Simulationstechnik I
“Mechanics”

Marek Behr
Chair for Computational Analysis of Technical Systems
Center for Computational Engineering Science
About CATS

• Established in 10.2004
• Originally: Rechnergestützte Analyse Technischer Systeme
• Now: Computergestützte Analyse Technischer Systeme
• 1 Prof. (me!), 2 non-sci. staff, 7 doctoral and post-doctoral students
• Close collaborations with:
  ★ Rice University
  ★ Baylor College of Medicine
  ★ MicroMed Technology, Inc.
  ★ University of Houston
  ★ Chuo University, Tokyo
CATS and CES

- Center for Computational Engineering Science established 12.2004:
  - 2 Prof. in FB 1 (Dieter Bothe, NN),
  - 1 Prof. in FB 4 (me!)
  - 1 Prof. in FB 5 (Heike Emmerich)
  - 1 Prof. in FB 6 (coming in 2006)
  - Director of the Computing Center (Christian Bischof)
  - Additional associated faculty later on

- Evolution of CES teaching:
  - Mechanics sequence is being replaced with CES-specific content
  - Mathematics sequence will have regular lecturers
  - Software engineering will be CES-specific
Outline

• What is mechanics?
• Why computation?
• Simulation basics
• Example application: civil engineering
• Faster simulations
• Example application: biomedical engineering
• Model development
• Optimization
• Outlook
What is mechanics?

• Rigid body mechanics: robotics, machine elements
• Deformable body mechanics: civil engineering, biomechanics, materials
• Fluid mechanics: aerospace engineering, chemical engineering

• Common features:
  ★ mathematical models, often in the form of differential equations
  ★ all three branches may interact in a single problem:
  ★ crucial in technology and design

This talk will concentrate on incompressible fluid mechanics
Why computation? (1)

- Three methods of prediction and investigation:
- Brains are not getting faster!
- Experiments remain costly

- Computation benefits from Moore’s Law
**Why computation? (2)**

- Computer performance has been increasing exponentially:

- Gordon Moore predicted this in 1965

- Parallelism is the key
Simulations basics (I)

- Wanted: fluid velocity and pressure fields: $u(x, t)$ and $p(x, t)$
- Governed by nonlinear partial differential equation system:
  \[
  \rho \left( \frac{\partial u}{\partial t} + u \cdot \nabla u - f \right) - \mu \Delta u + \nabla p = 0
  \]
  \[
  \nabla \cdot u = 0
  \]
- Depends on strength of $\rho \ u \cdot \nabla u$ term relative to $\mu \Delta u$ term, i.e., Reynolds number
- Can be solved *analytically* only in special cases
- Flow around a cylinder at low Reynolds number (vorticity $\omega = \nabla \times u$):
Simulation basics (2)

- Flow around a cylinder at moderate Reynolds number:
  ![Flow around a cylinder at moderate Reynolds number](image1)
  ![Flow around a cylinder at moderate Reynolds number](image2)

- Flow around a cylinder at high Reynolds number:
  ![Flow around a cylinder at high Reynolds number](image3)
  ![Flow around a cylinder at high Reynolds number](image4)

- Followed by turbulence in 3D
- Solution impossible to write down in *closed form*
Examples of typical patterns generated in various kinds of fluid flow. Note the frequent occurrence of seemingly random turbulence.
Simulation basics (3)

• What is discretization?

<table>
<thead>
<tr>
<th>continuum problem</th>
<th>discrete problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f(x) )</td>
<td>( f(x) )</td>
</tr>
<tr>
<td>( x )</td>
<td>( x )</td>
</tr>
</tbody>
</table>

- infinite-
- analytical-
- limited-
- generally no

- unknowns-
- approach-
- applicability-
- approximation-

- finite-
- numerical-
- wide-
- generally yes

• Finite difference, finite volume and finite element methods
Finite Difference (FD) method

- Differential equation converted into stencils at mesh nodes:

- Simple, fast

- Inflexible
Finite Volume (FV) method

- Differential equation reformulated in terms of *fluxes* at cell boundaries:

- Flexible

- Mesh generation difficult in 3D
Finite Element (FE) method

- Differential equation reformulated in *weak* (variational) form
- Interpolation functions are defined piecewise over simple shapes

- Flexible
- Mesh generation difficult in 3D
Simulation basics (4)

- End result is always an equation system
- Often too large to be solved directly
- Iterative solution techniques are used
- Sequence of solution guesses, introducing another layer of approximation
Simulation basics (5)

- Refinement

In 3D, doubling the refinement leads to 8 times as many elements!

![Coarse Discretization](image1)

- **fewer**
  - **low**
  - **unknowns**
  - **accuracy**

![Fine Discretization](image2)

- **many**
  - **high**
Simulation basics (6)

- How many finite elements?

