The ALPS Project 2.0

Open Source Software for Strongly Correlated Systems

Matthias Troyer, ETH Zürich

for the ALPS collaboration

The ALPS collaboration

ETH Zürich, Switzerland
- Bela Bauer
- Lukas Gamper
- Jan Gukelberger
- Tama Ma
- Brigitte Surer
- Matthias Troyer
- Philipp Werner

LMU Munich
- Ulrich Schollwöck

Universität Stuttgart, Germany
- Stefan Wessel

MPI Dresden, Germany
- Andreas Läuchli

Universität Göttingen, Germany
- Sebastian Fuchs
- Andreas Honecker

University of Wyoming, USA
- Adrian Feiguin

Harvard University, USA
- Lode Pollet

UC Santa Barbara, USA
- Simon Trebst

Columbia University, USA
- Emanuel Gull

University of Tokyo, Japan
- Ryo Igarashi
- Synge Todo

Université de Toulouse, France
- Fabien Alet

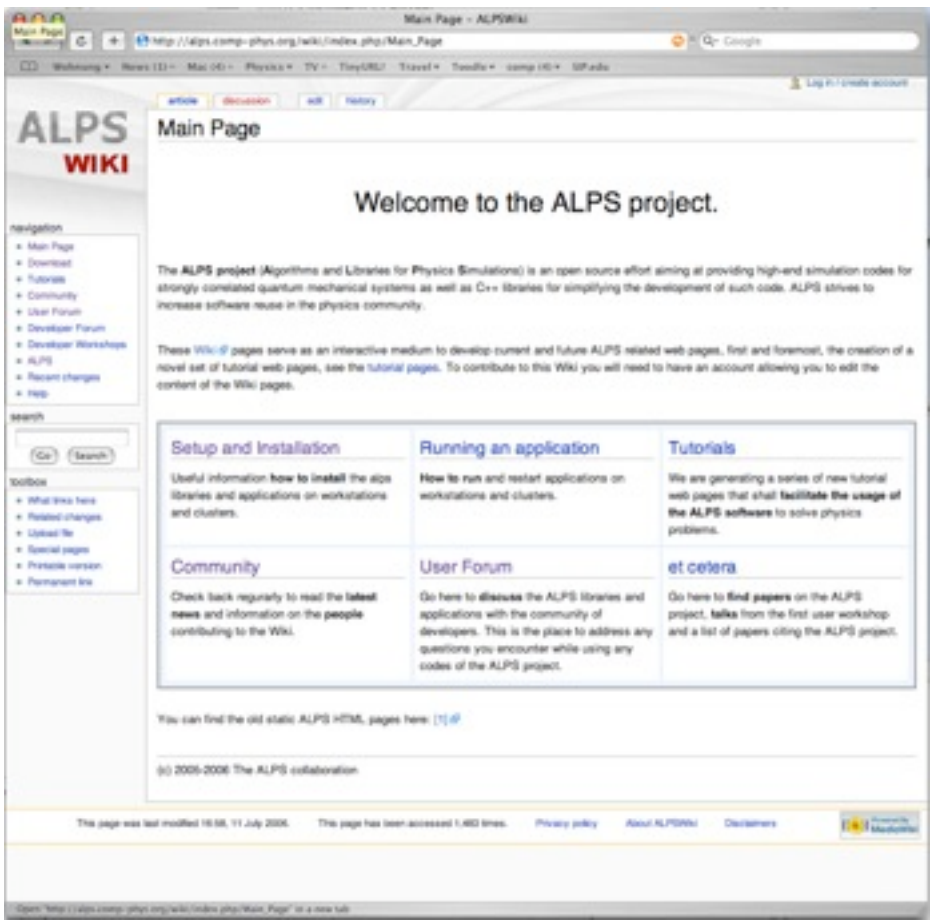
A. Mickiewicz University, Poznan, Poland
- Grzegorz Pawłowski

The ALPS project

Algorithms and **L**ibraries for **P**hysics **S**imulations

- **open source** data formats, libraries and simulation codes for quantum lattice models

- download codes from website **http://alps.comp-phys.org**



Simulation codes of quantum lattice models

- **The status quo**
  - individual codes
  - model-specific implementations
  - growing complexity of methods
- **ALPS**
  - community codes
  - generic implementations
  - simplified code development
  - common file formats

