多体問題の計算科学

Computational Science for Many-Body Problems

#1 Many-body problems in physics

15:10-16:40 April 6, 2021

1. Background of the lecture
2. Many-body problems in physics
3. Finite size and discretization
4. Why many-body problems is hard to solve
5. (Super)computers
6. Details and exercises

Computational Science for Many-Body Problems

Lecture Materials

You will find our lecture materials at ITC-LMS or Github

https://github.com/compsci-alliance/many-body-problems

Online Guidance for Computational Science Alliance

令和３年度　計算科学アライアンス　学生募集説明会
第１回：2021年4月6日 (火曜日) 17時00分 ~ 18時35分
第２回：2021年4月9日 (金曜日) 17時00分 ~ 18時35分
URL: https://u-tokyo-ac-jp.zoom.us/j/89239347429?pwd=TTJ6UEdYdkNFbCtrZzUzZHg3MWgvdz09
ミーティングID: 892 3934 7429
パスコード: 711356
URL: https://u-tokyo-ac-jp.zoom.us/j/82077948129?pwd=dng0YkJwU3ZVczg2SDdKN1NWS2RnZz09
ミーティングID: 820 7794 8129
パスコード: 916287

Background of This Lecture

Computational
Science
Alliance
The University of Tokyo

UTokyo
Computational Science Alliance

Fugaku project

Program for Promoting Researches on the Supercomputer Fugaku
Basic Science for Emergence and Functionality in Quantum Matter
— Innovative Strongly-Correlated Electron Science by
Integration of “Fugaku” and Frontier Experiments —

富岳

Lecturer

山地 洋平 YAMAJI, Youhei
YAMAJI.Youhei@nims.go.jp

Senior Researcher
IFCS Group, GREEN,
National Institute for Materials Science

Research:
Theoretical condensed matter physics
Computational method of many-body quantum systems

Developer of open source codes for supercomputers

Quantum lattice model solver HΦ

http://ma.cms-initiative.jp/ja/index/ja/listapps/hphi

Background of This Lecture

Computational Science

1st Experimental Science
2nd Theoretical Science
3rd Computational Science
4th Data Science

Computational Science 計算科学
Computer Science 計算機科学

Background of This Lecture

Computational Science Education

Computational
Science
Alliance

The University of Tokyo

Background of This Lecture

Computational Science Education

Computational
Science
Alliance
The University of Tokyo

This lecture is in Category F Computational Science
計算科学の各分野におけるシミュレーション手法と
その研究成果について学ぶ。
電子状態計算、分子動力学、量子多体計算、数値流体力学、
構造計算、ゲノム解析など、さらには社会科学や経済分野
における最先端手法を学習する。
また、実際にソフトウェアを用いて大規模計算科学
シミュレーションを実行する

Many-Body Problems

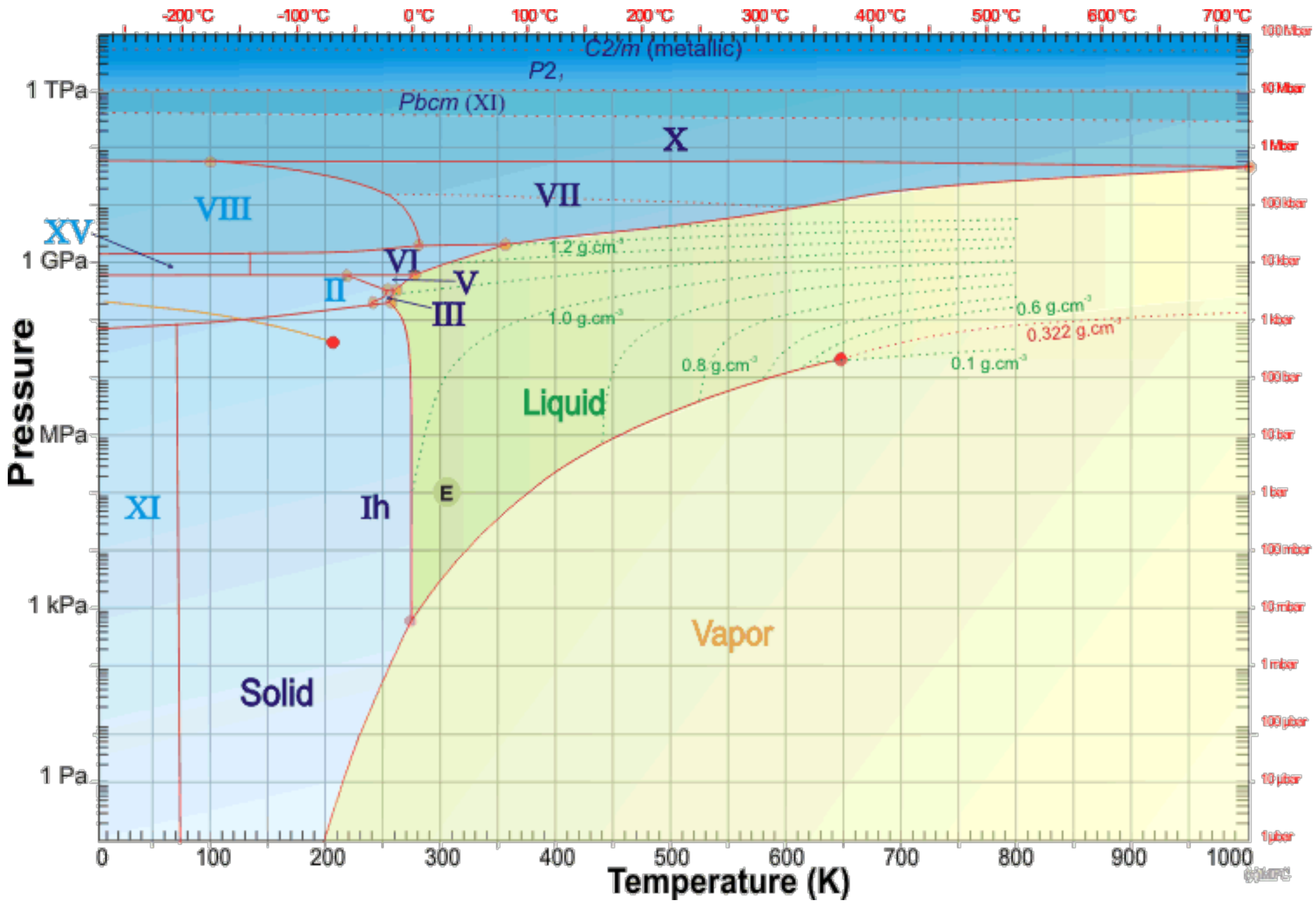
Many-body problems in Physics

Examples:
-Celestial objects
-Proteins and molecules
(molecular mechanics, quantum mechanics/molecular mechanics)
-Electrons in molecules and solids
-Quantum Chromodynamics

