

# The ALPS Project

Open Source Software for  
Strongly Correlated Systems

---

Matthias Troyer, ETH Zürich

*for the ALPS collaboration*

# The ALPS collaboration

## **ETH Zürich, Switzerland**

- Philippe Corboz
- Lukas Gamper
- Emanuel Gull
- Lode Pollet
- Matthias Troyer

## **IRRMA Lausanne, Switzerland**

- Andreas Läuchli

## **RWTH Aachen, Germany**

- Ulrich Schollwöck

## **Universität Marburg, Germany**

- Reinhard Noack
- Salvatore Manmana

## **Institute of Metal Physics, Ekaterinburg, Russia**

- Anton Kozhevnikov

## **A. Mickiewicz University, Poznan, Poland**

- Grzegorz Pawłowski

## **University of Queensland, Australia**

- Ian McCulloch

## **University of New South Wales, Australia**

- Fabricio Albuquerque

## **Universität Stuttgart, Germany**

- Stefan Wessel

## **Université de Toulouse, France**

- Fabien Alet

## **TU Graz, Austria**

- Franz Michel

## **UC Santa Barbara, USA**

- Adrian Feiguin
- Simon Trebst

## **UC Davis, USA**

- Munehisa Matsumoto

## **Columbia University, USA**

- Philipp Werner

## **Honk Kong University, China**

- Siegfried Gürtler

## **University of Tokyo, Japan**

- Ryo Igarashi
- Syngae Todo

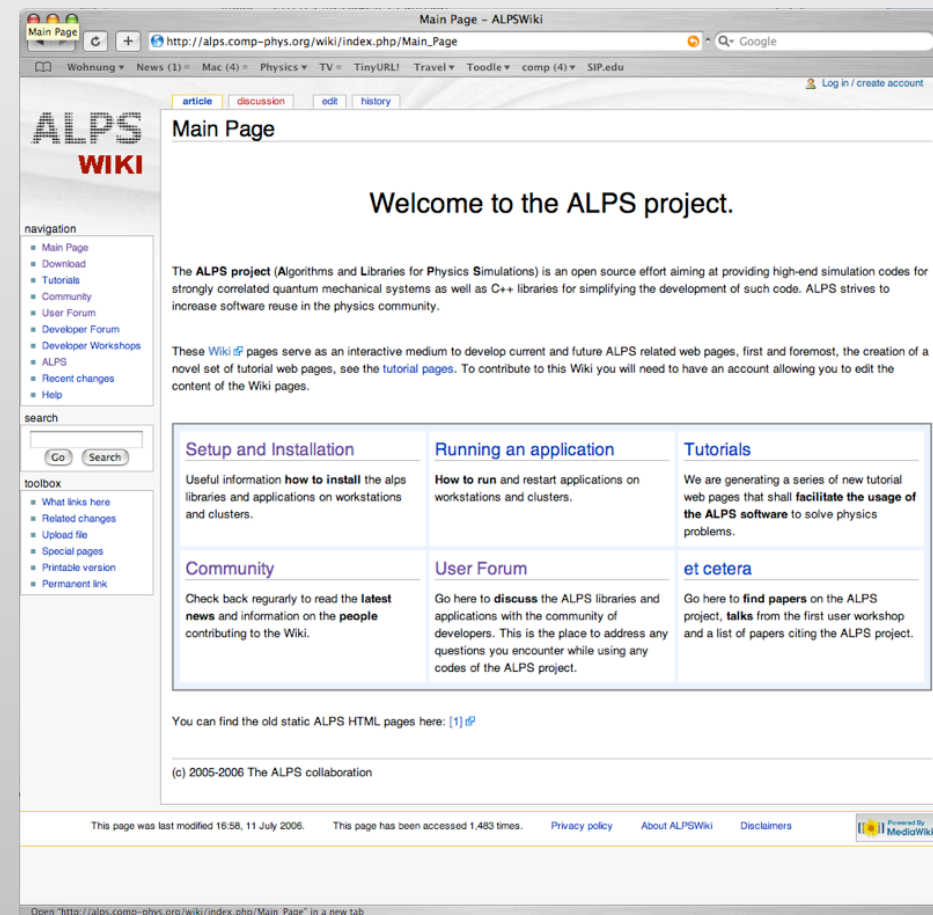
## **Universität Göttingen, Germany**

- Sebastian Fuchs
- Andreas Honecker
- Thomas Pruschke

# The ALPS project

## Algorithms and Libraries for Physics Simulations

- **open source** data formats, libraries and simulation codes for quantum lattice models
- download codes from website **<http://alps.comp-phys.org>**



The screenshot shows the main page of the ALPS Wiki. The page title is "Main Page - ALPSWiki" and the URL is "http://alps.comp-phys.org/wiki/index.php/Main\_Page". The page features a navigation menu on the left with links to Main Page, Download, Tutorials, Community, User Forum, Developer Forum, Developer Workshops, ALPS, Recent changes, and Help. Below the navigation menu is a search box and a toolbox with links to What links here, Related changes, Upload file, Special pages, Printable version, and Permanent link. The main content area is titled "Main Page" and contains a welcome message: "Welcome to the ALPS project." Below the welcome message is a paragraph describing the ALPS project as an open source effort aiming at providing high-end simulation codes for strongly correlated quantum mechanical systems. The page also includes a table with three columns: "Setup and Installation", "Running an application", and "Tutorials". The "Setup and Installation" column contains the text: "Useful information **how to install** the alps libraries and applications on workstations and clusters." The "Running an application" column contains the text: "How to run and restart applications on workstations and clusters." The "Tutorials" column contains the text: "We are generating a series of new tutorial web pages that shall **facilitate the usage of the ALPS software** to solve physics problems." Below the table is a section titled "Community" with the text: "Check back regularly to read the **latest news** and information on the **people** contributing to the Wiki." To the right of the "Community" section is a section titled "User Forum" with the text: "Go here to **discuss** the ALPS libraries and applications with the community of developers. This is the place to address any questions you encounter while using any codes of the ALPS project." To the right of the "User Forum" section is a section titled "et cetera" with the text: "Go here to **find papers** on the ALPS project, **talks** from the first user workshop and a list of papers citing the ALPS project." At the bottom of the page, there is a footer with the text: "You can find the old static ALPS HTML pages here: [1]". The footer also includes the copyright notice: "(c) 2005-2006 The ALPS collaboration" and a footer bar with the text: "This page was last modified 16:58, 11 July 2006. This page has been accessed 1,483 times. Privacy policy About ALPSWiki Disclaimers" and a logo for "Powered by MediaWiki".

