Today’s Lecture

- 2D grid
  - colocated arrangement
  - staggered arrangement
- Exercise: Make a Fortran program which solves a system of linear equations using an iterative method
- SIMPLE algorithm
  - Pressure-velocity correction method
- Finite Volume solution of Navier-Stokes equations
- Exercise: Finish solving the Navier Stokes equations
  - Lid driven cavity flow, Fortran program
What did we learn last time?

- Navier-Stokes equations
  \[
  \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left( \mu \frac{\partial u_i}{\partial x_j} \right) + S_M^i
  \]
  \[
  \frac{\partial \rho u_i}{\partial x_i} = 0
  \]

- Name the equations
- Name the different terms
- Which terms have we discretized?
- Which discretization schemes have been used for which terms?
Numerical Methods in Aerodynamics
Lecture 2: Numerical Solution of the Navier-Stokes equations

What needs to be considered?

- Previous $u$ was assumed known.
- $u$ needs to be found from the governing equations
  - momentum $\frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left( \mu \frac{\partial u_i}{\partial x_j} \right) + S_{M_i}$
  - continuity $\frac{\partial \rho u_i}{\partial x_i} = 0$

- Pressure gradients have not been considered yet.
  - No real governing equation for pressure
- The convective terms are nonlinear
- 2D grid vs. 1D grid
  $x_1 = x, \quad x_2 = y, \quad u_1 = u, \quad u_2 = v$
2D grid (colocated)

- Pressure, and velocities are evaluated at nodal points.
- The pressure gradients are needed where the velocities are located. I.e. at nodal points.
Checker-board pressure field

\[
\frac{\partial p}{\partial x} = \frac{p_e - p_w}{\Delta x} = \frac{p_E + p_P}{2} - \frac{p_E + p_W}{2} = \Delta x
\]

\[
\frac{\partial p}{\partial y} = \frac{p_n - p_s}{\Delta y} = \frac{p_N + p_P}{2} - \frac{p_P + p_S}{2} = \Delta y
\]

Is this true???
2D grid (staggered)

- Pressure is evaluated at nodal points.
- Velocities are evaluated at CV boundaries.
- \( u \) velocities are located at \( w \) and \( e \) boundary points
- \( v \) velocities are located at \( s \) and \( n \) boundary points
- The pressure gradients are needed where the velocities are located. I.e. at CV boundary points
Checker-board pressure field

\[ \frac{\partial p}{\partial x} \bigg|_w = \frac{p_P - p_W}{\Delta x_u} = \ldots \]

\[ \frac{\partial p}{\partial x} \bigg|_s = \frac{p_P - p_S}{\Delta y_v} = \ldots \]

- Is this better, than with the colocated arrangement???
- No interpolation of pressure is needed.
Numbering of staggered grid

- Grid lines are numbered by capital letters
  - x-direction $I-2, I-1, I, I+1, ...$
  - y-direction $J-2, J-1, J, J+1, ...$
- The scalar cell faces are numbered by lower case letters
  - x-direction $i-1, i, i+1, ...$
  - y-direction $j-1, j, j+1, ...$
- $u$-velocity points are numbered by (lower case, capital), e.g. $(i, J)$
- $v$-velocity points are numbered by (capital, lower case), e.g. $(I, j)$
The Navier Stokes equations for stationary, incompressible 2D flow

- x-momentum equation
  \[ \frac{\partial \rho uu}{\partial x} + \frac{\partial \rho uv}{\partial y} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x} \left( \mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial u}{\partial y} \right) + S_u \]

- y-momentum equation
  \[ \frac{\partial \rho vu}{\partial x} + \frac{\partial \rho vv}{\partial y} = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial x} \left( \mu \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial v}{\partial y} \right) + S_v \]

- continuity equation
  \[ \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0 \]
Discretization of the x-momentum equation \((u \text{ CV})\)