- Problems computed for *thousands* of time steps

- Optimization or model identification may require *tens* of simulations
Example: civil engineering (1)

- Challenge: prediction of water flow in or around hydraulic structures:
  - dams, spillways and channels
  - offshore platforms
  - storage tanks and basins
- Design objectives:
  - minimize pressure variations
  - prevent erosion
  - minimize overspill
- Modeling issues:
  - free-surface flow with boundaries unknown in advance
  - deforming computational meshes
  - large scale span: 1 mm to 1 km
Example: civil engineering (2)

- Spillway of the Olmsted dam on the Ohio River
- Experiments in a scale model at U.S. Army Corps of Engineers Waterways Experiment Station:
  - Simulation restricted to a periodic spillway section
- Obstacles dissipate flow energy to reduce erosion downstream:
Example: civil engineering (3)

- Computation involves 418,000 tetrahedra and 1000 time steps:
Example: civil engineering (4)

- Mesh is deforming in response to the water surface movement:
  - Streamwise velocity component color-coded in the right movie
  - Quasi-steady state reached
  - Numerical prediction accurately reflects position of a hydraulic jump
Example: civil engineering (5)

- Trapezoidal channel with bridge supports
- Experiments in a scale model at U.S. Army CEWES:

- Blockage due to the columns could submerge the bridge surface
Example: civil engineering (6)

- Computation involves 163,000 tetrahedra and 2000 time steps:
Example: civil engineering (7)

- Streamwise component of velocity is animated viewed from upstream:
Faster simulations (1)

- Can you recognize any of these machines?

1 TeraFLOPS = $1 \times 10^{12}$ 64-bit floating-point ops per second

1 TeraByte = 137,438,953,472 64-bit floating-point numbers can be held in memory
Faster simulations (2)

- **Earth Simulator** in Yokohama:
  - 5120 custom NEC CPUs
  - 41 TFLOPS
  - 10 TBytes of memory
Faster simulations (3)

- **IBM Blue Gene/L** in Livermore:
  - 65,536 PowerPC 440 CPUs
  - 280 TFLOPS + 16 TBytes RAM
  - very low power usage

- **Sun E25K** in Aachen
  - 1536 UltraSPARC IV CPUs
  - 3.5 TFLOPS + 3 TBytes RAM
  - large shared memory

- **“Normal” PC**
  - 1–2 x86 CPUs
  - 0.01 TFLOPS + 0.002 TBytes
  - drives the Moore’s Law!
Example: bioengineering (1)

- Heart disease statistics:
  - Number one killer of adults
  - 50,000 people in the US need heart transplant
  - 2500 donor hearts available in the US annually
  - Same percentages worldwide

- Medical technology timeline:
  - 1953 heart-lung machine enables first open-heart surgery
  - 1964 NIH-funded research projects on heart replacement (TAH) or augmentation (LVAD) initiated
  - 1966 first implantable TAH and LVAD under development
  - 1982 Jarvik-7 TAH implanted at Utah for 4 months
  - 2001 Abiocor TAH implanted, one patient survives 17 months
Where do heart-assist devices go?

- **Left ventricle** supplies the aorta
- **Right ventricle** supplies the pulmonary artery

Heart-assist device configurations:

- Roller pump\(^1\) for surgery support
- Pulsatile TAH (Jarvik, Abiocor\(^2\)) and LVAD (Thoratec, DLR\(^3\))
- Axial rotary LVAD (Impella, MicroMed\(^4\), Pitt Streamliner)
- Centrifugal rotary LVAD (Medtronic, Nikisso, Baylor GYRO)
Example: bioengineering (3)

- GYRO centrifugal LVAD:

- Under development at the Baylor College of Medicine

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chamber diameter</td>
<td>56.4 mm</td>
</tr>
<tr>
<td>Inflow tube diameter</td>
<td>11.0 mm</td>
</tr>
<tr>
<td>Outflow tube diameter</td>
<td>10.7 mm</td>
</tr>
<tr>
<td>Shear layer diameter</td>
<td>51.8 mm</td>
</tr>
<tr>
<td>Shear layer thickness</td>
<td>0.13–1.02 mm</td>
</tr>
<tr>
<td>Rotation speed</td>
<td>3000 rpm</td>
</tr>
<tr>
<td>Flow volume</td>
<td>~5 l/min</td>
</tr>
</tbody>
</table>
Example: bioengineering (4)

- Hybrid structured/unstructured mesh:

- Shear layer allows one part of computational grid to rotate

<table>
<thead>
<tr>
<th>elements</th>
<th>1,138,145</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>393,786</td>
</tr>
<tr>
<td>shear layer elements</td>
<td>58,200</td>
</tr>
<tr>
<td>time step size</td>
<td>0.0003 s</td>
</tr>
<tr>
<td>time steps</td>
<td>1000</td>
</tr>
<tr>
<td>non-linear iterations</td>
<td>4</td>
</tr>
<tr>
<td>GMRES iterations</td>
<td>80(2)</td>
</tr>
</tbody>
</table>
Pressure and particle traces at 2000 rpm and 90 mm Hg
Example: bioengineering (6)

Stress iso-surface at 2000 rpm and 90 mm Hg
Example: bioengineering (7)

- Impeller forces history at 2000 rpm and 80 mm Hg:
Example: bioengineering (8)

- Flux history at 2000 rpm and 80 mm Hg:
Example: bioengineering (9)

- **Hydraulic performance chart:**

![Graph showing hydraulic performance chart with data points for different rpm experiments and simulations.](image-url)
Example: bioengineering (10)

- Parallel computation necessary for reasonable turnaround
- ~20 steps per hour on 32 CPUs of a Pentium 4 cluster
- ~150 steps per hour on 256 CPUs of a Blue Gene/L
- **Scalability** of GYRO pump computations:

![Graph showing scalability of GYRO pump computations](f-pump.1138-timing-2005-10-jubl.ps)
Model development (1)

- Flow fields in GYRO pump are nice to look at
- Surgeons at Baylor don’t care!
- Need relation between flow and blood damage (hemolysis)
- Let’s look at what is known:
  - Hemolysis in LVADs depends on shear stress and exposure time
  - A standard measure (NIH) is related to $\Delta \frac{Hb}{Hb}$ (hemoglobin release)
  - Short-exposure correlation available from steady-shear experiments:
    \[
    \frac{\Delta Hb}{Hb} = 3.62 \times 10^{-7} \sigma^{2.416} \Delta t^{0.785}
    \]
  - Stress-based models integrate stress along the pathlines
  - They *overpredict* hemolysis
Model development (2)

- Red blood cells (RBCs) are elastic and don’t adjust to stress right away
- Should we model each RBC as it passes through the pump?

A $3 \times 3$ morphology tensor $\mathbf{B}$ can roughly represent an RBC.
Model development (3)

- Evolution equation for the morphology tensor:

\[
\frac{d\mathbf{B}}{dt} - [\Omega \cdot \mathbf{B} - \mathbf{B} \cdot \Omega] = -f_1 [\mathbf{B} - g(\mathbf{B})\mathbf{I}] + f_2 [\tilde{\mathbf{E}} \cdot \mathbf{B} + \mathbf{B} \cdot \tilde{\mathbf{E}}] + f_3 [\tilde{\mathbf{W}} \cdot \mathbf{B} - \mathbf{B} \cdot \tilde{\mathbf{W}}]
\]

- Accounts for elasticity and membrane kinematics (tank-treading)
- Flow field enters through strain rate tensor \( \tilde{\mathbf{E}} \) and vorticity tensor \( \tilde{\mathbf{W}} \)
- Strain-based model can be calibrated using the steady-shear correlation
- Averaged over many pathlines:

<table>
<thead>
<tr>
<th>Hemolysis [g/100l]</th>
<th>1600 rpm</th>
<th>2000 rpm</th>
<th>2350 rpm</th>
</tr>
</thead>
<tbody>
<tr>
<td>stress-based</td>
<td>0.0821</td>
<td>0.1560</td>
<td>0.2276</td>
</tr>
<tr>
<td>strain-based</td>
<td>0.0085</td>
<td>0.0114</td>
<td>0.0184</td>
</tr>
<tr>
<td>experiment</td>
<td>0.0070</td>
<td>0.0070</td>
<td>0.0200</td>
</tr>
</tbody>
</table>
Model development (4)

- Stress-based versus strain-based models simply explained:
Optimization (1)

• Standard way of testing MicroMed axial pump design changes:
  ★ Machine new part
  ★ Run test-loop experiments
  ★ 2–3 configurations considered
  ★ non-intuitive changes avoided

• Automatic design optimization:
  ★ Large range of configurations
  ★ Best parameters pinpointed
  ★ Less reliance on intuition
Optimization (2)

- Example of inverse problem:

- Direct solutions only reproduce or replace analysis or experiments
- Inverse solutions are unique to the computational approach!
Outlook (1)

- Increased complexity of engineered systems
Outlook (2)

• Increased complexity of engineered systems:
  ★ Multiphysics: air flow + combustion, chemical flow + reaction
  ★ Multiscale: from molecules to factories

• Need for optimal design and inverse techniques:
  ★ Repeated simulations to pinpoint best design
  ★ Repeated simulations to validate and fine-tune a model

• Will Moore’s law still apply?
  ★ Parallelism is the key
  ★ Quantum computing
  ★ Molecular computing
  ★ DNA computing
Simulationstechnik I

Thank You!