20151023 wiki (Simulation) \_JamboreeVer4.0

2015年10月23日

12:53

To confirm our design concept, we developed a simulation software using the physical engine Unity [1], which allows us to explore the space of design parameters and predict the behavior of real nano objects.

|  |
| --- |
|  |
| Movie 1 The scenes of the simulation |

Model

Since it takes too much computational cost to represent the 3D model of the monomer in detail, we have to made necessary simplifications in the simulation.

* Simulation space is set as a cube with periodic boundaries to reduce the computing time. Hundreds of monomers placed randomly in a space with determined concentration

|  |
| --- |
|  |
| Fig 1. The simulation space |

* The monomers are simplified as cylinders, which is cost-efficient for the physical engine.
* The stacking of the monomers takes place as the collisions among the cylinders.

Interface

The cylinders collide mutually based on Brownian motion. We created a user interface with several scripts:

* To adjust the space size to fit with the desired concentration.
* To modify the velocity and angular velocity of each monomer.
* To output the result into an appropriate file format.

To simulate the collision efficiently, we modify the position of monomer, which goes out of the boundaries to satisfy the periodical boundary. Instead of combining two objects, we erased one of the object and generate a new object that matches the length with the addition monomer. It is possible to separate the object into two pieces.

Results and Vision

In our simulation, the Brownian motion of each monomer and the stacking behavior among them is reproduced. The results correspond to the temperature at 8 and 250 nM concentration of DNA origami in a span of 8 seconds. The graph (Fig. 2) and the histogram (Fig. 3) show that monomer concentration decreases while the multimers such as dimers and trimers increase overtime.

|  |
| --- |
|  |
| Fig.2 The monomer repartition |

|  |
| --- |
|  |
| Fig. 3 Histogram of the number of monomer |

There are still some unrealistic motions due to the limitation of the Unity engine, since this engine is not made to work on the molecular scale, it cannot handle colliding objects at high speed. We plan to improve the simulation in the following ways: (1) Introducing CUDA to reduce the computational time. (2) Employing parallel processing to compute the time development in parallel to obtain average data.

If you are interested in this program, please visit here (URL).

Reference

[1] Unity: http://unity3d.com/