Key Technologies

Generic Programming in C++
- flexibility
- high-performance

Python for evaluation
- flexibility, object orientation
- custom C++ extensions

XML / XSLT and HDF5 for I/O
- portability
- self-explanatory

MPI/OpenMP for Parallelization

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

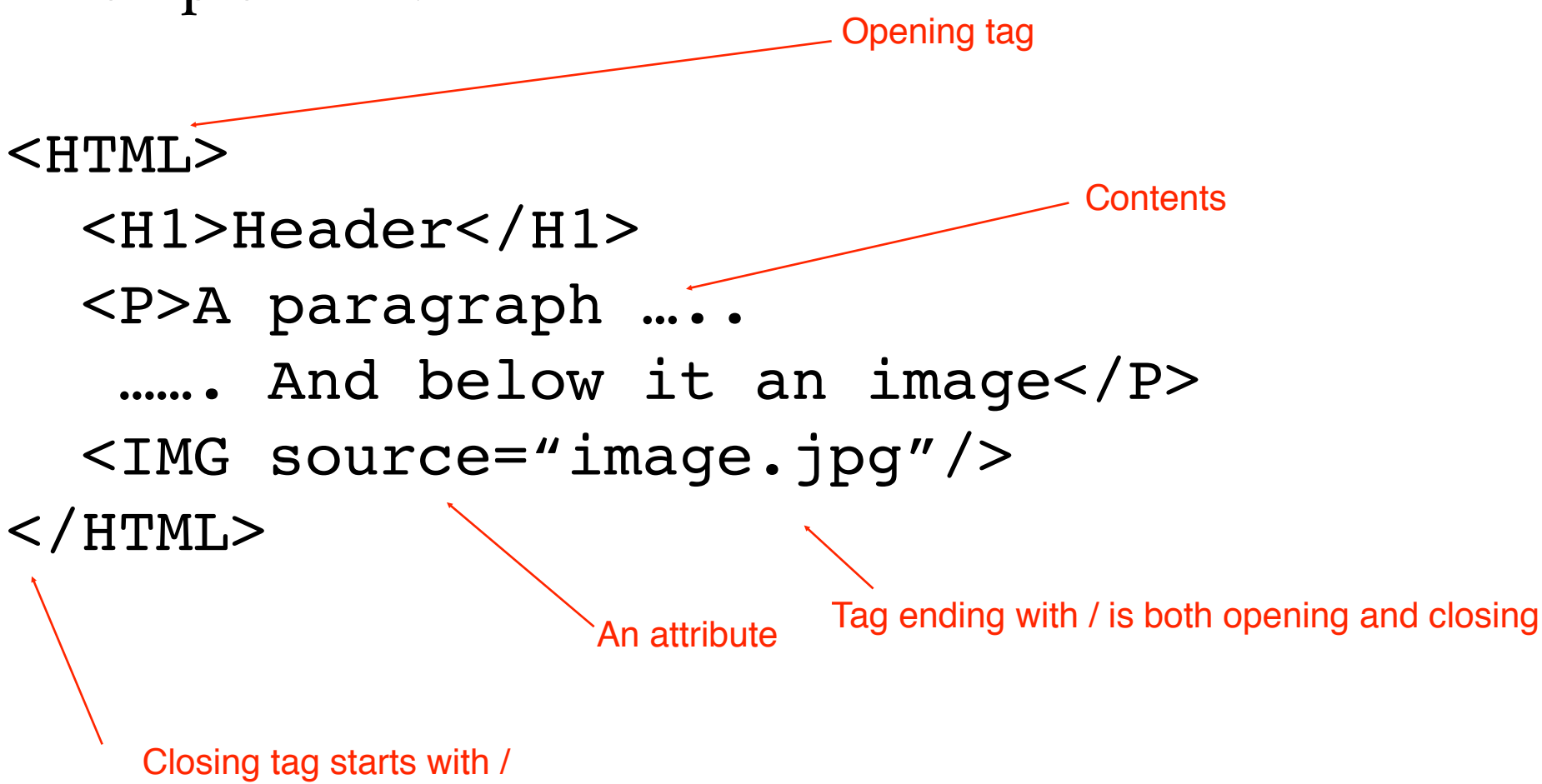
The ALPS project

Algorithms and Libraries for Physics Simulations

- The **simulation codes** include
  - Classical and Quantum Monte Carlo
  - Exact and Full Diagonalization
  - Dynamical Mean Field Theory (DMFT)
  - Density Matrix Renormalization Group (DMRG)
- **Motivation**
  - established algorithms
  - increased demand for reliable simulations from theorists and experimentalists

What is XML?

- eXtensible Markup Language
- We mix text with "tags" defining the function of the text
- Example: HTML



Why use XML?

- Plain text file:

```
# first row parameters
10 0.5 10000 1000
# mean, error
-10.451 0.043
```

XML:

```
<PARAMETER name="L">10</PARAMETER>
<PARAMETER name="T">0.5</PARAMETER>
<PARAMETER name="SWEEPS">10000</PARAMETER>
<PARAMETER name="THERMALIZATION">1000</PARAMETER>
<AVERAGE name="Energy">
  <MEAN> -10.451 </MEAN>
  <ERROR> 0.043 </ERROR>
</AVERAGE>
```

- Which is easier to understand?
- Which is better machine-readable?
- Which one will you understand in a few years?

Why use XML?

- Extending the data format: let's add random number generator type and seed

- Plain text file:

```
# first row parameters
10 0.5 10000 1000 12
# random number generator
"Mersenne Twister"
# mean, error
-10.451 0.043
```

XML:

```
<PARAMETER name="L">10</PARAMETER>
<PARAMETER name="T">0.5</PARAMETER>
<PARAMETER name="SWEEPS">10000</PARAMETER>
<PARAMETER name="THERMALIZATION">1000
</PARAMETER>

<PARAMETER name="SEED">12</PARAMETER>
<RNG name="Mersenne Twister"/>
<AVERAGE name="Energy">
  <MEAN> -10.451 </MEAN>
  <ERROR> 0.043 </ERROR>
</AVERAGE>
```

- The change in the text file format might break your program
- The additional XML tag is no problem

Why use XML?

- Contents marked up with context
  - Reduces data rot
  - Increases portability of data
- Extensible
  - Can add new contents without breaking old programs
- XSLT
  - Can use “stylesheets” to display/convert contents into any other format
- ISO standard
  - Many tools available: editors, browsers, databases, ...

What is HDF-5?

- Hierarchical Data Format, version 5
- Similar hierarchical and extensible structure as XML
- But for large binary data
- Supported by many standard tools

