High Performance and Parallel Computing (MA 6241)

Lecture 1

Welcome and Introduction

Ulrich Rüde
Lehrstuhl für Simulation
Universität Erlangen-Nürnberg

visiting Professor at
Department of Mathematics
National University of Singapore

Setup of Course

- Taught jointly by
  - Prof. Ulrich Ruede
  - Prof. Weizhu Bao (NUS)
  - Further guest lectures in Feb/March
    - Dr. B. Gmeiner (Post Doc at LSS, visiting NUS)
    - others ?

Who am I?

- Prof. Dr. U. Rüde, Lehrstuhl für Systemsimulation
  - https://www10.informatik.uni-erlangen.de/en/~ruede/
  - e-mail: ulrich.ruede@fau.de
- Personal background
  - studies of math, computer science and applied math at TU Munich and The Florida State University
  - PhD and Habilitation Degrees from TU Munich
  - Post Doc at Univ. of Colorado
  - Professor at
    - TU Chemnitz
    - Univ. of Augsburg
    - Univ. of Erlangen (since 1998)
    - Univ. of Colorado (visiting)
    - Natl. Univ. of Singapore (visiting Jan 2015 - Mar 2015)

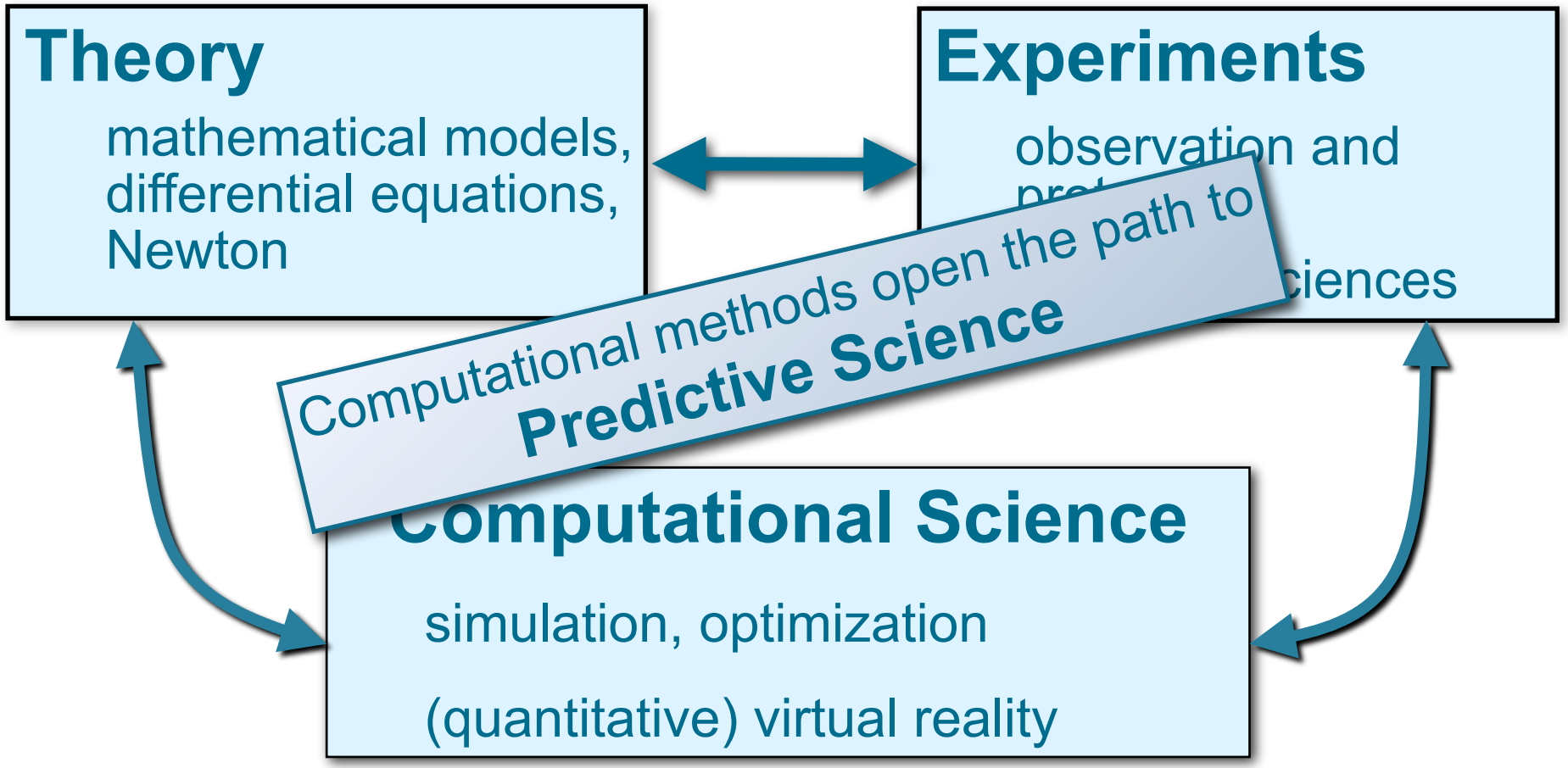
Organization of High Performance and Parallel Computing

- weekly Lecture: Tuesday 19:00-22:00
- Syllabus with more information will come soon
- No final exam
- 4 homework assignments
- grading (preliminary)
  - 10%
  - 20%
  - 20%
  - 50% (final project)
- Textbooks
  - Hager, G., & Wellein, G. (2010). Introduction to high performance computing for scientists and engineers. CRC Press.
  - Goedecker, S., & Hoisie, A. (2001). Performance optimization of numerically intensive codes. Siam.

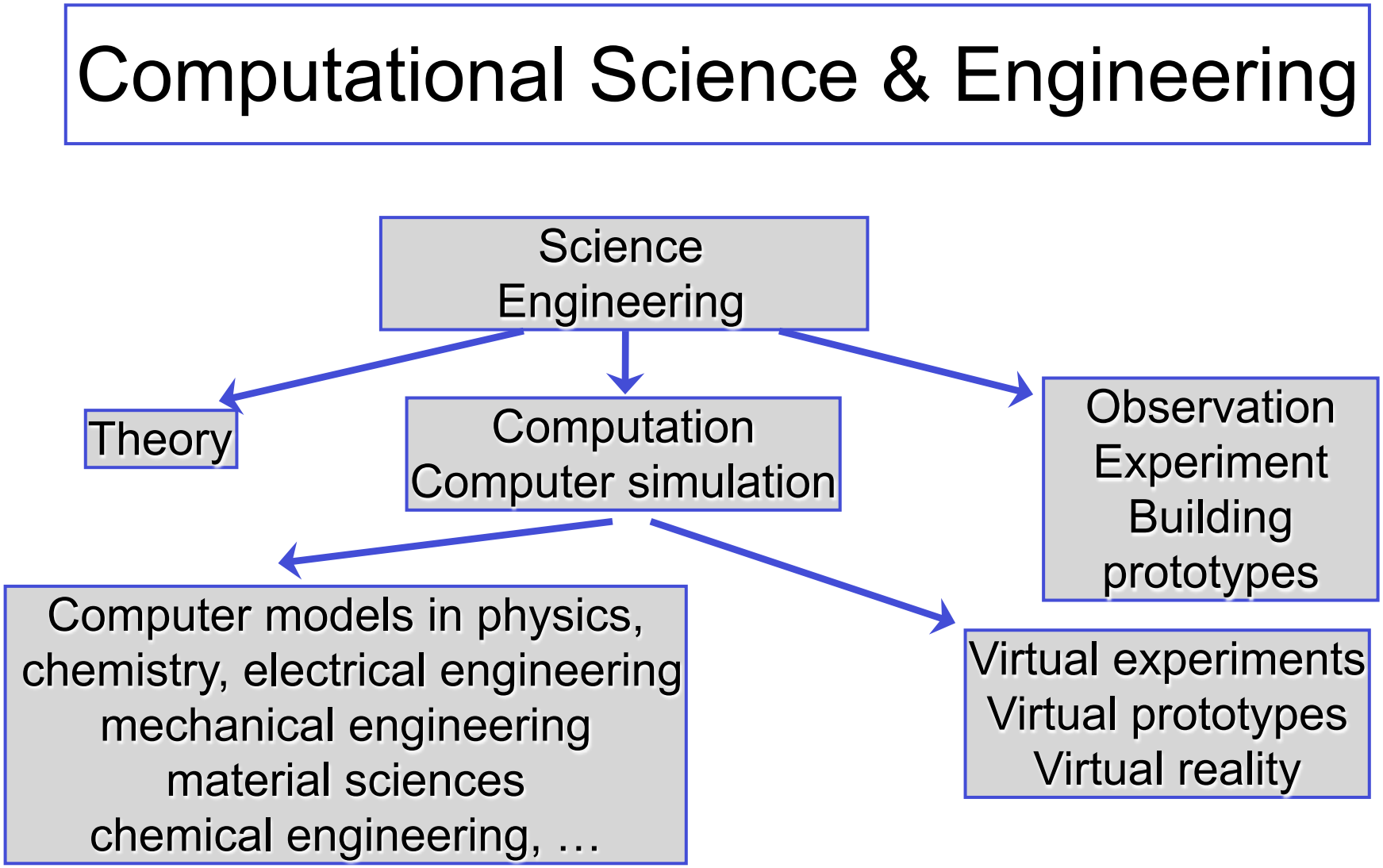
Contents of High Performance and Parallel Computing

- Introduction
  - Computational Science and Engineering
  - High Performance Computing
- Computer Architecture
  - memory hierarchy
  - pipeline
  - instruction level parallelism
  - multi-core systems
  - parallel clusters
- Efficient Programming
  - computational complexity
  - efficient coding
  - architecture aware programming
  - shared memory parallel programming (OpenMP)
  - distributed memory parallel programming (MPI)
  - parallel programming with Graphics Cards

The ~~Two~~ Three Principles of Science



High Performance and Parallel Computing is a central foundation of CSE



TECHNISCHE FAKULTÄT
Friedrich-Alexander-Universität Erlangen-Nürnberg
Institut für Informatik

The Logic of Predictive Science: What is it?

thanks to T. J. Oden, UT Texas, ICES

Mathematical constructions based on physical principles or empirical relations - generally derived from inductive theories that attempt to characterize abstractions of physical reality.

