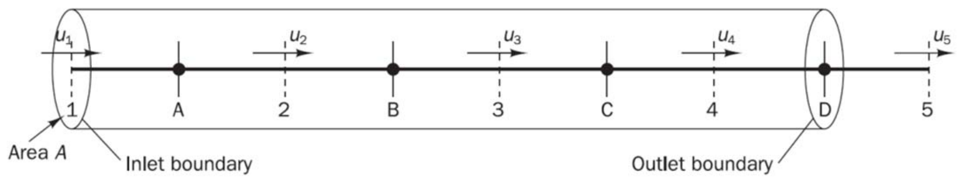
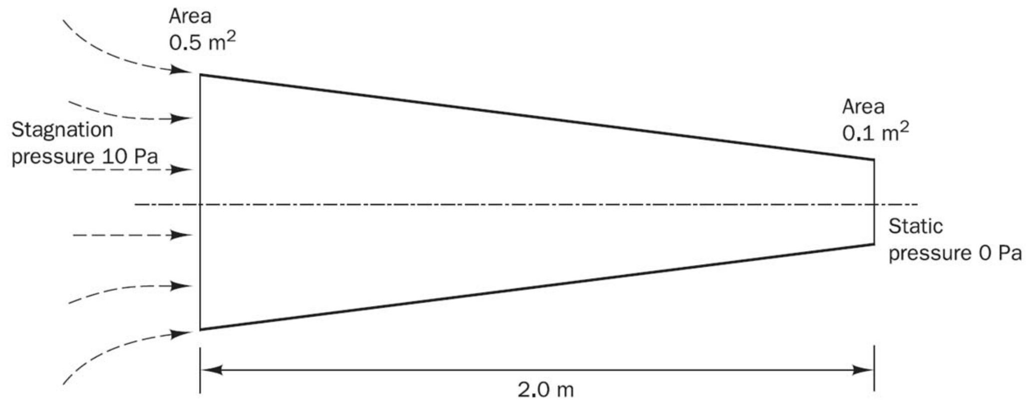
Contents

Worked examples of the SIMPLE algorithm

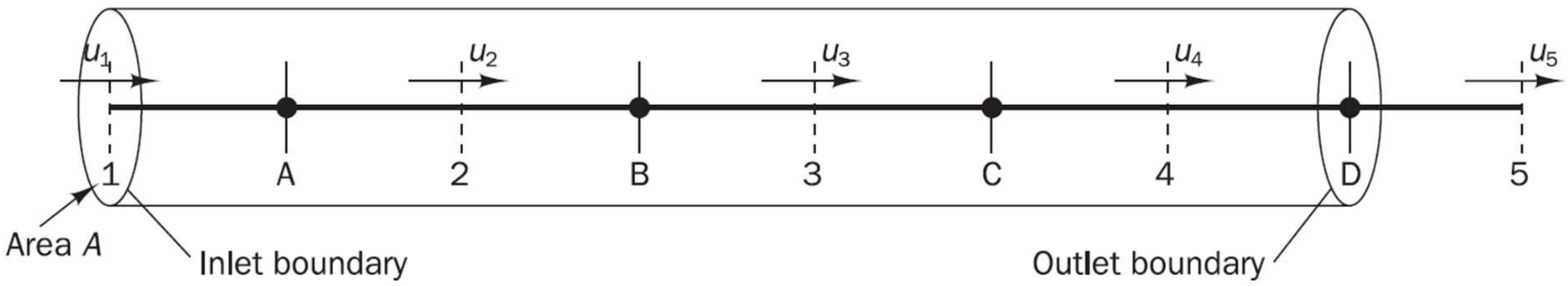
- 1D Flows
  - 1^{st} example: frictionless, incompressible flow through a duct of constant cross-sectional area
    - Constant velocity $\Rightarrow$ trivial solution
    - How an initial guess with varying velocities along the length of the duct is updated to satisfy mass conservation using the pressure correction equation.
  - 2^{nd} example: frictionless, incompressible flow through a planar, converging nozzle
    - With an assumption that the flow is unidirectional and all flow variables are uniformly distributed throughout every cross-section perpendicular to the flow direction
    - We can develop a set of one-dimensional governing equations for the problem.





Worked examples of the SIMPLE algorithm

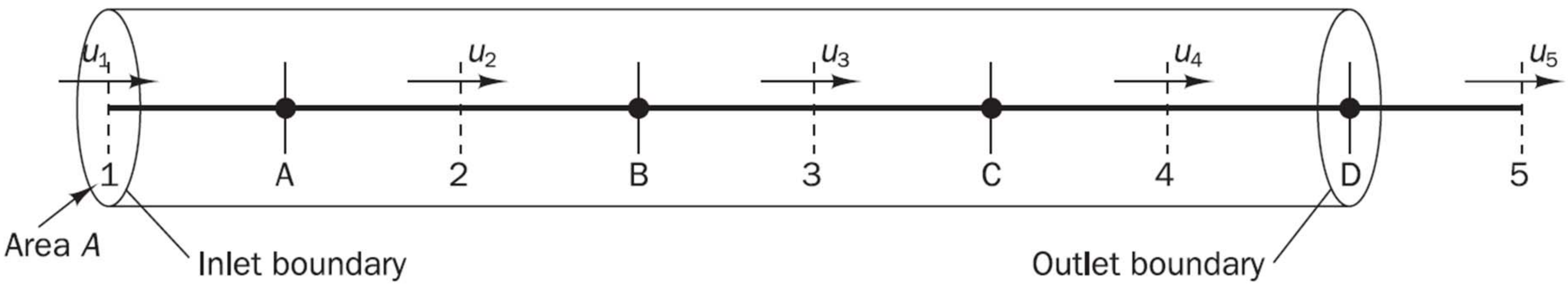
- Example 6.1
  - Steady, one-dimensional flow of a constant-density fluid through a duct with constant cross-sectional area
  - Starting point
    - Assume that we have used a guessed pressure field p^* in the discretized momentum equation to obtain a guessed velocity field u^*.
    - Pressure correction equation only!
  - We demonstrate the guess-and-correct procedure that forms the basis of the SIMPLE algorithm.



Worked examples of the SIMPLE algorithm

- Example 6.1
  - Problem data
    - Density $\rho = 1.0\ kg/m^3$ is constant.
    - Duct area A is constant.
    - $d = 1.0 \Rightarrow$ not realistic, just for practice
    - $u_1 = 10$ m/s and $p_D = 0$ Pa
    - Initial guessed velocity field: $u_2^* = 8.0$ m/s, $u_3^* = 11.0$ m/s and $u_4^* = 7.0$ m/s.

  - In this very straightforward problem with constant area and constant density, it is easy to see that the velocity must be constant everywhere by continuity.
  - Hence, we will be able to compare our computed solution against the exact solution $u_2 = u_3 = u_4 = 10$ m/s.



Worked examples of the SIMPLE algorithm

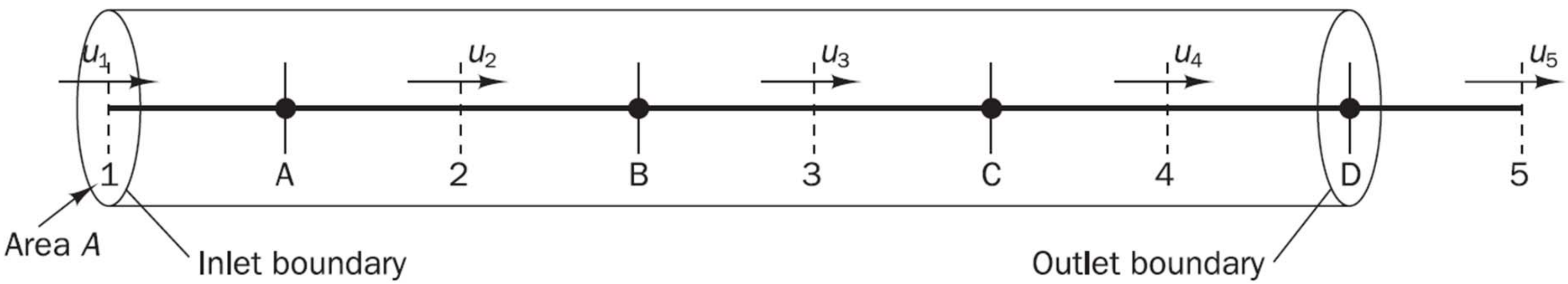
❖ Example 6.1

$$a_{I,J}p'_{I,J} = a_{I+1,J}p'_{I+1,J} + a_{I-1,J}p'_{I-1,J} + a_{I,J+1}p'_{I,J+1} + a_{I,J-1}p'_{I,J-1} + b'_{I,J}$$

$a_{I+1,J}$	$a_{I-1,J}$	$a_{I,J+1}$	$a_{I,J-1}$	$b'_{I,J}$
$(\rho dA)_{i+1,J}$	$(\rho dA)_{i,J}$	$(\rho dA)_{I,j+1}$	$(\rho dA)_{I,j}$	$(\rho u^* A)_{i,J} - (\rho u^* A)_{i+1,J}$ $+ (\rho v^* A)_{I,j} - (\rho v^* A)_{I,j+1}$

$$u'_{i,J} = d_{i,J}(p'_{I-1,J} - p'_{I,J}) \qquad u' = d(p'_I - p'_{I+1}) \qquad u = u^* + u'$$

where $d_{i,J} = \dfrac{A_{i,J}}{a_{i,J}}$ and $d_{I,j} = \dfrac{A_{I,j}}{a_{I,j}}$



Worked examples of the SIMPLE algorithm

- Example 6.1
  - Solution

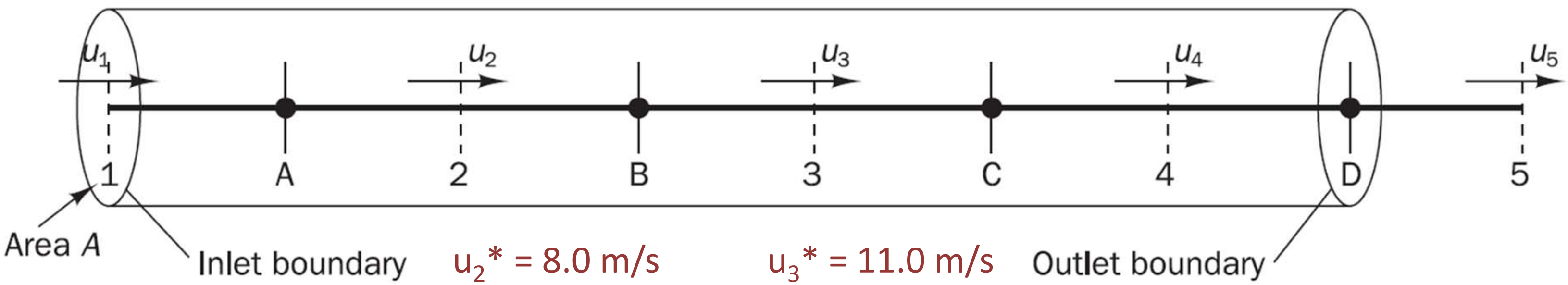
$$a_P p'_P = a_W p'_W + a_E p'_E + b'$$

$$a_W = (\rho dA)_w \qquad a_E = (\rho dA)_e \qquad a_P = a_W + a_E \qquad b' = (\rho u^* A)_w - (\rho u^* A)_e$$

Node B

$$\begin{aligned}
a_W &= (\rho dA)_w = (\rho dA)_2 = 1.0 \times 1.0 \times A = 1.0A \\
a_E &= (\rho dA)_e = (\rho dA)_3 = 1.0 \times 1.0 \times A = 1.0A \\
a_P &= a_W + a_E = 1.0A + 1.0A = 2.0A \\
b' &= (\rho u^* A)_w - (\rho u^* A)_e = (\rho u^* A)_2 - (\rho u^* A)_3 \\
&= (1.0 \times 8. \times A) - (1.0 \times 11. \times A) = -3.0A
\end{aligned}$$

$$(2.0A)p'_B = (1.0A)p'_A + (1.0A)p'_C + (-3.0A) \qquad \boxed{2p'_B = p'_A + p'_C - 3}$$



$u_2^* = 8.0$ m/s

$u_3^* = 11.0$ m/s

$u_4^* = 7.0$ m/s

Worked examples of the SIMPLE algorithm

- Example 6.1
  - Solution

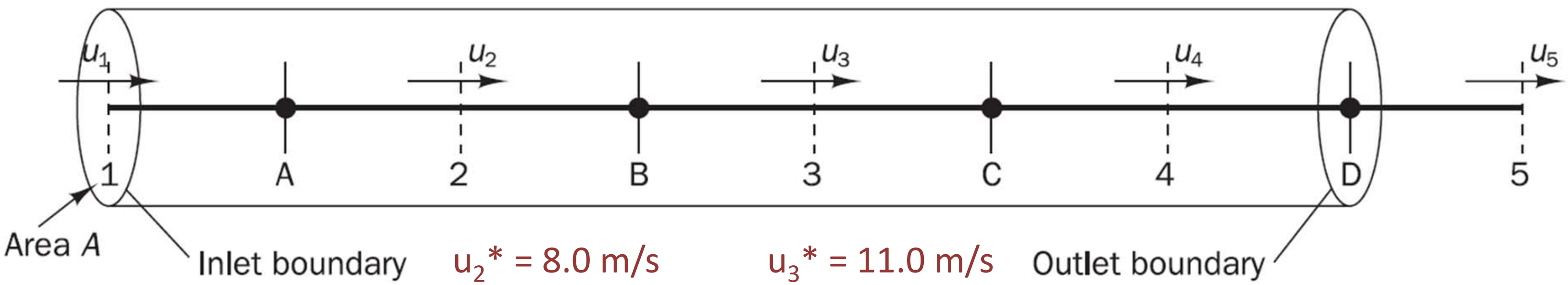
$$a_P p'_P = a_W p'_W + a_E p'_E + b'$$

$$a_W = (\rho d A)_w \qquad a_E = (\rho d A)_e \qquad a_P = a_W + a_E \qquad b' = (\rho u^* A)_w - (\rho u^* A)_e$$

Node C

$$\begin{aligned}
a_W &= (\rho dA)_w = (\rho dA)_3 = 1.0 \times 1.0 \times A = 1.0A \\
a_E &= (\rho dA)_e = (\rho dA)_4 = 1.0 \times 1.0 \times A = 1.0A \\
a_P &= a_W + a_E = 1.0A + 1.0A = 2.0A \\
b' &= (\rho u^* A)_w - (\rho u^* A)_e = (\rho u^* A)_3 - (\rho u^* A)_4 \\
&= (1.0 \times 11. \times A) - (1.0 \times 7. \times A) = 4.0A
\end{aligned}$$

$$(2.0A)p'_C = (1.0A)p'_B + (1.0A)p'_D + (4.0A) \qquad \boxed{2p'_C = p'_B + p'_D + 4}$$



$u_2^* = 8.0$ m/s $\qquad u_3^* = 11.0$ m/s $\qquad u_4^* = 7.0$ m/s

Worked examples of the SIMPLE algorithm

- Example 6.1
  - Solution

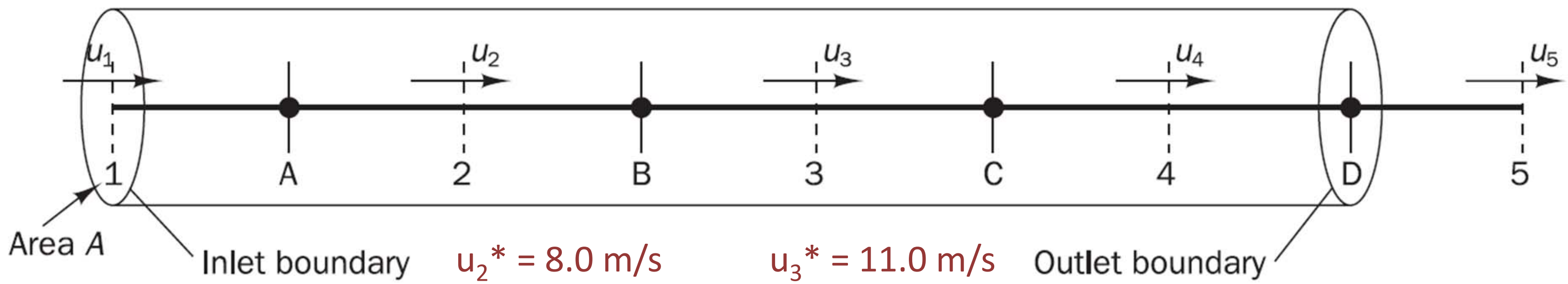
$$a_P p'_P = a_W p'_W + a_E p'_E + b'$$

$$a_W = (\rho dA)_w \qquad a_E = (\rho dA)_e \qquad a_P = a_W + a_E \qquad b' = (\rho u^* A)_w - (\rho u^* A)_e$$

