U.C. Berkeley
Computational Nanoscience
NSE C242 & Phys C203
Spring 2008

Lecture 1: Introduction

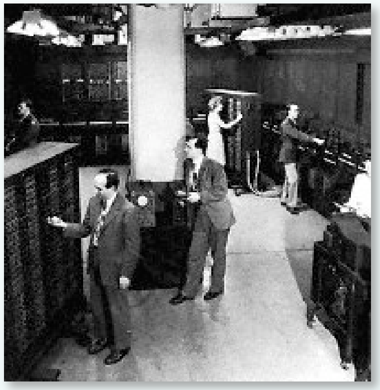
January 22, 2008

Jeffrey C. Grossman

Elif Ertekin

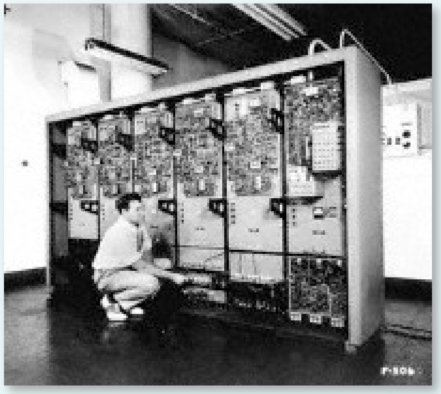
Computational Science: Historical Perspective

Computer Hardware Historical Milestones



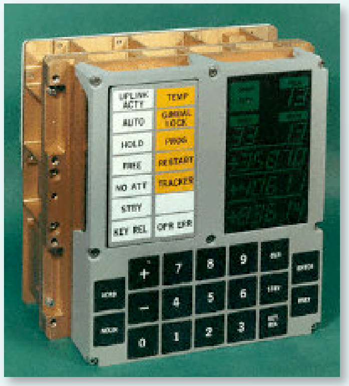
1946: Eniac
Op/s: 5000
Sq. ft: 3000

1952: IBM SSEC
Op/s: 2000
Sq. ft: 1000

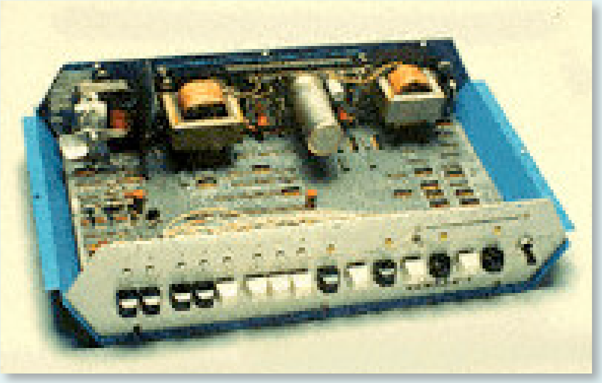


1951: MIT Whirlwind
Op/s: 200,000
Sq. ft: 3100

1964: CDC 6600
Op/s: 3,000,000
Sq. ft: 3100



1968: Apollo Guide
Apollo 7&11 missions



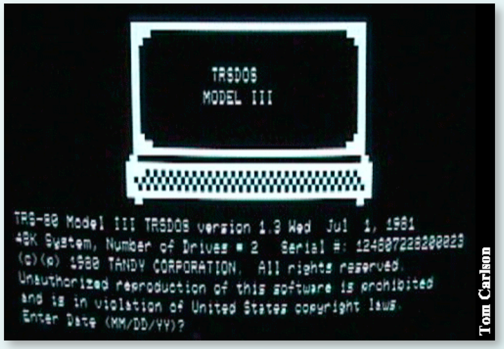
1971: Kenbak-1
First personal computer
256 Bytes of memory

1974: Xerox Alto
Built-in mouse
Connect to network

JCG Personal Computer History

Tandy TRS-80 (a.k.a. “Trash-80”)

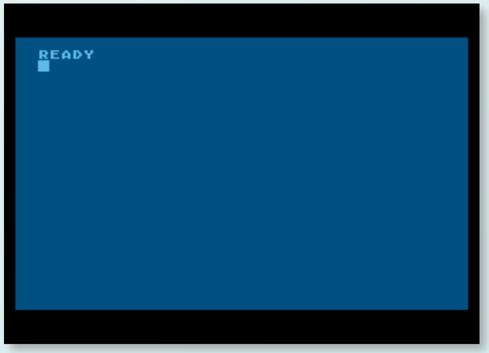
Atari 400 (note the stylish keyboard)



```
TRSDOS
MODEL III

TRS-80 Model III TRSDOS version 1.3 Wed  Jul  1, 1981
48K System, Number of Drives = 2   Serial #: 124607228200023
(c)(p) 1980 TANDY CORPORATION.  All rights reserved.
Unauthorized reproduction of this software is prohibited
and is in violation of United States copyright laws.
Enter Date (MM/DD/YY)?
```

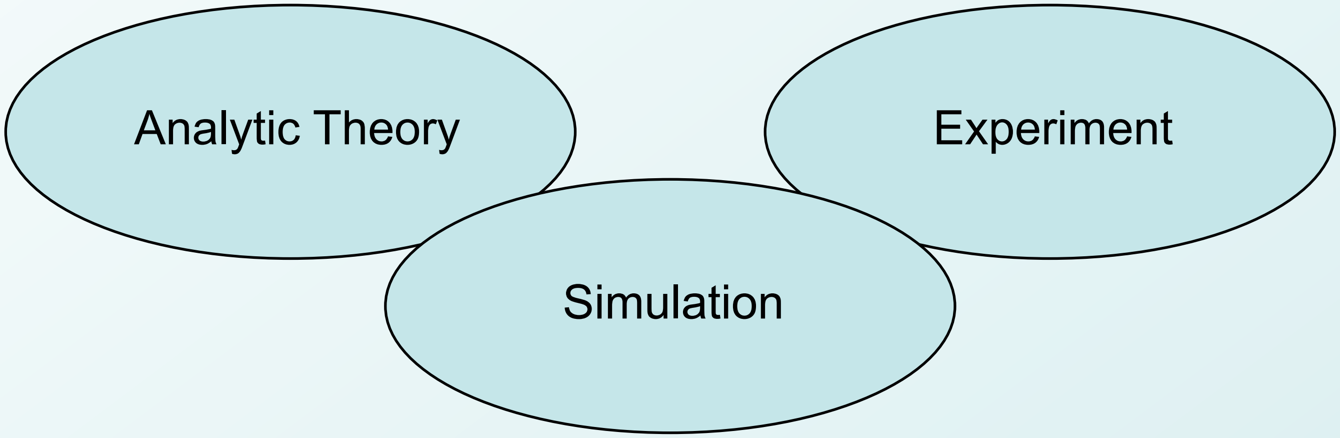
Start-up screens



```
READY
```

Computational Science

The Third Branch of Science – along with Analytic Theory and Experiment



Modeling and simulation can be thought of as “experimental theory” or “theoretical experiments”

The field is still in its infancy – 40 years vs. > 400 years of experimental sciences

Computational Science is not Computer Science

Scientific Computing Milestones

So…what has computational science done for us thus far?