The process of adjusting the parameters of a model to improve the predictions of th emodel to better agree with experimental measurements

The process of determining the accuracy with which a model can predict features of reality

Predictive Science: the scientific discipline concerned with assessing the predictability of mathematical and computational **models** of reality. It embraces the processes of model selection, **calibration**, **validation**, verification, and their use in forecasting **the relevant features of reality** with **quantified uncertainty**.

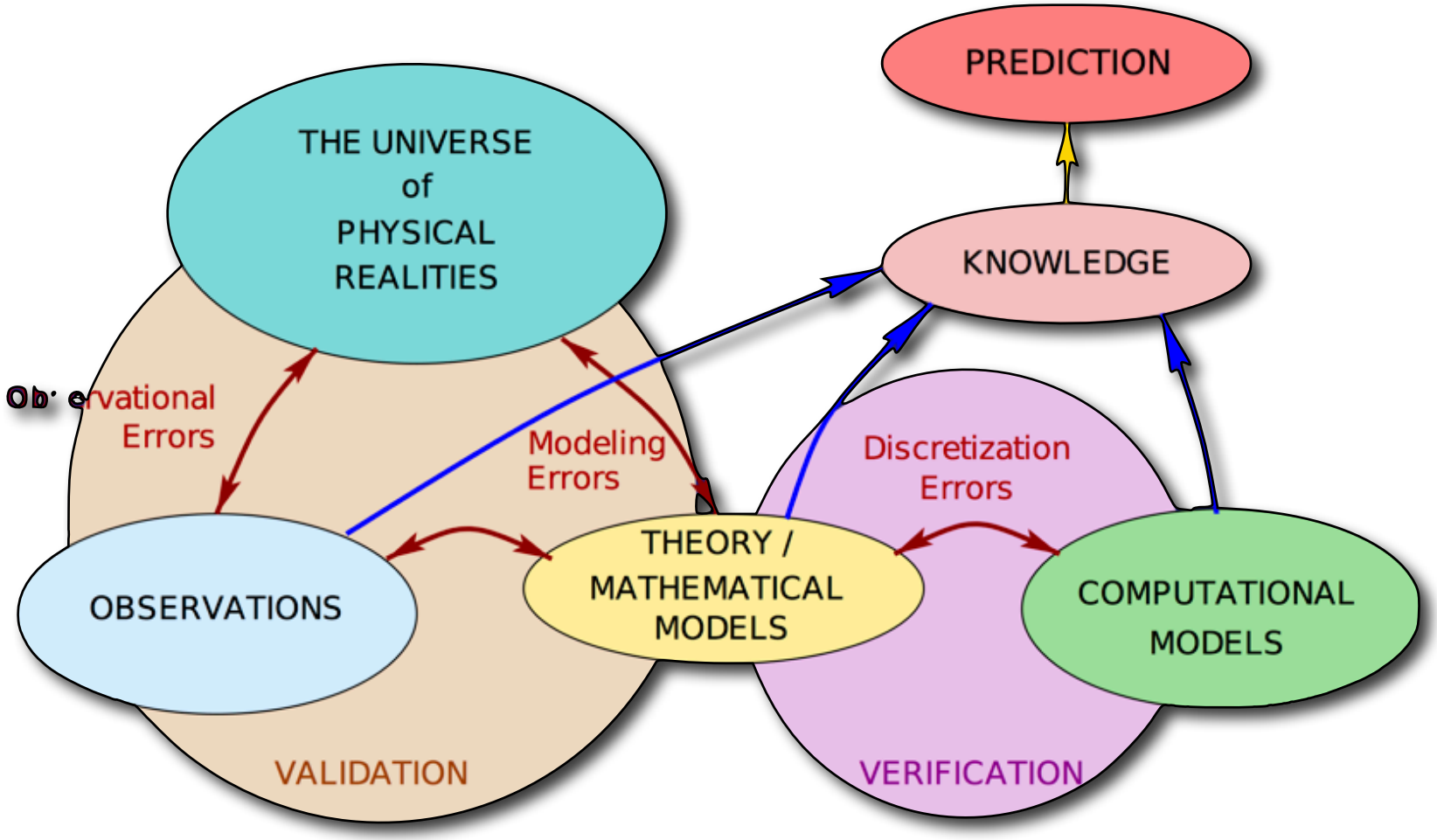
The *quantities of interest (QI):* the goals of the simulation

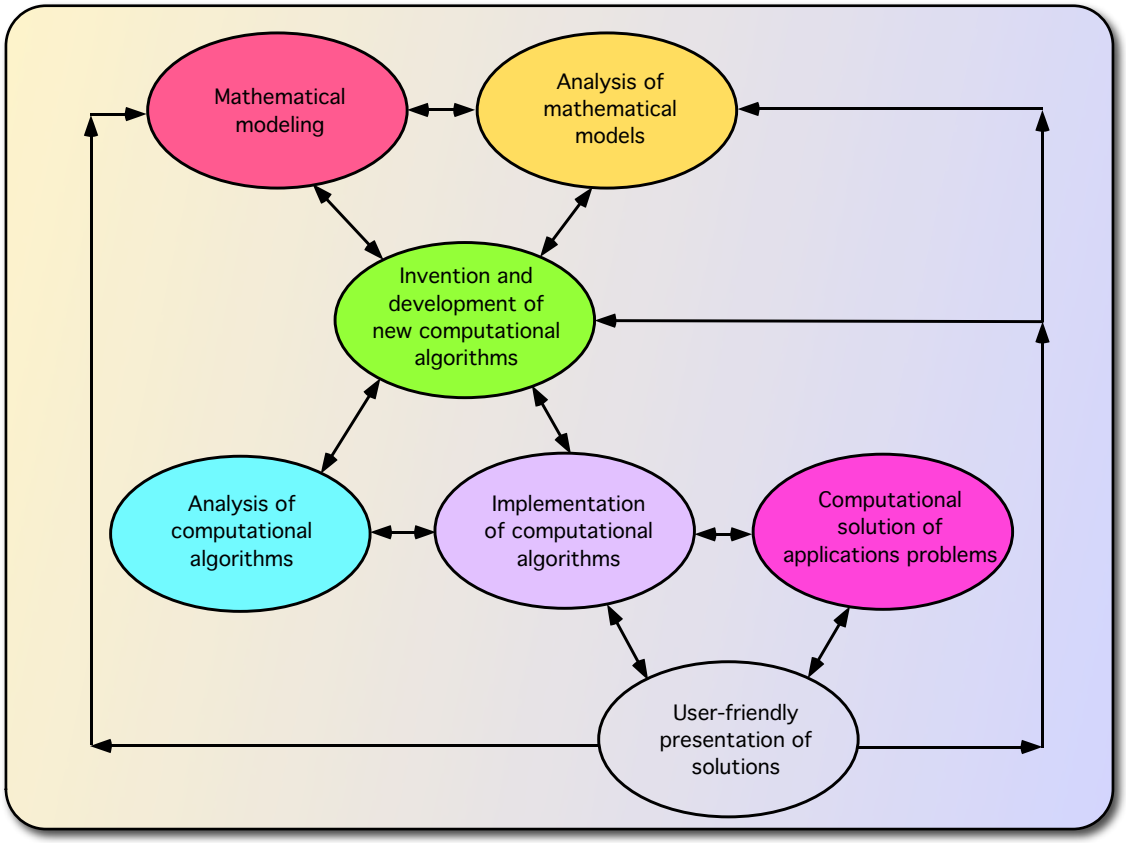
UQ: quantitative analysis of the uncertainties in the predicted QI

Predictability requires **knowledge** of the physical laws that are proposed to explain realities and requires recognizing and quantifying uncertainties.