http://www.hdfgroup.org/HDF5/

Simulations with ALPS

Lattice

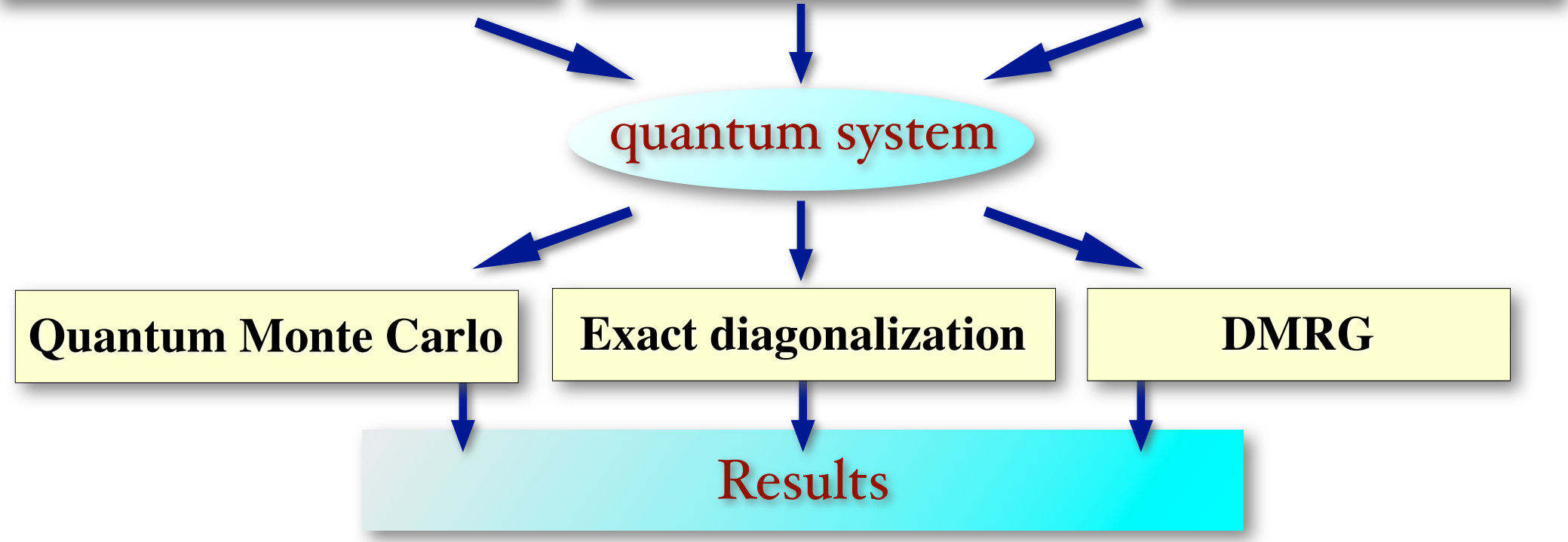
```
<LATTICEGRAPH name = "square lattice">
  <FINITELATTICE>
    <LATTICE dimension="2"/>
    <EXTENT dimension="1" size="L"/>
    <EXTENT dimension="2" size="L"/>
    <BOUNDARY type="periodic"/>
  </FINITELATTICE>
  <UNITCELL>
    ...
  </UNITCELL>
</LATTICEGRAPH>
```

Model

```
<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 source="i" target="j">
   Jxy/2*(Splus(i)*Sminus(j)+Sminus(i)*Splus(j))
   + Jz*Sz(i)*Sz(j)
  </BONDTERM>
</HAMILTONIAN>
```

Parameters

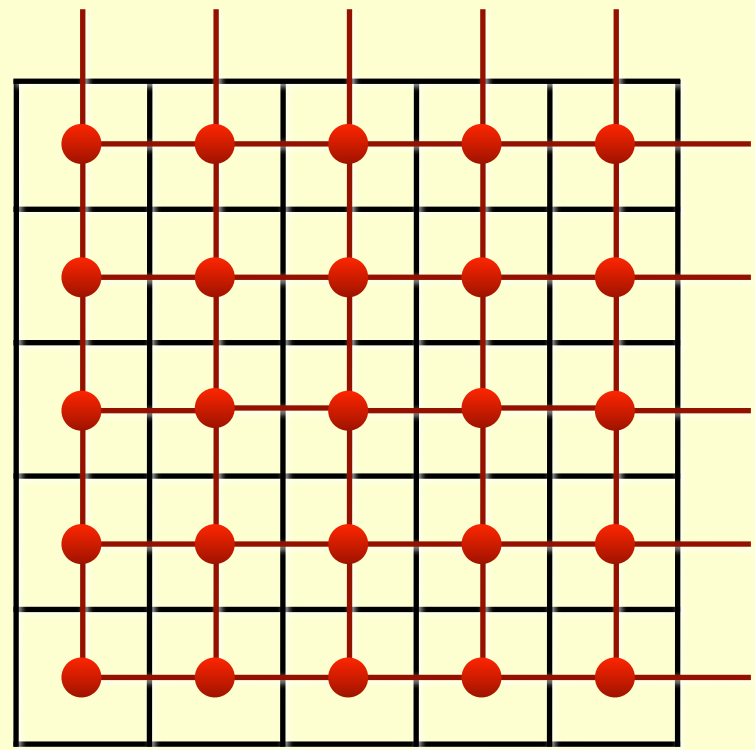
```
LATTICE = “square lattice”
L = 100

MODEL = “spin”
Jxy = 1
Jz  = 1
h   = 0

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



The ALPS lattice library

A lattice

```
<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

$$H_{XXZ} = \frac{J_{xz}}{2} \sum_{\langle i,j \rangle} (S_i^+ S_j^- + S_i^- S_j^+) + J_z \sum_{\langle i,j \rangle} S_i^z S_j^z - h \sum_i S_1^z$$

```
<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>
```

Current applications

- **Classical Monte Carlo**
  - local and cluster updates for classical spin systems, M. Troyer
- **Quantum Monte Carlo**
  - stochastic series expansions (SSE), F. Alet, L. Pollet, M. Troyer
  - 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:** A. Feiguin
- **DMFT:** E. Gull, B. Surer, P. Werner

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/2.6.5/tutorial/introduction.html
- ALPS exports the main functionality to Python
  - Creating input files
  - Running simulations
  - Loading results
  - Evaluating data
  - Making plots
  - ....

