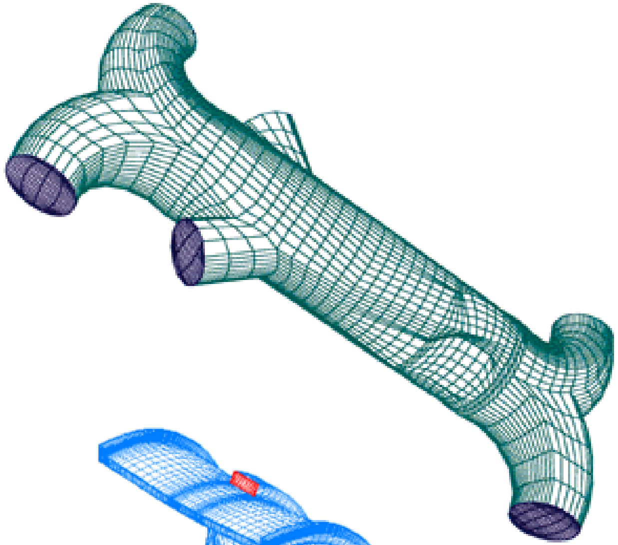
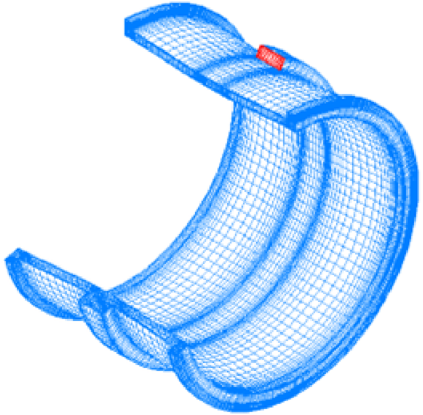
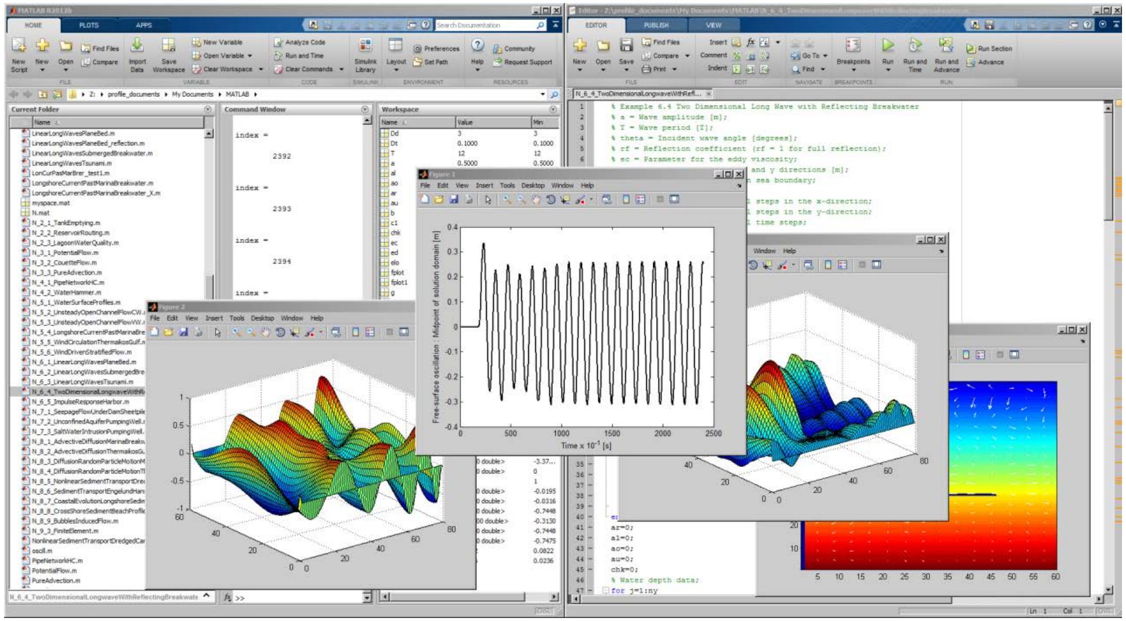
Computational Hydrodynamics

- Chapter 1 : Introduction







$$f(x \pm \Delta x) = f(x) \pm \frac{df}{dx}\Delta x + \frac{d^2 f}{dx^2}\frac{(\Delta x)^2}{2!} \pm \frac{d^3 f}{dx^3}\frac{(\Delta x)^3}{3!} + \ldots$$

Mathematical Modelling

- **Mathematical Models** approximate *cause-response* relations in a wide variety of applications in engineering, physical and social sciences, and economics.

- Depending on the methodology used, mathematical models can be *deterministic / stochastic / a combination of both*.

- **Deterministic Models** are *using equations* based on underlying principles and physical laws.

- **Stochastic Models** are based on *probabilistic* and *statistical* methods.

Mathematical Modelling

- The solution of a mathematical model can be either *analytical* or ***numerical***(focus of this class).

- **Analytical Solutions** are limited only to simple(or idealized) problems, and are presented as *closed-form* or *open-form* (series with infinite number of terms) solutions.

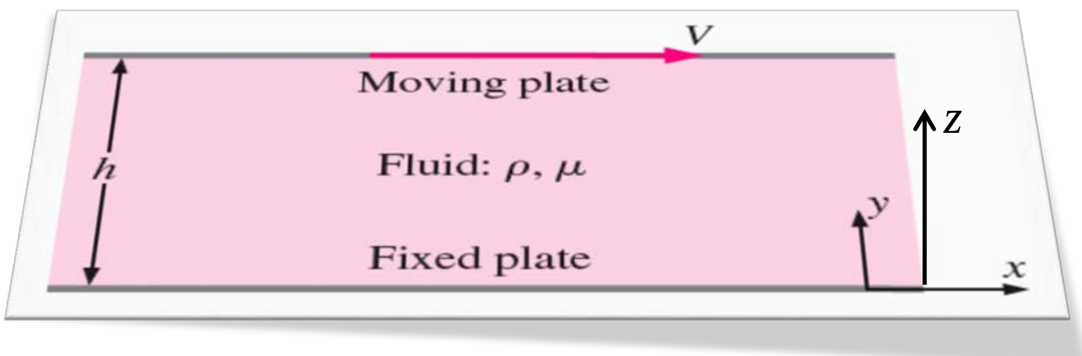
- **Numerical Solutions** are approximate(or alternative) of analytical solution and require the usage of a computer.

Analytical Solutions

- **<u>Example : Fully developed Couette Flow</u>**

- For the given geometry and BC's, calculate the velocity and pressure fields, and estimate the shear force per unit area acting on the bottom plate

- Step 1: Geometry, dimensions, and properties



Analytical Solutions

- Step 2: Assumptions and BC's

- Assumptions 1. Plates are infinite in x and z (no edge effects)
  2. Flow is steady, $\partial/\partial t = 0$
  3. Parallel flow, V=0
  4. Incompressible, Newtonian, laminar, constant properties
  5. No pressure gradient in x-direction
  6. 2D, W=0, $\partial/\partial z = 0$
  7. Gravity acts in the -z direction, $\vec{g} = -g\vec{k}, g_z = -g$

- Boundary conditions 1. Bottom plate (y=0) : u=0, v=0, w=0
  2. Top plate (y=h) : u=V, v=0, w=0

Analytical Solutions

- Step 3: Simplify

Note: these numbers refer to the assumptions on the previous slide

Continuity

$$\frac{\partial U}{\partial x} + \underbrace{\cancel{\frac{\partial V}{\partial y}}}_{3} + \underbrace{\cancel{\frac{\partial W}{\partial z}}}_{6} = 0$$

$$\boxed{\frac{\partial U}{\partial x} = 0}$$

This means the flow is *"fully developed"* or not changing in the direction of flow

