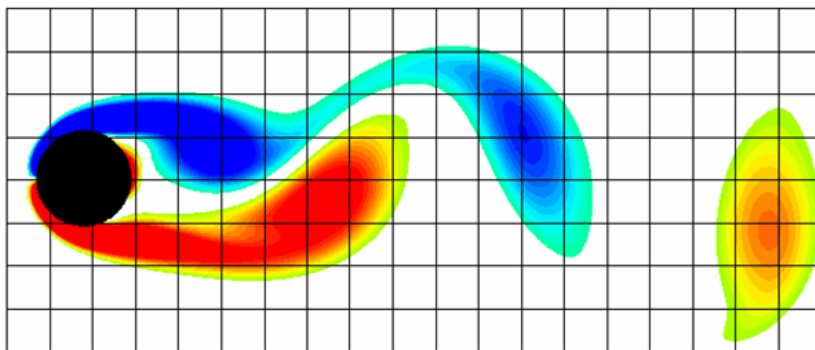
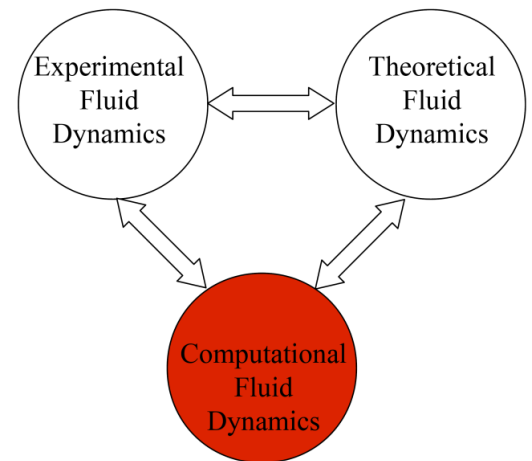
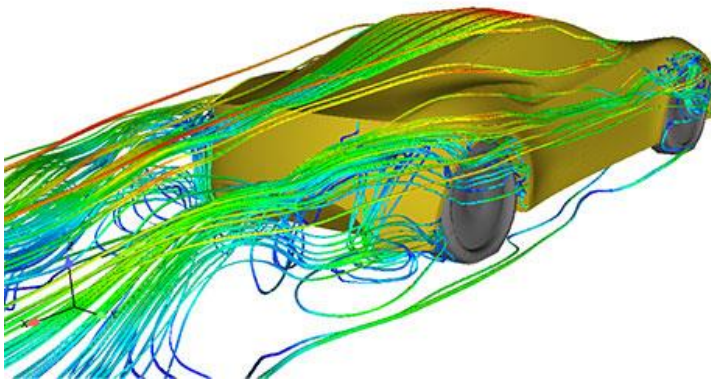


COMPUTATIONAL FLUID DYNAMICS (CFD)

Master 1 ENERGIE



Olivier BOTELLA

LEMTA - Université de Lorraine - CNRS
2 avenue de la Forêt de Haye, TSA 60604
54518 Vandoeuvre (France)

Email : Olivier.Botella@univ-lorraine.fr

Lectures web page : <http://arche.univ-lorraine.fr/course/view.php?id=9396>

What is CFD ?

• **Computational Fluid dynamics (CFD)** is the **science of predicting fluid flows**, heat transfer, mass transfer, chemical reactions and related phenomena **by solving the mathematical equations** that represent physical laws, **using a numerical (computer) process**.

• CFD complements testing and experimentation:

- Reduces the total effort to design a prototype.
- Rule of thumb for airplane design:
1 hour of CFD ~ \$ 1k
1 hour of wind tunnel ~ \$ 10k
1 hour of flight test ~ \$ 100k

• The result of CFD analysis is relevant to :

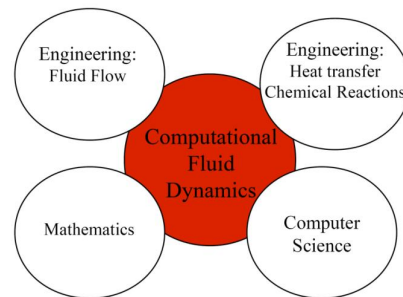
- Understanding complex physical phenomena
- Conceptual study of new designs
- Troubleshooting

• CFD is interdisciplinary. Skills needed :

- Fluid mechanics / engineering science
- Computer science (software, programming language)
- Numerical analysis :
 - Building of numerical methods for fluid flows
 - Analyse of their convergence properties



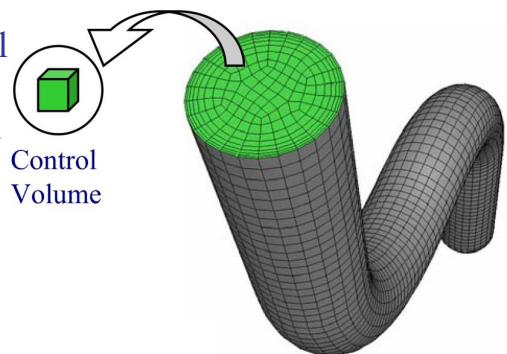
CFD contribution to A380 design



How does CFD work (1/2) ?

- CFD analysis begins with a **mathematical model** of a physical problem.
- Conservation of matter, momentum, and energy (**Navier-Stokes Eqs**) must be satisfied in the region of interest :

$$\underbrace{\frac{\partial}{\partial t} \int_V \rho \phi dV}_{\text{Unsteady}} + \underbrace{\oint_A \rho \phi \mathbf{V} \cdot d\mathbf{A}}_{\text{Convection}} = \underbrace{\oint_A \Gamma \nabla \phi \cdot d\mathbf{A}}_{\text{Diffusion}} + \underbrace{\int_V S_\phi dV}_{\text{Generation}}$$



- Fluid properties (density, viscosity, ...) are modeled empirically.
- **Simplifying assumptions** are made in order to make the problem tractable (e.g., **steady-state, incompressible, inviscid, two-dimensional**).
- Provide appropriate **initial and boundary conditions** for the problem.

Equation	Variable ϕ
Continuity	1
X momentum	u
Y momentum	v
Z momentum	w
Energy	h

How does CFD work (2/2) ?

- CFD solvers are based on the **finite difference method (FDM)** or **finite volume method (FVM)**, mostly.

- Flow domain is discretized into a finite set of **control volumes (CVs)**
- The collection of **CVs** is called the **grid** (or **mesh**)
- Transport Eqs (PDEs) for mass, momentum, energy (i.e., the **Navier-Stokes Eqs**) :

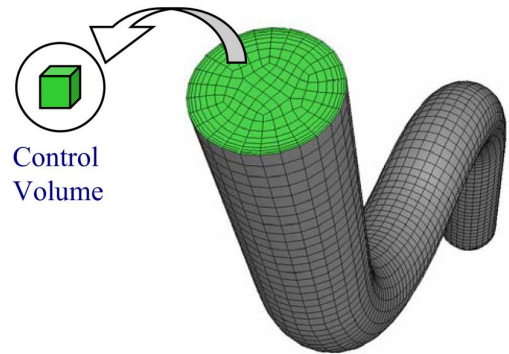
$$\underbrace{\frac{\partial}{\partial t} \int_V \rho \phi dV}_{\text{Unsteady}} + \underbrace{\oint_A \rho \phi \mathbf{V} \cdot d\mathbf{A}}_{\text{Convection}} = \underbrace{\oint_A \Gamma \nabla \phi \cdot d\mathbf{A}}_{\text{Diffusion}} + \underbrace{\int_V S_\phi dV}_{\text{Generation}}$$

are solved in each **CVs**.

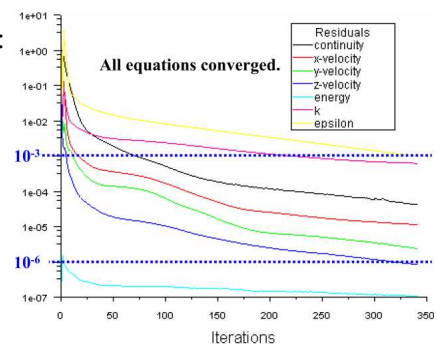
- These PDEs are discretized into a system of algebraic equations :

$$\frac{\partial \rho \phi}{\partial t} V + \sum_f^{N_{\text{faces}}} \rho_f \bar{\mathbf{v}}_f \phi_f \cdot \bar{\mathbf{A}}_f = \sum_f^{N_{\text{faces}}} \Gamma_\phi \nabla \phi_f \cdot \bar{\mathbf{A}}_f + S_\phi V$$

- All algebraic equations are solved on a PC to give the flow solution
- The solution is **post-processed** to extract quantities of interest (e.g. lift, drag, torque, heat transfer, separation, pressure loss, etc.).

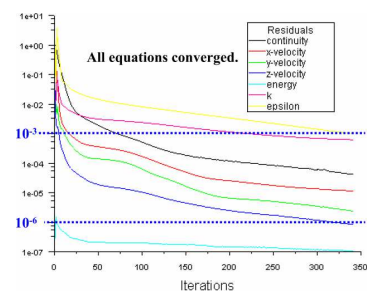
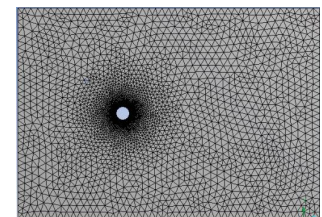
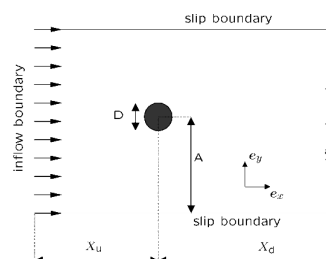


Fluid region discretized into a set of CVs



1) Preprocessing :

- Formulation of the problem (material properties, physical models, governing equations, initial and boundary conditions)
- Build computational mesh :
 - CAD model of flow domain
 - Meshing of flow domain (set of control volumes)



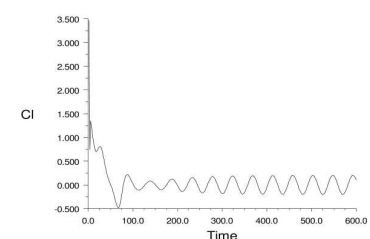
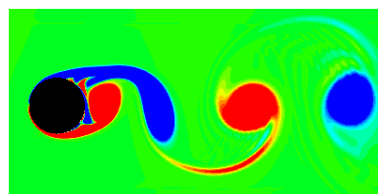
2) Solving :

- Discretization of the governing equations by FDM or FVM
- Solution of the resulting algebraic equations

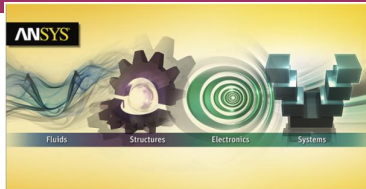
$$a_p \phi_p + \sum_{nb} a_{nb} \phi_{nb} = b_p$$

3) Post processing :

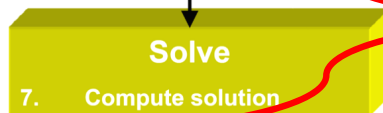
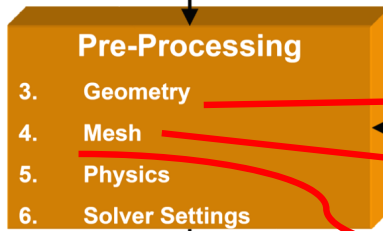
- Visualisation of the solution
- Analysis of results (forces, flow rates, etc ...)



CFD Analysis with ANSYS Workbench (FLUENT flow solver)



CFD Workflow



- **FLUENT** is a software for CFD analysis, heat transfer and related problems for industrial applications
- The software company **ANSYS** bought **FLUENT** in 2006
- **Worbench** is a common platform for **ANSYS** analysis tools
 - ✓ Geometry creation / CAD importation
 - ✓ Meshing
 - ✓ Solving
 - ✓ Post-processing

Components in Worbench

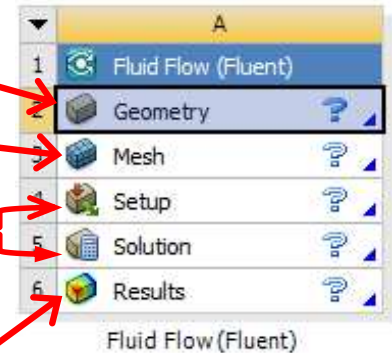
Workbench interface

DesignModeler

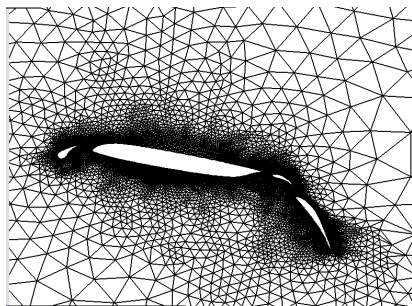
ANSYS Meshing

ANSYS FLUENT

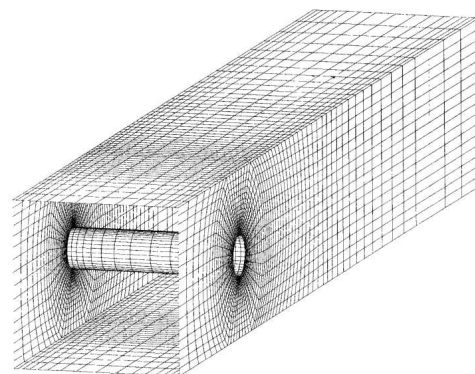
ANSYS FLUENT
or
ANSYS CFD- POST



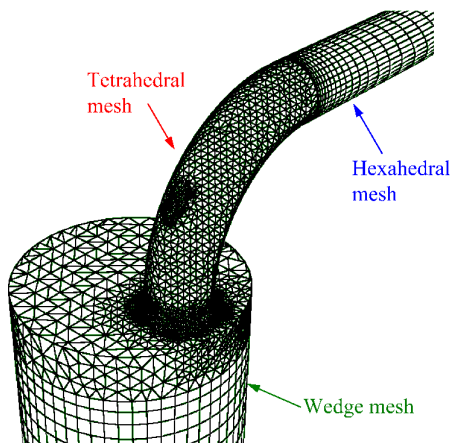
Exemples of grid used in CFD analysis



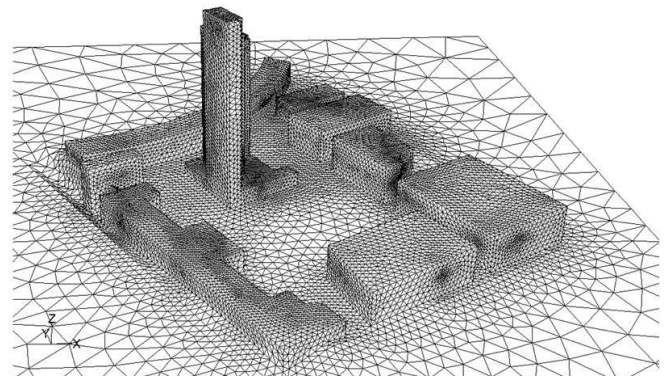
Triangle mesh for flow past 2D airfoil



Curvilinear mesh for 3D flow past circular cylinder



Hybrid mesh for an
IC engine valve port



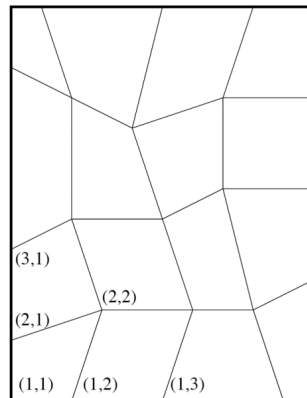
Mesh for the Thiers square in Nancy (France)

Classification of grids (or meshes) for CFD

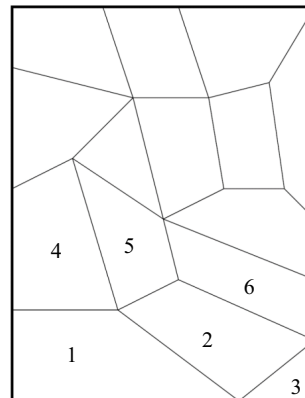
- 2 basic types of physical domains : simple and complex



- 2 basic types of grids : structured and unstructured



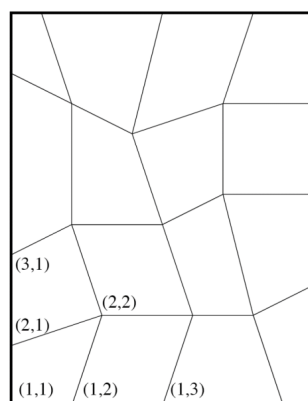
2D structured grid



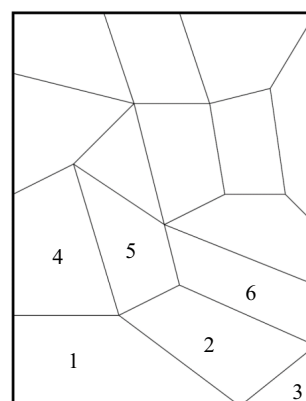
2D unstructured grid

Classification of grids (or meshes) for CFD

- 2 basic types of grids : structured and unstructured



2D structured grid

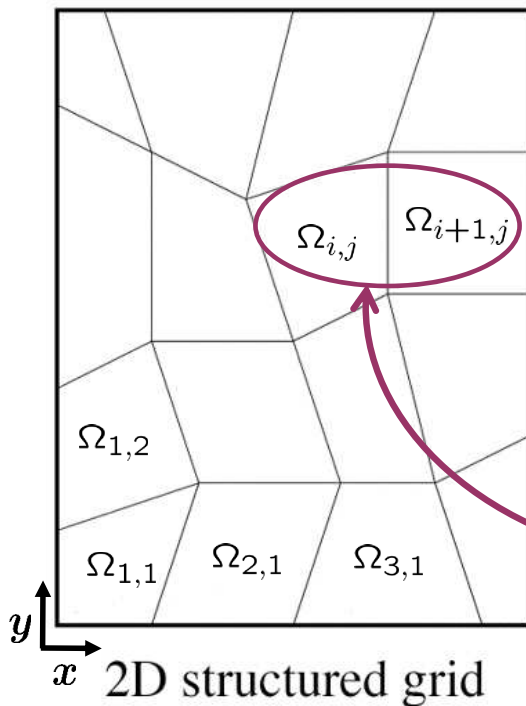


2D unstructured grid

- 2 basic types of physical domains : simple and complex



Structured Grids



- Structured grids are characterized by their **regular connectivity**.

- Cells are usually quadrilaterals in 2D. They can be arranged in a 2D array such as :

$$\Omega_{i,j} \text{ with } i = 1, \dots, N_x, j = 1, \dots, N_y$$

The total numbers of cells in the mesh is :

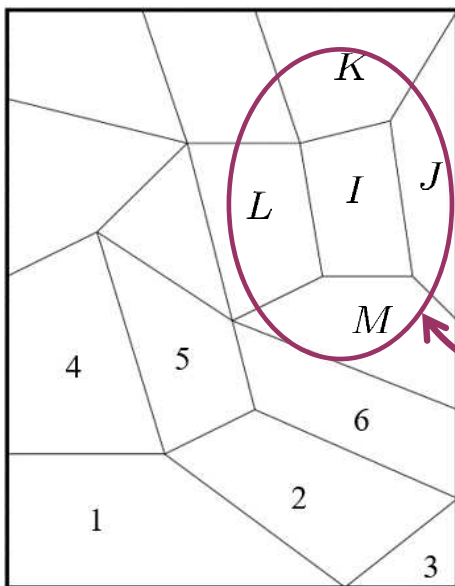
$$N = N_x \times N_y$$

- **Neighboring cells** can be directly identified by their indices $\Rightarrow$ the evaluation of gradients, flux, interpolations, is easily performed, *i.e.* :

$$\phi_{i+\frac{1}{2},j} \cong \frac{1}{2}\phi_{i,j} + \frac{1}{2}\phi_{i+1,j}$$

$\Rightarrow$ **Efficient computer implementations** (memory size, CPU time) of numerical methods (FDM, FVM) on structured grids

Unstructured Grids



2D unstructured grid

- Unstructured grids are characterized by their **irregular connectivity**.

- The cells (generated by powerful meshing softwares) have **no particular ordering**.

$\Rightarrow$ Use of global indexing

- To perform numerical computations, **explicit storage of neighborhood relationship** is needed.

- Example : the indices of the 4 neighbours of cell I are stored in the array `pointer_cell(N, 4)` as :

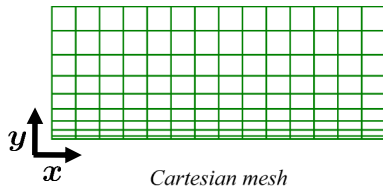
$$J = \text{pointer_cell}(I, 1), K = \text{pointer_cell}(I, 2)$$

$$L = \text{pointer_cell}(I, 3), M = \text{pointer_cell}(I, 4)$$

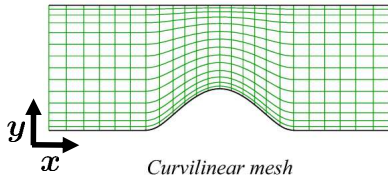
$\Rightarrow$ Algorithms are more complex, and they are more memory and CPU intensive.

- FDM not possible on unstructured grids, only FVM can be used

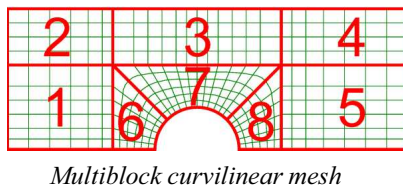
Types of structured grids



- **Cartesian grid** : Grid lines are parallel to the xy Cartesian axes. Cartesian grids are only used for simple physical domains (rectangles in 2D)



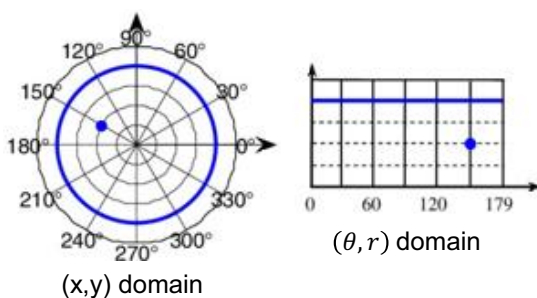
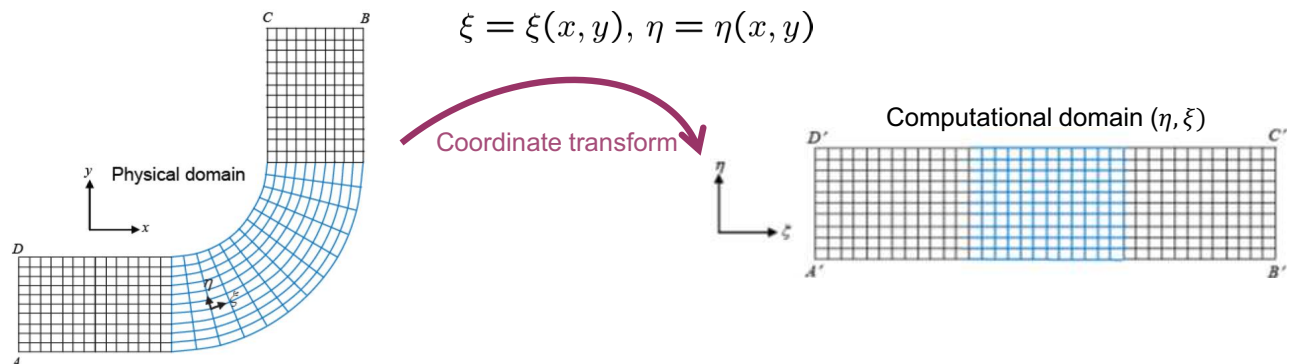
- **Curvilinear grid** : useful for complex domains. Grid lines are curved to fit the boundaries of the physical domain.



- **Multiblock grid** : To ease the generation of grids for complex domains, it is customary to divide the physical domain into blocks of simpler domains, such as each block remains structured.

Principles of curvilinear grid generation

- Map a physical domain (x, y) into a Cartesian computational domain (ξ, η) by coordinate transform :



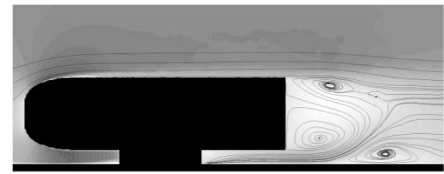
- Example : solve Laplace PDE $\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0$ in a circle of radius R . Using polar coordinates $x = r \cos \theta$, $y = r \sin \theta$ the PDE reads :

$$\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial \phi}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 \phi}{\partial \theta^2} = 0, \quad (r, \theta) \in [0, R] \times [0, 2\pi]$$

- PDEs on curvilinear grids involves **metrics**, and are more intricate to solve than in Cartesian grid.
- FDM and FVM can be applied.
- Curvilinear grids can be generated analytically, or numerically with **grid generation software**.

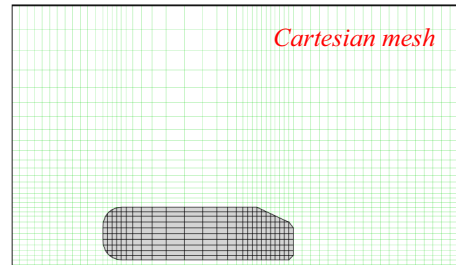
Grids for complex geometries

- Example from car aerodynamics : the Ahmed body

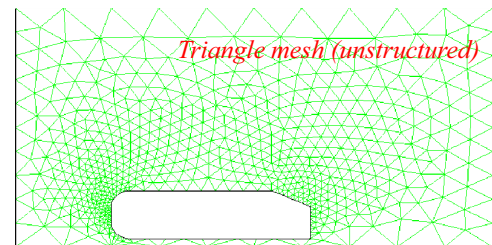
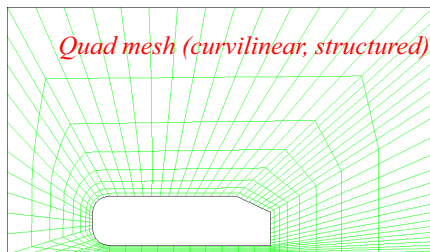


Industrial CFD applications are done in a fluid domain with complex shape => 2 strategies to build the mesh

- Non body-fitted grids

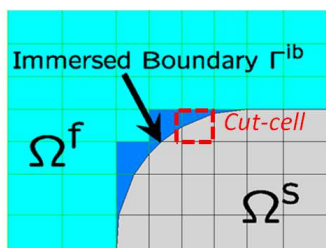
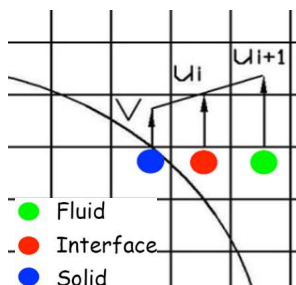
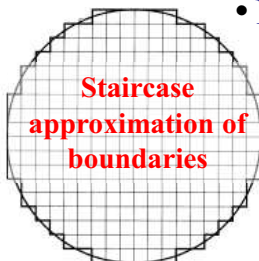


- Body-fitted grids



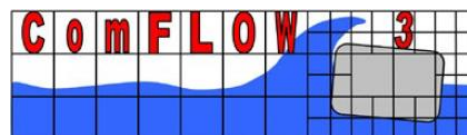
Non Body-Fitted Methods

- Non body-fitted grids : grid lines do not follow the domain boundaries



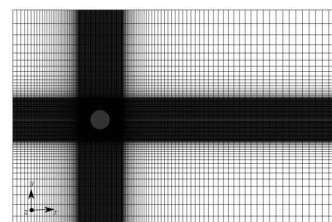
- Advantages :

- Most of the cells have a regular Cartesian shape
- Cartesian methods are accurate, robust and fast.
- Cartesian grids are popular for multiphase flows (e.g VOF or level-set methods for two-phase gas-liquid flows)



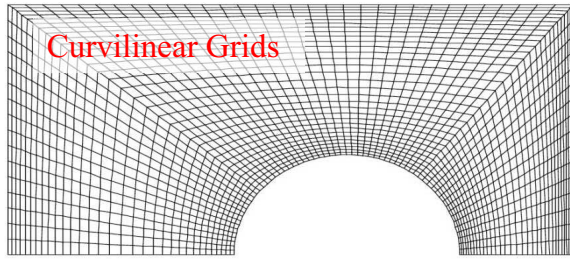
- Drawbacks :

- Difficult to impose BCs at complex boundaries. Several techniques : Staircase, Immersed boundary or Cut-cell methods.
- Local grid refinement is not possible.
- Not widely used in commercial CFD codes.



Body-Fitted Methods

- Body-fitted grids : grid lines follow the domain boundaries

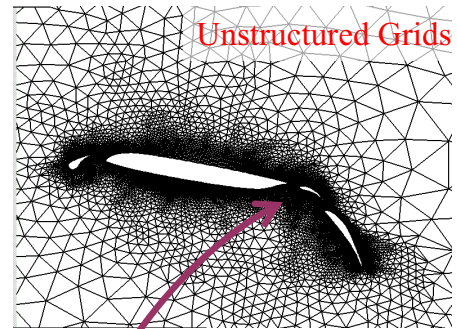
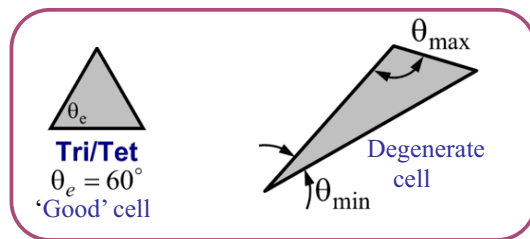


Advantages :

- Grid is structured => Efficient CFD solvers

Drawbacks :

- Software helps the grid generation, but good-quality grid is time-consuming to generate by the user.



Advantages :

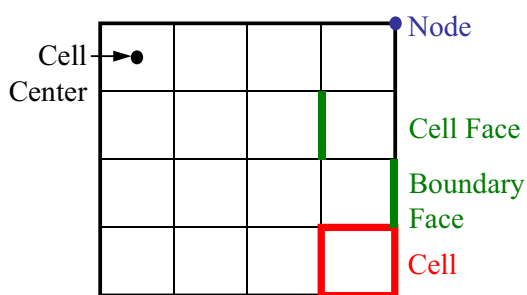
- Property of 'local grid refinement': possible to cluster grid points near boundary layers.
- Grid generation is fully automated (e.g. ANSYS Meshing)
- Unstructured meshes are now the standard for industrial CFD applications.

Drawbacks :

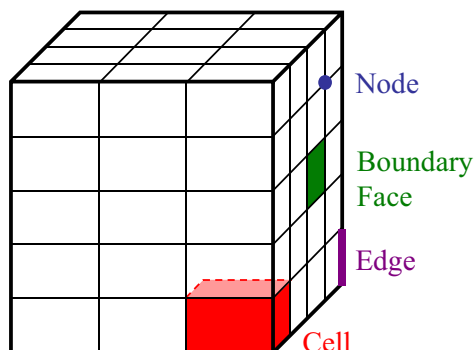
- Use of indirect addressing reduces the computational efficiency of the CFD solvers compared to structured methods.
- Degenerate cells affect the accuracy of the solution

Mesh components in 2D and 3D

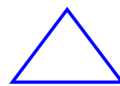
2D mesh



3D mesh



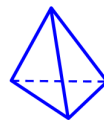
Basic types of cell in 2D and 3D



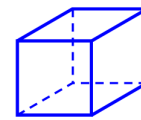
Triangle



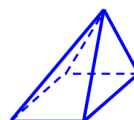
Quadrilateral



Tetrahedron



Hexahedron



Pyramid



Prism/Wedge

- Tri (2D) / tets (3D) : automatic meshing of extremely complicated regions
- Quad/hex : high quality elements for wall regions
- Pyramid : transition between tet and hex cells
- Wedge : formed when a tet mesh is extruded

Meshing strategy

- Mesh density should be high enough to capture all flow features.
- Mesh adjacent to solid boundaries should be **fine enough to resolve the boundary layer flow**
- **Quad, hex and prism** cells give better accuracy and should be preferred over triangles, tetrahedrons and pyramids.
- **Quad and hex** cells can be stretched where the flow is fully developed and 1D, as in boundary layers.

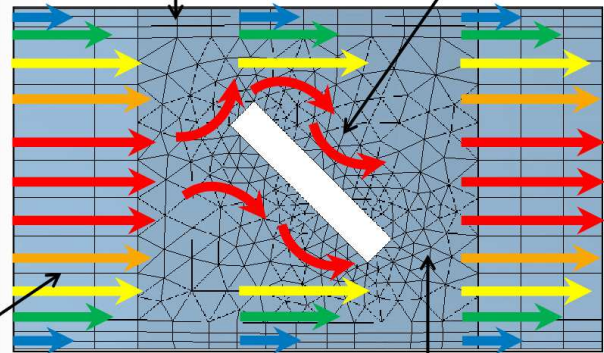
- **Physics**
- **Geometry**
- **Resources**

– **Mesh could require just one or a combination of methods.**

Hex (3d) or Quad (2d) cells used to mesh simple regions

High aspect ratio cells (on) near wall to capture boundary layer gradients

Cells refined around small geometric details and complex flow



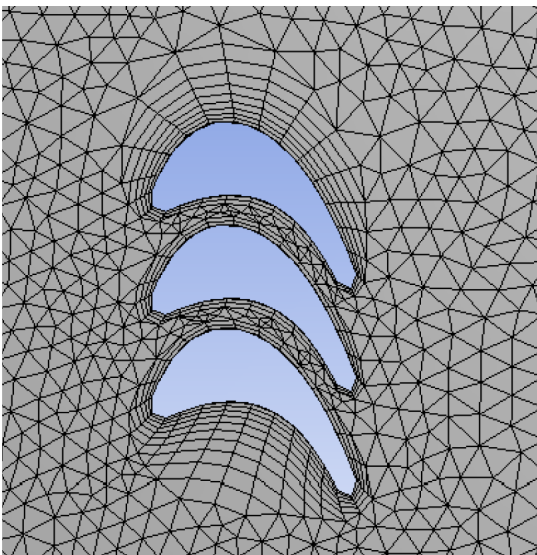
Tet (3d) or Tri (2d) cells used here to mesh complex region

4

© 2015 ANSYS, Inc.

February 12, 2015

Mesh quality

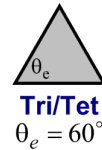
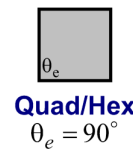
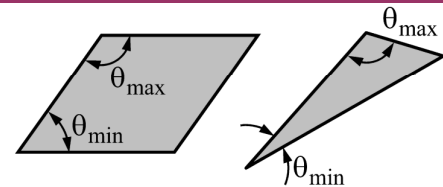


- **Example of hybrid 2D unstructured grid (tri/quad) for array of airfoils**
 - Automatic generation of triangle cells (constant size) away from solid boundaries
 - Regular-shaped cells (quadrilaterals) in boundary layers
 - High aspect-ratio cells near solid bodies
 - Graded mesh is used away from solid bodies
 - Distorted cells (consequence of computer-aided grid design algorithms) found between solid bodies
- **Distorted cells have negative impact on :**
 - Local accuracy of the solution
 - Convergence of the iterative solution of the discrete system of equations

Qeas	0-0.25	0.25-0.50	0.50-0.80	0.80-0.95	0.95-0.98	0.98-1.00*
Cell quality	Excellent	very good	good	acceptable	bad	Inacceptable*

Assessing mesh quality

- Mesh density should be high enough to capture all flow features.
- Mesh adjacent to solid boundaries should be **fine enough to resolve the boundary layer flow**
- **Quad, hex and prism** cells give better accuracy and should be preferred over triangles, tetrahedrons and pyramids.
- **Quad and hex** cells can be stretched where the flow is fully developed and 1D, as in boundary layers.



- **Mesh quality is assessed by 3 indicators**

- **Skewness** (Equiangle skew), defined by :

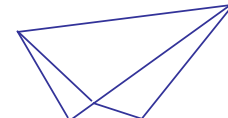
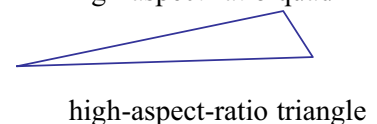
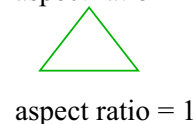
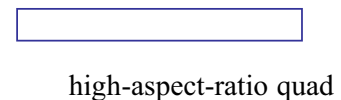
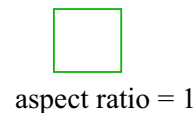
$$Q_{EAS} = \max \left[\frac{\theta_{\max} - \theta_e}{180 - \theta_e}, \frac{\theta_e - \theta_{\min}}{\theta_e} \right] \in [0, 1]$$

where $Q_{EAS} = 0$ is the ideal case. In practice, Q_{EAS} should not exceed 0.85-0.90 to ensure accuracy.

- **Aspect ratio** $\Delta x / \Delta y$ It should be close to 0.8-1.2 to ensure best results.

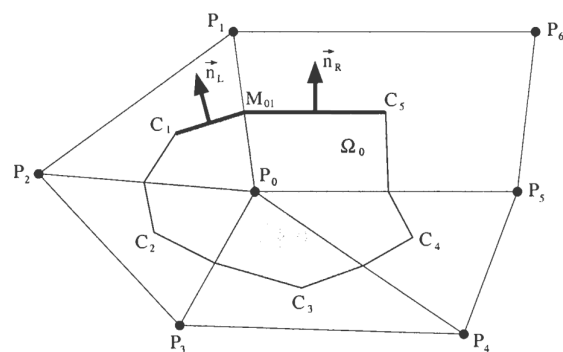
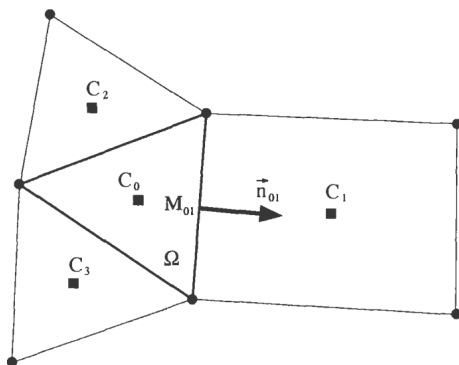
- **Smoothness**. Sudden jumps in the size of the cell may lead to large errors in the vicinity.

- **Grid generation software** have tools to examine mesh quality and perform mesh improvements.



