# Iterative techniques in matrix algebra

Tsung-Ming Huang

Department of Mathematics
National Taiwan Normal University, Taiwan

August 28, 2011

# Outline

1. Norms of vectors and matrices
2. Eigenvalues and eigenvectors
3. Iterative techniques for solving linear systems
4. Error bounds and iterative refinement
5. The conjugate gradient method

## Definition 1

$\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ is a vector norm if

**(i)** $\|x\| \geq 0, \ \forall\, x \in \mathbb{R}^n$,

**(ii)** $\|x\| = 0$ if and only if $x = 0$,

**(iii)** $\|\alpha x| = |\alpha|\|x\| \ \forall\, \alpha \in \mathbb{R}$ and $x \in \mathbb{R}^n$,

**(iv)** $\|x + y\| \leq \|x\| + \|y\| \ \forall\, x, y \in \mathbb{R}^n$.

## Definition 2

The $\ell_2$ and $\ell_\infty$ norms for $x = [x_1, x_2, \cdots, x_n]^T$ are defined by

$$\|x\|_2 = (x^T x)^{1/2} = \left\{ \sum_{i=1}^{n} x_i^2 \right\}^{1/2} \quad \text{and} \quad \|x\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

The $\ell_2$ norm is also called the Euclidean norm.

## Theorem 3 (Cauchy-Bunyakovsky-Schwarz inequality)

*For each* $x = [x_1, x_2, \cdots, x_n]^T$ *and* $y = [y_1, y_2, \cdots, y_n]^T$ *in* $\mathbb{R}^n$,

$$x^T y = \sum_{i=1}^{n} x_i y_i \leq \left\{ \sum_{i=1}^{n} x_i^2 \right\}^{1/2} \left\{ \sum_{i=1}^{n} y_i^2 \right\}^{1/2} = \|x\|_2 \cdot \|y\|_2.$$

*Proof:* If $x = 0$ or $y = 0$, the result is immediate.
Suppose $x \neq 0$ and $y \neq 0$. For each $\alpha \in \mathbb{R}$,

$$0 \leq \|x - \alpha y\|_2^2 = \sum_{i=1}^{n} (x_i - \alpha y_i)^2 = \sum_{i=1}^{n} x_i^2 - 2\alpha \sum_{i=1}^{n} x_i y_i + \alpha^2 \sum_{i=1}^{n} y_i^2,$$

and

$$2\alpha \sum_{i=1}^{n} x_i y_i \leq \sum_{i=1}^{n} x_i^2 + \alpha^2 \sum_{i=1}^{n} y_i^2 = \|x\|_2^2 + \alpha^2 \|y\|_2^2.$$

Since $\|x\|_2 > 0$ and $\|y\|_2 > 0$, we can let

$$\alpha = \frac{\|x\|_2}{\|y\|_2}$$

to give

$$\left(2\frac{\|x\|_2}{\|y\|_2}\right)\left(\sum_{i=1}^{n} x_i y_i\right) \leq \|x\|_2^2 + \frac{\|x\|_2^2}{\|y\|_2^2}\|y\|_2^2 = 2\|x\|_2^2.$$

Thus

$$x^T y = \sum_{i=1}^{n} x_i y_i \leq \|x\|_2 \|y\|_2.$$

For each $x, y \in \mathbb{R}^n$,

$$\begin{aligned} \|x+y\|_\infty &= \max_{1\le i\le n} |x_i + y_i| \le \max_{1\le i\le n} (|x_i| + |y_i|) \\ &\le \max_{1\le i\le n} |x_i| + \max_{1\le i\le n} |y_i| = \|x\|_\infty + \|y\|_\infty \end{aligned}$$

and

$$\begin{aligned} \|x+y\|_2^2 &= \sum_{i=1}^{n} (x_i + y_i)^2 = \sum_{i=1}^{2} x_i^2 + 2\sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} y_i^2 \\ &\le \|x\|_2^2 + 2\|x\|_2\|y\|_2 + \|y\|_2^2 = (\|x\|_2 + \|y\|_2)^2, \end{aligned}$$

which gives

$$\|x+y\|_2 \le \|x\|_2 + \|y\|_2.$$

## Definition 4

A sequence $\{x^{(k)} \in \mathbb{R}^n\}_{k=1}^{\infty}$ is convergent to $x$ with respect to the norm $\|\cdot\|$ if $\forall\ \varepsilon > 0$, $\exists$ an integer $N(\varepsilon)$ such that

$$\|x^{(k)} - x\| < \varepsilon,\ \forall\ k \geq N(\varepsilon).$$

## Theorem 5

*$\{x^{(k)} \in \mathbb{R}^n\}_{k=1}^{\infty}$ converges to $x$ with respect to $\|\cdot\|_\infty$ if and only if*

$$\lim_{k\to\infty} x_i^{(k)} = x_i,\ \forall\ i = 1, 2, \ldots, n.$$

*Proof:* "$\Rightarrow$" Given any $\varepsilon > 0$, $\exists$ an integer $N(\varepsilon)$ such that

$$\max_{1\leq i\leq n} |x_i^{(k)} - x_i| = \|x^{(k)} - x\|_\infty < \varepsilon,\ \forall\ k \geq N(\varepsilon).$$

This result implies that

$$|x_i^{(k)} - x_i| < \varepsilon, \ \forall\, i = 1, 2, \ldots, n.$$

Hence

$$\lim_{k\to\infty} x_i^{(k)} = x_i, \ \forall\, i.$$

"$\Leftarrow$" For a given $\varepsilon > 0$, let $N_i(\varepsilon)$ represent an integer with

$$|x_i^{(k)} - x_i| < \varepsilon, \ \text{ whenever } \ k \geq N_i(\varepsilon).$$

Define

$$N(\varepsilon) = \max_{1\leq i\leq n} N_i(\varepsilon).$$

If $k \geq N(\varepsilon)$, then

$$\max_{1\leq i\leq n} |x_i^{(k)} - x_i| = \|x^{(k)} - x\|_\infty < \varepsilon.$$

This implies that $\{x^{(k)}\}$ converges to $x$ with respect to $\|\cdot\|_\infty$. ■

**Theorem 6**

*For each $x \in \mathbb{R}^n$,*

$$\|x\|_\infty \le \|x\|_2 \le \sqrt{n}\|x\|_\infty.$$

*Proof:* Let $x_j$ be a coordinate of $x$ such that

$$\|x\|_\infty^2 = |x_j|^2 \le \sum_{i=1}^n x_i^2 = \|x\|_2^2,$$

so $\|x\|_\infty \le \|x\|_2$ and

$$\|x\|_2^2 = \sum_{i=1}^n x_i^2 \le \sum_{i=1}^n x_j^2 = nx_j^2 = n\|x\|_\infty^2,$$

so $\|x\|_2 \le \sqrt{n}\|x\|_\infty$. ∎

## Definition 7

A matrix norm $\|\cdot\|$ on the set of all $n \times n$ matrices is a real-valued function satisfying for all $n \times n$ matrices $A$ and $B$ and all real number $\alpha$:

- **(i)** $\|A\| \geq 0$;
- **(ii)** $\|A\| = 0$ if and only if $A = 0$;
- **(iii)** $\|\alpha A\| = |\alpha|\|A\|$;
- **(iv)** $\|A + B\| \leq \|A\| + \|B\|$;
- **(v)** $\|AB\| \leq \|A\|\|B\|$;

## Theorem 8

*If $\|\cdot\|$ is a vector norm on $\mathbb{R}^n$, then*

$$\|A\| = \max_{\|x\|=1} \|Ax\|$$

*is a matrix norm.*

For any $z \neq 0$, we have $x = z/\|z\|$ as a unit vector. Hence

$$\|A\| = \max_{\|x\|=1} \|Ax\| = \max_{z\neq 0} \left\| A\left(\frac{z}{\|z\|}\right)\right\| = \max_{z\neq 0} \frac{\|Az\|}{\|z\|}.$$

## Corollary 9

$$\|Az\| \leq \|A\| \cdot \|z\|.$$

## Theorem 10

*If $A = [a_{ij}]$ is an $n \times n$ matrix, then*

$$\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|.$$

*Proof:* Let $x$ be an $n$-dimension vector with

$$1 = \|x\|_\infty = \max_{1 \le i \le n} |x_i|.$$

Then

$$\begin{aligned} \|Ax\|_\infty &= \max_{1 \le i \le n} \left| \sum_{j=1}^{n} a_{ij} x_j \right| \\ &\le \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}| \max_{1 \le j \le n} |x_j| = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|. \end{aligned}$$

Consequently,

$$\|A\|_\infty = \max_{\|x\|_\infty = 1} \|Ax\|_\infty \le \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|.$$

On the other hand, let $p$ be an integer with

$$\sum_{j=1}^{n} |a_{pj}| = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|,$$

and $x$ be the vector with

$$x_j = \begin{cases} 1, & \text{if } a_{pj} \geq 0, \\ -1, & \text{if } a_{pj} < 0. \end{cases}$$

Then

$$\|x\|_\infty = 1 \;\text{ and }\; a_{pj}x_j = |a_{pj}|, \; \forall\, j = 1, 2, \ldots, n,$$

so

$$\|Ax\|_\infty = \max_{1\leq i\leq n} \left| \sum_{j=1}^{n} a_{ij}x_j \right| \geq \left| \sum_{j=1}^{n} a_{pj}x_j \right| = \left| \sum_{j=1}^{n} |a_{pj}| \right| = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|.$$

This result implies that

$$\|A\|_\infty = \max_{\|x\|_\infty=1} \|Ax\|_\infty \geq \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|.$$

which gives

$$\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|.$$

# Eigenvalues and eigenvectors

**Definition 11 (Characteristic polynomial)**

If $A$ is a square matrix, the characteristic polynomial of $A$ is defined by

$$p(\lambda) = \det(A - \lambda I).$$

**Definition 12 (Eigenvalue and eigenvector)**

If $p$ is the characteristic polynomial of the matrix $A$, the zeros of $p$ are eigenvalues of the matrix $A$. If $\lambda$ is an eigenvalue of $A$ and $x \neq 0$ satisfies $(A - \lambda I)x = 0$, then $x$ is an eigenvector of $A$ corresponding to the eigenvalue $\lambda$.