- Many early engineering and control examples, used by NASA (Apollo missions, 1960's), the Census Bureau (1970's), and even election tabulation
  - Note that these are not very "scientific" in nature

- Early applications that solved equations numerically in order to probe new science include:
  - Enrico Fermi's Monte Carlo calculation of neutron diffusion (1930's)
    - 12 "hard sphere" atoms
  - Hydrodynamics simulations required for the Manhattan Project (1940's)
  - Molecular dynamics simulations for materials science (1960's) and later (1970's) for protein structure
  - Quantum Monte Carlo calculation of electron gas density (1980) - became the basis for density functional theory calculations

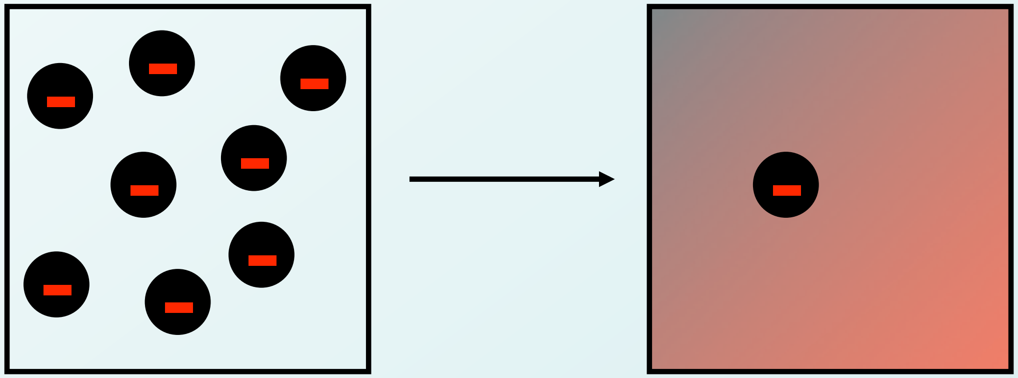
Density functional theory

Energy ⟷ Electron density

$$E_0 = E[n_0]$$

Hohenberg & Kohn, 1964

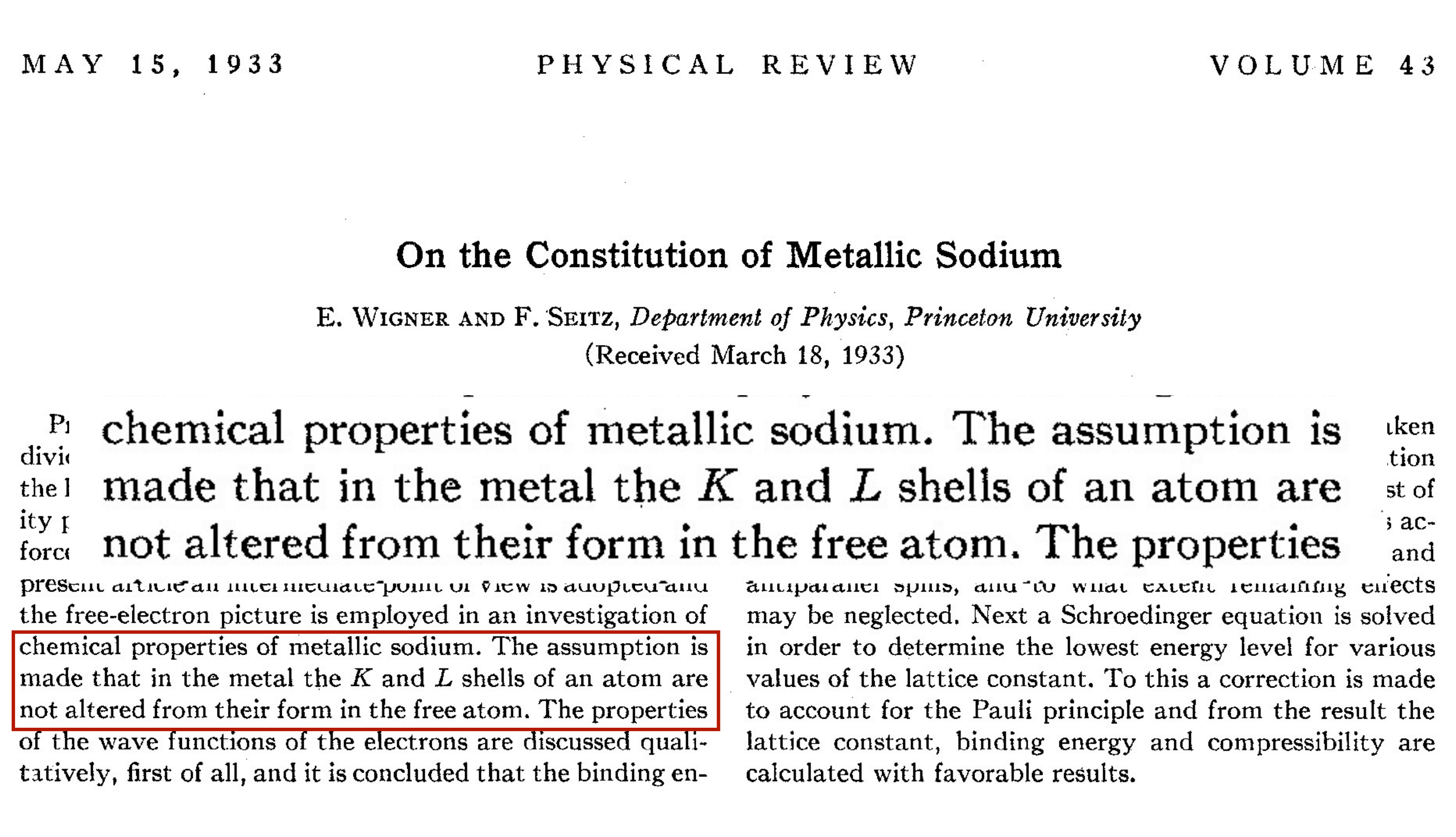
Interacting → Non-interacting



Kohn & Sham, 1965

Walter Kohn (left), receiving the Nobel prize in chemistry in 1998.

