PAM 252 Metode Numerik

Bab 3 Sistem Persamaan Linier

Mahdhivan Syafwan

Jurusan Matematika FMIPA Universitas Andalas

Semester Genap 2016/2017

Bentuk-bentuk khusus matriks persegi

- Matriks simetrik
- Matriks diagonal
- Matriks identitas
- Matriks segitiga atas
- Matriks segitiga bawah
- Matriks tridiagonal
- Matriks Hessenberg (pentadiagonal)

Bentuk umum

- Bentuk umum dari SPL:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & b_n \end{array}$$

- Dalam bentuk matriks, SPL di atas dapat ditulis dengan

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

dimana

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

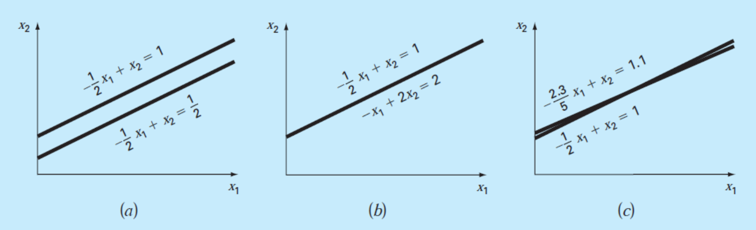
- Bagaimana menentukan solusi untuk $\mathbf{x} = [x_1 \quad x_2 \quad \ldots \quad x_n]^T$?

Tentang solusi SPL

Ada tiga kemungkinan mengenai solusi SPL:

(a) Tidak ada solusi
(b) Tak-hingga solusi
(c) Solusi tunggal

Tafsiran geometris:



Matriks koefisien, matriks lengkap SPL, dan OBE

- Pada persamaan sebelumnya, A disebut matriks koefisien.
- Matriks yang dibentuk oleh matriks A dengan penambahan vektor kolom $\mathbf{b}$ disebut matriks lengkap dari SPL, yaitu

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right].$$

- Operasi baris elementer (OBE):
    - Menukarkan dua buah baris
    - Mengalikan suatu baris dengan suatu konstanta tak-nol
    - Menambahkan k kali baris ke-i pada baris ke-j
- Sifat: *OBE tidak mengubah penyelesaian SPL.*

SPL segitiga atas

- Bentuk umum dari SPL segitiga atas:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_1 & = & b_1 \\ & & a_{22}x_2 & + & \cdots & + & a_{2n}x_2 & = & b_2 \\ & & & & \ddots & & \vdots & & \vdots \\ & & & & & & a_{nn}x_n & = & b_n \end{array}$$

- Matriks lengkap dari SPL segitiga atas:

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ & a_{22} & \cdots & a_{2n} & b_2 \\ & & \ddots & \vdots & \vdots \\ & & & a_{nn} & b_n \end{array}\right].$$

- Sifat: *SPL segitiga atas mempunyai solusi tunggal jika dan hanya jika setiap elemen diagonal dari matriks koefisiennya tidak nol, yaitu* $a_{kk} \neq 0, k = 1, 2, ..., n.$

SPL segitiga atas - contoh

Selesaikan SPL segitiga atas berikut:

$$
\begin{aligned}
4x_1 - x_2 + 2x_3 + 3x_4 &= 20 \\
-2x_2 + 7x_3 - 4x_4 &= -7 \\
6x_3 + 5x_4 &= 4 \\
3x_4 &= 60
\end{aligned}
$$

SPL segitiga atas - substitusi mundur

- Solusi dari SPL segitiga atas secara umum dapat dihitung sebagai berikut:

$$
\begin{aligned}
x_n &= b_n / a_{nn} \\
x_{n-1} &= (b_{n-1} - a_{n-1,n} x_n) / a_{n-1,n-1} \\
x_{n-2} &= (b_{n-2} - (a_{n-2,n-1} x_{n-1} + a_{n-2,n} x_n)) / a_{n-2,n-2} \\
&\vdots \\
x_k &= \left(b_k - \sum_{i=k+1}^{n} a_{ki} x_i \right) / a_{kk} \\
&\vdots \\
x_1 &= \left(b_1 - \sum_{i=2}^{n} a_{1i} x_i \right) / a_{11}
\end{aligned}
$$

- Proses perhitungan di atas dinamakan substitusi mundur, karena ...

Algoritma substitusi mundur

Apa saja yang harus diperhatikan?

- Dalam setiap iterasi, sebelum nilai x_k dihitung, dilakukan pemeriksaan terlebih dahulu terhadap elemen diagonal a_{kk} (proses dihentikan jika ...)
- Misalkan $\tilde{A}$ adalah matriks lengkap. Maka vektor **b** berada pada kolom ke ... dari matriks $\tilde{A}$.

```
Masukan:  n      ukuran SPL
          a[i,j] i := 1,2, ...,n  j := i, i+1 , ...,n, n+1
Keluaran: x[i]   i := 1,2, ...,n    solusi SPL
Langkah-Langkah:
          1. Jika |a[n,n]| < 1E-15 maka 'proses gagal', stop
          2. x[n] := a[n, n+1] / a[n,n]
          3. Untuk k := n-1, n-2, ..., 1
             │ jika |a[k,k]| < 1E-15 maka '   ses gagal', stop
             │ s := 0
             │ Untuk i := k+1, k+2, ..., n
             │ └s := s + a[k,i] * x[i]
             └x[k] := (a[k,n+1] - s) / a[k,k]
```

Algoritma substitusi maju ? (pada SPL segitiga bawah)

Metode eliminasi Gauss - contoh

Dengan menggunakan metode eliminasi Gauss, selesaikan SPL berikut ini:

$$\begin{array}{rcrcrcl} -x_1 & + & x_2 & + & 2x_3 & = & 1, \\ 3x_1 & - & x_2 & + & x_3 & = & 1, \\ -x_1 & + & 3x_2 & + & 4x_3 & = & 1. \end{array}$$

Dua tahap besar pada metode eliminasi Gauss

1. Tahap eliminasi (maju), yaitu mengubah SPL semula menjadi SPL segitiga atas melalui serangkaian OBE (operasi ini tidak mengubah solusi dari SPL semula).

