KAJIAN NUMERIK MASALAH SYARAT BATAS MELALUI PENYELESAIAN MATRIKS TRIDIAGONAL
(Studi Kasus : Menghitung Potensial Listrik)

Tatik Juwariyah
Fakultas Teknik,UPN "Veteran" Jakarta
Kampus Jalan RS. Fatmawati Pondok Labu Jakarta 12450, Indonesia,
email : tatikjuwariyah@gmail.com

Abstract

A numerical study of the solution of boundary value problems with the tridiagonal matrix approach has been done. The case studied is computing the electrical potential expressed by 1D Poisson's equation with boundary conditions. The 1D Poisson's equations are then discreted with the finite different method so as to form a system of linier equations. The system of non-homogeneous linear equations that can be formed is a tridiagonal matrix. Then the matrix is solved by Gaussian elimination algorithm. On this study, the algorithm is implemented on three programming languages namely, Fortran, Java and MATLAB.
Keywords: Gaussian elimination, finite different, tridiagonal matrix, Poisson's equation

PENDAHULUAN

Masalah syarat batas (*boundary-value problems*) sering muncul pada persamaan diferensial baik persamaan diferensial biasa ataupun persamaan diferensial parsial yang merupakan pondasi kajian ilmu-ilmu fisika terapan. Beberapa kasus masalah syarat batas dijumpai pada perhitungan potensial kelistrikan, potensial gravitasi, gelombang elektromagnetika, aliran fluida dan aliran panas. Terdapat dua cara yang dapat ditempuh untuk menyelesaikan masalah syarat batas pada sebuah persamaan diferensial yaitu secara analitik dan secara numerik. Cara numerik lebih dipilih oleh para insinyur dikarenakan lebih efisien dalam menyelesaikan masalah di lapangan yang biasanya lebih kompleks dan rumit. Sementara cara analitik ditempuh jika syarat batasnya ideal dan sederhana dikarenakan penyelesaian secara analitik membutuhkan piranti matematika lanjutan seperti deret Fourier, transformasi Laplace ataupun fungsi-fungsi khas (fungsi Gamma, fungsi Beta, fungsi Green, fungsi Bessel, fungsi Legendre, dsb).

Perkembangan komputer sebagai alat hitung yang handal sangat membantu penyelesaian masalah-masalah di bidang keteknikan. Berbagai piranti lunak yang telah aplikatif dan berbentuk GUI *(grafic user interface)* semakin mempermudah dalam mensimulasikan masalah-masalah nyata. Meskipun telah banyak piranti lunak siap pakai yang lebih aplikatif untuk menyelesaikan kasus-kasus yang melibatkan persamaan diferensial memilih bahasa pemrograman terstruktur (seperti Fortran, Basic, Pascal, C/C++ dan Java) untuk menyelesaikan masalah langkah demi langkah merupakan salah satu cara yang baik digunakan oleh akademisi pendidikan. Hal ini dikarenakan melatih dan mengasah ketrampilan proses pendidikan/pengajaran.

TINJAUAN PUSTAKA

Potensial Elektrostatik

Medan listrik yang menembus suatu permukaan khayal dapat dijelaskan oleh hukum Gauss dalam bentuk diferensial dinyatakan sebagai :

$$\nabla \bullet E = \rho / \varepsilon_0 \qquad (1)$$

Medan listrik **E** adalah gradien potensial istrik ϕ dinyatakan oleh ungkapan :

$$E = -\nabla \phi \qquad (2)$$

Substitusi persamaan (2) ke persamaan (1) diperoleh:

$$\nabla \bullet \nabla \phi = -\rho / \varepsilon_0$$

Dikarenakan operator nabla ∇ adalah vektor maka dapat disederhanakan sehingga berbentuk

$$\nabla^2 \phi = -\rho / \varepsilon_0 \qquad (3)$$

Bentuk matematis persamaan (3) dikenal dengan Persamaan Poisson. Jika ruas kanan persamaan Poisson bernilai nol persamaan akan mereduksi menjadi persamaan Laplace seperti berikut,

$$\nabla^2 \phi = 0 \qquad (4)$$

Persamaan Laplace dapat ditemukan di banyak fenomena seperti aliran fluida, elektrostatika, gelombang elektromagnetika dan aliran panas.

