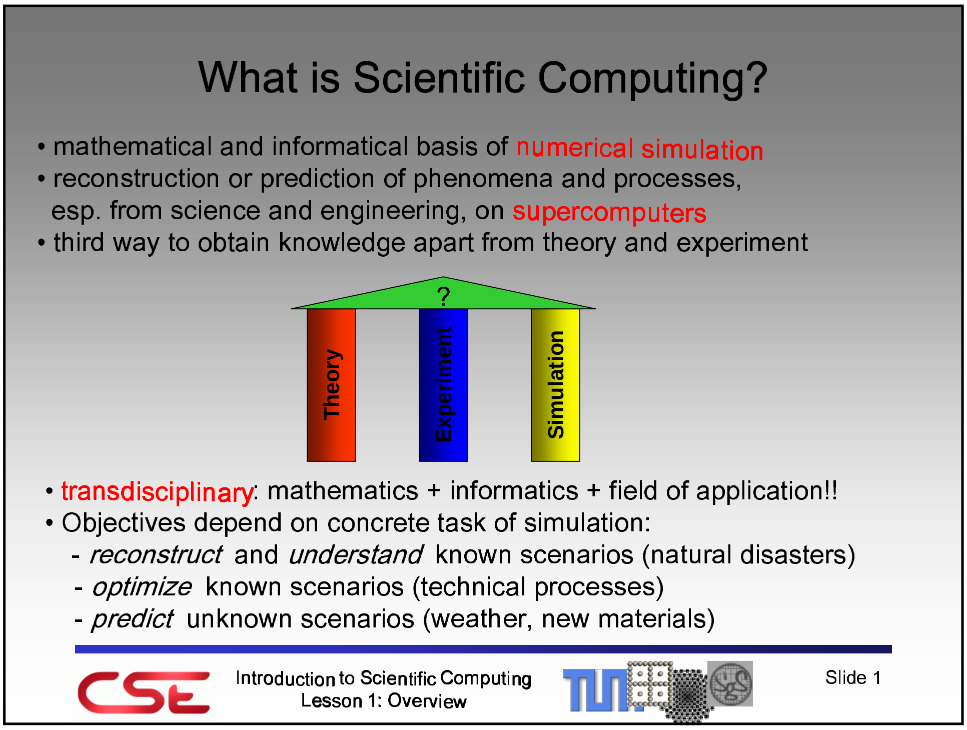
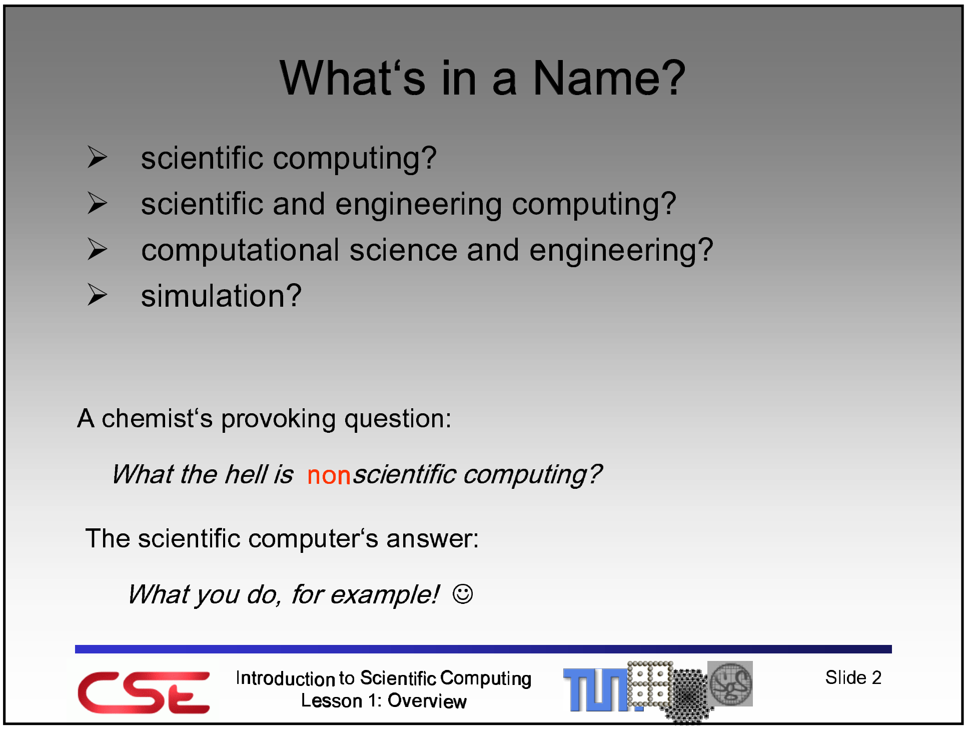


What is Scientific Computing?

- mathematical and informational basis of numerical simulation
- reconstruction or prediction of phenomena and processes, esp. from science and engineering, on supercomputers
- third way to obtain knowledge apart from theory and experiment

?
Theory
Experiment
Simulation

- transdisciplinary: mathematics + informatics + field of application!!
- Objectives depend on concrete task of simulation:
  - *reconstruct* and *understand* known scenarios (natural disasters)
  - *optimize* known scenarios (technical processes)
  - *predict* unknown scenarios (weather, new materials)



What‘s in a Name?

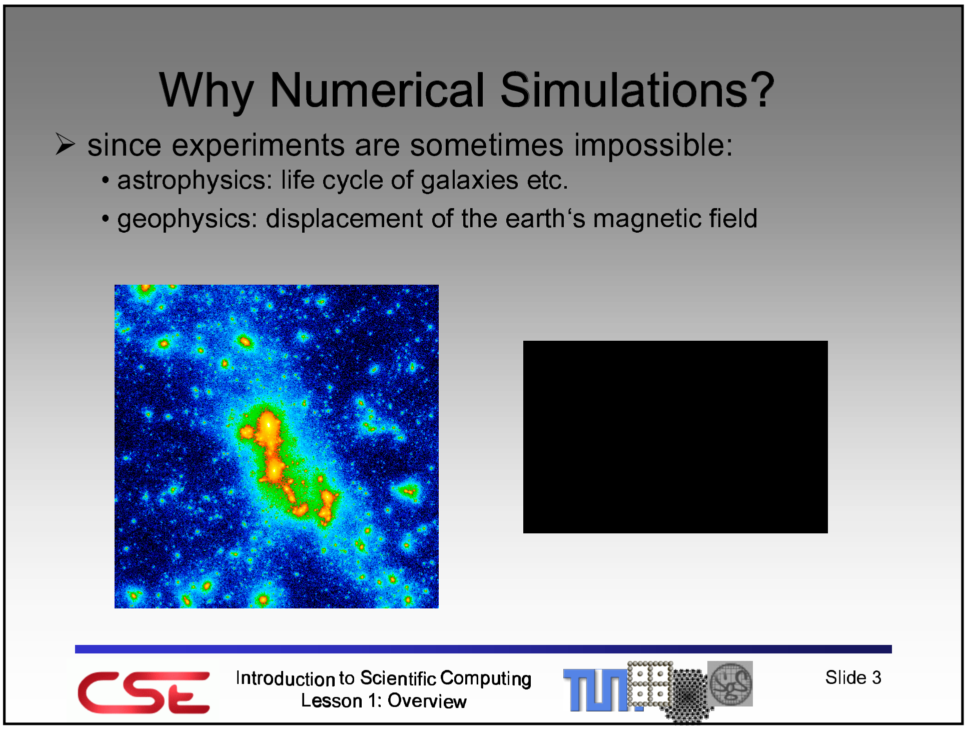
- scientific computing?
- scientific and engineering computing?
- computational science and engineering?
- simulation?