Principles:
・Classical mechanics
(Newtonian equation of motion, Post Newtonian, …)
・ Quantum mechanics
(Schrödinger equation, …)
・ Classical/quantum statistical mechanics

An Example of Many-Body Problems: H_2O

Phase diagram of H_2O



Martin Chaplin
Water Structure and Science
http://www1.lsbu.ac.uk/water/

An Example of Many-Body Problems: Proteins

Proteins in water

David E. Shaw *et al.*,
D. E. Shaw Research
SC09 (2009)

The state of the system is determined by position and velocity of each particle

$$(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N; \vec{v}_1, \vec{v}_2, \ldots, \vec{v}_N)$$

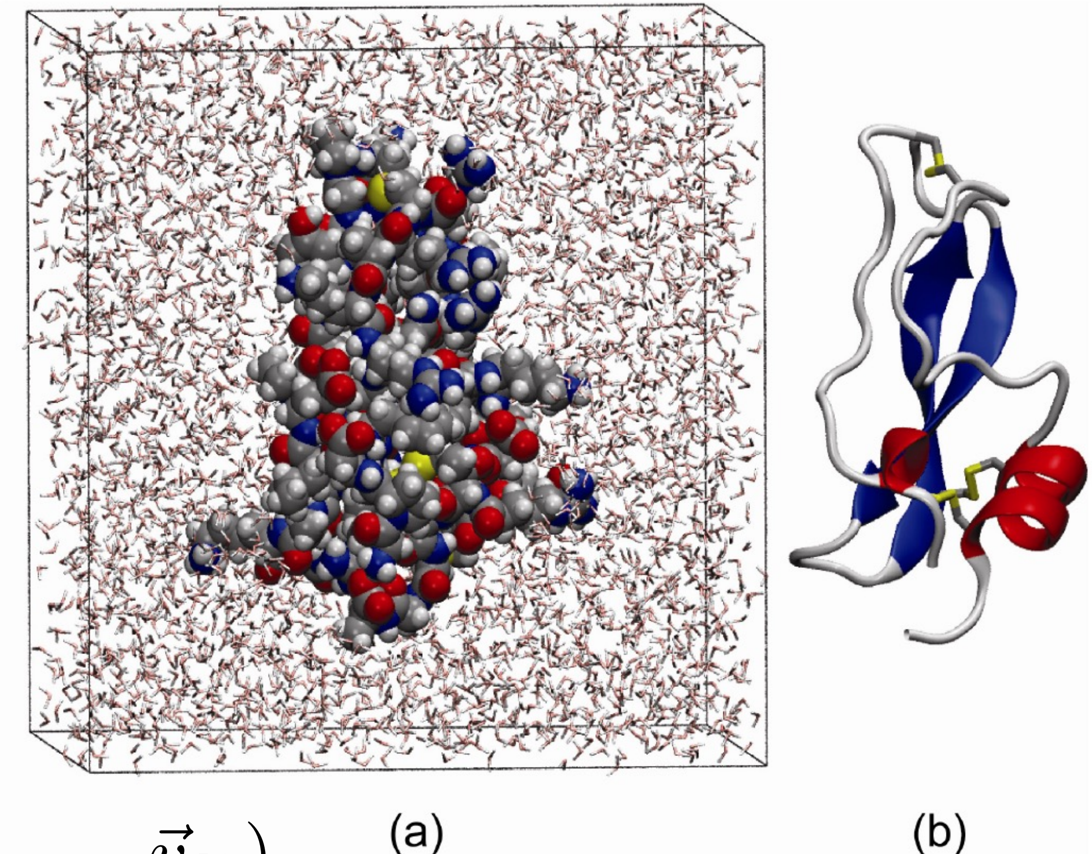


Figure 1: Two renderings of a protein (BPTI) taken from a molecular dynamics simulation on Anton. (a) The entire simulated system, with each atom of the protein represented by a sphere and the surrounding water represented by thin lines. For clarity, water molecules in front of the protein are not pictured. (b) A "cartoon" rendering showing important structural elements of the protein (secondary and tertiary structure).

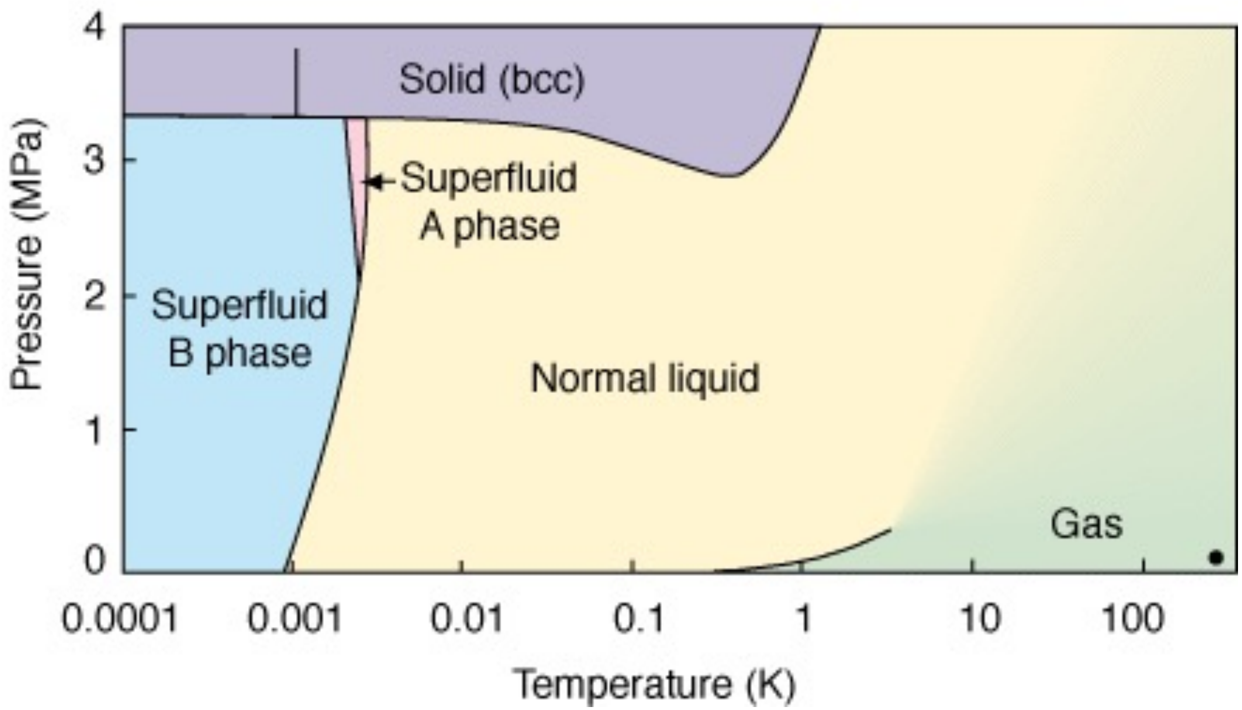
An Example of Many-Body Problems: Quantum Liquids