# Simulation codes of quantum lattice models

# Simulation codes of quantum lattice models

- **The status quo**
  - individual codes
  - model-specific implementations
  - growing complexity of methods

# 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

# 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

### Standard C++ Libraries

- fast development

### XML / XSLT for Input/Output

- portability
- self-explanatory

### MPI/OpenMP for Parallelization

# Three tiers of ALPS



# Three tiers of ALPS

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

# Three 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

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

# The ALPS project

**A**lgorithms and **L**ibraries for **P**hysics **S**imulations

# The ALPS project

Algorithms and Libraries for Physics Simulations

- The **simulation codes** include
  - Classical and Quantum Monte Carlo
  - Exact and Full Diagonalization
  - Density Matrix Renormalization Group (DMRG)

# The ALPS project

Algorithms and Libraries for Physics Simulations

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

# What is XML?

- e**X**tensible **M**arkup **L**anguage
- We mix text with “tags” defining the function of the text
- Example: HTML

```
<HTML>
```

```
  <H1>Header</H1>
```

```
  <P>A paragraph ....
```

```
  ..... And below it an image</P>
```

```
  <IMG source="image.jpg" />
```

```
</HTML>
```

# What is XML?

- e**X**tensible **M**arkup **L**anguage
- We mix text with “tags” defining the function of the text
- Example: HTML

Opening tag

<HTML>

<H1>Header</H1>

<P>A paragraph ....

..... And below it an image</P>

<IMG source="image.jpg" />

</HTML>



# What is XML?

- e**X**tensible **M**arkup **L**anguage
- We mix text with “tags” defining the function of the text
- Example: HTML

```
<HTML>  
  <H1>Header</H1>  
  <P>A paragraph ....  
  ..... And below it an image</P>  
  <IMG source="image.jpg" />  
</HTML>
```

Opening tag



Closing tag starts with /



# What is XML?

- e**X**tensible **M**arkup **L**anguage
- We mix text with “tags” defining the function of the text
- Example: HTML

```
<HTML>  
  <H1>Header</H1>  
  <P>A paragraph ....  
  ..... And below it an image</P>  
  <IMG source="image.jpg" />  
</HTML>
```

Opening tag

Tag ending with / is both opening and closing

Closing tag starts with /

# What is XML?

- e**X**tensible **M**arkup **L**anguage
- We mix text with “tags” defining the function of the text
- Example: HTML

```
<HTML>
  <H1>Header</H1>
  <P>A paragraph ....
  ..... And below it an image</P>
  <IMG source="image.jpg" />
</HTML>
```

Opening tag

Contents

Tag ending with / is both opening and closing

Closing tag starts with /

# What is XML?

- e**X**tensible **M**arkup **L**anguage
- We mix text with “tags” defining the function of the text
- Example: HTML

```
<HTML>  
  <H1>Header</H1>  
  <P>A paragraph ....  
  ..... And below it an image</P>  
  <IMG source="image.jpg" />  
</HTML>
```

Opening tag

Contents

An attribute

Tag ending with / is both opening and closing

Closing tag starts with /

# 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: XML:

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

<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

# Calculating $\pi$

- We calculated  $\pi$

```
<RNG name="RanF" />
```

```
<AVERAGE name="Pi">
```

```
  <MEAN> 3.1566 </MEAN>
```

```
  <ERROR> 0.0048 </ERROR>
```

```
  <COUNT> 33554432 </COUNT>
```

```
</AVERAGE>
```

- Now that we know that RanF is a bad generator we know which data to throw away

# XSLT transformations



# XSLT transformations

- But the XML file is ugly to look at ...

# XSLT transformations

- But the XML file is ugly to look at ...

# XSLT transformations

- But the XML file is ugly to look at ...
- No problem, it is meant only for the computer's eyes

# XSLT transformations

- But the XML file is ugly to look at ...
- No problem, it is meant only for the computer's eyes

# XSLT transformations

- But the XML file is ugly to look at ...
- No problem, it is meant only for the computer's eyes
- XSLT transforms (“stylesheets”) allow conversions into any format
  - Other XML
  - HTML
  - Plain text

# XSLT transformations

- But the XML file is ugly to look at ...
- No problem, it is meant only for the computer's eyes
- XSLT transforms (“stylesheets”) allow conversions into any format
  - Other XML
  - HTML
  - Plain text
- I'll show you an example ...

# 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, ...

# Simulations with ALPS



# 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>
```

# 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>
```

# 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 }
```

# 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 }
```

quantum system

# 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 }
```

quantum system

Quantum Monte Carlo

Exact diagonalization

DMRG

# 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 }
```