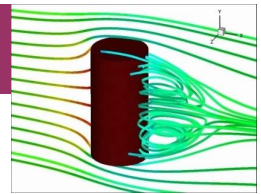
2 popular approaches for defining FVM control-volumes

- Consider an unstructured mesh with mixed cells (tri or quad) $C_0, C_1, C_2, \dots$

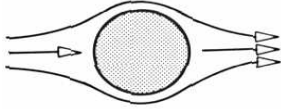
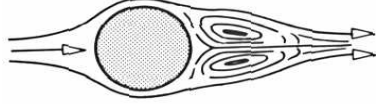
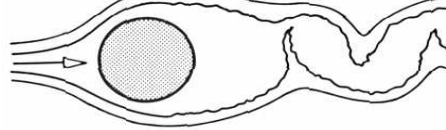
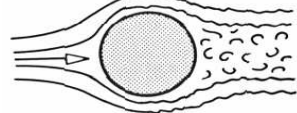
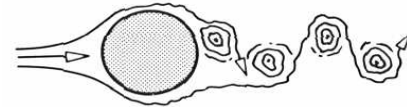


- **Cell-centred approach**
 - Unknowns ϕ_I located at the **centroid** ■ of the cells
 - The CVs Ω_I coincides with the cells
- **Vertex-centred approach**
 - Unknowns located at the **vertex** ● of the cells
 - Various choices for defining the CVs boundaries
- **Finite-volume discretization is similar, but choosing 1 of these 2 approaches has impact on :**
 - Implementation of BCs and accuracy of the solution
 - Memory requirements (more variables for cell-vertex approach)
 - Computational work for calculation numerical fluxes (depends on type of cells)
- **FLUENT uses cell-centred approach**

Quizz: Simplification of computational domain (2/2)



• Flow past circular cylinder :

$Re < 5$		Creeping flow (no separation)
$5-15 < Re < 40$		A pair of stable vortices in the wake
$40 < Re < 150$		Laminar vortex street (unsteady, 2D)
$150 < Re < 3 \times 10^5$		Laminar boundary layer up to the separation point, turbulent wake (unsteady, 3D)
$3 \times 10^5 < Re < 3.5 \times 10^6$		Boundary layer transition to turbulent
$Re > 3.5 \times 10^6$		Turbulent vortex street, but the separation is narrower than the laminar case

The Navier-Stokes equations (conservative form)

$$\left\{ \begin{array}{ll} \frac{\partial \rho}{\partial t} + \frac{\partial (\rho v_i)}{\partial x_i} = 0 & \text{Conservation of mass (1 eq.)} \\ \frac{\partial \rho v_i}{\partial t} + \frac{\partial (\rho v_j v_i)}{\partial x_j} = \frac{\partial \tau_{ij}}{\partial x_j} - \frac{\partial p}{\partial x_i} & \text{Conservation of momentum (3 eq.)} \\ \frac{\partial (\rho E)}{\partial t} + \frac{\partial (\rho v_j E)}{\partial x_j} = \frac{\partial}{\partial x_j} \left(k \frac{\partial T}{\partial x_j} \right) + \frac{\partial}{\partial x_j} (\tau_{ij} v_i) & \text{Conservation of energy (1 eq.)} \end{array} \right.$$

$$\text{Newtonian fluid} \quad \tau_{ij} = \mu \left[\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial v_k}{\partial x_k} \delta_{ij} \right]$$

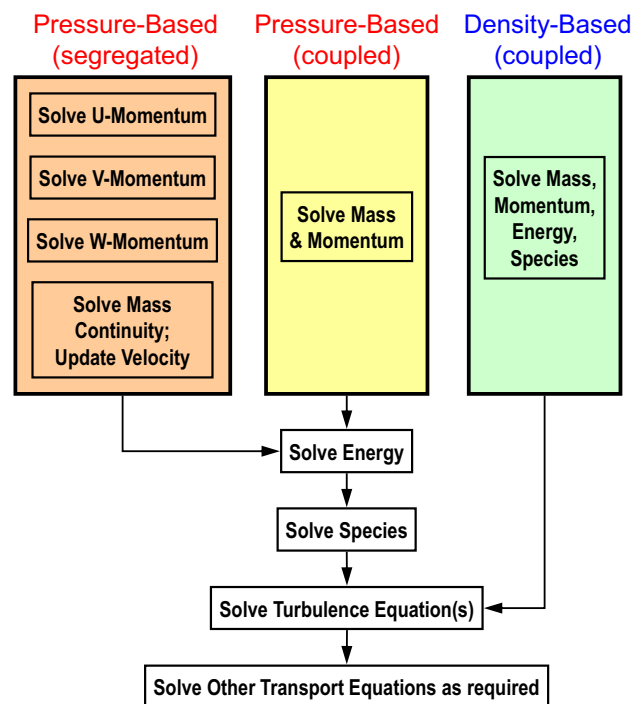
In 3D: 5 equations & 6 unknowns: $p, \rho, v_i, E(T)$

Need supplemental information: equation of state

- For compressible flows, the mass conservation is a transport equation for density. With an additional energy equation, pressure p can be specified from a thermodynamic relation (ideal gas law)
- For incompressible flows, density variations are not linked to the pressure. The mass conservation is a constraint on the velocity field.
- These differences in the role of pressure leads to different solutions algorithms (or solvers) for incompressible and compressible flows.

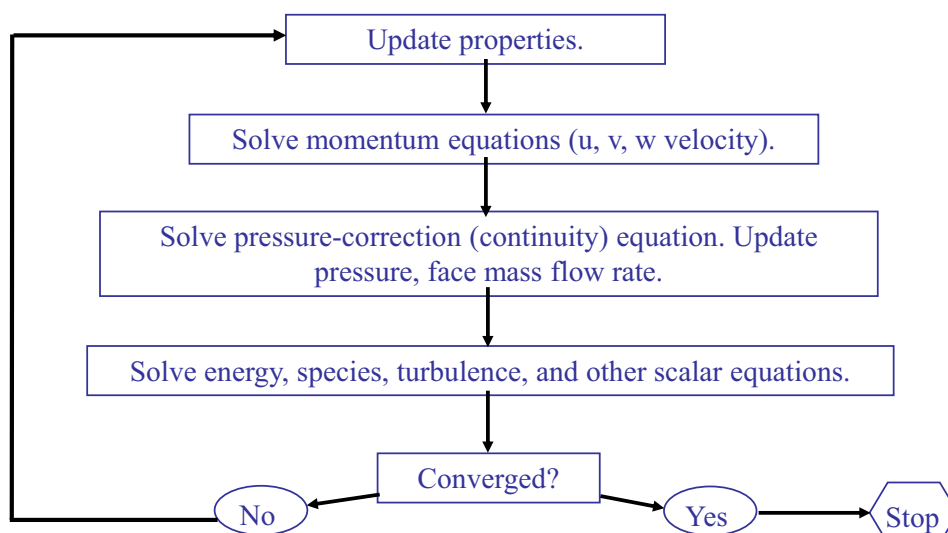
Available Solvers for Navier-Stokes Equations

- There are two kinds of solvers available in FLUENT – **Pressure based (incompressible flows)** and **Density based (compressible flows)**.
- The **pressure-based solvers** take momentum and pressure (or pressure correction) as the primary variables.
 - Pressure-velocity coupling algorithms are derived by reformatting the continuity equation as a pressure equation
- Two algorithms are available with the **pressure-based solvers**:
 - Segregated solver (PBSS)** – Solves for pressure correction and momentum sequentially.
 - Coupled Solver (PBCS)** – Solves pressure and momentum simultaneously.



Pressure-Based segregated solver (incompressible flows)

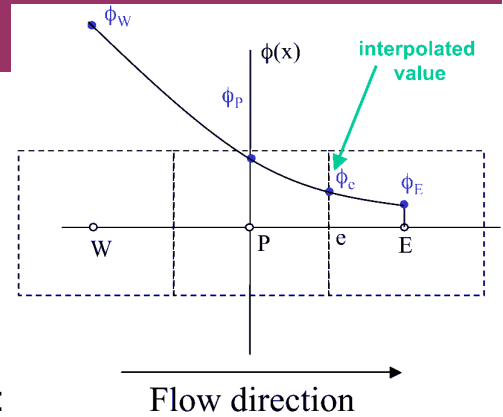
- For **incompressible flows**, density variations are not linked to the pressure. The **mass conservation is a constraint on the velocity field**.
- The **continuity equation** (combined with the **momentum**) is used to derive an **equation for the pressure (PPE)**
- The momentum and pressure equations are **solved sequentially** to fulfill **mass conservation** in the computational domain.



Discretization of Convection

- Field variables (stored at cell centers) **must be interpolated** to the faces of the control volumes.

$$V_{\text{cell}} \frac{\partial(\rho\phi)}{\partial t} + \sum_f^{N_{\text{faces}}} \rho_f \mathbf{V}_f \phi_f \cdot \mathbf{A}_f = \sum_f^{N_{\text{faces}}} \Gamma_\phi \nabla \phi_f \cdot \mathbf{A}_f + S_\phi V_{\text{cell}}$$



- Available Interpolation schemes** for convection:

- **First-Order Upwind** – Easiest to converge, only first-order accurate.
- **Power Law** – More accurate than first-order for flows when $Re_{\text{cell}} < 5$ (typ. low Re flows)
- **Second-Order Upwind** – Uses larger stencils for 2nd order accuracy, essential with tri/tet mesh or when flow is not aligned with grid; convergence may be slower.
- **Central Differencing (CDS)** – **Only available for LES simulations**
- **Quadratic Upwind Interpolation (QUICK)** – Applies to quad/hex and hybrid meshes, useful for rotating/swirling flows, 3rd-order accurate on uniform mesh.
- **Monotone Upstream-Centered Schemes for Conservation Laws (MUSCL)** – Locally 3rd order convection discretization scheme for unstructured meshes; more accurate in predicting secondary flows, vortices, forces, etc.

Interpolation Methods for Gradients

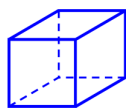
- Gradients of solution variables are required in order to evaluate **diffusive fluxes, velocity derivatives**, and for higher-order discretization schemes.

$$V_{\text{cell}} \frac{\partial(\rho\phi)}{\partial t} + \sum_f^{N_{\text{faces}}} \rho_f \mathbf{V}_f \phi_f \cdot \mathbf{A}_f = \sum_f^{N_{\text{faces}}} \Gamma_\phi \nabla \phi_f \cdot \mathbf{A}_f + S_\phi V_{\text{cell}}$$

- The gradients** of solution variables at cell centers can be determined using **three approaches**:
 - **Green-Gauss Cell-Based** – The default method; solution may have false diffusion (smearing of the solution fields).
 - **Green-Gauss Node-Based** – **More accurate**; minimizes false diffusion; recommended for tri/tet meshes. Not available for polyhedral meshes (pyramid, wedges)
 - **Least-Squares Cell-Based** – Recommended for polyhedral meshes; has the same accuracy and properties as node-based Gradients.



Tetrahedron



Hexahedron



Pyramid



Prism/Wedge

Interpolation Methods for Pressure

- Interpolation schemes (based on Rhie-Chow momentum-weighted interpolation) for calculating cell-face pressures when using the segregated solver in FLUENT are available as follows:
 - **Standard** – The default scheme; reduced accuracy for flows exhibiting large surface-normal pressure gradients near boundaries (but should not be used when steep pressure changes are present in the flow – **PRESTO! scheme should be used instead.**)
 - **PRESTO! (recommended)** – Use for highly swirling flows, flows involving steep pressure gradients (porous media, fan model, etc.), or in strongly curved domains
 - **Linear** – Use when other options result in convergence difficulties or unphysical behavior
 - **Second-Order** – Use for compressible flows; not to be used with porous media, jump, fans, etc. or VOF/Mixture multiphase models
 - **Body Force Weighted** – Use when body forces are large, e.g., high Ra natural convection or highly swirling flows

Pressure-Velocity Coupling for Segregated Solver

- **Pressure-velocity coupling** refers to the **numerical algorithm** which uses a combination of continuity and momentum equations to derive an **equation for pressure** (PPE or pressure correction equation) for the **iterative solution** of the Navier-Stokes equations

$$\frac{\partial u_i}{\partial x_i} = 0$$

Conservation of **mass** (continuity)

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\mu \frac{\partial u_i}{\partial x_j} \right) - \frac{1}{\rho} \frac{\partial p}{\partial x_i}$$

Conservation of **momentum**

- Four algorithms are available in FLUENT:
 - **Semi-Implicit Method for Pressure-Linked Equations (SIMPLE)**
 - The **default scheme**, robust
 - **SIMPLE-Consistent (SIMPLEC)**
 - Allows faster convergence for simple problems (e.g., laminar flows with no physical models employed).
 - **Pressure-Implicit with Splitting of Operators (PISO)**
 - Useful for **unsteady flow problems** or for meshes containing cells with higher than average skewness
 - **Fractional Step Method (FSM)** for unsteady flows.
 - Used with the NITA scheme; similar characteristics as PISO, 20% cheaper.

①

Lecture notes:

Finite-volume Method (FVM)

1) Introduction

For the convection-diffusion balance equation:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\vec{v} \phi) - \nabla \cdot (\nu \nabla \phi) = q \quad (1)$$

with non-constant coefficients:

- Convection velocity $\vec{v} = \begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix}$

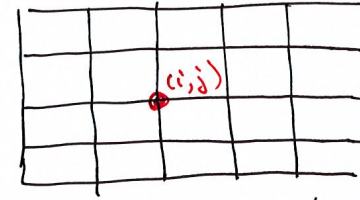
- viscosity $\nu = \nu(x,y)$

the previous lectures focused on the basics of the FDM in 1D. We have seen:

- Truncation error of a scheme
- Discretization formulas for:
 - diffusion: centred method $\delta_{xx}^0 \phi$
 - convection: centred method $\delta_x^0 \phi$
upwind method $\delta_x^-, \delta_x^{+I}$
- Time-integration based on the M.O.L. using EP1, ER1, CN, etc...
- Stability of the numerical solution (e.g.: CFL criterion)

②

However, the FDM method applies only to Cartesian mesh:



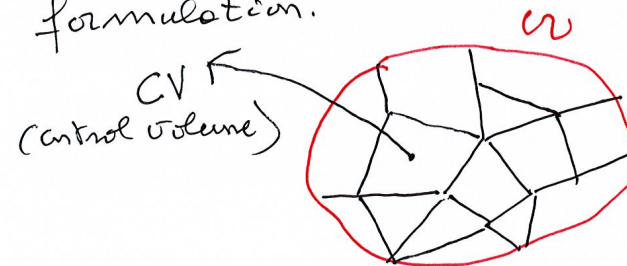
and its application is restricted to simple geometries.

What remains to be described in this lecture:

- Application to "real-life" problems (fluid flows, thermal problems) in 2D or 3D

- Numerical methods applicable to complex geometries: 

FVM has now a widespread use for engineering applications. Its main characteristic is to use the integral formulation of balance equations, in analogy to the FDM that uses its differential formulation.



- ③ • In the FVM, the physical domain Ω is divided in CVs, and the balance equation is discretized in each CV $\Rightarrow$ 1 algebraic equation per CV.

- FVM can be applied to CVs of arbitrary shape:



and thus can be applied to complex geometries, in analogy to FDM.

- Nevertheless, FDM and FVM have common grounds. For simple applications using:

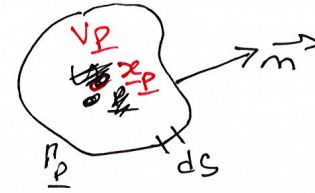
- Cartesian grids
- Constant coefficients (e.g. $\vec{v}$, ν in Eq. (1))
- uniform grid spacing $\Delta x, \Delta y$

the FVM and FDM are identical. Thus, the basic notions of FDM (truncation error, stability, upwinding, ...) apply to FVM. For more general application, it is preferable to use the FVM.

④ 2) Principles of the FVM

- Volume of Ω_P is V_P

- Center is $\underline{x}_P$



The basics of FDM are explained for the convection-diffusion Eq. (1), ~~in the~~ which is written in integral form in the CV Ω_P by using the flux-divergence theorem:

$$\forall \text{ vector } \vec{V}, \quad \int_{\Omega_P} \nabla \cdot \vec{V} d\Omega = \int_{\Gamma_P} \vec{V} \cdot \vec{n} dS \quad (2)$$

The integral form of (1) is thus:

$$\underbrace{\frac{d}{dt} \int_{\Omega_P} \phi dV}_{\text{Volume integrals}} + \underbrace{\int_{\Gamma_P} (\vec{F}^c - \vec{F}^d) \cdot \vec{n} dS}_{\text{Surface integrals}} = \underbrace{\int_{\Omega_P} q dV}_{\text{Volume int.}} \quad (3)$$

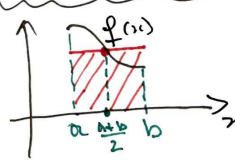
with the "numerical" fluxes $\vec{F} = \begin{pmatrix} F_x \\ F_y \end{pmatrix}$

- Convective fluxes $\vec{F}^c = \phi \vec{v} = \begin{pmatrix} \phi u \\ \phi v \end{pmatrix} \quad (4a)$

- Diffusive fluxes $\vec{F}^d = \nu \vec{\nabla} \phi = \begin{pmatrix} \nu \partial \phi / \partial x \\ \nu \partial \phi / \partial y \end{pmatrix} \quad (4b)$

- ⑤ The integrals in Eq. (3) will be approximated using the mid-point quadrature, which is a second-order discretization.

Definition (Mid-point quadrature in 1D)



$$\int_a^b f(x) dx \approx (b-a) f\left(\frac{a+b}{2}\right) \quad (5)$$

With this quadrature, the volume integrals are easily calculated, for example:

$$\int_{V_P} \Phi dV \approx V_P \Phi_P \Rightarrow \Phi_P \approx \frac{1}{V_P} \int_{V_P} \Phi dV \quad (6)$$

which means that Φ_P (value at center x_P) can be considered as the volume average of Φ in V_P .

After discretization of the volume integrals, the convection-diffusion Eq. (3) reads in V_P :

~~$$\frac{d}{dt} \Phi_P + \frac{1}{V_P} \left[\vec{F}_E(\Phi_P) - \vec{F}_W(\Phi_P) \right] \cdot \vec{m} dS = q_P \quad (4)$$~~

and the finite-volume discretization will be achieved by the approximation of the numerical fluxes (4) (convection and diffusive) on the FV mesh.

$$\frac{d}{dt} \Phi_P + \frac{1}{V_P} \int_{P_f} \vec{F} \cdot \vec{m} dS = q_P, \text{ with } \vec{F} = \vec{F}_c - \vec{F}_d \quad (7)$$

⑥

3) Flux evaluation on a Cartesian mesh.

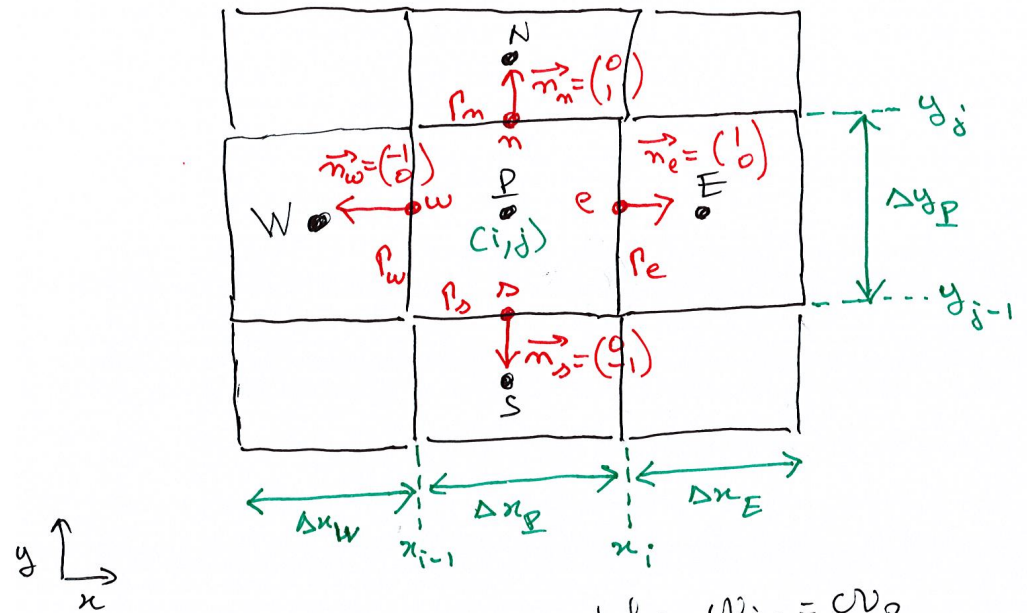


Fig 1: Compass notation used for $V_{i,j} = V_P$

The flux evaluation is best described by using the compass notation of Fig. 1. In particular:

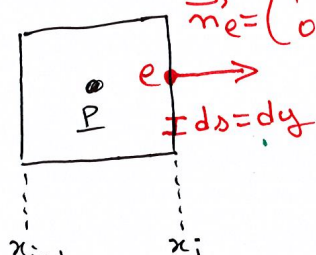
- The volume of cell V_P is $V_P = \Delta x_P \Delta y_P$. Its center is x_P
- Cell P shares frontiers ("faces") with 4 neighboring cells: E, W, N and S. (Capital letters used for CVs)
- Thus, the surface P_f of cell V_P is decomposed in 4 elementary faces P_f :
- (8) $P_f = P_n \cup P_w \cup P_e \cup P_s$ (small letters for faces)
- Each face P_f has a center (x_f) and a normal vector $\vec{m}_f$

④ With these definitions, the numerical flux (4) is decomposed as:

$$\frac{1}{V_P} \int_P \vec{F} \cdot \vec{n} dS = \frac{1}{V_P} \sum_{\text{face } f} \int_{P_f} \vec{F} \cdot \vec{n}_f dS \quad (9)$$

$$= \frac{1}{\Delta x_P \Delta y_P} \left[\int_{P_e} F_x dy - \int_{P_w} F_x dy + \int_{P_n} F_y dx - \int_{P_s} F_y dx \right]$$

Remark 2: The calculation on face P_e is done as:



$$\int_{P_e} \vec{F} \cdot \vec{n}_e dS = \int_{P_e} \begin{pmatrix} F_x \\ F_y \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} dy$$

$$= \int_{P_e} F_x(x_i, y) dy$$

$$\approx \Delta y_P F_x(x_e)$$

by using mid-point rule.

After a similar calculation on all 4 faces, the numerical flux (4) reads finally:

$$\frac{1}{V_P} \int_P \vec{F} \cdot \vec{n} dS = \frac{(F_x)_e - (F_x)_w}{\Delta x_P} + \frac{(F_y)_n - (F_y)_s}{\Delta y_P} \quad (10)$$

notation for $F_x(x_e)$

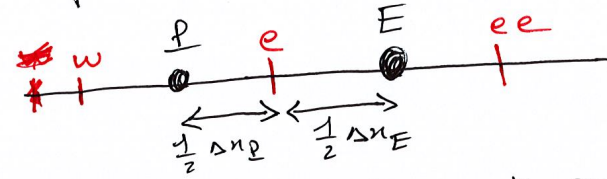
This equation involves the flux vectors $\vec{F}$ calculated at the face centers of CP_P . To achieve the FVM discretization, we need to interpolate them at the cell centers P, W, E, N and S . We need to distinguish the case of diffusive $\vec{F}^d$ and convective $\vec{F}^c$ fluxes.

⑧ 4) Interpolation of diffusive fluxes $\vec{F}^d = \nu \nabla \Phi$

When $\vec{F}$ is given by (4b), Eq. (10) reads:

$$\frac{1}{V_P} \int_P \vec{F} \cdot \vec{n} dS = \frac{\nu_e \left(\frac{\partial \Phi}{\partial x} \right)_e - \nu_w \left(\frac{\partial \Phi}{\partial x} \right)_w}{\Delta x_P} + \frac{\nu_n \left(\frac{\partial \Phi}{\partial y} \right)_n - \nu_s \left(\frac{\partial \Phi}{\partial y} \right)_s}{\Delta y_P} \quad (11)$$

in the case of non-constant viscosity, it is evaluated at ~~the~~ face centers.



For calculating $\left(\frac{\partial \Phi}{\partial x} \right)_e$ at face center x_e , we consider the centred FD formula:

$$\left(\frac{\partial \Phi}{\partial x} \right)_e = \frac{\Phi_E - \Phi_P}{x_E - x_P} = \frac{\Phi_E - \Phi_P}{\frac{1}{2} \Delta x_P + \frac{1}{2} \Delta x_E} \quad (12)$$

and use similar FD formulas for the other 3 faces, i.e. at the north face:

$$\left(\frac{\partial \Phi}{\partial y} \right)_n = \frac{\Phi_N - \Phi_P}{\frac{1}{2} \Delta y_N + \frac{1}{2} \Delta y_P} \quad (13)$$

the resulting FV discretization of (11) is equivalent to the centred FD formulas $\delta_{xx}^0 \Phi_P$ and $\delta_{yy}^0 \Phi_P$. In effect, in the case of uniform mesh ($\Delta x, \Delta y = \text{const}$) and constant viscosity, the first term in the RHS of (11) simplifies as:

$$\frac{V_e \left(\frac{\partial \phi}{\partial n} \right)_e - V_w \left(\frac{\partial \phi}{\partial n} \right)_w}{\Delta x_P} = \nu \left[\frac{\frac{\phi_E - \phi_P}{\Delta n} - \frac{\phi_P - \phi_W}{\Delta n}}{\Delta n} \right] \quad (14)$$

$$= \nu \left(\frac{\phi_E - 2\phi_P + \phi_W}{\Delta n^2} \right)$$

$$= \nu \nabla_{xx}^0 \Phi_P.$$

order accurate.

5) Interpolation of convective fluxes $\vec{F}^c = \phi \vec{u}$

$$\frac{1}{V_P} \int_{P_P} \vec{F} \cdot \vec{n} dS = \frac{\mu_e \phi_e - \mu_w \phi_w}{\Delta x_P} + \frac{u_m \phi_m - u_d \phi_d}{\Delta y_P}$$

nodes, as in the FDM, is continuous. ϕ_e : interpolated values.

The graph shows a horizontal axis with three points labeled W , P , and E . Above the axis, a green curve represents the function $\phi(x)$. At the points W , P , and E , there are black dots representing nodes. Above these dots are green labels ϕ_W , ϕ_P , and ϕ_E respectively. A green line segment connects the points (W, ϕ_W) and (P, ϕ_P) , and another green line segment connects the points (P, ϕ_P) and (E, ϕ_E) . A red arrow points from the text ϕ_e : interpolated values to the green line segment between P and E . There are also red dashed vertical lines extending from the axis at points between W and P , and between P and E .

→
flow direction: $u_e > 0$

5-1) Central Differencing scheme (CDS). (10)
The ~~simplest~~ ^{most common} ~~method~~ ^{method} is the central interpolation:

such as the flip a face e reds:

This FVM scheme is usually called CDS ad, in the case of constant coefficient ($\vec{v} = c\vec{e}$; $\Delta x, \Delta y$ constant) is equivalent to the central FD $\nabla_{\text{CD}}^2 \phi_P$ since, in this case the first term in Eq. (15) simplifies as:

From the FDM lecture, we can infer that CDS is $O(\Delta u^2)$ but, when convection dominates ($Pe = \frac{u \Delta u}{\nu} \gg 1$), ~~leads~~ to large oscillations in the solution. In this case, it is advised to use upwind-biased interpolation.

These schemes are based on the flow direction (see Fig. 2) and distinguish the case where $u_e < 0$ and $u_e > 0$.

(10)

S-2-1) Upwind Differencing Scheme (UDS)

(11)

This scheme consists in a constant interpolation with respect to the local flow direction u_e :

$$\phi_e = \begin{cases} \phi_P & \text{if } u_e \geq 0, \\ \phi_W & \text{if } u_e < 0. \end{cases} \quad (19)$$

In the case of a constant ~~convect~~ ^{convect} velocity, the UDS is equivalent to the upwind FD scheme δ_n^- or δ_n^+ .

For example if $u_w = u_e = u > 0$, the first term in Eq. (15) simplify as:

$$\frac{u_e \phi_e - u_w \phi_w}{\Delta x} = \frac{u \phi_P - u \phi_W}{\Delta x} = u \delta_n^- \phi_P. \quad (20)$$

For the FD lectures, we can state that UDS gives no oscillation, in contrary to CDS, but ~~it is very diffusive~~ is only $O(\Delta x)$ and leads to large diffusion errors.

In order to enhance ~~the~~ the accuracy of UDS, we need to consider upwind-biased interpolation with polynomials of higher-order ~~than~~ the constant polynomial used in (19).

(12)

S-2-2) QUICK Scheme

Q.U.I.C.K = Quadratic Upwind Interpol for Convection Kinematics.

For $u_e > 0$

This scheme uses a ~~quadratic~~ ^{quadratic} polynomial ~~that interpolates ϕ at~~ ^{that interpolates ϕ at} points E , P and W , as shown in Fig. (2) to obtain the interpolation formula:

$$\phi_e = \frac{3}{8} \phi_W + \frac{3}{4} \phi_P - \frac{1}{8} \phi_E, \quad (21)$$

as seen ~~in~~ in Exercise B.2. This interpolator can be written as:

$$\phi_e = \frac{\phi_{i+1} + \phi_i}{2} - \frac{\Delta x^2}{8} \delta_{xx}^0 \phi_i. \quad (22)$$

which shows that the scheme stabilizes the CDS scheme by adding a second-order diffusion term. ~~Due to its good~~ ^{due to its good} properties of accuracy and stability, the QUICK has seen a widespread use in commercial CFD like such as ANSYS FLUENT.

6) Implementation of boundary conditions (BCs)

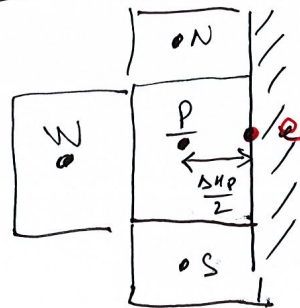


Fig. 3 Sketch of the mesh near the East boundary.

Let us consider the case depicted in Fig. 3, where the CV CV_P is next to the East boundary. The interpolations of fluxes at the w, s, n are the same as in section 4 and 5, but the flux $(F_n)_e$ at the east face needs to be modified to take into account Dirichlet BCs:

$$\phi_e = f_e, \quad f_e \text{ is a given data} \quad (23)$$

or Neuman BCs:

$$\left(\frac{\partial \phi}{\partial n}\right)_e = g_e, \quad g_e \text{ is a given data.} \quad (24)$$

6-1) BCs in convective fluxes

For the convective flux $F_n = u \phi$, the Dirichlet BC (23) is easily implemented as:

$$(F_n)_e = u_e \phi_e = u_e f_e. \quad (25)$$

The implementation of the Neuman BC (24) is more involved since it necessitates the Taylor expansion:

$$\phi_P \approx \phi_e - \frac{\Delta x_P}{2} \left(\frac{\partial \phi}{\partial x}\right)_e. \quad (26)$$

From this Taylor expansion, we obtain:

$$(F_n)_e = u_e \phi_e = u_e \left(\phi_P + \frac{\Delta x_P}{2} g_e \right) \quad (27)$$

6-2) BCs in diffusive fluxes

For the diffusive flux $F_n = D \frac{\partial \phi}{\partial n}$, the Neuman BC (24) is easily implemented as:

$$(F_n)_e = D_e \left(\frac{\partial \phi}{\partial n}\right)_e = D_e g_e. \quad (28)$$

The Dirichlet boundary condition ~~requires~~ requires the Taylor expansion (26), i.e.:

$$\left(\frac{\partial \phi}{\partial n}\right)_e \approx \frac{\phi_e - \phi_P}{\Delta x_P/2} \quad (29)$$

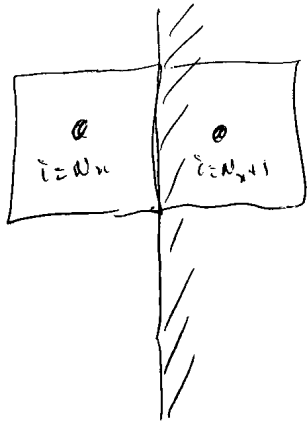
which yields finally:

$$(F_n)_e = D_e \frac{\phi_e - \phi_P}{\Delta x_P/2}. \quad (30)$$

Remark: In a computer code, these BCs are efficiently implemented by using the concept of ghost-cells described in the next pages.

Utilisation des conditions aux limites

(1)



Direchlet $\frac{\phi_{N_x+1} + \phi_{N_x}}{2} = f_e$

$\Rightarrow \phi_{N_x+1} = 2f_e - \phi_{N_x}$

Neumann

$\frac{\phi_{N_x+1} - \phi_{N_x}}{\Delta x} = g_e$

$\Rightarrow \phi_{N_x+1} = \phi_{N_x} + \Delta x g_e$

• Utilisation des Formules auxiliaires usuelles:

$\rightarrow$ Cas Neuman:

$F_e^e = (u_e f_e) = u_e \frac{\phi_{N_x+1} + \phi_{N_x}}{2} = u_e \phi_{N_x} + \frac{\Delta x}{2} g_e$

$F_d^e = \left(\frac{\partial \phi}{\partial x} \right)_e = \frac{\phi_{N_x+1} - \phi_{N_x}}{\Delta x} = g_e$

on retrouve les conditions aux limites du cas.

Sec. 15.3, 4.

$\rightarrow$ Cas Direchlet

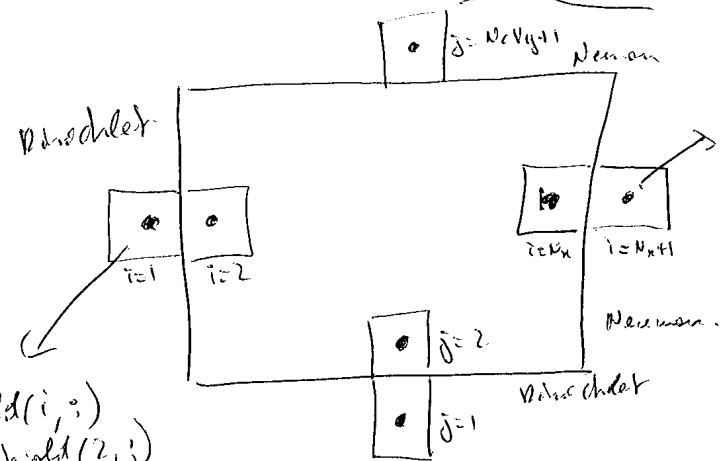
$F_e^e = u_e \frac{\phi_{N_x+1} + \phi_{N_x}}{2} = u_e f_e$

$F_d^e = \left(\frac{\partial \phi}{\partial x} \right)_e = \frac{\phi_{N_x+1} - \phi_{N_x}}{\Delta x} = \frac{f_e - \phi_{N_x}}{\Delta x/2}$

idem cas.

Application sur TP

$\text{phiold}(:, N_{\text{vgy}}+1) = \text{phiold}(:, N_{\text{vgy}})$



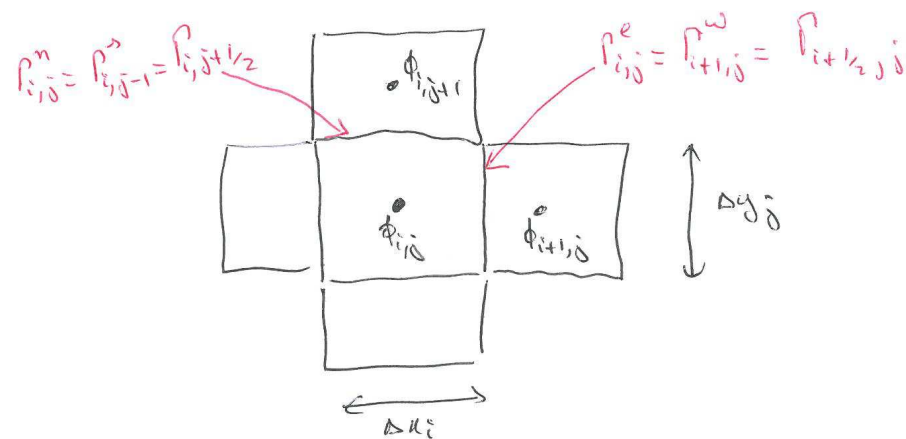
$\text{phiold}(i, :) = \text{phiold}(2, i)$

19/04/2014

Section 15.5 Global (and local) Conservation properties of FVM

FV Discretization of convection-diffusion eqn reads:

for cell $\omega_{i,j}$ (1)
$$\frac{d}{dt} \phi_{i,j} + \frac{F_{i,j}^e - F_{i,j}^w}{\Delta x_i} + \frac{F_{i,j}^n - F_{i,j}^s}{\Delta y_j} = q_{i,j}$$



An important property that FVM should verify is the local conservation of fluxes

$$F_{i,j}^e = F_{i+1,j}^w, \quad F_{i,j}^n = F_{i,j+1}^s \quad (2)$$

This is usually verified by even discretization, e.g. CDS discretization of convective flux:

$$F_{i,j}^e = u \left(\frac{\phi_{i+1,j} + \phi_{i,j}}{2} \right) = F_{i+1,j}^w \equiv F_{i+1,j}$$

We can thus drop the supers notation and rewrite (2) simply (1) as:

$$\frac{d}{dt} \phi_{i,j} + \frac{(F_{i+1/2,j} - F_{i-1/2,j})}{\Delta x_i} + \frac{(F_{i,j+1/2} - F_{i,j-1/2})}{\Delta y_j} = q_{i,j} \quad (3)$$

The local conservation of numerical fluxes (2) is a property of FVM as long as the EDPDE is written in conservative form:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (F(\phi)) = q \quad (4)$$

$$\Rightarrow \frac{d}{dt} \int_{\omega} \phi dV + \int_{\partial \omega} F(\phi) dS = \int_{\omega} q dV \quad (5)$$

Example 1: the convection-diffusion eqn in conservative form is:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\vec{\sigma} \phi - \nu \nabla \phi) = q \quad (6)$$

The non conservative (or convective) form is:

$$\frac{\partial \phi}{\partial t} + \vec{\sigma} \cdot \nabla \phi - \nu \nabla^2 \phi = q \quad (7)$$

The convective and conservative are analytically identical when $\nabla \cdot \vec{\sigma} = 0$, but the discretization of (7) may lead to the violation of local conservation property (2) $\Rightarrow$ creation of artificial source or sink terms in the FV equations.

See !

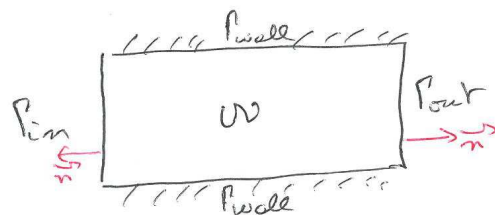
• Exercise

• lecture on NS eq^t in allocated arrangement

(3)

Example 2:

Conservation of mass in incompressible flows



The continuity eq^t reads:

$$\nabla \cdot \vec{v} = 0 \quad \forall \underline{x} \in U \quad (8a)$$

If we integrate this equation on whole domain U :

$$(8b) \quad \underbrace{\int_P \vec{v} \cdot \vec{n} ds}_{\dot{m}: \text{flow rate across } P} = 0 \quad \text{for } P = P_{in} \cup P_{out} \cup P_{wall}$$

which leads to:

$$\dot{m}_{in} - \dot{m}_{out} = 0 \quad (9)$$

since $\dot{m}_{wall} = 0$ due to B.C.s.