Phase diagram of ^{3}He

The system at zero temperature is represented by the wave function (complex function of particles' position)

$$\psi(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N; t)$$

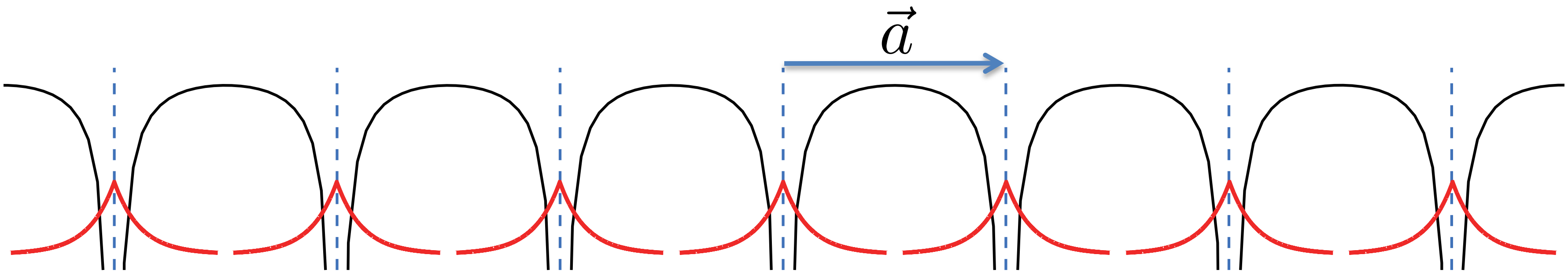
D. D. Osheroff, R. C. Richardson, and D. M. Lee, Phys. Rev. Lett. 28, 885 (1972).



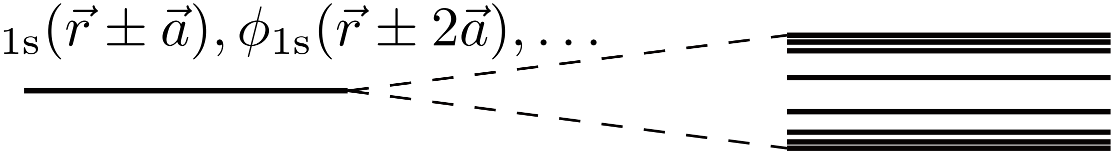
Erkki Thuneberg
http://ltl.tkk.fi/research/theory/helium.html

A Minimal Model of Many-Body Problems

One of the simplest many-body electrons in Crystalline solids: Hydrogen solid



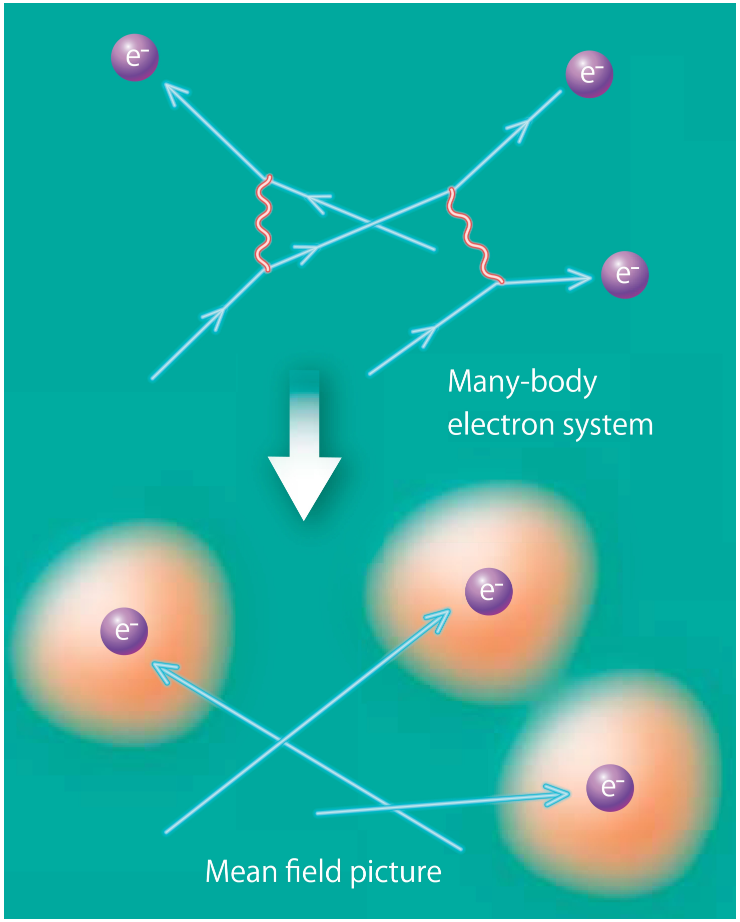
$\phi_{1\mathrm{s}}(\vec{r}), \phi_{1\mathrm{s}}(\vec{r} \pm \vec{a}), \phi_{1\mathrm{s}}(\vec{r} \pm 2\vec{a}), \ldots$



Gedankenexperiment of F. N. Mott

Starting from isolated one-body wave functions and constructing entangled $\psi(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N; t)$

Many-Body Problem & One-Body Picture in Solid State Physics



"One-body picture"

Basis of quantum many-body problem

Derive effective one-body picture

-Hartree-Fock (Mean-field)
-Landau's Fermi liquid (1956)
-Hohenberg-Kohn
& Kohn-Sham theory (1965)

