3-D NOM Simulation

Plans for a Potential Implementation

Eric Chanowich
University of Notre Dame
Department of Computer Science

Section

1

Background

A description of the NOM simulation and its current state

NOM Research Group

The NOM Research Group was formed to create a realistic stochastic simulator of the interactions of molecules in the top layer of the soil. The group, composed of biologists, geologists, chemists, and computer scientists from around the country, came together with a grant proposal that was accepted by the National Science Foundation.

The project, which started this summer, is expected to take up to 3 years for completion. This summer, we (the computer scientists) created a basic simulation, which the other scientists will then test against real-world experiments for validity. Feedback will be given back to the computer scientists as to what needs improvement in the simulation.

Unlike many scientific/computer collaborations where the computers would in support of some end scientific discovery, the NOM Research Group's goal is to create a realistic simulation that is supported by the work of scientists. It is speculated that the simulator created from this project may eventually be useful for evaluating the impact of such ecological problems as pollution and nuclear waste.

Current NOM Simulation

The current working NOM simulation was created in two dimensions. The layout is a two-dimensional grid of a size specified by the user. Molecule positions are referenced by x/y coordinates and movements are calculated only in the x and y directions.

Adsorption sites exist where molecules may become temporarily (or possibly even permanently) "stuck" to these sites based on both the properties of the molecule and the adsorption site. The adsorption sites are represented in a separate, parallel two-dimensional grid. If a molecule becomes adsorped, it is moved from the main two-dimensional grid to the adsorped two-dimensional grid. If the molecules desorps, it is moved back to the main two-dimensional grid and its movement is once again calculated at each time step.

The simulation was originally constructed in two-dimensions to simplify creation, as it was the first simulation created by the NOM Research Group. Many problems encountered while creating the two dimensional simulation would have been far more complex if the simulation had originally been built in three-dimensions.

Section
2

The Problem

Need, goals, and potential problems for a 3-D NOM simulation

A Three Dimensional NOM Simulation

As the NOM simulation becomes more complex and nears closer to reality, it will become essential to port the current two-dimensional simulation to three-dimensions. Simulating two-dimension is like simulating a flat slice out of the soil, whereas in real life molecules are clearly moving in three-dimensions.

The realism of adsorption will benefit most from moving the simulation to three-dimensions. In the current two-dimensional simulation, molecules have the opportunity to adsorp at every time step because they are constantly over adsorption sites. In the proposed three-dimensional simulation, the molecules will move freely

Swarm and Three Dimensions

Of course, porting the simulation to three-dimensions is not an easy task. Furthermore, it is not made any easier by Swarm, which supports only two-dimensional grids in its APIs. Thus, we arrive at a design dilemma.

Should we use a new technology other than Swarm or should we attempt to forge a three-dimensional environment in Swarm when only two-dimensions are supported. As a significant amount of work has already been done with Swarm technology, the remainder of this paper discusses the feasibility and a potential implementation for forging three-dimensions in Swarm.

Section

3

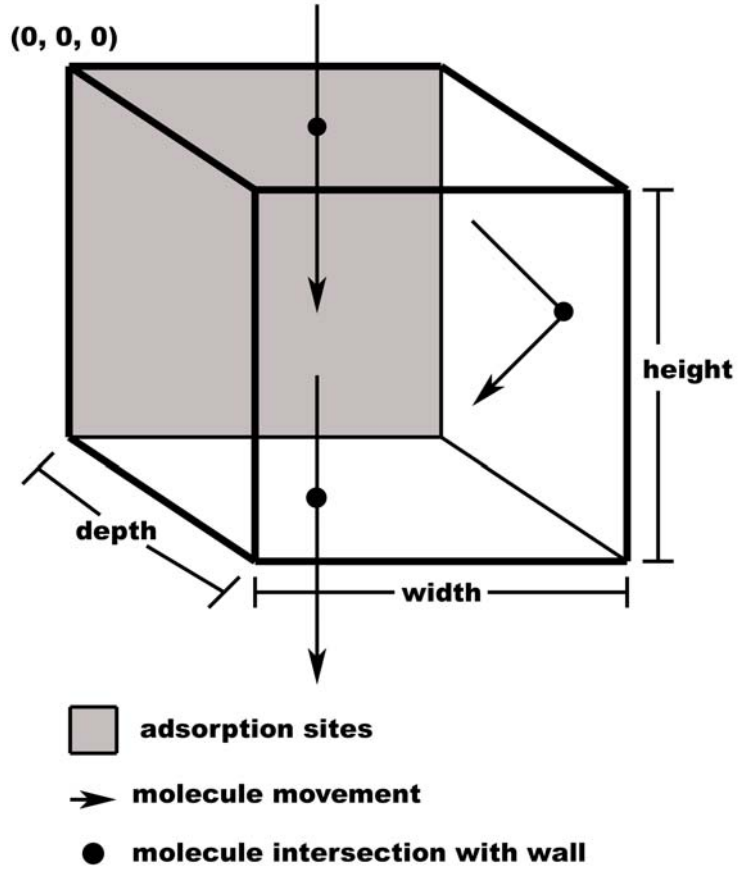
3-D Modeling of the NOM Simulation

A theoretical evaluation of the new representation

General Scheme

Swarm two-dimensional grids are discrete. Each two-dimensional grid has (width x height) unique squares that can only be occupied by one molecule. Likewise, as we attempt to create a three-dimensional simulation, there will be (width x height x depth) unique *cubes* that only one molecules may occupy.

The diagram below illustrates the proposed theoretical layout of the NOM simulation in three-dimensions.