Node A

$$a_W = 0.0$$
$$a_E = (\rho dA)_e = (\rho dA)_2 = 1.0 \times 1.0 \times A = 1.0A$$
$$a_P = a_W + a_E = 0.0 + 1.0A = 1.0A$$
$$\begin{aligned} b' = (\rho u^* A)_w - (\rho u^* A)_e + (\rho u A)_{boundary} &= -(\rho u^* A)_2 + (\rho u A)_1 = \\ &= -(1.0 \times 8. \times A) + (1.0 \times 10. \times A) \\ &= 2.0A \end{aligned}$$

$$(1.0A)p'_A = 0 + (1.0A)p'_B + (2.0A) \qquad \boxed{p'_A = p'_B + 2.0}$$



$u_2^* = 8.0$ m/s $\quad u_3^* = 11.0$ m/s $\quad u_4^* = 7.0$ m/s

Worked examples of the SIMPLE algorithm

- Example 6.1
  - Solution

$$a_P p'_P = a_W p'_W + a_E p'_E + b'$$

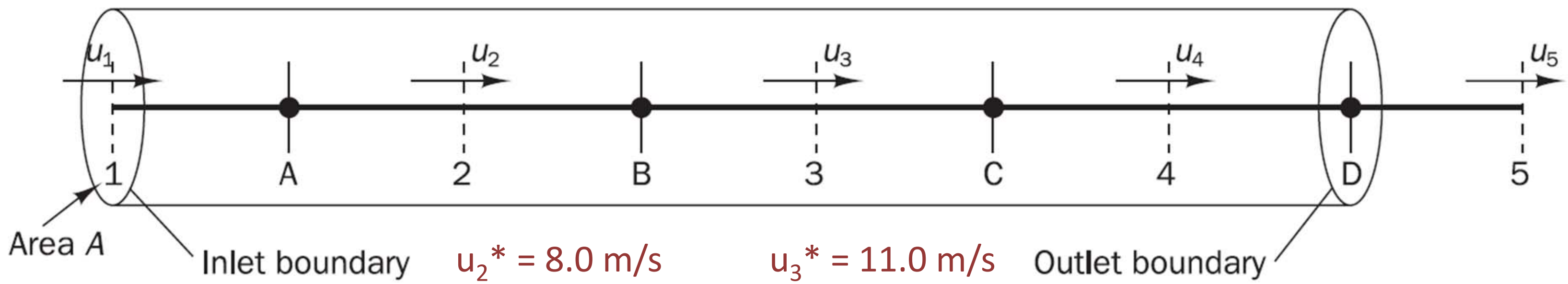
$$a_W = (\rho d A)_w \qquad a_E = (\rho d A)_e \qquad a_P = a_W + a_E \qquad b' = (\rho u^* A)_w - (\rho u^* A)_e$$

Node D $p_D = 0$

$p'_D = 0$

For all cells,

$$\boxed{\begin{aligned} p'_A &= p'_B + 2 \\ 2p'_B &= p'_A + p'_C - 3 \\ 2p'_C &= p'_B + p'_D + 4 \\ p'_D &= 0 \end{aligned}}$$



$u_2^* = 8.0$ m/s $\quad u_3^* = 11.0$ m/s $\quad u_4^* = 7.0$ m/s

Worked examples of the SIMPLE algorithm

- Example 6.1
  - System pressure correction equation

$$\boxed{\begin{aligned} p'_A &= p'_B + 2 \\ 2p'_B &= p'_A + p'_C - 3 \\ 2p'_C &= p'_B + p'_D + 4 \\ p'_D &= 0 \end{aligned}}$$

$$2p'_C = p'_B + 4$$

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} p'_A \\ p'_B \\ p'_C \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \\ 4 \end{bmatrix}$$

$p'_A = 4.0$, $p'_B = 2.0$ and $p'_C = 3.0$ (with $p'_D = 0$ as before)

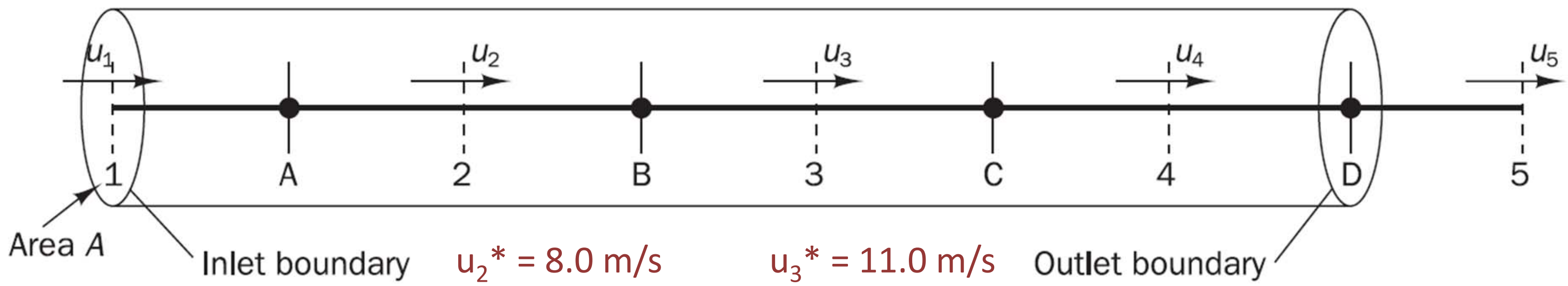
$$u = u^* + d(p'_I - p'_{I+1})$$

Velocity node 2: $u_2 = u_2^* + d(p'_A - p'_B) = 8.0 + 1.0 \times [4.0 - 2.0] = 10.0 \text{ m/s}$

Velocity node 3: $u_3 = u_3^* + d(p'_B - p'_C) = 11.0 + 1.0 \times [2.0 - 3.0] = 10.0 \text{ m/s}$

Velocity node 4: $u_4 = u_4^* + d(p'_C - p'_D) = 7.0 + 1.0 \times [3.0 - 0.0] = 10.0 \text{ m/s}$

Exact velocity field in a single iteration!



$u_2^* = 8.0$ m/s

$u_3^* = 11.0$ m/s

$u_4^* = 7.0$ m/s

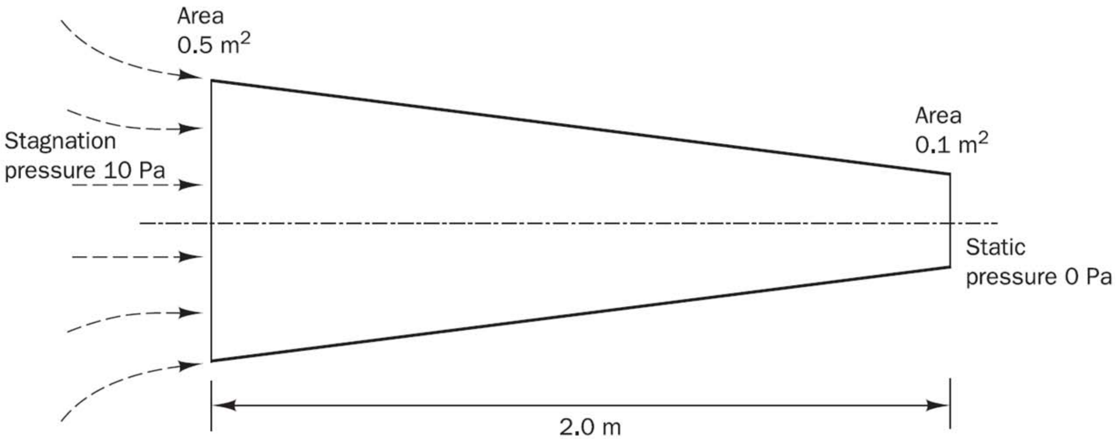
Worked examples of the SIMPLE algorithm

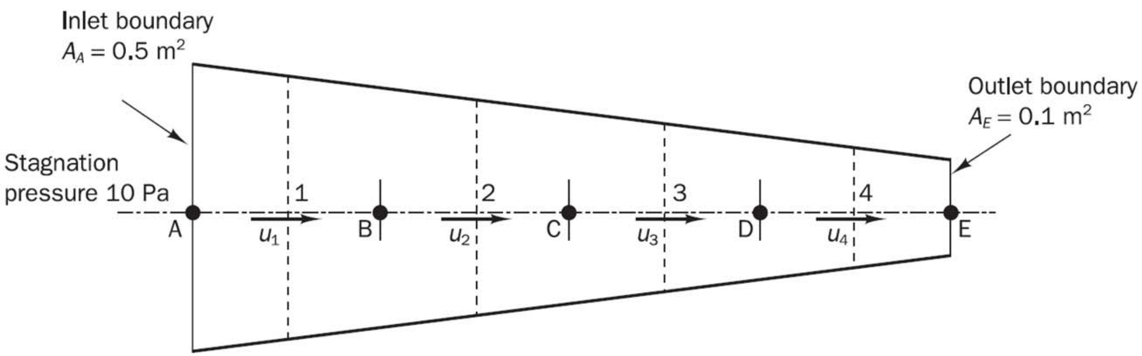
- Example 6.1
  - Discussion
    - Pressure correction equation only
    - In more general problem, the pressure and velocity fields are coupled.
    - So the pressure correction equation must be solved along with the discretized momentum equations.
    - The value of *d:* assumed to be constant
      - Normally, the value of *d* will vary from node to node
      - Should be calculated using control volume face areas and central coefficient values from the discretized momentum equations.
    - This process will be illustrated in the next example.

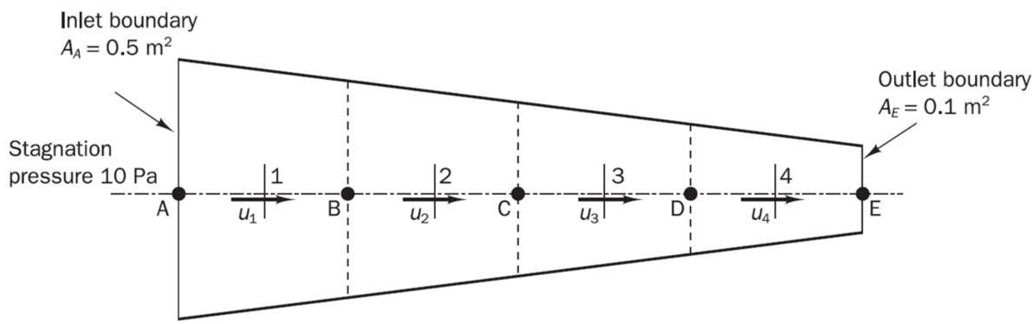
Worked examples of the SIMPLE algorithm

- Example 6.2
  - Planar two-dimensional nozzle
  - The flow is steady and frictionless and the density of the fluid is constant.
  - Backward-staggered grid with five pressure nodes and four velocity nodes
  - The stagnation pressure is given at the inlet and the static pressure is specified at the exit.

$$p_A = p_0 - \frac{1}{2}(\rho u_A^2)$$







Worked examples of the SIMPLE algorithm

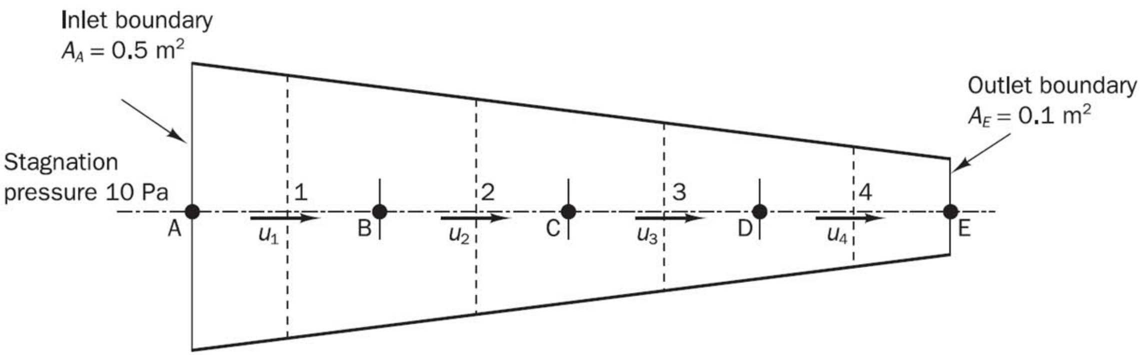
- Example 6.2
  - Problem data
    - ρ= 1.0 kg/m^3
    - L = 2.00 m; $\Delta x = L/4 = 2.00/4 = 0.5$ m
    - Inlet area: A_A=0.5 m^2, outlet area: A_E=0.1 m^2
      - Area change is a linear function of distance from the nozzle inlet.
    - Boundary conditions
      - Inlet p_0 = 10 Pa
      - Exit p_E = 0 Pa
    - Initial guess
      - m=1.0 kg/s
      - Linear pressure variation

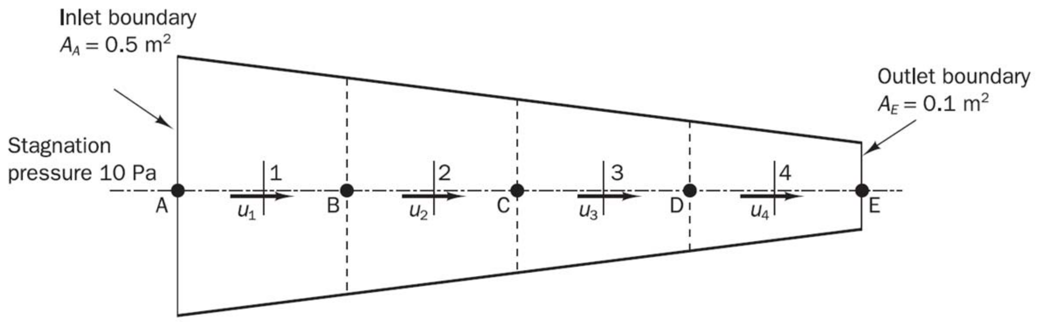
Node	*A (m²)*	*Node*	*A (m²)*
A	0.5	1	0.45
B	0.4	2	0.35
C	0.3	3	0.25
D	0.2	4	0.15
E	0.1		

$$u_1 = \dot{m}/(\rho A_1) = 1.0/(1.0 \times 0.45) = 2.22222 \text{ m/s}$$
$$u_2 = \dot{m}/(\rho A_2) = 1.0/(1.0 \times 0.35) = 2.85714 \text{ m/s}$$
$$u_3 = \dot{m}/(\rho A_3) = 1.0/(1.0 \times 0.25) = 4.00000 \text{ m/s}$$
$$u_4 = \dot{m}/(\rho A_4) = 1.0/(1.0 \times 0.15) = 6.66666 \text{ m/s}$$

$$p_A^* = p_0 = 10.0 \text{ Pa}, \; p_B^* = 7.5 \text{ Pa}, \; p_C^* = 5.0 \text{ Pa}, \; p_D^* = 2.5 \text{ Pa}$$

$$p_E = 0.0 \text{ Pa}$$





Worked examples of the SIMPLE algorithm

- Example 6.2
  - Solution
    - The governing equations for steady, one-dimensional, incompressible, frictionless equations through the planar nozzle

Mass conservation: $$\frac{d}{dx}(\rho A u) = 0$$

Momentum conservation: $$\rho u A \frac{du}{dx} = -A\frac{dp}{dx}$$

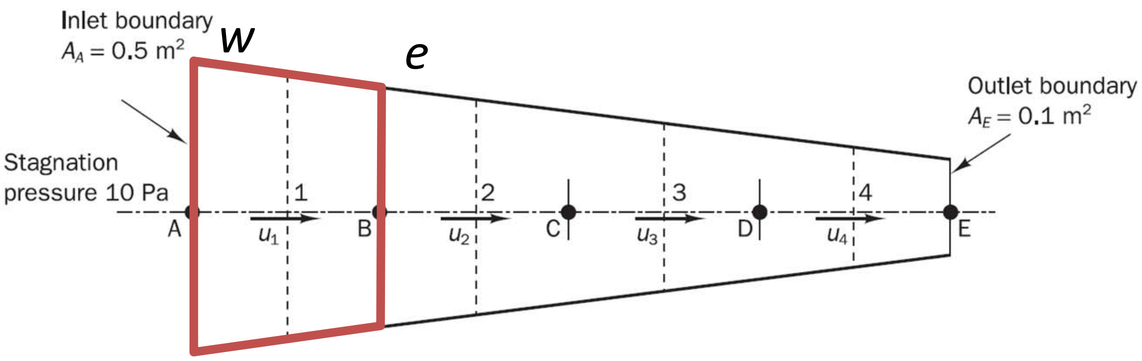
From NS equation

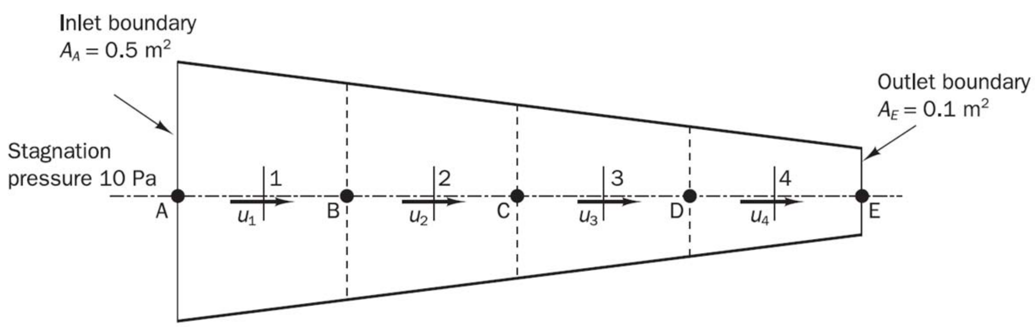
    - Discretization of u-momentum equation

$$(\rho u A)_e u_e - (\rho u A)_w u_w = \frac{\Delta p}{\Delta x}\Delta V \qquad \Delta p = p_w - p_e$$

$$F_e u_e - F_w u_w = \frac{p_w - p_e}{\Delta x}\Delta V$$

Upwind differencing scheme!





