Scientific Computing I

Module 10: Case Study – Computational Fluid Dynamics

Michael Bader

Winter 2012/2013
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

Many fields of application:

- aerodynamics: aircraft design, car design, ...
- thermodynamics: heating, cooling, ...
- process engineering: combustion
- material science: crystal growth
- astrophysics: accretion disks
- geophysics: mantle convection, climate/weather prediction, tsunami simulation, ...
Part I: Modelling

Mathematical Models for CFD

Advection and Diffusion
  Advection Equation
  Advection-Diffusion Equation

Euler Equations
  1D Euler Equations
  Conservation Laws in Higher Dimensions
  2D Euler Equations

Navier-Stokes Equations
  Conservation and Convection Form
  Incompressible Equations
  Viscous Forces

Boundary Conditions
Fluids and Flows

- *ideal* or *real* fluids
  - “ideal”: no resistance to tangential forces
- *compressible* or *incompressible* fluids
  - volume change of gases (vs. liquids?) under pressure
- *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
Mathematical Models for CFD

- typically: all require different models
- our focus here: incompressible, viscous, Newtonian, laminar
  → incompressible Navier-Stokes Equations
  → Shallow Water Equations
- starting point: continuum mechanics
  → macroscopic properties (pressure, density, velocity field)
  → compared to stochastic or micro-/mesoscopic approaches (lattice Boltzman method, e.g.)
- relies on basic conservation laws (remember the heat equation): conservation of mass and momentum (and energy)
- additionally: slight focus on Finite Volume Methods
Conservation of some quantity $q$ in a fluid domain $\Omega = [a, b]$ with given velocity $v(x, t)$:

- total amount/mass of $q$ in $\Omega = [a, b]$ is given by $\int_a^b q(x, t) \, dx$
- change of mass can only happen due to in-/outflow at $a$ and $b$:
  \[
  \frac{\partial}{\partial t} \int_a^b q(x, t) \, dx = F(a, t) - F(b, t) = -F(x, t)|_a^b = -\int_a^b \frac{\partial}{\partial x} F(x, t) \, dx
  \]

- note: $F(a, t)$ and $-F(b, t)$ denote an inflow into the domain $\Omega$
Consider

- flux function $F(x, t)$ depends on velocity $v(x, t)$, density $q(x, t)$ and the pipe’s cross-sectional area $A(x)$:

$$F(x, t) = A(x)v(x, t)q(x, t)$$

- for simplicity, we set $A(x) = 1$, and obtain:

$$\frac{\partial}{\partial t} \int_{a}^{b} q(x, t) \, dx = -\int_{a}^{b} \frac{\partial}{\partial x} F(x, t) \, dx = -\int_{a}^{b} \frac{\partial}{\partial x} (v(x, t)q(x, t)) \, dx$$
Advection Equation (3)

**Advection Equation:**
- for smooth functions, we may write:

\[
\int_a^b \frac{\partial}{\partial t} q(x, t) \, dx = \frac{\partial}{\partial t} \int_a^b q(x, t) \, dx = - \int_a^b \frac{\partial}{\partial x} (v(x, t)q(x, t)) \, dx
\]

- as this equation has to hold for any $\Omega = [a, b]$, we demand:

\[
\frac{\partial}{\partial t} q(x, t) = -\frac{\partial}{\partial x} (v(x, t)q(x, t)) \quad \text{or short:} \quad q_t + (vq)_x = 0
\]
Advection and Diffusion

Diffusion

- even in a fluid at rest, an inhomogeneous density \( q(x, t) \) will slowly change towards a uniform density \( q_0 \) due to molecular processes → **diffusion**
- **Fick’s law of diffusion**: resulting flux is prop. to gradient of \( q \)
  \[
  -F_{\text{diff}} = \beta q_x
  \]
- to model both advection and diffusion, we have
  \[
  -F = -vq + \beta q_x, \text{ and thus}
  \]
  \[
  q_t + (vq)_x = \beta q_{xx}
  \]
- special case \( q_t = 0 \) ↩ **“advection-diffusion equation”**:
  \[
  -\beta q_{xx} + (vq)_x = 0
  \]
1D Euler Equations

• with our quantity $q$ being the mass density $\rho$, we obtain an equation for the conservation of mass:

$$\rho_t + (\nu \rho)_x = 0$$

• another conservation property is that of momentum $\rho \nu$; here, the flux term includes the pressure $p$:

$$F_{\text{mom}} = \rho \nu^2 + p$$

• thus, we obtain as equation for the conservation of momentum:

$$(\rho \nu)_t + (\rho \nu^2 + p)_x = 0$$

• we obtain a system of two PDEs, the 1D Euler Equations

• to close the system, we need a relation between $\rho$ and $p$ (using the ideal gas law, e.g.)