The imperfect paths to knowledge

diagram from J.T. Oden





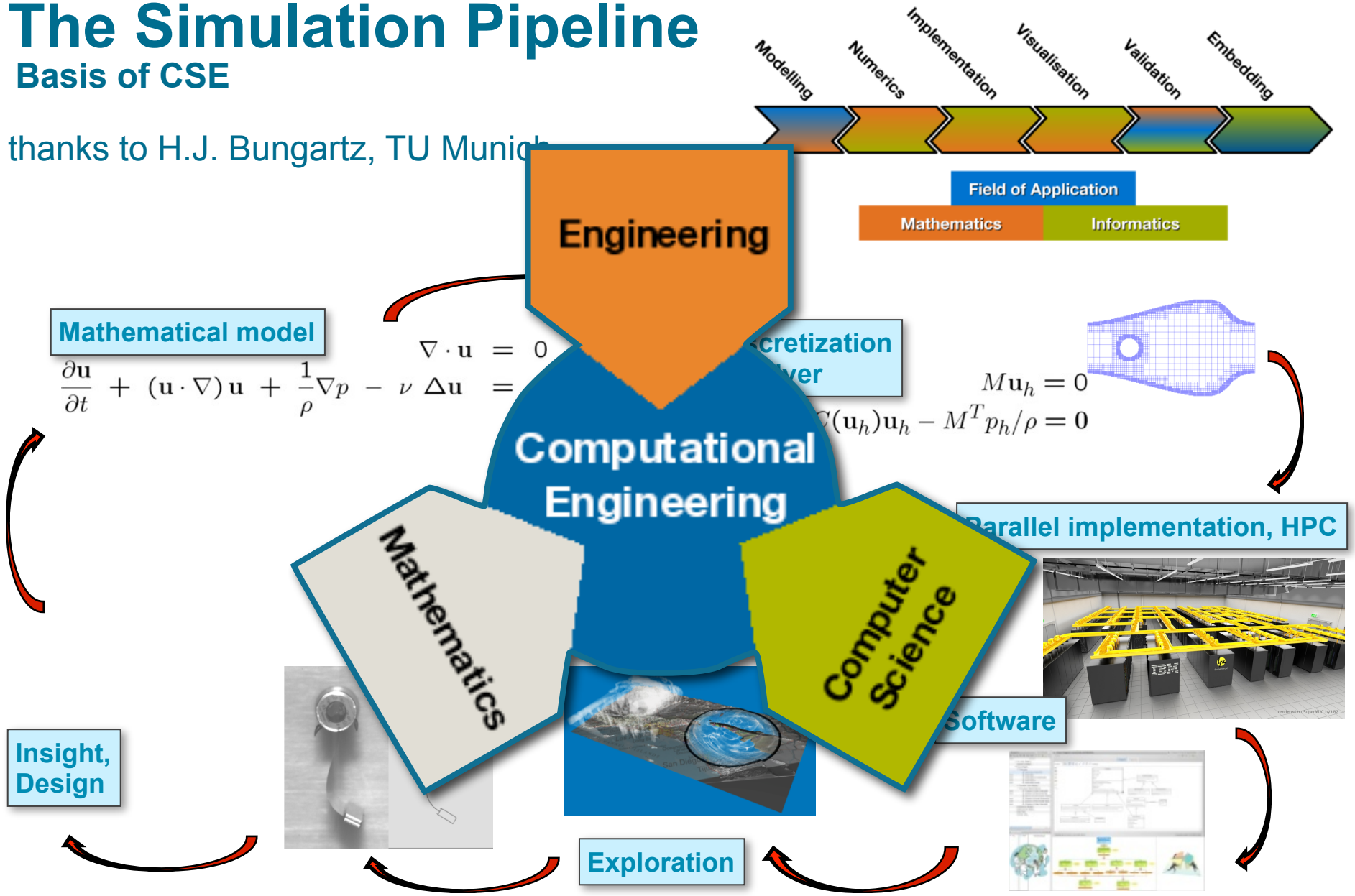
Max Gunzburger's (FSU) diagram on CSE

Bringing Things Together in Computational Science and Engineering

The Simulation Pipeline

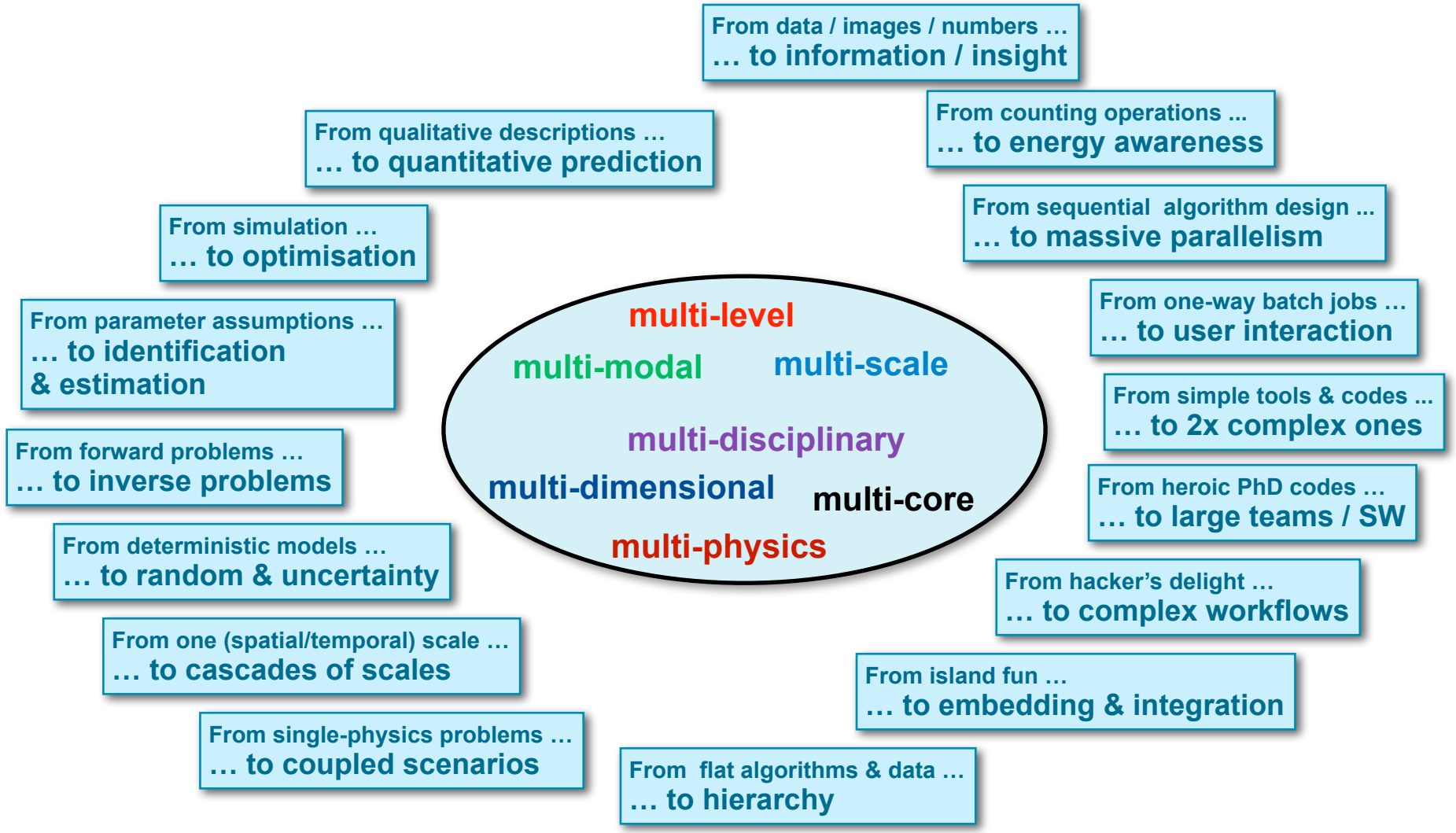
Basis of CSE

thanks to H.J. Bungartz, TU Munich



All this is just about to take off…

slide thanks to H.J. Bungartz, TU Munich



The CSE Pipeline

... many CS&E problems can be characterized by a "pipeline" that includes

- Modeling techniques (mathematical and geometric)
- Simulation techniques (discretizations, algorithms, data structures, software frameworks, and problem solving environments), and
- Analysis techniques (data mining, data management, visualization, and error, sensitivity, stability, and uncertainty analyses).

CS&E tackles problems from the real world and is characterized by employing several stages of the CS&E pipeline for solving the problem.

Research in CS&E should illustrate new and useful techniques and tools for solving realistic problems, which often have complicated three-dimensional geometries, multiple scales, heterogeneities, anisotropies, and multi-physical or biological descriptions.

Such problems often thwart proofs of accuracy or efficiency, but leading edge work in CS&E should address validation and verification through reduction to analyzable cases and convergence studies, as applicable, and comparisons with alternative approaches.

A Definition of CSE (and what it is NOT)

- development of problem-solving methodologies
- robust tools for solutions to scientific and engineering problems
- ever-increasing complexity
- differs from mathematics or computer science
  - directed specifically at the solution of problems
    - from science and engineering
    - Requires
      - detailed application knowledge
      - substantial collaboration from those disciplines

It is more than a scientist or engineer using a canned code
to generate and visualize results
(skipping all of the intermediate steps).

