PENYELESAIAN NUMERIK PERSAMAAN KONDUKSI 1D DENGAN SKEMA FTCS, LAASONEN DAN CRANK-NICOLSON

Eko Prasetya Budiana[1]
Syamsul Hadi[2]

Abstract, Finite difference method (FTCS, Laasonen and Crank-Nicholson scheme) have been develop for solving 1D conduction enguation. Forward difference is used for temporal discretization and central difference is used for spatial discretization. Numerical results are comparison to the exac solution, Crank-Nicolson scheme has minimum error in comparison with FTCS and Laasonen scheme.

Keywords : Finite difference, forward difference, central difference,conduction

PENDAHULUAN

Metode beda hingga adalah salah satu pendekatan numerik yang dapat digunakan untuk menyelesaikan persamaan konduksi 1D. Untuk penyelesain kasus 1D terdapat beberapa skema pendekatan anatara lain skema FTCS, Laasonen dan Crank-Nicolson.
Ketiga skema tersebut didasarkan pada pendekatan beda maju untuk turunan waktu dan pendekatan beda tengah untuk turuan ruang.

Skema FTCS dapat diselesaikan secara ekplisit sedangkan skema Laasonen dan Crank-Nicolson penyelesaiannya secara implisit. Bentuk matrik koefisien penyelesaian implisit adalah matrik pita tridiagonal yang dapat diselesaikan dengan algoritma Thomas.

Dalam paper ini akan disajikan penyelesaian persamaan konduksi 1D dengan pendekatan skema FTCS, Laasonen dan Crank-Nicholson. Hasil penyelesaian dengan tiga skema tersebut dibandingkan dengan hasil penyelesaian eksak kemudian dicari penyelesaian yang memiliki ketelitian terbaik.

LANDASAN TEORI

Model matematika untuk persamaan konduksi 1D adalah :

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad \text{.....} \quad (1)$$

Model matematika tersebut diselesaiakn secara numerik dengan skema FTCS, Laasonen dan Crank-Nicolson.

Skema FTCS

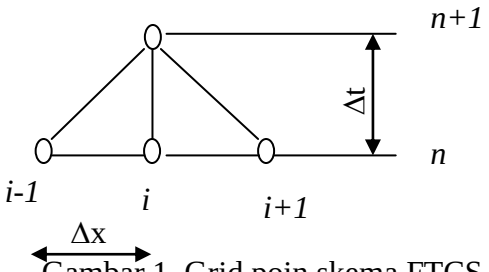
Diskretisasi persaman konduksi 1D dengan skema FTCS adalah :

$$\frac{T^{n+1} - T^n}{\Delta t} = \alpha \frac{T_{i-1}^n - 2T_i^n + T_{i+1}^n}{\Delta x^2} \quad \text{.....} \quad (2)$$

[1] Staf Pengajar Jurusan Teknik Mesin FT UNS
[2] Staf Pengajar Jurusan Teknik Mesin FT UNS

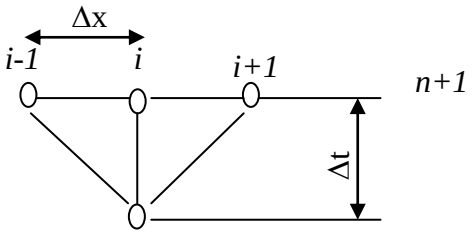
Dari persamaan 2 diatas hanya variabel T_i^{n+1} yang tidak diketahui maka persaman tersebut dapat disusun menjadi :

$$T_i^{n+1} = T_i^n + \frac{\alpha \Delta t}{\Delta x^2}\left(T_{i-1}^n - 2T_i^n + T_{i+1}^n\right) \quad \text{.....} \quad (3)$$



Gambar 1. Grid poin skema FTCS

Skema Laasonen



Gambar 2. Grid poin skema Laasonen

Diskretisasi persaman konduksi 1D dengan skema Laasonen adalah :

$$\frac{T^{n+1} - T^n}{\Delta t} = \alpha \frac{T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1}}{\Delta x^2} \quad \text{.....} \quad (4)$$

