Numerical Methods in Computational Fluid Dynamics (CFD)

Zhaoyuan Wang, Maysam Mousaviraad, Tao Xing and Fred Stern

IIHR—Hydroscience & Engineering
C. Maxwell Stanley Hydraulics Laboratory
The University of Iowa

ME:5160 Intermediate Mechanics of Fluids
http://css.engineering.uiowa.edu/~me_160/
Sep. 25, 2024

Outline

1. Introduction to Numerical Methods
2. Components of Numerical Methods
   - 2.1. Properties of Numerical Methods
   - 2.2. Discretization Methods
   - 2.3. Application of Numerical methods in PDE
   - 2.4. Numerical Grid and Coordinates
   - 2.5. Solution of Linear Equation System
   - 2.6. Convergence Criteria
3. Methods for Unsteady Problems
4. Solution of Navier-Stokes Equations
5. Example

Introduction to numerical methods

- Approaches to Fluid Dynamical Problems:
  1. Simplifications of the governing equations→ AFD
  2. Experiments on scale models→ EFD
  3. Discretize governing equations and solve by computers→ CFD
- CFD is the simulation of fluids engineering system using modeling and numerical methods
- Possibilities and Limitations of Numerical Methods:
  1. Coding level: quality assurance, programming defects, inappropriate algorithm, etc.
  2. Simulation level: iterative error, truncation error, grid error, etc.

Components of numerical methods (Properties)

- Consistence
  1. The discretization should become exact as the grid spacing tends to zero
  2. Truncation error: Difference between the discretized equation and the exact one
- Stability: does not magnify the errors that appear in the course of numerical solution process.
  1. Iterative methods: not diverge
  2. Temporal problems: bounded solutions
  3. Von Neumann's method
  4. Difficulty due to boundary conditions and non-linearities present.
- Convergence: solution of the discretized equations tends to the exact solution of the differential equation as the grid spacing tends to zero.

Components of numerical methods (Properties, Cont'd)

- Conservation

  1. The numerical scheme should on both local and global basis respect the conservation laws.
  2. Automatically satisfied for control volume method, either individual control volume or the whole domain.
  3. Errors due to non-conservation are in most cases appreciable only on relatively coarse grids, but hard to estimate quantitatively

- Boundedness:

  1. Numerical solutions should lie within proper bounds (e.g. non-negative density and TKE for turbulence; concentration between 0% and 100%, VOF between 0 and 1, etc.)
  2. Difficult to guarantee, especially for higher order schemes.

- Realizability: models of phenomena which are too complex to treat directly (turbulence, combustion, or multiphase flow) should be designed to guarantee physically realistic solutions.
- Accuracy: 1. Modeling error 2. Discretization errors 3. Iterative errors

Components of numerical methods (Discretization Methods)

- Finite Difference Method (focused in this lecture)

  1. Introduced by Euler in the 18th century.
  2. Governing equations in differential form→ domain with grid→ replacing the partial derivatives by approximations in terms of node values of the functions→ one algebraic equation per grid node→ linear algebraic equation system.
  3. Applied to structured grids

- Finite Volume Method (not focused in this lecture)

  1. Governing equations in integral form→ solution domain is subdivided into a finite number of contiguous control volumes→ conservation equation applied to each CV.
  2. Computational node locates at the centroid of each CV.
  3. Applied to any type of grids, especially complex geometries
  4. Compared to FD, FV with methods higher than 2nd order will be difficult, especially for 3D. Good mass conservation.

- Finite Element Method (not covered in this lecture):

  1. Similar to FV
  2. Equations are multiplied by a weight function before integrated over the entire domain. Often used for Solid Mechanics.

Discretization methods (Finite Difference, introduction)

- First step in obtaining a numerical solution is to discretize the geometric domain→ to define a numerical grid
- Each node has one unknown and need one algebraic equation, which is a relation between the variable value at that node and those at some of the neighboring nodes.
- The approach is to replace each term of the PDE at the particular node by a finite-difference approximation.
- Numbers of equations and unknowns must be equal

Discretization methods (Finite Difference, approximation of the first derivative)

- Taylor Series Expansion: Any continuous differentiable function, in the vicinity of x_i , can be expressed as a Taylor series:

$$\Phi(x)=\Phi(x_i)+(x-x_i)\left(\frac{\partial\Phi}{\partial x}\right)_i+\frac{(x-x_i)^2}{2!}\left(\frac{\partial^2\Phi}{\partial x^2}\right)_i+\frac{(x-x_i)^3}{3!}\left(\frac{\partial^3\Phi}{\partial x^3}\right)_i+\ldots+\frac{(x-x_i)^n}{n!}\left(\frac{\partial^n\Phi}{\partial x^n}\right)_i+H$$

$$\left(\frac{\partial\Phi}{\partial x}\right)_i=\frac{\Phi_{i+1}-\Phi_i}{x_{i+1}-x_i}-\frac{x_{i+1}-x_i}{2}\left(\frac{\partial^2\Phi}{\partial x^2}\right)_i-\frac{(x_{i+1}-x_i)^2}{6}\left(\frac{\partial^3\Phi}{\partial x^3}\right)_i+H$$

- Higher order derivatives are unknown and can be dropped when the distance between grid points is small.
- By writing Taylor series at different nodes, x_{i-1}, x_{i+1}, or both x_{i-1} and x_{i+1}, we can have:

$$\left(\frac{\partial\Phi}{\partial x}\right)_i\approx\frac{\Phi_{i+1}-\Phi_i}{x_{i+1}-x_i}$$ **Forward-FDS** (forward difference scheme)

$$\left(\frac{\partial\Phi}{\partial x}\right)_i\approx\frac{\Phi_i-\Phi_{i-1}}{x_i-x_{i-1}}$$ **Backward-BDS**

$$\left(\frac{\partial\Phi}{\partial x}\right)_i\approx\frac{\Phi_{i+1}-\Phi_{i-1}}{x_{i+1}-x_{i-1}}$$ **Central-CDS**

1st order, order of accuracy $P_{kest}=1$

2nd order, order of accuracy $P_{kest}=2$

Discretization methods (Finite Difference, approximation of the first derivative, Cont'd)

- Polynomial fitting: fit the function to an interpolation curve and differentiate the resulting curve.

  Example: fitting a parabola to the data at points x_{i-1},x_i, and x_{i+1}, and computing the first derivative at x_i, we obtain:

$$\left(\frac{\partial \Phi}{\partial x}\right)_i \approx \frac{\Phi_{i+1}\left(\Delta x_i\right)^2 - \Phi_{i-1}\left(\Delta x_{i+1}\right)^2 + \Phi_i\left[\left(\Delta x_{i+1}\right)^2 - \left(\Delta x_i\right)^2\right]}{\Delta x_{i+1}\Delta x_i\left(\Delta x_i + \Delta x_{i+1}\right)}$$

$$\Delta x_i = x_i - x_{i-1}$$

  2nd order truncation error on any grid. For uniform meshing, it reduced to the CDS approximation given in previous slide.