Worked examples of the SIMPLE algorithm

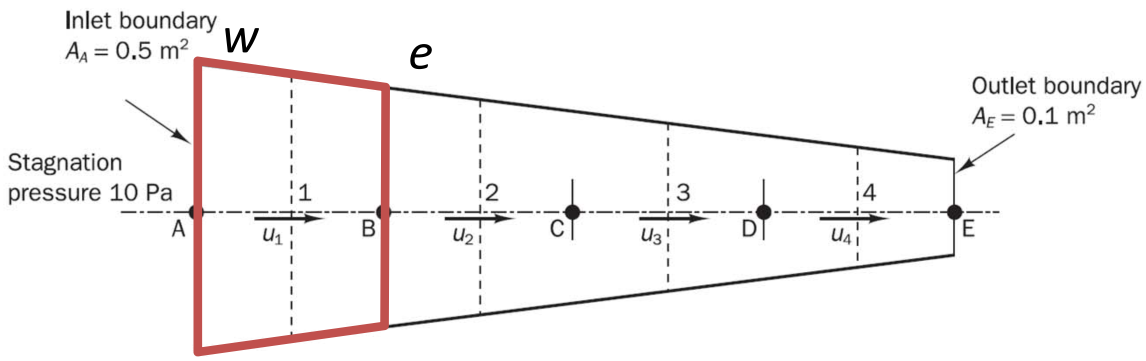
- Example 6.2
  - Solution
    - With UDS

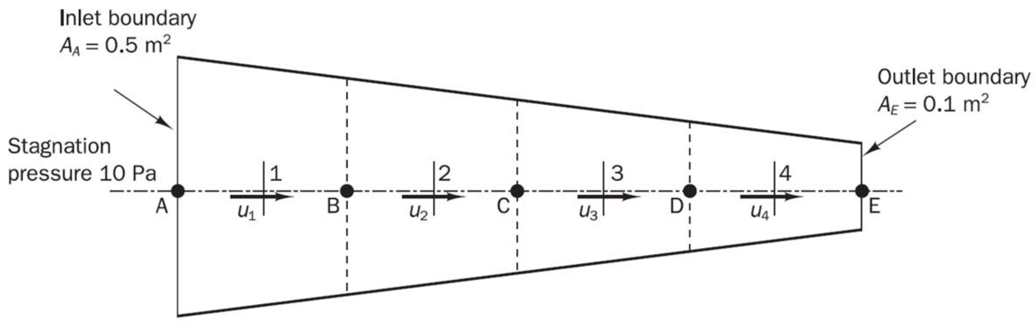
$$F_e u_e - F_w u_w = \frac{p_w - p_e}{\Delta x}\Delta V \qquad (u_e = u_P \quad and \quad u_w = u_W) \; or \; (u_e = u_E \quad and \quad u_w = u_P)$$

$$
\begin{aligned}
&-\max(-F_e,0)u_E - \min(-F_e,0)u_P - \max(F_w,0)u_W - \min(F_w,0)u_P \\
&= -\max(-F_e,0)u_E - \max(F_w,0)u_W - [\min(-F_e,0) + \min(F_w,0)]u_P \\
&= -a_E u_E - a_W u_W - [\min(-F_e,0) + \min(F_w,0)]u_P \\
&= -a_E u_E - a_W u_W + [a_E + a_W + (F_e - F_w)]u_P = S_u
\end{aligned}
$$

$$[a_E + a_W + (F_e - F_w)]u_p = a_E u_E + a_W u_W + S_u$$

$$
\begin{aligned}
a_E &= \max(-F_e,0) \\
a_W &= \max(F_w,0) \\
a_P &= a_E + a_W + (F_e - F_w)
\end{aligned}
$$





Worked examples of the SIMPLE algorithm

- Example 6.2
  - Solution
    - For intermediate velocity

$$a_P u_P^* = a_W u_W^* + a_E u_E^* + S_u$$

$$a_E = \max(-F_e, 0)$$
$$a_W = \max(F_w, 0)$$
$$a_P = a_E + a_W + (F_e - F_w)$$

– For face velocities needed for F_w and F_e

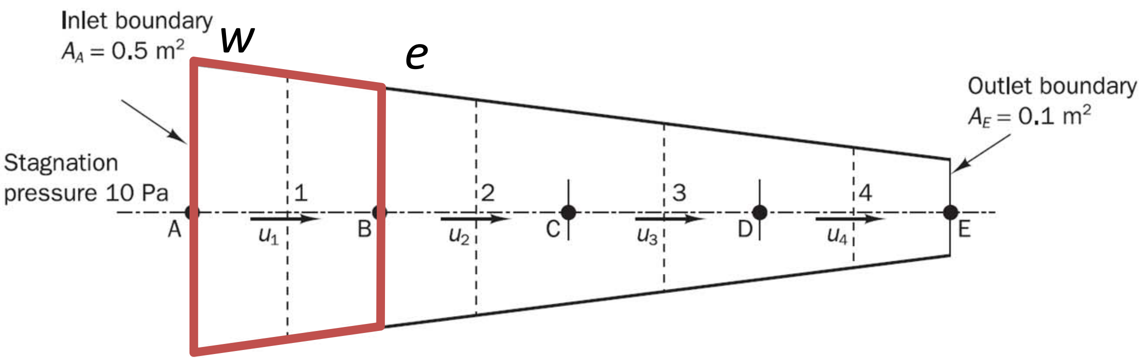
$$F = \rho u A$$

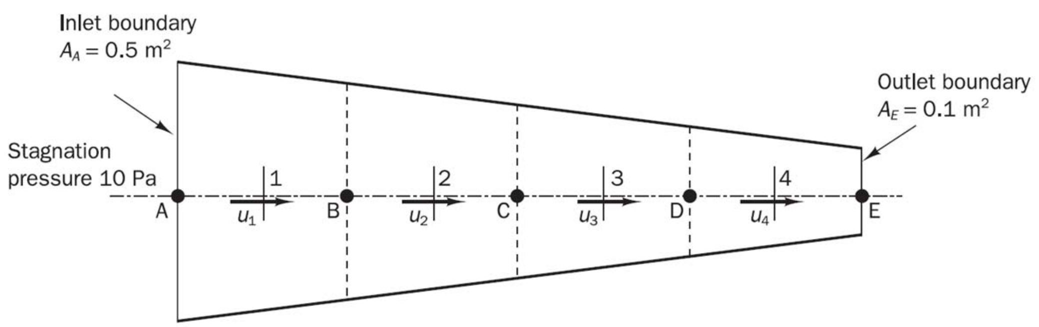
- For initial step: initial guess
- During iteration, the corrected velocity obtained after solving the pressure correction equation.

– For source term,

$$S_u = \frac{p_w^* - p_e^*}{\Delta x}\Delta V = (p_w^* - p_e^*)A_{av} = (p_w^* - p_e^*)\frac{1}{2}(A_w + A_e)$$

Crude approximation? The accuracy order is no worse than the UDS





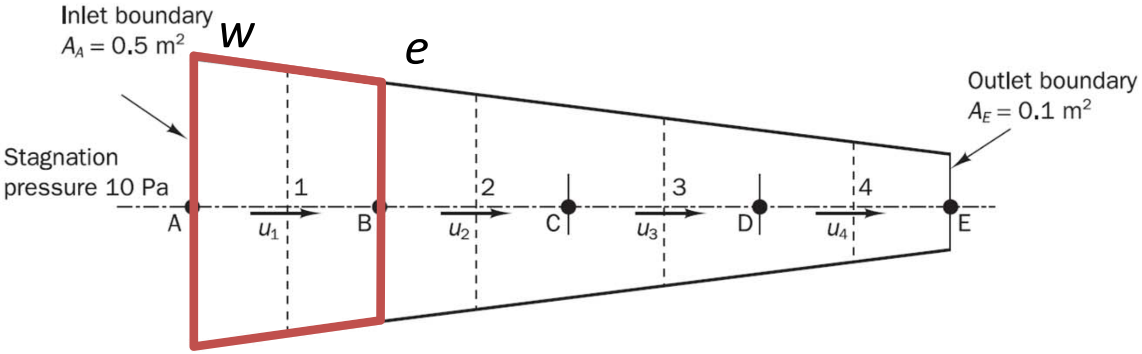
Worked examples of the SIMPLE algorithm

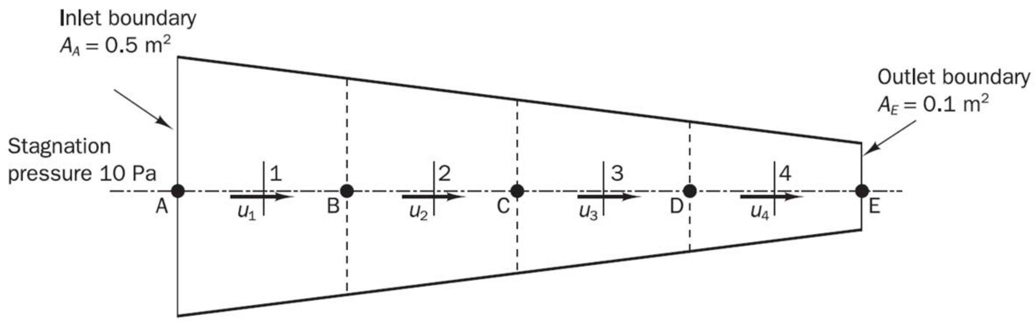
- Example 6.2
  - Solution
    - For intermediate velocity

$$a_P u_P^* = a_W u_W^* + a_E u_E^* + S_u$$

$$a_E = \max(-F_e, 0)$$
$$a_W = \max(F_w, 0)$$
$$a_P = a_E + a_W + (F_e - F_w)$$

$$a_E = 0$$
$$a_W = F_w$$
$$a_P = a_E + a_W + (F_e - F_w)$$
$$S_u = (p_w^* - p_e^*)\frac{1}{2}(A_w + A_e) = \Delta p^* A_P$$





Worked examples of the SIMPLE algorithm

- Example 6.2
  - Solution
    - Pressure correction vs. velocity correction

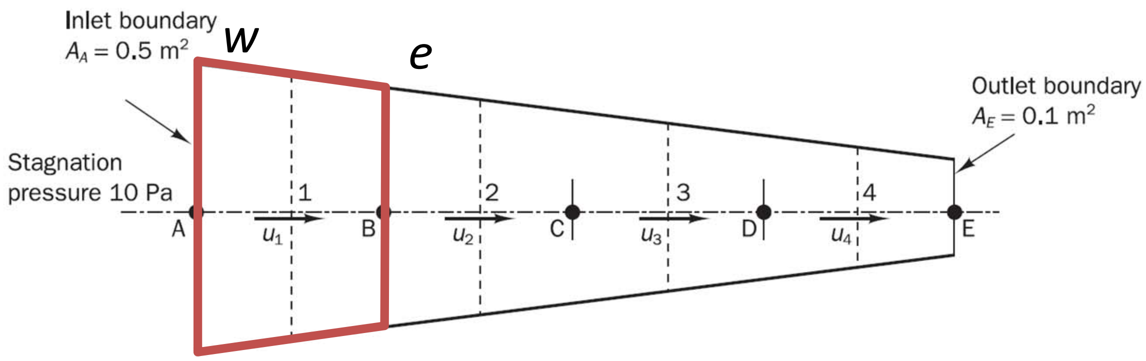
$$a_P u_P^* = a_E u_E^* + a_W u_W^* + \Delta p^* A_P$$

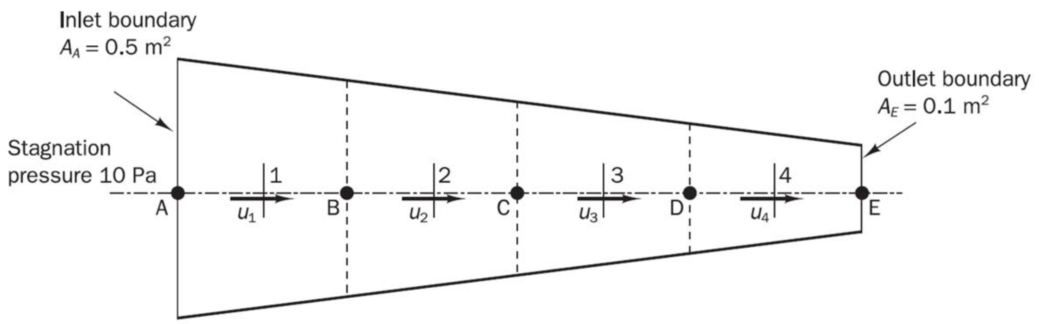
$$a_P u_P = a_E u_E + a_W u_W + \Delta p \; A_P$$

$$a_P\left(u_P - u_P^*\right) = a_E\left(u_E - u_E^*\right) + a_W\left(u_W - u_W^*\right) + \left(\Delta p \; - \Delta p^*\right)A_P$$

$$a_P\left(u_P - u_P^*\right) = a_E\left(u_E - u_E^*\right) + a_W\left(u_W - u_W^*\right) + \left[\left(p - p^*\right)_w - \left(p - p^*\right)_e\right]A_P$$

$$a_P u_P' = a_E u_E' + a_W u_W' + \left[p'_w - p'_e\right]A_P$$





Worked examples of the SIMPLE algorithm

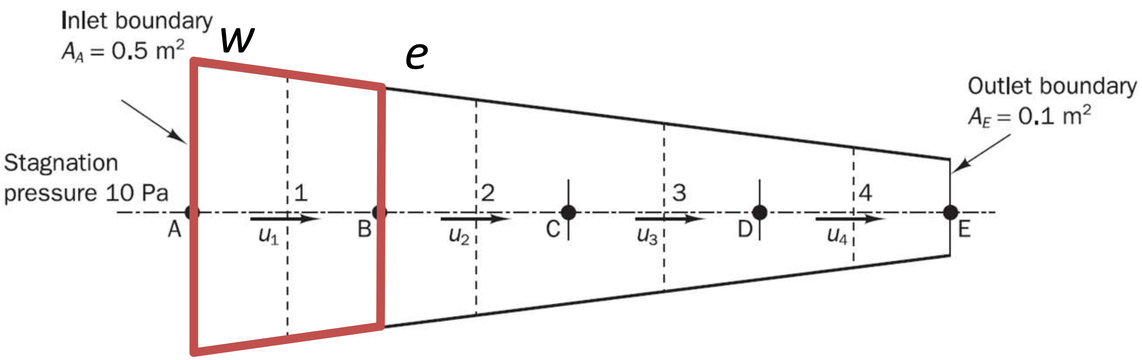
- Example 6.2
  - Solution
    - Pressure correction vs. velocity correction

