ME 747 Introduction to computational fluid dynamics

Lecture 5_2

Introduction to solve engineering problems with finite-difference method

By Chainarong Chaktranond

Lecture schedule

Session	Topics
1	**1.Overviews of computational fluid dynamics** - Overviews and importance of heat transfer in real applications
2 - 3	**2. Introduction to Fortran programming** - Basic commands in Fortran programming
4	**3. Overviews of governing equations for flow and heat transfer** -Elliptic, Parabolic and Hyperbolic equations
5	**4. Introduction to numerical methods** - Finite different method, Finite volume method, Finite element method, etc.
6 – 7	**5. Introduction to solve engineering problems with finite-difference method** - Taylor series expansion, Approximation of the second derivative, Initial condition and Boundary conditions

Contents

- Iteration method
- Gauss-Seidel Method
- Successive Over Relaxation Method

Iterative method: Gauss-Seidel Method

$$[A]\{X\} = \{C\}$$

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1m}x_m = c_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2m}x_m = c_2$$
$$\vdots \qquad\qquad\qquad\qquad \vdots$$
$$a_m x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{mm}x_m = c_m$$

Controlled error

$$\left|\varepsilon_{a,i}\right| = \left|\frac{x_i^n - x_i^{n-1}}{x_i^n}\right| \times 100 < \varepsilon_s$$

Iterative method: Gauss-Seidel Method

Example

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

1st Step:

$$x_1 = \frac{7.85 + 0.1x_2 + 0.2x_3}{3}$$
$$x_2 = \frac{-19.3 - 0.1x_1 + 0.3x_3}{7}$$
$$x_3 = \frac{71.4 - 0.3x_1 + 0.2x_2}{10}$$

Iterative method: Gauss-Seidel Method

2^{nd} Step: Trail with $x_2 = x_3 = 0$

$$x_1 = \frac{7.85 + 0.1(0) + 0.2(0)}{3} = 2.616666667$$

3^{rd} Step: Trail with $x_3 = 0$ and employ x_1 from 1^{st} step

$$x_2 = \frac{-19.3 - 0.1(2.6166667) + 0.3(0)}{7} = -2.79452381$$

4^{th} Step: Trail with employ x_1 from 1^{st} step and x_2 from 2^{nd} step

$$x_3 = \frac{71.4 - 0.3(2.616666667) + 0.2(-2.79452381)}{10} = 7.005609524$$

Iterative method: Gauss-Seidel Method

5th step: Control error

$$\left|\varepsilon_{a,i}\right| = \left|\frac{x_i^n - x_i^{n-1}}{x_i^n}\right| \times 100 < \varepsilon_s$$

$$\left|\varepsilon_{x_1}\right| = \left|\frac{0 - 2.616666667}{2.616666667}\right| \times 100 = 100$$

$$\left|\varepsilon_{x_2}\right| = \left|\frac{0 - (-2.79452381)}{(-2.79452381)}\right| \times 100 = 100$$

$$\left|\varepsilon_{x_2}\right| = \left|\frac{0 - 7.005609524}{7.005609524}\right| \times 100 = 100$$

6th step: Repeat from 1st to 5th step

```
      program gaussseidel
      parameter (n=20)
      real*8 x1(0:n),x2(0:n),x3(0:n)
      open(1000,file='Gauss-seidel',status='unknown')
      x1(0) = 0.d0
      x2(0) = 0.d0
      x3(0) = 0.d0
      error1 = 100.d0
      error2 = 100.d0
      error3 = 100.d0
      write(1000,1001) i,x1(0),x2(0),x3(0),error1,error2,error3
      do 10 i = 1,n
            x1(i) = (7.85+0.1d0*x2(i-1)+0.2d0*x3(i-1))/3.d0
            x2(i) = (-19.3d0-0.1d0*x1(i)+0.3d0*x3(i-1))/7.d0
            x3(i) = (71.4d0-0.3d0*x1(i)+0.2d0*x2(i))/10.d0
            error1 = dabs((x1(i)-x1(i-1))/x1(i))*100.d0
            error2 = dabs((x2(i)-x2(i-1))/x2(i))*100.d0
            error3 = dabs((x3(i)-x3(i-1))/x3(i))*100.d0
            write(1000,1001) i,x1(i),x2(i),x3(i),error1,error2,error3
1001        format(i2,2x,3e15.7,2x,3e15.7)
10    continue
      close(1000)
      stop
```

