# 2.29 Numerical Fluid Mechanics
# Spring 2015 – **Lecture 18**

**REVIEW Lecture 17:**

- End of Finite Volume Methods – Cartesian grids
  - Higher order (interpolation) schemes
- Solution of the Navier-Stokes Equations

$$\frac{\partial \rho \vec{v}}{\partial t} + \nabla.(\rho \vec{v}\ \vec{v}) = -\nabla p + \mu \nabla^2 \vec{v} + \rho \vec{g}$$
$$\nabla.\vec{v} = 0$$

  - Discretization of the convective and viscous terms
  - Discretization of the pressure term

$$\tilde{p} = p - \rho \mathbf{g.r} + \mu \frac{2}{3} \nabla.\mathbf{u} \quad (p\ \vec{e}_i - \rho g_i x_i \vec{e}_i + \frac{2}{3}\mu \frac{\partial u_j}{\partial x_j} \vec{e}_i)$$

$$\int_S -\tilde{p}\ \vec{e}_i.\vec{n} dS$$

  - Conservation principles
    - Momentum and Mass
    - Energy

$$\frac{\partial}{\partial t}\int_{CV} \rho \frac{\|\vec{v}\|^2}{2} dV = -\int_{CS} \rho \frac{\|\vec{v}\|^2}{2} (\vec{v}.\vec{n}) dA - \int_{CS} p\ \vec{v}.\vec{n}\ dA + \int_{CS} (\vec{\vec{\varepsilon}}.\vec{v}).\vec{n}\ dA + \int_{CV} \left(-\vec{\vec{\varepsilon}} : \nabla \vec{v} + p \nabla.\vec{v} + \rho \vec{g}.\vec{v}\right) dV$$

  - Choice of Variable Arrangement on the Grid
    - Collocated and Staggered
  - Calculation of the Pressure

# TODAY (Lecture 18): Numerical Methods for the Navier-Stokes Equations

- Solution of the Navier-Stokes Equations
  - Discretization of the convective and viscous terms
  - Discretization of the pressure term
  - Conservation principles
  - Choice of Variable Arrangement on the Grid
  - Calculation of the Pressure
  - Pressure Correction Methods
    - A Simple Explicit Scheme
    - A Simple Implicit Scheme
      - Nonlinear solvers, Linearized solvers and ADI solvers
    - Implicit Pressure Correction Schemes for steady problems
      - Outer and Inner iterations
    - Projection Methods
      - Non-Incremental and Incremental Schemes
    - Fractional Step Methods:
      - Example using Crank-Nicholson

# References and Reading Assignments

- Chapter 7 on "Incompressible Navier-Stokes equations" of "J. H. Ferziger and M. Peric, Computational Methods for Fluid Dynamics. Springer, NY, 3rd edition, 2002"
- Chapter 11 on "Incompressible Navier-Stokes Equations" of T. Cebeci, J. P. Shao, F. Kafyeke and E. Laurendeau, *Computational Fluid Dynamics for Engineers*. Springer, 2005.
- Chapter 17 on "Incompressible Viscous Flows" of Fletcher, *Computational Techniques for Fluid Dynamics*. Springer, 2003.

# Calculation of the Pressure

- The Navier-Stokes equations do not have an independent equation for pressure
  - But the pressure gradient contributes to each of the three momentum equations
  - For incompressible fluids, mass conservation becomes a kinematic constraint on the velocity field: we then have no dynamic equations for both density and pressure
  - For compressible fluids, mass conservation is a dynamic equation for density
    - Pressure can then be computed from density using an equation of state
    - For incompressible flows (or low Mach numbers), density is not a state variable, hence can't be solved for
- For incompressible flows:
  - Momentum equations lead to the velocities $\Rightarrow$
  - Continuity equation should lead to the pressure, but it does not contain pressure! How can $p$ be estimated?

# Calculation of the Pressure

- Navier-Stokes, incompressible:

$$\frac{\partial \rho \vec{v}}{\partial t} + \nabla.(\rho \vec{v}\, \vec{v}) = -\nabla p + \mu \nabla^2 \vec{v} + \rho \vec{g}$$

$$\nabla.\vec{v} = 0$$

- Combine the two conservation eqs. to obtain an equation for $p$
  - Since the cons. of mass has a divergence form, take the divergence of the momentum equation, using cons. of mass:
    - For constant viscosity and density:

$$\nabla.\nabla p = \nabla^2 p = -\nabla.\frac{\partial \rho \vec{v}}{\partial t} - \nabla.\left(\nabla.(\rho \vec{v}\, \vec{v})\right) + \nabla.\left(\mu \nabla^2 \vec{v}\right) + \nabla.\left(\rho \vec{g}\right) = -\nabla.\left(\nabla.(\rho \vec{v}\, \vec{v})\right)$$