Hence, the global conservation of mass in U leads to a compatibility condition for B.C.s at P_{in} and P_{out} :

$$\int_{P_{in}} \vec{v}_{in} \cdot \vec{n} ds = \int_{P_{out}} \vec{v}_{out} \cdot \vec{n} ds \quad (10)$$

Rq1: Conservation of mass is one of the global compatibility properties of NS eq^ts (along with a search of momentum and kinetic energy). (4)

Compatibility condition (10) must be verified analytically to lead to a well-posed problem (solution exists and is unique). If the FV discretized version of this condition is violated, the discrete ~~NS~~ FV eq^ts may not be solvable $\Rightarrow$ blow up of the solution.

What is the equivalent of (8)-(10) in FVM?

In a cell $U_{i,j}$, Eq. (8a) leads to:

$$\dot{m}_{i,j} = \dot{m}_e^{i,j} - \dot{m}_w^{i,j} + \dot{m}_n^{i,j} - \dot{m}_s^{i,j} = 0 \quad (11)$$

(the relation between flow rate and $\vec{v}_{i,j}$ will be detailed in NS lectures). If local conservation (2) holds in all cells $U_{i,j}$, i.e.:

$$\dot{m}_e^{i,j} = \dot{m}_w^{i+1,j}, \quad \dot{m}_n^{i,j} = \dot{m}_s^{i,j+1}$$

then discrete continuity eq^t reads:

$$\dot{m}_{i+1/2,j} - \dot{m}_{i-1/2,j} + \dot{m}_{i,j+1/2} - \dot{m}_{i,j-1/2} = 0 \quad (12)$$

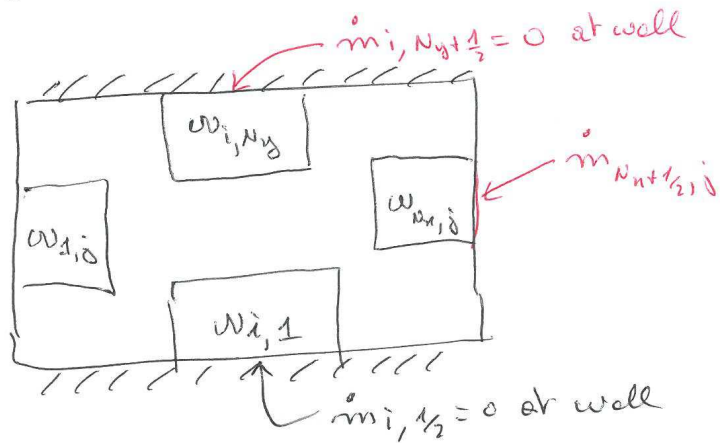
Now, discretization of continuity eq^t (8b) in whole fluid domain (8b) reads:

$$\sum_{i,j} \dot{m}_{i,j} = 0$$

$$\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} (\dot{m}_{i+1/2,j} - \dot{m}_{i-1/2,j} + \dot{m}_{i,j+1/2} - \dot{m}_{i,j-1/2}) = 0 \quad (5)$$

this sum cancels out except at boundaries P_{in} and P_{out} :

$$\underbrace{\sum_{j=1}^{N_y} \dot{m}_{1/2,j}}_{\text{mass flow at } P_{in}} - \underbrace{\sum_{j=1}^{N_y} \dot{m}_{N_x+1/2,j}}_{\text{mass flow at } P_{out}} = 0 \quad (13)$$



Eqⁿ (13) is a discretization of ~~mass~~ global mass ~~conservation~~ (10).

Hence, FVM is globally conservative if:

- Local conservation of flux numerical fluxes (2) is verified. (easy)
- BCs are "correctly" imposed at boundary cells. (more difficult)

If this property is not verified, some terms in FVM eq^s act as source or sinks, which alter the discrete conservation of quantities such as mass, momentum or kinetic energy.

⇒ See lecture on FVM for NS eq^s.

16 Les équations de Navier-Stokes pour un fluide incompressible

16.1 Dérivation des équations de Navier-Stokes

16.1.1 Conservation de la masse

Pour un fluide compressible, l'équation de conservation de la masse s'écrit :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} = 0. \quad (16.1)$$

En utilisant l'identité

$$\nabla \cdot q \mathbf{u} = q \nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla q, \quad (16.2)$$

qui est valable pour tout scalaire q et tout vecteur $\mathbf{u}$, on réécrit la conservation de la masse sous la forme :

$$\frac{1}{\rho} \frac{D\rho}{Dt} + \nabla \cdot \mathbf{v} = 0. \quad (16.3)$$

où D/Dt est la *dérivée particulaire*, définie par :

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \mathbf{v} \cdot \nabla \rho. \quad (16.4)$$

Généralités sur les fluides incompressibles

Dimensionnellement $[\nabla \cdot \mathbf{v}] = U/L$, où U et L sont la vitesse et longueur caractéristique de l'écoulement. Par définition, un fluide est considéré incompressible si la densité d'une particule matérielle varie peu :

$$\frac{1}{\rho} \frac{D\rho}{Dt} \ll \frac{U}{L}. \quad (16.5)$$

Ainsi, dans cette approximation l'équation de conservation de la masse s'écrit :

$$\nabla \cdot \mathbf{v} = 0 \quad (16.6)$$

Enlevons quelques idées préconçues :

- L'incompressibilité est une propriété de *l'écoulement*, et non du fluide. On peut montrer par analyse dimensionnelle que l'hypothèse d'incompressibilité implique :

$$M \ll 1,$$

où M est le nombre de Mach. On considère généralement que si $M \lesssim 0.2$ le nombre de Mach est suffisamment faible pour que l'hypothèse d'incompressibilité soit valide. Cela vaut en particulier pour les liquides, les écoulements internes dans les moteurs, autour d'une voiture, ...

- L'incompressibilité n'implique pas nécessairement que la masse volumique ρ est constante. Par exemple, l'écoulement de l'air qui est comprimé dans un moteur à piston de voiture est tel que $M \ll 1$, mais ρ varie en fonction du volume de la chambre de compression.

16.1.2 Conservation de la quantité de mouvement

L'équation de conservation de la quantité de mouvement s'écrit :

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v} - [\boldsymbol{\sigma}]) = \rho \mathbf{f}, \quad (16.7)$$

où $\mathbf{f}$ est un terme représentant les forces volumiques, et $[\boldsymbol{\sigma}]$ le tenseur des contraintes de Cauchy. Afin de clore le système, il est nécessaire de relier $[\boldsymbol{\sigma}]$ au mouvement du fluide. Une telle relation est appelée *relation*

constitutive. La plus simple de ces relations, proposée par Stokes, suppose que $[\boldsymbol{\sigma}]$ dépend linéairement du gradient de vitesses. En décomposant le tenseur des contraintes comme :

$$[\boldsymbol{\sigma}] = -p[\mathbf{I}] + [\boldsymbol{\tau}], \quad (16.8)$$

où $[\mathbf{I}]$ est le tenseur identité et p la pression, on fait apparaître $[\boldsymbol{\tau}]$, la partie visqueuse du tenseur des contraintes, qui s'écrit dans l'hypothèse de Stokes par :

$$[\boldsymbol{\tau}] = -\frac{2}{3}\mu\nabla \cdot \mathbf{v}[\mathbf{I}] + 2\mu[\mathbf{S}], \quad (16.9)$$

où μ est la viscosité dynamique du fluide, et $[\mathbf{S}]$ le tenseur des déformations :

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) = \frac{1}{2} (\nabla \mathbf{v} + (\nabla \mathbf{v})^T)_{ij}. \quad (16.10)$$

Les fluides qui satisfont à la relation de Stokes (16.9) avec μ constante sont appelés *fluides newtoniens*. Par exemple, l'eau ou le mercure peuvent être considérés comme newtoniens. Des fluides tels que le sang ou les polymères n'obéissent pas à la relation (16.9) : on les appelle *non-newtoniens*. La *rhéologie* est la discipline qui étudie les fluides non-newtoniens.

Dans la suite de ce cours, nous considérerons uniquement le cas de fluides newtoniens incompressibles tels que (16.6) est vérifié. Le tenseur de Stokes (16.8) s'écrit alors simplement :

$$[\boldsymbol{\sigma}] = -p[\mathbf{I}] + 2\mu[\mathbf{S}]. \quad (16.11)$$

16.1.3 Conservation de l'énergie

In order to make long story short : pour un écoulement subsonique, il suffit de considérer l'équation de transport de la température T :

$$c_p \frac{\partial \rho T}{\partial t} + c_p \nabla \cdot \rho \mathbf{v} T - \nabla \cdot \kappa \nabla T = \rho q, \quad (16.12)$$

avec κ la conductivité thermique, et c_p la chaleur spécifique à pression constante.

16.2 Adimensionnement des équations

En mécanique des fluides, il y a exactement 4 unités physiques indépendantes : celle de longueur, vitesse, masse et température, qui seront notées L , U , M et T_r respectivement. Toutes les autres unités doivent être formulées à partir de ces 4 unités de base de façon à éviter d'introduire des coefficients superflus dans les équations. Par exemple, l'unité de temps doit être définie comme L/U , l'unité de force F comme MU^2/L d'après la loi de Newton. Souvent, il se révèle commode de choisir ces unités à partir du problème physique à modéliser, et d'utiliser la *forme adimensionnelle* des équations de Navier-Stokes. Cela entraîne l'identification de nombres sans dimensions, tels le *nombre de Reynolds*, qui gouvernent le régime de l'écoulement.

Adimensionnement de l'équation de quantité de mouvement

Soient L et U des échelles de longueur et de vitesse typiques de l'écoulement considéré, que nous choisissons comme unités de référence pour notre calcul. L'unité de masse est choisie comme étant $M = \rho_r L^3$, où ρ_r est une masse volumique de référence, par exemple la masse volumique du fluide au repos. On définit alors les variables sans dimensions avec des primes :

$$\mathbf{x}' = \mathbf{x}/L, \quad \mathbf{v}' = \mathbf{v}/U, \quad \rho' = \rho/\rho_r. \quad (16.13)$$

Ces nouvelles variables doivent être introduites dans l'équation (16.7) avec le tenseur des contraintes (16.11). L'équation résultante est laissée en exo. Ensuite, lorsqu'on introduit d'autres grandeurs sans dimensions :

$$t' = Ut/L, \quad p' = \frac{p}{\rho_r U^2}, \quad \mathbf{f}' = \frac{L}{U^2} \mathbf{f}. \quad (16.14)$$

on obtient la *forme adimensionnelle* de l'équation de quantité de mouvement, qui s'écrit

$$\frac{\partial \rho' \mathbf{v}'}{\partial t'} + \nabla' \cdot \rho' \mathbf{v}' \mathbf{v}' + \nabla' p' - \nabla' \cdot \frac{1}{\text{Re}} [\mathbf{S}]' = \rho' \mathbf{f}'. \quad (16.15)$$

Dans cette équation, on a fait apparaître le nombre de Reynolds :

$$\text{Re} = \frac{\rho_r U L}{\mu}. \quad (16.16)$$

Le changement de variables (16.13) montrent que le terme convectif (ou inertiel) a pour dimension $\rho_r U^2/L$ et le terme dissipatif a pour dimension $\mu U/L^2$. Ainsi, le nombre de Reynolds mesure le rapport entre les forces inertielles et visqueuses en compétition au sein de l'écoulement. Pour $\text{Re} \gg 1$ l'inertie domine ; et pour $\text{Re} \ll 1$ la dissipation visqueuse domine. Ces deux effets sont contrebalancés par le gradient de pression.

Adimensionnement de la conservation de la masse

En utilisant le changement de variable (16.13), la forme adimensionnelle de l'équation de continuité (16.6) s'écrit simplement :

$$\nabla' \cdot \mathbf{v}' = 0. \quad (16.17)$$

Adimensionnement de l'équation de température

Comme unité de référence de la température, on utilise couramment un écart de température $T_H - T_C$, par exemple la différence de température entre une paroi chaude et une paroi froide. La variable sans dimension utilisée est alors :

$$T' = \frac{T - T_C}{T_H - T_C}.$$

Après adimensionnement, l'équation de transport de température (16.12) devient alors :

$$\frac{\partial \rho' T'}{\partial t'} + \nabla' \cdot \rho' \mathbf{v}' T' - \nabla' \cdot \frac{1}{\text{Pe}} \nabla' T' = \rho' q', \quad (16.18)$$

où apparaît le nombre de Péclet Pe défini par :

$$\text{Pe} = \frac{UL}{c_p \kappa}. \quad (16.19)$$

16.3 Les équations qu'on utilisera dans la suite

Dans la suite du cours, nous allons considérer un écoulement incompressible, tel que la conservation de la masse adimensionnée (16.17) s'écrit en enlevant les primes :

$$\nabla \cdot \mathbf{v} = 0. \quad (16.20)$$

Examinons la forme de l'équation de température (16.18), qui s'écrit sans les primes :

$$\frac{\partial \rho T}{\partial t} + \nabla \cdot \rho \mathbf{v} T - \nabla \cdot \frac{1}{\text{Pe}} \nabla T = \rho q. \quad (16.21)$$

Cette équation est similaire à une équation d'advection-diffusion scalaire. Sa discrétisation s'effectue par application directe des techniques du Chap. 15, et ne sera pas détaillée plus loin. Dès que la vitesse d'advection $\mathbf{v}$ est connue, Eq. (16.21) est une équation pour une scalaire passif advecté par l'écoulement : on peut se dispenser de considérer cette équation si T n'est pas une quantité intéressante où si μ est indépendante de T .

Tournons nous maintenant vers l'équation de quantité de mouvement (16.15), soit :

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \rho \mathbf{v} \mathbf{v} + \nabla p - \nabla \cdot \frac{1}{\text{Re}} [\mathbf{S}] = \rho \mathbf{f}. \quad (16.22)$$

Cette équation vectorielle est très proche d'une équation d'advection-diffusion, comme on le remarque en l'écrivant pour la $k^{\text{ième}}$ composante :

$$\frac{\partial \rho u_k}{\partial t} + \nabla \cdot \rho u_k \mathbf{v} + \frac{\partial p}{\partial x_k} - \nabla \cdot \frac{1}{\text{Re}} \mathbf{S}_k = \rho f_k, \quad (16.23)$$

avec

$$\mathbf{v} = (u_1, u_2)^T \quad (\text{ou } \mathbf{v} = (u, v)^T), \quad (16.24a)$$

$$\mathbf{S}_1 = \begin{pmatrix} 2 \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{pmatrix}, \quad \mathbf{S}_2 = \begin{pmatrix} \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ 2 \frac{\partial v}{\partial y} \end{pmatrix}. \quad (16.24b)$$

Pour discrétiser (16.23), on va s'appuyer sur notre expérience de l'équation d'advection-diffusion. Néanmoins, il apparaît des complications qui se révéleront significatives dans leur résolution numérique :

- Absence d'une équation d'évolution pour la pression p .
- Pas de conditions aux limites pour p .
- Contrainte d'incompressibilité : $\nabla \cdot \mathbf{v} = 0$.

16.4 Formulation intégrale

Afin d'écrire la formulation intégrale des équations de Navier-Stokes, on utilisera le théorème de la divergence 4.2, soit :

$$\int_{\Omega} \nabla \cdot \mathbf{u} \, dV = \int_{\Gamma} \mathbf{u} \cdot \mathbf{n} \, d\Gamma, \quad (16.25)$$

où $\mathbf{n}$ désigne la normale extérieure unitaire de Γ , et aussi :

$$\int_{\Omega} \nabla \phi \, dV = \int_{\Gamma} \phi \mathbf{n} \, d\Gamma, \quad (16.26a)$$

qui s'écrit composante par composante :

$$\int_{\Omega} \frac{\partial \phi}{\partial x_k} \, dV = \int_{\Gamma} \phi \mathbf{e}_k \cdot \mathbf{n} \, d\Gamma, \quad (16.26b)$$

où $\mathbf{e}_k$ désigne le vecteur unitaire dans la $k^{\text{ième}}$ direction.

La forme intégrale (ou conservative) de l'équation de continuité (16.20) s'écrit simplement :

$$\int_{\Gamma} \mathbf{v} \cdot \mathbf{n} \, d\Gamma = 0. \quad (16.27)$$

La forme conservative de l'équation de quantité de mouvement (16.23) s'écrit :

$$\frac{d}{dt} \int_{\Omega} \rho u_k \, dV + \int_{\Gamma} (\mathbf{F}_k^c - \mathbf{F}_k^d) \cdot \mathbf{n} \, d\Gamma + \int_{\Gamma} p \mathbf{e}_k \cdot \mathbf{n} \, d\Gamma = \int_{\Omega} \rho f_k \, dV, \quad (16.28)$$

avec la fonction de flux convectif :

$$\mathbf{F}_k^c = \rho u_k \mathbf{v} = \rho \begin{pmatrix} u_k u \\ u_k v \end{pmatrix}, \quad (16.29a)$$

et le flux diffusif :

$$\mathbf{F}_k^d = \frac{1}{\text{Re}} \mathbf{S}_k. \quad (16.29b)$$

Remarque 16.1. Il est intéressant de décomposer le flux diffusif dans la direction x comme

$$\mathbf{F}_1^d = \mathbf{F}_{1,\text{impl}}^d + \mathbf{F}_{1,\text{expl}}^d, \quad (16.30a)$$

avec

$$\mathbf{F}_{1,\text{impl}}^d = \frac{1}{\text{Re}} \nabla u, \quad \mathbf{F}_{1,\text{expl}}^d = \frac{1}{\text{Re}} \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial x} \end{pmatrix}. \quad (16.30b)$$

Une décomposition analogue est valable pour $\mathbf{F}_2^d$. Lorsque $\mu = C^{\text{te}}$ uniquement, en utilisant la conservation de la masse on obtient :

$$\int_{\Gamma} \mathbf{F}_{1,\text{expl}}^d \cdot \mathbf{n} \, d\Gamma = 0.$$

Ainsi, lorsque $\mu = C^{\text{te}}$, les flux diffusifs des équations de Navier-Stokes ont la même forme que dans l'équation de convection-diffusion (15.4). On peut alors utiliser directement les schémas d'intégration en temps implicites ou semi-implicites de la section 15.4. Pour utiliser ces schémas dans le cas $\mu \neq C^{\text{te}}$, il est recommandé de traiter le terme $\int_{\Gamma} \mathbf{F}_{1,\text{expl}}^d \cdot \mathbf{n} \, d\Gamma$ explicitement.

16.5 Conditions initiales et aux limites

Les équations de Navier-Stokes pour un fluide incompressible sont de nature parabolique, et des conditions aux limites et initiales convenables doivent être choisies. Ces conditions doivent répondre à deux besoins :

- Respecter la physique du problème à modéliser.

- Conduire à un problème mathématiquement bien posé, qui est tel que sa solution existe, est unique, et n'est pas sensible aux perturbations des données. Dans le cas contraire, son approximation numérique peut souffrir de graves problèmes de convergence et de stabilité.

Les conditions aux limites doivent être imposées en tout point de la frontière $\partial\Omega$ du domaine de calcul. En règle générale, pour les équations de Navier-Stokes 2D possédant 3 inconnues scalaires u , v et p , deux conditions doivent être imposées en chaque point¹ de $\partial\Omega$; pour l'équation de température (scalaire), une condition en chaque point de $\partial\Omega$ doit être imposée (cf. Fletcher [5]).

La discussion sur les conditions initiales est reportée à la section 16.5.3.

16.5.1 Conditions aux limites pour les écoulements confinés

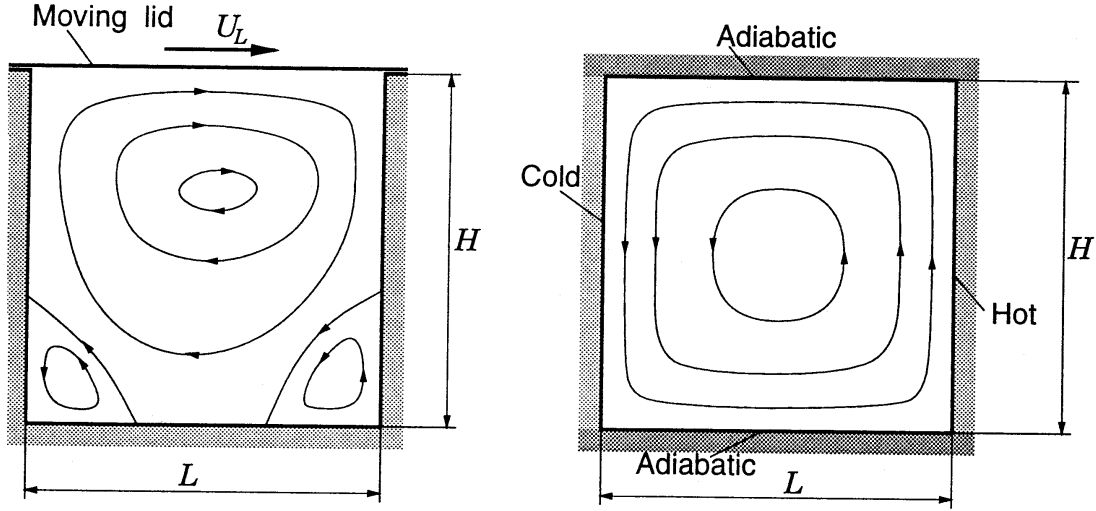


Fig. 16.1. Deux écoulements confinés classiques. *A gauche* : un fluide initialement au repos dans une cavité est entraîné par le mouvement de translation de la paroi supérieure, de vitesse $\mathbf{w}_L = (U_L, 0)$. On prescrit donc les conditions $\mathbf{v} = \mathbf{w}_L$ sur la paroi supérieure et les conditions d'adhérence $\mathbf{v} = \mathbf{0}$ sur les trois murs fixes. Il n'y a pas besoin de considérer d'équation pour la température.

A droite : écoulement de convection dans une cavité différentiellement chauffée, régi par les équations de Navier-Stokes dans l'approximation de Boussinesq. Les parois horizontales sont adiabatiques, ce qui est modélisé par une condition de flux nul pour la température ($g = 0$ dans la condition de Neumann (15.2b)). Un écart de température est imposé entre les deux parois verticales, modélisé par des conditions de Dirichlet non-homogènes de type (15.2a). Enfin, le fluide adhère aux parois ($\mathbf{v} = \mathbf{0}$ sur $\partial\Omega$).

Le cas le mieux maîtrisé mathématiquement concerne les écoulements à l'intérieur de domaines fermés. Deux exemples typiques sont représentés sur la figure 16.1. Pour ces écoulements, la frontière $\partial\Omega$ correspond à des parois solides sur lesquelles on impose généralement des conditions de Dirichlet pour le champ de vitesse :

$$\forall t, \quad \mathbf{v} = \mathbf{w} \quad \text{sur } \partial\Omega, \quad (16.31)$$

telle $\mathbf{w}$ est une vitesse donnée. Le cas d'une paroi fixe $\mathbf{w} = \mathbf{0}$ correspond à imposer des conditions de non glissement pour la composante tangentielle de $\mathbf{w}$ (i.e., $\mathbf{w} \times \mathbf{n} = 0$) et de non pénétration pour sa composante normale (i.e., $\mathbf{w} \cdot \mathbf{n} = 0$).

La donnée $\mathbf{w}$ sur le mur ne doit pas être choisie au hasard, et doit respecter le principe physique qu'il n'y a pas de création de masse dans le domaine Ω . Ce principe s'exprime mathématiquement par l'équation de continuité sous forme locale (16.20) ou conservative (16.27). En utilisant cette dernière équation, on en déduit une contrainte sur le choix de la condition aux limites :

$$\forall t, \quad \int_{\partial\Omega} \mathbf{w} \cdot \mathbf{n} \, d\Gamma = 0, \quad (16.32)$$

qui s'appelle *condition de compatibilité*. Cette condition est trivialement vérifiée pour les écoulements de la figure 16.1.

Pour l'équation de température (16.21), on prescrit des conditions de Dirichlet et Neumann, comme (15.2a) et (15.2b). Un exemple est donné dans la figure 16.1.

Enfin, il est important de rappeler qu'il n'y a pas de conditions aux limites sur la pression dans le cas de parois solides.

¹ En 3D il y a 4 inconnues scalaires et 3 conditions en chaque point de la frontière.

16.5.2 Conditions aux limites pour les écoulements externes

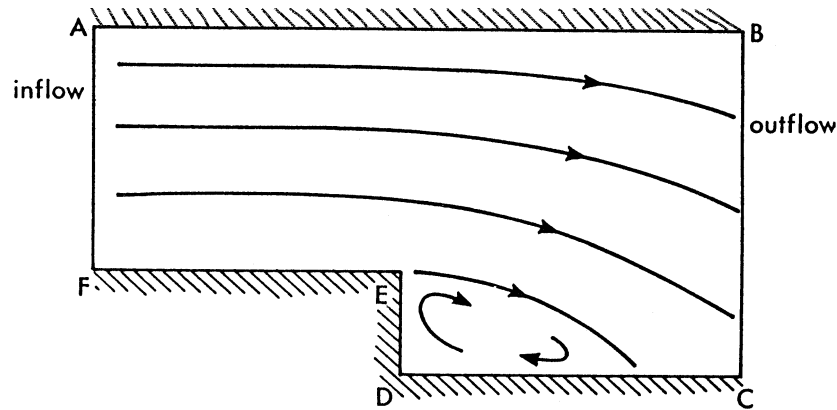


Fig. 16.2. Un exemple de domaine de calcul pour l'écoulement autour d'une marche descendante. Des conditions d'entrée à la frontière $\partial\Omega_{\text{in}} = AF$, et des conditions de sortie en $\partial\Omega_{\text{out}} = BC$. Les autres frontières correspondent à des parois solides où des conditions d'adhérence sont prescrites.

Le second cas important concerne les écoulements dits “externes” ou “ouverts”, tels que le domaine physique n'est pas entièrement délimité par des parois solides : le prototype est le problème de la marche représenté sur la figure 16.2. Pour ce type de configurations, il faut restreindre le domaine de calcul à une portion “pertinente” du domaine physique. En conséquence, la modélisation mathématique nécessite d'imposer des conditions aux limites artificielles sur les portions ouvertes du domaine (segments AF et BC dans la figure 16.2) même si, la plupart du temps, on ne dispose pas suffisamment d'informations sur la physique de l'écoulement pour formuler le bon nombre de conditions aux limites.

En première approximation, on considère deux types de frontières ouvertes : la partie $\partial\Omega_{\text{in}}$ où le fluide s'écoule à l'intérieur du domaine de calcul, *i.e.* :

$$\mathbf{v} \cdot \mathbf{n} < 0, \quad \text{sur } \partial\Omega_{\text{in}},$$

sur laquelle on impose des conditions d'entrée (“*inflow*”). Sur la partie $\partial\Omega_{\text{out}}$, où le fluide sort :

$$\mathbf{v} \cdot \mathbf{n} > 0, \quad \text{sur } \partial\Omega_{\text{out}},$$

des conditions de sorties (“*outflow*”) sont imposées. Ces conditions aux limites artificielles doivent être choisies de façon à obtenir un problème mathématiquement bien posé tout en influençant le moins possible la dynamique de l'écoulement. Nous nous bornerons dans ce cours à esquisser les conditions les plus communes, sans soucis d'exhaustivité.

Conditions d'inflow

Les expériences numériques et l'analyse mathématique montrent que à l'entrée du domaine, des conditions de Dirichlet ou de Neumann sur la vitesse peuvent être légitimement prescrites. Pour le problème de la marche dans le régime laminaire², on impose généralement un profil de Poiseuille pour la vitesse en $\partial\Omega_{\text{in}}$, de type (16.31) (voir aussi la Fig. 16.3).

Conditions d'outflow

La formulation de conditions de sortie est plus délicate. En premier lieu, elles doivent être le plus “transparente” possible, de façon à laisser sortir le fluide sans provoquer de phénomènes de réflexion parasite, qui influeraient sur l'écoulement amont. Un grand nombre de conditions furent et sont toujours proposées. Leur point commun est que :

- Comme énoncé plus haut, 2 conditions scalaires portant sur u , v ou p sont formulées en $2D$, et une seule pour T .
- Pour $\mathbf{v}$ et T , les conditions de Dirichlet sur $\partial\Omega_{\text{out}}$ conduisent généralement à des problèmes mal posés. On leur préfère des conditions de Neumann, qui ont l'avantage de n'affecter l'écoulement en amont de $\partial\Omega_{\text{out}}$ que dans une couche limite d'épaisseur $\mathcal{O}(1/\text{Re})$ pour $\text{Re} \gg 1$ (*cf.* [14]).

² Générer des conditions d'inflow turbulente est une autre histoire.

- Les conditions que l'on impose sur $\partial\Omega_{\text{in}}$ et $\partial\Omega_{\text{out}}$ ne doivent pas créer de la masse dans le domaine, *i.e.* la condition de compatibilité (16.32) doit être vérifiée numériquement.

Une classe de conditions d'outflow suppose que $\partial\Omega_{\text{out}}$ est placée à l'interface d'un écoulement externe dont on connaît *a priori* le tenseur des contraintes $[\boldsymbol{\sigma}]^\infty$. Comme dans les problèmes de surface libre, on suppose que le tenseur des contraintes (16.11), sous forme adimensionnelle, est continu au travers de l'interface :

$$[\boldsymbol{\sigma}] \cdot \mathbf{n} = [\boldsymbol{\sigma}]^\infty \cdot \mathbf{n}, \quad \text{sur } \partial\Omega_{\text{out}}. \quad (16.33)$$

Pour l'écoulement de la marche descendante, on fait l'hypothèse que suffisamment en aval de la marche, l'écoulement atteint un régime développé tel que

$$[\boldsymbol{\sigma}]^\infty = -p^\infty [\mathbf{I}],$$

où p^∞ est la pression atmosphérique extérieure. On déduit de (16.33) les deux conditions aux limites scalaires :

$$([\boldsymbol{\sigma}] \cdot \mathbf{n}) \cdot \mathbf{n} \equiv -p + \frac{2}{\text{Re}} S_{11} = -p^\infty, \quad \text{sur } \partial\Omega_{\text{out}}, \quad (16.34a)$$

$$([\boldsymbol{\sigma}] \cdot \mathbf{n}) \cdot \mathbf{t} \equiv \frac{2}{\text{Re}} S_{21} = 0, \quad \text{sur } \partial\Omega_{\text{out}}. \quad (16.34b)$$

Pour la température, on imposerait la continuité du flux de température au travers de l'interface, soit :

$$\frac{\partial T}{\partial \mathbf{n}} = q^\infty. \quad (16.35)$$

Ces conditions sont très utilisées dans la méthode des éléments finis.

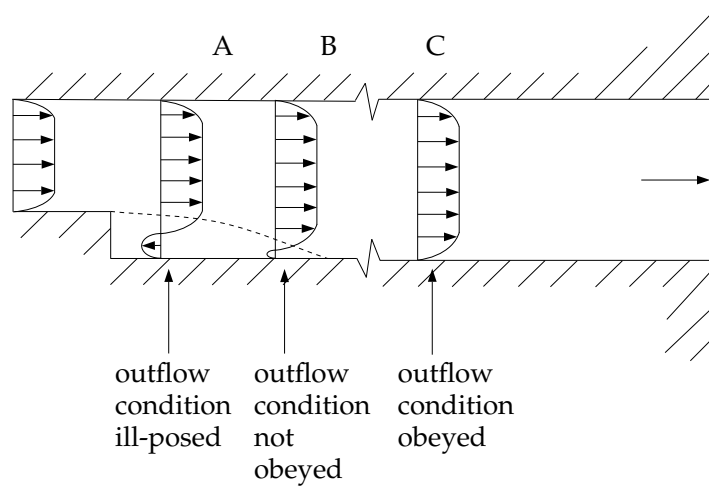


Fig. 16.3. Conditions d'inflow/outflow pour le problème de la marche descendante. En $\partial\Omega_{\text{in}}$, on impose un profil de Poiseuille. En $\partial\Omega_{\text{out}}$, on utilise la condition (16.36) par exemple. Si on place la frontière $\partial\Omega_{\text{out}}$ au niveau de la zone de recirculation (en A ou B par exemple), l'écoulement n'est pas unidirectionnel et un phénomène de "backflow" est observé. Des problèmes de convergence numérique ou des résultats erronés peuvent être observés. La frontière $\partial\Omega_{\text{out}}$ doit être positionnée au point C suffisamment loin de la marche pour que l'écoulement retrouve son profil unidirectionnel (cela peut entraîner un domaine de calcul exagérément long, de l'ordre d'une dizaine de fois la longueur de la marche).

Une simplification de (16.34) consiste à écrire simplement :

$$S_{11} = 0, \quad S_{21} = 0, \quad \text{sur } \partial\Omega_{\text{out}}, \quad (16.36)$$

ce qui correspond à annuler la composante des flux diffusif qui est normale à $\partial\Omega_{\text{out}}$. Les équations (16.36) et (16.35) correspondent aux conditions "Outflow" utilisée par *FLUENT*³. Une interprétation de ces conditions peut être faite en observant que, lorsque la viscosité est constante et d'après la remarque (16.1), Eq. (16.36) devient :

$$\frac{\partial u}{\partial x} = 0, \quad \frac{\partial v}{\partial x} = 0, \quad \text{sur } \partial\Omega_{\text{out}},$$

ce qui consiste à supposer qu'à la sortie du domaine l'écoulement est devenu unidirectionnel : cette approximation est valide suffisamment en aval de la marche (*cf.* Fig. 16.3). Si $\partial\Omega_{\text{out}}$ est situé trop près de la marche, on observe alors un phénomène de "backflow" : l'écoulement est susceptible de rentrer à l'intérieur du domaine de calcul. Il faut alors utiliser des conditions aux limites plus sophistiquées : *FLUENT* prétend gérer des phénomènes de backflow avec la condition "Pressure Outlet".

³ Cette condition est implémentée numériquement par extrapolation de la solution à l'intérieur du domaine physique, puis corrigée de manière à respecter la condition de compatibilité (16.32).

16.5.3 Conditions initiales

Pour l'équation de quantité de mouvement (16.22), de nature parabolique, il faut définir une conditions initiale du type :

$$\mathbf{v}(t = 0, \mathbf{x}) = \mathbf{v}_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega. \quad (16.37)$$

La donnée initiale doit vérifier certaines conditions afin que le problème soit mathématiquement bien posé (existence et unicité de la solution) :

- Il faut que $\mathbf{v}_0$ vérifie la contrainte d'incompressibilité :

$$\nabla \cdot \mathbf{v}_0 = 0, \quad \forall \mathbf{x} \in \Omega.$$

- Il faut que la composante *normale* de $\mathbf{v}_0$ vérifie les conditions aux limites, par exemple dans le cas (16.31) :

$$\mathbf{v}_0 \cdot \mathbf{n} = \mathbf{w} \cdot \mathbf{n}, \quad \text{sur } \partial\Omega.$$

Le non-respect de ces conditions lors d'une simulation numérique peut entraîner des difficultés de convergence et l'instabilité inconditionnelle du calcul.

Pour l'équation de température (16.21), également parabolique, on prescrit une condition initiale de la forme (15.3).

Enfin, il n'y a pas de conditions initiales sur la pression, car il n'y a pas de terme en $\partial p / \partial t$ dans les équations de Navier-Stokes.

16.6 Propriétés analytiques de conservation (ébauche)

Le mouvement d'un fluide visqueux est complètement décrit par le système de Navier-Stokes (16.22), (16.20) et (16.21). En particulier, il exprime le bilan ou loi de "conservation"⁴ de la quantité de mouvement

$$\int_{\Omega} \rho \mathbf{v} \, dV, \quad (16.38)$$

et de la masse

$$\int_{\Omega} \rho \, dV, \quad (16.39)$$

d'une particule fluide de volume Ω . Ces lois de conservations peuvent être qualifiées de "primaires" car elles sont directement exprimées par les équations (16.22) et (16.20) respectivement. Il existe d'autre lois de conservation importantes, comme celles gouvernant l'énergie cinétique

$$E_c = \int_{\Omega} e_c \, dV, \quad \text{avec } e_c = \frac{1}{2} \|\mathbf{v}\|^2 \quad (16.40)$$

et la vortacité

$$\int_{\Omega} \boldsymbol{\omega} \, dV, \quad \text{avec } \boldsymbol{\omega} = \nabla \times \mathbf{v}, \quad (16.41)$$

qui sont considérées comme "secondaires" car elles dérivent indirectement du système de Navier-Stokes : elles peuvent être construites analytiquement à partir du bilan de masse et de quantité de mouvement.

Construisons, par exemple, le bilan d'énergie cinétique pour un fluide de densité constante ($\rho = 1$). Dans ce cas, l'équation (16.22) s'écrit :

$$\frac{D\mathbf{v}}{Dt} = -\nabla p + \nabla \cdot \frac{2}{\text{Re}} [\mathbf{S}] + \rho \mathbf{f}. \quad (16.42)$$

en utilisant la conservation de la masse (16.20). Le produit scalaire de cette équation avec $\mathbf{v}$ conduit à l'équation de transport de l'énergie cinétique par unité de volume e_c :

$$\frac{De_c}{Dt} = -\mathbf{v} \cdot \nabla p + \mathbf{v} \cdot \nabla \cdot \frac{2}{\text{Re}} [\mathbf{S}] + \rho \mathbf{v} \cdot \mathbf{f}. \quad (16.43)$$

Dans le cas d'un écoulement non forcé ($\mathbf{f} = 0$ et conditions aux limites homogènes), examinons le taux de variation de l'énergie cinétique "globale" E_c : d'après (16.43), on obtient

$$\frac{dE_c}{dt} = \int_{\Omega} \frac{De_c}{Dt} \, dV = - \int_{\Omega} \mathbf{v} \cdot \nabla p \, dV + \int_{\Omega} \mathbf{v} \cdot \nabla \cdot [\mathbf{S}] \, dV. \quad (16.44)$$

⁴ Ces quantités sont en fait conservées dans la limite $\mu \rightarrow 0$.