A chemist‘s provoking question:

What the hell is non*scientific computing?*

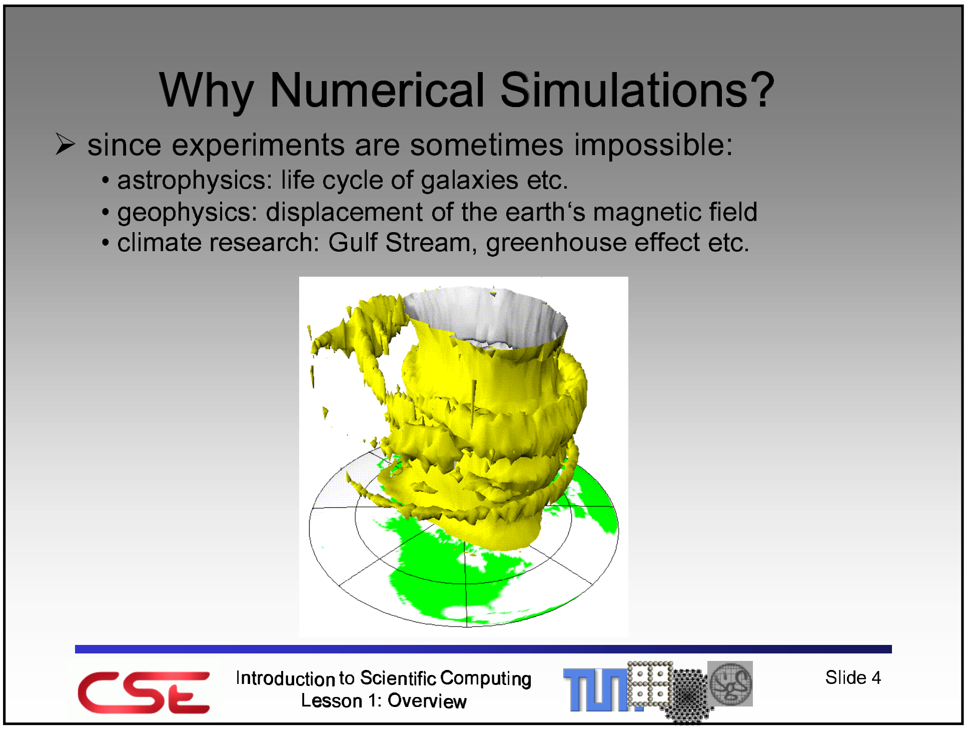
The scientific computer‘s answer:

What you do, for example! ☺



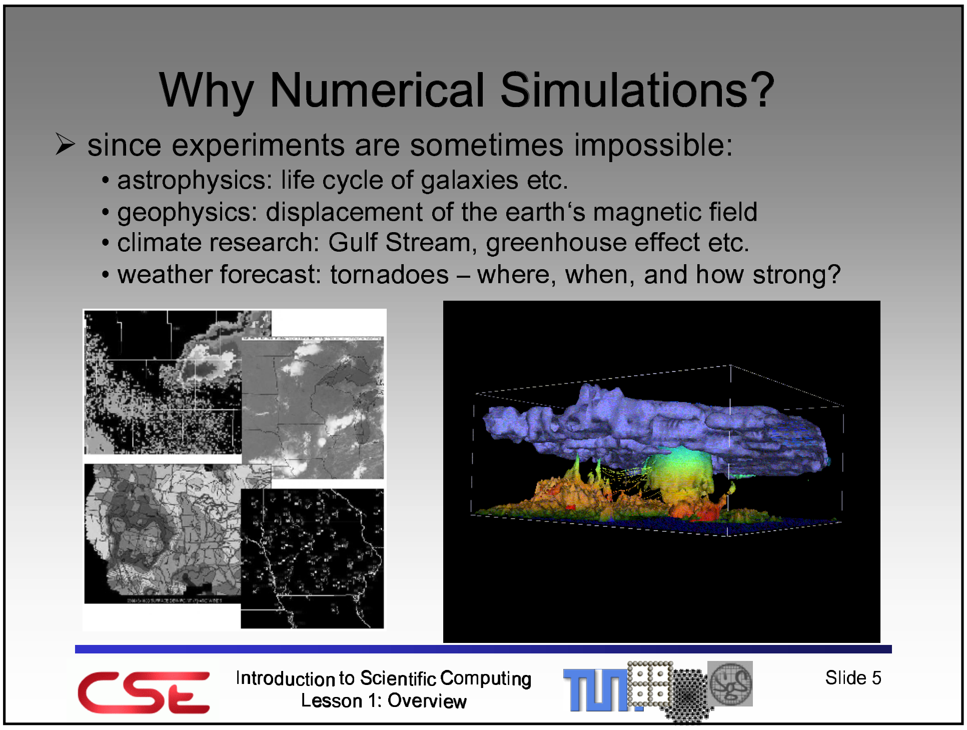
Why Numerical Simulations?

- since experiments are sometimes impossible:
  - astrophysics: life cycle of galaxies etc.
  - geophysics: displacement of the earth‘s magnetic field



Why Numerical Simulations?

- since experiments are sometimes impossible:
  - astrophysics: life cycle of galaxies etc.
  - geophysics: displacement of the earth‘s magnetic field
  - climate research: Gulf Stream, greenhouse effect etc.

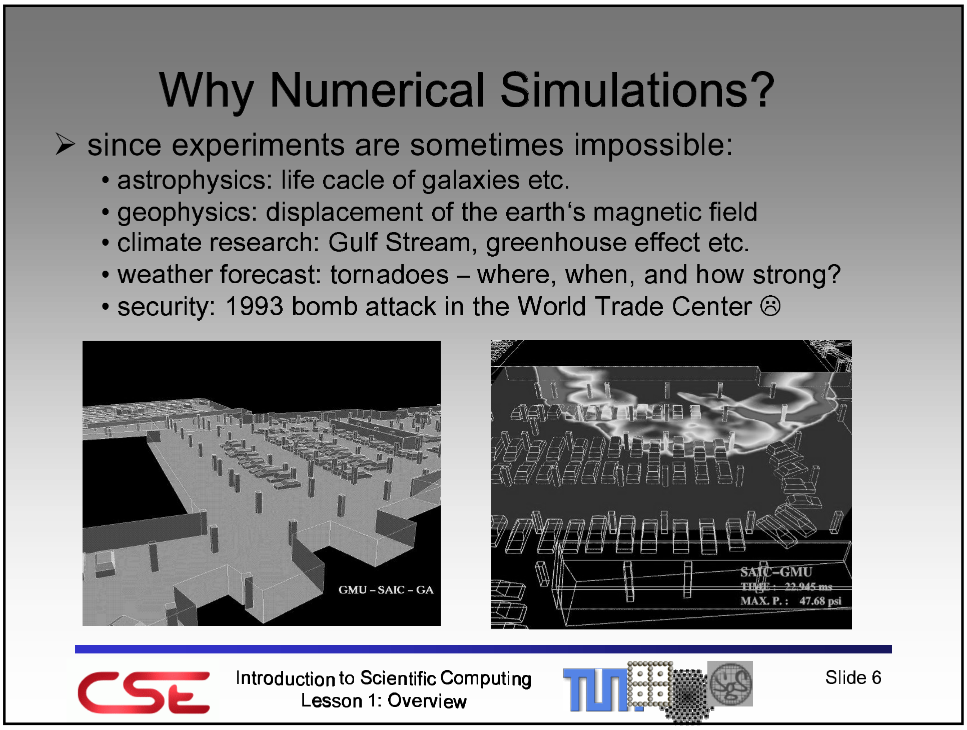


Why Numerical Simulations?