- Compact schemes: Depending on the choice of parameters α, β, and γ, 2nd order and 4th order CDS, 4th order and 6th order Pade scheme are obtained.

$$\alpha\left(\frac{\partial \Phi}{\partial x}\right)_{i+1} + \left(\frac{\partial \Phi}{\partial x}\right)_i + \alpha\left(\frac{\partial \Phi}{\partial x}\right)_{i-1} = \beta\frac{\Phi_{i+1} - \Phi_{i-1}}{2\Delta x} + \gamma\frac{\Phi_{i+2} - \Phi_{i-2}}{4\Delta x}$$

- Non-Uniform Grids: to spread the error nearly uniformly over the domain, it will be necessary to use smaller Δx in regions where derivatives of the function are large and larger Δx where function is smooth. Save computational resources.

Discretization methods (Finite Difference, approximation of the second derivative)

- Geometrically, the second derivative is the slope of the line tangent to the curve representing the first derivative.

$$\left(\frac{\partial^2\Phi}{\partial x^2}\right)_i \approx \frac{\left(\frac{\partial\Phi}{\partial x}\right)_{i+1} - \left(\frac{\partial\Phi}{\partial x}\right)_i}{x_{i+1} - x_i}$$

Estimate the outer derivative by FDS, and estimate the inner derivatives using BDS, we get

$$\left(\frac{\partial^2\Phi}{\partial x^2}\right)_i \approx \frac{\Phi_{i+1}(x_i - x_{i-1}) + \Phi_{i-1}(x_{i+1} - x_i) - \Phi_i(x_{i+1} - x_{i-1})}{(x_{i+1} - x_i)^2(x_i - x_{i-1})}$$

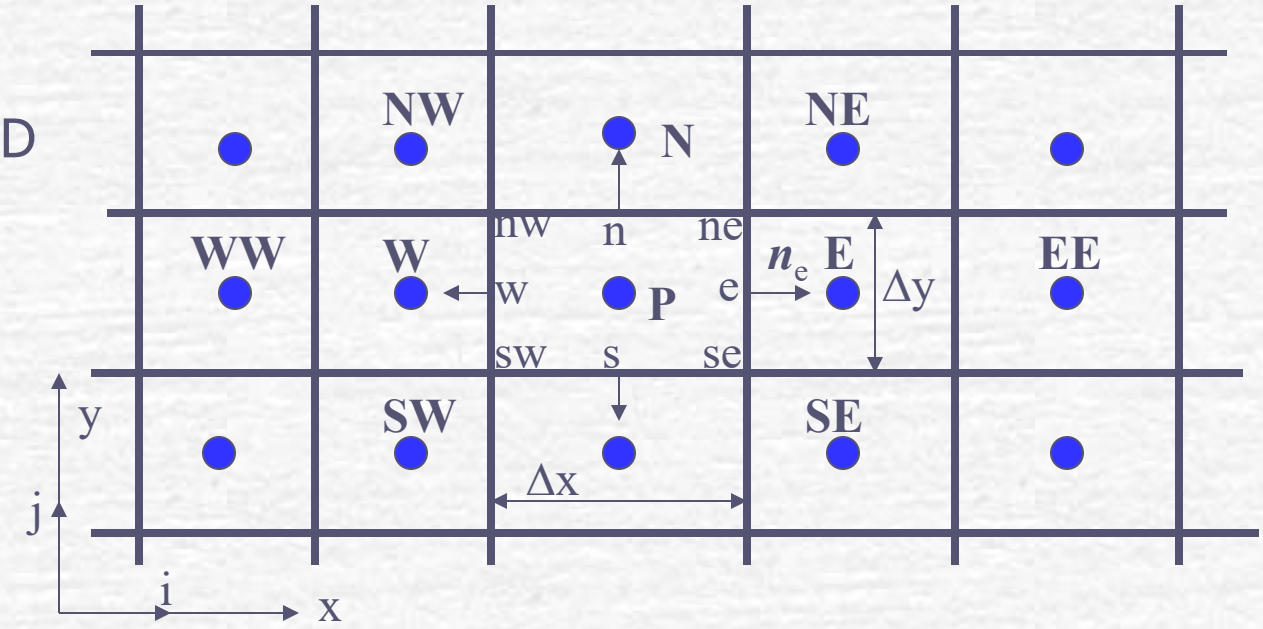
For equidistant spacing of the points:

$$\left(\frac{\partial^2\Phi}{\partial x^2}\right)_i \approx \frac{\Phi_{i+1} - 2\Phi_i + \Phi_{i-1}}{(\Delta x)^2}$$

Higher-order approximations for the second derivative can be derived by including more data points, such as x_{i-2}, and x_{i+2}, even x_{i-3}, and x_{i+3}

Discretization methods (Finite Volume)

- FV methods uses the integral form of the conservation equation
- FV defines the control volume boundaries while FD define the computational nodes
- Computational node located at the Control Volume center
- Global conservation automatically satisfied



Typical CV and the notation for Cartesian 2D

- FV methods use the integral form of the conservation equation

$$\int_S f dS = \sum_k \int_{S_k} f dS$$

Application of numerical methods in PDE

- Fluid Mechanics problems are governed by the laws of physics, which are formulated for unsteady flows as initial and boundary value problems (IBVP), which is defined by a continuous partial differential equation (PDE) operator L_T (no modeling or numerical errors, T is the true or exact solution)

$$L_T(T)=0 \qquad IC: T(x,t=0)=G_T(x) \qquad BC: T(x_B,t)=H_T(t) \qquad \text{A1}$$

- Analytical and CFD approaches formulate the IBVP by selection of the PDE, IC, and BC to model the physical phenomena

$$L_M(M)=0 \qquad IC: M(x,t=0)=G_M(x) \qquad BC: M(x_B,t)=H_M(t) \qquad \text{A2}$$

- Using numerical methods, the continuous IBVP is reduced to a discrete IBVP (computer code), and thus introduce numerical errors:

$$L_N(S)=\Gamma_1 \qquad IC: S(x,t=0)=g_N(x) \qquad BC: S(x_B,t)=h_N(t) \qquad \text{A3}$$

- Numerical errors can be defined and evaluated by transforming the discrete IBVP back to a continuous IBVP.

A4

$$S=S_C+\sum_{j=1}^{J}\sum_{i=1}^{\infty}\frac{(\Delta x_j)^i}{i!}\frac{\partial^i S}{\partial x_j^i}$$

$$L_{Modified}(S)=L_M(S)=\Gamma_N \qquad BC: S(x_B,t)=H_{Modified}(t)$$

$$\Gamma_N=\Gamma_1+\sum_{j=1}^{J}\Gamma_j \qquad IC: S(x,t=0)=G_{Modified}(x)$$

Truncation error

Application of numerical methods in PDE (Truncation and Discretization errors)