Persamaan 4 dapat disusun menjadi :

$$-\frac{\alpha \Delta t}{\Delta x^2} T_{i-1}^{n+1} + \left(1 + 2\frac{\alpha \Delta t}{\Delta x^2}\right) T_i^{n+1} - \frac{\alpha \Delta t}{\Delta x^2} T_{i+1}^{n+1} = T_i^n \quad \text{.....} \quad (5)$$

$$a_i T_{i-1}^{n+1} + b_i T_i^{n+1} + c_i T_{i+1}^{n+1} = d_i \quad \text{.....} \quad (6)$$

dimana :

$$a_i = -\frac{\alpha \Delta t}{\Delta x^2}$$

$$b_i = 1 + 2\frac{\alpha \Delta t}{\Delta x^2}$$

$$c_i = -\frac{\alpha \Delta t}{\Delta x^2}$$

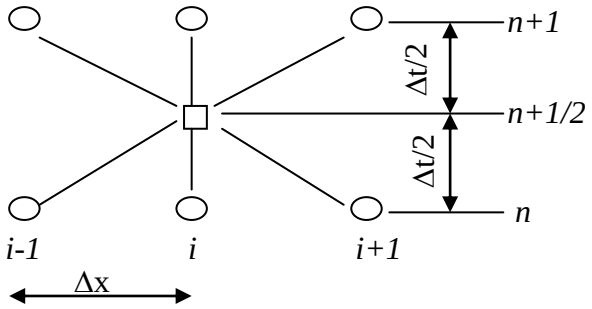
$$d_i = T_i^n$$

Persamaan 6 dapat disusun menjadi bentuk formulasi matrik sebagai berikut :

$$\begin{pmatrix} b_2 & c_2 & & & & & \\ a_3 & b_3 & c_3 & & & & \\ & a_4 & b_4 & c_4 & & & \\ & & & & am\text{-}2 & bm\text{-}2 & cm\text{-}2 \\ & & & & & am\text{-}1 & bm\text{-}1 \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \\ u_4 \\ \\ u_{n-2} \\ u_{n-1} \end{pmatrix} = \begin{pmatrix} {}_{d2}\text{-}a_2u_2 \\ d_3 \\ d_4 \\ \\ d_{m-2} \\ d_{m-1}\text{-}c_{m-1}u_m \end{pmatrix}$$

Kemudian persamaan matrik diselesaikan dengan algoritma Thomas.

Skema Crank-Nicolson



Gambar 3. Grid poin skema Crank-Nicolsonn

Diskretisasi skema Crank-Nicholson adalah sebagai berikut :

$$\frac{T^{n+1}-T^{n}}{\Delta t}=\left(\frac{\alpha}{2}\right)\left(\frac{T_{i-1}^{n+1}-2T_{i}^{n+1}+T_{i+1}^{n+1}}{\Delta x^{2}}+\frac{T_{i-1}^{n}-2T_{i}^{n}+T_{i+1}^{n}}{\Delta x^{2}}\right) \quad \text{.....} \quad (8)$$

Persamaan 8 dapat disusun menjadi :

$$a_iT_{i-1}^{n+1}+b_iT_i^{n+1}+c_iT_{i+1}^{n+1}=d_i \quad \text{.....} \quad (9)$$

dimana :

$$a_i=-\frac{\alpha\Delta t}{2\Delta x^2}$$

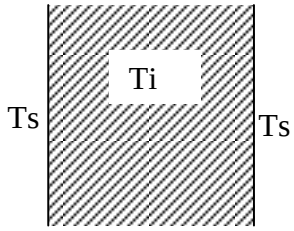
$$b_i=1+\frac{\alpha\Delta t}{\Delta x^2}$$

$$c_i=-\frac{\alpha\Delta t}{2\Delta x^2}$$

$$d_i=T_i^n+\frac{\alpha\Delta t}{2\Delta x^2}\left(T_{i-1}^n-2T_i^n+T_{i+1}^n\right)$$

Persamaan 9 disusun menjadi bentuk formulasi matriks seperti pada skema Laasonen kemudian diselesaikan dengan algoritma Thomas.

Kasus perpindahan panas konduksi 1D



Gambar 4. Domain dan syarat batas.