- since experiments are sometimes impossible:
  - astrophysics: life cycle of galaxies etc.
  - geophysics: displacement of the earth‘s magnetic field
  - climate research: Gulf Stream, greenhouse effect etc.
  - weather forecast: tornadoes – where, when, and how strong?

CSE Introduction to Scientific Computing
Lesson 1: Overview

Slide 5



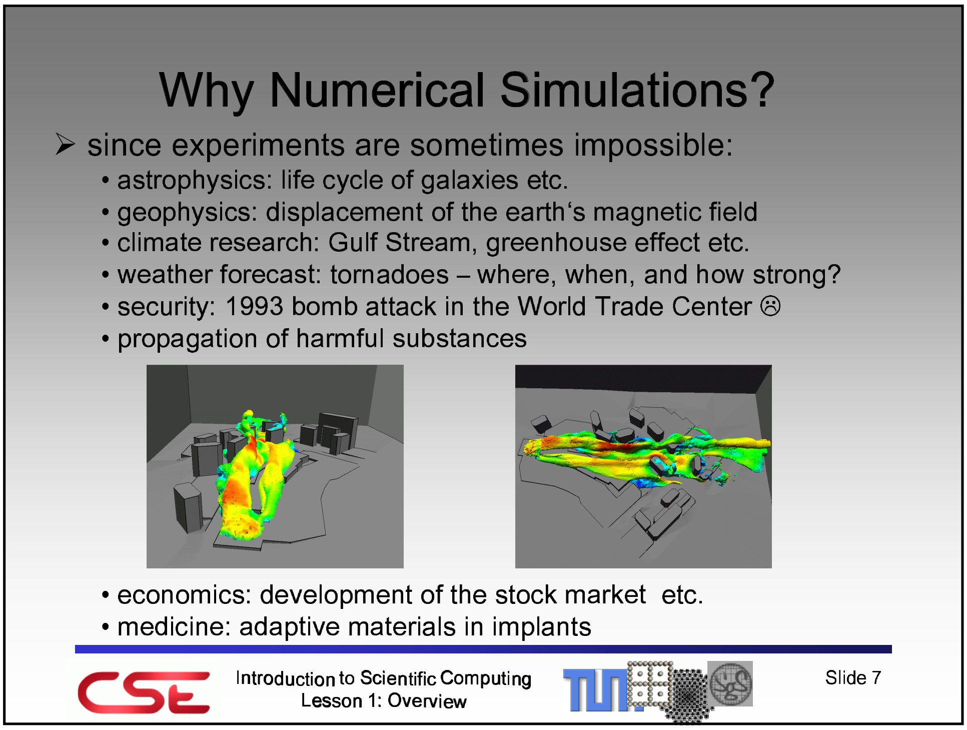
Why Numerical Simulations?

- since experiments are sometimes impossible:
  - astrophysics: life cacle of galaxies etc.
  - geophysics: displacement of the earth‘s magnetic field
  - climate research: Gulf Stream, greenhouse effect etc.
  - weather forecast: tornadoes – where, when, and how strong?
  - security: 1993 bomb attack in the World Trade Center ☹

GMU – SAIC – GA
SAIC–GMU
TIME : 22.945 ms
MAX. P. : 47.68 psi

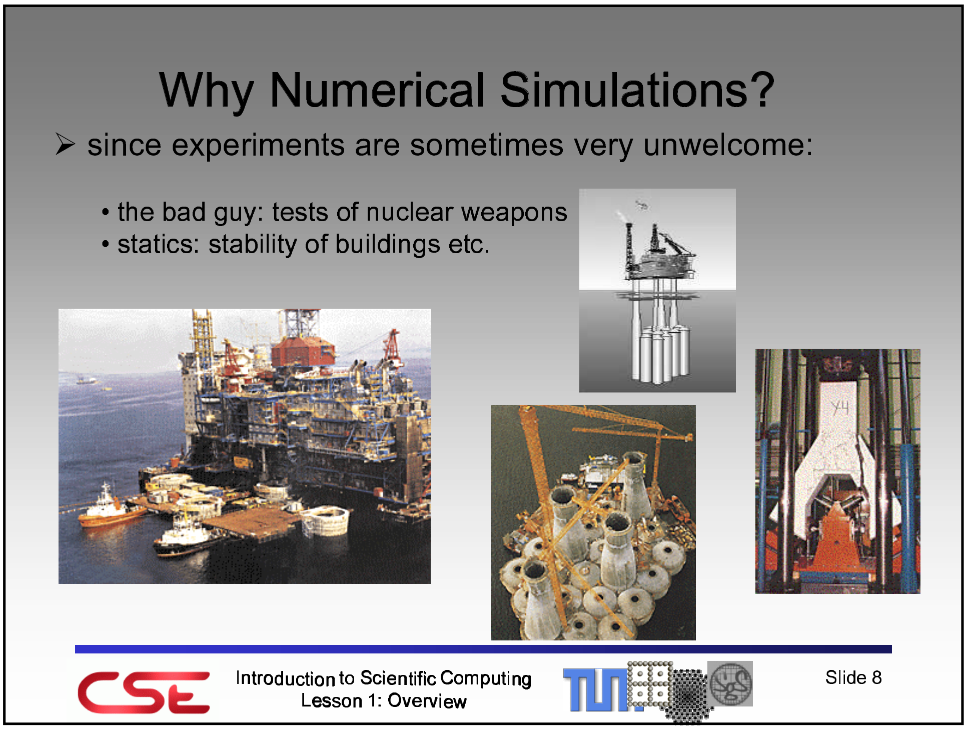
CSE Introduction to Scientific Computing
Lesson 1: Overview

Slide 6



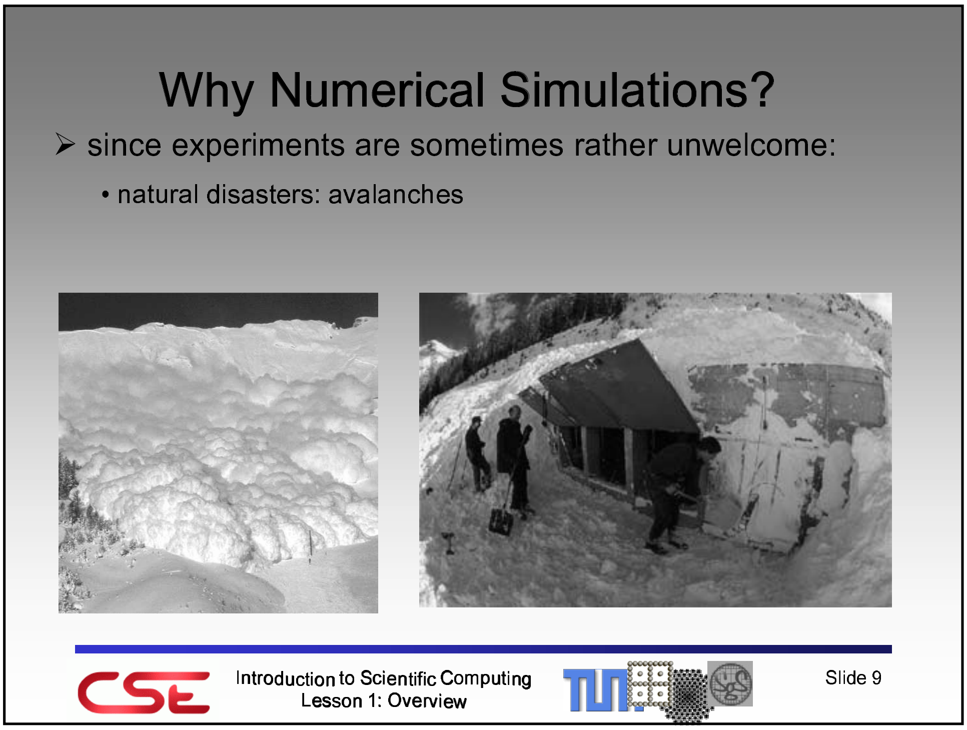
Why Numerical Simulations?