• we might add an equation for temperature (derived from the conservation of internal energy)
Conservation Laws in Higher Dimensions

• in 2D, a conservation law for quantity $q$ takes the form:

$$q_t + F(q)_x + G(q)_y = 0$$

• or similar in 3D:

$$q_t + F(q)_x + G(q)_y + H(q)_z = 0$$

• for advection, the flux functions are

$$F(q) = uq \quad G(q) = vq \quad H(q) = wq$$

where $u, v, w$ are the velocity components in the three space dimensions $x, y, z$

• hence, for 2D we obtain a conservation law such as

$$q_t + (uq)_x + (vq)_y = 0$$
2D Euler Equations

- in 2D, with velocity components \( u(x, y, t) \) and \( v(x, y, t) \) the equation for conservation of mass reads:

\[
\rho_t + (\rho u)_x + (\rho v)_y = 0
\]

- similar, the two equation for conservation of momentum are:

\[
(\rho u)_t + (\rho u^2 + p)_x + (\rho uv)_y = 0
\]
\[
(\rho v)_t + (\rho uv)_x + (\rho v^2 + p)_y = 0
\]

- again, we assume constant temperature, and we need a relation between \( \rho \) and \( p \) to close the system

- the Euler equations model an inviscid (ideal) fluid

- we also neglect additional source terms, such as for gravity forces, etc.
Navier-Stokes Equations

- mass conservation/continuity equation is the same as for the Euler equations:

\[ \rho_t + (\rho u)_x + (\rho v)_y + (\rho w)_z = 0 \]

or, written in vector notation:

\[ \frac{\partial}{\partial t} \rho + \nabla \cdot (\rho \vec{u}) = 0, \quad \nabla \cdot \vec{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \]

- momentum conservation/momentum equations

\[ \frac{\partial}{\partial t} (\rho \vec{u}) + \nabla \cdot (\vec{u} \otimes \rho \vec{u}) - \nabla \sigma - f = 0 \]

- with \( \sigma \) being the stress tensor, which includes the pressure \( p \) and viscous forces: \( \sigma = -pl + \ldots \)

- \( f \) models external (volume) forces (gravity, e.g.)
Navier-Stokes Equations
Conservation and Convection Form

- the equations for mass and momentum, on the previous slide, are given in the so-called **conservation form**
- with the equations
  \[ \nabla \cdot (\rho \vec{u}) = \vec{u} \cdot \nabla \rho + \rho \nabla \cdot \vec{u} \quad \text{and} \quad \nabla \cdot (\rho \vec{u} \otimes \vec{u}) = \vec{u} \left( \nabla \cdot (\rho \vec{u}) \right) + (\rho \vec{u} \cdot \nabla) \vec{u}, \]
we obtain:
  \[ \frac{\partial}{\partial t} \rho + \vec{u} \cdot \nabla \rho + \rho \nabla \cdot \vec{u} = 0 \]
  \[ \frac{\partial}{\partial t} (\rho \vec{u}) + \vec{u} \left( \nabla \cdot (\rho \vec{u}) \right) + (\rho \vec{u} \cdot \nabla) \vec{u} - \nabla \sigma - f = 0 \]
- with \( \frac{\partial}{\partial t} (\rho \vec{u}) = \rho \frac{\partial}{\partial t} \vec{u} + \vec{u} \frac{\partial}{\partial t} \rho \)
and applying \( \vec{u} \frac{\partial}{\partial t} \rho + \vec{u} \left( \nabla \cdot (\rho \vec{u}) \right) = \vec{u} \left( \frac{\partial}{\partial t} \rho + \nabla \cdot (\rho \vec{u}) \right) = 0, \)
we obtain for the momentum equation in **convection form**
  \[ \rho \left( \frac{\partial}{\partial t} \vec{u} + (\vec{u} \cdot \nabla) \vec{u} \right) - \nabla \sigma - f = 0 \]
Navier-Stokes Equations

