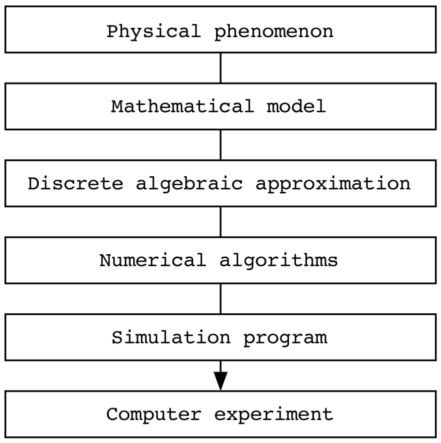
Introduction

Computer Experiment

- **Computational science**: An area of scientific investigation, where computers play a central role.
- **Scientific computing**: An area in computer science to support computational sciences by innovative use of computer systems; it involves the development of numerical algorithms, software tools, scientific visualization, etc.
- **Computational-science approach**[1,2]



1. **Mathematical model** is developed for the **physical phenomenon** of interest.

   Example: Newton's second law of motion for interacting particles. Three laws of motion were published by Isaac Newton in *Mathematical Principles of Natural Philosophy* (1686).

$$m\frac{d^2\vec{r}_i(t)}{dt^2} = \vec{F}_i(t) \quad (i = 1,...,N) \tag{1}$$

2. The equations of the mathematical model are cast into a **discrete algebraic form**, which is amenable to numerical solution.

   Example: Time discretization of the Newton's second law of motion.

$$m\frac{\vec{r}_i(t+\Delta t) - 2\vec{r}_i(t) + \vec{r}_i(t-\Delta t)}{\Delta t^2} = \vec{F}_i(t) \quad (i = 1,...,N; t = 0, \Delta t, 2\Delta t,...)$$

3. **Numerical algorithms** are used to convert the algebraic equation system into a **simulation program**.

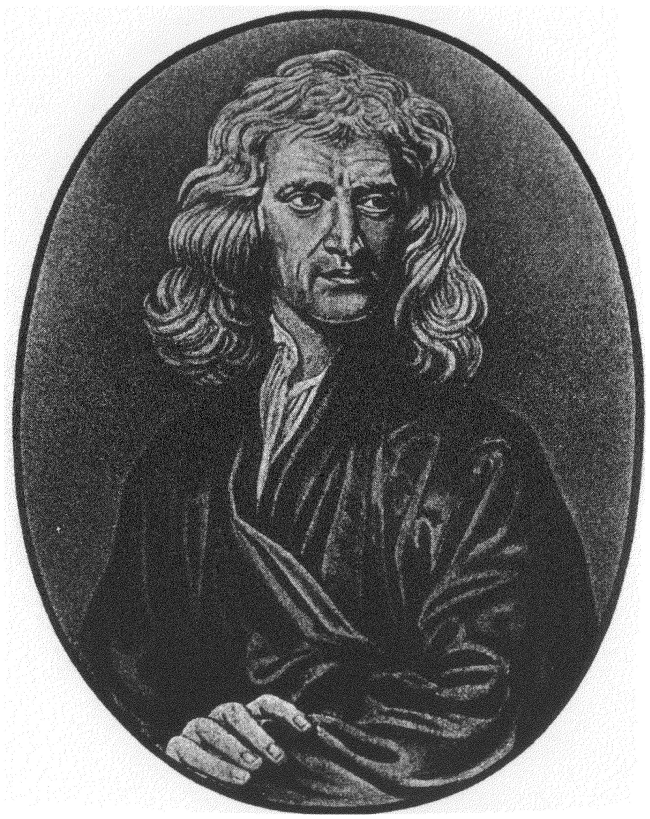
   Example 1: Verlet algorithm.

```
compute $F_i$(t) as a function of $x_i$(t)
$x_i$(t+Dt) = 2$x_i$(t) - $x_i$(t-Dt) + $F_i$(t)Dt$^2$/m
$v_i$(t) = ($x_i$(t+Dt) - $x_i$(t-Dt))/2Dt
```

   Example 2: velocity-Verlet algorithm.

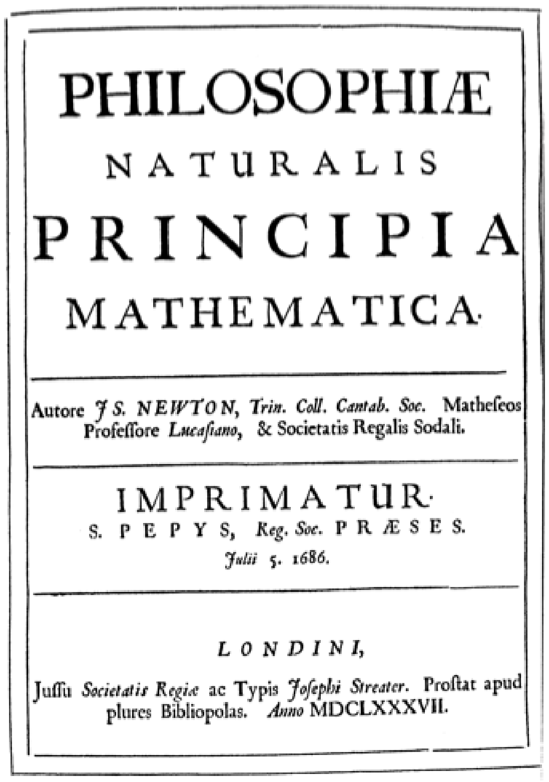
```
compute $F_i$(t) as a function of $x_i$(t)
$v_i$(t+Dt/2) = $v_i$(t) + $F_i$(t)Dt/2m
$x_i$(t+Dt) = $x_i$(t) + $v_i$(t+Dt/2)Dt
compute $F_i$(t+Dt) as a function of $x_i$(t+Dt)
$v_i$(t+Dt) = $v_i$(t+Dt/2) + $F_i$(t+Dt)Dt/2m
```

4. **Computer experiments** are performed to follow the time evolution of the model physical system.