X-momentum

$$\rho\left(\underbrace{\cancel{\frac{\partial U}{\partial t}}}_{2} + \underbrace{\cancel{U\frac{\partial U}{\partial x}}}_{\text{Cont.}} + \underbrace{\cancel{V\frac{\partial U}{\partial y}}}_{3} + \underbrace{\cancel{W\frac{\partial U}{\partial z}}}_{6}\right) = -\underbrace{\cancel{\frac{\partial P}{\partial x}}}_{5} + \underbrace{\cancel{\rho g_x}}_{7} + \mu\left(\underbrace{\cancel{\frac{\partial^2 U}{\partial x^2}}}_{\text{Cont.}} + \frac{\partial^2 U}{\partial y^2} + \underbrace{\cancel{\frac{\partial^2 U}{\partial z^2}}}_{6}\right)$$

$$\boxed{\frac{d^2 u}{dy^2} = 0}$$

Analytical Solutions

- Step 3: Simplify, (cont'd)

Y-momentum

$$\rho\left(\overset{\boxed{2,3}}{\cancel{\frac{\partial V}{\partial t}}}+\overset{\boxed{3}}{\cancel{U\frac{\partial V}{\partial x}}}+\overset{\boxed{3}}{\cancel{V\frac{\partial V}{\partial y}}}+\overset{\boxed{3,6}}{\cancel{W\frac{\partial V}{\partial z}}}\right)=-\frac{\partial P}{\partial y}+\overset{\boxed{7}}{\cancel{\rho g_y}}+\mu\left(\overset{\boxed{3}}{\cancel{\frac{\partial^2 V}{\partial x^2}}}+\overset{\boxed{3}}{\cancel{\frac{\partial^2 V}{\partial y^2}}}+\overset{\boxed{3}}{\cancel{\frac{\partial^2 V}{\partial z^2}}}\right)$$

$$\frac{\partial p}{\partial y}=0 \quad \Rightarrow \quad \boxed{p=p(z)}$$

Z-momentum

$$\rho\left(\overset{\boxed{2,6}}{\cancel{\frac{\partial W}{\partial t}}}+\overset{\boxed{6}}{\cancel{U\frac{\partial W}{\partial x}}}+\overset{\boxed{6}}{\cancel{V\frac{\partial W}{\partial y}}}+\overset{\boxed{6}}{\cancel{W\frac{\partial W}{\partial z}}}\right)=-\frac{\partial P}{\partial z}+\overset{\boxed{7}}{\cancel{\rho g_z}}+\mu\left(\overset{\boxed{6}}{\cancel{\frac{\partial^2 W}{\partial x^2}}}+\overset{\boxed{6}}{\cancel{\frac{\partial^2 W}{\partial y^2}}}+\overset{\boxed{6}}{\cancel{\frac{\partial^2 W}{\partial z^2}}}\right)$$

$$\frac{\partial p}{\partial z}=\rho g_z \quad \Rightarrow \quad \boxed{\frac{dp}{dz}=-\rho g}$$

Analytical Solutions

- Step 4: Integrate

X-momentum

$$\frac{d^2u}{dy^2} = 0 \xrightarrow{\text{integrate}} \frac{du}{dy} = C_1 \xrightarrow{\text{integrate}} \boxed{u(y) = C_1 y + C_2}$$

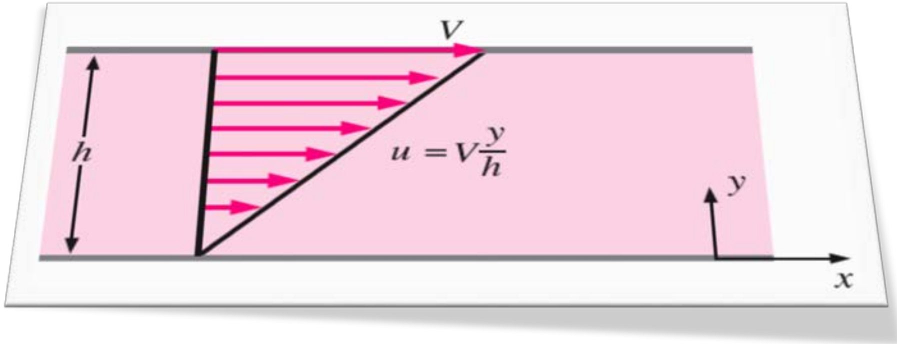
Z-momentum

$$\frac{dp}{dz} = -\rho g \xrightarrow{\text{integrate}} \boxed{p = -\rho g z + C_3}$$

Analytical Solutions

- Step 5: Apply BC's
  - $y=0,\ u=0=C_1\times 0 + C_2 \Rightarrow C_2 = 0$
  - $y=h,\ u=V=C_1 h \Rightarrow C_1 = V/h$
  - This gives

$$\boxed{u(y) = V\frac{y}{h}}$$



  - For pressure, no explicit BC, therefore C_3 can remain an arbitrary constant (recall only ∇P appears in NSE).
    - Let $p = p_0$ at $z = 0$ (C_3 renamed p_0)

$$\boxed{p(z) = p_0 - \rho g z}$$

1. Hydrostatic pressure
2. Pressure acts independently of flow

Analytical Solutions

- Step 6: Verify solution by back-substituting into differential equations
  - Given the solution (u,v,w)=(Vy/h, 0, 0)

$$\frac{\partial u}{\partial x} = 0, \frac{\partial v}{\partial y} = 0, \frac{\partial w}{\partial z} = 0$$

  - Continuity is satisfied

0 + 0 + 0 = 0

  - X-momentum is satisfied

$$\rho\left(\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} + W\frac{\partial U}{\partial z}\right) = -\frac{\partial P}{\partial x} + \rho g_x + \mu\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2}\right)$$

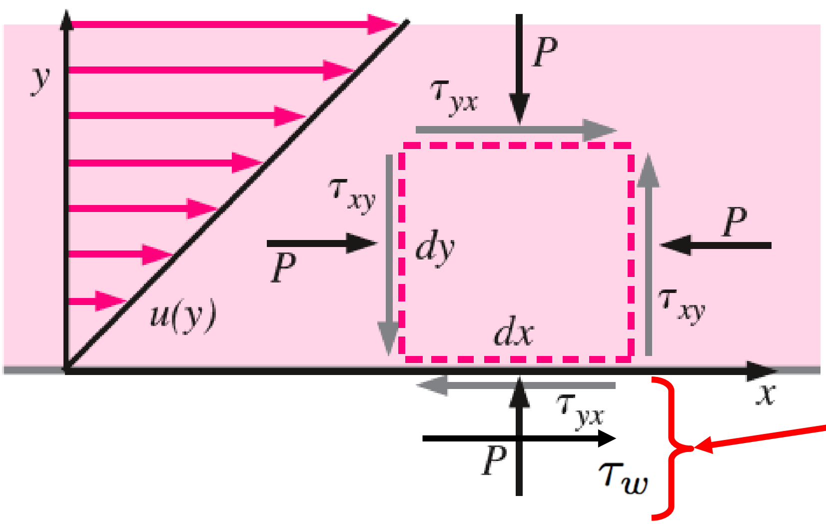
$$\rho\left(0 + V\frac{y}{h}\cdot 0 + 0 \cdot V/h + 0 \cdot 0\right) = -0 + \rho \cdot 0 + \mu\left(0 + 0 + 0\right)$$

$$0 = 0$$

Analytical Solutions

- Finally, calculate shear force on bottom plate

$$\tau_{ij} = \begin{pmatrix} 2\mu\frac{\partial U}{\partial x} & \mu\left(\frac{\partial U}{\partial y}+\frac{\partial V}{\partial x}\right) & \mu\left(\frac{\partial U}{\partial z}+\frac{\partial W}{\partial x}\right) \\ \mu\left(\frac{\partial V}{\partial x}+\frac{\partial U}{\partial y}\right) & 2\mu\frac{\partial V}{\partial y} & \mu\left(\frac{\partial V}{\partial z}+\frac{\partial W}{\partial y}\right) \\ \mu\left(\frac{\partial W}{\partial x}+\frac{\partial U}{\partial z}\right) & \mu\left(\frac{\partial W}{\partial y}+\frac{\partial V}{\partial z}\right) & 2\mu\frac{\partial W}{\partial z} \end{pmatrix} = \begin{pmatrix} 0 & \mu\frac{V}{h} & 0 \\ \mu\frac{V}{h} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$



Shear force per unit area acting on the wall

$$\boxed{\frac{\vec{F}}{A} = \tau_w = \mu\frac{V}{h}\hat{i}}$$

Note that τ_w is equal and opposite to the shear stress acting on the fluid τ_{yx} (Newton's third law).