Persamaan Poisson 1D ditulis sebagai berikut,

$$\frac{d^2\phi}{dx^2} = -\rho / \varepsilon_0 \qquad (5)$$

Bentuk umum persamaan Poisson 1D secara matematis tidak lain adalah persaman diferensial berbentuk :

$$\frac{d^2 y(x)}{dx^2} = f(x) \qquad (6)$$

Penyelesaian/solusi khusus persamaan (6) dapat diperoleh jika kondisi batas y(x) diketahui.

Diskritisasi Persamaan Diferensial

Diskritisasi dengan metode beda hingga yaitu beda pusat *(central difference)* suku ruas kiri persamaan (6) berubah menjadi :

$$\frac{d^2 y(x_i)}{dx^2} = \frac{y_{i-1} - 2y_i + y_{i+1}}{(\Delta x)^2} + O(\Delta x)^2 \qquad (7)$$

Dengan $O(\Delta x)^2$ adalah kesalahan perhitungan numerik. Apabila syarat/kondisi batas y(x) diketahui maka dengan metode beda hingga dengan mengambil sejumlah N titik-titik grid, penyelesaian yang diinginkan adalah y(x) pada ranah $x_0 \le x \le x_N$ dimana nilai atau bentuk penyelesaian pada kedua titik batas yaitu $y(x_0) = y_0$ dan $y(x_N) = y_N$ sudah diketahui. Dengan metode beda hingga persamaan (6) akan berubah menjadi

$$y_{i-1} - y_i + y_{i+1} = h^2 f_i \qquad (8)$$

Dengan $h=\Delta x$ adalah lebar langkah perhitungan numerik.

Berbeda dengan masalah syarat awal, disini nilai y(x) yang akan dicari harus sesuai pada batas x_0 dan x_N. Ini berarti persamaan (8) tidak dapat diselesaikan satu persatu untuk tiap i tertentu tetapi harus diselesaikan secara serentak untuk seluruh i yang ada. Artinya apabila nilai N besar maka cacah persamaan menjadi banyak sekali. Salah satu metode paling efektif untuk memecahkan banyak persamaan secara serentak adalah menggunakan cara sistem persamaan linier/matriks.

Persamaan (8) berubah menjadi sistem persamaan linier(matriks) yang unik karena hanya berisi tiga koefisien di setiap barisnya sehingga matriks koefisien yang unik ini dikenal dengan matriks tridiagonal.

$$\begin{pmatrix} -2 & 1 & 0 & \dots & 0 & 0 \\ 1 & -2 & 1 & & & 0 \\ 0 & 1 & -2 & 1 & & \\ \vdots & & & \ddots & & \\ 0 & & & 1 & -2 & 1 \\ 0 & 0 & \dots & 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ y_{N-2} \\ y_{N-1} \end{pmatrix} = \begin{pmatrix} h^2 f_1 - y_0 \\ h^2 f_2 \\ . \\ . \\ h^2 f_{N-2} \\ h^2 f_{N-1} - y_N \end{pmatrix} \qquad (9)$$

Salah satu metode efisien untuk menyelesaikan sistem persamaan simultan yang tersusun atas matriks tridiagonal adalah dengan eleminasi Gauss yaitu dengan mengeliminasi unsur-unsur yang terletak persis di bawah diagonal utama. Bentuk umum persamaan (9) seperti berikut:

$$\begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 & 0 \\ a_2 & b_2 & c_2 & & & 0 \\ 0 & & b_3 & & & \\ \vdots & & & \ddots & & \\ 0 & & & a_{N-2} & b_{N-2} & c_{N-2} \\ 0 & 0 & \dots & 0 & a_{N-1} & b_{N-1} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ y_{N-2} \\ y_{N-1} \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ . \\ . \\ r_{N-2} \\ r_{N-1} \end{pmatrix} \qquad (10)$$

