Numerical Techniques in Fluid Dynamics: Session 1

Practicalities and MATLAB framework

Frederik Aerts
Turbulent Flow Simulation and Optimization Group
AY 25-26

1 Outline

1 Practical Arrangements

Basics

- Requirements: fluid mechanics, numerical modeling, MATLAB
- Weekly contact moments on Wednesday (16h05-18h00)
- Primarily self-study with additional theory & explanation on blackboard
- Grading: 70% project work, 30% presentation to peers

Checkpoints

1. Geometrical implementation of normals, tangents, . . . (week 1)
2. 2D steady-state diffusion (week 2)
3. 2D steady-state convection-diffusion (week 3)
4. 2D channel flow with imposed pressure field (week 4-5)
5. SIMPLE pressure correction 2D channel flow (week 6 - . . .)
6. Rhie-Chow interpolation to avoid checkerboarding (week 6 - . . .)
7. 2D lid-driven cavity (. . .)
8. . . .

1 Practical Arrangements

Final report

- Discretization (spatial & temporal)
- Grid convergence
- Order estimation
- Verification of numerical results (analytical or numerical)
- ...
- Be concise and to the point
- Email report as PDF to us and prof. Meyers
- Due date will be specified (around start of January exam period)

1 Practical Arrangements

Contact details

Frederik Aerts
frederik.aerts1@kuleuven.be
Room 02.26

2 Outline

2 Governing equations: Concepts

All (continuum) fluid dynamics based on fundamental conservation laws.

- Conservation of mass - continuity equation
- Conservation of momentum (Newton's second law) - momentum equation
- Conservation of energy - energy equation

2 Governing equations: Concepts

All (continuum) fluid dynamics based on fundamental conservation laws.

- Conservation of mass - continuity equation
- Conservation of momentum (Newton's second law) - momentum equation
- Conservation of energy - energy equation

These statements are cast into a mathematical PDE model under some basic assumptions. In this course we assume that

- The fluid is a continuum
- The fluid is Newtonian
- The flow is incompressible
- The flow is two-dimensional
- The influence of gravity is negligible
- ...

2 Governing equations: 2D Navier-Stokes equations

Conservation of mass (continuity)

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{1}$$

2 Governing equations: 2D Navier-Stokes equations

Conservation of mass (continuity)

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{1}$$

Conservation of momentum

$$\frac{\partial u}{\partial t} + \frac{\partial uu}{\partial x} + \frac{\partial uv}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{\mathsf{Re}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{2}$$

$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial vv}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{\mathsf{Re}}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{3}$$

3 equations in 3 unknowns (u, v, p) ...

2 Governing equations: 2D Navier-Stokes equations

Conservation of mass (continuity)

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{1}$$

Conservation of momentum

$$\frac{\partial u}{\partial t} + \frac{\partial uu}{\partial x} + \frac{\partial uv}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{\mathsf{Re}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{2}$$

$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial vv}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{\mathsf{Re}}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{3}$$

3 equations in 3 unknowns (u, v, p) ...

In order to solve the PDE's numerically, we apply a spatial (finite difference, finite volume, spectral, ...) and a temporal (explicit Euler, implicit Euler, Runge-Kutta, ...) discretization technique.

2 Governing equations: Finite volume discretization

1. Integrate PDE's over control volume Ω
2. Apply Gauss divergence theorem to individual terms: volume integrals $\rightarrow$ surface integrals
3. Numerical approximations: approximate surface integrals (mid-point integration rule), interpolate cell centered values, ...

$$\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{U}\mathbf{U} = -\boldsymbol{\nabla} p + \frac{1}{\mathsf{Re}} \nabla^2 \mathbf{U} \tag{4}$$

$$\iiint_\Omega \left\{ \frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{U}\mathbf{U} \right\} \mathrm{d}\Omega = \iiint_\Omega \left\{ -\boldsymbol{\nabla} p + \frac{1}{\mathsf{Re}} \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} \mathbf{U} \right\} \mathrm{d}\Omega \tag{5}$$

cfr. blackboard ($\iiint \boldsymbol{\nabla} \cdot \mathbf{f} \mathrm{d}\Omega = \iint_\Gamma \mathbf{f} \cdot \mathbf{n} \mathrm{d}s$ with $\mathbf{f} = \boldsymbol{\nabla} \varphi$)

2 Governing equations: Discretized equations

We end up with a system of linear algebraic equations of the form:

$$a_P \varphi_P + \sum_{nb} a_{nb} \varphi_{nb} = b_P \qquad \forall P \tag{6}$$

$$\mathrm{A} \boldsymbol{\varphi} = \mathbf{b} \tag{7}$$

$$\boldsymbol{\varphi}, \mathbf{b} \in \mathbb{R}^{N_x N_y}, \quad \mathrm{A} \in \mathbb{R}^{N_x N_y \times N_x N_y}$$

