PENGANTAR MATLAB UNTUK SISTEM PERSAMAAN LINEAR

Muh. Irwan[i]

[i] *Prodi Matematika, UIN Alauddin, muhirwan @uin-alauddin.ac.id*

ABSTRAK,
Pada paper ini dijelaskan beberapa alternatif dalam menyelesaikan sistem persamaan linear (SPL) dengan menggunakan MATLA. Metode yang digunakan diantaranya adalah metode matriks ($X = A^{-1}B$) , rref($[A, B]$) dan eliminasi Gauss.

Kata Kunci *Sistem persamaan linear, MATLAB.*

1. PENDAHULUAN

Sistem persamaan linear (SPL) merupakan gabungan dari beberapa persamaan linear yang saling berkorelasi. Berbagai metode yang digunakan seperti eliminasi, substitusi, eliminasi gauss, eliminasi Gauss, eliminasi Gauss Jordan dan lain sebagainya. Metode-metode tersebut jika dikerjakan secara manual seringkali membutuhkan waktu yang lama untuk menyelesaikan SPL. Sehingga untuk memudahkan pengerjaan maka digunakan software MATLAB.
MATLAB singkatan dari *Matrix Laboratory* yang merupakan software matematika yang telah digunakan pada berbagai bidang ilmu. MATLAB sebagai software tingkat menengah dan lanjut, dibutuhkan untuk menyelesaikan persoalan-persoalan matematika khususnya dalam menyelesaikan SPL. Oleh karena itu pada paper ini akan diperkenalkan penyelesaian SPL menggunakan MATLAB.

2. TINJAUAN PUSTAKA

Persamaan linear aljabar berbentuk,

$$a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n = b$$

Solusi untuk persamaan ini, merupakan sesuatu yang penting untuk berbagai aplikasi. Di dalam Pemrograman MATLAB, untuk menyelesaikan persamaan ini, dibutuh setidaknya dua metode yaitu[3]:

a. Menggunakan representasi matriks
b. Menggunakan fungsi solve

Pada bagian ini, kita akan diperkenalkan mengenai operasi matriks dan vektor dan menggunakannya untuk menyelesaikan system persamaan linear aljabar.

Matriks

Definisi

Suatu matriks adalah susunan segiempat siku-siku dari bilangan-bilangan. Bilangan-bilangan dalam susunan tersebut dinamakan entri matriks.

Suatu matriks dapat dianggap sebagai suatu tabel nilai yang memiliki baris dan kolom. Bentuk umum suatu matriks A ditulis sebagai

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = a_{ij},$$

Dimana $i = 1,2,..m, j = 1,2..n$. Matriks ini memiliki m baris dan n kolom, berukuran $m \times n$.
Suatu vektor merupakan kasus khusus suatu matriks, yaitu matriks yang berdimensi satu (salah satu dari m atau n yang sama dengan 1). Suatu vektor baris mamiliki orde $1 \times n$, sedangkan vektor kolom memiliki ordo $m \times 1$. Selanjutnya, untuk skalar juga merupakan kasus khusus matriks $1 \times 1, n = 1, m = 1$.

Sifat-sifat Matriks

Dua matriks dikatakan sama jika dan hanya jika semua elemen-elemen matriks yang berkorespondensi memiliki nilai yang sama. Serta ukuran matriksnya harus sama. Atau atau dapat dituliskan sebagai
$[A] = [B]$ jika dan hanya jika $a_{ij} = b_{ij}$ untuk semu i dan j.

Konsep pemrograman

Untuk mengetahui apakah dua matriks sama atau tidak, dapat dibuat suatu fungsi untuk mengecek hal tersebut. Jika fungsinya bernilai 1 maka dikatakan dua matriks tersebut sama, tetapi jika

bernilai 0, maka matriks tersebut tidak sama. Berikut diberikan programnya

```
function myflag = myisequal(mata,matb)
% kita mengasumsikan bahwa jika fungsi
%  bernilai 1 maka kedua matriks sama
% jika fungsi bernilai 0, maka matriks
%  tidak sama

myflag = logical(1);
[r c] = size(mata);
if all(size(mata) ~= size(matb))
myflag = logical(0);
else
for i=1:r
      for j = 1:c
            if mata(i,j) = matb(i,j)
                  myflag = logical(0);
            end
      end
end
end
```

selanjutnya, dengan menuliskan pada worksheet matlab dengan,

```
>> mata = [2 5 8; 1:3];
>> matb = [2:3:8; 1 2 3];
>> myisequal(mata,matb)
ans =
1
```

Metode yang lebih efisien dapat digunakan adalah,

```
>> isequal(mata,matb)
ans =
1
```

Fungsi `isequal` akan bernilai 0 jika kedua matriks tidak sama.

