infty} \sum_{j=0}^{i} Q^j R$ exists and

$$(I-\tilde{Q})^{-1} R = \sum_{j=0}^{\infty} Q^j R,$$

where R may not be removed as a factor from the series.

2. The $n\times m$ matrix $(I-Q)^{-1} R$, which will be denoted by G, is a generalized inverse of the matrix A such that

$$AGA = A, \quad GAG = G, \quad GA = P_{\mathscr{I}}, \quad AG = P,$$

where P is the projection onto $\mathscr{Im}\, A$ along $\mathscr{Ker}\, R$ and $GA = P_{\mathscr{I}}$ may be replaced by $(GA)^* = GA$.

Proof. For any $x \in \mathscr{Ker}\, A$, we have, for $i=1, \dots, m$

$$\langle Q_{i-1} a_i, x\rangle = \langle a_i, Q_{i-1}^* x\rangle = \langle a_i, x\rangle = 0,$$

since $Q_i^* = P_i P_{i-1} \dots P_1$. Then the column vectors $\frac{1}{\alpha_i} Q_{i-1} a_i$ of the matrix R are in the subspace $\mathscr{Im}\, A^*$. Thus we have

$$\sum_{j=0}^{i} Q^j R = \sum_{j=0}^{i} \tilde{Q}^j R.$$

Since $\|\tilde{Q}\| < 1$, we have

$$\sum_{j=0}^{\infty} \tilde{Q}^j R = \left(\sum_{j=0}^{\infty} \tilde{Q}^j\right) R = (I-\tilde{Q})^{-1} R.$$

Consequently

$$(I-\tilde{Q})^{-1} R = \sum_{j=0}^{\infty} Q^j R.$$

Next, since the column vectors of A^* are in $\mathscr{Im}\, A^*$,

$$(I-Q) A^* = (I-\tilde{Q}) A^*.$$

From Proposition 1, we have $RA = I - Q$. Then

$$RAA^* = (I-\tilde{Q}) A^*,$$

whence $(I-\tilde{Q})^{-1} RAA^* = A^*$. This means that

$$AGA = A \quad \text{and} \quad (GA)^* = GA.$$

We have

$$\begin{aligned} RA(I-\tilde{Q})^{-1} &= (I-Q)(I-\tilde{Q})^{-1} \\ &= (I-\tilde{Q}-P_{\mathscr{K}})(I-\tilde{Q})^{-1} \\ &= I - P_{\mathscr{K}}(I-\tilde{Q})^{-1} \\ &= I - P_{\mathscr{K}}, \end{aligned}$$

since $P_{\mathscr{X}}\tilde{Q}=0$. Then we have

$$RA(I-\tilde{Q})^{-1}R=(I-P_{\mathscr{X}})R=R,$$

since the column vectors of R are in the subspace $\mathscr{I}m\,A^*$.

Premultiplying this equation by $(I-\tilde{Q})^{-1}$, we have

$$(I-\tilde{Q})^{-1}RA(I-\tilde{Q})^{-1}R=(I-\tilde{Q})^{-1}R,$$

that is, $GAG=G$. AG is idempotent and $\mathscr{I}m(AG)\supset\mathscr{I}m\,A$, since $AGA=A$. It is clear that

$$\mathscr{K}er(AG)=\mathscr{K}er(A(I-\tilde{Q})^{-1}R)\supset\mathscr{K}er\,R.$$

Since $\mathscr{I}m(AG)\oplus\mathscr{K}er(AG)=\mathscr{C}^m$, then $\mathscr{I}m\,A\cap\mathscr{K}er\,R=\{0\}$. But $\mathscr{I}m\,A^*=\mathscr{I}m\,R$, because the i-th column vector of R is of the form

$$\frac{1}{\alpha_i}Q_{i-1}a_i=\frac{1}{\alpha_i}\left(a_i-\sum_{j=1}^{i-1}c_j a_j\right).$$

Then

$$\dim(\mathscr{I}m\,A)=\dim(\mathscr{I}m\,R^*),$$

and thus

$$\dim(\mathscr{K}er\,A^*)=\dim(\mathscr{K}er\,R).$$

Consequently we have

$$\mathscr{I}m\,A\oplus\mathscr{K}er\,R=\mathscr{C}^m.$$

This completes the proof.

Now we prove the convergence of the algorithm (5).

