Applications and Libraries for Physics Simulations

Emanuel Gull
University of Michigan, Ann Arbor
...for the ALPS collaboration...

http://alps.comp-phys.org

Mardi Gras Conference 2014
Application Development for Exascale Computing
February 27 - March 1, 2014
No physics/results in this talk!

1. Introduction/Motivation
2. The ALPS software package
3. Problems we are trying to address
4. Future plans?
   1. How to go towards exascale?
   2. Problems and limitations?
The beginnings

Classical physics: Ising model

\[ H(\sigma) = - \sum_{\langle i \, j \rangle} J_{ij} \sigma_i \sigma_j - \mu \sum_j h_j \sigma_j \]

Stat Mech Partition function

\[ Z_\beta = \sum_\sigma e^{-\beta H(\sigma)} \]

… compute derivatives of it to get interesting physics: order parameter, phase transitions, critical exponents, finite size scaling, universality,…
The beginnings

Beautiful and relevant physics:

- If the two spins are antiparallel, we never connect them; then we put all connected sites into the stack $Q$, and pull out the last element in the list as our next examining site.

6. Repeat steps 2-5 until the stack $Q$ is empty.

3. Simulation results

3.1. Magnetization

Figure 2 shows the measurements of magnetization in different sized systems. We can clearly see how different finite size systems behave differently near the critical region. Also we can see the fluctuation gets bigger after the critical temperature. This is because the slowing down of the Wolff algorithm beyond $T_c$. Since for disordered phase, the average cluster size is very small, the Wolff algorithm in this scenario is nothing but almost equivalent to the single flip algorithm.

3.2. Energy

Figure 3 shows the measurements of energy in different sized systems. The scaling effect is not obvious in this case because we have basically small systems. We can predict that the same thing would happen for specific heat and we hardly could an accurate result for critical exponent $\chi$.

3.3. Magnetic susceptibility

Figure 4 shows the measurements of scaled magnetic susceptibility in different sized systems to better demonstrate finite size effect. Similar to magnetization, we can easily see the curves behave differently near the critical region.

3.4. Specific heat

Figure 5 shows the measurements of scaled specific heat in different sized systems to better demonstrate finite size effect. As predicted, it does not show the scaling behavior very well.

4. Finite Size Scaling

Finite size scaling is a method to find the values for the critical exponents and the transition temperature by observing how measured quantities vary for different lattice sizes. Here are some critical exponents we intend to obtain:

First year graduate student homework: Qiaoyuan Dong
Binder cumulant, critical exponents

Figure 7: Binder cumulants plot in temperature range $T = 4.40 \ll 4.60$ with discretization $= 0.001$.

Figure 8: Binder cumulants plot in temperature range $T = 4.48 \ll 4.52$ with discretization $= 0.0001$.

Figure 9: Plot of $\ln U$ versus $\ln L$. The slope of this straight line gives us $1/\nu$.

Figure 10: Plot of $\ln h | m | i$ versus $\ln L$. The slope of this straight line gives us $1/\nu$.

Figure 11: Plot of $\ln g$ versus $\ln L$. The slope of this straight line gives us $\nu/\nu$.

Figure 12: Plot of $\ln c V$ versus $\ln L$. The slope of this straight line gives us $1/\nu$.

First year graduate student homework: Qiaoyuan Dong
...but how to compute it?

1. Use Monte Carlo

2. Metropolis single spin flip can get these things done:

3. The basic form of the algorithm is as follows:

   1. Pick a spin site using a selection probability and calculate the contribution to the energy involving this spin.
   2. Flip the value of the spin and calculate the new contribution.
   3. If the new energy is less, keep the flipped value.
   4. If the new energy is more, only keep with probability $e^{-\beta(H_{\nu} - H_{\mu})}$.
   5. Repeat.

To speed it up: Parallelize, vectorize, use OpenMP and MPI or any of the techniques of Days 1 and 2
...but how to compute it?

1. Exponential scaling thanks to Moore's law: computers are getting faster and faster

2. ...but another development is more important: exponential scaling due to different algorithms: No more single flip Metropolis but cluster updates give much larger speedup

Plot/results: D.P. Landau, UGA
cluster updates

1. Find a way of formulating the equations differently, adapt physics to problem rather than apply brute force scaling

2. Physics algorithm development!
our mission

...to make realizing this difference as easy and painless as possible

...especially for quantum systems.

