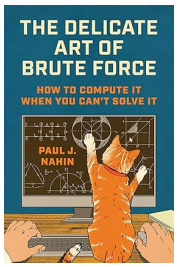
Paul J. Nahin, **The delicate art of brute force**. How to compute if when you can't solve it, Princeton University Press, 2026, isbn: 978-0-691-26746-3 (hbk)



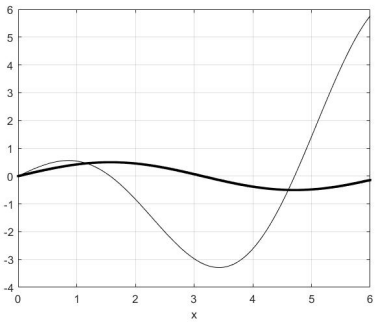


Paul J. Nahin

Paul J. Nahin is an electrical engineer, but one that loves mathematics and he has shown that in many of his books in which he illustrates that his love is in particular because it allows him to solve, with rather elementary tools, some problems that are at first sight rather complex. Hence in many of these books we find explicit computations that should trigger the interest of young students that could become possible scientists, engineers, or mathematicians in later life.

The same acclaim holds for this book. Why would he change at the age of 84? Although the title seems to suggest that the computer can help when mathematics alone cannot always provide the solution. But that is just a gambit, a perspective, needed to present yet another collection of his witty problems. Indeed there are still examples where solutions are computed using formulas, but then a simple matlab program is used to check that solution. Often the code is written in such a way that a very simple modification will solve (or approximate) a much more general problem where the math would be much more involved, or the formulas may be so complicated from the start that a simple computer simulation is preferred. But still in those cases there is some mathematics to verify that the computer produced sensible results.

As long as the computer has to check a finite number N of possibilities, it can check each and every one, but some problems become so complex that even that is impossible, and it certainly holds for problems where N has to approach infinity. Then only approximations are possble, which sometimes rely on statistical tests. But the overall lesson to take home after reading the book is that it is often optimal to use the computational power and speed of the computer in combination with your own mathematical skills for the analysis or the manipulation of the formulas to solve the problem or verify the solution.



Los Alamos equation

The book is a collection with 10 examples in as many chapters of the ideas I explained above. An introduction starts with some history: the birth of electronic and digital computers around the middle of the 20th century with ENIAC, IAS, MANIAC being the first vacuum tube computers. It is illustrated that just a computer plot of the two functions of the equation $x\cos(x) = (1-a)\sin(x)$ immediately shows that there are infinite solutions and an estimate of their location is simple to see. The problem at Los Alamos was to find how the solution evolved with a. The plot on the left is generated using the matlab code from the book. The thin line is the plot of the left-hand side the thick line repesents the right-hand side. By varying the value of a, one can see how the intersections of the curves will change.

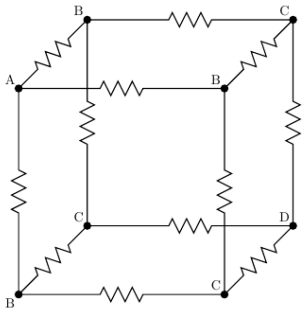
The point being made, there are first 3 chapters that are stochastic (e.g. gamble related) problems that are (approximately) solved by simulation. What is the probability that a chess match between two players of the same strength will end in a tie? The theoretical computation requires Stirling's formula for factorials to find asymptotics, but a computer can simply evaluate these formulas and see what happens when there have been n draws already, or when the players do not have equal strength.

The carpenter's problem is the following: cut 2 beams of the same length each at arbitrary places into 2 parts. Take 3 of the resulting 4 parts, then what is the probability that they can be used to form a triangle (e.g. for a window)? What if you cut one beam into 3 random pieces?