Step	x1	x2	x3	error1	error2	error3
0	0.000000	0.000000	0.000000	1.00E+02	1.00E+02	1.00E+02
1	2.616667	-2.794524	7.005610	1.00E+02	1.00E+02	1.00E+02
2	2.990556	-2.499625	7.000291	1.25E+01	1.18E+01	7.60E-02
3	3.000032	-2.499988	6.999999	3.16E-01	1.45E-02	4.16E-03
4	3.000000	-2.500000	7.000000	1.05E-03	4.82E-04	1.01E-05
5	3.000000	-2.500000	7.000000	1.18E-05	1.41E-06	1.62E-07
6	3.000000	-2.500000	7.000000	6.44E-08	1.83E-08	6.97E-10
7	3.000000	-2.500000	7.000000	4.01E-10	9.05E-11	5.80E-12
8	3.000000	-2.500000	7.000000	3.42E-12	6.39E-13	5.08E-14
9	3.000000	-2.500000	7.000000	1.48E-14	1.78E-14	0.00E+00
10	3.000000	-2.500000	7.000000	0.00E+00	0.00E+00	0.00E+00
11	3.000000	-2.500000	7.000000	0.00E+00	0.00E+00	0.00E+00

Successive Over Relaxation method (SOR)

$$x_i^{new} = \lambda x_i^{new} + \left(1-\lambda\right) x_i^{old}$$ **where** $1 < \lambda < 2$

Example

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

1st Step:

$$x_1 = \frac{7.85 + 0.1x_2 + 0.2x_3}{3}$$
$$x_2 = \frac{-19.3 - 0.1x_1 + 0.3x_3}{7}$$
$$x_3 = \frac{71.4 - 0.3x_1 + 0.2x_2}{10}$$

Iterative method: Gauss-Seidel Method

2nd Step: Trail with $x_2 = x_3 = 0$

$$x_1 = \frac{7.85 + 0.1(0) + 0.2(0)}{3} = 2.616666667$$

3rd Step: find new x_1

$$x_i^{new} = \lambda x_i^{new} + (1-\lambda)\, x_i^{old}$$

$$\lambda = 1.0005$$

$$x_1 = (1.0005)(2.616666667) + (1-1.0005)(0) = 2.617975$$

Iterative method: Gauss-Seidel Method

4th Step: substitute x_1 and Trail with x_3 = 0

$$x_2 = \frac{-19.3 - 0.1(2.617975) + 0.3(0)}{7} = \text{-}2.7945425$$

5th Step: find new x_2

$$x_i^{new} = \lambda x_i^{new} + (1-\lambda)\, x_i^{old}$$

$$\lambda = 1.0005$$

$$x_2 = (1.0005)(\text{-}2.7945425) + (1-1.0005)(0) = \text{-}2.79594$$

Iterative method: Gauss-Seidel Method

6th Step: substitute x_1 and x_2

$$x_3 = \frac{71.4 - 0.3(2.617975) + 0.2(-2.79594)}{10} = 7.00554195$$

7th Step: find new x_2

$$x_i^{new} = \lambda x_i^{new} + (1-\lambda)\, x_i^{old}$$

$$\lambda = 1.0005$$

$$x_2 = (1.0005)(7.00554195) + (1-1.0005)(0) = 7.009044$$

8th step: Repeat from 1st to 7th step

```fortran
      program sor
      parameter (n=20)
      real*8 x1(0:n),x2(0:n),x3(0:n)
      open(1000,file='SOR-output',status='unknown')
      x1(0) = 0.d0
      x2(0) = 0.d0
      x3(0) = 0.d0
      error1 = 100.d0
      error2 = 100.d0
      error3 = 100.d0
      ramda = 1.0005d0
      write(1000,1001) i,x1(0),x2(0),x3(0),error1,error2,error3
      do 10 i = 1,n
            x1(i) = (7.85+0.1d0*x2(i-1)+0.2d0*x3(i-1))/3.d0
            x1(i) = ramda*x1(i)+(1.d0-ramda)*x1(i-1)
            x2(i) = (-19.3d0-0.1d0*x1(i)+0.3d0*x3(i-1))/7.d0
            x2(i) = ramda*x2(i)+(1.d0-ramda)*x2(i-1)
            x3(i) = (71.4d0-0.3d0*x1(i)+0.2d0*x2(i))/10.d0
            x3(i) = ramda*x3(i)+(1.d0-ramda)*x3(i-1)
            error1 = dabs((x1(i)-x1(i-1))/x1(i))*100.d0
            error2 = dabs((x2(i)-x2(i-1))/x2(i))*100.d0
            error3 = dabs((x3(i)-x3(i-1))/x3(i))*100.d0
            write(1000,1001) i,x1(i),x2(i),x3(i),error1,error2,error3
1001        format(i2,2x,3e15.7,2x,3e15.7)
10    continue
      close(1000)
      stop
      end
```