2. Tahap substitusi mundur, yaitu menyelesaikan SPL segitiga atas yang terbentuk.

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right] \Big\downarrow \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ & a'_{22} & a'_{23} & b'_2 \\ & & a''_{33} & b''_3 \end{array}\right]$$

(*a*) Forward elimination

$$\Big\downarrow$$

$$x_3 = b''_3/a''_{33}$$

$$x_2 = (b'_2 - a'_{23}x_3)/a'_{22}$$

$$x_1 = (b_1 - a_{13}x_3 - a_{12}x_2)/a_{11}$$

(*b*) Back substitution

Metode eliminasi Gauss - tahap eliminasi

Langkah pertama:
membuat agar elemen-elemen kolom pertama mulai baris ke-$2, 3, ..., n$ (yaitu $a_{21}, a_{31}, ..., a_{n1}$) menjadi nol.

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & \cdots & a_{2n} & a_{2,n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & a_{n,n+1} \end{array}\right] \sim \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ 0 & a_{22} & \cdots & a_{2n} & a_{2,n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2} & \cdots & a_{nn} & a_{n,n+1} \end{array}\right]$$

Catatan:

- Notasi $\sim$ menyatakan bahwa proses yang dilakukan adalah melalui serangkaian OBE.
- Elemen-elemen pada kedua matriks lengkap di atas menggunakan notasi yang sama, yaitu a_{ij}. Hal ini tidak berarti bahwa nilainya juga sama. Pemakaian notasi yang sama ini ditujukan untuk keperluan pada pemrograman komputer.

Metode eliminasi Gauss - tahap eliminasi

Langkah pertama - ilustrasi

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & \cdots & a_{2n} & a_{2,n+1} \\ a_{31} & a_{32} & \cdots & a_{3n} & a_{3,n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & a_{n,n+1} \end{array}\right] \quad (b)_2 \leftarrow (b)_2 - \frac{a_{21}}{a_{11}}(b)_1 \quad \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ 0 & a_{22} & \cdots & a_{2n} & a_{2,n+1} \\ a_{31} & a_{32} & \cdots & a_{3n} & a_{3,n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & a_{n,n+1} \end{array}\right]$$

$$(b)_3 \leftarrow (b)_3 - \frac{a_{31}}{a_{11}}(b)_1 \quad \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ 0 & a_{22} & \cdots & a_{2n} & a_{2,n+1} \\ 0 & a_{32} & \cdots & a_{3n} & a_{3,n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & a_{n,n+1} \end{array}\right]$$

$$\vdots$$

$$(b)_n \leftarrow (b)_n - \frac{a_{n1}}{a_{11}}(b)_1 \quad \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ 0 & a_{22} & \cdots & a_{2n} & a_{2,n+1} \\ 0 & a_{32} & \cdots & a_{3n} & a_{3,n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2} & \cdots & a_{nn} & a_{n,n+1} \end{array}\right]$$

Metode eliminasi Gauss - tahap eliminasi

Langkah pertama - algoritma

$$(b)_2 \leftarrow (b)_2 - \frac{a_{21}}{a_{11}}(b)_1$$
$$(b)_3 \leftarrow (b)_3 - \frac{a_{31}}{a_{11}}(b)_1$$
$$\vdots$$
$$(b)_n \leftarrow (b)_n - \frac{a_{n1}}{a_{11}}(b)_1$$

```
untuk i=2,3,…,n
  p:=a[i,1]/a[1,1]
  untuk j=1,2,…,n+1
    a[i,j]:=a[i,j]-p*a[1,j]
```

Metode eliminasi Gauss - tahap eliminasi

Langkah kedua: mengeliminasi kolom kedua dari matriks lengkap SPL.

$$\left[\begin{array}{cccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & & a_{1,n+1} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} & & a_{2,n+1} \\ 0 & a_{32} & a_{33} & \cdots & a_{3n} & & a_{3,n+1} \\ \vdots & \vdots & & \vdots & \vdots & & \\ 0 & a_{n2} & a_{n3} & \cdots & a_{nn} & & a_{n,n+1} \end{array}\right] \begin{array}{c} (b)_3 \leftarrow (b)_3 - \frac{a_{32}}{a_{22}}(b)_2 \\ (b)_4 \leftarrow (b)_4 - \frac{a_{42}}{a_{22}}(b)_2 \\ \vdots \\ (b)_n \leftarrow (b)_n - \frac{a_{n2}}{a_{22}}(b)_2 \end{array} \left[\begin{array}{cccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & & a_{1,n+1} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} & & a_{2,n+1} \\ 0 & 0 & a_{33} & \cdots & a_{3n} & & a_{3,n+1} \\ \vdots & \vdots & & \vdots & \vdots & & \\ 0 & 0 & a_{n3} & \cdots & a_{nn} & & a_{n,n+1} \end{array}\right]$$

```
untuk i=3,4,…,n
  p:=a[i,2]/a[2,2]
  untuk j=2,3,…,n+1
    a[i,j]:=a[i,j]-p*a[2,j]
```

Langkah ke-$3, 4, \ldots, n-1$: mengeliminasi kolom ke-$3, 4, \ldots, n-1$ dari matriks lengkap SPL.

Hasil akhir dari tahap eliminasi adalah suatu SPL segitiga atas. Solusi SPL dapat diperoleh dengan menjalankan algoritma substitusi mundur.