Computational Hydraulics

- **Numerical Analysis** is the branch of mathematics that develops and analyzes methodologies for *numerical solutions*.

- **Numerical Models** are mathematical models that utilize numerical analysis and computers to provide numerical solutions. Numerical models are relatively easy to develop and very easy to modify and adapt to different scenarios.

- **Computational Hydraulics** or **Computational Hydrodynamics** is the discipline that seeks solutions of hydraulic/hydrodynamic problems by means of numerical models.

Computational Hydraulics

- Computational Hydraulics is part of the broader discipline of *Computational Fluid Dynamics* (CFD).

- Nowadays, *Computational Hydraulics* simulations have replace, almost exclusively, the application of *Physical Modelling*.

- **Physical Models** involve the study of engineering applications by using small-scale replicas of the *prototype*. Physical Models are expensive to construct and maintain, and once built are very difficult to modify.

Numerical Algorithms

- **Numerical Algorithm** describes a pre-determined sequence of basic *arithmetic* and *logical* operations, for the solution of a mathematical problem:

$$Y = L(X)$$

where X is a set of *input data*, Y is a set of *output data* (the solution in numerical form), and L is an *operator.*

- Problems in hydraulics can be described mainly by means of *partial differential equations* (PDEs). These are, most of the time, *linear* or *linearized homogeneous 2nd order PDEs*.

- The problems that can be described and solved by using PDEs include but are not limited to flows in: *Pressurized Conduits*; *Open Channels*; *Waves Mechanics*; *Coastal Hydraulics*; *Pollutant Transport*; and *Sedimentation Processes*.

- Those equations in their general form can only be solved by using some numerical algorithm.

Well-Posed Mathematical Models

- A mathematical model of a physical system is ***well-posed*** under the following conditions:

  - The solution algorithm produces a solution for all sets of *input data*, under specified conditions and limitations.

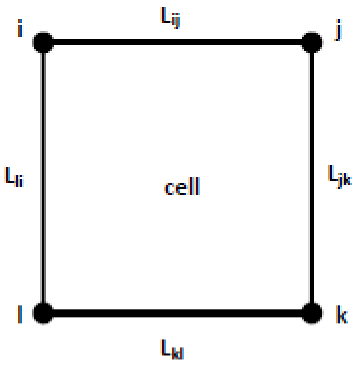
  - The produced solution is *unique*, i.e. only one solution output corresponds to each set of *input data.*

  - The *output* has to be related to the *input*, (via a *"Lipschitz" condition*) - i.e. each infinitesimal change of input values δX, results into a finite change of output δY.

- Note that for a mathematical model to be *well-posed*, it is necessary that the *governing PDEs*, the *auxiliary data* (i.e. boundary and initial conditions) and the *numerical algorithm* are all well-posed.

Discretization and Numerical Solution of Mathematical Models

- The solution domain is *discretized* appropriately into one-, two- or three-dimensional "*cells*" (or "grids") and the solution is approximated at: the corner nodes (i,j,k,l), or the sides (L_{ij}, L_{jk}, L_{kl}, L_{li}) or the interior of the cell. Some numerical methods (e.g. Finite Elements) can handle cells of arbitrary shape.



- The independent variables in space (x, y, z) and time (t) are also discretized into small steps Δx, Δy, Δz, and Δt.

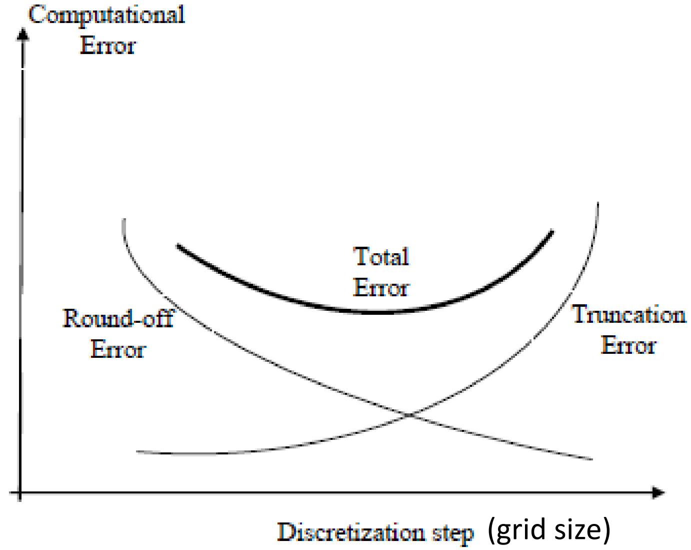
- The *Differentials* are approximated by *Differences* and the *Differential Equations* are converted into *Difference Equations*.

Truncation and Round-off Errors

- The *truncation error* is introduced by the discretization of the solution domain. This error increases with increasing discretization steps.

- The *round-off* error is the result of arithmetic operations. This error increases with decreasing discretization steps.

- The *total computational error* is the summation of both the *truncation* and the *round-off* errors.



(grid size)

Numerical Solutions

- Any numerical solution method needs to satisfy three conditions:

- It has to be ***consistent***, i.e. the approximation used for the derivatives has to be correct, according to the numerical method used.

- It has to be ***convergent***, i.e. the numerical solution must tend asymptotically towards the analytical solution, as the discretization steps (Δx, Δy, Δz, Δt) tend to zero. A *non-convergent* method is of no practical use.

- It has to be ***numerically stable***. For stable methods, the inevitably introduced errors during the solution procedure do not increase indefinitely, but *decay* and become negligible after some solution steps.

Consistency + Convergence + Stability = **Good Model**

Taylor Series

- A function f(x) can be expanded into a neighbouring point by means of the Taylor series

$$f(x \pm \Delta x) = f(x) \pm \frac{df}{dx}\Delta x + \frac{d^2 f}{dx^2}\frac{(\Delta x)^2}{2!} \pm \frac{d^3 f}{dx^3}\frac{(\Delta x)^3}{3!} + \ldots$$

where Δx is a very small number and n! is the factorial (n! = 1×2×3×... ×n).

- Since Δx is a very small number, $(\Delta x)^m$ -> 0 for any exponent m that is a positive integer greater than 1.

- That leads to *truncate*d Taylor Series

$$f(x \pm \Delta x) = f(x) \pm \frac{df}{dx}\Delta x + O[(\Delta x)^2]$$

where O[ ] is the order of the *truncation error*.