Plot/results: D.P. Landau, UGA
The ALPS collaboration

University of Tokyo, Japan
• Ryo Igarashi
• Synge Todo

ETH Zürich, Switzerland
• Lukas Gamper
• Jan Güdelberger
• Tama Ma
• Matthias Troyer
• Sergei Isakov

University of Fribourg, Switzerland
• Philipp Werner

University of Michigan, Ann Arbor
• Emanuel Gull
• Andrey Antipov

Universität Göttingen, Germany
• Thomas Pruschke
• Andreas Honecker

UC Santa Barbara, USA
• Bela Bauer

University of Innsbruck, Austria
• Andreas Läuchli

LMU Munich, Germany
• Ulrich Schollwöck
• Lode Pollet

RWTH Aachen, Germany
• Stefan Wessel

Colorado School of Mines, Golden, CO, USA
• Michael Wall
• Lincoln Carr

A. Mickiewicz University, Poznan, Poland
• Grzegorz Pawłowski

Northeastern University, USA
• Adrian Feiguin

University of Bonn, Germany
• Simon Trebst
ALPS: The key players

Matthias Troyer, ETH Zurich

Synge Todo, Tokyo U

Lukas Gamper, ETH Zurich

Bela bauer, Tama Ma, Jan Gukelberger, Andreas Hehn, Michele Dolfi
ALPS applications

ALPS library

Experimentalists on lab computers running Windows to simulate experimental systems

summer school students running tutorials to gain insight into many-body or classical spin physics

Researchers running benchmarks to test new implementations and programs

HPC applications running on newest generation Cray/BlueGene/…
Example: a quantum lattice model

Given a simple lattice of sites and bonds, e.g. this 2x2 cluster…

…with 2 spaces for spin 1/2 fermions living on the ‘sites’ but able to move (‘hop’) along bonds from site to site.

…physics described by the Hubbard Hamiltonian…

\[
H = - \sum_{\langle ij \rangle, \sigma} t_{ij} (c_{i\sigma}^\dagger c_{j\sigma} + c_{j\sigma}^\dagger c_{i\sigma}) + U \sum_i n_{i\uparrow} n_{i\downarrow}.
\]
Example: a quantum lattice model: solve it!

1. Choose a basis, build the Hamiltonian matrix, diagonalize it explicitly using a diagonalization routine like Lanczos (sparse) or dsyev (full)
2. Use a determinantal quantum Monte Carlo code or a diagrammatic code to get finite temperature estimates
3. Use a wave function Monte Carlo code to get an approximation to the ground state wave function
4. Use a DMRG code or an extension to 2d to get an approximation to the ground state wave function
5. Use series expansion method to get the high-T behavior
6. Study the system using your own new method and compare to all of these algorithms?
ALPS Library: The goals

1. To provide a robust and versatile foundation on which codes for simulating quantum systems can be built.

2. To facilitate access to standard libraries, to provide a standard build system.

3. To facilitate access to large high performance computers.

4. To define standards for data formats and data interchange for simulations and to provide sample tools for working with those formats.

5. C++
ALPS Applications: goals

1. To provide a set of standard, well established applications to simulate simple quantum systems for benchmarking and testing purposes.

2. To define standards for data formats and data interchange for simulations and to provide sample tools for working with those formats.

3. To provide sample tutorials and programs for students learning numerical statistical mechanics, many-body and correlation physics (e.g. as part of a summer school).
Simulation codes of quantum lattice models

• **The status quo**
  - Individual codes, built by talented physicists over the course of years (PhD, postdoc).
  - Model-specific implementations.
  - Complex, well adapted codes for a particular problem.
  - Code implementation starts from scratch, few outside tools.
  - Knowledge preservation difficult

• **Codes built on top of the ALPS libraries**
  - Simplified code development, Minimize code copying, common parts are delegated to ALPS
  - Standard parts are independently maintained, tested, and documented
  - Reduce ‘time to market’ of complex codes
  - common file formats
  - common libraries
The tiers of ALPS

1. **Standard data formats and interfaces** to facilitate
   - exchange, archiving and querying of simulation results
   - exchange of simulation and analysis tools

2. **Libraries**
   - to support standard data formats and interfaces
   - to ease building of parallel simulation programs

3. **Evaluation tools**
   - to ease data evaluation and plotting
   - to record provenance information