Pour simplifier le terme de pression, on utilise l'identité (16.2) et la contrainte d'incompressibilité, ce qui conduit à :

$$\int_{\Omega} \mathbf{v} \cdot \nabla p \, dV = \int_{\Omega} \nabla \cdot p \mathbf{v} \, dV = \int_{\Gamma} p \mathbf{v} \cdot \mathbf{n} \, d\Gamma.$$

Pour le terme visqueux, on utilise l'identité vectorielle (Candel, p.122, Coirier, p. 439)

$$\nabla \cdot ([\mathbf{t}] \mathbf{u}) = \mathbf{u} \cdot \nabla \cdot [\mathbf{t}]^T + [\mathbf{t}] : \nabla \mathbf{u}, \quad \forall \mathbf{u}, [\mathbf{t}], \quad (16.45)$$

ce qui donne :

$$\int_{\Omega} \mathbf{v} \cdot \left(\nabla \cdot \frac{2}{\text{Re}} [\mathbf{S}] \right) dV = \int_{\Gamma} \left(\frac{2}{\text{Re}} [\mathbf{S}] \mathbf{v} \right) \cdot \mathbf{n} \, d\Gamma - \int_{\Omega} \frac{2}{\text{Re}} [\mathbf{S}] : \nabla \mathbf{v} \, dV.$$

L'équation (16.44) prend ainsi la forme :

$$\frac{dE_c}{dt} = \int_{\Gamma} ([\boldsymbol{\sigma}] \mathbf{v}) \cdot \mathbf{n} \, d\Gamma - \int_{\Omega} \hat{\Phi} \, dV,$$

où

$$\hat{\Phi} = \frac{2}{\text{Re}} [\mathbf{S}] : \nabla \mathbf{v},$$

est la fonction de dissipation visqueuse (*cf.* Candel). Cette fonction est toujours positive ou nulle : par exemple, si on suppose que la viscosité est constante, on obtient :

$$\hat{\Phi} = \frac{1}{\text{Re}} \nabla \mathbf{v} : \nabla \mathbf{v} = \frac{1}{\text{Re}} \|\nabla \mathbf{v}\|^2 \geq 0.$$

Pour un écoulement non forcé, pour lequel des conditions homogènes de type “wall” (16.31) ou “outflow” (16.34) sont utilisées, on observe donc que :

$$\frac{dE_c}{dt} = - \int_{\Omega} \hat{\Phi} \, dV \leq 0. \quad (16.46)$$

Ainsi, seules les forces visqueuses dissipent l'énergie cinétique en énergie interne : le transport convectif $\nabla \cdot \rho \mathbf{v} \mathbf{v}$ et la différence de pression ∇p n'interviennent pas dans ce bilan.

Une méthode numérique pour résoudre les équations de Navier-Stokes est dite “conservative” ou “non-dissipative” si elle conserve la contrepartie discrète de la quantité de mouvement (16.38), de la masse (16.39) et de l'énergie cinétique (16.40). La conservation de cette dernière quantité (*i.e.* , obtenir l'équivalent discret de (16.46)) est particulièrement importante pour la simulation des écoulements turbulents, afin de ne pas briser le subtil équilibre entre le transport convectif et la dissipation visqueuse décrit par la cascade de Kolmogorov (Sect. 11.2). Les propriétés de conservation des schémas numériques pour les équations de Navier-Stokes seront examinées dans la section *quidevraitexistermaisqu'existepasencore*. Mentionons dès à présent que l'ingrédient de base d'une bonne méthode numérique est la conservation de la masse, qui est une propriété cruciale pour obtenir la contrepartie discrète de la loi de “conservation” analytique (16.46).

DFV Discretization of the Navier-Stokes eqns: ①

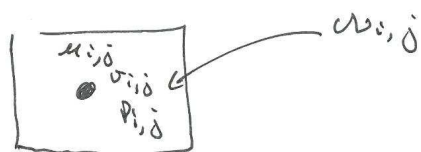
Navier-Stokes eqns:

18 Feb 2015
STIG

1) Arrangement of the variables on a mesh:

2 popular arrangements:

1-1) Collocated grid arrangement.



flow variables
- All ~~variables~~ or stored ("collocated") at the center of the cell.

⇒ Easy to generalize this approach to non-structured grids and to apply techniques for ~~best~~ ^{straightforward} lectures.

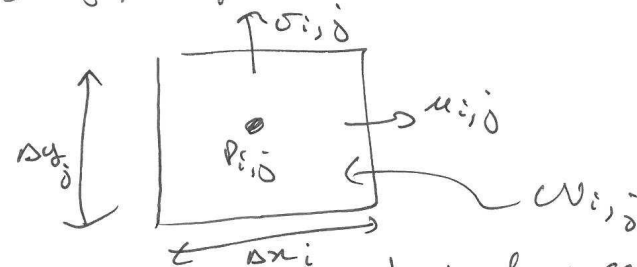
⇒ Most of commercial codes who use non-structured grids use collocated arrangement.

⇒ Problems related to ~~acoustic~~ ^{compressibility} properties.

② Staggered as we shall see next.

There is no problem related to pressure oscillations as we shall see next, which is related to the interpolation of pressure in collocated mesh. A common ^{straight} way to cure this pressure oscillations is to use a different interpolation of pressure (Moukhanov's interpolation of ρ & ρu , see next lecture), which has the effect of ~~decreasing~~ ^{improving} the compressibility properties of the collocated approach.

2) Staggered grid arrangement ②



Velocities are stored at face centers.
- Pressure and other ~~scalar~~ ^{mass} scalar variables (Temperature, etc...) are stored at cell centers.

⇒ The continuity equation is discretized in a straightforward way without any interpolation, i.e.:

$$\dot{m}_e \approx \Delta y u_{i,j}$$

⇒ It is the first method for solving the NS eqns in velocity-pressure formulation. (MAC Meth of Harlow & Welch, 1964).

⇒ Generalization of mass, momentum, etc. is easily verified, ^{if it is a numerical method of NS eqns} there is no problem of pressure oscillations as in the collocated arrangement.

But:

• Since the variables ($u_{i,j}$, $v_{i,j}$, $p_{i,j}$) are stored at different locations in the cell, we must define $\neq$ control volumes for the momentum eqns in x and y directions, mass eqn.
⇒ Coupling the staggered method is more involved.

In the beginning of CFD 1970-1980, there were conceptual difficulties in generalizing the staggered method to non-structured mesh (3)
 $\Rightarrow$ Collocated methods were preferred, that is why current commercial CFD codes (e.g. FLUENT) still uses collocated method, despite of its flaws.

$\Rightarrow$ Staggered method or still be preferred choice when structured grid are used.

2) FVM for NS eqt on collocated grid (4)

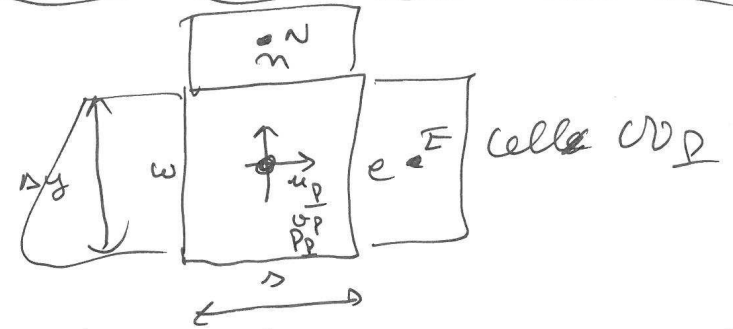


Fig 1: cell CV_P
 We will consider the case of a uniform mesh with cell $\Delta x \times \Delta y \Rightarrow V_P = \Delta x \Delta y$. The U-P variables are stored in the cell centers, i.e.:

$$\phi_P = \frac{1}{V} \int_{CV_P} \phi dV \quad \text{for } \phi = u, v, \text{ and } p.$$

The integral (or conservative) form of the NS eqts is:
 For $h = u, v$ ($\rho(u_h)_P = \rho_P$, $(u_h)_P = u_P$):
 Conservation of momentum:
 (1) $\frac{d}{dt} \left(\rho u \right)_P + \frac{1}{V_P} \int_{\Gamma_P} (F_h^c - F_h^d) \cdot \vec{n} dS + \frac{1}{V} \int_{\Gamma_P} p \vec{e}_h \cdot \vec{n} d\Gamma = f_h$

for $h = u, v$.

where for putting the pressure term in integral form we have used Green's theorem:

$$\int_{CV} \nabla \phi dV = \int_{\Gamma} \phi \vec{n} dS \Leftrightarrow \int_{CV} \frac{\partial \phi}{\partial x_h} dV = \int_{\Gamma} \phi \vec{e}_h \cdot \vec{n} d\Gamma$$

where $\vec{e}_h$ is the unit vector in the h th direction.

The convective and diffusive fluxes are respectively: (5)

$$(2) \quad F_h^c = \rho \vec{v} u_h, \quad F_h^d = V \frac{D}{Dt} = \frac{1}{2} (\underline{D} \vec{v} + \vec{v} \underline{D})$$

Conservation of mass:

$$\int_{\Omega} \vec{v} \cdot \vec{n} dS = 0. \quad (3)$$

To these NS equations, we must add boundary conditions on the boundary Γ (see lecture ---), and initial conditions for the velocity at $t=0$.

The NS eqt (1)-(3) needs to be discretized on the CV Ω_P of Reg. 1.

Some remarks on the N-S eqts (1)-(3):
They are really intended to be a continuum description except that:
• There is no explicit equation for pressure,
• and no initial conditions.

• There is no initial conditions for pressure.

• The incompressibility constraint:

$$\int_{\Omega} \vec{v} \cdot \vec{n} dS = 0 \Leftrightarrow \nabla \cdot \vec{v} = 0.$$

must be fulfilled at each time. This is a real difference with the con-diff eqt, and we will see next that the pressure will be computed such as this incompressibility constraint is fulfilled at each time-step.

2-1) Discretization of the continuity eqt: (6)

By applying the midpoint rule to the 4 faces of control volume Ω_P , we get:

$$\int_{\Omega_P} \vec{v} \cdot \vec{n} dS \approx \Delta y (u_e - u_w) + \Delta x (v_n - v_s) \quad (4)$$

The velocity at the face centers need now to be interpreted to the center of the CV. This is done straightforwardly with central interpolation:

$$u_e \approx \frac{1}{2} (u_P + u_E), \quad v_n \approx \frac{1}{2} (v_P + v_N) \quad (5)$$

We finally get:

$$\begin{aligned} \int_{\Omega_P} \vec{v} \cdot \vec{n} dS &= \Delta y \frac{(u_E - u_W)}{2} + \Delta x \frac{(v_N - v_S)}{2} \\ &= V \left(\delta_x^0 u_{i,j} + \delta_y^0 v_{i,j} \right) \quad (6) \end{aligned}$$

by using stencil notation for centered FD formulae.

→ Discrete system of PDEs: $\approx \approx \approx$

2-2) Secur. discretization of the momentum eqts:

For Eq. (1), the discretization of conv & diff flux follow the lines of lecture 1 on con-diff eqt, what remains is the pressure term; which reads in the x-dim:

$$\begin{aligned} \frac{1}{V} \int_{\Omega_P} p \vec{e}_x \cdot \vec{n} dS &= \frac{1}{V} \left[\int_{\Omega_e} p dV - \int_{\Omega_w} p dS \right] \\ &\approx \frac{1}{\Delta x} [p_e - p_w] \quad (7) \end{aligned}$$

We use standard central interpolation: (1)

$$P_c \approx \frac{1}{2} (P_E + P_W), \quad P_w \approx \frac{1}{2} (P_E + P_W) \quad (8)$$

and we finally get:

$$\frac{1}{V} \int_{P_P} p \cdot \vec{e}_x \cdot \vec{n} ds \approx \frac{(P_E - P_W)}{2\Delta x} = \sum_n P_{i,j} \quad (9)$$

and for the y direction:

$$\frac{1}{V} \int_{P_P} p \cdot \vec{e}_y \cdot \vec{n} ds \approx \frac{(P_N - P_S)}{2\Delta y} = \sum_n P_{i,j} \quad (10)$$

In matrix formulation, the sea-discrete NS eqs read:

$$\frac{d}{dt} U + \left(\frac{G_h(U) - K_h}{N_h} \right) U + G_h P = F \quad (11a)$$

$$D_h U = 0 \quad (11b)$$

where the unknowns are:

$$U = (u_1, u_2, \dots, u_N, v_1, \dots, v_N) = (u_l, v_l)_{l=1 \dots N_h}$$

$$P = (P_1, \dots, P_N)_{l=1 \dots N_h}$$

Since the collocated mesh has $l=1, \dots, N$ unknowns ($N=N_h N_y$).

Note that the velocity mesh have $2N$ unknowns in 2D.

The matrix G_h represents the discrete pressure gradient, such as:

$$G_h P = \begin{bmatrix} \delta_x^0 \\ \delta_y^0 \end{bmatrix} P = \begin{pmatrix} \delta_x^0 P_x \\ \delta_y^0 P_y \end{pmatrix}$$

The matrix D_h is the discrete divergence, such as:

$$D_h U = \begin{bmatrix} \delta_x^0 & \delta_y^0 \end{bmatrix} U = (\delta_x^0 u_l + \delta_y^0 v_l)_l$$

3) An "explicit" time-stepping scheme for the NS eqs - the Semi-Discrete EP1 (SDEP1). (8)

Since Eq. (11a) is very similar to the cur-diff Eq, we let's apply the EP1 scheme:

$$\xrightarrow{2N \text{ eqs}} \frac{U^{n+1} - U^n}{\Delta t} + N_h U^n + G_h P^n = F^n \quad (12a)$$

Now, observe that the divergence eq has not ~~temporal~~ time derivatives. It must be resolved at each current time-step t^{n+1} , i.e:

$$N_h P^n \rightarrow D_h U^{n+1} = 0 \quad (12b)$$

~~this eq~~ ^{system} ~~has~~ ^{then} ~~2N eqs~~ ^{then} ~~3N unknowns: U^{n+1}~~ ^{and P^n} .
Note that this system has ~~3N unknowns~~ ^{3N unknowns} $U^{n+1} = \begin{pmatrix} u^{n+1} \\ v^{n+1} \end{pmatrix}$ and P^n .
This system has ~~3N eqs~~ ^{3N eqs}, so we must consider ~~3N unknowns~~ ^{3N unknowns} $U^{n+1} = \begin{pmatrix} u^{n+1} \\ v^{n+1} \end{pmatrix}$ and P^n which must be solved simultaneously. As we shall see next, this system is explicit only for the velocity, ~~and we~~ must derive a ~~pressure~~ ^{pressure} eq for the pressure in order to get P^n .

3-1] the EP1 as a fractional step scheme

For ease of computation, it is customary to rewrite EP1 scheme (12a) - (12b) as a 2-step scheme, by introducing the auxiliary variable $\tilde{U}$, as:

$$\frac{\tilde{U} - U^n}{\Delta t} + N_h U^n = F^n \quad (13)$$

$$\text{Then: } \begin{cases} \frac{U^{n+1} - \tilde{U}}{\Delta t} + G_h P^n = 0 & (14a) \\ D_h U^{n+1} = 0 & (14b) \end{cases}$$

After cancel of (14a) and (14b) to get rid of U^{n+1} , we obtain the pressure eqⁿ for the pressure. (8)

$$D_h G_h P^n = \frac{1}{\Delta t} \sum D_h \vec{U}. \quad (15)$$

Algorithm (EPI schem for NS eqⁿ)

The soln of EPI schem (13)-(14) is actually done by the follow steps:

1) Initialization: at $t=0$: U^0 is given by initial cond.

2) Evolve from t^n to $t^{n+1} = t_n + \Delta t$. The soln (U^{n+1}, P^n) is obtained by the current steps:

a) Apply Eq. (13) to get ~~a predicted~~ the predicted velocity $\vec{U}$:

$$\vec{U} = U^n + \Delta t (F^n - A_h U^n) \quad (16a)$$

Note that $\vec{U}$ is not divergence-free, i.e. $\nabla \cdot \vec{U} \neq 0$.

b) Compute P^n by solving the pressure eqⁿ (15).

c) Now that P^n is known, use Eq (14a) to get U^{n+1} :

$$U^{n+1} = \vec{U} - \Delta t G_h P^n \quad (16b)$$

It can be verified a posteriori that U^{n+1} is divergence-free, i.e. $D_h U^{n+1} = 0$.

This algorithm is the prototype of the pressure-correction method, where P^n is computed and used to correct the velocity (16b) so that it is divergence-free at each time step. It means mass is conserved along the time integration. If it is not the case, the time-integration would blow up.

As a consequence, the EPI schem is explicit only

3-2) the Pressure Poisson Equation for the Pressure (PPE) (2)

Last paragraph showed that the EPI scheme (12) is explicit only for the velocity, and the pressure must be solved "implicitly" by the solution of the PPE:

(17) $A_h D_h G_h P^{n+1} = \sum D_h \vec{U}$, where $A_h = D_h G_h$ and $\sum$ depends on the time-stepping scheme. The explicit nature of the calculation of P is a common characteristic of all numerical methods for incompressible flows that needs at any time step to solve eqⁿ (17).

The pressure operator $A_h = D_h G_h$ is the product of the discrete divergence and gradient, so it should look like a discrete gradient. Let's have a formula:

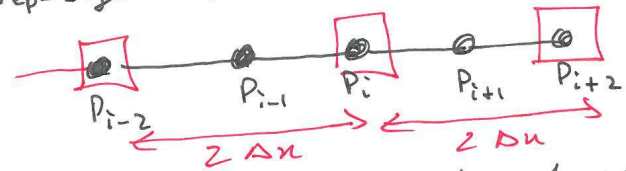
$$D_h G_h P^l = \begin{bmatrix} \delta_x^0 & \delta_y^0 \end{bmatrix} \begin{bmatrix} \delta_x^0 \\ \delta_y^0 \end{bmatrix} = \delta_x^0 \delta_x^0 P^l + \delta_y^0 \delta_y^0 P^l. \quad (17)$$

Note that $\delta_x^0 \delta_x^0 \neq \delta_{xx}^0$, but rather:

$$\delta_x^0 (\delta_x^0 P_{i,j}) = \frac{P_{i+3,j} - 2P_{i,j} + P_{i-2,j}}{(2\Delta x)^2} \quad (18a)$$

$$\delta_y^0 (\delta_y^0 P_{i,j}) = \frac{P_{i,j+2} - 2P_{i,j} + P_{i,j-2}}{(2\Delta y)^2} \quad (18b)$$

which shows that A_h corresponds to a discrete Laplacian of step-size $2\Delta x, 2\Delta y$, which skips a grid point:



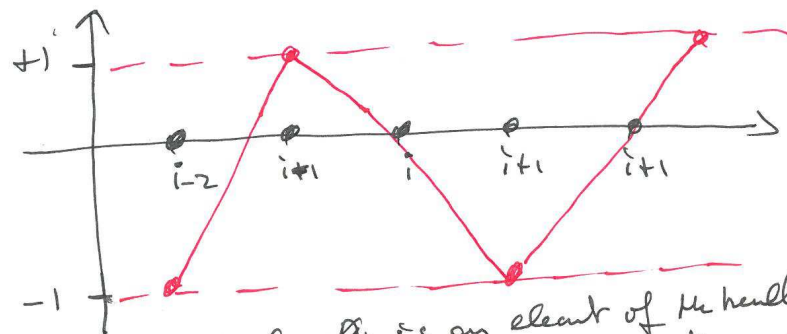
the skipping of every other grid point is dangerous, as we shall see next.

First, let us recall that the term: (10)

$$\pi_{i,j}^1 = cte \quad (15)$$

Cancels the product $A_h \pi_{i,j}^1$. It is the discrete version of the fact that pressure is defined up to a constant value. Math, it means that $\pi_{i,j}^1$ lies in the kernel of values A_h .

Let us also observe that term $\pi_{i,j}^2 = (-1)^i$, (20a)



also ~~cancel~~ A_h is an element of the kernel of A_h .
Same for the term $\pi_{i,j}^3 = (-1)^j$ and $\pi_{i,j}^4 = (-1)^{i+j}$. (20b)

Hence, the solution of the PPE (17) has the form:

$$P_{i,j} + \sum_{m=1}^4 \alpha_m \pi_{i,j}^m$$

and is thus polluted by the spurious pressure terms π^2, π^3 and π^4 that make goods a note-to-note pressure oscillations $\Rightarrow$ this is the so-called Pressure checkerboard phenomenon.

Hence, the ~~use~~ collocation formulation yields a pressure solution which has oscillate, rendering this this method ~~is~~ unusable in practice. There are nonetheless 2 remedies:

- 1) Use the staggered formulation, where we see in the next lecture. (11)
- 2) Modify the ~~discretization~~ ~~in order~~ discretization of the continuity eq (4) in order to get a PPE which is free of pressure oscillations. This is the basis of the unimodular weighted unimodular of $\nabla \cdot \mathbf{c}$ (see lecture) which is used by all commercial codes that use collocation approach.

25 Feb 2015

(12)

12) FVM for NS eq^t on staggered grid

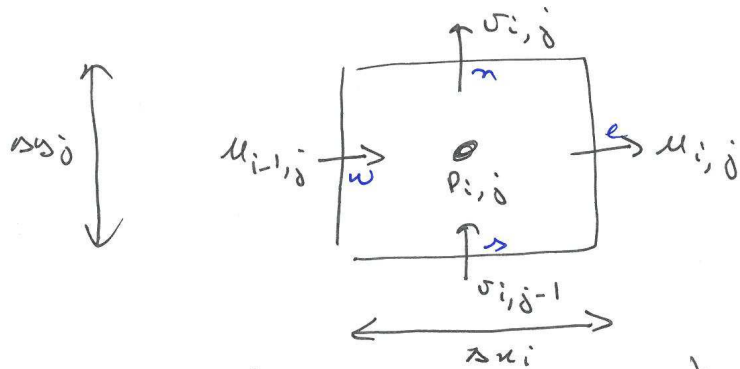


Fig 1: Control volume V_P for continuity eq^t:

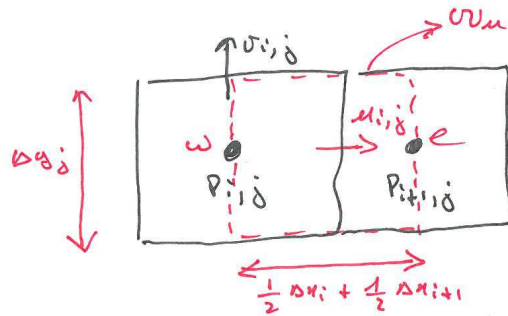


Fig 2: Staggered control volume V_u for the u -momentum eq^t. Its width $V_u = (\frac{1}{2} \Delta x_i + \frac{1}{2} \Delta x_{i+1}) = \Delta x$ on a uniform mesh.

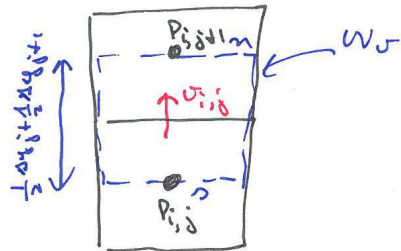


Fig 3: Staggered control volume V_v for the v -momentum eq^t. $V_v = \Delta x \Delta y$ on a uniform mesh.

(13)

13-1) Discretization of the continuity eq^t

As in Sect 2.1, the discretization of the continuity eq^t is:

$$\int_{P_P} \vec{v} \cdot \vec{n} dS \equiv \Delta y (u_e - u_w) + \Delta x (v_n - v_s) \quad (21)$$

there is no need to do further interpolation since $u_e = u_{i,j}$, $v_n = v_{i,j}$, etc... so we finally get:

$$\int_{P_P} \vec{v} \cdot \vec{n} dS \equiv V_P (\bar{S}_x u_{i,j} + \bar{S}_y v_{i,j}) \quad (22)$$

by using the FD formulas:

$$\bar{S}_x \phi_{i,j} = \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x}, \quad \bar{S}_y \phi_{i,j} = \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y} \quad (23)$$

N.B: It does not mean that upwind biased formulae are used for calculating the diffusion divergence! everything is computed at face center with central formulae.

13-2) Semi-discretization of the momentum eq^t

The u -momentum eq^t needs to be discretized on CV V_u of Fig 2, ie:

$$\frac{d}{dt} u_{i,j} + \frac{1}{V_u} \int_{P_u} (F_x^c - F_x^d) \cdot \vec{n} dS + \frac{1}{V_u} \int_{P_u} \vec{p} \cdot \vec{e}_n \cdot \vec{n} dS = 0 \quad (24)$$

The discretization of F_x^c and F_x^d will be detailed in an exercise.

The pressure gradient needs to be approximated by:

$$\frac{1}{V_n} \int_{P_n} \vec{p} \cdot \vec{e}_n \cdot \vec{n} dS = \frac{1}{\Delta x \Delta y} \left[\frac{\int p dx}{\Delta y P_e} - \frac{\int p dy}{\Delta y P_w} \right] \quad (24)$$

$$= \frac{1}{\Delta x} (P_{i+1,j} - P_{i,j}) = \delta_n^+ P_{i,j} \quad (25)$$

As for the continuity eq^t, there is no need to make further interpolatⁿ $\Rightarrow$ big difference with the collocated arrangement (see Sect. 2.2).

Note that the pressure gradient for the y-momentum eq^t (Fig. 5) reads:

$$\frac{1}{V_o} \int_{P_o} \vec{p} \cdot \vec{e}_y \cdot \vec{n} dS = \delta_y^+ P_{i,j} \quad (26)$$

3-3) The ED1 scheme on the PDE for staggered mesh.

As for the collocated mesh (Sect 3.), we will use the ED1 scheme (12), or equivalently the fractional-step sch^m (13)-(14), where now the pressure and divergence unknown are:

$$D_h = \begin{bmatrix} \delta_n^- & \delta_n^- \end{bmatrix}, \quad G_h = \begin{bmatrix} \delta_n^+ \\ \delta_y^+ \end{bmatrix} \quad (27)$$

and the PDE (19) reads: pressure the following pressure eqⁿ:

$$D_h G_h p_{\ell} = (\delta_n^- \delta_n^+ + \delta_y^- \delta_y^+) p_{\ell} \quad (28)$$

and it is well known that (see FD exercise 1)

$$\delta_n^- \delta_n^+ = \delta_{nn}^0, \quad \delta_y^- \delta_y^+ = \delta_{yy}^0 \quad (29)$$

So that the PDE now looks like a standard Poisson equation with step-size Δx and Δy :

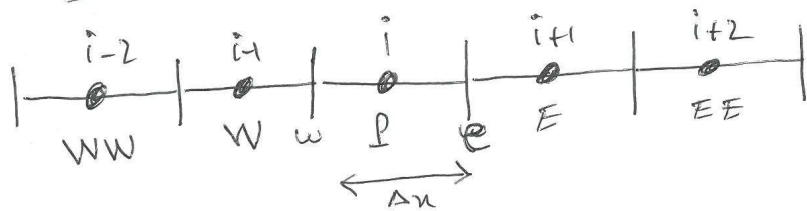
$$\frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{\Delta x^2} + \frac{P_{i,j+1} - 2P_{i,j} + P_{i,j-1}}{\Delta y^2} = \frac{1}{\Delta t} (\delta_n^- \bar{u}_{i,j} + \delta_y^- \bar{v}_{i,j}) \quad (30)$$

We can now readily observe that the spurious pressure terms (20) of the collocated method are no more part of the kernel, only $\bar{u}_{i,j} = c_k$ cancels Eq. (30).
the physical pressure mode.

$\Rightarrow$ There is no spurious pressure oscillations in the staggered method!

Rhee - Chow momentum-weighted interpolation for collocated variable arrangement

①



1) Statement of the problem: pressure checkerboarding

In cell V_P the continuity eq. reads for face velocities:

$$\Delta y (u_e - u_w) + \Delta x (v_n - v_s) = 0 \quad (1)$$

In the collocated arrangement (pressure and velocities stored at cell center), the face velocities need to be interpolated. When we use central interpolation, as the one used for the discretization of momentum Eqs:

$$u_e = \frac{1}{2} (u_E + u_P), \quad (2)$$

the continuity eq. reads:

$$V_P \left[\underbrace{\frac{(u_E - u_W)}{2\Delta x} + \frac{(v_N - v_S)}{2\Delta y}}_{m_P} \right] = 0 \quad (3).$$

The PPE is constructed from the projection step (18.21a) of the EP1 scheme:

②

$$u^{n+1} = u^n - \Delta t \left(\frac{\partial P}{\partial x} \right)_E \Rightarrow u_E^{n+1} = u_E^n - \Delta t \left(\frac{\partial P}{\partial x} \right)_E \quad (4)$$

with $\left(\frac{\partial P}{\partial x} \right)_E = \frac{P_{EE} - P_P}{2\Delta x}$. When inserting Eq. (4)

into the continuity eq. (3), the PPE reads:

$$\left[\frac{P_{EE} - P_P}{2\Delta x} - \frac{P_P - P_{WW}}{2\Delta x} \right] + \left[\dots \right] = \Delta t \nabla \cdot u \quad (5)$$

The PPE amounts to a discrete Laplacian which uncouples odd nodal values (P_{WW}, P_P, P_{EE}) from even nodal values $(P_W, P_E, \dots)$. In other words, this odd-even decoupling or checkerboarding effect leads to oscillations in the pressure field in the form $(-1)^i$, $(-1)^j$ and $(-1)^{i+j}$ in 2D, which renders the collocated method unsuitable for actual CFD computations.

2) Momentum-weighted (MWI) interpolation

Rhee & Chow (1983) observed that pressure checkerboarding is caused by the central ~~face~~ interpolation (2) that straddles the ^{PPE} pressure stencil. They proposed a modification of (2) that uses pressure nodal values as weights in the fashion of the R-scheme for convection. This pressure

- weighted interpolation (PWI) is also known ③
as momentum-weighted interpolation (MWI)
because it is based on the discretized momentum
eqⁿ. that reads for cells U_P and U_E :

$$a_P u_P + \left(\frac{\partial P}{\partial x} \right)_P = - \left(\sum_{nb} a_{nb} u_{nb} \right)_P \quad (6a)$$

$$a_P u_E + \left(\frac{\partial P}{\partial x} \right)_E = - \left(\sum_{nb} a_{nb} u_{nb} \right)_E \quad (6b).$$

Note that the coefficients of the discretization ~~are~~
(a_P , a_{nb}) are identical in the case of uniform mesh.

their idea is to use momentum discretization (6)
to interpolate face velocity u_e as:

$$\begin{aligned} a_P u_e + \left(\frac{\partial P}{\partial x} \right)_e &= - \left(\sum_{nb} a_{nb} u_{nb} \right)_e \\ &= - \frac{1}{2} \left(\sum_{nb} a_{nb} u_{nb} \right)_E - \frac{1}{2} \left(\sum_{nb} a_{nb} u_{nb} \right)_P \\ &= \frac{1}{2} a_P u_P + \frac{1}{2} \left(\frac{\partial P}{\partial x} \right)_P + \frac{1}{2} a_P u_E + \frac{1}{2} \left(\frac{\partial P}{\partial x} \right)_E \end{aligned}$$

by using (6a) and (6b). Finally, the MWI for u_e reads:

$$u_e = \frac{1}{2} (u_E + u_P) + \frac{1}{a_P} \left[\frac{1}{2} \left(\frac{\partial P}{\partial x} \right)_P + \frac{1}{2} \left(\frac{\partial P}{\partial x} \right)_E - \left(\frac{\partial P}{\partial x} \right)_e \right] \quad (7)$$

with the pressure gradient at CV center evaluated as:

$$\left(\frac{\partial P}{\partial x} \right)_P = \frac{P_E - P_W}{2 \Delta x} \quad (8a)$$

and at face center as:

$$\left(\frac{\partial P}{\partial x} \right)_e = \frac{P_E - P_P}{\Delta x} \quad (8b) \quad \text{④}$$

The MWI interpolation (7) amounts to the central
interpolation (2) with additional pressure terms.

If we switch to index notations, Eq. (7) reads:

$$\begin{aligned} u_e &= \frac{1}{2} (u_i + u_{i+1}) + \frac{1}{a_P} \left[\frac{1}{2} \left(\frac{P_{i+1} - P_{i-1}}{2 \Delta x} \right) + \frac{1}{2} \left(\frac{P_{i+2} - P_i}{2 \Delta x} \right) - \left(\frac{P_{i+1} - P_i}{\Delta x} \right) \right] \\ &= \frac{1}{2} (u_i + u_{i+1}) + \frac{1}{4 a_P \Delta x} [P_{i+2} - 3 P_{i+1} + 3 P_i - P_{i-1}] \quad (9) \end{aligned}$$

In this interpolation, we recognize the non-centred
finite-difference quotient:

$$\left(\frac{d^3 P}{dx^3} \right)_i = \frac{P_{i+2} - 3 P_{i+1} + 3 P_i - P_{i-1}}{\Delta x^3} + O(\Delta x), \quad (10)$$

so that the PWI interpolation is equivalent to:

$$u_e = \frac{1}{2} (u_i + u_{i+1}) + \frac{\Delta x^2}{4 a_P} \left(\frac{\partial^3 P}{\partial x^3} \right)_i \quad (11)$$

the PWI
and amounts to add a small regularization term in
 $O(\Delta x^2)$ to the central interpolation (2), with the aim
of suppressing the checkerboard oscillations.
When the MWI (7) is used for discretizing
the continuity eqⁿ (1), we obtain:

$$\begin{aligned} \Delta y (u_e - u_w) &= \Delta y \left[\frac{1}{2} (u_i + u_{i+1}) + \frac{1}{4 a_P \Delta x} (P_{i+2} - 3 P_{i+1} + 3 P_i - P_{i-1}) \right. \\ &\quad \left. - \frac{1}{2} (u_{i-1} + u_i) + \frac{1}{4 a_P \Delta x} (P_{i+1} - 3 P_i + 3 P_{i-1} - P_{i-2}) \right] \end{aligned}$$

$$\Delta y (u_e - u_w) = V_p \left[\frac{u_E - u_W}{2 \Delta n} + \frac{1}{4 q_p \Delta x^2} (P_{i+2} - 4P_{i+1} + 6P_i - 4P_{i-1} + P_{i-2}) \right] \quad (5)$$

(12)

where the centred finite difference quotient appears:

$$\left(\frac{d^4 p}{dx^4} \right)_i = \frac{P_{i+2} - 4P_{i+1} + 6P_i - 4P_{i-1} + P_{i-2} + O(\Delta x^2)}{\Delta x^4}$$

with MWI interpolation.

The continuity eqⁿ is thus equivalent to:

$$\dot{m}_p + \frac{1}{4 q_p} \left[\underbrace{\Delta x^2 \left(\frac{\partial^4 p}{\partial x^4} \right)_i}_{\text{dissipative terms}} + \Delta y^2 \left(\frac{\partial^4 p}{\partial y^4} \right)_i \right] = 0 \quad (13)$$

which has to be compared with the continuity Eq. (3) obtained with central interpolation. The PWI has the effect to add a dissipative term to the continuity eqⁿ in order to damp the checkerboarding oscillations. In effect, checkerboarding is absent from the PPE obtained with PWI since spurious pressure solutions of the form $(-1)^i$ do not cancel in Eq. (12).

Concluding remarks: The PWI (or MWI) is an effective way to get rid of the checkerboarding problem of the collocated method, and is widely used in commercial codes such as FLUENT. The regularizing terms in Eq. (15) and (12) do not alter the truncation error of the collocated method (in $O(\Delta x^2)$) but they may have an effect on its stability. Ventures of the PWI (7) have been devised for the case of

non-uniform grids, non-rectangular (unstructured) cells, unsteady NS eqs, presence of body terms, etc.... (6)

We nonetheless observe that PWI affects the conservation properties of the method since Eq. (13) shows that total mass is not conserved during a computation.

Basics of FVM on unstructured 2D meshes

1) Introduction

We will review the FV discretization of the unsteady convection-diffusion equation on 2D unstructured meshes:

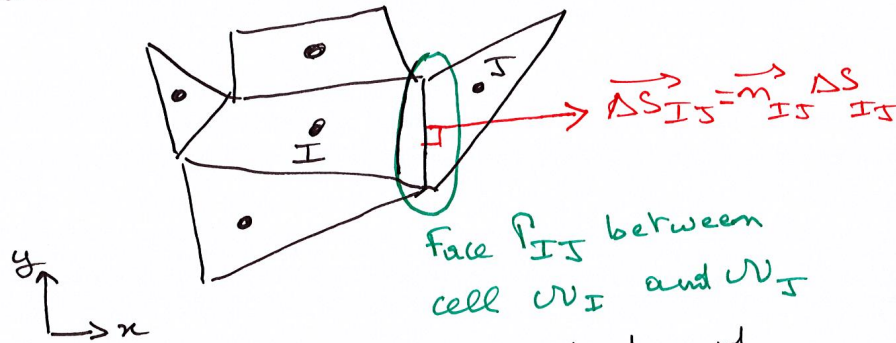


Fig 1: Example of a 2D unstructured grid. This grid is hybrid (with Tri and Quad cells).

Unstructured mesh in 2D have:

- Quadrangle cells ("Quad") with 4 faces, and 4 neighboring cells.
- Triangular cells ("Tri") with 3 faces

We consider only the cell-centred FV method, where the unknown Φ_I is located at the cell centers (centroids, barycenters).

(1)

2) Discretization of the convection-diffusion Eq^t

The integral form of the convection-diffusion in cell V_I is:

$$\frac{d}{dt} \int_{V_I} \phi dV + \int_{P_I} \vec{F} \cdot \vec{\Delta S} = \int_{V_I} q dV \quad (1)$$

where:

- $\vec{F} = \vec{F}^c - \vec{F}^d$ are the convection and diffusive flows:
 - $\vec{F}^c = \phi \vec{u}$
 - $\vec{F}^d = \nu \vec{\nabla} \phi$

• $\vec{\Delta S}$ is the area-vector of the faces of V_I , such that:

$$\vec{\Delta S}_{IJ} = \vec{n}_{IJ} \Delta S_{IJ} \quad (2)$$

with:

- $\vec{n}_{IJ}$ is the normal vector of the face between I and J ($\|\vec{n}_{IJ}\| = 1$)
- ΔS_{IJ} is the area of the face.

By using mid-point rule, the FV discretization of Eq. (1) is:

$$V_I \frac{d}{dt} \phi_I + \sum_{\text{faces of } V_I} \vec{F}_f \cdot \vec{\Delta S}_f = V_I q_I \quad (3)$$

where, for example,

$$\vec{F}_f \cdot \vec{\Delta S}_f = F_x(x_f) \Delta S_f^x + F_y(x_f) \Delta S_f^y$$

with $\vec{\Delta S}_f = \begin{pmatrix} \Delta S_f^x \\ \Delta S_f^y \end{pmatrix}$ and x_f is the center of the face.

Example: On the rectangular cell of Fig. 2

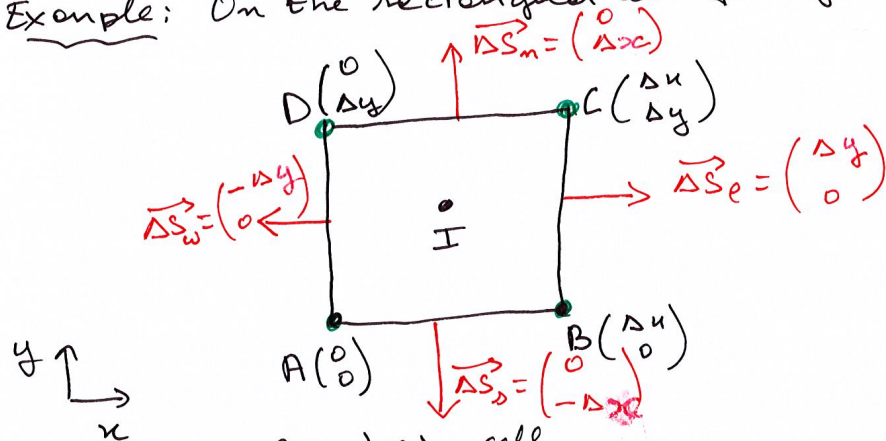


Fig. 2 A Cartesian cell
we recover the usual FV discretization on structured grid:

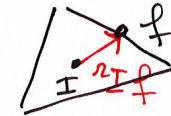
$$(4) \quad \frac{d}{dt} \phi_I + \frac{(F_x)_e - (F_x)_w}{\Delta x} + \frac{(F_y)_n - (F_y)_s}{\Delta y} = q_I$$

3) Calculation of the geometry parameters of a cell

In a structured mesh, we only need to store in the PC memory the grid size ($\Delta x, \Delta y$) to perform a FV discretization such as (4).