Metode eliminasi Gauss - algoritma tahap eliminasi

```
untuk i=2,3,…,n
  p:=a[i,1]/a[1,1]
  untuk j=1,2,…,n+1
    a[i,j]:=a[i,j]-p*a[1,j]

untuk i=3,4,…,n
  p:=a[i,2]/a[2,2]
  untuk j=2,3,…,n+1
    a[i,j]:=a[i,j]-p*a[2,j]

  ⋮

untuk i=n
  p:=a[i,n-1]/a[n-1,n-1]
  untuk j=n-1,n,n+1
    a[i,j]:=a[i,j]-p*a[n-1,j]
```

$\rightarrow$

```
untuk k=2,3,…,n
  jika abs(a[k-1,k-1])<1e-15
    maka "proses gagal", stop
  untuk i=k,k+1,…,n
    p:=a[i,k-1]/a[k-1,k-1]
    untuk j=k-1,k,…,n+1
      a[i,j]:=a[i,j]-p*a[k-1,j]
```

Catatan: elemen pembagi pada tahap eliminasi, yaitu $a[k,k]$ dinamakan elemen penumpu (pivot).

Algoritma metode eliminasi Gauss

```
Masukan:    n        ukuran SPL
            a[i,j]   i=1,2,…,n; j=1,2,…,n+1   elemen matriks lengkap
Keluaran:   x[i]     i=1,2,…,n   solusi SPL

Langkah-Langkah:

     1. (*tahap eliminasi*)
        untuk k=2,3,…,n
        | jika abs(a[k-1,k-1])<1e-15
        |    maka "proses gagal", stop
        | untuk i=k,k+1,…,n
        | | p:=a[i,k-1]/a[k-1,k-1]
        | | untuk j=k-1,k,…,n+1
        |_|_|__ a[i,j]:=a[i,j]-p*a[k-1,j]

     2. (*tahap substitusi mundur*)
        jika abs(a[n,n])<1e-15 maka "proses gagal", stop
        x[n]:=a[n,n+1]/a[n,n]
        untuk k:=n-1,n-2,…,1
        |  s:=0
        |  untuk i=k+1,k+2,…,n
        |  |__s:=s+a[k,i]*x[i]
        |__x[k]:=(a[k,n+1]-s)/a[k,k]
```

Kelemahan metode eliminasi Gauss & perbaikannya

Kelemahan metode eliminasi Gauss:

- Proses eliminasi tidak dapat dilakukan jika elemen penumpu (pivot) bernilai nol.
- Jika nilai mutlak dari elemen pivot sangat kecil, maka pada realisasi komputer akan menimbulkan perambatan galat pembulatan yang besar.

Cara memperbaikinya?

- Perlu dipilih elemen penumpu yang nilai mutlaknya besar.
- Hal ini direalisasikan dengan melakukan pertukaran baris dan/atau kolom pada matriks lengkap.
- Pertukaran baris tidak mengubah solusi SPL.
- Pertukaran kolom bagaimana?

Teknik pemilihan elemen penumpu ini dinamakan teknik penumpuan (pivoting).

Beberapa macam teknik penumpuan

- Penumpuan total
  - Elemen penumpu diambil dari $\max\limits_{k \leq i,j \leq n} |a_{ij}|$
  - Memerlukan pertukaran baris dan/atau kolom
- Penumpuan parsial
  - Elemen penumpu diambil dari $\max\limits_{k \leq i \leq n} |a_{ik}|$
  - Hanya memerlukan pertukaran baris saja
- Penumpuan parsial terskala
  - Elemen penumpu diambil dari $\max\limits_{k \leq i \leq n} |a_{ik}/a_{kk}|$
  - Hanya memerlukan pertukaran baris saja

Elemen penumpu yang dipilih kemudian ditempatkan pada posisi (k, k) dari matriks lengkap SPL.

Eliminasi Gauss dengan penumpuan parsial - contoh

$$\begin{bmatrix} 1 & 2 & 1 & 4 & 13 \\ 2 & 0 & 4 & 3 & 28 \\ \boxed{4} & 2 & 2 & 1 & 20 \\ -3 & 1 & 3 & 2 & 6 \end{bmatrix} \quad b_1 \longleftrightarrow b_3 \quad \begin{bmatrix} 4 & 2 & 2 & 1 & 20 \\ 2 & 0 & 4 & 3 & 28 \\ 1 & 2 & 1 & 4 & 13 \\ -3 & 1 & 3 & 2 & 6 \end{bmatrix} \quad \begin{array}{l} b_2 \leftarrow b_2 - \frac{2}{4} b_1 \\ b_3 \leftarrow b_3 - \frac{1}{4} b_1 \\ b_4 \leftarrow b_4 - \frac{-3}{4} b_1 \end{array}$$

$$\begin{bmatrix} 4 & 2 & 2 & 1 & 20 \\ 0 & -1 & 3 & 2.5 & 18 \\ 0 & 1.5 & 0.5 & 3.75 & 8 \\ 0 & \boxed{2.5} & 4.5 & 2.75 & 21 \end{bmatrix} \quad b_2 \longleftrightarrow b_4 \quad \begin{bmatrix} 4 & 2 & 2 & 1 & 20 \\ 0 & 2.5 & 4.5 & 2.75 & 21 \\ 0 & 1.5 & 0.5 & 3.75 & 8 \\ 0 & -1 & 3 & 2.5 & 18 \end{bmatrix} \quad \begin{array}{l} b_3 \leftarrow b_3 - \frac{1.5}{2.5} b_2 \\ b_4 \leftarrow b_4 - \frac{-1}{2.5} b_2 \end{array}$$