SOR

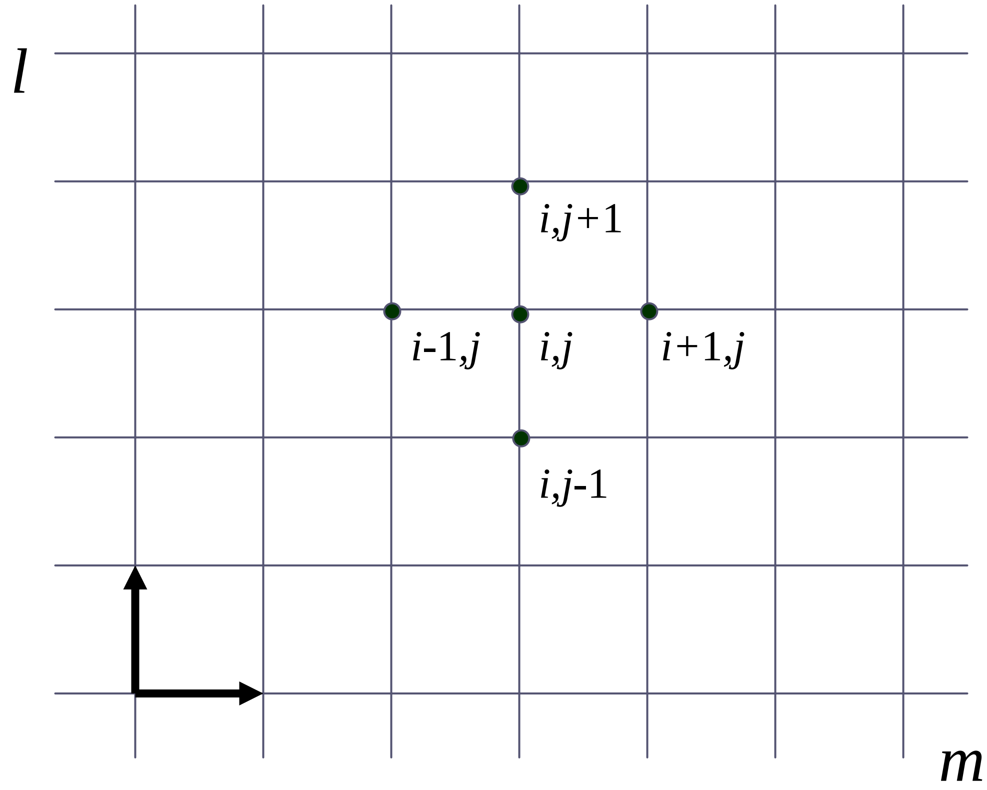
Step	x1	x2	x3	error1	error2	error3
0	0.000000	0.000000	0.000000	1.00E+02	1.00E+02	1.00E+02
1	2.617975	-2.795940	7.009044	1.00E+02	1.00E+02	1.00E+02
2	2.990925	-2.499335	7.000281	1.25E+01	1.19E+01	1.25E-01
3	3.000045	-2.499989	6.999999	3.04E-01	2.62E-02	4.04E-03
4	3.000000	-2.500000	7.000000	1.51E-03	4.46E-04	1.82E-05
5	3.000000	-2.500000	7.000000	8.79E-06	2.56E-06	1.22E-07
6	3.000000	-2.500000	7.000000	9.46E-08	1.18E-08	1.19E-09
7	3.000000	-2.500000	7.000000	9.42E-11	1.51E-10	1.67E-12
8	3.000000	-2.500000	7.000000	4.50E-12	3.55E-14	5.08E-14
9	3.000000	-2.500000	7.000000	0.00E+00	1.78E-14	0.00E+00
10	3.000000	-2.500000	7.000000	0.00E+00	0.00E+00	0.00E+00

Gauss-Seidel

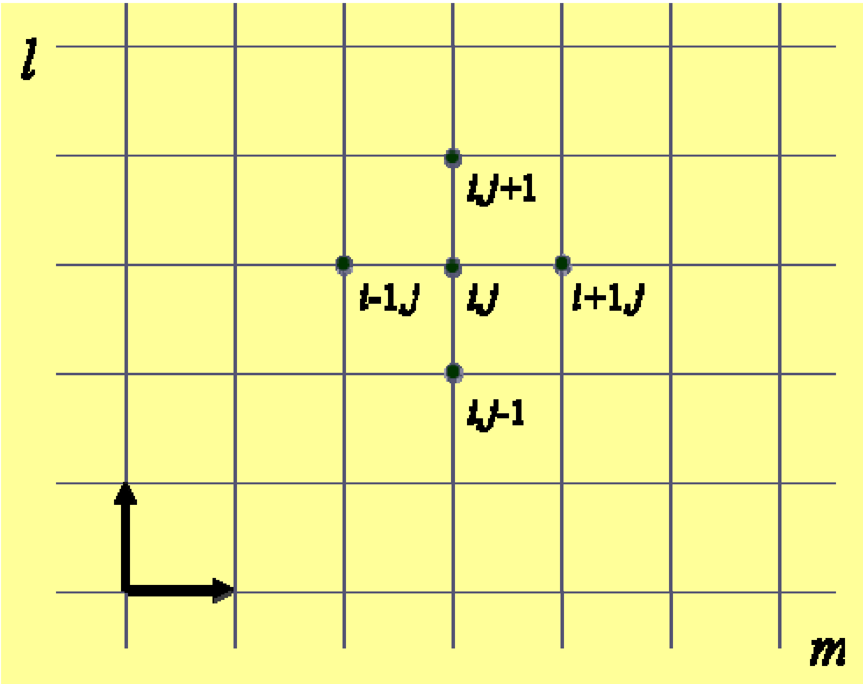
9	3.000000	-2.500000	7.000000	1.48E-14	1.78E-14	0.00E+00
10	3.000000	-2.500000	7.000000	0.00E+00	0.00E+00	0.00E+00