The first ‘first-principles’ calculation



MAY 15, 1933 PHYSICAL REVIEW VOLUME 43

On the Constitution of Metallic Sodium

E. Wigner and F. Seitz, *Department of Physics, Princeton University*

(Received March 18, 1933)

chemical properties of metallic sodium. The assumption is made that in the metal the K and L shells of an atom are not altered from their form in the free atom. The properties

Pr
divi
the l
ity p
forc
present article an intermediate point of view is adopted and the free-electron picture is employed in an investigation of chemical properties of metallic sodium. The assumption is made that in the metal the K and L shells of an atom are not altered from their form in the free atom. The properties of the wave functions of the electrons are discussed qualitatively, first of all, and it is concluded that the binding en-

ken
tion
st of
; ac-
and
antiparallel spins, and to what extent remaining effects may be neglected. Next a Schroedinger equation is solved in order to determine the lowest energy level for various values of the lattice constant. To this a correction is made to account for the Pauli principle and from the result the lattice constant, binding energy and compressibility are calculated with favorable results.

Scientific Computing State-of-the-Art

And what can computational science do for us now?

Computational:

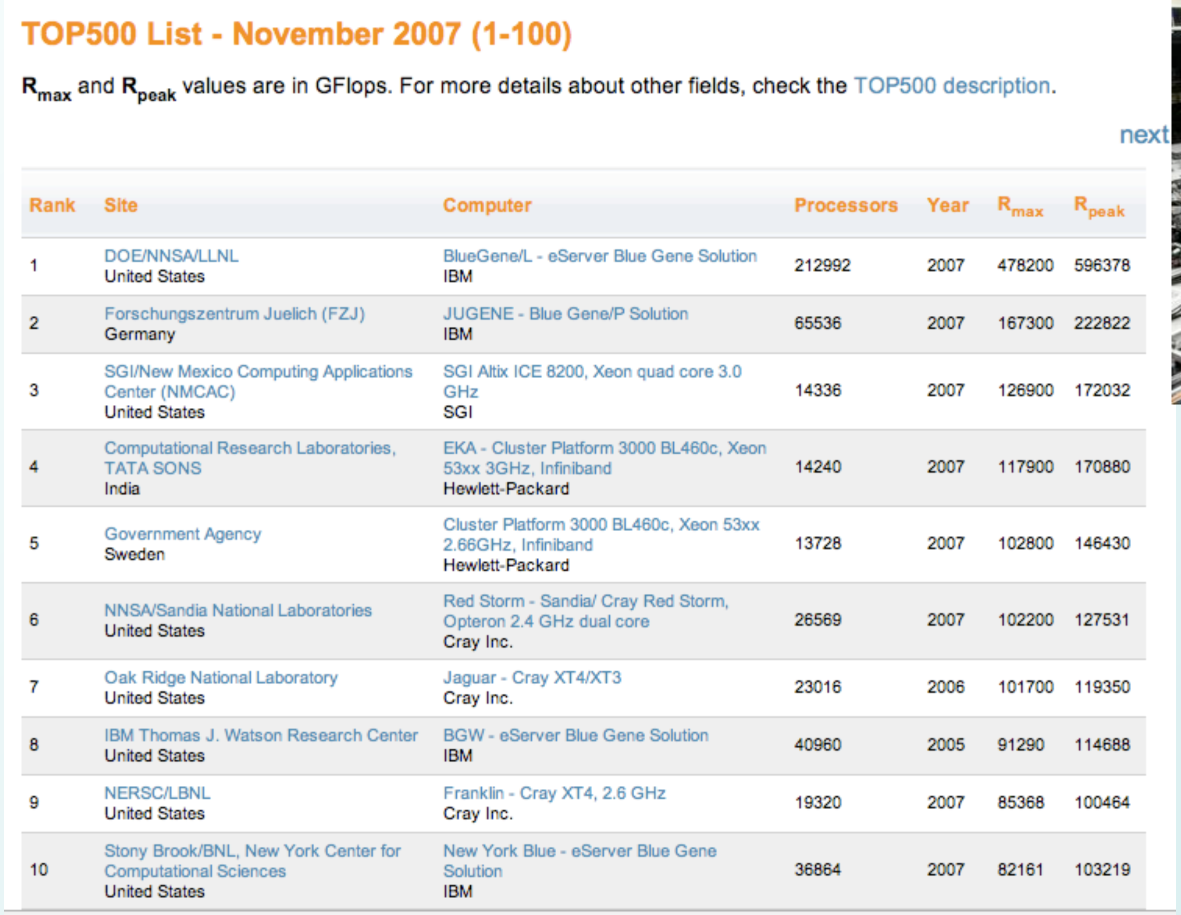
- Physics
- Chemistry
- Biology
- Cosmology
- Climate Modeling
- …and much more

Depending on what is needed to solve the problem, we can now simulate:

- the (almost) exact quantum behavior of a thousand atoms
- the approximate quantum behavior of billions of atoms
- 3-dimensional, transient, turbulent reacting combustive flows
- global climate with 50 km resolution for next 200 years
- protein folding geometries during entire folding process

Top Supercomputers Available Today

from http://www.top500.org/list/2007/11/100:



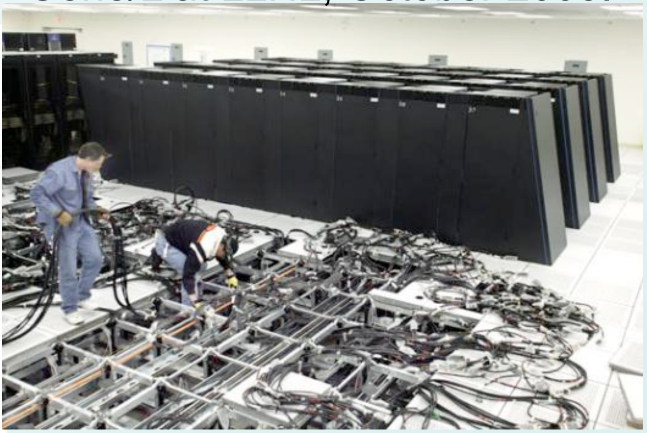
TOP500 List - November 2007 (1-100)