  - This pressure equation is elliptic (Poisson eqn. once velocity is known)
    - It can be solved by methods we have seen earlier for elliptic equations

$$\frac{\partial}{\partial x_i}\left(\frac{\partial p}{\partial x_i}\right) = -\frac{\partial}{\partial x_i}\left(\frac{\partial \left(\rho u_i u_j\right)}{\partial x_j}\right)$$

  - Important Notes
    - RHS: Terms inside divergence (derivatives of momentum terms) must be approximated in a form consistent with that of momentum eqns. However, divergence is that of cons. of mass.
    - Laplacian operator comes from i) divergence of cons. of mass and ii) gradient in momentum eqns.: consistency must be maintained, i.e. divergence and gradient discrete operators in Laplacian should be those of the cons. of mass and of the momentum eqns., respectively
    - Best to derive pressure equation from discretized momentum/continuity equations

# Pressure-correction Methods

- First solve the momentum equations to obtain the velocity field for a known pressure
- Then solve the Poisson equation to obtain an updated/corrected pressure field
- Another way: modify the continuity equation so that it becomes hyperbolic (even though it is elliptic)
  - Artificial Compressibility Methods
- Notes:
  - The general pressure-correction method is independent of the discretization chosen for the spatial derivatives $\Rightarrow$ in theory any discretization can be used
  - We keep density in the equations (flows are assumed incompressible, but small density variations are considered)

# A Simple Explicit Time Advancing Scheme

- Simple method to illustrate how the numerical Poisson equation for the pressure is constructed and the role it plays in enforcing continuity
- Specifics of spatial derivative scheme not important, hence, we look at the equation discretized in space, but not in time.
  - Use $\frac{\delta}{\delta x_i}$ to denote discrete spatial derivatives.

  This gives: $\frac{\partial \rho u_i}{\partial t} = -\frac{\partial(\rho u_i u_j)}{\partial x_j} - \frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \Rightarrow \frac{\partial \rho u_i}{\partial t} = -\frac{\delta(\rho u_i u_j)}{\delta x_j} - \frac{\delta p}{\delta x_i} + \frac{\delta \tau_{ij}}{\delta x_j} = H_i - \frac{\delta p}{\delta x_i}$

  Note: $p = p_{real} - \rho g_i x_i$
  - Simplest approach: Forward Euler for time integration, which gives:

$$\left(\rho u_i\right)^{n+1} - \left(\rho u_i\right)^n = \Delta t \left( H_i^{\,n} - \frac{\delta p^{\,n}}{\delta x_i} \right)$$

    - In general, the new velocity field we obtain at time $n+1$ does not satisfy the discrete continuity equation:

$$\frac{\delta\left(\rho u_i\right)^{n+1}}{\delta x_i} = 0$$

# A Simple Explicit Time Advancing Scheme

- How can we enforce continuity at $n+1$?
- Take the discrete numerical divergence of the NS eqs.:

$$(\rho u_i)^{n+1} - (\rho u_i)^n = \Delta t \left( H_i^n - \frac{\delta p^n}{\delta x_i} \right) \quad \Rightarrow \quad \boxed{\frac{\delta (\rho u_i)^{n+1}}{\delta x_i} - \frac{\delta (\rho u_i)^n}{\delta x_i} = \Delta t \left[ \frac{\delta}{\delta x_i} \left( H_i^n - \frac{\delta p^n}{\delta x_i} \right) \right]}$$

  – The first term is the divergence of the new velocity field, which we want to be zero, so we set it to zero.

  – Second term is zero if continuity was enforced at time step $n$

  – Third term can be zero or not, but the two above conditions set it to zero

- All together, we obtain:

$$\boxed{\frac{\delta}{\delta x_i} \left( \frac{\delta p^n}{\delta x_i} \right) = \frac{\delta H_i^n}{\delta x_i}}$$

  – Note that this includes the divergence operator from the continuity eqn. (outside) and the pressure gradient from the momentum equation (inside)

  – Pressure gradient could be explicit ($n$) or implicit ($n+1$)

# A Simple Explicit Time Advancing Scheme: Summary of the Algorithm

- Start with velocity at time $t_n$ which is divergence free
- Compute RHS of pressure equation at time $t_n$
- Solve the Poisson equation for the pressure at time $t_n$
- Compute the velocity field at the new time step using the momentum equation: It will be *discretely* divergence free
- Continue to next time step