Corollary 9. For any $m\times n$ matrix A with nonzero rows and any m-dimensional column vector b, the algorithm (5) generates a convergent sequence of vectors $\{x^i\}$ such that

$$\lim_{i\to\infty}x^i=P_{\mathscr{X}}x_0+Gb,$$

where $x^0\in\mathscr{C}^n$ is an arbitrary initial vector.

Proof. From the relation (5) and (7), we have

$$x^i=Q^i x^0+\left(\sum_{j=0}^{i-1}Q^j R\right)b. \tag{10}$$

It is an immediate consequence of Corollary 6 and the previous theorem that the first and the second terms of (10) converge to $P_{\mathscr{X}}x^0$ and Gb, respectively, as i tends to infinity. Thus we have the desired result.

It is well known that the system (1) has a solution if and only if $AGb=b$. In this case, $P_{\mathscr{X}}x^0+Gb$ is a solution of the system for arbitrary x_0, and Gb is the solution with minimum norm. Then the procedure for solving the system (1) may be the following:

1. Compute Gb by applying the algorithm (5) to the initial vector $x^0=0$.

2. Check the consistency of the system, either computing $A(Gb)$ directly or observing the movement in the last cycle of the iteration. For if the system is

inconsistent, then any vector is necessarily changed by some mapping f_i $(1 \leq i \leq m)$ in one cycle of the iteration.

3. If the system proves to be consistent, then calculate the matrix $P_{\mathscr{X}}$ by applying the process (9) as mentioned before. Thus we determine the affine space $\mathscr{I}m\, P_{\mathscr{X}} + Gb$ formed by the solution vector of the system (1).

We can also compute the generalized inverse G, applying the algorithm (5) to m right hand vectors $b_1 = (1, 0, \ldots, 0)^t$, $b_2 = (0, 1, \ldots, 0)^t, \ldots, b_m = (0, \ldots, 0, 1)^t$ with the same initial vector $x^0 = 0$.

4. Numerical Examples

It is easily seen that if nonzero column vectors a_i are mutually orthogonal then $\tilde{Q} = 0$, and thus a single iteration of the algorithm yields an answer. Hence, when the a_i are mutually "nearly" orthogonal, it is likely that $\|\tilde{Q}\|$ is small and the algorithm converges fast. Our computational experience confirms this.

If we calculate $\alpha_i = \langle a_i, a_i \rangle$, $i = 1, \ldots, m$ in advance (s multiplicative operations are required, where s is the number of nonzero element in A), then a single iteration of the algorithm (5) requires $2s + m$ multiplicative operations. Thus if the matrix A if of large order, this method has an advantage over other ones for both sparse and dense matrices as long as $\|\tilde{Q}\|$ is reasonably small. When $\|\tilde{Q}\|$ is close to 1, the algorithm converges slowly. In this case we need to improve the convergence by such a process as the Aitken's Δ^2-method. But the application of Aitken's process to a singular system will generally disturb the dependence of the limit point on the initial vector that is stated in Corollary 9.

Now we show some numerical examples which were computed in floating arithmetic with a 39-bit mantissa (about 11 decimals).

Problem 1. Find x which satisfies

$$\begin{pmatrix} -3.2 & 2.9 & 1.6 & 0.1 \\ 0.0 & -1.1 & 2.3 & 1.0 \\ 5.1 & 4.8 & 0.2 & 4.9 \\ 2.0 & 1.1 & 1.9 & -2.9 \end{pmatrix} x = \begin{pmatrix} 1.4 \\ 2.2 \\ 15.0 \\ 2.1 \end{pmatrix}.$$

The computer solution is shown in Table 1. The rows of the left-hand-side matrix are mutually "nearly" orthogonal. Hence the iteration converged fast.

Table 1

Iteration	Inner cycle	x_1	x_2	x_3	x_4
Initial		0.0	0.0	0.0	0.0
2	4	1.000 173 4495 − 00	9.999 458 0032 − 01	9.999 395 0672 − 01	1.000 059 4283 − 00
4	1	1.000 000 3410 − 00	1.000 000 4843 − 00	9.999 997 9444 − 01	1.000 000 1557 − 00
	2	1.000 000 3410 − 00	1.000 000 3597 − 00	1.000 000 0551 − 00	1.000 000 2690 − 00
	3	1.000 000 0065 − 00	1.000 000 0448 − 00	1.000 000 0419 − 00	9.999 999 4761 − 01
	4	9.999 999 7237 − 01	1.000 000 0261 − 00	1.000 000 0095 − 00	9.999 999 9708 − 01
6	4	1.000 000 0000 − 00	1.000 000 0000 − 00	1.000 000 0000 − 00	1.000 000 0000 − 00
True solution		1.0	1.0	1.0	1.0

