Nama : M.Irfansyah
NPM : 2106701255
Kelas : Persamaan Differensial Numerik (B)

TUGAS Akhir Praktikum PD-Numerik

1. Diberikan masalah nilai batas:

$$y'' = \frac{1}{2}(1 - (y')^2 - y\sin x), 0 \le x \le \pi$$
$$y(0) = 2, \;\; y(\pi) = 2$$

dengan solusi eksak:

$$y(x) = 2 + \sin x$$

Gunakan *nonlinear shooting method* dengan $TOL = 10^{-4}$ dan $N = 20$ untuk mengaproksimasi solusi permasalahan tersebut. Lalu, bandingkan dengan solusi eksaknya.

Berdasarkan perintah soal kita akan menyelesaikan permasalahan pada soal menggunakan *nonlinear shooting method*.Berikut adalah algoritma yang digunakan untuk menyelesaikan persoalan diatas.

```
clc; % untuk membersihkan jendela perintah atau konsol
clear all; % menghapus semua variabel dari ruang kerja
close all; % menutup semua jendela gambar atau plot

function [x_i, w_1i, w_2i] = nonlinshoot(f, fy, fyp, a, b, n, alpha, beta, m, tol)  % Mendefinisikan fungsi nonlinearshoot
  h = (b - a)/n;% stepsize
  k = 1; % parameter k
  tk = (beta - alpha)/(b - a); % parameter tk
  x_i = w_1i = w_2i = []; % mendefinisikan variabel variabel xi,w1i dan w2i sebagai sebuah himpunan kosong
  while k <= m % while condition agar fungsi berjalan
    w = [alpha;tk]; % Mendefinisikan w(1,0) dan w(2,0)
    u = [0,1]; % Mendefinisikan nilai u1 dan u2
    for i = 1:n
      x = a + (i-1)*h; % Mendefinisikan nilai x
      % proses rungkuta orde 4
      k_11 = h*w(2,i);
      k_12 = h*f(x, w(1,i), w(2,i));

      k_21 = h*(w(2,i)+(k_12/2));
      k_22 = h*f((x+(h/2)), (w(1,i)+(k_11/2)), (w(2,i)+(k_12/2)));

      k_31 = h*(w(2,i)+(k_22/2));
      k_32 = h*f((x+(h/2)), (w(1,i)+(k_21/2)), (w(2,i)+(k_22/2)));

      k_41 = h*(w(2,i)+k_32);
      k_42 = h*f((x+h), (w(1,i)+k_31), (w(2,i)+k_32));

      w(1,i+1) = w(1,i) + ((k_11 + 2*k_21 + 2*k_31 + k_41)/6);
      w(2,i+1) = w(2,i) + ((k_12 + 2*k_22 + 2*k_32 + k_42)/6);

      kp_11 = h*u(2);
      kp_12 = h*(fy(x, w(1,i), w(2,i))*u(1) + fyp(x, w(1,i), w(2,i))*u(2));

      kp_21 = h*(u(2) + (kp_12/2));
      kp_22 = h*(fy((x+(h/2)), w(1,i), w(2,i))*u(1) + fyp((x+(h/2)), w(1,i), w(2,i))*(u(2) + (kp_12/2)));

      kp_31 = h*(u(2)+(kp_22/2));
      kp_32 = h*(fy((x+(h/2)), w(1,i), w(2,i))*(u(1) + (kp_21/2)) + fyp((x+(h/2)), w(1,i), w(2,i))*(u(2) + (kp_22/2)));

      kp_41 = h*(u(2)+kp_32);
      kp_42 = h*(fy((x+h), w(1,i), w(2,i))*(u(1)+kp_31) + fyp((x+h), w(1,i), w(2,i))*(u(2) + kp_32));

      u(1) = u(1) + (kp_11 + 2*kp_21 + 2*kp_31 + kp_41)/6;
      u(2) = u(2) + (kp_12 + 2*kp_22 + 2*kp_32 + kp_42)/6;
    endfor

  if abs(w(1,n+1) - beta) <= tol       % jika sudah mencapai batas toleransi maka program berhenti
    for i = 1:(n+1)
      x = a+(i-1)*h;
      x_i(i) = x;
      w_1i(i) = w(1,i);
```