Dengan mendefinisikan kaitan

$$\beta_j = b_j - \frac{a_j}{\beta_{j-1}} c_{j-1} \quad \text{dan} \quad \rho_j = r_j - \frac{a_j}{\beta_{j-1}} \rho_{j-1},$$

untuk j = 2,3,..., N-1 dimana $\beta_1 = b_1$ dan $\rho_1 = r_1$ maka persamaan (10) berubah menjadi bentuk persamaan matriks berikut

$$\begin{pmatrix} \beta_1 & c_1 & 0 & \dots & 0 & 0 \\ 0 & \beta_2 & c_2 & & & 0 \\ 0 & & \beta_3 & & & \\ \vdots & & & \ddots & & \\ 0 & & & 0 & \beta_{N-2} & c_{N-2} \\ 0 & 0 & \dots & 0 & 0 & \beta_{N-1} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ y_{N-2} \\ y_{N-1} \end{pmatrix} = \begin{pmatrix} \rho_1 \\ \rho_2 \\ . \\ . \\ \rho_{N-2} \\ \rho_{N-1} \end{pmatrix} \qquad (11)$$

Bentuk persamaan (11) sangat menarik karena memungkinkan semua y_i dapat diperoleh dengan cara substitusi balik yaitu pertama dihitung y_{N-1} melalui kaitan

$$y_{N-1} = \frac{\rho_{N-1}}{\beta_{N-1}} \qquad (12)$$

Setelah y_{N-1} didapatkan maka y_i yang lain diperoleh melalui kaitan untuk j = 2,3,..., N-1 sebagai berikut

$$y_{N-j} = \frac{\rho_{N-1} - c_{N-j}\, y_{N-j+1}}{\beta_{N-j}} \qquad (13)$$

METODE PENELITIAN

Langkah-langkah penyelesaian kasus masalah di penelitian ini dijabarkan sebagai berikut.

Pemodelan Matematis Masalah

Penyelesaian masalah menghitung potensial listrik satu dimensi yang menjadi kasus masalah pada penelitian ini adalah andaikan ditinjau suatu daerah antara 0 ≤ x ≤ 3 yang memiliki rapat muatan berbentuk $\rho(x) = 3\varepsilon_0 x$ dan adanya syarat batas nilai potensial ϕ(0) = 10 dan ϕ (3) = 0. Masalah ini dapat dirumuskan seperti persamaan (6) dengan mengambil y(x) = ϕ (x) dan $f(x) = \rho(x)/\varepsilon_0 = 3\varepsilon_0 x/\varepsilon_0 = -3x$. Sehingga diperoleh model matematis masalah yaitu PD berbentuk :

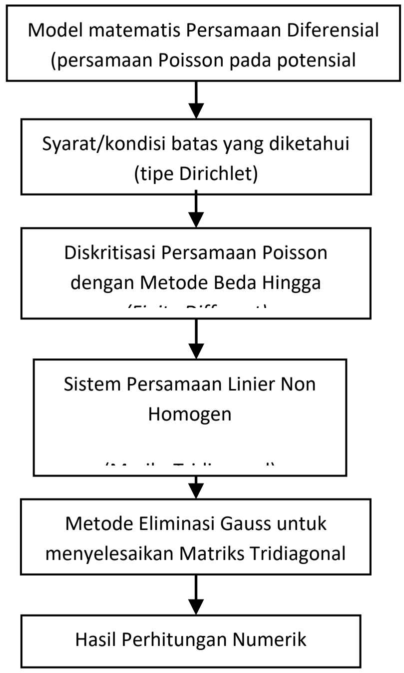
$$\frac{d^2\phi}{dx^2} = -3x \qquad (14)$$

Dengan syarat batas diketahui ϕ(0) = 10 dan ϕ (3) = 0.