Incompressible Equations

• in the convective forms

\[
\begin{align*}
\frac{\partial}{\partial t} \rho + \vec{u} \cdot \nabla \rho + \rho \nabla \cdot \vec{u} &= 0 \\
\rho \left( \frac{\partial}{\partial t} \vec{u} + (\vec{u} \cdot \nabla) \vec{u} \right) - \nabla \sigma - \vec{f} &= 0
\end{align*}
\]

we assume that the density \( \rho \) is constant: \( \frac{\partial}{\partial t} \rho = 0, \nabla \rho = 0 \)

• we obtain the incompressible Navier-Stokes equations:

\[
\begin{align*}
\nabla \cdot \vec{u} &= 0 \\
\rho \left( \frac{\partial}{\partial t} \vec{u} + (\vec{u} \cdot \nabla) \vec{u} \right) - \nabla \sigma - \vec{f} &= 0
\end{align*}
\]

• “incompressible”: the density does not change due to pressure or temperature, e.g.
Viscous Forces

Open question: stress tensor $\sigma$

- $\sigma$ includes pressure $p$ and viscosity tensor $\tau$: $\sigma = -pl + \tau$
- *Newtonian* fluids: viscous stresses proportional to the strain rate (first derivatives)
- isotropic, *incompressible* fluids, Stokes assumption (no volume viscosity), then $\nabla \sigma = -\nabla p + \mu \Delta \vec{u}$
- $\mu$ the dynamic viscosity

**Incompressible Navier-Stokes equations:**

\[
\nabla \cdot \vec{u} = 0
\]

\[
\rho \left( \frac{\partial}{\partial t} \vec{u} + (\vec{u} \cdot \nabla) \vec{u} \right) = -\nabla p + \mu \Delta \vec{u} + f
\]
Dynamic Similarity of Flows
Dimensionless Form of the Navier-Stokes Equations

- we scale our unknowns to typical length scale $L$ and velocity $u_\infty$:
  
  $\begin{align*}
  x &\rightarrow \frac{x}{L} \\
  t &\rightarrow \frac{u_\infty t}{L} \\
  u &\rightarrow \frac{u}{u_\infty} \\
  p &\rightarrow \frac{p - p_\infty}{\rho u_\infty^2}
  \end{align*}$

- and obtain the dimensionless form of the Navier-Stokes equations:

  \begin{align*}
  \nabla \cdot \vec{u} &= 0 \\
  \frac{\partial}{\partial t} \vec{u} + (\vec{u} \cdot \nabla) \vec{u} &= -\nabla p + \frac{1}{\text{Re}} \Delta \vec{u} + f
  \end{align*}

  introducing the **Reynolds number** $\text{Re} := \frac{\mu}{\rho u_\infty L}$

- important corollary: flows with the same Reynolds number will show the same behaviour
Boundary Conditions (here only velocity)

- **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
  \[ u_n = 0, \quad \frac{\partial \vec{u}}{\partial n} = 0. \]

- **inflow**: both tangential and normal velocity components are prescribed
  \[ \vec{u} = \vec{u}_{\text{inflow}}. \]

- **outflow**: should be “do nothing”; simple option: all velocity components do not change in normal direction
  \[ \frac{\partial \vec{u}}{\partial n} = 0. \]

- **periodic**: same velocity and pressure at inlet and outlet
  \[ \vec{u}_{\text{in}} = \vec{u}_{\text{out}}. \]
Part II: A Finite Difference/Volume Method for the Incompressible Navier-Stokes Equations

Numerical Treatment – Spatial Derivatives
- Finite Volume Discretisation and Upwind Flux
- Marker-and-Cell Method, Staggered Grid
- Discretization of Continuity Equation
- Discretization of Momentum Equation

Time Discretization
- Chorin Projection

Implementation
Finite Volume Discretisation – Advection-Diffusion Equation

- compute tracer concentration $q$ with diffusion $\beta$ and convection $v$:
  $$-\beta q_{xx} + (vq)_x = 0 \quad \text{on } \Omega = (0, 1)$$
  with boundary conditions $q(0) = 1$ and $q(1) = 0$.
- equidistant grid points $x_i = ih$, grid cells $[x_i, x_{i+1}]$
- back to representation via conservation law (for one grid cell):
  $$\int_{x_i}^{x_{i+1}} \frac{\partial}{\partial x} F(x) \, dx = F(x) \bigg|_{x_{i+1}}^{x_i} = 0$$
  with $F(x) = F(q(x)) = -\beta q_x(x) + vq(x)$.
- we need to compute the flux $F$ at the boundaries of the grid cells; however, assume $q(x)$ piecewise constant within the grid cells
Finite Volume Discretisation – Advection-Diffusion Equation (2)

