

# Spin models on random bipartite graphs

Shane Harding  
Supervisor: Mike Peardon

July 2, 2015

## Abstract

A software system to investigate the properties of spin models on random graphs will be developed. To facilitate easy parallelism, the project will restrict its attention to bipartite graphs and develop MPI software. An investigation of the best way to partition a random graph into sub-domains to be handled on MPI cores will be carried out. The phase structure of these spin models will be studied in detail.

## 1 Background of problem

First we need to know what a bipartite graph is. In graph theory, a graph is a set of objects (vertices) where some pairs of objects are connected by links (edges). A bipartite graph is a graph whose vertices can be divided into two disjoint sets  $A$  and  $B$ , such that every edge connects a vertex in  $A$  to one in  $B$ .

The problem is closely related to the Ising model, in fact it is an Ising model that isn't represented on a lattice. In the Ising model each lattice site is given a spin, up or down, and then in these interact with their neighbouring sites, following rules and the spins develop and may reach a steady state.

## 2 Implementation

For this project each site, of our random bipartite graph, will be given a random spin and will only interact with the vertices that it is connected with. I will write code to simulate the process of this interactions. MPI will be used to parallelise the problem. Various methods of partitioning the graphs will be investigated and the performance of each analysed. The underlying physical meaning of the problem will also be explored.

## 3 Plan

A rough outline for what I plan to do during the course of this project is as follows: First, I will get familiar with graph theory and do the background

reading on various Ising model techniques. Once a solid understanding has been established I will write a general MPI code for the problem. I expect this to be the most time consuming part of the project. Then I will specialise the code to look at specific problems. These special cases could be graphs with unique or unusual properties that would make for some interesting parallelisation or interesting physics.

Then I will analyse performance aspects of the code. In particular the most efficient methods of dividing the graph between processors. And finally the physical meaning of the interactions and the results of the simulations will be studied in detail.