Trend after Landau & Kohn

Ab initio
LDA, GGA, HF

Model calculations (direct approaches)
Hubbard model, Heisenberg model,…
ED,QMC,DMFT,…

cf.) Solvable model

Recent trends

GW, post HF, LDA+DMFT, GW+DMFT…

Finite Size and Discretization

- *Theory* handles infinite number of and continuous degrees of freedom

Naïvely, computers can not handle them

→ Reformulate the problems with finite number of and discrete degrees of freedom

Size of Quantum Many-Body Problems

Hilbert space dimension can be stored in memory

～50 qubits or 50 spins
(Heisenberg-like hamiltonian with N (< 50) *spins*)

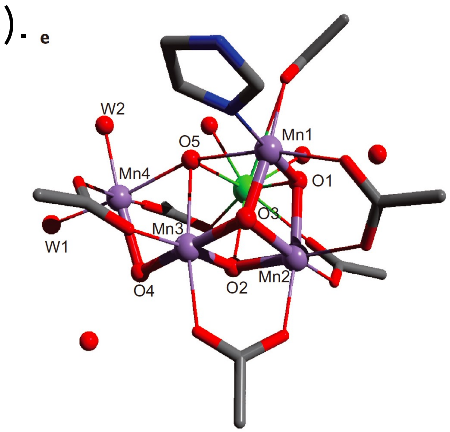
Examples of finite size systems from chemistry

-A H_2O molecule: 5↑ & 5↓ electrons in 41 orbitals
→ 5.6×10^{11} dimensional (～2^{39})

G. K.-L. Chan & M. Head-Gordon, J. Chem. Phys. 118, 8551 (2003).

-Manganese cluster in photosystem II:
44 electrons in 35 orbitals
→ 2×10^{18} dimensional (～2^{61})

Y. Kurashige, G. K.-L. Ghan, & T. Yanai, Nat. Chem. 5, 660 (2013).



Size of Quantum Many-Body Problems

Hilbert space dimension can be stored in memory

∼50 qubits or 50 spins
(Heisenberg-like hamiltonian with N (< 50) *spins*)

How about crystalline lattice ?
(with periodic boundary)

-Finite N calculations are useful?

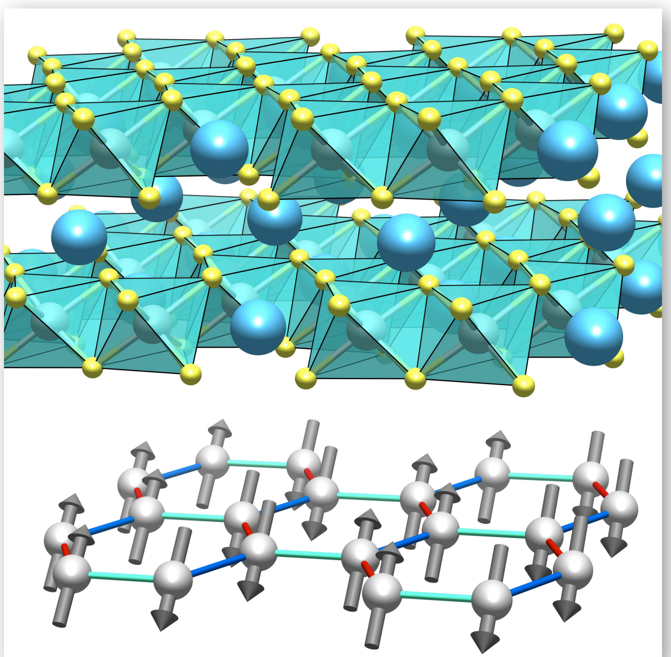
Finite size scaling (2nd & 3rd lectures):
Extracting information of infinite systems from finite ones
Nearsightedness
W. Kohn, Phys. Rev. Lett. 76, 3168 (1996).

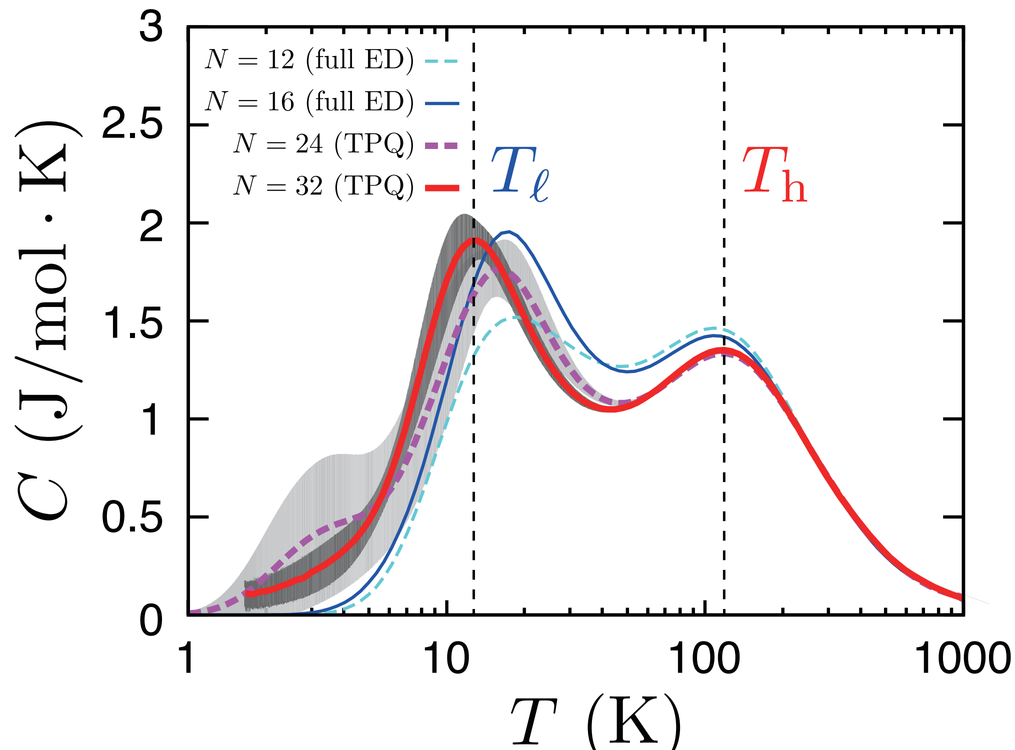
Nearsightness

For example, W. Kohn, Phys. Rev. Lett. 76, 3168 (1996).