• wanted: compute $F(x_i)$ with $F(q(x)) = -\beta q_x(x) + vq(x)$
• where $q(x) := q_i$ for each $\Omega_i = [x_i, x_{i+1}]$
• computing the diffusive flux is straightforward:

\[-\beta q_x \bigg|_{x_{i+1}} = -\beta \frac{q(x_{i+1}) - q(x_i)}{h}\]

• options for advective flux $vq$:
  • symmetric flux:
    \[ vq \bigg|_{x_{i+1}} = \frac{vq(x_i) + vq(x_{i+1})}{2} \]
  • “upwind” flux:
    \[ vq \bigg|_{x_{i+1}} = \begin{cases} vq(x_i) & \text{if } v > 0 \\ vq(x_{i+1}) & \text{if } v < 0 \end{cases} \]
Finite Volume Discretisation – Advection-Diffusion Equation (3)

- system of equations: for all $i$
  \[ F(x) \bigg|_{x_i}^{x_{i+1}} = F(x_{i+1}) - F(x_i) = 0 \]

- for symmetric flux:
  \[ -\beta \frac{q(x_{i+1}) - 2q(x_i) + q(x_{i-1})}{h^2} + v \frac{q(x_{i+1}) - q(x_{i-1})}{2h} = 0 \]

leads to non-physical behaviour as soon as $\beta < \frac{vh}{2}$
(observe signs of matrix elements!)

- system of equations for upwind flux (assume $v > 0$):
  \[ -\beta \frac{q(x_{i+1}) - 2q(x_i) + q(x_{i-1})}{h^2} + \frac{v}{h} \frac{q(x_i) - q(x_{i-1})}{h} = 0 \]

→ stable, but overly diffusive solutions (positive definite matrix)
Marker-and-Cell method (Harlow and Welch, 1965):

• discretization scheme: Finite Differences
• can be shown to be equivalent to Finite Volumes, however
• based on a so-called **staggered grid**:
  • *Cartesian* grid (rectangular grid cells), with cell centres at $x_{i,j} := (ih, jh)$, e.g.
  • pressure located in cell centres
  • velocities (those in normal direction) located on cell edges
Spatial Discretisation – Continuity Equation:

- mass conservation: discretise $\nabla \cdot \vec{u}$
  $\rightarrow$ evaluate derivative at cell centres, allows central derivatives:

\[
(\nabla \cdot \vec{u})\big|_{i,j} = \left. \frac{\partial u}{\partial x} \right|_{i,j} + \left. \frac{\partial v}{\partial y} \right|_{i,j} \approx \frac{u_{i,j} - u_{i-1,j}}{h} + \frac{v_{i,j} - v_{i,j-1}}{h}
\]

remember: $u_{i,j}$ and $v_{i,j}$ located on cell edges

- notation: $(\nabla \cdot \vec{u})\big|_{i,j} := (\nabla \cdot \vec{u})\big|_{x_{i,j}}$
  (evaluate expression at cell centre $x_{i,j}$)
Spatial Discretisation – Pressure Terms

- note: velocities located on midpoints of cell edges

\[
\frac{\partial u}{\partial t}\bigg|_{i+\frac{1}{2},j} = \ldots \quad \frac{\partial v}{\partial t}\bigg|_{i,j+\frac{1}{2}} = \ldots
\]

thus, all derivatives need to be approximated at midpoints of cell edges!

- pressure term \( \nabla p \): central differences for first derivatives (as pressure is located in cell centres)

\[
\frac{\partial p}{\partial x}\bigg|_{i+\frac{1}{2},j} \approx \frac{p_{i+1,j} - p_{i,j}}{h} \quad \frac{\partial p}{\partial y}\bigg|_{i,j+\frac{1}{2}} \approx \frac{p_{i,j+1} - p_{i,j}}{h}
\]
Spatial Discretisation – Diffusion Term

• for diffusion term $\Delta \bar{u}$: use standard 5- or 7-point stencil

• 2D:

$$\Delta u_{|i,j} \approx \frac{u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1}}{h^2}$$

