NDMansfield, a Monte-Carlo (lattice) polymer simulator

“ndmansfield” is a simple program which generates a series of self-avoiding space-filling curves (also called “lattice Hamiltonian paths”).

It has been used (among other things) to simulate the shapes of polymers packed into confined spaces.

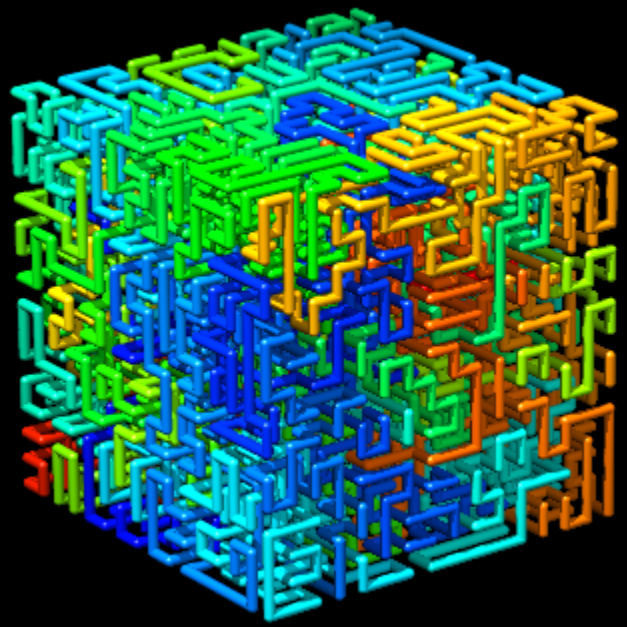
“polymer”

=

a long (floppy) molecule

Example: randomly-generated self-avoiding polymer (L = 4096 b)

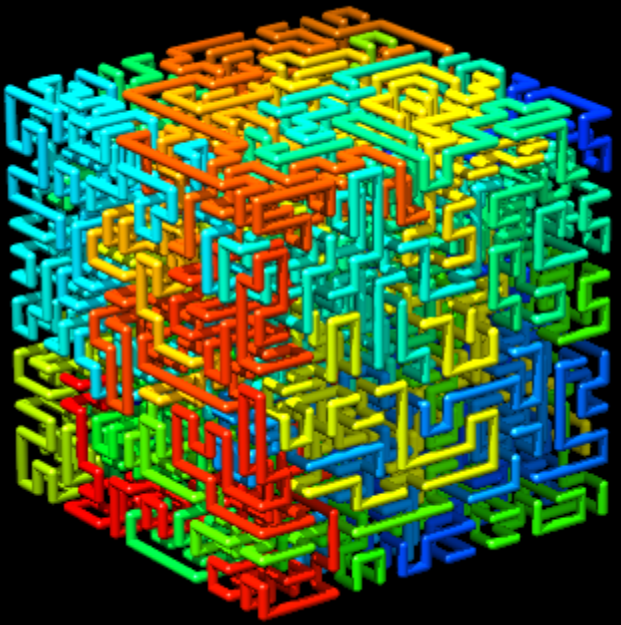
Lieberman-Aiden et al., Science, 2009



Mansfield, J.Chem.Phys. 2006

Example: randomly-generated self-avoiding polymer (L = 4096 b)

Lieberman-Aiden et al., Science, 2009



Mansfield, J.Chem.Phys. 2006

Generating random polymer shapes

- What do you mean by “random”?

Generating random polymer shapes

- What do you mean by “random”?

  Let's restrict ourselves to polymers which live on a “lattice”. (For example, a checker board.)

Generating random polymer shapes

- What do you mean by “random”?

  Let's restrict ourselves to polymers which live on a “lattice”. (For example, a checker board.)

- One strategy: Start at a randomly chosen location and move in random directions until you fill the box.

Generating random polymer shapes

- What do you mean by “random”?