- since experiments are sometimes impossible:
  - astrophysics: life cycle of galaxies etc.
  - geophysics: displacement of the earth‘s magnetic field
  - climate research: Gulf Stream, greenhouse effect etc.
  - weather forecast: tornadoes – where, when, and how strong?
  - security: 1993 bomb attack in the World Trade Center ☹
  - propagation of harmful substances
  - economics: development of the stock market etc.
  - medicine: adaptive materials in implants



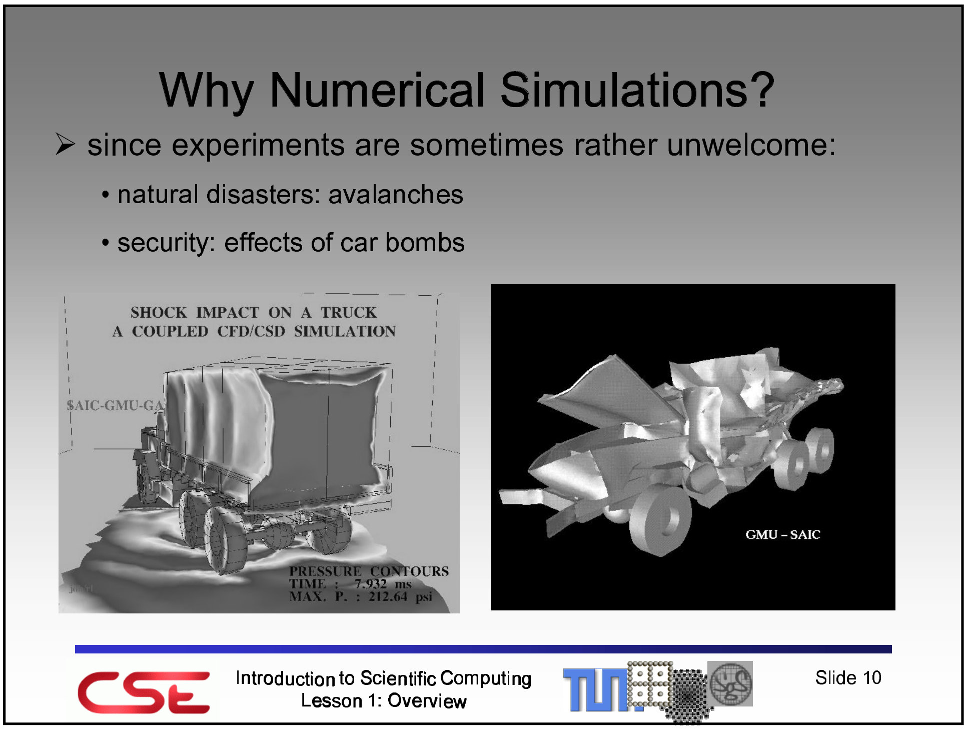
Why Numerical Simulations?

- since experiments are sometimes very unwelcome:
  - the bad guy: tests of nuclear weapons
  - statics: stability of buildings etc.



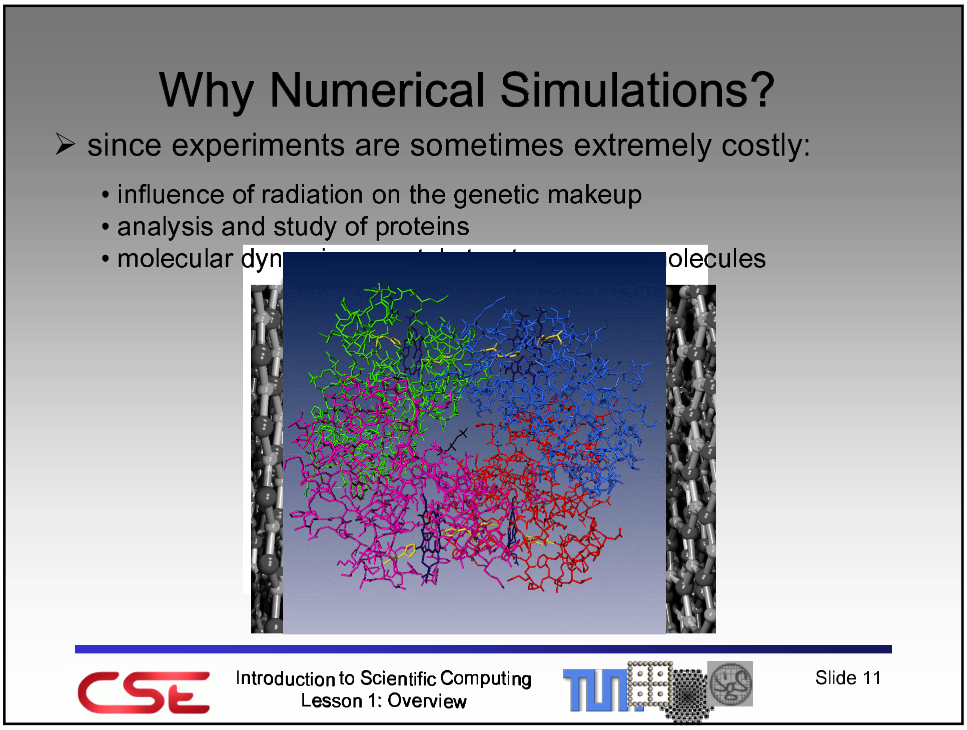
Why Numerical Simulations?

- since experiments are sometimes rather unwelcome:
  - natural disasters: avalanches



Why Numerical Simulations?