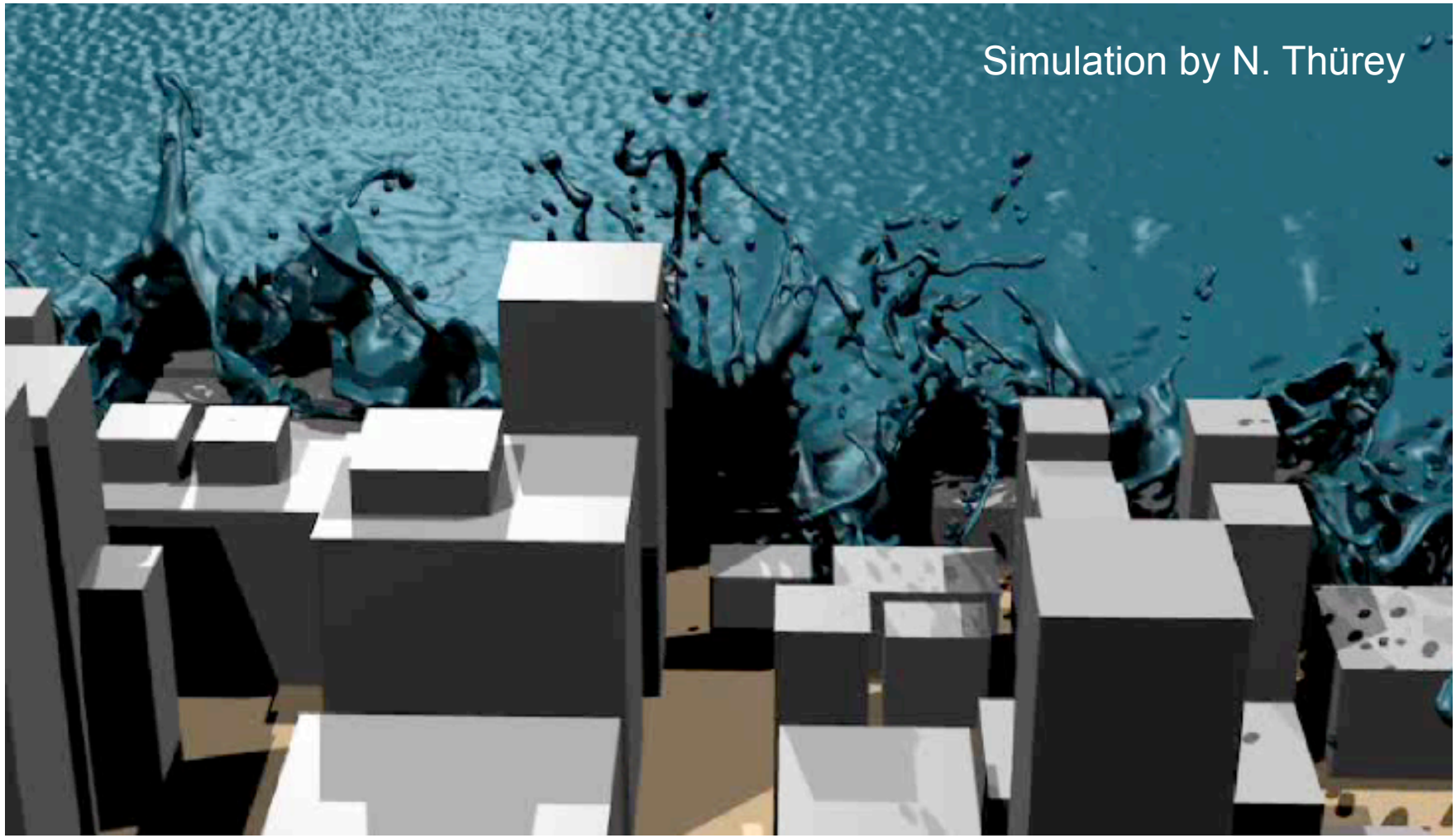
Motivation &
Review of Computer Architecture

Computational Intensity for Problems in Scientific Computing

- Example Problem 1: Flow Simulation
  - Airplane: compute lift and drag, or e.g. reacting flow in engine
  - Car: reduce drag and noise, or understand burning in cylinder
  - Tsunami hitting the coast
  - Biomedical flows (blood, eye liquid, ...)
  - Weather prediction
  - Ocean recirculation
  - Animation for movies and games
  - ... and many, many, ... more
- Example 2: Molecular Dynamics, i.e. simulate behavior of a large collection of atoms/molecules to understand
  - chemical reactions
  - drug design
  - material behavior/ failure

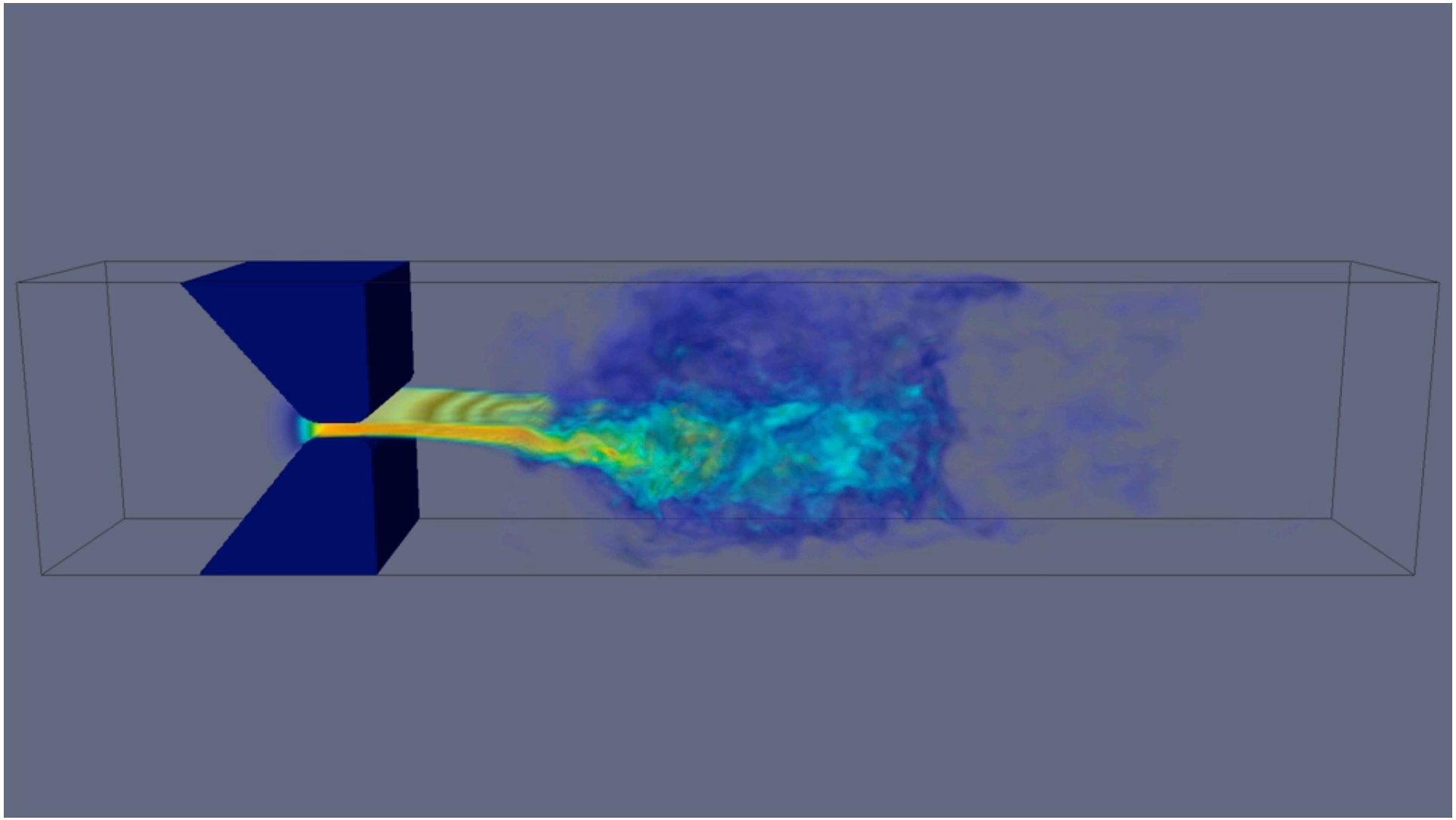
Examples of Flow Simulation

- Resolution: 880*880*336; 260M cells, 6.5M active on average



Examples of Flow Simulation

Computational Fluid Dynamics with waLBerla



with: F. Schornbaum, C. Godenschwager, E. Fattahi:
Simulation of phantom vocal fold geometry

Why do we need so much computing power?

For many physical problems, a 3D domain must be discretized.

Assume that we use

- $k_x \times k_y \times k_z = 10000 \times 10000 \times 10000$ cells
- $k_t = 10000$ time steps

Even in the *best case*, we need

$$O(k_x \times k_y \times k_z \times k_t)$$

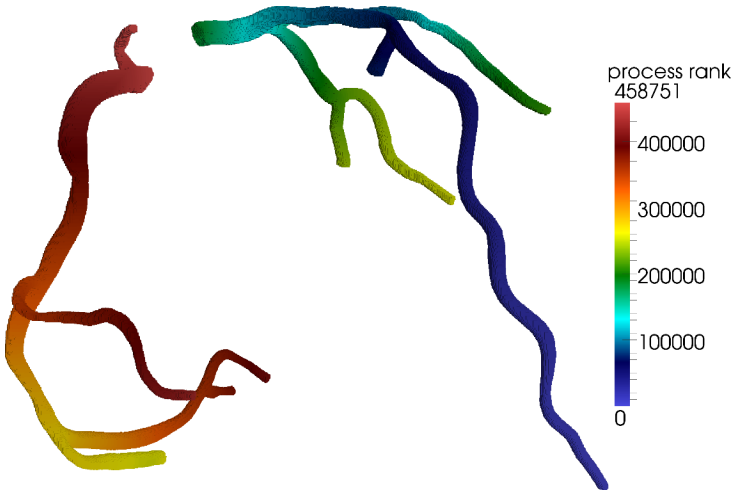
floating point operations. In our eaxmple, when 100 operations are used per cell and time step, we require 1000 000 000 000 000 000 = 10^{18} floating point operations.

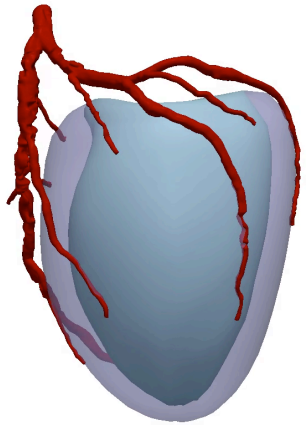
Also note that if we need 10 variables per cell, and 8 Byte per variable (double precision) then we need 80 000 Gbyte of memory.

The quick growth of complexity for higher dimensional problems is called the

curse of dimensionality.

Summary of Performance on Coronary Geometry





- Weak scaling on JUQUEEN with 458,752 processes
- over a trillion (10^{12}) fluid lattice cells
  - Cell sizes of 1.27µm (diameter of red blood cells about 7µm)
  - 2.1 10^{12} cell updates per second
  - 0.41 PFlops
- Strong scaling at cell sizes of 0.1 mm
  - In excess of 6000 time steps per second on 32,768 cores of SuperMUC
  - 2.1 million fluid cells

Godenschwager, C., Schornbaum, F., Bauer, M., Köstler, H., & UR (2013). A framework for hybrid parallel flow simulations with a trillion cells in complex geometries. In *Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis* (p. 35). ACM.

So What?

