

PENYELESAIAN PERSAMAAN DIFERENSIAL PARSIAL SECARA NUMERIK

Tugas Kuliah
SK5001 Analisis Numerik Lanjut

Mohammad Rizka Fadhli
NIM: 20921004

21 November 2021

PENDAHULUAN

Metode Beda Hingga

Pada tugas ini, metode yang digunakan untuk menyelesaikan persamaan diferensial parsial parabolik secara numerik adalah metode beda hingga. Akan ada tiga metode yang akan dibuatkan program-nya, yakni:

1. Metode eksplisit (**FTCS**: *forward difference in time, central difference in space*).
2. Metode implisit (**BTCS**: *backward difference in time, central difference in space*).
3. Metode **Crank-Nicolson**.

Bahasa yang Digunakan

Untuk membuat ketiga program di atas, saya menggunakan bahasa **R** yang bisa dieksekusi pada versi 4.1.1.

PDP Parabolik

Perhatikan PDP parabolik berikut:

$$\begin{aligned}u_t &= u_{xx} & 0 < x < l, t > 0 \\u(0, t) &= u(l, t) = 0 & t > 0 \text{ (syarat batas)} \\u(x, 0) &= f(x) & 0 \leq x \leq l \text{ syarat batas}\end{aligned}$$

Persamaan di atas disebut sebagai persamaan panas atau persamaan difusi.

METODE EKSPLISIT

Metode ini melakukan diskritisasi untuk memperoleh persamaan beda hingga dengan membagi selang $[0, l]$ ke dalam m sub-selang yang sama panjang dengan lebar $h = \frac{l}{m}$ dengan m bulat positif dan pilih diskritisasi pada sumbu t dengan lebar sub-selang yang sama panjang sebesar k , diperoleh titik bagi: (x_i, t_j) dengan $x_i = ih, i = 0, 1, \dots, m$ dan $t_j = jk, j = 0, 1, \dots$

Diperoleh:

$$u_t(x_i, t_j) = \frac{u(x_i, t_j + k) - u(x_i, t_j)}{k} + O(k)$$

$$u_{xx}(x_i, t_j) = \frac{u(x_i + h, t_j) - 2u(x_i, t_j) + u(x_i - h, t_j)}{h^2} + O(h^2)$$

dengan $w_{i,j}$ dinotasikan sebagai hampiran untuk $u(x_i, t_j)$ dan $\lambda = \frac{k}{h^2}$ kita dapat tuliskan:

Skema Iterasi

$$w_{i,j+1} = \lambda w_{i-1,j} + (1 - 2\lambda)w_{i,j} + \lambda w_{i+1,j}$$

Skema Iterasi dalam Aljabar Untuk $j = 0$, nilai-nilai $w_{i,0}$ sudah diberikan oleh syarat awal $w_{i,0} = u(x_i, 0) = f(x_i), 0 \leq i \leq m$. Sementara pada batas diberikan $w_{0,j} = w_{m,j} = 0$.

Kita tuliskan nilai-nilai w pada waktu j dalam vektor berikut:

$$w^{(j)} = \begin{pmatrix} w_{1,j} \\ w_{2,j} \\ \dots \\ w_{m-1,j} \end{pmatrix}$$

dan matriks tridiagonal berukuran $(m-1) \times (m-1)$ berikut:

$$A = \begin{pmatrix} 1-2\lambda & \lambda & 0 & \dots & 0 \\ \lambda & 1-2\lambda & \lambda & \dots & \dots \\ 0 & \lambda & 1-2\lambda & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \lambda & 1-2\lambda \end{pmatrix}$$

sehingga skema iterasi di atas, kita bisa tuliskan dalam bentuk matriks sebagai berikut:

$$w^{(j+1)} = Aw^{(j)}, j = 0, 1, 2, \dots$$

dengan orde galat $O(k + h^2)$.