R_{max} and R_{peak} values are in GFlops. For more details about other fields, check the TOP500 description.

next

Rank	Site	Computer	Processors	Year	R_{max}	R_{peak}
1	DOE/NNSA/LLNL United States	BlueGene/L - eServer Blue Gene Solution IBM	212992	2007	478200	596378
2	Forschungszentrum Juelich (FZJ) Germany	JUGENE - Blue Gene/P Solution IBM	65536	2007	167300	222822
3	SGI/New Mexico Computing Applications Center (NMCAC) United States	SGI Altix ICE 8200, Xeon quad core 3.0 GHz SGI	14336	2007	126900	172032
4	Computational Research Laboratories, TATA SONS India	EKA - Cluster Platform 3000 BL460c, Xeon 53xx 3GHz, Infiniband Hewlett-Packard	14240	2007	117900	170880
5	Government Agency Sweden	Cluster Platform 3000 BL460c, Xeon 53xx 2.66GHz, Infiniband Hewlett-Packard	13728	2007	102800	146430
6	NNSA/Sandia National Laboratories United States	Red Storm - Sandia/ Cray Red Storm, Opteron 2.4 GHz dual core Cray Inc.	26569	2007	102200	127531
7	Oak Ridge National Laboratory United States	Jaguar - Cray XT4/XT3 Cray Inc.	23016	2006	101700	119350
8	IBM Thomas J. Watson Research Center United States	BGW - eServer Blue Gene Solution IBM	40960	2005	91290	114688
9	NERSC/LBNL United States	Franklin - Cray XT4, 2.6 GHz Cray Inc.	19320	2007	85368	100464
10	Stony Brook/BNL, New York Center for Computational Sciences United States	New York Blue - eServer Blue Gene Solution IBM	36864	2007	82161	103219

Installation of cables for Blue Gene/L at LLNL, October 2005.



Current and recent projects:

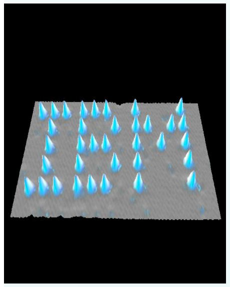
(1) (On-going) Blue Brain - attempt to map the entire human brain at the molecular level. That's 30 million synapses! (Ecole Polytechnique, Switzerland). Computer: Blue Gene.

(2) (2005) Simulation of the ribosome - the motion of 2.64 million atoms! Computer: Q Machine, Los Alamos. Largest biomolecular dynamics simulation to date.

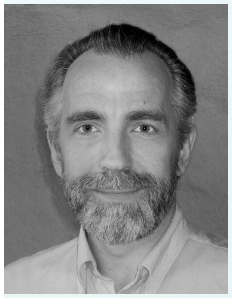
Nanoscience

Nanotechnology Historical Milestones

“It would be, in principle, possible (I think) for a physicist to synthesize any chemical substance that the chemist writes down. . . . Put the atoms down where the chemist says, and so you make the substance.” - Richard Feynman, *There’s Plenty of Room at the Bottom* (1959).



Don Eigler and Erhard Schweizer spelled “IBM” with 35 xenon atoms, 1989. This logo could fit 350 million times in the area at the point of a pin.



"We cannot afford certain types of accidents” - Eric Drexler, *Engines of creation*, 1986, the book where the term “grey goo” was coined.

“My budget supports a major new National Nanotechnology Initiative, worth $500 million. More than 40 years ago, Caltech’s own Richard Feynman asked, ‘What would happen if we could arrange the atoms one by one the way we want them?’” - President Clinton, Caltech (2000).

Nanotechnology Definition…

Google

```
define: nanotechnology
```

- Technology development at the atomic, molecular, or macromolecular range of approximately 1-100 nanometers to create and use structures, devices, and systems that have novel properties.
- The application of nanoscience in order to control processes on the nanometer scale, ie between 0.1 nm and 100 nm.
- The development and use of devices that have a size of only less than 200 nanometers.

- This is a general term for technology research on the scale of less than about 1000 nanometers.

- First coined by K. Eric Drexler in 1986
- According to the Oxford English Dictionary, the term “nanotechnology” was coined in 1974.

Nanotechnology Press

“How Super-Cows and Nanotechnology will Make Ice Cream Healthy”

telegraph.co.uk, August 21, 2005

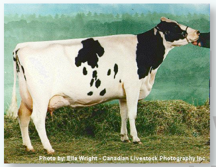


Photo by: Ella Wright - Canadian Livestock Photography Inc.

somewhere in County Down, Northern Ireland, is a herd of 40 ws that could take all the poisonous guilt out of bingeing on ice cream. Unilever, the manufacturer of Persil and PG Tips, is sponsoring a secret research project by a leading British agricultural science institution into how to reduce the levels of saturated fat in cow's milk.

It is also experimenting with nanotechnology, or the science of invisibly tiny things. Unilever believes that by halving the size of particles that make up the emulsion - or fatty oil - that it uses to make ice cream, it could use 90 per cent less of the emulsion.

(Ed.'s note: just shrinking the particle size into the nano-realm does not make it nanotechnology, even by today's materials science standards. For that, the particles would have to exhibit new properties.)