Python example

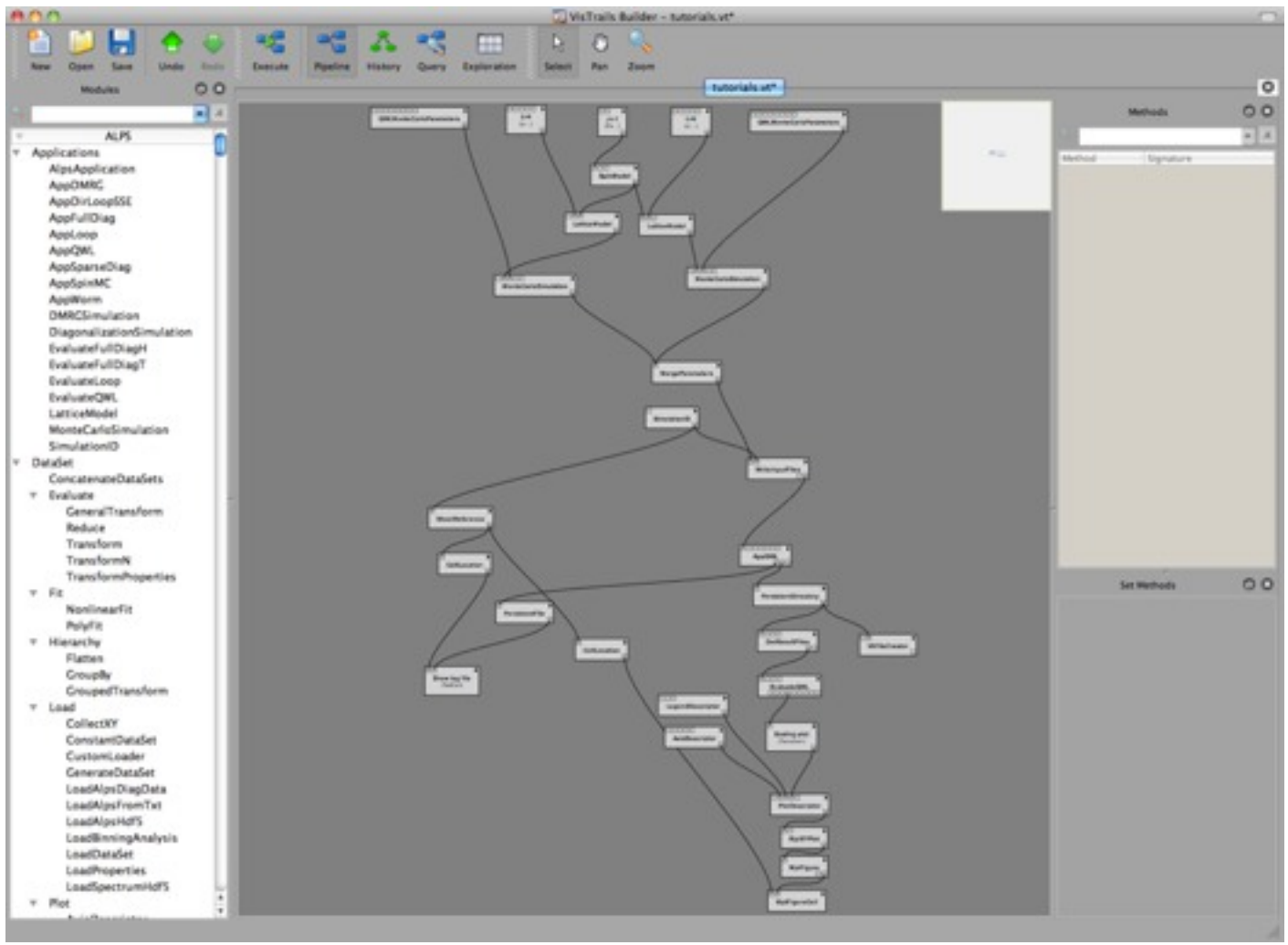
```
#prepare the input parameters
parms = []
for t in [0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.25,
1.5, 1.75, 2.0]:
    parms.append(
        {
          'LATTICE'        : "chain lattice",
          'T'              : t,
          'J'              : -1 ,
          'THERMALIZATION'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

Python example (cont.)