SIR ISAAC NEWTON

(See Appendix, Note 1, page 627)



PHILOSOPHIÆ

NATURALIS

PRINCIPIA

MATHEMATICA.

Autore *J. S. NEWTON*, *Trin. Coll. Cantab. Soc.* Matheſeos Profeſſore *Lucaſiano*, & Societatis Regalis Sodali.

IMPRIMATUR.

S. PEPYS, *Reg. Soc.* PRÆSES.

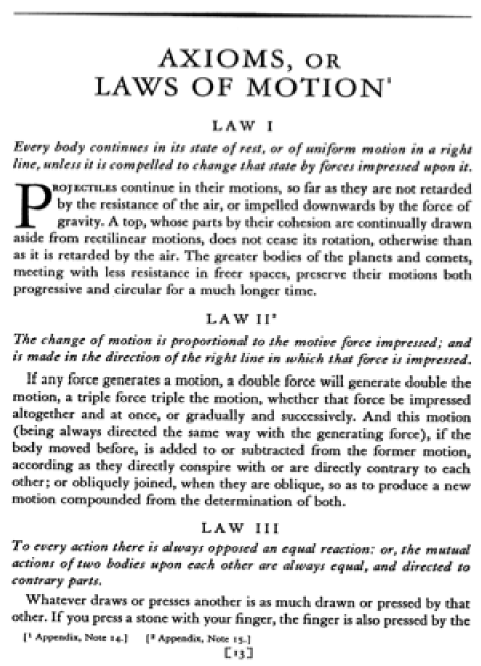
Julii 5. 1686.

LONDINI,

Juſſu *Societatis Regiæ* ac Typis *Joſephi Streater*. Proſtat apud plures Bibliopolas. *Anno* MDCLXXXVII.

TITLE PAGE OF THE FIRST EDITION OF THE PRINCIPIA

(See Appendix, Note 2, page 627)



AXIOMS, OR LAWS OF MOTION[1]

LAW I

Every body continues in its state of rest, or of uniform motion in a right line, unless it is compelled to change that state by forces impressed upon it.

PROJECTILES continue in their motions, so far as they are not retarded by the resistance of the air, or impelled downwards by the force of gravity. A top, whose parts by their cohesion are continually drawn aside from rectilinear motions, does not cease its rotation, otherwise than as it is retarded by the air. The greater bodies of the planets and comets, meeting with less resistance in freer spaces, preserve their motions both progressive and circular for a much longer time.

LAW II[2]

The change of motion is proportional to the motive force impressed; and is made in the direction of the right line in which that force is impressed.

If any force generates a motion, a double force will generate double the motion, a triple force triple the motion, whether that force be impressed altogether and at once, or gradually and successively. And this motion (being always directed the same way with the generating force), if the body moved before, is added to or subtracted from the former motion, according as they directly conspire with or are directly contrary to each other; or obliquely joined, when they are oblique, so as to produce a new motion compounded from the determination of both.

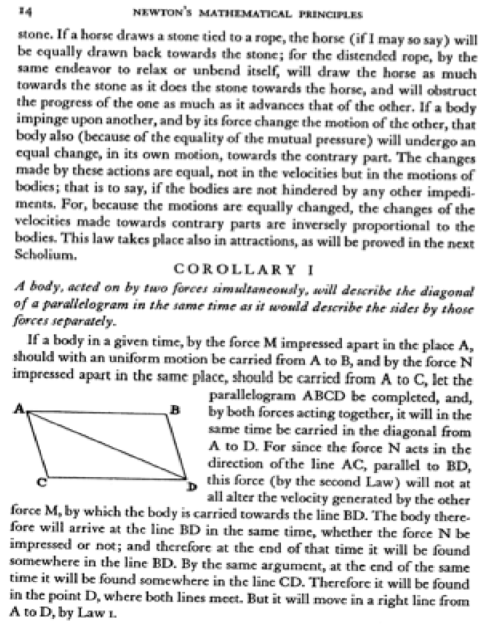
LAW III

To every action there is always opposed an equal reaction: or, the mutual actions of two bodies upon each other are always equal, and directed to contrary parts.

Whatever draws or presses another is as much drawn or pressed by that other. If you press a stone with your finger, the finger is also pressed by the

[1 Appendix, Note 14.] [2 Appendix, Note 15.]

[13]



14 NEWTON'S MATHEMATICAL PRINCIPLES

stone. If a horse draws a stone tied to a rope, the horse (if I may so say) will be equally drawn back towards the stone; for the distended rope, by the same endeavor to relax or unbend itself, will draw the horse as much towards the stone as it does the stone towards the horse, and will obstruct the progress of the one as much as it advances that of the other. If a body impinge upon another, and by its force change the motion of the other, that body also (because of the equality of the mutual pressure) will undergo an equal change, in its own motion, towards the contrary part. The changes made by these actions are equal, not in the velocities but in the motions of bodies; that is to say, if the bodies are not hindered by any other impediments. For, because the motions are equally changed, the changes of the velocities made towards contrary parts are inversely proportional to the bodies. This law takes place also in attractions, as will be proved in the next Scholium.

COROLLARY I

A body, acted on by two forces simultaneously, will describe the diagonal of a parallelogram in the same time as it would describe the sides by those forces separately.

If a body in a given time, by the force M impressed apart in the place A, should with an uniform motion be carried from A to B, and by the force N impressed apart in the same place, should be carried from A to C, let the parallelogram ABCD be completed, and, by both forces acting together, it will in the same time be carried in the diagonal from A to D. For since the force N acts in the direction of the line AC, parallel to BD, this force (by the second Law) will not at all alter the velocity generated by the other force M, by which the body is carried towards the line BD. The body therefore will arrive at the line BD in the same time, whether the force N be impressed or not; and therefore at the end of that time it will be found somewhere in the line BD. By the same argument, at the end of the same time it will be found somewhere in the line CD. Therefore it will be found in the point D, where both lines meet. But it will move in a right line from A to D, by Law I.