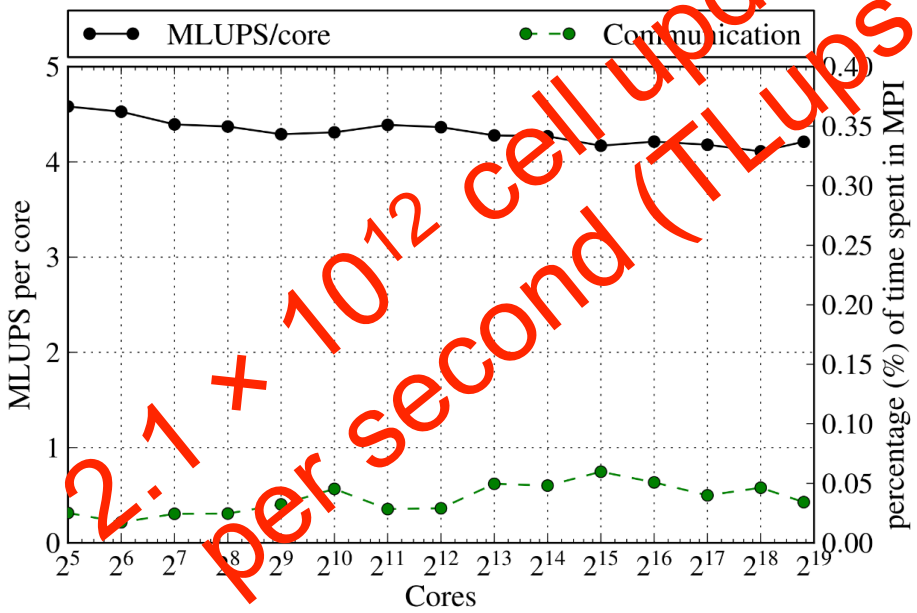
Example application:

- Flow with moving objects on the micro-scale (suspensions)
- 1μm lattice resolution, object size 5-10 μm (example: blood cells)
- We can simulate (with this resolution)
- on a laptop: 2 Million lattice cells = 0.002 mm^3 of liquid.
- on a current supercomputer with 40 TByte of memory: 80 Billion (8×10^{10}) lattice cells = 80 mm^3= 0.08 ml of liquid
  - for blood additionally required: 400 Million blood cells
- on systems that will reach peta scale peak performance (will become available in the next generation of supercomputers)
  - 2 Trillion lattice cells = 2 cm^3 = 2 ml of liquid
  - ~ 10 Billion blood cells
- on an systems that will reach peta scale performance on real applications (should become available next decade)
  - 20 Trillion lattice cells = 20 ml of liquid

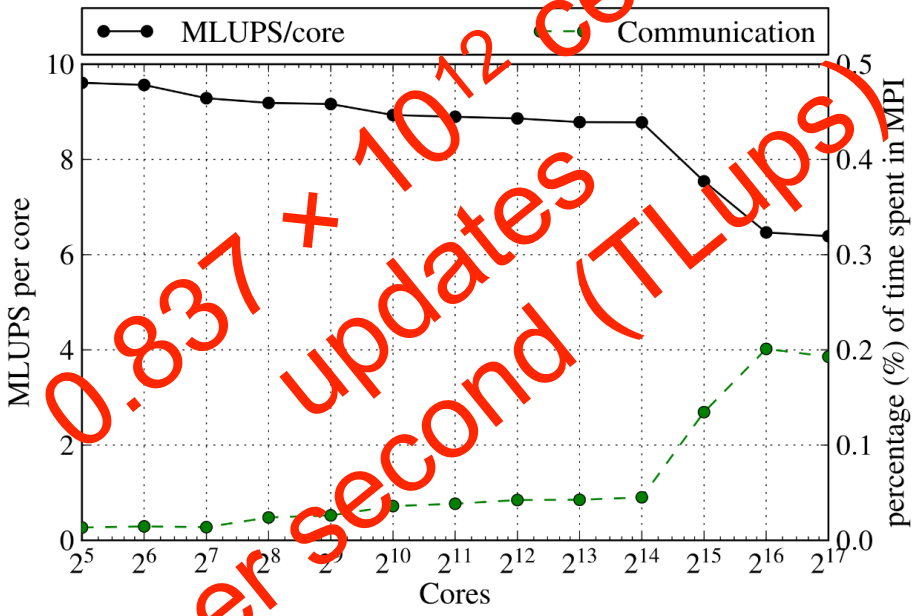
Is this good for anything?