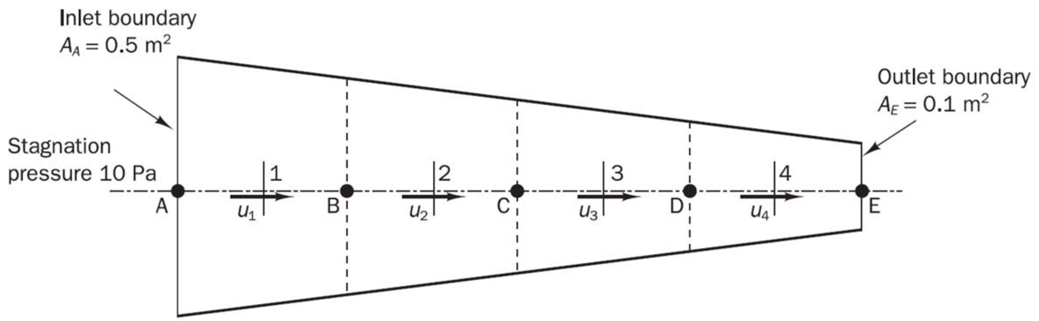
$$a_P u'_P = a_E u'_E + a_W u'_W + \left[p'_w - p'_e\right] A_P$$

$$a_P u'_P = \left[p'_w - p'_e\right] A_P$$

$$d = \frac{A_{av}}{a_P} = \frac{(A_w + A_e)}{2a_P}$$

$$u'_P = d\left[p'_w - p'_e\right]$$





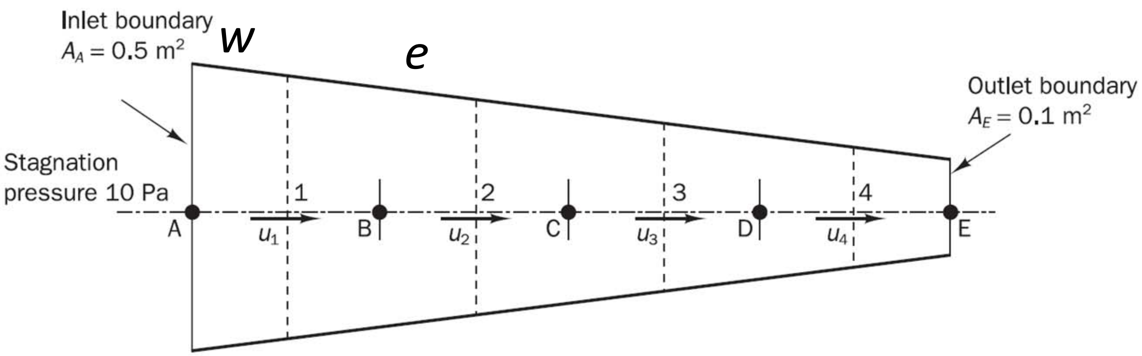
Worked examples of the SIMPLE algorithm

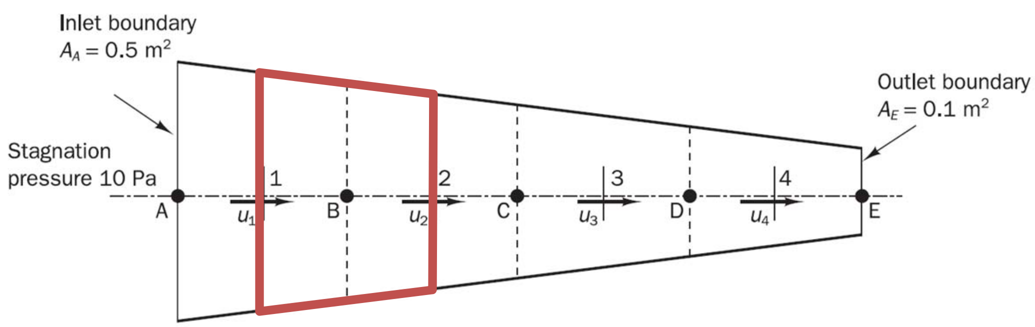
- Example 6.2
  - Solution
    - Continuity equation

$$(\rho u A)_e - (\rho u A)_w = 0$$

$$u_e = u_e^* + u_e' = u_e^* + d_e\left[p_P' - p_E'\right]$$
$$u_w = u_w^* + u_w' = u_w^* + d_w\left[p_W' - p_P'\right]$$

$$\rho u_e A_e - \rho u_w A_w = \rho A_e\left[u_e^* + d_e\left(p_P' - p_E'\right)\right] - \rho A_w\left[u_w^* + d_w\left(p_W' - p_P'\right)\right]$$
$$= \left(\rho d_e A_e + \rho d_w A_w\right)p_P' - \rho d_e A_e p_E' - \rho d_w A_w p_W' + \rho\left(u_e^* A_e - u_w^* A_w\right) = 0$$





Worked examples of the SIMPLE algorithm

- Example 6.2
  - Solution
    - Pressure correction equation

$$\left(\rho d_e A_e + \rho d_w A_w\right) p'_P - \rho d_e A_e p'_E - \rho d_w A_w p'_W + \rho\left(u^*_e A_e - u^*_w A_w\right) = 0$$

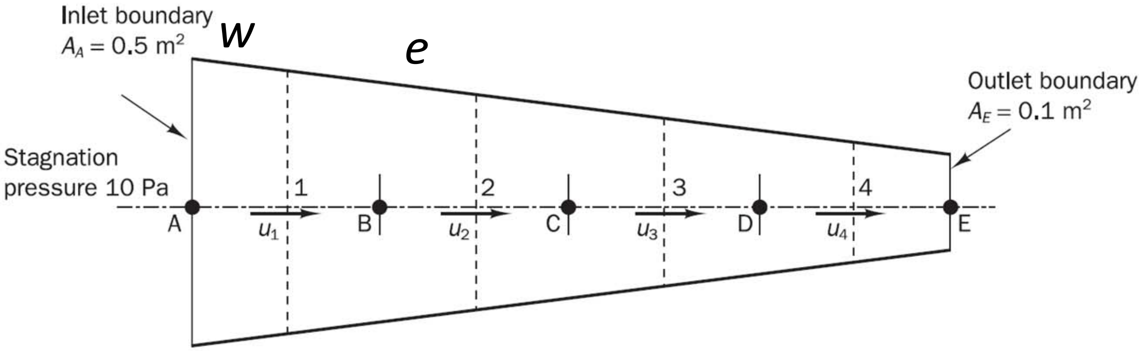
$$a_P p'_P = a_W p'_W + a_E p'_E + b'$$

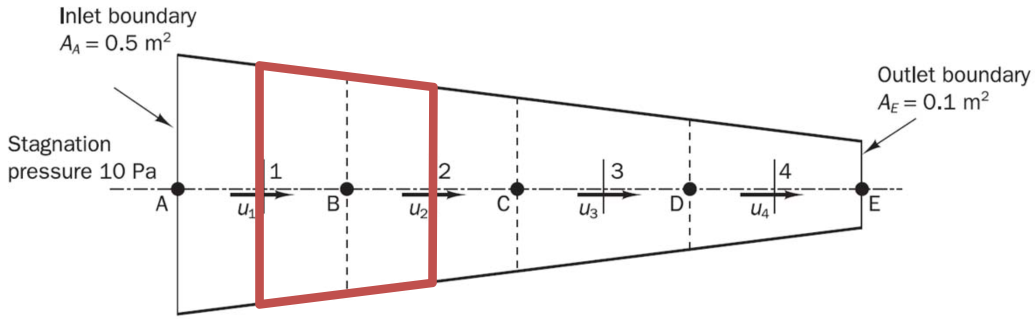
$$a_W = \rho d_w A_w$$
$$a_E = \rho d_e A_e$$
$$b' = \rho\left(u^*_w A_w - u^*_e A_e\right) = F^*_w - F^*_e$$

System pressure correction equation !

$$u' = d(p'_I - p'_{I+1})$$
$$p = p^* + p'$$
$$u = u^* + u'$$

Return to the momentum eq.





Worked examples of the SIMPLE algorithm

- Example 6.2
  - Numerical values
    - Momentum equation for intermediate velocity (node2)

$$a_P u_P^* = a_W u_W^* + a_E u_E^* + S_u$$

$$F = \rho u A$$

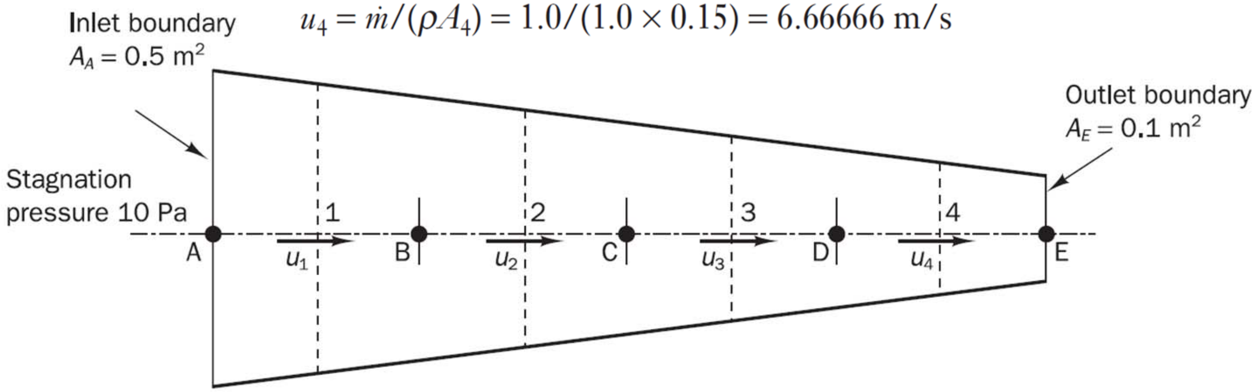
$$a_E = \max(-F_e, 0) = 0$$

$$a_W = \max(F_w, 0) = F_w$$

$$a_P = a_E + a_W + (F_e - F_w)$$

$$S_u = \Delta p^* A_P$$

$$u_1 = \dot{m}/(\rho A_1) = 1.0/(1.0 \times 0.45) = 2.22222 \text{ m/s}$$
$$u_2 = \dot{m}/(\rho A_2) = 1.0/(1.0 \times 0.35) = 2.85714 \text{ m/s}$$
$$u_3 = \dot{m}/(\rho A_3) = 1.0/(1.0 \times 0.25) = 4.00000 \text{ m/s}$$
$$u_4 = \dot{m}/(\rho A_4) = 1.0/(1.0 \times 0.15) = 6.66666 \text{ m/s}$$



- *Velocity node 2*

$$F_w = (\rho u A)_w = 1.0 \times [(u_1 + u_2)/2] \times 0.4$$
$$= 1.0 \times [(2.2222 + 2.8571)/2] \times 0.4 = 1.01587$$

$$F_e = (\rho u A)_e = 1.0 \times [(u_2 + u_3)/2] \times 0.3$$
$$= 1.0 \times [(2.8571 + 4.0)/2] \times 0.3 = 1.02857$$

$$a_W = F_w = 1.01587$$

$$a_E = 0$$

$$a_P = a_W + a_E + (F_e - F_w) = 1.01587 + 0 + (1.02857 - 1.01587)$$
$$= 1.02857$$

$$S_u = \Delta P \times A_2 = (p_B - p_C) \times A_2 = (7.5 - 5.0) \times 0.35 = 0.875$$

$$1.02857 u_2 = 1.01587 u_1 + 0.875$$

$$d_2 = A_2 / a_P = 0.35/1.02857 = 0.34027$$

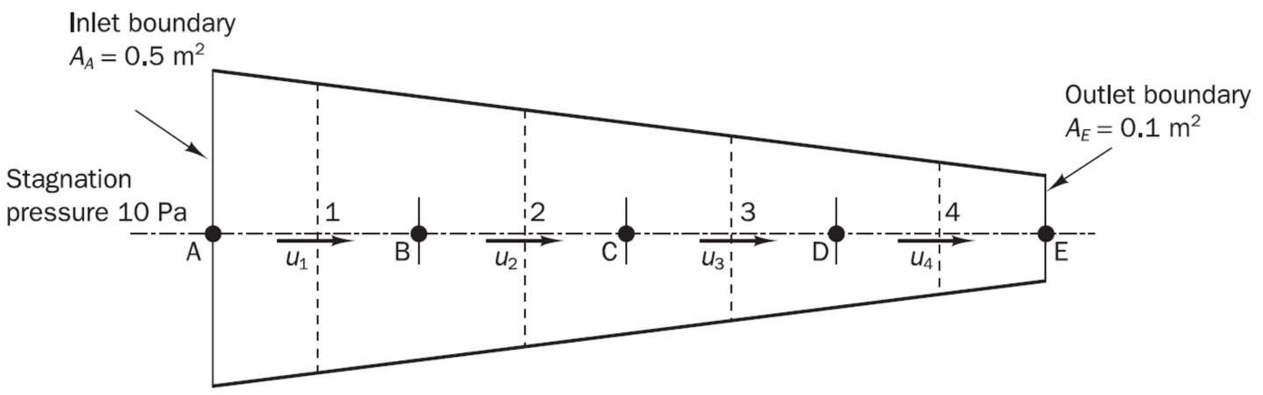
Worked examples of the SIMPLE algorithm

- Example 6.2
  - Numerical values
    - Momentum equation for intermediate velocity

Velocity node 3

$$1.06666u_3 = 1.02857u_2 + 0.625$$

$$d_3 = A_3/a_P = 0.25/1.06666 = 0.23437$$



Worked examples of the SIMPLE algorithm

- Example 6.2
  - Numerical values
    - Momentum equation for intermediate velocity