- since experiments are sometimes rather unwelcome:
  - natural disasters: avalanches
  - security: effects of car bombs

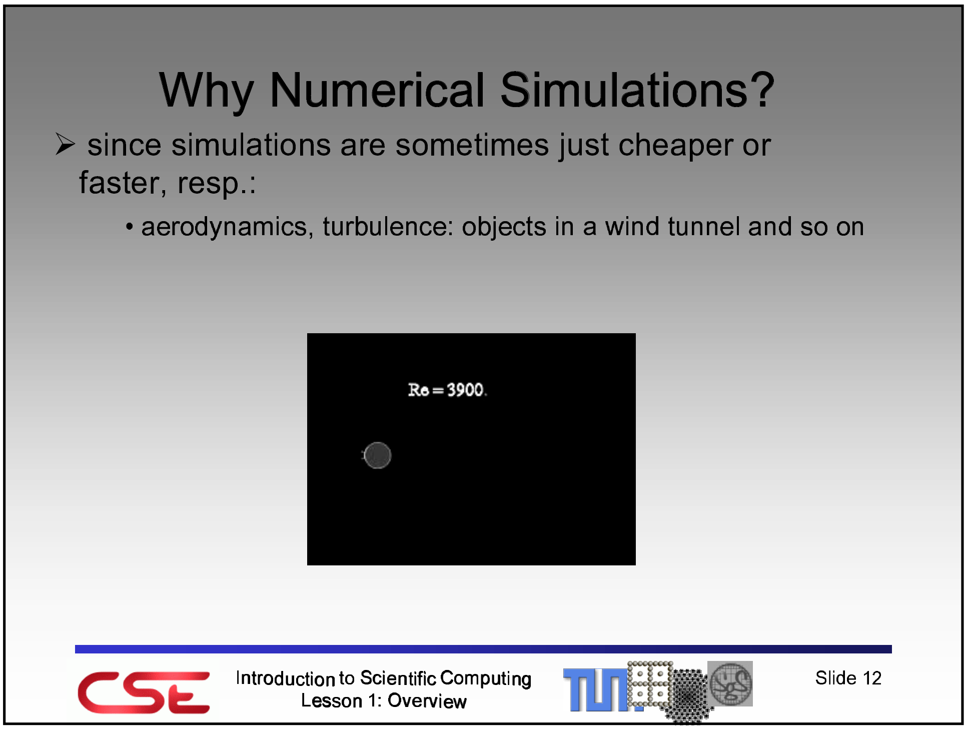


Why Numerical Simulations?

- since experiments are sometimes extremely costly:
  - influence of radiation on the genetic makeup
  - analysis and study of proteins
  - molecular dyn olecules

Introduction to Scientific Computing
Lesson 1: Overview

Slide 11

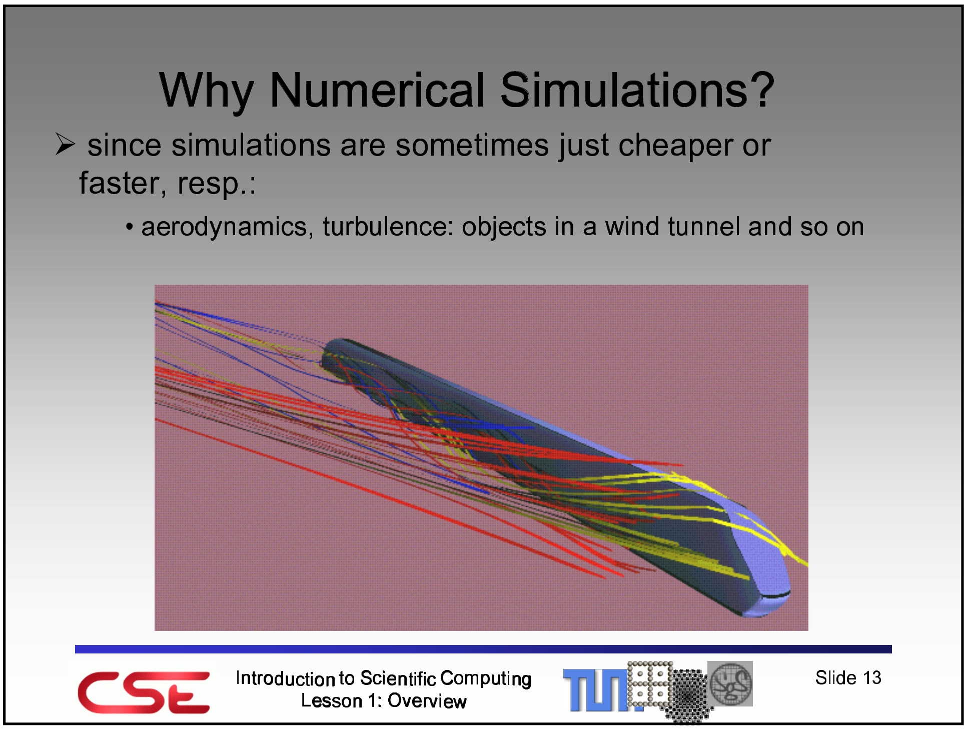


Why Numerical Simulations?

- since simulations are sometimes just cheaper or faster, resp.:
  - aerodynamics, turbulence: objects in a wind tunnel and so on

Introduction to Scientific Computing
Lesson 1: Overview

Slide 12

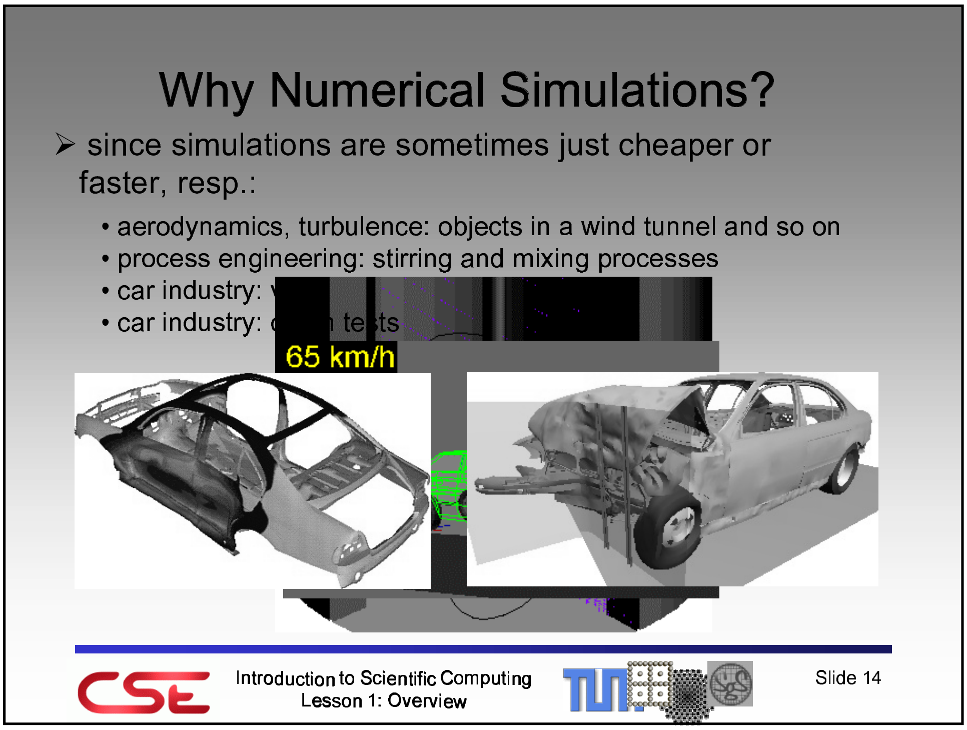


Why Numerical Simulations?