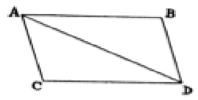


Figure. First few pages of Newton's *Mathematical Principles of Natural Philosophy* (1686).

- **Type of mathematical models**

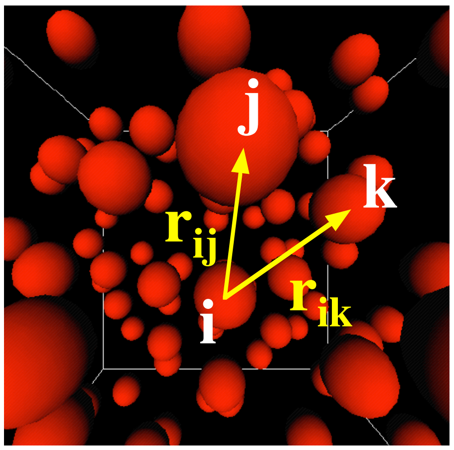
	Particle model (ordinary differential equations)	Continuum model (partial differential equations)
Deterministic	molecular dynamics	computational fluid dynamics, continuum mechanics
Stochastic	Monte Carlo particle simulation	quantum Monte Carlo

- **Particle vs. continuum models**: Mathematical models are either the particle type or the continuum type. Particle models trace the motion of many interacting particles, an example being Newton's second law of motion. Particle-type laws are typically formulated as coupled ordinary differential equations.

  Continuum models deal with functions extending over the space. For example, the dynamics of a quantum particle is described by a parabolic partial differential equation called the Schrödinger equation,

$$i\hbar\frac{\partial}{\partial t}\psi(\vec{r},t) = \left(-\frac{1}{2m}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}\right)+v(\vec{r})\right)\psi(\vec{r},t)\,, \tag{2}$$

  where $i = \sqrt{-1}$, $\hbar = 1.05\times10^{-34}$ J•s is the Planck constant, and $\psi(\vec{r},t)$ is a complex-valued wave function. The square, $|\psi(\vec{r},t)|^2$, of the wave function is the probability to find the particle at position $\vec{r}$ at time t.



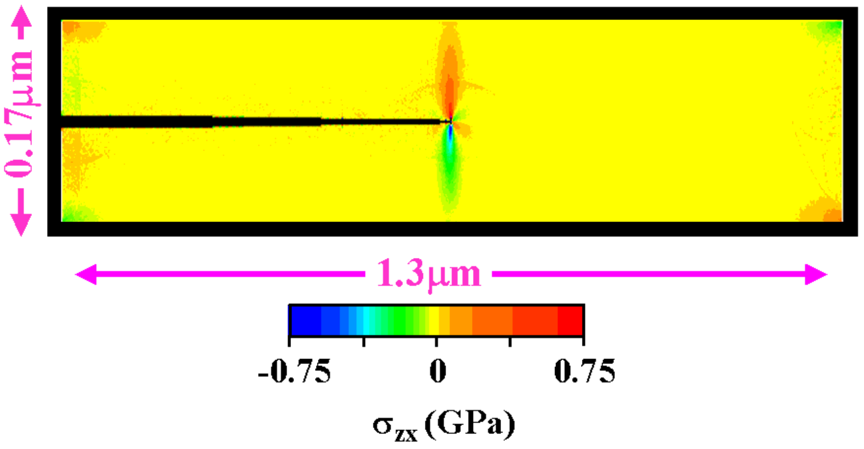


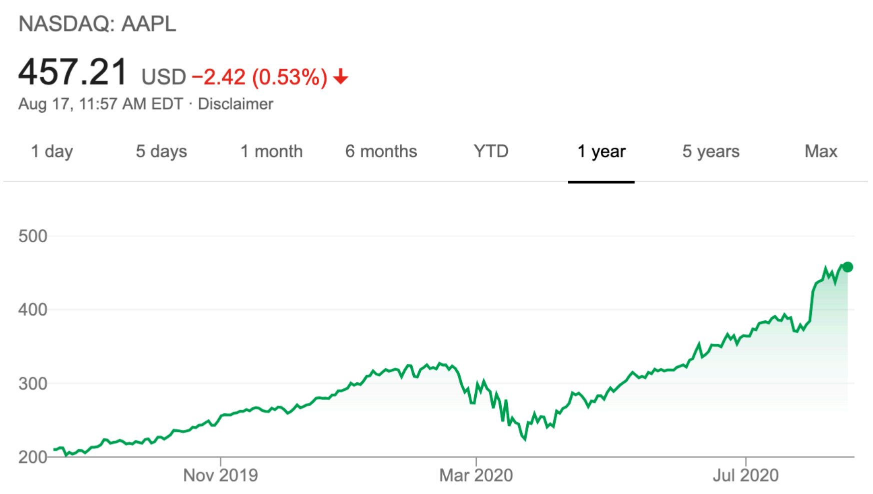
Figure. (Left) A molecular-dynamics simulation consists of a collection of atoms, which exert forces to each other, depending on their mutual interactions and relative positions. (Right) An example of continuum data—uniaxial stress field in a cracked thin plate made of gallium-arsenide material.

- **Molecular dynamics (MD)**: Follows Newton's second law of motion for interacting particles.
- **Deterministic vs. stochastic simulations**: Computer simulations are either deterministic or stochastic. Deterministic simulations usually deal with mathematical initial value problems, i.e., differential equations such as Eqs. (1) and (2) are integrated forward in time starting with some initial configuration.

  Stochastic simulations use random numbers to: 1) provide approximate solutions to large-scale problems where deterministic solutions are intractable (e.g., statistical mechanics in physics); or 2) simulate stochastic natural phenomena (e.g., stock price).