```
#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()
```

Graphical workflow systems

- 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 yo 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

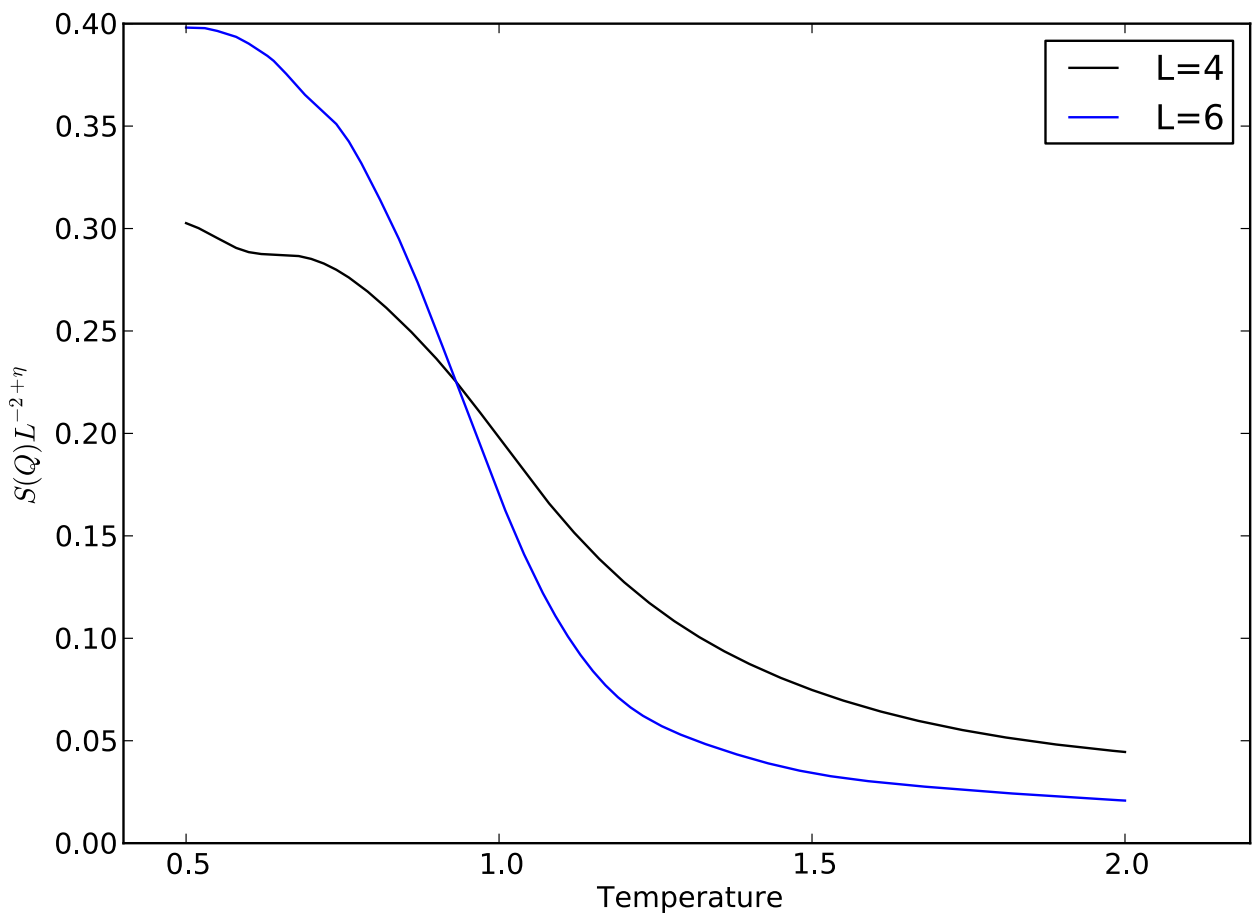
- **Our goal:**
  - clicking a figure or data item in a published paper recalls the complete workflow leading to that figure or number
  - the workflow can be inspected
  - the workflows could be included in an arXiv submission
  - the same simulation run with modified parameters

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

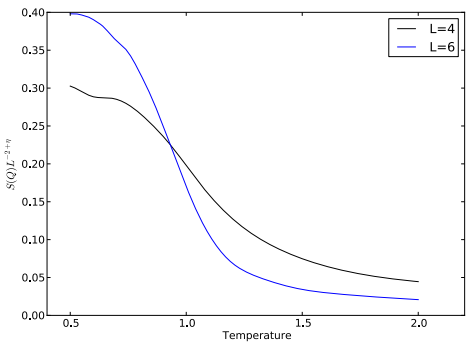
- **We should be able to do the same in physics !**
  - VisTrails support integrated with ALPS