Matriks Persegi

Jika suatu matriks memiliki jumlah matriks sama dengan jumlah kolomnya, maka matriks tersebut dikatakan matriks persegi. Diagonal utama matriks persegi disimbolkan denga a_{ii}. Sebagai contoh,

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Elemen $a_{11} = 1, a_{22} = 5$ dan $a_{33} = 9$ disebut sebagai diagonal matriks A. MATLAB memiliki fungsi diagonal `(diag)` yang bertujuan untuk menampilkan diagonal suatu matriks.

```
>> mymat = reshape(1:16,3,3)'
mymat =
1     2     3
4     5     6
7     8     9
>> diag(mymat)'
ans =
1     5     9
```

Trace matriks bujur sangkar adalah jumlahand ari semua diagonalnya. Sebagai contoh: 1 + 5 + 9 =15.

Konsep pemrograman

Bagaimana menentukan trace matriks dalam pemrograman MATLAB? Hal ini mudah dilakukan, mengingat bahwa trace suatu matriks bujur sangkar adalaj jumlahan elemenya yang memiliki indeks sama. Sehingga hanya dibutuhkan satu perulangan. Adapun perulangan/iterasi akan berhenti ketika jumlah perulangan sama dengan ukuran matriksnya. Berikut diberikan programnya.

```
function outsum = mytrace(mymat)
% Menetukan Trace matriks bujur sangkar
% return kosong  jika matriksnya tidak
sama
[r c] = size(mymat);
if r = c
      outsum = [];
else
      outsum = 0;
      for i = 1:r
        outsum = outsum+mymat(i,i);
      end
end
```

selanjutnya, menuliskan pada worksheet perintah berikut:

```
>> mymat = reshape(1:16,3,3)'
mymat =
1     2     3
4     5     6
7     8     9
>> mytrace(mymat)
ans =
      34
```

Operasi Matriks

Operasi matriks yang akan dijelaskan pada bagian ini yaitu penjumlahan, pengurangan dan perkalian yaitu sebagai berikut.

Penjumlahan

Definisi

Jika $\boldsymbol{A}$ *dan* $\boldsymbol{B}$ *adalah dua matriks yang ukurannya sama, maka jumlah* $\boldsymbol{A+B}$ *adalah matriks yang diperoleh dengan menambahkan bersama-sama entri yang bersesuaian dalam kedua matriks tersebut. Matriks yang ukurannya berbeda tidak dapat dijumlahkan[3].*

Dua matriks dapat dijumlahkan jika dan hanya jika kedua matriks tersebut memiliki ukuran yang sama. Penjumlahan matriks A dan B dapat dituliskan sebagai

$$c_{ij} = a_{ij} + b_{ij}.$$

Konsep pemrograman

Untuk penjumlahan matriks, *user* harus hati-hati dalam menempatkan perulangan bertingkat (nested loop)[3].

```
function outmat = mymatadd(mat1,mat2)
outmat = [];
if all(size(mat1) == size(mat2))
outmat = zeros(size(mat1));
[r c] = size(mat1);
for i = 1:r
for j = 1:c
outmat(i,j) = mat1(i,j)+mat2(i,j);
end
end
end
```

selanjutnya, dituliskan pada worksheet MATLAB,

```
>> A = [1:3;4:6];
>> B = [100 10 1; 10 100 1];
>> C = mymatadd(A,B)
C =
101    12     4
14     105    7
```

Operasi pengurangan dua matriks, mirip dengan proses penjumlahan. Perbedaanya, terletak pada tanda operator, yaitu

$$c_{ij} = a_{ij} - b_{ij}.$$

Sehingga untuk programnya, hanya diganti tanda "+" dengan tanda "–" pada program penjumlahan.

