Doing physics with microcomputers

An ordinary personal computer can be used to do large-scale calculations in physics at a great savings in cost and added personal convenience for the researcher.

Per Bak

In the past year or so, the sale of microcomputers has increased from a few thousand to millions a year. Today, you can walk into a toy store and buy a computer for a few hundred dollars. Most micros are probably used for recreation, and until recently they have not been taken very seriously by scientists.

But we are in the middle of a revolution; I believe that in the near future most calculations in physics will be done on these "home computers." The cost will be but a fraction of that for central-processor time at a typical computer center. Moreover, the microcomputer is accessible to physicists 24 hours a day; when a problem arises, physicists can start calculating, thereby avoiding the delay and other annoyances that arise when dealing with funding and computer-center bureaucracies. Physicists will be in full control and will be able to plan their work accordingly. In this article, I plan to demonstrate that today's off-the-shelf micros can perform very demanding physics calculations at speeds not much slower than those of modern full-size computers.

Technically, there is little difference between a microcomputer and a large computer. In both, the brain is the central processing unit, which performs extremely simple manipulations—additions, subtractions and logical operations, for example—with numbers stored in the memory. All information must pass through this bottleneck. In the home computer, the cpu is a simple microprocessor. A large computer has more memory and more peripheral equipment, and so it is well suited for handling huge quantities of data, each piece of which requires a small amount of simple processing (large-scale bookkeeping and so on). But in physics we are typically dealing with a relatively small quantity of data that requires a large amount of processing, so most physicists really do not need the facilities provided by a large computer. There is a significant difference between the computational needs for running a bank or organizing a library and doing calculations in physics! With the microcomputer, the physicist pays only for what he needs: central processing time. Because of the enormous overhead (buildings, personnel and so on), computer centers charge typically \$500 per cpu hour; the cost of doing the same calculation on a microcomputer may be only a few cents! Furthermore, home computers are certainly going to be even more powerful in the near future. In this light, it is very likely that most calculations in physics will be performed by the home computer in the future.

To be specific, let me illustrate the considerations above with a concrete example. (The box on page 27 describes another.) In my own field, solid-state physics, Monte Carlo simulations are among the most demanding types of calculations, requiring days or weeks of cpu time on large computers. Monte Carlo simulations are used typically to find phase diagrams and investigate critical properties near phase transitions. A standard model for studying phase transitions is the Ising model of a ferromagnet. In three dimensions, the model cannot be solved analytically, so one is left with a numerical approach. As I will show, even a "state-of-the-art" Monte Carlo simulation on the three-dimensional Ising model can be performed efficiently on a micro.

Monte Carlo simulation

We consider the simple d-dimensional Ising model with spins $\sigma_i = \pm 1$ arranged on a simple cubic lattice with linear dimension N. The energy of the various states of the model is given by the Hamiltonian

$$H = -\sum_{ij} \sigma_i \sigma_j$$

where the summation is over nearest neighbor pairs of spins. Clearly, the ground state at $T=0$ is one where all the spins are aligned, $\sigma_i = +1$ (or -1). For an infinite lattice there is a transition at a critical temperature T_c from a high-temperature disordered paramagnetic phase to the low-temperature ordered phase where the spins are partly aligned so that there is a net magnetization, $M = \Sigma\langle\sigma_i\rangle/N^d$. The thermal expectation value $\langle\sigma_i\rangle$ is defined as a weighted average over the Boltzmann factor

$$\langle\sigma_i\rangle = \sum_{\text{all states}} \sigma_i e^{-H/T} \Big/ \sum e^{-H/T}$$

Because the summation is over 2^{N^d} states each involving N^d spins, an exact calculation is not possible even for lattices of moderate size. In the Monte Carlo method, a representative sample of states is generated by a sampling technique that weighs states according to their importance. Successive states are generated by moving from site to site and flipping spins with probability

$$p = \begin{cases} e^{-\Delta E/T} & \text{for } \Delta E > 0 \\ 1 & \text{for } \Delta E < 0 \end{cases}$$

where ΔE is the change in energy caused by the spin flip. Estimates of thermodynamic functions such as the order parameter M are obtained by averaging over the states obtained in this way.