  Let's restrict ourselves to polymers which live on a “lattice”

- One strategy: Start at a randomly chosen location and move in random directions until you fill the box. Problems:

  You usually “box yourself” into a corner.

Generating random polymer shapes

- What do you mean by “random”?

  Let's restrict ourselves to polymers which live on a “lattice”

- One strategy: Start at a randomly chosen location and move in random directions until you fill the box. Problems:

  You usually “box yourself” into a corner.

  It's not clear if it is random ***(depends on where you start)***

Generating random polymer shapes

- What do you mean by “random”?

  Let's restrict ourselves to polymers which live on a “lattice”

- One strategy: Start at a randomly chosen location and move in random directions until you fill the box. Problems:

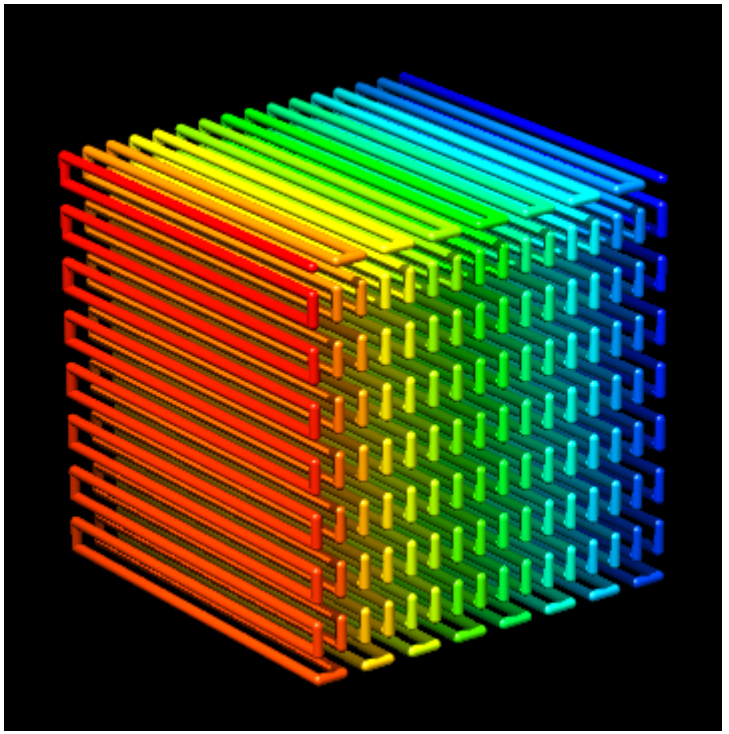
  You usually “box yourself” into a corner.

  It's not clear if it is random ***(depends on where you start)***

- ***Alternate approach:*** Use ***“Monte Carlo”*** (explained later…)