- since simulations are sometimes just cheaper or faster, resp.:
  - aerodynamics, turbulence: objects in a wind tunnel and so on

Introduction to Scientific Computing
Lesson 1: Overview

Slide 13

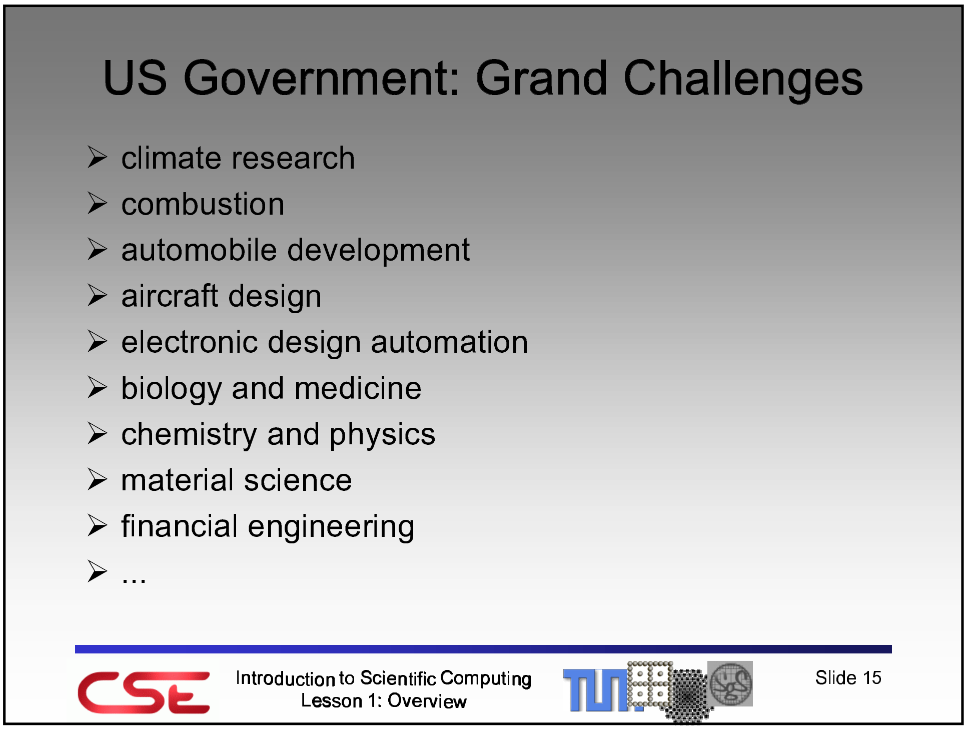


Why Numerical Simulations?

- since simulations are sometimes just cheaper or faster, resp.:
  - aerodynamics, turbulence: objects in a wind tunnel and so on
  - process engineering: stirring and mixing processes
  - car industry: v[illegible]
  - car industry: c[illegible]n te[illegible]ts

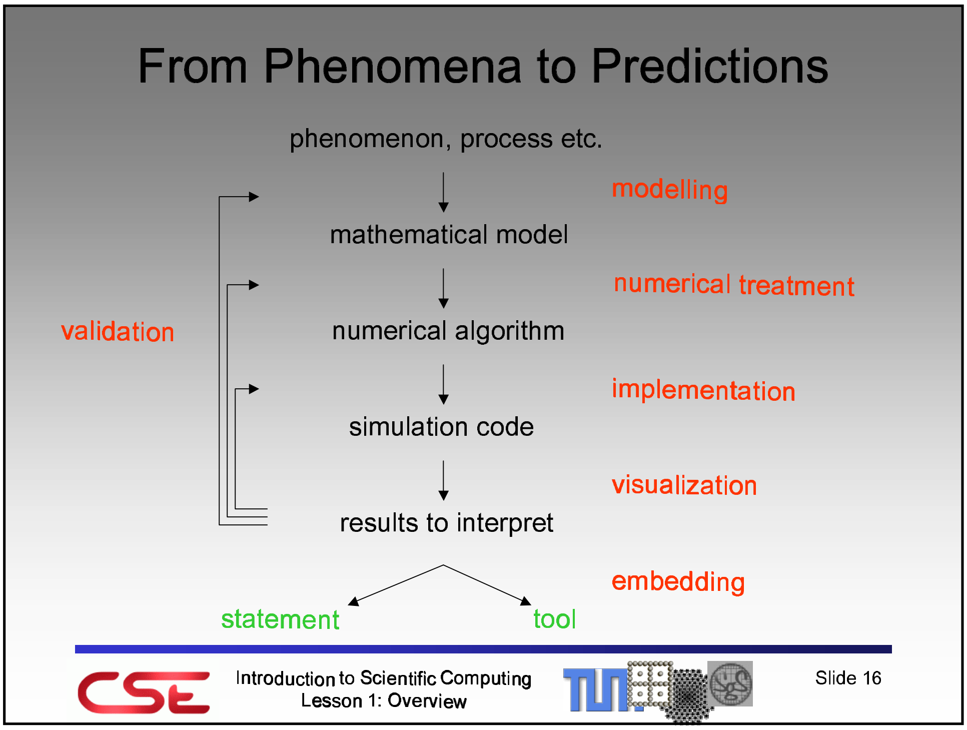
Introduction to Scientific Computing
Lesson 1: Overview

Slide 14



US Government: Grand Challenges

- climate research
- combustion
- automobile development
- aircraft design
- electronic design automation
- biology and medicine
- chemistry and physics
- material science
- financial engineering
- ...



From Phenomena to Predictions

phenomenon, process etc.