Nama : M.Irfansyah
NPM : 2106701255
Kelas : Persamaan Differensial Numerik (B)

```
      w_2i(i) = w(2,i);
    endfor
    return
  endif
  tk = tk-((w(1,n+1) - beta)/u(1));
  k = k + 1;
  endwhile
  disp('max iteration')
endfunction

xi = w1i = w2i = []; %mendefinisikan variabel variabel xi,w1i dan w2i sebagai sebuah himpunan kosong
f = @(x, y, yp) ((1/2)*(1-((yp) ^ 2)- y*sin(x))); %mendefinisikan fungsi
fy = @(x, y, yp) (-(sin(x)) / 2); % melakukan turunan terhadap y
fyp = @(x, y, yp) (-yp); % melakukan turunan terhadap yp
a = 0; % mendefinisikan batas bawah fungsi
b = pi; % mendefinisikan batas atas fungsi
alpha = 2; % Mendefinisikan nilai awal fungsi
beta = 2; % Mendefinisikan nilai batas fungsi
tol = 10^(-4); % batas toleransi error
n = 20; % mendefinisikan parameter n
m = 10; % mendefinisikan parameter m

[xi, w1i, w2i] = nonlinshoot(f, fy, fyp, a, b, n, alpha, beta, m, tol); % Mendefinisikan fungsi nonlinearshoot

% Mencari solusi eksak
sln = @(x) (2 + sin(x));
w = [];
for i = 1:length(xi)
  w(i) = sln(xi(i));
endfor

[xi', w1i', w', abs(w-w1i)'] % Mencetak tabelx,aproksimasi w, solusi eksak dan eror


hold on;
fplot(sln, [0,pi], 'k'); % plot fungsi eksaknya
scatter(xi, w1i, 'r'); % xi-wli scatter plot
legend('Solusi Eksak', 'Metode Nonlinear Shooting'); % memberi legenda pada plot
title('Nonlinear shooting method'); % judul plot
```

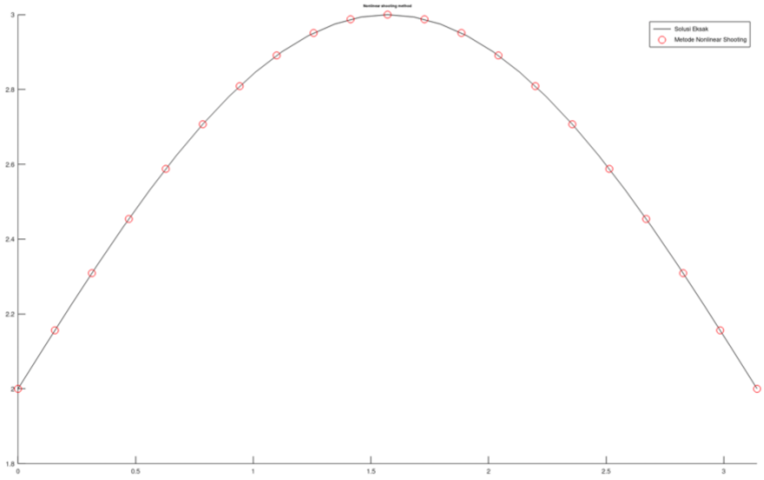
Dari metode ini didapatkan hasil aproksimasi sebagai berikut:

0	2.0000	2.0000	0
0.1571	2.1564	2.1564	0.0000
0.3142	2.3090	2.3090	0.0000
0.4712	2.4540	2.4540	0.0000
0.6283	2.5878	2.5878	0.0000
0.7854	2.7071	2.7071	0.0000
0.9425	2.8090	2.8090	0.0000
1.0996	2.8910	2.8910	0.0000
1.2566	2.9510	2.9511	0.0000
1.4137	2.9877	2.9877	0.0000
1.5708	3.0000	3.0000	0.0000
1.7279	2.9877	2.9877	0.0000
1.8850	2.9510	2.9511	0.0000
2.0420	2.8910	2.8910	0.0000
2.1991	2.8090	2.8090	0.0000
2.3562	2.7071	2.7071	0.0000
2.5133	2.5878	2.5878	0.0000
2.6704	2.4540	2.4540	0.0000
2.8274	2.3090	2.3090	0.0000
2.9845	2.1564	2.1564	0.0000
3.1416	2.0000	2.0000	0.0000

Nama : M.Irfansyah
NPM : 2106701255
Kelas : Persamaan Differensial Numerik (B)

(Keterangan: Tabel 1 menunjukan nilai x_i, Tabel 2 menunjukkan aproksimasi $w_{1,i}$, Tabel 3 menunjukkan solusi eksak $(y(x_i))$dan Tabel 4 menunjukkan eror, penomoran tabel dari kiri ke kanan).

Algoritma ini menghasilkan grafik berikut:



Dari grafik yang telah didapatkan, diketahui bahwa:

1.Fungsi terbuka kebawah dengan range nilai x yaitu $[0, \pi]$

2. Eror yang didapat cukup kecil sehingga bisa dikatakan bahwa metode yang digunakan cukup akurat.

2. Gunakan metode beda hingga nonlinear dengan $h = 0.1$ dan toleransi 10^{-4} untuk mengaproksimasi BVP berikut:

$$\begin{aligned} y'' &= y' + 2(y - \ln\ x) \quad -\frac{1}{x},\ 2 \leq x \leq 3 \\ y(2) &= \frac{1}{2} + \ln\ 2,\ y(3) = \frac{1}{3} + \ln\ 3 \end{aligned}$$

Berdasarkan perintah soal kita akan menyelesaikan permasalahan pada soal menggunakan metode beda hingga nonlinear.Berikut adalah algoritma yang digunakan untuk menyelesaikan persoalan diatas.