# A Simple Implicit Time Advancing Scheme

- Some additional difficulties arise when an implicit method is used to solve the (incompressible) NS equations
- To illustrate, let's first try the simplest: backward/implicit Euler
  - Recall: $$\frac{\partial \rho u_i}{\partial t} = -\frac{\delta(\rho u_i u_j)}{\delta x_j} - \frac{\delta p}{\delta x_i} + \frac{\delta \tau_{ij}}{\delta x_j} = H_i - \frac{\delta p}{\delta x_i}$$
  - Implicit Euler: $$(\rho u_i)^{n+1} - (\rho u_i)^n = \Delta t \left( H_i^{n+1} - \frac{\delta p}{\delta x_i}^{n+1} \right) = \Delta t \left( -\frac{\delta(\rho u_i u_j)}{\delta x_j}^{n+1} + \frac{\delta \tau_{ij}}{\delta x_j}^{n+1} - \frac{\delta p}{\delta x_i}^{n+1} \right)$$
- Difficulties (specifics for incompressible case)

  1) Set numerical divergence of velocity field at new time-step to be zero

  - Take divergence of momentum, assume velocity is divergence-free at time $t_n$ and demand zero divergence at $t_{n+1}$. This leads to:

$$\frac{\delta(\rho u_i)^{n+1}}{\delta x_i} - \frac{\delta(\rho u_i)^n}{\delta x_i} = \Delta t \left[ \frac{\delta}{\delta x_i} \left( H_i^{n+1} - \frac{\delta p}{\delta x_i}^{n+1} \right) \right] \Rightarrow \frac{\delta}{\delta x_i} \left( \frac{\delta p}{\delta x_i}^{n+1} \right) = \frac{\delta}{\delta x_i} \left( -\frac{\delta(\rho u_i u_j)}{\delta x_j}^{n+1} + \frac{\delta \tau_{ij}}{\delta x_j}^{n+1} \right)$$

  - Problem: The RHS can not be computed until velocities are known at $t_{n+1}$ (and these velocities can not be computed until $p^{n+1}$ is available)
  - Result: Poisson and momentum equations have to be solved simultaneously

# A Simple Implicit Time Advancing Scheme, Cont'd

2) Even if $p^{n+1}$ known, a large system of nonlinear momentum equations must be solved for the velocity field:

$$(\rho u_i)^{n+1} - (\rho u_i)^n = \Delta t \left( -\frac{\delta(\rho u_i u_j)}{\delta x_j}^{n+1} + \frac{\delta \tau_{ij}}{\delta x_j}^{n+1} - \frac{\delta p}{\delta x_i}^{n+1} \right)$$

Three approaches for solution:

– First approach: nonlinear solvers

- Use velocities at $t_n$ for initial guess of $u_i^{n+1}$ (or use explicit-scheme as first guess) and then employ a nonlinear solver (Fixed-point, Newton-Raphson or Secant methods) at each time step
- Nonlinear solver is applied to the nonlinear algebraic equations

$$(\rho u_i)^{n+1} - (\rho u_i)^n = \Delta t \left( -\frac{\delta(\rho u_i u_j)}{\delta x_j}^{n+1} + \frac{\delta \tau_{ij}}{\delta x_j}^{n+1} - \frac{\delta p}{\delta x_i}^{n+1} \right)$$

$$\frac{\delta}{\delta x_i}\left(\frac{\delta p}{\delta x_i}^{n+1}\right) = \frac{\delta}{\delta x_i}\left( -\frac{\delta(\rho u_i u_j)}{\delta x_j}^{n+1} + \frac{\delta \tau_{ij}}{\delta x_j}^{n+1} \right)$$

# A Simple Implicit Time Advancing Scheme, Cont'd

– Second approach: linearize the equations about the result at $t_n$

$$u_i^{n+1} = u_i^n + \Delta u_i \quad \Rightarrow$$

$$u_i^{n+1} u_j^{n+1} = u_i^n u_j^n + u_i^n \Delta u_j + u_j^n \Delta u_i + \Delta u_i \Delta u_j$$