• 3D:

$$\Delta u_{|i,j,k} \approx \frac{u_{i-1,j,k} + u_{i,j-1,k} + u_{i,j,k-1} - 6u_{i,j,k} + u_{i+1,j,k} + u_{i,j+1,k} + u_{i,j,k+1}}{h^2}$$
Spatial Discretisation – Convection Terms

- treat derivatives of nonlinear terms \((\vec{u} \cdot \nabla) \vec{u}\):
- central differences (for momentum equation in x-direction):

\[
\left. u \frac{\partial u}{\partial x} \right|_{i+\frac{1}{2},j} \approx u_{i,j} \frac{u_{i+1,j} - u_{i-1,j}}{2h} \quad \left. v \frac{\partial u}{\partial y} \right|_{i+\frac{1}{2},j} \approx v_{x_{i+\frac{1}{2},j}} \frac{u_{i,j+1} - u_{i,j-1}}{2h}
\]

with \(v_{x_{i+\frac{1}{2},j}} = \frac{1}{4} \left( v_{i,j} + v_{i,j-1} + v_{i+1,j} + v_{i+1,j-1} \right)\)

- upwind differences (for momentum equation in x-direction):

\[
\left. u \frac{\partial u}{\partial x} \right|_{x_{i+\frac{1}{2},j}} \approx u_{i,j} \frac{u_{i,j} - u_{i-1,j}}{2h} \quad \left. v \frac{\partial u}{\partial y} \right|_{x_{i+\frac{1}{2},j}} \approx v_{x_{i+\frac{1}{2},j}} \frac{u_{i,j} - u_{i,j-1}}{2h}
\]

if \(u_{i,j} > 0\) and \(v_{x_{i+\frac{1}{2},j}} > 0\)

- code for CFD lab will mix central and upwind differences (and is based on conservation form of convection terms)
Time Discretisation

- recall the incompressible Navier-Stokes equations:

\[ \nabla \cdot \vec{u} = 0 \]

\[ \frac{\partial}{\partial t} \vec{u} + (\vec{u} \cdot \nabla) \vec{u} = -\nabla p + \frac{1}{Re} \Delta \vec{u} + f \]

- note the role of the unknowns:
  - 2 or 3 equations for velocities (x, y, and z component) resulting from momentum conservation
  - 4th equation (mass conservation) to “close” the system; required to determine pressure \( p \)
  - however, \( p \) does not occur explicitly in mass conservation

- possible approach: Chorin’s projection method
  - \( p \) acts as a variable to enforce the mass conservation as “side condition”
Time Discretisation – Chorin Projection

- explicit Euler scheme for momentum equation:
  \[
  \vec{u}^{(n+1)} = \vec{u}^{(n)} + \tau \left( -\nabla p + \frac{1}{Re} \Delta \vec{u}^{(n)} - \left( \vec{u}^{(n)} \cdot \nabla \right) \vec{u}^{(n)} + \vec{g} \right)
  \]

- Chorin projection
  \[
  \vec{u}^{(n+\frac{1}{2})} = \vec{u}^{(n)} + \tau \left( \frac{1}{Re} \Delta \vec{u}^{(n)} - \left( \vec{u}^{(n)} \cdot \nabla \right) \vec{u}^{(n)} + \vec{g} \right),
  \]
  \[
  \vec{u}^{(n+1)} = \vec{u}^{(n+\frac{1}{2})} - \tau \nabla p
  \]