Truncation Errors

- From the previous equation, *truncation error* (T_E)can be expressed as

$$T_E = \frac{(\Delta x)^{n+1}}{(n+1)!}\frac{d^{(n+1)}f}{dx^{(n+1)}}$$

- How to reduce truncation errors?
  - (a) Reduce grid spacing, use smaller Δx
  - (b) Increase order of accuracy, use larger n

The Finite Differences (F.D.) Method

- The Finite Differences Method is a classical method of *Numerical Analysis* where *differentials* are approximated by *differences* using the truncated Taylor series as:

- Forward or Upwind F.D.
- 1^{rst} derivative - 1^{rst} order

$$\frac{df}{dx} = \frac{f(x+\Delta x) - f(x)}{\Delta x} + O(\Delta x)$$

- Backward or Downwind F.D.
- 1^{rst} derivative - 1^{rst} order

$$\frac{df}{dx} = \frac{f(x) - f(x-\Delta x)}{\Delta x} + O(\Delta x)$$

- Central F.D.
- 1^{rst} derivative - 2^{nd} order

$$\frac{df}{dx} = \frac{f(x+\Delta x) - f(x-\Delta x)}{2(\Delta x)} + O\left[(\Delta x)^2\right]$$

- Central F.D.
- 2^{nd} derivative - 2^{nd} order

$$\frac{d^2f}{dx^2} = \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{(\Delta x)^2} + O\left[(\Delta x)^2\right]$$

The Finite Differences Solution Schemes

- After replacement of the differentials by "FINITE" differences, the resulting Finite Differences algebraic equation(s) is known as the *Finite Differences Scheme* or *Algorithm*.

- Depending on the procedure for the solution of the algebraic equation(s) corresponding to all the points in the solution domain, the finite difference scheme used, is characterized as:

- Explicit, when the deriving algebraic equation(s) can be solved independently, or

- Implicit, when those equations need to be solved simultaneously, as a system of algebraic equations.

- *Explicit schemes* are much easier to use but commonly are subject to certain restrictions due to *instability criteria*.

- Note that a Finite Difference Scheme may not always lead to a physically correct and operationally acceptable solution .

Background Knowledge Requirements

- Some fundamental knowledge of the following topics is necessary for taking this class, as follows
- Calculus including differential equations.
- Introductory numerical analysis.
- Any computer language (e.g., FORTRAN, C++, BASIC, ***MATLAB***, etc.)
- Basic understanding of mass and momentum conservation principles as apply to fluid flow.
- In addition, all the students must have access to MATLAB R2015a (or earlier version) on campus in order to run the computer models, and to conduct the suggested exercises throughout the class.

Brief Introduction of MATLAB
Current Folder-Editor-Command Window-Workspace

- The *computer programs*, *subroutines*, and *data files* are all listed in the “Current Folder” tile (left).

Editor

Workspace

Current Folder

Command Window

Brief Introduction of MATLAB
Run-Output windows

- All programs and data related to MATLAB should be stored in the MATLAB folder under *Libraries-Documents* (Microsoft platform).

- Once a program is selected from the "*Current Folder*" tile (right), a new window the "*Editor*" appears.
- The program can be executed by clicking on the "*Run*" icon on the top of the "*Editor*" window.

- The program will run on the "*Command Window*" (center) and the results of the various variables will appear on the "*Workplace*" tile (right).

- Any plot will be shown on a separate "*Figure*" window.

Output Windows

Solving the PDE's using Numerical Methods

- The are a number of methods for the solution of the governing PDE's on the discretized domain

- The most important discretization methods are:
  - Finite Difference Method (FDM)
  - Finite Volume Method (FVM)
  - Finite Element Method (FEM)

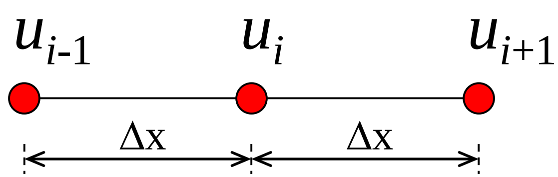
Finite Difference Method - Introduction

- Oldest method for the numerical solution of PDE's

- Procedure:
  - Start with the conservation equation in differential form
  - Solution domain is covered by grid
  - Approximate the differential equation at each grid point by approximating the partial derivatives from the nodal values of the function giving one algebraic equation per grid point
  - Solve the resulting algebraic equations for the whole grid. At each grid point you solve for the unknown variable value and the value of it's neighboring grid points

Finite Difference Method - Concept

- The finite difference method is based on the Taylor series expansion about a point, x



$$u_{i-1} = u_i - \left(\frac{\partial u}{\partial x}\right)_i \Delta x + \left(\frac{\partial^2 u}{\partial x^2}\right)_i \frac{\Delta x^2}{2} + \text{H.O.T.}$$

where u_{i-1} is defined as $u(x-\Delta x)$

$$u_{i+1} = u_i + \left(\frac{\partial u}{\partial x}\right)_i \Delta x + \left(\frac{\partial^2 u}{\partial x^2}\right)_i \frac{\Delta x^2}{2} + \text{H.O.T.}$$

where u_{i+1} is defined as $u(x+\Delta x)$

Subtracting the two eqns above gives

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i+1} - u_{i-1}}{2\Delta x} + O\left(\Delta x^2\right)$$

Adding the two eqns above gives

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} + O\left(\Delta x^2\right)$$

Finite Difference Method - Application

- Consider the steady 1-dimensional convection/diffusion equation:

$$\frac{\partial(\rho u \phi)}{\partial x} = \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right)$$

- From the Taylor series expansion, get

$$-\left[\frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right)\right]_i \approx -\frac{\left(\Gamma \frac{\partial \phi}{\partial x}\right)_{i+\frac{1}{2}} - \left(\Gamma \frac{\partial \phi}{\partial x}\right)_{i-\frac{1}{2}}}{\Delta x} = -\frac{1}{\Delta x}\left(\Gamma \frac{\phi_{i+1} - \phi_i}{\Delta x} - \Gamma \frac{\phi_i - \phi_{i-1}}{\Delta x}\right)$$

$$\frac{\partial(\rho u \phi)}{\partial x} \approx \rho u \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x}$$

Finite Difference Method – Application(Cont'd)

- Substitute the discrete forms of the differentials to get:

$$\rho u \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} - \frac{1}{\Delta x}\left(\Gamma \frac{\phi_{i+1} - \phi_i}{\Delta x} - \Gamma \frac{\phi_i - \phi_{i-1}}{\Delta x} \right) = 0$$

$$\boxed{\frac{2\Gamma}{\Delta x}\phi_i = \left(-\frac{\rho u}{2} + \frac{\Gamma}{\Delta x} \right)\phi_{i+1} + \left(\frac{\rho u}{2} + \frac{\Gamma}{\Delta x} \right)\phi_{i-1}}$$