$$\begin{bmatrix} 4 & 2 & 2 & 1 & 20 \\ 0 & 2.5 & 4.5 & 2.75 & 21 \\ 0 & 0 & -2.2 & 2.1 & -4.6 \\ 0 & 0 & \boxed{4.8} & 3.6 & 26.4 \end{bmatrix} \quad b_3 \longleftrightarrow b_4 \quad \begin{bmatrix} 4 & 2 & 2 & 1 & 20 \\ 0 & 2.5 & 4.5 & 2.75 & 21 \\ 0 & 0 & 4.8 & 3.6 & 26.4 \\ 0 & 0 & -2.2 & 2.1 & -4.6 \end{bmatrix} \quad b_4 \leftarrow b_4 - \frac{-2.2}{4.8} b_3$$

$$\begin{bmatrix} 4 & 2 & 2 & 1 & 20 \\ 0 & 2.5 & 4.5 & 2.75 & 21 \\ 0 & 0 & 4.8 & 3.6 & 26.4 \\ 0 & 0 & 0 & 3.75 & 7.5 \end{bmatrix}$$

$$x_4 := 7.5/3.75 = 2$$
$$x_3 := (26.4 - 2 * 3.6)/4.8 = 4$$
$$x_2 := (21 - 4.5 * 4 - 2.75 * 2)/2.5 = -1$$
$$x_4 := (20 - 2 * (-1) - 2 * 4 - 1 * 2)/4 = 3$$

Eliminasi Gauss dengan penumpuan parsial - algoritma?

```
Masukan:   n        ukuran SPL
           a[i,j]   i=1,2,…,n; j=1,2,…,n+1   elemen matriks lengkap

Keluaran:  x[i]     i=1,2,…,n   solusi SPL

Langkah-Langkah:

     1. (*tahap eliminasi*)
        untuk k=2,3,…,n
        |  1. Pencarian elemen penumpu dan posisinya?
        |  2. Pertukaran baris jika posisi elemen
        |     penumpu berubah?
        |  jika abs(a[k-1,k-1])<1e-15
        |     maka "proses gagal", stop
        |  untuk i=k,k+1,…,n
        |  |  p:=a[i,k-1]/a[k-1,k-1]
        |  |  untuk j=k-1,k,…,n+1
        |__|______ a[i,j]:=a[i,j]-p*a[k-1,j]

     2. (*tahap substitusi mundur*)
        jika abs(a[n,n])<1e-15 maka "proses gagal", stop
        x[n]:=a[n,n+1]/a[n,n]
        untuk k:=n-1,n-2,…,1
        |  s:=0
        |  untuk i=k+1,k+2,…,n
        |  |__s:=s+a[k,i]*x[i]
        |__x[k]:=(a[k,n+1]-s)/a[k,k]
```

Eliminasi Gauss dengan penumpuan parsial - algoritma

```
Masukan:    n        ukuran SPL
            a[i,j]   i=1,2,…,n; j=1,2,…,n+1   elemen matriks lengkap

Keluaran:   x[i]     i=1,2,…,n   solusi SPL

Langkah-Langkah:

        1. (*tahap eliminasi*)
           untuk k=2,3,…,n
           |  m:=k-1
           |  untuk i=k,k+1,…,n
           |  └─jika abs(a[i,k-1])>abs(a[m,k-1]) maka m:=i
           |  jika m~=k-1
           |    maka untuk j=k-1,k,…,n+1
           |            s:=a[k-1,j]
           |            a[k-1,j]:=a[m,j]
           |            a[m,j]:=s
           |
           |  jika abs(a[k-1,k-1])<1e-15
           |     maka "proses gagal", stop
           |  untuk i=k,k+1,…,n
           |  |  p:=a[i,k-1]/a[k-1,k-1]
           |  |  untuk j=k-1,k,…,n+1
           └──└──└──a[i,j]:=a[i,j]-p*a[k-1,j]

        2. (*tahap substitusi mundur*)
           jika abs(a[n,n])<1e-15 maka "proses gagal", stop
           x[n]:=a[n,n+1]/a[n,n]
           untuk k:=n-1,n-2,…,1
           |  s:=0
           |  untuk i=k+1,k+2,…,n
           |  └─s:=s+a[k,i]*x[i]
           └─x[k]:=(a[k,n+1]-s)/a[k,k]
```

Beberapa SPL dengan matriks koefisien sama

Pandang dua SPL berikut:

$$\begin{cases} x_1 + 2x_2 + x_3 + 4x_4 = 13 \\ 2x_1 + 4x_3 + 3x_4 = 28 \\ 4x_1 + 2x_2 + 2x_3 + x_4 = 20 \\ -3x_1 + x_2 + 3x_3 + 2x_4 = 0 \end{cases}$$

$$\begin{cases} x_1 + 2x_2 + x_3 + 4x_4 = 8 \\ 2x_1 + 4x_3 + 3x_4 = 9 \\ 4x_1 + 2x_2 + 2x_3 + x_4 = 9 \\ -3x_1 + x_2 + 3x_3 + 2x_4 = 3 \end{cases}$$

Matriks lengkap dari dua SPL tersebut dapat ditulis:

$$\left[\begin{array}{cccc|cc} 1 & 2 & 1 & 4 & 13 & 8 \\ 2 & 0 & 4 & 3 & 28 & 9 \\ 4 & 2 & 2 & 1 & 20 & 9 \\ -3 & 1 & 3 & 2 & 6 & 3 \end{array}\right].$$

Solusi dari masing-masing SPL dapat ditentukan dengan metode eliminasi Gauss.