To perform the discretization (2) on an unstructured mesh, we need the following geometry parameters of the cells:

- Volume V_I
- N_f , the number of faces of V_I
- For all faces, the face-area vectors $\vec{\Delta S}_f$
- $\vec{r}_{If}$, the vector joining the center of cell V_I to the center of the face.



Since these geometry parameters are not as easily calculated as in a Cartesian mesh, they need to be stored in the PC memory.

example: For a 2D quad mesh, we store for each cell:

- the volume V_I
- the face-area vectors $\begin{pmatrix} \Delta S_f^x \\ \Delta S_f^y \end{pmatrix}$ for the 4 faces
- The distance $\begin{pmatrix} r_{If}^x \\ r_{If}^y \end{pmatrix}$ to the 4 faces.

$\Rightarrow$ 17 reals for the N cells of the grid.

(5)

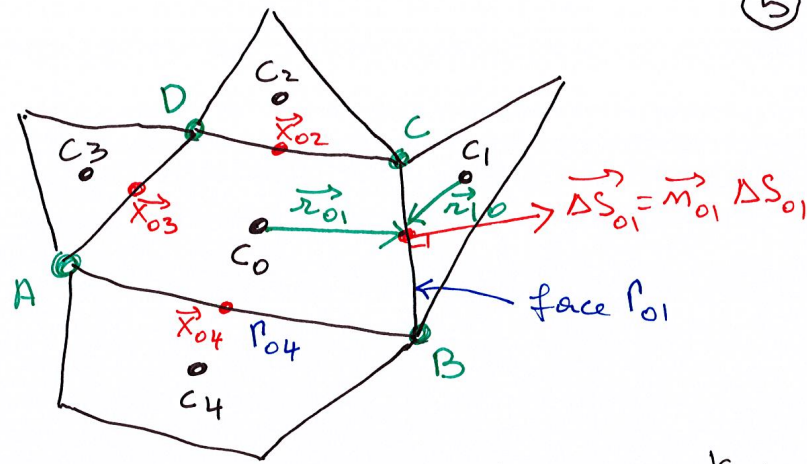


Fig. 3 Notations for calculating geometry parameters.

We consider the quad cell in Fig. 3, which has 4 neighboring cells C_1 to C_4 :

- The cell center is simply denoted C_0 .
- The volume of C_0 is V_0 .
- Cell C_0 has 4 vertices (or nodes) A, B, C and D with coordinates in the basis (x, y) : $\vec{X}_A = \begin{pmatrix} x_A \\ y_A \end{pmatrix} = x_A \vec{x} + y_A \vec{y}$.
- The face between C_0 and C_1 is denoted P_{01}
 - $\vec{DS}_{01}$ is the face-area vector (oriented from centre 0 to face P_{01})
 - $\vec{X}_{01}$ is the center of face P_{01}
 - $\vec{r}_{01}$ is the distance between center C_0 and the center of face P_{01} .

3-1) Calculation of cell volumes with the Gauss formula (6)

The volume of any type of cells in 2D and 3D is computed with the Gauss formula:

$$V_0 = \frac{1}{N_{\text{dim}}} \sum_{i=1}^4 \vec{X}_{0i} \cdot \vec{DS}_{0i} \quad (4)$$

with $N_{\text{dim}} = 2$ in 2D ($N_{\text{dim}} = 3$ in 3D).

Proof: For any position vector $\vec{X} = \begin{pmatrix} x \\ y \end{pmatrix}$ we have the identity $\nabla \cdot \vec{X} = 2$ in 2D. After integration in cell C_0 , we get $\int_{C_0} \nabla \cdot \vec{X} dV = 2 V_0$.

The flux-divergence theorem gives:

$$V_0 = \frac{1}{2} \int_{C_0} \nabla \cdot \vec{X} dV = \frac{1}{2} \int_{P_0} \vec{X} \cdot \vec{DS} = \frac{1}{2} \sum_{\text{faces}} \vec{X}_f \cdot \vec{DS}_f$$

Remark 1: The point $\vec{X}_f$ can be any point on face f ,

since:

$$\vec{X}_f \cdot \vec{DS}_f = [\vec{X}_{f'} + (\vec{X}_f - \vec{X}_{f'})] \cdot \vec{DS}_f = \vec{X}_{f'} \cdot \vec{DS}_f + \underbrace{(\vec{X}_f - \vec{X}_{f'}) \cdot \vec{DS}_f}_{\perp \vec{DS}_f}$$

$\Rightarrow$ choose $\vec{X}_f$ that gives the easiest calculations!

Remark 2: The volume of the Cartesian cell of (4)

Fig. 2 is calculated as:

$$\begin{aligned} 2V_I &= \vec{B} \cdot \vec{\Delta S}_e + \vec{C} \cdot \vec{\Delta S}_n + \vec{D} \cdot \vec{\Delta S}_w + \vec{A} \cdot \vec{\Delta S}_s \\ &= \Delta x \Delta y + \Delta x \Delta y + 0 + 0 \\ \Rightarrow V_I &= \Delta x \Delta y. \end{aligned}$$

3-2) Calculation of face-area vectors

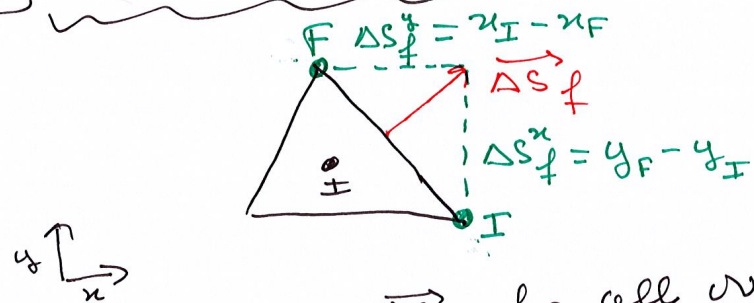


Fig. 4: Calculation of $\vec{\Delta S}_f$ for cell V_I .
The vector $\vec{\Delta S}_f$ is easily calculated from the coordinates of its initial vertex I and final vertex F

$$\vec{\Delta S}_f = \begin{pmatrix} \Delta S_f^x \\ \Delta S_f^y \end{pmatrix} = \begin{pmatrix} y_F - y_I \\ x_I - x_F \end{pmatrix} \quad (5)$$

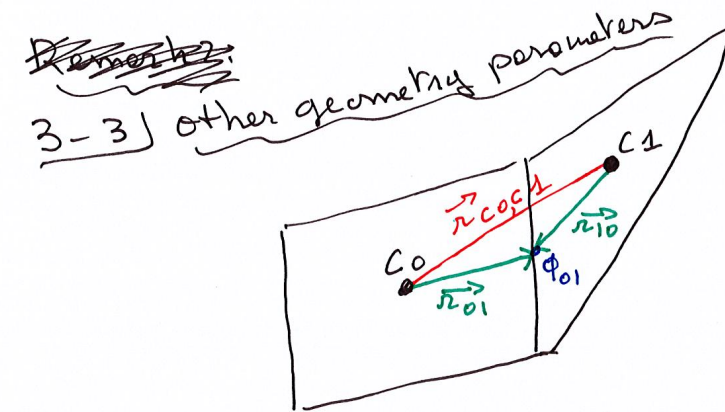
Remark 1: For the Cartesian cell of Fig. 2, Eq. (5) recovers the face-area vectors of the 4 faces, for example:

• face e: $I=B, F=C$; $\vec{\Delta S}_e = \begin{pmatrix} y_C - y_B \\ x_B - x_C \end{pmatrix} = \begin{pmatrix} \Delta y - 0 \\ \Delta x - \Delta x \end{pmatrix}$

• face n: $I=C, F=D \Rightarrow \vec{\Delta S}_n = \begin{pmatrix} \Delta y - \Delta y \\ \Delta x - 0 \end{pmatrix}$ (8)

• face w: $I=D, F=A \Rightarrow \vec{\Delta S}_w = \begin{pmatrix} 0 - \Delta y \\ 0 - 0 \end{pmatrix}$

• face s: $I=A, F=B \Rightarrow \vec{\Delta S}_s = \begin{pmatrix} 0 - 0 \\ 0 - \Delta x \end{pmatrix}$



For calculating the viscous fluxes (section 6), it will be useful to define the distance between Centroids C_0 and C_1 as:

$$\vec{r}_{C_0 C_1} = \vec{r}_{01} - \vec{r}_{10}$$

It will also be useful to interpolate the solution at face center of P_{01} . The most accurate way is to use the length-weighted average of cell values ϕ_0 and ϕ_1 :

$$\phi_{01} = d_{10} \phi_0 + (1 - d_{10}) \phi_1 \quad (6a)$$

$$\text{with } d_{10} = |\vec{r}_{10}| / |\vec{r}_{C_0 C_1}| \quad (6b)$$

A less accurate option is to use arithmetic average: (9)

$$\phi_{o1} \cong \frac{1}{2} \bar{\phi}_0 + \frac{1}{2} \bar{\phi}_1 \quad (7)$$

4) Green-Gauss calculation of gradients

An important tool for the calculation of fluxes in unstructured mesh is the calculation of the solution gradient $\nabla \phi$ at the center of the cells. This calculation is based on the Green-Gauss (GG) formula, which states that for any closed domain Ω (of boundary Γ):

$$\int_{\Omega} \nabla \phi \, dV = \int_{\Gamma} \bar{\phi} \, \vec{n} \, dS \quad (8)$$

In cell C_0 , discretization with the midpoint rule gives:

$$\nabla \bar{\phi} \Big|_{C_0} \cong \frac{1}{V_0} \sum_{\text{faces of } C_0} \bar{\phi}_f \vec{\Delta S}_f \quad (9)$$

The gradient components are thus:

$$\frac{\partial \phi}{\partial x} \Big|_{C_0} \cong \frac{1}{V_0} \sum_f \bar{\phi}_f \Delta S_f^x ; \quad \frac{\partial \phi}{\partial y} \Big|_{C_0} \cong \frac{1}{V_0} \sum_f \bar{\phi}_f \Delta S_f^y$$

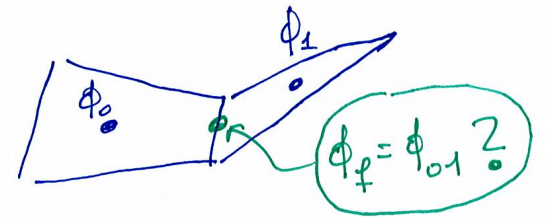
Remark: When the GG formula (9) is applied to $\bar{\phi} = 1$, one gets:

$$\sum_f \vec{\Delta S}_f = 0,$$

which states that the sum of the face area vectors sum up to zero. This property is useful for checking that in an exercise that the results obtained with Eq. (8) are correct.

The calculation of gradients with GG formula (9) will be useful for:

- Diffusive fluxes: calculation of its non-orthogonal contribution (see section 6)
- Convective fluxes: Implementation of high-order upwind discretization with the M.U.S.C.L. approach.



The values of $\bar{\phi}_f$ (or ϕ_{01} in the figure) needs now to be interpolated from values at the cell centers.

Commercial CFD codes like ANSYS FLUENT gives several different techniques to perform these

interpolation, with varying degrees of accuracy and applicability to the various type of cells encountered in actual 2D and 3D unstructured grids. Two type of interpolations are available:

- Cell-based interpolation of ϕ_f
- node-based interpolation of ϕ_f

In addition, a closely related technique for computing $\nabla \phi|_{c_0}$ is available in the latest version of ANSYS FLUENT:

- Least-square calculation of gradients.

We will review these techniques now.

4-1) Cell-based interpolation of Φ_f

The easiest interpolation of ϕ_f from values at cell centers is to use the values at cell centers C_0 and C_1 by linear interpolation (6), or arithmetic interpolation:

$$\Phi_f \approx \frac{1}{2} \phi_0 + \frac{1}{2} \phi_1 \quad (10)$$

The gradient calculation Eq. (9), (10) is named G-G cell-based interpolation. It is the first gradient calculation method developed in CFD.

The main properties are:

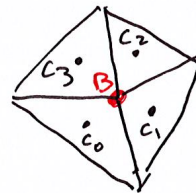
- fast calculation
- Valid for any type of 2D and 3D cells (Tri, quad, etc...)
- Low accuracy on ~~mixed~~ hybrid grids and highly distorted grids.

4-2) Node-based interpolation of Φ_f

An improvement of the cell-based interpolation is to define Φ_f from the nodal (vertex) values of cell C_0 , for example in the case of face P_{01} of Fig-3:

$$\Phi_{01} \approx \frac{1}{2} \Phi_B + \frac{1}{2} \Phi_C \quad (11)$$

The nodal values at B and C are constructed from weighted averages of cell-values surrounding the nodes, for examples:



$$\Phi_B = \frac{1}{4} (\phi_0 + \phi_1 + \phi_2 + \phi_3) \quad (12)$$

This method is called the G-G-node based calculation of gradients in ANSYS-FLUENT.

Its main properties are: (13)

- More accurate than the cell-based approach.
- More expensive to calculate.
- Recommended for tri (2D) and tet (3D) grids.
- Not available for polyhedral (pyramids, wedges) cells in 3D.

This method was for many years the best available method in ANSYS-FLUENT. In the latest version, the least-squares gradient calculation has replaced this method as the default gradient calculation available.

4-3) Least-squares (LS) gradient calculation.

This calculation is based on unidirectional Taylor expansion of the solution at cell centers (see notation of Fig. 3):

$$\Phi_1 \approx \Phi + \vec{\nabla} \Phi \cdot \vec{r}_{c0c1} + O(\|\vec{r}_{c0c1}\|^2) \quad (13)$$

If we neglect the second-order term, we get:

$$\vec{\nabla} \Phi \cdot \vec{r}_{c0c1} = \Phi_1 - \Phi_0. \quad (14)$$

Following this idea, the gradient of Φ is looked such that (14) is verified for all 4 neighboring cells of C_0 :

$$\forall i=1, \dots, 4: \vec{\nabla} \Phi \cdot \vec{r}_{c0ci} = \Phi_i - \Phi_0. \quad (15) \quad (14)$$

Eq. (15) is a linear system of 8 scalar equations for 2 scalar unknowns: $\frac{\partial \Phi}{\partial x}|_{C_0}$ and $\frac{\partial \Phi}{\partial y}|_{C_0}$.

This overdetermined system is solved by the least-squares method (see lectures on optimization).

This gradient calculation technique is called least-squares cell-based gradient evaluation in ANSYS-FLUENT. Its main properties are:

- Accuracy comparable to node-based GG (much better than cell-based GG).
- More expensive to calculate than GG methods.
- Recommended for hybrid grids.

5) Discretization of numerical fluxes

The geometry parameters of Section 3 and the gradients calculation of Section 4 are now used to calculate the convective $F_f^c = \phi \vec{u}$ and diffusive $F_f^d = \nu \vec{\nabla} \phi$ in the FV discretization given by Eq. (3):

$$\sum_{\substack{\text{faces} \\ \text{of } C_0}} \vec{F}_f \cdot \vec{\Delta S}_f = \sum_f F_f^u \Delta S_f^u + F_f^d \Delta S_f^d, \quad (16)$$

which involves, in particular, the interpolation of the fluxes at 'f' faces with respect to the unknowns at cell centers.

5-1) Calculation of diffusive fluxes $F_f^d = \nu \nabla \phi|_f$ (15)

Due to the non-orthogonality of unstructured grid, the calculation of F^d at cell faces is more involved than in Cartesian grids. On face $f = f_{01}$, the diffusive flux is calculated as:

$$F_f^d \cdot \vec{\Delta S}_f = \nu \vec{\nabla} \phi|_{o_1} \cdot \vec{\Delta S}_{o_1} = \Delta S_{o_1} \frac{\partial \phi}{\partial n}|_{o_1} \quad (17)$$

where $\frac{\partial \phi}{\partial n}|_{o_1}$ is the derivative in the direction $\vec{n}_{o_1}$.

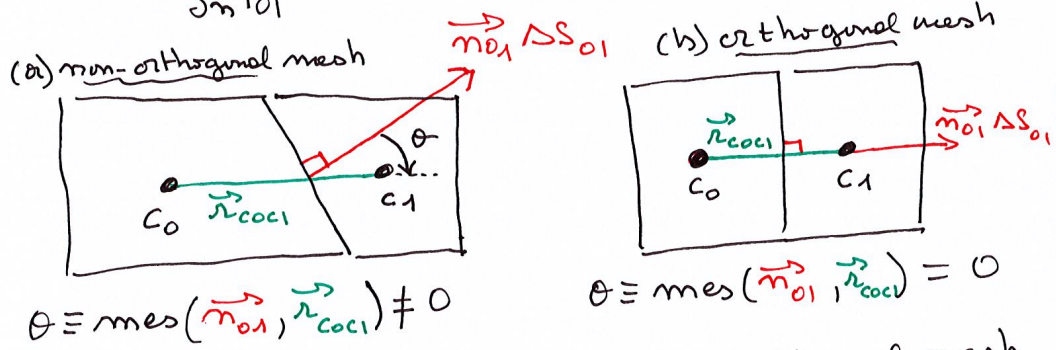


Fig. 5: Example of orthogonal and non-orthogonal mesh. For both type of mesh, $\vec{r}_{c_0c_1}$ is the line joining c_0 and c_1 , so the directional derivative in the $\vec{r}$ direction is given by the 2-point formula (2PT):

$$\frac{\partial \phi}{\partial r}|_{o_1} \approx \frac{\phi_1 - \phi_0}{\|\vec{r}_{c_0c_1}\|} \quad (18)$$

5-1-1) The 2PT approximation.

For the case of orthogonal mesh, since $\vec{n}_{o_1} \parallel \vec{r}_{c_0c_1}$ we can approximate $\frac{\partial \phi}{\partial r}|_{o_1}$ by the 2-PT formula (18)

which gives:

$$\frac{\partial \phi}{\partial n}|_{o_1} = \frac{\phi_1 - \phi_0}{\|\vec{r}_{c_0c_1}\|} \text{ in Eq. (17). } (19)$$

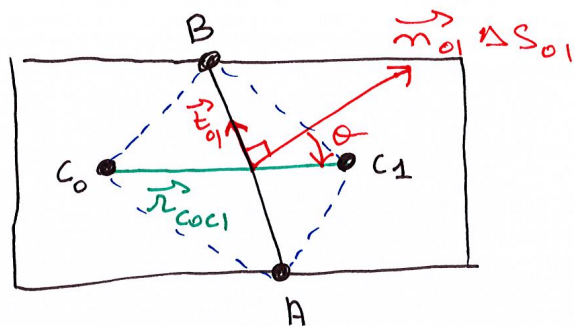
However, when the mesh is non-orthogonal (Fig. 5 left), the use of this 2-PT approximation gives large errors when the non orthogonality angle $\theta = \text{mes}(\vec{n}_{o_1}, \vec{r}_{c_0c_1})$ becomes large.

In this case, we need to add a non-orthogonal correction to the 2-PT approximation (19) by using the Diamond-scheme method (DCM)

5-1-2) The DCM approximation.

S-1-2] The DCM approximation

(17)



The name Diamond-cell method (DCM) comes from the fact that this gradient approximation was first constructed from the GG formula (8) in the diamond-shaped cell (C_0, A, C_1, B) . In these notes, the DCM formula is obtained by writing the gradient in the local basis $(\vec{n}_{o1}, \vec{t}_{o1})$ of face B_{01} :

$$\nabla \phi|_{o1} = \left(\frac{\partial \phi}{\partial n} \right)_{o1} \vec{n}_{o1} + \left(\frac{\partial \phi}{\partial t} \right)_{o1} \vec{t}_{o1} \quad (20)$$

thus:

$$\nabla \phi|_{o1} \cdot \vec{r}_{coc1} = \left(\frac{\partial \phi}{\partial n} \right)_{o1} \underbrace{\vec{n}_{o1} \cdot \vec{r}_{coc1}}_{\cos \theta} + \left(\frac{\partial \phi}{\partial t} \right)_{o1} \underbrace{\vec{t}_{o1} \cdot \vec{r}_{coc1}}_{\sin \theta}$$

The directional derivative in the $\vec{r}$ direction is discretized with 2-pt formula (14):

$$\frac{\partial \phi}{\partial n}|_{o1} = \nabla \phi|_{o1} \cdot \vec{r}_{coc1} = \frac{\phi_1 - \phi_0}{\|\vec{r}_{coc1}\|}$$

which is substituted in Eq. (21) to give

the DCM approximation of the gradient:

$$\frac{\partial \phi}{\partial n}|_{o1} = \underbrace{\frac{\phi_1 - \phi_0}{\|\vec{r}_{coc1}\| \cos \theta}}_{(1)} - \underbrace{\left(\frac{\partial \phi}{\partial t} \right)_{o1} \tan \theta}_{(2)} \quad (18)$$

• Term (1): is the orthogonal contribution of the gradient. It is equal to 2-pt formula (19) when the grid is orthogonal ($\theta = 0$).

• Term (2): is the non-orthogonal correction. This term is zero when $\theta = 0$. In ANSYS FLUENT, this ~~term~~ term is calculated as:

$$\frac{\partial \phi}{\partial t}|_{o1} = \nabla \phi|_{o1} \cdot \vec{t}_{o1}$$

where $\nabla \phi|_{o1}$ is the face center gradient calculated with techniques discussed in Section 4.

5-2] Calculation of convective fluxes $F_f^c = (\vec{u} \cdot \vec{\Delta S})_f \phi_f$ (19)

As in the case of Cartesian grids, convective discretization involves centered (CDS) and upwind-type (UDS) methods. The implementation in unstructured grid will be discussed for the case of Fig. 5

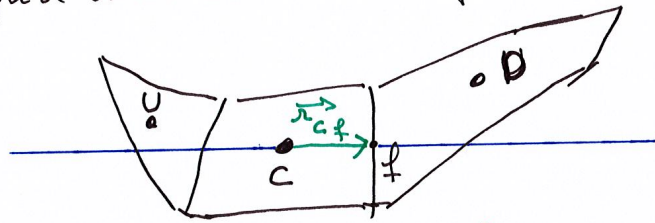


Fig. 5 Notation for convective discretization.

In the case the flow direction at face f of cell U_C is to the right, we have the following notation:

- C : is the central point
- D : Downstream point
- U : Upstream point.

As shown in the figure, these points are not aligned. Thus the implementation of high-order methods such as the k_2 -scheme (see exercise) will be more complicated than on regular grids.

The convective flux on face f is:

$$F_f^c \cdot \vec{\Delta S}_f = (\vec{u}_f \cdot \vec{\Delta S}_f) \phi_f, \quad (19)$$

and we will discuss the various interpolations of ϕ_f .

5-2-1] Upwind scheme (UDS) (20)

The first-order upwind interpolation is as easy as in Cartesian grid:

$$\phi_f = \begin{cases} \phi_c & \text{if } (\vec{u} \cdot \vec{\Delta S})_f > 0 \\ \phi_D & \text{otherwise.} \end{cases} \quad (20)$$

As mentioned in the previous lecture, the accuracy of UDS is very low, and more accurate methods are recommended.

5-2-2] Central scheme (CDS)

It is tempting to simply write the CDS as the arithmetic average:

$$\phi_f = \frac{1}{2} \phi_c + \frac{1}{2} \phi_D. \quad (21)$$

However, a more accurate implementation is due by considering the k_2 -scheme described in the next section.

5-2-3) Higher order interpolation for correction: the k -scheme ⁽²¹⁾

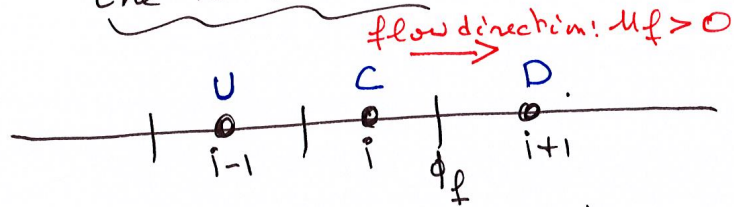


Fig. 6 Notation used for the k -scheme in a 1D grid

In exercise A.2, we have seen that the k -scheme use the following interpolation on a Cartesian grid:

$$\phi_f = \frac{\phi_{i+1} + \phi_i}{2} + \frac{(k-1)}{4} [\phi_{i+1} - 2\phi_i + \phi_{i-1}] \quad (22)$$

where the values of k gives the following methods:

- $k=1$: CDS scheme
- $k=-1$: upwind second-order
- $k=\frac{1}{2}$: Q.U.I.C.K scheme.

All these schemes are second-order accurate, and are available in ANSYS FLUENT.

For applying the k -scheme (22) to the unstructured mesh of Fig. 5, we need to rewrite as:

$$\phi_f = \underbrace{\phi_i}_{\phi_c} + \underbrace{\frac{(1-k)}{2} [\frac{\phi_{i+1} - \phi_{i-1}}{2}]}_{S_n^0 \phi_c \Delta x} + \underbrace{\frac{k}{2} [\phi_{i+1} - \phi_i]}_{S_n^0 \phi_f \Delta x} \quad (23)$$

As in Section 4.3, the k -scheme is extended to unstructured grids by remarking: (22)

$$S_n^0 \phi_c \Delta x \cong \vec{\nabla} \phi_c \cdot \vec{r}_{cf}, \quad (24a)$$

$$S_n^0 \phi_f \Delta x \cong \vec{\nabla} \phi_f \cdot \vec{r}_{cf}, \quad (24b)$$

which finally gives the generalization of the k -scheme:

$$\phi_f = \phi_c + (1-\frac{k}{2}) \vec{\nabla} \phi_c \cdot \vec{r}_{cf} + \frac{k}{2} \vec{\nabla} \phi_f \cdot \vec{r}_{cf}. \quad (24)$$

The gradients are calculated with the techniques described in Section 4.

14/03/2014

Analysis of the accuracy of FVM on arbitrary grids. Assessing the quality of the grid

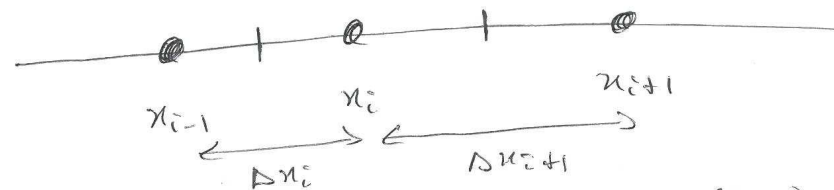
①

For meshing "real life" CFD problems, the grid is generally non-uniform (for clustering cells in high-gradient flow regions, like boundary layers), and meshing complex geometries give rise to mixed grids (tri and tet) which may be highly distorted, so that the accuracy and convergence of the computation may be severely diminished.

This lecture will review the accuracy properties of FVM on arbitrary mesh by checking the local truncation error (LTE) of the schemes. Then, we will derive some general guidelines for building high-quality grids for CFD in complex geometries.

② 1) Effects of grid non-uniformity: 1D example.

We consider a 1D mesh of cell $[x_{i-1/2}, x_{i+1/2}]$ of centroid x_i , such as the grid $\rightarrow$ step $\Delta x_i = x_i - x_{i-1}$ is non-uniform, i.e. $\Delta x_i \neq \Delta x_{i+1}$.



We will build a FD approximation of $(\frac{\partial u}{\partial x})_i$ by the usual method of Taylor expansions:

$$u_{i-1} = u_i - \Delta x_i \left(\frac{\partial u}{\partial x} \right)_i + \Delta x_i^2 \left(\frac{\partial^2 u}{\partial x^2} \right)_i - \frac{\Delta x_i^3}{3!} \left(\frac{\partial^3 u}{\partial x^3} \right)_i \quad (1)$$

$$u_{i+1} = u_i + \Delta x_{i+1} \left(\frac{\partial u}{\partial x} \right)_i + \Delta x_{i+1}^2 \left(\frac{\partial^2 u}{\partial x^2} \right)_i + \frac{\Delta x_{i+1}^3}{3!} \left(\frac{\partial^3 u}{\partial x^3} \right)_i \quad (2)$$

By combining Eqs (1) and (2), we get the FD quotient:

$$\frac{u_{i+1} - u_{i-1}}{\Delta x_i + \Delta x_{i+1}} = \left(\frac{\partial u}{\partial x} \right)_i + \tau_i \quad (3)$$

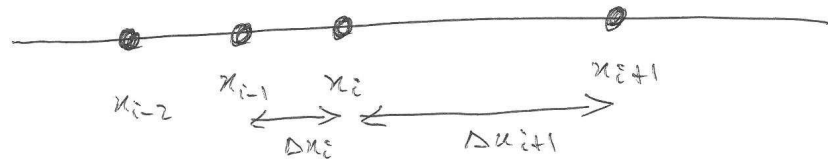
with the LTE:

$$\tau_i = (\Delta x_{i+1} - \Delta x_i) \left(\frac{\partial^2 u}{\partial x^2} \right)_i + \frac{1}{6} \frac{\Delta x_i^3 + \Delta x_{i+1}^3}{\Delta x_i + \Delta x_{i+1}} \left(\frac{\partial^3 u}{\partial x^3} \right)_i \quad (4)$$

On a uniform grid, $\Delta x_i = \Delta x \forall i$,
 we recover the usual CDS
 formula $\delta_x^0 u_i = \frac{u_{i+1} - u_{i-1}}{2 \Delta x}$ with LTE
 $\tau_i = O(\Delta x^2)$, but when $\Delta x_i \neq \Delta x_{i+1}$, the
 LTE is only first order!

Rule 1: In general, FD formulas lose 1 (or more!)
 order of accuracy on arbitrary grids.

This is especially true when there is an abrupt
 change in the grid, e.g. $\Delta x_{i+1} \gg \Delta x_i$



Local loss of accuracy near x_i is inevitable!

Rule 2: For "almost" uniform grids which are
 called smooth grids, such as:

$$\forall i, |\Delta x_{i+1} - \Delta x_i| = O(\Delta x_{\max}^2), \text{ where } \Delta x_{\max} = \max_i \Delta x_i$$

then Eq. (4) shows that $\tau_i = O(\Delta x_{\max}^2)$ and FD
 formula (3) recovers the second-order accuracy observed
 on uniform mesh.

Best practice 1: Avoid brutal change in
 the grid distribution or 2nd order accuracy is
 lost! (4)

Rule 3: Popular meshing strategies in grid
 generation software use a graded mesh such as:

$$\Delta x_{i+1} = r \Delta x_i, \text{ with clustering factor } r \geq 1.$$

For such grids, the LTE (4) can be written as:

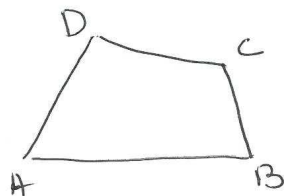
$$\tau_i = (r-1) \Delta x_i \left(\frac{\partial^2 u}{\partial x^2} \right)_i + O(\Delta x_i^2) \quad (5)$$

and if r is chosen close to 1 then LTE is
 near 2nd order and the local errors remain
 acceptable.

Best practice 2: Use smoothly varying grids
 (recommended values: $\frac{1}{4} \leq r \leq 1, 1-\frac{1}{4}$) in regions of
 strong flow variations to minimize numerical errors.

2) Error analysis for a 2D arbitrary quad (5)

We consider a quadrilateral cell $\mathcal{Q}_I$ (of volume V_I)



that is not highly distorted, ie such as faces lengths are:

$$x_D - x_A, x_C - x_D \approx \Delta x$$

$$V_I \approx \mathcal{O}(\Delta x \Delta y)$$

$$y_D - y_A, y_C - y_B \approx \Delta y$$

Note that a triangle is a degenerate case (eg. if $A=B$).

For a cell-vertex discretization of the steady compressible Euler eqs, ie:

$$\sum_{\text{faces}} \vec{F} \cdot \vec{\Delta S}_f = V_I q_I, \quad (6)$$

When we assume that $\Delta x \approx \Delta y = h$, Roe (1987) has shown that the LTE τ_I reads:

$$V_I \tau_I = \cancel{E_I} + \mathcal{O}(h^4) \quad (7)$$

with:

$$E_I = -E_2 \left(\frac{\partial^2 F_x^c}{\partial x^2} \right) + 2E_3 \left(\frac{\partial^2 F_x^c}{\partial x \partial y} \right) - E_4 \left(\frac{\partial^2 F_x^c}{\partial y^2} \right) - E_1 \left(\frac{\partial^2 F_y^c}{\partial x^2} \right) + 2E_2 \left(\frac{\partial^2 F_y^c}{\partial x \partial y} \right) - E_3 \left(\frac{\partial^2 F_y^c}{\partial y^2} \right) \quad (8)$$

$$E_1 = (x_A - x_C)(x_B - x_D)(x_A - x_B + x_C - x_D) \quad (9a)$$

$$= \mathcal{O}(\Delta x^3)$$

$$E_2 = 2V_I(x_A - x_B + x_C - x_D) + (x_A - x_C)(x_B - x_D) \times (y_A - y_B + y_C - y_D) \quad (9b)$$

$$= \mathcal{O}(\Delta x^2 \Delta y^2)$$

$$E_3 = 2V_I(y_A - y_B + y_C - y_D) + (y_A - y_C)(y_B - y_D)(x_A - x_B + x_D - x_C) \quad (9c)$$

$$= \mathcal{O}(\Delta x \Delta y^2)$$

$$E_4 = (y_A - y_C)(y_B - y_D)(y_A - y_B + y_C - y_D) \quad (9d)$$

$$= \mathcal{O}(\Delta y^3)$$

So that for a smooth quad such as $\Delta x \approx \Delta y \approx h$, the LTE

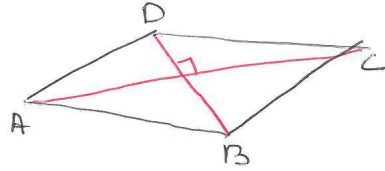
(7) reads:

$$\tau_I = \frac{\mathcal{O}(h^3)}{h^2} + \mathcal{O}(h^2). \quad (10)$$

Remark 1: For an arbitrary mesh $\tau_I = \mathcal{O}(h)$ the FVM is only 1st order accurate!

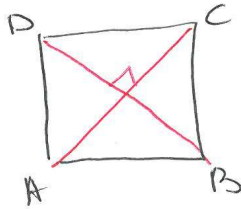
Remark 2: For particular cases of quadrilateral cells, because E_1 to E_4 in (9) cancel out such that $E_5 = 0$ and the method is second order accurate.

This is the case for parallelograms:



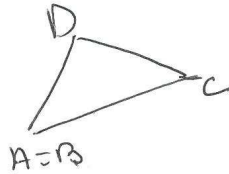
which are such that $\frac{\vec{x}_A + \vec{x}_C}{2} = \frac{\vec{x}_B + \vec{x}_D}{2}$.

The only interesting parallelogram cell is the rectangular cell:



⇒ On smooth Cartesian grids, the FVM is 2nd order accurate!

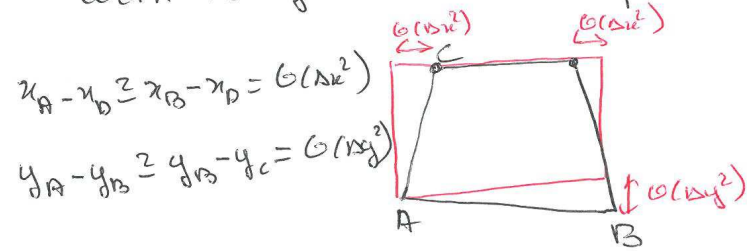
Remark 3: A triangle is a degenerate case of quadrilateral:



For triangles, cells, we always have $E_{\pm} \neq 0$
 ⇒ FVM on triangular grids is 1st-order accurate!

(4)

Remark 4: Let us consider a quad-cell with slight deviations from a ~~rect~~ rectangle:



$$x_A - x_D \approx x_B - x_C = O(hx^2)$$

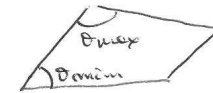
$$y_A - y_B \approx y_D - y_C = O(hy^2)$$

then $E_{\pm} = O(h^2)$ and the method would be second-order accurate.

This "slight deviation from rectangle" property is measured by the skewness factor Q_s .

For a quad, it is defined as:

$$Q_s = \max \left[\frac{\theta_{\max} - \pi/2}{\pi/2}, \frac{\pi/2 - \theta_{\min}}{\pi/2} \right] \in [0, 1].$$



$$Q_s \in [0, 1]$$



$$Q_s = 0.$$

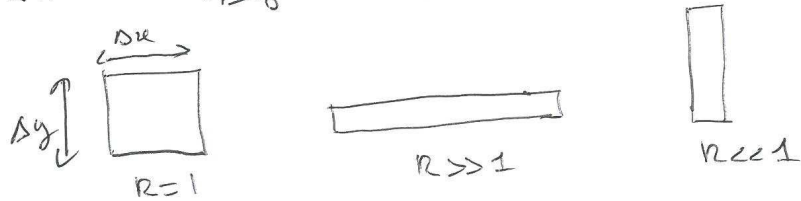
Quads are close to rectangles when $Q_s \approx 0$, hence numerical errors will be minimal for low skewness elements.

(8)

Remark 5) For arbitrary quad cells, the LTE (9)
 can be written as:

$$\begin{aligned} \gamma_{\pm} &= \underbrace{\frac{\mathcal{O}(\Delta x^3)}{\Delta x \Delta y}}_{E_1} + \underbrace{\frac{\mathcal{O}(\Delta x \Delta y^2)}{\Delta x \Delta y}}_{E_3} + \underbrace{\frac{\mathcal{O}(\Delta x^2 \Delta y)}{\Delta x \Delta y}}_{E_2} + \underbrace{\frac{\mathcal{O}(\Delta y^3)}{\Delta x \Delta y}}_{E_4} \\ &= \mathcal{O}(R \Delta x) + \mathcal{O}(\Delta y) + \mathcal{O}(\Delta x) + \mathcal{O}\left(\frac{\Delta y}{R}\right) \end{aligned}$$

with $R = \Delta x / \Delta y$ the cell aspect ratio:



The LTE is minimal for cell with aspect ratio close to 1, but can become $\mathcal{O}(1)$ for cells with high aspect ratio!

3) Summary and best practice guidelines.

See slides.