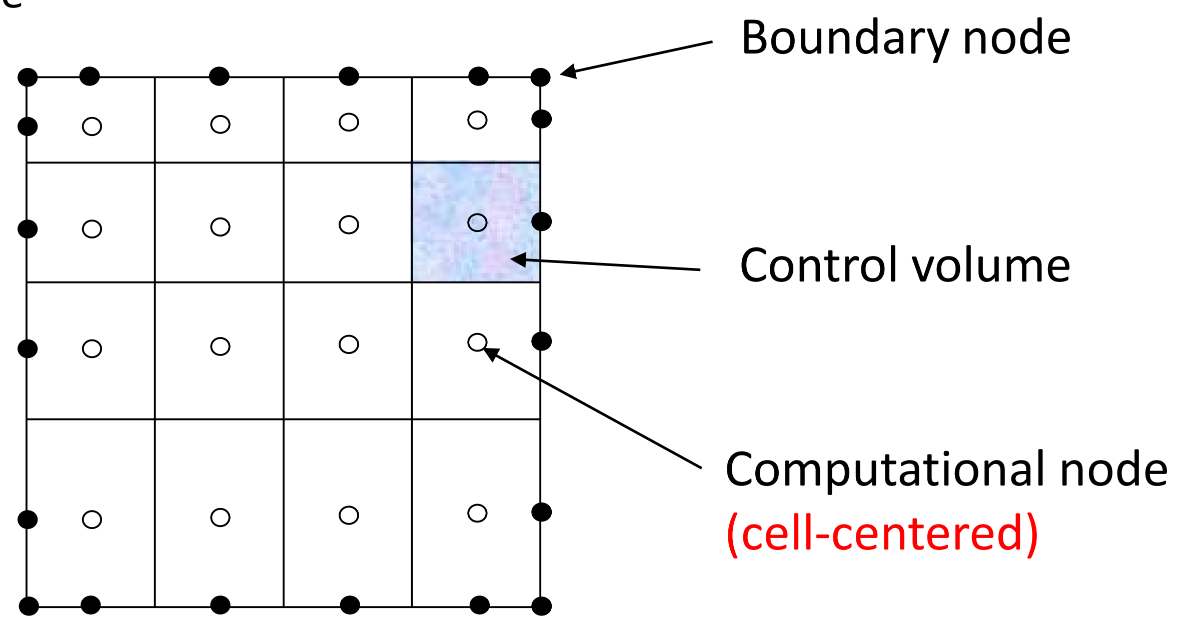
Algebraic form of PDE

Finite Difference Method – Summary

- Discretized the one-dimensional convection/diffusion equation
- The derivatives were determined from a Taylor series expansion
- Advantages of FDM: simple and effective on structured grids
- Disadvantages of FDM: *conservation is not enforced* unless with special treatment, restricted to simple geometries

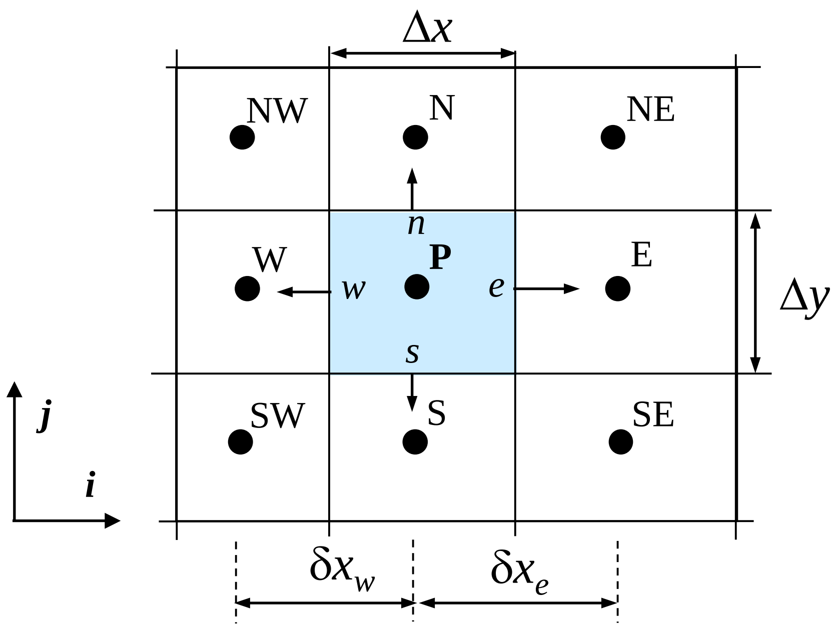
Finite Volume Method - Introduction

- Using Finite *Volume* Method, the solution domain is subdivided into a finite number of small control volumes by a grid
- The grid defines to boundaries of the control volumes while the computational node lies at the center of the control volume
- The advantage of FVM is that the integral conservation is satisfied exactly over the control volume



<Grid definition>

Finite Volume Method – Typical Control Volume



- The net flux through the control volume boundary is the sum of integrals over the four control volume faces (six in 3D). The control volumes do not overlap
- The value of the integrand is not available at the control volume faces and is determined by interpolation *(Why?)*

Finite Volume Method – Application

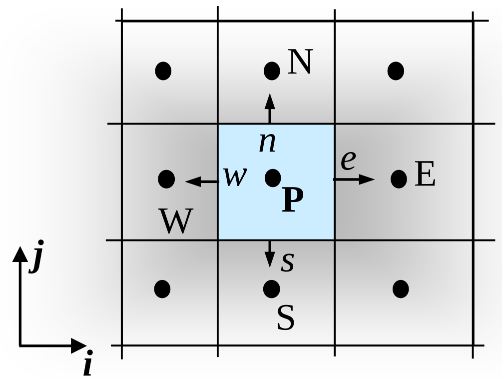
- Consider the one-dimensional convection/diffusion equation
- The finite volume method (FVM) uses the integral form of the conservation equations over the control volume: *(please recall what FDM does for the equation)*

$$\oint_V \left[\frac{\partial(\rho u \phi)}{\partial x} - \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x} \right) \right] dV = \oint_V S dV$$

- Integrating the above equation in the *x*-direction across faces *e* and *w* of the control volume and leaving out the source term gives

$$(\rho u \phi)_e - (\rho u \phi)_w = \left(\Gamma \frac{\partial \phi}{\partial x} \right)_e - \left(\Gamma \frac{\partial \phi}{\partial x} \right)_w$$

- The values of ϕ at the faces *e* and *w* are needed



Finite Volume Method – Interpolation

- Using a piecewise-linear interpolation between control volume centers gives

$$(\rho u\phi)_e - (\rho u\phi)_w = \left(\Gamma\frac{\partial\phi}{\partial x}\right)_e - \left(\Gamma\frac{\partial\phi}{\partial x}\right)_w$$

$$\tfrac{1}{2}(\rho u)_e(\phi_E+\phi_P) - \tfrac{1}{2}(\rho u)_w(\phi_P+\phi_W) = \frac{\Gamma_e(\phi_E-\phi_P)}{(\delta x)_e} - \frac{\Gamma_w(\phi_P-\phi_W)}{(\delta x)_w}$$

where $\left.\begin{aligned}\phi_e &= \tfrac{1}{2}(\phi_E+\phi_P)\\ \phi_w &= \tfrac{1}{2}(\phi_P+\phi_W)\end{aligned}\right\}$

- linear interpolation between nodes
- face is midway between nodes
- equivalent to *Central Difference Scheme* (*CDS*)

$$\boxed{\left(\frac{2\Gamma}{\delta x}\right)\phi_P = \left(\tfrac{1}{2}\rho u_w + \left(\frac{\Gamma}{\delta x}\right)_w\right)\phi_W + \left(-\tfrac{1}{2}\rho u_e + \left(\frac{\Gamma}{\delta x}\right)_e\right)\phi_E}$$

- ***Under assumption of continuity, discrete form of PDE from FVM is identical to FDM***

Finite Volume Method – Interpolation

- The piecewise-linear or CDS interpolation may give rise to (oscillatory) numerical errors. CDS was used only as an example of discretization and is inappropriate for most convection/diffusion flows.

