2004 NSF Division of Materials Research ITR Workshop
The Materials Computation Center
Funded by NSF DMR 03-25939

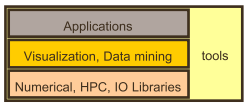
Duane D. Johnson and Richard M. Martin (PIs)

Materials Simulation Toolkits

Jeongnim Kim, D. M. Ceperley, R. M. Martin and D. D. Johnson
Students: J. Vincent and D. Das
National Center for Supercomputing Applications and the Departments of Physics and Materials Science and Engineering,
University of Illinois at Urbana-Champaign

Motivations and goals

- Ever-growing capability of hardware, combined with new software engineering technology
  - powerful materials simulation tools
- Use of standard, open-source software
  - dynamic, maintainable and adaptable scientific code development
- Use of standard IO
  - communications between diverse applications
- Use of standard tools
  - portability in diverse and changing HPC environment



Vision for **Materials Simulation Toolkits**

Guiding principles:
- **Standard** libraries and components.
- **Open source** development

Status:
- OHMMS, qmcplusplus, atomicHF, TBPW. …
- Standard IO based on XML/HDF5
- Data analysis and visualization tools: dataspork, matsimviz
- Open source utility tools

Standard tools for portable MS Toolkits

Why tools are important:
- No standard environment exists for scientific software.
- Conventional desktop package utility (e.g., rpm) is not suitable for HPC systems.
- Needs automatic configuration but specialization for the hardware and software.

What tools are utilized:
- C/C++ compilers (ANSI standard)
- OpenMP capable compilers: intel, ibm, sgi, compaq/hp
- doxygen for code documentation
- GNU autoconf/automake/libtool
- CMake (Cross Platform Makefile): http://www.cmake.org

Toward Standard IO

- Communications between applications can benefit from standard IO and tools.
- Light-weight IO: XML
  - Job management and project descriptions.
  - Setting up complex simulations.
  - Human-readable and easily transformed to various formats.
  - Solution for the complex problems of analyzing, mining, storing, and accessing vast stores of data.
- Data-intensive IO: HDF (http://hdf.ncsa.uiuc.edu)
  - Efficient IO on serial and parallel computers.
  - Facilitate scientific data exchange, access, analysis, archiving and discovery.

Epilogue

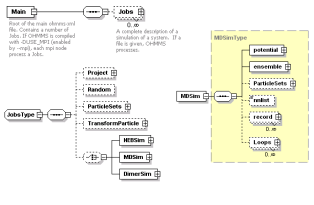
The Open-Source development community has a role to create new powerful tools to facilitate collaborations among the developers and users. Such tools enable effective management of resources and knowledge. In a leading role, the Materials Computation Center is developing software infrastructure that:

- takes full advantage of the advances in *algorithms* and *computing power.*
- adopts development protocol of Open-Source Projects.
- uses documentation with xml-based open-source standards.

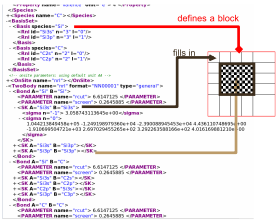
This tools-based approach enables faster "learning-curve" for students and research scientists to utilize and/or integrate code for their research needs, leading to more rapid progress in developing, testing, and applying codes and algorithms.

XML in action for materials simulations

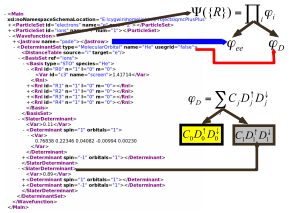
- Portable input and output formats
  - Atomistic simulations (OHMMS): molecular dynamics, free-energy, dimer simulations.
  - Input files for empirical classical and tight-binding potentials.
  - *DFT (basis functions, pseudopotentials etc)*
- Multi-level job and processor controls → Grid Computing.
  - SOAP (Simple Object Access Protocol)
  - Mixing scripting languages and programming languages.
- Hierarchical IO that mirrors the data and interfaces of an application.
  - Static and dynamic code generations.
  - Instantiation of template classes tailored to an application: physical dimension, precision, system size.



Execution of materials simulations



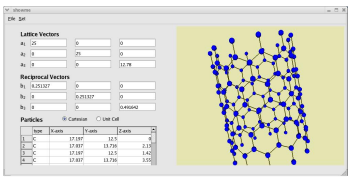
Mapping of input on the objects: standard for empirical tight-binding potentials



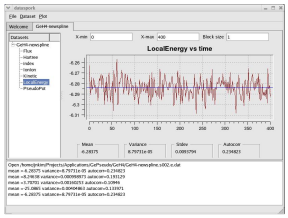
qmcPlusPlus: abstraction for a many-body electron wavefunction of a He atom.

Visualization, analysis and mining of materials simulation data

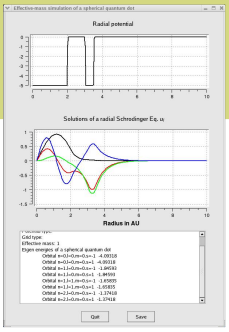
- Portable, efficient tools are developed to
  - Design, manipulate and visualize complex systems
  - Analyze massive data from multiple runs on multiple platforms
  - Build large-scale database for materials simulations
  - Provide knowledge database and gateway to software, data & literatures
- Utilizing open source libraries for cross-platform applications
  - OpenGL, Qt, libxml, perl, python …..
  - Ported on multiple platforms: linux, Mac OS X, Windows/cygwin
  - Documentation by docbook and doxygen



Visualization tool: presenting the atomic structure of a carbon nanotube and GUI to manipulate the crystalline/molecuar system.

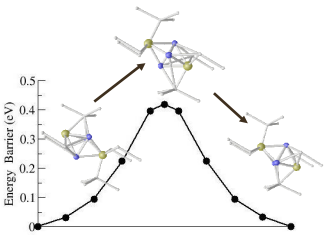


Analysis tool: presenting the trace and autocorrelation analysis of the local energy of GeH_4 during Diffusion Monte Carlo Simulation.

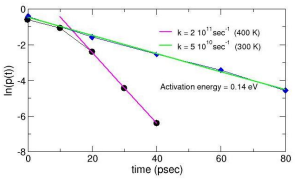


Educational tool: presenting the electronic structure of a spherical multi-shell quantum dot within Hartree-Fock approximation

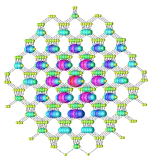
Image Gallery: Simulation snapshots, HPC performance profile



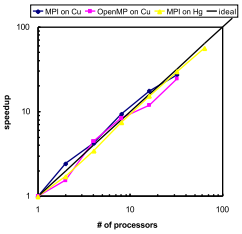
Energy barrier along the diffusion path of a Si-interstitial defect quantified by nudged elastic-band method [to be published in ACESS magazine J. Kim, 2004].



Accelerated dynamics simulations of rare events: diffusion constants of a vacancy in bulk Si predicted by parallel-replica simulation using empirical tight-binding potential [J. Kim, *et al.* 2003].



Atomic charge density of a negatively charged InAs nanocrystal [S. Lee, J. Kim, *et al.* 2003].



Materials simulation on HPC: parallel scaling of large-scale sparse eigen value problems on IBM pSeries 690, 32-way node and Intel Itanium2 [All hands meeting, J. Fettig, NCSA, 2003].