- Integrating over CV
- Using Gauss' theorem
  - Slide 11, lecture 1
- Convective terms
  - slide 31, lecture 1
- Viscous terms
  - slide 23, lecture 1
- Pressure gradient term
  - CDS
- Constant area, viscosity and density

\[
\begin{align*}
(\rho u u)_e - (\rho u u)_w + (\rho v u)_n - (\rho v u)_s & \\
= (\mu \frac{\partial u}{\partial x})_e - (\mu \frac{\partial u}{\partial x})_w + (\mu \frac{\partial u}{\partial y})_n - (\mu \frac{\partial u}{\partial y})_s - \Delta V \frac{p_e - p_w}{\Delta x} + \Delta V S_u
\end{align*}
\]
Discretization of the x-momentum equation (u CV)

- assume that $p$ is known at $e$ and $w$ face
- Nonlinear convective terms
  - Assume the convective fluxes are known
    \[ F_e = (\rho u)_e, F_w = (\rho u)_w \]
    \[ F_n = (\rho v)_n, F_w = (\rho v)_s \]
- UDS is used for convective terms (slide 32, lecture 1)
- CDS is used for viscous terms (slide 24, lecture 1)

\[
F_e u_P - F_w u_W + F_n \frac{1}{2}(u_N + u_P) - F_s \frac{1}{2}(u_P + u_S) \\
= \left( \frac{\mu}{\Delta x} \right)(u_E - u_P) - \left( \frac{\mu}{\Delta x} \right)(u_P - u_W) + \left( \frac{\mu}{\Delta y} \right)(u_N - u_P) - \left( \frac{\mu}{\Delta y} \right)(u_P - u_S) \\
- \Delta V \frac{p_e - p_w}{\Delta x} + \Delta V S_u
\]
Rearranging and using the global numbered notation

- **Convective fluxes**

\[ F_e = (\rho u)_e = \rho \frac{u_{i,J} + u_{i+1,J}}{2} \]

\[ F_w = (\rho u)_w = \rho \frac{u_{i-1,J} + u_{i,J}}{2} \]

\[ F_n = (\rho u)_n = \rho \frac{v_{I-1,j+1} + v_{I,j+1}}{2} \]

\[ F_s = (\rho u)_s = \rho \frac{v_{I-1,j} + v_{I,j}}{2} \]
Rearranging and using the numbered notation

- The x-momentum equation

\[ a_{i,j} u_{i,j} = a_{i-1,j} u_{i-1,j} + a_{i+1,j} u_{i+1,j} + a_{i,j-1} u_{i,j-1} + a_{i,j+1} u_{i,j+1} \]

\[ + S_u \Delta V \left( \frac{p_{I,j} - p_{I-1,j}}{\Delta x} \right) \Delta V \]
Rearranging and using the numbered notation

\[ a_{i,J}u_{i,J} = a_{i-1,J}u_{i-1,J} + a_{i+1,J}u_{i+1,J} + a_{i,J-1}u_{i,J-1} + a_{i,J+1}u_{i,J+1} + S_u\Delta V - \frac{p_{i,J} - p_{i-1,J}}{\Delta x} \Delta V \]

- Blue is assumed known from guessed (previous) values of \( u, v \) and \( p \)

\[ a_{i,J} = F_e + \frac{1}{2}F_n - \frac{1}{2}F_s + D_e + D_w + D_n + D_s \]

\[ a_{i-1,J} = F_w + D_w \]

\[ a_{i+1,J} = D_e \]

\[ a_{i,J-1} = \frac{1}{2}F_s + D_s \]

\[ a_{i,J+1} = -\frac{1}{2}F_n + D_n \]

\[ D_e = \frac{\mu}{\Delta x}, \quad D_w = \frac{\mu}{\Delta x}, \quad D_n = \frac{\mu}{\Delta y}, \quad D_s = \frac{\mu}{\Delta y} \]
Discretization of the y-momentum equation ($\nu \ CV$)
Discretization of the y-momentum equation ($\nu$ CV)