- \( \vec{u}^{(n+1)} \) needs to satisfy mass conservation: \( \nabla \cdot \vec{u}^{(n+1)} = 0 \)
  \[
  \nabla \cdot \left( \vec{u}^{(n+\frac{1}{2})} - \tau \nabla p \right) = 0 \quad \Rightarrow \quad \Delta p = \frac{1}{\tau} \left( \nabla \cdot \vec{u}^{(n+\frac{1}{2})} \right)
  \]
  thus, system of linear equations to be solved in each time step
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 1 1 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 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 0 0 0
```

- input data (boundary conditions) and output data (computed results) as arrays
Implementation (2)

Lab course "Scientific Computing – Computational Fluid Dynamics":

- modular C-code
- parallelization:
  - simple data parallelism, domain decomposition
  - straightforward MPI-based parallelization (exchange of ghost layers)
- target architectures:
  - parallel computers with distributed memory
  - clusters
- possible extensions:
  - free-surface flows ("the falling drop")
  - multigrid solver for the pressure equation
  - heat transfer or turbulence models
Part III: The Shallow Water Equations and Finite Volumes Revisited

The Shallow Water Equations
Modelling Scenario: Tsunami Simulation

Finite Volume Discretisation
Central and Upwind Fluxes
Lax-Friedrichs Flux

Towards Tsunami Simulation
Wave Speed of Tsunamis
Treatment of Bathymetry Data

The SWE Code
Model and Discretisation
The Shallow Water Equations

\[
\frac{\partial}{\partial t} \begin{pmatrix} h \\ hu \\ hv \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} hu^2 + \frac{1}{2}gh^2 \\ hu v \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix} + \frac{\partial}{\partial y} \begin{pmatrix} hv \\ hv u \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix} = S(t, x, y)
\]

Comments on modelling:

- generalized 2D hyperbolic PDE: \( q = (h, hu, hv)^T \)

\[
\frac{\partial}{\partial t} q + \frac{\partial}{\partial x} F(q) + \frac{\partial}{\partial y} G(q) = S(t, x, y)
\]

derived from conservation laws for mass and momentum

- may be derived by vertical averaging from the 3D incompressible Navier-Stokes equations

- compare to Euler equations: density \( \rho \) vs. water depth \( h \)
Modelling Scenario: Tsunami Simulation

The Ocean as “Shallow Water”??
- compare horizontal ($\sim 1000$ km) to vertical ($\sim 5$ km) length scale
- wave lengths large compared to water depth
- vertical flow may be neglected; movement of the “entire water column”
Tsunami Modelling with the Shallow Water equations:

- source term $S(x, y)$ includes bathymetry data (i.e., elevation of ocean floor)
- Coriolis forces, friction, etc., as possible further terms
- boundary conditions are difficult: coastal inundation, outflow at domain boundaries
Finite Volume Discretisation

- discretise system of PDEs

\[
\frac{\partial}{\partial t} q + \frac{\partial}{\partial x} F(q) + \frac{\partial}{\partial y} G(q) = S(t, x, y)
\]

- with

\[
q := \begin{pmatrix} h \\ hu \\ hv \end{pmatrix} \quad F(q) := \begin{pmatrix} hu \\ hu^2 + \frac{1}{2} gh^2 \\ huv \end{pmatrix} \quad G(q) := \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2} gh^2 \end{pmatrix}
\]

- basic form of numerical schemes:

\[
Q_{i,j}^{(n+1)} = Q_{i,j}^{(n)} - \frac{\tau}{h} \left( F_{i+\frac{1}{2},j}^{(n)} - F_{i-\frac{1}{2},j}^{(n)} \right) - \frac{\tau}{h} \left( G_{i,j+\frac{1}{2}}^{(n)} - G_{i,j-\frac{1}{2}}^{(n)} \right)
\]

where \( F_{i+\frac{1}{2},j}^{(n)} \), \( G_{i,j+\frac{1}{2}}^{(n)} \), ... approximate the flux functions \( F(q) \) and \( G(q) \) at the grid cell boundaries.