Nama : M.Irfansyah
NPM : 2106701255
Kelas : Persamaan Differensial Numerik (B)

```
clc;% untuk membersihkan jendela perintah atau konsol
clear all;% menghapus semua variabel dari ruang kerja
close all;% menutup semua jendela gambar atau plot

function [t_grid,w]=nonlinear_FDM_naive(f,f_y,f_yp,a,b,n,alpha,beta,max_iter,TOL) % Mendefinisikan beda hingga nonlinear
  h=(b-a)/(n+1); %stepsize
  w=zeros(n,1); %vektor solusi aproksimasi
  t_grid=[a:h:b]; %Membuat mesh_point
  J=zeros(n,n); %Membuat matriks jacobian
  F=zeros(n,1); %Membuat vektor fungsi  F=(f_1,f_2,...,f_n) yang dievaluasi di x_k

  for i=1:n %inisialisasi solusi awal
    w(i)=alpha+i*(beta-alpha)/(b-a)*h;
  endfor
  k=1;
  while k<=max_iter %lakukan iterasi jika masih belum didapat kriteria stopnya

    %solve nonlinear sistem tersebut dengan metode newton
    x=a+h;
    %kontruksi matriks Jacobian, dan vektor F-nya
    t=(w(2)-alpha)/(2*h);
    J(1,1)=2+h^2*f_y(x,w(1),t); %main diagoanal
    J(1,2)=-1+(h/2)*f_yp(x,w(1),t); %right diagonal
    F(1)=(2*w(1)-w(2)-alpha+h^2*f(x,w(1),t));
    for i =2:n-1
      x=a+i*h;
      t=(w(i+1)-w(i-1))/(2*h);
      J(i,i)=2+h^2*f_y(x,w(i),t); %main diagoanal
      J(i,i+1)=-1+(h/2)*f_yp(x,w(i),t); %main diagoanal
      J(i,i-1)=-1-(h/2)*f_yp(x,w(i),t); %left diagoanal
      F(i)=(2*w(i)-w(i+1)-w(i-1)+h^2*f(x,w(i),t));
    endfor
     x=b-h;
     t=(beta-w(n-1))/(2*h);
     J(n,n)=2+h^2*f_y(x,w(n),t); %main diagonal
     J(n,n-1)=-1-(h/2)*f_yp(x,w(n),t); %right diagonal
     F(n)=(2*w(n)-w(n-1)-beta+h^2*f(x,w(n),t));



    v=inverse(J)*F; %vector v adalah product dari J^-1 F
    w= w-v; % lakukan update nilai pada w

    if norm(v,2)<= TOL %kriteria stop jika norm(v)<=toleransinya
      break;
     else
        k=k+1; %jika belum memenuhi kriteria stop terus lanjut iterasinya (memperbaiki nilai w)
    endif
  endwhile
  w=[alpha ; w ; beta]; %konstruksi akhir w
  t_grid=transpose(t_grid); % %transpose meshpoint
```

```
  % untuk konsistensi dimensi saja

endfunction


f=@(x,y,yp) yp+2*(y-log(x))-1/x ; %mendefinisikan fungsi
f_y=@(x,y,yp) 2*(y-log(x)); %melakukan turunan terhadap y
f_yp=@(x,y,yp) 1; %melakukan turunan terhadap yp
a=2; %mendefinisikan batas bawah fungsi
b=3; %mendefinisikan batas atas fungsi
alpha=0.5+log(2); %Mendefinisikan nilai awal fungsi
beta=1/3+ log(3); %Mendefinisikan nilai batas fungsi
n=9; %banyaknya partisi (pilih n=9 sehingga h=0.1)
max_iter=30; %Memilih maksimal iterasi
TOL=10^(-4); %batas toleransi error

%memanggil fungsi nonlinear_FDM_naive
[x_grid,w]=nonlinear_FDM_naive(f,f_y,f_yp,a,b,n,alpha,beta,max_iter,TOL)
f_anal= @(x) 1./x +log(x); %sol analitik

%membuat grafiknya
fplot(f_anal, [a,b],'b')  % plot fungsi eksaknya
hold on;
scatter(x_grid,w,'r') % scatter plot
legend('solusi analitik', 'solusi linear FDM'); % memberi legenda pada plot
legend("location", "northwest");



%membuat tabel saja.
sol_anal=f_anal(x_grid); %sol analitik di meshpoint
error=abs(w-sol_anal); %error
[x_grid,w,sol_anal,error]
```