Nanotechnology in Hollywood*

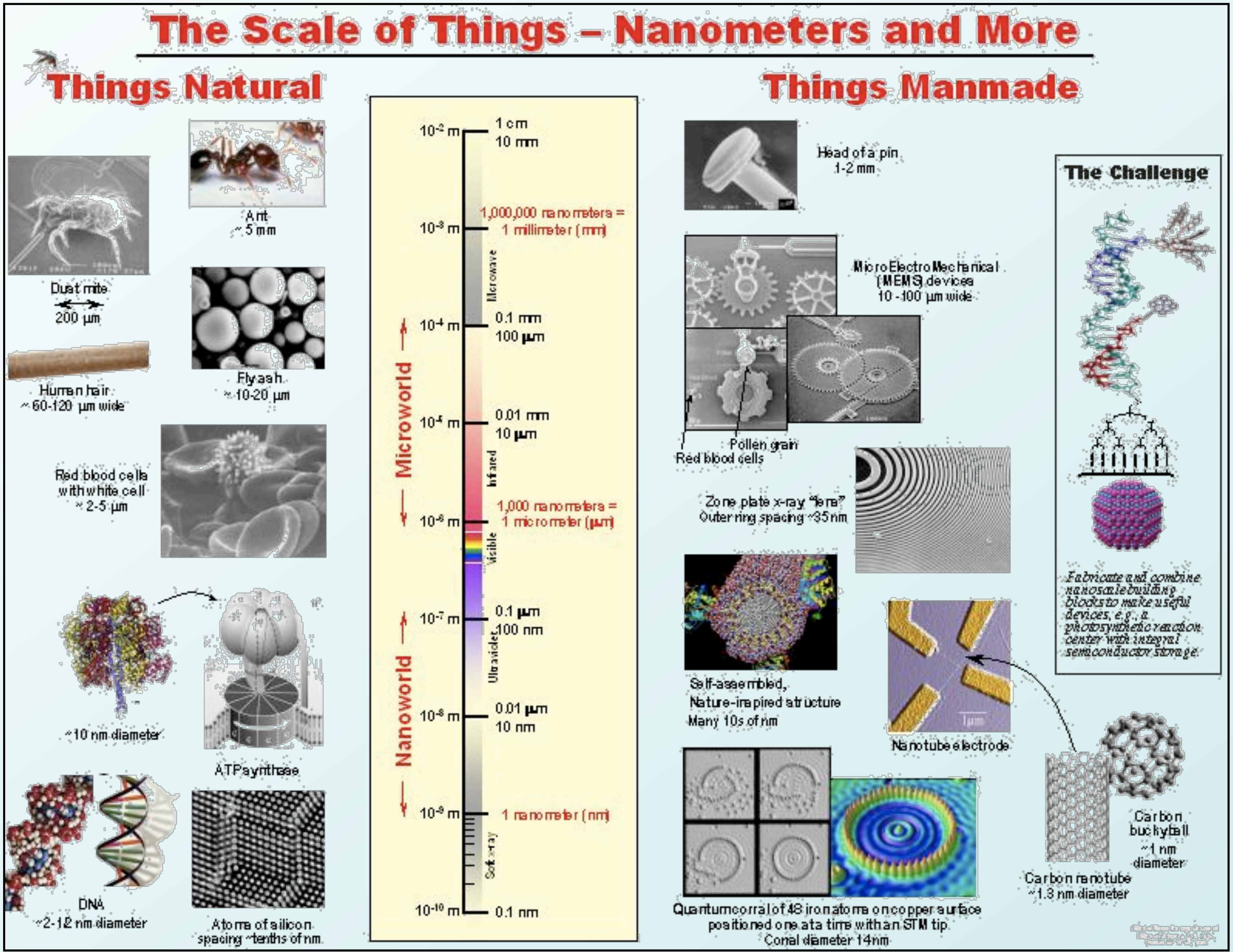
Nanotechnology Scientist in *Spiderman*[1]

Nanotechnology created *The Hulk*[2]

[1] Copyright 2002, Columbia Pictures, used without permission
[2] Copyright 2003, Universal Pictures, used without permission

*After M. Hersam

Ask the Government (www.nano.gov)



The Scale of Things – Nanometers and More

Things Natural

Dust mite
200 µm

Ant
~ 5 mm

Human hair
~ 60-120 µm wide

Fly ash
~ 10-20 µm

Red blood cells
with white cell
~ 2-5 µm

~10 nm diameter

ATP synthase

DNA
~2-1/2 nm diameter

Atoms of silicon
spacing ~tenths of nm

Things Manmade

Head of a pin
1-2 mm

MicroElectroMechanical
(MEMS) devices
10 -100 µm wide

Pollen grain
Red blood cells

Zone plate x-ray "lens"
Outer ring spacing ~35 nm

Self-assembled,
Nature-inspired structure
Many 10s of nm

Nanotube electrode

Carbon nanotube
~1.3 nm diameter

Carbon buckyball
~1 nm diameter

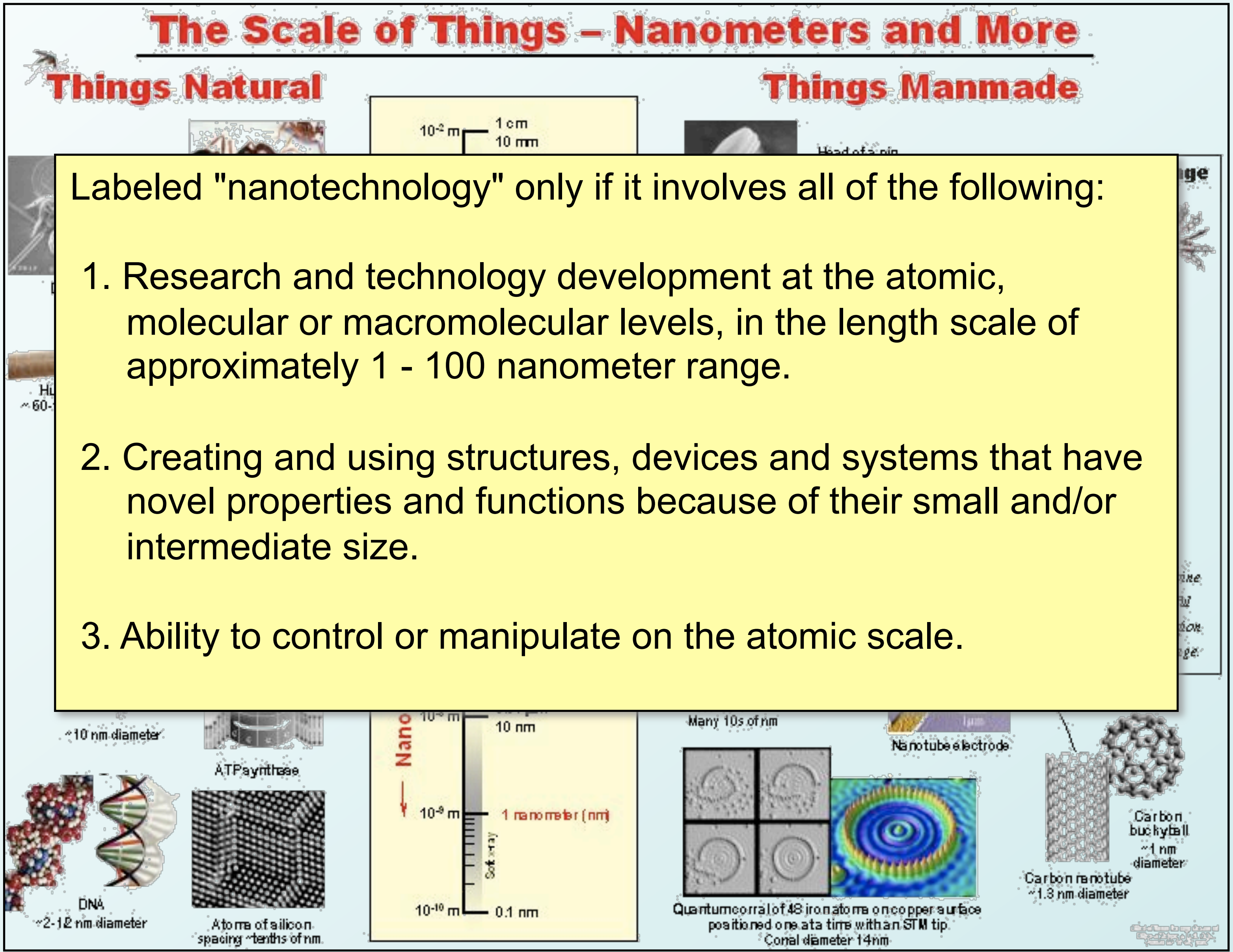
Quantum corral of 48 iron atoms on copper surface
positioned one at a time with an STM tip
Corral diameter 14 nm