**Definition 13 (Spectrum and Spectral Radius)**

The set of all eigenvalues of a matrix $A$ is called the spectrum of $A$. The spectral radius of $A$ is

$$\rho(A) = \max\{|\lambda|; \lambda \text{ is an eigenvalue of} A\}.$$

**Theorem 14**

*If $A$ is an $n \times n$ matrix, then*

**(i)** $\|A\|_2 = \sqrt{\rho(A^T A)}$;

**(ii)** $\rho(A) \leq \|A\|$ *for any matrix norm.*

*Proof:* Proof for the second part. Suppose $\lambda$ is an eigenvalue of $A$ and $x \neq 0$ is a corresponding eigenvector such that $Ax = \lambda x$ and $\|x\| = 1$. Then

$$|\lambda| = |\lambda|\|x\| = \|\lambda x\| = \|Ax\| \leq \|A\|\|x\| = \|A\|,$$

that is, $|\lambda| \leq \|A\|$. Since $\lambda$ is arbitrary, this implies that $\rho(A) = \max |\lambda| \leq \|A\|$. ■

**Theorem 15**

*For any $A$ and any $\varepsilon > 0$, there exists a matrix norm $\| \cdot \|$ such that*

$$\rho(A) < \|A\| < \rho(A) + \varepsilon.$$

## Definition 16

We call an $n \times n$ matrix $A$ convergent if

$$\lim_{k\to\infty} (A^k)_{ij} = 0 \ \forall\ i = 1, 2, \ldots, n \ \text{ and } \ j = 1, 2, \ldots, n.$$

## Theorem 17

*The following statements are equivalent.*

1. $A$ *is a convergent matrix;*
2. $\lim_{k\to\infty} \|A^k\| = 0$ *for some matrix norm;*
3. $\lim_{k\to\infty} \|A^k\| = 0$ *for all matrix norm;*
4. $\rho(A) < 1$;
5. $\lim_{k\to\infty} A^k x = 0$ *for any* $x$.

# Iterative techniques for solving linear systems

- For small dimension of linear systems, it requires for direct techniques.
- For large systems, iterative techniques are efficient in terms of both computer storage and computation.

The basic idea of iterative techniques is to split the coefficient matrix $A$ into

$$A = M - (M - A),$$

for some matrix $M$, which is called the splitting matrix. Here we assume that $A$ and $M$ are both nonsingular. Then the original problem is rewritten in the equivalent form

$$Mx = (M - A)x + b.$$

This suggests an iterative process

$$x^{(k)} = (I - M^{-1}A)x^{(k-1)} + M^{-1}b \equiv Tx^{(k-1)} + c,$$

where $T$ is usually called the iteration matrix. The initial vector $x^{(0)}$ can be arbitrary or be chosen according to certain conditions.
Two criteria for choosing the splitting matrix $M$ are

- $x^{(k)}$ is easily computed. More precisely, the system $Mx^{(k)} = y$ is easy to solve;
- the sequence $\{x^{(k)}\}$ converges rapidly to the exact solution.

Note that one way to achieve the second goal is to choose $M$ so that $M^{-1}$ approximate $A^{-1}$,
In the following subsections, we will introduce some of the mostly commonly used classic iterative methods.

# Jacobi Method

If we decompose the coefficient matrix $A$ as

$$A = L + D + U,$$

where $D$ is the diagonal part, $L$ is the strictly lower triangular part, and $U$ is the strictly upper triangular part, of $A$, and choose $M = D$, then we derive the iterative formulation for Jacobi method:

$$x^{(k)} = -D^{-1}(L+U)x^{(k-1)} + D^{-1}b.$$

With this method, the iteration matrix $T_J = -D^{-1}(L+U)$ and $c = D^{-1}b$.
Each component $x_i^{(k)}$ can be computed by

$$x_i^{(k)} = \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k-1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} \right) \Bigg/ a_{ii}.$$

$$
\begin{array}{rcl}
a_{11}x_1^{(k)} + a_{12}x_2^{(k-1)} + a_{13}x_3^{(k-1)} + \cdots + a_{1n}x_n^{(k-1)} & = & b_1 \\
a_{21}x_1^{(k-1)} + a_{22}x_2^{(k)} + a_{23}x_3^{(k-1)} + \cdots + a_{2n}x_n^{(k-1)} & = & b_2 \\
& \vdots & \\
a_{n1}x_1^{(k-1)} + a_{n2}x_2^{(k-1)} + a_{n3}x_3^{(k-1)} + \cdots + a_{nn}x_n^{(k)} & = & b_n.
\end{array}
$$

**Algorithm 1 (Jacobi Method)**

Given $x^{(0)}$, tolerance $TOL$, maximum number of iteration $M$.
Set $k = 1$.
While $k \le M$ and $\|x - x^{(0)}\|_2 \ge TOL$
  Set $k = k+1$, $x^{(0)} = x$.
  For $i = 1, 2, \ldots, n$

$$x_i = \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(0)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(0)} \right) \Big/ a_{ii}$$

  End For
End While

## Example 18

Consider the linear system $Ax = b$ given by

$$
\begin{array}{rrcrcrcrcr}
E_1: & 10x_1 & - & x_2 & + & 2x_3 & & & = & 6, \\
E_2: & -x_1 & + & 11x_2 & - & x_3 & + & 3x_4 & = & 25, \\
E_3: & 2x_1 & - & x_2 & + & 10x_3 & - & x_4 & = & -11, \\
E_4: & & & 3x_2 & - & x_3 & + & 8x_4 & = & 15
\end{array}
$$

which has the unique solution $x = [1, 2, -1, 1]^T$.

Solving equation $E_i$ for $x_i$, for $i = 1, 2, 3, 4$, we obtain

$$
\begin{array}{rcrcrcrcrcr}
x_1 & = & & & 1/10x_2 & - & 1/5x_3 & & & + & 3/5, \\
x_2 & = & 1/11x_1 & & & + & 1/11x_3 & - & 3/11x_4 & + & 25/11, \\
x_3 & = & -1/5x_1 & + & 1/10x_2 & & & + & 1/10x_4 & - & 11/10, \\
x_4 & = & & - & 3/8x_2 & + & 1/8x_3 & & & + & 15/8.
\end{array}
$$

Then $Ax = b$ can be rewritten in the form $x = Tx + c$ with

$$T = \begin{bmatrix} 0 & 1/10 & -1/5 & 0 \\ 1/11 & 0 & 1/11 & -3/11 \\ -1/5 & 1/10 & 0 & 1/10 \\ 0 & -3/8 & 1/8 & 0 \end{bmatrix} \text{ and } c = \begin{bmatrix} 3/5 \\ 25/11 \\ -11/10 \\ 15/8 \end{bmatrix}$$

and the iterative formulation for Jacobi method is

$$x^{(k)} = Tx^{(k-1)} + c \text{ for } k = 1, 2, \ldots .$$

The numerical results of such iteration is list as follows:

| k | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|
| 0 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 1 | 0.6000 | 2.2727 | -1.1000 | 1.8750 |
| 2 | 1.0473 | 1.7159 | -0.8052 | 0.8852 |
| 3 | 0.9326 | 2.0533 | -1.0493 | 1.1309 |
| 4 | 1.0152 | 1.9537 | -0.9681 | 0.9738 |
| 5 | 0.9890 | 2.0114 | -1.0103 | 1.0214 |
| 6 | 1.0032 | 1.9922 | -0.9945 | 0.9944 |
| 7 | 0.9981 | 2.0023 | -1.0020 | 1.0036 |
| 8 | 1.0006 | 1.9987 | -0.9990 | 0.9989 |
| 9 | 0.9997 | 2.0004 | -1.0004 | 1.0006 |
| 10 | 1.0001 | 1.9998 | -0.9998 | 0.9998 |

## Matlab code of Example

```
clear all; delete rslt.dat; diary rslt.dat; diary on;
n = 4; xold = zeros(n,1); xnew = zeros(n,1); T = zeros(n,n);
T(1,2) = 1/10; T(1,3) = -1/5; T(2,1) = 1/11;
T(2,3) = 1/11; T(2,4) = -3/11; T(3,1) = -1/5;
T(3,2) = 1/10; T(3,4) = 1/10; T(4,2) = -3/8; T(4,3) = 1/8;
c(1,1) = 3/5; c(2,1) = 25/11; c(3,1) = -11/10; c(4,1) = 15/8;
xnew = T * xold + c; k = 0;
fprintf(' k      x1      x2      x3      x4      \n');
while ( k <= 100 & norm(xnew-xold) > 1.0d-14 )
    xold = xnew; xnew = T * xold + c; k = k + 1;
    fprintf('%3.0f ',k);
    for jj = 1:n
        fprintf('%5.4f ',xold(jj));
    end
    fprintf('\n');
end
```

# Gauss-Seidel Method

When computing $x_i^{(k)}$ for $i > 1$, $x_1^{(k)}, \ldots, x_{i-1}^{(k)}$ have already been computed and are likely to be better approximations to the exact $x_1, \ldots, x_{i-1}$ than $x_1^{(k-1)}, \ldots, x_{i-1}^{(k-1)}$. It seems reasonable to compute $x_i^{(k)}$ using these most recently computed values.