- We'd expect the last term to be of 2nd order in $\Delta t$, it can thus be neglected (for a 2nd order in time, e.g. C-N scheme, it would still be of same order as spatial discretization error, so can still be neglected).
- Hence, doing the same in the other terms, the (incompressible) momentum equations are then approximated by:

$$(\rho u_i)^{n+1} - (\rho u_i)^n = \rho \Delta u_i = \Delta t \left( -\frac{\delta(\rho u_i u_j)^n}{\delta x_j} - \frac{\delta(\rho u_i^n \Delta u_j)}{\delta x_j} - \frac{\delta(\rho \Delta u_i u_j^n)}{\delta x_j} + \frac{\delta \tau_{ij}{}^n}{\delta x_j} + \frac{\delta \Delta \tau_{ij}}{\delta x_j} - \frac{\delta p^n}{\delta x_i} - \frac{\delta \Delta p}{\delta x_i} \right)$$

- One then solves for $\Delta u_i$ and $\Delta p$ (using the above mom. eqn. and its $\Delta p$ eqn.)
- This linearization takes advantage of the fact that the nonlinear term is only quadratic
- However, a large coupled linear system ($\Delta u_i$ & $\Delta p$) still needs to be inverted. Direct solution is not recommended: use an iterative scheme

– A third interesting solution scheme: an Alternate Direction Implicit scheme

# Parabolic PDEs: Two spatial dimensions

(from Lecture 14)

# **ADI scheme** (Two Half steps in time)

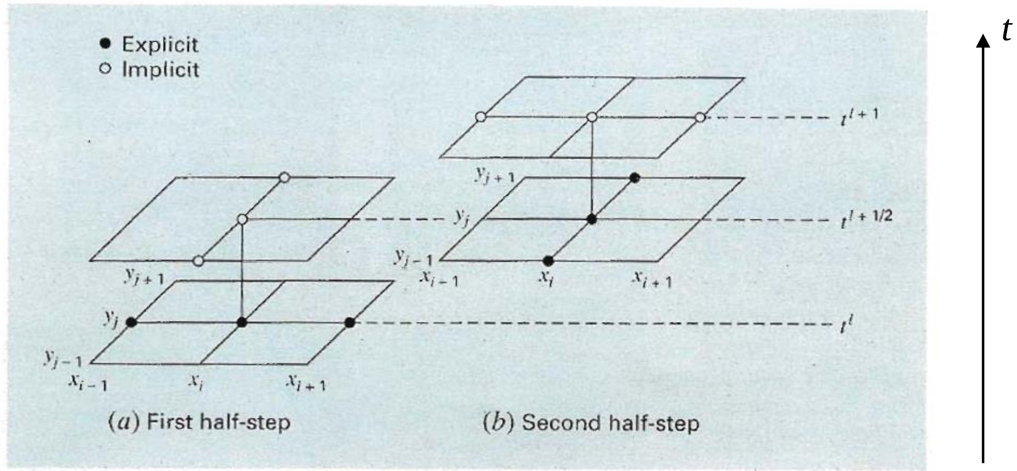



**FIGURE 30.10**
The two half-steps used in implementing the alternating-direction implicit scheme for solving parabolic equations in two spatial dimensions.


Source: Chapra, S., and R. Canale. *Numerical Methods for Engineers*. McGraw-Hill, 2005.

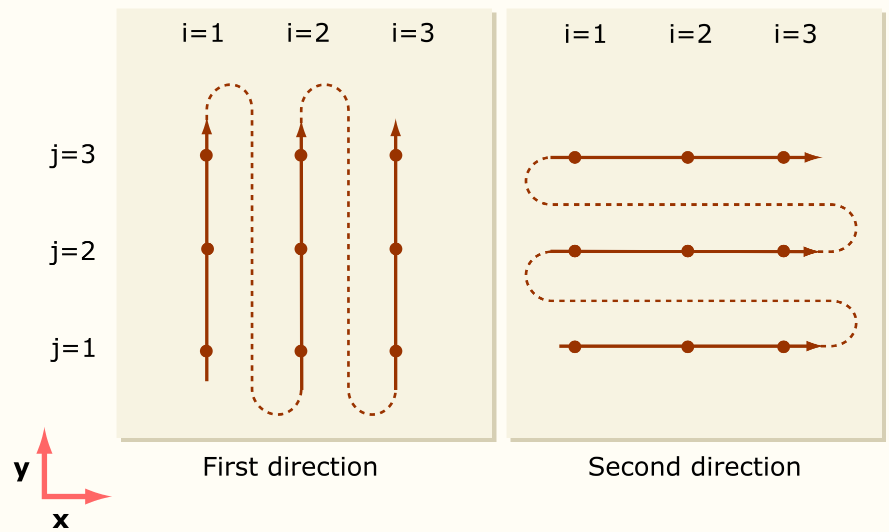
1) From time $n$ to $n+1/2$: Approximation of 2nd order $x$ derivative is explicit, while the $y$ derivative is implicit. Hence, tri-diagonal matrix to be solved:

$$\frac{T_{i,j}^{n+1/2} - T_{i,j}^{n}}{\Delta t/2} = c^2 \frac{T_{i-1,j}^{n} - 2T_{i,j}^{n} + T_{i+1,j}^{n}}{\Delta x^2} + c^2 \frac{T_{i,j-1}^{n+1/2} - 2T_{i,j}^{n+1/2} + T_{i,j+1}^{n+1/2}}{\Delta y^2} \qquad \left(O(\Delta x^2 + \Delta y^2)\right)$$

2) From time $n+1/2$ to $n+1$: Approximation of 2nd order $x$ derivative is implicit, while the $y$ derivative is explicit. Another tri-diagonal matrix to be solved:

$$\frac{T_{i,j}^{n+1} - T_{i,j}^{n+1/2}}{\Delta t/2} = c^2 \frac{T_{i-1,j}^{n+1} - 2T_{i,j}^{n+1} + T_{i+1,j}^{n+1}}{\Delta x^2} + c^2 \frac{T_{i,j-1}^{n+1/2} - 2T_{i,j}^{n+1/2} + T_{i,j+1}^{n+1/2}}{\Delta y^2} \qquad \left(O(\Delta x^2 + \Delta y^2)\right)$$

# Parabolic PDEs: Two spatial dimensions
# **ADI scheme** (Two Half steps in time)




The ADI method applied along the y direction and x direction. This method only yields tridiagonal equations if applied along the implicit dimension.



For $\Delta x = \Delta y$:

1) From time $n$ to $n+1/2$: (1st tri-diagonal sys.)

$$-rT_{i,j-1}^{n+1/2} + 2(1+r)T_{i,j}^{n+1/2} - rT_{i,j+1}^{n+1/2} = rT_{i-1,j}^{n} + 2(1-r)T_{i,j}^{n} + rT_{i+1,j}^{n}$$

2) From time $n+1/2$ to $n+1$: (2nd tri-diagonal sys.)

$$-rT_{i-1,j}^{n+1} + 2(1+r)T_{i,j}^{n+1} - rT_{i+1,j}^{n+1} = rT_{i,j-1}^{n+1/2} + 2(1-r)T_{i,j}^{n+1/2} + rT_{i,j+1}^{n+1/2}$$

# A Simple Implicit Time Advancing Scheme, Cont'd

- Alternate Direction Implicit method
  - Split the NS momentum equations into a series of 1D problems, e.g. each being block tri-diagonal. Then, either:
  - ADI nonlinear: iterate for the nonlinear terms, or,
  - ADI with a local linearization:
    - $\Delta p$ can first be set to zero to obtain a new velocity $u_i^*$ which does not satisfy continuity:

$$\left(\rho u_i^*\right)^{n+1} - \left(\rho u_i\right)^n = \Delta t \left( -\frac{\delta(\rho u_i u_j)^n}{\delta x_j} - \frac{\delta(\rho u_i^n \Delta u_j)}{\delta x_j} - \frac{\delta(\rho \Delta u_i u_j^n)}{\delta x_j} + \frac{\delta \tau_{ij}{}^n}{\delta x_j} + \frac{\delta \Delta \tau_{ij}}{\delta x_j} - \frac{\delta p^n}{\delta x_i} \right)$$

    - Solve a Poisson equation for the pressure correction. Taking the divergence of:

$$\left(\rho u_i\right)^{n+1} - \left(\rho u_i\right)^n = \Delta t \left( -\frac{\delta(\rho u_i u_j)^n}{\delta x_j} - \frac{\delta(\rho u_i^n \Delta u_j)^n}{\delta x_j} - \frac{\delta(\rho \Delta u_i u_j^n)^n}{\delta x_j} + \frac{\delta \tau_{ij}{}^n}{\delta x_j} + \frac{\delta \Delta \tau_{ij}}{\delta x_j} - \frac{\delta p^n}{\delta x_i} - \frac{\delta \Delta p}{\delta x_i} \right)$$

$$\Leftrightarrow \quad \left(\rho u_i\right)^{n+1} = \left(\rho u_i^*\right)^{n+1} - \Delta t \frac{\delta \Delta p}{\delta x_i}$$

gives, $\frac{\delta}{\delta x_i}\left(\frac{\delta \Delta p}{\delta x_i}\right) = \frac{1}{\Delta t} \frac{\delta(\rho u_i^*)^{n+1}}{\delta x_i}$ , from which $\Delta p$ can be solved for.

