Simulating the Ising model using the Metropolis algorithm

Johan Nereng

Department of Physics, University of Oslo, Norway

Nov 21, 2018

1 Abstract

2 Introduction

For the purpose of approximating the derivatives of u, three methods are applied: explicit forward Euler, implicit Backward Euler, and the implicit Crank-Nicolson with a time-centered scheme.

3 Methods

$$\nabla^2 u(x,t) = \frac{\partial u(x,t)}{\partial t} \tag{1}$$

$$u_x x = u_t \tag{2}$$

Initial conditions, boundry conditions

Before approximating the derivatives, the problem discretized. The spacial domain , $x \in [0, L]$, $L = 1$, is discretized over $n + 1$ grid points, such that $x_i = i\Delta x$, $i = 0, 1, ..., n, n + 1$, where $\Delta x = \frac{1}{n+1}$. Time, $t = [0, T]$, is is expressed as $t_j = j\Delta t$, $j \geq 0$, where $\Delta t = \frac{1}{T}$. The dizcrete approximation of the function u is then defined as $u(x_i, t_j) = u_{i,j}$, with boundary conditions (B.C) $u_{0,j} = 0$ and $u_{n+1,j} = 1$.

The solution to the problem is assumed to be u, and in order to dimdum, a change of variables from u to v is applied, where v is discretized similarly to x: $v(x, t) = v_{i,j}$. $u_{i,j} = v_{i,j} + x \implies v_{i,j} = u_{i,j} - x_i$. This changes the B.C to

$$v_{0,j} = v_{1,j} = 0$$

Recalling the initial conditions, $u_{i,0} = 0$, for $i < L$, $\implies v_{i,0} = u_{i,0} - x_i$

bytt med v Approximating derivatives: Explicit forward Euler (assignment paper/lecture notes):

$$u_t \approx \frac{u(x_i, t_j + \Delta t) - u(x_i, t_j)}{\Delta t} \tag{3}$$

$$u_{xx} \approx \frac{u(x_i + \Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j)}{\Delta x^2} \tag{4}$$

$$\tag{5}$$

Applying compact notation introduced in the discretization:

$$u_t \approx \frac{u_{i,j+1} - u_{i,j}}{\Delta t} \tag{6}$$

$$u_{xx} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \tag{7}$$

$$\tag{8}$$

Where u_t has local truncation error (LTE) $O(\Delta t)$ and $u_x x$ has $O(\Delta x^2)$.

Setting $u_t = u_{xx}$, defining $\alpha = \Delta t / \Delta x^2$, and solving for $v_{i,j+1}$ yields what is known as the explicit scheme (ref lec notes):

$$v_{i,j+1} = \alpha v_{i-1,j} + (1 - 2\alpha) v_{i,j} + \alpha v_{i+1,j} \tag{9}$$

As $v(0, j) = v(n + 1, j) = 0$ for all j, the time dependent part of the solution is now expressed as:

$$V_j = \begin{bmatrix} v_{1,i} \\ v_{2,i} \\ \dots \\ v_{n,i} \end{bmatrix} \tag{10}$$

Similarly, implicit backward euler:

$$u_t \approx \frac{u_{i,j} - u_{i,j-1}}{\Delta t} \tag{11}$$

$$u_{xx} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \tag{12}$$

$$\tag{13}$$

4 Algorithms