Skema Beda Hingga untuk Persamaan Gelombang dan Difusi

Persamaan Gelombang

Persamaan gelombang memiliki bentuk:

$$u_{tt} - c^2 u_{xx} = 0, \qquad c = \frac{T}{\rho},$$
$$u(x,0) = \phi(x), \qquad u_t(x,0) = \psi(x).$$

Seperti sebelumnya, interval x dan t dipartisi dengan *stepsize* Δx dan Δt berturut-turut sehingga diperoleh $x_j = j\Delta x,\ j = 0, 1, 2, \ldots, N_x$, dan $t_n = n\Delta t,\ n = 0, 1, 2, \ldots, N_t$. Dari sini diperoleh suatu *grid* dengan titik-titik (x_j, t_n). Untuk nilai hampiran di titik-titik *grid* tersebut, akan digunakan notasi $u_j^n \equiv u(x_j, t_n)$.

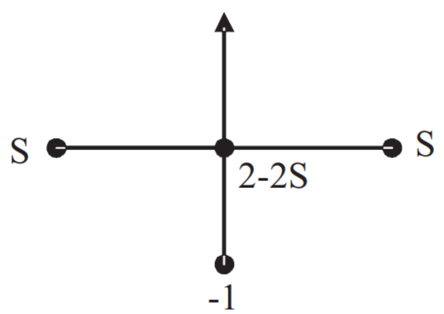
Metode pada persamaan gelombang memiliki persamaan beda

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

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

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

Metode ini membutuhkan syarat batas kiri dan kanan.



Pada *stencil* terlihat bahwa titik yang dicari membutuhkan dua baris t ke bawah, sehingga kita butuh dua baris nilai awal. Namun, hanya satu nilai awal yang tersedia, yaitu $u_j^0 = \phi\left(x_j\right),\ j = 0, 1, \ldots, N_x$. Untuk mendapatkan aproksimasi pada u_j^1, kita gunakan persamaan beda:

$$u_j^1 = \frac{S}{2}\left(u_{j+1}^0 + u_{j-1}^0\right) + (1 - S)u_j^0 + \Delta t\psi(x_j), \qquad j = 1, 2, \dots, N_x - 1$$

Berikut kode algoritma metode untuk persamaan gelombang menggunakan Octave.

```
function [x, t, u] = wave(c, phi, psi, lb, rb, xb, xu, tb,
tu, dx, dt)
  x = xb:dx:xu;
  t = tb:dt:tu;
  u = [];
  nt = length(t);
  nx = length(x);

  S = (c^2 * dt^2) / dx^2;

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

  for j = 2:nx-1
    u(j, 2) = (S/2) * (phi(x(j+1)) + phi(x(j-1))) + (1-S) *
phi(x(j)) + dt * psi(x(j));
  endfor

  for n = 2:nt
    u(1, n) = lb(t(n));
    u(nx, n) = rb(t(n));
  endfor

  for n = 2:nt-1
    for j = 2:nx-1
      u(j, n+1) = S * (u(j+1, n) + u(j-1, n)) + 2 * (1-S) *
u(j, n) - u(j, n-1);
    endfor
  endfor
endfunction
```

Penjelasan input:

- c adalah nilai c pada $u_{tt} - c^2 u_{xx} = 0$.
- phi dan psi adalah nilai awal $\phi(x)$ dan $\psi(x)$ pada persamaan gelombang.
- lb dan rb adalah syarat batas kiri pada persamaan gelombang.
- 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.

Akan kita uji solusi menggunakan persamaan gelombang:

$$u_{tt} - (0.25)^2 u_{xx} = 0, \qquad 0 < x < 1, t > 0,$$
$$u(0,t) = u(1,t) = 0, \qquad t > 0,$$
$$u(x,0) = 0, u_t(x,0) = x(1-x), \qquad 0 \le x \le 1$$

Solusi eksak dari PD tersebut adalah:

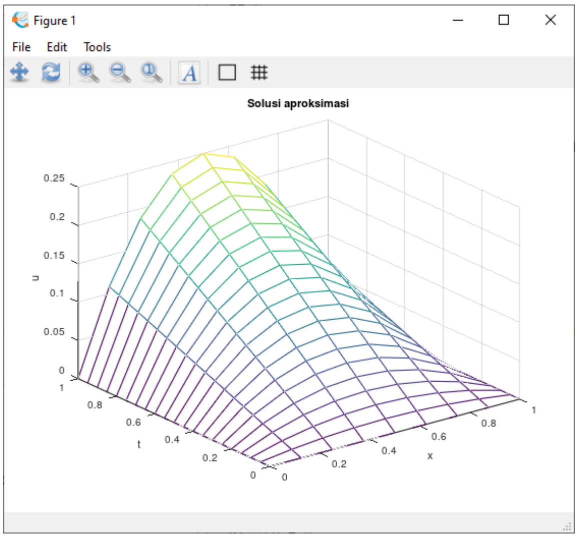
$$u(x,t) = \sum_{n=1}^{\infty} c_n \sin 0.25 n\pi t \sin n\pi x$$
$$c_n = \frac{2}{0.25 n\pi} \int_0^1 x(1-x) \sin n\pi x \; dx, \qquad n = 1, 2, \ldots$$

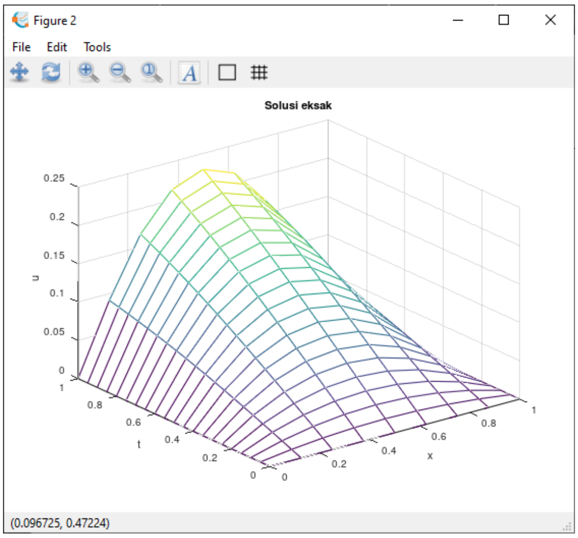
Untuk keperluan komputasi, akan kita ambil 10 suku pertama dari ekspansi deret Fourier dari $u(x,t)$.

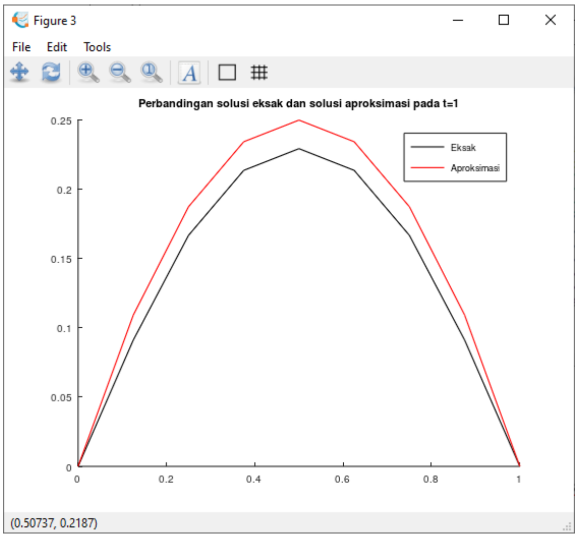
Untuk visualisasi dari solusi, sekarang kita akan mencoba membuat “animasi” jalannya solusi seiring bertambahnya t. Visualisasi ini akan disimpan pada figure(4);.