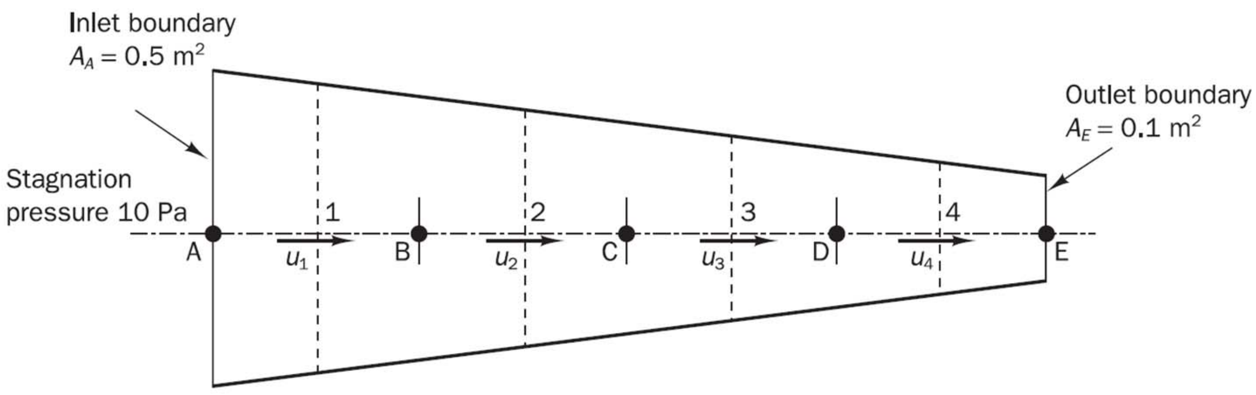
Velocity node 1

$$p_A = p_0 - \frac{1}{2}(\rho u_A^2) \qquad u_A = u_1 A_1 / A_A \qquad p_A = p_0 - \frac{1}{2}\rho u_1^2\left(\frac{A_1}{A_A}\right)^2$$

$$F_e u_e - F_w u_w = \frac{p_w - p_e}{\Delta x}\Delta V \qquad F_e u_1 - F_w u_A = (p_A - p_B) \times A_1 \qquad F_w = \rho u_A A_A = \rho u_1 A_1$$

$$F_e u_1 - F_w u_1 A_1 / A_A = [(p_0 - \tfrac{1}{2}\rho u_1^2 (A_1/A_A)^2) - p_B] \times A_1$$

$$[F_e - F_w A_1/A_A + F_w \times \tfrac{1}{2}(A_1/A_A)^2]u_1 = (p_0 - p_B)A_1$$



Worked examples of the SIMPLE algorithm

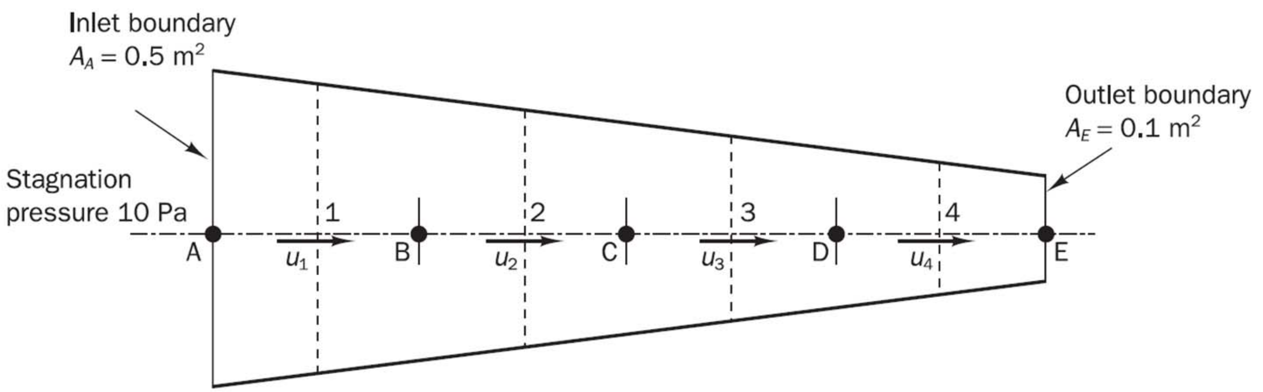
- Example 6.2
  - Numerical values
    - Momentum equation for intermediate velocity

Velocity node 1

$$[F_e - F_w A_1/A_A + F_w \times \tfrac{1}{2}(A_1/A_A)^2]u_1 = (p_0 - p_B)A_1$$

$$[F_e + F_w \times \tfrac{1}{2}(A_1/A_A)^2]u_1 = (p_0 - p_B)A_1 + F_w A_1/A_A \times u_1^{old}$$

- Place the negative contribution to coefficient on the right hand side
- Deferred correction approach
- Effective in stabilizing the iterative process if the initial velocity field is very poor.



Worked examples of the SIMPLE algorithm

- Example 6.2
  - Numerical values
    - Momentum equation for intermediate velocity

Velocity node 1

$$[F_e + F_w \times \tfrac{1}{2}(A_1/A_A)^2]u_1 = (p_0 - p_B)A_1 + F_w A_1/A_A \times u_1^{old}$$

$$u_A = u_1 A_1/A_A = 2.22222 \times 0.45/0.5 = 2.0$$

$$F_w = (\rho u A)_w = \rho u_A A_A = 1.0 \times 2.0 \times 0.5 = 1.0$$

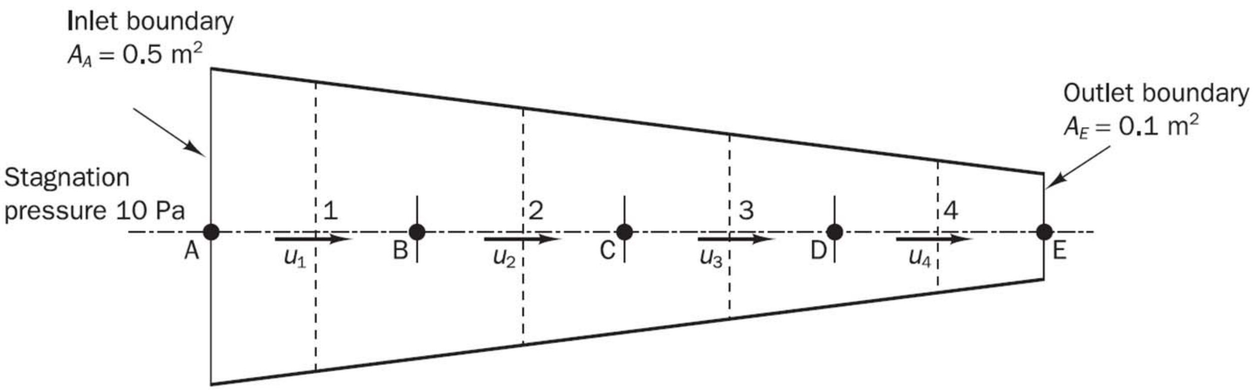
$$\begin{aligned} S_u &= (p_0 - p_B)A_1 + F_w(A_1/A_A) \times u_1^{old} \\ &= (10 - 7.5) \times 0.45 + 1.0 \times (0.45/0.5) \times 2.22222 \\ &= 3.125 \end{aligned}$$

$$\begin{aligned} F_e &= (\rho u A)_e = 1.0 \times [(u_1 + u_2)/2] \times 0.4 \\ &= 1.0 \times [(2.2222 + 2.8571)/2] \times 0.4 = 1.01587 \end{aligned}$$

$$a_W = 0$$

$$a_E = 0$$

$$\begin{aligned} a_P &= F_e + F_w \times \tfrac{1}{2}(A_1/A_A)^2 = 1.01587 + 1.0 \times 0.5 \times (0.45/0.5)^2 \\ &= 1.42087 \end{aligned}$$



$$1.42087 u_1 = 3.125$$

$$d_1 = A_1/a_P = 0.45/1.4209 = 0.31670$$

Worked examples of the SIMPLE algorithm

- Example 6.2
  - Numerical values
    - Momentum equation for intermediate velocity

From the continuity

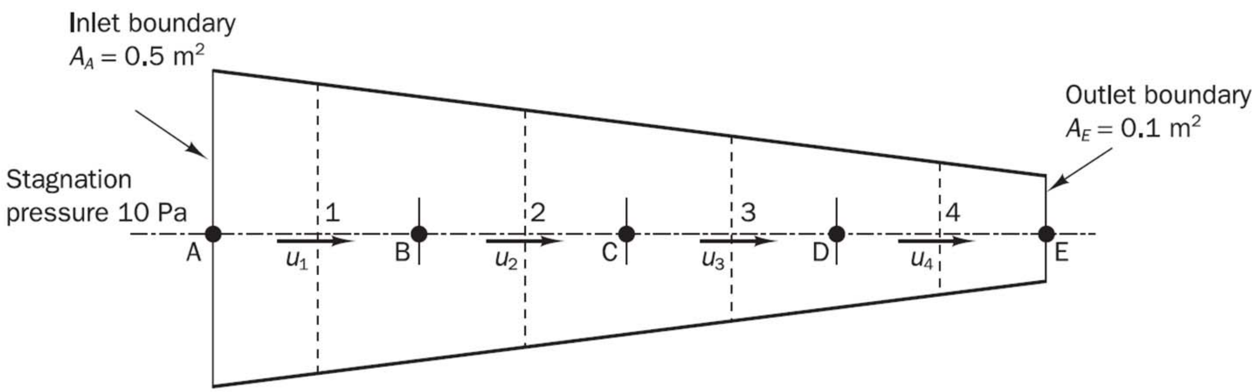
Velocity node 4

$$F_w = (\rho u A)_w = 1.0 \times [(u_3 + u_4)/2] \times 0.2 = 1.06666 \qquad F_e = (\rho u A)_4 \qquad F_e = 1.0 \text{ kg/s}$$

$$a_W = F_w = 1.06666$$
$$a_E = 0$$
$$a_P = a_W + a_E + (F_e - F_w) = 1.06666 + 0 + (1.0 - 1.06666) = 1.0$$

$$S_u = \Delta P \times A_{av} = (p_D - p_E) \times A_4 = (2.5 - 0.0) \times 0.15 = 0.375$$

$$u_1 = \dot{m}/(\rho A_1) = 1.0/(1.0 \times 0.45) = 2.22222 \text{ m/s}$$
$$u_2 = \dot{m}/(\rho A_2) = 1.0/(1.0 \times 0.35) = 2.85714 \text{ m/s}$$
$$u_3 = \dot{m}/(\rho A_3) = 1.0/(1.0 \times 0.25) = 4.00000 \text{ m/s}$$
$$u_4 = \dot{m}/(\rho A_4) = 1.0/(1.0 \times 0.15) = 6.66666 \text{ m/s}$$



$$1.0 u_4 = 1.0666 u_3 + 0.375$$

$$d_4 = A_4/a_P = 0.15/1.0 = 0.15$$

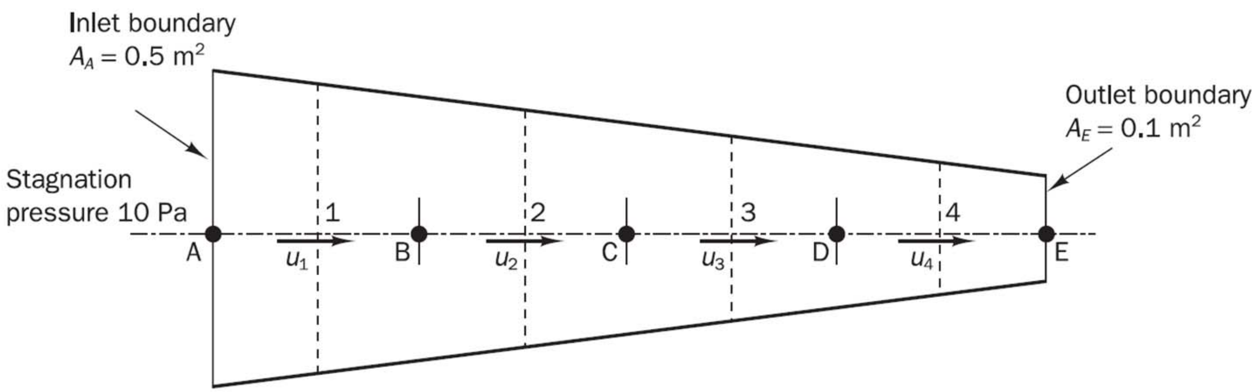
Worked examples of the SIMPLE algorithm

- Example 6.2
  - Numerical values
    - Momentum equation for intermediate velocity

$$1.42087u_1 = 3.125$$
$$1.02857u_2 = 1.01587u_1 + 0.875$$
$$1.06666u_3 = 1.02857u_2 + 0.625$$
$$1.00000u_4 = 1.06666u_3 + 0.375$$

u_1 m/s	u_2 m/s	u_3 m/s	u_4 m/s
2.19935	3.02289	3.50087	4.10926

d_1	d_2	d_3	d_4
0.31670	0.34027	0.23437	0.15000



Worked examples of the SIMPLE algorithm

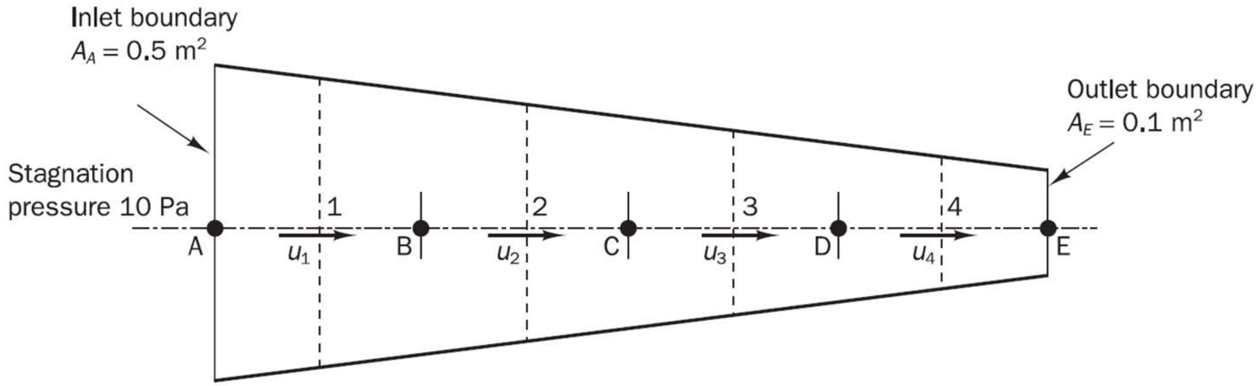
- Example 6.2
  - Numerical values
    - Pressure correction equation

$$a_P p'_P = a_W p'_W + a_E p'_E + b'$$

$$a_W = \rho d_w A_w$$

$$a_E = \rho d_e A_e$$

$$b' = \rho\left(u^*_w A_w - u^*_e A_e\right) = F^*_w - F^*_e$$



$$p'_A = 0.0$$

$$p'_E = 0.0$$

System pressure correction equation !

$$0.26161 p'_B = 0.11909 p'_C - 0.06830$$

$$0.17769 p'_C = 0.11909 p'_B + 0.058593 p'_D + 0.18279$$

$$0.081093 p'_D = 0.058593 p'_C + 0.25882$$

p'_A	p'_B	p'_C	p'_D	p'_E
0.0	1.63935	4.17461	6.20805	0.0

Worked examples of the SIMPLE algorithm

- Example 6.2
  - Numerical values
    - Pressure and velocity correction

Mass conservation is satisfied!
Momentum conservation?

Return to the momentum eq.

$$p_B = p_B^* + p_B' = 7.5 + 1.63935 = 9.13935$$

$$p_C = p_C^* + p_C' = 5.0 + 4.17461 = 9.17461$$

$$p_D = p_D^* + p_D' = 2.5 + 6.20805 = 8.70805$$

$$u_1 = u_1^* + d_1(p_A' - p_B') = 2.19935 + 0.31670 \times [0.0 - 1.63935] = 1.68015 \text{ m/s}$$

$$u_2 = u_2^* + d_2(p_B' - p_C') = 3.02289 + 0.34027 \times [1.63935 - 4.17461] = 2.16020 \text{ m/s}$$

$$u_3 = u_3^* + d_3(p_C' - p_D') = 3.50087 + 0.23437 \times [4.17461 - 6.20805] = 3.02428 \text{ m/s}$$

$$u_4 = u_4^* + d_4(p_D' - p_E') = 4.10926 + 0.15 \times [6.20805 - 0.0] = 5.04047 \text{ m/s}$$

Continuity check				
Node	1	2	3	4
$\rho u A$	0.75607	0.75607	0.75607	0.75607

Worked examples of the SIMPLE algorithm

- Example 6.2
  - Numerical values
    - The computed velocity solution at the end of an iteration cycle is not yet in balance with the computed pressure field
    - Momentum is not yet conserved.
      - The entries in the discretized momentum equations were computed on the basis of an assumed initial velocity field.
      - The velocity and the pressure were corrected.
      - We need to perform iterations until both continuity and momentum equations are satisfied.
    - Under relaxation
      - Necessary in the iteration process

Mass conservation is satisfied!
Momentum conservation?

Return to the momentum eq.

$$u_{new} = (1 - 0.8) \times u_{old} + 0.8 \times u_{calculated}$$
$$p_{new} = (1 - 0.8) \times p_{old} + 0.8 \times p_{calculated}$$

a_P, S_u and d

u_1 m/s	u_2 m/s	u_3 m/s	u_4 m/s
1.78856	2.29959	3.21942	5.36571
1.68015 m/s	2.16020 m/s	3.02428 m/s	5.04047 m/s
2.22222 m/s	2.85714 m/s	4.00000 m/s	6.66666 m/s

Worked examples of the SIMPLE algorithm

- Example 6.2
  - Numerical values
    - Iterative convergence and residuals
    - Update the coefficient with new velocity and pressure

$$\frac{a_{i,J}}{\alpha_u}u_{i,J} = \sum a_{nb}u_{nb} + (p_{I-1,J} - p_{I,J})A_{i,J} + b_{i,J} + \left[(1-\alpha_u)\frac{a_{i,J}}{\alpha_u}\right]u_{i,J}^{(n-1)}$$

$$1.20425u_1 = 1.98592 \qquad 1.143596u_1 = 1.830413?$$

Need to check!