- Subtracting equations A2 and A4 gives the IBVP that governs the simulation numerical error $\delta_{SN} = S - M$

$$L_M(S-M) = L_M(\delta_{SN}) = \Gamma_N = \Gamma_1 + \sum_{j=1} \Gamma_j$$

$$IC: \delta_{SN}(x, t=0) = G_{Modified}(x) - G_M(x)$$

$$BC: \delta_{SN}(x_B, t) = H_{Modified}(t) - H_M(t)$$

A5

- An IBVP for the modeling error M-T can be obtained by subtracting A1 and A2:

$$L_M(M-T) = L_M(\delta_{SM}) = \Gamma_M = -L_M(T)$$

$$IC: \delta_{SM}(x, t=0) = G_M(x) - G_T(x)$$

$$BC: \delta_{SM}(x_B, t) = H_M(t) - H_T(t)$$

A6

- Adding A5 and A6

$$L_M(S-T) = L_M(\delta_S) = \Gamma_N + \Gamma_M$$

$$IC: \delta_S(x, 0) = G_{Modified}(x) - G_T(x)$$

$$BC: \delta_S(x_B, t) = H_{Modified}(t) - H_T(t)$$

$$\delta_S = S - T = \delta_{SN} + \delta_{SM}$$

Numerical grids and coordinates

- The discrete locations at which the variables are to be calculated are defined by the numerical grid
- Numerical grid is a discrete representation of the geometric domain on which the problem is to be solved. It divides the solution domain into a finite number of sub-domains
- Type of numerical grids: 1. structured (regular grid), 2. Block-structured grids, and

  3. Unstructured grids
- Detailed explanations of numerical grids will be presented in the last lecture of this CFD lecture series.
- Different coordinates have been covered in "Introduction to CFD"

Components of numerical methods (Solution of linear equation systems, introduction)

- The result of the discretization using either FD or FV, is a system of algebraic equations, which are linear or non-linear

$$A\Phi = Q$$

- For non-linear case, the system must be solved using iterative methods, i.e. initial guess→ iterate→ converged results obtained.
- The matrices derived from partial differential equations are always sparse with the non-zero elements of the matrices lie on a small number of well-defined diagonals

Solution of linear equation systems (direct methods)

- Gauss Elimination: Basic methods for solving linear systems of algebraic equations but does not vectorize or parallelize well and is rarely used without modifications in CFD problems.

$$A=\begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ A_{n1} & A_{n2} & A_{n3} & \cdots & A_{nn} \end{pmatrix} \quad U=\begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} \\ 0 & A_{22} & A_{23} & \cdots & A_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & A_{nn} \end{pmatrix}$$

$$\Phi_n = \frac{Q_n}{A_{nn}}$$

$$\Phi_i = \frac{Q_i - \sum_{k=i+1}^{n} A_{ik}\Phi_k}{A_{ii}}$$

- LU Decomposition: the factorization can be performed without knowing the vector Q

$$A = LU \qquad U\Phi = Y \qquad LY = Q$$

- Tridiagonal Systems: *Thomas Algorithm* or *Tridiagonal Matrix Algorithm (TDMA) P95*

$$A_W^i \Phi_{i-1} + A_P^i \Phi_i + A_E^i \Phi_{i+1} = Q_i$$

$$A_P^i = A_P^i - \frac{A_W^i A_E^{i-1}}{A_P^{i-1}} \qquad Q_i^* = Q_i - \frac{A_W^i Q_{i-1}^*}{A_P^{i-1}} \qquad \Phi_i = \frac{Q_i^* - A_E^i \Phi_{i+1}}{A_P^i}$$

Solution of linear equation systems (iterative methods)

- Why use iterative methods:
  1. in CFD, the cost of direct methods is too high since the triangular factors of sparse matrices are not sparse.
  2. Discretization error is larger than the accuracy of the computer arithmetic
- Purpose of iteration methods: drive both the residual and iterative error to be zero
- Rapid convergence of an iterative method is key to its effectiveness.

$$A\Phi = Q \qquad A\Phi^n = Q - \rho^n \qquad \varepsilon^n = \Phi - \Phi^n$$

$$A\varepsilon^n = \rho^n$$

Φ^n Approximate solution after n iteration

ρ^n residual

ε^n Iteration error

Solution of linear equation systems (iterative methods, cont'd)

- Typical iterative methods:
  1. Jacobi method
  2. Gauss-Seidel method
  3. Successive Over-Relaxation (SOR), or LSOR
  4. Alternative Direction Implicit (ADI) method
  5. Conjugate Gradient Methods
  6. Biconjugate Gradients and CGSTAB
  7. Multigrid Methods

Solution of linear equation systems (iterative methods, examples)

- Jacobi method:

$$\Phi_i^{k+1} = \Phi_i^k + \frac{R_i^k}{A_{ii}} \qquad R_i^k = Q_i - \sum_{j=1}^{n} A_{ij}\Phi_j^k \qquad (i = 1,2,\ldots,n)$$

- Gauss-Seidel method: similar to Jacobi method, but most recently computed values of all Φ_i are used in all computations.

$$\Phi_i^{k+1} = \Phi_i^k + \frac{R_i^k}{A_{ii}} \qquad R_i^k = Q_i - \sum_{j=1}^{i-1} A_{ij}\Phi_j^{k+1} - \sum_{j=i}^{n} A_{ij}\Phi_j^k \qquad (i = 1,2,\ldots,n)$$

- Successive Overrelaxation (SOR):

$$\Phi_i^{k+1} = \Phi_i^k + \omega\frac{R_i^k}{A_{ii}} \qquad R_i^k = Q_i - \sum_{j=1}^{i-1} A_{ij}\Phi_j^{k+1} - \sum_{j=i}^{n} A_{ij}\Phi_j^k \qquad (i = 1,2,\ldots,n)$$

Solution of linear equation systems (coupled equations and their solutions)

- Definition: Most problems in fluid dynamics require solution of coupled systems of equations, i.e. dominant variable of each equation occurs in some of the other equations
- Solution approaches:
  1. Simultaneous solution: all variables are solved for simultaneously
  2. Sequential Solution: Each equation is solved for its dominant variable, treating the other variables as known, and iterating until the solution is obtained.
- For sequential solution, inner iterations and outer iterations are necessary

Solution of linear equation systems (non-linear equations and their solutions)

- Definition:

  Given the continuous nonlinear function f(x), find the value x=α, such that f(α)=0 or f(α)=β

- Solution approaches:

  1. Newton-like Techniques: root finding algorithm, faster but need good estimation of the solution. Seldom used for solving Navier-Stokes equations.

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \qquad x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})}$$

  2. Global: guarantee not to diverge but slower, such as sequential decoupled method

Solution of linear equation systems (convergence criteria and iteration errors)

- Convergence Criteria: Used to determine when to quit for iteration method
  1. Difference between two successive iterates
  2. Order drops of the residuals
  3. Integral variable vs. iteration history

$$\left|\Delta\Phi_{i,j}^{k+1}\right| < \varepsilon \quad \text{(for all i, j)} \qquad \left|\Delta\Phi_{i,j}^{k+1}\right| < \varepsilon\left|f_{i,j}^{k}\right| \quad \text{(for all i, j)}$$

$$\sum_{i,j}^{n}\left|\Delta\Phi_{i,j}^{k+1}\right| < \varepsilon \qquad \sum_{i,j}^{n}\left|\frac{\Delta\Phi_{i,j}^{k+1}}{\Phi_{i,j}^{k}}\right| < \varepsilon \qquad \left(\sum_{i,j}^{n}\left(\Delta\Phi_{i,j}^{k+1}\right)^2\right)^{1/2} < \varepsilon$$

- Inner iterations can be stopped when the residual has fallen by one to two orders of magnitude.
- Details on how to estimate iterative errors have been presented in CFD lecture.

Methods for unsteady problems (introduction)

- Unsteady flows have a fourth coordinate direction– time, which must be discretized.
- Differences with spatial discretization: a force at any space location may influence the flow anywhere else, forcing at a given instant will affect the flow only in the future (parabolic like).
- These methods are very similar to ones applied to initial value problems for ordinary differential equations.