Movement

Molecule movement will be very similar to previous simulations, with the exception that molecules are of course allowed to move in three-dimension. In the two-dimensional simulation, molecules enter the simulation in a designated rectangle at the top of the larger simulation rectangle. Likewise, in the three-dimensional simulation, the molecules will still enter from the top. The will now be places in a three-dimensional box at the top, as designated by the arrow entering the cube in the diagram above.

Molecules will exit through the bottom of the cube, as is also consistent with our current two-dimensional simulation. If they are not considered adsorption sites, the side walls of the cube may be used to reflect molecules back into the simulation or may also be used as another exit from the simulation, though the former is more likely.

Locations will be referenced in (x, y, z) coordinate triples. Height corresponds to the y-axis, width corresponds to x-axis, and depth corresponds to z-axis.

Adsorption

Any of the sides of the cube may be used as adsorption sites. A literal translation from the two-dimensional simulation yields only one adsorption "wall" on the back on the cube, as designated by the grey wall in the diagram.

Adsorption is perhaps much more realistic in this model, as molecules must collide with the adsorption sites to have the opportunity to adsorp. In the previous two-dimensional model, molecules were always located adjacent to an adsorption site, which is unrealistic. This model will clearly allow the NOM Research Group to advance the realism of adsorption in the simulation.

Graphical User Interface

The graphical user interface (GUI) will likely remain exactly the same. The (x, y) coordinates will be projected onto a two-dimensional surface for viewing. As Swarm does not support three-dimensions, its GUI capabilities are also limited to two-dimensions.

To create a three-dimensional GUI would likely mean writing your own APIs in Java to interface between Swarm and Tcl/Tk, the graphics language used to create Swarm GUIs. Obviously, this is a somewhat daunting task, especially for a GUI that is primarily used for debugging anyway.

Section
4

Implementation

Programming specifics

A 3-D Structure

Forging an effective and efficient three-dimensional structure in an environment that supports only two-dimensions is not an easy task. Swarm provides a structure called grid2d that can contain any object type in its grid.

Thus, I propose each grid location contain a gridLocation object. The object will essentially be a container for an array of size *depth* where each array location is unique cube in the three dimensional structure. Picture a sky scraper coming up from the ground. If there are many sky scrapers directly adjacent to each other, the three-dimensional area above the ground will be completely covered until it reach the top of the sky scrapers. In our case, the ground is an (x, y) plane and the skyscrapers extend in the y direction. When molecules need to be referenced, the gridLocation object will provide methods to obtain the molecules, just as the skyscraper provides offices so companies have their own unique space. An array seems an obvious choice because it fully represents the space of the third dimension and its member can be quickly and easily accessed by their y coordinate (the y coordinate of a molecule will correspond to its array index at (x,y)).

Adsorption will be represented as a two-dimensional surface in the plane of the desired wall of the cube. Thus, the only change to the adsorption grid, will be that multiple adsorption grids are required if multiple walls are to act as adsorption sites. The coordinates on the molecule in (x, y, z) will then be translated to (a, x, y) where a is the number of the adsorption wall and (x, y) are the two-dimensional coordinates on adsorption wall a.

Finding Proximity

Proximity knowledge is only necessary when two molecules seek to combine and become one, where the two molecules must be located closely enough to each other that such a reaction is possible. In the current two-dimensional simulation, an algorithm is used that "spirals" outward from the first molecule to find a second molecule. Of course, the same theory could be applied to the three-dimensional simulation, checking the surface of increasingly large cubes, but this seems overly complicated. The best solution is perhaps to randomly pick a plane and spiral out from it as before. This will not necessarily find the closest molecule, but it will either find a molecule that is close enough to react or abort the reaction if the molecule is too far away. If the reaction is aborted, another plane can be searched instead. Of course, it is highly unlikely that multiple planes would ever need to be searched, as the molecules are reasonably dense.

Section

5

Conclusions

General thoughts about need and feasibility of 3-D model

Advantages

The NOM simulation has clearly evolved to the point where a three-dimensional model is necessary to obtain the most realistic results. Originally programmed in two-dimensions for simplicity, it has outgrown its initial model and the need exists to transform this model into three dimensions.

There is only one advantage to making this transformation and that is to increase the realism of the simulation. If the NOM simulation is not constantly refined to make it as realistic as possible is it even worth making?

Disadvantages

The disadvantages of a three-dimensional system seem fairly obvious: decreased performance and greater memory requirements. The biggest decreases in performance will come in the overhead of translating three-dimensional movement coordinates to two-dimensional adsorption sites and corresponding walls, as well finding the closest molecule for a combination reaction.

Other that these two relatively rare occurrences, the only slow-downs will be from: (1) accessing an object in an array element of another object, rather than directly accessing the object itself and (2) calculating an additional semi-random coordinate for movement (the y coordinate). However, this should prove relatively insignificant.

Much greater amounts of memory will be used with a huge array at every location in the 2d grid. Smaller structures could be used, such as a linked list with a y-coordinate attribute, but these lists would not be nearly as efficient to access, as they would have to be fully searched each time to make sure a location is empty, rather than simply checking that location in an array.

As recent memory issues have been nearly (if not completely) eliminated, the added memory strain of a three-dimensional system should not be a problem. Had the previous problems not already been eliminated,, however, the memory could be a complete mess in three-dimensions, illustrating again why the simulation was originally created in two-dimensions.

Feasibility

The three-dimensional model seems completely feasible. Both computational slow-downs and greater memory requirements should not serve as obstacles in creating the most realistic simulation possible. Initially, it seemed as if three-dimensional support in Swarm would be impossible, but a clever design will make the three-dimensional system not only feasible, but also quite efficient.