Monte Carlo

- Start with any shape



Monte Carlo

- Start with any shape
- Move it in a random way

J. Chem. Phys. **125**, 154103 (2006)

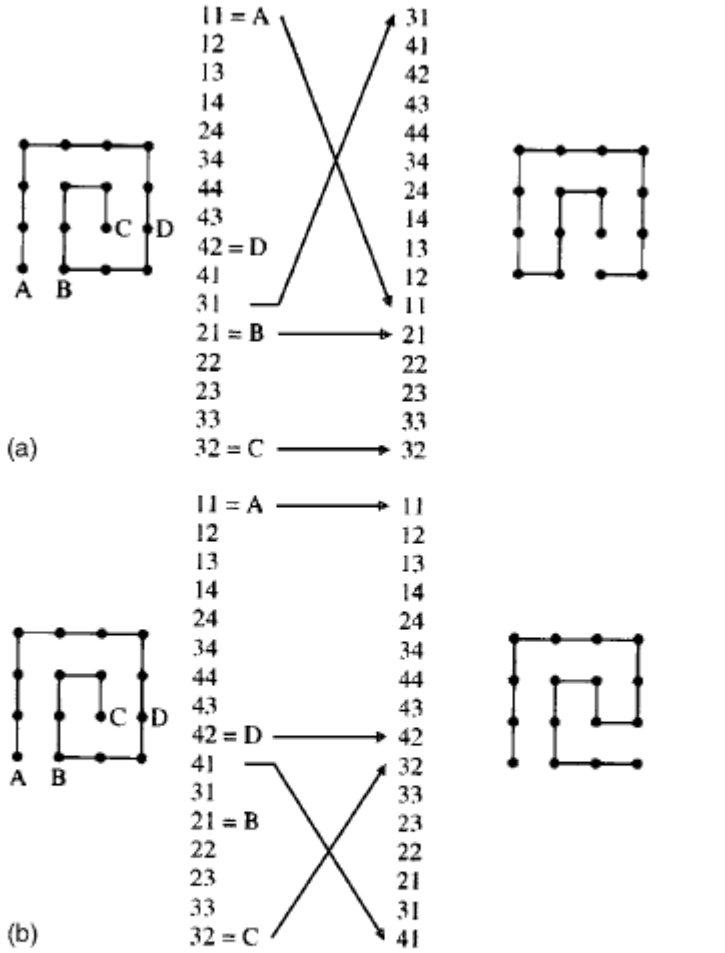
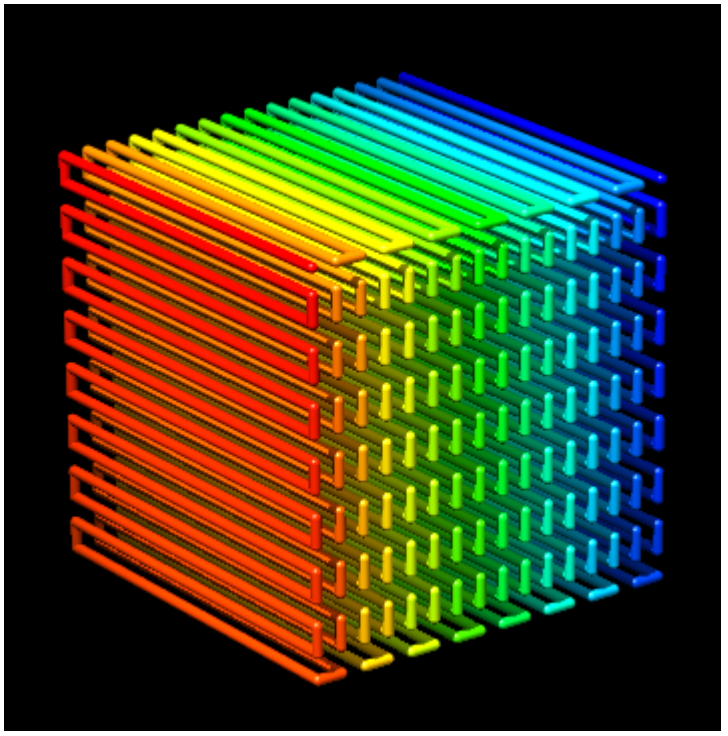


FIG. 1. (a) A Hamilton path on a $4\times4\times1$ lattice and its associated list, in which sites are specified by their coordinates. The head of the path is site A with coordinates $xy=11$. One of the nearest neighbors of A is the site $B=21$. A new path may be generated from the old one by reversing the order of the half-list lying above B. (b) Another possible outcome occurs if we select the tail, $C=32$, and its neighbor $D=42$. Then we reverse the half-list lying below D. After each maneuver, only a single edge has moved, but the end of the walk has jumped to a new lattice site.

