Simulated Snowflakes

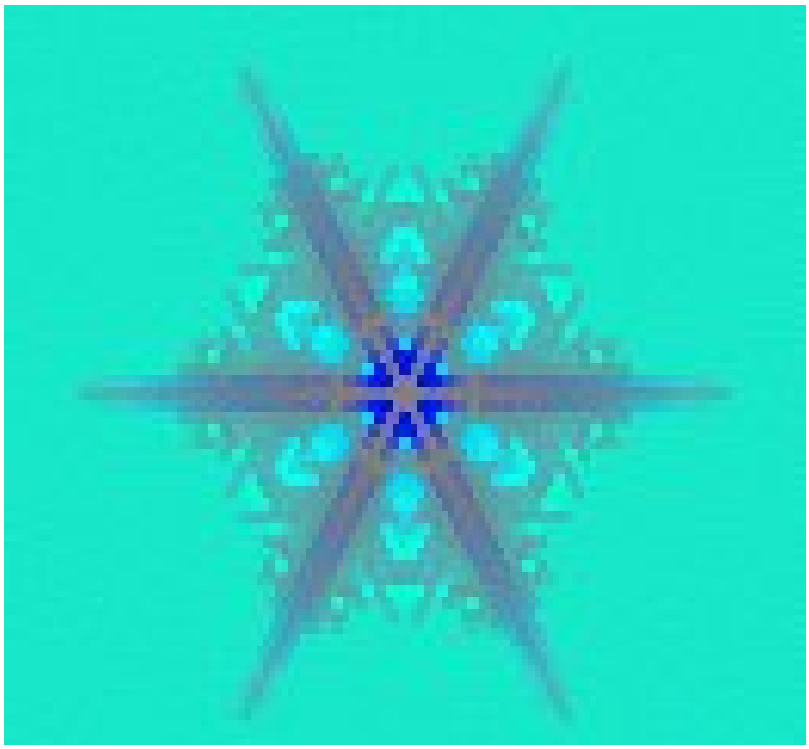

Table of Contents

Introduction

There is much that scientists don't yet know about the physical world. One area of much ongoing research is how some molecules bond together. Computer software is now available to simulate how atoms and molecules move and how many of them bond. Some types of molecular bonding are still not well understood. Research to discover more about the process of molecular bonding may increase our ability to understand, predict, and shape our environment.

There are two primary methods of studying molecular bonding. One approach is the development of a controlled physical experiment. A second approach is to develop specific computer simulations of the molecular bonding process.

In this project, an attempt will be made to develop a computer simulation of the formation of a snowflake, a process characterized by the crystal lattice form of molecular bonding. While understanding the process of snowflake formation has little immediate practical use, the crystal form of molecular bonding is very common in our environment. For example, silicon crystals would be a useful bond to know more about because of the growing computer chip industry. Snowflakes are a universal medium in which the crystal growth can be easily displayed.

Experiment

Problem: Can a computer simulate the formation of snowflakes and, if so, to what degree of accuracy?

Null hypothesis: Computers will not be able to simulate the formation of snowflakes with a molecular dynamics program because of the compute intensity. There will be some other kind of program that will achieve the purpose.

Alternative hypothesis 1: Computers will not be able to simulate the formation of snowflakes because of the vast number of elements involved.

Alternative hypothesis 2: Computers will not be able to simulate the formation of snowflakes because of the compute power necessary.

Alternative hypothesis 3: Computers will not be able to simulate the formation of snowflakes correctly because how molecules move when 'freezing' and 'melting' is too complex to simulate.

Alternative hypothesis 4: Computers will be able to simulate snowflakes when the variables such as air currents, relative humidity, etc. are randomly set and regulated by the computer or imputed manually.

Alternative hypothesis 5: Computers will not be able to simulate the formation of snowflakes because our knowledge of crystalline structure formation is incomplete and/or inaccurate.

Procedures

GROMACS

First, a program must be found or written that will simulate the formation of the crystal lattice[1]. As this is a popular issue, many programmers have made their simulators available for free download so that other programmers can use and modify them. GROMACS is a highly recommended program. GROMACS is an acronym that stands for GROMACS Runs On Most of All Computer Systems. Funny, it doesn't work on Microsoft Windows.

Little is known about how the water molecules bond (see research) and create branching instability. Furthermore, it would take about 578,703 days on a 1 gigaflop computer to produce one second of simulation with only 360 molecules. Even on the fastest computer in the world today (Red Storm, currently being built by Sandia Labs) it would take 57 days for the same simulation, assuming there are no power outages, bugs, and system or network failures. Snowflakes take minutes (in real-time) to form and have thousands of molecules.