Kuis

Diberikan SPL ukuran $n \times n$. Tuliskan algoritma untuk mengeliminasi x_n pada persamaan ke-1 sampai $(n-1)$ dengan elemen penumpu $a_{n,n}$ (lihat ilustrasi berikut):

$$\begin{bmatrix} x & x & x & \cdots & x & x \\ x & x & x & \cdots & x & x \\ x & x & x & \cdots & x & x \\ & & & \vdots & & \\ x & x & x & \cdots & x & x \\ x & x & x & \cdots & x & x \end{bmatrix} \sim \begin{bmatrix} x & x & x & \cdots & 0 & x \\ x & x & x & \cdots & 0 & x \\ x & x & x & \cdots & 0 & x \\ & & & \vdots & & \\ x & x & x & \cdots & 0 & x \\ x & x & x & \cdots & x & x \end{bmatrix}$$

Perhitungan determinan - dasar teori

Teorema (determinan matriks segitiga atas)

Jika A matriks segitiga atas berukuran $n \times n$, maka $\det(A) = \prod_{i=1}^{n} a_{ii}$.

Bukti. Lakukan ekspansi kofaktor berkali-kali sepanjang baris terakhir. ■

Teorema (pengaruh OBE terhadap nilai determinan suatu matriks)

Misalkan A matriks berukuran $n \times n$.

- Jika B adalah matriks hasil dari perkalian suatu baris (kolom) matriks A dengan konstanta k, maka $\det(B) = k \det(A)$.
- Jika B adalah matriks hasil dari pertukaran dua baris (kolom) matriks A, maka $\det(B) = -\det(A)$.
- Jika B adalah matriks hasil penambahan k kali baris (kolom) ke baris (kolom) lain dari matriks A, maka $\det(B) = \det(A)$.

Bukti. ...

Perhitungan determinan - penerapan, contoh, & algoritma

Untuk menghitung determinan suatu matriks, lakukan serangkaian OBE terhadap matriks tersebut sedemikian sehingga menjadi matriks segitiga atas. Perhatikan perubahan nilai determinan selama melakukan OBE.

Contoh. Dengan menggunakan metode eliminasi Gauss dengan penumpuan parsial, tentukan determinan dari

$$\begin{bmatrix} 1 & 2 & 1 & 4 \\ 2 & 0 & 4 & 3 \\ 4 & 2 & 2 & 1 \\ -3 & 1 & 3 & 2 \end{bmatrix}.$$

Algoritmanya?

Perhitungan determinan - algoritma

```
Masukan:   n        ukuran matriks
           a[i,j]   i=1,2,…,n; j=1,2,…,n   elemen-elemen matriks

Keluaran:  d        nilai determinan matriks

Langkah-Langkah:

        1. f:=0
        2. d:=1
        3. (*tahap eliminasi dengan pivoting*)
           untuk k=2,3,…,n
           |  m:=k-1
           |  untuk i=k,k+1,…,n
           |  └jika abs(a[i,k-1])>abs(a[m,k-1]) maka m:=i
           |  jika m~=k-1
           |    maka f:=f+1
           |         untuk j=k-1,k,…,n
           |         | s:=a[k-1,j]
           |         | a[k-1,j]:=a[m,j]
           |         └a[m,j]:=s
           |
           |  jika abs(a[k-1,k-1])<1e-15
           |     maka “proses gagal”, stop
           |   untuk i=k,k+1,…,n
           |   |  p:=a[i,k-1]/a[k-1,k-1]
           |   |  untuk j=k-1,k,…,n
           |   └──└══ a[i,j]:=a[i,j]-p*a[k-1,j]
           └──── d:=d*a[k-1,k-1];
        4. d:=(-1)^f*d*a[n,n];
```

Perhitungan invers

Gunakan metode eliminasi Gauss-Jordan (eliminasi maju dan mundur):

$$[A \mid I] \sim \cdots \sim [I \mid A^{-1}]. \quad \text{[justifikasi!]}$$

Contoh.

$$\left(\begin{array}{rrr|rrr} 1 & 0 & 2 & 1 & 0 & 0 \\ 2 & -1 & 3 & 0 & 1 & 0 \\ 4 & 1 & 8 & 0 & 0 & 1 \end{array}\right) \to \left(\begin{array}{rrr|rrr} 1 & 0 & 2 & 1 & 0 & 0 \\ 0 & -1 & -1 & -2 & 1 & 0 \\ 0 & 1 & 0 & -4 & 0 & 1 \end{array}\right)$$

$$\to \left(\begin{array}{rrr|rrr} 1 & 0 & 2 & 1 & 0 & 0 \\ 0 & -1 & -1 & -2 & 1 & 0 \\ 0 & 0 & -1 & -6 & 1 & 1 \end{array}\right)$$

$$\to \left(\begin{array}{rrr|rrr} 1 & 0 & 0 & -11 & 2 & 2 \\ 0 & -1 & 0 & 4 & 0 & -1 \\ 0 & 0 & -1 & -6 & 1 & 1 \end{array}\right) \to \left(\begin{array}{rrr|rrr} 1 & 0 & 0 & -11 & 2 & 2 \\ 0 & 1 & 0 & -4 & 0 & 1 \\ 0 & 0 & 1 & 6 & -1 & -1 \end{array}\right)$$

Algoritmanya?

Perhitungan invers - algoritma

```
Masukan:   n        ukuran matriks
           a[i,j]   i=1,2,…,n; j=1,2,…,n   elemen matriks

Keluaran:  b[i,j]   i=1,2,…,n; j=1,2,…,n   elemen invers matriks

Langkah-Langkah:
   1. (*tahap menggandengkan matriks identitas*)
      untuk i=1,2,…,n
        untuk j=n+1,n+2,…,2*n
          jika i=j+n maka a[i,j]:=1
          jikatidak a[i,j]:=0

   2. (*tahap eliminasi maju dengan pivoting*)
      untuk k=2,3,…,n
        m:=k-1
        untuk i=k,k+1,…,n
          jika abs(a[i,k-1])>abs(a[m,k-1]) maka m:=i
        jika m~=k-1
              untuk j=k-1,k,…,2*n
                s:=a[k-1,j]
                a[k-1,j]:=a[m,j]
                a[m,j]:=s

        jika abs(a[k-1,k-1])<1e-15
           maka "proses gagal", stop
        untuk i=k,k+1,…,n
          p:=a[i,k-1]/a[k-1,k-1]
          untuk j=k-1,k,…,2*n
            a[i,j]:=a[i,j]-p*a[k-1,j]

   3. (*tahap eliminasi mundur*)
      untuk k=n-1,n-2,…,1
        jika abs(a[k+1,k+1])<1e-15
            maka "proses gagal", stop
        untuk i=k,k-1,…,1
          p:=a[i,k+1]/a[k+1,
          untuk j=1,2,…,k+1
            a[i,j]:=a[i,j]   +1,j]
          untuk j=n+1,n+2,…,2*n
            a[i,j]:=a[i,j]-p*a[k+1,j]

   4. (*tahap mendapatkan invers matriks*)
      untuk i=1,2,…,n
        untuk j=n+1,n+2,…,2*n
          b[i,j-n]:=a[i,j]/a[i,i]
```