To perform this calculation, I used a very inexpensive home computer, the Commodore VIC20, with 20 kilobytes of memory and a cassette recorder to store programs. The computer is con-

Per Bak was a professor at the H. C. Ørsted Institute in Copenhagen, Denmark, and is now a scientist at BNL.

nected to a standard color tv, which is the output medium. (Each byte consists of eight bits). To use the memory in the most efficient way, each spin is stored in just one bit ("multispin coding"[1]). The program uses not more than 2 kbytes of memory, so there are at least 18 kbytes left for storing spins, that is, there is room for $18\,000 \times 8$, or 144 000 spins. The linear dimensions of the lattices used in my calculation are not larger than $N = 32$, giving a total of 32^3, or 33 000 spins for a three-dimensional lattice.

To calculate whether or not a spin should be flipped, the probability p given by the equation above is compared with a random number between 0 and 1. If the random number is less than p the spin is flipped, otherwise not. Usually the random number generation is the most time consuming part of the calculation. In this instance I used a very fast, high-quality number generator constructed[2] by Eric Stoll and Scott Kirkpatrick. The only operation to be performed is a logical "exclusive or" operation between two previously calculated numbers (the R250 method described in reference 2).

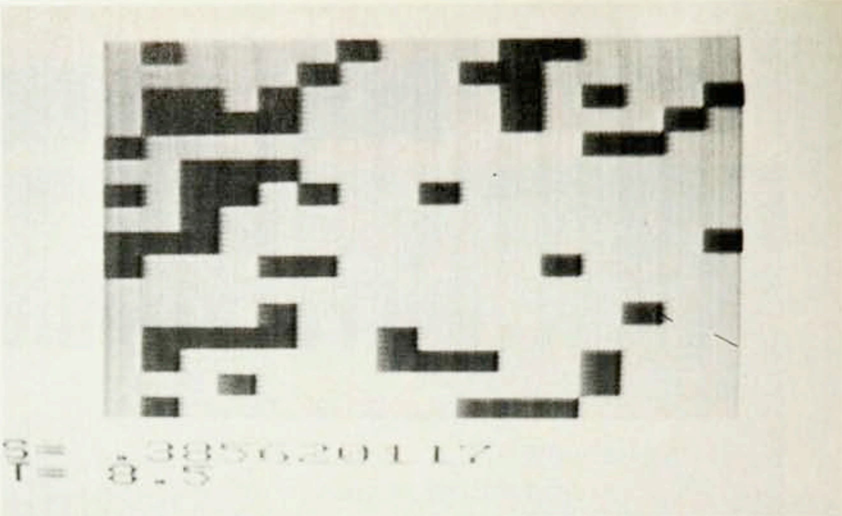
Usually home computers appear to be rather slow. This is in part because they use a BASIC "interpreter," which translates each statement into the machine language consecutively as it is being performed by the computer. If a program line is run 10^8 times, it is translated 10^8 times. A large computer always has a compiler, which translates a program once and for all; the cheap home computers now on the market now have no compiler. However, by using "machine language," one can increase the speed by approximately a factor of 100. A machine language is simply a series of two-digit numbers, each representing a specific order for the cpu microprocessor to carry out. It is really not difficult to to use machine language. I would like to stress that I am by no means an expert in the field of computing: I had no previous experience with machine language. However, it takes only a few hours to get acquainted with the technique.[3] Had I spent about $30 on an assembler program, the task would have been even simpler.

A Monte Carlo program (like all other programs) consists of parts that are performed only a few times and parts that are performed many times and thus require speed. The former includes reading initial parameters, calculating a table with $e^{-\Delta E/T}$ for each temperature for the small number of possible nearest-neighbor configurations, and calculating 250 random numbers to initiate the random number generator R250. The latter includes checking the near-neighbor configurations, calculating random numbers and updating thermodynamic averages, which are all done for every single spin-flip attempt. Because it is much simpler to write and change a BASIC program, I wrote only the latter parts, which require speed, as a machine-language program; these are then called from a slow BASIC program.