- **y-momentum**
  \[
  \frac{\partial \rho u \nu}{\partial x} + \frac{\partial \rho u \nu}{\partial y} = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial x} \left( \mu \frac{\partial \nu}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial \nu}{\partial y} \right) + S_v
  \]

- **Integration over CV, Gauss' theorem**
  \[
  (\rho u \nu)_e - (\rho u \nu)_w + (\rho u \nu)_n - (\rho u \nu)_s
  = (\mu \frac{\partial \nu}{\partial x})_e - (\mu \frac{\partial \nu}{\partial x})_w + (\mu \frac{\partial \nu}{\partial y})_n - (\mu \frac{\partial \nu}{\partial y})_s - \Delta V \frac{p_n - p_s}{\Delta y} + \Delta V S_v
  \]

- **Convective fluxes known. Constant area, viscousity and density**
  \[
  F_e = (\rho u)_e = \rho \frac{u_{i+1,j-1} + u_{i+1,j}}{2}, \quad F_w = (\rho u)_w = \rho \frac{u_{i,j} + u_{i,j-1}}{2},
  \]
  \[
  F_n = (\rho v)_n = \rho \frac{v_{i,j} + v_{i,j+1}}{2}, \quad F_s = (\rho v)_s = \rho \frac{v_{i,j} + v_{i,j-1}}{2}
  \]
Discretization of the y-momentum equation ($v \text{ CV}$)

- UDS is used for convective terms (slide 32, lecture 1)
- CDS is used for viscous terms (slide 24, lecture 1)

\[
F_c \frac{1}{2} (v_P + v_E) - F_w \frac{1}{2} (v_W + v_P) + F_n v_P - F_s v_S \\
= \left( \frac{\mu}{\Delta x} \right) (v_E - v_P) - \left( \frac{\mu}{\Delta x} \right) (v_P - v_W) + \left( \frac{\mu}{\Delta y} \right) (v_N - v_P) - \left( \frac{\mu}{\Delta y} \right) (v_P - v_S) \\
- \Delta V \frac{p_n - p_s}{\Delta x} + \Delta V S_u
\]
Rearranging and using the global numbered notation

\[ a_{I,j} v_{I,j} = a_{I,j-1} v_{I,j-1} + a_{I+1,j} v_{I+1,j} + a_{I,j+1} v_{I,j+1} + a_{I-1,j} v_{I-1,j} \]

\[ + S_v \Delta V - \frac{p_{I,j} - p_{I,j-1}}{\Delta y} \Delta V \]

- Blue is assumed known from guessed (previous) values of \( u, v \) and \( p \)

\[ a_{I,j} = F_n + \frac{1}{2} F_e - \frac{1}{2} F_w + D_e + D_w + D_n + D_s \]

\[ a_{I,j-1} = F_s + D_s \]

\[ a_{I+1,j} = -\frac{1}{2} F_e + D_e \]

\[ a_{I,j+1} = D_n \]

\[ a_{I-1,j} = \frac{1}{2} F_w + D_w \]

\[ D_e = \frac{\mu}{\Delta x}, \quad D_w = \frac{\mu}{\Delta x}, \quad D_n = \frac{\mu}{\Delta y}, \quad D_s = \frac{\mu}{\Delta y} \]
Summary, momentum equations

\[ a_{i,J}u_{i,J} = a_{i-1,J}u_{i-1,J} + a_{i+1,J}u_{i+1,J} + a_{i,J-1}u_{i,J-1} + a_{i,J+1}u_{i,J+1} + S_u \Delta V - \frac{p_{I,J} - p_{I-1,J}}{\Delta x} \Delta V \]

\[ a_{I,j}v_{I,j} = a_{I,j-1}v_{I,j-1} + a_{I+1,j}v_{I+1,j} + a_{I,j+1}v_{I,j+1} + a_{I-1,j}v_{I-1,j} + S_v \Delta V - \frac{p_{I,J} - p_{I,J-1}}{\Delta y} \Delta V \]

- The equations are solved iteratively
  - The fluxes giving the \( a \) coefficients and pressure are guessed.
  - The momentum equations are solved giving new values of \( u \) and \( v \)
  - The fluxes are updated

- The pressure is only guessed, i.e. the velocities doesn't satisfy continuity.