Program Metode Eksplisit Saya akan menggunakan skema iterasi aljabar untuk membuat programnya. Berikut adalah *codes* dalam bahasa **R**:

```
k = Te/Nt
lambda = k/h^2

x = seq(0,1,by = h)

u_x0 = function(x){tuliskan fungsi soal}
w = u_x0(x)
w = w[-1]
w = w[-length(w)]

A = matrix(0,
           ncol = length(w),
           nrow = length(w))

for(i in 1:length(w)){
  A[i,i] = 1-(2*lambda)
}

for(i in 1:(length(w)-1)){
  A[i+1,i] = lambda
  A[i,i+1] = lambda
}

# iterasi sesuai dengan NT
for(i in 1:Nt){
  w = A %*% w
}
```

Contoh Soal

Gunakan $h = 0.1$ dan $k = 0.0005$ untuk menghampiri *heat equation* berikut ini:

$$u_t - u_{xx} = 0, \quad \begin{array}{l} 0 < x < 1 \\ 0 \leq t \end{array}$$

boundary condition berikut:

$$u(0, t) = u(1, t) = 0, 0 < t$$

initial condition berikut:

$$u(x, 0) = \sin(\pi x), 0 \leq x \leq 1$$

Bandingkan pada $t = 0.5$ menggunakan solusi eksak:

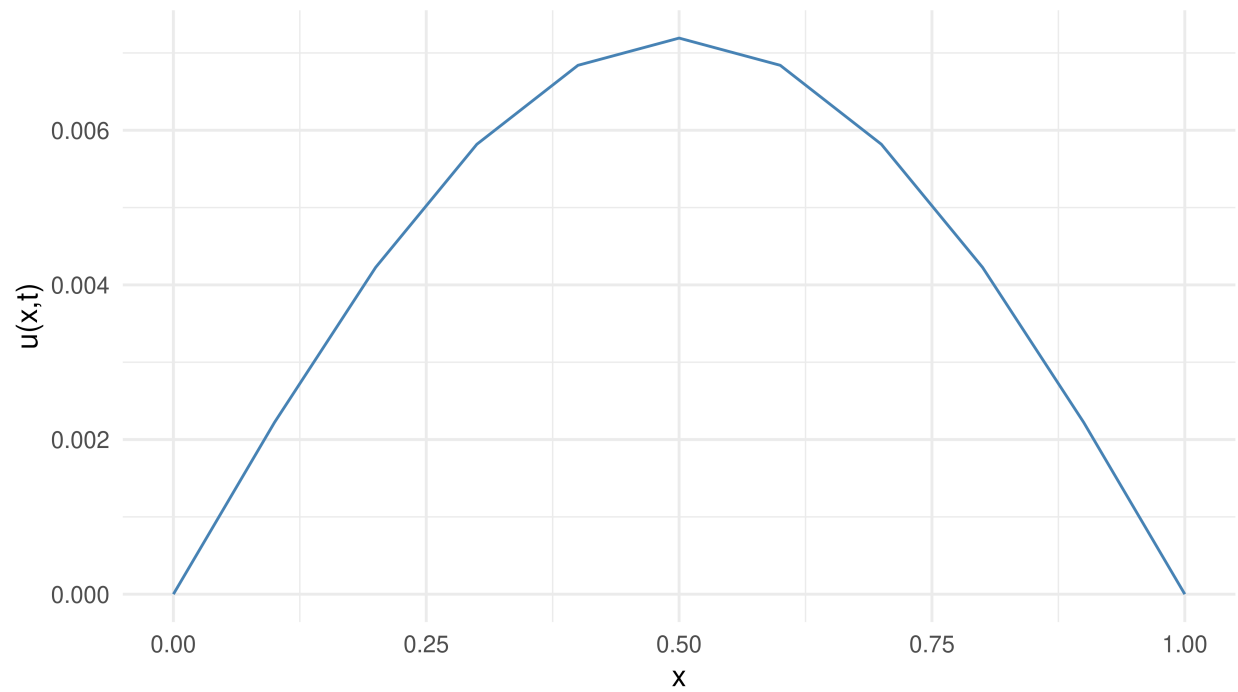
$$u(x, t) = e^{-\pi^2 t} \sin(\pi x)$$

Jawab Saya coba hitung terlebih dahulu solusi eksaknya dan membuat grafiknya pada selang $0 \leq x \leq 1$ berikut:

Table 1: Tabel Solusi Eksak

x	u(x,t)
0.0	0.0000000
0.1	0.0022224
0.2	0.0042273
0.3	0.0058184
0.4	0.0068399
0.5	0.0071919
0.6	0.0068399
0.7	0.0058184
0.8	0.0042273
0.9	0.0022224
1.0	0.0000000

