Guide through the project

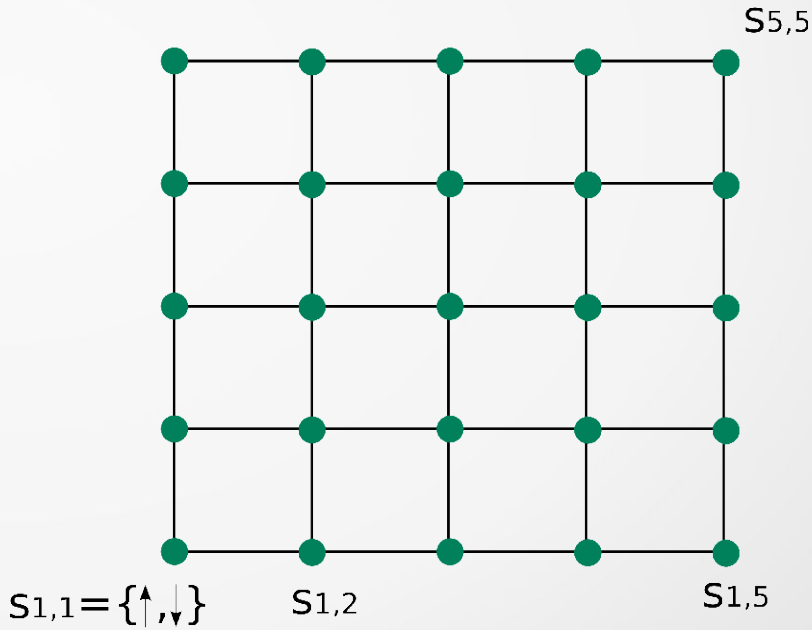
The aim of the project

To provide a numerical laboratory **(tool)** for testing the critical properties of the Ising model:

C++ implementation is provied for 2 models:

Ising chain – Ising1D C++ class

Ising lattice – Ising2D C++ class



Periodic boundary conditions are assumed.

Methodology

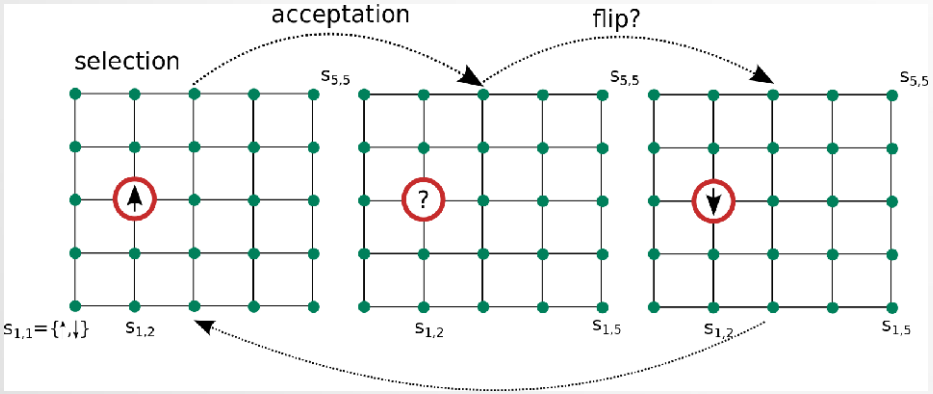
In order to simulate the above models, two numerical Monte Carlo (MC) algorithms were implemented:

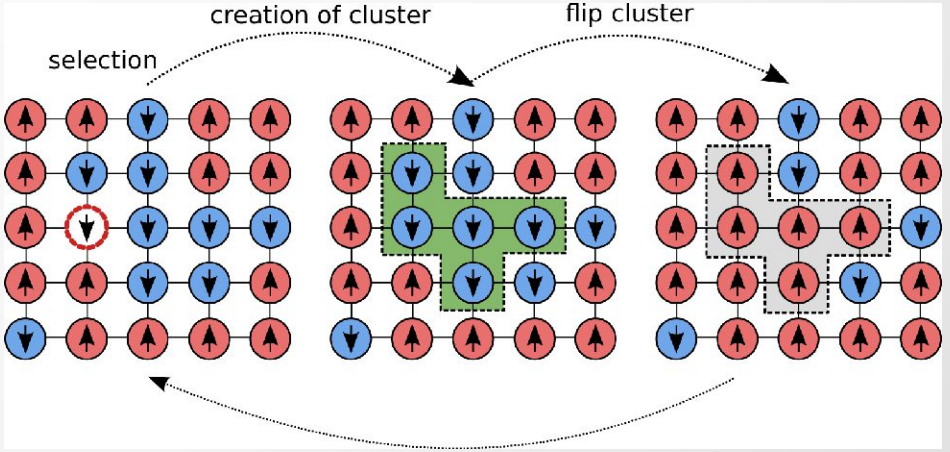
- **Metropolis method**

flips one spin with special propability at each MC cycle. The method is universal, easy to implement but not efficient at critical temperature (critical slowing down).

- **Wolff method**

derived to deal with critical phenomena allows us to flip whole clusters of spins. Very efficient at critical points, but less broadly implementable.





More details about the moethods above will be given later.

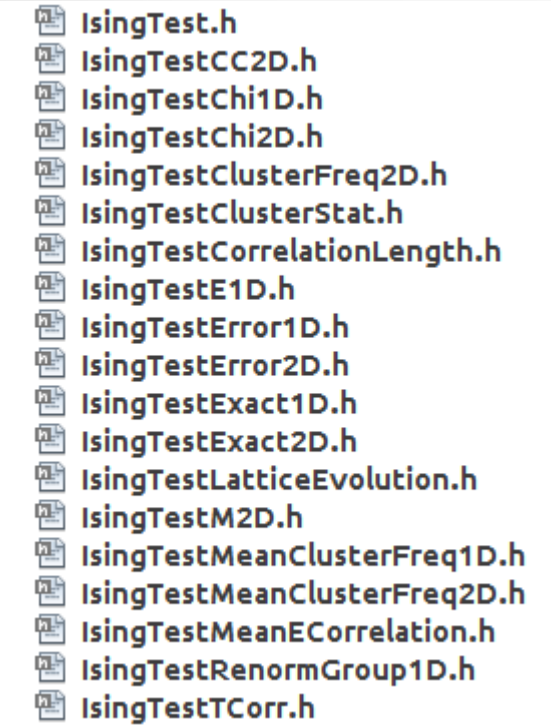
What can we simulate?

For both 1D and 2D model simulation data were used to calculate the following physical properies:

- **magnetization,**
- **magnetic susceptibility,**
- **specific heat,**
- **average energy,**
- **spin-spin correlation function,**
- **energy-energy correlation function,**
- **temperature correlation function,**
- **statictics of cluster distribution,**
- **and elaborate the following helpful tools:**
- **error analysis with Bootstrap method**
- **exact solution for small lattices.**

How can we simulate?

There are provided **test classes** for each of the above physical properties. Creating such objects of test class runs the simulation for a default input simulation



List of available tests

Each test performs following operations:

- displays basic information about test
- runs simulation/simulations
- saves data to propper files
- generates plots based on data above