A geometric problem: Assume that you place some Wi-fi antennas in a large room, what percentage of the area in the room will be covered? In 3D this can be placed in the context of firing a torpedo at a randomly moving submarine. What is the probability the submarine will

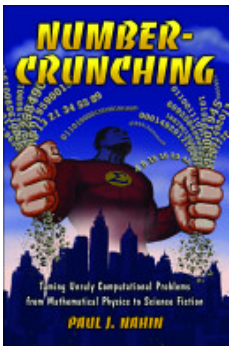
survive when the bomb explodes at the submarine's original position?

The electrical engineer still lingers in the author, so there are a number of possible configurations of circuits represented as regular graphs (hypercube, ladder, lattice, pinwheel, etc. networks) with various resistors at the branches. Problem is to find the voltages at the nodes. Of course this is 1-1 applicable to water supply plumbing or traffic control.



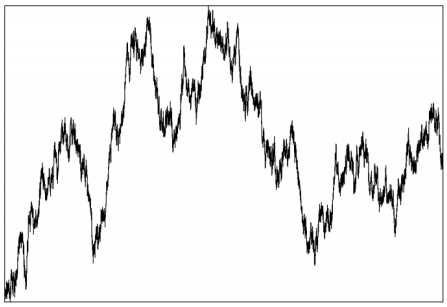
resistor cube

Next is a simple matlab implementation of a (binary) Turing machine and a discussion of the halting problem. Chapter 7 is announced as the toughest problem in the book. It is a classic leapfrog problem and can be formulated as follows: Two friends F and G start at the same time in city C to reach a destination D at a distance d, but they have only one bike. First F starts walking and G takes the bike. After a distance d_1, G leaves the bike and starts walking. When F reaches the bike he cycles for a distance d_2 (overtaking G) then he leaves the bike and starts walking. When G reaches the bike he takes it etc. Problem: Since between C and D one of both is always walking, does this define the time needed for the last one to reach D? This problem was also considered in a previous book by Nahin (*Number crunching* Princeton University Press, 2011) and I do not think that it is a spoiler to say that the answer is no. However, the simulation seemed to give an unexpected bias that was swept under the carpet. So the problem is reconsidered here to remove that bias and the reason is explained and that reason I will not spoil.



Number crunching

The 3 remaining chapters are a bit less 'recreational'. Chapter 8 is dealing with random walks (in n dimensions), and the problem is whether the random walk will escape to infinity or will there always be the possibility that at some point in the future there is a positive chance that the path will return to its starting point. The plot on the left shows the distance from the origin one has moved as a function of the number of steps. The problem is: will this curve ever cross 0 again? Suppose in 1D, the chance to go forward is p and the chance to go back is $q = 1 - p$. The remarkable answer is that escape to infinity is certain whenever at every binary choice during the walk the chance is fifty-fifty ($p = q = 0.5$), but as soon as one deviates a fraction ε from this, one is certain that sooner or later one shall return to the start. This involves some more statistical analysis than what was previously needed.


random walk

Chapter 9 is about Monte Carlo integration which is often used to compute very high dimenional integrals (into the thousands). Nahin uses matlab's random number generator although currently quasi Monte Carlo, (not depending on random number generators) are more common for practical problems.

Finally Chapter 10 is about computing the distance that gamma rays travels through a body. This is simplified to 2D where the body is the upper half of a circle and the rays reach it at random places on the diameter at random angles. Find the distribution of the lengths that a ray travels in the semicircle.

There is a rather extensive appendix with explanations about some of mathematics used in the rest of the book but everything should be accessible for anyone with a background of high school mathematics. All the problems are illustrated and tested with a matlab code. It is not a problem if the reader is not familiar with matlab because the code is very readable and Nahin goes through the steps to explain what is happening. So it is also some introduction to matlab and it is fun to run the codes and experiment on your own when you have matlab available.

In conclusion anyone who has read some of Nahin's previous books and liked it, will certainly be eager to read this one too. The style and the quality is not at all deteriorating with the age of the author.

Adhemar Bultheel