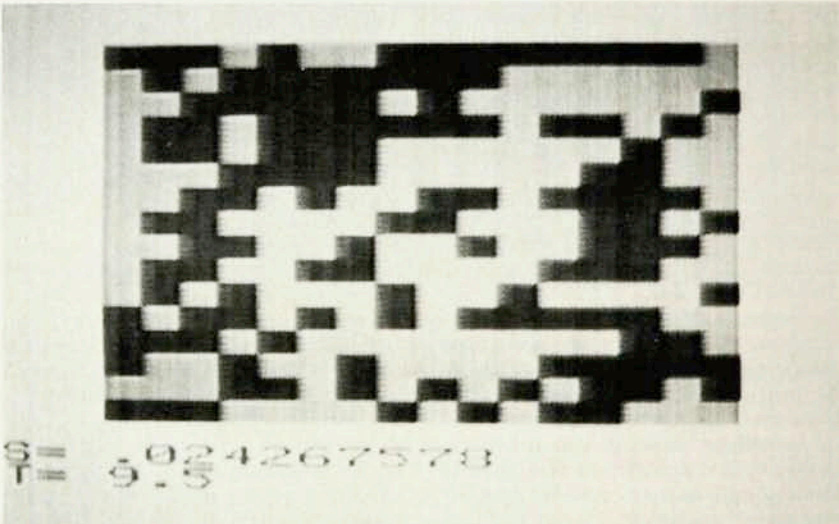
The speed that can be obtained is determined by the clock frequency, which is 1.1 MHz for the Commodore VIC20. The calculation of one Monte Carlo step takes 150–250 clock pulses, which gives a speed of between 4000 and 6000 Monte Carlo steps per second. The clock frequency of a large computer (such as the CDC 7600) is about 35 times higher, and the computer manipulates words that are 4 to 8 times longer. So in principle the larger computer is 100 times faster, but in practice it usually isn't because it uses inefficient FORTRAN or similar programs. A very important factor is that the microcomputer can be run precisely when the need arises. With a larger computer, there may be idle periods of arbitrary length because of conflicts with other users, maintenance and so on, increasing the effective time needed for computations.

Typically, my calculations involved 5000–20 000 Monte Carlo steps per spin for each data point. Because the computer is attached to a standard color tv, color graphics are immediately available. Figure 1 shows photographs taken directly from the TV screen of cross sections of spin configurations on a $16 \times 16 \times 16$ lattice. "Up" spins and "down" spins are represented by two different colors. One can follow visually on the screen the growth of clusters of aligned spins as the transition temperature is approached from above.

Figure 2 shows the resulting graphs



Spin configurations on a $16 \times 16 \times 16$ lattice. Up spins and down spins are represented by two different colors, respectively. The photo above shows the situation below T_c where a majority of spins are pointing in one direction ("green"). The photo below shows a disordered state slighly above T_c, where there are clusters of aligned spins. Figure 1



of order parameter versus temperature for different lattice sizes. The point indicating the $T = 4.5$ magnetization for the $N = 32$ lattice represents 6×10^8 Monte Carlo steps. The transition temperature is identified as the inflection point of the curve. For the lattice with $N = 32$ we find T_c is about 4.50 ± 0.02, which is in fair agreement with the best estimate[4] from a high-temperature series expansion ($T_c = 4.511$). Also, of course, the magnetization curves agree with those calculated using large computers.[5] In addition to the magnetization, the program was also used to calculate the internal energy and the specific heat versus temperature. It is quite straightforward to extend the program to perform more sophisticated analyses of the Monte Carlo results (such as the methods combining Monte Carlo and renormalization-group arguments, developed by Robert Swendsen[6] and Kenneth Wilson, which require the formation of block spins to calculate critical indices), but these are beyond the scope of the present article. The important point is that the above basic Monte Carlo calculation could in fact be done swiftly and accurately.