Modifikasi eliminasi Gauss untuk SPL tridiagonal

Perhatikan matriks SPL tridiagonal berikut:

$$\begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}.$$

Pada SPL tersebut, banyak sekali koefisiennya yang bernilai nol.

Bagaimana algoritma yang paling efisien untuk mencari solusi SPL tersebut? $\longrightarrow$ TUGAS BACA!

Definisi faktorisasi LU dan kegunaannya

Definisi (Faktorisasi LU/Segitiga)

Matriks nonsingular A dikatakan mempunyai faktorisasi LU (juga dikenal dengan faktorisasi segitiga) jika ia dapat ditulis sebagai perkalian matriks segitiga bawah L dan matriks segitiga atas U, yaitu

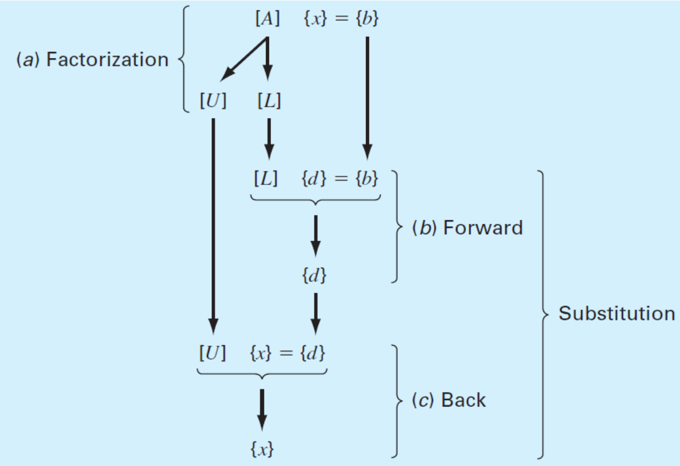
$$A = LU.$$

Misalkan matriks koefisien A dari SPL $A\mathbf{x} = \mathbf{b}$ mempunyai faktorisasi LU. Maka

$$A\mathbf{x} = \mathbf{b} \Leftrightarrow (LU)\mathbf{x} = \mathbf{b} \Leftrightarrow L(U\mathbf{x}) = \mathbf{b}.$$

Sekarang misalkan $\mathbf{d} = U\mathbf{x}$. SPL segitiga bawah $L\mathbf{d} = \mathbf{b}$ dapat diselesaikan dengan substitusi maju. Setelah $\mathbf{d}$ diperoleh, solusi $\mathbf{x}$ dapat dicari dari SPL segitiga atas $U\mathbf{x} = \mathbf{d}$ dengan substitusi mundur.

Ilustrasi


(a) Factorization
[A] {x} = {b}
[U] [L]
[L] {d} = {b}
(b) Forward
{d}
Substitution
[U] {x} = {d}
(c) Back
{x}

Beberapa jenis faktorisasi LU

- Secara umum faktorisasi LU tidak tunggal.
- Agar hasilnya tunggal, biasanya dilakukan dengan memilih matriks L dan U yang memiliki sifat tertentu.
- Beberapa faktorisasi LU yang dikenal:
  - Faktorisasi/dekomposisi Doolitle, yaitu elemen diagonal utama matriks L dipilih bernilai 1.
  - Faktorisasi/dekomposisi Crout, yaitu elemen diagonal utama matriks U dipilih bernilai 1.
  - Faktorisasi/dekomposisi Cholesky, yaitu matriks U dibuat sama dengan L^T jika A matriks simetris.

Faktorisasi LU Doolitle dengan eliminasi Gauss - proses

- Misalkan dari tahap eliminasi pada eliminasi Gauss diperoleh

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \sim \cdots \sim \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & a_{nn}^{(n-1)} \end{bmatrix} = U,$$

  dimana $a_{ij}^{(k)}$ menyatakan elemen matriks A pada posisi (i,j) yang nilainya merupakan hasil dari OBE pada iterasi ke-k.
- Meskipun tidak muncul secara langsung, matriks L juga dihasilkan dari proses eliminasi ini, yaitu diberikan oleh

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix},$$

  dimana $l_{ij} = a_{ij}^{(j-1)}/a_{jj}^{(j-1)}$ dan $a_{i1}^{(0)} = a_{i1}$ [periksa!].