Given the number of complexities in the snowflake making process, simulating one on a MD program such as GROMACS is currently not possible. With more research, it is possible that there are other kinds of less compute intensive programs or methods that could be used. For the time being, a MD program can be used to simulate the freezing and melting of ice/water. Inferences can then be made concerning how close a match exists between the computer simulation and the physical reality of the snowflake making process.

GROMACS displays a box with molecules in it that can be rotated. The box is set with periodic boundary conditions. This means that if a molecule floats through a side of the box, it will appear on the opposite side. This is useful because it keeps the number of molecules constant while not having them interact with walls that are made of, say,

[1] *First do all this with a MD program, and then move onto different kinds of programs.*

metal. The molecules don't bounce off the edges and disturb the simulation. This situation is never found in nature, but is a useful simplification on the computer.

Results

Figure 1 and Figure 2 is how ice appears in GROMACS. GROMACS illustrates the freezing and melting of water, but there is no branching instability.

This is not what is needed. Although snowflakes are made of ice, the structure and build is completely different. Through this experiment and a few others, it has been determined that simulating a snowflake with this program (or any other one on the molecular level) isn't possible with today's technology. This idea must be completely discarded.

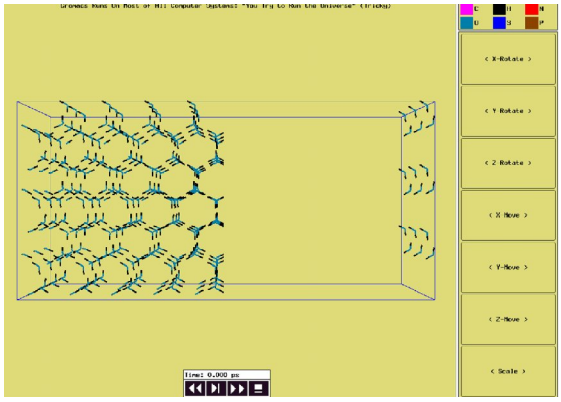

Figure 1
Notice the honeycomb-like structure and periodic boundary conditions.

Cellular Automata

In a paper by Angela M. Coxe and Clifford A. Reiter (see bibliography), another method is described. They do not have molecules bond onto a dust particle and then onto each other. Instead, they have a hexagonal grid, each hexagon cell representing a 'particle' that assigned a value of 0-1. If the cell is of value 0, it is not frozen. From 0.5+, it is frozen.

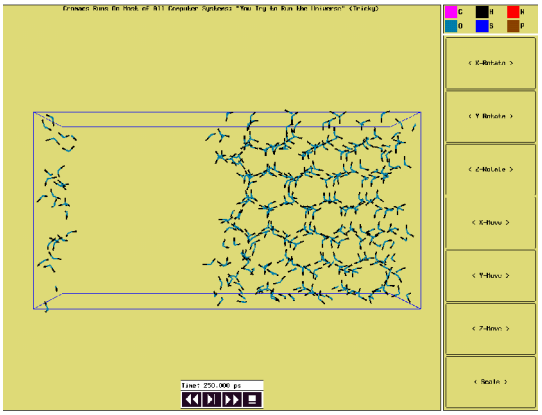

Figure 2
This is what it looks like after 250 ps of simulation, in which it has melted slightly.

Each circle represents a hexagonal cell in a cluster is called the neighborhood as shown in Figure 3. The two yellow lines are the axes used in this program. The small number on the left and right of each cell is the y and x-coordinates.

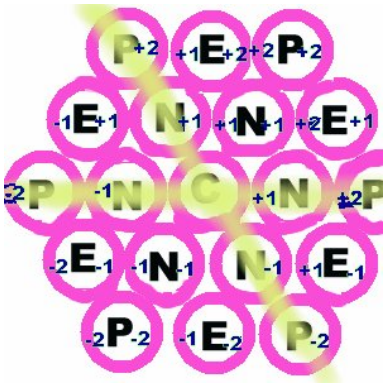

Figure 3
This is a 'neighborhood' of cells. Cell C is being calculated. Each neighborhood is divided into close neighbors (N), far neighbors (E), and corner neighbors (P).

In the initial conditions of the program[2], all cells, except one, are set to a background temperature. The one cell is set to 1. Then the computer calculates whether each cell will freeze on the next timestep by adding up the number of N, E, and P cells. For more details on how this is computed, see program pages 10 and 11.