Weak scaling (Lid Driven Cavity) TRT

JUQUEEN
16 processes per node
4 threads per process

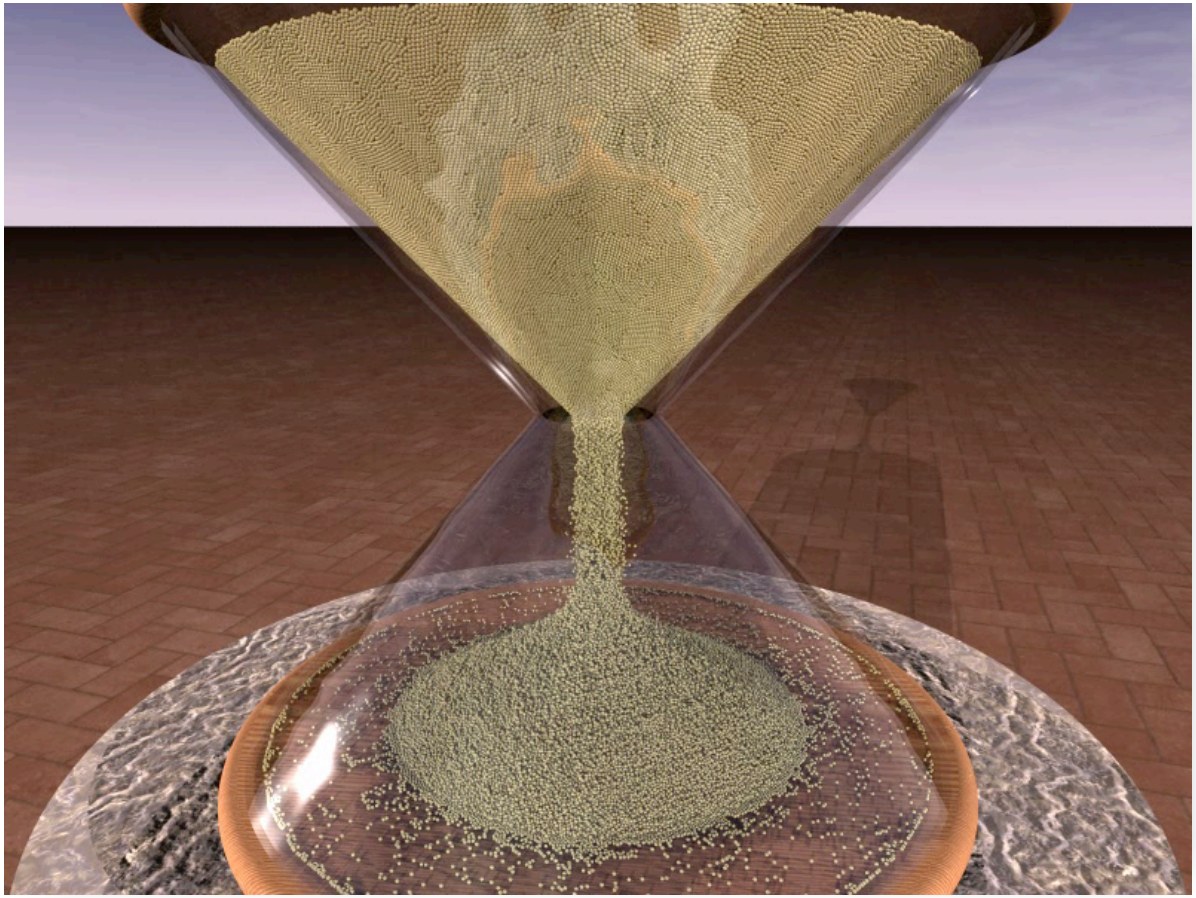


SuperMUC
4 processes per node
4 threads per process



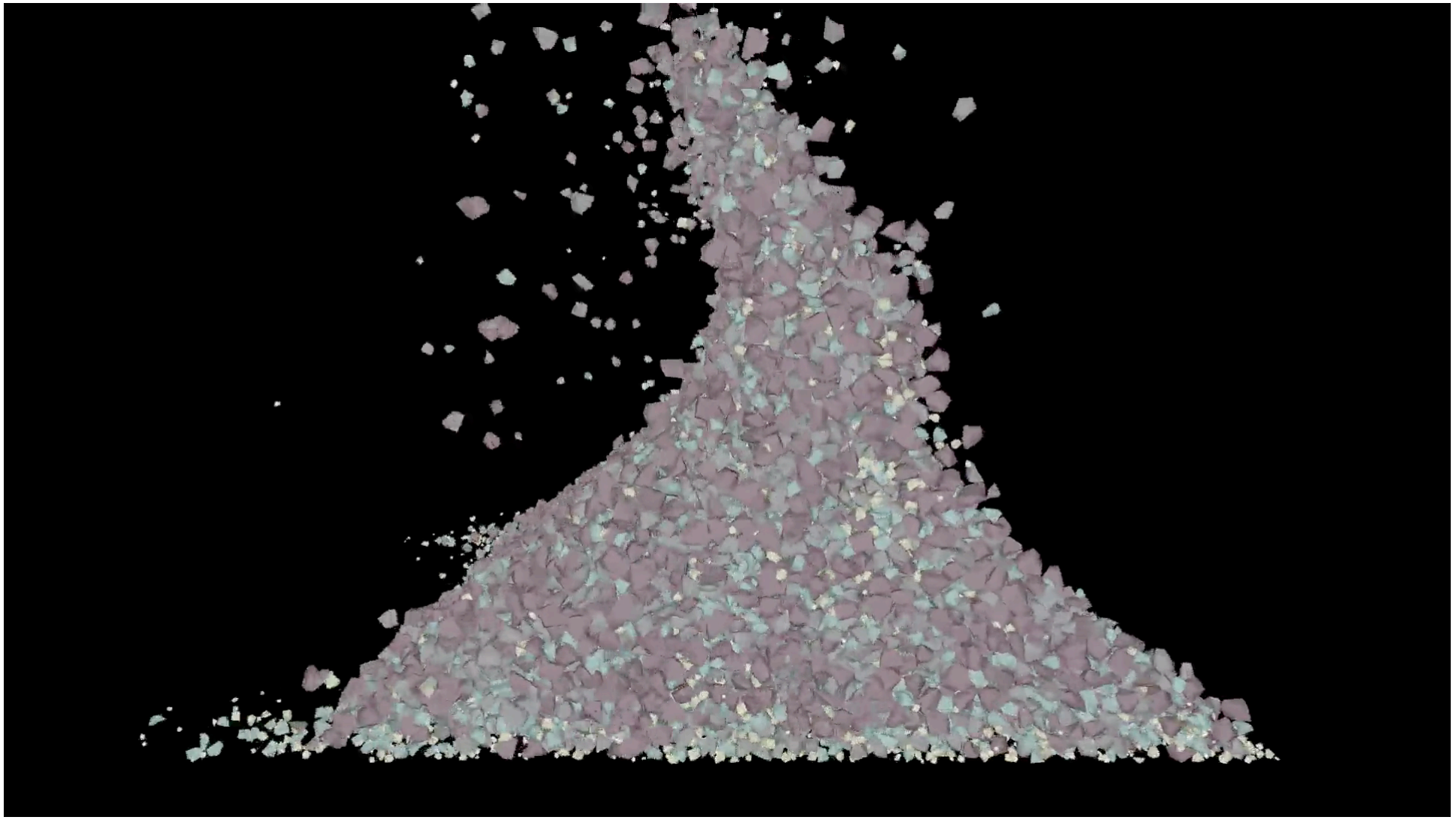
Example: Dynamics of many objects (rigid body dynamics)

Simulation by Klaus Iglberger



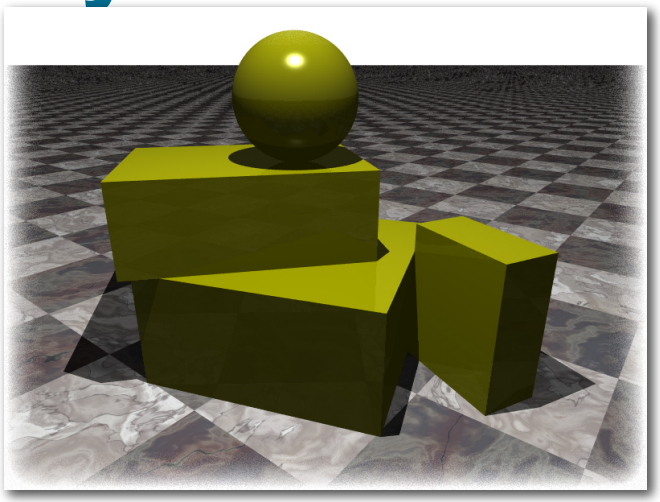
1250000 spherical particles, 256 CPUs, 300300 time steps, runtime: 48h (including data output), texture mapping, ray tracing

Shaker scenario with sharp edged hard objects

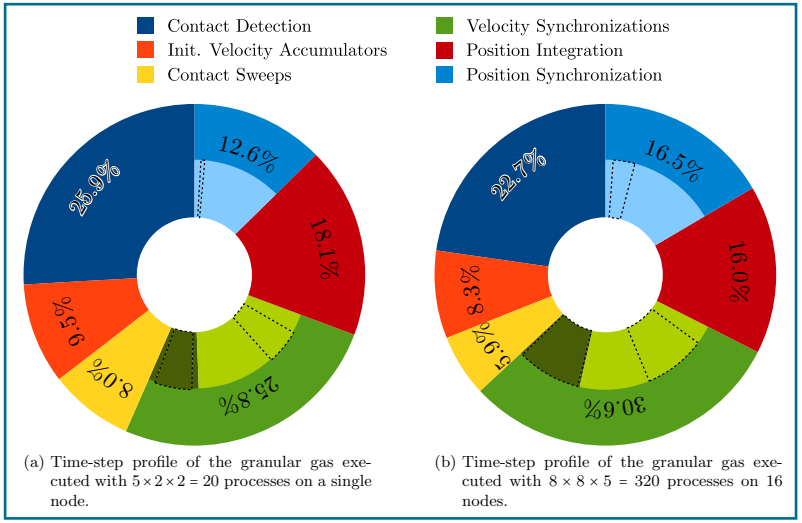


Computational Granular Dynamics

- Alternative contact models
  - FFD - Fast Frictional Dynamics (D. Kaufman et. al. 2005)
  - smooth: Discrete Element Method (DEM)
  - nonsmooth: Linear (LCP)/ Nonlinear Complementarity (NCP)
- MPI parallelization scaling up to full size
- largest simulation up to date:
  - **28,311,552,000 fully resolved particles** (NCP solver)
  - O(10^{11}) collisions
- novel contact model (T. Preclik) based on
  - generalized maximum dissipation principle
  - large-scale MPCC = mathematical program with complementarity constraints



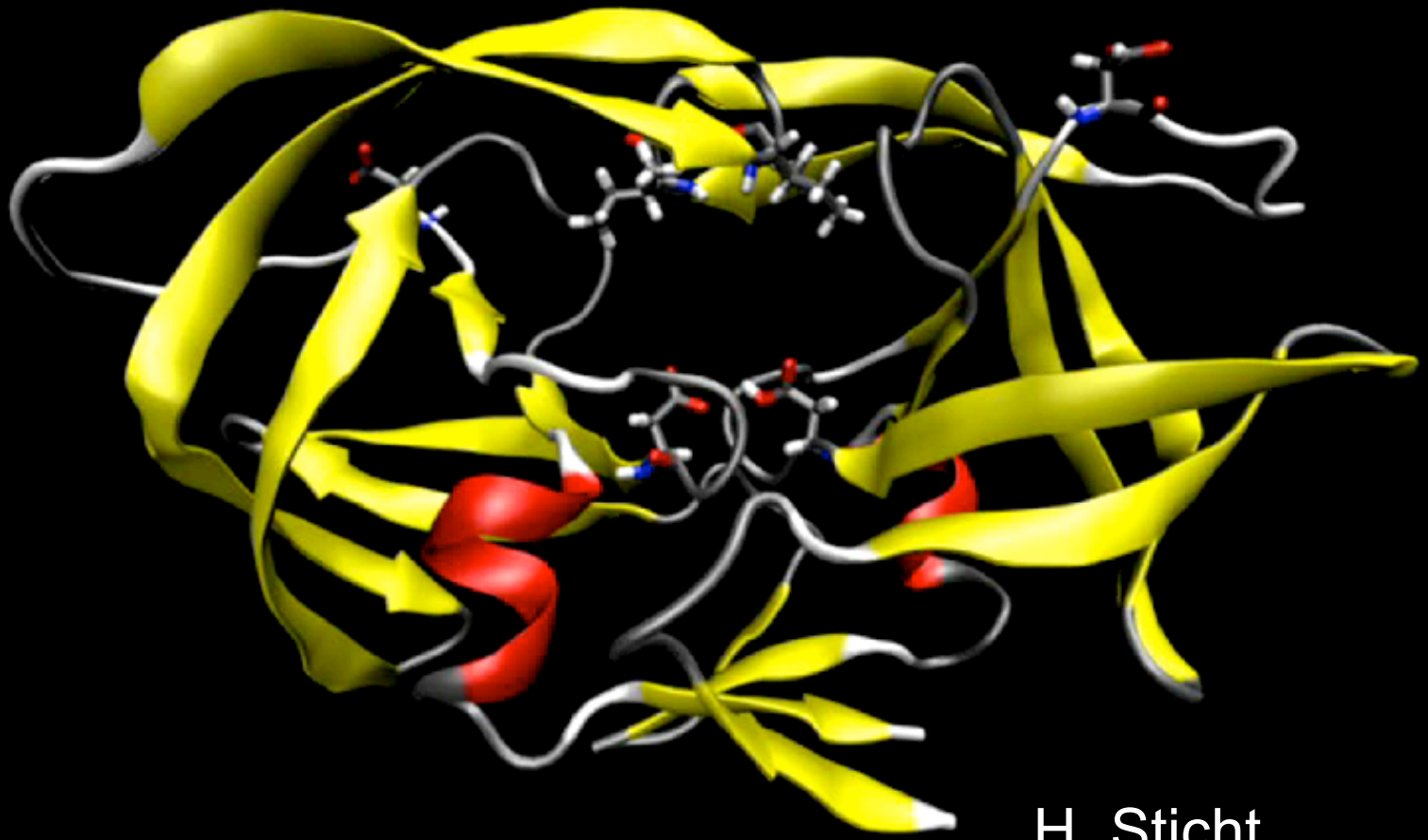
Collisions & Contacts between Geometric Objects



(a) Time-step profile of the granular gas executed with $5 \times 2 \times 2 = 20$ processes on a single node.