Algoritma Penyelesaian Numerik

1. Input :
   a. Banyaknya N (cacah perhitungan dalam hal ini terkait dengan ukuran matriks persegi).
   b. Daerah selang x yang ditinjau.
   c. Lebar langkah perhitungan
   d. Syarat batas y(0) dan y(N).
2. Mengisi unsur-unsur matrik tridiagonal dan vektor kolom ruas kanan a(i), b(i), c(i) dan r(i). Khusus untuk i=1 dan i=N-1, unsur matriks r(i) berbeda dengan yang lain karena ada pengurangan dengan y_0 dan y_n.
3. Algoritma Eliminasi Gauss : mengubah matriks tridiagonal ke bentuk matriks atas yang hanya mengandung dua larik yaitu beta(i) dan c(i) serta r(i) ke rho(i).
   Menghitung y(i) dengan substitusi balik yaitu dihitung lebih dahulu y(n-1) dan dilanjutkan ke y(n-2) dan seterusnya sampai y(1).
4. Menampilkan hasil.
5. Membuat subprogram/subroutine untuk mendefinisikan fungsi Ruas Kanan persamaan Poisson.

Pada penelitian ini Algoritma Penyelesaian Masalah diimplementasikan pada beberapa bahasa pemrograman yaitu Fortran, Java dan MATLAB.
Diagam alir Metode Penyelesaian Masalah yang diangkat pada Penelitian ini dilukiskan pada Gambar 1.



Gambar 1. Diagram alir metode penyelesaian masalah

PEMBAHASAN

Berikut adalah implementasi algoritma penyelesaian masalah dengan bahasa Fortran.

```
PROGRAM syarat_batas
IMPLICIT NONE
REAL, DIMENSION(0:100) :: x,y,a,b,c,r,beta,rho
REAL :: h
INTEGER ::n,i
n=30
x(0)=0.0
x(n)=3.0
h=(x(n)-x(0))/n
y(0)=10.0
y(n)=0.0

!Memasukkan unsur-unsur matrik a(i), b(i), c(i) dan r(i)
!Khusus untuk i=1 dan i=n-1, unsur matriks r(i) berbeda
!dengan yang lain karena ada pengurangan dengan y0 dan yn

DO i=1,(n-1)
   x(i)=x(0)+i*h
   b(i)=-2.0
   IF (i .EQ. 1) THEN
        r(i)=f(x(i))*h**2-y(0)
   ELSE
      IF (i .EQ. (n-1)) THEN
         r(i)=f(x(i))*h**2-y(n)
      ELSE
         r(i)=f(x(i))*h**2
      END IF
   END IF
END DO
DO i=1,(n-2)
c(i)=1.0
END DO
DO i=2,(n-1)
a(i)=1.0
END DO
```

```
!--------------------------------------------------------
!Mengubah matriks tridiagonal ke bentuk matriks atas yang
!hanya mengandung dua larik yaitu beta(i) dan c(i) serta
!r(i) ke rho(i)
!--------------------------------------------------------
CALL tridiagonal(n-1,a,b,c,r,beta,rho)
!--------------------------------------------------------
!Menghitung y(i) dengan substitusi balik yaitu dihitung
!lebih dahulu y(n-1) dan dilanjutkan ke y(n-2) dan
!seterusnya sampai y(1)
!--------------------------------------------------------
y(n-1)=rho(n-1)/beta(n-1)
DO i=2,(n-1)
y(n-i)=(rho(n-i)-c(n-i)*y(n-i+1))/beta(n-i)
END DO
!--------------------------------------------------------
!Menampilkan hasil
!--------------------------------------------------------
DO i=0,n
WRITE(*,*) i,x(i),y(i)
END DO

CONTAINS
FUNCTION f(xx)
IMPLICIT NONE
REAL :: f
REAL, INTENT(in) :: xx
f=-3*x
END FUNCTION f
!--------------------------------------------------------
!Mengubah matriks tridiagonal dan perluasannya a(i), b(i),
!c(i) dan r(i) agar menjadi beta(i), c(i) dan rho(i)
!--------------------------------------------------------
SUBROUTINE tridiagonal(m,a,b,c,r,beta,rho)
IMPLICIT NONE
REAL, DIMENSION(0:100), INTENT(in) :: a,b,c,r
REAL, DIMENSION(0:100), INTENT(out) :: beta,rho
INTEGER, INTENT(in) :: m
REAL :: pengali
beta(1)=b(1)
rho(1)=r(1)
DO i=2,m
pengali=a(i)/beta(i-1)
beta(i)=b(i)-pengali*c(i-1)
rho(i)=r(i)-pengali*rho(i-1)
END DO
END SUBROUTINE tridiagonal
END PROGRAM syarat_batas
```