    - Finally, update the velocity: $\left(\rho u_i\right)^{n+1} = \left(\rho u_i^*\right)^{n+1} - \Delta t \frac{\delta \Delta p}{\delta x_i}$

# Methods for solving (steady) NS problems: Implicit Pressure-Correction Methods

- Simple implicit approach based on linearization is most useful for unsteady problems (with limited time-steps)
  - It is not accurate for large (time) steps (because the linearization would then lead to a large error)
  - Thus, it should not be used for steady problems (which often use large time-steps)
- Steady problems are often solved with an implicit method (with pseudo-time), but with large time steps (no need to reproduce the pseudo-time history)
  - The aim is to rapidly converge to the steady nonlinear solution
- Many steady-state solvers are based on variations of the implicit schemes
  - They use a pressure (or pressure-correction) equation to enforce continuity at each “pseudo-time” steps, also called “outer iteration”

# Methods for solving (steady) NS problems: Implicit Pressure-Correction Methods, Cont'd

- For a fully implicit scheme, the steady state momentum equations are:

$$(\rho u_i)^{n+1} - (\rho u_i)^n = 0 \quad \Rightarrow \quad \left( -\frac{\delta(\rho u_i u_j)}{\delta x_j}^{n+1} + \frac{\delta \tau_{ij}}{\delta x_j}^{n+1} - \frac{\delta p}{\delta x_i}^{n+1} \right) = 0$$

- With the discretized matrix notation, the result is a nonlinear algebraic system

$$\mathbf{A}^{\mathbf{u}_i^{n+1}} \mathbf{u}_i^{n+1} = \mathbf{b}_{\mathbf{u}_i}^{n+1} - \frac{\boldsymbol{\delta} p}{\boldsymbol{\delta} x_i}^{n+1}$$

  - The $\mathbf{b}$ term in the RHS contains all terms that are explicit (in $\boldsymbol{u}_i^n$) or linear in $\boldsymbol{u}_i^{n+1}$ or that are coefficients function of other variables at $t_{n+1}$, e.g. temperature
  - Pressure gradient is still written in symbolic matrix difference form to indicate that any spatial derivatives can be used
  - The algebraic system is nonlinear. Again, nonlinear iterative solvers can be used. For steady flows, the tolerance of the convergence of these nonlinear-solver iterations does not need to be as strict as for a true time-marching scheme
  - Note two types of successive iterations can be employed with pressure-correction:
    - Outer iterations: (over one pseudo-time step) use nonlinear solvers which update the elements of matrix $\mathbf{A}^{\mathbf{u}_i^{n+1}}$ as well as $\mathbf{u}_i^{n+1}$ (uses no or approx. pressure term, then corrects it)
    - Inner Iterations: linear algebra to solve the linearized system with fixed coefficients

# Methods for solving (steady) NS problems: Implicit Pressure-Correction Methods, Cont'd

- Outer iteration $m$ (pseudo-time): nonlinear solvers which update the elements of the matrix $\mathbf{A}^{\mathbf{u}_i^{m*}}$ as well as $\mathbf{u}_i^{m*}$ :

best estimate of exact $\mathbf{u}$ without any p-grad.

$$\mathbf{A}^{\mathbf{u}_i^{m*}} \mathbf{u}_i^{m*} = \mathbf{b}_{\mathbf{u}_i^{m*}}^{m-1} - \frac{\delta p}{\delta x_i}^{m-1} \Rightarrow \text{formally, } \mathbf{u}_i^{m*} = \left(\mathbf{A}^{\mathbf{u}_i^{m*}}\right)^{-1} \mathbf{b}_{\mathbf{u}_i^{m*}}^{m-1} - \left(\mathbf{A}^{\mathbf{u}_i^{m*}}\right)^{-1} \frac{\delta p}{\delta x_i}^{m-1} \equiv \tilde{\mathbf{u}}_i^{m*} - \left(\mathbf{A}^{\mathbf{u}_i^{m*}}\right)^{-1} \frac{\delta p}{\delta x_i}^{m-1}$$