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

c = 0.25;
phi = @(x) 0;
psi = @(x) x * (1-x);
lb = rb = @(t) 0;
xb = 0;
xu = 1;
tb = 0;
tu = 1;
dx = 0.125;
dt = 0.05;

[x, t, u] = wave(c, phi, psi, lb, rb, xb, xu, tb, tu, dx,
dt);

for i = 1:length(x)
  for j = 1:length(t)
    w(i, j) = 0;
    for n = 1:10
      G = @(x) x .* (1-x) .* sin(n.*pi.*x);
      cn(n) = (2/(n*c*pi)) * integral(G, 0, 1);
      w(i, j) += cn(n) * sin(n*pi*c*t(j)) * sin(n*pi*x(i));
    endfor
```

```
  endfor
endfor

figure(1);
mesh(x, t, u');
xlabel("x");
ylabel("t");
zlabel("u");
title("Solusi aproksimasi");

figure(2);
mesh(x, t, w');
xlabel("x");
ylabel("t");
zlabel("u");
title("Solusi eksak");

figure(3);
hold on;
plot(x, w(:, length(t)), 'k');
plot(x, u(:, length(t)), 'r');
legend('Eksak', 'Aproksimasi');
title("Perbandingan solusi eksak dan solusi aproksimasi pada
t=1");

figure(4);
for j = 1:length(t)
  plot(x, u(:, j), 'k', 'linewidth', 1.5);
  ylim([0, 0.25]);
  title("Animasi solusi aproksimasi u(x, t) seiring
berjalannya t");
  drawnow;
  pause(0.001);
endfor
```







Persamaan Difusi (Panas) | Metode Eksplisit

Persamaan difusi (panas) memiliki bentuk:

$$u_t - du_{xx} = 0, \qquad u(x,0) = f(x),$$

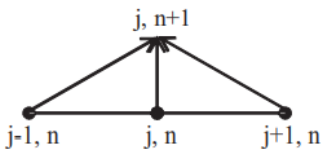
Metode pada persamaan panas memiliki persamaan beda

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

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

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

Metode ini membutuhkan syarat batas kiri dan kanan.



Berikut kode algoritma metode eksplisit untuk persamaan panas menggunakan Octave

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

```
  u = [];
  nt = length(t);
  nx = length(x);

  S = (d * dt) / dx^2;

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

  for n = 2:nt
    u(1, n) = lb(t(n));
    u(nx, n) = rb(t(n));
  endfor

  for n = 1:nt-1
    for j = 2:nx-1
      u(j, n+1) = (1 - 2*S) * u(j, n) + S * (u(j+1, n) + u(j-
1, n));
    endfor
  endfor
endfunction
```

Akan kita uji solusi menggunakan persamaan panas:

$$\begin{aligned} u_t - u_{xx} &= 0, \qquad 0 < x < 1, t > 0, \\ u(0,t) = u(1,t) &= 0, \qquad t \geq 0 \\ u(x,0) &= 10x^3(1-x), \qquad 0 \leq x \leq 1 \end{aligned}$$

Solusi eksak dari PD tersebut adalah:

$$u(x,t) = \sum_{n=1}^{\infty} c_n e^{-n^2\pi^2 t} \sin n\pi x$$

$$c_n = 20 \int_0^1 x^3(1-x) \sin n\pi x \ dx, \qquad n = 1, 2, \dots$$

Untuk keperluan komputasi, akan kita ambil 10 suku pertama dari ekspansi deret Fourier dari $u(x,t)$.

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

```
d = 1;
f = @(x) 10 * x^3 * (1-x);
lb = rb = @(t) 0;
xb = 0;
xu = 1;
tb = 0;
tu = 1;
dx = 0.2;
dt = 0.02;

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

for i = 1:length(x)
  for j = 1:length(t)
    w(i, j) = 0;
    for n = 1:10
      F = @(x) x.^3 .* (1-x) .* sin(n.*pi.*x);
      cn(n) = 20 * integral(F, 0, 1);
      w(i, j) += cn(n) * exp(-n^2*pi^2*t(j)) *
sin(n*pi*x(i));
    endfor
  endfor
endfor


figure(1);
mesh(x, t, u');
xlabel("x");
ylabel("t");
zlabel("u");

figure(2);
mesh(x, t, w');
xlabel("x");
ylabel("t");
zlabel("u");

figure(3);
for j = 1:length(t)
  plot(x, u(:, j), 'k', 'linewidth', 1.5);
  ylim([0, 1.5]);
  title("Animasi solusi aproksimasi u(x, t) seiring
berjalannya t");
  drawnow;
  pause(0.001);
endfor
```