Simulation of Brownian motion

Modelling of Physical Systems

Piotr Cenda
Kraków, 08.03.2022

1 2D Brownian motion simulation programme

To simulate 2D Brownian motion the following code was used:

```
n_particles = 1000;
n_steps = 1000;

x = zeros(n_particles, 1);
y = zeros(n_particles, 1);

for n=2:n_steps
    dx = randn(n_particles, 1);
    dy = randn(n_particles, 1);
    x = [x x(:, n-1)+dx];
    y = [y y(:, n-1)+dy];
    plot(x', y');
    xlabel("x coordinate");
    ylabel("y coordinate");
    xlim([-50 50]);
    ylim([-50 50]);
    pause(0.001);
end
```

It computed, for given number of particles and steps, motion for each particle, by appending new coordinates to the vector, which were calculated by adding random (normally distributed) number to last particles' positions.

2 2D Brownian motion simulation visualisation examples

Using shown code, visualisations were made for different numbers of particles and steps.

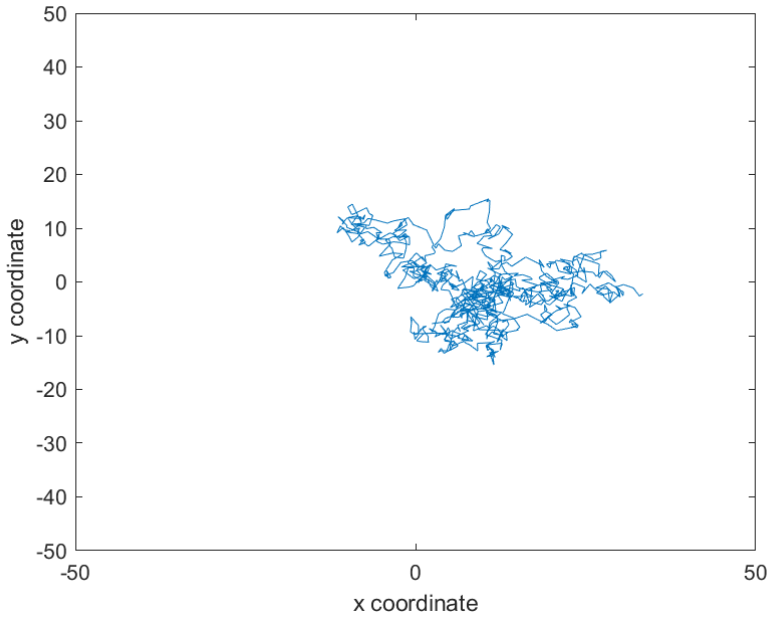