2 Governing equations: Discretized equations

We end up with a system of linear algebraic equations of the form:

$$a_P \varphi_P + \sum_{nb} a_{nb} \varphi_{nb} = b_P \qquad \forall P \tag{6}$$

$$\mathrm{A} \boldsymbol{\varphi} = \mathbf{b} \tag{7}$$

$$\boldsymbol{\varphi}, \mathbf{b} \in \mathbb{R}^{N_x N_y}, \quad \mathrm{A} \in \mathbb{R}^{N_x N_y \times N_x N_y}$$

- main task is to fill up (sparse) A matrix

2 Governing equations: Discretized equations

We end up with a system of linear algebraic equations of the form:

$$a_P \varphi_P + \sum_{nb} a_{nb} \varphi_{nb} = b_P \qquad \forall P \tag{6}$$

$$\mathrm{A}\boldsymbol{\varphi} = \mathbf{b} \tag{7}$$

$$\boldsymbol{\varphi}, \mathbf{b} \in \mathbb{R}^{N_x N_y}, \quad \mathrm{A} \in \mathbb{R}^{N_x N_y \times N_x N_y}$$

- main task is to fill up (sparse) A matrix
- apply boundary conditions and solve the linear system for the unknown vector $\boldsymbol{\varphi}$

3 Outline

1. Practical Arrangements
2. Navier-Stokes discretization
3. FVMLab
4. Implementation of geometry

3 FVMLab

Framework skeleton provided on Toledo

- Laminar flow solver
- Finite volume discretization
- Co-located unstructured grid

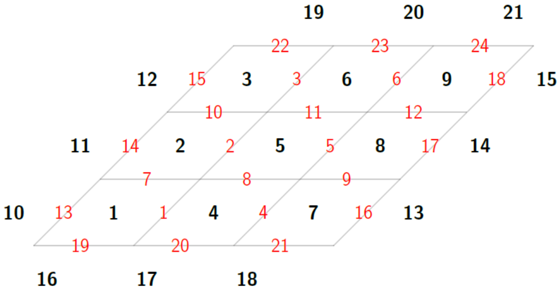
Code structure

- launchfvframework4students.m sets MATLAB path
- src folder
  - contains all source files
  - you don't need to look at all of them
- work folder
  - contains your simulation cases
  - initializes mesh, fields, ...
  - calls the solver

Tip: use CTRL+D to navigate through the code!

3 FVMLab: Unstructured grid

Simulation grid with cell and face numbers



→ Today: implement basic geometric properties grid

3 FVMLab: Unstructured grid

Finite volume methods use fluxes over volume faces, therefore a lot of what we will do is based on the volume faces rather than the volume cells.

Several structures and arrays available, generated by the `FvDomain` function (check it out).

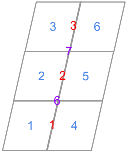
- fNbC: For every face, the neighboring cells
- fNbV: For every face, the neighboring vertices
- cNbF: For every cell, the neighboring faces
- ...

cfr. next slide

3 FVMLab: Unstructured grid

$$\mathrm{fNbX} = \begin{bmatrix} f_1Nb_1 \\ f_1Nb_2 \\ f_2Nb_1 \\ f_2Nb_2 \\ \vdots \end{bmatrix}$$

Example:



$$\mathrm{fNbC} = \begin{bmatrix} 1 \\ 4 \\ 2 \\ 5 \\ \vdots \end{bmatrix} \qquad \mathrm{fNbV} = \begin{bmatrix} \dots \\ \dots \\ 6 \\ 7 \\ \vdots \end{bmatrix}$$

$$\mathrm{vCoord} = \begin{bmatrix} x_{v1} & x_{v2} & \dots & x_{v6} & \dots \\ y_{v1} & y_{v2} & \dots & y_{v6} & \dots \end{bmatrix} \qquad \mathrm{fCentr} = \begin{bmatrix} x_{f1} & x_{f2} & \dots \\ y_{f1} & y_{f2} & \dots \end{bmatrix}$$

Conclusion:

$$\text{face } i \Rightarrow \begin{cases} f_i NbX_1 \text{ at fNbX}[2i-1] \\ f_i NbX_2 \text{ at fNbX}[2i] \end{cases} \quad \text{and} \quad \begin{cases} x_f \text{ at fCentr}[i] \\ y_f \text{ at fCentr}[i] \end{cases}$$

3 FVMLab: Demo

1. Run `launchfvframework4students.m`
2. Try and run the code by running runexamplecase.m
3. MATLAB will throw an error: `FVMLab:ImplementationNotStarted`
4. Go to function `calc_fCentr` either by
   - clicking on hyperref error message
   - using CTRL+D within code
   - going to `src/domain/num/calc_fCentr`
5. Interpret inputs and outputs
   - meaning?
   - type?
   - use documentation in `FvDomain`