Plot $u(x,t)$ pada $t = 0.5$
Solusi eksak



Digambar dengan R
20921004@mahasiswa.itb.ac.id

Figure 1: Grafik Solusi Eksak

Sekarang kita akan hitung solusinya menggunakan program yang kita telah buat sebelumnya:

```
L = 1
Nx = 10
h = L/Nx
Te = 0.5
Nt = 100
k = Te/Nt
lambda = k/h^2

x = seq(0,1,by = h)

u_x0 = function(x){sin(pi*x)}
w = u_x0(x)
w = w[-1]
w = w[-length(w)]

A = matrix(0,
           ncol = length(w),
           nrow = length(w))

for(i in 1:length(w)){
  A[i,i] = 1-(2*lambda)
}

for(i in 1:(length(w)-1)){
  A[i+1,i] = lambda
  A[i,i+1] = lambda
}
```



```

# iterasi sesuai dengan NT
for(i in 1:Nt){
  w = A %*% w
}

solusi_numerik = data.frame(x,y = c(0,w,0))

```

Table 2: Tabel Solusi Numerik

x	u(x,t) numerik	u(x,t) eksak	numerik - eksak
0.0	0.0000000	0.0000000	0.0000000
0.1	0.0020446	0.0022224	0.0001778
0.2	0.0038891	0.0042273	0.0003382
0.3	0.0053529	0.0058184	0.0004654
0.4	0.0062927	0.0068399	0.0005472
0.5	0.0066166	0.0071919	0.0005753
0.6	0.0062927	0.0068399	0.0005472
0.7	0.0053529	0.0058184	0.0004654
0.8	0.0038891	0.0042273	0.0003382
0.9	0.0020446	0.0022224	0.0001778
1.0	0.0000000	0.0000000	0.0000000

Kita dapatkan nilai solusi hampiran yang relatif sangat dekat dengan nilai solusi eksaknya.

Berikut adalah grafik dari solusi numeriknya:

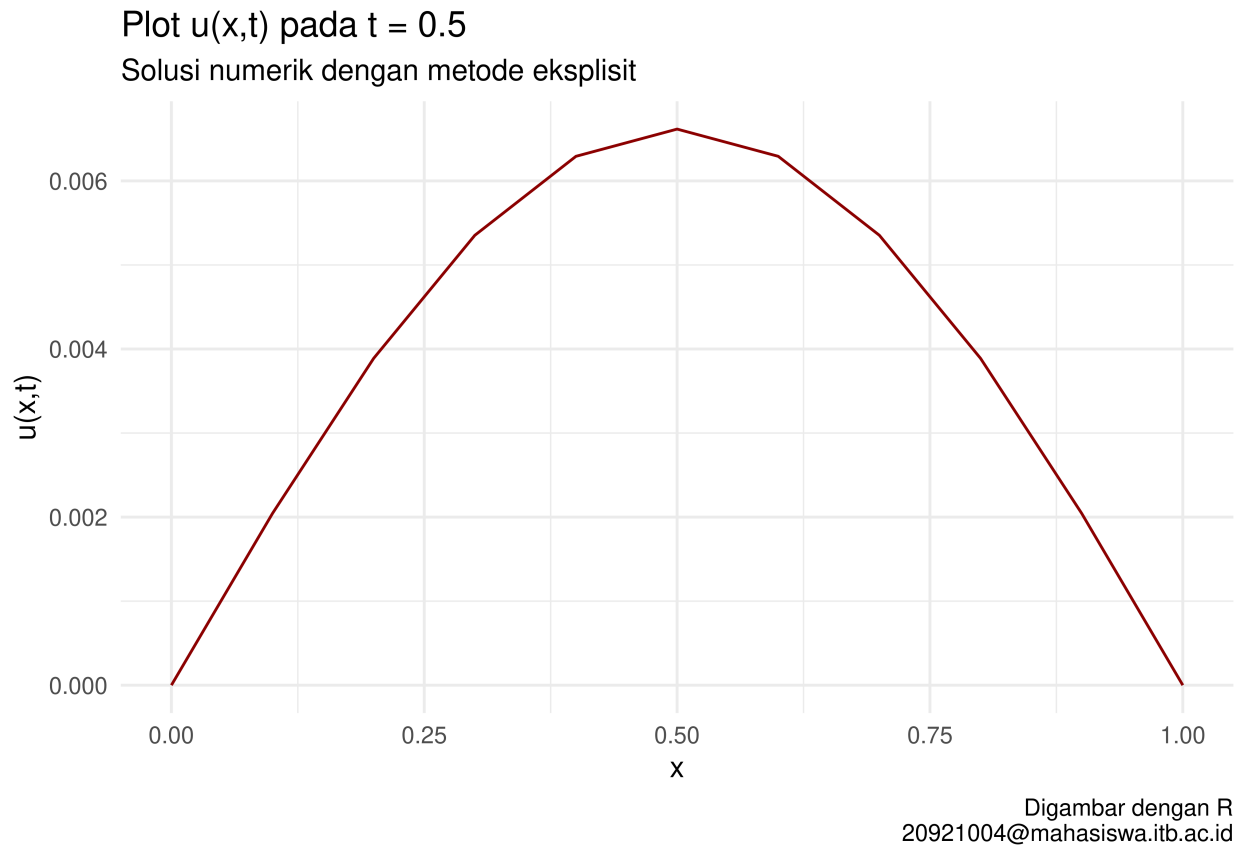


Figure 2: Solusi Numerik dengan Metode Eksplisit

METODE IMPLISIT

Metode implisit menggunakan skema *backward difference* untuk $u_t(x_i, t_j)$ dan *central difference* untuk $u_{xx}(x_i, t_j)$.

$$u_t(x_i, t_j) = \frac{u(x_i, t_j) - u(x_i, t_j - k)}{k} + O(k)$$

$$u_{xx}(x_i, t_j) = \frac{u(x_i + h, t_j) - 2u(x_i, t_j) + u(x_i - h, t_j)}{h^2} + O(h^2)$$

Skema Iterasi dari bentuk di atas adalah:

$$w_{i,j-1} = -\lambda w_{i+1,j} + (1 + 2\lambda)w_{i,j} - \lambda w_{i-1,j}$$

dengan $w_{i,j}$ menghampiri $u(x_i, t_j)$ dan $\lambda = \frac{k}{h^2}$.

Perhatikan dari skema iterasi di atas, nilai-nilai w pada saat ke j dikaitkan pada saat waktu ke $j - 1$. Sehingga vektor $w^{(j)}$ secara implisit didapatkan dari sistem persamaan linear:

$$Aw^{(j)} = w^{(j-1)}, j = 1, 2, \dots, j_{max}$$

dengan nilai-nilai batas $w_{0,j} = w_{m,j} = 0$. Jika matriks A **tidak singular**, maka kita akan dapatkan skema iterasi dalam bentuk aljabar berikut:

Skema Iterasi dalam Aljabar

$$w^{(j)} = A^{-1}w^{(j-1)} = \dots = (A^{-1})^{(j)}w^{(0)}, j = 1, 2, \dots, j_{max}$$

dengan orde galat $O(k + h^2)$. Persamaan di atas harus selalu diselesaikan setiap waktu j .

dengan:

$$w^{(j)} = \begin{pmatrix} w_{1,j} \\ w_{2,j} \\ \dots \\ w_{m-1,j} \end{pmatrix}$$

dan matriks tridiagonal berukuran $(m - 1) \times (m - 1)$ berikut:

$$A = \begin{pmatrix} 1+2\lambda & -\lambda & 0 & .. & 0 \\ -\lambda & 1+2\lambda & -\lambda & .. & .. \\ 0 & -\lambda & 1+2\lambda & .. & .. \\ .. & .. & .. & .. & .. \\ 0 & .. & 0 & -\lambda & 1+2\lambda \end{pmatrix}$$

Program Metode Implisit Saya akan menggunakan skema iterasi aljabar untuk membuat programnya. Berikut adalah *codes* dalam bahasa **R**:

```
k = Te/Nt
lambda = k/h^2

x = seq(0,1,by = h)

u_x0 = function(x){tuliskan fungsi soal}
w = u_x0(x)
w = w[-1]
w = w[-length(w)]

A = matrix(0,
           ncol = length(w),
           nrow = length(w))

for(i in 1:length(w)){
  A[i,i] = 1+(2*lambda)
}

for(i in 1:(length(w)-1)){
  A[i+1,i] = -lambda
  A[i,i+1] = -lambda
}
```

```

}

A_inv = solve(A)
# iterasi sesuai dengan NT
for(i in 1:Nt){
    w = A_inv %*% w
}

```

Contoh Soal

Gunakan $h = 0.1$ dan $k = 0.01$ untuk menghampiri *heat equation* berikut ini:

$$u_t - u_{xx} = 0, \begin{matrix} 0 < x < 1 \\ 0 \leq t \end{matrix}$$

boundary condition berikut:

$$u(0, t) = u(1, t) = 0, 0 < t$$

initial condition berikut:

$$u(x, 0) = \sin(\pi x), 0 \leq x \leq 1$$

Bandingkan pada $t = 0.5$ menggunakan solusi eksak:

$$u(x, t) = e^{-\pi^2 t} \sin(\pi x)$$

Jawab Soal ini adalah soal yang sama dengan soal pada bagian metode eksplisit. Di sini saya akan memperlihatkan bahwa dengan nilai h dan k yang relatif besar, metode implisit akan menghasilkan nilai yang tetap baik dan stabil.

