Skema Beda Hingga untuk Persamaan Transport

Persamaan transport adalah salah satu bentuk PDP paling sederhana. Persamaan transport memiliki bentuk:

$$u_t + du_x = 0, \qquad (x,t) \in [0,L] \times [0,T],$$
$$u(x,0) = f(x), \qquad 0 \le x \le L.$$

Solusi eksak dari persamaan transport adalah $u(x,t) = f(x - dt)$. Interval x dan t dipartisi dengan *stepsize* Δx dan Δt berturut-turut sehingga diperoleh $x_j = (j-1)\Delta x,\ j = 1, 2, \dots, N_x$, dan $t_n = (n-1)\Delta t,\ n = 1, 2, \dots, N_t$. Dari sini diperoleh suatu *grid* dengan titik-titik (x_j, t_n). Untuk nilai hampiran di titik-titik *grid* tersebut, akan digunakan notasi $u_j^n \equiv u(x_j, t_n)$.

Akan dijelaskan tiga metode untuk mengaproksimasi persamaan transport:

- Metode Courant-Isaacson-Rees (FTBS, upwind)
- Metode Richardson (FTCS)
- Metode Lax

Metode Courant-Isaacson-Rees

Metode ini adalah metode FTBS (*Forward Time Backward Space*) yang juga dikenal dengan metode *upwind* dengan akurasi $O(\Delta t, \Delta x)$. Metode ini memiliki persamaan beda hingga

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + d\,\frac{u_j^n - u_{j-1}^n}{\Delta x} = 0.$$

Jika dituliskan dalam *term* u_j^{n+1} menjadi

$$u_j^{n+1} = (1 - C)u_j^n + Cu_{j-1}^n, \qquad C \equiv \frac{d\Delta t}{\Delta x}$$

Metode ini membutuhkan syarat batas kiri.

Berikut kode algoritma metode Courant-Isaacson-Rees menggunakan Octave.

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

  for j = 1:length(x)
```

```
    u(j, 1) = f(x(j));
  end

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

  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);
    end
  end
end
```

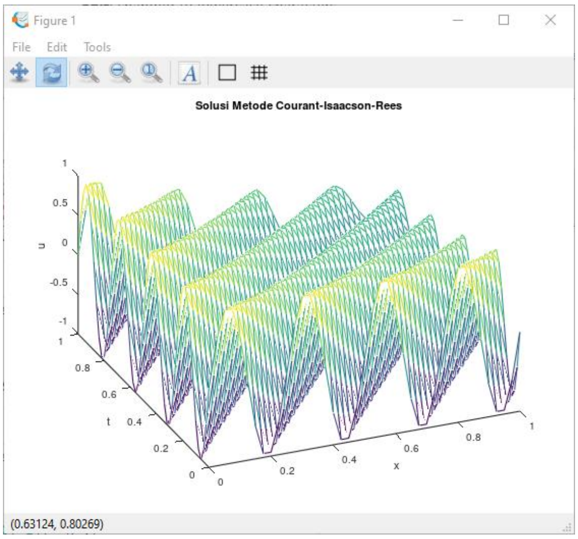
Penjelasan input:

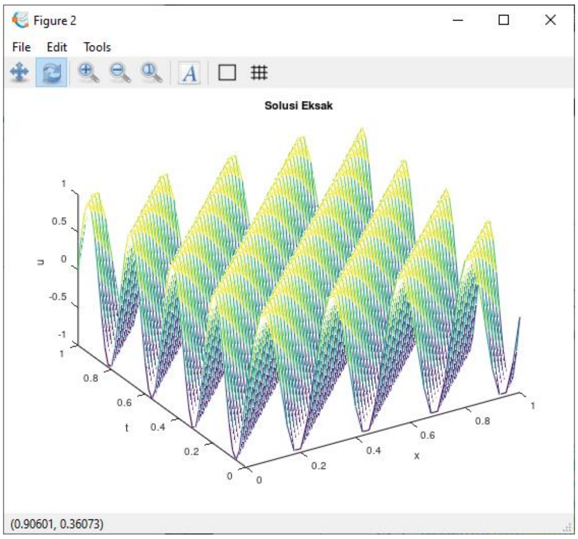
- d adalah nilai d pada $u_t + du_x = 0$.
- f adalah nilai awal $f(x)$ pada persamaan transport.
- lb adalah syarat batas kiri pada persamaan transport.
- xb dan xu berturut-turut adalah batas bawah dan atas untuk variabel x.
- tb dan tu serupa, namun untuk variabel t.
- dx dan dt berturut-turut adalah *stepsize* Δx dan Δt.

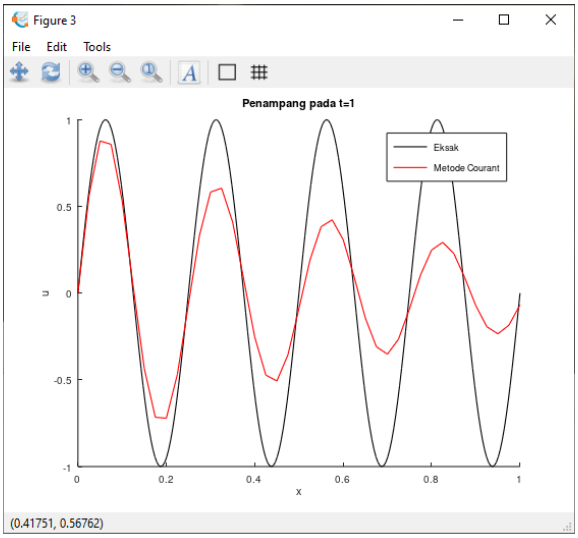
Dalam membandingkan dengan solusi eksak, sulit jika kita menumpuk *plot* seperti biasa. Kita akan menggunakan *syntax* `figure(n)` pada Octave untuk membuat lebih dari satu jendela *plot* untuk kebutuhannya masing-masing. Variabel sol akan digunakan untuk menyimpan solusi eksak dari persamaan transport.