Solving the fluid dynamic equations

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$



Solving the fluid dynamic equations



$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

$$\left.\frac{\Delta u}{\Delta x}\right|_{i+\frac{1}{2},j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x}$$

$$\left.\frac{\Delta u}{\Delta x}\right|_{i-\frac{1}{2},j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x}$$

$$\left.\frac{\Delta^2 u}{\Delta x^2}\right|_{i,j} = \frac{1}{\Delta x}\left(\frac{\left(u_{i+1,j} - u_{i,j}\right)}{\Delta x} - \frac{\left(u_{i,j} - u_{i-1,j}\right)}{\Delta x}\right) = \left(\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2}\right)$$

$$\left.\frac{\Delta^2 u}{\Delta y^2}\right|_{i,j} = \frac{1}{\Delta y}\left(\frac{\left(u_{i,j+1} - u_{i,j}\right)}{\Delta y} - \frac{\left(u_{i,j} - u_{i,j-1}\right)}{\Delta y}\right) = \left(\frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2}\right)$$

Implicit Euler method

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \left(\frac{u_{i-1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i+1,j}^{n+1}}{\Delta x^2} \right) + \left(\frac{u_{j-1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j+1}^{n+1}}{\Delta y^2} \right)$$

$$u_{i,j}^{n} = -\frac{\Delta t}{\Delta x^2} u_{i-1,j}^{n+1} + \left(1 + \frac{2\Delta t}{\Delta x^2} + \frac{2\Delta t}{\Delta y^2} \right) u_{i,j}^{n+1} - \frac{\Delta t}{\Delta x^2} u_{i+1,j}^{n+1} - \frac{\Delta t}{\Delta y^2} u_{i,j-1}^{n+1} - \frac{\Delta t}{\Delta y^2} u_{i,j+1}^{n+1}$$

a **b** **c** **d** **e**

$$[A]\left\{ X_i^{n+1} \right\} = \left\{ X_i^{n} \right\}$$

Implicit Euler method

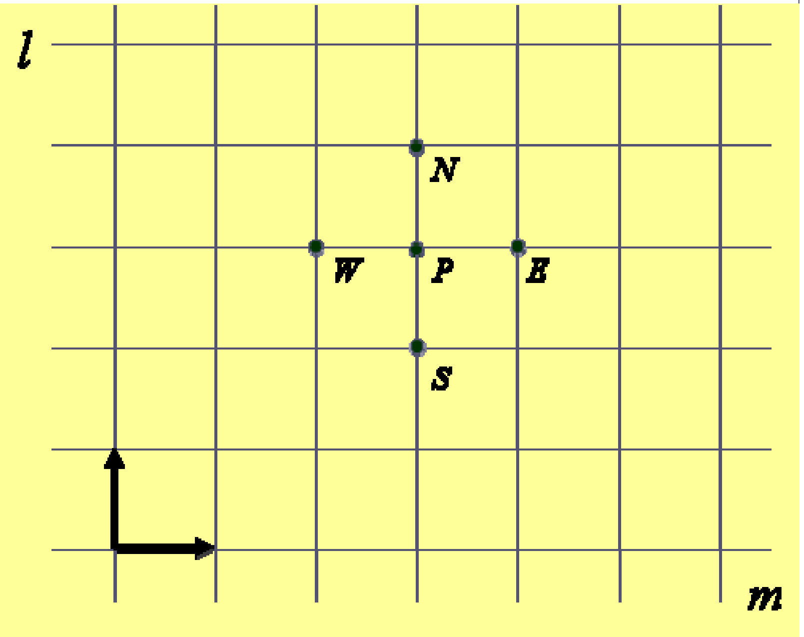
$$\left\{X_i^n\right\} = [A]\left\{X_i^{n+1}\right\}$$

$$u_{i,j}^n = -\frac{\Delta t}{\Delta x^2}u_{i-1,j}^{n+1} + \left(1+\frac{2\Delta t}{\Delta x^2}+\frac{2\Delta t}{\Delta y^2}\right)u_{i,j}^{n+1} - \frac{\Delta t}{\Delta x^2}u_{i+1,j}^{n+1} - \frac{\Delta t}{\Delta y^2}u_{i,j-1}^{n+1} - \frac{\Delta t}{\Delta y^2}u_{i,j+1}^{n+1}$$