Saya coba hitung terlebih dahulu solusi eksaknya pada selang $0 \leq x \leq 1$ berikut:

Table 3: Tabel Solusi Eksak

x	u(x,t)
0.0	0.00000000
0.1	0.0022224
0.2	0.0042273
0.3	0.0058184
0.4	0.0068399
0.5	0.0071919
0.6	0.0068399
0.7	0.0058184
0.8	0.0042273
0.9	0.0022224
1.0	0.00000000

Berikut adalah penyelesaiannya dengan program yang telah dibuat sebelumnya:

```
L = 1
Nx = 10
h = L/Nx
Te = 0.5
Nt = 50
k = Te/Nt
lambda = k/h^2

x = seq(0,1,by = h)

u_x0 = function(x){sin(pi*x)}
w = u_x0(x)
```

```

w = w[-1]
w = w[-length(w)]

A = matrix(0,
           ncol = length(w),
           nrow = length(w))

for(i in 1:length(w)){
  A[i,i] = 1+(2*lambda)
}

for(i in 1:(length(w)-1)){
  A[i+1,i] = -lambda
  A[i,i+1] = -lambda
}

A_inv = solve(A)
# iterasi sesuai dengan NT
for(i in 1:Nt){
  w = A_inv %*% w
}

solusi_numerik = data.frame(x,y = c(0,w,0))

```

Berikut adalah hasil dan perbandingannya dengan solusi eksak:

Table 4: Hasil Solusi Numerik dari Metode Implisit

x	u(x,t) numerik	u(x,t) eksak	numerik - eksak
0.0	0.0000000	0.0000000	0.0000000
0.1	0.0028980	0.0022224	0.0006756
0.2	0.0055124	0.0042273	0.0012851
0.3	0.0075871	0.0058184	0.0017688
0.4	0.0089192	0.0068399	0.0020793
0.5	0.0093782	0.0071919	0.0021863
0.6	0.0089192	0.0068399	0.0020793
0.7	0.0075871	0.0058184	0.0017688
0.8	0.0055124	0.0042273	0.0012851
0.9	0.0028980	0.0022224	0.0006756
1.0	0.0000000	0.0000000	0.0000000

Kita dapatkan solusi numerik yang relatif dekat dengan solusi eksak.

METODE CRANK-NICOLSON

Metode ini didapatkan dengan merata-ratakan metode eksplisit dan implisit.

Skema Iterasi Dengan $\lambda = \frac{k}{h^2}$, maka:

$$-\frac{\lambda}{2}w_{i-1,j+1} + (1 + \lambda)w_{i,j+1} - \frac{\lambda}{2}w_{i+1,j+1} = \frac{\lambda}{2}w_{i-1,j} + (1 - \lambda)w_{i,j} + \frac{\lambda}{2}w_{i+1,j+1}$$

dengan syarat batas $w_{0,j} = w_{m,j} = 0$.

Skema Iterasi dalam Aljabar Skema di atas bisa kita tuliskan menjadi:

$$Aw^{(j+1)} = Bw^{(j)}$$

$$w^{(j+1)} = A^{-1}Bw^{(j)}$$

dengan:

$$w^{(j)} = \begin{pmatrix} w_{1,j} \\ w_{2,j} \\ \dots \\ w_{m-1,j} \end{pmatrix}$$

dan matriks tridiagonal berukuran $(m-1) \times (m-1)$ berikut:

$$A = \begin{pmatrix} 1 + \lambda & -\lambda/2 & 0 & \dots & 0 \\ -\lambda/2 & 1 + \lambda & -\lambda/2 & \dots & \dots \\ 0 & -\lambda/2 & 1 + \lambda & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & -\lambda/2 & 1 + \lambda \end{pmatrix}$$

dan

$$B = \begin{pmatrix} 1 - \lambda & \lambda/2 & 0 & \dots & 0 \\ \lambda/2 & 1 - \lambda & \lambda/2 & \dots & \dots \\ 0 & \lambda/2 & 1 - \lambda & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \lambda/2 & 1 - \lambda \end{pmatrix}$$