- The basic problem is to find the solution Φ a short time Δt after the initial point. The solution at $t_1=t_0+\Delta t$, can be used as a new initial condition and the solution can be advanced to $t_2=t_1+\Delta t$, $t_3=t_2+\Delta t$, ….etc.

Methods for unsteady problems

- Methods for Initial Value Problems in ODEs
  1. Two-Level Methods (explicit/implicit Euler)
  2. Predictor-Corrector and Multipoint Methods
  3. Runge-Kutta Methods
  4. Other methods: Lagrangian and Semi-Lagrangian scheme

$$\frac{d\Phi(t)}{dt} = f(t, \Phi(t)) \qquad \Phi(t_0) = \Phi^0$$

- Application to the Generic Transport Equation
  1. Explicit methods
  2. Implicit methods
  3. Other methods

$$\frac{\partial \Phi}{\partial t} = -u\frac{\partial \Phi}{\partial x} + \frac{\Gamma}{\rho}\frac{\partial^2 \Phi}{\partial x^2}$$

Methods for unsteady problems (examples)

- Methods for Initial Value Problems in ODEs (explicit and implicit Euler method)

explicit $\Phi^{n+1} = \Phi^n + f\left(t_n, \Phi^n\right)\Delta t$

$\Phi^{n+1} = \Phi^n + f\left(t_{n+1}, \Phi^{n+1}\right)\Delta t$ implicit

- Methods for Initial Value Problems in ODEs (4th order Runge-Kutta method)

$$\Phi^*_{n+\frac{1}{2}} = \Phi^n + \frac{\Delta t}{2} f\left(t_n, \Phi^n\right)$$

$$\Phi^{**}_{n+\frac{1}{2}} = \Phi^n + \frac{\Delta t}{2} f\left(t_{n+\frac{1}{2}}, \Phi^*_{n+\frac{1}{2}}\right)$$

$$\Phi^*_{n+1} = \Phi^n + \Delta t f\left(t_{n+\frac{1}{2}}, \Phi^{**}_{n+\frac{1}{2}}\right)$$

$$\Phi^{n+1} = \Phi^n + \frac{\Delta t}{6}\left[f\left(t_n, \Phi^n\right) + 2f\left(t_{n+\frac{1}{2}}, \Phi^*_{n+\frac{1}{2}}\right) + 2f\left(t_{n+\frac{1}{2}}, \Phi^{**}_{n+\frac{1}{2}}\right) + f\left(t_{n+1}, \Phi^*_{n+1}\right)\right]$$

Methods for unsteady problems (examples)

- Application to the Generic Transport Equation (Explicit Euler methods)

$$\Phi_i^{n+1} = \Phi_i^n + \left[-u \frac{\Phi_{i+1}^n - \Phi_{i-1}^n}{2\Delta x} + \frac{\Gamma}{\rho} \frac{\Phi_{i+1}^n + \Phi_{i-1}^n - 2\Phi_i^n}{(\Delta x)^2} \right] \Delta t$$

Assume constant velocity

$$\Phi_i^{n+1} = (1-2d)\Phi_i^n + \left(d - \frac{c}{2} \right) \Phi_{i+1}^n + \left(d + \frac{c}{2} \right) \Phi_{i-1}^n$$

$$d = \frac{\Gamma \Delta t}{\rho (\Delta x)^2}$$

Time required for a disturbance to be transmitted By diffusion over a distance Δx

$$c = \frac{u \Delta t}{\Delta x}$$

Courant number (or CFL number) when diffusion negligible, Courant number should be smaller than unity to make the scheme stable

Methods for unsteady problems (examples)

- Application to the Generic Transport Equation (Implicit Euler methods)

$$\Phi_i^{n+1} = \Phi_i^n + \left[-u \frac{\Phi_{i+1}^{n+1} - \Phi_{i-1}^{n+1}}{2\Delta x} + \frac{\Gamma}{\rho} \frac{\Phi_{i+1}^{n+1} + \Phi_{i-1}^{n+1} - 2\Phi_i^{n+1}}{(\Delta x)^2} \right] \Delta t$$

Assume constant velocity

$$(1+2d)\Phi_i^{n+1} + \left(\frac{c}{2} - d \right) \Phi_{i+1}^{n+1} + \left(-\frac{c}{2} - d \right) \Phi_{i-1}^{n+1} = \Phi_i^n$$

- Advantage: Use of the implicit Euler method allows arbitrarily large time steps to be taken
- Disadvantage: first order truncation error in time and the need to solve a large coupled set of equations at each time step, and more computational time for iterations.

Treatment of Advection Equations

- **Lagrangian Approach**: Tracks individual fluid parcels as they move through space.
- **Eulerian Approach**: Focuses on fixed points in space and examines how fluid properties change at those points.
- **Semi-Lagrangian Approach**: combines both Lagrangian and Eulerian approaches,
  Advantages: stability, accuracy;
  Disadvantages: interpolation errors, complexity;
  Applications: Weather Forecasting, Climate Modeling, Ocean Engineering.

- In the **Eulerian** method, the advection of some quantity C is described by:

$$\frac{DC}{Dt} \equiv \frac{\partial C}{\partial t} + \mathbf{u} \bullet \nabla C = 0$$

- In the *pure* **Lagrangian** formulation,

$$\frac{dC}{dt} = 0,$$
$$\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}, t).$$

- The **Semi-Lagrangian** scheme

$$\frac{C^{n+1} - C_d^n}{\Delta t} = 0,$$
$$\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}, t),\ \mathbf{x}^{n+1} = \mathbf{x}(t^{n+1}) = \mathbf{x}_a$$

where *a* and *d* denote arrival and departure point, respectively.

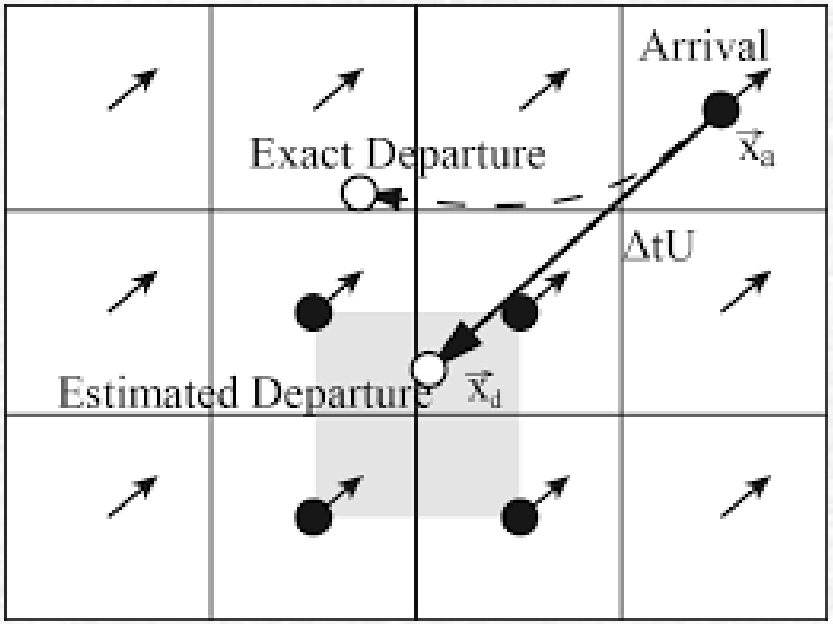
Semi-Lagrangian Scheme

- Backward Integration:

Find the departure point, x_d, of the fluid particle arriving at the grid point x^{n+1}

$$\hat{\mathbf{x}} = \mathbf{x}_a - \frac{\Delta t}{2}\mathbf{u}(\mathbf{x}_a, t^n),$$

$$\mathbf{x}_d = \mathbf{x}_a - \Delta t \mathbf{u}(\hat{\mathbf{x}}, t^n + \frac{\Delta t}{2}).$$



(Jin and Chen, 2015, IJNMHFF)

- Interpolation:

The departure point usually does not coincide with the grid point, the solution value needs to be interpolated at the departure point.

$$C^{n+1}(\mathbf{x}_a, t^{n+1}) = C_d^n(\mathbf{x}_d, t^n)$$

Semi-Lagrangian scheme for VOF

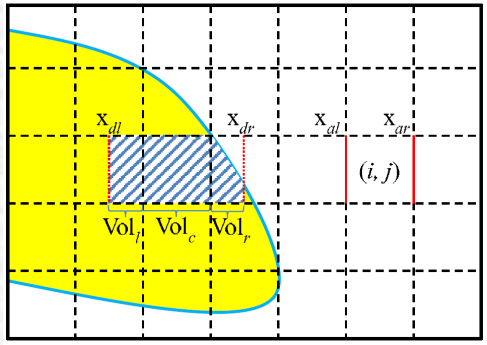
- Unconditionally stable: allow very large CFL for significant speedup of overall computation
- Comparable to an benchmark Eulerian VOF method in terms of interface position errors
- Improved with regard to mass conservation, even with much larger CFL numbers

* Z. Wang, J. Yang, F. Stern, A simple and conservative operator-splitting semi-Lagrangian volume-of-fluid advection scheme, J. Comp. Phys., in review, 2012

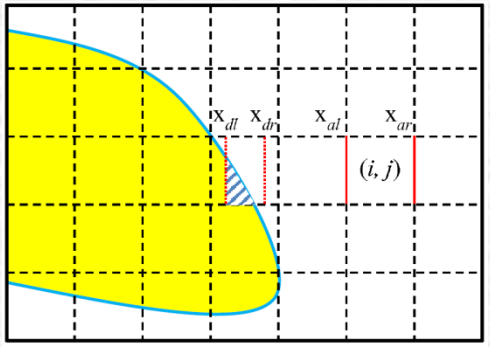
Semi-Lagrangian discretization of VOF Equation

$$\frac{dF}{dt} = 0, \qquad \frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}, t). \quad \Rightarrow \quad \frac{F^{n+1} - F_d^n}{\Delta t} = 0, \qquad \frac{\mathbf{x}^{n+1} - \mathbf{x}_d^n}{\Delta t} = \mathbf{u}(\mathbf{x}, t),$$

The "departure cell" of each computational cell is tracked backward instead of the grid point, which can be determined by locating the two "departure faces" of the cell in each spatial direction.



A stretched departure cell



A compressed departure cell

VOF reconstruction

Total volume in the departure cell

$$Vol_{tot} = Vol_c + Vol_r + Vol_l,$$

where

$$Vol_c = -(F^n \Delta x \Delta y)_{i_{dl}, j} + \sum_{ii=i_{dl}}^{i_{dr}-1} (F^n \Delta x \Delta y)_{ii, j}.$$

The VOF value updated by

$$F^{n+1}(\mathbf{x}_a, t^{n+1}) = \frac{Vol_{tot}(\mathbf{x}_d, t^n)}{|x_{dr} - x_{dl}| \, \Delta y_{i,j}}.$$

Semi-Lagrangian Scheme for the Navier-Stokes Equations

Navier-Stokes equations in Lagrangian form:

$$\frac{d\mathbf{u}}{dt} = \frac{1}{\rho}\nabla\cdot(-p\mathbf{I}+\mathbf{T})+\mathbf{g}$$

$$\nabla\cdot\mathbf{u} = 0$$

Semi-Lagrangian time-discretization using a second-order scheme:

$$\frac{\frac{3}{2}\mathbf{u}^{n+1} - 2\mathbf{u}_d^n + \frac{1}{2}\mathbf{u}_d^{n-1}}{\Delta t} = \left[\frac{1}{\rho}\nabla\cdot(-p\mathbf{I}+\mathbf{T})+\mathbf{g}\right]^{n+1}$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}^n(\mathbf{x},t),\ \ \mathbf{x}(t^{n+1}) = \mathbf{x}_a$$

The values at the departure points can be obtained using the same ENO scheme as in the level set equation.

Solution of Navier-Stokes equations

- Special features of Navier-Stokes Equations
- Choice of Variable Arrangement on the Grid
- Pressure Poisson equation
- Solution methods for N-S equations

Solution of N-S equations (special features)

- Navier-Stokes equations (3D in Cartesian coordinates)

$$\rho\frac{\partial u}{\partial t}+\rho u\frac{\partial u}{\partial x}+\rho v\frac{\partial u}{\partial y}+\rho w\frac{\partial u}{\partial z}=-\frac{\partial \hat{p}}{\partial x}+\mu\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right]$$

$$\rho\frac{\partial v}{\partial t}+\rho u\frac{\partial v}{\partial x}+\rho v\frac{\partial v}{\partial y}+\rho w\frac{\partial v}{\partial z}=-\frac{\partial \hat{p}}{\partial y}+\mu\left[\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right]$$

$$\rho\frac{\partial w}{\partial t}+\rho u\frac{\partial w}{\partial x}+\rho v\frac{\partial w}{\partial y}+\rho w\frac{\partial w}{\partial z}=-\frac{\partial \hat{p}}{\partial z}+\mu\left[\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right]$$

Local acceleration

Convection

Piezometric pressure gradient

Viscous terms

$$\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}+\frac{\partial(\rho w)}{\partial z}=0$$

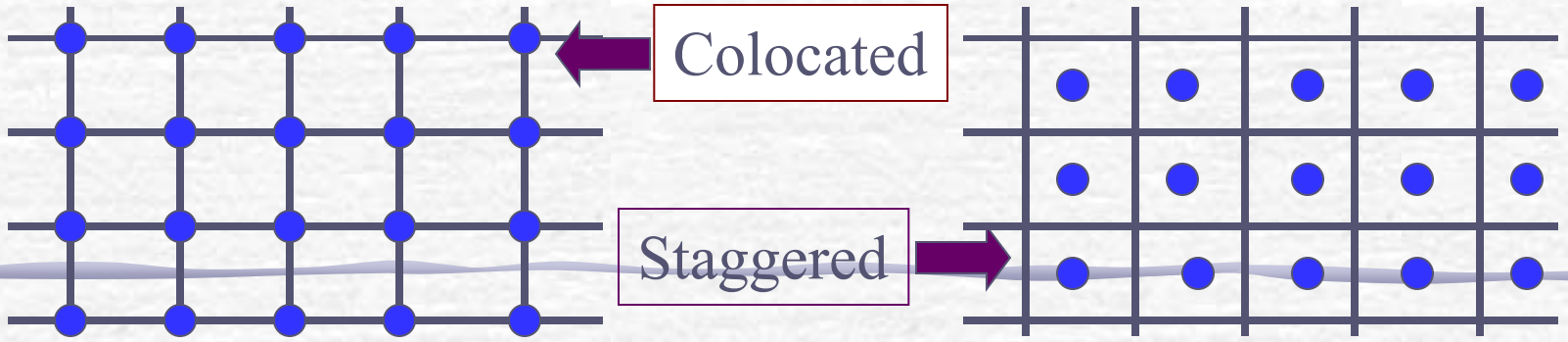
Continuity equation

- Discretization of Convective, pressure and Viscous terms
- Conservation properties: 1. Guaranteeing global energy conservation in a numerical method is a worthwhile goal, but not easily attained;