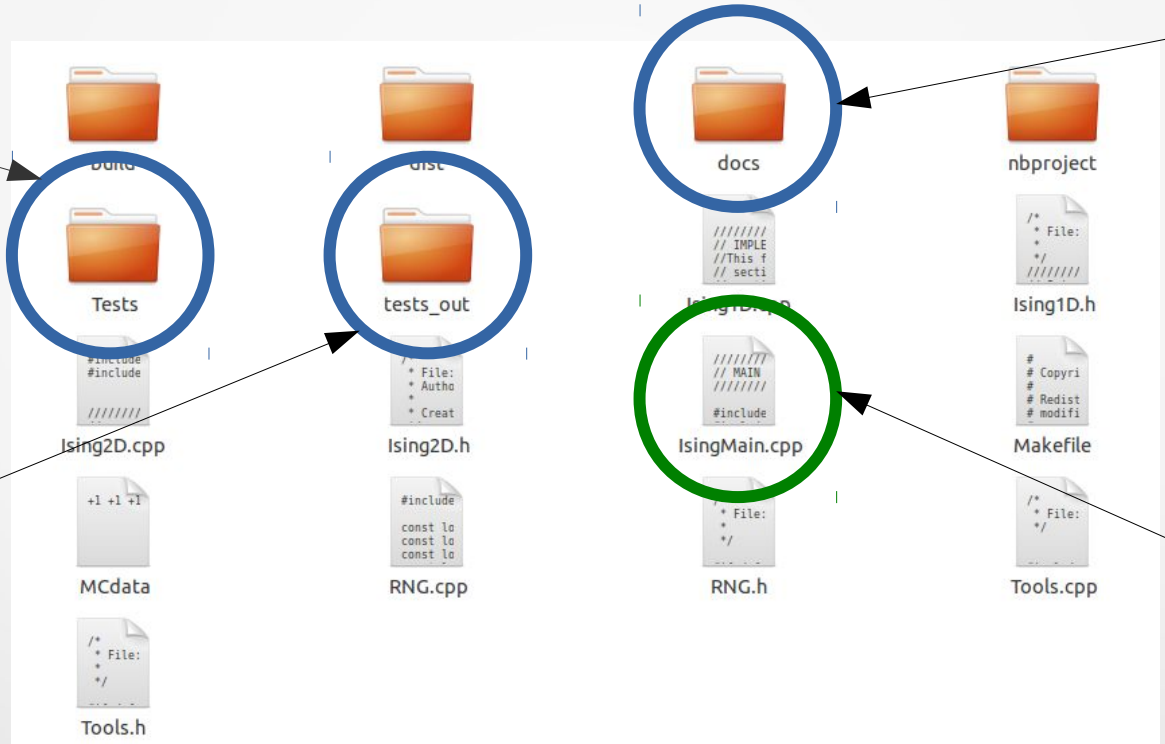
Case study – technical part

Let us consider the example where we want to run a test which will calculate the specific heat for 2D lattice.

1. We have to open the main file, which is: **IsingMain.cpp.**

Test class location

Test documentation and theory



Test data and plot output location

IsingMain.cpp

View of the folder containing the common project

Case study

Let us consider the example where we want to run a test which will calculate the specific heat for a 2D lattice.

2. To run a specific test we have to find it in **main()** function**.**

```
// IsingTestChi2D isingTestChi2D;
// IsingTestError2D isingTestError2D;
// IsingTestCv2D isingTestCv2D;
// IsingTestM2D isingTestM2D;
// IsingTestExact2D isingTestExact2D;
```

3. Uncomment selected test.

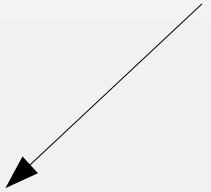
```
// IsingTestChi2D isingTestChi2D;
// IsingTestError2D isingTestError2D;
 IsingTestCv2D isingTestCv2D;
// IsingTestM2D isingTestM2D;
// IsingTestExact2D isingTestExact2D;
```

Case study

Let us consider the example where we want to run test which will calculate the specific heat for 2D lattice.