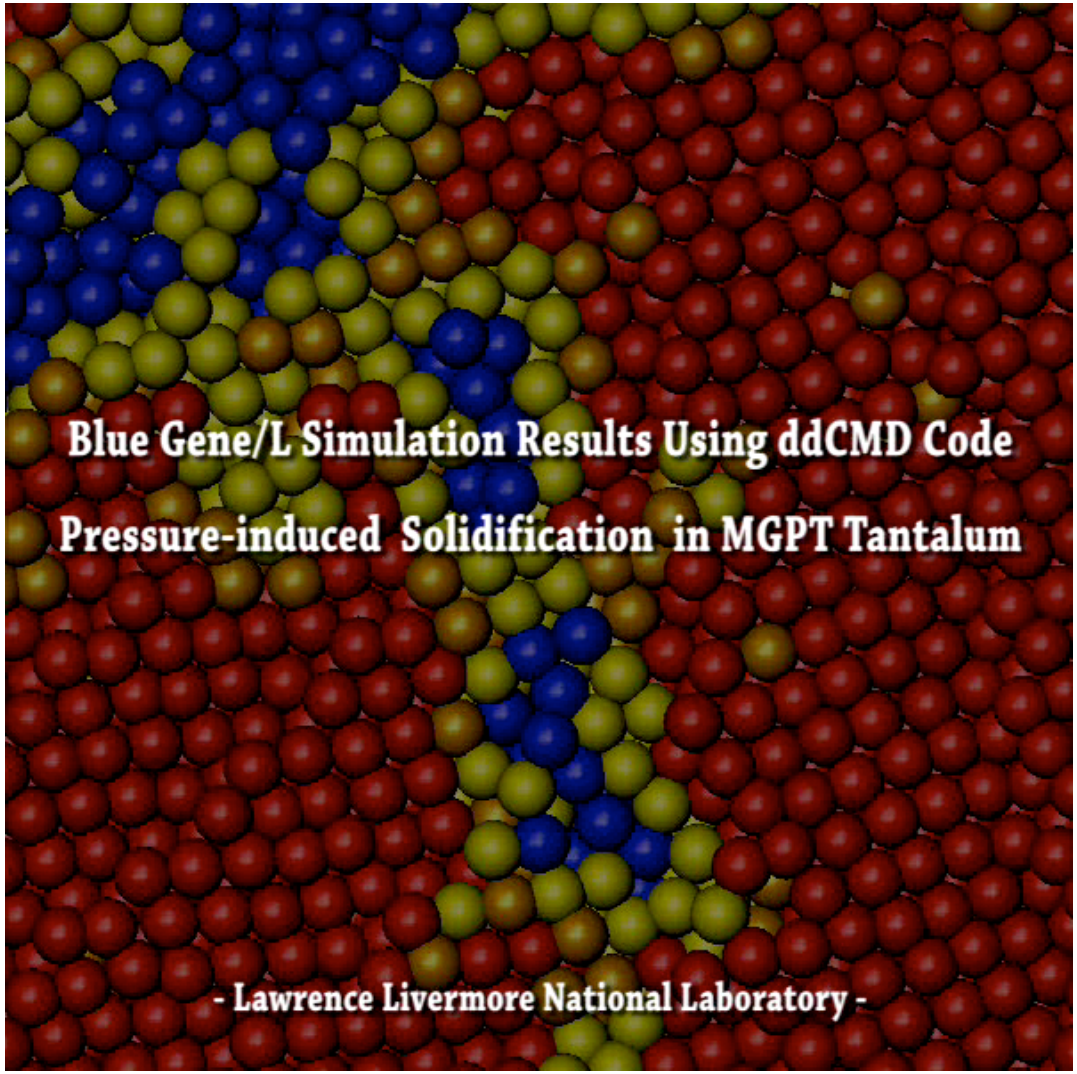
(b) Time-step profile of the granular gas executed with $8 \times 8 \times 5 = 320$ processes on 16 nodes.

Molecular dynamics simulation of HIV-Protease



H. Sticht
(FAU, Inst. for Biochemie)

Visualization illustrates the power of supercomputing for science (LLNL, S. Ashby)



- World's most powerful supercomputer, IBM BlueGene/L at LLNL, enables simulation of physical systems at a resolution heretofore impossible
- Movie shows detail resolved by increasing number of atoms in a molecular dynamics simulation from 5000 to 500,000,000

Some questions so that I understand better your background

Who of you have

- Programming experience?
  - Matlab
  - Fortran
  - C
  - C++
  - others?
- worked with (large?) (complex?) software?
- knowlege about computer architecture
  - Cache
  - SIMD
  - GPU
  - TLB

Who of you know?

- Message passing
- thread
- deadlock
- barrier
- MPI
- OpenMP
- Cuda

How many of you know about

- algorithmic complexity?
  - sorting (a list of numbers) has O(?) complexity?
  - solving a dense linear system has O(?) complexity?
  - solving Laplaces equation with the xxxx algorithm has O(y) complexity?
  - solving the traveling salesman problem has ... complexity?
- Single precision accuracy has x digits, double precision has y digits and is s times slower.
- When solving an elliptic PDE the computationally most expensive part is ...
  - ... generating the mesh
  - ... assembling the stiffness matrix
  - ... performing the ... iteration
  - ... writing the output to a file
  - ...

Have you learned about ...?

- The conjugate gradient algorithm works only for symmetric positive definite matrices. T/F
- The conjugate gradient algorithm gets as input a matrix and a vector. T/F?
- A sparse matrix can be stored efficiently in the CRS format T/F.
- What is a V-cycle?
- For linear systsms with sparse matrices, iterative algorithms are always more efficient than direct methods T/F.
- The condition number is always larger than the spectral radius.

First homework (10%)

- Inform yourself in the internet about
  - LINPACK
  - TOP 500
  - what is the LINPACK performance of your own PC/ Laptop and predict how long will it therefore take to solve a dense system on your PC with N=1000 unknowns.
- Starting from there search for web sites with information on
  - computer performance and its evolution
  - benchmarking computer performance for scientific comp.
  - critique of LINPACK and TOP 500
  - technological trends
- Summarize your findings in an essay
  - with $2 \leqq p \leqq 4$ pages, and
  - submit as pdf file, due date Jan 23 (midnight).

Current and Future High Performance Supercomputers

Performance Requirements for MD Simulations?

In molecular dynamics (MD), the forces between atoms must be computed in each time step. This must be kept as cheap as possible, in particular we must avoid that each atom/molecule interacts individually with each other one.

In some applications (e.g. in material science), we would like to compute ensembles of many millions of atoms.

(Remember Loschmidt's constant = $2{,}687 \times 10^{25}\ \mathrm{m}^{-3}$)

Due to physical restrictions, a time step is very short: Picoseconds (10^{-12}) or Femtoseconds (10^{-15}) since we must resolve thermal oscillations. Therefore even to simulate a nanosecond (10^{-9}), we may need thousands to millions of time steps.

Molecular dynamics has an insatiable hunger for computer performance!

What is the performance of current computers?

Relevant performance metrics:

- Floating Point operations/ second (Flops)
- Memory capacity (Byte)

Throughput vs. Latency:

- Throughput = number of operations performed per second
- Latency = how long does it take for an operation to complete

(operation = e.g. floating point operation, memory read operation, communication operation, etc.)

Parallel execution $\Rightarrow$ Throughput $\neq$ 1/Latency

Especially important on (all) computer systems:

- Throughput and Latency of the
  - memory system and of the
  - network connecting different parts of the computer system.
- The throughput of data transfer operations is also called "bandwidth" (measured in bytes per second).

A little quiz ...

1 Kflops = 10^3, 1 Mflops = 10^6, 1 Gflops = 10^9, 1 Tflops = 10^{12},

1 Pflops = 10^{15} *floating point operations per second*

1. What is the speed of your PC?
2. What is the speed of the fastest computer currently available (and where is it located)
3. What was the speed of the fastest computer in
   - 1995?
   - 2000?
   - 2005?

A little quiz ...

1 Kflops = 10^3, 1 Mflops = 10^6, 1 Gflops = 10^9, 1 Tflops = 10^{12},
1 Pflops = 10^{15} *floating point operations per second*

1. What is the speed of your PC?
   probably between 10 and 100 GFlops

A little quiz ...

1 Kflops = 10^3, 1 Mflops = 10^6, 1 Gflops = 10^9, 1 Tflops = 10^{12},

1 Pflops = 10^{15} *floating point operations per second*

1. What is the speed of your PC?
2. What is the speed of the fastest computer currently available (and where is it located)
3. What was the speed of the fastest computer in
   - 1995?
   - 2000?
   - 2005?

A little quiz ...

1 Kflops = 10^3, 1 Mflops = 10^6, 1 Gflops = 10^9, 1 Tflops = 10^{12},

1 Pflops = 10^{15} *floating point operations per second*

1. What is the speed of your PC?
2. What is the speed of the fastest computer currently available (and where is it located)

   TIANHE-2 (MILKYWAY-2) : NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY: No. 1 system since June 2013, retains its position as the world's No. 1 system with a performance of 33.86 petaflop/s on the Linpack benchmark. 16,000 computer nodes, each two Intel Ivy Bridge Xeon processors and three Xeon Phi chips, counting a total of 3,120,000 cores. Each of the 16,000 nodes possess 88 gigabytes of memory, together approximately 1.34 PiB

A little quiz ...

1 Kflops = 10^3, 1 Mflops = 10^6, 1 Gflops = 10^9, 1 Tflops = 10^{12},

1 Pflops = 10^{15} *floating point operations per second*

1. What is the speed of your PC?
2. What is the speed of the fastest computer currently available (and where is it located)
3. What was the speed of the fastest computer in
   - 1995?
   - 2000?
   - 2005?
   - 2010?
   - 2015?

A little quiz ...