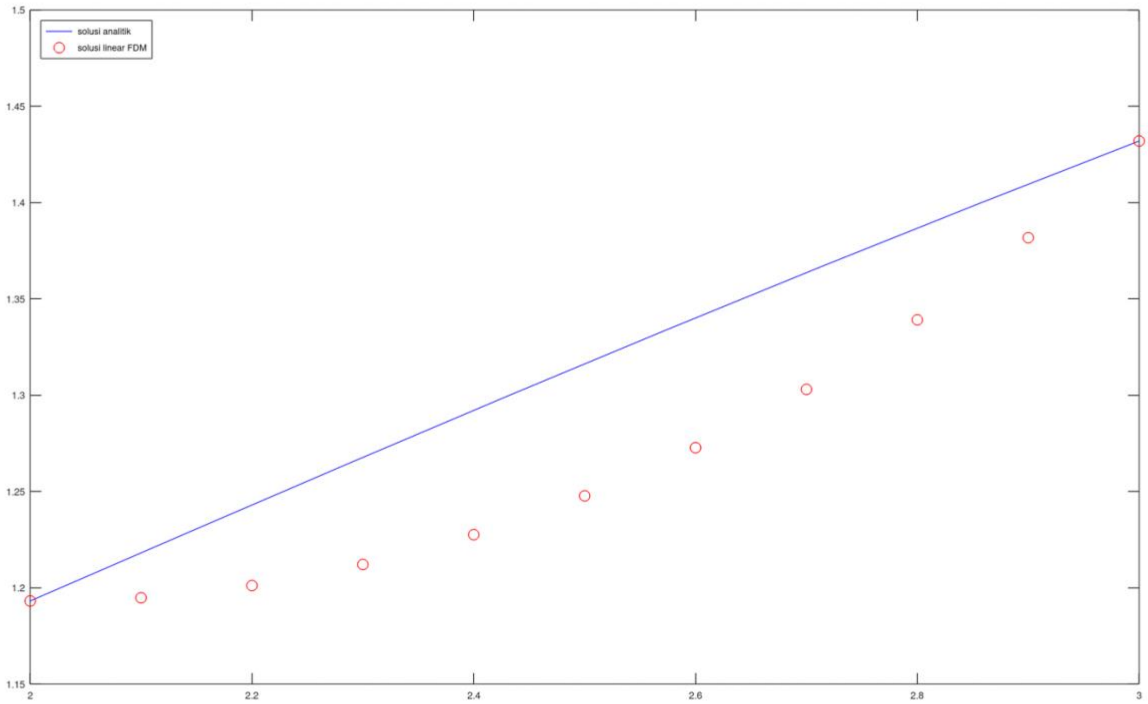
Nama : M.Irfansyah
NPM : 2106701255
Kelas : Persamaan Differensial Numerik (B)

Dari metode ini didapatkan hasil aproksimasi sebagai berikut:

2.0000	1.1931	1.1931	0
2.1000	1.1948	1.2181	0.0233
2.2000	1.2012	1.2430	0.0418
2.3000	1.2121	1.2677	0.0556
2.4000	1.2276	1.2921	0.0646
2.5000	1.2477	1.3163	0.0686
2.6000	1.2727	1.3401	0.0674
2.7000	1.3030	1.3636	0.0606
2.8000	1.3391	1.3868	0.0477
2.9000	1.3818	1.4095	0.0278
3.0000	1.4319	1.4319	0

(Keterangan: Tabel 1 menunjukan nilai x_grid, Tabel 2 menunjukkan aproksimasi w, Tabel 3 menunjukkan solusi analitik dan Tabel 4 menunjukkan eror, penomoran tabel dari kiri ke kanan).

Algoritma ini menghasilkan grafik berikut:



Dari grafik yang telah didapatkan, diketahui bahwa:

1.Nilai eksak (solusi analitik) mendekati nilai aproksimasi hanya ketika x nya berada di nilai batas

Nama : M.Irfansyah
NPM : 2106701255
Kelas : Persamaan Differensial Numerik (B)

2.Selain pada nilai nilai batasnya, solusi analitik bernilai lebih besar dari pada hasil aproksimasi

3. Diberikan masalah persamaan *transport*:

$$u_t + 2u_x = 0, \qquad 0 < x < 2,\ t > 0;$$
$$u(x,0) = e^{-x^2}, \qquad 0 \le x \le 2.$$

Hitung nilai aproksimasinya menggunakan ketiga metode yang telah diajarkan (Metode *upwind*, metode Richardson, metode Lax). *Output* yang dihasilkan minimal berupa 4 *plot figure* yang memuat solusi eksak dan solusi aproksimasi dari setiap metode.

Catatan: Jika dibutuhkan syarat batas, ambil dari solusi eksak.

Berdasarkan perintah soal kita akan menyelesaikan permasalahan pada soal menggunakan tiga metode yaitu metode *upwind*, metode Richardson,dan metode Lax.