4. Save, compile and run the program.

program output



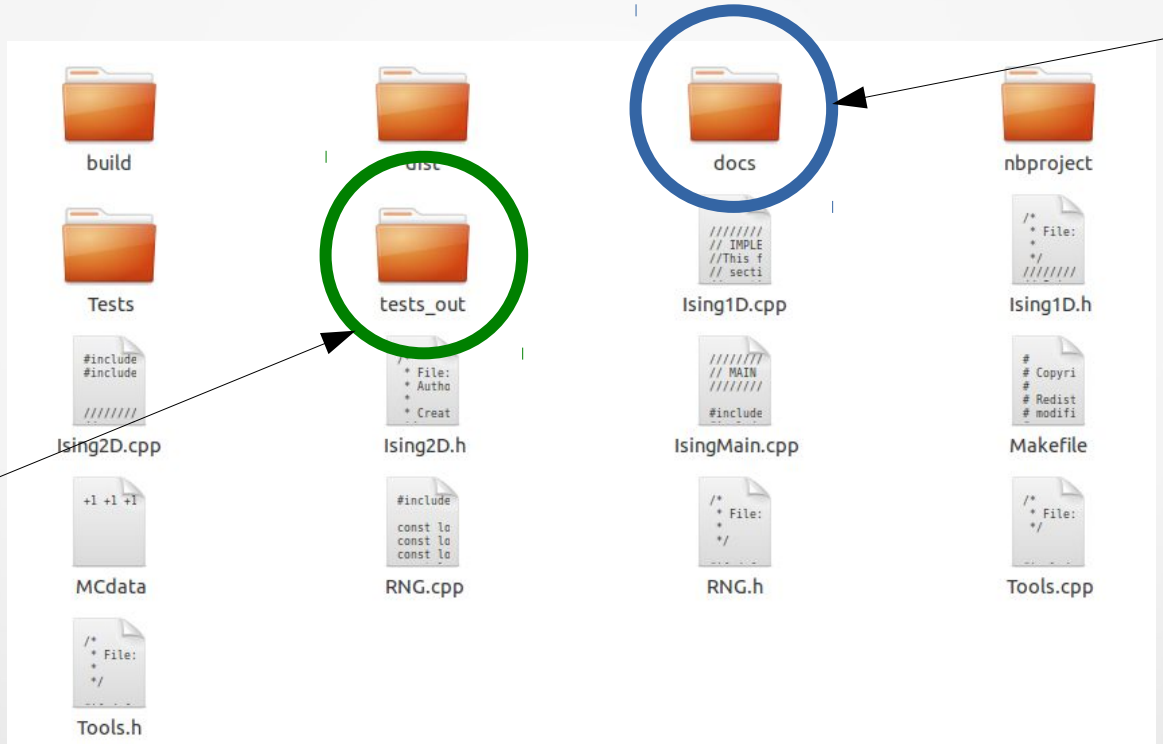
```
-----------------------------------------------------
 Starting test: Test of specific heat in function of T .
-----------------------------------------------------
 Run test to calculate the Cv value in function of
 temperature T for the 2D Ising lattice. The test is performed
 for different lattice sizes: 2x2, 4x4, 8x8, 16x16. Critical
 temperature is Tc=2.27. From theoretical results we know that
 Cv should obtain maximum value near the Tc value. For large and
 very small values of T the Cv should tend to zero.
 The results are saved in following files:
 'IsingTestCv2D.txt' and 'IsingTestCv2D.png' in 'tests_out' directory.
 See run() function for more details.
-----------------------------------------------------
 Running test...
 It may take some time...
Starting simulation for lattice 2x2...
T=1     :0%     10%     20%     30%     40%     50%     60%     70%     80%     90%
T=1.1   :0%     10%     20%     30%     40%     50%     60%     70%     80%     90%
T=1.2   :0%     10%     20%     30%     40%     50%     60%     70%     80%     90%
T=1.3   :0%     10%     20%     30%     40%     50%     60%     70%     80%     90%
T=1.4   :0%     10%     20%     30%     40%     50%     60%     70%     80%     90%
T=1.5   :0%     10%     20%     30%     40%     50%     60%     70%     80%     90%
T=1.6   :0%     10%     20%     30%     40%     50%     60%     70%     80%     90%
T=1.7   :0%     10%     20%     30%     40%     50%     60%     70%     80%     90%
T=1.8   :0%     10%     20%     30%     40%     50%     60%     70%     80%     90%
```

Case study – technical part

Let us consider the example where we want to run test which will calculate the specific heat for 2D lattice.