Dalam bahasa Java, algoritma penyelesaian masalah ditulis seperti berikut.

```
public class syaratBatas {
public static void main(String args[]) {
// DIFINISI VARIABEL DALAM ARRAY
double  x[]=new double[100];
double  y[]=new double[100];
double  a[]=new double[100];
double  b[]=new double[100];
double  c[]=new double[100];
double  r[]=new double[100];
double  rho[]=new double[100];
double  beta[]=new double[100];
double h,pengali;
int N,i;
// SYARAT BATAS
    N=30;
    x[0]=0.0;
    x[N]=3.0;
    h=(x[N]-x[0])/N;
    y[0]=10.0;
    y[N]=0.0;
/*  mengisi  matrik  tridiagonal  dan  vektor  kolom  ruas  kanan
(a,b,c,r);   Memasukkan unsur-unsur matrik a(i), b(i), c(i) dan
r(i);     Khusus untuk i=1 dan i=n-1, unsur matriks r(i) berbeda
dengan yang lain karena ada pengurangan dengan y0 dan yn */

    for (i=1;i<=N-1;i++){
         x[i]=x[0]+i*h;
         b[i]=-2.0;
         if (i==1){
            r[i]=fs_x(x[i])*h*h-y[0];}
         else
          if (i==(N-1)) {
          r[i]=fs_x(x[i])*h*h-y[N];}
      else{
      r[i]=fs_x(x[i])*h*h;      }
         }
   for (i=1;i<=N-2;i++) {
     c[i]=1.0;   }
         for (i=2;i<=(N-1);i++) {
     a[i]=1.0;   }

/*  Mengubah  matriks  tridiagonal  dan  perluasannya  a(i),  b(i),
c(i) dan r(i) agar menjadi beta(i), c(i) dan rho(i) */

 beta[1]=b[1];
rho[1]=r[1];
for (i=2;i<=(N-1);i++) {
pengali=a[i]/beta[i-1];
beta[i]=b[1]-pengali*c[i-1];
rho[i]=r[i]-pengali*rho[i-1];
}
/*Menghitung y(i) dengan substitusi balik yaitu dihitung
  lebih dahulu y(n-1) dan dilanjutkan ke y(n-2) dan
  seterusnya sampai y(1)  */

y[N-1]=rho[N-1]/beta[N-1];
for (i=2;i<=(N-1);i++){
y[N-i]=(rho[N-i]-c[N-i]*y[N-i+1])/beta[N-i];
}
// tampilkan hasil perhitungan lengkap beserta batas
for (i=0;i<=N;i++){
 System.out.format("%02d"  +  "%10.4f"  +  "%10.4f%n",  (i),  x[i],
y[i]);}}
}
// DEFINISI FUNGSI
 public static  double fs_x(double x){
    return -3*x;   }
}
```

Dengan MATLAB bentuk source code penyelesaian masalah adalah sebagai berikut.