quantum system

Quantum Monte Carlo

Exact diagonalization

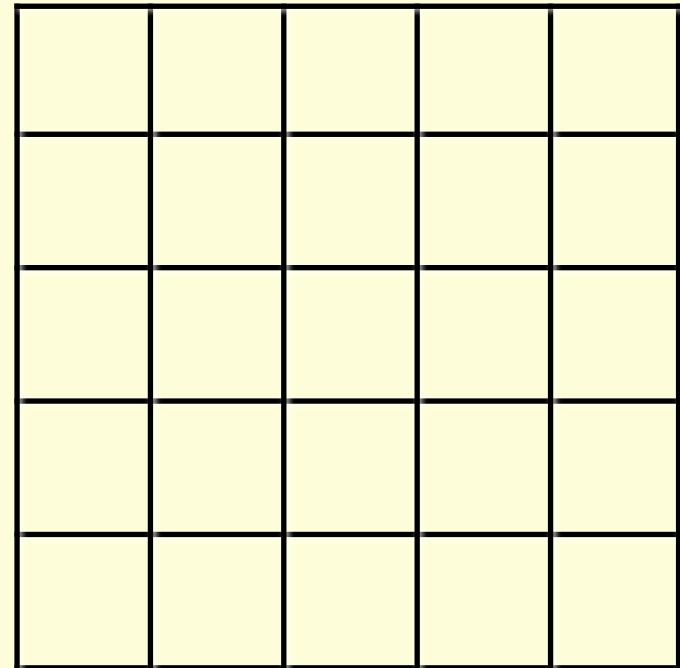
DMRG

Results

# The ALPS lattice library

## A lattice

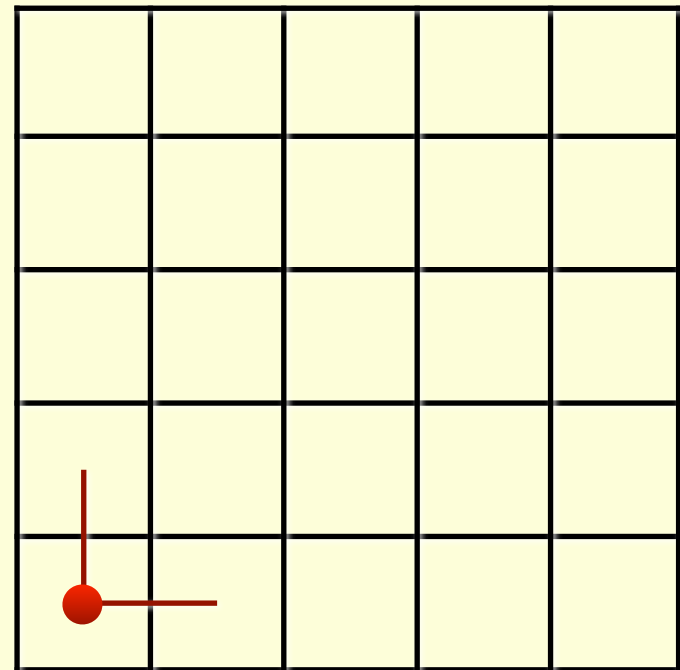
```
<LATTICEGRAPH name = "square lattice">
  <FINITELATTICE>
    <LATTICE dimension="2"/>
    <EXTENT dimension="1" size="L"/>
    <EXTENT dimension="2" size="L"/>
    <BOUNDARY type="periodic"/>
  </FINITELATTICE>
  <UNITCELL>
    <VERTEX/>
    <EDGE type="1">
      <SOURCE vertex="1" offset="0 0"/>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

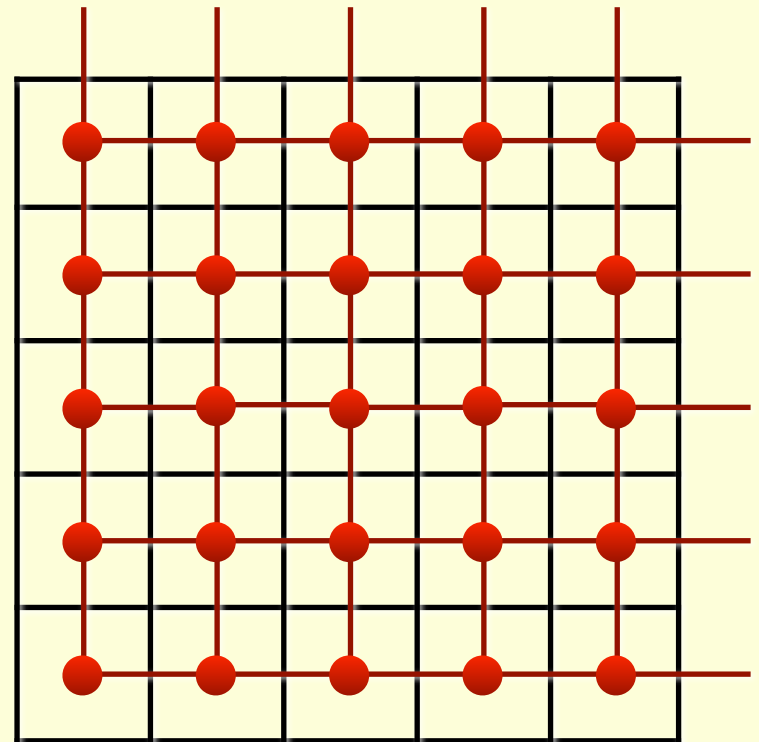




# The ALPS lattice library

## 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

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