Gauss-Seidel

$$u_{i,j}^{n+1} = \frac{u_{i,j}^n + \frac{\Delta t}{\Delta x^2}u_{i-1,j}^{n+1} + \frac{\Delta t}{\Delta x^2}u_{i+1,j}^{n+1} + \frac{\Delta t}{\Delta y^2}u_{i,j-1}^{n+1} + \frac{\Delta t}{\Delta y^2}u_{i,j+1}^{n+1}}{\left(1+\frac{2\Delta t}{\Delta x^2}+\frac{2\Delta t}{\Delta y^2}\right)}$$

$$\phi_P^{n+1} = \frac{Q_P - A_S\phi_S^{n+1} - A_N\phi_N^{n+1} - A_W\phi_W^{n+1} - A_E\phi_E^{n+1}}{A_P}$$



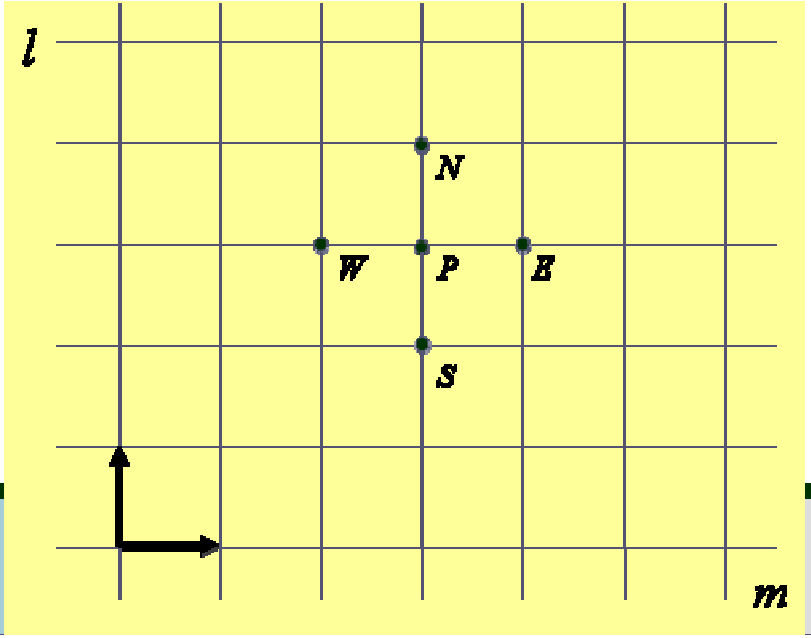
Implicit Euler method

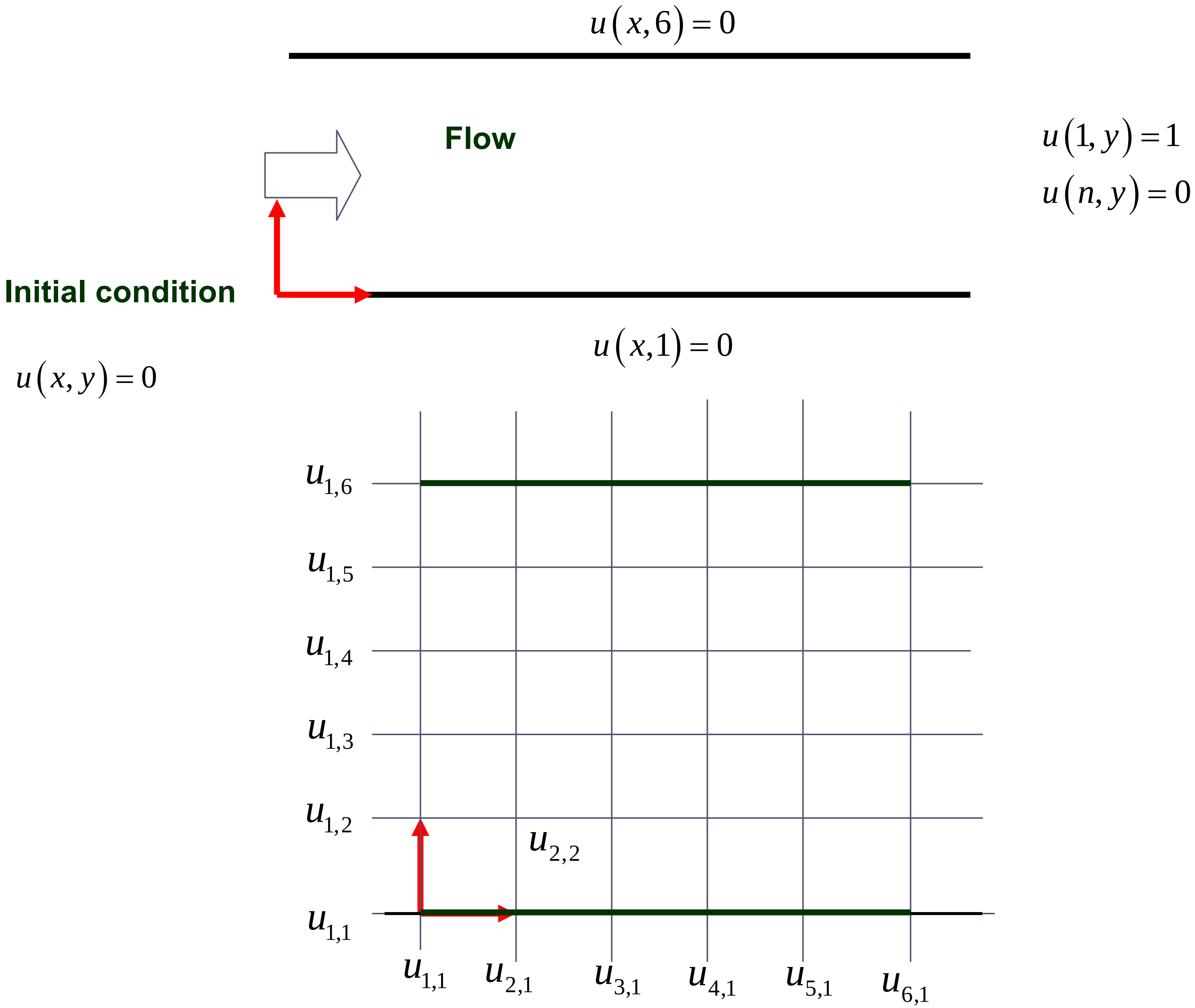
$$u_{i,j}^{n} = -\frac{\Delta t}{\Delta x^2}u_{i-1,j}^{n+1} + \left(1+\frac{2\Delta t}{\Delta x^2}+\frac{2\Delta t}{\Delta y^2}\right)u_{i,j}^{n+1} - \frac{\Delta t}{\Delta x^2}u_{i+1,j}^{n+1} - \frac{\Delta t}{\Delta y^2}u_{i,j-1}^{n+1} - \frac{\Delta t}{\Delta y^2}u_{i,j+1}^{n+1}$$