- Momentum residual
  - The difference between the left and right hand sides of the discretized momentum equation at every velocity node.

$$u_1 = 1.78856$$

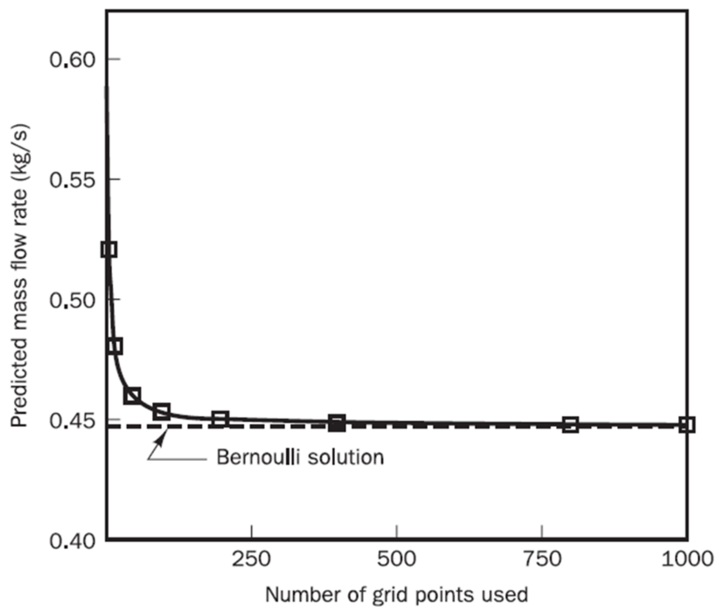
$$1.20425 \times 1.78856 - 1.98592 = 0.16795$$

$<10^{-5}$, accept the solution

Worked examples of the SIMPLE algorithm

- Example 6.2
  - Solution

Converged pressure and velocity field after 19 iterations							
Pressure (Pa)				*Velocity (m/s)*			
Node	*Computed*	*Exact*	*Error (%)*	*Node*	*Computed*	*Exact*	*Error (%)*
A	9.22569	9.60000	−3.9	1	1.38265	0.99381	39.1
B	9.00415	9.37500	−4.0	2	1.77775	1.27775	39.1
C	8.25054	8.88889	−7.2	3	2.48885	1.78885	39.1
D	6.19423	7.50000	−17.4	4	4.14808	2.98142	39.1
E	0	0	–				



With proper number of the computational grid, reasonable solution can be obtained.

Contents

The SIMPLER algorithm

- SIMPLER (SIMPLE Revised)
  - Proposed by Patankar (1980)
  - Derive a discretized equation for pressure
    - Instead of a pressure correction equation as in SIMPLE
    - Intermediate pressure field is obtained directly without the use of a correction.
  - Velocities are, however, still obtained through the velocity corrections.

Discretized momentum equation

$$a_{i,J} u^*_{i,J} = \sum a_{nb} u^*_{nb} + (p^*_{I-1,J} - p^*_{I,J}) A_{i,J} + b_{i,J}$$

$$a_{I,j} v^*_{I,j} = \sum a_{nb} v^*_{nb} + (p^*_{I,J-1} - p^*_{I,J}) A_{I,j} + b_{I,j}$$

$$u_{i,J} = \frac{\sum a_{nb} u_{nb} + b_{i,J}}{a_{i,J}} + \frac{A_{i,J}}{a_{i,J}} (p_{I-1,J} - p_{I,J})$$

$$v_{I,j} = \frac{\sum a_{nb} v_{nb} + b_{I,j}}{a_{I,j}} + \frac{A_{I,j}}{a_{I,j}} (p_{I,J-1} - p_{I,J})$$

Pseudo velocity

$$\hat{u}_{i,J} = \frac{\sum a_{nb} u_{nb} + b_{i,J}}{a_{i,J}}$$

$$\hat{v}_{I,j} = \frac{\sum a_{nb} v_{nb} + b_{I,j}}{a_{I,j}}$$

$$u_{i,J} = \hat{u}_{i,J} + d_{i,J} (p_{I-1,J} - p_{I,J})$$

$$v_{I,j} = \hat{v}_{I,j} + d_{I,j} (p_{I,J-1} - p_{I,J})$$

The SIMPLER algorithm

- Continuity equation

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

$$u_{i,J} = \hat{u}_{i,J} + d_{i,J}(p_{I-1,J} - p_{I,J})$$
$$v_{I,j} = \hat{v}_{I,j} + d_{I,j}(p_{I,J-1} - p_{I,J})$$

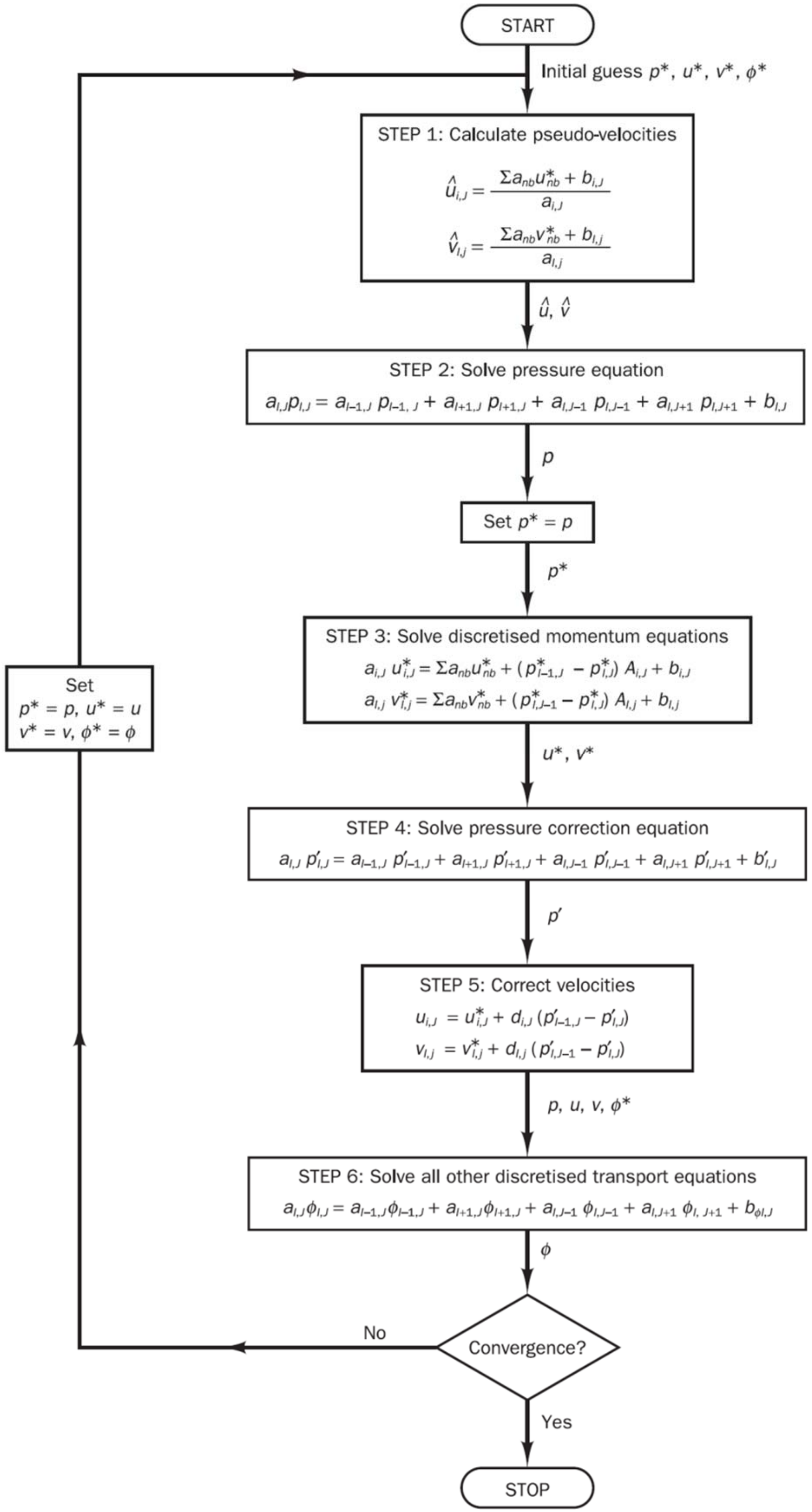
$$\begin{aligned}
&[\rho_{i+1,J} A_{i+1,J}(\hat{u}_{i+1,J} + d_{i+1,J}(p_{I,J} - p_{I+1,J})) - \rho_{i,J} A_{i,J}(\hat{u}_{i,J} \\
&\quad + d_{i,J}(p_{I-1,J} - p_{I,J}))] + [\rho_{I,j+1} A_{I,j+1}(\hat{v}_{I,j+1} + d_{I,j+1}(p_{I,J} - p_{I,J+1})) \\
&\quad - \rho_{I,j} A_{I,j}(\hat{v}_{I,j} + d_{I,j}(p_{I,J-1} - p_{I,J}))] = 0
\end{aligned}$$

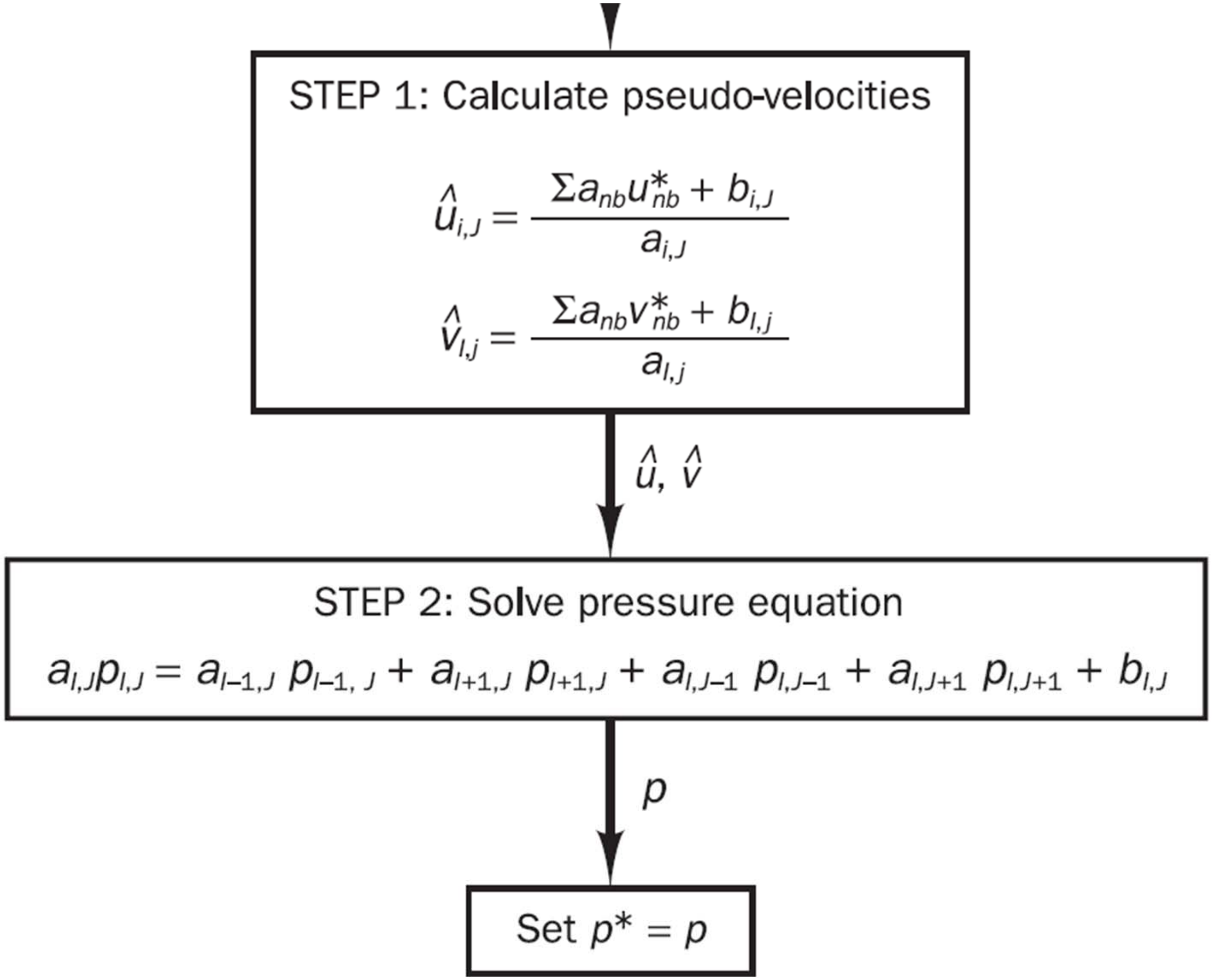
$$a_{I,J} p_{I,J} = a_{I+1,J} p_{I+1,J} + a_{I-1,J} p_{I-1,J} + a_{I,J+1} p_{I,J+1} + a_{I,J-1} p_{I,J-1} + b_{I,J}$$

$$a_{I,J} = a_{I+1,J} + a_{I-1,J} + a_{I,J+1} + a_{I,J-1}$$

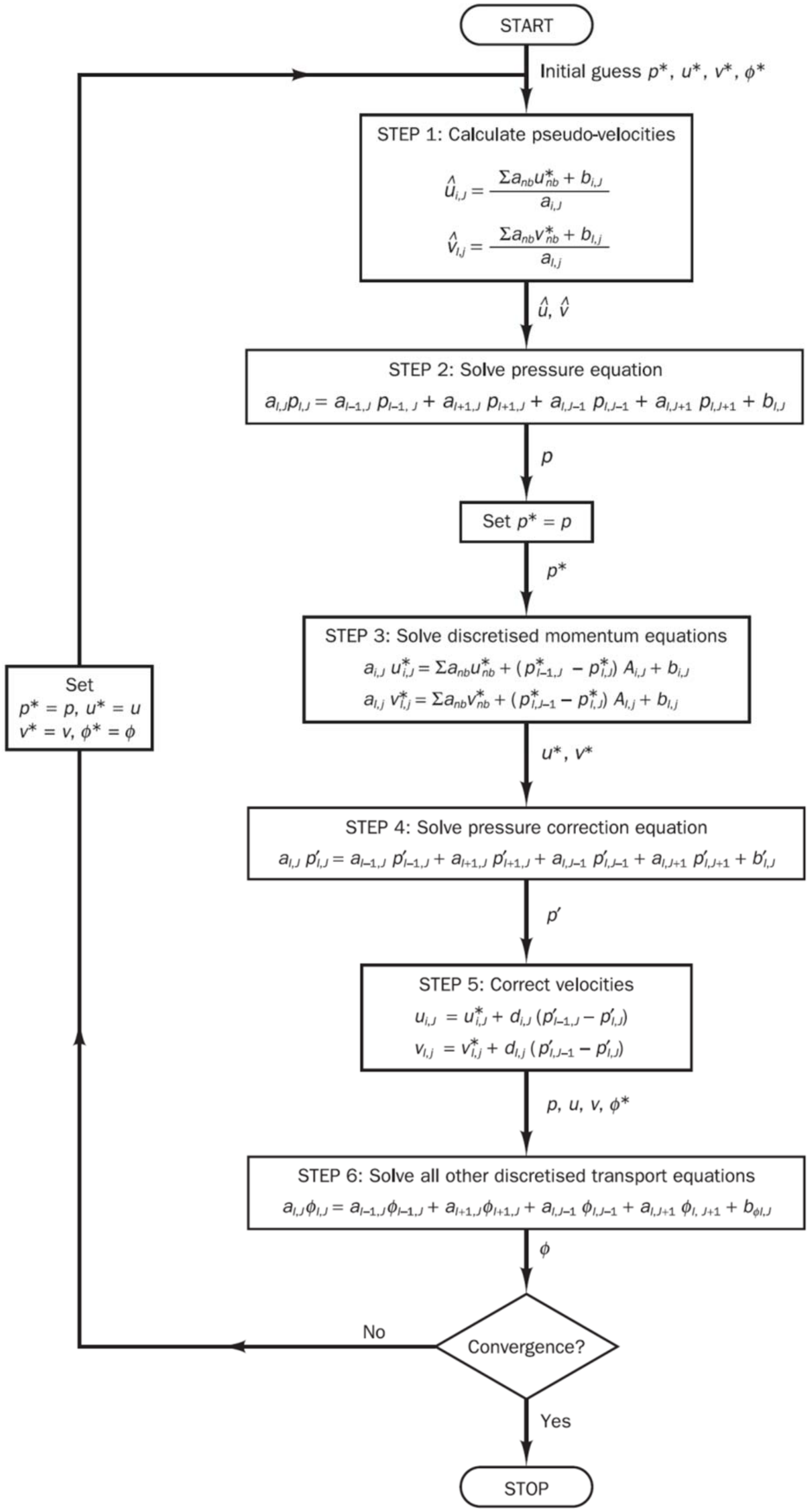
$a_{I+1,J}$	$a_{I-1,J}$	$a_{I,J+1}$	$a_{I,J-1}$	$b_{I,J}$
$(\rho d A)_{i+1,J}$	$(\rho d A)_{i,J}$	$(\rho d A)_{I,j+1}$	$(\rho d A)_{I,j}$	$(\rho \hat{u} A)_{i,J} - (\rho \hat{u} A)_{i+1,J} + (\rho \hat{v} A)_{I,j} - (\rho \hat{v} A)_{I,j+1}$