- **Monte Carlo (MC) method**: A computational method that utilizes random numbers.

    Example: Stock price.


NASDAQ: AAPL

457.21 USD −2.42 (0.53%) ↓

Aug 17, 11:57 AM EDT · Disclaimer

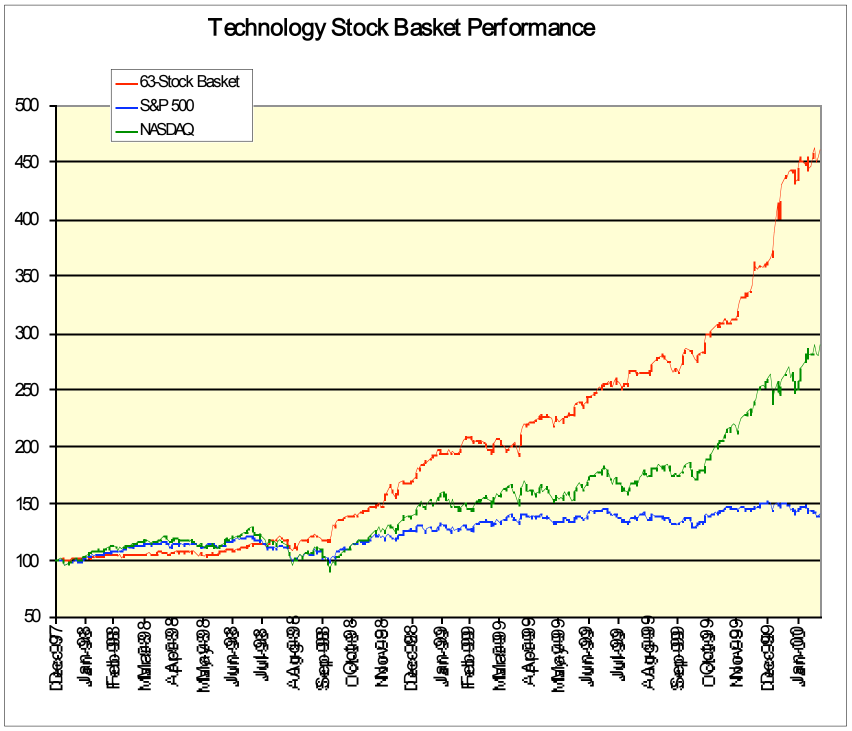


Figure. (Top) Typical fluctuation in stock price. (Bottom) Stock portfolio trading at Quantlab Financial LLC (courtesy of Dr. Andrey Omeltchenko, CACS graduate).

Application of Molecular Dynamics

- Drug design
- Materials design
- Robotics
- Computer graphics[1]
- Games

[1]One example of the use of particle systems for computer animation is found in *Twister* (1996), in which particle systems are used to simulate a tornado.

Figure. (Top) A scene from movie *Twister*. (Bottom) Particle-based dust visualization by Prof. Jim Chen at George Mason University (`http://cs.gmu.edu/~jchen`).

History of Particle Simulations

1944 John von Neumann was attracted to the ENIAC project (the world's first operational electronic, general-purpose computer); he wrote a memo proposing a stored-program computer. He believed: "Our present analytical methods seem unsuitable for the solution of the important problems arising in connection with nonlinear partial differential equations. The really efficient high-speed computing devices may provide us with those heuristic hints which are needed in all parts of mathematics for genuine progress."

1953 The first MC simulation of a liquid by Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller performed on the MANIAC computer at Los Alamos National Laboratory.

1955 Enrico Fermi, John Pasta, and Stanislaw Ulam studied the dynamics of a one-dimensional array of particles coupled by anharmonic springs on the MANIAC.

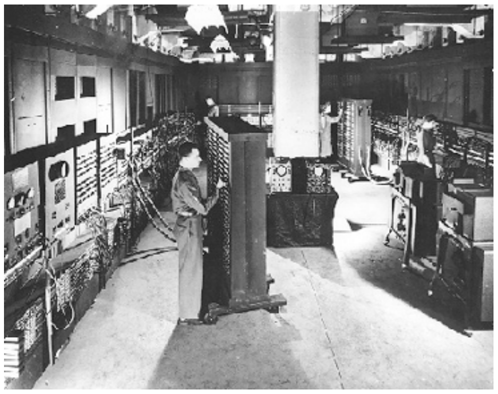
1956 Dynamics of hard spheres (billiards) studied by Alder and Wainwright at the Lawrence Livermore National Laboratory.

1960 Radiation damage in crystalline Cu studied with short-range repulsion and uniform attraction toward the center by George Vineyard's group at the Brookhaven National Laboratory.

1964 The first MD simulation of a liquid (864 argon atoms) using interatomic potentials by Aneesur Rahman at the Argonne National Laboratory using a CDC 3600 computer.

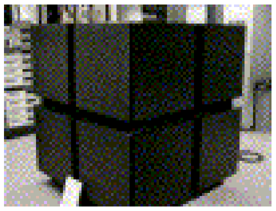
Current state-of-the-art MD simulations consist of over trillion atoms.[3,4] For example, an MD program developed at the Collaboratory for Advanced Computing and Simulations (CACS) at the University of Southern California (USC) has simulated material consisting of 4.95 trillion atoms using 786,432 IBM BlueGene/Q processors. For a harder problem—atomistic simulation in which interatomic forces are computed quantum mechanically, CACS scientists have performed 39.8 trillion electronic degrees-of-freedom calculation.

History of Supercomputers



The world's first general-purpose electronic computer was ENIAC (see the figure in the right) built by Presper Eckert and John Mauchly at the University of Pennsylvania during World War II. The computer was intended to be used for calculating ballistic trajectories—a kind of particle dynamics. (Eckert was only 22 years old when he and Mauchly started the project with $half-million support from Army.) However, rewiring this computer to solve a new problem required days of work by a number of operators.