- The pressure needs to be updated. We still haven't used the continuity equation. But only velocities enters this equation!

\[ \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0 \]
Exercise:

- Problem

\[
\begin{bmatrix}
375 & -125 & 0 & 0 & 0 \\
-125 & 250 & -125 & 0 & 0 \\
0 & -125 & 250 & -125 & 0 \\
0 & 0 & -125 & 250 & -125 \\
0 & 0 & 0 & -125 & 375 \\
\end{bmatrix}
\begin{bmatrix}
T_1 \\
T_2 \\
T_3 \\
T_4 \\
T_5 \\
\end{bmatrix}
= 
\begin{bmatrix}
29000 \\
4000 \\
4000 \\
4000 \\
54000 \\
\end{bmatrix}
\]

- solve the system of linear equations using an iterative method

\[
T_{1}^{n+1} = \frac{1}{375} (125T_{2}^{n} + 29000)
\]

\[
T_{I}^{n+1} = \frac{1}{250} (125T_{I-1}^{n} + 125T_{I+1}^{n}) + 4000, \quad I = 2, 3, 4
\]

\[
T_{5}^{n+1} = \frac{1}{375} (125T_{4}^{n} + 54000)
\]

- write solution to a file
- load and plot solution in Matlab
BREAK

- Next: SIMPLE algorithm
  - Continuity equation
  - Pressure-velocity correction method
SIMPLe-algorithm

- Semi-Implicit Method for Pressure-Linked Equations (SIMPLE)

Variations
  - SIMPLER (SIMPLE-Revised), Patankar (1980)
  - SIMPLEC (SIMPLE-Consistent), Van Doormal and Raithby (1984)
  - PISO, Issa (1986)

Guess-and-correct procedure
Initiation of the procedure

- Guessed fields $p^*, u^*$ and $v^*$ are used
  
  $p = p^* + p', \quad u = u^* + u', \quad v = v^* + v'$

- $p', u'$ and $v'$ are the differences between the guessed fields and the correct $p, u$ and $v$. 
Inserting the guessed fields in the discretized momentum equations

- **x-momentum**

\[
a_{i,J}u_{i,J}^* = a_{i-1,J}u_{i-1,J}^* + a_{i+1,J}u_{i+1,J}^* + a_{i,J-1}u_{i,J-1}^* + a_{i,J+1}u_{i,J+1}^* + S_u \Delta V - \frac{p_{I,J}^* - p_{I-1,J}^*}{\Delta x} \Delta V
\]

\[
a_{i,J}u_{i,J}^* = \sum_{nb} a_{nb}u_{nb}^* + b_{i,J} - (p_{I,J}^* - p_{I-1,J}^*)A_{i,J}
\]

- **y-momentum**

\[
a_{I,j}v_{I,j}^* = a_{I,j-1}v_{I,j-1}^* + a_{I+1,j}v_{I+1,j}^* + a_{I,j+1}v_{I,j+1}^* + a_{I-1,j}v_{I-1,j}^* + S_v \Delta V - \frac{p_{I,J}^* - p_{I,J-1}^*}{\Delta y} \Delta V
\]

\[
a_{I,j}v_{I,j}^* = \sum_{nb} a_{nb}v_{nb}^* + b_{I,j} - (p_{I,J}^* - p_{I,J-1}^*)A_{I,j}
\]
Subtracting the discretized equations for guessed and correct fields

- **Correct field**
  \[ a_{i,J}u_{i,J} = \sum_{nb} a_{nb}u_{nb} + b_{i,J} - (p_{I,J} - p_{I-1,J})A_{i,J} \]