Economics

The computer was running about a week to generate the data I reported above, and the total number of spin-flip attempts was three billion! The cost of the computer, including the cassette recorder, was about \$400. Assuming that the computer is written off over three years, the cost of one week's computations is \$3 plus maybe \$1 for electricity, so the total cost is around \$4 for 160 hours of cpu time.

What would be the cost of running the same calculation on a large computer (assuming that we do not belong to the lucky few who have unlimited access to one)? Typically, a computer center charges something like \$500 per cpu hour. If we assume that a factor of 40 can be gained in speed, the job would need around 4 hours cpu time at a cost of maybe \$2000. By using the small computer, the cost has thus been reduced by a factor of 500! And this reduction has been achieved without compromising in any way the quality of the calculation.

Admittedly, the three-dimensional Ising model is particularly well-suited for the small computer. Some very ambitious Monte Carlo simulations (on lattice gauge theories, for instance) requiring huge amounts of floating-point calculations cannot yet be carried out in this way. On the other hand, more typical simulations that form the basis of up-to-date research projects[7] may need only a fraction of the processing time of the example presented here, and the considerations above apply.

Turning to problems outside statisti-

Software for the Apple

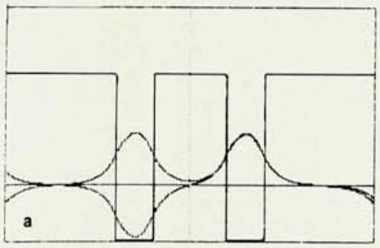
As physicists, we know that a complete description of a physical process requires first formulating the laws that govern the process into a mathematical expression and then solving the mathematical expression. The physical insight used to formulate the mathematical expression is often lost in the purely mathematical task of solving the expression. One such case in point is wave mechanics. Schrödinger's equation can be readily expressed as a time-independent differential equation for wave functions in an arbitrary potential, whose normalizable solutions give the allowed energy values and wave functions for the potential under consideration. Finding these normalizable solutions and their associated energy eigenvalues is, as a formal mathematical task, not trivial, but also not particularly enlightening. However, with the advent of personal computers with speed and graphic capabilities, differential equations such as the time-independent Schrödinger equation can be solved by numerical techniques. The graphical display of a numerical integration can also give new insight into the relationship between the laws and the physical processes the laws govern.

I programmed an Apple computer to integrate Schrödinger's equation numerically for a variety of potentials and to display graphically the evolving trial solution. The project started as an educational tool for an introductory course in quantum mechanics at Williams College. Once completed, it was an easy task to give the program additional potentials that I encountered in teaching advanced courses and in my research. To my surprise, the program could even handle scattering problems. After some fancy additions, including a draw-your-own potential and a set of recallable demonstrations, the program was published under the name "Schrödinger's Equation" by EduTech (624 Commonwealth Ave., Newton Centre, MA 02159).

The program numerically integrates the time-independent Schrödinger equation for a trial energy value and a set of initial conditions supplied by the user. The evolving wave function is displayed graphically, demonstrating visually whether or not the trial energy and initial conditions result in a normalizable solution. The user modifies the trial energy value or initial conditions until the solution is normalizable. This procedure results in a wave function and energy eigenvalue for the selected potential and provides insight into what aspects of the potential and initial conditions cause convergent solutions.

Shown in the figure are **(a)** the two lowest-energy solutions to a double potential well, **(b)** the $n = 20$, $l = 9$ and $n = 10$, $l = 9$ states of hydrogen, **(c)** the $n = 0$ and $n = 21$ levels of a harmonic oscillator and **(d)** a wave scattering off of a potential barrier. I find this method of solving Schrödinger's equation using a personal computer with graphics capabilities much more exciting and rewarding than matching boundary conditions or using the method of Frobenius.