Faktorisasi *LU* Doolitle dengan eliminasi Gauss - algoritma

(TANPA PIVOTING)

```
Masukan:   n        ukuran SPL
           a[i,j]   i=1,2,…,n; j=1,2,…,n+1   elemen matriks lengkap

Keluaran:  x[i]     i=1,2,…,n   solusi SPL

Langkah-Langkah:
```

```
1. (*membuat matriks identitas sementara untuk
   matriks L*)
   untuk i=1,2,…,n
   | untuk j=1,2,…,n
   | | jika i=j maka l[i,j]:=1
   |___jikatidak l[i,j]:=0

2. (*tahap eliminasi untuk memperoleh matriks U
   dan L*)
   untuk k=2,3,…,n
   | jika abs(a[k-1,k-1])<1e-15
   |    maka "proses gagal", stop
   | untuk i=k,k+1,…,n
   | | p:=a[i,k-1]/a[k-1,k-1]
   | | l[i,k-1]:=p
   | | untuk j=k-1,k,…,n
   |___|___ a[i,j]:=a[i,j]-p*a[k-1,j]
   (*a[i,j] sekarang adalah elemen matriks U*)
```

```
3. (*tahap substitusi maju untuk memperoleh solusi d*)
   d[1]:=a[1,n+1]
   untuk k:=2,3,…,n
   | s:=0
   | untuk i=1,2,…,k
   | └─s:=s+l[k,i]*d[i]
   └── d[k]:=a[k,n+1]-s

4. (*tahap substitusi mundur untuk memperoleh solusi
   x*)
   jika abs(a[n,n])<1e-15 maka "proses gagal", stop
   x[n]:=d[n]/a[n,n]
   untuk k:=n-1,n-2,…,1
   | s:=0
   | untuk i=k+1,k+2,…,n
   | └── s:=s+a[k,i]*x[i]
   └── x[k]:=(d[k]-s)/a[k,k]
```

Perhitungan invers matriks - dasar teori & penerapan

Diberikan sistem

$$A\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad A\mathbf{x}_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots \quad , A\mathbf{x}_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix},$$

dimana A adalah matriks berukuran $n \times n$ dan $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ adalah vektor-vektor berukuran $n \times 1$. Jika A dapat diinverskan, maka

$$A^{-1} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \ldots \quad \mathbf{x}_n]. \quad [\text{tunjukkan!}]$$

Solusi $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ dapat ditentukan dengan menggunakan faktorisasi LU. Karena sistem di atas memiliki matriks koefisien yang sama, maka matriks L dan U cukup dihitung sekali.

Perhitungan invers matriks - algoritma

(TANPA PIVOTING)

```
Masukan:  n        ukuran SPL
          a[i,j]   i=1,2,…,n; j=1,2,…,n   elemen matriks A
Keluaran: x[i,j]   i=1,2,…,n; j=1,2,…,n   elemen invers matriks A
```

Langkah-Langkah:

```
1. (*membuat matriks identitas sementara untuk
   matriks L*)
   untuk i=1,2,…,n
     untuk j=1,2,…,n
       jika i=j maka l[i,j]:=1
       jikatidak l[i,j]:=0

2. (*tahap eliminasi untuk memperoleh matriks U
   dan L*)
   untuk k=2,3,…,n
     jika abs(a[k-1,k-1])<1e-15
        maka "proses gagal", stop
     untuk i=k,k+1,…,n
        p:=a[i,k-1]/a[k-1,k-1]
        l[i,k-1]:=p
        untuk j=k-1,k,…,n
           a[i,j]:=a[i,j]-p*a[k-1,j]
   (*a[i,j] sekarang adalah elemen matriks U*)

3. (*tahap substitusi maju dan mundur untuk memperoleh
   invers matriks A dalam x[i,j]*)

   untuk j=1,2,…,n
     untuk k=1,…,j-1
       d[k]:=0
     d[j]=1
     untuk k:=j+1,j+2,…,n
       s:=0
       untuk i=j,j+1,…,k-1
         s:=s+l[k,i]*d[i]
       d[k]:=-s

     jika abs(a[n,n])<1e-15 maka "proses gagal", stop
     x[n,j]:=d[n]/a[n,n]
     untuk k:=n-1,n-2,…,1
       s:=0
       untuk i=k+1,k+2,…,n
         s:=s+a[k,i]*x[i,j]
       x[k,j]:=(d[k]-s)/a[k,k]
```

Contoh 1

Tentukan matriks dekomposisi LU yang memenuhi

$$A = \begin{bmatrix} 1 & 3 & 6 \\ 4 & 8 & -1 \\ -2 & 3 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} = LU$$

Contoh 2

Problem Statement. Employ LU decomposition to determine the matrix inverse for the matrix

$$[A] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}$$

Recall that the decomposition resulted in the following lower and upper triangular matrices:

$$[U] = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0 & 7.00333 & -0.293333 \\ 0 & 0 & 10.0120 \end{bmatrix} \qquad [L] = \begin{bmatrix} 1 & 0 & 0 \\ 0.0333333 & 1 & 0 \\ 0.100000 & -0.0271300 & 1 \end{bmatrix}$$

Metode iterasi?

- Alternatif metode untuk menyelesaikan SPL (dan juga SPNL).
- Metode iteratif dimulai dengan sebuah tebakan awal, kemudian digunakan suatu metode sistematis untuk memperoleh barisan yang diharapkan konvergen ke solusi yang ingin dicari.
- Metode iteratif untuk SPL: metode Jacobi dan metode Gauss-Seidel.
- Metode iteratif untuk SPNL: metode substitusi berturutan dan metode Newton-Raphson (kasus multivariat) [tidak dipelajari].

Ilustrasi

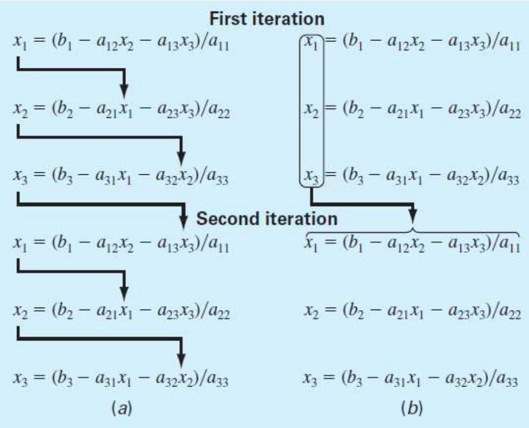


Figure : (a) Metode Gauss-Seidel, (b) Metode Jacobi

Metode Jacobi

- Diberikan SPL berikut:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & b_n \end{array}$$

- Rumus iterasi dari metode Jacobi:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{(k)} \right) / a_{ii}, \quad i = 1, 2, ..., n.$$

Catatan: indeks (k) menyatakan langkah iterasi.