- **Guessed field**
  \[ a_{i,J}u_{i,J}^* = \sum_{nb} a_{nb}u_{nb}^* + b_{i,J} - (p_{I,J}^* - p_{I-1,J}^*)A_{i,J} \]

- **Difference**
  \[ a_{i,J}(u_{i,J} - u_{i,J}^*) = \sum_{nb} a_{nb}(u_{nb} - u_{nb}^*) - ((p_{I,J} - p_{I,J}^*) - (p_{I-1,J} - p_{I-1,J}^*))A_{i,J} \]

- **Correction equation, disregarding neighbour points**
  \[ a_{i,J}u_{i,J}' = \sum_{nb} a_{nb}u_{nb}' - (p_{I,J}' - p_{I-1,J}')A_{i,J} \]
CV for continuity equation (scalar CV)

\[ \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0 \]

- Integrating over CV, Gauss’ theorem

\[ (\rho u A)_e - (\rho u A)_w + (\rho v A)_n - (\rho v A)_s = 0 \]

- We know \( u \) and \( v \) at faces.

\[ u_{i,J} = u_{i,J}^* + u_{i,J}', \quad v_{I,j} = v_{I,j}^* + v_{I,j}' \]

- Global numbering

\[ (\rho u A)_{i+1,J} - (\rho u A)_{i,J} + (\rho v A)_{I,j+1} - (\rho v A)_{I,j} = 0 \]
Velocities and correction equations

\[(\rho u A)_{i+1,j} - (\rho u A)_{i,j} + (\rho v A)_{I,j+1} - (\rho v A)_{I,j} = 0\]

- **Velocities**
  \[u_{i,j} = u^*_{i,j} + u'_{i,j}, \quad \nu_{I,j} = \nu^*_{I,j} + \nu'_{I,j}\]
  \[u_{i+1,j} = u^*_{i+1,j} + u'_{i+1,j}, \quad \nu_{I,j+1} = \nu^*_{I,j+1} + \nu'_{I,j+1}\]

- **Correction equations**
  \[u'_{i,j} = -(p'_{I,j} - p'_{I,j-1}) \frac{A_{i,j}}{a_{i,j}}, \quad \nu'_{I,j} = -(p'_{I,j} - p'_{I,j-1}) \frac{A_{I,j}}{a_{I,j}}\]
  \[u'_{i+1,j} = -(p'_{I+1,j} - p'_{I,j}) \frac{A_{i+1,j}}{a_{i+1,j}}, \quad \nu'_{I,j+1} = -(p'_{I,j+1} - p'_{I,j}) \frac{A_{I,j+1}}{a_{I,j+1}}\]
Inserting velocities and correction equations

\[(\rho u A)_{i+1,J} - (\rho u A)_{i,J} + (\rho v A)_{I,j+1} - (\rho v A)_{I,j} = 0\]

- Discretized continuity equation

\[
(\rho A)_{i+1,J} \left( u_{i+1,J}^* - (p'_{I+1,J} - p'_{I,J}) \frac{A_{i+1,J}}{a_{i+1,J}} \right) \\
- (\rho A)_{i,J} \left( u_{i,J}^* - (p'_{I,J} - p'_{I-1,J}) \frac{A_{i,J}}{a_{i,J}} \right) \\
+ (\rho A)_{I,j+1} \left( v_{I,j+1}^* - (p'_{I,j+1} - p'_{I,j}) \frac{A_{I,j+1}}{a_{I,j+1}} \right) \\
- (\rho A)_{I,j} \left( v_{I,j}^* - (p'_{I,j} - p'_{I,j-1}) \frac{A_{I,j}}{a_{I,j}} \right) = 0
\]
Rearranging terms