14/04/2015

Convergence accelerators and relaxation methods

(1)

For solving linear system $Ax = F$ we consider the "splitting" method (A.19 from lecture notes):

$$P X^{k+1} = P X^k + R^k \quad (1)$$

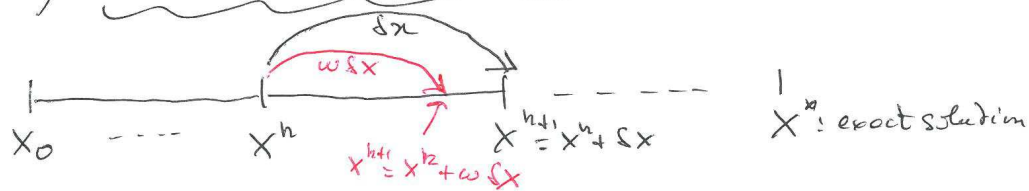
Algorithm A.5 from lecture notes)

1) X^0 given, calculate $R^0 = F - AX^0$

2) For $k \geq 0$,

- compute solve $P \delta x = R^k$
- compute $X^{k+1} = X^k + \delta x$
- compute R^{k+1}
- test if $|R^{k+1}| < \epsilon$

1) Relaxation version of the "splitting" method



For certain linear systems, it has been observed that using steps of length δx is causing convergence problems, and that it was worthwhile to reduce the step as:

$$w \delta x = w (X^* - X^k) \quad (2)$$

where the real $w \in [0, 1]$ is called the under-relaxation factor. The iteration is now:

$$X^{k+1} = X^k + w \frac{\delta x}{X^* - X^k} \quad (3a)$$

$$= w X^* + (1-w) X^k \quad (3b)$$

The under-relaxed "splitting" method (1) reads now:

$$P X^* = P X^k + R^k \quad (4a)$$

$$X^{k+1} = w X^* + (1-w) X^k \quad (4b)$$

it is also known as the preconditioned Richardson method. Its algorithm reads (compare with Algorithm A.5):

Algorithm (A.6)

1) X^0 given, calculate $R^0 = F - AX^0$

2) For $k \geq 0$,

- solve $P \delta x = R^k$
- compute $X^{k+1} = X^k + w \delta x$
- compute R^{k+1}
- test if $|R^{k+1}| < \epsilon$

2) Description of some classical relaxation methods

2-1) Static relaxation methods (ω is fixed) (3)

Original method	Relaxation method
Jacobi $P = D$	S.O.R $P = \frac{D}{\omega}$
G-S $P = D - L$	S.O.R: $P = \frac{D}{\omega} - L$
	SSOR: $P = \frac{1}{2-\omega} \left(\frac{1}{\omega} D - L \right) \left(\frac{1}{\omega} D \right)^{-1} \left(\frac{1}{\omega} D + L \right)^T$

the interest of S.S.O.R over S.O.R is that it gives a symmetric preconditioner (in the case A is symmetric).

2-2) Practical remarks on the value of ω

- If $\omega = 1$, we recover the original method (1)
- If $\omega > 1$, it is called an over-relaxed method, which may have stability problems (not recommended).
- If $\omega \in [0, 1]$, it is an under-relaxed method (we advance slower towards exact solution x^*), but it helps the stability in case the original method ($\omega = 1$) is not convergent.
- For many applications (e.g. FLUENT software), the value of ω is ~~is~~ given by the user (experience and best practice guidelines are needed)

• If ω is reduced from 1 to 0, the convergence is slowed (it is halted for $\omega = 0$), but the stability is enhanced, (4)

• For some linear systems, an optimal value of ω_{opt} can be calculated for faster convergence (see optimal SOR method of L3 lecture notes).

2-3) Dynamic relaxation methods

Modern iterative methods are able to compute dynamically (at each iteration) the "optimal" value of ω for faster convergence.

$\Rightarrow$ Krylov methods such as conjugate gradient (~~CG~~) method, GMRES, BiCG-STAB (~~CG~~) method (for ~~is~~ symmetric systems), GMRES, BiCG-stab (for non-symmetric systems) use dynamical relaxation factors.

A Travaux Dirigés

A.1 Finite-volume method for convection-diffusion equation

Exercice A.1 (The *QUICK* and κ -scheme for convection discretization).

This exercise concerns the construction of higher-order upwind-biased finite-volume scheme for the convective fluxes. We will use the standard notations of the lecture notes for the 2D convection diffusion equation: the convective flux $F_x^c = u\phi$ is discretized on the mesh depicted in Fig 15.2 using mid-point rule as $(F_x^c)_e = u_e \Delta y_P \phi_e$ at face Γ_e . The CDS and upwind interpolation of ϕ_e have been described in the lectures notes, and in the following we will build high-order interpolation for ϕ_e . Throughout this exercise, we will suppose that $u_e > 0$ and that the mesh is uniform of size Δx .

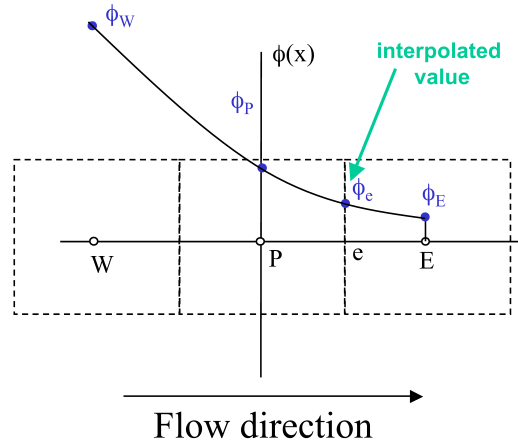


Fig. A.1. Parabolic interpolation used by the *QUICK* scheme.

- 1) While the CDS scheme uses linear polynomial interpolation between ϕ_E and ϕ_P to build a second-order formula, the *QUICK* scheme¹ uses quadratic interpolation between ϕ_E , ϕ_P and ϕ_W to achieve better accuracy. Therefore, as the UDS scheme, the *QUICK* scheme uses upwind-biased interpolation following the flow direction (sign of convective velocity u_e).
 - a) Build² the quadratic polynomial $\pi_2[\phi](x)$ that fits ϕ at values ϕ_W , ϕ_P and ϕ_E (see Fig. A.1).
 - b) Evaluate this interpolation polynomial at face $x_e = x_P + \Delta x/2$ to obtain the *QUICK* interpolation formula:

¹ Acronym for “Quadratic Upwind Interpolation for Convection Kinematics”, introduced by Leonard (1979).

² Hint: use the Lagrange interpolation polynomial $\pi_2[\phi](x) = \sum_{j=0}^2 \phi(x_j) L_j(x)$ at points $x_0 \equiv x_W = x_P - \Delta x$, $x_1 \equiv x_P$, $x_2 \equiv x_E = x_P + \Delta x$, where the j th Lagrange polynomial is :

$$\phi_e = \frac{3}{8}\phi_{i+1} + \frac{3}{4}\phi_i - \frac{1}{8}\phi_{i-1}, \quad (\text{A.1})$$

where $\phi_{i+1} = \phi_E$, $\phi_i = \phi_P$, and $\phi_{i-1} = \phi_W$.

c) Write the *QUICK* interpolation scheme as:

$$\phi_e = \frac{\phi_{i+1} + \phi_i}{2} - \frac{\Delta x^2}{8} \delta_{xx}^0 \phi_i, \quad (\text{A.2})$$

and give an interpretation of this scheme.

d) Determine the truncation error of interpolation (A.2) at face x_e . Can you guess the order of accuracy of the *QUICK* scheme for convection?

e) What nodes would the *QUICK* scheme use in the case $u_e < 0$?

2) A generalization of the *QUICK* scheme (A.2) is given by the family of upwind-biased κ -scheme that reads on uniform mesh :

$$\phi_e = \begin{cases} \frac{\phi_{i+1} + \phi_i}{2} + \frac{\kappa - 1}{4} \Delta x^2 \delta_{xx}^0 \phi_i & \text{if } u_e \geq 0, \\ \frac{\phi_{i+1} + \phi_i}{2} + \frac{\kappa - 1}{4} \Delta x^2 \delta_{xx}^0 \phi_{i+1} & \text{if } u_e < 0, \end{cases} \quad (\text{A.3})$$

where κ is a real parameter, such as special values of this parameter correspond to well-known convection discretization:

- $\kappa = 1$ yields the CDS scheme,
- $\kappa = -1$ yields the 2nd order upwind scheme,
- $\kappa = 1/3$ yields the 3rd order upwind scheme,
- $\kappa = 1/2$ yields the *QUICK* scheme.

a) Assume that $u_e = u_w = u > 0$ and write down the κ -scheme as a finite-difference quotient:

$$uD_x^{(\kappa)} \phi_i = \frac{u\phi_e - u\phi_w}{\Delta x}, \quad (\text{A.4})$$

and determine the coefficients of the scheme with respect to κ . Show that the special cases $\kappa = 1$ and $\kappa = -1$ yield the CDS and 2nd order upwind scheme respectively:

$$\delta_x^0 \phi_i = \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x}, \quad D_x^- \phi_i = \frac{3\phi_i - 4\phi_{i-1} - \phi_{i-2}}{2\Delta x}. \quad (\text{A.5})$$

3) Calculate the truncation error of the κ -scheme and show that it is 3rd order accurate for $\kappa = 1/3$ and 2nd order for all other values of κ .

Exercice A.2 (On conservative and non-conservative discretization methods).

The transport equation for a conserved scalar ϕ in a steady incompressible 2D flow (such as $\nabla \cdot \mathbf{v} = 0$) reads in conservative form :

$$\nabla \cdot \mathbf{v} \phi = 0. \quad (\text{A.1})$$

This equation is to be solved in a rectangular domain Ω such as the boundary conditions are: inflow condition at west boundary, outflow at east boundary, wall ($\phi = 0$) at north and south boundaries.

$$L_j(x) = \prod_{\substack{i=0 \\ j \neq i}}^2 \frac{x - x_i}{x_j - x_i}.$$

- 1) Show that Eq. (A.1) is analytically equivalent to the convective (or nonconservative) form of the transport equation :

$$\mathbf{v} \cdot \nabla \phi = 0. \quad (\text{A.2})$$

- 2) Integrate Eq. (A.1) in the whole fluid domain and express the compatibility condition verified by ϕ at inflow and outflow boundaries.
- 3) Discretize Eq. (A.1) with FVM on a Cartesian grid with uniform cells of size Δx and Δy . Interpolation at cell centers is not needed: the values at cell faces will be denoted as $\phi_{i+1/2,j}$, $u_{i+1/2,j}$ on east face, etc
- 4) Integrate the FV scheme of question 3) in the whole fluid domain Ω and compare the result with the analytical compatibility condition of question 2).
- 5) Discretize Eq. (A.2) with FDM using same notations as question 3). Compare the FD and FV schemes and show that they are equivalent for constant convection velocity only.
- 6) Same question as 4) for the FD scheme. Show that the compatibility condition is asymptotically verified when the mesh is refined.

Exercice A.3. On considère l'équation d'advection-diffusion stationnaire

$$\frac{du\phi}{dx} - \frac{d}{dx} \left(\nu \frac{d\phi}{dx} \right) = q(x), \quad (\text{A.3})$$

dans le domaine $\Omega =]0, 1[$. La viscosité $\nu = \nu(x)$ est strictement positive pour tout $x \in \Omega$. La vitesse d'advection $u = u(x)$ est de signe quelconque. On suppose néanmoins que $u(x=0) > 0$ et $u(x=1) > 0$: c'est-à-dire qu'au bord $x=0$ le fluide entre, et en $x=1$ le fluide sort. Pour que le problème soit mathématiquement bien posé, on choisit donc les conditions aux limites suivantes : on impose une condition d'entrée (inflow) en $x=0$,

$$\phi(0) = g_-, \quad (\text{A.4})$$

et une condition de sortie (outflow) en $x=1$, qui est généralement de type Neumann :

$$\frac{d\phi}{dx}(1) = g_+. \quad (\text{A.5})$$

On se propose de résoudre ce problème par la méthode des volumes finis sur un maillage non-uniforme centré en cellules.

- 1) *Construction du maillage.* Soit $\{x_{i+\frac{1}{2}}, i = 0, \dots, N\}$ une suite de réels tels que :

$$0 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \dots < x_{N+\frac{1}{2}} = 1.$$

Construisez un maillage centré en cellules tels que les $x_{i+\frac{1}{2}}$ représentent les frontières des cellules élémentaires Ω_i . On notera x_i le centre de Ω_i , et Δx_i son volume. L'espacement entre les nœuds x_i et x_{i+1} sera noté $\widetilde{\Delta x}_i$. Combien y-a-t'il de cellules ? Représentez ce maillage sur un schéma. Soignez ce schéma en particulier autour d'une cellule Ω_i quelconque, et près des bords $x=0$ et $x=1$.

- 2) *Définition des inconnues.* On définit les valeurs moyennes dans une cellule :

$$\phi_i = \frac{1}{\Delta x_i} \int_{\Omega_i} \phi(x) dx, \quad q_i = \frac{1}{\Delta x_i} \int_{\Omega_i} q(x) dx, \quad i = 1, \dots, N.$$

Effectuez une approximation des intégrales permettant de relier ces moyennes aux valeurs nodales.

- 3) *Formulation volumes finis*. En intégrant l'équation (A.3) dans le volume Ω_i , montrez que la discretisation volumes finis s'écrit :

$$F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}} = \Delta x_i q_i, \quad i = 1, \dots, N, \quad (\text{A.6})$$

avec $F = F^c - F^d$, les fonctions de flux conservatif et dissipatif dont vous donnerez l'expression. Le système discret est-il clos ? Quelle différence notable observez-vous par rapport au cas bidimensionnel ?

- 4) *Interpolation des flux*. Il faut maintenant exprimer les flux discrets $F_{i+\frac{1}{2}}$, $F_{i-\frac{1}{2}}$ en fonction des valeurs nodales ϕ_i .

- Dans le cas d'une vitesse d'advection quelconque, établissez les interpolations UDS et CDS du flux convectif $F_{i+\frac{1}{2}}^c$ pour $i = 1, \dots, N-1$, c'est-à-dire loin des bords. En déduire l'expression de $F_{i-\frac{1}{2}}^c$. Retrouvez la formule de différences finies upwind dans le cas $u(x) = u > 0$.
- Effectuez l'interpolation du flux diffusif $F_{i+\frac{1}{2}}^d$ pour $i = 1, \dots, N-1$. Sur maillage uniforme et $\nu(x) = \nu > 0$, retrouvez la formule de différences finies :

$$\frac{F_{i+\frac{1}{2}}^d - F_{i-\frac{1}{2}}^d}{\Delta x} = \nu \delta_{xx}^0 \phi_i.$$

- 5) *Implémentation des conditions aux limites*. Il faut maintenant tenir compte des conditions aux limites dans l'interpolation des flux discrets près des parois.

- Représentez les C.L. (A.4) et (A.5) sur le dessin du maillage. D'après Eq. (A.6), quels sont les flux discrets dont l'interpolation nécessite d'être modifiée *a priori* pour tenir compte des conditions aux limites ?
 - Implémentez les C. L. dans les flux convectifs pour l'approximation CDS.
 - Même question pour l'approximation UDS. D'après l'énoncé, $u_{\frac{1}{2}} > 0$: la condition de sortie (A.5) est-elle nécessaire ?
 - Implémentez les C. L. dans les flux diffusifs.
 - On se place maintenant dans le cadre d'un problème purement diffusif ($u = 0$ dans (A.3)), à viscosité constante. Exprimez le schéma (A.6) pour $i = 1$ et $i = N$ dans le cas d'un maillage uniforme. Comparez avec les différences finies. Peut-on interpréter cette implémentation des C. L. comme "le retour du point fantôme" ?
- 6) *Représentation matricielle du schéma*. Montrez que la formulation volumes finis s'écrit sous la forme matricielle, de taille $N \times N$:

$$(\mathcal{C}_h(u) - \mathcal{D}_h) \Phi = Q'.$$

Dans le cas CDS, donnez l'expression des matrices tridiagonales $\mathcal{C}_h(u)$ et $\mathcal{D}_h$, ainsi que le second-membre Q' .

A.2 Finite-volume method for Navier-Stokes equations

Exercice A.4. (Approximation "Symmetry-Preserving" du flux convectif dans la direction x sur maillages décalés)

On considère le maillage centré en cellules :

$$\Omega_{ij} =]x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}[\times]y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}[,$$

pour $i = 1, \dots, N_x$, et $j = 1, \dots, N_y$. On rappelle que Ω_{ij} a pour volume $\Delta x_i \Delta y_j$, et pour centre $\mathbf{x}_{ij} = \begin{pmatrix} x_i \\ y_j \end{pmatrix}$, soit :

$$x_i = \frac{1}{2}(x_{i+\frac{1}{2}} + x_{i-\frac{1}{2}}), \quad y_j = \frac{1}{2}(y_{j+\frac{1}{2}} + y_{j-\frac{1}{2}}).$$

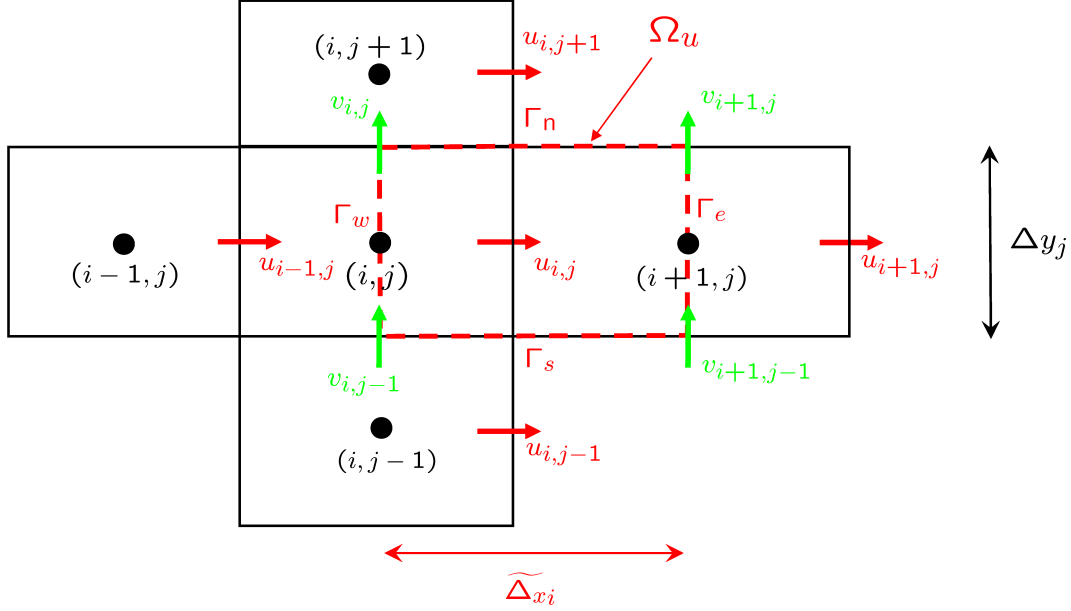


Fig. A.2. Arrangement décalé des inconnues pour le volume de contrôle Ω_u , de taille $V_u = \widetilde{\Delta x}_i \Delta y_j$. Les notations géographiques sont utilisées pour Ω_u et ses faces. Les notations indicielles sont réservées à la cellule Ω_{ij} et aux inconnues du problème.

De plus, on définit :

$$\widetilde{\Delta x}_i = x_{i+1} - x_i, \quad i = 1, \dots, N_x - 1, \quad (\text{A.7a})$$

$$\widetilde{\Delta y}_j = y_{j+1} - y_j, \quad j = 1, \dots, N_y - 1. \quad (\text{A.7b})$$

Soit $\mathbf{F}_x^c = \rho \mathbf{v} u$ la fonction de flux convectif qui apparaît dans l'équation quantité de mouvement dans la direction x . Ce problème a pour objet de calculer l'approximation discrète :

$$\int_{\Gamma_u} \mathbf{F}_x^c \cdot \mathbf{n} \, dS \cong F_{x,h}^c, \quad (\text{A.8})$$

pour le volume de contrôle Ω_u centré autour de u_{ij} , pour $i = 1, \dots, N_x - 1$ et $j = 1, \dots, N_y$, cf. Fig. A.2.

En préambule, notons le lien avec l'équation de continuité. Si $u = \mathbf{C}^{\text{te}}$ dans $\mathbf{F}_x^c$, on remarque que ce flux est proportionnel au *débit massique* $\int_{\Gamma_u} \rho \mathbf{v} \cdot \mathbf{n} \, dS$ au travers des faces du volume de contrôle Ω_u . Ce débit doit être nul pour un écoulement incompressible à masse volumique constante. Cette propriété de conservation doit être retrouvée dans l'approximation discrète $F_{x,h}^c$ pour la réalité physique de la simulation. En corollaire, cela assure de bonnes propriétés aux systèmes discrets à résoudre, tels leur (anti-)symétrie, même si les pas de maillages $\widetilde{\Delta x}_i$ et Δy_j ne sont pas uniformes. Ces propriétés permettront l'utilisation d'algorithmes de résolution performants tels que le gradient conjugué.

Dans la suite du problème, on considère un fluide de masse volumique constante telle que $\rho = 1$. De plus, il sera commode de noter avec des barres le débit massique au travers des faces des divers volumes de contrôle. Par exemple, sur la face 'est' de Ω_{ij} , on définit :

$$\int_{(\Gamma_{ij})_e} \mathbf{v} \cdot \mathbf{n} \, dS = \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x_{i+\frac{1}{2}}, y) \, dy \equiv \bar{u}_{ij}.$$

Notez qu'aucune discrétisation n'est encore faite. On l'effectue, comme d'habitude, avec l'approximation de la moyenne, *i.e.* :

$$\bar{u}_{i,j} \cong \Delta y_j u_{ij}.$$

En procédant de façon identique sur les trois autres faces, on retrouve la discrétisation de l'équation de continuité dans Ω_{ij} :

$$\bar{u}_{i,j} - \bar{u}_{i-1,j} + \bar{v}_{i,j} - \bar{v}_{i,j-1} \cong \Delta y_j (u_{i,j} - u_{i-1,j}) + \Delta x_i (v_{i,j} - v_{i,j-1}) = 0. \quad (\text{A.9})$$

Cette relation assure la conservation de la masse au travers de la cellule Ω_{ij} . D'une façon analogue, on utilisera des barres pour noter le débit massique au travers des faces de Ω_u , par exemple au travers de sa face 'est' :

$$\int_{\Gamma_e} \mathbf{v} \cdot \mathbf{n} \, dS = \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x_{i+1}, y) \, dy \equiv \bar{u}_e.$$

Partie I. Dans cette partie, on effectue la discrétisation pour le volume de contrôle générique Ω_u situé loin des parois du domaine, *cf.* Fig. A.2. L'implémentation des conditions aux limites sera effectuée dans la seconde partie.

- 1) Dans une première étape, on effectue une discrétisation partielle de $\mathbf{F}_x^c$, qui ne porte que sur la vitesse convectée u .
 - a) Montrez que, en effectuant une approximation de la vitesse u uniquement, on obtient :

$$\int_{\Gamma_u} \mathbf{F}_x^c \cdot \mathbf{n} \, dS \cong \bar{F}_x^c \equiv \bar{u}_e u_e - \bar{u}_w u_w + \bar{v}_n u_n - \bar{v}_s u_s. \quad (\text{A.10})$$

- b) Complétez la discrétisation avec une interpolation centrée de la vitesse u en fonction des inconnues u_{ij} . Par exemple, d'après la figure A.2 :

$$u_e \cong \frac{1}{2} u_{i+1,j} + \frac{1}{2} u_{i,j}.$$

Montrez alors que, pour le volume de contrôle $\Omega_u(i, j)$, le flux (semi)-discret s'écrit :

$$\bar{F}_x^c(i, j) = \bar{A}_E(i, j) u_{i+1,j} + \bar{A}_W(i, j) u_{i-1,j} + \bar{A}_P(i, j) u_{i,j} + \bar{A}_N(i, j) u_{i,j+1} + \bar{A}_S(i, j) u_{i,j-1},$$

et exprimez les coefficients $\bar{A}_E(i, j)$, $\bar{A}_W(i, j)$, $\dots$, en fonction des débits massiques au travers des faces de Ω_u .

- 2) Pour achever la discrétisation, il reste maintenant à discrétiser les coefficients de $\bar{F}_x^c$, *i.e.* calculer l'approximation $\bar{u}_{e,h}$ du débit massique $\bar{u}_e$, *etc.* $\dots$, de manière *consistante* avec l'équation de continuité discrète (A.9).
 - a) Pour cela, interpolez les $\bar{u}_e$, $\bar{u}_w$, $\dots$, en fonction des $\bar{u}_{ij}$ qui apparaissent dans (A.9).
 - b) Enfin, discrétisez les débits $\bar{u}_{ij}$ et donnez l'expression finale de $\bar{u}_{e,h}$, *etc.* $\dots$
- 3) En déduire que le flux discret loin des bords s'écrit :

$$F_{x,h}^c(i, j) \equiv A_E(i, j) u_{i+1,j} + A_W(i, j) u_{i-1,j} + A_P(i, j) u_{i,j} + A_N(i, j) u_{i,j+1} + A_S(i, j) u_{i,j-1}, \quad (\text{A.11})$$

et exprimez les coefficients en fonctions des inconnues de vitesse et des pas du maillage.

- 4) Soient U , V les vecteurs contenant les inconnues discrètes u_{ij} et v_{ij} respectivement. Avec ces notations, le flux discret $F_{x,h}^c$ prend la forme matricielle $\mathcal{C}_{x,h}(U, V)U$, où les $A_E(i, j)$, $A_W(i, j)$, *etc.* $\dots$, sont les uniques coefficients non nuls de la matrice $\mathcal{C}_{x,h}(U, V)$ pour chaque "ligne" (i, j) .

Montrez que cette matrice est antisymétrique, *i.e.* :

$$A_N(i, j) = -A_S(i, j+1), \quad A_E(i, j) = -A_W(i+1, j),$$

et, si la continuité discrète est vérifiée dans les cellules $\Omega_{i+1,j}$ et $\Omega_{i,j}$:

$$A_P(i, j) = 0.$$

Partie II. On considère maintenant les conditions aux limites de Dirichlet sur les parois du domaine. L'implémentation des conditions sur la *vitesse normale* est immédiate, et a été effectuée en cours. On en déduit que la discrétisation (A.11) est valable pour la cellule $\Omega_{i,j}$, tel que $i = 1, \dots, N_x - 1$, et $j = 2, \dots, N_y - 1$. Il reste à implémenter les conditions portant sur la *vitesse tangentielle*. Pour le flux $\mathbf{F}_x^c = \rho \mathbf{v} u$, cela revient à modifier la discrétisation de la partie I près de la paroi inférieure (cellule $\Omega_{i,1}$, Fig. A.3), et supérieure (cellule Ω_{i,N_y} , Fig. A.4).

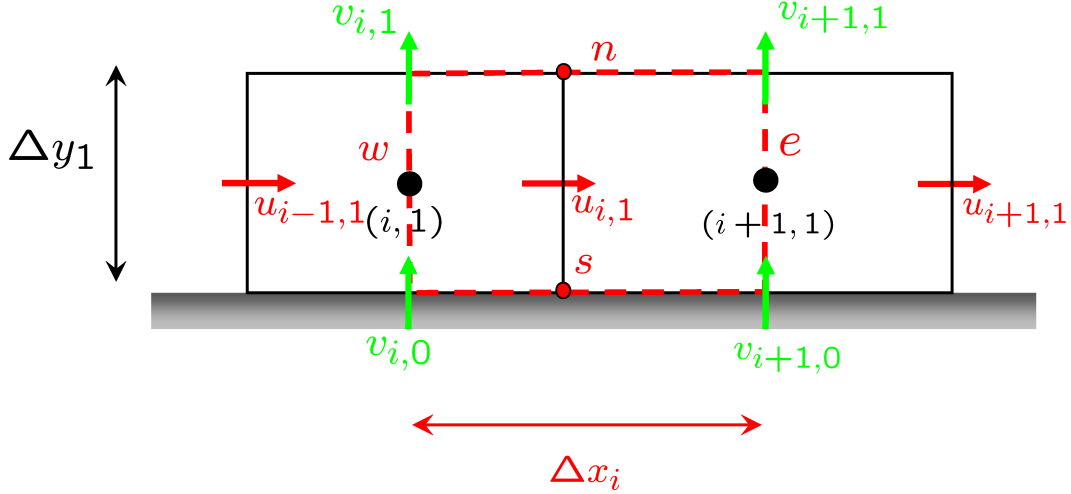


Fig. A.3. Arrangement des inconnues près de la paroi inférieure. On notera $u_{i,0}$ la valeur de la vitesse tangentielle au nœud $\mathbf{x}_s$.

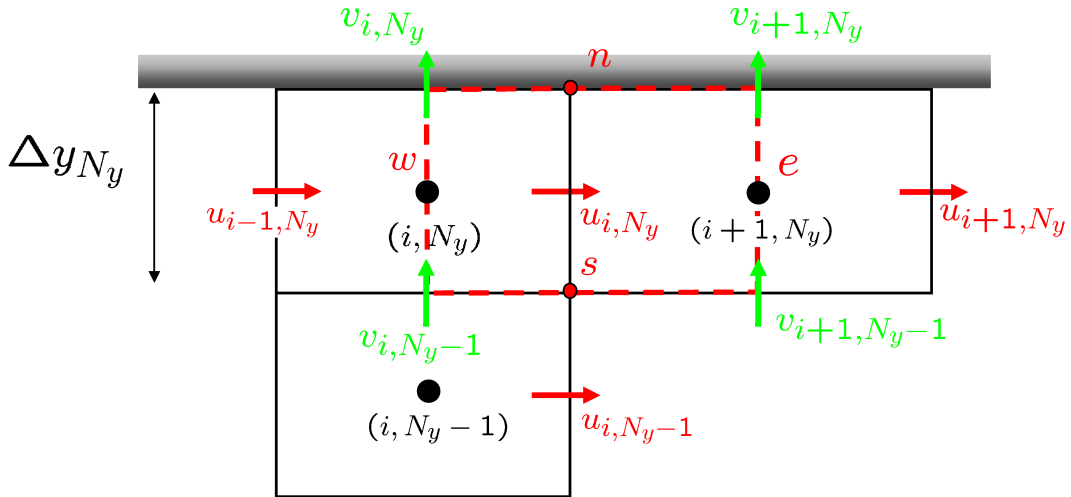


Fig. A.4. Arrangement des inconnues près de la paroi supérieure. On notera u_{i,N_y+1} la valeur de la vitesse tangentielle au nœud $\mathbf{x}_n$.

- 1) Proposez une modification des interpolations effectuées dans la partie I, question 1), qui permettent de tenir compte de chacune de ces conditions aux limites.
- 2) Pour faciliter l'implémentation informatique des conditions aux limites, on introduit les "témoins de bord" définis par :

$$\epsilon_u^N(j) = \begin{cases} 1 & \text{si } 1 \leq j \leq N_y - 1, \\ 0 & \text{si } j = N_y, \end{cases} \quad \epsilon_u^S(j) = \begin{cases} 0 & \text{si } j = 1, \\ 1 & \text{si } 2 \leq j \leq N_y. \end{cases} \quad (\text{A.12})$$

Utilisez ces paramètres pour exprimer de façon unifiée les interpolations précédentes, dans les cas près du bord et loin du bord.

- 3) En déduire l'expression de $F_{x,h}^c$, de la forme (A.11), qui soit valable en tout point (i, j) du maillage. Sachant que ce terme doit être pris de façon explicite en temps, décrivez l'implémentation informatique de ce flux.

Exercice A.5. (Approximation “Symmetry-Preserving” du flux diffusif dans la direction x sur maillages décalés)

On reprend les notations du problème précédent, et on considère l'approximation du flux diffusif :

$$\mathbf{F}_x^d = \mathbf{F}_{x,\text{impl}}^d + \mathbf{F}_{x,\text{expl}}^d, \quad (\text{A.13a})$$

avec

$$\mathbf{F}_{x,\text{impl}}^d = \frac{1}{\text{Re}} \nabla u, \quad \mathbf{F}_{x,\text{expl}}^d = \frac{1}{\text{Re}} \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial v} \\ \frac{\partial u}{\partial x} \end{pmatrix}. \quad (\text{A.13b})$$

Dans la suite, on considérera le cas $\text{Re} = \text{C}^{\text{te}}$.

Partie I. On considère dans cette partie la discrétisation de $\int_{\Gamma_u} \mathbf{F}_{x,\text{expl}}^d \cdot \mathbf{n} \, dS$, que l'on notera $F_{x,\text{expl},h}^d$.

- 1) Décomposez ce flux selon ses contributions élémentaires sur chacune des 4 faces de Ω_u .
- 2) En faisant l'hypothèse de linéarité du flux discret au travers de chacune de ces faces, effectuez l'approximation de chacune de ces contributions.
- 3) Lorsque $\text{Re} = \text{C}^{\text{te}}$, montrez que $F_{x,\text{expl},h}^d$ est nul en utilisant l'équation de continuité discrète (A.9) dans $\Omega_{i,j}$ et $\Omega_{i+1,j}$.

Partie II. On considère maintenant la discrétisation de $\int_{\Gamma_u} \mathbf{F}_{x,\text{impl}}^d \cdot \mathbf{n} \, dS$, loin des bords du domaine. On notera $F_{x,\text{impl},h}^d$ son approximation discrète.

- 1) En suivant les grandes étapes de la discrétisation du flux explicite, donnez l'expression du flux discret sous la forme :

$$F_{x,\text{impl},h}^d = A_E(i, j) u_{i+1,j} + A_W(i, j) u_{i-1,j} + A_P(i, j) u_{i,j} + A_N(i, j) u_{i,j+1} + A_S(i, j) u_{i,j-1}. \quad (\text{A.14})$$

- 2) D'après la question précédente, le flux discret prends la forme matricielle $\mathcal{K}_{x,h} U$. Montrez que cette matrice est symétrique, *i.e.* les coefficients de cette matrice vérifient pour chaque “ligne” (i, j) :

$$A_N(i, j) = A_S(i, j + 1), \quad A_E(i, j) = A_W(i + 1, j).$$

Partie III. On considère maintenant l'implémentation des conditions de Dirichlet dans $F_{x,\text{impl},h}^d$. Comme pour le problème précédent, il faut modifier l'interpolation des flux pour prendre en compte les vitesses tangentielles en $j = 1$ et $j = N_y$.

- 1) On considère le cas $j = 1$. Utilisez la méthode du cours pour implémenter la condition de Dirichlet dans $(\partial u / \partial y)_s$. Montrez que, en posant $\widetilde{\Delta y}_0 = y_1 - y_{\frac{1}{2}}$, on retrouve une formule analogue à (A.14) pour le flux discret (on rappelle que les $\widetilde{\Delta y}_j$ sont définis par (A.7b)).
- 2) Même question pour le cas $j = N_y$.

Partie IV. D'après les parties précédentes, le flux discret $F_{x,\text{expl},h}^d$ est identiquement nul ; $F_{x,\text{impl},h}^d$ a la forme (A.14) et les conditions aux limites sont implémentées.

On considère maintenant l'implémentation pratique de $F_{x,\text{impl},h}^d$, dans le cadre de la résolution d'une équation de la forme :

$$\mathcal{K}_{x,h}U = \text{RHS}. \quad (\text{A.15})$$

Ce système est de taille $N_u = (N_x - 1)N_y$, et chaque ligne correspond à une cellule du maillage. Pour Ω_{ij} loin du bord, une ligne de ce système prend la forme (cf. (A.14)) :

$$A_E(i, j) u_{i+1, j} + A_W(i, j) u_{i-1, j} + A_P(i, j) u_{i, j} + A_N(i, j) u_{i, j+1} + A_S(i, j) u_{i, j-1} = \text{RHS}(i, j). \quad (\text{A.16})$$

Pour Ω_{ij} adjacente aux parois, il faut tenir compte des conditions aux limites. Par exemple, pour $j = 1$ et $i = 2, \dots, N_x - 2$, le système s'écrit :

$$A_E(i, 1) u_{i+1, 1} + A_W(i, 1) u_{i-1, 1} + A_P(i, 1) u_{i, 1} + A_N(i, 1) u_{i, 2} = \text{RHS}(i, 1) - A_S(i, 1) u_{i, 0}, \quad (\text{A.17})$$

puisque $u_{i,0}$ est une donnée au bord et n'est donc pas une inconnue du système linéaire (cf. Fig. A.3). Il faudra donc tenir compte de ce type de modifications lors de l'implémentation informatique du système linéaire.

1) En utilisant les “témoins de bord” (A.12), réexprimez (A.16) sous la forme générique :

$$A_E(i, 1) u_{i+1, 1} + A_W(i, 1) u_{i-1, 1} + A_P(i, 1) u_{i, 1} + A_N(i, 1) u_{i, 2} + A_S^\epsilon(i, 1) u_{i, 0} = \text{RHS}^\epsilon(i, 1).$$

2) Définissons les nouveaux “témoins de bord”

$$\epsilon_u^W(i) = \begin{cases} 0 & \text{si } i = 1, \\ 1 & \text{si } 2 \leq i \leq N_x - 1, \end{cases} \quad \epsilon_u^E(i) = \begin{cases} 1 & \text{si } 1 \leq i \leq N_x - 2, \\ 0 & \text{si } i = N_x - 1. \end{cases} \quad (\text{A.18})$$

De même que dans la question précédente, écrivez la forme générale du système :

$$A_E^\epsilon(i, j) u_{i+1, j} + A_W^\epsilon(i, j) u_{i-1, j} + A_P^\epsilon(i, j) u_{i, j} + A_N^\epsilon(i, j) u_{i, j+1} + A_S^\epsilon(i, j) u_{i, j-1} = \text{RHS}^\epsilon(i, j), \quad (\text{A.19})$$

qui prend en compte les conditions aux limites sur chacune des 4 parois, et qui est donc valable pour *toutes* les “lignes” $i = 1, \dots, N_x - 1$ et $j = 1, \dots, N_y$ du système linéaire.

Partie V. On considère maintenant l'implémentation pratique d'algorithmes itératifs pour résoudre le système linéaire (A.15). Cette implémentation permettra de résoudre ce système *sans stocker la totalité* de la matrice $\mathcal{K}_{x,h}$ en mémoire, mais seulement *ses coefficients non nuls*, i.e. les coefficients qui apparaissent dans (A.19).

Pour la clarté de l'exercice, on considère un algorithme itératif “basique”, mais présentant la plupart des ingrédients d'une méthode sophistiquée : la méthode de Richardson stationnaire avec préconditionneur de Jacobi, qui s'écrit sous forme matricielle :

$$\mathcal{P}(U^{k+1} - U^k) = \alpha(\text{RHS} - \mathcal{K}_{x,h}U^k), \quad \text{pour } k \geq 0, \quad (\text{A.20})$$

avec $\mathcal{P}$ la diagonale de $\mathcal{K}_{x,h}$, et $\alpha \in \mathbb{R}$ un paramètre d'accélération donné.