The Challenge

Fabricate and combine nanoscale building blocks to make useful devices, e.g., a photosynthetic reaction center with integral semiconductor storage.

Jeffrey C. Grossman & Elif Ertekin, *NSE C242 & Phys C203, Spring 2008, U.C. Berkeley*

Ask the Government (www.nano.gov)



Labeled "nanotechnology" only if it involves all of the following:

1. Research and technology development at the atomic, molecular or macromolecular levels, in the length scale of approximately 1 - 100 nanometer range.

2. Creating and using structures, devices and systems that have novel properties and functions because of their small and/or intermediate size.

3. Ability to control or manipulate on the atomic scale.

Nanotechnology Definition

Not “nano by accident”

“The purposeful engineering of matter at scales of less than 100 nanometers to achieve size-dependent properties and functions.”

Matthew Nordan, Lux Research, 2005.

Really small

Not just “small;” “small and different”

Computational Nanoscience

- There was a little perspective on Computational Science and on Nanoscience

- Now how about Computational Nanoscience? Here’s a great quote:

> “The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known, and the difficulty lies only in the fact that application of these laws leads to equations that are too complex to be solved.” (P.A.M. Dirac)

- With a combination of computer power and algorithm development over the last 50 years, these equations - the ones that govern nanoscience - are no longer too complex to solve

- And that’s a good thing, because it is much needed…

Simulation as Computer Experiments

- Why is simulation so important in nanoscience and nanotechnology?
  - Because nanoscale systems are "complex"
  - Because experimental characterization is challenging
  - Because optimization is nonlinear
  - Because classical models and simple theories may fail

- Complex means (for now):
  - There is more than one individual constituent in the system
  - Constituents cannot be considered independent, as they interact with one another

How Does One Begin?

Generally speaking, two steps are involved:

1. A *mathematical model* must be introduced to describe, as faithfully as possible, the physical system under exam. For example, in the case of a classical monatomic liquid, atoms can be described as massive, point-like particles, mutually interacting via a two-body potential.
   A model is essentially an *idealized* version of the system under exam, conceived to be mathematically well-defined, so that calculations can be performed, and to incorporate most of the effects that are relevant to the science in which one is interested.

2. The model introduced in must be studied by solving appropriate equations of motion, either classical (i.e., **F=ma**) or quantum-mechanical, according to whether the model is based on Newtonian or quantum mechanics. The solution of the equation of motion enables one to calculate values of physical quantities of interest.

Defining vs. Solving a Model

Both are, of course, important. In this class, we will focus more on *solving* a specific model, for a number of reasons:

- Very accurate and realistic models often exist in both basic and applied research.
- Artificial models are also of interest - not quantitative, but can incorporate essential chemistry/physics of an entire class of systems.
- Even when a model is known to a high degree of accuracy, an exact analytical solution of the equations of motion is usually out of question.

Why is this so difficult?

Solving a Model is Difficult

The presence of interactions among constituents renders the many-body problem usually unsolvable exactly.

As of now, no general solution is known when the number of interacting bodies is greater than two.

One can sometimes obtain useful results by making simplifying approximations, for example:

- neglecting interactions altogether (sometimes OK, e.g., for gases at high temperature)
- reducing the problem to an effective one-body problem, describing the interactions by means of an *effective single-particle external field* (e.g., in the Hartree method)

However, these simplifications can have *uncontrolled approximations*, meaning it may be impossible to estimate reliably how far off one is from the exact answer (the one corresponding to the initial model).

Advantage of Computer Simulations

Computer simulations can obtain *essentially exact* numerical solutions to the equations of motion for the model of interest, with no approximation.

- obtain reliable theoretical predictions for the system of interest
- gauge quantitatively the accuracy of approximate methods and theories, by comparing their predictions against computer-generated answers
- much easier and faster to do, even for very complex systems, than analytical calculations, even approximate ones
- scale up with the increase of computer power; thus, something which is not feasible today might become so tomorrow
- generally allow to calculate the most relevant quantities, those that are measured in an experiment, or that are most directly related to our understanding, or that would be too difficult to measure experimentally

Some Important Issues with Simulations

- results are *numbers*, as opposed to elegant equations; thus, one has to make sense out of a set of numbers, i. e. interpret the results.

- results are obtained for *finite systems* only, whereas often one is seeking estimates for infinite systems, to eliminate surface effects; thus, one has to assess the sensitivity of the results to the system size (finite-size scaling).

- even when the simulation is free from approximation, the results are usually affected by *statistical errors*, which have to be estimated.

- Because the field of computer simulations is no more than 50 years old, most interesting techniques have likely not yet been invented or tried.

Some Important Issues with Simulations

Some good initial questions to ask before trying to solve a problem with a computer:

- What is the optimal algorithm for a given problem?

- What do we want to calculate?

- Is it feasible, given the computational facilities available?

- What are the limits, underlying assumptions, and underlying approximations in our model? Is it still useful?

Analogies with Experiment

- The goal is to obtain a (hopefully) reliable numerical estimate for some relevant physical quantity **A** characterizing a system of interest, such as a liquid, or a solid, or a biological membrane.

- Let us assume that a mathematical model, deemed suitable, has been defined for the system of interest.
  The simulation essentially consists of performing an *experiment on the model*, i. e. on a *fictitious* system, described by the mathematical model specified.

- For example, we may put a bunch of particles in a box, specify the way in which they interact (the potential), assign initial positions and velocities and let them move according to the laws of classical mechanics (this is a Molecular Dynamics simulation).