-Excitation gap, temperature, frustration make correlation length shorter and, thus, finite-size effect smaller

An example: Frustrated magnet Na_2IrO_3 (12th)





From Continuous to Discrete

We need to rewrite *first principles* equations to calculate them in suitable and efficient way with (super)computers

A typical example: Discretization

Differential equation

$$\frac{\partial}{\partial t} u(t,r) = D \frac{\partial^2}{\partial r^2} u(t,r)$$

Difference equation

$$u(n+1,x) = \frac{1}{2}\left[u(n,x+1) + u(n,x-1)\right]$$

Other examples: Symplectic integral, variational methods, ⋯

Difference and Differential: Diffusion Eq.

Difference eq. $$u(n+1,x) = \frac{1}{2}\left[u(n,x+1) + u(n,x-1)\right]$$

Differential eq. $$\frac{\partial}{\partial t}u(t,r) = D\frac{\partial^2}{\partial r^2}u(t,r)$$

$$\boxed{D = \lim_{\delta t\to 0, \delta r\to 0}\frac{(\delta r)^2}{\delta t}}$$

$$u(t+\delta t, r) = \frac{1}{2}\left[u(t, r+\delta r) + u(t, r-\delta r)\right]$$

$$\begin{cases} u(t+\delta t, r) = u(t,r) + \delta t\frac{\partial}{\partial t}u(t,r) + \frac{\delta t^2}{2}\frac{\partial^2}{\partial t^2}u(t,r) + \mathcal{O}(\delta t^3) \\ u(t, r+\delta r) = u(t,r) + \delta r\frac{\partial}{\partial r}u(t,r) + \frac{\delta r^2}{2}\frac{\partial^2}{\partial r^2}u(t,r) + \mathcal{O}(\delta r^3) \end{cases}$$

$$\Longrightarrow \frac{\partial}{\partial t}u(t,r) = D\frac{\partial^2}{\partial r^2}u(t,r)$$

Classical vs Quantum Mechanics

classical point particles $(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N; \vec{v}_1, \vec{v}_2, \ldots, \vec{v}_N)$

Every set of locations/positions and velocities of particles gives you an *eigenstate* of the hamiltonian:
The hamiltonian is already *diagonal*

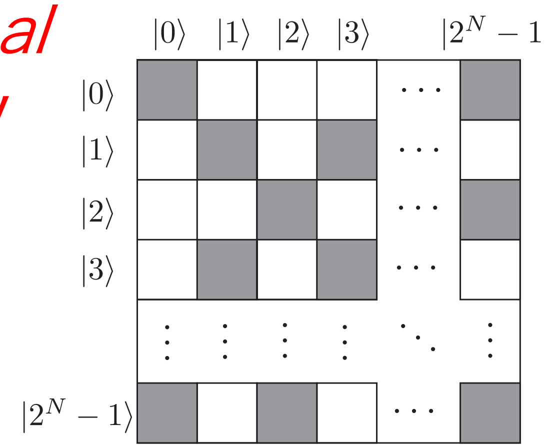
$$E = \sum_{j=1}^{N} \frac{m_j}{2} |\vec{v}_j|^2 + V(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N)$$

Example of potentila: gravitational or Coulomb force

$$V(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N) = g \sum_{i<j} \frac{1}{|\vec{r}_i - \vec{r}_j|}$$

quantum The hamiltonian is not *diagonal* when the basis set is given by positions of particles

$$|\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N\rangle$$



cf.) Uncertainty principle,
zero point motion, or quantum fluctuation

Why Many-Body Problem Is Hard to Solve

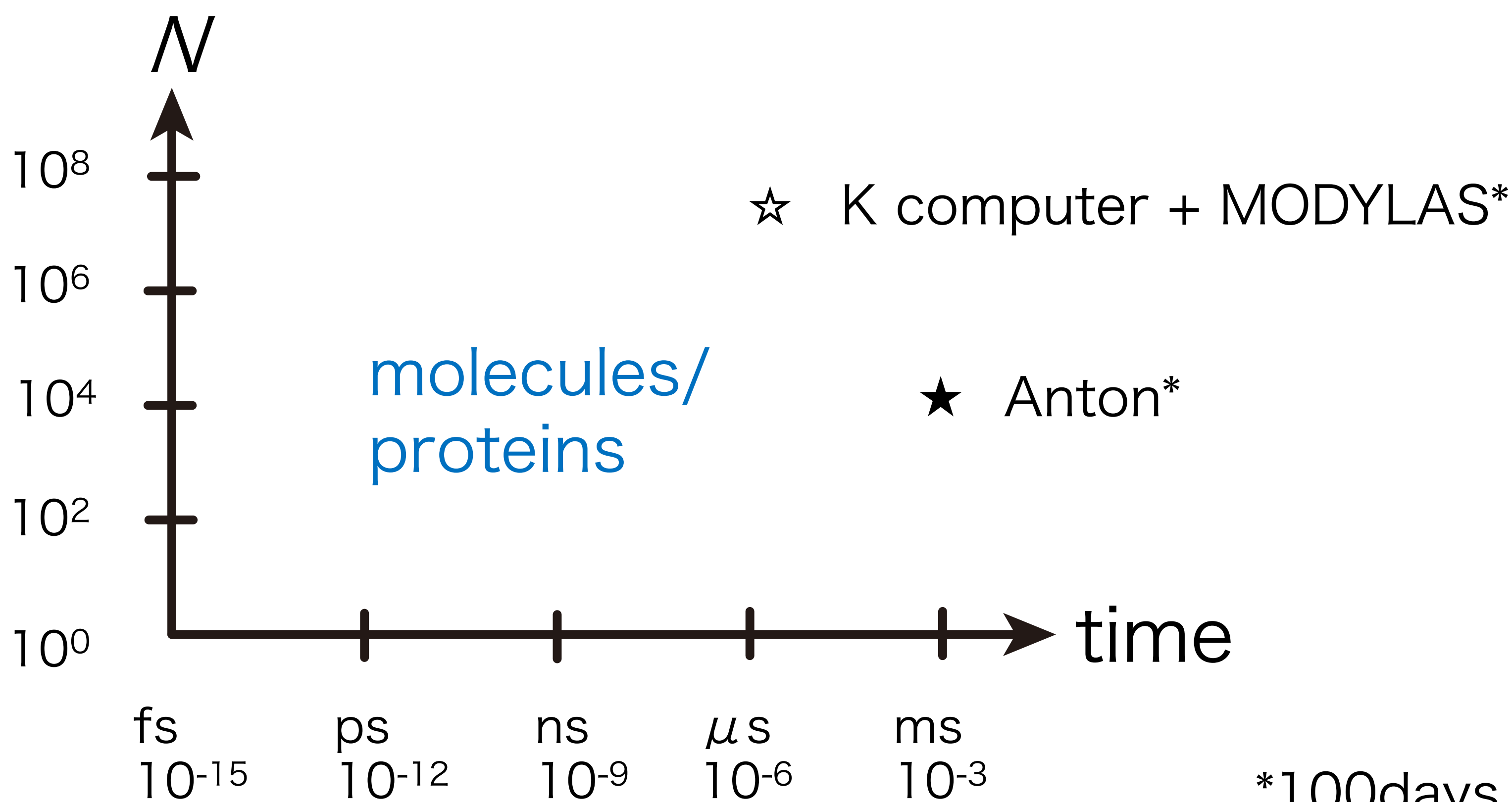
Target of this lecture:

1. N-body Newtonian equation of motion
2. N-body classical statistical mechanics
3. N-body Schrödinger equation
4. N-body quantum statistical mechanics

Why Many-Body Problem Is Hard to Solve

1. N-body Newtonian equation of motion
-Time evolution of $6N$ degrees of freedom

parallelization



*100days

An Example of Many-Body Problems: Proteins

Proteins in water

David E. Shaw *et al.*,
D. E. Shaw Research
SC09 (2009)

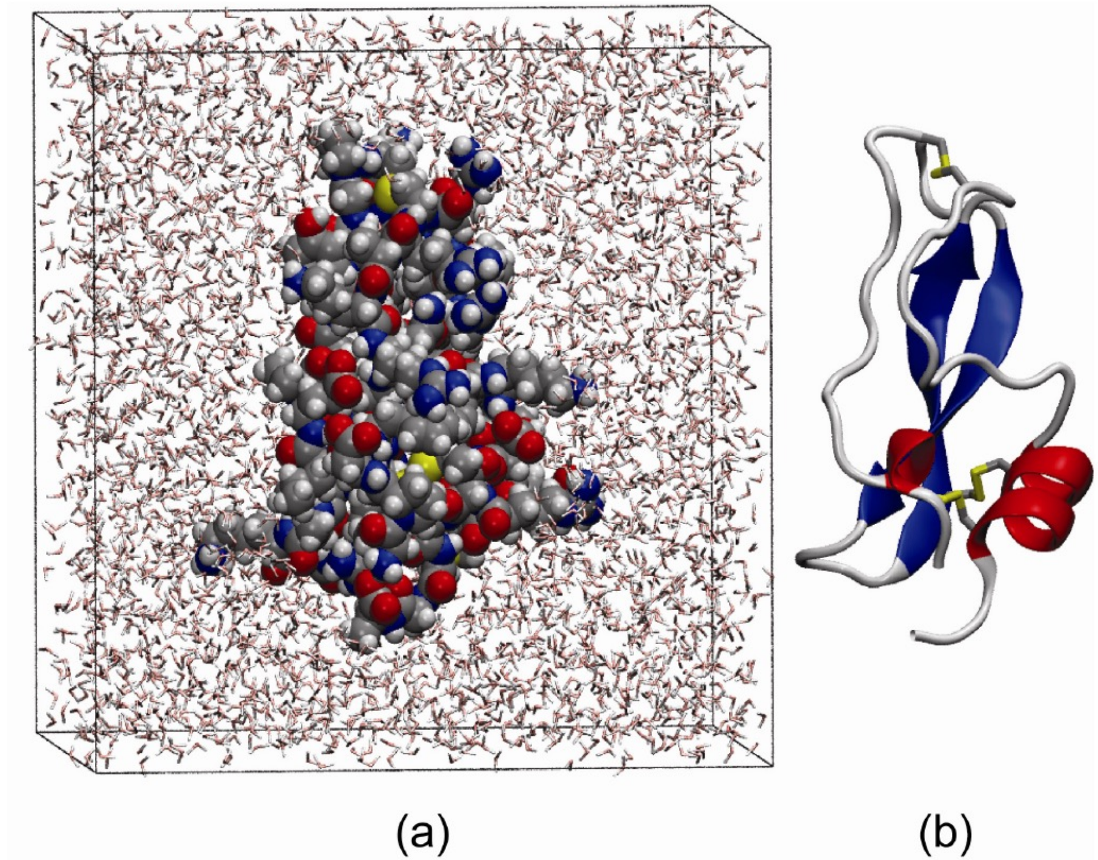


Figure 1: Two renderings of a protein (BPTI) taken from a molecular dynamics simulation on Anton. (a) The entire simulated system, with each atom of the protein represented by a sphere and the surrounding water represented by thin lines. For clarity, water molecules in front of the protein are not pictured. (b) A "cartoon" rendering showing important structural elements of the protein (secondary and tertiary structure).

Eigenvalue Problems

Diagonalizing Hermitian matrices

Standard approach:
Call LAPACK (Linear Algebra PACKage)

subroutine zheev:
Householder reflection + QR algorithm
$\rightarrow O(N^3)$ numerical cost

Computational and Memory Costs

1. Vector-vector product $$\sum_{j=0}^{N_{\mathrm{H}}-1} u_j^* v_j$$

   Computational: $\mathcal{O}(N_{\mathrm{H}})$
   Memory: $\mathcal{O}(N_{\mathrm{H}})$

2. Matrix-vector product $$v_i = \sum_{j=0}^{N_{\mathrm{H}}-1} A_{ij} u_j$$

   Computational: $\mathcal{O}(N_{\mathrm{H}}^2)$
   Memory: $\mathcal{O}(N_{\mathrm{H}}^2)$

3. Matrix-matrix product $$C_{ij} = \sum_{k=0}^{N_{\mathrm{H}}-1} A_{ik} B_{kj}$$

   Computational: $\mathcal{O}(N_{\mathrm{H}}^3)$
   Memory: $\mathcal{O}(N_{\mathrm{H}}^2)$

Difficulties in Many-Body Problems

-Longer simulation

-Summation over exponetially large
 # of configurations

←Monte Carlo (3rd to 9th lecture), ⋯

-$O(N^3)$ numerical cost and $O(N^2)$ memory

←Krylov subspace method (11th to 12th), ⋯

Computer

Supercomputers in UTokyo

Ohtaka (CPU server)

