Case Study: CFD

- fluid mechanics as a 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
- many fields of application:
  - aerodynamics: aircraft design, car design, ...
  - thermodynamics: heating, cooling, ...
  - process engineering: combustion
  - material science: crystal growth
  - astrophysics: accretion disks

Some Small Part of the World ...

- 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 schon 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
- typically: all require different models

The Mathematical Model

- starting point: continuum mechanics
  - basic conservation laws (remember heat conduction in the modelling section): conservation of *mass* and *momentum*
  - with the transport theorem and Newton's second law, we get

$$\frac{\partial}{\partial t}\rho + \mathrm{div}(\rho\vec{u}) = 0 \qquad \text{continuity equation}$$

$$\frac{\partial}{\partial t}(\rho\vec{u}) + (\vec{u}\cdot\mathrm{grad})(\rho\vec{u}) + (\rho\vec{u})\mathrm{div}\vec{u} - \rho\vec{g} - \mathrm{div}\sigma = 0 \qquad \text{momentum equation}$$

  - above quantities:

$\vec{u} = (u, v, w)$ 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
- incompressible case: density is constant
- introducing Reynolds number *Re* (dimensionless, essentially reciprocal of viscosity and some scaling), we finally get the famous Navier-Stokes equations:

$$\frac{\partial}{\partial t}\vec{u} + (\vec{u}\cdot\mathrm{grad})\vec{u} + \mathrm{grad}p = \frac{1}{\mathrm{Re}}\Delta\vec{u} + \vec{g}$$

$$\mathrm{div}\vec{u} = 0$$

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

The Mathematical Model 3

- what about boundary conditions?
  - no-slip:
    - The fluid can not penetrate the wall and sticks to it (tangential and normal velocity components are 0).
  - free-slip:
    - The fluid can not penetrate the wall but does not stick to it (the normal velocity component and the tangential component's normal derivative are 0).
  - inflow:
    - Both tangential an normal velocitiy components are prescribed (defined inflow).
  - outflow:
    - Both velocity components do not change in normal direction (free outlet without restrictions).
  - periodic:
    - Same velocity and pressure at inlet and outlet.

The Numerical Treatment

- discretization scheme: Finite Differences (can be shown to be equivalent to Finite Volumes, here)
  - grid:
    - strictly orthogonal
    - staggered grid
  - spatial derivatives:
    - Laplacian: standard 5- or 7-point stencil
    - first derivatives: mixture of central differences and upwind
    - derivatives of nonlinear terms: as first derivatives
  - time discretization:
    - explicit Euler scheme (simple, but stability restrictions)
  - coupling of equations:
    - Chorin's projection method; leads to a Poisson equation for the pressure
  - solution of SLE:
    - SOR

The Implementation

- geometry representation as a flag field (cf. *Marker-and-Cell*)
- input data (boundary conditions) and output data (computed results) as arrays
- modular C-code
- parallelization:
  - simple data parallelism, domain decomposition
  - straightforward MPI-based parallelization
- target architectures:
  - (real) parallel computers
  - clusters (NOW)
- The rest is (some ☺) programming!

The Visualization

- techniques: all the stuff discussed before:
  - isosurfaces
  - orthoslices
  - streamlines
  - streaklines
  - particle tracing
  - ...
- finally some examples for visualized flows:

Ein paar NAST-Bildchen!

Outlook

- Studying this simulation cycle for CFD as presented very shortly here will be the topic of next semester's practical *Scientific Computing and Visualization* .
- There, you will develop your own simple simulation code and run your own fluid flow simulations, including some pretty pictures and movies to see what you've done.
- In addition to that, you will meet again many of the topics discussed in this introductory course during the CSE program – in more detail, and related to other topics or applications.
- For now, that's all ☺ - see you!