Figure 1: 2D Brownian motion for 1 particle for 1000 steps

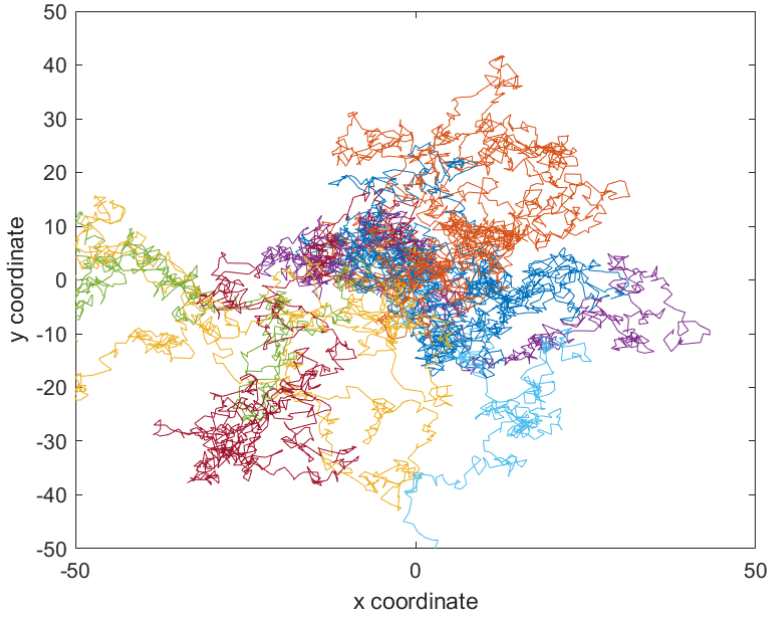

Figure 2: 2D Brownian motion for 10 particles for 1000 steps

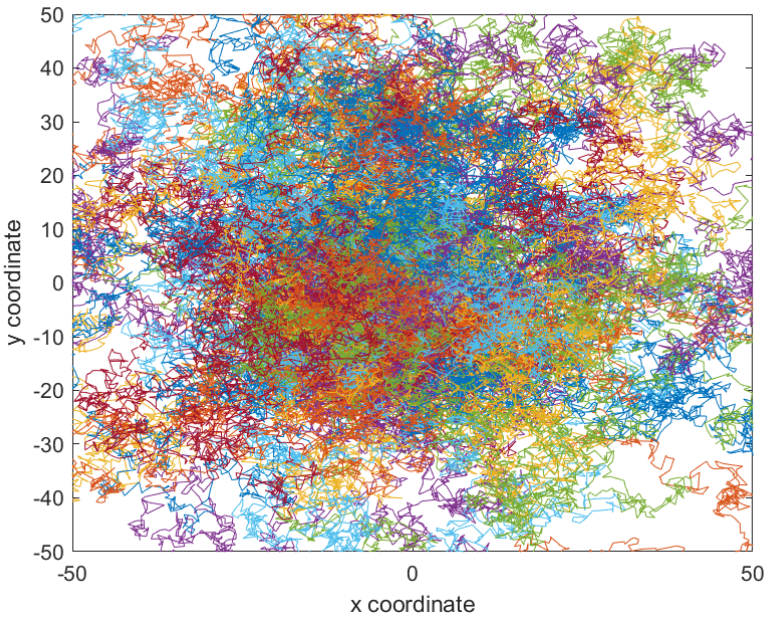

Figure 3: 2D Brownian motion for 100 particles for 1000 steps

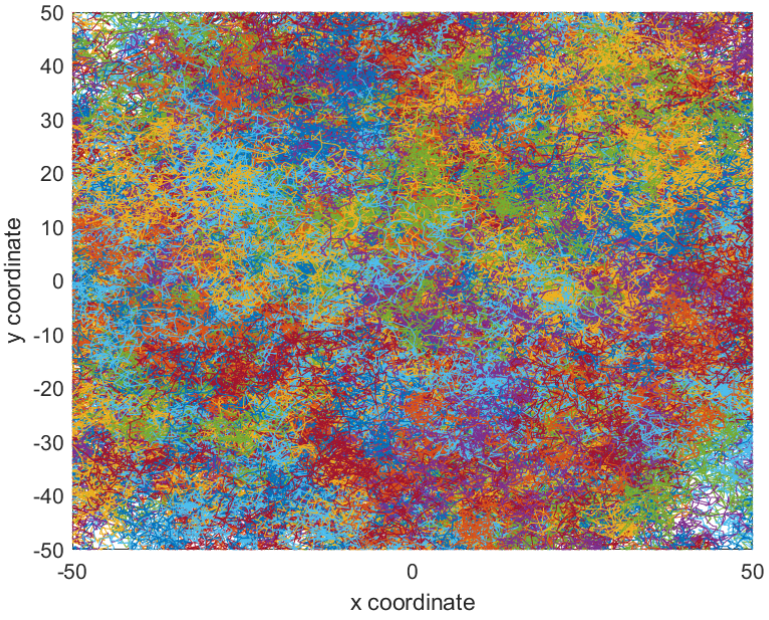

Figure 4: 2D Brownian motion for 1000 particles for 1000 steps

All of these simulation proves that particles move randomly and are independent from each other.

3 Mean square of displacement

To compute and plot mean square of displacement the following code was used:

```
means_sq = mean((x.^2 + y.^2));
plot(means_sq, '.b');
xlabel("step");
ylabel("mean square of displacement");
```

Mean square of displacement is calculated as mean of sum of squares of all particles' coordinates for each step. Relationship between mean square of displacement and number of steps using model created in point 1 for 1000 particles gives function similar to linear ($f(x) = 2x$).