- Ambil tebakan awal $x = \begin{bmatrix} x_1^{(0)} & x_2^{(0)} & \cdots & x_n^{(0)} \end{bmatrix}^T$.
- Kriteria penghentian iterasi: $\max\limits_{1\leq i\leq n} \left| x_i^{(k+1)} - x_i^{(k)} \right| < \epsilon$.

Algoritma metode Jacobi

```
Masukan:   n        ukuran SPL
           a[i,j]   i=1,2,…,n; j=1,2,…,n   elemen matriks koefisien
           b[i]     i=1,2,…,n  nilai SPL
           x[i]     i=1,2,…,n  tebakan awal
           epsl     batas galat
           maks     jumlah maksimum iterasi

Keluaran:  x[i]     i=1,2,…,n   solusi SPL

Langkah-Langkah:
        1. iter:=1
        2. galat:=0
        3. untuk i=1,2,…,n
           | s:=0
           | untuk j=1,2,…n
           | |   jika j~=i
           | └─── maka s:=s+a[i,j]*x[j]
           | xbaru[i]:=(b[i]-s)/a[i,i]
           | p:=abs(xbaru[i]-x[i])
           └─jika p>galat maka galat:=p
        4. untuk i=1,2,…n
           └─x[i]:=xbaru[i]
        5. jika galat < epsl
              maka "proses selesai"
        6. iter=iter+1
        7. jika iter>maks
              maka "proses belum konvergen", stop
        8. kembali ke langkah 2
```

```
2. galat:=epsl+1
3. selagi galat >= epsl dan iter <= maks
   galat:=0
     untuk i=1,2,…,n
     | s:=0
     | untuk j=1,2,…n
     | |  jika j~=i
     | └─── maka s:=s+a[i,j]*x[j]
     | xbaru[i]:=(b[i]-s)/a[i,i]
     | p:=abs(xbaru[i]-x[i])
     └─jika p>galat maka galat:=p
    untuk i=1,2,…n
     └─x[i]:=xbaru[i]
    iter:=iter+1
4. jika iter > maks
     maka "proses belum konvergen"
```

Metode Gauss-Seidel

- Diberikan SPL berikut:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & b_n \end{array}$$

- Rumus iterasi dari metode Gauss-Seidel:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) / a_{ii}, \quad i = 1, 2, ..., n.$$

Catatan: indeks (k) menyatakan langkah iterasi.

- Ambil tebakan awal $x = \begin{bmatrix} x_1^{(0)} & x_2^{(0)} & \cdots & x_n^{(0)} \end{bmatrix}^T$.
- Kriteria penghentian iterasi: $\max_{1 \le i \le n} \left| x_i^{(k+1)} - x_i^{(k)} \right| < \epsilon$.

Algoritma metode Gauss-Seidel

```
Masukan:   n        ukuran SPL
           a[i,j]   i=1,2,…,n; j=1,2,…,n   elemen matriks koefisien
           b[i]     i=1,2,…,n  nilai SPL
           x[i]     i=1,2,…,n  tebakan awal
           epsl     batas galat
           maks     jumlah maksimum iterasi

Keluaran:  x[i]     i=1,2,…,n   solusi SPL

Langkah-Langkah:
       1. iter:=1
       2. galat:=0
       3. untuk i=1,2,…,n
          |  s1:=0
          |  untuk j=1,2,…i-1
          |  └──s1:=s1+a[i,j]*x[j]
          |  s2:=0
          |  untuk j=i+1,i+2,…n
          |  └──s2:=s2+a[i,j]*x[j]
          |  xbaru[i]:=(b[i]-s1-s2)/a[i,i]
          |  p:=abs(xbaru[i]-x[i])
          |  jika p>galat maka galat:=p
          └─x[i]:=xbaru[i]
       4. jika galat < epsl
             maka "proses selesai"
       5. iter=iter+1
       6. jika iter>maks
             maka "proses belum konvergen", stop
       7. kembali ke langkah 2
```

```
2. galat:=epsl+1
3. selagi galat >= epsl dan iter <= maks
   galat:=0
     untuk i=1,2,…,n
     |  s1:=0
     |  untuk j=1,2,…i-1
     |  └── s1:=s1+a[i,j]*x[j]
     |  s2:=0
     |  untuk j=i+1,i+2,…n
     |  └── s2:=s2+a[i,j]*x[j]
     |   xbaru[i]:=(b[i]-s1-s2)/a[i,i]
     |   p:=abs(xbaru[i]-x[i])
     |  jika p>galat maka galat:=p
     └─x[i]:=xbaru[i]
    iter:=iter+1
4. jika iter > maks
     maka "proses belum konvergen"
```

Kekonvergenan metode Jacobi dan Gauss-Seidel

- Metode Jacobi dan Gauss-Seidel tidak selalu konvergen.
- Syarat cukup agar kedua metode tersebut konvergen adalah matriks koefisien A bersifat dominan kuat secara diagonal, yaitu

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

- Sebelum metode Jacobi dan Gauss-Seidel digunakan, lakukan dulu pemeriksaan apakah matriks koefisien A bersifat dominan kuat secara diagonal.
- Salah satu cara agar matriks koefisien A bersifat dominan kuat secara diagonal adalah dengan menukarkan baris-baris dari SPL tersebut.
- Bila proses penukaran baris tidak berhasil membuat matriks koefisien A bersifat dominan kuat secara diagonal, maka metode Jacobi dan Gauss-Seidel biasanya tidak dapat digunakan.