That is

$$
\begin{array}{rcl}
a_{11}x_1^{(k)} + a_{12}x_2^{(k-1)} + a_{13}x_3^{(k-1)} + \cdots + a_{1n}x_n^{(k-1)} & = & b_1 \\
a_{21}x_1^{(k)} + a_{22}x_2^{(k)} + a_{23}x_3^{(k-1)} + \cdots + a_{2n}x_n^{(k-1)} & = & b_2 \\
a_{31}x_1^{(k)} + a_{32}x_2^{(k)} + a_{33}x_3^{(k)} + \cdots + a_{3n}x_n^{(k-1)} & = & b_3 \\
& \vdots & \\
a_{n1}x_1^{(k-1)} + a_{n2}x_2^{(k-1)} + a_{n3}x_3^{(k-1)} + \cdots + a_{nn}x_n^{(k)} & = & b_n.
\end{array}
$$

This improvement induce the Gauss-Seidel method.

The Gauss-Seidel method sets $M = D + L$ and defines the iteration as

$$x^{(k)} = -(D+L)^{-1}Ux^{(k-1)} + (D+L)^{-1}b.$$

That is, Gauss-Seidel method uses $T_G = -(D+L)^{-1}U$ as the iteration matrix. The formulation above can be rewritten as

$$x^{(k)} = -D^{-1}\left(Lx^{(k)} + Ux^{(k-1)} - b\right).$$

Hence each component $x_i^{(k)}$ can be computed by

$$x_i^{(k)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\right) \Bigg/ a_{ii}.$$

- For Jacobi method, only the components of $x^{(k-1)}$ are used to compute $x^{(k)}$. Hence $x_i^{(k)}, i = 1, \ldots, n$, can be computed in parallel at each iteration $k$.
- At each iteration of Gauss-Seidel method, since $x_i^{(k)}$ can not be computed until $x_1^{(k)}, \ldots, x_{i-1}^{(k)}$ are available, the method is not a parallel algorithm in nature.

## Algorithm 2 (Gauss-Seidel Method)

Given $x^{(0)}$, tolerance $TOL$, maximum number of iteration $M$.
Set $k = 1$.
For $i = 1, 2, \ldots, n$

$$x_i = \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^{n} a_{ij} x_j^{(0)} \right) \Big/ a_{ii}$$

End For
While $k \leq M$ and $\|x - x^{(0)}\|_2 \geq TOL$
    Set $k = k + 1$, $x^{(0)} = x$.
    For $i = 1, 2, \ldots, n$

$$x_i = \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^{n} a_{ij} x_j^{(0)} \right) \Big/ a_{ii}$$

    End For
End While

## Example 19

Consider the linear system $Ax = b$ given by

$$
\begin{array}{rrcrcrcrcr}
E_1: & 10x_1 & - & x_2 & + & 2x_3 & & & = & 6, \\
E_2: & -x_1 & + & 11x_2 & - & x_3 & + & 3x_4 & = & 25, \\
E_3: & 2x_1 & - & x_2 & + & 10x_3 & - & x_4 & = & -11, \\
E_4: & & & 3x_2 & - & x_3 & + & 8x_4 & = & 15
\end{array}
$$

which has the unique solution $x = [1, 2, -1, 1]^T$.

Gauss-Seidel method gives the equation

$$
\begin{array}{rcrcrcrcrcr}
x_1^{(k)} & = & & & \frac{1}{10}x_2^{(k-1)} & - & \frac{1}{5}x_3^{(k-1)} & & & + & \frac{3}{5}, \\
x_2^{(k)} & = & \frac{1}{11}x_1^{(k)} & & & + & \frac{1}{11}x_3^{(k-1)} & - & \frac{3}{11}x_4^{(k-1)} & + & \frac{25}{11}, \\
x_3^{(k)} & = & -\frac{1}{5}x_1^{(k)} & + & \frac{1}{10}x_2^{(k)} & & & + & \frac{1}{10}x_4^{(k-1)} & - & \frac{11}{10}, \\
x_4^{(k)} & = & & - & \frac{3}{8}x_2^{(k)} & + & \frac{1}{8}x_3^{(k)} & & & + & \frac{15}{8}.
\end{array}
$$

The numerical results of such iteration is list as follows:

| k | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|
| 0 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 1 | 0.6000 | 2.3273 | -0.9873 | 0.8789 |
| 2 | 1.0302 | 2.0369 | -1.0145 | 0.9843 |
| 3 | 1.0066 | 2.0036 | -1.0025 | 0.9984 |
| 4 | 1.0009 | 2.0003 | -1.0003 | 0.9998 |
| 5 | 1.0001 | 2.0000 | -1.0000 | 1.0000 |

- The results of Example appear to imply that the Gauss-Seidel method is superior to the Jacobi method.
- This is almost always true, but there are linear systems for which the Jacobi method converges and the Gauss-Seidel method does not.
- See Exercises 17 and 18.

## Matlab code of Example

```
clear all; delete rslt.dat; diary rslt.dat; diary on;
n = 4; xold = zeros(n,1); xnew = zeros(n,1); A = zeros(n,n);
A(1,1)=10; A(1,2)=-1; A(1,3)=2; A(2,1)=-1; A(2,2)=11; A(2,3)=-1; A(2,4)=3; A(3,1)=2; A(3,2)=-1;
A(3,3)=10; A(3,4)=-1; A(4,2)=3; A(4,3)=-1; A(4,4)=8; b(1)=6; b(2)=25; b(3)=-11; b(4)=15;
for ii = 1:n
    xnew(ii) = b(ii);
    for jj = 1:ii-1
        xnew(ii) = xnew(ii) - A(ii,jj) * xnew(jj);
    end
    for jj = ii+1:n
        xnew(ii) = xnew(ii) - A(ii,jj) * xold(jj);
    end
    xnew(ii) = xnew(ii) / A(ii,ii);
end
k = 0; fprintf(' k       x1       x2       x3       x4       \n');
while ( k <= 100 & norm(xnew-xold) > 1.0d-14 )
    xold = xnew; k = k + 1;
    for ii = 1:n
        xnew(ii) = b(ii);
        for jj = 1:ii-1
            xnew(ii) = xnew(ii) - A(ii,jj) * xnew(jj);
        end
        for jj = ii+1:n
            xnew(ii) = xnew(ii) - A(ii,jj) * xold(jj);
        end
        xnew(ii) = xnew(ii) / A(ii,ii);
    end
    fprintf('%3.0f ',k);
    for jj = 1:n
        fprintf('%5.4f ',xold(jj));
    end
    fprintf('\n');
end
```

## Lemma 20 (20)

*If $\rho(T) < 1$, then $(I-T)^{-1}$ exists and*

$$(I-T)^{-1} = \sum_{i=0}^{\infty} T^i = I + T + T^2 + \cdots .$$

*Proof:* Let $\lambda$ be an eigenvalue of $T$, then $1-\lambda$ is an eigenvalue of $I-T$. But $|\lambda| \le \rho(A) < 1$, so $1-\lambda \neq 0$ and $0$ is not an eigenvalue of $I-T$, which means $(I-T)$ is nonsingular. Next we show that $(I-T)^{-1} = I + T + T^2 + \cdots$. Since

$$(I-T)\left(\sum_{i=0}^{m} T^i\right) = I - T^{m+1},$$

and $\rho(T) < 1$ implies $\|T^m\| \to 0$ as $m \to \infty$, we have

$$(I-T)\left(\lim_{m\to\infty} \sum_{i=0}^{m} T^i\right) = (I-T)\left(\sum_{i=0}^{\infty} T^i\right) = I. \qquad \blacksquare$$

## Theorem 21

*For any* $x^{(0)} \in \mathbb{R}^n$ *, the sequence produced by*

$$x^{(k)} = Tx^{(k-1)} + c, \quad k = 1, 2, \ldots,$$

*converges to the unique solution of* $x = Tx + c$ *if and only if*

$$\rho(T) < 1.$$

*Proof:* Suppose $\rho(T) < 1$. The sequence of vectors $x^{(k)}$ produced by the iterative formulation are

$$\begin{aligned} x^{(1)} &= Tx^{(0)} + c \\ x^{(2)} &= Tx^{(1)} + c = T^2 x^{(0)} + (T + I)c \\ x^{(3)} &= Tx^{(2)} + c = T^3 x^{(0)} + (T^2 + T + I)c \\ &\vdots \end{aligned}$$

In general

$$x^{(k)} = T^k x^{(0)} + (T^{k-1} + T^{k-2} + \cdots T + I)c.$$

Since $\rho(T) < 1$, $\lim_{k\to\infty} T^k x^{(0)} = 0$ for any $x^{(0)} \in \mathbb{R}^n$. By Lemma 20,

$$(T^{k-1} + T^{k-2} + \cdots T + I)c \to (I - T)^{-1}c, \quad \text{as} \quad k \to \infty.$$

Therefore

$$\lim_{k\to\infty} x^{(k)} = \lim_{k\to\infty} T^k x^{(0)} + \left(\sum_{j=0}^{\infty} T^j\right) c = (I - T)^{-1}c.$$

Conversely, suppose $\{x^{(k)}\} \to \; x = (I - T)^{-1}c$. Since

$$\begin{aligned} x - x^{(k)} &= Tx + c - Tx^{(k-1)} - c = T(x - x^{(k-1)}) = T^2(x - x^{(k-2)}) \\ &= \cdots = T^k(x - x^{(0)}). \end{aligned}$$

Let $z = x - x^{(0)}$. Then

$$\lim_{k\to\infty} T^k z = \lim_{k\to\infty} (x - x^{(k)}) = 0.$$

It follows from theorem $\rho(T) < 1$.