SOR

$$\phi_P^{n+1} = \lambda\frac{Q_P - A_S\phi_S^{n+1} - A_N\phi_N^{n+1} - A_W\phi_W^{n+1} - A_E\phi_E^{n+1}}{A_P} + (1-\lambda)\phi_P^{n}$$

$$u_P^{n+1} = \lambda\frac{u_P^{n} - A_S u_S^{n+1} - A_W u_W^{n+1} - A_N u_N^{n+1} - A_E u_E^{n+1}}{A_P} + (1-\lambda)u_P^{n}$$





$u(x,6)=0$
Flow
$u(1,y)=1$
$u(n,y)=0$
Initial condition
$u(x,1)=0$
$u(x,y)=0$
$u_{1,6}$
$u_{1,5}$
$u_{1,4}$
$u_{1,3}$
$u_{1,2}$
$u_{2,2}$
$u_{1,1}$
$u_{1,1}$ $u_{2,1}$ $u_{3,1}$ $u_{4,1}$ $u_{5,1}$ $u_{6,1}$

$$u_{i,j}^{n} = -\frac{\Delta t}{\Delta x^{2}} u_{i-1,j}^{n+1} + \left(1 + \frac{2\Delta t}{\Delta x^{2}} + \frac{2\Delta t}{\Delta y^{2}}\right) u_{i,j}^{n+1} - \frac{\Delta t}{\Delta x^{2}} u_{i+1,j}^{n+1} - \frac{\Delta t}{\Delta y^{2}} u_{i,j-1}^{n+1} - \frac{\Delta t}{\Delta y^{2}} u_{i,j+1}^{n+1}$$