- A computer can do that for us, thereby simulating the behavior that our ideal system would have if it existed, and rendering it possible to make measurements on it.
  Naturally, the “measurements” are performed by the computer itself.

Simulations as Computer Experiments

Just as in an experiment, the estimate of the value of the quantity **A** is obtained by measuring it a sufficiently large number of times during the simulation, and averaging the results in some way.

An important difference between a real experiment and the simulation is that the latter makes it possible to “measure” quantities that cannot be easily (or at all) measured in a real experiment (for example, the trajectory of a single particle).

This way of “solving” the problem has obvious methodological and deep *philosophical* differences with respect to tackling the problem analytically.

Sources of Error

Just like in a real experiment, in computer simulations the measured quantity is assigned a numerical value, which is in principle affected by *statistical and systematic errors*.

Statistical errors

Always affect numerical simulations, just like they affect experimental measurements. This is because the experiment (e.g., the simulation) necessarily takes a finite time.

As it turns out, assigning statistical errors is a rather delicate part of the job (as we'll see in a moment).

Systematic Errors

The hardest to deal with; i.e., if my instrument is not calibrated the outcomes of my measurements will be biased.

Systematic errors can be generally associated with “lack of calibration”, or “incomplete calibration”, even when discussing computer simulations, for which the word “calibration” takes a slightly generalized meaning.

In a computer simulation, typical sources of systematic errors are:

- The finite size of the simulation sample
- A defective random number generator in Monte Carlo simulations
- The finite time-step in Molecular Dynamics simulations
- Round-off errors due to the finite precision of the computer (any numerical calculation suffers from this)
- Sometimes, approximations in the numerical procedure (hopefully, this is never the case!)

There are, with the sole exception of the last item in the above list, well-defined methods to deal with systematic errors in the context of computer simulations.

Computation of Physical Averages

Again, let's draw analogy with experiment

(Basically) two types of experiments:

- those in which the system of interest is in *thermal equilibrium*, i. e. for which physical averages do not change with time
- those in which the system is not in equilibrium, or it is under the influence of an external perturbation

For the first case, the physical system is initially "prepared" into a given state, and then allowed to "relax" toward equilibrium. The experimenter has to:

- determine when equilibrium has been reached
- perform measurements to determine the values of the sought physical averages for the system in equilibrium

Computation of Physical Averages

Analogy with experiment, continued

If the experiment could be carried on for an infinitely long time, the experimenter could collect an infinite number of measurements of the desired physical quantity for the system in equilibrium, and determine its *expectation value*, an error-free average.

Naturally, instead, the experiment can only last a finite time, and so one can only attempt to estimate the given expectation value by computing an *average* over a finite number of measurements, determining the *statistical error.*

A fundamental assumption underlying this procedure is ergodicity, according to which a system, given enough time, will explore all its accessible states.*

This is equivalent to taking instantaneous measurements on a very large number of independent systems in equilibrium.

* See, for example, F. Reif, *Fundamentals of Statistical and Thermal Physics*, McGraw-Hill

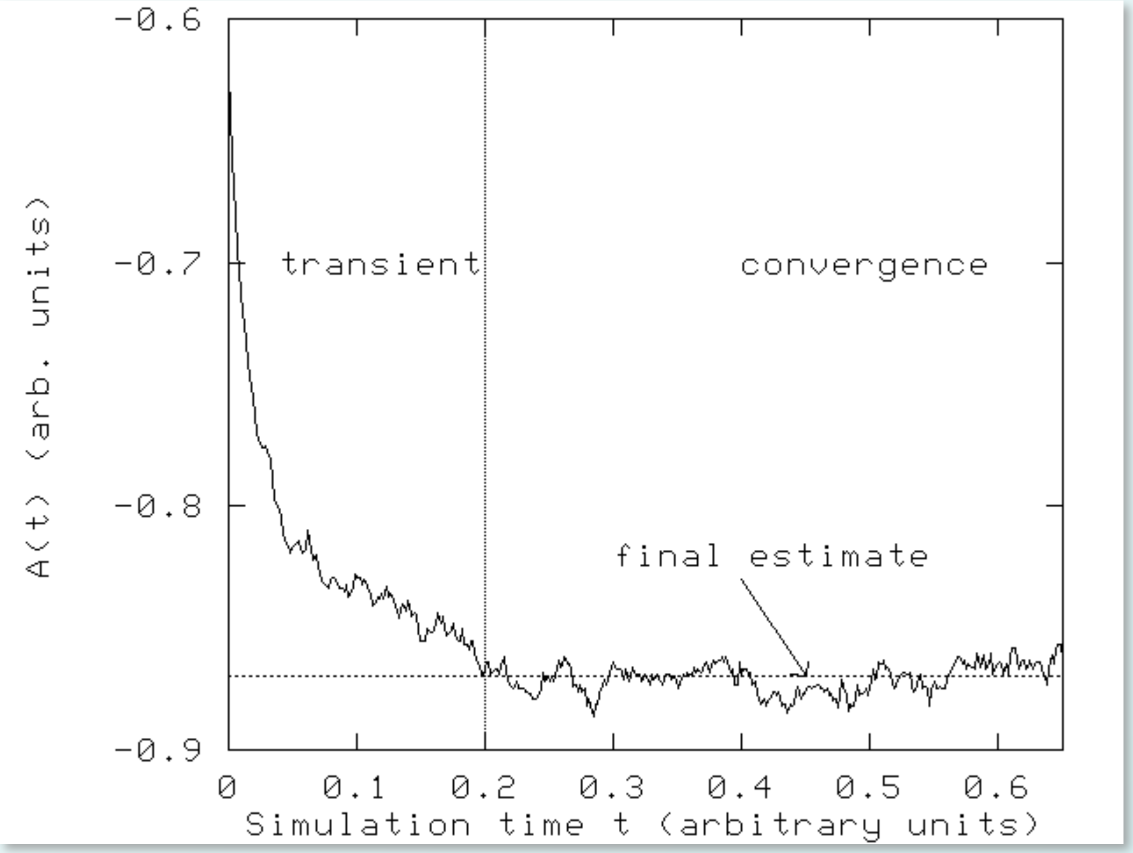
Computation of Physical Averages

Simulations of systems in equilibrium typically start from a given configuration of the system. For example:

- if one is considering a classical liquid one may put N particles at random positions inside a container, with initial velocities specified according to the selected value of the temperature
- if one is interested in the solid phase of a given element or material, it is usually a safe choice to start with particles at well-defined lattice positions
- in a quantum simulation, it is impossible to assign both positions and velocities, so one simply assigns the positions

An Example Result

Here’s a typical simulation result:



Computation of Physical Averages

If the simulation could last for an arbitrarily long time, <**A**> could approach arbitrarily the expectation value of **A**, which is what, ideally, we would like to determine, and the initial transient would be rendered immaterial, as it would represent a vanishingly small fraction of the simulation time.