modelling

mathematical model

numerical treatment

validation

numerical algorithm

implementation

simulation code

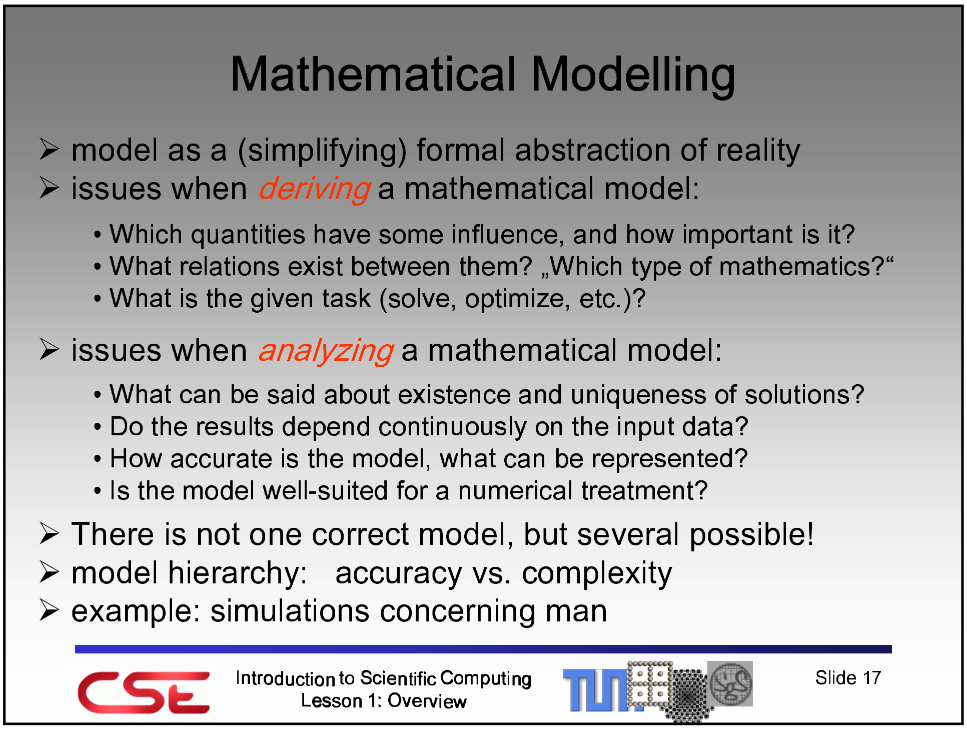
visualization

results to interpret

embedding

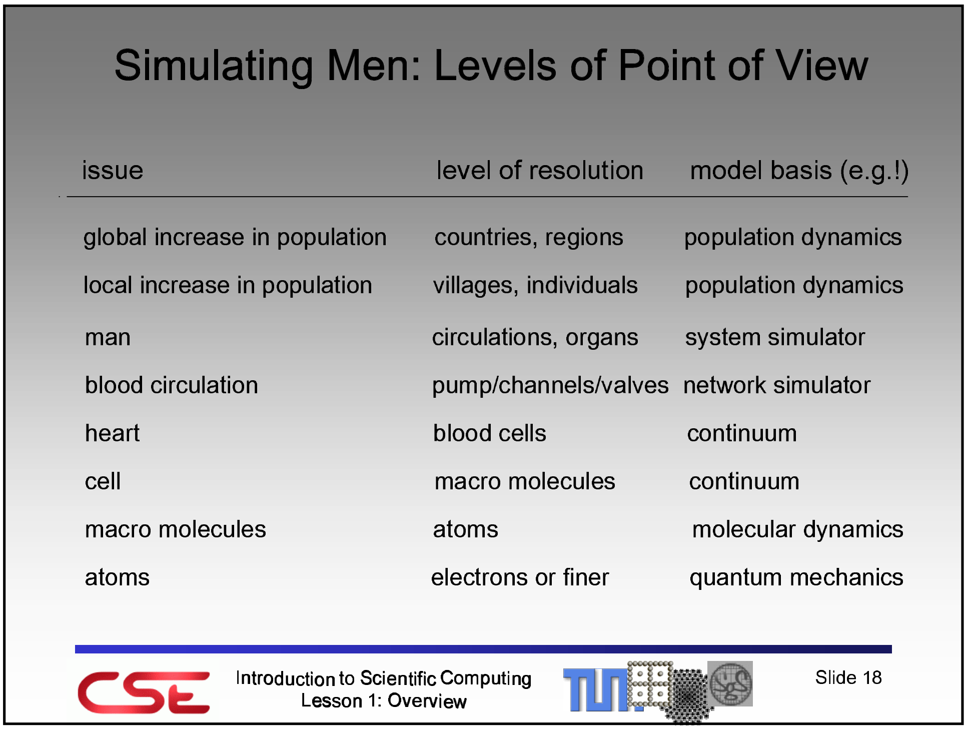
statement

tool



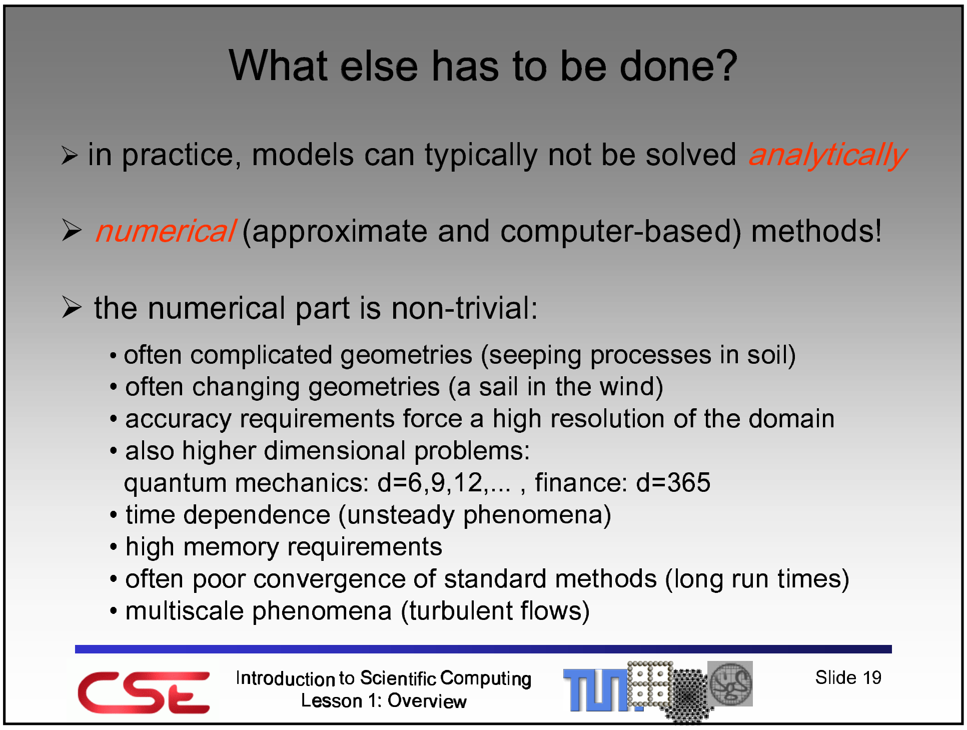
Mathematical Modelling

- model as a (simplifying) formal abstraction of reality
- issues when *deriving* a mathematical model:
  - Which quantities have some influence, and how important is it?
  - What relations exist between them? „Which type of mathematics?“
  - What is the given task (solve, optimize, etc.)?
- issues when *analyzing* a mathematical model:
  - What can be said about existence and uniqueness of solutions?
  - Do the results depend continuously on the input data?
  - How accurate is the model, what can be represented?
  - Is the model well-suited for a numerical treatment?
- There is not one correct model, but several possible!
- model hierarchy: accuracy vs. complexity
- example: simulations concerning man



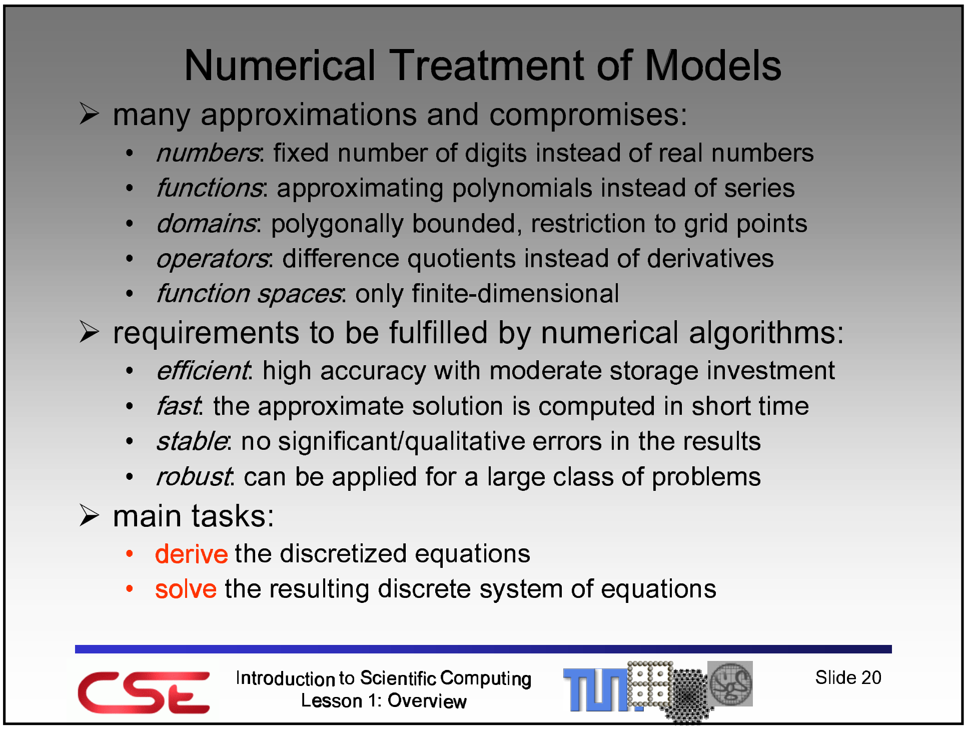
Simulating Men: Levels of Point of View

issue	level of resolution	model basis (e.g.!)
global increase in population	countries, regions	population dynamics
local increase in population	villages, individuals	population dynamics
man	circulations, organs	system simulator
blood circulation	pump/channels/valves	network simulator
heart	blood cells	continuum
cell	macro molecules	continuum
macro molecules	atoms	molecular dynamics
atoms	electrons or finer	quantum mechanics



