Percolation of a Liquid Through a Porous Solid

Andrew Kindschuh

Physics Department, the College of Wooster, Wooster, Ohio 44691

February 24, 2005

This experiment utilizes a computer program to simulate the flow of a liquid through a randomly porous material. Three graphs were made, showing the probability of percolation for differently sized blocks of material of different densities. It was predicted that increasing the area of the top of the block would shift the critical density towards higher densities, increasing the height of the block would shift the critical density towards lower densities, and increasing both proportionally would increase the steepness of the slope in the density versus percolation probability plots. These predictions were found to be correct.

INTRODUCTION

In an object populated by randomly placed mass and vacancy there is some probability that a clear path from the top to bottom exists, through which a liquid could flow. This probability will go to 100% at densities such that if all the matter were arrayed in a horizontal wall or sheet, there would still be holes, and the probability will go to 0% at densities such that if all the vacancy was arrayed in a vertical tube, it would be insufficient to join the top with the bottom. For this experiment I have created a number of objects of differing dimensions and densities, and tabulated how these differences affected the probability of percolation. I theorized that a plot of the probabilities (as a %) for a specific shape over all densities could be approximated by the equation:

$$F(x) = \frac{100}{e^{(x-A)/B} + 1}$$

It was also theorized that increasing the area of the object will shift the transition from 100% to 0% to the right (towards higher densities) and that increasing the height of the object will shift the transition to the left. What effect increasing both (effectively increasing the resolution of the simulation) will have is hard to say, possibly it will increase the slope of the transition, or maybe it will provide more possible avenues of travel, shifting the transition to the right.

EXPERIMENT

I began by creating the object through which percolation would occur. This was accomplished by creating a one-dimensional array with a length equal to the volume of the final product, and a subroutine to convert between Cartesian coordinates and locations in this array, effectively coiling the array into a cubic shape. A printing routine was used to trouble shoot the creation and manipulation of the array, but commented out before data collection as too cumbersome when the array became large. Additional print lines were placed throughout the program, allowing me to follow its progress; most of these were deleted, although some were commented out for future reference. A random number generator was acquired from “Numerical recipes in C: the art of scientific computing” by William H.Press[1], although it turned out to be less random than I had hoped. It needed to be fed a seed number, and if given the same seed number multiple times, it would spit out the same seemingly random string of numbers every time. This problem was circumvented by a routine that first checked to see if a seed file accompanied the program, and if so, read it and used that number. If not, it would use the default seed, 12345678, in the program. Another routine at the very end of the program took the final seed and saved it to a file, to be opened upon the next running of the program. As the random number generator only returned finite decimals between 0 and 1, another routine was added to convert these decimals to integer values between a and b.

Finally I was ready to populate the array. Integer numbers between 0 and 99 were asked from the generator, and if they were less than the density then a 1 was placed in the array, otherwise a 0 was. If/else statements were added to insure the user provided positive, non-negative volumes for the array, and positive densities no greater than 100 (zeros were allowed). A routine added water (2's) to any open areas on the top of the object, and another removed water from the bottom.

Initially anther finite array of user-defined volume (x and z coordinates were fixed to those given for the main array, but y, or the height of water in the bowl was user defined) was placed on top of the object, and designed to drain into it, and the removing routine tabulated the water removed. The idea was to determine which densities let how much water through, although the bowl was soon replaced with a single integer, decremented whenever water entered the object. I later realized that the object would trap a specific amount, rather than a percentage, and thus how much got through was much more dependant on the amount placed on top than anything else. The finite reservoir was replaced with an infinite one, and the program designed to terminate when any water made it through.

Data collection was the next step. The simulation was placed within a while loop, designed to run it a user defined number of times, and expel the results and a percentage. 100 arrays were passed through 10,000 iterations, with the testing routine returning either a 1 if water passed through, or a 0 if not. This was passed to an if/else statement that printed and compiled their results. Seven computers in the 101 physics lab, and 3 in the Jr. I.S. lab were used in parallel to collect data from the simulation.