4 Outline

1. Practical Arrangements
2. Navier-Stokes discretization
3. FVMLab
4. Implementation of geometry

KU LEUVEN

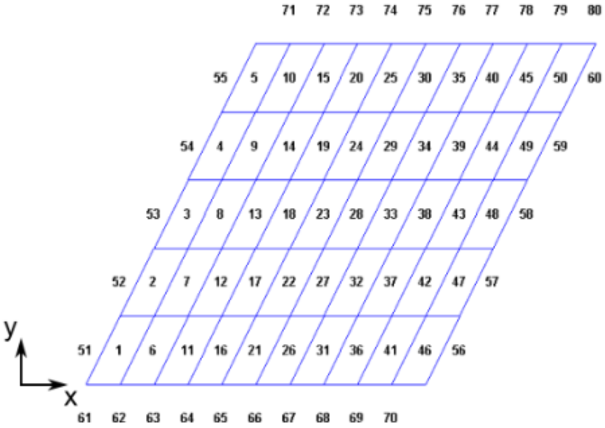
4 Implementation of geometry: Steps

1. Remainder of the session, complete these functions used in FvDomain:
   - `calc_fCentr`: calculates coordinates of face centers
   - `calc_fArea` : calculates length of face
   - `calc_fNormal` : calculates unit length normal vector $\mathbf{n}$ on face (direction $\mathbf{n}$: low cell number to high cell number!)
   - `calc_fTangent` : calculate unit length tangent vector $\mathbf{t}$ on face such that $\mathbf{t} \times \mathbf{n} \geq 0$
   - `calc_fXi` : calculate vector $\boldsymbol{\xi}$ between adjacent cell centers such that $\boldsymbol{\xi} \cdot \mathbf{n} \geq 0$
2. Check results (demo)
3. Homework: finish this

Next week: diffusion of a passive scalar with Dirichlet/Neumann boundary conditions (cfr. heat diffusion and Fourier's law)

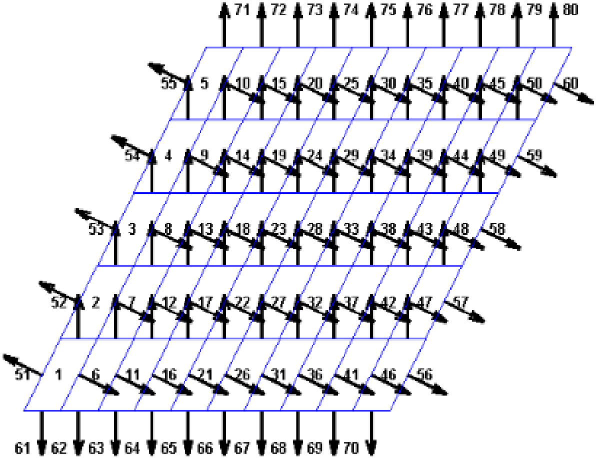
4 Implementation of geometry: Results

Mesh



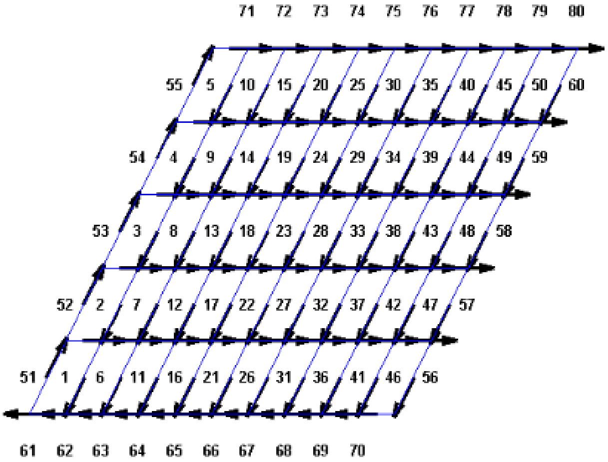
4 Implementation of geometry: Results

Normals



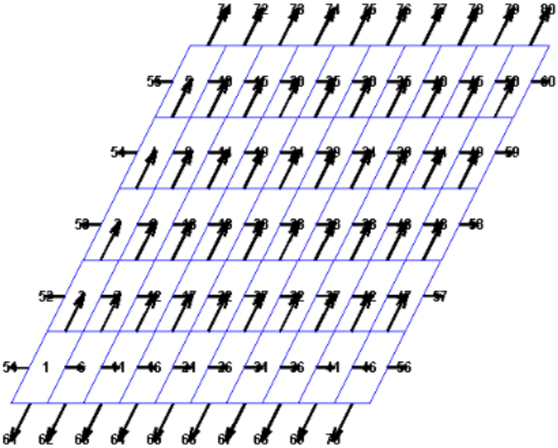
4 Implementation of geometry: Results

Tangents



4 Implementation of geometry: Results

ξ vectors (length incorrectly plotted)



4 Implementation of geometry: Plotting

Example code:

```
normal = Field(casedef.dom.allFaces,1);

set(normal,[casedef.dom.fNormal]);

figure; hold on; axis off; axis equal; colormap(jet(50));
scale = 'lin'; lw = 2;
fvmplotvectorfield(normal,lw);
fvmplotmesh(casedef.dom,1);
fvmplotcellnumbers(casedef.dom,8);
```