Problem 2. Find the component of x projected on the subspace $[a_1, a_2, a_3]^\perp$, where

$$\begin{aligned} x &= (\ \ 1.0,\ 3.0,\ 5.0,\ -1.0),\\ a_1 &= (\ \ 0.0,\ 5.0,\ 8.0,\ -5.0),\\ a_2 &= (-2.0,\ 0.0,\ 5.0,\ \ \ 2.0),\\ a_3 &= (\ \ 2.0,\ 0.0,\ 4.0,\ -2.0). \end{aligned}$$

The computer solution is shown in Table 2.

Table 2

Iteration	Inner cycle	x_1	x_2	x_3	x_4
Initial		1.0	3.0	5.0	−1.0
6	3	1.055 260 2354−00	9.287 881 0843−01	−1.965 428 9654−02	1.015 951 6561−00
11	3	1.005 704 1078−00	9.926 789 0562−01	−2.043 560 6384−03	1.001 616 9865−00
21	3	1.000 060 1582−00	9.999 227 9204−01	−2.155 434 9972−05	1.000 017 0495−00
31	3	1.000 000 6343−00	9.999 991 8570−01	−2.273 057 8885−07	1.000 000 1795−00
41	3	1.000 000 0066−00	9.999 999 9131−01	−2.395 831 5655−09	1.000 000 0018−00
51	3	9.999 999 9994−01	9.999 999 9978−01	−2.322 824 0254−11	9.999 999 9989−01
61	3	9.999 999 9986−01	9.999 999 9986−01	3.257 868 0723−13	9.999 999 9986−01
True solution		1.0	1.0	0.0	1.0

Problem 3. Find the component of x projected on the subspace $[a_1, a_2, a_3]^\perp$, where

$$\begin{aligned} x &= (\ \ 1.0,\ \ 3.0,\ \ 5.0,\ \ -1.0),\\ a_1 &= (\ \ 0.0,\ \ 3.8,\ 10.4,\ \ -3.8),\\ a_2 &= (-0.6,\ \ 6.6,\ 15.3,\ \ -6.0),\\ a_3 &= (\ \ 1.0,\ 10.5,\ 26.0,\ -11.5). \end{aligned}$$

The vectors a_1, a_2, a_3 are almost of the same direction. The computer solution is shown in Table 3. Since the problem is nearly singular, the convergence is

Table 3

Iteration	Inner cycle	x_1	x_2	x_3	x_4
Initial		1.0	3.0	5.0	−1.0
1	3	1.000 284 3198−00	1.131 623 3244−00	−1.115 102 7444−01	8.680 923 5572−01
6	3	1.001 670 7998−00	1.128 752 9782−00	−1.097 480 9690−01	8.695 762 2188−01
101	1	1.018 715 0467−00	1.086 803 5048−00	−7.027 152 1045−02	8.944 814 4722−01
	2	1.018 943 6340−00	1.084 289 0448−00	−7.610 049 6465−02	8.967 673 1994−01
	3	1.018 821 2397−00	1.083 003 9052−00	−7.928 274 7094−02	8.981 748 5387−01
501	3	1.014 952 3048−00	1.007 103 8867−00	−1.319 959 1744−02	9.779 438 0260−01
True solution		1.0	1.0	0.0	1.0

very slow. Yet at the sixth iteration we have

$$\frac{\langle x^6, a_1\rangle}{\|x^6\|\cdot\|a_1\|} \fallingdotseq 0.00766,$$

$$\frac{\langle x^6, a_2\rangle}{\|x^6\|\cdot\|a_2\|} \fallingdotseq 0.00155,$$

$$\frac{\langle x^6, a_3\rangle}{\|x^6\|\cdot\|a_3\|} \fallingdotseq 0.000\,000\,000\,002\,27.$$

We are often satisfied with such approximations in solving nonlinear programming problems.