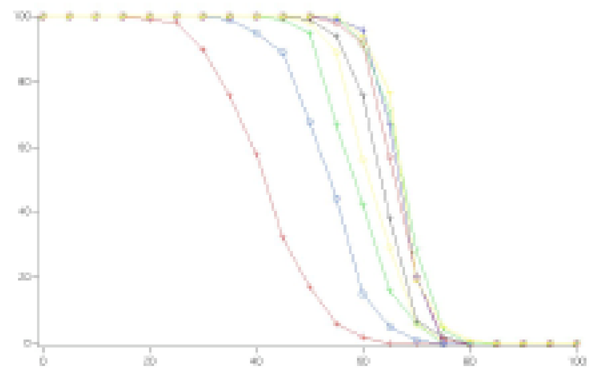

Fig 1: A plot of the probability of percolation versus density for ten objects of height and width 10. The depths of the objects range from 1 to 10, with 1 being the red line (+ symbol) on the far left, and 10 being the yellow line (x symbol) on the far right. This plot clearly shows that increasing either horizontal dimension increases the critical density, as well as showing the increasing steepness found at increasing volumes.

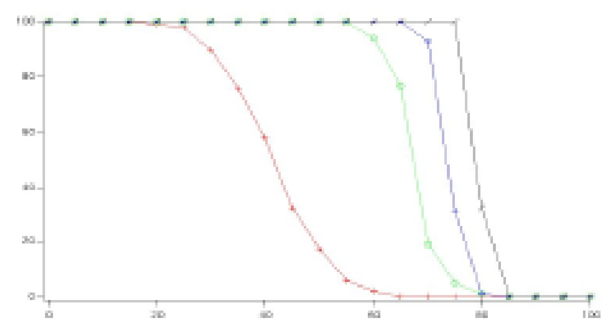

Fig. 2: A plot of the probability of percolation versus density for four objects of height and width 10. As opposed to the last plot, the depths of these objects go by powers of 10, ranging from 1 to 1000, with 1 being the red line (+) on the far left, and 1000 being the black line on the far right. This plot further shows that increasing the horizontal dimensions increases the critical density, as well as more clearly showing the increasing steepness found at increasing volumes.

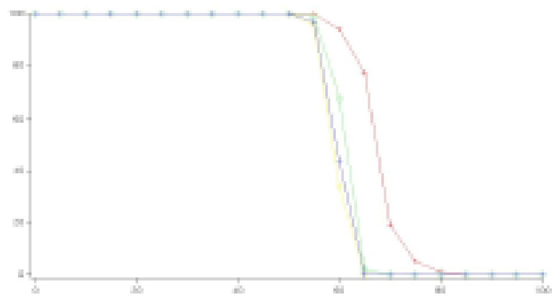

Fig 3: A plot of the probability of percolation versus density for four objects of depth and width 10, and varying heights: Red (+) is 10; green (o) is 40; blue is 70, and yellow is 100. The progression towards lower densities for increasing heights is shown.

ANALYSIS AND INTERPRETATION

Differently proportioned arrays were tested at 5% density intervals, although some trials that would obviously give 0% or 100% were skipped (such as 0% - 25% and 95%, 100%), and the graphs produced came very close to expectations. While I was unable to do certain geometries (such as 100x100x10 or 100x100x100), the geometries the 9 computers at my disposal were capable of clearly showed most of my early predictions to be correct. Graphs 1 and 2 show quite clearly that increasing the area shifts the slope right, Graph 3 shows increasing y shifting the slope left, and all graphs show that increasing the volume steepens the slope; all as predicted. My prediction that increasing the resolution would provide more possible avenues of flow was incorrect, however, probably due to it decreasing the probability that any particular path would permit flow at the same rate as it produces new paths. My guess at a formula for the curve fitting was later found to be correct, albeit with additional constants.

CONCLUSION

13 differing geometries were tested at 5% density intervals, and produced data very similar to that expected, each having a characteristic slope, with specifics defined by their geometries. Additional data could be obtained and allow an increase in the resolution on multiple trials with the same ratio of height to area. My observations on increasing slope lead me to theorize that each geometry will have a critical cut-off point where the slope is infinite, that it will approach as the resolution of the model is increased. It would be interesting to compare the cut off points for a flat object, a cube, and rectangles of various ratios. It would also have been interesting to try other geometries, a cylinder or spiral perhaps.

ACKNOWLEDGMENTS

I would like to thank Dr. Boone for much help with the initial programming, as well as later troubleshooting, Mark Lightfoot for the idea, and Dr. Jacobs for general guidance and help with my report.

1. Numerical recipes in C: the art of scientific computing / William H. Press ... [et al.] Cambridge [Cambridge shire]; New York : Cambridge University Press, 1992 2nd ed