The era of **vector supercomputers** started in 1976 when Seymour Cray built Cray-1 (see the figure in the left). Vector processing is a type of **parallelism**, which speeds up computation.



In late 80's, **massively parallel computers** such as the Thinking Machines CM-2 (right) became the central technology for supercomputing. The Cray T3E computer, for example, is a collection of 450MHz Digital Alpha processors connected to form three-dimensional grids.

Another important development is the invention of the **microprocessor**—a computer on a single semiconductor chip. The first microprocessor, Intel 4004 introduced in 1971, contained 2300 transistors and was 0.3 by 0.4 cm in size. In comparison, Pentium 4 contains 125 million transistors. Intel had a 16-bit processor two years before its competitors' more elegant architectures, such as Motorola 68000, and this head start led to the selection of 8086 as the CPU for the IBM PC in 1981. (Andy Grove, the CEO of Intel, was the '97 TIME Man of the Year.)

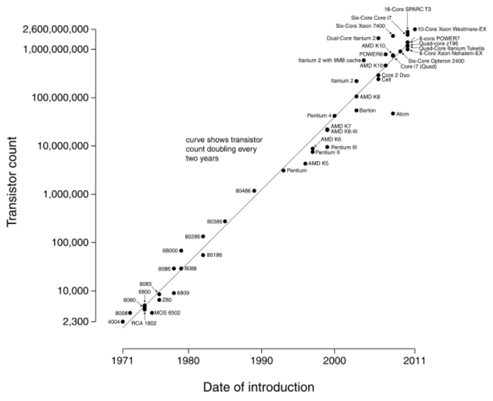


Figure. Dramatic increase of the number of transistors in a microprocessor.

Merge of PC and supercomputer technologies: The later trend in computer technology was that the PC and supercomputer technologies are merging. For example, we acquired a Linux cluster consisting of 512 dual Intel Xeon 1.8 GHz nodes (i.e., 1,024 processors) connected by Myricom's Myrinet interconnect at Louisiana State University (LSU) in 2002. The performance of the $2.6 million cluster, *SuperMike*, was rated as 2.21 teraflops (10^{12} floating-point operations per second), according to the standard High Performance Linpack (HPL) benchmark (`http://www.netlib.org/benchmark/hpl`), and *SuperMike* was ranked as the 11th fastest supercomputer in the world in August 2002 (`http://www.top500.org`).

Figure. The 1,024-processor Xeon cluster, *SuperMike*, at LSU.

The current world's fastest supercomputer is the Fugaku computer at RIKEN in Japan. Fugaku consists of 7.3 million A64FX ARM cores, and its theoretical peak performance is 514 petaflops (1 petaflops = 10^{15} floating-point operations per second;[5] see the figure). The actual measured performance of Fugaku for the Linpack benchmark program is 416 petaflops.

Figure. 416 petaflops Fugaku.

You can also build PC clusters by yourself.[6,7] The CACS at the USC has a 4,096-processor PC cluster. The Center for Advanced Research Computing (CARC) at the USC has a 13,440-processor Linux cluster with 0.62 petaflops Linpack performance, which you will use in this class.

Beginning of the many-core parallel computing era: Computer industry is facing a historical shift, in which Moore's law due to ever increasing clock speeds has been subsumed by increasing numbers of "cores" per microchip. Intel has earlier demonstrated an 80-core microchip that executed trillion operations per second with mere 62 W of power (`http://en.wikipedia.org/wiki/Teraflops_Research_Chip`), and the number of cores per microchip is expected to double at each generation, reaching thousands in 10 years. In addition to such many-core central processing units (CPUs), advanced graphics processing units (GPUs) in desktop computers will add extra multi-trillion operations with minimal (~ $ hundred) cost. The many-core revolution will mark the end of the free-ride era (i.e., legacy software will run faster on newer chips), resulting in a dichotomy—subsiding speed-up of conventional software and exponential speed-up of scalable parallel-computing applications. To develop a many-core CPU/GPU computing framework applied to broad applications, we have constructed a test bed consisting of a cluster of 9 Playstation3 consoles (each Playstation3 box containing one power processing and 8 streaming processing units as well as a GPU) at CACS (see the figure).

Figure. Playstation3 cluster at CACS.

Enabling Technologies

Parallel Computing

Parallel computing technology has extended the scope of computer simulations in terms of simulated system size. In order to perform parallel computer simulations efficiently, however, algorithms developed for serial computers must often be modified. We will learn parallel MD algorithms in this lecture.[8,9]

Parallel computing requires decomposing the computation into subtasks and mapping them to multiple processors. For MD simulations, the divide-and-conquer strategy based on spatial decomposition is commonly used. The total volume of the system is divided into *P* subsystems of equal volume, and each subsystem is assigned to a node in an array of *P* processors (see the figure below).[10] In this lecture, you will learn message passing interface (MPI) programming on multiple computers and thread (OpenMP) programming on multicore processors. In addition, a brief overview will be given on the programming on Cell Broadband Engine on PlayStation3 consoles,[11] GPUs using compute unified device architecture (CUDA),[12] and prototype multicore processors such as 64-core Godson-T.[13]

Visualization

Impact of these grand-challenge simulations on parallel computers cannot be fully realized without major breakthroughs in scientific visualization. The current practice of sequentially processing visualization data is highly ineffective for large-scale applications that produce terabytes of data. The only viable solution is to integrate visualization into simulation so that they are both performed concurrently on multiple parallel machines and then to examine the results in real time in three-dimensional immersive and interactive virtual environments.

