

3.4.3 Source-code dasar

Proses triangularisasi dan substitusi-mundur dibakukan menjadi algoritma metode eliminasi gauss. Berikut ini saya tampilkan *source-code* dalam Matlab sebagaimana langkah-langkah diatas

```
clear all
clc

%---- inisialisasi matrik A ----
A = [1 1 0 3
     2 1 -1 1
     3 -1 -1 2
     -1 2 3 -1];

%---- inisialisasi vektor b ----
b = [4 ; 1 ; -3 ; 4];

%---- membentuk matrik augmentasi ----
dim = size(A);
n = dim(1);
for i = 1:n
    A(i,n+1) = b(i);
end

%==== Proses Triangularisasi ====
%---- menghilangkan variabel x1 dari P2, P3 dan P4 ----
m=A(2,1)/A(1,1); % huruf m mewakili simbol lambda
A(2,1)=A(2,1)-m*A(1,1);
A(2,2)=A(2,2)-m*A(1,2);
A(2,3)=A(2,3)-m*A(1,3);
A(2,4)=A(2,4)-m*A(1,4);
A(2,5)=A(2,5)-m*A(1,5);

m=A(3,1)/A(1,1);
A(3,1)=A(3,1)-m*A(1,1);
A(3,2)=A(3,2)-m*A(1,2);
A(3,3)=A(3,3)-m*A(1,3);
A(3,4)=A(3,4)-m*A(1,4);
A(3,5)=A(3,5)-m*A(1,5);

m=A(4,1)/A(1,1);
A(4,1)=A(4,1)-m*A(1,1);
A(4,2)=A(4,2)-m*A(1,2);
A(4,3)=A(4,3)-m*A(1,3);
A(4,4)=A(4,4)-m*A(1,4);
A(4,5)=A(4,5)-m*A(1,5);

%---- menghilangkan variabel x2 dari P3 dan P4 ----
m=A(3,2)/A(2,2);
A(3,1)=A(3,1)-m*A(2,1);
A(3,2)=A(3,2)-m*A(2,2);
```

```
A(3,3)=A(3,3)-m*A(2,3);
A(3,4)=A(3,4)-m*A(2,4);
A(3,5)=A(3,5)-m*A(2,5);
```

```
m=A(4,2)/A(2,2);
A(4,1)=A(4,1)-m*A(2,1);
A(4,2)=A(4,2)-m*A(2,2);
A(4,3)=A(4,3)-m*A(2,3);
A(4,4)=A(4,4)-m*A(2,4);
A(4,5)=A(4,5)-m*A(2,5);
```

```
%---- menghilangkan variabel x3 dari P4 ----
m=A(4,3)/A(3,3);
A(4,1)=A(4,1)-m*A(3,1);
A(4,2)=A(4,2)-m*A(3,2);
A(4,3)=A(4,3)-m*A(3,3);
A(4,4)=A(4,4)-m*A(3,4);
A(4,5)=A(4,5)-m*A(3,5);
```

```
%==== Proses Substitusi Mundur ====
x(4,1)=A(4,5)/A(4,4);
x(3,1)=(A(3,5)-A(3,4)*x(4,1))/A(3,3);
x(2,1)=(A(2,5)-(A(2,3)*x(3,1)+A(2,4)*x(4,1)))/A(2,2);
x(1,1)=(A(1,5)-
(A(1,2)*x(2,1)+A(1,3)*x(3,1)+A(1,4)*x(4,1)))/A(1,1);
```

3.4.4 Optimasi *source code*

Singkatnya, tujuan dari dilakukannya proses optimasi adalah untuk memperkecil jumlah baris statemen pada *source code* dasar. Seperti kita ketahui bersama, *source code* dasar eliminasi gauss yang tertulis di atas terdiri atas 70 baris statemen, sehingga perlu dilakukan proses optimasi untuk memperkecil jumlah baris statemen (tanpa menyalahi hasil perhitungan).

3.4.4.1 Optimasi *source code* bagian triangular

Langkah optimasi *source code* bagian triangularisasi dimulai dari baris statemen ke 23 hingga ke 27, yaitu

```
m=A(2,1)/A(1,1); % huruf m mewakili simbol lambda
A(2,1)=A(2,1)-m*A(1,1);
A(2,2)=A(2,2)-m*A(1,2);
A(2,3)=A(2,3)-m*A(1,3);
A(2,4)=A(2,4)-m*A(1,4);
A(2,5)=A(2,5)-m*A(1,5);
```

Bagian ini dapat dioptimasi menjadi

```
m=A(2,1)/A(1,1); % huruf m mewakili simbol lambda
```

```
for k = 1:5
    A(2,k) = A(2,k)-m*A(1,k);
End
```

Langkah optimasi yang sama juga bisa diterapkan untuk rangkaian baris statemen dari baris

ke 30 hingga 34 dan baris ke 37 hingga 41 (yang terdapat pada *source-code* dasar), sehingga masing-masing akan menjadi

```
for k = 1:5
    A(3,k) = A(3,k) - m*A(1,k);
End
```