ISSP, UTokyo@Kashiwa
Theoretical Peak: 6,881TFlop/s
AMD EPYC 7720 2.0GHz, 64 cores x 2
メモリ: 258GiB
1,680 nodes

Oakforest-PACS

JCAHPC@Kashiwa (Utokyo & Tsukuba U)
Theoretical Peak: 24,913.5TFlop/s
Intel Xeon Phi 7250 (1.4GHz, 68 cores, 112GB)
8,208 nodes

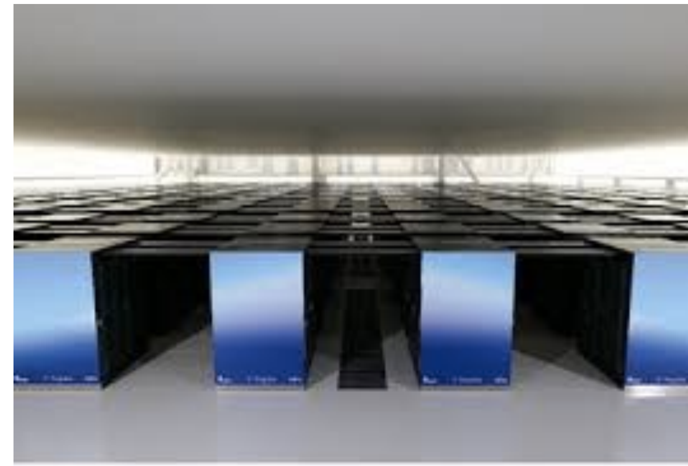
Oakbridge-CX (You can use this when you join CSA!)

Intel Xeon
1,368 nodes

“Fugaku” Supercomputer

Fugaku 富岳 @RIKEN, Kobe
-158,976 nodes
 A64FX
   48 cores/node
   32 GiB/node

- 1st in Top500 & 9th in Green500 on June 2020
- 1st in Top500 & 10th in Green500 on November 2020



How Computer Handles Numbers

Numbers in Computers

Integer

Integer(4): 31bit+1bit

$$-2^{31} \leq j \leq 2^{31} - 1 \ (j \in \mathbb{Z})$$

Real number

IEEE Std 754-2008 binary64
(double-precision floating-point number)

Double precision/real(8):
Sign (1 bit) + Exponent (11 bit) + Significand (52 bit)

$$(-1)^s \times 2^e \times m$$

$$-1022 \leq e \leq 1023 \ (e \in \mathbb{Z}) \qquad m = \frac{\sum_{\ell=0}^{p-1} d_\ell \cdot 2^\ell}{2^p} \qquad p = 53$$

Performance of computer

How many times does the computer multiply/add per second?
How much data does the computer memorize?
How much data does the computer read/write per second?

FLOP/s

Floating-point Operations Per Second

An example: Intel Xeon Phi Nights Landing

Intel Xeon Phi 7250 (1.4GHz, 68 cores, 112GB)

-1.4×10^9 instructions per second

(instruction to perform double precision add or multiply)

-Intel AVX-512 instruction*

double precision floating point number (8byte=64bit)

8 double-precision multiply-add** operations

$\rightarrow 1.4 \times 10^9 \times 16 \times 68$ FLOP/s per processor

*Other instruction set:

SVE (for example, A64FX of Fugaku)

**Multiply-add: $a \leftarrow a + (b \times c)$

Increasing FLOP/s

Clock rate saturated

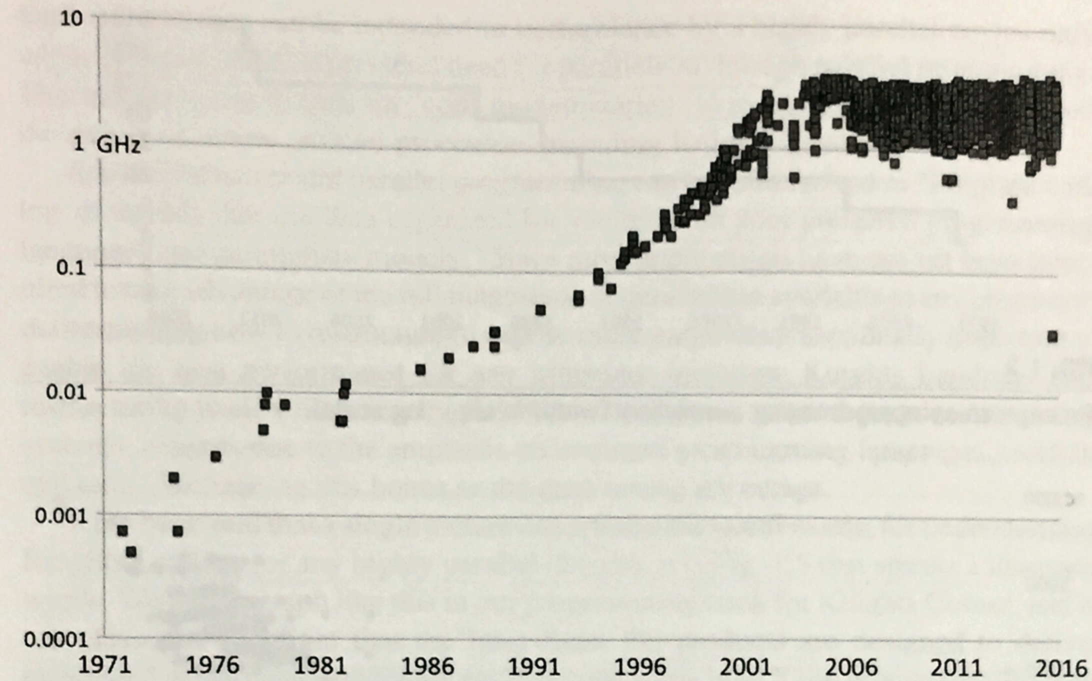


FIG. 1.1

Processor/coprocessor speed era (log scale).