Berikut adalah algoritma yang digunakan untuk menyelesaikan persoalan diatas.

```
clc;% untuk membersihkan jendela perintah atau konsol
clear all;% menghapus semua variabel dari ruang kerja
close all;% menutup semua jendela gambar atau plot


function [x, t, u] = courant(d, f, lb, xb, xu, tb, tu, dx, dt) % Mendefinisikan fungsi courant(upwind)
  t = tb:dt:tu;
  x = xb:dx:xu;
  u = [];

  for j = 1:length(x)
    u(j, 1) = f(x(j));
  endfor

  for n = 1:length(t)
    u(1, n) = lb(t(n));
  endfor

  c = d * dt / dx;
  for n = 1:length(t)-1
    for j = 2:length(x)
      u(j, n+1) = (1-c) * u(j, n) + c * u(j-1, n);
    endfor
  endfor
endfunction

function [x, t, u] = richardson(d, f, lb, rb, xb, xu, tb, tu, dx, dt) % Mendefinisikan fungsi richardson
  t = tb:dt:tu;
  x = xb:dx:xu;
  nt = length(t);
  nx = length(x);
  u = [];

  for j = 1:nx
    u(j, 1) = f(x(j));
  endfor

  for n = 1:nt
    u(1, n) = lb(t(n));
    u(nx, n) = rb(t(n));
  endfor

  c = (d*dt) / (2*dx);
  for n = 1:nt-1
    for j = 2:nx-1
      u(j, n+1) = u(j, n) - (c * (u(j+1, n) - u(j-1, n)));
    endfor
  endfor
endfunction
```

Nama : M.Irfansyah
NPM : 2106701255
Kelas : Persamaan Differensial Numerik (B)

```
function [x, t, u] = lax(d, f, lb, rb, xb, xu, tb, tu, dx, dt) % Mendefinisikan fungsi lax
  t = tb:dt:tu;
  x = xb:dx:xu;
  nt = length(t);
  nx = length(x);
  u = [];

  for j = 1:nx
    u(j, 1) = f(x(j));
  endfor

  for n = 1:nt
    u(1, n) = lb(t(n));
    u(nx, n) = rb(t(n));
  endfor

  c = (d*dt) / (2*dx);
  for n = 1:nt-1
    for j = 2:nx-1
      u(j, n+1) = (u(j+1, n) + u(j-1, n))/2 - (c * (u(j+1, n) - u(j-1, n)));
    endfor
  endfor
endfunction
```

Kemudian akan ditentukan beberapa parameter yang dibutuhkan:

$d = 2$ diperoleh dari $u_t + 2u_x = 0 \rightarrow u_t + du_x = 0$

$lb = e^{-(2t)^2}$

$rb = e^{-(2-2t)^2}$ yang mana lb dan rb diperoleh dari solusi eksak

$dx = 0.5$ (step size x)

$dt = 0.1$ (step size t)

Sehingga algoritmanya menjadi

Nama : M.Irfansyah
NPM : 2106701255
Kelas : Persamaan Differensial Numerik (B)

```
d = 2; # d adalah nilai d pada ut + dux = 0.
f = @(x) e^(-x.^2); # f adalah nilai awal f(x) pada persamaan transport
lb = @(t) e^(-(2*t).^2); # syarat batas kiri pada persamaan transport
rb = @(t) e^(-(2-2*t).^2) # syarat batas kanan pada persamaan transport
xb = 0; # batas bawah untuk variabel x
xu = 2; # batas atas untuk variabel x
tb = 0; # batas bawah untuk variabel t
tu = 1; # batas atas untuk variabel t
dx = 0.5; # stepsize x
dt = 0.1; # stepsize t

# hasil aproksimasi
[x, t, u1] = courant(d, f, lb, xb, xu, tb, tu, dx, dt);
[x, t, u2] = richardson(d, f, lb, rb, xb, xu, tb, tu, dx, dt);
[x, t, u3] = lax(d, f, lb, rb, xb, xu, tb, tu, dx, dt);

# print
[x'] # nilai x
[t'] # nilai t
[u1'] # hasil aproksimasi metode courant
[u2'] # hasil aproksimasi metode richardson
[u3'] # hasil aproksimasi metode lax

sol = @(x, t) e^(-1*(x-2*t)^2); # solusi eksak
for j = 1:length(x)
  for n = 1:length(t)
    y(j, n) = sol(x(j), t(n));
  endfor
endfor

# Plot 1 (Metode Upwind)
figure(1);
mesh(x, t, u1');
xlabel('x');
ylabel('t');
zlabel('u');
title("Solusi Metode Upwind");

# Plot 2 (Metode Richardson)
figure(2);
mesh(x, t, u2');
xlabel('x');
ylabel('t');
zlabel('u');
title("Solusi Metode Richardson");
```

Nama : M.Irfansyah
NPM : 2106701255
Kelas : Persamaan Differensial Numerik (B)

```
# Plot 3 (Metode Lax)
figure(3);
mesh(x, t, u3');
xlabel('x');
ylabel('t');
zlabel('u');
title("Solusi Metode Lax");

# Plot 4 (Solusi Eksak)
figure(4);
mesh(x, t, y');
xlabel('x');
ylabel('t');
zlabel('u');
title("Solusi Eksak")
```

