0.1 Setting Up Computing Environment

https://github.com/ankitjainmeiitk

Wikibooks home

Listing 1: Some Code

```
public void here() {
        goes().the().code()
}
```

80% of supercomputers using linux ubuntu: most popular, well-documented, large-community discussion OS-supported hardware: lenovo, HP, dell, apple OS is an acceptable substitute given that it is linux kernel.

setting up environment: $PATH, sudo apt-get install gfortran, which gfortran gnu compilers vs intel: gnu freely available tar -zxvf lammps.tar, make serial, sudo apt-get install openmpi, make openmpi

suggested reading:

Statistical Mechanics/Condensed Matter MacQuarrie Ashcroft Wikipedia

Lattice Dynamics Dove: harmonic autocorrelation (DOS), harmonic (dispersion), thermodynamics, structural phase transition, Srivistava: quantum formalism, perturbation theory peierls: ziman: BTE, RTA Wikipedia

Quantum Chemistry Griffiths Quantum Mechanics: Pauli-Exclusion -¿ Exp repulsion Martin Electron Strucutre: DFT Wikipedia

Math Theory/Numerical Methods MacQuarrie Moin Wikipedia

Wikipedia == open source

list of useful packages: pacakges used during this thesis: open-source: lammps, gulp, abinit, qe, vesta, phonopy, ntpy other: vasp

other packages: dlpoly, siesta, GAMES, cp2k, boltztrap, PHON,

Useful Linux Commands

google: linux sed

sed, grep, ls, cd, pwd, export/setenv, scp -r, ssh, sudo, nohup, vi, cat, which (program), echo, qstat -u -n, qdel,

Running Linux System Commands in Python

python vs other languages: combines strengths of the 3 important: PERL+java+matlab 1) shell-scripting: controlling the OS, system(cmd), (PERL)

lmp.in.iseed

2) hi-level organization: object-oriented (module.method(input)) (Java) 3) lin alg + numerical methods: numpy.method, scipy.method (fortran, matlab) python deficiencies more about this in the appendix

0.2 Coding Languages

linux/unix shell

excellent c++ tutorial

excellent fortran tutorial

matlab has an excellent built-in guide, google search will typically yield useful results.

matlab
octave

0.3 Online Resources

wikipedia
octave
https://en.wikipedia.org/wiki/Lennard-Jones$_potential$
http://www.sklogwiki.org/SklogWiki/index.php/Lennard-Jones$_model$
we can do better, needs to be organization.
http://nanohub.org/
does not provide good HPC resources.

0.4 Latex Resources

http://tex.stackexchange.com/questions/1596/how-to-compile-a-latex-document
http://stackoverflow.com/questions/2461905/compiling-latex-bib-source
http://publish.aps.org/revtex
used
http://kile.sourceforge.net/
http://www.zotero.org/

0.5 Google Searching

Web-searching.
keywords: integrated development environment (IDE),

0.6 Symmetry Discussion

rub abinit for si, get S
$k_{ij} = k_{ji}$ and $k_{ii} = kjj$ show using symmetry operations
$S = [-100; 010; 001]$
This applies to any vector or tensor property of the crystal.
[?]
lennard-jones FCC primitive conventional mapping

0.6.1 KPT symmetries

property(k) = property(-k) for all
show how kpts are reduced to first octant using just rotations
show how kpts in first octant are reduced.

0.7 Normal Mode Decomposition

harmonic assumption MD is anharmonic

0.7.1 Eigvec mapping

0.8 VC-NMD versus Gamma-NMD

1 Computability

research.idea = 'string'

2 Unsorted

the smartest ppl ion the world shud be using the same shit

3 Perspectives

3.1 Preparing Journal Articles

1) Use latex - reccomend Kile editor, gedit works well too. - reccomend zotero to maintain library.bib

1) Time to prepare each paper

reccomendation: student advisor should try and exchange editing a written research document every day, at least every week.

Such a research document could be the running collection of journal articles which turn into the thesis. Maintenance of this document can be achieved with Dropbox or Github. Github offers to advantage of smart version control and internet wiki.

2) Computational cost of each paper

CPU, memory parallel cpu, parallel memory

Trends: small single cpu jobs -¿ large parallel cpu jobs single predictive method -¿ several predictive methods

reccomendation: explore all predictive methods and models possible

3.2 Plottable Code

It is important to explore various predictive methods visually. reccomendation: matplotlib provides an enormous number of plotting tools.

Here is a simple example using the python wrapper for lammps with matplotlib to visualize. An MD simuation of LJ argon is visualized. An atom is placed into an unusually high energy position, which initiates melt to the liquid phase.

matplotlib, code version with and without plotting capability.

3.3 Scaling into Multi-CPU Dimension

3.3.1 Python w/o shared Memory

3.3.2 Python w/ shared Memory (Parallel Python)

3.3.3 Compiled Code w/ Python Wrapper: Fortran and C++

cython: example LAMMPS python wrapper f2py: simple example. suggest warpper be built for GULP.

4 Importance of Open-Source Code

video of python development tree

google+ history with gource

python history with gource

grand-daddy of them all, linux kernel history with gource

It is important that results become more open-source. It's important that our communication becomes open-source. It's important that the entire numerical process be carried out open-source.

4.1 Redundancy

There are 3 different codes (shiomi/esfarjani, broido, ankit) for performing the same basic calculations.

For LD, there is GULP, phonopy, SIESTA, etc.

4.2 Code Development

took ankit 10 months to re-develop esfarjani code.

code development time can be drastically reduced using pre-existing code. codes written in a modular fashion can be added to easily.

4.3 Experiment Pre-dating Simulation

The ideal goal of simulation is to pre-date experiment. This has not been achieved yet. See Fig.

findings for: CNT/Graphene Si Thermoeletric (LUC, alloys, SL, etc) Perovskites PCM

4.4 Experiment Pre-dating Simulation

ntpl most cited paper 1-5: maradudin, ladd, dove, ziman,

4.5 Scaling Calculations

combination of increasing hardware, improving software. The improvement of these aspects depends partly on current research.

Have most control over what calculations

Time-scales for data points: 1 sec, 1 min, 1 hour, 1 day, 1 week, 1 month, 1 year number of calculations per PhD: 1E8 , 1E6 , 1E4, 1E3 , 1E2 , 1E1 , 1

1) Run system sizes by doubling. Most calcualtions scale poorly (i.e., (SYS$_SIZE^alphawithalpha > 1).Pickamaximumsystemsizebasedoncomputatinalresourcesandtimeconstraints.Runallsystemsizeshal$

2) Run all trivial (¡ 5 mins) calculations on a given system using all methods possible.

Explore all trivial calculations first. Increase the time-scale and decrease the numner of data-point accordingly.