Dan

```
for k = 1:5
    A(4,k) = A(4,k) - m*A(1,k);
End
```

Ternyata, pola optimasi yang sama juga masih bisa ditemui mulai baris ke 45 hingga baris statemen ke 64. Dengan demikian, setidaknya, tahapan pertama ini akan menghasilkan *sourcecode* baru hasil optimasi awal yaitu

```
clear all
clc

%---- inisialisasi matrik A ----
A = [1 1 0 3
     2 1 -1 1
     3 -1 -1 2
     -1 2 3 -1];

%---- inisialisasi vektor b ----
b = [4 ; 1 ; -3 ; 4];

%---- membentuk matrik augmentasi ----
dim = size(A);
n = dim(1);
for i = 1:n
    A(i,n+1) = b(i);
end

%==== Proses Triangularisasi ====
%---- menghilangkan variabel x1 dari P2, P3 dan P4 ----
m=A(2,1)/A(1,1); % huruf m mewakili simbol lambda
for k = 1:5
    A(2,k) = A(2,k) - m*A(1,k);
end

m=A(3,1)/A(1,1);
for k = 1:5
    A(3,k) = A(3,k) - m*A(1,k);
end

m=A(4,1)/A(1,1);
for k = 1:5
    A(4,k) = A(4,k) - m*A(1,k);
```

end

%---- menghilangkan variabel x2 dari P3 dan P4 ----

m=A(3,2)/A(2,2);

for k = 1:5

 A(3,k) = A(3,k)-m*A(2,k);

end

m=A(4,2)/A(2,2);

for k = 1:5

 A(4,k) = A(4,k)-m*A(2,k);

end

%---- menghilangkan variabel x3 dari P4 ----

m=A(4,3)/A(3,3);

for k = 1:5

 A(4,k) = A(4,k)-m*A(3,k);

end

%==== Proses Substitusi Mundur ====

x(4,1)=A(4,5)/A(4,4);

x(3,1)=(A(3,5)-A(3,4)*x(4,1))/A(3,3);

x(2,1)=(A(2,5)-(A(2,3)*x(3,1)+A(2,4)*x(4,1)))/A(2,2);

x(1,1)=(A(1,5)-

(A(1,2)*x(2,1)+A(1,3)*x(3,1)+A(1,4)*x(4,1)))/A(1,1);

3.4. MATRIK DAN ELIMINASI GAUSS

Sekarang, *source-code* eliminasi gauss telah mengecil menjadi hanya 58 baris statemen saja (sebelumnya ada 70 baris statemen). Namun ini belum merupakan akhir proses optimasi. *Sourcecode* yang terakhir ini masih bisa dioptimasi kembali.

Coba anda perhatikan pola yang nampak mulai pada baris statemen ke-22 hingga ke-35. Optimasi tahap dua dilakukan untuk menyederhanakan bagian tersebut, yaitu

```
for j = 2:4
    m=A(j,1)/A(1,1);
    for k = 1:5
        A(j,k) = A(j,k)-m*A(1,k);
    end
end
```

Demikian halnya untuk baris ke-38 sampai baris ke-46

```
for j = 3:4
    m=A(j,2)/A(2,2);
    for k = 1:5
        A(j,k) = A(j,k)-m*A(2,k);
    end
end
```

serta baris ke-49 hingga baris ke-52

```
for j = 4:4
    m=A(j,3)/A(3,3);
    for k = 1:5
        A(j,k) = A(j,k)-m*A(3,k);
    end
end
```

Dengan demikian hasil optimasi sampai dengan tahap ini adalah

```
clear all
clc

%---- inisialisasi matrik A ----
A = [1 1 0 3
     2 1 -1 1
     3 -1 -1 2
     -1 2 3 -1];

%---- inisialisasi vektor b ----
b = [4 ; 1 ; -3 ; 4];

%---- membentuk matrik augmentasi ----
dim = size(A);
n = dim(1);
for i = 1:n
    A(i,n+1) = b(i);
end
```

```

%==== Proses Triangularisasi ====
%---- menghilangkan variabel x1 dari P2, P3 dan P4 ----

for j = 2:4
    m=A(j,1)/A(1,1);
    for k = 1:5
        A(j,k) = A(j,k)-m*A(1,k);
    end
end

%---- menghilangkan variabel x2 dari P3 dan P4 ----
for j = 3:4
    m=A(j,2)/A(2,2);
    for k = 1:5
        A(j,k) = A(j,k)-m*A(2,k);
    end
end

%---- menghilangkan variabel x3 dari P4 ----
for j = 4:4
    m=A(j,3)/A(3,3);
    for k = 1:5
        A(j,k) = A(j,k)-m*A(3,k);
    end
end

%==== Proses Substitusi Mundur ====
x(4,1)=A(4,5)/A(4,4);
x(3,1)=(A(3,5)-A(3,4)*x(4,1))/A(3,3);
x(2,1)=(A(2,5)-(A(2,3)*x(3,1)+A(2,4)*x(4,1)))/A(2,2);
x(1,1)=(A(1,5)-
(1,2)*x(2,1)+A(1,3)*x(3,1)+A(1,4)*x(4,1)))/A(1,1);

```

Jika saya munculkan indeks i pada bagian proses triangularisasi

```

%==== Proses Triangularisasi ====
%---- menghilangkan variabel x1 dari P2, P3 dan P4 ----
i = 1;
    for j = i+1:4
        m=A(j,i)/A(i,i);
        for k = 1:5
            A(j,k) = A(j,k)-m*A(i,k);
        end
    end
%---- menghilangkan variabel x2 dari P3 dan P4 ----
i = 2;
    for j = i+1:4
        m=A(j,i)/A(i,i);
        for k = 1:5
            A(j,k) = A(j,k)-m*A(i,k);
        end
    end

```

```

        end
    end
%---- menghilangkan variabel x3 dari P4 ----
i = 3;
    for j = i+1:4
        m=A(j,i)/A(i,i);
        for k = 1:5
            A(j,k) = A(j,k)-m*A(i,k);
        end
    end
end

```