Dari metode ini didapatkan hasil aproksimasi sebagai berikut:

1.000000	0.778801	0.367879	0.105399	0.018316
0.960789	0.867280	0.532248	0.210391	0.053149
0.852144	0.904684	0.666261	0.339134	0.116046
0.697676	0.883668	0.761630	0.469985	0.205281
0.527292	0.809271	0.810445	0.586643	0.311163
0.367879	0.696480	0.809976	0.676164	0.421355
0.236928	0.565040	0.764577	0.729689	0.523278
0.140858	0.433795	0.684762	0.743644	0.605842
0.077305	0.316620	0.584375	0.720091	0.660963
0.039164	0.220894	0.477273	0.665805	0.684614
0.018316	0.148202	0.374722	0.590392	0.677091

Hasil metode aproksimasi courant (upwind)

1.000000	0.778801	0.367879	0.105399	0.018316
0.960789	0.905225	0.502560	0.175312	0.039164
0.852144	0.996871	0.648542	0.267991	0.077305
0.697676	1.037591	0.794318	0.382239	0.140858
0.527292	1.018263	0.925389	0.512931	0.236928
0.367879	0.938643	1.026455	0.650623	0.367879
0.236928	0.806928	1.084059	0.782338	0.527292
0.140858	0.637502	1.088977	0.893691	0.697676
0.077305	0.447878	1.037739	0.971952	0.852144
0.039164	0.255791	0.932925	1.009071	0.960789
0.018316	0.077039	0.782269	1.003498	1.000000

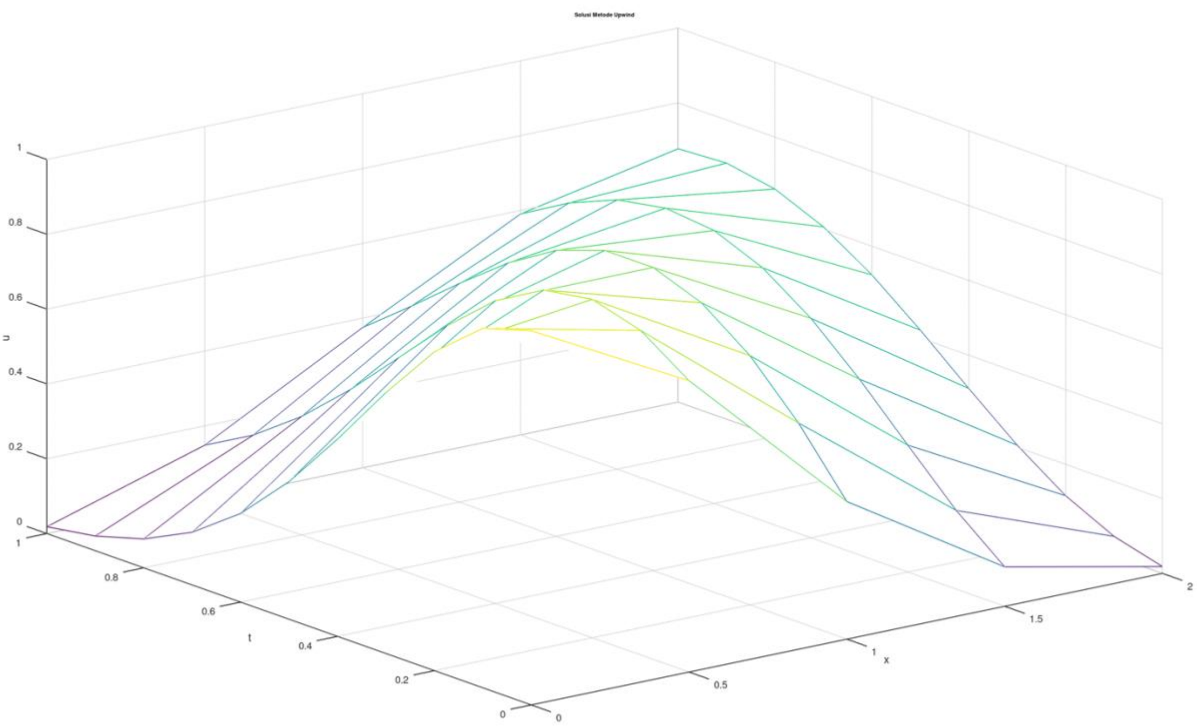
Nama : M.Irfansyah
NPM : 2106701255
Kelas : Persamaan Differensial Numerik (B)