Syarat awal (t=0) : Ti=100. ^oF
Syarat batas : Ts=300. ^oF
Difusivitas termal : α = 0.1 ft^2/jam
Dimensi domain : L=1. ft

Algoritma Pemrograman
1. Tentukan jumlah grid, langkah waktu dan properti bahan.
2. Tentukan syarat awal dan syarat batas
3. Hitung Tn+1 dengan metode FTCS, Laasonen dan Crank-Nicolson
4. Apakah sudah sampai batas perhitungan (t)?
   Jika belum, kembali ke-2
5. Tulis hasil
6. Selesai

HASIL DAN PEMBAHASAN

Program ditulis dalam Bahasa Fortran, sistem grid yang digunakan adalah Δx=0.05 untuk langkah ruang dan Δt=0.01 untuk langkah waktu. Skema FTCS memerlukan syarat kestabilan $\frac{\alpha\Delta t}{\Delta x^2} \le 0.5$. Untuk perhitungan ini $\frac{\alpha\Delta t}{\Delta x^2} = \frac{0.1x0.01}{(0.05)^2} = 0.4$ sehingga syarat kestabilan telah dipenuhi. Untuk skema Laasonen dan Crank-Nicolson tidak memerlukan syarat kestabilan.

Selanjutnya untuk mengetahui ketelitian dari penyelesaian numerik maka dibuat perbandingan antara penyelesaian numerik dengan penyelesaian analitis. Penyelesaian analitis dari kasus ini adalah :

$$T = T_s + 2(T_i - T_s)\sum_{m=1}^{\infty} e^{-\left(\frac{m\pi}{L}\right)\alpha t} \frac{1-(-1)^m}{m\pi} \sin\left(\frac{m\pi x}{L}\right) \quad \text{.....} \quad (10)$$

Hasil perbandingan ditunjukkan dalam tabel 1. Kesalahan dihitung dengan persamaan :

$$ER = \left|\frac{hasilanalitis - hasi\ln umerik}{hasi\ln analitis}\right| x100\%$$

Tabel 1. Kesalahan (ER) untuk skema FTCS, Laasonen dan Crank-Nicolson untuk t=0.5

x	FTCS	Laasonen	Crank-Nicolson
0.00	0.000	0.000	0.000
0.05	0.112	0.165	0.222
0.10	0.213	0.321	0.436
0.15	0.268	0.425	0.591
0.20	0.244	0.438	0.641
0.25	0.123	0.335	0.553
0.30	0.097	0.112	0.324
0.35	0.383	0.193	0.007
0.40	0.674	0.512	0.360
0.45	0.894	0.756	0.632
0.50	0.976	0.847	0.734
0.55	0.894	0.756	0.632
0.60	0.674	0.512	0.360
0.65	0.383	0.193	0.007
0.70	0.097	0.112	0.324
0.75	0.123	0.335	0.553
0.80	0.244	0.438	0.641
0.85	0.268	0.425	0.591
0.90	0.213	0.321	0.436
0.95	0.112	0.165	0.222
1.00	0.000	0.000	0.000

Dari tabel 1 diketahui kesalahan maksimum skema FTCS. Laasonen dan Crank-Nicolson masing-masing adalah 1.411%. 1.224% dan 1.062%. Dari hasil perbandingan diketahui bahwa skema Crank-Nicolson memiliki ketelitian terbaik.

KESIMPULAN

Dalam paper ini perhitungan numerik persamaan konduksi 1D telah dilakukan. Hasil perbandingan menunjukkan bahwa skema Crank-Nicolson memiliki ketelitian terbaik. Skema FTCS memerlukan syarat kstabilan $\frac{\alpha \Delta t}{\Delta x^2} \leq 0.5$ sedangkan skema Laasonen dan Crank-Nicolson stabil tanpa syarat.

DAFTAR PUSTAKA

Duchateu. P. & Zachmann. D.W.. 1986. *Theory and Problems of Partial Differential Equations*.
McGraw-Hill. Inc

Hoffmann. K.A.. 1989. *Computational Fluid Dynamics for Engineer*. University of Texas. Austin