Results

The results of the cellular automata program are

[2] *The program used in this project is not the same one used in the paper by Coxe and Reiter. Pages 1-9 and 12-15 of the program already existed on the computer. The main function of the pre-existing program was changed into the main program (see separate section labeled program). The equations and technique are the same as the ones described in the paper by Coxe and Reiter.*

satisfactory. Each time you change the background temperature, the snowflake changes. The board displays a few of these highlights. Eventually, a color function was inserted that displays color according to what timestep the part was formed on.

Conclusion

With the molecular dynamics program (GROMACS) you were able to simulate the freezing and melting of ice on the molecular level. If you ran the program with the same initial conditions, it would yield the same results. It was slow and accurate for it's purposes. It did not, however, form snowflakes because of the compute power necessary and the complexity of branching instability.

The cellular automata program created a snowflake-like pattern. It used a modified theory of how water freezes to create the effect. If you ran the program with the same background temperature, it would yield the same results. It was solely 2-d. This program was not as realistic or true-to-life as GROMACS, but it achieved a practical and useful outcome, from which more data concerning the snowflake bonding process can be illistrated.

Research

Snowflake Formation

Snow or, more correctly, ice crystals, are created when water molecules are cooled past freezing point, and condense on microscopic dust particles. The temperature and humidity of the air are the two factors that most influence the shape of an ice crystal. There are other events that make the ice crystals unique, such as collision with other crystals on the way down, or melting and re-freezing. Only about 1% of all snowflakes are completely symmetrical.

Ice crystals are always six-sided because of the shape of the water. When the crystallization takes place, each new crystal bud attaches at a 60-degree angle, creating a number of triangles, or a crystal lattice, as in Figure 4. After these are attached, more add on, and because the molecules only attach on in one way, the crystal stays in a relatively symmetrical shape on all sides.

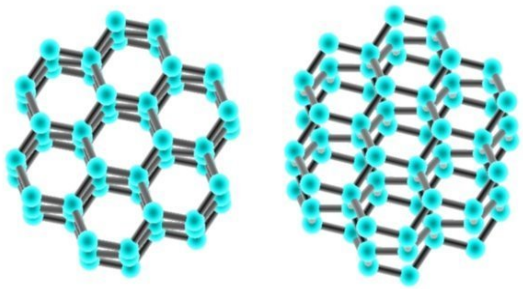

Figure 4
This is a picture of a crystal lattice.

This adding on crystallization method also means that the tiniest defects in the crystal are magnified. If there is one water molecule that sticks out on the crystal, other atoms are more likely to stick to it. This process is called branching instability. This is also why snowflakes sometimes have a lacy appearance.

When it is very cold, the snowflakes are more powdery because water molecules don't bond as well in cold temperatures. The flakes are just a few molecules that fused, creating a small ball. At warmer temperatures, near freezing, the snow is intricate and larger; a formation called a stellar dendrite. You may even get a 12-sided snowflake, from two six-sided ones that fused right on each other. If there is little wind, these large, delicate snowflakes are less likely to break.

However, stellar dendrites make up a very small percentage of the snow that actually falls. Most is 'dry' snow created at cold temperatures. The following are other common kinds of snow;

Sectored plates: A number of plates put in different combinations, these are the flakes that most resemble stellar dendrites.

Hollow snow: A hollow crystal.

Needles: These are created at very cold temperatures.

Spatial dendrites: A clump of different kinds of snowflakes that fuse, creating an interesting, if not beautiful, jumble of patterns.

Rimed crystals: These are created when a cloud freezes, but there is little wind, so plates form that are covered with other small growths.

Irregular flakes: A mix of fused and broken crystals, normally formed at low temperatures and high winds.

Molecular Dynamics Simulations

More commonly know as MD simulations, these are programs that calculate how molecules move. With modern high-speed computers, these simulations can apply various forces to atoms. These forces can include attraction and repulsion between

molecules, gravity, energy, electricity, temperature, magnetic forces, air currents, light, and many other things, depending on the nature of the simulation.

The computer calculates in timesteps, or the amount of real-time between each calculation. The forces affecting each molecule are calculated, and the molecule will move in whatever direction for however long the timestep is. If the timestep is too long, molecules may 'teleport' right next to each other, causing them to shoot away from each other at infinite speed on the next timestep because of molecule repulsion. This adds energy to the simulation at an increasing high rate.