Program Metode Crank-Nicolson Berikut adalah program yang saya tulis dalam bahasa R:

```
k = Te/Nt
lambda = k/h^2

x = seq(0,1,by = h)

u_x0 = function(x){tuliskan fungsi soal}
w = u_x0(x)
w = w[-1]
w = w[-length(w)]

# matrix A
A = matrix(0,ncol = length(w),nrow = length(w))
for(i in 1:length(w)){
  A[i,i] = 1+lambda
}
for(i in 1:(length(w)-1)){
  A[i+1,i] = -lambda/2
  A[i,i+1] = -lambda/2
}

# matrix B
B = matrix(0,ncol = length(w),nrow = length(w))
for(i in 1:length(w)){
  B[i,i] = 1-lambda
}
for(i in 1:(length(w)-1)){
  B[i+1,i] = lambda/2
```

```

    B[i,i+1] = lambda/2
}

A_inv = solve(A)
# iterasi sesuai dengan NT
for(i in 1:Nt){
    w = A_inv %*% B %*% w
}

```

Contoh Soal

Gunakan metode Crank-Nicolson dengan $h = 0.1$ dan $k = 0.01$ untuk mengaproksimasi solusi dari masalah:

$$u_t - u_{xx} = 0, \begin{matrix} 0 < x < 1 \\ 0 \leq t \end{matrix}$$

boundary condition berikut:

$$u(0, t) = u(1, t) = 0, 0 < t$$

initial condition berikut:

$$u(x, 0) = \sin(\pi x), 0 \leq x \leq 1$$

Bandingkan pada $t = 0.5$ menggunakan solusi eksak:

$$u(x, t) = e^{-\pi^2 t} \sin(\pi x)$$

Jawab Saya coba hitung terlebih dahulu solusi eksaknya pada selang $0 \leq x \leq 1$ berikut:

Table 5: Tabel Solusi Eksak

x	u(x,t)
0.0	0.00000000
0.1	0.0022224
0.2	0.0042273
0.3	0.0058184
0.4	0.0068399
0.5	0.0071919
0.6	0.0068399
0.7	0.0058184
0.8	0.0042273
0.9	0.0022224
1.0	0.00000000

Berikut adalah penyelesaiannya dengan program yang telah dibuat sebelumnya:

```

L = 1
Nx = 10
h = L/Nx
Te = 0.5
Nt = 50
k = Te/Nt
lambda = k/h^2

x = seq(0,1,by = h)

u_x0 = function(x){sin(pi*x)}
w = u_x0(x)
w = w[-1]

```

```

w = w[-length(w)]

# matrix A
A = matrix(0,ncol = length(w),nrow = length(w))
for(i in 1:length(w)){
  A[i,i] = 1+lambda
}
for(i in 1:(length(w)-1)){
  A[i+1,i] = -lambda/2
  A[i,i+1] = -lambda/2
}

# matrix B
B = matrix(0,ncol = length(w),nrow = length(w))
for(i in 1:length(w)){
  B[i,i] = 1-lambda
}
for(i in 1:(length(w)-1)){
  B[i+1,i] = lambda/2
  B[i,i+1] = lambda/2
}

A_inv = solve(A)
# iterasi sesuai dengan NT
for(i in 1:Nt){
  w = A_inv %*% B %*% w
}

solusi_numerik = data.frame(x,y = c(0,w,0))

```

Berikut adalah hasil dan perbandingannya dengan solusi eksak:

Table 6: Hasil Solusi Numerik dari Metode Crank-Nicolson

x	u(x,t) numerik	u(x,t) eksak	numerik - eksak
0.0	0.0000000	0.0000000	0.0000000
0.1	0.0023051	0.0022224	0.0000827
0.2	0.0043846	0.0042273	0.0001573
0.3	0.0060349	0.0058184	0.0002165
0.4	0.0070944	0.0068399	0.0002546
0.5	0.0074595	0.0071919	0.0002677
0.6	0.0070944	0.0068399	0.0002546
0.7	0.0060349	0.0058184	0.0002165
0.8	0.0043846	0.0042273	0.0001573
0.9	0.0023051	0.0022224	0.0000827
1.0	0.0000000	0.0000000	0.0000000