Perkalian Matriks

Definisi

Jika $\boldsymbol{A}$ *adalah matriks* $\boldsymbol{m \times r}$ *dan* $\boldsymbol{B}$ *adalah matriks yang berukuran* $\boldsymbol{r \times n}$*, maka hasil kalai* $\boldsymbol{AB}$ *adalah matriks berukuran* $\boldsymbol{m \times n}$ *yang entri-entrinya ditentukan sebagai berikut, untuk mencari entri dalam baris* $\boldsymbol{i}$ *dan kolom* $\boldsymbol{j}$ *dari* $\boldsymbol{AB}$*, pilihlah baris* $\boldsymbol{i}$ *dari matriks* $\boldsymbol{A}$ *dan kolom* $\boldsymbol{j}$ *dari matriks* $\boldsymbol{B}$*. Kalikan entri-entri yang bersesuaian dari baris dan kolom tersebut bersama-sama dean kemudian tambahkan hasil kali yang dihasilkan[3].*

Konsep Pemrograman

Untuk melakukan perkalian dua matriks, dibutuhkan 3 perulangan. Dengan ketentuan, 2 perulangan khusus untuk menentukan iterasi baris dan kolom matriks C yang berukuran $m \times n$. Untuk setiap elemen pada C, perulangan yang di dalam bertujuan untuk menjumlahkan hasil dari $a_{ik} * b_{kj}$, dimana $k = 1..n$

```
A = [3 8 0; 1 2 5];
B = [1 2 3 1; 4 5 1 2; 0 2 3 0];
[m n] = size(A);
[nb p] = size(B);
if n ~= nb
disp('matriks tidak berukuran sama')
else
C = zeros(m,p);
 for i=1:m
for j = 1:p
mysum = 0;
 for k = 1:n
 mysum = mysum+ A(i,k) * B(k,j);
end
 C(i,j) = mysum;
 end
end
end
```

simpan script di atas dengan nama `mymatmult.m`, selanjutnya dengan mengetik perintah berikut di worksheet

```
>> mymatmult
C =
35    46    17    19
9     22    20    5
```

Metode yang lebih efisien yang dapat digunakan yaitu dengan menuliskan perintah berikut,

```
>> A = [3 8 0; 1 2 5];
>> B = [1 2 3 1; 4 5 1 2; 0 2 3 0];
>> C = A*B
```

```
C =
35      46      17      19
9       22      20      5
```

Selain perkalian, Matlab juga menyediakan fungsi built-in untuk menentukan invers suatu matriks dengan menggunakan sintaks **inv**[3]**.**

3. PEMBAHASAN

Solusi Matriks Untuk Sistem Persamaan Linear

Persamaan aljabar linear berbentuk
$a_1x_1 + a_2x_2 + a_3x_3 + \cdots a_nx_n = b,$
Dimana $a_1, a_2, \dots, a_n$ koefisien konstant, $x_1, x_2, x_3, \dots, x_n$ tidak diketahui dan b adalah konstant [1],[2],[3],[4],[5],[6]. Sedangkan system persamaan aljabar linear berbentuk

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots a_{2n}x_n &= b_2 \\ \vdots \qquad\quad \vdots \qquad\qquad & \vdots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \cdots a_{mn}x_n &= b_m, \end{aligned}$$

selanjutnya, system persamaan ini akan diubah dalam bentuk matriks $AX = B$ dengan A adalah matriks koefisien x, X adalah vektor kolom yang tidak diketahui dan B vektor kolom konstant sisi kanan sisitem persamaan. Selanjutnya, bentuk $Ax = B$ dapat dijabarkan sebagai,

$$A^{-1}AX = A^{-1}B$$
$$IX = A^{-1}B$$
$$X = A^{-1}B$$

Jadi, nilai X dapat ditentukan dengan menentukan nilai A^{-1} terlebih dahulu, kemudian dikalikan dengan matriks B.
Misalkan diberikan persamaan sebagai berikut

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

Jad, matriks

$$A = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7.0 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}, \quad B = \begin{bmatrix} 7.85 \\ -19.3 \\ 71.4 \end{bmatrix}$$

selanjutnya, solusi persamaan linear ditentukan menggunakan MATLAB yaitu,

```
>> A=[3 -0.1 -0.2;0.1 7 -0.3;0.3 -0.2
10]

A =

    3.0000   -0.1000   -0.2000
    0.1000    7.0000   -0.3000
    0.3000   -0.2000   10.0000

>> b=[7.85;-19.3;71.4]

b =
    7.8500
  -19.3000
   71.4000

>> inv(A)*b
ans =

    3.0000
   -2.5000
    7.0000
```