Problem 4. Find x which satisfies

$$\begin{pmatrix} 1.0 & 3.0 & 2.0 & -1.0 \\ 1.0 & 2.0 & -1.0 & -2.0 \\ 1.0 & -1.0 & 2.0 & 3.0 \\ 2.0 & 1.0 & 1.0 & 1.0 \\ 5.0 & 5.0 & 4.0 & 1.0 \\ 4.0 & -1.0 & 5.0 & 7.0 \end{pmatrix} x = \begin{pmatrix} 5.0 \\ 0.0 \\ 5.0 \\ 5.0 \\ 15.0 \\ 15.0 \end{pmatrix}.$$

The rank of the left-hand-side matrix is three and the system is consistent, hence x is overdetermined. The computer solution is shown in Table 4.

Table 4

Iteration	Inner cycle	x_1	x_2	x_3	x_4
Initial		7.0	6.0	10.0	6.0
6	6	1.266 846 6149 − 00	9.515 214 9018 − 01	7.206 380 6131 − 01	1.040 134 9606 − 00
21	6	1.002 250 7795 − 00	9.995 910 9678 − 01	9.976 436 5714 − 01	1.000 338 5276 − 00
51	2	1.000 000 1976 − 00	9.999 999 4104 − 01	9.999 998 7093 − 01	1.000 000 1044 − 00
	4	1.000 000 1234 − 00	9.999 999 3510 − 01	9.999 998 0265 − 01	1.000 000 0153 − 00
	6	1.000 000 1602 − 00	9.999 999 7081 − 01	9.999 998 3243 − 01	1.000 000 0240 − 00
72	6	1.000 000 0003 − 00	9.999 999 9983 − 01	9.999 999 9989 − 01	9.999 999 9989 − 01
True solution		1.0	1.0	1.0	1.0

Problem 5. Find the generalized inverse G of the matrix

$$A = \begin{pmatrix} 1.0 & 0.0 & -1.0 & 1.0 \\ 0.0 & 1.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 1.0 & 1.0 \end{pmatrix}.$$

The solution computed by our method is

$$G_c = \begin{pmatrix} 2.499\,999\,9998-01 & -9.094\,947\,0177-13 & 2.500\,000\,0001-01 \\ 4.999\,999\,9994-01 & 9.999\,999\,9999-01 & -4.999\,999\,9995-01 \\ -4.999\,999\,9997-01 & 1.818\,989\,4035-12 & 4.999\,999\,9997-01 \\ 2.499\,999\,9998-01 & -9.094\,947\,0177-13 & 2.500\,000\,0001-01 \end{pmatrix},$$

while the exact solution is

$$G = \begin{pmatrix} 1/4 & 0 & 1/4 \\ 1/2 & 1 & -1/2 \\ -1/2 & 0 & 1/2 \\ 1/4 & 0 & 1/4 \end{pmatrix}.$$

See Table 5 for the second column of G.

Table 5

Iteration	Inner cycle	x_1	x_2	x_3	x_4
Initial		0.0	0.0	0.0	0.0
1	3	−1.666 666 6667−01	5.000 000 0000−01	3.333 333 3333−01	−1.666 666 6667−01
6	3	−1.060 956 7901−02	9.622 127 9149−01	2.121 913 5803−02	−1.060 956 7901−02
11	3	−7.818 889 2500−04	9.972 152 6130−01	1.563 777 8500−03	−7.818 889 2500−04
31	3	−2.306 296 6648−08	9.999 999 1786−01	4.612 593 3295−08	−2.306 296 6648−08
41	3	−1.255 102 6884−10	9.999 999 9955−01	2.510 205 3769−10	−1.255 102 6884−10
51	3	−9.094 947 0177−13	9.999 999 9999−01	1.818 989 4035−12	−9.094 947 0177−13
True solution		0.0	1.0	0.0	0.0

Problem 6. Find x which satisfies the 84-dimensional non-singular system

$$\underbrace{\begin{pmatrix} 6 & 1 & & & & & \\ 8 & 6 & 1 & & & & \\ & 8 & 6 & 1 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot \\ & & & & 8 & 6 & 1 \\ & & & & & 8 & 6 & 1 \\ & & & & & & 8 & 6 \end{pmatrix}}_{84} x = \begin{pmatrix} 7 \\ 15 \\ 15 \\ \cdot \\ \cdot \\ \cdot \\ 15 \\ 15 \\ 14 \end{pmatrix} 84,$$

which was investigated by Nagasaka in the study of error propagation in Gaussian elimination. The exact solution is $x_1 = x_2 = \cdots = x_{83} = x_{84} = 1.0$ and the solution computed by Gaussian elimination without pivoting or iterative improvement is