Peter B. Kramer
Cambridge Research Laboratory
Cambridge, Massachusetts



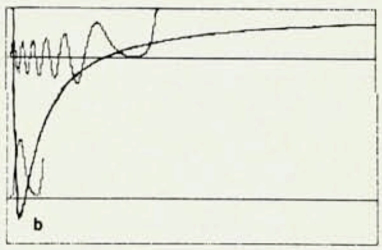
X full scale = 20 Å

Energy Top = 14 eV

Energy Bottom = 0 eV

Energy Value = 3.25 eV

Energy Value = 3.27 eV



X full scale = 1440 Å

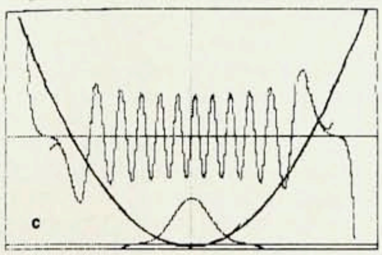
Energy Top = 0 eV

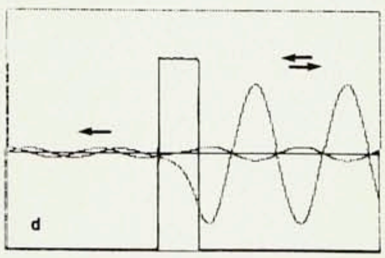
Energy Bottom = − 0.166666667 eV

L value is 9

Energy Value = − 0.1362 eV

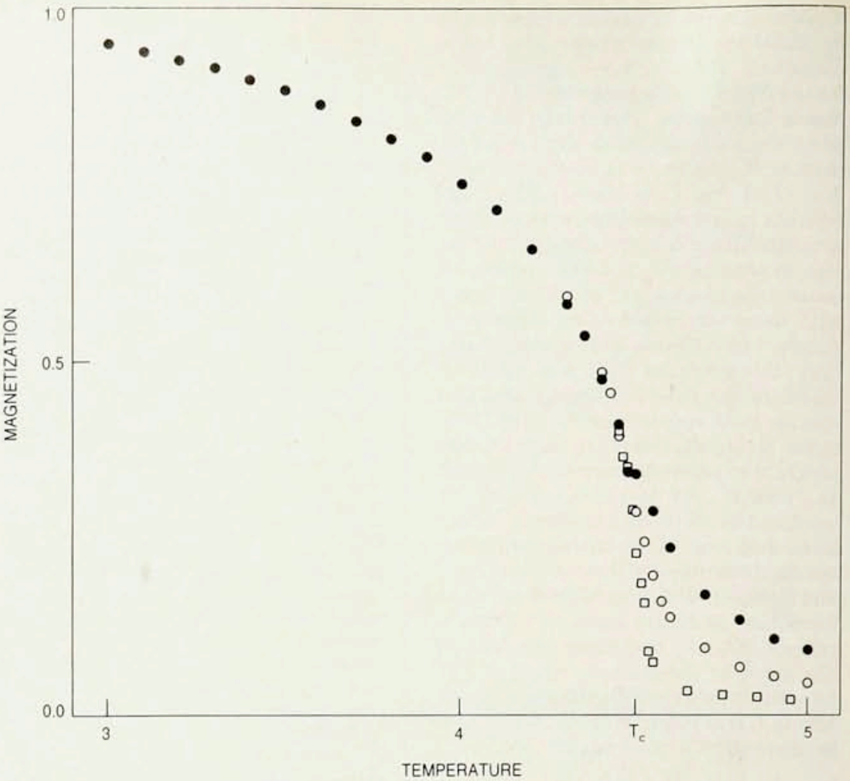
Energy Value = 0.03483 eV





cal mechanics, electronic-structure calculations, as well as many other theoretical efforts to develop accurate models of tiny regions of nature, require all the processing speed that a modern large computer can provide and can be performed only by the most patient physicist on a home computer at this point. Also, the nature of these calculations is often such that the speed mentioned above might increase by a factor of around 5, reducing the cost improvement correspondingly.