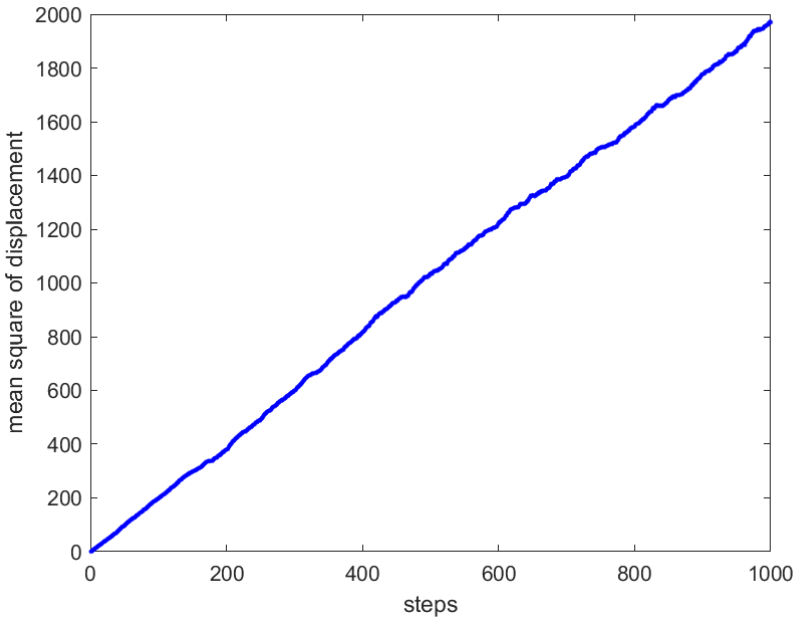

Figure 5: Mean square of displacement for 1000 particles

4 Autocorrelation

To demonstrate self-similarity property of brownian trajectories autocorrelation was calculated using the following code:

```
plot(xcorr(x(1, :)));
xlabel("samples");
ylabel("autocorrelation");
```

It was used to plot autocorrelation of one random particle's Brownian motion.

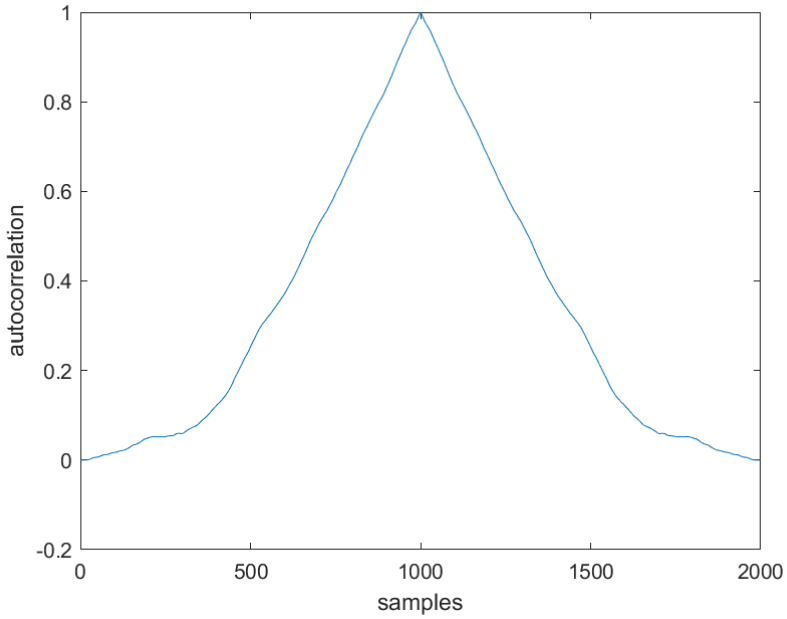

Figure 6: Autocorrelation of Brownian motion for one of particles

As shown, computed autocorrelation proves that postion of particle in Brownian motion is dependent on last coordinates of particle.

5 Conclusions

Presented simulations of Brownian motion show that motion for one particle is autocorrelated, but particles between themselves are independent. Additionally it shows that Monte Carlo methods are very easy and reliable methods to simulate physical systems. Theirs only dependency is need to use huge number of samples to perform well and give accurate results - the more samples the better.