```
clc;
clear all;
close all;
format long;

d = 1;
f = @(x) sin(8*pi*x);
lb = @(t) sin(-8*pi*t);
xb = 0;
xu = 1;
tb = 0;
tu = 1;
dx = 0.025;
dt = 0.02;

[x, t, u] = courant(d, f, lb, xb, xu, tb, tu, dx, dt);

sol = @(x, t) sin(8*pi*(x-t));
```

```
for j = 1:length(x)
  for n = 1:length(t)
    y(j, n) = sol(x(j), t(n));
  endfor
endfor

u1 = @(x) sin(8*pi*(x-1));

figure(1);
hold on;
mesh(x, t, u');
xlabel('x');
ylabel('t');
zlabel('u');
title("Solusi Metode Courant-Isaacson-Rees");

figure(2);
hold on;
mesh(x, t, y');
xlabel('x');
ylabel('t');
zlabel('u');
title("Solusi Eksak")

figure(3);
hold on;
fplot(u1, [0, 1], 'k');
plot(x, u(:, length(t)), 'r');
legend("Eksak", "Metode Courant");
title("Penampang pada t=1");
```







Kita membuat 3 jendela *plot* masing-masing untuk solusi eksak, solusi aproksimasi, dan penampang keduanya saat $t = 1$. Terlihat pada *figure* 3 bahwa solusi aproksimasi mengalami *damping* seiring $x \to 1$.

Kalian dapat memutar *plot* 3D yang dihasilkan dengan meng-klik ikon  pada jendela *figure*. Untuk menutup semua jendela *figure* yang muncul, gunakan *syntax* `close all;` pada *command window*.

Metode Richardson

Metode ini adalah metode FTCS (*Forward Time Center Space*) dengan akurasi $O(\Delta t, \Delta x^2)$. Metode ini memiliki persamaan beda hingga

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + d\,\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0.$$

Jika dituliskan dalam *term* u_j^{n+1} menjadi

$$u_j^{n+1} = u_j^n + C\left(u_{j+1}^n - u_{j-1}^n\right), \qquad C \equiv \frac{d\Delta t}{2\Delta x}$$

Metode ini membutuhkan syarat batas kiri dan kanan.

Berikut kode algoritma metode Lax menggunakan Octave.