a **b** **c** **d** **e**

$$\left[\begin{array}{cccccccccccccccc}
b & c & 0 & 0 & e & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a & b & c & 0 & 0 & e & 0 & & & & & & & & & 0 \\
0 & a & b & c & 0 & 0 & e & & & & & & & & & 0 \\
0 & 0 & a & b & c & 0 & 0 & e & & & & & & & & 0 \\
d & 0 & 0 & a & b & c & 0 & 0 & e & & & & & & & 0 \\
0 & d & 0 & 0 & a & b & c & 0 & 0 & e & & & & & & 0 \\
0 & & d & 0 & 0 & a & b & c & 0 & 0 & e & & & & & 0 \\
0 & & & d & 0 & 0 & a & b & c & 0 & 0 & e & & & & 0 \\
0 & & & & d & 0 & 0 & a & b & c & 0 & 0 & e & & & 0 \\
0 & & & & & d & 0 & 0 & a & b & c & 0 & 0 & e & & 0 \\
0 & & & & & & d & 0 & 0 & a & b & c & 0 & 0 & e & 0 \\
0 & & & & & & & d & 0 & 0 & a & b & c & 0 & 0 & e \\
0 & & & & & & & & d & 0 & 0 & a & b & c & 0 & 0 \\
0 & & & & & & & & & d & 0 & 0 & a & b & c & 0 \\
0 & & & & & & & & & & d & 0 & 0 & a & b & c \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d & 0 & 0 & a & b
\end{array}\right]
\left\{\begin{array}{c}
u_{2,2}^{n+1} \\ u_{3,2}^{n+1} \\ u_{4,2}^{n+1} \\ u_{5,2}^{n+1} \\ u_{2,3}^{n+1} \\ u_{3,3}^{n+1} \\ u_{4,3}^{n+1} \\ u_{5,3}^{n+1} \\ u_{2,4}^{n+1} \\ u_{3,4}^{n+1} \\ u_{4,4}^{n+1} \\ u_{5,4}^{n+1} \\ u_{2,5}^{n+1} \\ u_{3,5}^{n+1} \\ u_{4,5}^{n+1} \\ u_{5,5}^{n+1}
\end{array}\right\}
=
\left\{\begin{array}{c}
d_{2,2}^{n} \\ d_{3,2}^{n} \\ d_{4,2}^{n} \\ d_{5,2}^{n} \\ d_{2,3}^{n} \\ d_{3,3}^{n} \\ d_{4,3}^{n} \\ d_{5,3}^{n} \\ d_{2,4}^{n} \\ d_{3,4}^{n} \\ d_{4,4}^{n} \\ d_{5,4}^{n} \\ d_{2,5}^{n} \\ d_{3,5}^{n} \\ d_{4,5}^{n} \\ d_{5,5}^{n}
\end{array}\right\}
=
\left\{\begin{array}{c}
u_{2,2}^{n} - a u_{1,2}^{n+1} - d u_{2,1}^{n+1} \\
u_{3,2}^{n} - d u_{3,1}^{n+1} \\
u_{4,2}^{n} - d u_{4,1}^{n+1} \\
u_{5,2}^{n} - d u_{5,1}^{n+1} \\
u_{2,3}^{n} \\
u_{3,3}^{n} \\
u_{4,3}^{n} \\
u_{5,3}^{n} \\
u_{2,4}^{n} \\
u_{3,4}^{n} \\
u_{4,4}^{n} \\
u_{5,4}^{n} \\
u_{2,5}^{n} - e u_{2,6}^{n+1} \\
u_{2,5}^{n} - e u_{3,6}^{n+1} \\
u_{2,5}^{n} - e u_{4,6}^{n+1} \\
u_{2,5}^{n} - c u_{6,5}^{n+1} - d u_{5,6}^{n+1}
\end{array}\right\}$$

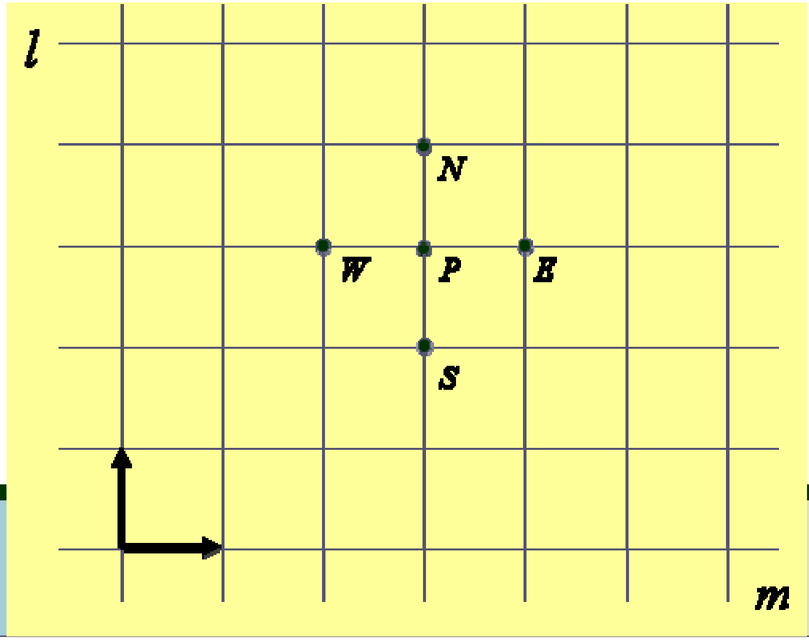
Implicit Euler method

$$u_{i,j}^{n} = -\frac{\Delta t}{\Delta x^2}u_{i-1,j}^{n+1} + \left(1+\frac{2\Delta t}{\Delta x^2}+\frac{2\Delta t}{\Delta y^2}\right)u_{i,j}^{n+1} - \frac{\Delta t}{\Delta x^2}u_{i+1,j}^{n+1} - \frac{\Delta t}{\Delta y^2}u_{i,j-1}^{n+1} - \frac{\Delta t}{\Delta y^2}u_{i,j+1}^{n+1}$$