  - The resulting velocities $\mathbf{u}_i^{m*}$ do not satisfy continuity (hence the *) since the RHS is obtained from $p^{m-1}$ at the end of the previous outer iteration → needs to correct $\mathbf{u}_i^{m*}$.
  - The final $\mathbf{u}_i^m$ needs to satisfy: $\mathbf{A}^{\mathbf{u}_i^m} \mathbf{u}_i^m = \mathbf{b}_{\mathbf{u}_i^m}^m - \frac{\delta p}{\delta x_i}^m$ and $\frac{\delta\, \mathbf{u}_i^m}{\delta x_i} = 0 \Rightarrow$

$$\mathbf{u}_i^m = \left(\mathbf{A}^{\mathbf{u}_i^m}\right)^{-1} \mathbf{b}_{\mathbf{u}_i^m}^{m-1} - \left(\mathbf{A}^{\mathbf{u}_i^m}\right)^{-1} \frac{\delta p}{\delta x_i}^m$$

$$\approx \left(\mathbf{A}^{\mathbf{u}_i^{m*}}\right)^{-1} \mathbf{b}_{\mathbf{u}_i^{m*}}^{m-1} - \left(\mathbf{A}^{\mathbf{u}_i^{m*}}\right)^{-1} \frac{\delta p}{\delta x_i}^m \Rightarrow 0 \approx \frac{\delta\, \tilde{\mathbf{u}}_i^{m*}}{\delta x_i} - \frac{\delta}{\delta x_i}\left( \left(\mathbf{A}^{\mathbf{u}_i^{m*}}\right)^{-1} \frac{\delta p}{\delta x_i}^m \right)$$

- Inner iteration: After solving a Poisson equation for the pressure, the final velocity is calculated using the inner iteration (fixed coefficient $\mathbf{A}$)

$$\mathbf{A}^{\mathbf{u}_i^{m*}} \mathbf{u}_i^m = \mathbf{b}_{\mathbf{u}_i^{m*}}^m - \frac{\delta p}{\delta x_i}^m$$

- Finally, increase $m$ to $m+1$ and iterate (outer, then inner)

This scheme is a variation of previous time-marching schemes. Main differences: i) no time-variation terms, and, ii) the terms in RHS can be explicit or implicit in outer iteration.

# Methods for solving (steady) NS problems: Projection Methods

- These schemes that first construct a velocity field that does not satisfy continuity, but then correct it using a pressure gradient are called "projection methods":
  - The divergence producing part of the velocity is "projected out"
- One of the most common methods of this type are the pressure-correction schemes
  - Substitute $\mathbf{u}_i^m = \mathbf{u}_i^{m*} + \mathbf{u}'$ and $p^m = p^{m-1} + p'$ in the previous equations
  - Variations of these pressure-correction methods include:
    - SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) method:
      - Neglects contributions of $\boldsymbol{u}$' in the pressure equation
    - SIMPLEC: approximate $\boldsymbol{u}$' in the pressure equation as a function of p' (better)
    - SIMPLER and PISO methods: iterate to obtain $\boldsymbol{u}$'
  - There are many other variations of these methods: all are based on outer and inner iterations until convergence at $m$ $(n+1)$ is achieved.

# Projection Methods: Example Scheme 1
# Guermond et al, CM-AME-2006

## Non-Incremental (Chorin, 1968):

- No pressure term used in predictor momentum equation
- Correct pressure based on continuity
- Update velocity using corrected pressure in momentum equation

$$\left(\rho u_i^{*}\right)^{n+1} = \left(\rho u_i\right)^{n} + \Delta t \left( -\frac{\delta\left(\rho u_i u_j\right)^{n+1}}{\delta x_j} + \frac{\delta \tau_{ij}^{\;n+1}}{\delta x_j} \right) \quad ; \quad \left(\rho u_i^{*}\right)^{n+1}\Big|_{\partial D} = (\text{bc})$$

$$\left.\begin{aligned} \left(\rho u_i\right)^{n+1} &= \left(\rho u_i^{*}\right)^{n+1} - \Delta t \,\frac{\delta p^{\,n+1}}{\delta x_i} \\ \frac{\delta\left(\rho u_i\right)^{n+1}}{\delta x_i} &= 0 \end{aligned}\right\} \Rightarrow \frac{\delta}{\delta x_i}\left(\frac{\delta p^{\,n+1}}{\delta x_i}\right) = \frac{1}{\Delta t}\frac{\delta}{\delta x_i}\left(\left(\rho u_i^{*}\right)^{n+1}\right) \quad ; \quad \frac{\delta p^{\,n+1}}{\delta n}\Big|_{\partial D} = 0$$

$$\left(\rho u_i\right)^{n+1} = \left(\rho u_i^{*}\right)^{n+1} - \Delta t \,\frac{\delta p^{\,n+1}}{\delta x_i}$$