4. **Applications**
   - to be used also by non-experts
   - implement modern algorithms for a large class of models
Current applications

- **Classical Monte Carlo**
  - local and cluster updates for classical spin systems, M. Troyer

- **Quantum Monte Carlo**
  - stochastic series expansions (SSE), S. Isakov
  - loop code for spin systems, S. Todo
  - continuous time worm code, S. Trebst, M. Troyer
  - extended ensemble simulations, S. Wessel, N. Stoop

- **Exact diagonalization**
  - full and sparse, A. Honecker, A. Läuchli, M. Troyer

- **DMRG and variants**: A. Feiguin, L. Carr, M. Wall

- **DMFT**: E. Gull, B. Surer, P. Werner
Simulations with ALPS

Lattice

< LATTICEGRAPH name = "square lattice">
  <FINITELATTICE>
    <LATTICE dimension="2"/>
    <EXTENT dimension="1" size="L"/>
    <EXTENT dimension="2" size="L"/>
    <BOUNDARY type="periodic"/>
  </FINITELATTICE>
  <UNITCELL>
    ...
  </UNITCELL>
</LATTICEGRAPH>

Model

<Model>
  <BASIS>
    <SITEBASIS name="spin">
      <PARAMETER name="S" default="1/2"/>
      <QUANTUMNUMBER name="Sz" min="-S" max="S"/>
    </SITEBASIS>
  </BASIS>
  <HAMILTONIAN name="spin">
    <BASIS ref="spin"/>
    <SITETERM> -h*Sz </SITETERM>
    <BONDTERM> Jxy/2*(Splus(i)*Sminus(j)+Sminus(i)*Splus(j)) 
               + Jz*Sz(i)*Sz(j) </BONDTERM>
  </HAMILTONIAN>
</Model>

Parameters

PARAMETERS = LATTICE = "square lattice" 
L = 100 
MODEL = "spin" 
Jxy = 1 
Jz = 1 
h = 0 

{ T = 0.1 } 
{ T = 0.2 } 
{ T = 0.5 } 
{ T = 1.0 }

Results

Quantum Monte Carlo

Exact diagonalization

DMRG

quantum system
The ALPS lattice library

A lattice

```xml
<LATTICEGRAPH name = "square lattice">
  <FINITELATTICE>
    <LATTICE dimension="2"/>
    <EXTENT dimension="1" size="L"/>
    <EXTENT dimension="2" size="L"/>
    <BOUNDARY type="periodic"/>
  </FINITELATTICE>
  <UNITCELL>
    <VERTEX/>
    <EDGE type="1">
      <SOURCE vertex="1" offset="0 0"/>
      <TARGET vertex="1" offset="0 1"/>
    </EDGE>
    <EDGE type="2">
      <SOURCE vertex="1" offset="0 0"/>
      <TARGET vertex="1" offset="1 0"/>
    </EDGE>
  </UNITCELL>
</LATTICEGRAPH>
```
The ALPS model library

A model

\[ H_{\text{XXZ}} = \frac{J_{xz}}{2} \sum_{\langle i,j \rangle} (S_i^+ S_j^- + S_i^- S_j^+) + J_z \sum_{\langle i,j \rangle} S_i^z S_j^z - h \sum_i S_i^z \]

```xml
<BASIS>
  <SITEBASIS name="spin">
    <PARAMETER name="S" default="1/2"/>
    <QUANTUMNUMBER name="Sz" min="-S" max="S"/>
  </SITEBASIS>
</BASIS>

<OPERATOR name="Splus" matrixelement="sqrt(S*(S+1)-Sz*(Sz+1))">
  <CHANGE quantumnumber="Sz" change="1"/>
</OPERATOR>

<OPERATOR name="Sminus" matrixelement="sqrt(S*(S+1)-Sz*(Sz-1))">
  <CHANGE quantumnumber="Sz" change="-1"/>
</OPERATOR>

<OPERATOR name="Sz" matrixelement="Sz"/>

<HAMILTONIAN name="spin">
  <BASIS ref="spin"/>
  <SITETERM> -h*Sz </SITETERM>
  <BONDTERM source="i" target="j">
    Jxy/2*(Splus(i)*Sminus(j)+Sminus(i)*Splus(j))+ Jz*Sz(i)*Sz(j)
  </BONDTERM>
</HAMILTONIAN>
```
Three ways of running ALPS