\[
a_{I,J} = \frac{\rho_{I+1,J} A_{I+1,J}^2}{a_{I+1,J}} + \frac{\rho_{I-1,J} A_{I-1,J}^2}{a_{I-1,J}} + \frac{\rho_{I,J+1} A_{I,J+1}^2}{a_{I,J+1}} + \frac{\rho_{I,J-1} A_{I,J-1}^2}{a_{I,J-1}} + b'_{I,J}
\]

\[
a_{I+1,J} = \frac{\rho_{I+1,J} A_{I+1,J}^2}{a_{I+1,J}}
\]

\[
a_{I-1,J} = \frac{\rho_{I-1,J} A_{I-1,J}^2}{a_{I-1,J}}
\]

\[
a_{I,J+1} = \frac{\rho_{I,J+1} A_{I,J+1}^2}{a_{I,J+1}}
\]

\[
a_{I,J-1} = \frac{\rho_{I,J-1} A_{I,J-1}^2}{a_{I,J-1}}
\]

\[
b'_{I,J} = (\rho u^* A)_{i,J} - (\rho u^* A)_{i+1,J} + (\rho v^* A)_{I,j} - (\rho v^* A)_{I,j+1}
\]
Under-relaxation

- The pressure correction equation may diverge. To avoid this only a fraction of the pressure correction is added to the guessed pressure (under-relaxation). \( n \) indicates iteration number.

\[
P^n = P^{*-1} + \alpha_p P'\]

- This stabilizes the algorithm but makes convergence slower
- Optimal choice of \( \alpha_p \) is problem dependent
- The updating of the velocities are also under-relaxed

\[
u^n = \alpha_u (u^{*-1} + u') + (1 - \alpha_u)u^{n-1}
\]

\[
v^n = \alpha_v (v^{*-1} + v') + (1 - \alpha_v)v^{n-1}
\]
Summary of the SIMPLE-algorithm

1. Initial guess of $p^*$, $u^*$ and $v^*$
2. Solve the discretized momentum equations for updated values of $u^*$ and $v^*$
   \[ a_{i,j}u_{i,j}^* = \sum_{nb} a_{nb}u_{nb}^* + b_{i,j} - (p_{I,J}^* - p_{I-1,J}^*)A_{i,j} \]
   \[ a_{I,j}v_{i,j}^* = \sum_{nb} a_{nb}v_{nb}^* + b_{I,j} - (p_{I,J}^* - p_{I,J-1}^*)A_{I,j} \]
3. Solve the pressure correction equation for $p'$
   \[ a_{I,J}p'_{I,J} = a_{I+1,J}p'_{I+1,J} + a_{I-1,J}p'_{I-1,J} + a_{I,J+1}p'_{I,J+1} + a_{I,J-1}p'_{I,J-1} + b'_{I,J} \]
4. Correct pressures and velocities using under-relaxation
   \[ p^n = p^{*n-1} + \alpha_pp^m \]
   \[ u_{i,j}^* = -(p'_{I,J} - p'_{I-1,J}) \frac{A_{i,j}}{a_{i,j}}, \quad v_{i,j}^* = -(p'_{I,J} - p'_{I,J-1}) \frac{A_{I,j}}{a_{I,j}} \]
   \[ u^n - \alpha_u(u^* + u^m) + (1 - \alpha_u)u^{n-1}, \quad v^n - \alpha_v(v^* + v^m) + (1 - \alpha_v)v^{n-1} \]
5. Repeat 2 with $p^* = p^n$, $u^* = u^n$ and $v^* = v^n$ until convergence is reached.
General comments

- The coefficients are dependent on the choice of discretization scheme. They are determined from the convective fluxes, i.e. they are updated each iteration.
- Solutions of the momentum and pressure correction equations have not been discussed. Iterative or direct methods may be adopted.
- Implementation of boundary conditions has not been discussed.
  - inlet, outlet, wall, prescribed pressure, symmetry, periodicity. See e.g. Versteeg
- Other discretization approaches can be used
  - Finite difference, Finite element
- Compressible and non-stationary flows have no been discussed.
- Only Cartesian meshes have been illustrated. Unstructured meshes are often used in commercial codes.
Generel comments