5. After execution, go to the **test_out** directory,

Tests documentation and theory



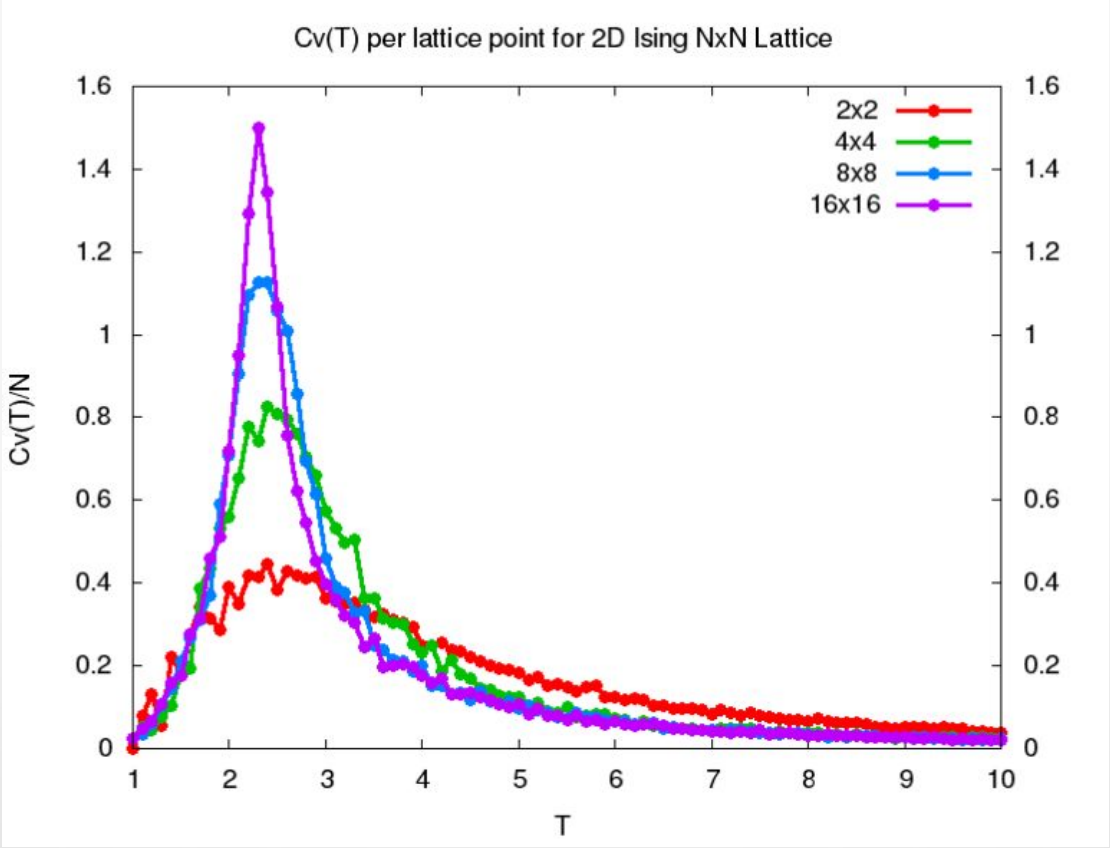
Test data and plot output location

View on the folder containing the common project

Case study – technical part

Let us consider the example where we want to run test which will calculate the specific heat for 2D lattice.

6. Find the graphics file (png or pdf) wth the same name as test class: In our case: IsingTestCv2D.png. See the result.



What else?

Additional comments:

- **Some of the tests classes have included additional documentation files (pdf files) in docs direcotry.**
- **In these files can be found: introduction to the problem and comments to the results.**
- **The whole project is located on online repository (https://code.google.com/p/pacp/) and can be downloaded from anywhere in the world.**

Thank you