```
<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

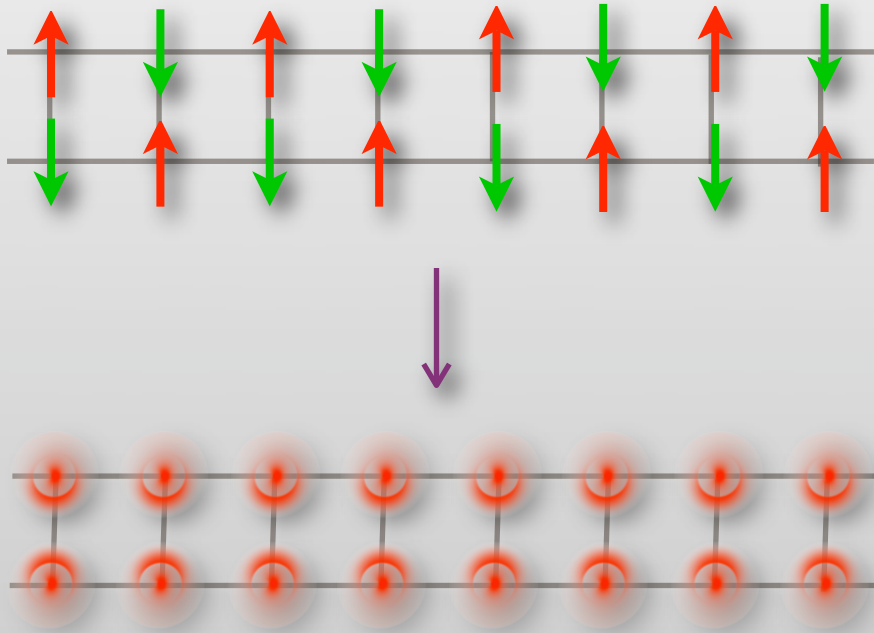
- **Exact diagonalization**

- full and sparse, A. Honecker, A. Läuchli, M. Troyer

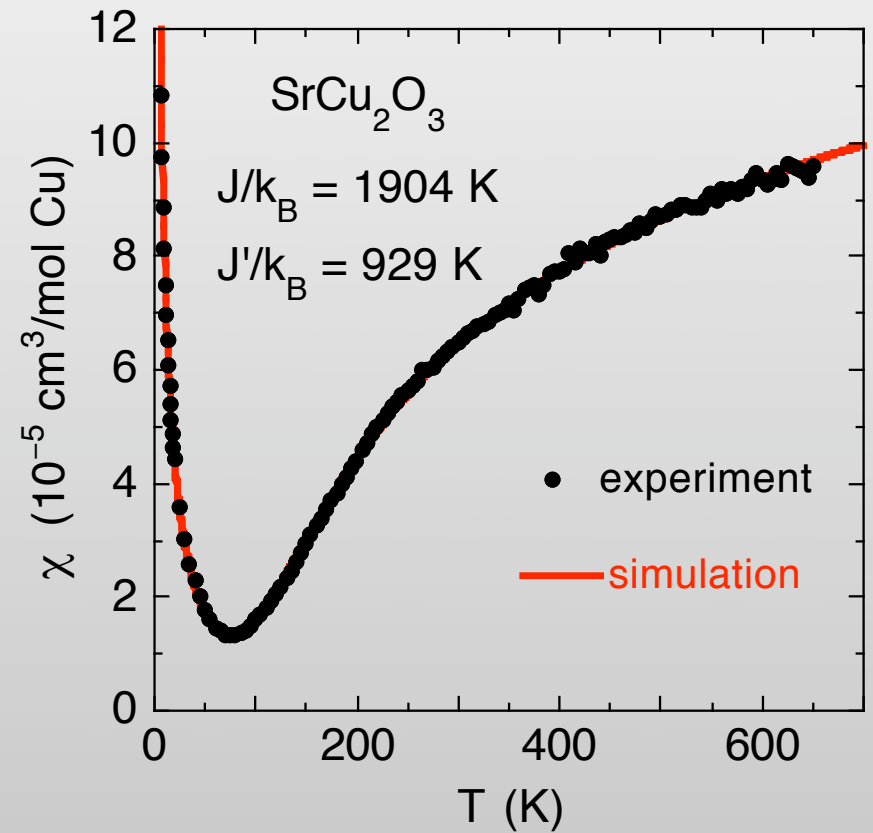
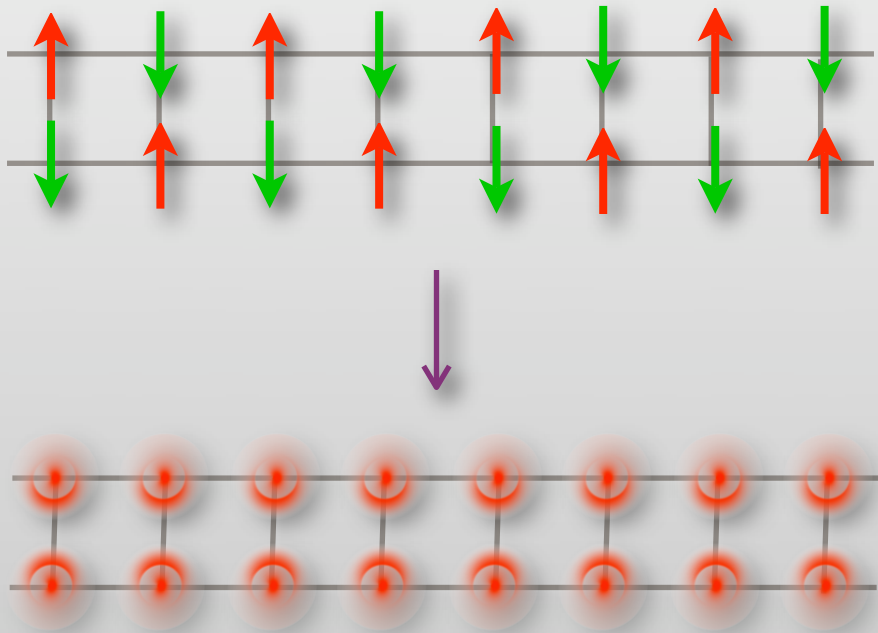
- **DMRG**

- single particle, S. Manmana, R. Noack, U. Schollwöck
- interacting particles, A. Feiguin