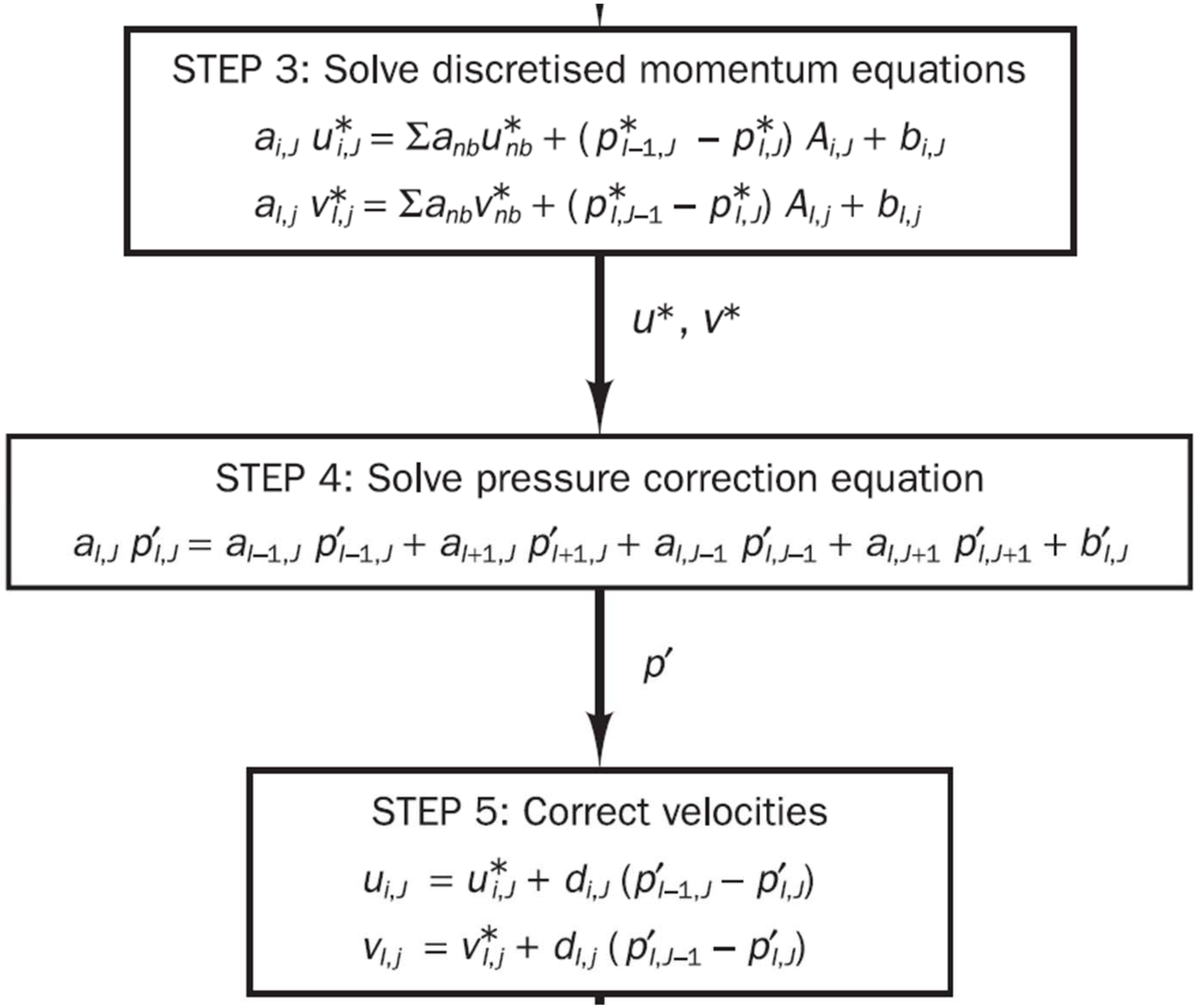
The SIMPLER algorithm





The SIMPLER algorithm





The SIMPLEC algorithm

- SIMPLEC (SIMPLE Consistent)
  - Proposed by Van Doormal and Raithby (1984)

$$u'_{i,J} = d_{i,J}(p'_{I-1,J} - p'_{I,J})$$

$$\text{where } d_{i,J} = \frac{A_{i,J}}{a_{i,J} - \sum a_{nb}}$$

$$v'_{I,j} = d_{I,j}(p'_{I,J-1} - p'_{I,J})$$

$$\text{where } d_{I,j} = \frac{A_{I,j}}{a_{I,j} - \sum a_{nb}}$$

$$a_{i,J}u'_{i,J} = \sum a_{nb}u'_{nb} + (p'_{I-1,J} - p'_{I,J})A_{i,J}$$

$$a_{I,j}v'_{I,j} = \sum a_{nb}v'_{nb} + (p'_{I,J-1} - p'_{I,J})A_{I,j}$$

$$u'_{i,J} = d_{i,J}(p'_{I-1,J} - p'_{I,J})$$

$$v'_{I,j} = d_{I,j}(p'_{I,J-1} - p'_{I,J})$$

$$\text{where } d_{i,J} = \frac{A_{i,J}}{a_{i,J}} \text{ and } d_{I,j} = \frac{A_{I,j}}{a_{I,j}}$$

The PISO algorithm

- PISO (Pressure Implicit with Splitting of Operators)
  - Proposed by Issa (1986)
  - One predictor + two corrector steps

  - Predictor step
    - Same method as the SIMPLE

  - Corrector step 1
    - The first corrector step of SIMPLE is introduced to give a velocity field (*u***, *v***) which satisfies the discretized continuity equation.
    - The resulting equations are the same as the velocity correction equations (6.21)–(6.22) of SIMPLE but, since there is a further correction step in the PISO algorithm, we use a slightly different notation

$$p^{**} = p^* + p' \qquad u^{**}_{i,J} = u^*_{i,J} + d_{i,J}(p'_{I-1,J} - p'_{I,J})$$
$$u^{**} = u^* + u' \qquad v^{**}_{I,j} = v^*_{I,j} + d_{I,j}(p'_{I,J-1} - p'_{I,J})$$
$$v^{**} = v^* + v'$$

The PISO algorithm

- PISO (Pressure Implicit with Splitting of Operators)
  - Corrector step 2
    - To enhance the SIMPLE procedure

$$a_{i,J}u_{i,J}^{**} = \sum a_{nb}u_{nb}^{*} + (p_{I-1,J}^{**} - p_{I,J}^{**})A_{i,J} + b_{i,J}$$
$$a_{I,j}v_{I,j}^{**} = \sum a_{nb}v_{nb}^{*} + (p_{I,J-1}^{**} - p_{I,J}^{**})A_{I,j} + b_{I,j}$$

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

$$a_{i,J}u_{i,J}^{***} = \sum a_{nb}u_{nb}^{**} + (p_{I-1,J}^{***} - p_{I,J}^{***})A_{i,J} + b_{i,J}$$
$$a_{I,j}v_{I,j}^{***} = \sum a_{nb}v_{nb}^{**} + (p_{I,J-1}^{***} - p_{I,J}^{***})A_{I,j} + b_{I,j}$$

$$u_{i,J}^{***} = u_{i,J}^{**} + \frac{\sum a_{nb}(u_{nb}^{**} - u_{nb}^{*})}{a_{i,J}} + d_{i,J}(p''_{I-1,J} - p''_{I,J})$$

$$v_{I,J}^{***} = v_{I,j}^{**} + \frac{\sum a_{nb}(v_{nb}^{**} - v_{nb}^{*})}{a_{I,j}} + d_{I,j}(p''_{I,J-1} - p''_{I,J})$$

$$p^{***} = p^{**} + p''$$

The PISO algorithm

- PISO (Pressure Implicit with Splitting of Operators)
  - Corrector step 2
    - To enhance the SIMPLE procedure

$$a_{I,J}p''_{I,J} = a_{I+1,J}p''_{I+1,J} + a_{I-1,J}p''_{I-1,J} + a_{I,J+1}p''_{I,J+1} + a_{I,J-1}p''_{I,J-1} + b''_{I,J}$$

$a_{I+1,J}$	$a_{I-1,J}$	$a_{I,J+1}$	$a_{I,J-1}$	$b''_{I,J}$
$(\rho dA)_{i+1,J}$	$(\rho dA)_{i,J}$	$(\rho dA)_{I,j+1}$	$(\rho dA)_{I,j}$	$\left[\left(\frac{\rho A}{a}\right)_{i,J}\sum a_{nb}(u^{**}_{nb} - u^{*}_{nb}) - \left(\frac{\rho A}{a}\right)_{i+1,J}\sum a_{nb}(u^{**}_{nb} - u^{*}_{nb}) + \left(\frac{\rho A}{a}\right)_{I,j}\sum a_{nb}(v^{**}_{nb} - v^{*}_{nb}) - \left(\frac{\rho A}{a}\right)_{I,j+1}\sum a_{nb}(v^{**}_{nb} - v^{*}_{nb})\right]$

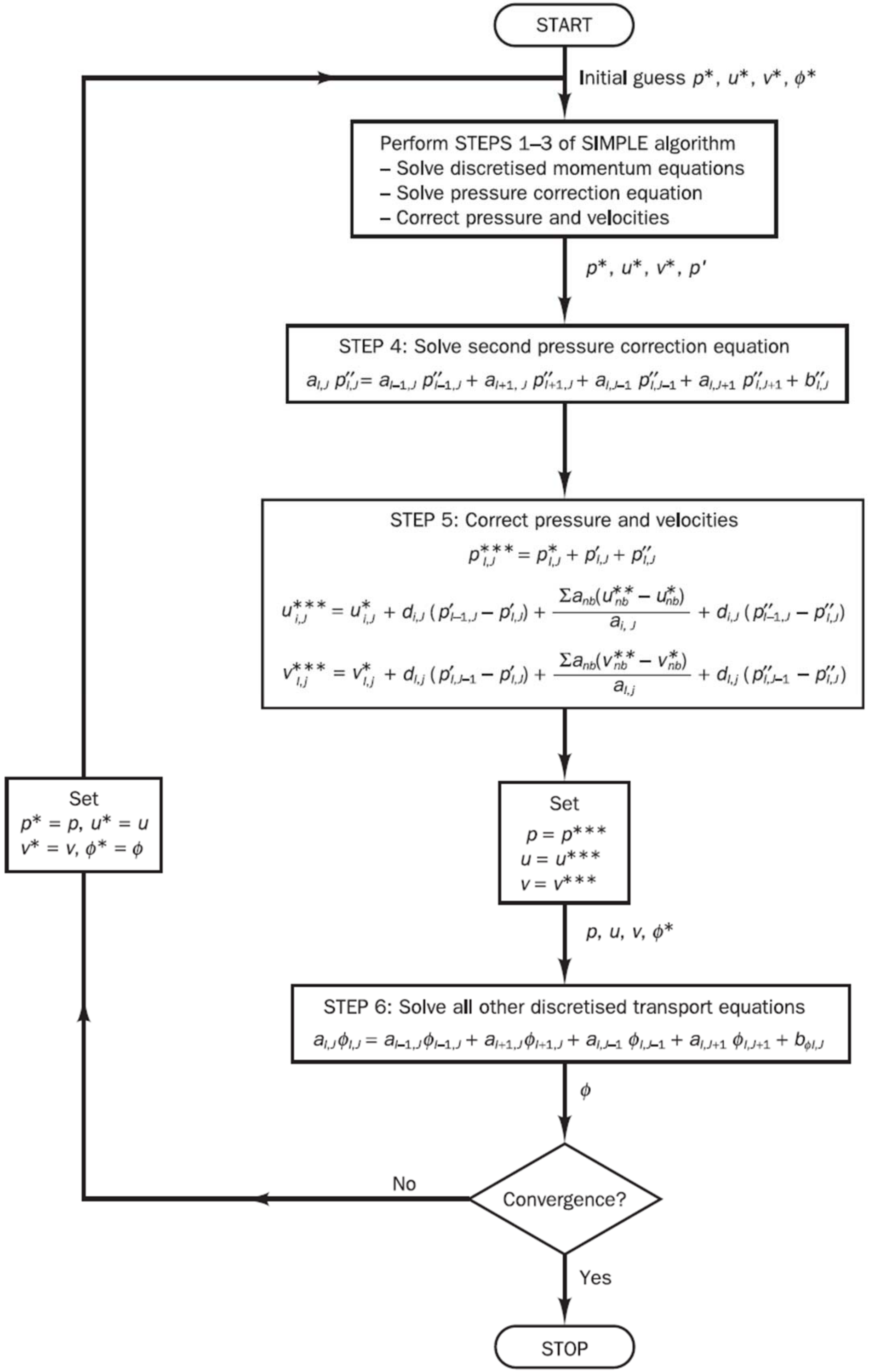
$$[(\rho A u^{**})_{i,J} - (\rho A u^{**})_{i+1,J} + (\rho A v^{**})_{I,j} - (\rho A v^{**})_{I,j+1}] = 0$$

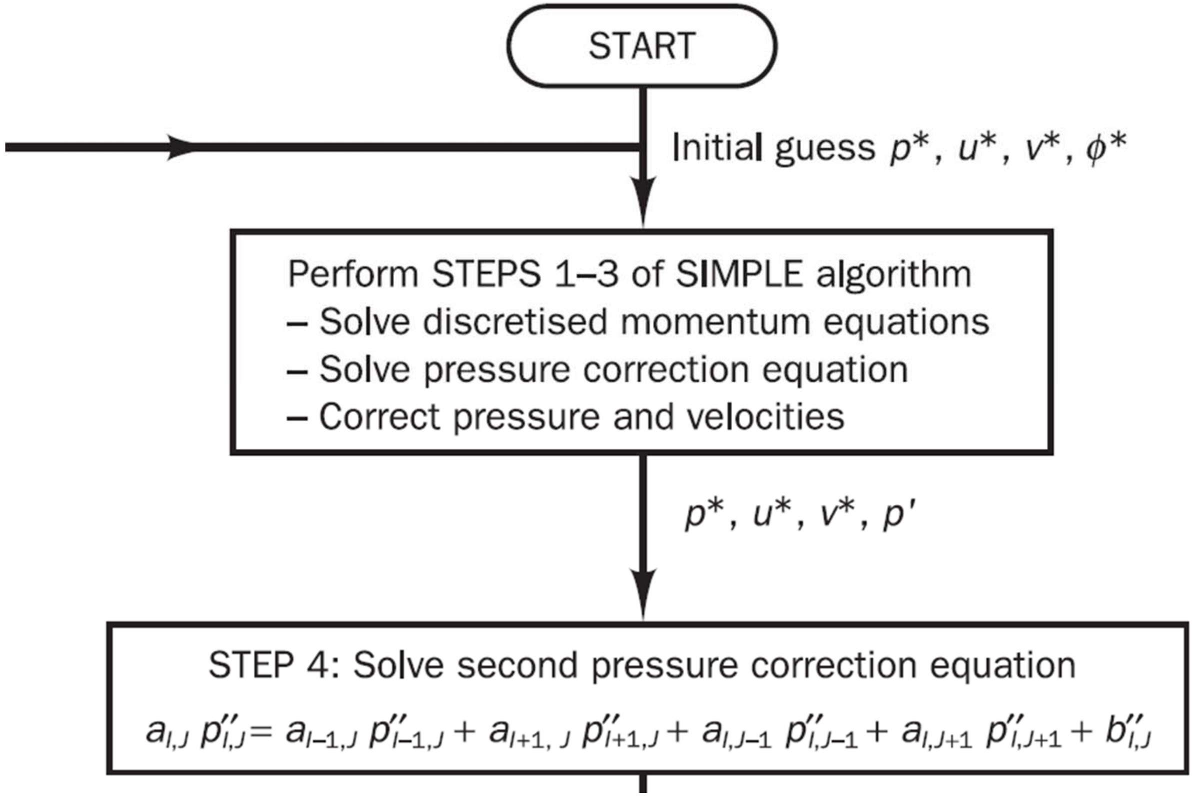
The PISO algorithm

- PISO (Pressure Implicit with Splitting of Operators)
  - Corrector step 2
    - Twice-corrected pressure field is obtained from