- 1) En posant $\delta U = U^{k+1} - U^k$, écrivez l'algorithme de cette méthode itérative sous forme matricielle.
- 2) D'après la question précédente, on peut identifier trois étapes importantes dans l'implémentation pratique de cet algorithme, qu'il faut maintenant expliciter sous forme indicielle (i, j) .
 - a) Explicitez le préconditionneur $\mathcal{P}$ sous forme indicielle, et décrivez l'algorithme de résolution de l'équation $\mathcal{P} \delta U = R$, où R est un second-membre donné.
 - b) Décrivez l'algorithme réalisant une opération algébrique de type $AXPY$, de la forme $Y := aX + Y$, pour des vecteurs Y, X et $a \in \mathbb{R}$.
 - c) Décrivez l'algorithme permettant de calculer le résidu de la méthode itérative.
- 3) Utilisez les algorithmes précédents dans le cadre de la méthode de la question 1).
- 4) Dans le cas où on souhaiterait calculer le paramètre d'accélération α de manière dynamique, décrivez l'algorithme permettant de calculer un produit scalaire.

Exercice A.6. (Discrétisation des équations de Navier-Stokes par un schéma de projection semi-implicite)

On reprend les notations des problèmes précédents, et on considère les équations de Navier-Stokes pour un fluide incompressible, soit l'équation de quantité de mouvement :

$$\frac{d}{dt} \int_{\Omega} u_k dV + \int_{\Gamma} (\mathbf{F}_k^c - \mathbf{F}_k^d) \cdot \mathbf{n} dS + \int_{\Gamma} p \mathbf{e}_k \cdot \mathbf{n} dS = 0, \quad (\text{A.21})$$

pour chaque direction $\mathbf{e}_x$ et $\mathbf{e}_y$, ainsi que l'équation de continuité :

$$\int_{\Gamma} \mathbf{v} \cdot \mathbf{n} dS = 0. \quad (\text{A.22})$$

Les flux convectifs et diffusifs de l'Eq. (A.21) avec $k = x$ ont déjà été discrétisés dans les deux problèmes précédents. La discrétisation de l'Eq. (A.22) est donnée par (A.9).

Partie I. Dans cette partie, on achèvera la discrétisation des ces équations par un schéma de projection semi-implicite. On rappelle que $U(t)$, $V(t)$ et $P(t)$ désignent les vecteurs contenant les inconnues discrètes $u_{ij}(t)$, $v_{ij}(t)$ et $p_{ij}(t)$ respectivement.

1) Complétez la semi-discrétisation de (A.21) avec $k = x$. Montrez qu'elle est de la forme :

$$\mathcal{M}_{x,h} \frac{dU}{dt} + \mathcal{F}_{x,h} - \mathcal{K}_{x,h} U + \mathcal{G}_{x,h} P = 0, \quad (\text{A.23})$$

et explicitez les coefficients de la matrice de masse $\mathcal{M}_{x,h}$ et du vecteur $\mathcal{G}_{x,h} P$.

Dans la suite, on supposera que la semi-discrétisation de (A.21) dans la direction $\mathbf{e}_y$ prend une forme similaire³.

2) Utilisez la variante “ δ -forme” du schéma *ER1-PROJ* pour achever la discrétisation.

Partie II. On se consacre maintenant à l'étape de prédiction dans la direction $\mathbf{e}_x$. On suppose que les champs U^n , V^n et P^n à l'instant $t_n = n\Delta t$ sont connus, et on souhaite calculer $\tilde{U}$, prédiction de la vitesse U^{n+1} à l'instant $t_{n+1} = (n+1)\Delta t$.

1) Montrez que pour obtenir $\tilde{U}$ on doit résoudre le système linéaire :

$$(\alpha \mathcal{M}_{x,h} - \mathcal{K}_{x,h}) \tilde{U} = \widetilde{\text{RHS}}_x. \quad (\text{A.24})$$

Explicitez le second-membre $\widetilde{\text{RHS}}_x$ et donnez la valeur du réel α .

2) En utilisant les problèmes précédents, écrivez le système (A.24) sous la forme indicelle (i, j) :

$$H_E(i, j) \tilde{u}_{i+1, j} + H_W(i, j) \tilde{u}_{i-1, j} + H_P(i, j) \tilde{u}_{i, j} + H_N(i, j) \tilde{u}_{i, j+1} + H_S(i, j) \tilde{u}_{i, j-1} = \widetilde{\text{RHS}}_x^\epsilon(i, j), \quad (\text{A.25})$$

et explicitez les coefficients de ce système en fonction de ceux apparaissant dans l'Eq. (A.19).

Partie III. On considère maintenant l'étape de projection. On suppose que $\tilde{U}$ a été calculé à partir de l'Eq. (A.25), et que $\tilde{V}$, prédiction de la vitesse V^{n+1} , a été calculé par une équation analogue.

1) Pour des commodités algorithmiques qui seront éclaircies par la suite, on introduit la variable auxiliaire⁴ Φ telle que $\Phi = (P^{n+1} - P^n)/\alpha$. En éliminant U^{n+1} et V^{n+1} de l'étape de projection, établissez le système linéaire vérifié par Φ sous la forme :

$$\mathcal{A}_h \Phi = \text{RHS}, \quad (\text{A.26})$$

et exprimez les composantes du second-membre en fonction de $\tilde{U}$ et $\tilde{V}$. On rappelle que les composantes de l'opérateur de pression $\mathcal{A}_h$ sont établies dans le cours.

³ On rappelle que le volume de contrôle Ω_v centré autour de v_{ij} est de taille $\Delta x_i \Delta y_j$ pour $i = 1, \dots, N_x$ et $j = 1, \dots, N_y - 1$, et que $\int_{\Gamma_v} p \mathbf{e}_y \cdot \mathbf{n} dS \cong \Delta x_i (p_{i, j+1} - p_{i, j})$.

⁴ Cette variable est souvent appelée “potentiel” de pression. On remarque qu'elle est proportionnelle à la correction de pression $\delta P = P^{n+1} - P^n$ du cours.

- 2) En déduire les corrections de vitesse U^{n+1} , V^{n+1} qui vérifient la contrainte de continuité discrète (A.9), ainsi que P^{n+1} .
- 3) Supposons que l'équation (A.26) soit résolue avec un algorithme itératif jusqu'à ce que la norme du résidu soit inférieure à une tolérance ϵ donnée. Quelles en sont les conséquences sur la vérification de la continuité discrète par U^{n+1} et V^{n+1} ? Discutez de l'intérêt de travailler avec la variable Φ plutôt qu'avec δP .

A.3 Finite-volume method on unstructured mesh

Exercice A.7 (Gradient approximation at cell faces and spurious oscillations).

In this exercise, we will apply the various unstructured gradient approximations of the lecture notes to the particular case of a (orthogonal) rectangular cell of size Δx and Δy . The notations used are the compass notations of Fig. 15.2 of the lecture notes.

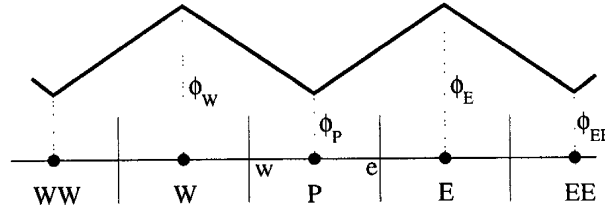


Fig. A.5. Example of spurious oscillations occurring in cell-based gradient approximation.

- 1) Calculate the face area vector $\Delta \mathbf{S}_f$ for each face e, w, n and s of the cell. Check that they sum up to zero.
- 2) Calculate the volume of the cell by using Gauss formula:

$$V_P = \frac{1}{2} \sum_{\text{faces}} \mathbf{x}_f \cdot \Delta \mathbf{S}_f,$$

where $\mathbf{x}_f$ is any position vector on face f.

- 3) Calculate the cell-average gradient of ϕ_P using Green-Gauss cell-based method. Compare with the usual FD central approximation $\delta_0^x \phi_P$.
- 4) Calculate the derivative $\left(\overline{\partial \phi / \partial x} \right)_e$ at face e using arithmetic interpolation. Show that this formula gives rise to checkerboard oscillations as shown in Fig. A.5 by calculating the derivative of $\phi_i = \text{Const.} + (-1)^i$. Compare with the formula obtained with the minimum correction approach.

Exercice A.8 (Steady incompressible flow in a general quadrilateral cell).

We consider a steady incompressible 2D flow over the quadrilateral cell Ω depicted in Fig. A.6.

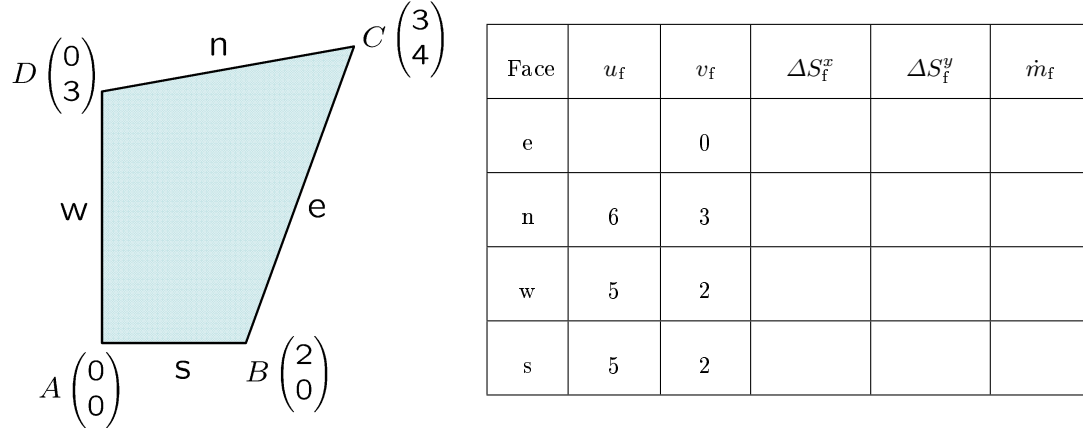


Fig. A.6. Quadrilateral cell Ω with vertices A , B , C and D . The velocity value $\mathbf{v}_f = (u_f, v_f)$ at each face centroid is given in the table.

- 1) Calculate the face area vector $\Delta \mathbf{S}_f$ for each face of the cell. Check that they sum up to zero and complete the table in Fig. A.6.
- 2) Calculate the mass flow

$$\dot{m}_f = \int_{\Gamma_f} \mathbf{v} \cdot \mathbf{n} \, dS \cong \mathbf{v}_f \cdot \Delta \mathbf{S}_f,$$

on faces n, w and s. Complete the table in Fig. A.6.

- 3) Use the continuity equation to find the x -velocity component on the east face.
- 4) Calculate the cell volume with Gauss formula.
- 5) Calculate the cell-averaged gradient of u and v . Deduce the cell-averaged values of the vorticity $\omega = \partial v / \partial x - \partial u / \partial y$ and the divergence $\partial u / \partial x + \partial v / \partial y$. Why is this last quantity equal to zero?
- 6) For 2D incompressible flows, the streamfunction ψ is defined in Cartesian coordinates as :

$$u = -\frac{\partial \psi}{\partial y}, \quad v = \frac{\partial \psi}{\partial x}.$$

Note that ψ is defined up to an arbitrary constant, and that opposite sign convention is also valid. A generalization of this definition is :

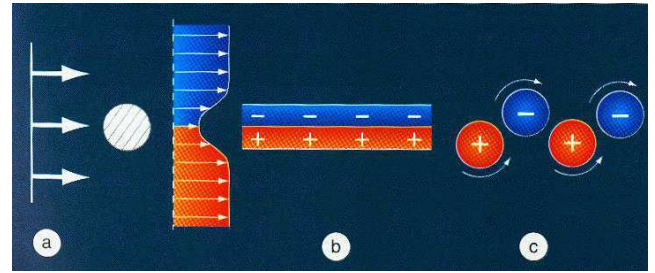
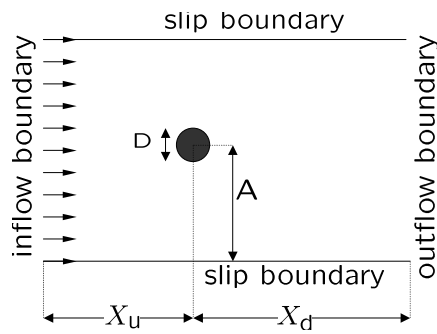
$$\psi_A - \psi_B = \int_A^B \mathbf{v} \cdot \mathbf{n} \, dS, \quad (\text{A.1})$$

which states that the change in ψ between points A and B is equal to the flow rate between any curve Γ connecting A to B .

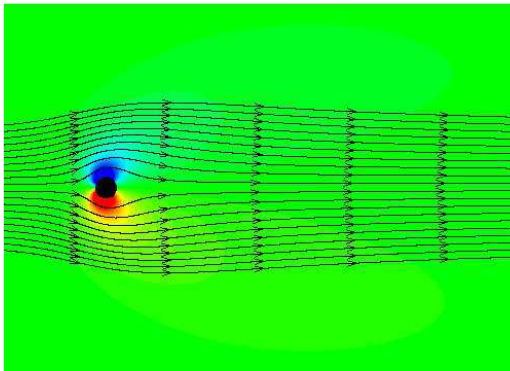
- a) Assuming $\psi_A = 0$, use Eq. (A.1) to calculate the streamfunction values at the other vertices of the cell.
- b) Calculate the streamfunction at vertex A by using the value of ψ_D you found at the previous question. Compare with the initial value $\psi_A = 0$.

Instabilité de Strouhal (1/2)

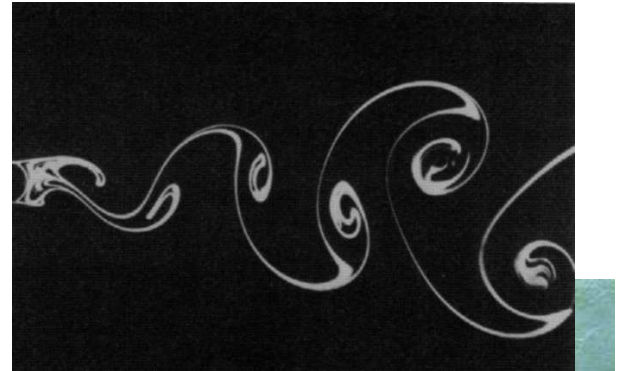
● Ecoulement autour d'un objet non profilé



● Différents régimes suivant la valeur du nombre de Reynolds $Re = \frac{U_\infty D}{\nu}$



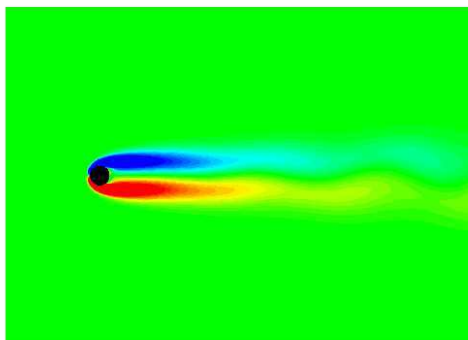
Re petit : Stationnaire



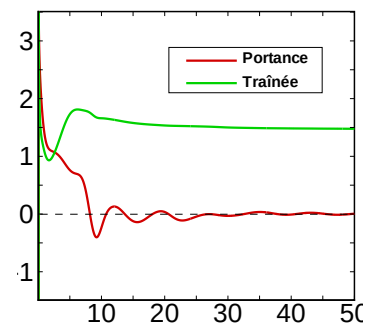
Re grand : Instationnaire (allées de von Kármán)

Instabilité de Strouhal : Systèmes Dynamiques

● Pour $Re < Re_c \simeq 47$: la solution stationnaire est stable :

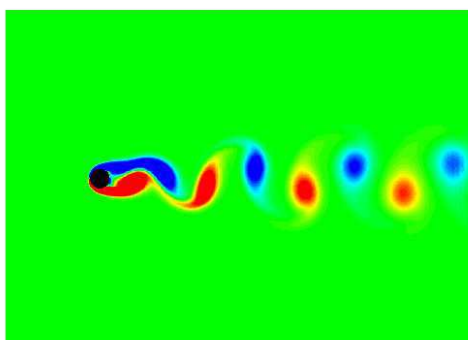


Re = 40 : écoulement stationnaire

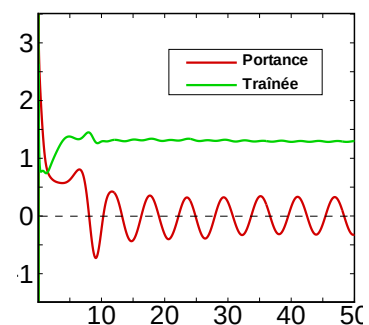


Forces sur le cylindre

● Pour $Re > Re_c$: la solution stationnaire est instable $\Rightarrow$ bifurcation de Hopf



Re = 100 : écoulement périodique

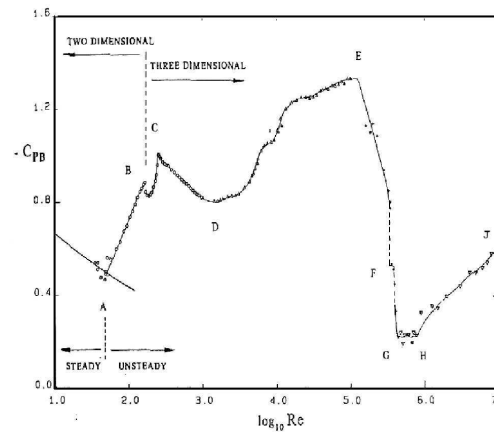


Forces sur le cylindre

Nombre de Strouhal $St = \frac{D}{U_\infty} f$, avec f fréquence du lâcher tourbillonnaire

Instabilité de Strouhal

- Bifurcation successive lorsque $Re \nearrow$ jusau'au régime turbulent



Force de pression en fonction du nombre de Reynolds

- Instabilité **dangereuse** lorsque la **fréquence du lâcher tourbillonnaire** est la même que la **fréquence de résonance** d'une structure. Exemple : *pont de Tacoma Narrows (1950)*



Unsteady flow past circular cylinder with ANSYS FLUENT

Instructor : Olivier Botella olivier.botella@univ-lorraine.fr

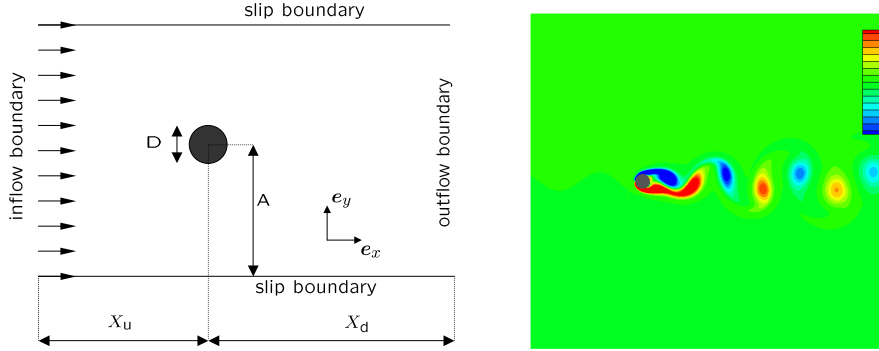


FIGURE 1 – At left : Computational domain and boundary conditions. The domain dimensions are : $D = 0.1$ m, $X_u = 0.8$ m, $X_d = 1.5$ m, $A = 0.8$ m. At left : instantaneous vorticity field $\omega = \partial v / \partial x - \partial u / \partial y$ at $Re = 100$.

1 Problem statement

This lab exercise concerns the CFD analysis of the laminar flow past a circular cylinder, see Fig. 1. The working fluid is *air*, and the *Reynolds* number of the flow is :

$$Re = \frac{\rho U_\infty D}{\mu},$$

with the velocity at the inlet U_∞ as reference velocity, and the cylinder diameter D as reference length.

At low Reynolds number the flow is steady, and above the critical value $Re_c \simeq 47$ the flow undergoes a Hopf bifurcation that causes unsteady oscillations of the wake : the well-known *von Kármán vortex street* (cf. Fig. 1). In the periodic regime ($47 \lesssim Re \lesssim 194$), the frequency f of the vortex shedding is usually given as the dimensionless *Strouhal* number St :

$$St = \frac{D}{U_\infty} f.$$

The Strouhal number is obtained by calculating the drag $F_x(t)$ and lift forces $F_y(t)$ exerted by the flow at the cylinder surface, and by performing a *FFT*¹ of the time-signal when the

1. FFT : Fast Fourier Transform.

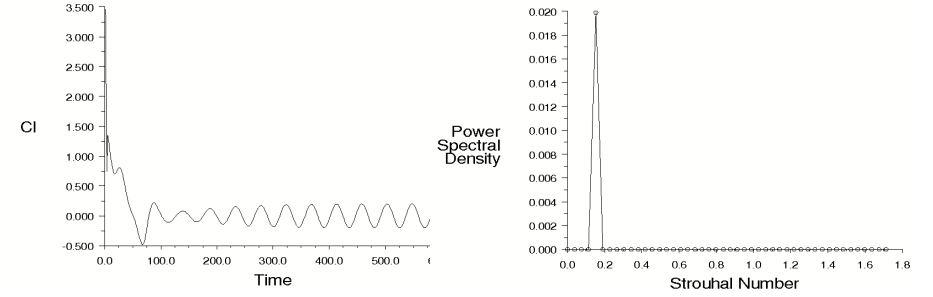


FIGURE 2 – At left : Time history of the lift coefficient C_L (the time is in seconds). At right : FFT of the lift signal in the time interval $t \in [400, 600]$, where the flow reached a periodic state.

flow has asymptotically reached its periodic regime, cf. Fig. 2. The hydrodynamic forces at the cylinder are usually given as the dimensionless *drag* and *lift coefficients* :

$$C_D(t) = \frac{F_x(t)}{\frac{1}{2} \rho U_\infty^2 S}, \quad C_L(t) = \frac{F_y(t)}{\frac{1}{2} \rho U_\infty^2 S},$$

where S (m^2 in SI units) is the projected area of the cylinder².

The purpose of this lab exercise is to compute the flow at $Re = 100$ with ANSYS Fluent , to study the vortex shedding and compare the values C_D , C_L et St given by your computations with the reference values of Table 1.

St	$\overline{C_D}$	$C_{L_{max}}$
0.16 – 0.17	1.21 – 1.41	0.31 – 0.40

TABLE 1 – Experimental and computational reference data of the flow at $Re = 100$. $\overline{C_D}$ denotes the mean value of the drag coefficient and $C_{L_{max}}$ is the maximal value of the drag coefficient, calculated when the flow reached its periodic state.

2 Geometry creation with DesignModeler

The computational domain to create is shown in Fig. 1. It is important to center the domain with respect to its origin (the cylinder center). To create this domain, we consider that the fluid domain consist in a rectangle from which a circle is subtracted (with a boolean operation). For this reason, it is worthwhile to draw the rectangle and the circle in 2 different sketches.

- 0) Go to the **Settings** toolbox to apply the adequate parameters for sketching : **Major Grid spacing** : 0,1 m, **Minor-steps per Major** : 10. For ease of sketching, show the grid and toggle the **Snap** option so the drawings will be aligned to the grid lines.

2. The projected area S is the frontal surface of the body projected on the spanwise direction. For 2D flow past circular cylinder of diameter D , we get $S = D$.

- 1) In a first sketch, create a rectangle with bottom-left corner of coordinates (-0.8 m, -0.8 m). It is then easy to center the rectangle with respect to the origin and give its correct dimensions. In a second sketch, create the circle³.
- 2) Create the surface sketch from these 2 sketches with the **Add Frozen** option. Rename the first surface body as “fluid” and the second as “cylinder”.
- 3) Now, you have to subtract the cylinder body to the fluid body by a boolean subtraction (**Create** → **Boolean** from the menu bar). Use the following details : **Operation** : Subtract, **Target body** : Fluid, **Tool body** : Cylinder, **Preserve Tool Body** : No.

After application of the boolean operation, you should have created 1 part, with 1 “fluid” body. Set its Fluid/solid detail to Fluid, save the project and exit DesignModeler .

3 Mesh creation with Ansys Meshing

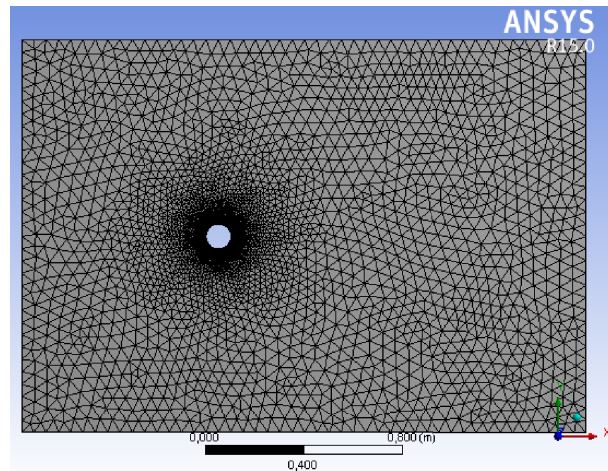


FIGURE 3 – Triangle mesh generated with the curvature size function and edge sizing of the cylinder surface. This mesh has 9715 cells.

The purpose of this part is to generate the unstructured triangle mesh shown in Fig. 3, which is refined around the cylinder surface in order to resolve the fine vortical structures in the boundary layers and the wake. To do so, Ansys Meshing use the following strategies :

- First, you define a **Global Meshing Method** in order to define the type of elements used (quad, tri, ...) and the average size of the elements. In addition, DesignModeler use a “size function” to automatically control the mesh density in the vicinity of highly curved boundaries by using various strategies (proximity, curvature, ..., see the CFD lectures). These global mesh parameters are set in the **Details View** of the mesh. By default, the size function is enabled (Use **Advanced Size Function** : On :Curvature).

3. If needed, use the **Coincident** constraint to put its center to the origin of the domain.

- Secondly, you may ameliorate the mesh in specific areas by inserting **Local Meshing Controls** such as edge or face sizing, inflation, etc. .(see CFD lectures). This step is optional if you only need a crude mesh, but is required to obtained quality meshes.

Following this strategy, the mesh is generated by the following steps :

- 1) **Global Meshing controls** : In the **Tree Outline**, select the mesh and insert a meshing method⁴ for the fluid domain. In its **Details View**, choose **Method** : **Triangles**. Now, define the global meshing controls : in the **Sizing Details** of the mesh, choose **Max Face Size** : 0,05 m and **Growth Rate** : 1,10 for the size function.
- 2) Generate the mesh. You may observe that the size function has detected the cylinder surface as a highly curved boundary and has refined the mesh in its vicinity⁵. However, the mesh is not as fine near the boundary as the one in Fig. 3.
- 3) Add **Local Meshing controls** : insert an **Edge Sizing** method in order to control the cell distribution on the cylinder surface with **Number of Divisions** : 200. The mesh generated will be similar to the one in Fig. 3.

Question 1. Display a figure of the various meshes you created in your Word report. The final mesh, to be used for the CFD analysis, should have the boundary conditions of Fig. 1 labeled with **Named Selections** before exiting Ansys Meshing .

4 Flow computation with Fluent

The unstructured mesh generated with DesignModeler is now used to compute the flow $Re = 100$ (reminder : the working fluid is **air**).

4.1 Step 1 : Solution setup

In this preliminary stage, you will define the flow physics (flow equations, boundary conditions, ...), set up the solver and the discretization methods. Follow the guidelines given in the first lab exercise. Only new steps will be detailed below.

Guidelines for solution setup

- Since the $Re = 100$ flow is time-periodic, use the **Unsteady** solver. The **PISO** algorithm for pressure-velocity coupling is recommended.
- For setting the **Inflow BC**, reply to the following question :

Question 2. Calculate the inflow velocity U_{∞} corresponding the the $Re = 100$ regime.

- Choose the most accurate and robust discretization methods, according to your experience with the first lab exercise.
- Select 2nd-order accurate time-stepping⁶ otherwise your computation may blow up.

4. Alternatively, toggle the volume selection icon , right-click **Insert** → **Method** on the fluid domain.

5. You may also generate a mesh with **Use Advanced Size Function** : **Off** and observe the differences.

6. In the **Solution Method Task Page**, choose **Second Order Implicit** in the **Transient Formulation** drop-down list.

Set up the time history of the drag C_D and lift C_L

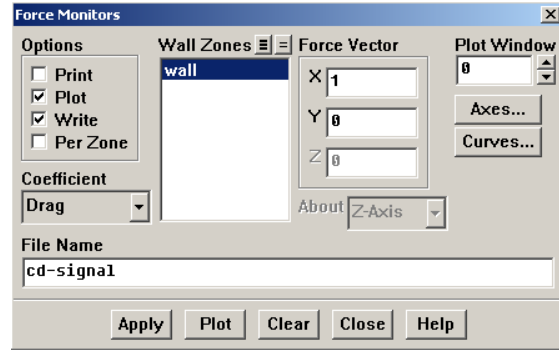


FIGURE 4 – Configuring the Drag Monitor dialog box. The drag is the horizontal component (Force Vector : e_x) of the hydrodynamic force acting on the cylinder surface (Wall Zones). The Plot option will plot the signal in a graphical window during the computation, while the Write option will write the data in the text file cd-signal.

During the unsteady computation, Fluent is able to calculate on-the-fly the forces coefficients $C_L(t)$ et $C_D(t)$ in order to postprocess these results as in Fig. 2. Prepare these calculations with the following steps :

- Define Reference Values of the flow (Solution Setup panel). The projected area S is denoted Area. These references values will be used by Fluent to calculate dimensionless flow quantities such as force coefficients.
- Set up the drag calculation with the Monitors task panel. Click on **Create** → **Drag...** and follow the instructions of the dialog box of Fig. 4. Do the same for the lift coefficient.

Setup the initial condition

In order to break the flow symmetry and trigger the Hopf instability, you will use the initial condition :

$$\mathbf{v}(\mathbf{x}, t = 0) = \mathbf{v}_0 + \mathbf{v}_{\text{pert}}(\mathbf{x}), \quad (1)$$

where $\mathbf{v}_0 = \mathbf{0}$ (fluid is initially at rest) and the perturbation $\mathbf{v}_{\text{pert}} = u_{\text{pert}}(y)\mathbf{e}_x$ such as :

$$u_{\text{pert}}(y) = U_{\infty} \frac{y + |y|}{2y}. \quad (2)$$

In this fashion, the initial condition correspond the sheared velocity :

$$u(y, t = 0) = \begin{cases} U_{\infty} & \text{if } y > 0 \text{ (upper part of the domain),} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

This initial condition is defined in Fluent with the following steps :

- Define analytically $u_{\text{pert}}(y)$ with the Custom Field Functions calculator⁷.

7. Note that the y coordinate is given in the Field Functions drop-down list **Mesh...**

- The initial condition is now defined with the Solution Initialization task panel. Define $\mathbf{v}_0 = \mathbf{0}$ in the fluid domain as in the first lab exercise. Then, add the perturbation $u_{\text{pert}}(y)$ by “patching” it⁸ to $\mathbf{v}_0$.

Question 3. Define this initial condition with Fluent . Plot the velocity field in the whole computational domain and on a vertical section. Displays these plots in your Word report.

4.2 Step 2 : Flow computation

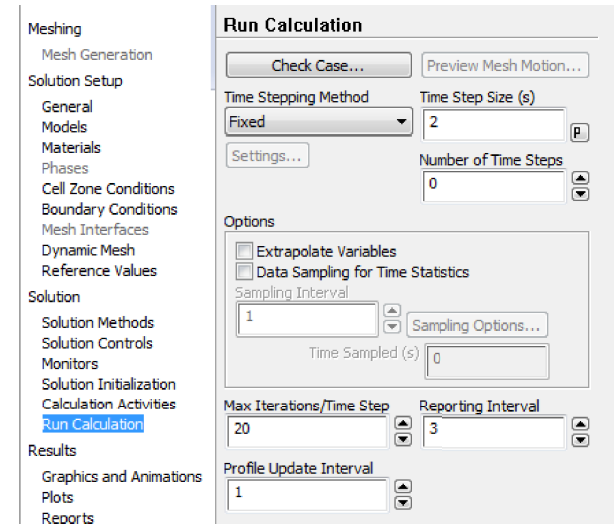


FIGURE 5 – Run calculation task page for the unsteady solver. The Fixed option uses a non-varying value of Δt (Time Step Size), necessary to perform a FFT of the time-signals. Max Iterations/Time Step defines the number of inner iterations of the velocity-pressure algorithm. Reporting Interval is the monitoring frequency of the algorithm.

The task page for the unsteady solver is shown in Fig. 5. You need to define the size of the time-step Δt (in seconds) and the number of time-steps. These values should be chosen from numerical considerations (such as the CFL condition be respected for stable computations), and also from the timescales of the flow at $Re = 100$, as done below.

8. Click on **Patch...** to open the patch dialog box. Choose Variable : X-Velocity and Zones to Patch : fluid, then click on **Patch**.

Determine the time-step size

Question 4. Use the value of the Strouhal number (Table 1) to calculate the period T_{shed} (in seconds) of the vortex shedding. Deduce the value of Δt that allows about 20 time-steps during one vortex-shedding period.

Determine the number of time-steps

Question 5. From the value of Δt and the relation $t_{max} = n_{max}\Delta t$ (in seconds), calculate the number of time-steps n_{max} for computing 600 seconds of the flow.

Compute the flow

Question 6. Click on Calculate to compute the flow. Check that the instability is triggered by monitoring the time evolution of the drag and lift coefficients. Export these plots to your Word reports.

Most Fluent solvers for incompressible flows⁹ are fully implicit, so that stable computations are obtained even with a CFL value greater than 1. Nonetheless, unstable computation may occur due to poor mesh quality, inadequate boundary conditions or stiff initial conditions. If solution blow-up happens in this lab exercise, review first your solution setup. If the computation is still unstable, restart the computation with a lower value of Δt for the first few time-steps, then increase Δt to its original value for the rest of the computation.

It is advised to perform an autosave of the solution during time-integration, see first lab exercise.

4.3 Step 3 : Postprocessing

Now that the computation has been completed, this part is devoted to study the flow physics.

9. With the notable exception of Non Iterative Time Advancement and Projection Method.

Solution visualization

Question 7. Export to your Word report the drag and lift history plots. Calculate $\overline{C_D}$ and C_{Lmax} when the flow reached the periodic regime, and compare your results with those Table1.

Question 8. Display the iso-vorticity lines (see Fig. 1) at a selected flow-time. In order to clearly visualize the cylinder wake, carefully choose an odd number of levels equally distributed around the zero isovalue (Min = -0.5 and Max = -0.5, for example).

Unfortunately, only the norm of the vorticity, named Vorticity Magnitude¹⁰, is defined in Fluent . You need to define the vorticity $\omega = \partial v / \partial x - \partial u / \partial y$ as a Custom Field Function in order to plot it.

Calculation of the Strouhal number

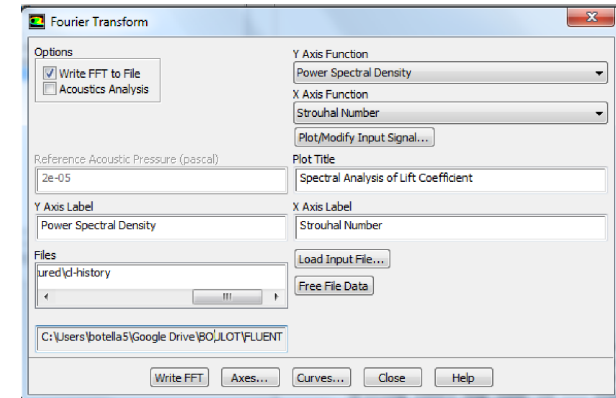


FIGURE 6 – Fourier Transform dialog box for FFT analysis of the lift coefficient.

The frequency of the vortex shedding can be estimated from the time history of the lift signal, cf. Question 7. However, it is customary to use signal processing tools such as the Fourier transform to precisely determine the flow regime (periodic, quasiperiodic, chaotic, ...). From the Plots task page, click on FFT to open the dialog box of Fig. 6, and follow these steps :

- Load the lift history file cl-signal.
- Use **Plot/Modify Input Signal...** to choose a time interval¹¹ where the periodic regime is reached, for example $t \in [400 \text{ s}, 600 \text{ s}]$.

10. In the Contours dialog box of the Graphics and Animation panel, Vorticity Magnitude is available in the Contours of **Velocity...** drop-down list.

11. We recall that the accuracy of frequency calculation with FFT is inversely proportional to the length of

- Click on **Plot FFT**

Question 9. Export the plot of the FFT of the lift signal in your Word report. Calculate the frequency of the vortex shedding and give the incertitude of your calculation. Compare your results with the reference (Table 1).

Graphical animation of the vortex shedding

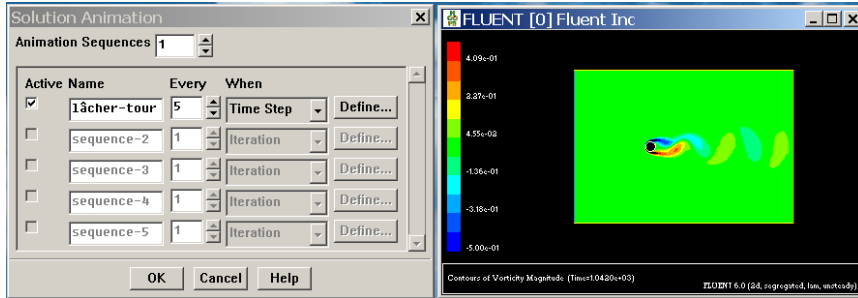


FIGURE 7 – At left : Solution Animation dialog box. At right : Iso-vorticity contours to be animated.

This part is devoted to the animation of the computed solution for visualization of the vortex shedding. This is done by performing a simulation on a few periods (for example $5 T_{\text{shedd}}$ seconds) and periodically saving snapshots of the iso-vorticity lines on the disk (for example every 5 time-steps). The few dozen of snapshots can then be animated in a video-clip.

In the Calculation Activities task panel, open the Solution Animation dialog box of Fig. 7, and follow these steps :

- Activate an animation and label it. With the option **Define...** enter the iso-vorticity lines of Question 8.
- Resume the flow computation for $5 T_{\text{shedd}}$ seconds to generate the snapshots.
- In the Graphics and Animation task page, use Solution Animation Playback to animate the iso-vorticity lines, and save the results as a video-clip.

Question 10. Generate a video-clip of the vortex shedding, and insert it in your Word report.

the time interval (the incertitude is $\pm \frac{1}{2} \frac{1}{\Delta t_{\text{FFT}}}$ s⁻¹ for an interval of length Δt_{FFT} s). So choose a large time interval for your FFT to have accurate results !