Monte Carlo

- Start with any shape

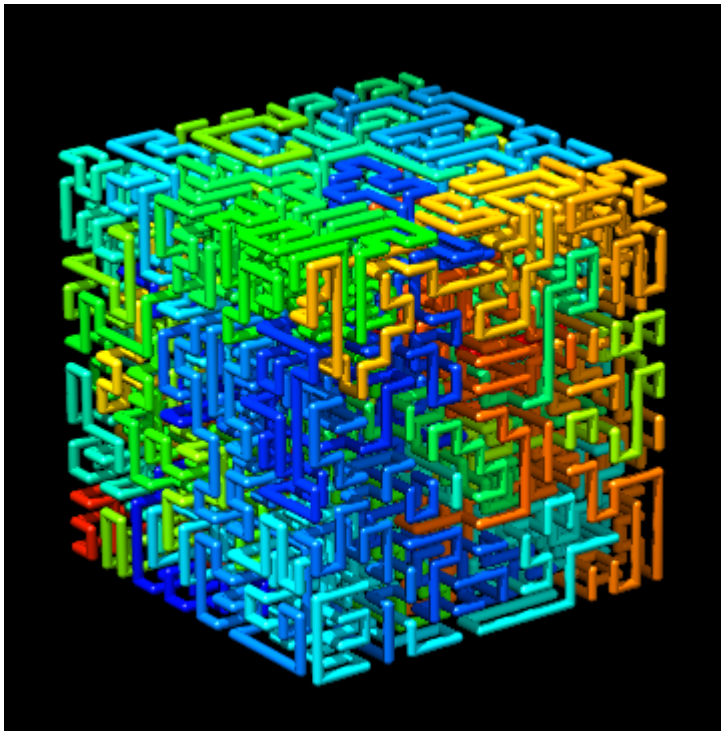
- Move it in a random way



Monte Carlo

- Start with any shape

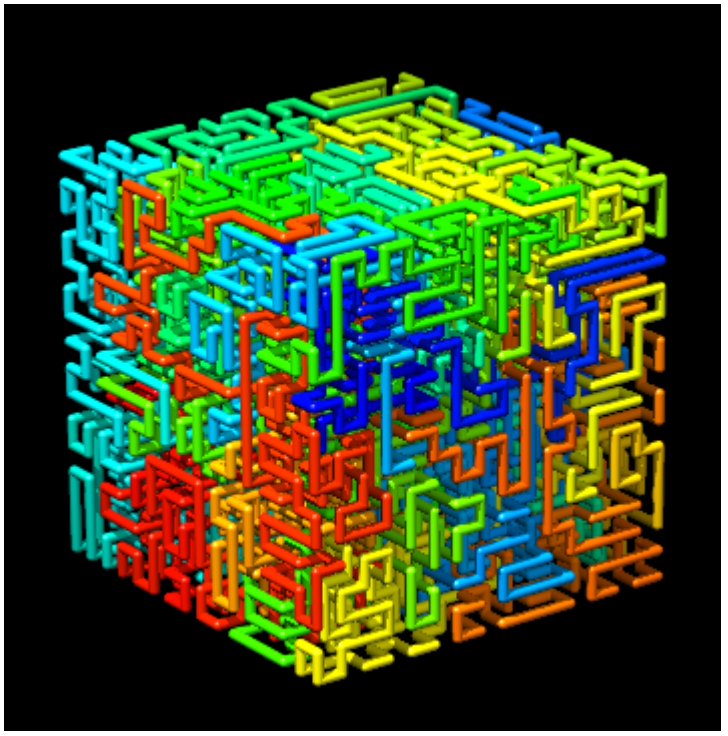
- Move it in a random way



Monte Carlo

- Start with any shape

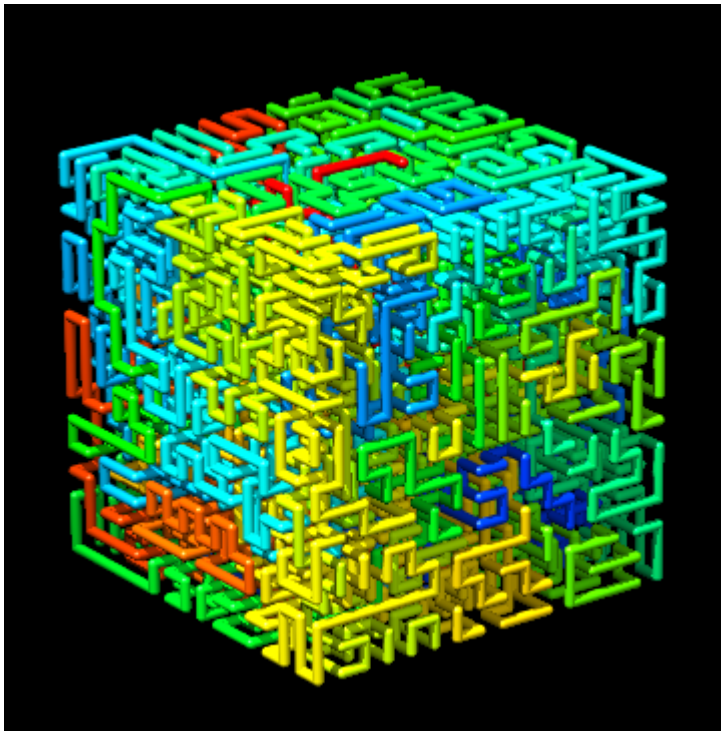
- Move it in a random way



Monte Carlo

- Start with any shape

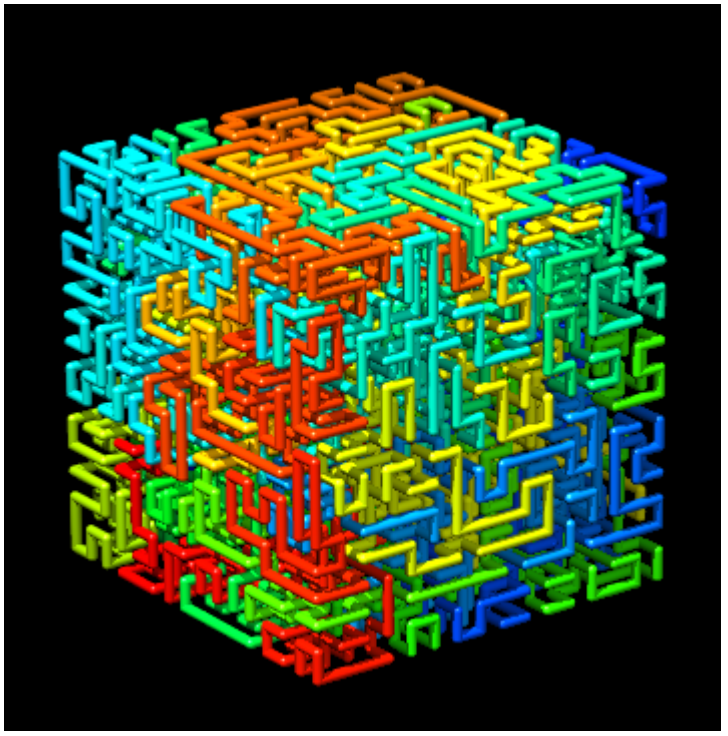
- Move it in a random way



Monte Carlo

- Start with any shape

- Move it in a random way



Monte Carlo

- Start with any shape
- Move it in a random way
- Do this for a LONG time.
- ***Eventually***, the shapes you get will be **random**

Code Outline:

- ***class Ndmansfield***

  This stores information about the shape of the polymer, and functions which change it.

- ***class Hamiltonian***

  This calculates the forces acting on the polymer which can bias one shape in favor of another.

Code Outline:

- ***class Ndmansfield***

  This stores information about the shape of the polymer, and functions which change it.

  MonteCarloStep(Hamiltonian& hamiltonian,

  double& total_energy);

- ***class Hamiltonian***

  This calculates the forces acting on the polymer which can bias one shape in favor of another.

  double CalcEnergy(NDmansfield& ndmansfield);