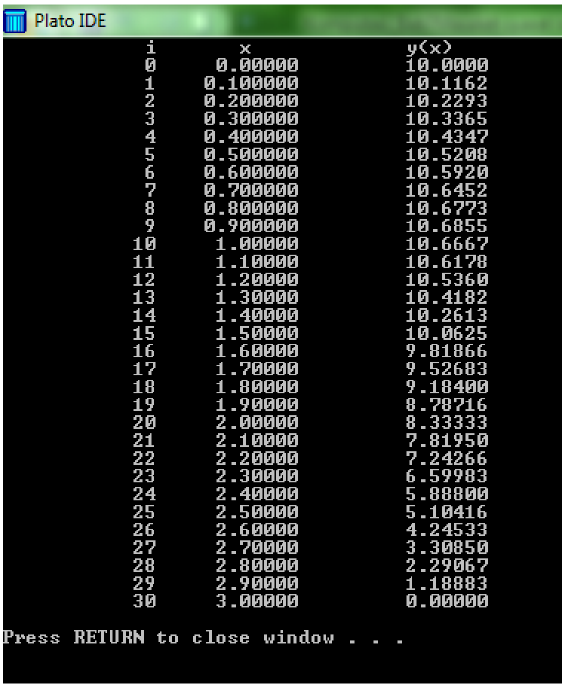
```
clc;clear;
%--------------------------------------------------------
%Masukan  banyaknya  cacah  titik  n,  batas  kiri  x(0)  dan  batas
kanan  x(n)  daerah  yang  ditinjau,  ukuran  langkah  h,serta  syarat
batas potensial di x0 dan xn
%--------------------------------------------------------
n=30;
x0=0.0;
x(n)=3.0;
h=(x(n)-x0)/n;
y0=10.0;
y(n)=0.0;
hasil=[];
%--------------------------------------------------------
%  mengisi  matrik  tridiagonal  dan  vektor  kolom  ruas  kanan
%(a,b,c,r);
%------------------------------------------------------
%Memasukkan unsur-unsur matrik a(i), b(i), c(i) dan r(i)
%Khusus untuk i=1 dan i=n-1, unsur matriks r(i) berbeda
%dengan yang lain karena ada pengurangan dengan y0 dan yn
%------------------------------------------------------

for i=1:(n-1)
x(i)=x0+i*h;
b(i)=-2.0;
    if (i == 1)
        r(i)=f1(x(i))*h^2-y0;
    else
    if (i == (n-1))
        r(i)=f1(x(i))*h^2-y(n);
       else
         r(i)=f1(x(i))*h^2;
    end
    end
end
for i=1:(n-2)
c(i)=1.0;
end
for i=2:(n-1)
a(i)=1.0;
end
%-------------------------------------------------------
%Mengubah matriks tridiagonal ke bentuk matriks atas yang
%hanya mengandung dua larik yaitu beta(i) dan c(i) serta
%r(i) ke rho(i)
%-------------------------------------------------------
beta(1)=b(1);
rho(1)=r(1);
for i=2:n-1
pengali=a(i)/beta(i-1);
beta(i)=b(i)-pengali*c(i-1);
rho(i)=r(i)-pengali*rho(i-1);
end
%-------------------------------------------------------
%Menghitung y(i) dengan substitusi balik yaitu dihitung
%lebih dahulu y(n-1) dan dilanjutkan ke y(n-2) dan
%seterusnya sampai y(1)
%-------------------------------------------------------
y(n-1)=rho(n-1)/beta(n-1);
for i=2:(n-1)
y(n-i)=(rho(n-i)-c(n-i)*y(n-i+1))/beta(n-i);
end
%-------------------------------------------------------
%Menampilkan hasil
%-------------------------------------------------------
for i=1:n
fprintf('%d   %8.6f    %8.6f\n', i, x(i), y(i));
end
```

M-file untuk menyatakan fungsi ruas kanan peramaan ditulis terpisah di M-file bernama `f1.m` seperti berikut

```
%   f1(x)   adalah   fungsi   rapat   muatan   (fungsi   ruas   kanan
%persamaan)
function f=f1(x)
f=-3*x;
return
```

Hasil Running Fortran, Java dan MATLAB berturut-turut ditunjukkan seperti Gambar 2, Gambar 3 dan Gambar 4.