1 Kflops = 10^3, 1 Mflops = 10^6, 1 Gflops = 10^9, 1 Tflops = 10^{12},
1 Pflops = 10^{15} *floating point operations per second*

1. What is the speed of your PC?
2. What is the speed of the fastest computer currently available (and where is it located)
3. What was the speed of the fastest computer in
   - 1995? 0.236 TFlops
   - 2000? 12.3 TFlops
   - 2005? 367 TFlops
   - 2010? 1,759 TFlops

... and how much has the speed of cars/airplanes/... improved in the same time?

additional question: When do you expect that computers exceed 1 ExaFlops?

How fast are computers today?

- 10^6 = 1 MegaFlops: Intel 486

  33MHz PC (~1989)
- 10^9 = 1 GigaFlops: Intel Pentium III

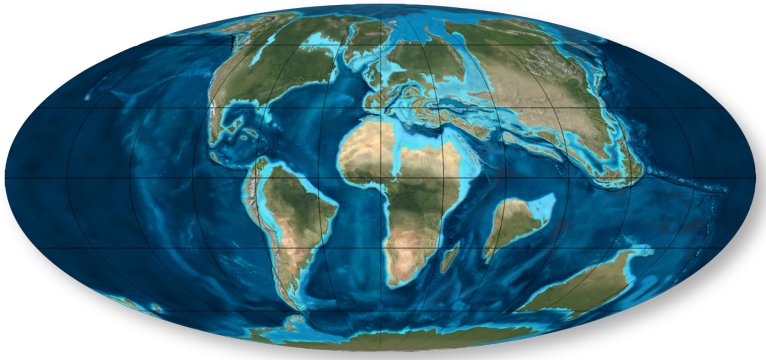
  1GHz (~2000)
  - If every person on earth computes one operation every 7 seconds, all humans together have ~1 GigaFlops performance (less than a current laptop)
- 10^{12}= 1 TeraFlops: HLRB-I

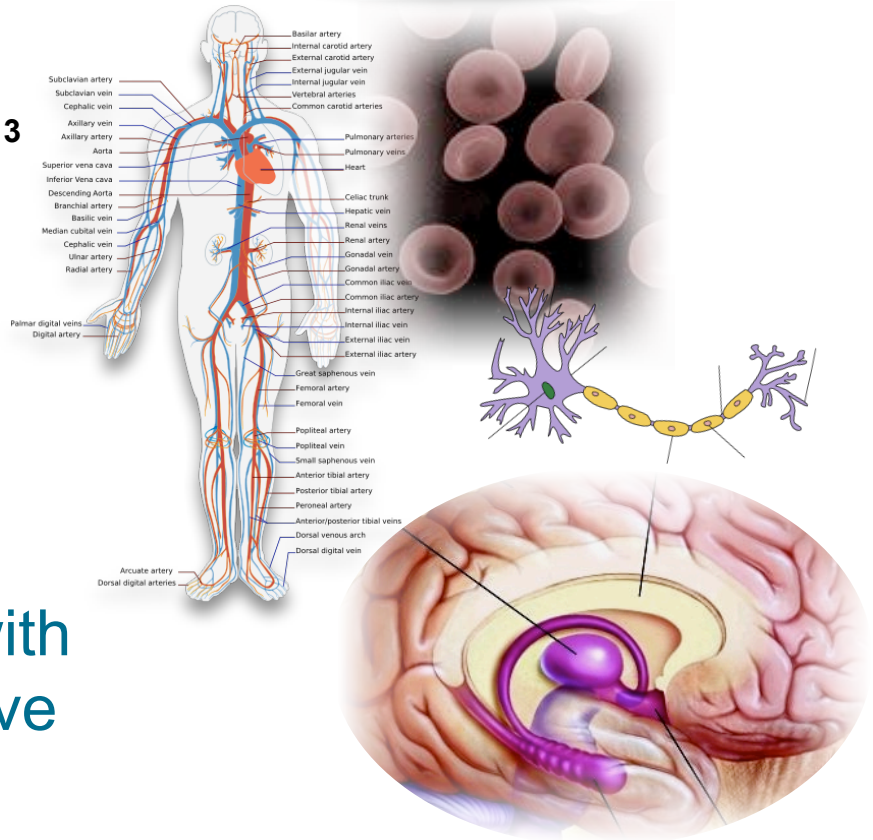
  1344 Proc., ~ 2000
- 10^{15}= 1 PetaFlops
  - 122 400 Cores (Roadrunner, 2008)
  - 294 912 Cores (Jugene, Jülich, 1 PFlops, 1.44 10^{14} Bytes Memory)
  - 155 000 Cores (SuperMuc, 3 PFlops, 3.33 10^{14} Bytes Memory)
- If every person on earth runs a 486 PC, we all together have an aggregate Performance of 7 PetaFlops.
- ExaScale (~10^{18} Flops) around 2020?

Mega=10^6, Giga=10^9, Tera=10^{12}, Peta=10^{15}, Exa=10^{18}

- World has a population of **7×10^9** humans
- Earth is **4.6×10^9** years old
  - the oceans together have ca. **1.3×10^9 km^3**
  - the mantle has **0.91×10^{12} km^3**
    10^{12} finite elements can resolve the volume of the mantle with ca. 1 km resolution
- Number of stars in the galaxy: **10^{11}**
- Avogadro's constant: **6×10^{23} mol^{-1}**
- The recirculatory system contains **2.5×10^{13}** red blood cells
- The *brain* has ca. **10^{11}** Neurons
- Processor chip has
  - **5×10^9** transistors
  - 3 GHz clock rate: **3×10^9**
  - can perform **10^{11}** (= 100 Giga) Flops
  - Supercomputer at 2020: **10^{18}** Flops

An exa-scale computer system with **10^{18}** Flops should suffice to resolve the meso-scale





All the performance that we can get?

All of these are important:

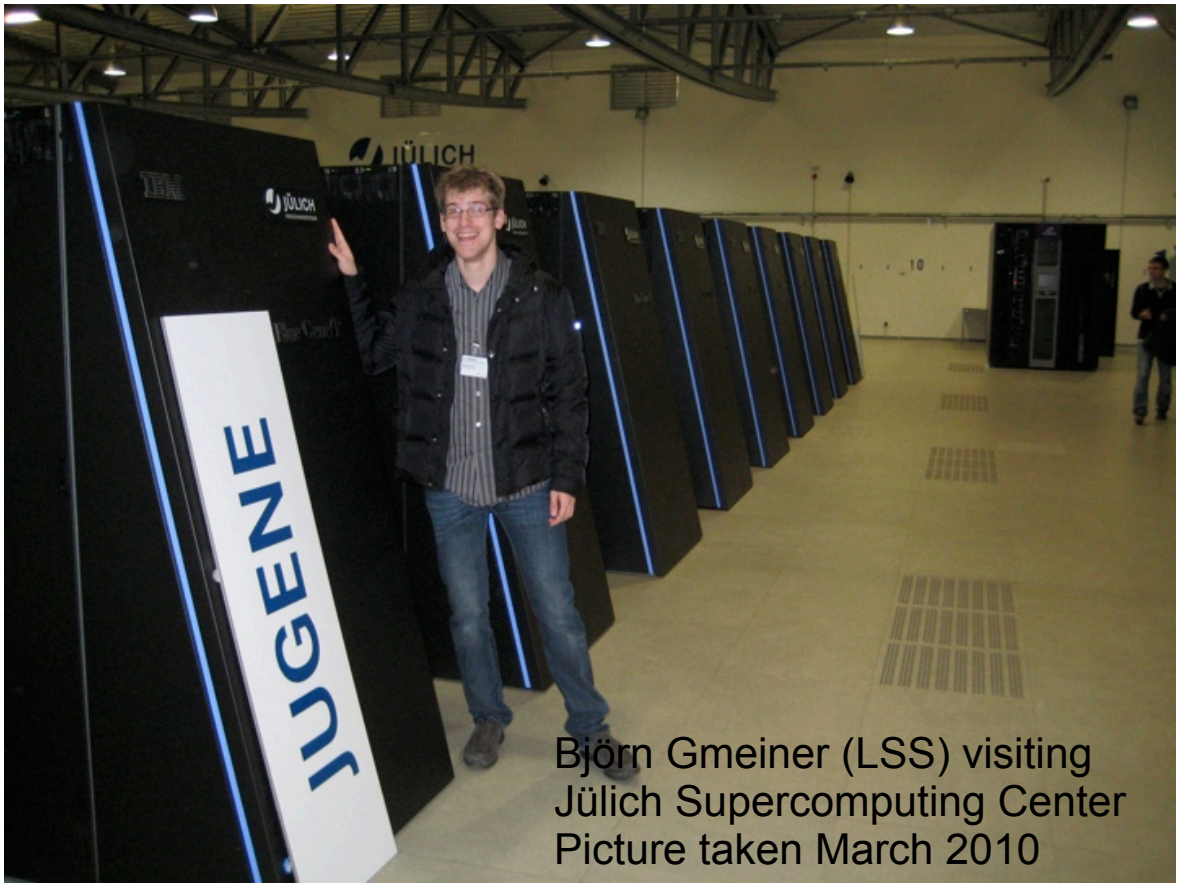
- Use best possible models (reduce model complexity)
- Use best possible discretization, adaptivity, (save grid points)
- Use best possible solver (to save operations)
- Use best possible implementation (to exploit the hardware well)
- Use fastest computer accessible

Jaguar @ Oak Ridge National Lab

Image courtesy of the National Center for Computational Sciences, Oak Ridge National Laboratory

- 1.75 petaflop/s performance speed running the Linpack benchmark.
- theoretical peak capability to 2.3 petaflop/s
- nearly a quarter of a million cores.
- #1 on TOP 500 @ Nov 2009

Jugene @ Jülich



- 0.825 petaflop/s performance speed running the Linpack benchmark.
- theoretical peak capability to 1.0027 petaflop/s
- 294912 cores.
- #4 on TOP 500 List Nov 2009