```
function [x, t, u] = richardson(d, f, lb, rb, xb, xu, tb, tu,
dx, dt)
  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));
  end

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

  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)));
    end
  end
end
```

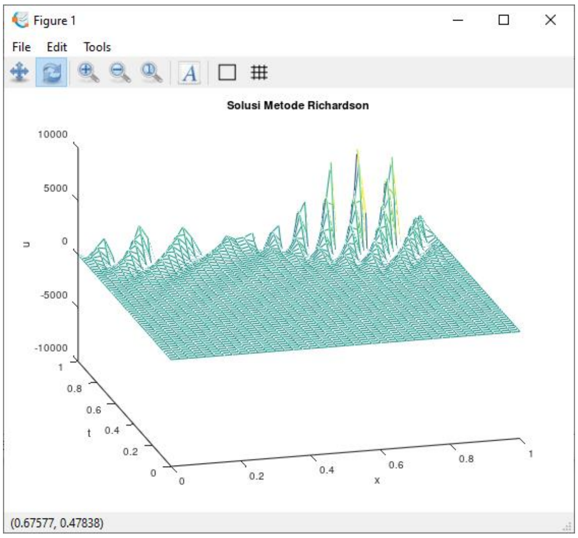
Penjelasan input:

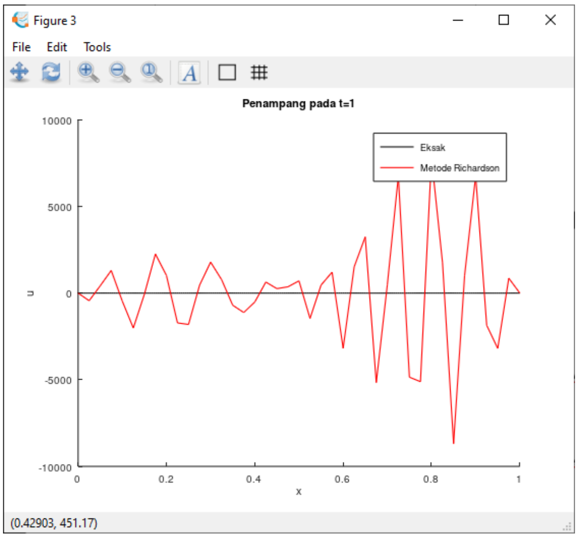
- d adalah nilai d pada $u_t + du_x = 0$.
- f adalah nilai awal $f(x)$ pada persamaan transport.
- lb dan rb berturut-turut adalah syarat batas kiri dan kanan pada persamaan transport.
- xb dan xu berturut-turut adalah batas bawah dan atas untuk variabel x.
- tb dan tu serupa, namun untuk variabel t.
- dx dan dt berturut-turut adalah *stepsize* Δx dan Δt.

Visualisasi untuk hasilnya:

```
clc;
clear all;
close all;
format long;

d = 1;
f = @(x) sin(8*pi*x);
lb = @(t) sin(-8*pi*t);
xb = 0;
xu = 1;
tb = 0;
tu = 1;
dx = 0.025;
dt = 0.02;

[x, t, u] = courant(d, f, lb, xb, xu, tb, tu, dx, dt);
```

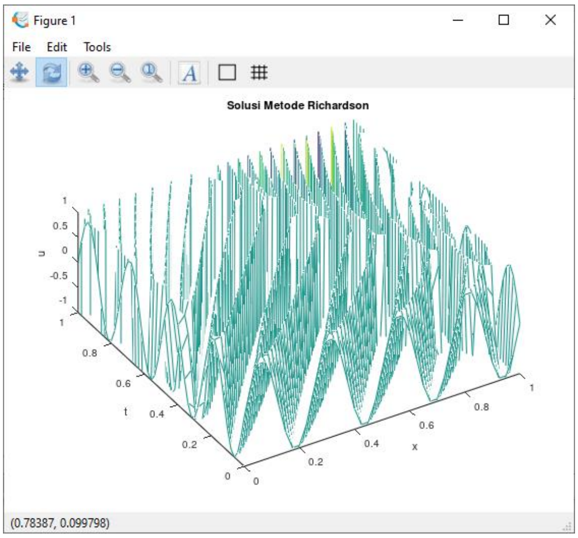
```
sol = @(x, t) sin(8*pi*(x-t));

for j = 1:length(x)
  for n = 1:length(t)
    y(j, n) = sol(x(j), t(n));
  endfor
endfor

u1 = @(x) sin(8*pi*(x-1));

figure(1);
hold on;
mesh(x, t, u');
xlabel('x');
ylabel('t');
zlabel('u');
title("Solusi Metode Courant-Isaacson-Rees");

figure(2);
hold on;
mesh(x, t, y');
xlabel('x');
ylabel('t');
zlabel('u');
title("Solusi Eksak")