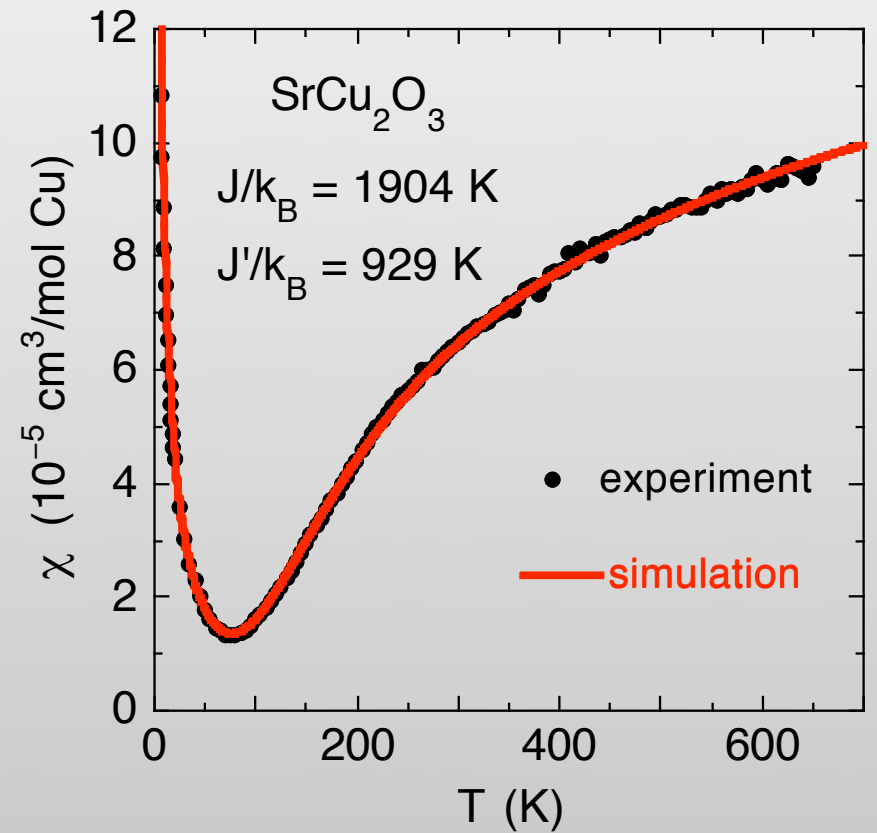
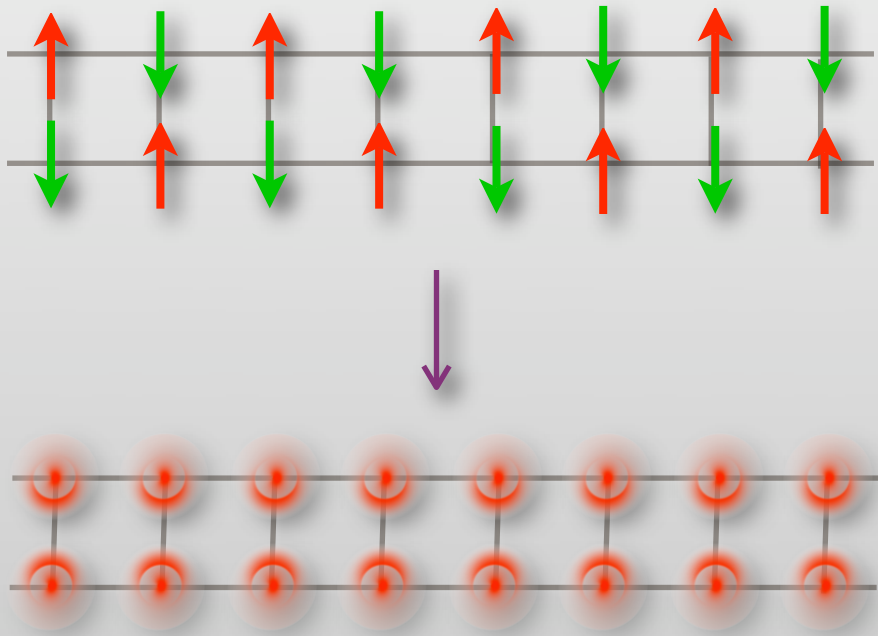
# Quantum spin ladders



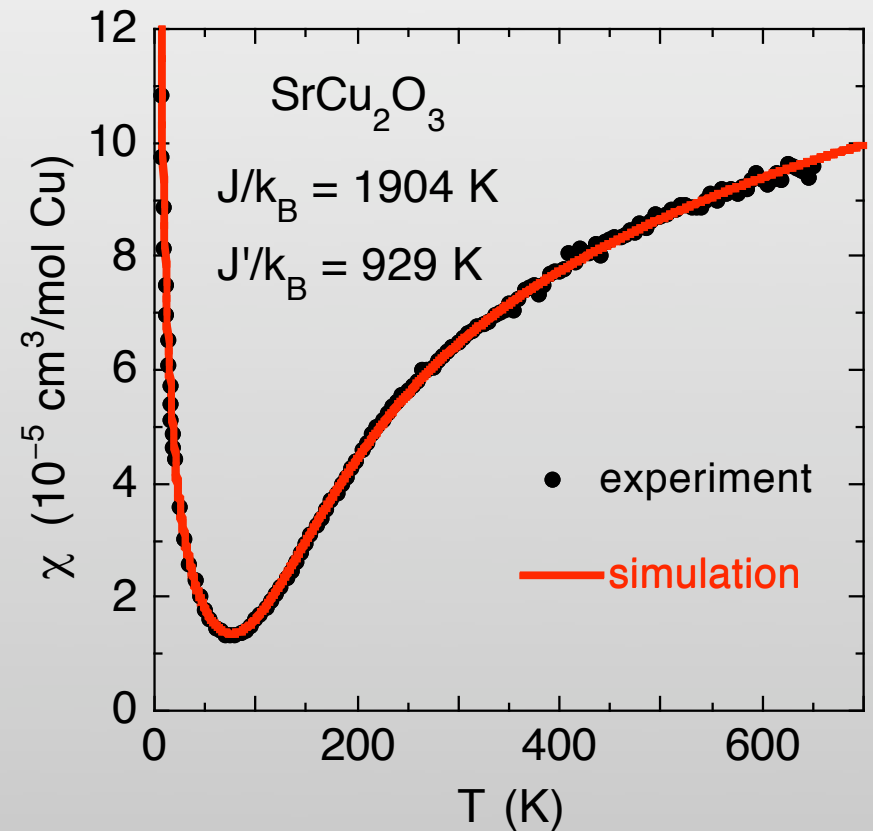
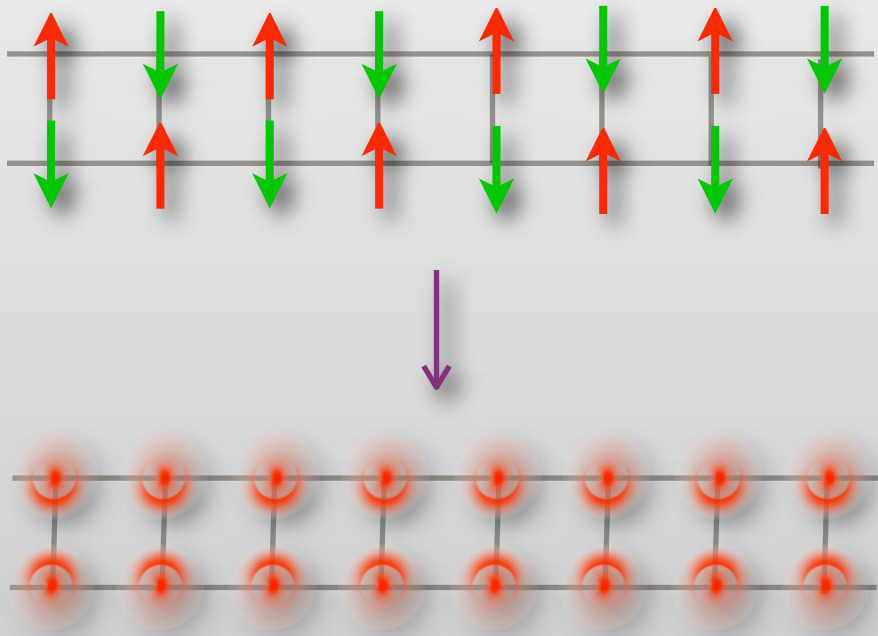
# Quantum spin ladders