### Theorem 22

*If $\|T\| < 1$, then the sequence $x^{(k)}$ converges to $x$ for any initial $x^{(0)}$ and*

1. $\|x - x^{(k)}\| \leq \|T\|^k \|x - x^{(0)}\|$
2. $\|x - x^{(k)}\| \leq \frac{\|T\|^k}{1-\|T\|} \|x^{(1)} - x^{(0)}\|$.

*Proof:* Since $x = Tx + c$ and $x^{(k)} = Tx^{(k-1)} + c$,

$$\begin{aligned} x - x^{(k)} &= Tx + c - Tx^{(k-1)} - c \\ &= T(x - x^{(k-1)}) \\ &= T^2(x - x^{(k-2)}) = \cdots\cdots = T^k(x - x^{(0)}). \end{aligned}$$

The first statement can then be derived

$$\|x - x^{(k)}\| = \|T^k(x - x^{(0)})\| \leq \|T\|^k \|x - x^{(0)}\|.$$

For the second result, we first show that

$$\|x^{(n)} - x^{(n-1)}\| \leq \|T\|^{n-1} \|x^{(1)} - x^{(0)}\| \text{ for } n \geq 1$$

Since

$$\begin{aligned} x^{(n)} - x^{(n-1)} &= Tx^{(n-1)} + c - Tx^{(n-2)} - c \\ &= T(x^{(n-1)} - x^{(n-2)}) \\ &= T^2(x^{(n-2)} - x^{(n-3)}) = \cdots\cdots = T^{n-1}(x^{(1)} - x^{(0)}), \end{aligned}$$

we have

$$\|x^{(n)} - x^{(n-1)}\| \le \|T\|^{n-1}\|x^{(1)} - x^{(0)}\|.$$

Let $m \ge k$,

$$\begin{aligned} & x^{(m)} - x^{(k)} \\ = & \left(x^{(m)} - x^{(m-1)}\right) + \left(x^{(m-1)} - x^{(m-2)}\right) + \cdots + \left(x^{(k+1)} - x^{(k)}\right) \\ = & T^{m-1}\left(x^{(1)} - x^{(0)}\right) + T^{m-2}\left(x^{(1)} - x^{(0)}\right) + \cdots + T^{k}\left(x^{(1)} - x^{(0)}\right) \\ = & \left(T^{m-1} + T^{m-2} + \cdots T^{k}\right)\left(x^{(1)} - x^{(0)}\right), \end{aligned}$$

hence

$$\begin{aligned} & \|x^{(m)} - x^{(k)}\| \\ \leq & \left(\|T\|^{m-1} + \|T\|^{m-2} + \cdots + \|T\|^{k}\right) \|x^{(1)} - x^{(0)}\| \\ = & \|T\|^{k} \left(\|T\|^{m-k-1} + \|T\|^{m-k-2} + \cdots + 1\right) \|x^{(1)} - x^{(0)}\|. \end{aligned}$$

Since $\lim_{m\to\infty} x^{(m)} = x$,

$$\begin{aligned} & \|x - x^{(k)}\| \\ = & \lim_{m\to\infty} \|x^{(m)} - x^{(k)}\| \\ \leq & \lim_{m\to\infty} \|T\|^{k} \left(\|T\|^{m-k-1} + \|T\|^{m-k-2} + \cdots + 1\right) \|x^{(1)} - x^{(0)}\| \\ = & \|T\|^{k} \|x^{(1)} - x^{(0)}\| \lim_{m\to\infty} \left(\|T\|^{m-k-1} + \|T\|^{m-k-2} + \cdots + 1\right) \\ = & \|T\|^{k} \frac{1}{1 - \|T\|} \|x^{(1)} - x^{(0)}\|. \end{aligned}$$

This proves the second result.

**Theorem 23**

*If $A$ is strictly diagonal dominant, then both the Jacobi and Gauss-Seidel methods converges for any initial vector $x^{(0)}$.*

*Proof:* By assumption, $A$ is strictly diagonal dominant, hence $a_{ii} \neq 0$ (otherwise $A$ is singular) and

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|, \quad i = 1, 2, \ldots, n.$$

For Jacobi method, the iteration matrix $T_J = -D^{-1}(L+U)$ has entries

$$[T_J]_{ij} = \begin{cases} -\frac{a_{ij}}{a_{ii}}, & i \neq j, \\ 0, & i = j. \end{cases}$$

Hence

$$\|T_J\|_\infty = \max_{1\leq i\leq n} \sum_{j=1, j\neq i}^{n} \left|\frac{a_{ij}}{a_{ii}}\right| = \max_{1\leq i\leq n} \frac{1}{|a_{ii}|} \sum_{j=1, j\neq i}^{n} |a_{ij}| < 1,$$

and this implies that the Jacobi method converges.

For Gauss-Seidel method, the iteration matrix $T_G = -(D+L)^{-1}U$. Let $\lambda$ be any eigenvalue of $T_G$ and $y$, $\|y\|_\infty = 1$, is a corresponding eigenvector. Thus

$$T_G y = \lambda y \quad \Longrightarrow \quad -Uy = \lambda(D+L)y.$$

Hence for $i = 1, \ldots, n$,

$$-\sum_{j=i+1}^{n} a_{ij} y_j = \lambda a_{ii} y_i + \lambda \sum_{j=1}^{i-1} a_{ij} y_j.$$

This gives

$$\lambda a_{ii} y_i = -\lambda \sum_{j=1}^{i-1} a_{ij} y_j - \sum_{j=i+1}^{n} a_{ij} y_j$$

and

$$|\lambda||a_{ii}||y_i| \le |\lambda| \sum_{j=1}^{i-1} |a_{ij}||y_j| + \sum_{j=i+1}^{n} |a_{ij}||y_j|.$$

Choose the index $k$ such that $|y_k| = 1 \geq |y_j|$ (this index can always be found since $\|y\|_\infty = 1$). Then

$$|\lambda||a_{kk}| \leq |\lambda| \sum_{j=1}^{k-1} |a_{kj}| + \sum_{j=k+1}^{n} |a_{kj}|$$

which gives

$$|\lambda| \leq \frac{\sum_{j=k+1}^{n} |a_{kj}|}{|a_{kk}| - \sum_{j=1}^{k-1} |a_{kj}|} < \frac{\sum_{j=k+1}^{n} |a_{kj}|}{\sum_{j=k+1}^{n} |a_{kj}|} = 1$$

Since $\lambda$ is arbitrary, $\rho(T_G) < 1$. This means the Gauss-Seidel method converges. ■

- The rate of convergence depends on the spectral radius of the matrix associated with the method.
- One way to select a procedure to accelerate convergence is to choose a method whose associated matrix has minimal spectral radius.

# Successive over-relaxation (SOR) method

**Definition 24**

Suppose $\tilde{x} \in \mathbb{R}^n$ is an approximated solution of $Ax = b$. The residual vector $r$ for $\tilde{x}$ is $r = b - A\tilde{x}$.

Let the approximate solution $\mathbf{x}^{(k,i)}$ produced by Gauss-Seidel method be defined by

$$\mathbf{x}^{(k,i)} = \left[x_1^{(k)}, \ldots, x_{i-1}^{(k)}, x_i^{(k-1)}, \ldots, x_n^{(k-1)}\right]^T$$

and

$$r_i^{(k)} = \left[r_{1i}^{(k)}, r_{2i}^{(k)}, \ldots, r_{ni}^{(k)}\right]^T = b - A\mathbf{x}^{(k,i)}$$

be the corresponding residual vector. Then the $m$th component of $r_i^{(k)}$ is

$$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i}^{n} a_{mj} x_j^{(k-1)},$$

or, equivalently,

$$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i+1}^{n} a_{mj} x_j^{(k-1)} - a_{mi} x_i^{(k-1)},$$

for each $m = 1, 2, \ldots, n$.
In particular, the $i$th component of $r_i^{(k)}$ is

$$r_{ii}^{(k)} = b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} - a_{ii} x_i^{(k-1)},$$

so

$$\begin{aligned} a_{ii} x_i^{(k-1)} + r_{ii}^{(k)} &= b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \\ &= a_{ii} x_i^{(k)}. \end{aligned}$$

Consequently, the Gauss-Seidel method can be characterized as choosing $x_i^{(k)}$ to satisfy

$$x_i^{(k)} = x_i^{(k-1)} + \frac{r_{ii}^{(k)}}{a_{ii}}.$$

Relaxation method is modified the Gauss-Seidel procedure to

$$\begin{aligned} x_i^{(k)} &= x_i^{(k-1)} + \omega \frac{r_{ii}^{(k)}}{a_{ii}} \\ &= x_i^{(k-1)} + \frac{\omega}{a_{ii}} \left[ b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} - a_{ii} x_i^{(k-1)} \right] \\ &= (1-\omega) x_i^{(k-1)} + \frac{\omega}{a_{ii}} \left[ b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \right] \quad (1) \end{aligned}$$

for certain choices of positive $\omega$ such that the norm of the residual vector is reduced and the convergence is significantly faster.

These methods are called for

$\omega < 1$: under relaxation,
$\omega = 1$: Gauss-Seidel method,
$\omega > 1$: over relaxation.