Hasil metode aproksimasi Richardson

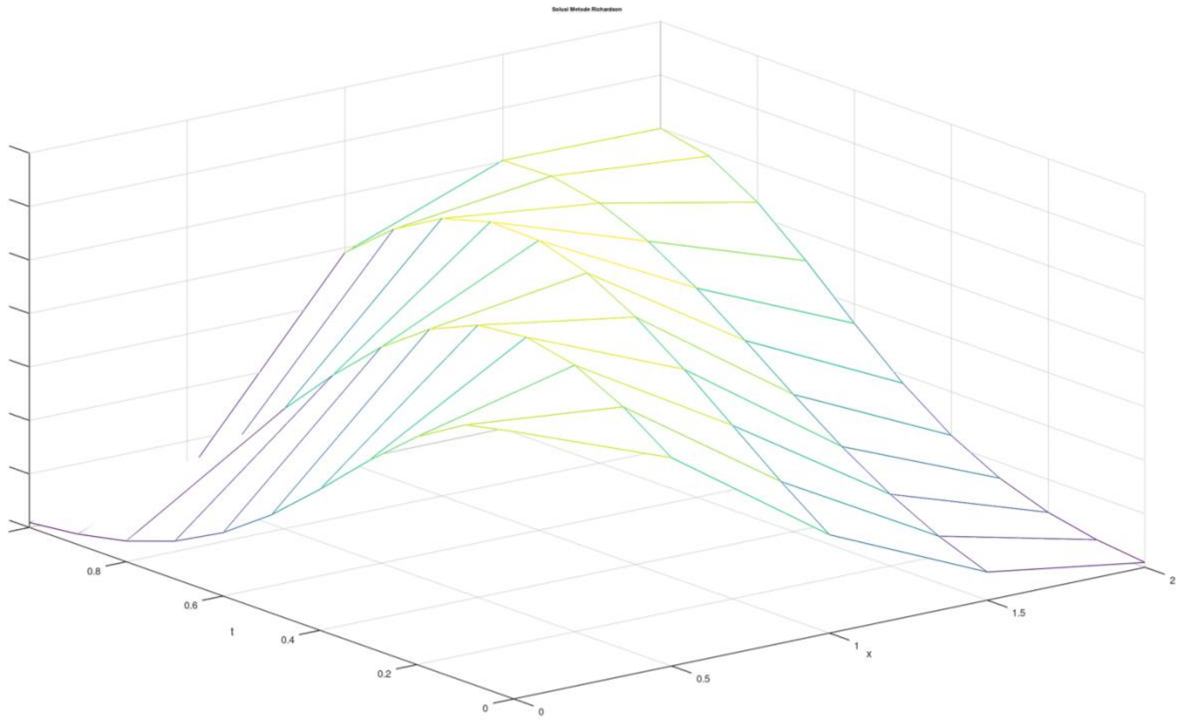
1.000000	0.778801	0.367879	0.105399	0.018316
0.960789	0.810364	0.576780	0.263010	0.039164
0.852144	0.845587	0.646158	0.415495	0.077305
0.697676	0.790348	0.716559	0.475502	0.140858
0.527292	0.703341	0.695894	0.543849	0.236928
0.367879	0.577873	0.655494	0.558204	0.367879
0.236928	0.454164	0.571972	0.569209	0.527292
0.140858	0.337441	0.488677	0.558568	0.697676
0.077305	0.245204	0.403779	0.551377	0.852144
0.039164	0.175247	0.337056	0.538289	0.960789
0.018316	0.128532	0.284160	0.524176	1.000000