Note: advection term can be treated:
- implicitly for $u^*$ at $n+1$ (need to iterate then), or,
- explicitly (evaluated with $u$ at $n$), as in 2d FV code and many others

# Projection Methods: Example Scheme 2
# Guermond et al, CM-AME-2006

## Incremental (Goda, 1979):

- Old pressure term used in predictor momentum equation
- Correct pressure based on continuity: $p^{n+1} = p^n + p'$
- Update velocity using pressure increment in momentum equation

$$\left(\rho u_i^*\right)^{n+1} = \left(\rho u_i\right)^n + \Delta t \left( -\frac{\delta(\rho u_i u_j)^{n+1}}{\delta x_j} + \frac{\delta \tau_{ij}^{\ n+1}}{\delta x_j} - \frac{\delta p^n}{\delta x_i} \right) \quad ; \quad \left(\rho u_i^*\right)^{n+1}\Big|_{\partial D} = (\text{bc})$$

$$\left.\begin{aligned} \left(\rho u_i\right)^{n+1} &= \left(\rho u_i^*\right)^{n+1} - \Delta t \frac{\delta\left(p^{n+1} - p^n\right)}{\delta x_i} \\ \frac{\delta\left(\rho u_i\right)^{n+1}}{\delta x_i} &= 0 \end{aligned}\right\} \Rightarrow \frac{\delta}{\delta x_i}\left(\frac{\delta\left(p^{n+1} - p^n\right)}{\delta x_i}\right) = \frac{1}{\Delta t}\frac{\delta}{\delta x_i}\left(\left(\rho u_i^*\right)^{n+1}\right); \quad \frac{\delta\left(p^{n+1} - p^n\right)}{\delta n}\Bigg|_{\partial D} = 0$$

$$\left(\rho u_i\right)^{n+1} = \left(\rho u_i^*\right)^{n+1} - \Delta t \frac{\delta\left(p^{n+1} - p^n\right)}{\delta x_i}$$

Notes:

- this scheme assumes $\boldsymbol{u}'=0$ in the pressure equation. It is as the SIMPLE method, but without the iterations
- As for the non-incremental scheme, the advection term can be explicit or implicit

# Projection Methods: Example Scheme 3
# Guermond et al, CM-AME-2006

## Rotational Incremental (Timmermans et al, 1996):

- Old pressure term used in predictor momentum equation
- Correct pressure based on continuity: $p^{n+1} = p^n + p' = p^n + \delta p^{n+1} + f(u')$
- Update velocity using pressure increment in momentum equation

$$\left(\rho u_i^*\right)^{n+1} = \left(\rho u_i\right)^n + \Delta t\left(-\frac{\delta(\rho u_i u_j)^{n+1}}{\delta x_j} + \frac{\delta \tau_{ij}^{\;n+1}}{\delta x_j} - \frac{\delta p^n}{\delta x_i}\right) \quad ; \quad \left(\rho u_i^*\right)^{n+1}\Big|_{\partial D} = (\text{bc})$$

$$\tau_{ij}^{\;n+1} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

$$\left.\begin{aligned} \left(\rho u_i\right)^{n+1} &= \left(\rho u_i^*\right)^{n+1} - \Delta t\,\frac{\delta\left(\delta p^{n+1}\right)}{\delta x_i} \\ \frac{\delta\left(\rho u_i\right)^{n+1}}{\delta x_i} &= 0 \end{aligned}\right\} \Rightarrow \frac{\delta}{\delta x_i}\left(\frac{\delta\left(\delta p^{n+1}\right)}{\delta x_i}\right) = \frac{1}{\Delta t}\frac{\delta}{\delta x_i}\left(\left(\rho u_i^*\right)^{n+1}\right); \quad \left.\frac{\delta\left(\delta p^{n+1}\right)}{\delta n}\right|_{\partial D} = 0$$

$$\left(\rho u_i\right)^{n+1} = \left(\rho u_i^*\right)^{n+1} - \Delta t\,\frac{\delta\left(\delta p^{n+1}\right)}{\delta x_i}$$

$$p^{n+1} = p^n + \delta p^{n+1} - \mu\frac{\delta}{\delta x_i}\left(\left(u_i^*\right)^{n+1}\right)$$

Notes:
- this scheme accounts for ***u***’ in the pressure eqn.
- It can be made into a SIMPLE-like method, if iterations are added
- Again, the advection term can be explicit or implicit. The rotational correction to the left assumes explicit advection

2.29 Numerical Fluid Mechanics
Spring 2015