Over-relaxation methods are called SOR (Successive over-relaxation). To determine the matrix of the SOR method, we rewrite (1) as

$$a_{ii}x_i^{(k)} + \omega \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} = (1-\omega)a_{ii}x_i^{(k-1)} - \omega \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} + \omega b_i,$$

so that if $A = L + D + U$, then we have

$$(D+\omega L)x^{(k)} = [(1-\omega)D - \omega U]\, x^{(k-1)} + \omega b$$

or

$$\begin{aligned} x^{(k)} &= (D+\omega L)^{-1}[(1-\omega)D-\omega U]\, x^{(k-1)} + \omega(D+\omega L)^{-1}b \\ &\equiv T_\omega x^{(k-1)} + c_\omega. \end{aligned}$$

## Example 25

The linear system $Ax = b$ given by

$$\begin{array}{rcrcrcr} 4x_1 & + & 3x_2 & & & = & 24, \\ 3x_1 & + & 4x_2 & - & x_3 & = & 30, \\ & - & x_2 & + & 4x_3 & = & -24, \end{array}$$

has the solution $[3, 4, -5]^T$.

- Numerical results of Gauss-Seidel method with $x^{(0)} = [1, 1, 1]^T$:

| k | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| 0 | 1.0000000 | 1.0000000 | 1.0000000 |
| 1 | 5.2500000 | 3.8125000 | -5.0468750 |
| 2 | 3.1406250 | 3.8828125 | -5.0292969 |
| 3 | 3.0878906 | 3.9267578 | -5.0183105 |
| 4 | 3.0549316 | 3.9542236 | -5.0114441 |
| 5 | 3.0343323 | 3.9713898 | -5.0071526 |
| 6 | 3.0214577 | 3.9821186 | -5.0044703 |
| 7 | 3.0134110 | 3.9888241 | -5.0027940 |

- Numerical results of SOR method with $\omega = 1.25$ and $x^{(0)} = [1, 1, 1]^T$:

| k | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| 0 | 1.0000000 | 1.0000000 | 1.0000000 |
| 1 | 6.3125000 | 3.5195313 | -6.6501465 |
| 2 | 2.6223145 | 3.9585266 | -4.6004238 |
| 3 | 3.1333027 | 4.0102646 | -5.0966863 |
| 4 | 2.9570512 | 4.0074838 | -4.9734897 |
| 5 | 3.0037211 | 4.0029250 | -5.0057135 |
| 6 | 2.9963276 | 4.0009262 | -4.9982822 |
| 7 | 3.0000498 | 4.0002586 | -5.0003486 |

- Numerical results of SOR method with $\omega = 1.6$ and $x^{(0)} = [1, 1, 1]^T$:

| k | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| 0 | 1.0000000 | 1.0000000 | 1.0000000 |
| 1 | 7.8000000 | 2.4400000 | -9.2240000 |
| 2 | 1.9920000 | 4.4560000 | -2.2832000 |
| 3 | 3.0576000 | 4.7440000 | -6.3324800 |
| 4 | 2.0726400 | 4.1334400 | -4.1471360 |
| 5 | 3.3962880 | 3.7855360 | -5.5975040 |
| 6 | 3.0195840 | 3.8661760 | -4.6950272 |
| 7 | 3.1488384 | 4.0236774 | -5.1735127 |

## Matlab code of SOR

```
clear all; delete rslt.dat; diary rslt.dat; diary on;
n = 3; xold = zeros(n,1); xnew = zeros(n,1); A = zeros(n,n); DL = zeros(n,n); DU = zeros(n,n);
A(1,1)=4; A(1,2)=3; A(2,1)=3; A(2,2)=4; A(2,3)=-1; A(3,2)=-1; A(3,3)=4;
b(1,1)=24; b(2,1)=30; b(3,1)=-24; omega=1.25;
for ii = 1:n
    DL(ii,ii) = A(ii,ii);
    for jj = 1:ii-1
        DL(ii,jj) = omega * A(ii,jj);
    end
    DU(ii,ii) = (1-omega)*A(ii,ii);
    for jj = ii+1:n
        DU(ii,jj) = - omega * A(ii,jj);
    end
end
c = omega * (DL \ b); xnew = DL \ ( DU * xold ) + c;
k = 0; fprintf(' k        x1        x2        x3        \n');
while ( k <= 100 & norm(xnew-xold) > 1.0d-14 )
    xold = xnew; k = k + 1; xnew = DL \ ( DU * xold ) + c;
    fprintf('%3.0f ',k);
    for jj = 1:n
        fprintf('%5.4f ',xold(jj));
    end
    fprintf('\n');
end
diary off
```

### Theorem 26 (Kahan)

*If $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$, then $\rho(T_\omega) \geq |\omega - 1|$. This implies that the SOR method can converge only if $0 < \omega < 2$.*

### Theorem 27 (Ostrowski-Reich)

*If $A$ is positive definite and the relaxation parameter $\omega$ satisfying $0 < \omega < 2$, then the SOR iteration converges for any initial vector $x^{(0)}$.*

### Theorem 28

*If $A$ is positive definite and tridiagonal, then $\rho(T_G) = [\rho(T_J)]^2 < 1$ and the optimal choice of $\omega$ for the SOR iteration is*

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_J)]^2}}.$$

*With this choice of $\omega$, $\rho(T_\omega) = \omega - 1$.*

## Example 29

The matrix

$$A = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix},$$

given in previous example, is positive definite and tridiagonal.

Since

$$\begin{aligned} T_J &= -D^{-1}(L+U) = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 0 & -3 & 0 \\ -3 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -0.75 & 0 \\ -0.75 & 0 & 0.25 \\ 0 & 0.25 & 0 \end{bmatrix}, \end{aligned}$$

we have

$$T_J - \lambda I = \begin{bmatrix} -\lambda & -0.75 & 0 \\ -0.75 & -\lambda & 0.25 \\ 0 & 0.25 & -\lambda \end{bmatrix},$$

so

$$\det(T_J - \lambda I) = -\lambda(\lambda^2 - 0.625).$$

Thus,

$$\rho(T_J) = \sqrt{0.625}$$

and

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_J)]^2}} = \frac{2}{1 + \sqrt{1 - 0.625}} \approx 1.24.$$

This explains the rapid convergence obtained in previous example when using $\omega = 0.125$

# Symmetric Successive Over Relaxation (SSOR) Method

Let $A$ be symmetric and $A = D + L + L^T$. The idea is in fact to implement the SOR formulation twice, one forward and one backward, at each iteration. That is, SSOR method defines

$$
\begin{aligned}
(D+\omega L)x^{(k-\frac{1}{2})} &= \left[(1-\omega)D - \omega L^T\right] x^{(k-1)} + \omega b, \qquad (2)\\
(D+\omega L^T)x^{(k)} &= \left[(1-\omega)D - \omega L\right] x^{(k-\frac{1}{2})} + \omega b. \qquad (3)
\end{aligned}
$$

Define

$$
\begin{cases}
M_\omega: &= D + \omega L,\\
N_\omega: &= (1-\omega)D - \omega L^T.
\end{cases}
$$

Then from the iterations (2) and (3), it follows that

$$
\begin{aligned}
x^{(k)} &= \left(M_\omega^{-T} N_\omega^T M_\omega^{-1} N_\omega\right) x^{(k-1)} + \omega \left(M_\omega^{-T} N_\omega^T M_\omega^{-1} + M_\omega^{-T}\right) b\\
&\equiv T(\omega) x^{(k-1)} + M(\omega)^{-1} b.
\end{aligned}
$$

But

$$\begin{aligned}
&((1-\omega)D-\omega L)\,(D+\omega L)^{-1}+I\\
&=(-\omega L-D-\omega D+2D)(D+\omega L)^{-1}+I\\
&=-I+(2-\omega)D(D+\omega L)^{-1}+I\\
&=(2-\omega)D(D+\omega L)^{-1}.
\end{aligned}$$

Thus

$$M(\omega)^{-1}=\omega\left(D+\omega L^T\right)^{-1}(2-\omega)D(D+\omega L)^{-1},$$

then the splitting matrix is

$$M(\omega)=\frac{1}{\omega(2-\omega)}(D+\omega L)D^{-1}\left(D+\omega L^T\right).$$

The iteration matrix is

$$T(\omega)=(D+\omega L^T)^{-1}\left[(1-\omega)D-\omega L\right](D+\omega L)^{-1}\left[(1-\omega)D-\omega L^T\right].$$

# Error bounds and iterative refinement

**Example 30**

The linear system $Ax = b$ given by

$$\begin{bmatrix} 1 & 2 \\ 1.0001 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3.0001 \end{bmatrix}$$

has the unique solution $x = [1, 1]^T$.

The poor approximation $\tilde{x} = [3, 0]^T$ has the residual vector

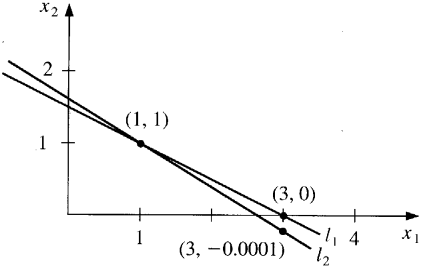
$$r = b - A\tilde{x} = \begin{bmatrix} 3 \\ 3.0001 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 1.0001 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -0.0002 \end{bmatrix},$$

so $\|r\|_\infty = 0.0002$. Although the norm of the residual vector is small, the approximation $\tilde{x} = [3, 0]^T$ is obviously quite poor; in fact, $\|x - \tilde{x}\|_\infty = 2$.

The solution of above example represents the intersection of the lines

$$\ell_1 : \quad x_1 + 2x_2 = 3 \quad \text{and} \quad \ell_2 : \quad 1.0001x_1 + 2x_2 = 3.0001.$$

$\ell_1$ and $\ell_2$ are nearly parallel. The point $(3, 0)$ lies on $\ell_1$ which implies that $(3, 0)$ also lies close to $\ell_2$, even though it differs significantly from the intersection point $(1, 1)$.