Jadi diperoleh nilai $x_1 = 3.0$, $x_2 = -2.5$ dan $x_3 = 7$. Cara yang lebih sederhana dapat dilakukan dengan menggunakan fungsi built-in `rref`. Fungsi `rref(A)` menghasilkan matriks berbentuk eselon baris tereduksi dari matriks A.
Penggunaan dalam matlab bisa diruliskan dengan

```
>> A=[3 -0.1 -0.2;0.1 7 -0.3;0.3 -0.2
10];
>> b=[7.85;-19.3;71.4];

>> rref([A b])

ans =

    1.0000         0         0    3.0000
         0    1.0000         0   -2.5000
         0         0    1.0000    7.0000
```

Hasil ini sama dengan menggunakan metode invers. Bagian terakhir yang akan dibahas adalah menyelesaikan SPL dengan menggunakan metode eliminasi Gauss.

Metode Eliminasi Gauss (EG)

Metode EG merupakan salah satu metode yang digunakan untuk menyelesaikan SPL. Sasaran EG adalah membuat matriks yang diperbesar menjadi matriks segitiga atas dengan menggunakan operasi baris elementer (OBE). Berikut diberikan OBE [1][

1. Mengalikan suatu baris dengan suatu konstanta yang bukan nol.
2. Menukarkan suatu baris dengan baris yang lain.
3. Mengurankan suatu baris dengan kelipatan baris lainnya.

Pada proses penggunaan OBE, terdapat dua langkah yang dilakukan untuk menyelesaikan SOL yaitu

1. Elimisasi $x_1, x_2, \dots x_n$
2. Substitusi ulang

Algoritma Eliminasi Gauss

Input : A adalah Matriks koefisien X, B vektor konstanta.
Output: Vektor X
Langkah-langkah.

Program [1]

```
clc
clear all
a=input('masukkan matriks A =');
b=input('masukkan matriks B= ');
[n,n]=size(a);
s=0;
for j=1:n-1
    if a(j,j)==0
        k=j;
        for k=k+1:n
            if a(k,j)==0
                continue
            end
            break
        end
        B=a(j,:);
        C=r(j);
        a(j,:)=a(k,:);
        b(j)=b(k);
        a(k,:)=B;
        b(k)=C;
    end
    for i=1+s:n-1
        L=a(i+1,j)/a(j,j);
        a(i+1,:)=a(i+1,:)-L*a(j,:);
        b(i+1)=b(i+1)-L*b(j);
    end
    s=s+1;
end
a;
x(n)=b(n)/a(n,n);
jum=0;
for i=n-1:-1:1
    for j=i+1:n
          jum=jum+a(i,j)*x(j);
    end
    x(i)=(b(i)-jum)/a(i,i);
end
    x
```

Output:

```
masukkan matriks A =
[3  -0.1  -0.2;0.1  7  -0.3;0.3  -0.2
10];
masukkan matriks B=
[7.85;-19.3;71.4];
x =
    3.6844   -2.5000    7.0000
A =
    3.0000   -0.1000   -0.2000
    0.1000    7.0000   -0.3000
    0.3000   -0.2000   10.0000

b =
    7.8500
  -19.3000
   71.4000

x =
    3.0000   -2.5000    7.0000
```

4. KESIMPULAN

Berdasarkan hasil pembahasan maka disimpulkan bahwa, MATLAB merupakan suatu software alternatif untuk memudahkan penyelesaian SPL baik menggunakan Algoritma maupun dengan menggunakan fungsi built-in yang disiapkan MATLAB.

5. DAFTAR PUSTAKA

[1] Anton Howard and Rorres Chris. 200 .Elementry Linear Algebra Ninth Edition.

[2] Attaway Stormy. 2009. MATLAB: A Practical Introduction to Programming and Problem Solving. Elsevier's Science &Technology Rights. USA.

[3] Epperson, James F. 2013. An Introduction to Numerical Methods and Analysis Secon Edition. John Wilety & Sons, Inc. Canada

[4] https://alimath.wordpress.com/2013/03/26 /eliminasi-gauss-dan-implementasinya-di-matlab/.

[5] Lyshevski Sergey E. 2003. Engineering and Scientific Computations Using MATLAB. John Wiley & Sons, Inc. Canada

[6] Suprianto Supomo. 2010. Komputasi untuk Sains dan Teknik Menggunakan Matlab Edisi III. Fisika-FMIPA, Universitas Indonesia.