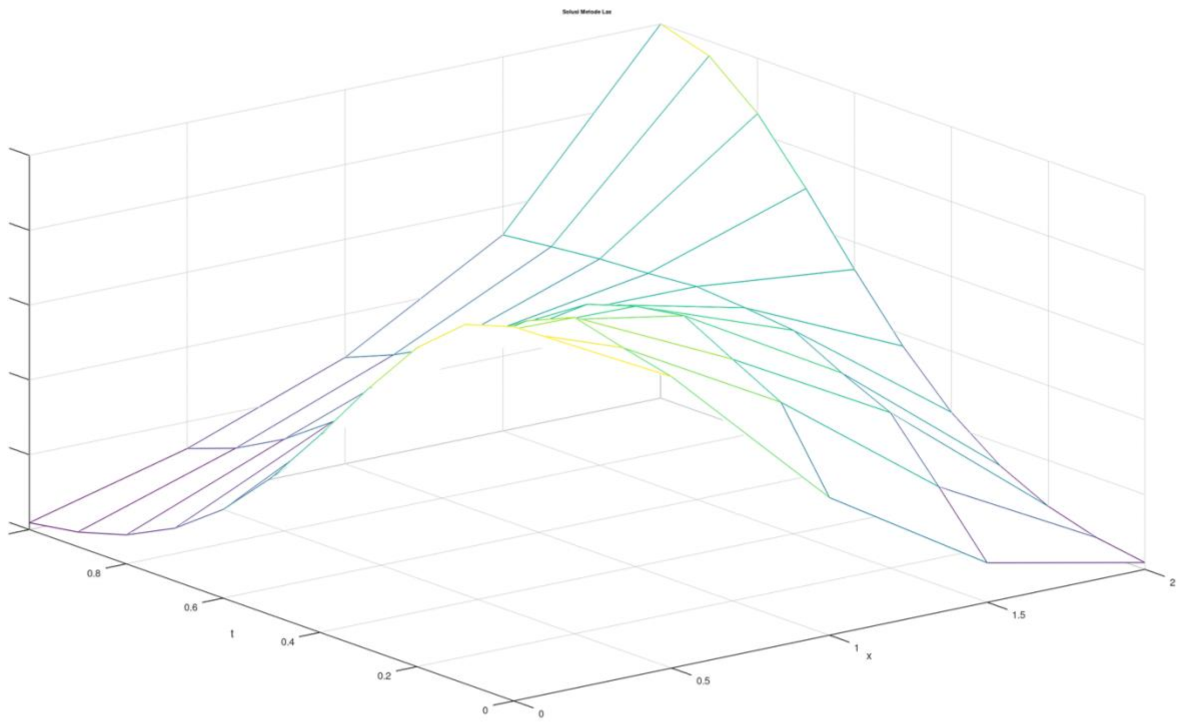
Hasil metode aproksimasi Lax

Algoritma ini menghasilkan grafik berikut:

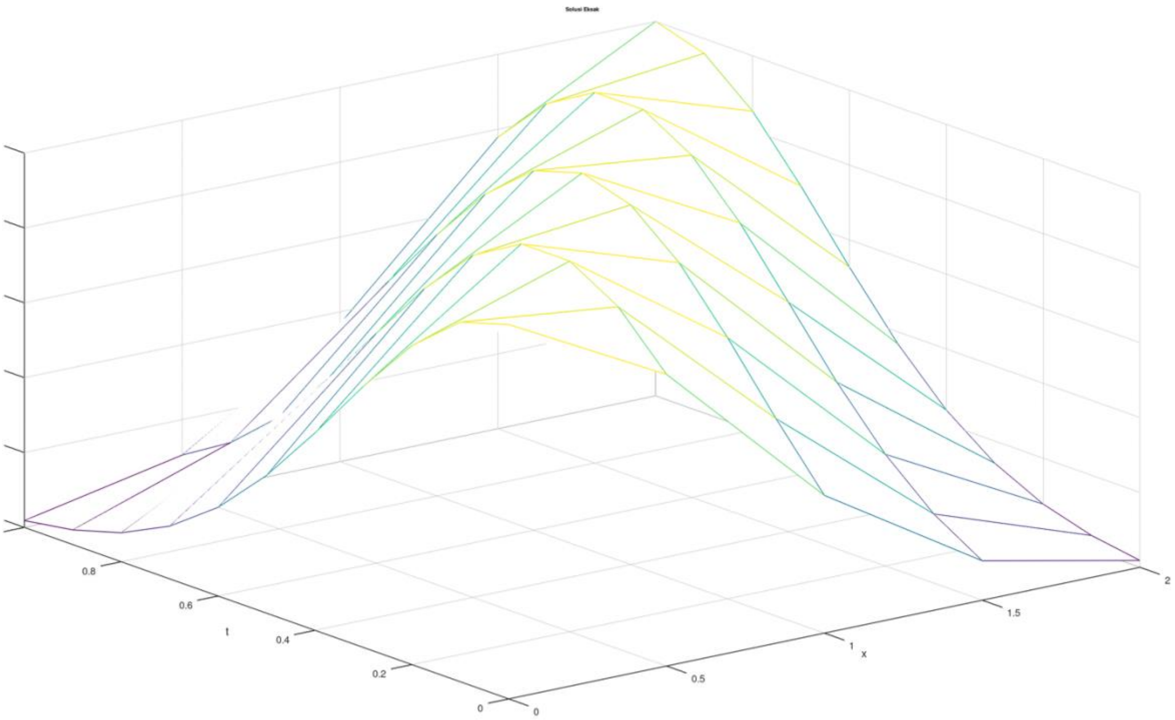


Nama : M.Irfansyah
NPM : 2106701255
Kelas : Persamaan Differensial Numerik (B)





Nama : M.Irfansyah
NPM : 2106701255
Kelas : Persamaan Differensial Numerik (B)



Dari grafik yang telah didapatkan, diketahui bahwa:

1.$C = \frac{d(\Delta t)}{\Delta x} = \frac{2(0{,}1)}{0{,}5} = 0{,}4$, hal ini mengisyaratkan bahwa $0 \leq C \leq 1$, yang menunjukkan bahwa metode upwind stabil

2.Kita tahu syarat suatu metode lax stabil yaitu $C \leq 1$ yang mana syarat ini pun terpenuhi, sehingga metode lax tabil

3.Metode Richardson selalu tidak stabil ketika $|\rho| \geq 1$