# Quantum spin ladders



# Quantum spin ladders

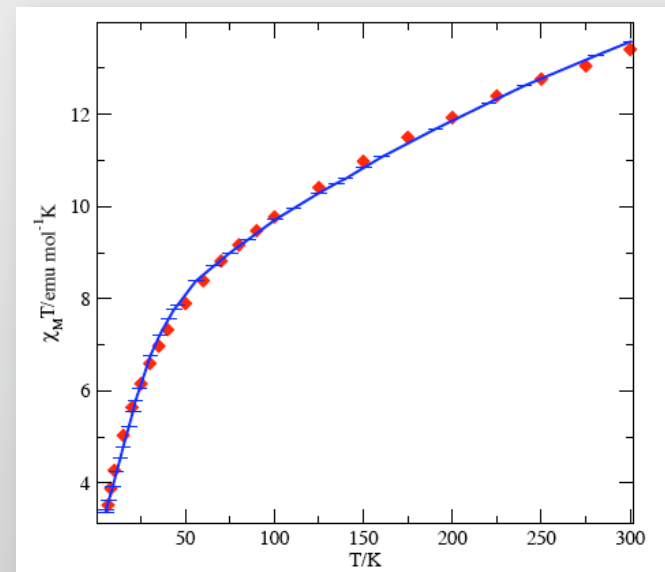
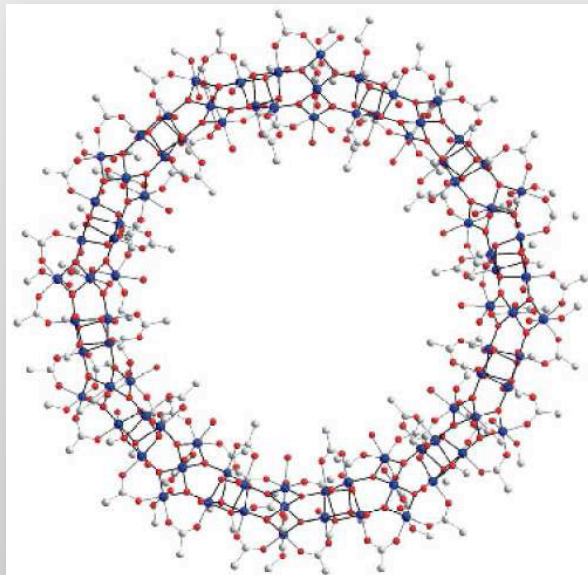


➔ compare microscopic models to experiments

# Mn-84 molecules

Vassilis Tangoulis, in preparation

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



## ALPS QMC codes

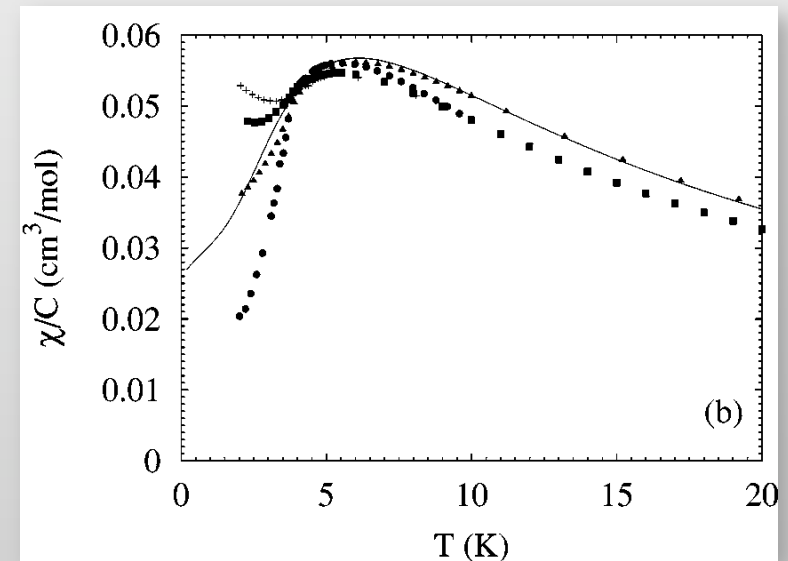
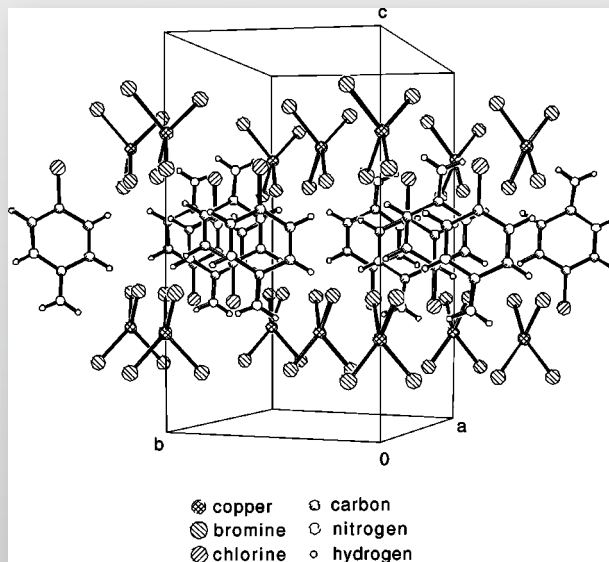
Numerical evaluation of susceptibility for full molecule:  
Fit of magnetic interaction strength.