2. Incompressible isothermal flows, significance is kinetic energy; 3. heat transfer: thermal energy>>kinetic energy

Solution of N-S equations (choice of variable arrangement on the grid)

- Collocated arrangement:
  1. Store all the variables at the same set of grid points and to use the same control volume for all variables
  2. Advantages: easy to code
  3. Disadvantages: pressure-velocity decoupling, approximation for terms
- Staggered Arrangements:
  1. Not all variables share the same grid
  2. Advantages: (1). Strong coupling between pressure and velocities, (2). Some terms interpolation in collocated arrangement can be calculated with interpolation.
  3. Disadvantages: higher order numerical schemes with order higher than 2^{nd} will be difficult



Solution of Navier-Stokes equations (Pressure Poisson equation)

- Why need equation for pressure: 1. N-S equations lack an independent equation for the pressure; 2. in incompressible flows, continuity equation cannot be used directly
- Derivation: obtain Poisson equation by taking the divergence of the momentum equation and then simplify using the continuity equation.
- Poisson equation is an elliptic problem, i.e. pressure values on boundaries must be known to compute the whole flow field

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

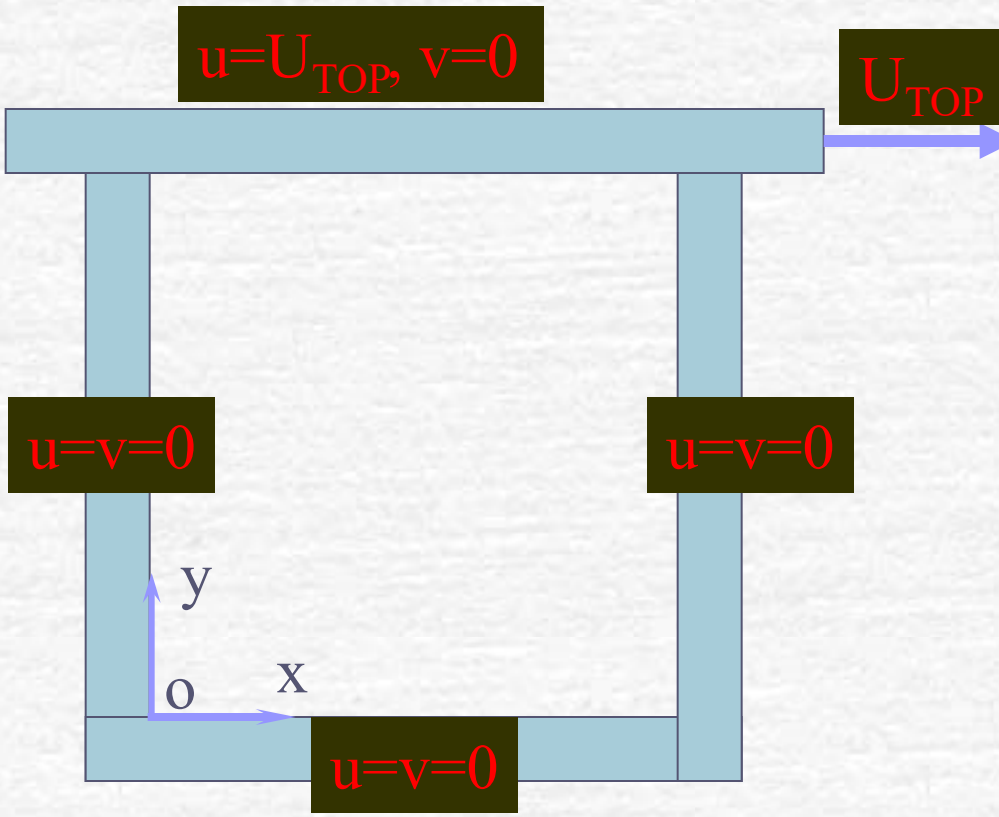
Solution methods for the Navier-Stokes equations

- Analytical Solution (fully developed laminar pipe flow)
- Vorticity-Stream Function Approach: eliminate pressure term
- The SIMPLE (Semi-Implicit Method for pressure-Linked Equations) Algorithm:

1. Guess the pressure field p*
2. Solve the momentum equations to obtain u*,v*,w*
3. Solve the p' equation (The pressure-correction equation)
4. p=p*+p'
5. Calculate u, v, w from their starred values using the velocity-correction equations
6. Solve the discretization equation for other variables, such as temperature, concentration, and turbulence quantities.
7. Treat the corrected pressure p as a new guessed pressure p*, return to step 2, and repeat the whole procedure until a converged solution is obtained.

Example (lid-driven cavity)

- The driven cavity problem is a classical problem that has wall boundaries surrounding the entire computational region.
- Incompressible viscous flow in the cavity is driven by the uniform translation of the moving upper lid.
- the vorticity-stream function method is used to solve the driven cavity problem.



Example (lid-driven cavity, governing equations)

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0$$

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{\partial p}{\partial x}+\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{\partial p}{\partial y}+\nu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)$$

$$\zeta=\frac{\partial v}{\partial x}-\frac{\partial u}{\partial y}$$

$$\frac{\partial \psi}{\partial y}=u$$

$$\frac{\partial \psi}{\partial x}=-v$$

$$x^*=\frac{x}{l},y^*=\frac{y}{l},t^*=\frac{t}{l/U},u^*=\frac{u}{U},v^*=\frac{v}{U},p^*=\frac{p}{\rho U^2},\zeta^*=\frac{\zeta}{U/l},\psi^*=\frac{\psi}{Ul}$$

$$\frac{\partial \zeta}{\partial t}+u\frac{\partial \zeta}{\partial x}+v\frac{\partial \zeta}{\partial y}=\frac{1}{\mathrm{Re}_l}\left(\frac{\partial^2 \zeta}{\partial x^2}+\frac{\partial^2 \zeta}{\partial y^2}\right)$$

$$\frac{\partial^2 \psi}{\partial x^2}+\frac{\partial^2 \psi}{\partial y^2}=-\zeta$$

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=2\left[\left(\frac{\partial^2 \psi}{\partial x^2}\right)\left(\frac{\partial^2 \psi}{\partial y^2}\right)-\left(\frac{\partial^2 \psi}{\partial x \partial y}\right)^2\right]$$

$$\mathrm{Re}_l=\frac{Ul}{\nu}$$

$$u\frac{\partial \zeta}{\partial x}+v\frac{\partial \zeta}{\partial y}=\frac{\partial (u\zeta)}{\partial x}+\frac{\partial (v\zeta)}{\partial y}$$