If the timestep is too short, the computer may end up calculating forever, or very nearly. This is a huge problem for many weather simulations. The smaller the timestep is, the more accurate the simulation. For example, gravity affects everything on the molecular scale instantaneously. Imagine two particles, one vastly larger than the other, floating in a vacuum with no gravity affecting them. The small particle would start to orbit the other. No matter how small the timestep is, the path of the small particle would be jerky. As the timestep gets smaller, so does the distortion of the orbit. Nevertheless, the computer would have to move and recalculate the particle's movement constantly for the path to be perfectly elliptical.

Another problem that can slow down the computer is the number of molecules. If there are too many molecules, then, for each one, the computer will have to calculate the attraction or repulsion for each molecule in relation to all the other ones. This means that if there is a molecule on one end of a yardstick and another on the opposite end, the computer will try to calculate how they affect each other's movement, even though, at that distance, they will have virtually no affect. A partial solution to this is to find the closest, say, 1000 molecules and only calculate their influence.

Programming Languages

The programming of a computer is what executes programs, starts up and shuts down the computer, and directs everything else a computer does. In early days, when people only had switchboards or the equivalent, the only way to 'program' a 'computer' was to pull wires in and out. After that, they graduated to punch cards. These cards had columns for ones and zeros, and the programmer punched out the appropriate slot. Each letter, character, and number is assigned a series of ones and zeros, thus creating a method of inputting commands to the computer. With the invention of monitors and keyboards, imputing commands is no longer as much of a challenge.

Programming languages are different formats for commands. It's really the difference between whether you say 'roger,' 'acknowledged,' or 'understood.' Different languages are sometimes specific to various tasks. For example, Java script is what pulls out the drop-down menus when you run your mouse over a certain words on the Internet. Some common languages are Visual Basic, Perl, Python, Fortran, C, C++, and LISP. In C, the command for 'add one to' is ++. (This is actually a joke. C is the original language, with no special features. Later on, C++, which is more advanced, came into existence. So C++ actually means C+1. Guess you had to be there.)

Another advancement in computer programming is the compiler. It's a program that helps you program. When there's error, such as a section doesn't return a value or some semicolons are out of whack, the compiler flags the problem. It also translates the program into even more unintelligible commands for the computer to execute (1s and 0s).

Conclusion

Although GROMACS gave a more accurate picture, the cellular automata program was more successful in making a snowflake pattern. The computer was able to simulate the formation of the snowflake, although not very accurately.

Acknowledgements

To Gayle Travis, Lomas-Tramway Library, who knows how to make research fun.

To Ms. Arnold, a great science teacher, who put “Ms. Arnold’s Wacky Science Page” on the web for those of us who can’t do science fair in school.

To my dad, for being patient while teaching me C, Linux commands, and for helping me reload Linux eight times on various computers.

To my mom, for giving me pointers on my report to make it coherent.

Bibliography

Author: Libbrecht, Kenneth George.
Title: The snowflake : winter's secret beauty / text by Kenneth Libbrecht ; photography by Patricia Rasmussen.
Publisher: Stillwater, MN : Voyageur Press, c2003.

Author: Bentley, W. A. (Wilson Alwyn), 1865-1931.
Title: Snow crystals / by W.A. Bentley and W.J. Humphreys.
Publisher: New York : Dover Publications, 1962, c1931.

Author: Williams, Terry Tempest.
Title: The secret language of snow / by Terry Tempest Williams and Ted Major ; illustrations by Jennifer Deyouy.
Publisher: San Francisco : Sierra Club/Pantheon Books, c1984.

Author: Heller, Steve
Title: Learning to Program in C++
Publisher: Prentice Hall PTR c2001

Author: Kernighan, Brian W.
Title: The C programming language / Brian W. Kernighan, Dennis M. Ritchie.
Publisher: Englewood Cliffs, N.J. : Prentice-Hall, c1978.

Author: Angela M. Coxe, Clifford A. Reiter
Title: Fuzzy hexagonal automata and snowflakes
Publisher: Computers & Graphics 27 (2003) 447-454

Web sites:
http://www.bbc.co.uk/weather/features/weatherbasics/snowflakes.shtml
http://www.suite101.com/article.cfm/13646/96746
http://snobear.colorado.edu/Markw/SnowHydro/Atmosphere/atmos.html
http://www.its.caltech.edu/~atomic/snowcrystals/primer/primer.htm
http://starryskies.com/articles/2002/12/physics-snow.html
http://ffden-2.phys.uaf.edu/212_fall2003.youb.dir/Yekaterina_Simonova/3.htm
www.GROMACS.org