# Low-dimensional quantum magnets

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

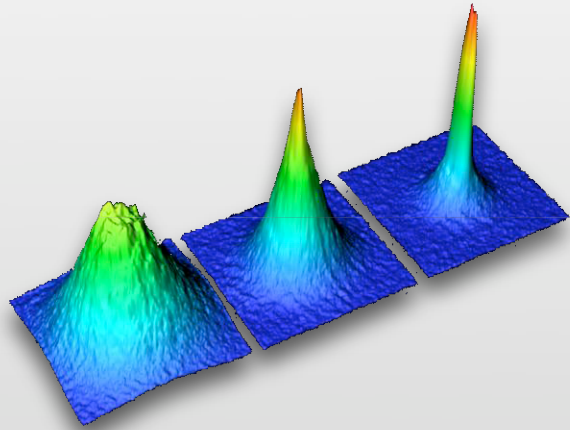
- How to characterize newly synthesized materials?



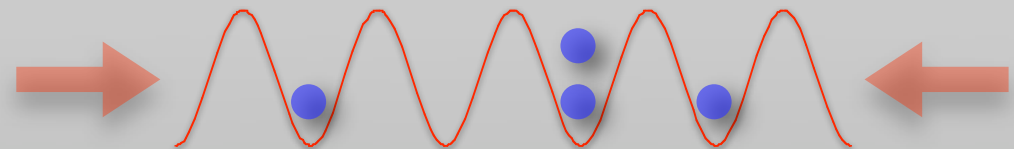
## ALPS QMC codes

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

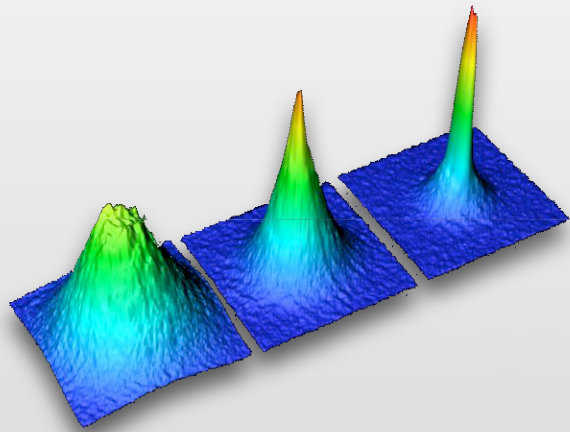
# BEC in ultracold atomic gases



T. Esslinger, ETH Zürich



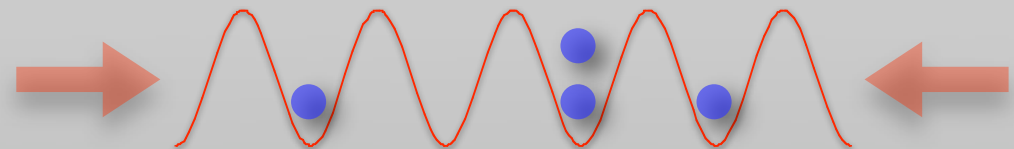
# BEC in ultracold atomic gases



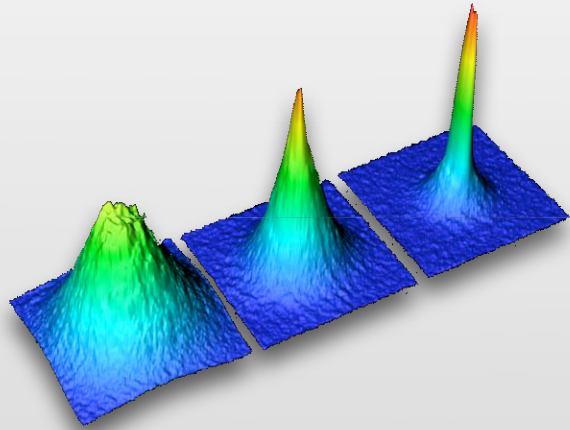
- Ultracold  $^{87}\text{Rb}$  atoms form a Bose-Einstein condensate (BEC).
  - first observed in 1995
  - sympathetic cooling of fermionic  $^4\text{He}$  atoms (2004)



T. Esslinger, ETH Zürich

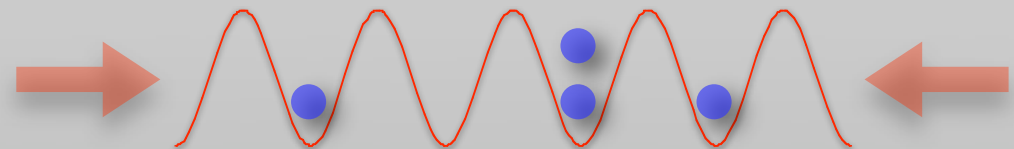


# BEC in ultracold atomic gases



T. Esslinger, ETH Zürich

- Ultracold  $^{87}\text{Rb}$  atoms form a Bose-Einstein condensate (BEC).
  - first observed in 1995
  - sympathetic cooling of fermionic  $4^{\circ}\text{K}$  atoms (2004)
- Standing laser waves form an optical lattice.

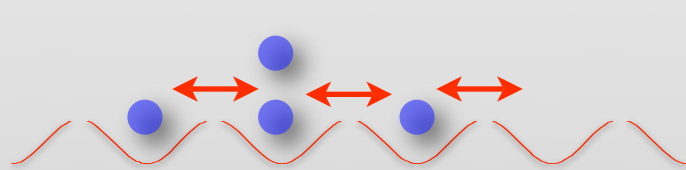


# Realization of the Bose-Hubbard model

S. Wessel et al., Phys. Rev. A **70**, 053615 (2004)

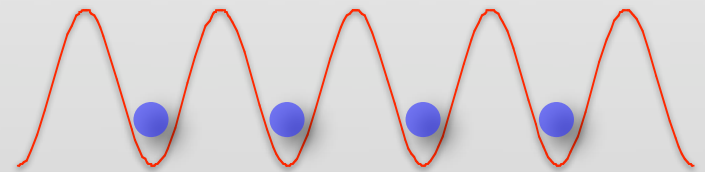
O. Gygi et al., Phys. Rev. A **73**, 063606 (2006)