Example (lid-driven cavity, boundary conditions)

$$\psi = 0 \quad \text{on all walls}$$

The top wall

$$\psi_{w+1} = \psi_w + \Delta n \frac{\partial \psi}{\partial n}\Big|_w + \frac{1}{2}(\Delta n)^2 \frac{\partial^2 \psi}{\partial n^2}\Big|_w + o\left[(\Delta n)^3\right]$$

$$\zeta_{i,NJ} = \frac{2(\psi_{i,NJ} - u_{i\ NJ}\Delta y - \psi_{i,NJ-1})}{(\Delta y)^2}$$

$$\zeta_w = \frac{2(\psi_{w+1} - \psi_w)}{(\Delta n)^2}$$

The other Three walls

$$\frac{\partial p}{\partial s}\Big|_w = -\frac{1}{\mathrm{Re}_l}\frac{\partial \zeta}{\partial n}\Big|_w$$

$$\frac{p_{s+1,w} - p_{s-1,w}}{2\Delta s} = -\frac{1}{\mathrm{Re}_l}\left(\frac{-3\zeta_{s,w} + 4\zeta_{s,w+1} - \zeta_{s,w+2}}{2\Delta n}\right)$$

For wall pressures, using the tangential momentum equation to the fluid adjacent to the wall surface, get:

* s is measured along the wall surface and n is normal to it

* Pressure at the lower left corner of the cavity is assigned 1.0

Example (lid-driven cavity, discretization methods)

$$\frac{\zeta_{i,j}^{n+1}-\zeta_{i,j}^{n}}{\Delta t}+\frac{u_{i+1,j}^{n}\zeta_{i+1,j}^{n}-u_{i-1,j}^{n}\zeta_{i-1,j}^{n}}{2\Delta x}+\frac{v_{i,j+1}^{n}\zeta_{i,j+1}^{n}-v_{i,j-1}^{n}\zeta_{i,j-1}^{n}}{2\Delta y}$$

$$=\frac{1}{\mathrm{Re}_l}\left(\frac{\zeta_{i+1,j}^{n}-2\zeta_{i,j}^{n}+\zeta_{i-1,j}^{n}}{(\Delta x)^2}+\frac{\zeta_{i,j+1}^{n}-2\zeta_{i,j}^{n}+\zeta_{i,j-1}^{n}}{(\Delta y)^2}\right)$$

1st order upwind for time derivative

$$\frac{\psi_{i+1,j}^{n}-2\psi_{i,j}^{n}+\psi_{i-1,j}^{n}}{(\Delta x)^2}+\frac{\psi_{i,j+1}^{n}-2\psi_{i,j}^{n}+\psi_{i,j-1}^{n}}{(\Delta y)^2}=-\zeta_{i,j}^{n}$$

$$\frac{p_{i+1,j}^{n}-2p_{i,j}^{n}+p_{i-1,j}^{n}}{(\Delta x)^2}+\frac{p_{i,j+1}^{n}-2p_{i,j}^{n}+p_{i,j-1}^{n}}{(\Delta y)^2}$$

$$=2\left[\left(\frac{\psi_{i+1,j}^{n}-2\psi_{i,j}^{n}+\psi_{i-1,j}^{n}}{(\Delta x)^2}\right)\left(\frac{\psi_{i,j+1}^{n}-2\psi_{i,j}^{n}+\psi_{i,j-1}^{n}}{(\Delta y)^2}\right)-\left(\frac{\psi_{i+1,j+1}^{n}-\psi_{i+1,j-1}^{n}-\psi_{i-1,j+1}^{n}+\psi_{i-1,j-1}^{n}}{4\Delta x\Delta y}\right)^2\right]$$

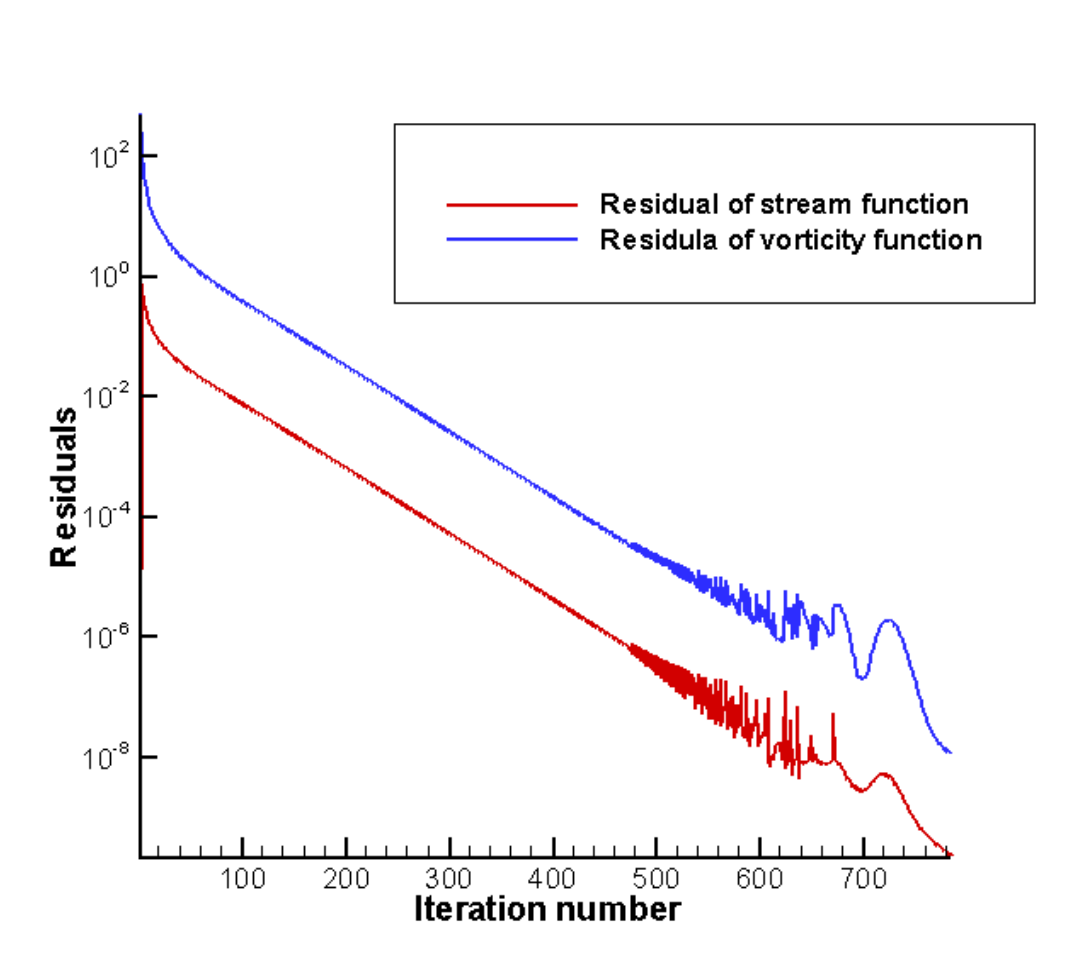
2nd order central difference scheme used for all spatial derivatives

Example (lid-driven cavity, solution procedure)

