Scientific Computing I

Module 10: Case Study – Computational Fluid Dynamics

Michael Bader

Lehrstuhl Informatik V

Winter 2007/2008

Outline

Fluid mechanics as a Discipline

Prominent discipline of application for numerical simulations:

- *experimental* fluid mechanics: wind tunnel studies, laser Doppler anemometry, hot wire techniques, ...
- *theoretical* fluid mechanics: investigations concerning the derivation of turbulence models, e.g.
- *computational* fluid mechanics (CFD): numerical simulations

Fluid mechanics – Fields of Applications

Many fields of application:

- aerodynamics: aircraft design, car design,. . .
- thermodynamics: heating, cooling,. . .
- process engineering: combustion
- material science: crystal growth
- astrophysics: accretion disks

Fluids and Flows

- *ideal* or *real* fluids
  - ideal: no resistance to tangential forces
- *compressible* or *incompressible* fluids
  - think of pressing gases and liquids
- *viscous* or *inviscid* fluids
  - think of the different characteristics of honey and water
- *Newtonian* and *non-Newtonian* fluids
  - the latter may show some elastic behaviour (e.g. in liquids with particles like blood)
- *laminar* or *turbulent* flows
  - turbulence: unsteady, 3D, high vorticity, vortices of different scales, high transport of energy between scales

The Mathematical Model

- typically: all require different models
- here: real, incompressible, viscous, Newtonian, laminar
- starting point: continuum mechanics
- basic conservation laws (remember the heat equation): conservation of *mass* and *momentum*

The Mathematical Model (2)

- with the transport theorem and Newton's second law, we get
  - mass conservation/continuity equation:

    $\frac{\partial}{\partial t}\rho + \mathrm{div}(\rho\vec{u}) = 0$
  - momentum conservation/momentum equations

    $\frac{\partial}{\partial t}(\rho\vec{u}) + (\vec{u}\cdot\mathrm{grad})(\rho\vec{u}) + (\rho\vec{u})\mathrm{div}\,\vec{u} - \rho\vec{g} - \mathrm{div}\,\sigma = 0$
- with the following quantities:
  - $\vec{u} = (u, v, w)$ three-dimensional velocity,
  - ρ density,
  - $\vec{g}$ gravity,
  - σ tension tensor,
  - $\mathrm{div}(\vec{u}) = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$,
  - $\mathrm{grad}\, p = \left(\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}, \frac{\partial p}{\partial z}\right)$.

The Mathematical Model (2)

What to do with the tensor σ?

- viscous case: not diagonal due to friction forces
- Newtonian case: isotrope, Stokes' postulate
- hence: pressure p and viscosity ν appear
- $\mathrm{div}\,\sigma \quad \rightarrow \quad \mathrm{grad}\,p - \nu\Delta\vec{u}$

Incompressible case: density is constant

$$\frac{\partial}{\partial t}\rho + \mathrm{div}(\rho\vec{u}) = 0 \qquad \rightarrow \quad \mathrm{div}(\vec{u}) = 0$$

$$\Rightarrow \quad (\rho\vec{u})\mathrm{div}(\vec{u}) = 0$$

The Navier Stokes Equations

- introducing the **Reynolds number** *Re* (dimensionless, essentially reciprocal of viscosity and some scaling), we finally get the famous **Navier-Stokes** equations:

$$
\begin{aligned}
\frac{\partial}{\partial t}\vec{u} + (\vec{u} \cdot \operatorname{grad})\vec{u} + \operatorname{grad} p &= \frac{1}{\mathrm{Re}}\Delta\vec{u} + \vec{g} \\
\operatorname{div}\vec{u} &= 0
\end{aligned}
$$

- two coupled PDE, nonlinear
- involving velocity and pressure, 1. and 2. spatial derivatives

Boundary Conditions

- **no-slip:**
  The fluid can not penetrate the wall and sticks to it

$$\vec{u} = 0.$$

- **free-slip:**
  The fluid can not penetrate the wall but does not stick to it

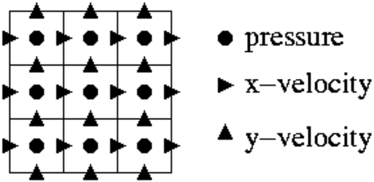
$$u_{\vec{n}} = 0, \frac{\partial \vec{u}_{|||}}{\partial \vec{n}} = 0.$$