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



suprafluid  
coherent BEC

local density



Mott-insulator  
incoherent

## ALPS QMC codes

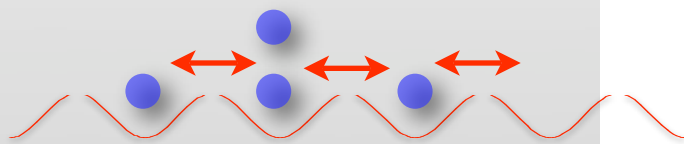
Numerical simulation of experimental setup:  
60<sup>2</sup> sites and harmonic trapping potential

# Realization of the Bose-Hubbard model

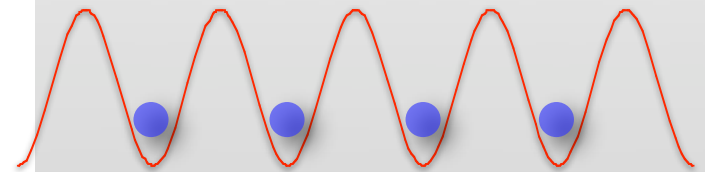
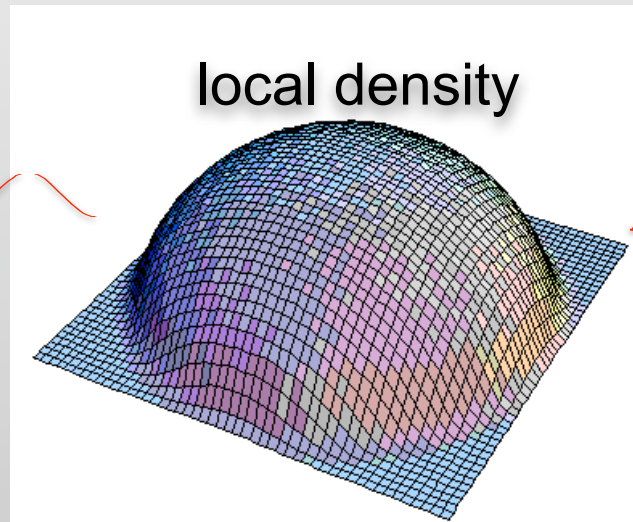
S. Wessel et al., Phys. Rev. A **70**, 053615 (2004)

O. Gygi et al., Phys. Rev. A **73**, 063606 (2006)

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



suprafluid  
coherent BEC



Mott-insulator  
incoherent

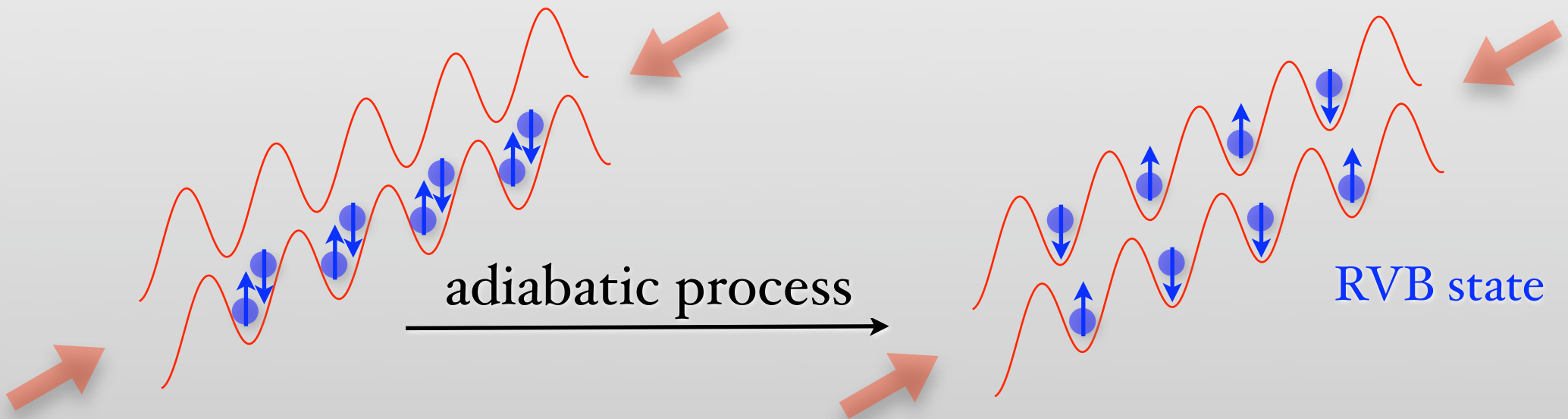
## ALPS QMC codes

Numerical simulation of experimental setup:  
60<sup>2</sup> sites and harmonic trapping potential

# Ultracold fermionic atoms

S. Trebst *et al.*, Phys. Rev. Lett. **96**, 250402 (2006)

- How can we cool down fermions to some  $0.01 T_F$ ?



## **ALPS exact diagonalization codes**

Excitation spectra of intermediate states.  
Time-evolution of proposed adiabatic processes.

# Ab-initio simulations of quantum magnets

- Simulate realistic magnetic models instead of toy models
  - obtain microscopic exchange constants from LDA+U
  - simulate quantum spin models using these exchange constants
- Was done by hand in the past
  - $\text{CaV}_2\text{O}_3$ ,  $\text{MgV}_2\text{O}_3$ ,  $\text{CaV}_3\text{O}_7$ ,  $\text{CaV}_4\text{O}_9$
  - Korotin, Elfimov, Anisimov, Troyer and Khomskii, PRL '99
- Can we automate this?



# ALPS Interface to band structure codes

- ORNL is developing standard XML I/O data formats and helper libraries for band structure codes
- Implementation in Stuttgart TB-LMTO-ASA band structure code by Anton Kozhevnikov (Ekaterinburg)
- Simple helper tool by Anton Kozhevnikov creates ALPS input file from XML output of LDA+U code
- Automated workflow from crystal structure to magnetic properties

# Example: SrCu<sub>2</sub>O<sub>3</sub>