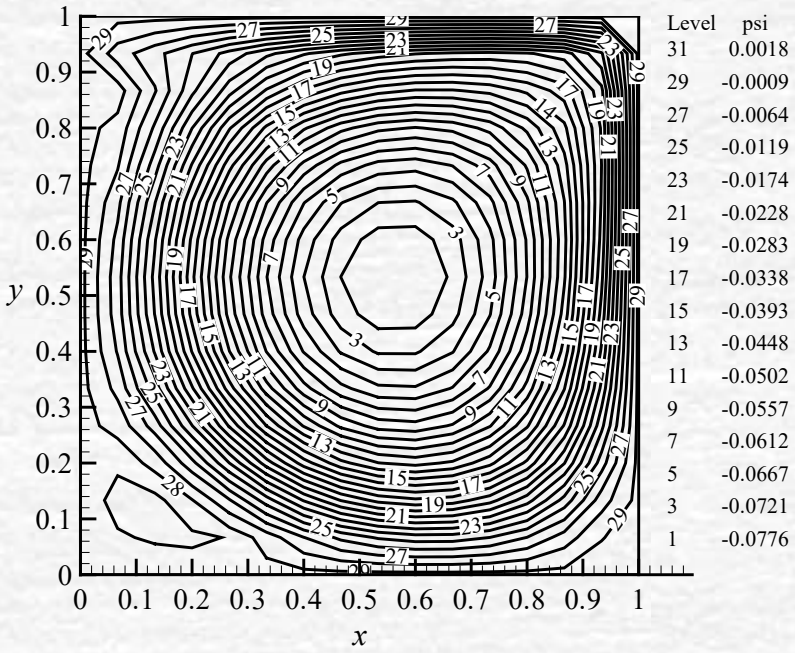
1. Specify the geometry and fluid properties
2. Specify initial conditions (e.g. u=v= ζ = ψ=0).
3. Specify boundary conditions
4. Determine Δt
5. Solve the vorticity transport equation for ζ^{n+1}
6. Solve stream function equation for ψ^{n+1}
7. Solve for u^{n+1} and v^{n+1}
8. Solve the boundary conditions for ζ^{n+1} on the walls
9. Continue marching to time of interest, or until the steady state is reached.

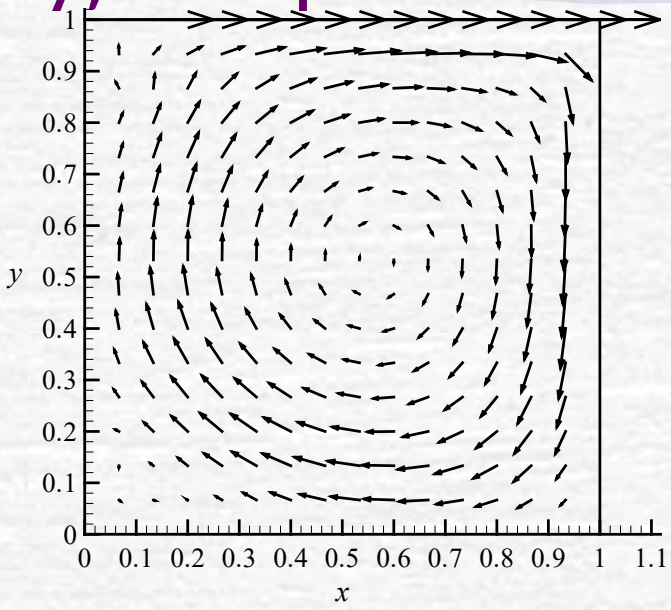
Example (lid-driven cavity, residuals)

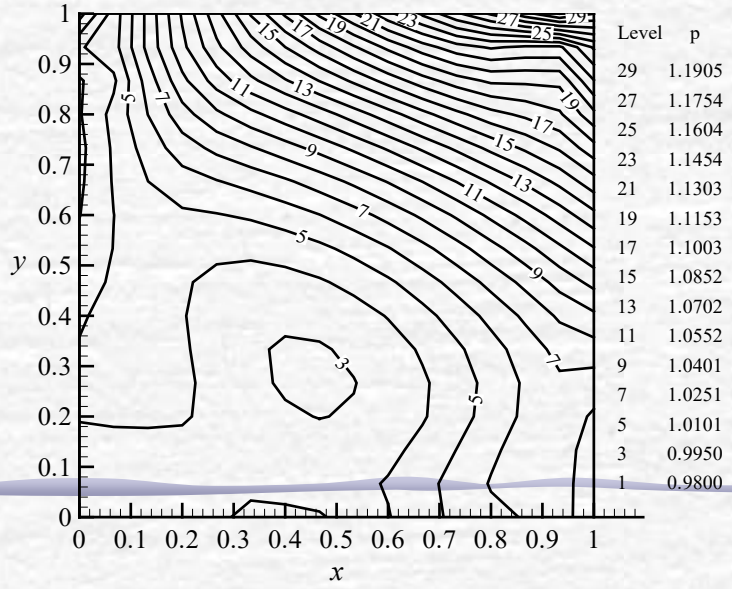
$$\frac{1}{NI \times NJ} \sum_{i=1,j=1}^{i=NI,j=NJ} \left| \zeta_{i,j}^{n+1} - \zeta_{i,j}^{n} \right| \le 1 \times 10^{-8} \quad \text{and} \quad \frac{1}{NI \times NJ} \sum_{i=1,j=1}^{i=NI,j=NJ} \left| \Psi_{i,j}^{n+1} - \Psi_{i,j}^{n} \right| \le 1 \times 10^{-8}$$

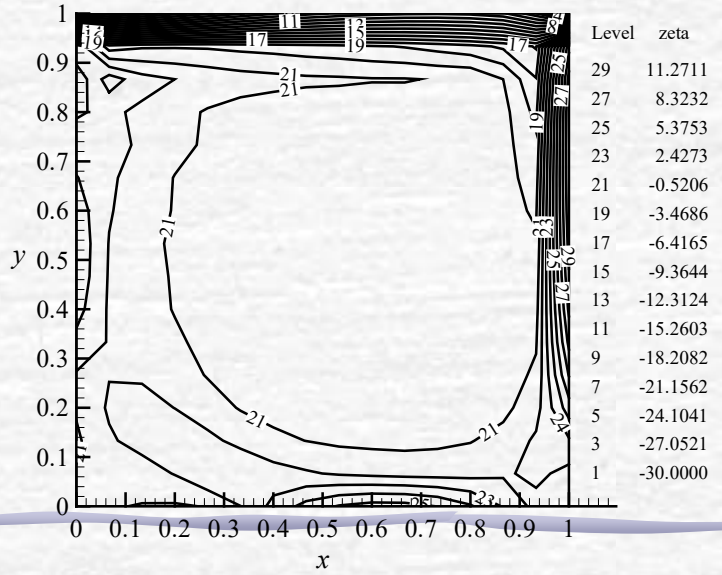


Example (lid-driven cavity, sample results)









Some good books

1. J. H. Ferziger, M. Peric, " **Computational Methods for Fluid Dynamics**," 3rd edition, Springer, 2002.
2. Patric J. Roache, "**Verification and Validation in Computational Science and Engineering**," Hermosa publishers, 1998
3. Frank, M. White, "**Viscous Fluid Flow**," 3rd edition, McGraw-Hill Inc., 2006