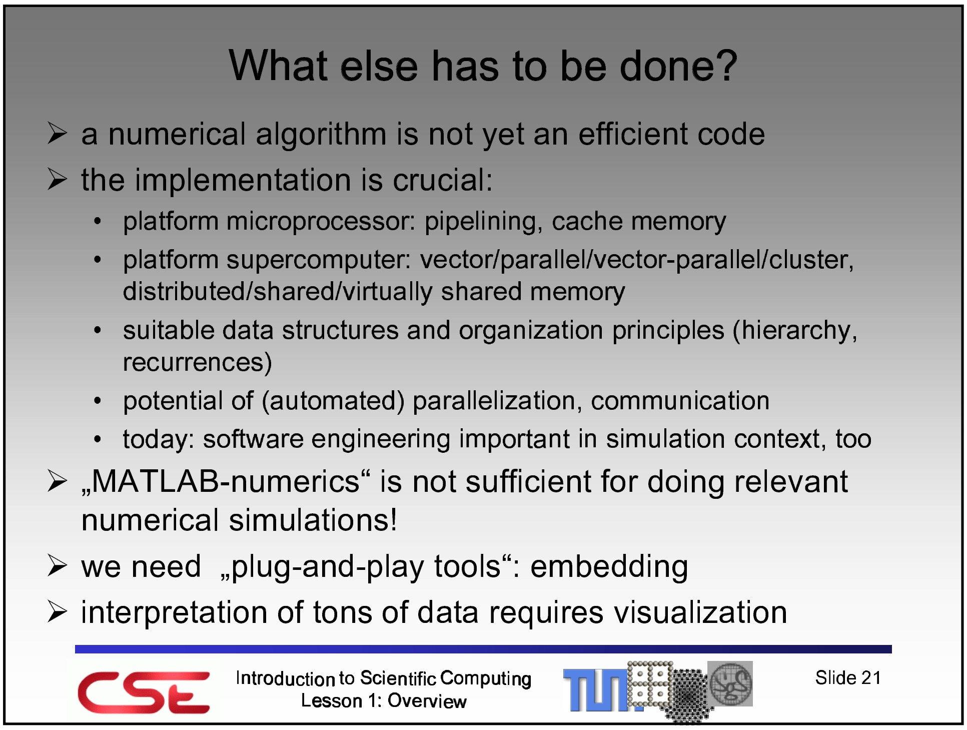
What else has to be done?

- in practice, models can typically not be solved *analytically*
- *numerical* (approximate and computer-based) methods!
- the numerical part is non-trivial:
  - often complicated geometries (seeping processes in soil)
  - often changing geometries (a sail in the wind)
  - accuracy requirements force a high resolution of the domain
  - also higher dimensional problems:
    quantum mechanics: d=6,9,12,... , finance: d=365
  - time dependence (unsteady phenomena)
  - high memory requirements
  - often poor convergence of standard methods (long run times)
  - multiscale phenomena (turbulent flows)



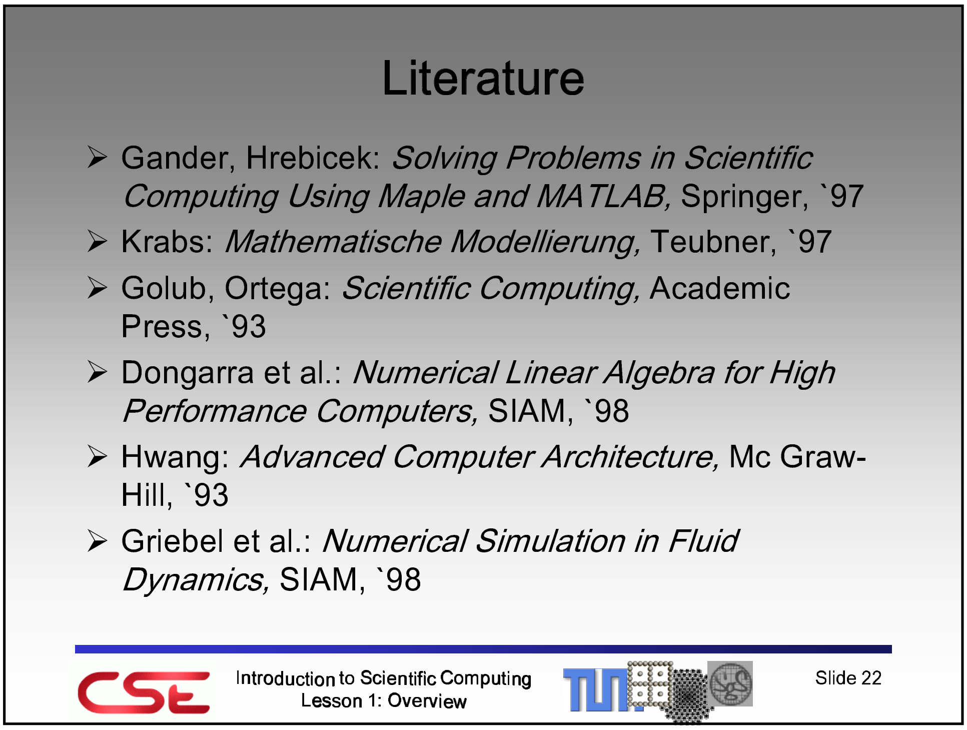
Numerical Treatment of Models

- many approximations and compromises:
  - *numbers*: fixed number of digits instead of real numbers
  - *functions*: approximating polynomials instead of series
  - *domains*: polygonally bounded, restriction to grid points
  - *operators*: difference quotients instead of derivatives
  - *function spaces*: only finite-dimensional
- requirements to be fulfilled by numerical algorithms:
  - *efficient*: high accuracy with moderate storage investment
  - *fast*: the approximate solution is computed in short time
  - *stable*: no significant/qualitative errors in the results
  - *robust*: can be applied for a large class of problems
- main tasks:
  - derive the discretized equations
  - solve the resulting discrete system of equations



What else has to be done?

- a numerical algorithm is not yet an efficient code
- the implementation is crucial:
  - platform microprocessor: pipelining, cache memory
  - platform supercomputer: vector/parallel/vector-parallel/cluster, distributed/shared/virtually shared memory
  - suitable data structures and organization principles (hierarchy, recurrences)
  - potential of (automated) parallelization, communication
  - today: software engineering important in simulation context, too
- „MATLAB-numerics“ is not sufficient for doing relevant numerical simulations!
- we need „plug-and-play tools“: embedding
- interpretation of tons of data requires visualization



Literature

- Gander, Hrebicek: *Solving Problems in Scientific Computing Using Maple and MATLAB,* Springer, `97
- Krabs: *Mathematische Modellierung,* Teubner, `97
- Golub, Ortega: *Scientific Computing,* Academic Press, `93
- Dongarra et al.: *Numerical Linear Algebra for High Performance Computers,* SIAM, `98
- Hwang: *Advanced Computer Architecture,* Mc Graw-Hill, `93
- Griebel et al.: *Numerical Simulation in Fluid Dynamics,* SIAM, `98