Boundary Conditions (2)

- **inflow:**
  Both tangential and normal velocitiy components are prescribed
  $$\vec{u} = \vec{u}_{\text{inflow}}.$$
- **outflow:**
  All velocity components do not change in normal direction
  $$\frac{\partial \vec{u}}{\partial \vec{n}} = 0.$$
- **periodic:**
  Same velocity and pressure at inlet and outlet
  $$\vec{u}_{\text{in}} = \vec{u}_{\text{out}}.$$

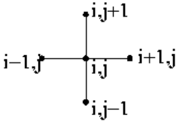
Numerical Treatment – Spatial Derivatives

- discretization scheme: **Finite Differences**
- can be shown to be equivalent to Finite Volumes, here
- grid:
  - strictly orthogonal, *cartesian*
  - staggered grid



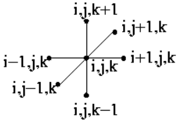
Numerical Treatment – Spatial Derivatives

- Laplacian $\Delta \vec{u}$: standard 5- or 7-point stencil
- 2D:



$$\Delta u(\vec{x}_{i,j}) \approx \frac{u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1}}{h^2}.$$

- 3D:

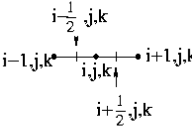


$$\Delta u(\vec{x}_{i,j,k}) \approx \frac{u_{i-1,j,k} + u_{i,j-1,k} + u_{i,j,k-1} - 6u_{i,j,k} + u_{i+1,j,k} + u_{i,j+1,k} + u_{i,j,k+1}}{h^2}.$$

Finite Differences (continued):

- first derivatives $\operatorname{grad} p, \operatorname{div} \vec{u}$: central differences

$$\frac{\partial p}{\partial x_1}(x_{i,j,k}) \approx \frac{p_{i+\frac{1}{2},j,k} - p_{i-\frac{1}{2},j,k}}{h}$$



- derivatives of nonlinear terms $(\vec{u} \cdot \operatorname{grad})\vec{u}$: mixture of central derivatives and upwind derivatives (one-sided derivatives, Donor-Cell-scheme)

Time Discretisation

- explicit Euler scheme:

$$\vec{u}^{(n+1)} = \vec{u}^{(n)} + dt\Big(-\operatorname{grad} p + \frac{1}{Re}\Delta\vec{u}^{(n)} - \left(\vec{u}^{(n)}\cdot\operatorname{grad}\right)\vec{u}^{(n)} + \vec{g}\Big)$$

- Chorin's projection method;

$$\vec{u}^{(n+\frac{1}{2})} = \vec{u}^{(n)} + dt\cdot\left(\frac{1}{Re}\Delta\vec{u}^{(n)} - \left(\vec{u}^{(n)}\cdot\operatorname{grad}\right)\vec{u}^{(n)} + \vec{g}\right),$$

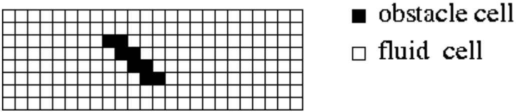
$$\Delta p = \frac{1}{dt}\cdot\operatorname{div}\vec{u}^{(n+\frac{1}{2})},$$

$$\vec{u}^{(n+1)} = \vec{u}^{(n+\frac{1}{2})} - dt\cdot\operatorname{grad} p.$$

- leads to a Poisson equation for the pressure (system of linear equations)

Implementation

- geometry representation as a flag field *(Marker-and-Cell)*



flag field as an array of booleans:

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

- input data (boundary conditions) and output data (computed results) as arrays

Implementation (2)

As in the CFD lab by SCCS:

- modular C-code
- parallelization:
  - simple data parallelism, domain decomposition
  - straightforward MPI-based parallelization
- target architectures:
  - (real) parallel computers
  - clusters
- possible extensions:
  - free-surface flows (“the falling drop”)
  - simple multigrid solvers
  - heat transfer or turbulence models