Workflow provenance using VisTrails

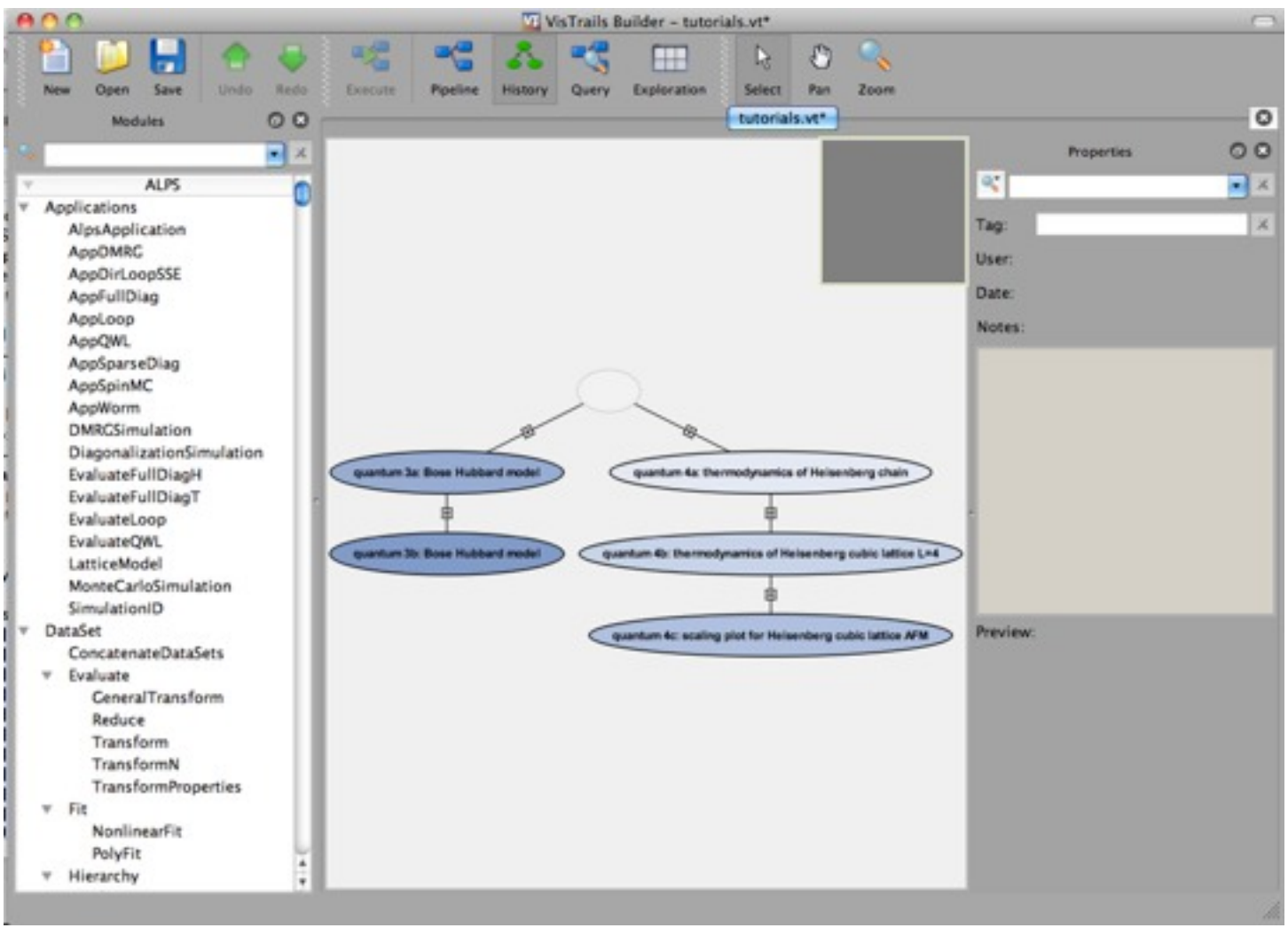
A figure from a paper (or the tutorials)



Workflow provenance using VisTrails

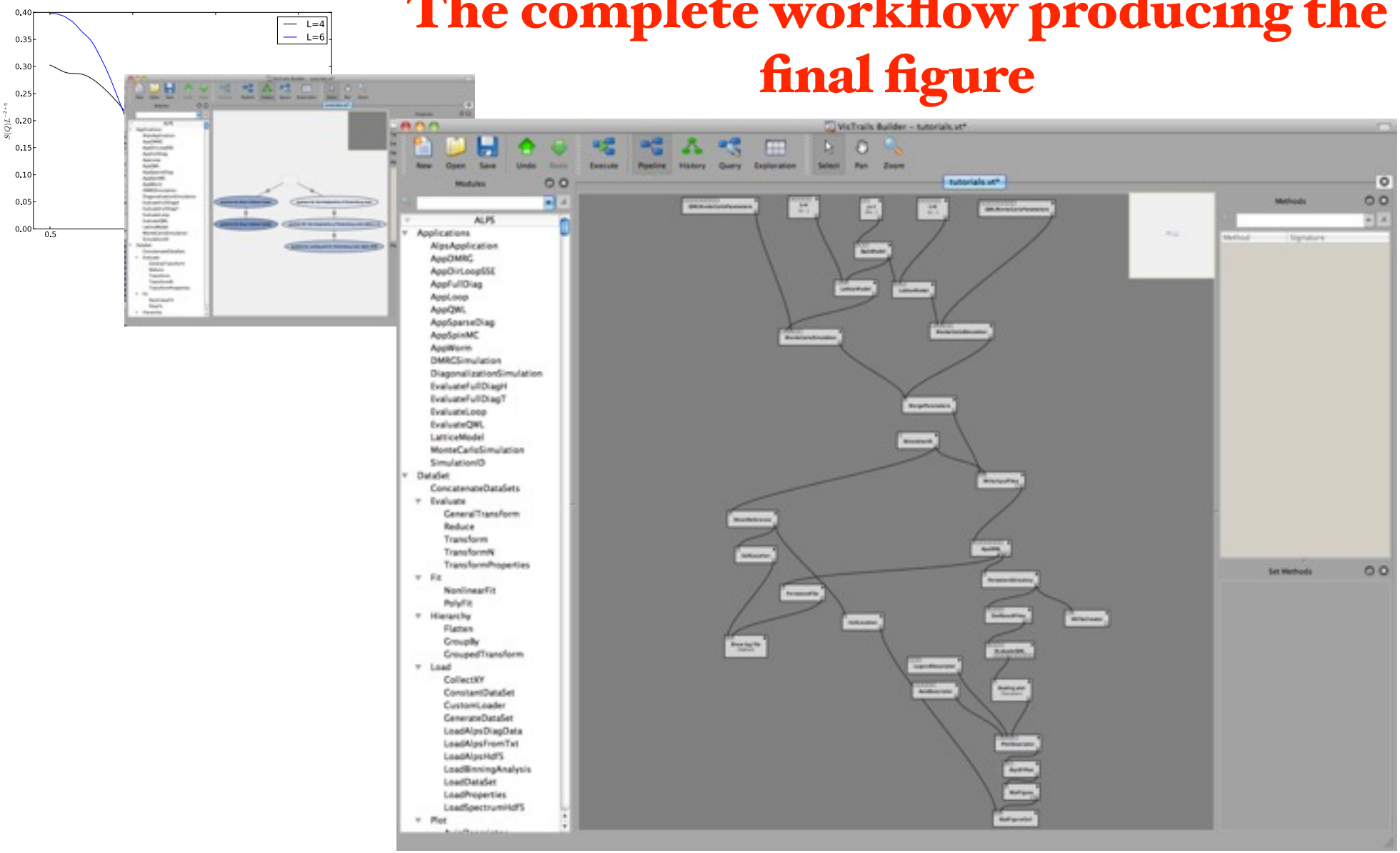


The history of explorations leading to the final figure

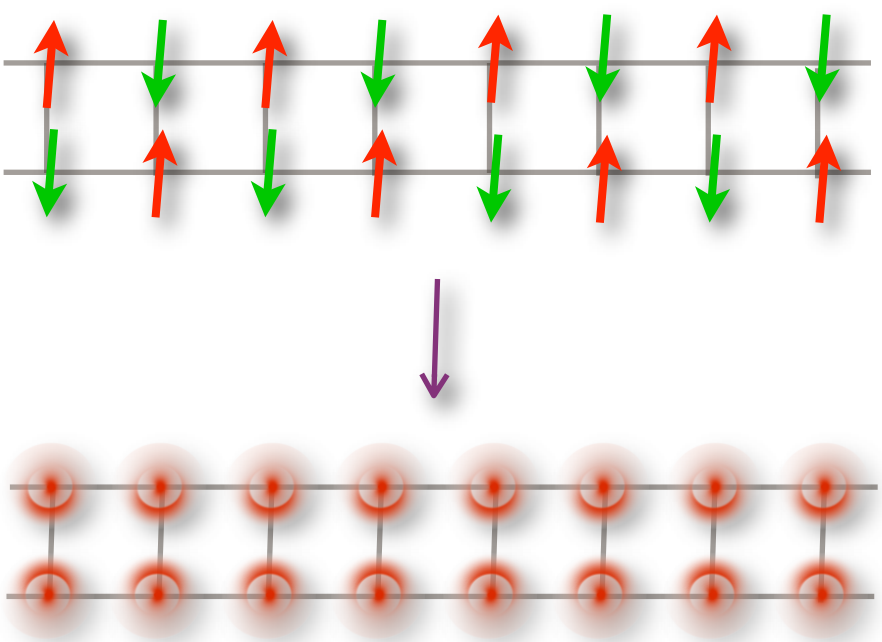


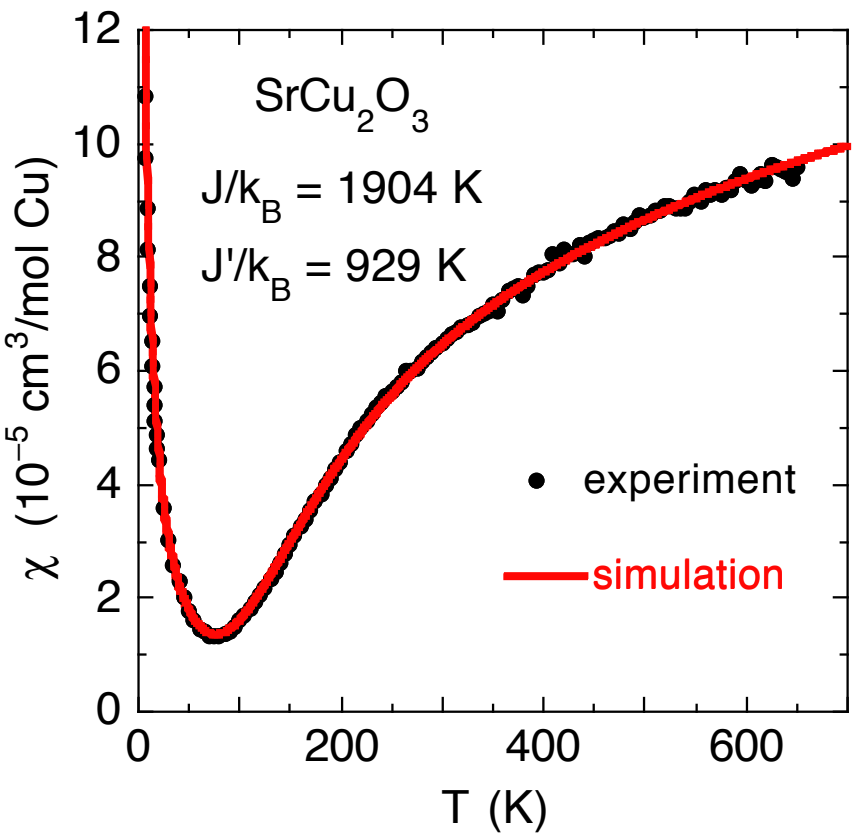
Workflow provenance using VisTrails

The complete workflow producing the final figure



Quantum spin ladders



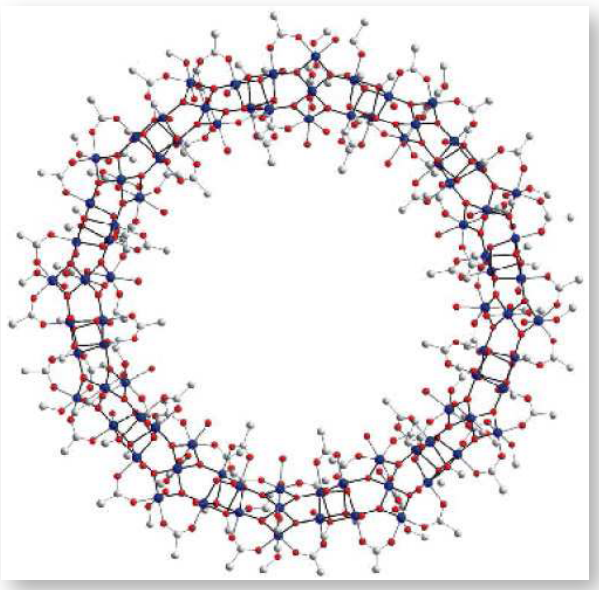


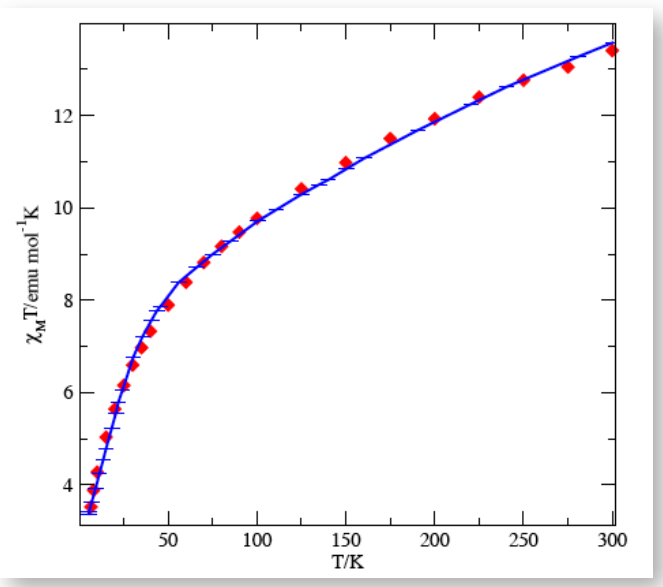
➡ compare microscopic models to experiments

Mn-84 molecules

Vassilis Tangoulis, in preparation

- How can we microscopically model interactions in Mn-84?