Holman. J.P.. 1994. *Perpindahan Kalor*. Penerbit Erlangga Jakarta

Daftar Program

Program FTCS

```
      dimension x(51),u(51),un(51)
      n=21
      v=0.1
      h=1.
      dx=h/(n-1)
      dt=0.01
      t=0.
      do j=1,n
      u(j)=100.
      enddo
      write(*,10)
      read(*,*)tt
10    format(' Batas waktu t= ',\)
      it=int(tt/dt)
123   k=k+1
      t=k*dt
      u(1)=300.
      u(n)=300.
      do j=2,n-1
      un(j)=u(j)+v*dt/dx/dx*(u(j-1)-2*u(j)+u(j+1))
      enddo
      do j=2,n-1
      u(j)=un(j)
      enddo
      if(k.lt.it)goto 123
      write(*,*)' it= ',it,' k= ',k ,' t= ',t
      open(1,file='d:\aa-eko\data\oftc.dat')
      do j=1,n
      x(j)=(j-1)*dx
      write(1,20)x(j),u(j)
      enddo
20    format(2x,f10.3,2x,f10.3)
      end
```

Program Laasonen

```
      dimension y(41),u(41)
      dimension a(41),b(41),c(41),d(41)
   n=21
      v=0.1
      h=1.0
      dy=h/(n-1)
      dt=0.01
      k=0
      do j=1,n
      u(j)=100.
      enddo
      u(1)=300.
      u(n)=300.
      write(*,10)
      read(*,*)tt
```

```
10      format(' Batas waktu t= ',\)
        kt=int(tt/dt)
123     k=k+1
        do j=2,n-1
        a(j)=-v*dt/dy/dy
        b(j)=1+2*v*dt/dy/dy
        c(j)=-v*dt/dy/dy
        d(j)=u(j)
        enddo
        d(2)=d(2)-a(2)*u(1)
        a(2)=0.
        d(n-1)=d(n-1)-c(n-1)*u(n)
        c(n-1)=0.
        call tridi(a,b,c,d,2,n-1)
        do j=2,n-1
        u(j)=d(j)
        enddo
        if(k.lt.kt)goto 123
        write(*,*)' kt= ',kt
        open(1,file='d:\aa-eko\data\olasonen.dat')
        do j=1,n
        y(j)=(j-1)*dy
        write(1,20)y(j),u(j)
        enddo
20      format(2x,f10.3,2x,f10.3)
        end

        subroutine tridi(a,b,c,d,l1,l2)
        dimension a(41),b(41),c(41),d(41)
        do 1 i=l1+1,l2
        r=-a(i)/b(i-1)
        b(i)=b(i)+r*c(i-1)
1       d(i)=d(i)+r*d(i-1)
        d(l2)=d(l2)/b(l2)
        do 2 j=l2-1,l1,-1
2       d(j)=(d(j)-c(j)*d(j+1))/b(j)
        return
        end
```

Program Crank-Nicolson

```
        dimension y(41),u(41)
        dimension a(41),b(41),c(41),d(41)
    n=21
        v=0.1
        h=1.0
        dy=h/(n-1)
        dt=0.01
        k=0
        do j=1,n
        u(j)=100.
        enddo
        u(1)=300.
        u(n)=300.
        write(*,10)
        read(*,*)tt
```

```
10      format(' Batas waktu t= ',\)
        kt=int(tt/dt)
123     k=k+1
        do j=2,n-1
        a(j)=-v*dt/dy/dy/2
        b(j)=1+v*dt/dy/dy
        c(j)=-v*dt/dy/dy/2
        d(j)=u(j)+v*dt/dy/dy/2*(u(j-1)-2*u(j)+u(j+1))
        enddo
        d(2)=d(2)-a(2)*u(1)
        a(2)=0.
        d(n-1)=d(n-1)-c(n-1)*u(n)
        c(n-1)=0.
        call tridi(a,b,c,d,2,n-1)
        do j=2,n-1
        u(j)=d(j)
        enddo
        if(k.lt.kt)goto 123
        write(*,*)' kt= ',kt
        open(1,file='d:\aa-eko\data\ocnk.dat')
        do j=1,n
        y(j)=(j-1)*dy
        write(1,20)y(j),u(j)
        enddo
20      format(2x,f10.3,2x,f10.3)
        end

        subroutine tridi(a,b,c,d,l1,l2)
        dimension a(41),b(41),c(41),d(41)
        do 1 i=l1+1,l2
        r=-a(i)/b(i-1)
        b(i)=b(i)+r*c(i-1)
1       d(i)=d(i)+r*d(i-1)
        d(l2)=d(l2)/b(l2)
        do 2 j=l2-1,l1,-1
2       d(j)=(d(j)-c(j)*d(j+1))/b(j)
        return
        end
```