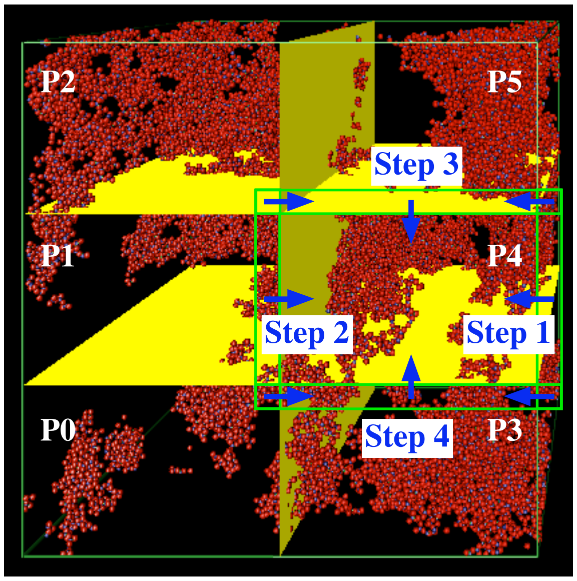


Figure. Spatial decomposition (2 × 3 × 1) of a porous silica material into 6 systems, which are mapped onto 6 processors (*P*0 - *P*5). Blue and red spheres represent silicon and oxygen atoms, respectively. Logical partition boundaries between subsystems are represented by yellow planes. Blue arrows represent flow of data by message passing between processors.

Virtual environment such *ImmersaDesk* (see the figure below) enables direct interaction with large datasets in three dimensions. Visual feedback allows us to zoom in on part of a simulation, change parameters, and thus guide the progress of a simulation in real time. Furthermore, with virtual environment the observer is inside the simulation instead of getting a glimpse of the data from the outside. These immersive and interactive features provide invaluable insight into the simulations. In this lecture, we will learn OpenGL programming as a foundation of scientific visualization, as well as virtual-reality programming using CAVE Library. In addition to an *ImmersaDesk*, the CACS has an 8' × 14' tiled display, which is driven by a 26-processor Linux cluster (see the figure below).

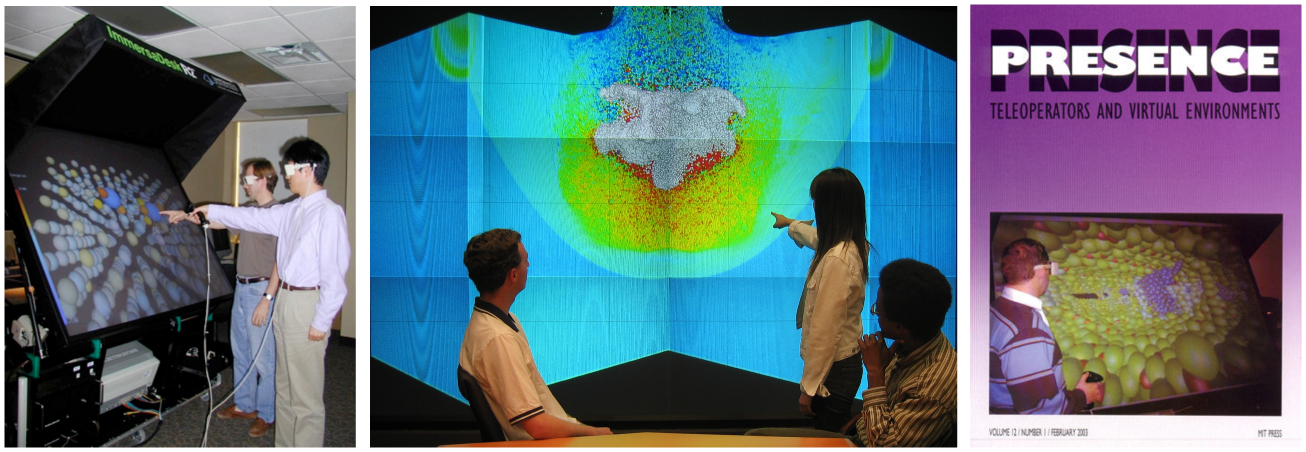


Figure. (Left) ImmersaDesk virtual environment at the CACS showing oxidation of an Al surface. (Center) Rendering of a molecular dynamics simulation on our tiled display to study hypervelocity impact damage. (Right) A scientist immersed in an atomistic model of a fractured ceramic nanocomposite material, featured on the cover of journal *Presence* (MIT Press).

CACS scientists have demonstrated real-time, interactive visualization of a billion-particle dataset in an immersive virtual reality environment, using advanced computer-science techniques such as: 1) multilevel view-frustum culling; 2) probabilistic occlusion culling; and 3) parallel/distributed preprocessing of visualization data on a remote PC cluster.[14,15]

Data Management

A serious technological gap exists between the growth in processor power and that of input/output (I/O) speed. Typically, a billion-atom MD simulation produces 0.1 terabytes (TB) of data (to store atom types, coordinates, velocities, and stresses) per frame, which amounts to 10 TB per day if the simulation runs for 1,000 steps and data are saved after every 10 steps. The I/O (including data transfer to remote archival storage devices) has thus become the bottleneck in large-scale simulations. Efficient schemes to manage these massive data are crucial. In this lecture, we will learn data compression approaches for the I/O problem.[16]

Visualization of collective motion of many atoms is a difficult task because of the high dimensionality ($3N$ dimensions for N atoms) of the space in which the collective motion occurs. A challenge is to extract topological defects, such as dislocations, and their activities from massive data with large thermal noises, especially at high temperatures. This will require nontrivial knowledge discovery or data-mining processes from very large noisy data sets.[17] We will learn the use of data-mining algorithms using graph data structures,[18,19] which is at the heart of recent federal and industrial initiatives on "Big Data".[20]

Distributed Computing

Often a single parallel supercomputer does not provide sufficient computing power for grand challenge problems. In such a case, "metacomputing" uses multiple supercomputers connected by wide-area, high-speed networks. The "Grid" of geographically distributed petaflops computers and immersive/interactive virtual reality environments connected via high-speed networks can revolutionize/democratize science, by enabling hybrid simulations that integrate multiple expertise distributed globally.[21] Such collaborative Grid computing will form the core of the new initiative on "Revolutionizing Science and Engineering through Cyberinfrastructure".[22]