J. Jeffers, J. Reinders, and A. Sodani,
Intel Xeon Phi Processor High Performance Programing

Increasing FLOP/s

Number of cores

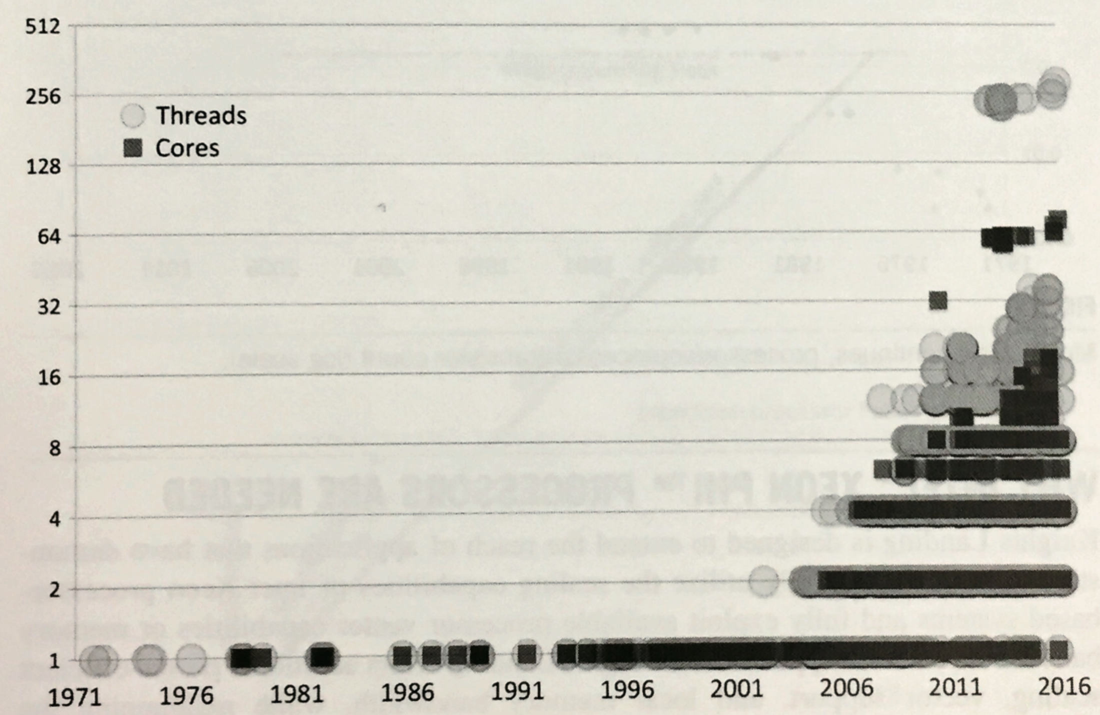


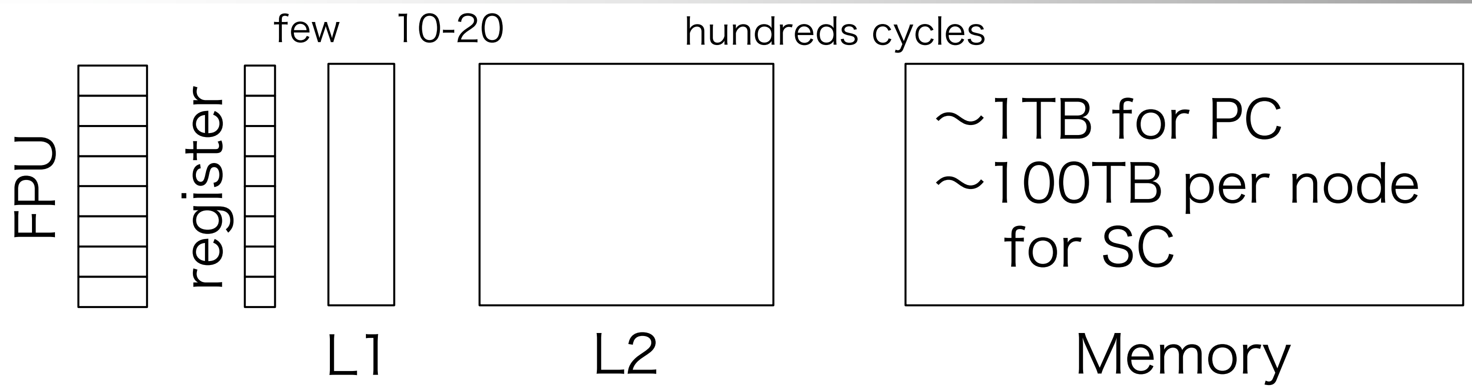
FIG. 1.2
Processor/coprocessor core/thread parallelism (log scale).

J. Jeffers, J. Reinders, and A. Sodani,
Intel Xeon Phi Processor High Performance Programing

Performance of computer

How many times does the computer multiply/add per second?
How much data does the computer memorize?
How much data does the computer read/write per second?

Cache, Memory, and Disk



Hierarchy:	Intel Xeon Phi 7250
Register	
L1 Cache	32KB per core
L2 Cache	512KB per core
Memory	∼1.6GB per core (∼10GB/s per core)
Disk	
Network	

Byte per flop (B/F)

Exercises (Not report problems)

1. Status of N qubits is represented as a complex vector in 2^N-dimensional Hilbert space. If you can use whole system of Oakforest-PACS, how many qubits can you store in the memory?

2. Usually, B/F of modern supercomputers is less than 1. Which kind of calculations is suitable for them?

Method of Evaluation

Based on 2 Reports:
-Exercise about algorithms
-Exercise of computer simulation

Closely Related Lectures

- Summer school and international symposium
  http://www.compsci-alliance.jp/
- 計算科学概論 (for undergraduate)
  Mon. 3rd period

- (archive/seminar)計算科学科学技術特論A,B,C

Exercises (Not report problems)

1. Status of N qubits is represented as a complex vector in 2^N-dimensional Hilbert space. If you can use whole system of Oakforest-PACS, how many qubits can you store in the memory?

2. Usually, B/F of modern supercomputers is less than 1. Which kind of calculations is suitable for them?

Lecture Schedule

Classical

Quantum

#1 Many-body problems in physics
#2 Why many-body problem is hard to solve
#3 Classical statistical model and numerical simulation
#4 Classical Monte Carlo method and its applications
#5 Molecular dynamics and its application
#6 Extended ensemble method for Monte Carlo methods
#7 Quantum lattice models and numerical approaches
#8 Quantum Monte Carlo methods
#9 Applications of quantum Monte Carlo methods
#10 Linear algebra of large and sparse matrices for quantum many-body problems
#11 Krylov subspace methods and their applications to quantum many-body problems
#12 Large sparse matrices and quantum statistical mechanics
#13 Parallelization for many-body problems