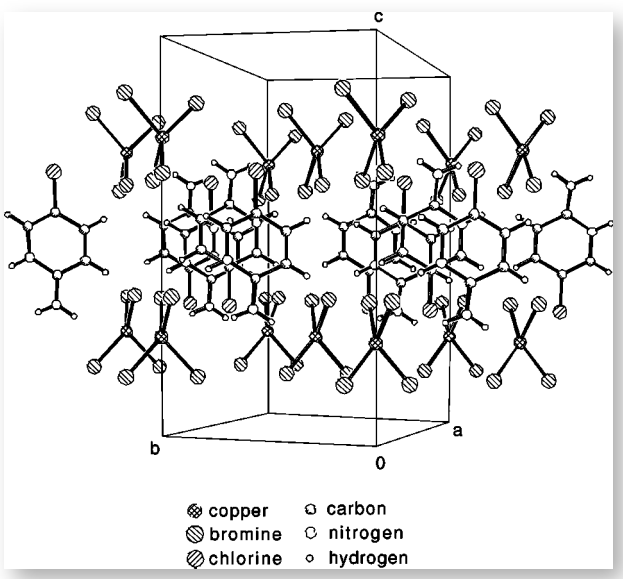
ALPS MC codes

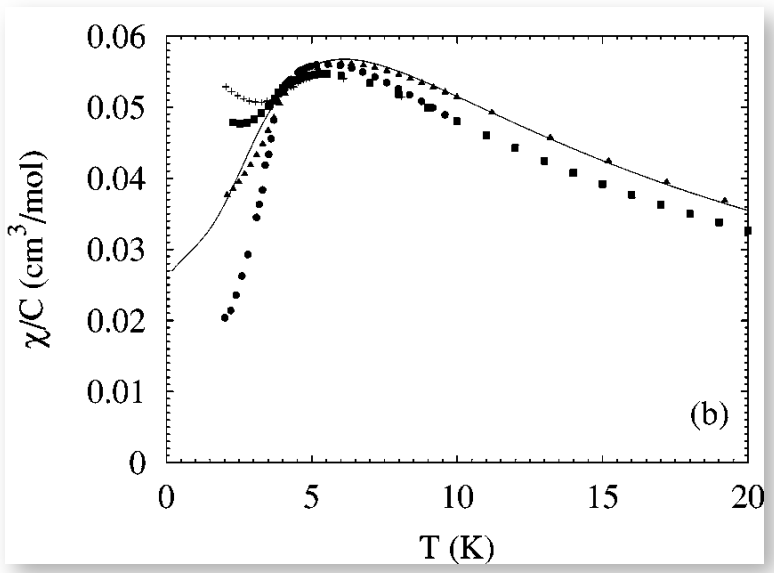
Numerical evaluation of susceptibility for full molecule:
Fit of magnetic interaction strength.

Low-dimensional quantum magnets

C.P. Landee et al., Phys. Rev. B **65**, 144412 (2002)

- How to characterize newly synthesized materials?

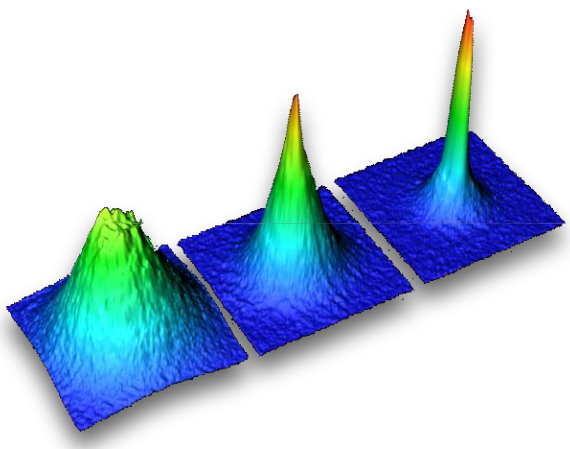




ALPS QMC codes

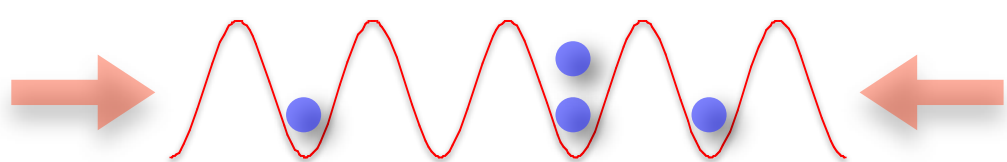
Numerical evaluation of susceptibility for 2D QHAF:
Fit of magnetic interaction strength.

BEC in ultracold atomic gases



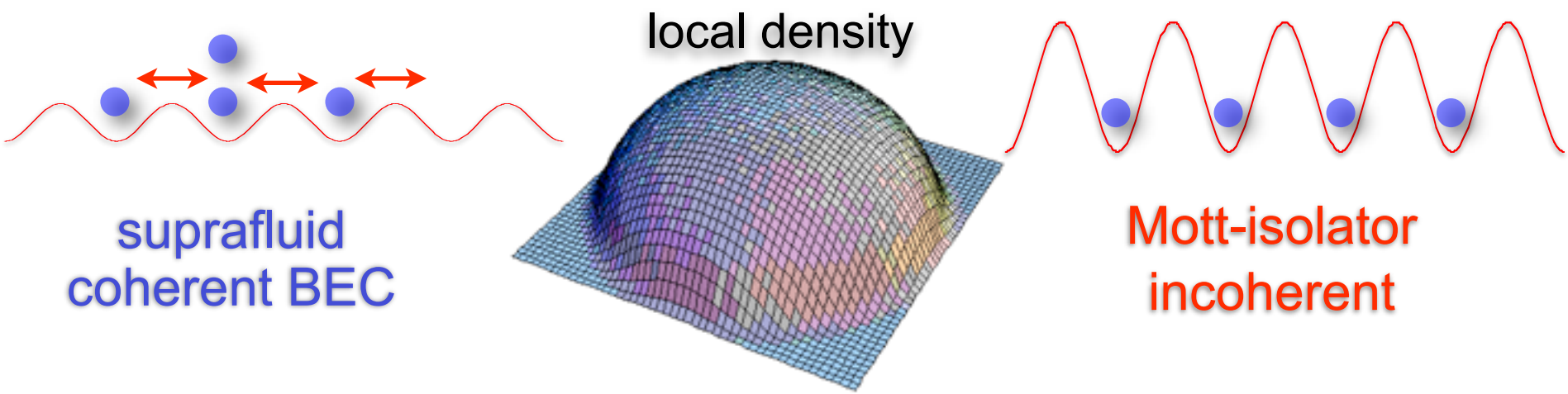
T. Esslinger, ETH Zürich

- Ultracold ^{87}Rb atoms form a Bose-Einstein condensate (BEC).
  - first observed in 1995
  - sympathetic cooling of fermionic ^{40}K atoms (2004)
- Standing laser waves form an optical lattice.



Realization of the Bose-Hubbard model

$$H = -t \sum_{\langle ij \rangle} \left(b_i^\dagger b_j + \text{h.c.} \right) + U \sum_i n_i(n_i - 1)/2 - \mu \sum_i n_i + V \sum_i r_i^2 n_i$$



ALPS QMC codes

Numerical simulation of experimental setup:
120^3 sites and harmonic trapping potential

Now it's your turn

- Install ALPS 2.0.0b2 from the ALPS web page http://alps.comp-phys.org/

- Run the first tutorials using Python or Vistrails:
  - Tutorial MC-01 on autocorrelations
  - Tutorial MC-02 on classical MC simulations

- Homework: write your own MC code for the Ising model
  - Use the template provided in Code-01 or Code-02