$$\begin{aligned}
x_1 &= 1.000\,000\,0000 \\
x_2 &= 1.000\,000\,0000 \\
x_3 &= 9.999\,999\,9998-01 \\
x_4 &= 1.000\,000\,0000 \\
x_5 &= 9.999\,999\,9993-01 \\
&\cdots\cdots\cdots \\
x_{80} &= 2.414\,274\,9372+12 \\
x_{81} &= -4.672\,790\,2010+12 \\
x_{82} &= 8.722\,541\,7086+12 \\
x_{83} &= -1.495\,292\,8643+13 \\
x_{84} &= 1.993\,723\,8191+13\,.
\end{aligned}$$

We obtained Table 6 and 7, applying our process.

Table 6

Iteration	Inner cycle	x_1	x_5	x_{80}	x_{84}
Initial		0.0	0.0	0.0	0.0
6	84	9.979 816 7924−01	9.978 048 8341−01	9.926 228 8024−01	7.013 465 3555−01
11	84	1.000 313 1902−00	1.000 284 1201−00	9.693 864 2414−01	7.070 335 0975−01
21	84	1.000 016 5118−00	1.000 024 0518−00	9.649 521 9062−01	7.082 487 5000−01
31	84	1.000 001 3869−00	1.000 002 4392−00	9.647 035 7023−01	7.083 258 5853−01
51	84	1.000 000 0147−00	1.000 000 0306−00	9.646 812 2669−01	7.083 332 5018−01
61	84	1.000 000 0017−00	1.000 000 0036−00	9.646 810 1655−01	7.083 333 2379−01
91	84	1.000 000 0000−00	1.000 000 0000−00	9.646 809 8965−01	7.083 333 3343−01
True solution		1.0	1.0	1.0	1.0

Table 7 (at 95-th iteration)

$x_1 = x_2 = \cdots = x_{45} = 1.000\,000\,0000-00$. The rest are as follows.

1.000 000 0000−00	9.999 999 9999−01	1.000 000 0000−00	9.999 999 9999−01	1.000 000 0000−00
1.000 000 0000−00	9.999 999 9994−01	1.000 000 0002−00	9.999 999 9960−01	1.000 000 0009−00
9.999 999 9805−01	1.000 000 0041−00	9.999 999 9161−01	1.000 000 0170−00	9.999 999 6562−01
1.000 000 0691−00	9.999 998 6136−01	1.000 000 2777−00	9.999 994 4416−01	1.000 001 1121−00
9.999 977 7522−01	1.000 004 4500−00	9.999 910 9951−01	1.000 017 8013−00	9.999 643 9758−01
1.000 071 2030−00	9.998 576 0228−01	1.000 284 7610−00	9.994 306 1688−01	1.001 138 2101−00
9.977 258 0476−01	1.004 539 4897−00	9.909 566 2441−01	1.017 944 3359−00	9.646 809 8962−01
1.068 359 3750−00	8.723 958 3338−01	1.218 749 9999−00	7.083 333 3344−01	

Note that those examples are computed without using double precision arithmetic.

The author would like to thank M. Shibuya for helpful suggestions.

References

1. Forsythe, G. E.: Solving linear algebraic equations can be interesting. Bull. Amer. Math. Soc. **59**, 299–329 (1953).
2. Kaczmarz, S.: Angenäherte Auflösung von Systemen linearer Gleichungen. Bull. Acad. Polon. Sciences et Lettres, A, 355–357 (1937).
3. Nagasaka, H.: Error propagation in the solution of tridiagonal linear equations. Information Processing in Japan **5**, 38–44 (1965).
4. Penrose, R.: A generalized inverse for matrices. Proc. Cambr. Phil. Soc. **51**, 406–413 (1955).
5. Robinson, D. W.: On the generalized inverse of an arbitrary linear transformation. Amer. Math. Monthly **69**, 412–416 (1962).
6. Rosen, J. B.: The gradient projection method for nonlinear programming. Part II, nonlinear constraints. J. Soc. Indust. Appl. Math. **9**, no. 4, 514–532 (1961).
7. Tanabe, K.: An algorithm for the constrained maximization in nonlinear programming, Research Memorandum No. 31, The Institute of Statistical Mathematics, 1969.

K. Tanabe
The Institute of Statistical Mathematics
4–6–7 Minami-Azabu
Minatoku, Tokyo/Japan