Central and Upwind Fluxes

- define fluxes $F^{(n)}_{i+\frac{1}{2},j}$, $G^{(n)}_{i,j+\frac{1}{2}}$, ... via 1D numerical flux function $\mathcal{F}$:

$$F^{(n)}_{i+\frac{1}{2}} = \mathcal{F}(Q^{(n)}_i, Q^{(n)}_{i+1}) \quad \text{and} \quad G^{(n)}_{j-\frac{1}{2}} = \mathcal{F}(Q^{(n)}_{j-1}, Q^{(n)}_j)$$

- central flux:

$$F^{(n)}_{i+\frac{1}{2}} = \mathcal{F}(Q^{(n)}_i, Q^{(n)}_{i+1}) := \frac{1}{2} \left( F(Q^{(n)}_i) + F(Q^{(n)}_{i+1}) \right)$$

leads to unstable methods for convective transport

- upwind flux (here, for $h$-equation, $F(h) = hu$):

$$F^{(n)}_{i+\frac{1}{2}} = \mathcal{F}(h^{(n)}_i, h^{(n)}_{i+1}) := \begin{cases} hu_i & \text{if } u_{i+\frac{1}{2}} > 0 \\ hu_{i+1} & \text{if } u_{i+\frac{1}{2}} < 0 \end{cases}$$

stable, but includes artificial diffusion
(Local) Lax-Friedrichs Flux

- classical **Lax-Friedrichs method** uses as numerical flux:

\[
F_{i+\frac{1}{2}}^{(n)} = \mathcal{F}(Q_i^{(n)}, Q_{i+1}^{(n)}) := \frac{1}{2} \left( F(Q_i^{(n)}) + F(Q_{i+1}^{(n)}) \right) - \frac{h}{2\tau} (Q_{i+1}^{(n)} - Q_i^{(n)})
\]

- can be interpreted as central flux plus diffusion flux:

\[
\frac{h}{2\tau} (Q_{i+1}^{(n)} - Q_i^{(n)}) = \frac{h^2}{2\tau} \cdot \frac{Q_{i+1}^{(n)} - Q_i^{(n)}}{h}
\]

with diffusion coefficient \( \frac{h^2}{2\tau} \), where \( c := \frac{h}{\tau} \) is some kind of velocity ("one grid cell per time step")

- idea of **local Lax-Friedrichs** method: use the "appropriate" velocity

\[
F_{i+\frac{1}{2}}^{(n)} := \frac{1}{2} \left( F(Q_i^{(n)}) + F(Q_{i+1}^{(n)}) \right) - \frac{a_{i+\frac{1}{2}}}{2} (Q_{i+1}^{(n)} - Q_i^{(n)})
\]
Wave Speed of Tsunamis

• consider the 1D case

\[
\frac{\partial}{\partial t} \begin{pmatrix} h \\ hu \end{pmatrix} + \frac{\partial}{\partial x} \left( hu^2 + \frac{1}{2} gh^2 \right) = 0
\]

• with \( q = (q_1, q_2)^T := (h, hu)^T \), we obtain

\[
\frac{\partial}{\partial t} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + \frac{\partial}{\partial x} \left( \frac{q_2^2}{q_1} + \frac{1}{2} gq_1^2 \right) = 0
\]

• write in convective form:

\[
\frac{\partial}{\partial t} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + f' \frac{\partial}{\partial x} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = 0
\]

with

\[
f' = \begin{pmatrix} \partial f_1 / \partial q_1 & \partial f_1 / \partial q_2 \\ \partial f_2 / \partial q_1 & \partial f_2 / \partial q_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -q_2^2/q_1^2 + gq_1 & 2q_2/q_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -u^2 + gh & 2u \end{pmatrix}
\]
Wave Speed of Tsunamis (2)

- compute eigenvectors and eigenvalues of \( f' \):
  \[
  \lambda^{1/2} = u \pm \sqrt{gh} \quad r^{1/2} = \left( u \pm \sqrt{gh} \right)
  \]

- and then with \( f' = R \Lambda R^{-1} \), where \( R := (r^1, r^2) \) and \( \Lambda := \text{diag}(\lambda^1, \lambda^2) \), we can diagonalise the PDE:
  \[
  \frac{\partial}{\partial t} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + \Lambda \frac{\partial}{\partial x} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = 0, \quad w = R^{-1}q
  \]

- for small changes in \( h \) and small velocities, we thus obtain that waves are “advected” (i.e., travel) at speed \( \lambda^{1/2} \approx \pm \sqrt{gh} \)

- recall local Lax-Friedrichs method:
  \[
  F_{i+\frac{1}{2}}^{(n)} := \frac{1}{2} \left( F(Q_i^{(n)}) + F(Q_{i+1}^{(n)}) \right) - \frac{a_{i+\frac{1}{2}}}{2} (Q_i^{(n)} - Q_{i-1}^{(n)})
  \]

→ choose \( a_{i+\frac{1}{2}} = \max\{\lambda^k\} \)
Shallow Water Equations with Bathymetry

\[
\begin{aligned}