## Theorem 31

*Suppose that $\tilde{x}$ is an approximate solution of $Ax = b$, $A$ is nonsingular matrix and $r = b - A\tilde{x}$. Then*

$$\|x - \tilde{x}\| \leq \|r\| \cdot \|A^{-1}\|$$

*and if $x \neq 0$ and $b \neq 0$,*

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \|A\| \cdot \|A^{-1}\| \frac{\|r\|}{\|b\|}.$$

*Proof:* Since

$$r = b - A\tilde{x} = Ax - A\tilde{x} = A(x - \tilde{x})$$

and $A$ is nonsingular, we have

$$\|x - \tilde{x}\| = \|A^{-1}r\| \leq \|A^{-1}\| \cdot \|r\|. \tag{4}$$

Moreover, since $b = Ax$, we have

$$\|b\| \leq \|A\| \cdot \|x\|.$$

It implies that

$$\frac{1}{\|x\|} \leq \frac{\|A\|}{\|b\|}. \tag{5}$$

Combining Equations (4) and (5), we have

$$\frac{\|x-\tilde{x}\|}{\|x\|} \leq \frac{\|A\| \cdot \|A^{-1}\|}{\|b\|}\|r\|.$$

■

**Definition 32 (Condition number)**

The condition number of nonsingular matrix $A$ is

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|.$$

For any nonsingular matrix $A$,

$$1 = \|I\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\| = \kappa(A).$$

## Definition 33

A matrix $A$ is well-conditioned if $\kappa(A)$ is close to 1, and is ill-conditioned when $\kappa(A)$ is significantly greater than 1.

In previous example,

$$A = \begin{bmatrix} 1 & 2 \\ 1.0001 & 2 \end{bmatrix}.$$

Since

$$A^{-1} = \begin{bmatrix} -10000 & 10000 \\ 5000.5 & -5000 \end{bmatrix},$$

we have

$$\kappa(A) = \|A\|_\infty \cdot \|A^{-1}\|_\infty = 3.0001 \times 20000 = 60002 \gg 1.$$

How to estimate the effective condition number in $t$-digit arithmetic without having to invert the matrix $A$?

- If the approximate solution $\tilde{x}$ of $Ax = b$ is being determined using $t$-digit arithmetic and Gaussian elimination, then

$$\|r\| = \|b - A\tilde{x}\| \approx 10^{-t}\|A\| \cdot \|\tilde{x}\|.$$

- All the arithmetic operations in Gaussian elimination technique are performed using $t$-digit arithmetic, but the residual vector $r$ are done in double-precision (i.e., $2t$-digit) arithmetic.
- Use the Gaussian elimination method which has already been calculated to solve

$$Ay = r.$$

Let $\tilde{y}$ be the approximate solution.

Then

$$\tilde{y} \approx A^{-1}r = A^{-1}(b - A\tilde{x}) = x - \tilde{x}$$

and

$$x \approx \tilde{x} + \tilde{y}.$$

Moreover,

$$\begin{aligned} \|\tilde{y}\| &\approx \|x - \tilde{x}\| = \|A^{-1}r\| \\ &\leq \|A^{-1}\| \cdot \|r\| \approx \|A^{-1}\|(10^{-t}\|A\| \cdot \|\tilde{x}\|) = 10^{-t}\|\tilde{x}\|\kappa(A). \end{aligned}$$

It implies that

$$\kappa(A) \approx \frac{\|\tilde{y}\|}{\|\tilde{x}\|}10^{t}.$$

**Iterative refinement**

In general, $\tilde{x} + \tilde{y}$ is a more accurate approximation to the solution of $Ax = b$ than $\tilde{x}$.

**Algorithm 3 (Iterative refinement)**

Given tolerance $TOL$, maximum number of iteration $M$, number of digits of precision $t$.
Solve $Ax = b$ by using Gaussian elimination in $t$-digit arithmetic.
Set $k = 1$
while ( $k \leq M$ )
    Compute $r = b - Ax$ in $2t$-digit arithmetic.
    Solve $Ay = r$ by using Gaussian elimination in $t$-digit arithmetic.
    If $\|y\|_\infty < TOL$, then stop.
    Set $k = k + 1$ and $x = x + y$.
End while

## Example 34

The linear system given by

$$\begin{bmatrix} 3.3330 & 15920 & -10.333 \\ 2.2220 & 16.710 & 9.6120 \\ 1.5611 & 5.1791 & 1.6852 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 15913 \\ 28.544 \\ 8.4254 \end{bmatrix}$$

has the exact solution $x = [1, 1, 1]^T$.

Using Gaussian elimination and five-digit rounding arithmetic leads successively to the augmented matrices

$$\left[\begin{array}{ccc|c} 3.3330 & 15920 & -10.333 & 15913 \\ 0 & -10596 & 16.501 & -10580 \\ 0 & -7451.4 & 6.5250 & -7444.9 \end{array}\right]$$

and

$$\begin{bmatrix} 3.3330 & 15920 & -10.333 & 15913 \\ 0 & -10596 & 16.501 & -10580 \\ 0 & 0 & -5.0790 & -4.7000 \end{bmatrix}.$$

The approximate solution is

$$\tilde{x}^{(1)} = [1.2001, 0.99991, 0.92538]^T.$$

The residual vector corresponding to $\tilde{x}$ is computed in double precision to be

$$\begin{aligned} r^{(1)} &= b - A\tilde{x}^{(1)} \\ &= \begin{bmatrix} 15913 \\ 28.544 \\ 8.4254 \end{bmatrix} - \begin{bmatrix} 3.3330 & 15920 & -10.333 \\ 2.2220 & 16.710 & 9.6120 \\ 1.5611 & 5.1791 & 1.6852 \end{bmatrix} \begin{bmatrix} 1.2001 \\ 0.99991 \\ 0.92538 \end{bmatrix} \\ &= \begin{bmatrix} 15913 \\ 28.544 \\ 8.4254 \end{bmatrix} - \begin{bmatrix} 15913.00518 \\ 28.26987086 \\ 8.611560367 \end{bmatrix} = \begin{bmatrix} -0.00518 \\ 0.27412914 \\ -0.186160367 \end{bmatrix}. \end{aligned}$$

Hence the solution of $Ay = r^{(1)}$ to be

$$\tilde{y}^{(1)} = [-0.20008, 8.9987 \times 10^{-5}, 0.074607]^T$$

and the new approximate solution $x^{(2)}$ is

$$x^{(2)} = x^{(1)} + \tilde{y}^{(1)} = [1.0000, 1.0000, 0.99999]^T.$$

Using the suggested stopping technique for the algorithm, we compute $r^{(2)} = b - A\tilde{x}^{(2)}$ and solve the system $Ay^{(2)} = r^{(2)}$, which gives

$$\tilde{y}^{(2)} = [1.5002 \times 10^{-9}, 2.0951 \times 10^{-10}, 1.0000 \times 10^{-5}]^T.$$

Since

$$\|\tilde{y}^{(2)}\|_\infty \leq 10^{-5},$$

we conclude that

$$\tilde{x}^{(3)} = \tilde{x}^{(2)} + \tilde{y}^{(2)} = [1.0000, 1.0000, 1.0000]^T$$

is sufficiently accurate. ■

In the linear system

$$Ax = b,$$

$A$ and $b$ can be represented exactly. Realistically, the matrix $A$ and vector $b$ will be perturbed by $\delta A$ and $\delta b$, respectively, causing the linear system

$$(A + \delta A)x = b + \delta b$$

to be solved in place of $Ax = b$.

## Theorem 35

*Suppose $A$ is nonsingular and*

$$\|\delta A\| < \frac{1}{\|A^{-1}\|}.$$

*Then the solution $\tilde{x}$ of $(A+\delta A)\tilde{x} = b + \delta b$ approximates the solution $x$ of $Ax = b$ with the error estimate*

$$\frac{\|x-\tilde{x}\|}{\|x\|} \leq \frac{\kappa(A)}{1-\kappa(A)(\|\delta A\|/\|A\|)}\left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|}\right).$$

- If $A$ is well-conditioned, then small changes in $A$ and $b$ produce correspondingly small changes in the solution $x$.
- If $A$ is ill-conditioned, then small changes in $A$ and $b$ may produce large changes in $x$.

# The conjugate gradient method

Consider the linear systems

$$Ax = b$$

where $A$ is large sparse and symmetric positive definite. Define the inner product notation

$$< x, y >= x^T y \text{ for any } x, y \in \mathbb{R}^n.$$

**Theorem 36**

*Let $A$ be symmetric positive definite. Then $x^*$ is the solution of $Ax = b$ if and only if $x^*$ minimizes*

$$g(x) =< x, Ax > -2 < x, b > .$$

## *Proof:*

(" $\Rightarrow$") Rewrite $g(x)$ as

$$\begin{aligned} g(x) &= <x-x^*, A(x-x^*)> + <x, Ax^*> + <x^*, Ax> \\ &\quad - <x^*, Ax^*> -2 <x, b> \\ &= <x-x^*, A(x-x^*)> - <x^*, Ax^*> \\ &\quad +2 <x, Ax^*> -2 <x, b> \\ &= <x-x^*, A(x-x^*)> - <x^*, Ax^*> +2 <x, Ax^*-b> . \end{aligned}$$

Suppose that $x^*$ is the solution of $Ax=b$, i.e., $Ax^*=b$. Then

$$g(x) = <x-x^*, A(x-x^*)> - <x^*, Ax^*>$$

which minimum occurs at $x = x^*$.