- From the command line
  - Recommended for clusters and supercomputers
- Using Python
  - Recommended for small simulations and all data evaluation
- Using VisTrails
  - Recommended for small simulations and all data evaluation
  - Captures full provenance information
Command line

• Advantage: Easy to use on every system
• Disadvantage: limited evaluation tools

• First create a parameter file parm2a
• Then create the input files and run the simulation

  parameter2xml parm2a
  spinmc --Tmin 10 --write-xml parm2a.in.xml

• Finally look at the output in a web browser
Evaluations in Python

- Python is an object-oriented scripting language
- Easy to interface to C, C++, and other languages
- Easy to learn in a short time: [http://docs.python.org/release/2.7/tutorial/index.html](http://docs.python.org/release/2.7/tutorial/index.html)
- ALPS exports the main functionality to Python
  - Creating input files
  - Running simulations
  - Loading results
  - Evaluating data
  - Making plots
  - ....
Python example

#prepare the input parameters
parms = []
for t in [0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.25, 1.5, 1.75, 2.0]:
    parms.append(
        {
            'LATTICE' : "chain lattice",
            'T' : t,
            'J' : -1,
            'THERMALIZATION' : 10000,
            'SWEEPS' : 500000,
            'UPDATE' : "cluster",
            'MODEL' : "Heisenberg",
            'L' : 60
        }
    )

#write the input file and run the simulation
input_file = pyalps.writeInputFiles('parm2a', parms)
pyalps.runApplication('spinmc', input_file, Tmin=5)
#load the susceptibility and collect it as function of temperature T
data = 
pyalps.loadMeasurements(pyalps.getResultFiles(prefix='parm2a'),'Susceptibility')
susceptibility = pyalps.collectXY(data,x='T',y='Susceptibility')

#make plot
plt.figure()
pyalps.pyplot.plot(susceptibility)
plt.xlabel('Temperature $T/J$')
plt.ylabel('Susceptibility $\chi J$')
plt.ylim(0,0.22)
plt.title('Classical Heisenberg chain')
plt.show()
• ALPS can also use the Vistrails workflow system for all simulations and evaluation
Advantages of VisTrails

- Graphical user interface and workflow system
- Integrated plotting capabilities
- Complete history of the workflow is available: you can go back to any version
- Automated caching of results: when the workflow is changed, only those parts which need to run again are executed again
- “Learning from the experts”: the complete workflow is available to students
- Automatic recording of provenance information
Computational provenance

• **Oxford English Dictionary definition of *provenance***
  
i) the fact of coming from some particular source or quarter; origin, derivation.

ii) the history or pedigree of a work of art, manuscript, rare book, etc.; concretely, a record of the ultimate derivation and passage of an item through its various owners.

• **Provenance information in science is important to**
  
  • reproduce results
  • answer questions like
    • Who created this result and when?
    • What process modified some data product and when?
    • What process created the result?
    • What was the raw data used?
    • What parameters and algorithms were used?
    • Is the result reliable?
Provenance as a key to scientific discovery

• Not a new issue! Lab notebooks have been used for a long time
  • Reproduce results
  • Evidence in clearing up discrepancies

• What is new?
  • Large volumes of data
  • Ease of producing computational results
  • Complex analyses and simulations

• Writing notes is no longer an option and not enough
  • Need systematic means to capture provenance
  • Need to change the way we perform simulations

• We all know the problem of trying to reproduce what exactly we did 10 years ago....
Using provenance-enabled workflow systems

- **Our goal:**
  - clicking a figure or data item in a published paper recalls the complete workflow leading to that figure or number
  - the workflow can be inspected
  - the workflows could be included in an arXiv submission
  - the same simulation run with modified parameters

- We are starting to use of the VisTrails systems to capture provenance
  - This has been successfully demonstrated for computer graphics papers

- **We should be able to do the same in physics!**
  - VisTrails support integrated with ALPS
Reproducible papers are possible

The arXiv version has all data and workflow!