SOR

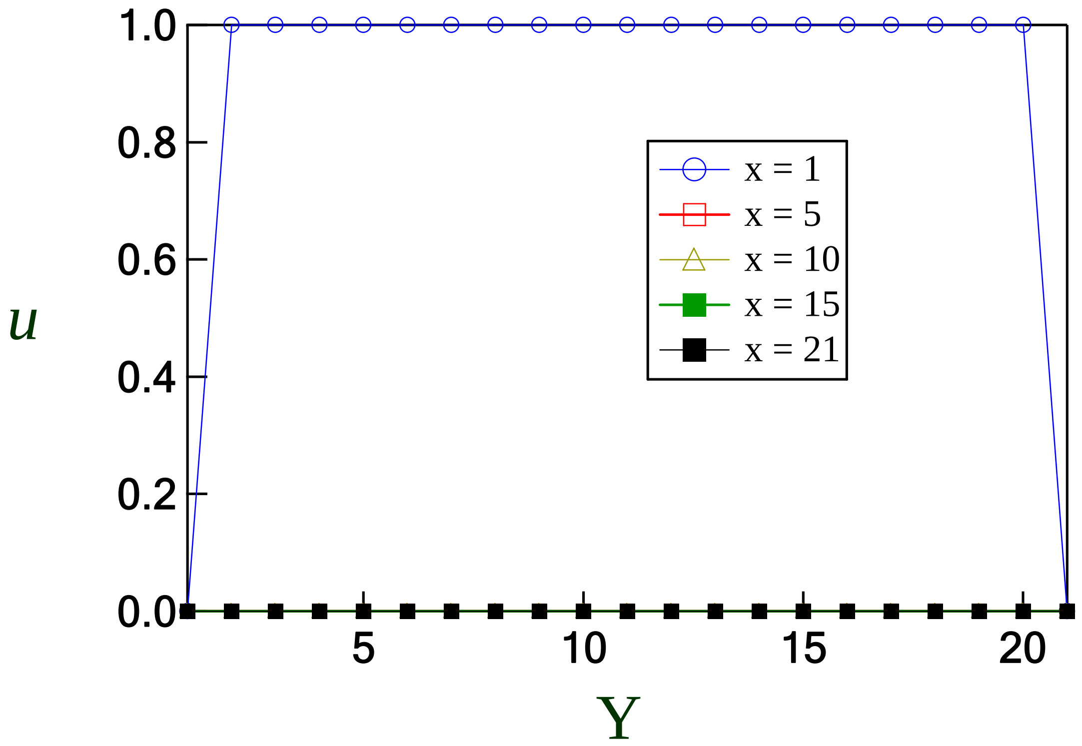
$$\phi_P^{n+1} = \lambda\frac{Q_P - A_S\phi_S^{n+1} - A_N\phi_N^{n+1} - A_W\phi_W^{n+1} - A_E\phi_E^{n+1}}{A_P} + (1-\lambda)\phi_P^{n}$$

$$u_P^{n+1} = \lambda\frac{d_P^{n} - A_S u_S^{n+1} - A_W u_W^{n+1} - A_N u_N^{n+1} - A_E u_E^{n+1}}{A_P} + (1-\lambda)u_P^{n}$$



```fortran
      program flowsor
      parameter(n=21,m =21,ntime=100)
      real*8 unew(n,m),uold(n,m),a,b,c,d,e
      open(1000,file='FlowSOR',status='unknown')
      do 10 j = 1,m
      do 20 i = 1,n
            unew(i,j) = 0.d0
            uold(i,j) = 0.d0
            unew(1,j) = 1.d0
            uold(1,j) = 1.d0
            unew(i,1) = 0.d0
            uold(i,m) = 0.d0
            unew(n,j) = 0.d0
20    continue
10    continue
      dx = 10.d0/dble(n-1)
      dy = 10.d0/dble(m-1)
      dt = 0.001d0
      a = -1.d0*dt/dx/dx
      b = 1.d0+2.d0/dx/dx+2.d0/dy/dy
      c = -1.d0*dt/dx/dx
      d = -1.d0*dt/dy/dy
      e = -1.d0*dt/dy/dy
      ramda = 1.5d0
```

```fortran
      do 100 nt = 1,ntime
            do 110 j = 2,m-1
            do 120 i = 2,n-1
            unew(1,j) = 1.d0
            uold(i,j) = unew(i,j)
            unew(n,j) = 0.d0
            unew(i,1) = 0.d0
            unew(i,m) = 0.d0
            unew(i,j) = ramda*(uold(i,j)-d*unew(i,j-1)-e*unew(i,j+1)
     #                  -a*unew(i-1,j)-c*unew(i+1,j))/b+(1.d0-ramda)*uold(i,j)
120   continue
110   continue
100   continue
      do 200 i = 1,n
      do 210 j = 1,m
            unew(i,1) = 0.d0
            unew(i,m) = 0.d0
      write(1000,1001) i,j,unew(i,j)
1001  format(2i2,2x,e15.8)
210   continue
200   continue
      close(1000)
      stop
      end
```



1.0
0.8
0.6
0.4
0.2
0.0
u
5
10
15
20
Y
x = 1
x = 5
x = 10
x = 15
x = 21