- The Reynolds number $Re$ is the ratio of inertial forces ($\frac{u\rho}{\mu}$) to viscous forces ($\frac{\mu}{L}$) and is used for determining whether a flow will be laminar or turbulent.
- Laminar flow occurs at low Reynolds numbers, where viscous forces are dominant, and is characterized by smooth, constant fluid motion, while turbulent flow, on the other hand, occurs at high Reynolds numbers and is dominated by inertial forces, producing random eddies, vortices and other flow fluctuations.
- Turbulence models are used to describe these fluctuations.
  - Large-eddy simulations (LES)
  - Reynolds-Averaged Navier-Stokes equations (RANS)
Next: Exercise problem
   ◆ Going through the main elements of the problem
   ◆ Exercise: Find out what is going on in the finer parts of the program
Numerical Methods in Aerodynamics

Lecture 2: Numerical Solution of the Navier-Stokes equations

Fortran project

Program COMET
This program incorporates the FV method for solving the Navier-Stokes equations using 2D, Cartesian grids and the staggered arrangement of variables. Variables are stored as 2D arrays. SIMPLE method is used for pressure calculation. UDS and CBS are implemented for the discretization of convective terms. CBS is used for the diffusive terms. The boundary conditions are set for the lid-driven cavity flow. Only steady flows are considered.

M. Peric, IFS, Hamburg, 1996

Program PLOT
This code produces plots of grid, velocity vectors, and profiles, contours lines and color fills for any quantity. The output are postscript files for each page. The code can easily be adapted for interactive use and screen window output. However, this is to some extent hardware-dependent. Only postscript output is provided here. The same code is used for both Cartesian and non-orthogonal grids, and for both single-grid and multi-grid solutions. Array dimensions NX and NY should be equal to or greater than the maximum number of nodes in respective directions on the finest grid. Up to 128 contours can be plotted. The plots are saved on files which carry the non-grid number, eg. VECD.1S for the plot of velocity vectors from grid 1.

M. Peric, IFS, Hamburg, 1996

Program EXPA
Program for interactive generation of rectilinear orthogonal grids for the 2D multigrid flow prediction code

M. PERIC, IFS Hamburg, 1995
Main parts of the project

- **Grid generation**
  - Read the input file containing the problem definition
  - Generating grid
  - Write the grid to an output file

- **Setting up and solve the system**
  - Read the input file containing the problem definition
  - Read the grid file
  - Define initial conditions including boundary conditions
  - Solve the system equations until convergence is reached
  - Write the result to an output file

- **Plotting the results**
  - Read the input file containing the problem definition
  - Read the solution from the output file
  - Plot the desired results
Exercise: Investigate the main parts of the program

- Look through the project files grid.f90, pstag.f90 and plot.f90
  - Find the sections where the previous mentioned parts are located
  - Try to identify the steps of the SIMPLE algorithm
- Run the files with various problem definitions
  - change the lid velocity
  - change the fluid properties
  - change the grid size
What have we learned?

- Discretize the governing equations for fluid flow (convection and diffusion) using the Finite Volume method
- Central difference scheme for viscous terms
- Upwind difference scheme for convective terms
- Non-linear convective terms are solved iteratively by defining convective fluxes from previous iteration
- The continuity equation is used as a pressure correction equation to introduce effects from the pressure gradient term
- There is A LOT OF BOOKKEEPING!!!
- Fortran
  - Read from and write to files (and screen)
  - loops
  - program, subroutines, functions
Next lecture

- Minor finite element structural models may be solved on a single PC.
- Minor flow problems may be solved on a single PC.
- When the problems increase in size the memory of one PC may not suffice.
- Calculation time may be decreased from dividing the system into minor problems and solved on separate PC's.
- We will look at a cluster of PC's and learn the basic programming skills for solving problems on parallel systems.
Thank you for your attention