Clicking on the figure downloads the workflow that reproduces the figure.
ALPS: Current Challenges

1. Heterogeneous user community, varied levels of expertise
2. Legacy codes
3. Difficulty to maintain codes on supercomputers and Windows
ALPS: Current Challenges

1. Several obsolete parts (library by now almost 20 years old).

2. Beautifully written but undocumented core libraries.

3. Maintenance done by a few core people, most not focused on libraries but on science.

4. Large barrier to get started with the library.
Dear all,
I am trying to install ALPS on a Ubuntu 12.04 (64bit) machine. I have downloaded all the needed libraries, and then I have downloaded and extracted on my desktop alps-2.2.b1-r7195-src-with-boost.
I then move into the folder (which contains the subfolders alps and boost) and give the command

```
sudo cmake -D Boost_ROOT_DIR:PATH=boost/ alps/
```

I get some errors, here is the output

Dear ALPS user
I have been trying to install 'alps-2.2.b1-r7195-src-with-boost' version on my pc.
I went through the ALPS building from source page prescription, but unfortunately something goes wrong and I don’t know how to fix it. Any help will be kindly appreciated

After running "make" I faced the following error:

```

```

Hi all,
For a long time I used alps on ubuntu 12.04 and I had no problem with the installation. Right know I wanted to try installing alps on ubuntu 13.10 and I encounter an problem that I can't solve.

I have extracted alps and boost (from alps-2.1.1-r6176-src-with-boost.tar.gz) into separated folders in ~/src/.
I made an folder for building. I installed dependencies with

```
sudo apt-get install cmake-curses-gui libboost-all-dev libhdf5-serial-dev libfftw3-dev gfortran python-matplotlib python-scipy python-h5py liblapack-dev xsltproc python-dev
```

when i ran dmtf-02-hybridization by alpspython tutorial2.py, I got errors shown below.
I know that info in dgesv must be zero when dgesv is working well. Did i do something wrong? if possible, please let me know what to do for this error.

thank you!
ALPS: Future Directions

- **User Applications and Tools**
  - Geared towards general users, not HPC
  - Supported from Zurich
  - Target audience: students (summer schools), experimentalists, etc.

- **Library for HPC**
  - Supported by ETH Zurich and Michigan
  - Target audience: ‘professionals’, computational physicists.
ALPS HPC: Future Directions

• Library for HPC
• Supported by ETH Zurich and Michigan

• Target audience: ‘professionals’, computational physicists.

• Goal: provide a platform for computational scientists to facilitate the development of computer codes, in particular Monte Carlo codes.

• To reduce the time it takes from inception of an algorithm to a high performance simulation.

• To facilitate access to high performance computers.
ALPS HPC (current state)

ALPS HPC Build system

- ALPS HDF5 library
- ALPS scheduler
- ALPS Auxiliary libraries
- ALPS Accumulator
ALPS HPC: the scheduler

• Distributes Monte Carlo simulations consisting of a large number of processes.

• Efficient job startup using only collective communications

• Mixed OpenMP/MPI communications

• Full error and binning analysis of results from Monte Carlo simulations over the network.
ALPS HPC: HDF5

- Convenient library for interfacing to HDF5 binary data storage.
- Simplifies reading/writing of HDF5 files.
- Used by other parts of the library (e.g. observables io) and applications
ALPS HPC: Accumulator (for Monte Carlo)

- Core of Monte Carlo part: handling of observables
- Integrates into scheduler (communications to master)
- Integrates into hdf5 for file I/O
- Correct error evaluation and error propagation, correct evaluation of functions of observables.
ALPS HPC: auxiliary libraries

- Parameter handling
  - integrates into scheduler and hdf5

- Matrix classes with binding to blas/lapack

- …several smaller ones, more to be added as they become useful
ALPS HPC: dependencies

- External dependencies exist but only on standard libraries:
  - several parts of the boost library (www.boost.org)
    - random
    - chrono
    - ...
  - HDF5 library
  - CMake as a build system
  - MPI / OpenMP dependencies for the scheduler
  - bindings for the Matrix bindings to blas and lapack
An application: CT-QMC

Average matrix size as a function of inverse temperature

Algorithms are $O([\text{Matrix size}]^3)$

Single site dynamical mean field calculation
...speedup via algorithms...

factors of 25 in matrix size translate to factors of $25^3=15’000$ in speedup. Add caching, numerics, ...
Open positions in Ann Arbor

- Scientific programming (HPC) and postdoc positions
- For more information:
  - alps.comp-phys.org
  - egull@umich.edu
Thank You!

Emanuel Gull
for the ALPS collaboration
special thanks to Matthias Troyer and Lukas Gamper

http://alps.comp-phys.org

SIMONS FOUNDATION

DOE Early Career

ALFRED P. SLOAN FOUNDATION

COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS
UNIVERSITY OF MICHIGAN