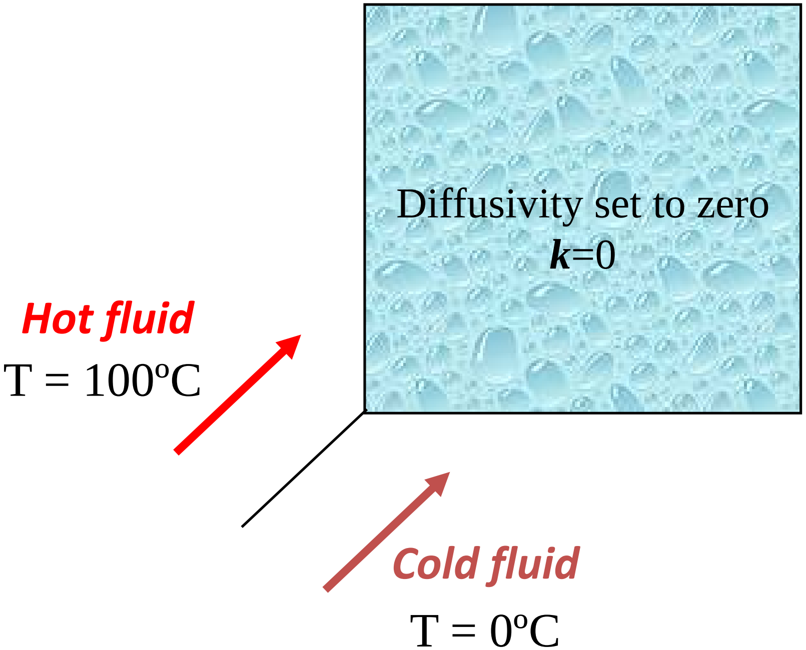
- A large number of interpolation techniques are improvements on the CDS. Some of these, in increasing level of accuracy, are:
  - First-Order Upwind Scheme
  - Power Law Scheme
  - Second-Order Upwind Scheme
  - Higher Order
    - Blended Second-Order Upwind/Central Difference
    - Quadratic Upwind Interpolation (QUICK)
    - MUSCL(Monotonic Upstream-Centered Scheme for Conservation Law)

Sources of Numerical Errors – FDM & FVM

- Discretization Errors from inexact interpolation of nonlinear profile (FVM)
- Truncation Errors due to exclusion of Higher Order Terms (FDM)
- Domain discretization not well resolved to capture flow physics
- Artificial or False Diffusion due to interpolation method and grid

Example of Numerical Errors – False Diffusion

- False diffusion is numerically introduced diffusion and arises in convection dominant flows, i.e., high *Pe* number flows (*Pe = advection rate/diffusion rate*)
- Consider the problem below:



◆ If there is no false diffusion, the temperature along the diagonal will be either 100ºC or 0ºC, exactly.

◆ False diffusion will occur due to the oblique flow direction and non-zero gradient of temperature in the direction normal to the flow

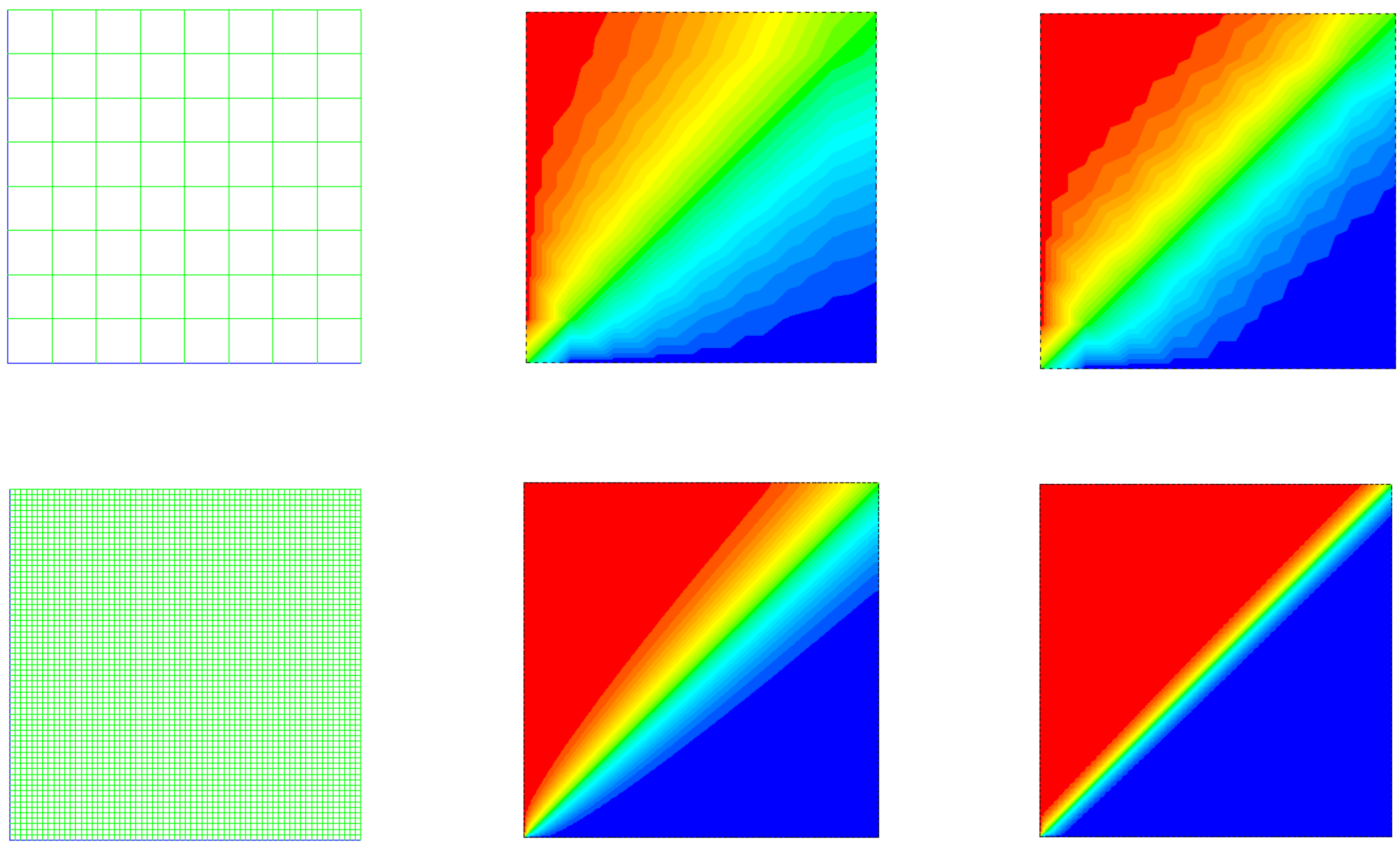
◆ Grid refinement coupled with a higher-order interpolation scheme will minimize the false diffusion *(see the next slide)*

Example of Numerical Errors – False Diffusion(Cont'd)

First-order Upwind *Second-order Upwind*

8 x 8

64 x 64



Finite Volume Method – Summary

- The FVM uses the integral conservation equation applied to control volumes which subdivide the solution domain, and to the entire solution domain
- The variable values at the faces of the control volume are determined by interpolation. False diffusion can arise depending on the choice of interpolation scheme
- The grid must be refined to reduce "smearing" of the solution as shown in the last example
- Advantages of FVM: Integral conservation is exactly satisfied, Not limited to grid type (structured or unstructured, Cartesian or body-fitted)

Finite Volume Method – Homework #1

- One governing equation can be written in either conservative or non-conservative form as shown below.

- Conservative form :

$$\frac{\partial \mathrm{F}}{\partial x} = \mathbf{0}$$

- Non-conservative form :

$$G\frac{\partial \mathrm{F}}{\partial x} = \mathbf{0}$$

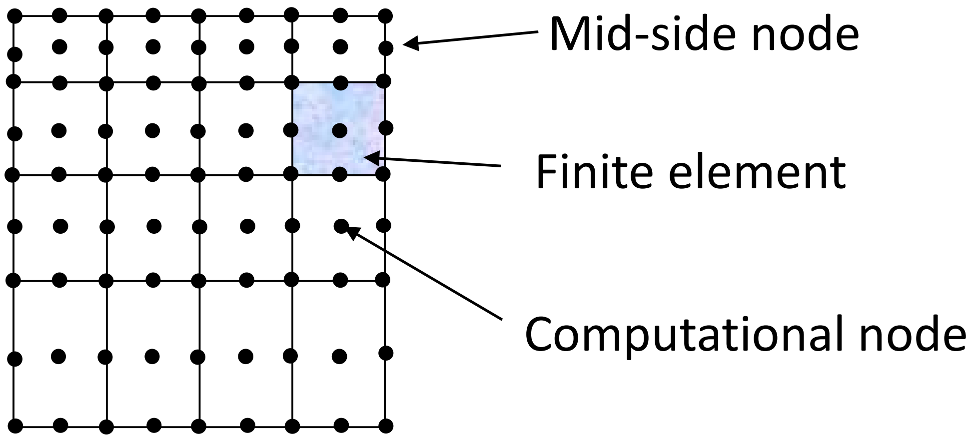
where F and G are unknowns to be discretized.