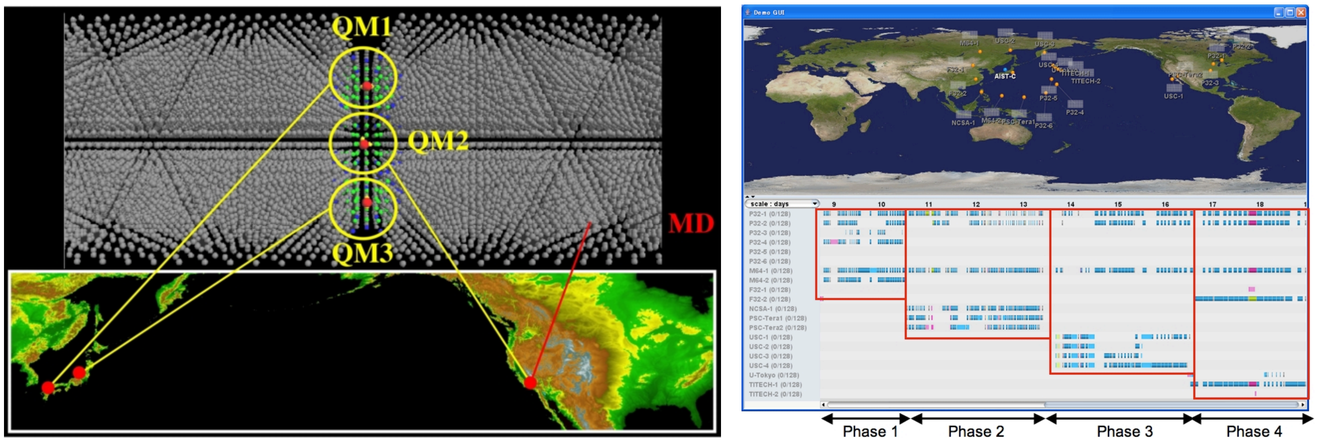


Figure. (Left panel) Multiscale MD/QM simulation of the reaction of water at a crack tip in silicon (top), on a Grid of distributed PC clusters in Japan and the US (bottom). (Right panel) Time chart (bottom) of an adaptive QM/MD simulation performed on globally distributed parallel supercomputers in the US (USC, PSC, NCSA) and Japan (AIST, Univ. of Tokyo, TITech). The blue line denotes the execution of QM simulation, the red line shows the failure in the initialization phase, and the light blue line the failure in the simulation phase.

Such a multidisciplinary application is emerging at the forefront of computational sciences. The multiscale simulation embeds accurate quantum mechanical (QM) calculations to handle chemical reactions within a molecular dynamics (MD) simulation to describe large-scale atomistic processes. CACS scientists have performed a preliminary MD/QM simulation on a Grid of distributed PC clusters in Japan and the US (see the figure below), in collaboration with Japanese scientists.[23] More recently, we have performed a larger Grid MD/QM simulation on a Grid consisting of 6 supercomputing centers in the US (USC and two NSF TeraGrid nodes at the Pittsburgh Supercomputing Center and the National Center for Supercomputing Applications) and Japan (National Institute of Advanced Industrial Science and

Technology, University of Tokyo, and Tokyo Institute of Technology). The simulation was sustained autonomously on ~700 processors for 2 weeks, involving in total of 150,000 CPU-hours, where the number of processors changed dynamically on demand and computations were migrated automatically according to both reservations and unexpected faults.[24] A recent trend in distributed computing is "cloud",[25] which has gained popularity in commerce and will be covered briefly in the class.

References

1. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (Adam Hilger, Bristol, 1988).
2. A. Nakano, "Physics computing," in *Encyclopedia of Electrical and Electronics Engineering, Vol. 16*, ed. J. G. Webster (John Wiley & Sons, NY, 1999) p. 420; online update, http://onlinelibrary.wiley.com/doi/10.1002/047134608X.W1675.pub2/full (2007).
3. A. Nakano, R. K. Kalia, K. Nomura, A. Sharma, P. Vashishta, F. Shimojo, A. C. T. van Duin, W. A. Goddard, III, R. Biswas, and D. Srivastava, "De novo ultrascale atomistic simulations on high-end parallel supercomputers," *Int'l J. High Performance Comput. Appl.* **22**, 113 (2008); K. Nomura, H. Dursun, R. Seymour, W. Wang, R. K. Kalia, A. Nakano, P. Vashishta, F. Shimojo, and L. H. Yang, "A metascalable computing framework for large spatiotemporal-scale atomistic simulations," in *Proc. of International Parallel and Distributed Processing Symposium, IPDPS 2009,* (IEEE, Rome, Italy, 2009); M. Kunaseth, R. K. Kalia, A. Nakano, K. Nomura, and P. Vashishta, "A scalable parallel algorithm for dynamic range-limited *n*-tuple computation in many-body molecular dynamics simulation," in *Proc. of Supercomputing, SC13* (ACM/IEEE, New York, NY, 2013).
4. F. Shimojo, R. K. Kalia, A. Nakano, and P. Vashishta, "Embedded divide-and-conquer algorithm on hierarchical real-space grids: parallel molecular dynamics simulation based on linear-scaling density functional theory," *Comput. Phys. Commun.* **167**, 151 (2005); "Metascalable molecular dynamics simulation of nano-mechano-chemistry," *J. Phys.: Condensed Matter* **20**, 294204 (2008); F. Shimojo, S. Hattori, R. K. Kalia, M. Kunaseth, W. Mou, A. Nakano, K. Nomura, S. Ohmura, P. Rajak, K. Shimamura, and P. Vashishta, "A divide-conquer-recombine algorithmic paradigm for large spatiotemporal quantum molecular dynamics simulations," *J. Chem. Phys.* **140**, 18A529 (2014); K. Nomura, R. K. Kalia, A. Nakano, P. Vashishta, K. Shimamura, F. Shimojo, M. Kunaseth, P. C. Messina, and N. A. Romero, "Metascalable quantum molecular dynamics simulations of hydrogen-on-demand," in *Proc. of Supercomputing, SC14* (ACM/IEEE, New York, NY, 2014).
5. R. F. Service, "Design for US exascale computer takes shape", *Science* **359**, 617 (2018).
6. T. L. Sterling, J. Salmon, D. J. Becker, and D. F. Savarese, *How to Build a Beowulf* (MIT Press, Cambridge, MA, 1999); T. L. Sterling, *Beowulf Cluster Computing with Linux* (MIT Press, Cambridge, MA, 2002).
7. R. Buyya, *High Performance Cluster Computing*, *Vol. 1 and 2* (Prentice Hall, Upper Saddle River, NJ, 1999).
8. D. C. Rapaport, *The Art of Molecular Dynamics Simulation*, *2nd Ed.* (Cambridge Univ. Press, Cambridge, 2004).
9. A. Nakano, R. K. Kalia, and P. Vashishta, "Multiresolution molecular dynamics algorithm for realistic materials modeling on parallel computers," *Comput. Phys. Commun.* **83**, 197 (1994).
10. A. Nakano, "Multiresolution load balancing in curved space: the wavelet representation," *Concurrency: Practice and Experience* **11**, 343 (1999).
11. K. Nomura, S. W. de Leeuw, R. K. Kalia, A. Nakano, L. Peng, R. Seymour, L. H. Yang, and P. Vashishta, "Parallel lattice Boltzmann flow simulation on a low-cost PlayStation3 cluster," *Int'l J. Comput. Sci.* **2**, 437 (2008); L. Peng, R. Seymour, K. Nomura, R. K. Kalia, A. Nakano, P. Vashishta,