Of course there are also problems involving "real-time" processing of large amounts of data (such as running an acclerator) that cannot yet be performed on the home computer. But the bulk of the problems in physics are not of this nature. As a reviewer for the National Science Foundation, I sometimes encounter proposals seeking funding for scientific computations to be performed at university (or other) computer centers. Most of these calculations could be performed on the smallest computers available, at a fraction of the cost. Some physicists, however, have unlimited access to large computers, typically because they run their jobs parasitically while more demanding jobs are being run at their large research labs. Of course there is no advantage for these groups to switch to microcomputers, but they surely are going to get strong competition. Apart from these situations, any calculation that can be done on a small computer should be done on a small computer.



Order parameter versus temperature for the three-dimensional Ising model obtained by author with Commodore computer. Each point represents at least 5000 Monte Carlo sweeps through the lattice. The transition temperature was found to be $T_c = 4.50 \pm 0.02$. Figure 2

The future

Clearly, we are only in the beginning of the microcomputer age. The microcomputer has been around in huge numbers only for a few months. The real revolution lies slightly in the future when we can have home computers powerful enough to save both time and money. Already there are microprocessors (such as the Motorola MC 68000) that can function as cpu's in 32-bit computers with a clock frequency of 12 MHz. In principle, there is absolutely no reason that a small computer should be slower than a large one. In fact, today it is the speed of light and hence the physical size of the computer that limits speed. Thus, the smallest computers are eventually going to be the fastest.

As we have ssen, the main reason that a traditional computer cannot be used efficiently is that all the data must pass through a single bottleneck—the cpu. A large computer is like a dinosaur: a large body with a small brain. Hence the cpu time is extremely expensive because one has to pay a tremendous overhead to finance peripheral units, buildings and personnel. Thus you pay $500 an hour to rent a $5 cpu. However, computers are now being built that allow for multiple processing, so that calculations can be performed simultaneously at different locations. I see no reason that the same construction cannot be implemented in the $300 home computer. An extreme solution would be to buy 100 home computers and run them simultaneously, still saving money.

Another interesting possibility is to build a computer custom-made ("hard wired") for a specific problem. A group of scientists at the University of California, Santa Barbara, has in fact constructed a computer dedicated solely to Monte Carlo simulations on the three-dimensional Ising model (see the article by Jorge E. Hirsch and Douglas J. Scalapino, PHYSICS TODAY, May, page 44.) Information on the content of neighboring cells is fed at several positions on the lattice at the same time. In this way they are able to carry out 25×10^6 steps per second—far beyond the speed of the home computer. However, such an approach appears useful for only a relatively small number of well-defined problems, and much effort is involved in planning and building a computer for every such problem. The whole computer must be written off on a single project, whereas the home computer can easily be programmed to perform other calculations.

In view of the fantastic developments in recent months, it is very difficult to extrapolate into the future. For me, the greatest asset of the home computer is that it makes me truly independent of computer organizations, bureaucracies and economic constraints, giving me time to concentrate on my scientific work. Surely, for most everyday problems in physics, large-scale efficient computer power will be available like water to the physicist in the future!

* * *

I am grateful to several of my colleagues, including O. Mouritsen, R. Swendsen, E. Stoll, D. Scalapino, J. Beyer Nielsen, W. Selke and K. Binder for discussions on many aspects on computations and physics.

References

1. L. Jacobs, C. Rebbi, J. Comp. Phys. **41**, 203 (1981).
2. S. Kirkpatrick and E. Stoll, J. Comp. Phys. **40**, 517 (1981).
3. N. Hampshire, *VIC Revealed*. Computabis Ltd., Yeovil, England (1981).
4. C. Domb, in *Phase Transitions and Critical Phenomena*, C. Domb and M. Green, eds., Academic, New York (1974), Vol. 3.
5. See, for instance, D. P. Landau, Phys. Rev. **B14**, 255 (1976).
6. R. H. Swendsen, Phys. Rev. **B20**, 2080 (1979); H. W. Blöte, R. H. Swendsen, Phys. Rev. **B20**, 2077 (1979).
7. W. Selke, M. E. Fisher, Phys. Rev. **B20**, 257 (1979). □