\frac{\partial}{\partial t} \begin{pmatrix} h \\ hu \\ hv \end{pmatrix} &+ \frac{\partial}{\partial x} \begin{pmatrix} hu^2 + \frac{1}{2}gh^2 \\ hu^2 \end{pmatrix} + \frac{\partial}{\partial y} \begin{pmatrix} hv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix} = \\
&\begin{pmatrix} 0 \\ -(ghb)_x \\ -(ghb)_y \end{pmatrix}
\end{aligned}
\]

Questions for numerics:
- treat \((bh)_x\) and \((bh)_y\) as source terms or include these into flux computations?
- preserve certain properties of solutions – e.g., “lake at rest”
Consider “Lake at Rest” Scenario:

- “at rest”: velocities $u = 0$ and $v = 0$
- examine local Lax-Friedrichs flux in $h$ equation:

$$F_{i+rac{1}{2}}^{(n)} = \frac{1}{2} \left( (hu)_i^{(n)} + (hu)_{i+1}^{(n)} \right) - \frac{a_{i+\frac{1}{2}}}{2} (h_{i+1}^{(n)} - h_i^{(n)}) = 0$$

$$\Rightarrow F_{i+\frac{1}{2}}^{(n)} - F_{i-\frac{1}{2}}^{(n)} = -\frac{a_{i+\frac{1}{2}}}{2} (h_{i+1}^{(n)} - h_i^{(n)}) + \frac{a_{i-\frac{1}{2}}}{2} (h_i^{(n)} - h_{i-1}^{(n)}) = 0$$

- note: $a_{i+\frac{1}{2}} \approx \sqrt{gh}$ and if $b_{i-1} \neq b_i \neq b_{i+1}$ then $h_{i-1} \neq h_i \neq h_{i+1}$
- thus: “lake at rest” not an equilibrium solution for local Lax-Friedrichs flux

Additional problems:

- complicated numerics close to the shore
- in particular: “wetting and drying” (inundation of the coast)
SWE – An Education-Oriented Shallow Water Code

Model & Discretisation

Simplified setting (no friction, no viscosity, no coriolis forces, etc.):

\[
\begin{pmatrix}
h \\
h u \\
h v \\
\end{pmatrix}_t + \begin{pmatrix}
h u \\
h u^2 + \frac{1}{2}gh^2 \\
h v \\
\end{pmatrix}_x + \begin{pmatrix}
h v \\
h u v \\
h v^2 + \frac{1}{2}gh^2 \\
\end{pmatrix}_y = S(t, x, y).
\]

Finite Volume Discretization:

- generalized 2D hyperbolic PDE: \( q = (h, hu, hv)^T \)

\[
\frac{\partial}{\partial t} q + \frac{\partial}{\partial x} F(q) + \frac{\partial}{\partial y} G(q) = S(t, x, y)
\]

- Wave propagation form:

\[
Q_{i,j}^{n+1} = Q_{i,j}^n - \frac{\Delta t}{\Delta x} \left( A^+ \Delta Q_{i-1/2,j} + A^- \Delta Q_{i+1/2,j} \right) \\
- \frac{\Delta t}{\Delta y} \left( B^+ \Delta Q_{i,j-1/2} + B^- \Delta Q_{i,j+1/2} \right).
\]
SWE – An Education-Oriented Shallow Water Code
Model & Discretisation

Simplified setting (no friction, no viscosity, no coriolis forces, etc.):

\[
\begin{pmatrix}
  h \\
  hu \\
  hv
\end{pmatrix}_t + \begin{pmatrix}
  hu \\
  hu^2 + \frac{1}{2}gh^2 \\
  huv
\end{pmatrix}_x + \begin{pmatrix}
  hv \\
  hv^2 + \frac{1}{2}gh^2 \\
  huv
\end{pmatrix}_y = S(t, x, y).
\]

Flux Computation on Edges:

- Wave propagation form:
  \[
  Q_{i,j}^{n+1} = Q_{i,j}^n - \frac{\Delta t}{\Delta x} \left( A^+ \Delta Q_{i-1/2,j}^n + A^- \Delta Q_{i+1/2,j}^n \right) \\
  - \frac{\Delta t}{\Delta y} \left( B^+ \Delta Q_{i,j-1/2}^n + B^- \Delta Q_{i,j+1/2}^n \right).
  \]

- simple fluxes: Rusanov/(local) Lax-Friedrich
- more advanced: f-Wave or (augmented) Riemann solvers (George, 2008; LeVeque, 2011), no limiters
Unknowns and Numerical Fluxes:

- unknowns $h$, $hu$, $hv$, and $b$ located in cell centers
- two sets of “net updates”/numerical fluxes per edge: $A^+ \Delta Q_{i-1/2,j}$, $B^- \Delta Q_{i,j+1/2}$, etc.
SWE – An Education-Oriented Shallow Water Code

Patches of Cartesian Grid Blocks

Spatial Discretization:
- regular Cartesian meshes; allow multiple patches
- ghost and copy layers to implement boundary conditions, for more complicated domains, and for parallelization
Course material is mostly based on:


Shallow Water Code SWE:
→ [http://www5.in.tum.de/SWE/](http://www5.in.tum.de/SWE/)