Natural convection from a cylinder in a square cavity

Instructors: Maximilien Danda maximilien.danda5@etu.univ-lorraine.fr,
Olivier Botella olivier.botella@univ-lorraine.fr

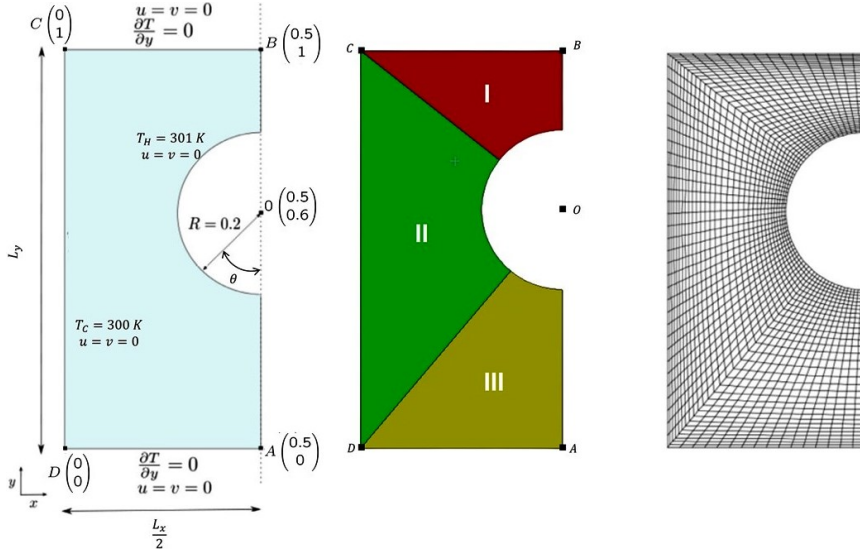


Figure 1: *Left* : Computational domain and boundary conditions. *Center*: Partition of the domain domain for mesh generation. *Right*: Curvilinear mesh used. Domain dimensions are given in meter (m) with $L_x = L_y = 1\text{ m}$.

1 Problem statement

This lab exercise concerns the computation of the natural (or free) convection flow generated by a heated cylinder placed in a square cavity, for which Demirdžić *et al.* [5] published highly accurate numerical solutions. In this problem, the heat equation is coupled with the Navier-Stokes equations in the Boussinesq approximation (see appendix). The computational domain and boundary conditions are shown in Fig. 1 (left). Since the flow displays a symmetry along the vertical axis AB, only half of the geometry is considered. The top and bottom walls of the domain are adiabatic and the temperatures of the circular cylinder and the right wall are respectively $T_H = 301\text{ K}$ and $T_C = 300\text{ K}$. The gravity force $\rho \mathbf{g} = \rho g \mathbf{e}_y$ is in the downward direction, with $g = -1\text{ m} \cdot \text{s}^{-2}$. The fluid used in this exercise has the following custom thermophysical properties in SI units :

- Density $\rho_0 = 1\text{ kg} \cdot \text{m}^{-3}$,
- Specific heat $c_p = 1\text{ J} \cdot \text{K}^{-1} \cdot \text{kg}^{-1}$.
- Thermal conductivity $\kappa = 10^{-4}\text{ W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$,
- Dynamic viscosity $\mu = 10^{-3}\text{ kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$,
- Thermal expansion coefficient $\beta = 0.1\text{ K}^{-1}$.

For comparing the results with [5], all reported quantities will be non-dimensionalized with the following reference values :

- $L_{\text{ref}} = L_x = 1\text{ m}$,
- $T_{\text{ref}} = T_C = 300\text{ K}$,
- $S_{\text{ref}} = L_x \times 1 = 1\text{ m}^2$.

Note that the appendix gives a summary of all the notions on heat transfer in fluids used in this lab exercise, and their definitions in ANSYS Fluent. Read this appendix carefully while performing the exercise.

1.1 Objectives of the lab exercise

The configuration of Fig. 1 is a typical example of the mechanisms of room heating by natural convection (see Appendix A.2), where air motion is triggered by the buoyancy forces¹ generated by the heated cylinder. One of the main objectives of the exercise is to compute the total heat flux Q (measured in watt W) exchanged at the surface of the heated cylinder.

In ANSYS Fluent, the Total Surface Heat Flux² at a surface S is defined by :

$$Q = \int_S \mathbf{q} \cdot \mathbf{n} dS, \quad (1.1)$$

where the (local) density of heat flux $\mathbf{q}$ (in $\text{W} \cdot \text{m}^{-2}$) is given by Fourier's law :

$$\mathbf{q} = -\kappa \nabla T. \quad (1.2)$$

A surface heat flux is usually reported in dimensionless form as the Nusselt number Nu , which is the ratio of convective to conductive heat fluxes (see Appendix A.2.2). For the flow of Fig. 1, Demirdžić *et al.* have performed in [5] a series of highly accurate computations and have calculated with a 7-digit accuracy the Surface Nusselt Number (in m^2) at the surface S of the half-cylinder :

$$NuS = 7.38376\text{ m}^2. \quad (1.3)$$

The aim of this lab exercise is to perform this numerical simulation with ANSYS Fluent and compare your results with the benchmark results of [5].

Question 1. Calculate the Prandtl and Rayleigh numbers of the flow (see appendix). Deduce the principal features of the flow, and verify that the Boussinesq approximation of the flow equations is valid.

¹En français : forces de flottabilité ou poussée d'Archimède.

²ANSYS Fluent expressions are written in **sans serif** font.

2 Geometry creation with DesignModeler

The computational domain of Fig. 1 (center) will be decomposed in 3 parts (or blocks) as shown in Fig. 2, in order to build a high quality curvilinear mesh in each part. To do so, each part needs to be created in a separate sketch (use the **New Sketch** tool ) .

The contour of Part 1 is created by first drawing triangle OBC and the circular arc of center O, then removing the excess lines. Here are the steps for sketching Part 1 :

Step 0) Go to the **Settings** toolbox to apply the appropriate settings for this sketch : **Major Grid Spacing** : 0.1 m, **Minor-Steps per Major** : 5. Show the grid and toggle the **Snap** option for aligning the drawings to the grid lines.

Step 1) **Sketching** of triangle OBC : Draw segments BC and BO and set their dimension. Draw segment OC.

Step 2) Draw a circular arc between OC and OB using the **Arc by center** tool and set the radius value.

Step 3) Remove excess lines using the **Trim** tool.

Once Part 1 is drawn, apply the same strategy for the other parts in distinct sketches. Here are some tips for the geometry creation :

- It is important that each sketch forms a closed surface.
- Set the correct dimensions of the triangle and the circular arc before using the **Trim** tool.
- For creating the surface sketches, use the **Add Frozen** operation to keep the 3 surfaces separated.

Question 2. To check that the generated geometry is correct, verify that its area A_{geom} (in m^2) is equal to its analytic value :

$$A_{geom} = \frac{(L_{ref}^2 - \pi R^2)}{2}.$$

Display the geometry in your Word report.

3 Mesh generation with Ansys Meshing

The purpose of this section is to generate the curvilinear mesh shown in Fig. 1 (right), which is refined near the boundaries of the domain, in particular near the left wall and the cylinder surface to account for the thin thermal boundary layers.

A curvilinear grid is a structured grid for complex domains where the grid lines are curved to fit the boundaries of the physical domain. Such grids are built by coordinate transform of simple Cartesian domains, as shown in Fig. 2. To ease the coordinate transform performed by the meshing software, the physical domain is decomposed in 3 blocks, such as each is block is meshed with a “bottom up” strategy starting with the edge meshing.

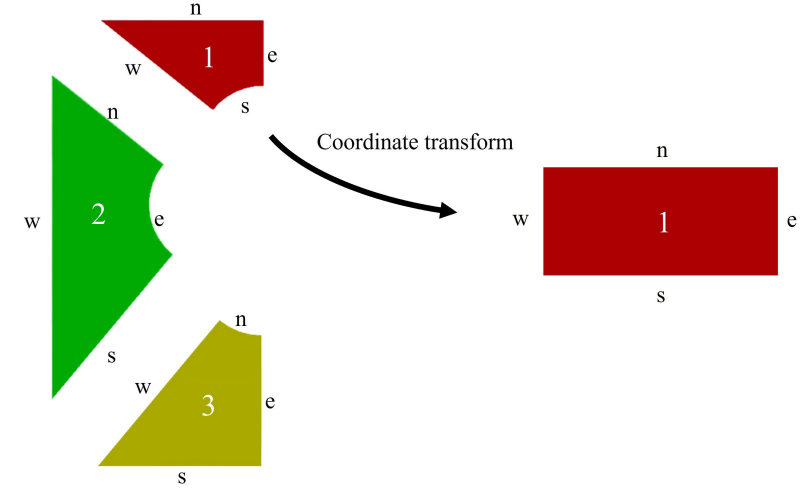


Figure 2: Edge labelling of the 3 blocks of the curvilinear grid.

		# of divisions	Bias type	Bias factor	Name
I	n	64	X	No bias	wall_top
	e	128	- - - - -	3	symmetry
	s	64	X	No bias	wall_h
	w	128	- - - - -	3	X
II	e	128	X	No bias	wall_h
	s	128	- - - - -	3	X
	w	128	X	No bias	wall_c
III	n	64	X	No bias	wall_h
	s	64	X	No bias	wall_bottom
	e	128	- - - - -	3	symmetry

Table 1: Sizing details for the edges shown in Fig. 2. **Number of Divisions** corresponds to the number of edges vertices N . **Bias Type** creates a graded mesh near the edge boundaries, where the **Bias Factor** is the ratio of the largest element to the smallest element.

Guidelines for mesh creation :

- The sizing details of the edges are given in Table 1.
- Use **Behavior** : Hard for all edges³.
- Use quadrangular cells for the face meshing.
- For the ease of the solution setup in **Fluent** , give meaningful names to the external edges of the domain with the **Named Selection** tool.

Question 3. *Display the mesh in your Word report. Calculate the expected number of cells of each block and verify that the total is correct in the Mesh Statistics of Ansys Meshing .*

4 CFD analysis with ANSYS Fluent

This part is dedicated to the computation and the analys of the results on the natural convection flow. Follow the guidelines of the previous lab exercises. Only the new steps will be detailed below.

4.1 Step 1 : Solution setup

In this preliminary stage, you will define the flow physics (flow equations, working fluid, cell zone and boundary conditions, operating conditions, reference values, ...), select the iterative solver and the discretization schemes.

4.1.1 Boussinesq setup

The setup of the Boussinesq model of Appendix A.2.1 amounts to set the values of the various parameters in the momentum equation (A.6) and the energy equation (A.7).

- The gravity acceleration $\mathbf{g} = g\mathbf{e}_y$ is defined in the **General** task page. It can also be set in the **Operating Conditions** panel of the **Physics** menu bar.
- The thermophysical properties of the custom Boussinesq fluid of Section 1 are set by creating a new material in the **Materials** task page. In the **Create/Edit Materials** panel, use the **boussinesq** property for the density with the value of ρ_0 .
- The average temperature T_0 corresponds to the **Operating Temperature (k)** in the **Operating Conditions** panel. Set its value according to the guidelines of Appendix A.2.1.

³In contrary to **Soft**, the **Hard** behavior overrides the **Max face Size** and other sizing properties used elsewhere in the mesh.

4.1.2 Guidelines for the solution setup

- **General** task page : Use the steady pressure-based solver and set the gravity acceleration.
- **Models** task page : Use the **Viscous (laminar)** model and activate the energy equation.
- **Materials** task page : Define the custom Boussinesq fluid and use it in the **Cell Zone Conditions**.
- **Boundary Conditions** task page : Set the boundary conditions of Fig. 1 (left) for both momentum and energy equations. Note that a Neumann boundary condition for temperature amounts to prescribing the heat flux density $\mathbf{q}$ (1.2).
- **Reference Values** task page : Set the reference values given in Section 1. They will be used for reporting the dimensionless quantities of Appendix A.3.

4.2 Step 2 : Flow computation

Here are some advice for the solution methods to use :

- Use the **SIMPLE** scheme for pressure-velocity coupling.
- Use the most accurate gradient interpolation available : for ANSYS V2020 and later versions, **Least Squares Cell Based** is recommended.
- For high-Rayleigh number natural convection, **Body Force Weighted** is the recommended pressure interpolation scheme.
- Use a second-order method for convection such as the **QUICK** scheme for all fluid flow and thermal equations.

Note that the choice of the discretization methods and the version of ANSYS may have an impact on the last digits of the alphanumeric data reported in Question 4 to 7, but will not change the outcome of the exercise. In your Word report, report your results with a 7-digit accuracy as in (1.3).

4.2.1 Initializing the natural convection flow simulation

Actual Problem	Initial Condition
Heat Transfer	Isothermal
Natural convection	Low Rayleigh number
Combustion / reacting flow	Cold flow (no combustion)
Turbulence	Inviscid (Euler) solution

Table 2: Best practices for initializing coupled flow simulations.

The simulation of fluid flows coupled with other transport phenomena like heat or mass transfer can be plagued with numerical convergence difficulties. To mitigate these issues, it is customary to start the computation with a “simplified” version of the flow, for example by

using the isothermal flow as the initial condition of a coupled heat and flow problem. The best practices for solving coupled problems are summarized in Table 2. However, in this lab exercise the convective flow motion is induced by the temperature differences in the domain : the heat equation should be solved first.

Question 4. Solve the heat transfer problem by conduction only (Tip : set the gravity acceleration to $g = 0$ in order to decouple the energy equation (A.7) to the momentum equation (A.6)). Verify there is no flow motion and display the isothermal lines in the computational domain (Tip : Use Colormap Size=10 in the Colormap Options ... panel to display sharper contour lines). Export this plot to your Word report and comment the results.

This solution will be used as the initial condition of the natural convection problem considered next.

4.2.2 Solving the natural convection flow with default convergence settings

In this part, you will perform the natural flow computation (with $g = -1$) with the basic procedure to monitor the convergence of the numerical solution, which is to check that the residuals of the discretized equations fall below a specific tolerance ϵ . By default, the tolerance thresholds of Fluent are :

- $\epsilon_u = 10^{-3}$ for flow equations,
- $\epsilon_T = 10^{-6}$ for thermal equations.

It is recommended that the thermal tolerance ϵ_T be 3 orders of magnitude below the fluid flow tolerance ϵ_u , for reasons that will be explained next.

One of the objectives of this simulation is to calculate the Surface Nusselt Number Nu_S at the surface S of the half-cylinder and compare with its benchmark value (1.3). For this purpose, set up the calculation of Nu_S on the Report Definitions task page. Appendix A.3 gives information on the ANSYS Fluent calculation of Nu_S and the other heat transfer quantities used in this lab exercise.

Question 5. Compute the flow with the default tolerance thresholds. In your Word report, report (and plot) the monitoring history of the residuals and Nu_S (Tip : Use Reporting Interval=10 in the Run Calculation task page to avoid printing the residuals at each iteration). When the solution is converged, report the value of Nu_S and its normalized error with respect to its benchmark value.

4.2.3 Checking overall flux conservation

The basic procedure for assessing convergence may not be sufficient to obtain a physically realistic computations. It is thus recommended to lower the threshold ϵ_u and ϵ_T , but it is also important to check the convergence of relevant physical quantities such as forces or heat fluxes at solid walls.

For example, the equation of energy (A.7) states that the total heat flux rate (1.1) is conserved at the external boundaries cb of the domain when the steady state is reached :

$$\sum_{cb} Q = 0. \quad (4.1)$$

A physically realistic computation should verify this balance of fluxes.

Question 6. For the simulation with $\epsilon_u = 10^{-3}$ and $\epsilon_T = 10^{-6}$ computed in the previous question, report the (normalized) value of the heat flux imbalance (4.1). Give your conclusions on the convergence of this simulation.

4.2.4 Solving the natural convection flow with improved convergence settings

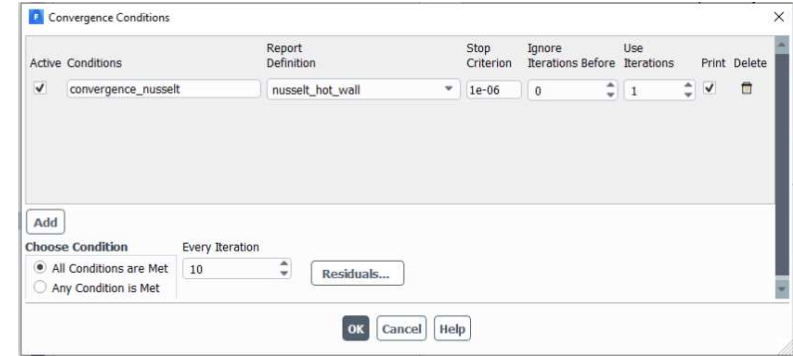


Figure 3: Convergence Conditions panel for Nu_S . This panel defines the condition $|Nu^k - Nu^{k-p}|S < \epsilon$ where $p = \text{Use Iterations}$ and $\epsilon = \text{Stop Criterion}$.

In addition to the residual history, Fluent allows to monitor the convergence of global physical quantities during the iterative solution procedure. Of particular importance in this lab exercise is the monitoring of the Surface Nusselt Number Nu_S set up previously. Use the Convergence Conditions panel (see Fig. 3) to activate this monitoring.

Question 7. Perform a simulation with $\epsilon_u = 10^{-6}$, $\epsilon_T = 10^{-9}$ and $\epsilon_{Nu} = 10^{-6}$ for the surface Nusselt number. In your Word report, report (and plot) the same quantities as the previous two questions. Give your conclusions.

4.3 Step 3 : Post-processing

This part is devoted to the analysis of the results obtained for the natural convection problem.

Question 8. Display the contour plot of the temperature in the domain. Export this graph to your Word report. Compare with the plot obtained in Question 4 for conduction only.

Question 9. Create a Custom Field Function for the Boussinesq density $\rho(T)$ given by (A.3). Display the contour plot of the density in your Word report. Comment your results.

Question 10. Display the velocity vectors in the domain (Tip : for better visualisation of the flow circulation, color the vectors by temperature, and adjust the values of the *Scale* and *Skip* options). Export this graph to your Word report. Use the plots obtained in Questions 8-10 to explain the phenomenon of natural convection in a closed domain.

The final question is devoted to the plotting of the local Surface Nusselt number along the surface S of the hot cylinder. Demirdžić has given in [5] benchmark values for this quantity at selected locations on S . The plot of these benchmark values is performed in the Matlab file `plot_local_surf_nusselt.m` available in the course website⁴.

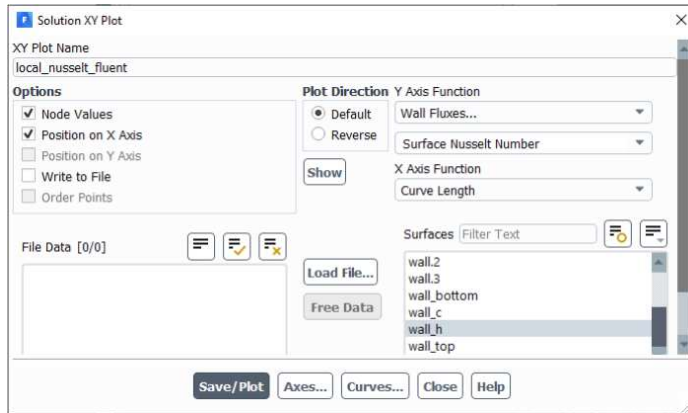


Figure 4: Panel for plotting the local Nusselt number on a selected surface. **Solution X Axis Function : Curve length** corresponds to the the curvilinear abscissa of the surface⁵. The option **Write to File** allows you to write the results in a text file ready to be uploaded in Matlab .

⁴Course Website on ARCHE : <https://arche.univ-lorraine.fr/course/view.php?id=9396>.

Question 11. Use the *Solution XY Plot* panel (see Fig. 4) to plot the local Nusselt number along the cylinder surface. Export your data in Matlab and compare your results with the benchmark values in the same Matlab plot.

A Appendix : Basics on convective heat transfer

This part briefly addresses the basic notions of heat transfer in fluids that are useful for performing this lab exercise. More information is available in standard textbooks, *e.g.* [3, 8, 2]. In addition, the last section gives the ANSYS Fluent definition of heat transfer quantities.

A.1 Basic regimes of heat transfer in fluids

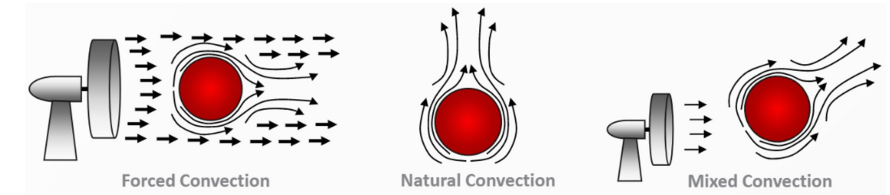


Figure 5: Basic regimes of convective heat transfer from a hot cylinder.

Convective heat transfer refers to the transport of energy in fluid flows, which is coupled to the transfer of momentum (the Navier-Stokes equations). The 3 main types of heat transfer in fluid are depicted in Fig. 5. In *forced convection*, fluid motion is driven an external force (pumps, fans, ...), enabling energy transfer due to temperature differences. In *natural (or free) convection*, fluid motion is caused by buoyancy forces due to the differences between hot and cold zones in the fluid. Finally, *mixed convection* refers to the regime where natural and forced convection are both present, with similar intensities.

A.2 Natural convection

As shown by the equation of state for a perfect gas :

$$p = \rho r T, \quad (\text{A.1})$$

at constant pressure the gas density decreases when the temperature increases, allowing the gas to expand. This effect plays a central role in natural convection, where fluid motion is generated by the spatial variations of density (*i.e.*, stratification) in conjunction with the gravitational field : the so-called buoyancy forces. In the configurations of Fig. 6, the fluid heated from below expands and rises up spontaneously due to the buoyancy forces generated by the temperature differences. In the closed domain of Fig. 6 (right), the fluid that reaches the top cools down and then sinks, enabling flow circulation and mixing in the domain.

⁵Issues with using *Curve length* may be related to the fact that Fluent is running in parallel mode. To enter the serial mode, close the Fluent application. Then, deselect the option *Run Parallel Version* in the properties of the *Setup* cell of your WorkBench project.

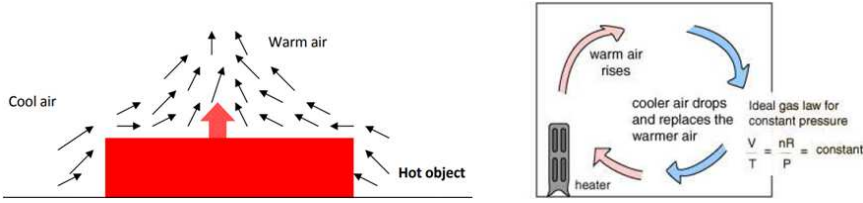


Figure 6: *At left* : Natural convection from a hot body in an open domain. *At right* : Heating in a closed room through natural convection.

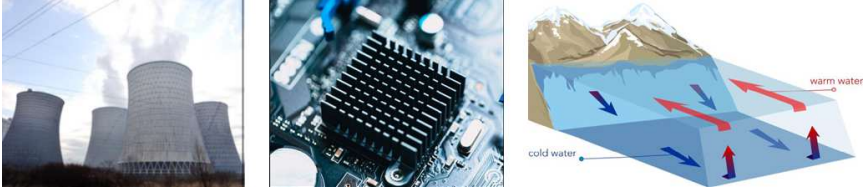


Figure 7: Examples of natural convection flows in industrial and natural processes.

In the absence of flow motion, heat transfer would occur by conduction only at a much lower rate. Hence, natural convection plays an important role in industrial applications such as power generation, food processing, HVAC⁶ or electronic cooling (Fig. 7 and Ref. [1]). It is also the driving mechanism of many natural processes such as oceanic and atmospheric currents.

A.2.1 Boussinesq approximation of the Navier-Stokes equations

The Boussinesq approximation (1903) is the most usual approach for modelling natural convection flows. This approximation is valid when the temperature variations are “sufficiently” small around an average temperature T_0 to induce small density fluctuations around $\rho_0 = \rho(T_0)$.

The Boussinesq approximation assumes that all thermophysics coefficients of the Navier-Stokes and energy equations are constant, except in the body force term $\rho \mathbf{g}$ of the momentum equation which accounts for gravity. In order to express the buoyancy force of natural convection, the density is assumed to depend on temperature only : $\rho = \rho(T)$. For constant pressure, the Taylor expansion of $\rho(T)$ at $T = T_0$ is :

$$\rho(T) = \rho(T_0) + \left. \frac{\partial \rho}{\partial T} \right|_p (T - T_0) + \mathcal{O}(T - T_0)^2, \quad (\text{A.2})$$

which yields the following linear approximation of the density :

$$\rho(T) = \rho_0 - \beta \rho_0 (T - T_0), \quad (\text{A.3})$$

where the thermal expansion coefficient is :

$$\beta = -\frac{1}{\rho_0} \left. \frac{\partial \rho}{\partial T} \right|_p. \quad (\text{A.4})$$

⁶HVAC: Heating, ventilation and air-conditioning.

In practical applications, the Boussinesq approximation is considered valid when the temperature variations δT do not cause large density changes, *i.e.* according to (A.3) :

$$\beta \delta T \ll 1. \quad (\text{A.5})$$

Inserting the Boussinesq approximation (A.3) into the momentum equation gives :

$$\rho_0 \left(\frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot (\mathbf{v} \otimes \mathbf{v}) \right) + \nabla (p + \rho_0 \mathbf{g} \cdot \mathbf{x}) - \mu \nabla^2 \mathbf{v} = -\beta \rho_0 (T - T_0) \mathbf{g}, \quad (\text{A.6})$$

where $\mathbf{x} = (x, y, z)$. The energy equation is simply :

$$\rho_0 c_p \left(\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{v} T) \right) - \kappa \nabla^2 T = 0. \quad (\text{A.7})$$

The RHS in the momentum equation (A.6) is responsible of the buoyancy effects and depends on the temperature T , which strongly couples the Navier-Stokes equations with the energy equation. The average values T_0 and ρ_0 in (A.6) are problem-dependent and have to be set by the user. Their “exact” values does not impact the solution to the Boussinesq equations, since $\rho_0 \mathbf{g} \cdot \mathbf{x}$ in the LHS of (A.6) accounts for the hydrostatic pressure. However, incorrect values of T_0 and ρ_0 may lead to round-off errors and impact the convergence of iterative solvers. For closed domains, it is advised to set T_0 and ρ_0 as their averaged values in the domain (using information at the boundaries, for example). For open domains, it is advised to use the external ambient density and temperature.

A.2.2 Heat transfer coefficient and dimensionless numbers for natural convection

For calculating the heat flux rate Q/S across a surface S between the fluid and the solid, one of the key step is to evaluate the heat transfer coefficient h (in $\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$) which is defined by Newton’s law of cooling $Q/S = h \Delta T$. Analytic calculations of h can be done for canonical configurations only. For complex configurations, h needs to be numerically computed (as in this lab exercise), or estimated through empirical correlations that involves dimensionless numbers representative of the flow under consideration, such as the Nusselt number $Nu = hL/\kappa$:

$$Nu = F(Pr, Gr), \quad \text{or simply} \quad Nu = F(Ra).$$

In the context of this lab exercise on ANSYS **Fluent**, the precise definition of the Nusselt number Nu , the Prandtl number Pr , the Grashof number Gr and the Rayleigh number $Ra = Pr Gr$ are given next.

Prandtl number

The Prandtl number Pr of a fluid is the ratio of its momentum diffusivity to its thermal diffusivity :

$$\begin{aligned} Pr &= \frac{\text{Momentum diffusivity}}{\text{Thermal diffusivity}} = \frac{\mu/\rho}{\kappa/(c_p \rho)} \\ &= \frac{c_p \mu}{\kappa}. \end{aligned} \quad (\text{A.8})$$

Since Pr does not depend on scale variables such as L_{ref} (or U_{ref}), the Prandtl number is representative of the material properties of the fluid only. For large values of Pr , momentum diffusion dominates and heat is exchanged at a lower rate. In consequence, the thermal boundary layer of laminar flows on a flat plate is thinner than the hydrodynamic boundary layer (see Fig. 8).

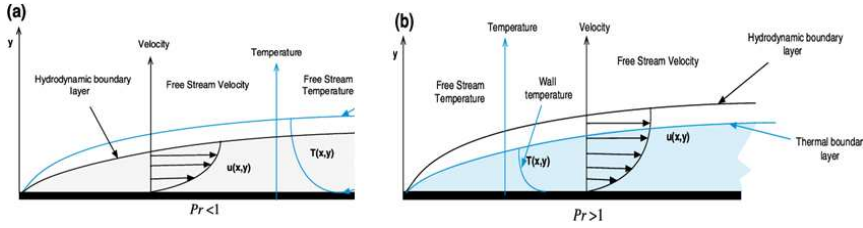


Figure 8: Thermal and momentum boundary layers for (a) $Pr < 1$ and (b) $Pr > 1$ (taken from [4]).

Grashof and Rayleigh numbers

The Grashof number Gr is defined as the ratio of the buoyancy forces (which drive the flow) to viscous forces (which resist flow motion). With the notations of this lab exercise, it is given by:

$$Gr = \frac{\text{Buoyancy forces}}{\text{Viscous forces}} = \frac{gL_{\text{ref}}^3 \beta (T_H - T_C) \rho_0^2}{\mu^2}. \quad (\text{A.9})$$

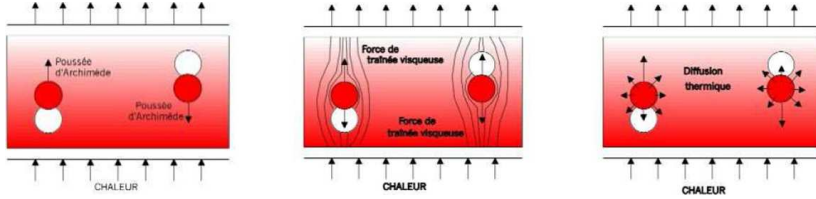


Figure 9: The Rayleigh number (A.10) expresses the competition between driving forces (buoyancy, at left) and resistive forces (viscous resistance, middle ; and thermal diffusion that increases the heat exchanged by the displaced fluid parcel, at right).

Many empirical correlations for the Nusselt number involves the Rayleigh number Ra given by :

$$\begin{aligned} Ra &= \frac{\text{Buoyancy forces}}{\text{Viscous forces}} \frac{\text{Momentum diffusivity}}{\text{Thermal diffusivity}} \\ &= Gr Pr = \frac{gL_{\text{ref}}^3 \beta (T_H - T_C) \rho_0}{\mu \kappa}, \end{aligned} \quad (\text{A.10})$$

which expresses the ratio of the driving buoyancy forces to the stabilizing forces of viscosity and thermal diffusivity (see Fig. 9).

In natural convection the Reynolds number no longer characterize the flow, since a reference value U_{ref} cannot be defined. Instead, the Grashof or Rayleigh numbers are used. For example, in the configuration of Fig. 10, the fluid is at rest for low values of Ra and heat transfer occurs by conduction only. Above a critical value of $Ra_c \approx 1700$, gravity forces become more prominent and induce fluid circulation, forming the so-called Bénard convection cells [7]. At higher values of Ra , the flow becomes turbulent. In engineering applications, the flow is considered turbulent when $Ra \geq 10^9$ [3, 2, 1].

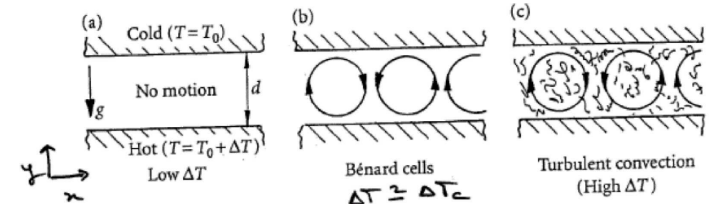


Figure 10: Example of Rayleigh-Bénard convection between 2 horizontal plates with $L_x \gg L_y$. The fluid is heated from below with a temperature difference of ΔT between the 2 plates. The various flow regimes are sketched for increasing value of the Rayleigh number $Ra = f(\Delta T)$.

Nusselt number and heat transfer coefficient

The Nusselt number Nu is a dimensionless parameter that measures the amplification of heat transfer across a surface S due to the convective fluid motion. It is expressed as the ratio of the heat flux in the presence of convection to the heat flux given by conduction only [6, 3] :

$$Nu = \frac{Q_{\text{convection}}}{Q_{\text{conduction}}}. \quad (\text{A.11})$$

In the case of the configuration of Fig. 1, $Q_{\text{convection}}$ is given by (1.1). The conduction heat flux can be approximated as $Q_{\text{conduction}} \simeq \kappa \nabla T_{\text{conduction}} S$, where $\nabla T_{\text{conduction}}$ represents the temperature difference between the hot cylinder and the cold left wall :

$$\nabla T_{\text{conduction}} \simeq \frac{T_H - T_C}{L_x}, \quad (\text{A.12})$$

resulting in the following expression of the Nusselt number (A.11) :

$$Nu = \frac{Q}{\kappa \frac{T_H - T_C}{L_x} S}. \quad (\text{A.13})$$

The Nusselt number is the dimensionless form of the heat transfer coefficient h given by Newton's law of cooling :

$$\frac{Q}{S} = h(T_H - T_C). \quad (\text{A.14})$$

By combining this expression with (A.13), one retrieves the well-known relation :

$$Nu = \frac{h L_x}{\kappa}. \quad (\text{A.15})$$

For practical flow configurations, the expression of the Nusselt number and heat transfer coefficient used by ANSYS Fluent will be given next.

A.3 Postprocessing heat transfer with ANSYS Fluent

The calculation of heat transfer quantities is available in the **Results** → **Reports** task page (see Fig. 11). The following definitions refer to calculations performed *on a given surface S of temperature T* . For calculating quantities such as the heat transfer coefficient or the Nusselt number, the value of L_{ref} and T_{ref} needs to be set in the **Reference Values** task page.



Figure 11: Reports panel from the Results menu.

Panel **Reports** → **Fluxes...**

The Total Heat Transfer Rate (in W) corresponds to the heat flux Q given by (1.1).

Panel **Reports** → **Surface Integrals...**

- The report type **Area** gives the value of S in m^2 .
- The report type **Integral** corresponds to quantities integrated on surface S . For the field variable **Wall Fluxes...**, the following quantities are available :
 - Total Surface Heat Flux (in W): It is the same as the Total Heat Transfer Rate Q defined above (up to a minus sign).
 - Surface Heat Transfer Coef. (in $W \cdot K^{-1}$): In analogy with (A.14), it corresponds to the quantity hS calculated by ANSYS Fluent as :

$$hS = \frac{Q}{T - T_{\text{ref}}}. \quad (\text{A.16})$$

- Surface Nusselt Number (in m^2): In analogy with (A.13), it corresponds to the quantity NuS calculated by ANSYS Fluent as :

$$NuS = \frac{Q}{\kappa \frac{T - T_{\text{ref}}}{L_{\text{ref}}}}. \quad (\text{A.17})$$

- The report type **Area-Weighted Average** gives the above quantities divided by the surface area S .

References

- [1] Natural Convection - ANSYS Innovation Courses. <https://courses.ansys.com/index.php/courses/natural-convection/>.
- [2] R. Bird, W. Stewart, and E. Lightfoot. *Transport Phenomena*. Wiley, 2007.
- [3] P. Boulet and D. Lacroix. Notes de cours M1 Energie - UE Transferts Thermiques. Université de Lorraine.

- [4] M. Darwish, L. Mangani, and F. Moukalled. *The Finite volume method in computational fluid dynamics: an advanced introduction with OpenFOAM® and Matlab®*. Springer, 2016.
- [5] I. Demirdžić, Ž. Lilek, and M. Perić. Fluid flow and heat transfer test problems for non-orthogonal grids: Bench-mark solutions. *International Journal for Numerical Methods in Fluids*, 15(3):329–354, 1992.
- [6] M. Griebel, T. Dornseifer, and T. Neunhoffer. *Numerical Simulation in Fluid Dynamics: A Practical Introduction*. SIAM, 1998.
- [7] E. Guyon, J.-P. Hulin, L. Petit, and C. D. Mitescu. *Physical Hydrodynamics*. Oxford University Press, 2015.
- [8] Y. Jannot, C. Moyne, and A. Degiovanni. *Transfert de Chaleur 1 : Conduction et Convection*. ISTE Editions, 2022.

Bibliographie

En ce qui concerne l'analyse numérique "classique", niveau Licence ou Maîtrise, citons le livre de Quarteroni [9], et celui de Ciarlet [2] portant notamment sur l'optimisation. Citons aussi sur le même sujet le récent ouvrage de Bergonioux [1]. On rappelle que le grand classique présentant le côté pratique de l'analyse numérique est le "Numerical Recipes" de Press et al. [8], qui regroupe de nombreux programmes en *C* ou *Fortran*, et qui est disponible gratuitement sur le *Net*⁷.

Au sujet des méthodes numériques, le bouquin de Sainsaulieu [11] est un excellent complément de ces notes de cours, bien qu'il possède un formalisme mathématique légèrement plus élevé. Pour le niveau d'une maîtrise, citons aussi le livre de Rappaz and Picasso [10], et surtout celui d'Euvrard [3], qui possède une grande sensibilité "mécanique".

Pour ceux qui veulent connaître plus sur la *mécanique des fluides numérique*, un livre en anglais est incontournable : Ferziger and Perić [4]. Citons aussi les "classiques" de Fletcher [5] et Peyret and Taylor [7], et celui de Versteeg and Malalasekera [13] portant sur la méthode des volumes finis et son utilisation dans les codes de calcul commerciaux tels *Fluent*. Pour les gens vraiment accrochés : le récent bouquin de Wesseling [14].

- [1] M. Bergonioux. *Optimisation et Contrôle des Systèmes Linéaires*. Dunod, Paris, 2001.
- [2] P. G. Ciarlet. *Introduction à l'Analyse Numérique Matricielle et à l'Optimisation*. Masson, Paris, 1990.
- [3] D. Euvrard. *Résolution Numérique des Equations aux Dérivées Partielles*. Masson, Paris, 1988.
- [4] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer, Berlin, 1996.
- [5] C. A. J. Fletcher. *Computational Techniques for Fluid Dynamics*. Springer, Berlin, 1991.
- [6] R. Peyret. *Spectral Methods for Incompressible Viscous flow*. Springer, New York, 2002.
- [7] R. Peyret and T. D. Taylor. *Computational Methods for Fluid Flow*. Springer, New York, 1983.
- [8] W. H. Press, B. P. Flannery, S. A. Teukolski, and W. T. Vetterling. *Numerical Recipes*. Cambridge Univ. Press, Cambridge, 1992.
- [9] A. Quarteroni. *Méthodes Numériques pour le Calcul Scientifique*. Springer, Paris, 2000.
- [10] J. Rappaz and M. Picasso. *Introduction à l'Analyse Numérique*. Presses polytechniques et universitaires romandes, Lausanne, 1998.
- [11] L. Sainsaulieu. *Calcul Scientifique*. Masson, Paris, 1996.
- [12] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, Wellesley, MA, 1986.
- [13] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics. The Finite Volume Method*. Longman, Harlow, 1995.
- [14] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer, Berlin, 2001.

⁷ <http://www.library.cornell.edu/nr/bookfpdf.html>