- Q1) Please prove that when using a simple numerical scheme(e.g., FDM) the equation of conservative form retains conservative property while the one of non-conservative form cannot.

- Q2) What will be the benefit through the conservative property in the computational hydraulics?

Finite Element Method – Introduction

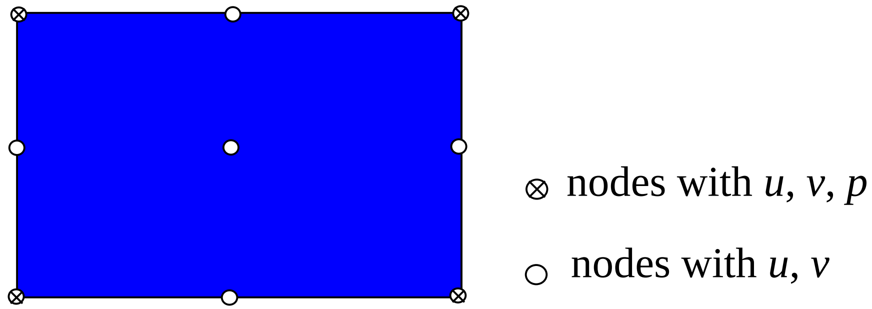
- Using Finite Element Method, the solution domain is subdivided into a finite number of small elements by a grid
- The grid defines to boundaries of the elements and location of nodes — for higher-order elements, there can be mid-side nodes also
- FEM uses multi-dimensional shape functions which afford geometric flexibility and limit false diffusion



<Element definition>

Finite Element Method – Typical Element

- 9-noded quadrilateral



- Within each element, the velocity and pressure fields are approximated by:

$$u = \sum_{i=1}^{r} u_i \varphi_i \quad v = \sum_{i=1}^{r} v_i \varphi_i \quad p = \sum_{i=1}^{s} p_i \psi_i$$

where u_i , v_i , p_i are the nodal point unknowns and ϕ_i and ψ_i are **interpolation functions**

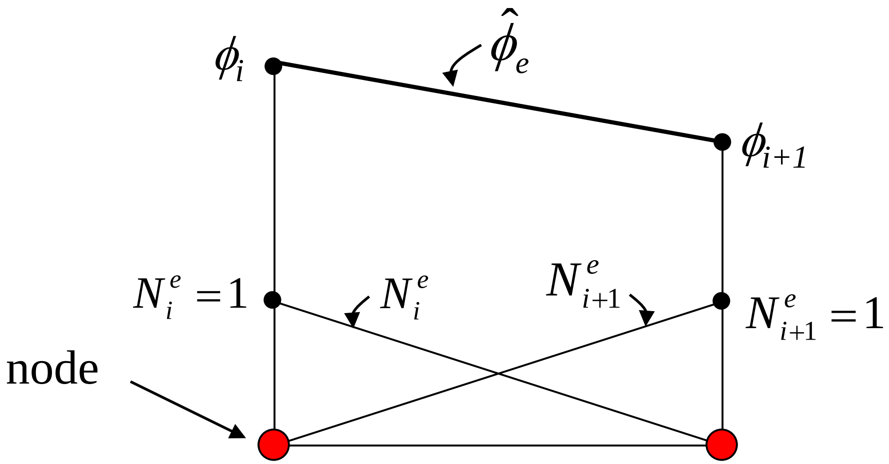
- Quadratic approximation for velocity, linear approximation for pressure required to avoid spurious pressure modes

Finite Element Method – Interpolation

- The solution on an element is represented as:

$$\phi \cong \hat{\phi}_e = \phi_i N_i^e + \phi_{i+1} N_{i+1}^e$$

where N are the basis functions. We choose basis functions that are 1 at one node of the element and 0 at all other the nodes.



Finite Element Method – Application

- Recall the one-dimensional convection/diffusion equation
- Most often, the finite element method (FEM) uses the Method of Weighted Residuals to discretize the equation
- Multiply governing equation by weight function W_i and integrate over the element

$$\int_e W_i \left\{ \rho \left(u \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial x} \left(\Gamma \frac{\partial \phi}{\partial x} \right) \right\} dx = 0$$

- How do we choose the W_i ? For Galerkin FEM, replace W_i by N_i , the shape or basis functions

Finite Element Method – “Weak” Form

- Use integration by parts to obtain the “weak” formulation — involves first derivatives rather than second derivatives

- We can now substitute the interpolation function for ϕ :

$$\phi = \sum_{i=1}^{r} \phi_i N_i$$

and evaluate the required integrals to produce the discrete equation:

$$\mathbf{K}(u)\phi = 0$$

$$\text{where} \quad K^e = \int_e \Gamma \frac{d^2 N_i}{dx^2} dx + \int_e \rho N_i u \frac{dN_i}{dx} dx$$

Finite Element Method – Summary

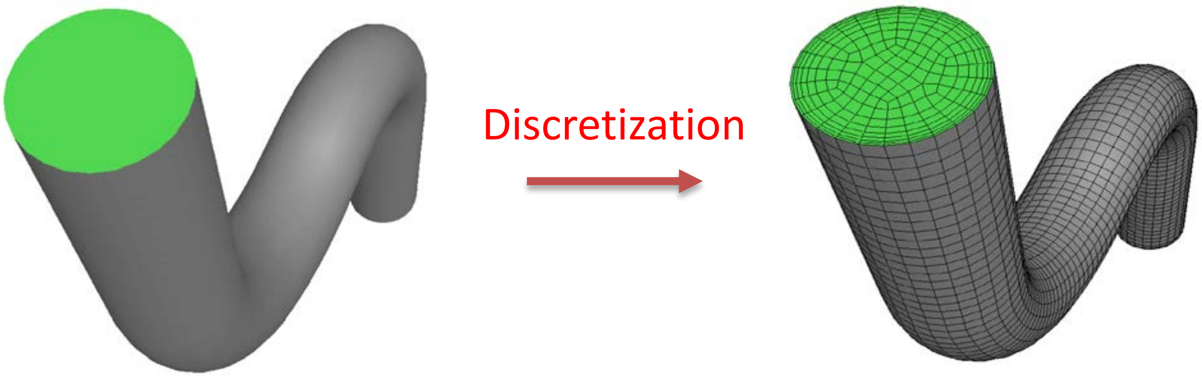
- FEM solves the “weak” form of the governing equations
  - weak form requires continuity of lower order operators only
  - very similar to using the divergence theorem in FVM
- The technique is conservative in a weighted sense (*recall the previous example*)

$$\int_e W_i \left\{ \rho\left(u\frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right)\right\} dx = 0$$

- The weight functions can easily be made multi-dimensional
  - this limits false diffusion

What is discretization?

- Discretization is the method of approximating the differential equations by a system of algebraic equations for the variables at some set of discrete locations in space and time
- The discrete locations are grid/mesh points or cells
- The continuous information from the exact solution of PDE's is replaced with discrete values
- Transforming the physical model into a form in which the equations governing the flow physics can be solved can be referred to as *discretizing the domain*



Continuous domain

Discretized domain

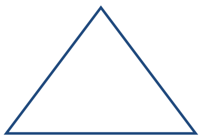
Numerical Grids – Why is a grid needed?