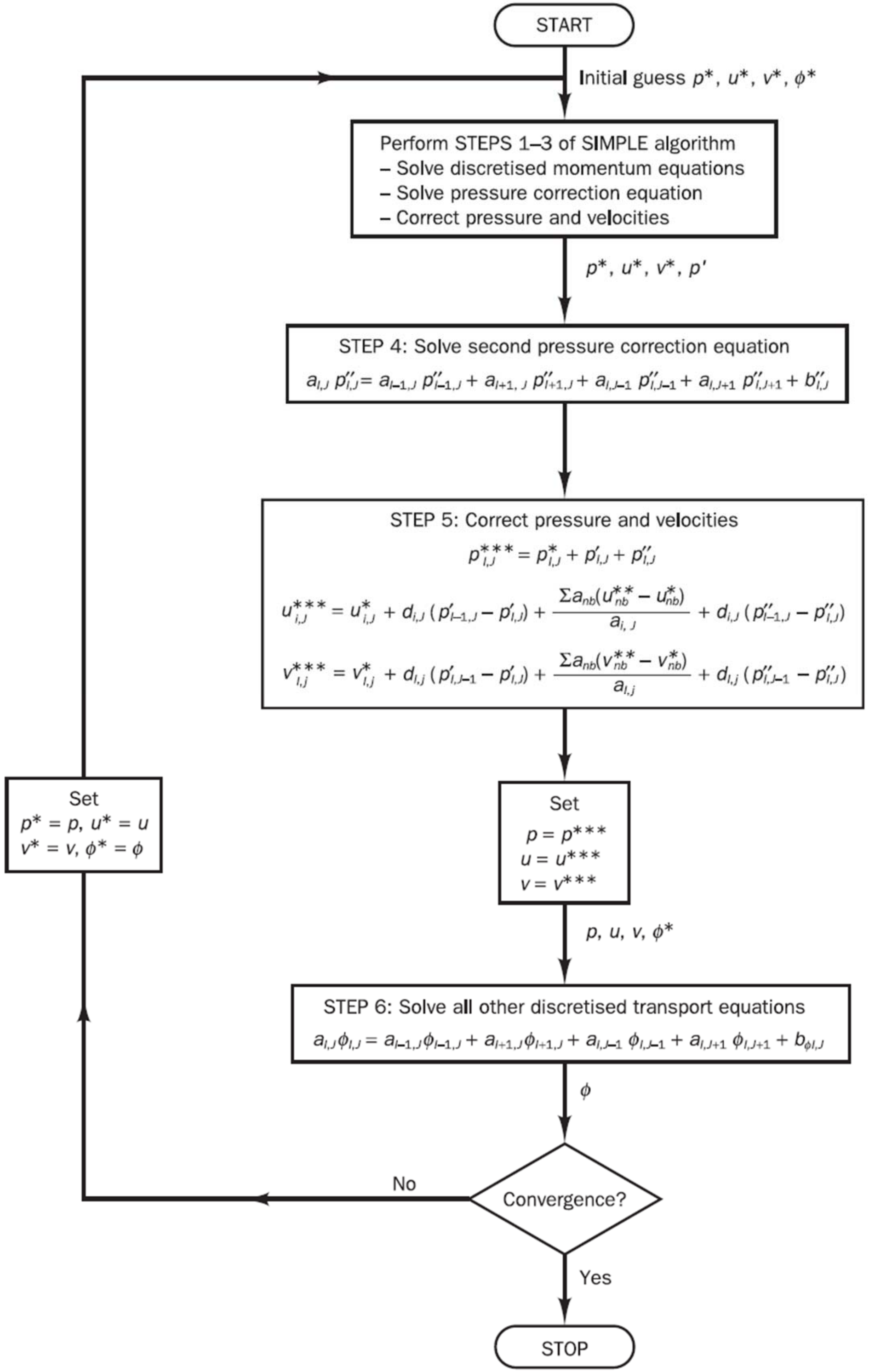
$$p^{***} = p^{**} + p'' = p^{*} + p' + p''$$

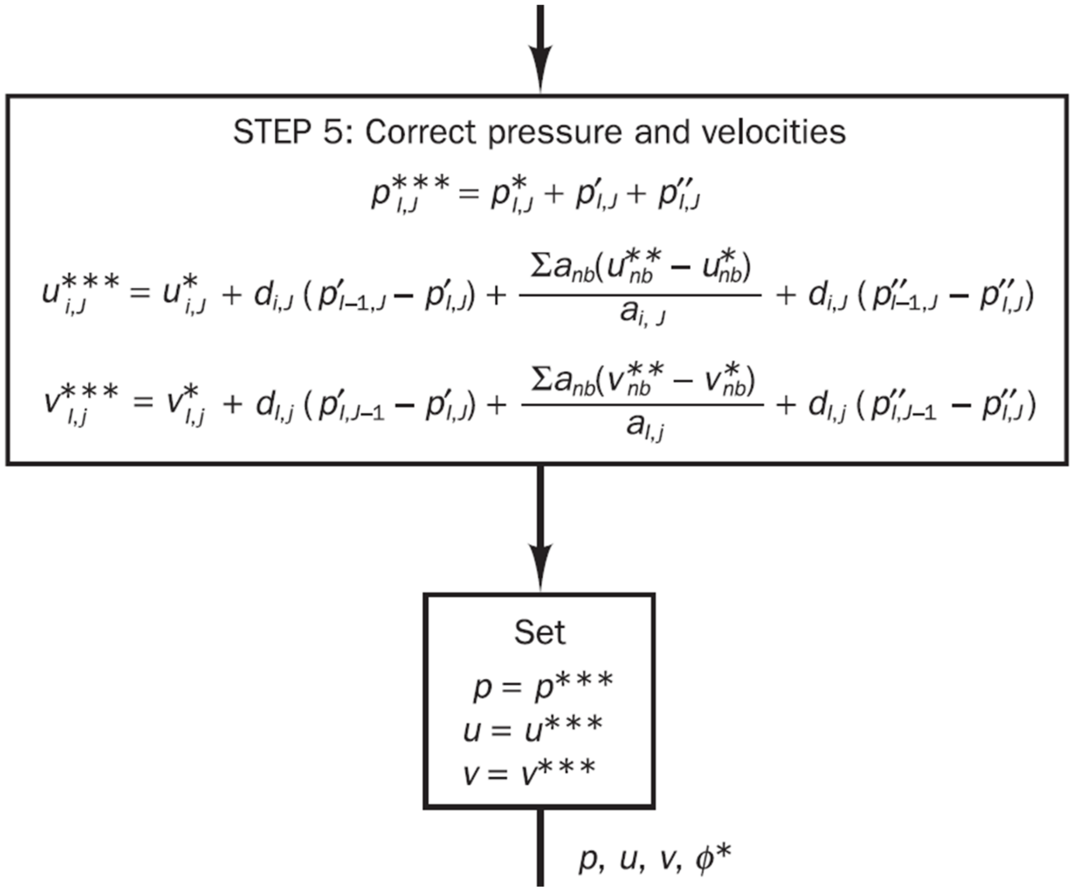
The PISO algorithm



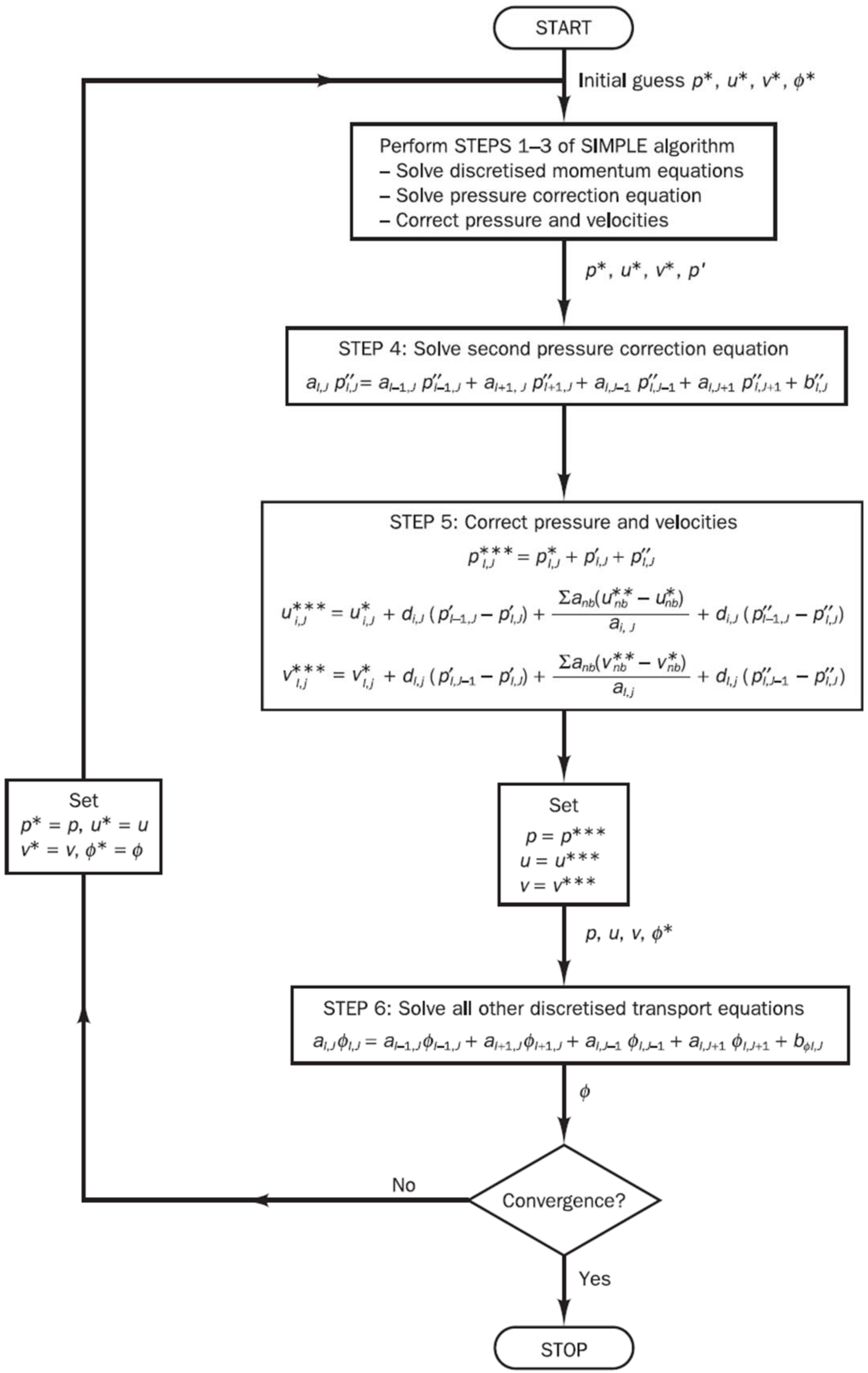


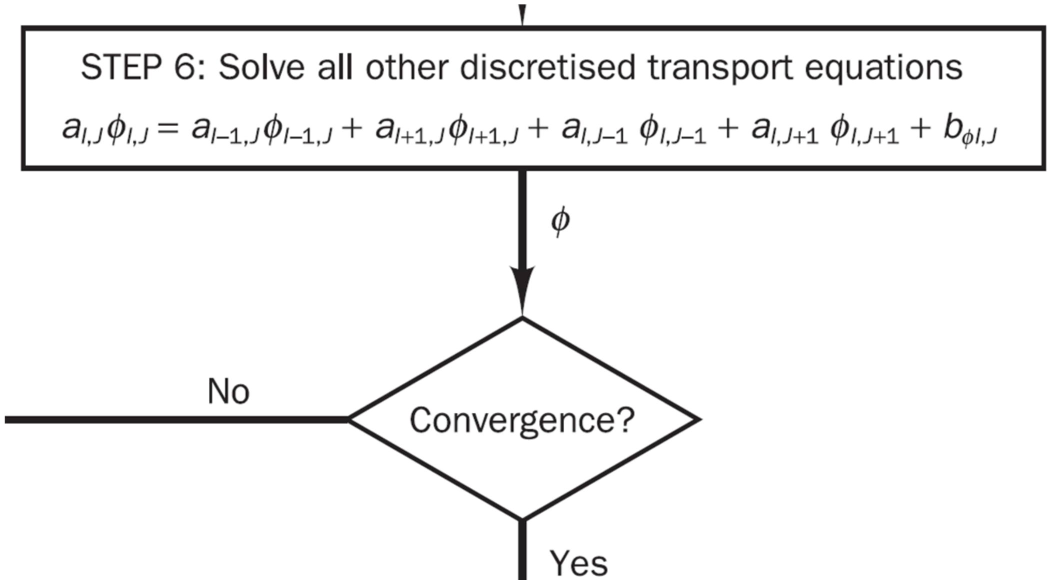
The PISO algorithm





The PISO algorithm





General Comments

- SIMPLE
  - Relatively straightforward and has been successfully implemented in numerous CFD procedures.
  - The other variations of SIMPLE can produce savings in computational effort due to improved convergence.
  - In SIMPLE, the pressure correction p' is satisfactory for correcting velocities but not so good for correcting pressure.
- SIMPLER
  - Hence the improved procedure SIMPLER uses the pressure corrections to obtain velocity corrections only.
  - A separate, more effective, pressure equation is solved to yield the correct pressure field. Since no terms are omitted to derive the discretized pressure equation in SIMPLER, the resulting pressure field corresponds to the velocity field.
  - Therefore, in SIMPLER the application of the correct velocity field results in the correct pressure field, whereas it does not in SIMPLE.
  - Consequently, the method is highly effective in calculating the pressure field correctly. This has significant advantages when solving the momentum equations. Although the number of calculations involved in SIMPLER is about 30% larger than that for SIMPLE, the fast convergence rate reportedly reduces the computer time by 30–50% (Anderson *et al.*, 1984).

General Comments

- SIMPLEC and PISO
  - have proved to be as efficient as SIMPLER in certain types of flows but it is not clear whether it can be categorically stated that they are better than SIMPLER.
  - Comparisons have shown that the performance of each algorithm depends on the flow conditions, the degree of coupling between the momentum equation and scalar equations, the amount of under-relaxation used, and sometimes even on the details of the numerical technique used for solving the algebraic equations.
  - A comprehensive comparison of PISO, SIMPLER and SIMPLEC methods for a variety of steady flow problems by Jang *et al.*(1986) showed that, for problems in which momentum equations are not coupled to a scalar variable, PISO showed robust convergence behavior and required less computational effort than SIMPLER and SIMPLEC.
  - It was also observed that when the scalar variables were closely linked to velocities, PISO had no significant advantage over the other methods.
  - Iterative methods using SIMPLER and SIMPLEC have robust convergence characteristics in strongly coupled problems, and it could not be ascertained which of SIMPLER or SIMPLEC was superior.