("$\Leftarrow$") Fixed vectors $x$ and $v$, for any $\alpha \in \mathbb{R}$,

$$
\begin{aligned}
& f(\alpha) \equiv g(x + \alpha v) \\
= \;& < x + \alpha v, Ax + \alpha Av > -2 < x + \alpha v, b > \\
= \;& < x, Ax > +\alpha < v, Ax > +\alpha < x, Av > +\alpha^2 < v, Av > \\
& -2 < x, b > -2\alpha < v, b > \\
= \;& < x, Ax > -2 < x, b > +2\alpha < v, Ax > -2\alpha < v, b > +\alpha^2 < v, A \\
= \;& g(x) + 2\alpha < v, Ax - b > +\alpha^2 < v, Av > .
\end{aligned}
$$

Because $f$ is a quadratic function of $\alpha$ and $< v, Av >$ is positive, $f$ has a minimal value when $f'(\alpha) = 0$. Since

$$f'(\alpha) = 2 < v, Ax - b > +2\alpha < v, Av >,$$

the minimum occurs at

$$\hat{\alpha} = -\frac{< v, Ax - b >}{< v, Av >} = \frac{< v, b - Ax >}{< v, Av >}.$$

and

$$\begin{aligned} g(x+\hat{\alpha}v) &= f(\hat{\alpha}) = g(x) - 2\frac{< v, b-Ax >}{< v, Av >} < v, b-Ax > \\ &\quad + \left(\frac{< v, b-Ax >}{< v, Av >}\right)^2 < v, Av > \\ &= g(x) - \frac{< v, b-Ax >^2}{< v, Av >}. \end{aligned}$$

So, for any nonzero vector $v$, we have

$$g(x+\hat{\alpha}v) < g(x) \text{ if } < v, b-Ax > \neq 0 \tag{6}$$

and

$$g(x+\hat{\alpha}v) = g(x) \text{ if } < v, b-Ax >= 0. \tag{7}$$

Suppose that $x^*$ is a vector that minimizes $g$. Then

$$g(x^*+\hat{\alpha}v) \geq g(x^*) \text{ for any } v. \tag{8}$$

From (6), (7) and (8), we have

$$< v, b - Ax^* >= 0 \text{ for any } v,$$

which implies that $Ax^* = b$. ■

Let

$$r = b - Ax.$$

Then

$$\alpha = \frac{< v, b - Ax >}{< v, Av >} = \frac{< v, r >}{< v, Av >}.$$

If $r \neq 0$ and if $v$ and $r$ are not orthogonal, then

$$g(x + \alpha v) < g(x)$$

which implies that $x + \alpha v$ is closer to $x^*$ than is $x$.

Let $x^{(0)}$ be an initial approximation to $x^*$ and $v^{(1)} \neq 0$ be an initial search direction. For $k = 1, 2, 3, \ldots$, we compute

$$\begin{aligned} \alpha_k &= \frac{< v^{(k)}, b - Ax^{(k-1)} >}{< v^{(k)}, Av^{(k)} >}, \\ x^{(k)} &= x^{(k-1)} + \alpha_k v^{(k)} \end{aligned}$$

and choose a new search direction $v^{(k+1)}$.

**Question:** How to choose $\{v^{(k)}\}$ such that $\{x^{(k)}\}$ converges rapidly to $x^*$?

Let $\Phi : \mathbb{R}^n \to \mathbb{R}$ be a differential function on $x$. Then it holds

$$\frac{\Phi(x + \varepsilon p) - \Phi(x)}{\varepsilon} = \nabla\Phi(x)^T p + O(\varepsilon).$$

The right hand side takes minimum at

$$p = -\frac{\nabla\Phi(x)}{\|\nabla\Phi(x)\|} \quad \text{(i.e., the largest descent)}$$

for all $p$ with $\|p\| = 1$ (neglect $O(\varepsilon)$).

Denote $x = [x_1, x_2, \ldots, x_n]^T$. Then

$$g(x) =< x, Ax > -2 < x, b >= \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j - 2 \sum_{i=1}^{n} x_i b_i.$$

It follows that

$$\frac{\partial g}{\partial x_k}(x) = 2 \sum_{i=1}^{n} a_{ki} x_i - 2b_k, \quad \text{for} \quad k = 1, 2, \ldots, n.$$

Therefore, the gradient of $g$ is

$$\nabla g(x) = \left[ \frac{\partial g}{\partial x_1}(x), \frac{\partial g}{\partial x_2}, \cdots, \frac{\partial g}{\partial x_n}(x) \right]^T = 2(Ax - b) = -2r.$$

## Steepest descent method (gradient method)

Given an initial $x_0 \neq 0$.
For $k = 1, 2, \ldots$
  $r_{k-1} = b - Ax_{k-1}$
  If $r_{k-1} = 0$, then stop;
  else $\alpha_k = \frac{r_{k-1}^T r_{k-1}}{r_{k-1}^T A r_{k-1}}$, $x_k = x_{k-1} + \alpha_k r_{k-1}$.
End for

## Theorem 37

*If $x_k$, $x_{k-1}$ are two approximations of the steepest descent method for solving $Ax = b$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$ are the eigenvalues of $A$, then it holds:*

$$\|x_k - x^*\|_A \leq \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right) \|x_{k-1} - x^*\|_A,$$

*where $\|x\|_A = \sqrt{x^T A x}$. Thus the gradient method is convergent.*

- If the condition number of $A$ $(= \lambda_1/\lambda_n)$ is large, then $\frac{\lambda_1-\lambda_n}{\lambda_1+\lambda_n} \approx 1$.
  The gradient method converges very slowly. Hence this method is not recommendable.
- It is favorable to choose that the search directions $\{v^{(i)}\}$ as mutually $A$-conjugate, where $A$ is symmetric positive definite.

**Definition 38**

Two vectors $p$ and $q$ are called $A$-conjugate ($A$-orthogonal), if $p^T Aq = 0$.

## Lemma 39

*Let $v_1, \ldots, v_n \neq 0$ be pairwisely $A$-conjugate. Then they are linearly independent.*

*Proof:* From

$$0 = \sum_{j=1}^{n} c_j v_j$$

follows that

$$0 = (v_k)^T A \left( \sum_{j=1}^{n} c_j v_j \right) = \sum_{j=1}^{n} c_j (v_k)^T A v_j = c_k (v_k)^T A v_k,$$

so $c_k = 0$, for $k = 1, \ldots, n$. ∎

## Theorem 40

*Let $A$ be symm. positive definite and $v_1, \ldots, v_n \in \mathbb{R}^n \backslash \{0\}$ be pairwisely $A$-orthogonal. Give $x_0$ and let $r_0 = b - Ax_0$. For $k = 1, \ldots, n$, let*

$$\alpha_k = \frac{< v_k, b - Ax_{k-1} >}{< v_k, Av_k >} \quad \text{and} \quad x_k = x_{k-1} + \alpha_k v_k.$$

*Then $Ax_n = b$ and*

$$< b - Ax_k, v_j >= 0, \quad \text{for each } j = 1, 2, \ldots, k-1.$$

*Proof:* Since, for each $k = 1, 2, \ldots, n$,

$$x_k = x_{k-1} + \alpha_k v_k,$$

we have

$$\begin{aligned} Ax_n &= Ax_{n-1} + \alpha_n Av_n = (Ax_{n-2} + \alpha_{n-1} Av_{n-1}) + \alpha_n Av_n = \cdots \\ &= Ax_0 + \alpha_1 Av_1 + \alpha_2 Av_2 + \cdots + \alpha_n Av_n. \end{aligned}$$

It implies that

$$\begin{aligned}
& < Ax_n - b, v_k > \\
= & < Ax_0 - b, v_k > + \alpha_1 < Av_1, v_k > + \cdots + \alpha_n < Av_n, v_k > \\
= & < Ax_0 - b, v_k > + \alpha_1 < v_1, Av_k > + \cdots + \alpha_n < v_n, Av_k > \\
= & < Ax_0 - b, v_k > + \alpha_k < v_k, Av_k > \\
= & < Ax_0 - b, v_k > + \frac{< v_k, b - Ax_{k-1} >}{< v_k, Av_k >} < v_k, Av_k > \\
= & < Ax_0 - b, v_k > + < v_k, b - Ax_{k-1} > \\
= & < Ax_0 - b, v_k > \\
& + < v_k, b - Ax_0 + Ax_0 - Ax_1 + \cdots - Ax_{k-2} + Ax_{k-2} - Ax_{k-1} > \\
= & < Ax_0 - b, v_k > + < v_k, b - Ax_0 > + < v_k, Ax_0 - Ax_1 > \\
& + \cdots + < v_k, Ax_{k-2} - Ax_{k-1} > \\
= & < v_k, Ax_0 - Ax_1 > + \cdots + < v_k, Ax_{k-2} - Ax_{k-1} > .
\end{aligned}$$

For any $i$

$$x_i = x_{i-1} + \alpha_i v_i \quad \text{and} \quad Ax_i = Ax_{i-1} + \alpha_i Av_i,$$

we have

$$Ax_{i-1} - Ax_i = -\alpha_i Av_i.$$

Thus, for $k = 1, \ldots, n$,

$$\begin{aligned} & < Ax_n - b, v_k > \\ = \quad & -\alpha_1 < v_k, Av_1 > - \cdots - \alpha_{k-1} < v_k, Av_{k-1} >= 0 \end{aligned}$$

which implies that $Ax_n = b$.
Suppose that

$$< r_{k-1}, v_j >= 0 \;\text{ for }\; j = 1, 2, \ldots, k-1. \tag{9}$$

By the result

$$r_k = b - Ax_k = b - A(x_{k-1} + \alpha_k v_k) = r_{k-1} - \alpha_k Av_k$$

it follows that

$$\begin{aligned} < r_k, v_k > &= < r_{k-1}, v_k > -\alpha_k < Av_k, v_k > \\ &= < r_{k-1}, v_k > -\frac{< v_k, b - Ax_{k-1} >}{< v_k, Av_k >} < Av_k, v_k > \\ &= 0. \end{aligned}$$

From assumption (9) and $A$-orthogonality, for $j = 1, \ldots, k-1$

$$< r_k, v_j > = < r_{k-1}, v_j > -\alpha_k < Av_k, v_j > = 0$$

which is completed the proof by the mathematic induction. ■

Method of conjugate directions:

Let $A$ be symmetric positive definite, $b$, $x_0 \in \mathbb{R}^n$. Given $v_1, \ldots, v_n \in \mathbb{R}^n \backslash \{0\}$ pairwisely $A$-orthogonal.