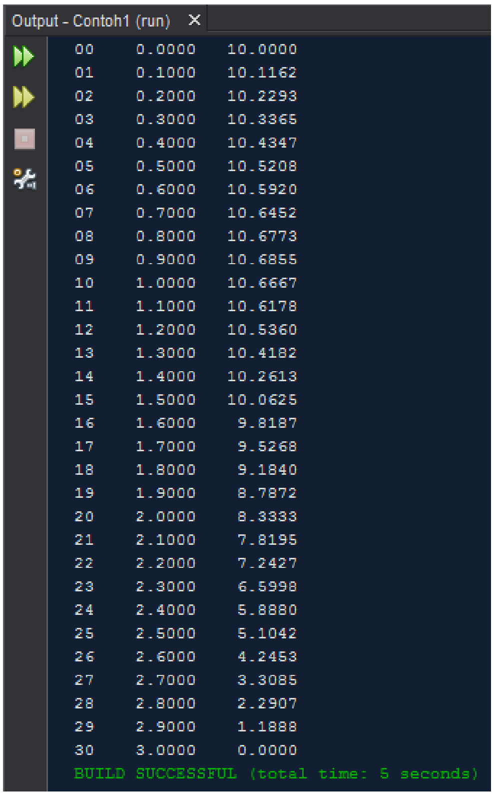


Plato IDE

i	x	y(x)
0	0.000000	10.0000
1	0.100000	10.1162
2	0.200000	10.2293
3	0.300000	10.3365
4	0.400000	10.4347
5	0.500000	10.5208
6	0.600000	10.5920
7	0.700000	10.6452
8	0.800000	10.6773
9	0.900000	10.6855
10	1.00000	10.6667
11	1.10000	10.6178
12	1.20000	10.5360
13	1.30000	10.4182
14	1.40000	10.2613
15	1.50000	10.0625
16	1.60000	9.81866
17	1.70000	9.52683
18	1.80000	9.18400
19	1.90000	8.78716
20	2.00000	8.33333
21	2.10000	7.81950
22	2.20000	7.24266
23	2.30000	6.59983
24	2.40000	5.88800
25	2.50000	5.10416
26	2.60000	4.24533
27	2.70000	3.30850
28	2.80000	2.29067
29	2.90000	1.18883
30	3.00000	0.00000

Press RETURN to close window . . .

Gambar 2. Hasil Running program di bahasa Fortran

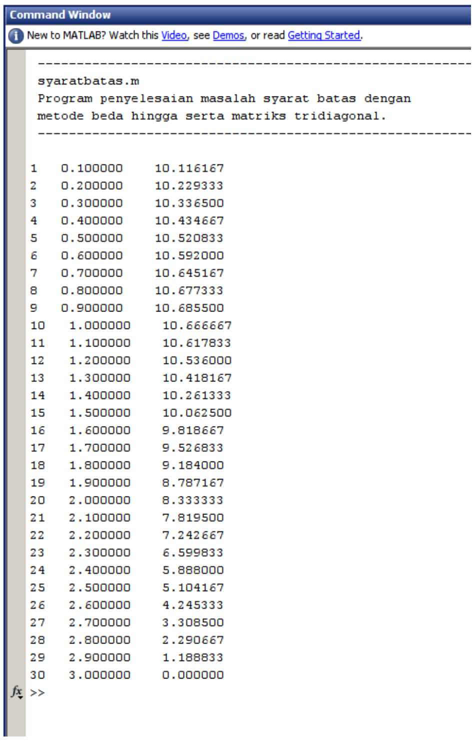


Output - Contoh1 (run)

00	0.0000	10.0000
01	0.1000	10.1162
02	0.2000	10.2293
03	0.3000	10.3365
04	0.4000	10.4347
05	0.5000	10.5208
06	0.6000	10.5920
07	0.7000	10.6452
08	0.8000	10.6773
09	0.9000	10.6855
10	1.0000	10.6667
11	1.1000	10.6178
12	1.2000	10.5360
13	1.3000	10.4182
14	1.4000	10.2613
15	1.5000	10.0625
16	1.6000	9.8187
17	1.7000	9.5268
18	1.8000	9.1840
19	1.9000	8.7872
20	2.0000	8.3333
21	2.1000	7.8195
22	2.2000	7.2427
23	2.3000	6.5998
24	2.4000	5.8880
25	2.5000	5.1042
26	2.6000	4.2453
27	2.7000	3.3085
28	2.8000	2.2907
29	2.9000	1.1888
30	3.0000	0.0000