But because in practice the simulation cannot take an arbitrarily long time, <**A**> can only be estimated, by averaging over a finite period of time, and the problem arises of assessing the reliability of the estimated value of <**A**>

Unfortunately, often the initial transient represents a significant fraction of the total simulation time; thus, how does one go about eliminating, or rendering as small as possible the effect of the transient on the final estimate of <**A**>?

Error bars prove crucial to overcome both problems

Error Bars

Error bars determine a range, centered around the estimated <**A**>, within which the “true” <**A**> is expected to lie with a specified *probability*.

Many simulation results are not meaningful without an associated error bar

Once a robust procedure is defined to calculate error bars reliably, then one can empirically attempt to establish how many initial data points should be discarded, in other words how long the transient is.

Typically, one wants to make sure that as little data as possible from the transient region is included in the average, but at the same time one does not want to throw away too much data.

What is usually done is to calculate the average < **A** > by discarding an increasing number of initial data points until the average does not change, within error bars.

Occasionally, one may fail to reach such a limit: in this case, the simulation should be carried on for a longer time, as the transient extends through most of it.

Review: Error Bars

Averages are computed as: $< A >= \frac{1}{n}\sum_{i=1}^{n} A_i$

Statistical error is then: $(\sigma)^2 \approx (\delta A)^2 \left(1 \pm \sqrt{\frac{2}{(n-1)}}\right)$

where: $(\delta A)^2 = \frac{\sum_i (A_i - < A >)^2}{n(n-1)}$

Actually, this is only true when two criteria are met: (1) the data is drawn from a gaussian probability distribution, and (2) the data are uncorrelated

Reblocking Transformation

Blocking or “binning” method is simplest way to eliminate data correlation.

In this method, the data set is transformed, for example into one that is 1/2 the size:

$$\left\{A_1, A_2, A_3, \ldots, A_n\right\} \rightarrow \left\{A_1^{'}, A_2^{'}, A_3^{'}, \ldots, A_{n/2}^{'}\right\}$$

The statistical error is then recomputed, and the reblocking is repeated.

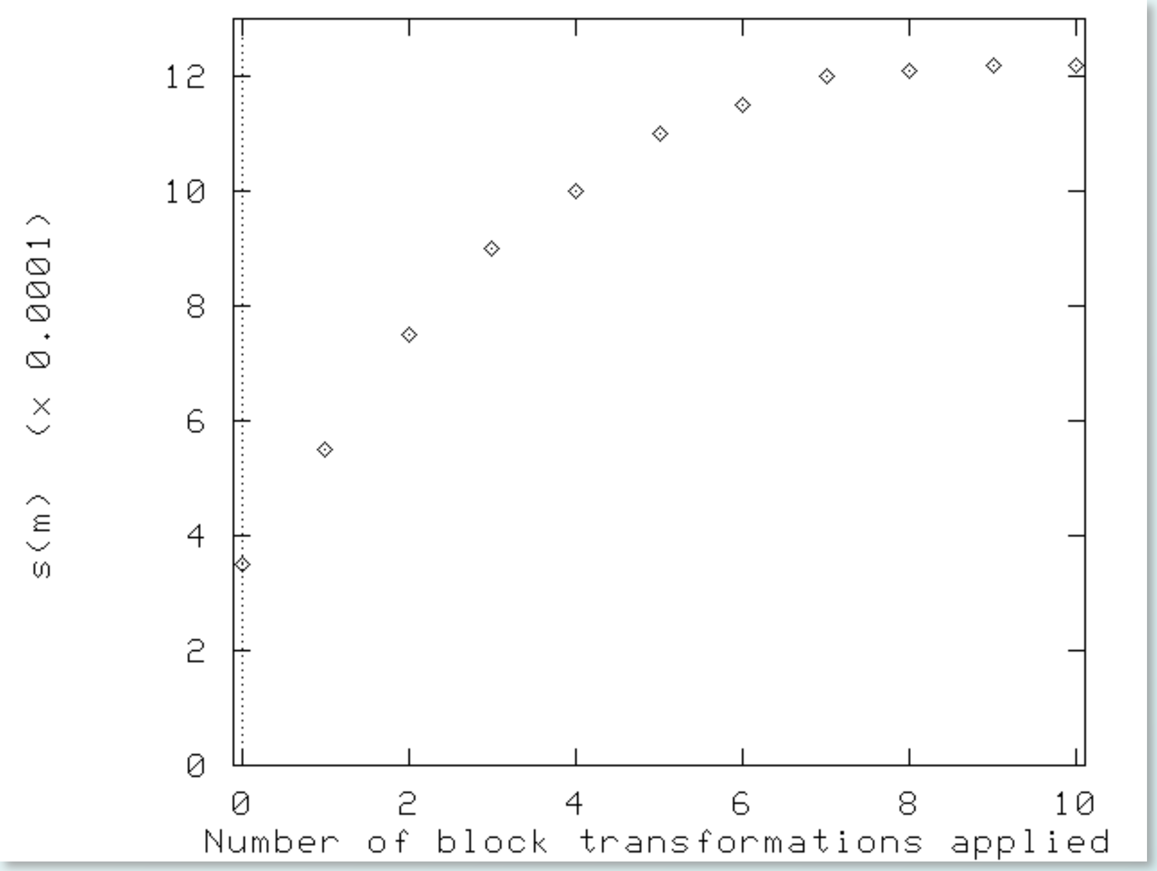
One can derive the following result (*):

$$\sigma^2 \geq (\delta A^{(n/2)})^2 \geq (\delta A^{(n/2-1)})^2 \geq \cdots \geq (\delta A^{''})^2 \geq (\delta A^{'})^2 \geq (\delta A)^2$$

* see, for example, H. Flyvbjerg and H. G. Petersen, *J. Chem. Phys.* **91**, 461 (1989).

Reblocking Transformations

An example of the blocking method applied to a data set of 137,072 elements from a Monte Carlo simulation.


s(m) (x 0.0001)
Number of block transformations applied