- The grid:
  - is a discrete representation of the geometry of the problem
  - indicates the location on which the flow is solved
  - has a cell group on the boundary zones — where BC’s are applied
  - is called cell/mesh/element depending on the numerical scheme used

- The grid has a significant impact on:
  - rate of convergence (or even *lack* of convergence)
  - solution accuracy
  - CPU time required (efficiency)

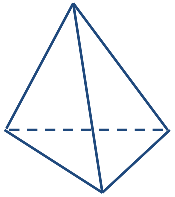
Grid/Cell/Element Types

- Many different cell/element and grid types are available — choice depends on the problem and the solver capabilities
- Different grid/cell/element types

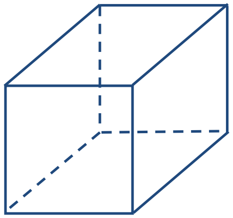


triangle

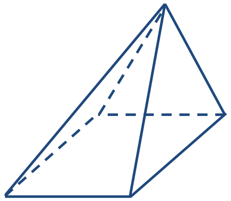
2D rectangular



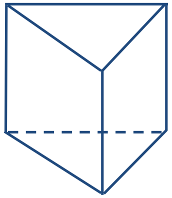
tetrahedron



prism with
quadrilateral base
(hexahedron)



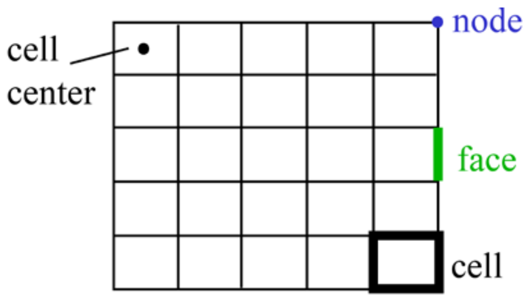
pyramid



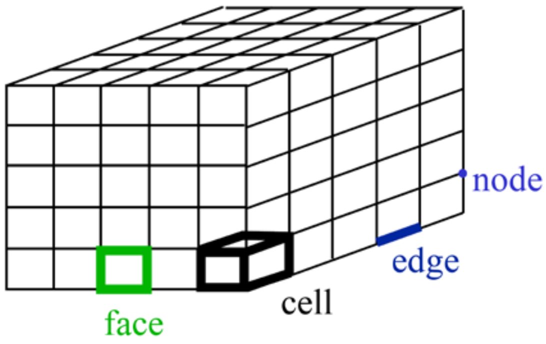
prism with
triangular base

Terminology

- Cell = control volume into which
- domain is broken up.
- Node = grid point.
- Cell center = center of a cell.
- Edge = boundary of a face.
- Face = boundary of a cell.
- Zone = grouping of nodes, faces, and cells:
- – Wall boundary zone.
- – Fluid cell zone.
- Domain = group of node, face and cell zones.



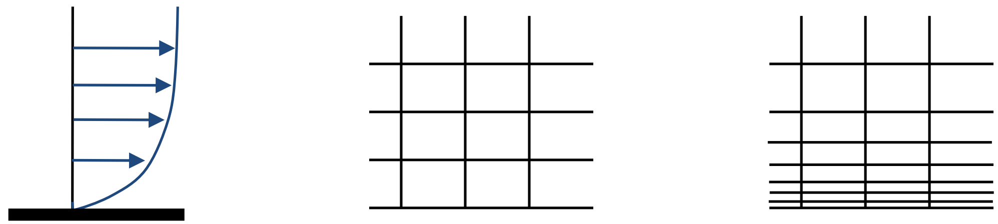
2D computational grid



3D computational grid

Grid Design Guideline : Resolution

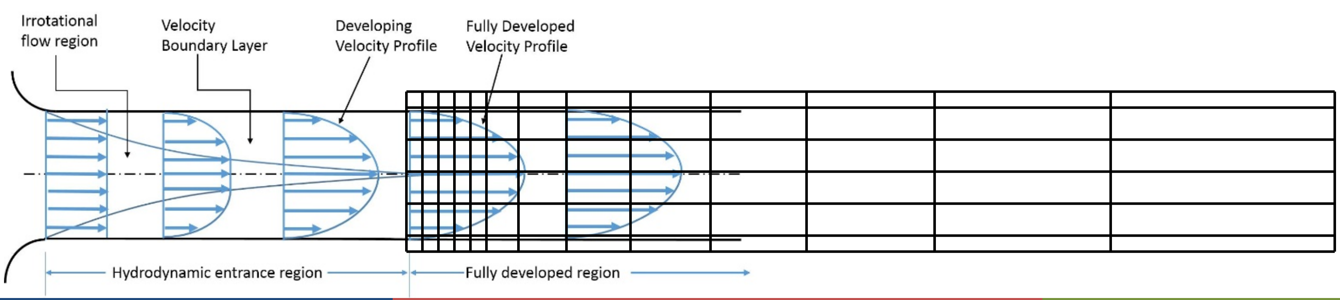
- Flow features should be adequately resolved



flow inadequate better

- Cell aspect ratio (width/height) should be near 1 where flow is multi-dimensional

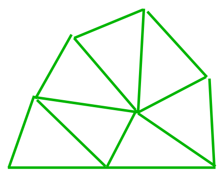
- Quad/hex cells can be stretched where flow is fully-developed and essentially one-dimensional



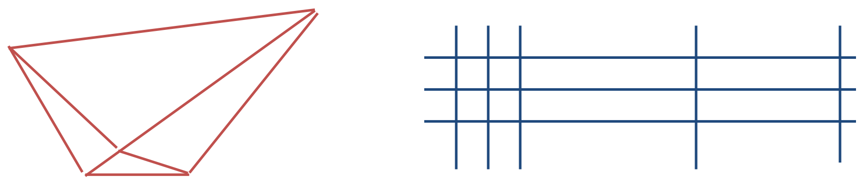
OK!

Grid Design Guideline : Smoothness

- Change in cell/element size should be gradual (smooth)

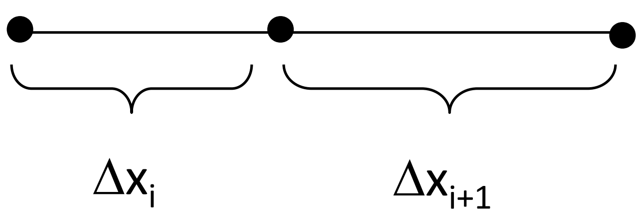


smooth change
in cell size



sudden change
in cell size — AVOIDED!

- Ideally, the maximum change in grid spacing should be less than 20%:



$$\frac{\Delta x_{i+1}}{\Delta x_i} \leq 1.2$$

Grid Design Guideline : Total Cell Count

- More cells can give higher accuracy — downside is increased memory and CPU time (will be a highly demanding computation) Trade-off problem!
- To keep cell count down, use a non-uniform grid to cluster cells only where they're needed
- Design and construction of a quality grid is crucial to the success of the CFD analysis.
- Appropriate choice of grid type depends on:
  - geometric complexity
  - flow field characteristics
  - cell/element types supported by numerical methods

Grid Design Guideline : Total Cell Count

- More cells can give higher accuracy — downside is increased memory and CPU time (will be a highly demanding computation) Trade-off problem!
- To keep cell count down, use a non-uniform grid to cluster cells only where they're needed
- Design and construction of a quality grid is crucial to the success of the CFD analysis.
- Appropriate choice of grid type depends on:
  - geometric complexity
  - flow field characteristics
  - cell/element types supported by numerical methods

Nested Grids (Multi-grids)

- At different sizes of grids can be used for several domains which are connected and nested to the grid of higher level.