BUILD SUCCESSFUL (total time: 5 seconds)

Gambar 3. Hasil Running program di bahasa Java



Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

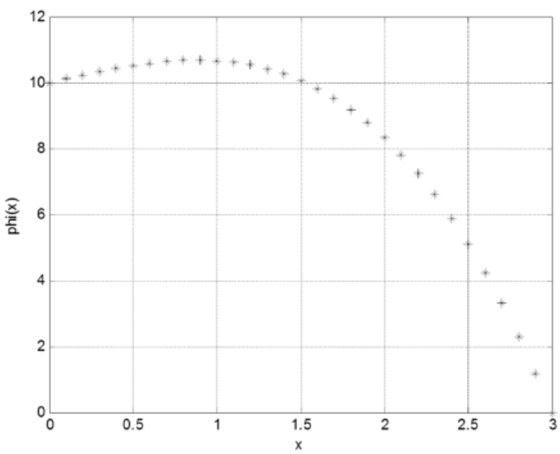
syaratbatas.m
Program penyelesaian masalah syarat batas dengan
metode beda hingga serta matriks tridiagonal.

1	0.100000	10.116167
2	0.200000	10.229333
3	0.300000	10.336500
4	0.400000	10.434667
5	0.500000	10.520833
6	0.600000	10.592000
7	0.700000	10.645167
8	0.800000	10.677333
9	0.900000	10.685500
10	1.000000	10.666667
11	1.100000	10.617833
12	1.200000	10.536000
13	1.300000	10.418167
14	1.400000	10.261333
15	1.500000	10.062500
16	1.600000	9.818667
17	1.700000	9.526833
18	1.800000	9.184000
19	1.900000	8.787167
20	2.000000	8.333333
21	2.100000	7.819500
22	2.200000	7.242667
23	2.300000	6.599833
24	2.400000	5.888000
25	2.500000	5.104167
26	2.600000	4.245333
27	2.700000	3.308500
28	2.800000	2.290667
29	2.900000	1.188833
30	3.000000	0.000000

>>

Gambar 4. Hasil Running program di MATLAB

Gambar 5 melukiskan grafik hasil perhitungan numerik nilai potensial listrik $\phi(x)$ terhadap jarak.



Gambar 5. Nilai potensial listrik terhadap jarak

KESIMPULAN

Dari hasil kajian numerik ini dapat disimpulkan bahwa metode beda hingga yang menghasilkan matriks tridiagonal yang diselesaikan dengan eliminasi Gauss dapat digunakan sebagai salah satu cara menyelesaikan persamaan diferensial biasa, orde dua, linier yang disertai dengan kondisi batas seperti kasus menghitung potensial elektrostatis yang berbentuk persamaan Poisson 1D. Algoritma penyelesaian masalah pada dapat diimplementasikan di beberapa bahasa pemrograman.

DAFTAR PUSTAKA

Koonin, S.E.,. 1990. *Computational Physics - Fortran Version*. Addison-Wesley Pu. Co.Inc. New York.

Lam,C.Y.,1994. *Applied Numerical Methods for Partial Differential Equations*. Prentice Hall, London.

Rubin H.L., Manuel J.P., Cristian C.B., 2007. *Computational Physics.* Wiley VCH Verlag. Weinheim.

Sahid. 2005. *Pengantar Komputasi Numerik dengan MATLAB*. Penerbit Andi. Yogyakarta.

Suarga. 2007. *Fisika Komputasi, Solusi Problema Fisika dengan MATLAB*. Penerbit Andi. Yogyakarta.

Smith,G.D., 1969. *Numerical Methods for Partial Differential Equations*. Oxford University Press. London.

Yang W.Y, dkk. 2005. *Apllied Numerical Methods using MATLAB*. John Wiley & Sons. New Jersey.