figure(3);
hold on;
fplot(u1, [0, 1], 'k');
plot(x, u(:, length(t)), 'r');
legend("Eksak", "Metode Courant");
title("Penampang pada t=1");
```

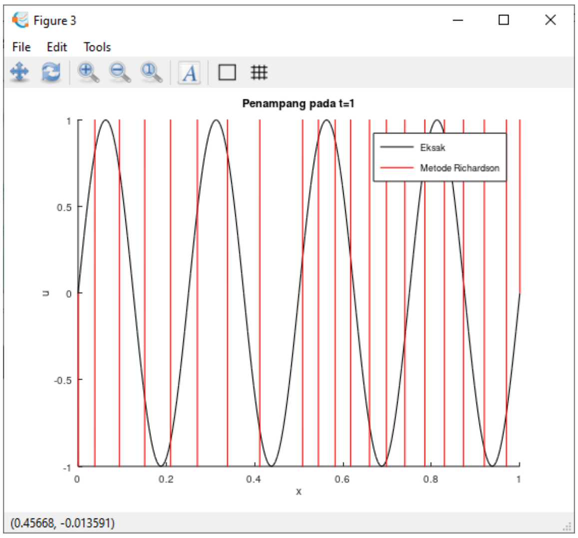




Jika diperhatikan, terlihat bahwa nilai u(x, 0) di *figure* 1 adalah 0, sehingga kalian mungkin akan mengira bahwa metodenya salah atau typo. Namun perhatikan sumbu u pada *figure* 1 dan

3. Nilainya mempunyai range yang berkisar $[-10000, 10000]$. Hal ini menunjukkan bahwa solusi metode Richardson akan tidak stabil. Pada faktanya, untuk persamaan transport, metode ini selalu tidak stabil. Kita dapat membatasi *range* dengan menambahkan `zlim([-1, 1]);` dan `ylim([-1, 1]);` berturut-turut pada `figure(1);` dan `figure(3);`.





Metode Lax

Metode ini adalah perbaikan dari metode Richardson dengan mengganti u_j^n dengan $\frac{1}{2}\left(u_{j+1}^n + u_{j-1}^n\right)$, sehingga persamaan bedanya menjadi

$$u_j^{n+1} = \frac{1}{2}\left(u_{j+1}^n + u_{j-1}^n\right) + C\left(u_{j+1}^n - u_{j-1}^n\right), \qquad C \equiv \frac{d\Delta t}{2\Delta x}$$

Metode ini membutuhkan syarat batas kiri dan kanan.

Berikut kode algoritma metode Lax menggunakan Octave.

```
function [x, t, u] = lax(d, f, lb, rb, xb, xu, tb, tu, dx,
dt)
  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));
  end

  for n = 1:nt
    u(1, n) = lb(t(n));
```

```
    u(nx, n) = rb(t(n));
  end

  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)));
    end
  end
end
```

Penjelasan input:

- d adalah nilai d pada $u_t + du_x = 0$.
- f adalah nilai awal $f(x)$ pada persamaan transport.
- lb dan rb berturut-turut adalah syarat batas kiri dan kanan pada persamaan transport.
- xb dan xu berturut-turut adalah batas bawah dan atas untuk variabel x.
- tb dan tu serupa, namun untuk variabel t.
- dx dan dt berturut-turut adalah *stepsize* Δx dan Δt.

Visualisasi untuk hasilnya:

```
clc;
clear all;
close all;
format long;

d = 1;
f = @(x) sin(8*pi*x);
lb = @(t) sin(-8*pi*t);
rb = @(t) sin(-8*pi*(1-t));
xb = 0;
xu = 1;
tb = 0;
tu = 1;
dx = 0.025;
dt = 0.02;

[x, t, u] = lax(d, f, lb, rb, xb, xu, tb, tu, dx, dt);

sol = @(x, t) sin(8*pi*(x-t));

for j = 1:length(x)
  for n = 1:length(t)
    y(j, n) = sol(x(j), t(n));
  endfor
endfor
```

```
u1 = @(x) sin(8*pi*(x-1));

figure(1);
hold on;
mesh(x, t, u');
xlabel('x');
ylabel('t');
zlabel('u');
title("Solusi Metode Lax");

figure(2);
hold on;
mesh(x, t, y');
xlabel('x');
ylabel('t');
zlabel('u');
title("Solusi Eksak")

figure(3);
hold on;
fplot(u1, [0, 1], 'k');
plot(x, u(:, length(t)), 'r');
xlabel('x');
ylabel('u');
legend("Eksak", "Metode Lax");
title("Penampang pada t=1");
```