$r_0 = b - Ax_0$,

For $k = 1, \ldots, n$,

$\alpha_k = \frac{< v_k, r_{k-1} >}{< v_k, Av_k >}$, $x_k = x_{k-1} + \alpha_k v_k$,

$r_k = r_{k-1} - \alpha_k Av_k = b - Ax_k$.

## Practical Implementation

- In $k$-th step a direction $v_k$ which is $A$-orthogonal to $v_1, \ldots, v_{k-1}$ must be determined.
- It allows for orthogonalization of $r_k$ against $v_1, \ldots, v_k$.
- Let $r_k \neq 0$, $g(x)$ decreases strictly in the direction $-r_k$. For $\varepsilon > 0$ small, we have $g(x_k - \varepsilon r_k) < g(x_k)$.

If $r_{k-1} = b - Ax_{k-1} \neq 0$, then we use $r_{k-1}$ to generate $v_k$ by

$$v_k = r_{k-1} + \beta_{k-1} v_{k-1}. \tag{10}$$

Choose $\beta_{k-1}$ such that

$$\begin{aligned} 0 &= < v_{k-1}, Av_k > = < v_{k-1}, Ar_{k-1} + \beta_{k-1} Av_{k-1} > \\ &= < v_{k-1}, Ar_{k-1} > + \beta_{k-1} < v_{k-1}, Av_{k-1} > . \end{aligned}$$

That is

$$\beta_{k-1} = -\frac{< v_{k-1}, Ar_{k-1} >}{< v_{k-1}, Av_{k-1} >}. \tag{11}$$

**Theorem 41**

*Let $v_k$ and $\beta_{k-1}$ be defined in (10) and (11), respectively. Then $r_0, \ldots, r_{k-1}$ are mutually orthogonal and*

$$< v_k, Av_i >= 0, \quad \textit{for} \;\; i = 1, 2, \ldots, k-1.$$

*That is $\{v_1, \ldots, v_k\}$ is an $A$-orthogonal set.*

Having chosen $v_k$, we compute

$$\begin{aligned} \alpha_k &= \frac{< v_k, r_{k-1} >}{< v_k, Av_k >} = \frac{< r_{k-1} + \beta_{k-1} v_{k-1}, r_{k-1} >}{< v_k, Av_k >} \\ &= \frac{< r_{k-1}, r_{k-1} >}{< v_k, Av_k >} + \beta_{k-1} \frac{< v_{k-1}, r_{k-1} >}{< v_k, Av_k >} \\ &= \frac{< r_{k-1}, r_{k-1} >}{< v_k, Av_k >}. \end{aligned} \tag{12}$$

Since

$$r_k = r_{k-1} - \alpha_k A v_k,$$

we have

$$< r_k, r_k > = < r_{k-1}, r_k > - \alpha_k < A v_k, r_k > = -\alpha_k < r_k, A v_k > .$$

Further, from (12),

$$< r_{k-1}, r_{k-1} > = \alpha_k < v_k, A v_k >,$$

so

$$\begin{aligned} \beta_k &= -\frac{< v_k, A r_k >}{< v_k, A v_k >} = -\frac{< r_k, A v_k >}{< v_k, A v_k >} \\ &= \frac{(1/\alpha_k) < r_k, r_k >}{(1/\alpha_k) < r_{k-1}, r_{k-1} >} = \frac{< r_k, r_k >}{< r_{k-1}, r_{k-1} >}. \end{aligned}$$

**Algorithm 4 (Conjugate Gradient method (CG-method))**

Let $A$ be s.p.d., $b \in \mathbb{R}^n$, choose $x_0 \in \mathbb{R}^n$, $r_0 = b - Ax_0 = v_0$.
If $r_0 = 0$, then $N = 0$ stop, otherwise for $k = 0, 1, \ldots$

(a). $\alpha_k = \frac{<r_k, r_k>}{<v_k, Av_k>}$,

(b). $x_{k+1} = x_k + \alpha_k v_k$,

(c). $r_{k+1} = r_k - \alpha_k A v_k$,

(d). If $r_{k+1} = 0$, let $N = k + 1$, stop.

(e). $\beta_k = \frac{<r_{k+1}, r_{k+1}>}{<r_k, r_k>}$,

(f). $v_{k+1} = r_{k+1} + \beta_k v_k$.

- Theoretically, the exact solution is obtained in $n$ steps.
- If $A$ is well-conditioned, then approximate solution is obtained in about $\sqrt{n}$ steps.
- If $A$ is ill-conditioned, then the number of iterations may be greater than $n$.

Select a nonsingular matrix $C$ so that

$$\tilde{A} = C^{-1}AC^{-T}$$

is better conditioned.
Consider the linear system

$$\tilde{A}\tilde{x} = \tilde{b},$$

where

$$\tilde{x} = C^T x \quad \text{and} \quad \tilde{b} = C^{-1}b.$$

Then

$$\tilde{A}\tilde{x} = (C^{-1}AC^{-T})(C^T x) = C^{-1}Ax.$$

Thus,

$$Ax = b \Leftrightarrow \tilde{A}\tilde{x} = \tilde{b} \text{ and } x = C^{-T}\tilde{x}.$$

Since

$$\tilde{x}_k = C^T x_k,$$

we have

$$\begin{aligned}\tilde{r}_k &= \tilde{b} - \tilde{A}\tilde{x}_k = C^{-1}b - \left(C^{-1}AC^{-T}\right)C^T x_k \\ &= C^{-1}(b - Ax_k) = C^{-1}r_k.\end{aligned}$$

Let

$$\tilde{v}_k = C^T v_k \quad \text{and} \quad w_k = C^{-1}r_k.$$

Then

$$\begin{aligned}\tilde{\beta}_k &= \frac{<\tilde{r}_k, \tilde{r}_k>}{<\tilde{r}_{k-1}, \tilde{r}_{k-1}>} = \frac{<C^{-1}r_k, C^{-1}r_k>}{<C^{-1}r_{k-1}, C^{-1}r_{k-1}>} \\ &= \frac{<w_k, w_k>}{<w_{k-1}, w_{k-1}>}.\end{aligned}$$

Thus,

$$\begin{aligned}\tilde{\alpha}_k &= \frac{<\tilde{r}_{k-1}, \tilde{r}_{k-1}>}{<\tilde{v}_k, \tilde{A}\tilde{v}_k>} = \frac{<C^{-1}r_{k-1}, C^{-1}r_{k-1}>}{<C^T v_k, C^{-1}AC^{-T}C^T v_k>} \\ &= \frac{<w_{k-1}, w_{k-1}>}{<C^T v_k, C^{-1}Av_k>}\end{aligned}$$

and, since

$$\begin{aligned}<C^T v_k, C^{-1}Av_k> &= (v_k)^T CC^{-1}Av_k = (v_k)^T Av_k \\ &= <v_k, Av_k>,\end{aligned}$$

we have

$$\tilde{\alpha}_k = \frac{<w_{k-1}, w_{k-1}>}{<v_k, Av_k>}.$$

Further,

$$\tilde{x}_k = \tilde{x}_{k-1} + \tilde{\alpha}_k \tilde{v}_k, \text{ so } C^T x_k = C^T x_{k-1} + \tilde{\alpha}_k C^T v_k$$

and

$$x_k = x_{k-1} + \tilde{\alpha}_k v_k.$$

Continuing,

$$\tilde{r}_k = \tilde{r}_{k-1} - \tilde{\alpha}_k \tilde{A} \tilde{v}_k,$$

so

$$C^{-1} r_k = C^{-1} r_{k-1} - \tilde{\alpha}_k C^{-1} A C^{-T} C^T v_k$$

and

$$r_k = r_{k-1} - \tilde{\alpha}_k A v_k.$$

Finally,

$$\tilde{v}_{k+1} = \tilde{r}_k + \tilde{\beta}_k \tilde{v}_k \ \text{ and } \ C^T v_{k+1} = C^{-1} r_k + \tilde{\beta}_k C^T v_k,$$

so

$$v_{k+1} = C^{-T} C^{-1} r_k + \tilde{\beta}_k v_k = C^{-T} w_k + \tilde{\beta}_k v_k.$$

## Algorithm 5 (Preconditioned CG-method (PCG-method))

Choose $C$ and $x_0$.
Set $r_0 = b - Ax_0$, solve $Cw_0 = r_0$ and $C^T v_1 = w_0$.
If $r_0 = 0$, then $N = 0$ stop, otherwise for $k = 1, 2, \ldots$

(a). $\alpha_k =< w_{k-1}, w_{k-1} > / < v_k, Av_k >$,
(b). $x_k = x_{k-1} + \alpha_k v_k$,
(c). $r_k = r_{k-1} - \alpha_k Av_k$,
(d). If $r_k = 0$, let $N = k + 1$, stop.
Otherwise, solve $Cw_k = r_k$ and $C^T z_k = w_k$,
(e). $\beta_k =< w_k, w_k > / < w_{k-1}, w_{k-1} >$,
(f). $v_{k+1} = z_k + \beta_k v_k$.