A. Loddoch, M. Netzband, W. R. Volz, and C. C. Wong, "High-order stencil computations on multicore clusters," in *Proc. of International Parallel and Distributed Processing Symposium, IPDPS 2009* (IEEE, Rome, Italy, 2009).

12. L. Peng, K. Nomura, T. Oyakawa, R. K. Kalia, A. Nakano, and P. Vashishta, "Parallel lattice Boltzmann flow simulation on emerging multi-core platforms," *Lecture Notes in Computer Science* **5168**, 763 (2008).
13. L. Peng, G. Tan, R. K. Kalia, A. Nakano, P. Vashishta, D Fan, H. Zhang, and F. Song, "Scalability study of molecular dynamics simulation on Godson-T many-core architecture," *J. Parallel Distrib. Comput.* **73**, 1469 (2013).
14. A. Sharma, A. Nakano, R. K. Kalia, P. Vashishta, S. Kodiyalam, P. Miller, W. Zhao, X. Liu, T. J. Campbell, and A. Haas, "Immersive and interactive exploration of billion-atom systems," *Presence: Teleoperators and Virtual Environments* **12**, 85 (2003).
15. C. Zhang, S. Callaghan, T. Jordan, R. K. Kalia, A. Nakano, and P. Vashishta, "ParaViz: a spatially decomposed parallel visualization algorithm using hierarchical visibility ordering," *Int'l J. Comput. Sci.* **1**, 407 (2007).
16. A. Omeltchenko, T.J. Campbell, R.K. Kalia, X. Liu, A. Nakano, and P. Vashishta, "Scalable I/O of large-scale molecular-dynamics simulations: a data-compression algorithm," *Comput. Phys. Commun.* **131**, 78 (2000).
17. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining* (MIT Press, Cambridge, MA, 1996); J. Han and M. Kamber, *Data Mining: Concepts and Techniques, 2nd Ed.* (Morgan Kaufmann, San Francisco, CA, 2005).
18. D. J. Cook and L. B. Holder, *Mining Graph Data* (Wiley, Hoboken, NJ, 2007).
19. C. Zhang, B. Bansal, P. S. Branicio, R. K. Kalia, A. Nakano, A. Sharma, and P. Vashishta, "Collision-free spatial hash functions for structural analysis of billion-vertex chemical bond networks," *Comput. Phys. Commun.* **175**, 339 (2006).
20. R. E. Bryant, R. H. Katz, and E. D. Lazowska, "Big-data computing: creating revolutionary breakthroughs in commerce, science, and society," http://www.cra.org/ccc/docs/init/Big_Data.pdf.
21. I. Foster and C. Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure* (Morgan Kaufmann, San Francisco, 2003).
22. National Science Foundation Advisory Committee for Cyberinfrastucture task force reports (`http://www.nsf.gov/cise/aci/taskforces`).
23. H. Kikuchi, R. K. Kalia, A. Nakano, P. Vashishta, H. Iyetomi, S. Ogata, T. Kouno, F. Shimojo, K. Tsuruta, and S. Saini, "Collaborative simulation Grid: multiscale quantum-mechanical/classical atomistic simulations on distributed PC clusters in the US and Japan," in *Proc. of Supercomputing, SC02* (IEEE/ACM, Los Alamitos, CA, 2002).
24. H. Takemiya, Y. Tanaka, S. Sekiguchi, S. Ogata, R. K. Kalia, A. Nakano, and P. Vashishta, "Sustainable adaptive Grid supercomputing: multiscale simulation of semiconductor processing across the Pacific," in *Proc. of Supercomputing, SC06* (IEEE/ACM, Los Alamitos, CA, 2006); Y. Song, Y. Tanaka, H. Takemiya, H. Nakada, S. Sekiguchi, A. Nakano, and S. Ogata, "The development and evaluation of an integrated framework supporting sustainable execution for large-scale computations on Grids," *Int'l J. Comput. Sci.* **3**, 18 (2009).
25. K. Hwang, J. Dongarra, and G. C. Fox, *Distributed and Cloud Computing: From Parallel Computing to the Internet of Things* (Morgan Kaufmann, Burlington, MA, 2011).