

**GERBRAND**

**CEDER:**

--recap a little the introduction to Monte Carlo. And then, I want to show you in different types of systems, systems with different types of degrees of freedom, how you would implement a Monte Carlo move. And then towards the end, depending on how far we get, I'll talk a little about error analysis in Monte Carlo and also about things like free energy integration and how to detect phase transitions in Monte Carlo. OK.

I wanted to go through the list of references first. For general statistical mechanics, the books I particularly like are David Chandler's book and Don McQuarrie's book. Those of you who've taken 320 with me, *Statistical Thermodynamics* is exactly the book we use in 320. But if you've never sort of seen the relation between statistical mechanics and thermo, then I would really recommend to Dave Chandler's little green book. It's really sort of an excellent but short tutorial, I think, to modern statistical mechanics.

More specifically on the Monte Carlo, my favorite is the book by Franklin Smith, which I think is now in a second edition. I sort of noticed that people had a book that looked very different. If it's anywhere like the first edition, I thought it was a really good book. I think it's a little deeper than all of the other books I've seen on Monte Carlo. Because of that, it may look a little more abstract on some occasions. And because these guys are chemists, it tends to focus heavily on molecular systems rather than solid state. But I think it's a great book to learn the proper statistical mechanics and sort of the theory of Monte Carlo.

The Newman book is pretty good as well. And then a classic for Monte Carlo books is all the books by Binder. Binder has actually several books-- Kurt Binder-- at least four or five, some thick, some thin. And Binder's expertise is essentially Monte Carlo and lattice models. I mean, he does other stuff as well, but that's really where his expertise lie.

And in particular, long chain molecules, for example if you do polymers on lattice models. And I'll show an example of that. Binder's book is great for that. But he has the standard introduction in there with lattice models with Ising like-- so spin models, how to do the statistical mechanics of those, how to look at transitions. It's a great reference if you're first starting out. OK.

So let me sort of recap how we got to Monte Carlo. Remember that we talked about an ensemble as a collection of the microscopic states that really belong to one macroscopic state. So the macroscopic state is thermodynamically defined. The ensemble is the collection of microscopic states that the system goes through for a given set of thermodynamic boundary conditions.

The probability that your state is in a given system is the exponential minus beta, the Hamiltonian, normalized by the partition function. And so the way you track averages is simply by averaging the property over the microstates with the relevant probability. We distinguish two types of sampling-- and I'll come back to this later to show you intermediates-- simple sampling, where you would randomly pick in the ensemble, and then weigh the property-- let's say that A is the property you're trying to average-- with the correct probability.

And I've shown you why in many cases, in systems with a large number of degrees of freedom, why simple sampling will not work. Because you will tend to pick the states that have high degeneracy-- so the type of states that have high frequency of occurrence, not the states that necessarily determine your average well. And the classic example is if you take the spin model-- if you're in a low temperature state, the dominant states are the ferromagnetic states. Whereas if you randomly picked spin states, you would all pick disordered states. You would barely ever have any states with significant ferromagnetic patches.

So that's why we went to importance sampling. And importance sampling is essentially you try to pick the states in your sample immediately with the correct probability or with a rate that's proportional to the correct probability. So if you pick them with the proper rate, you really just average the property you're getting. Because these already occur in the sample with a frequency proportional to the correct probability. That's the idea of importance sampling.

OK. So I've shown you how a typical Metropolis algorithm with Boltzmann's sampling would work. You would start with some configuration in the system, which could be pretty much anything. You would choose a perturbation of the system-- and I'll do a lot more examples of what these can be. You compute the energy of that perturbation. If the energy goes down, you accept the perturbation. If the energy goes up, you accept it with a probability of the Boltzmann factor-- so the exponential minus beta delta E-- and then you kind of circle around.

It's not totally clear who first developed modern Monte Carlo, but it's often attributed to Metropolis in this famous paper Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller. Her I know these Tellers were husband and wife. I'm not sure about the Rosenbluth, but I also saw-- these were couples doing research together.

And no surprise, the sort of modern version of Monte Carlo essentially started with the advent of computers. Because to do the statistical sampling, essentially what you need was a computer. So the time frame is about the same. I give you one example of a model that I'm going to come back to, the Ising spin model, where you literally have up and down spins on given lattice sites with a simple interaction Hamiltonian. And the reason these models are nice, you can very quickly calculate the excitation energy.

So for example, if you were in a ferromagnetic state-- so a central spin is surrounded by four up spins, so in a ferromagnetic state-- you can easily calculate what the spin flip energy is of these. Here you have four parallel bonds. If you flip this over, you have four antiparallel bonds So the delta E is essentially 8 times the bond energy, assuming this case  $\epsilon_j$  is positive. So an excitation occurs with the rate of exponential minus beta delta E.

If you ever want to play games and try to write really, really fast Monte Carlo code, if you just have a nearest neighbor interaction, you see that the number of different excitation energies you can get is very finite. Because there's really only a small number of possible nearest neighbor environments. And so what you can do is tabulate them. You can tabulate their exponential at a given temperature. And so then you save all the calculations of these exponentials.

And you can write extremely fast Monte Carlo code. This is code where you can easily reach 100 million spin flip attempts per second. So you can do, just for the fun of it, amazing statistics.

You can keep on doing that. It's something to keep in mind, that if you're going to do Monte Carlo where speed becomes an issue, sometimes if you have discrete environments, it can be efficient to tabulate them ahead of time. Because if you can tabulate them, you can tabulate their energy. That means you can tabulate exponential, and there's a big chunk of time you tend to save.

The way we implement the probability is with random numbers. And I want to come back to that issue that I brought up last time, because I may have not been very clear from the comments I got. Normally, random number generation works pretty well, but there's one issue. So remember, the way you implement a probability is that you take a random number, say between 0 and 1, and when that random number is less than the exponential, then you execute. I've really got to calibrate this pen.

But the way random numbers are typically made in a lot of computers is that you don't generate reals, you generate integers, and typically between 0 and some largest integer  $2^n - 1$  depending on the type of random number generator. And then you divide that by  $2^n$  to get the real between 0 and 1. But what that means is that you really have a set of discrete real random numbers, and they start from  $0 - \frac{1}{2^n}$  to  $\frac{2^n - 1}{2^n}$ .

So the important part of this is that on occasion will cause you trouble is that you draw 0 with a certain probability, with a finite probability, which is actually  $\frac{1}{2^n}$  you get 0. And you may think that's not a problem, but the problem is that whenever your random number is 0, you will execute any perturbation you attempt. So if you have a perturbation algorithm that can occasionally pick very high energy perturbations, normally that's a safe thing to do. Because if your random number generation was continuous, that means you would never hit 0, and the probability of hitting 0 is essentially 0. Then that would never be accepted.

OK. But because of the discreteness of the random number generator, you do accept that. And so this is a great way of sort of blowing up your system. I mean, you could imagine, let's say, for some silly reason you could occasionally do perturbation atoms in a liquid, where you put all the atoms on top of each other. So that would be a very high energy perturbation. It would never be accepted. But here it would be accepted at a rate of  $\frac{1}{2^n}$ . And usually, that means that in a sort of regular set of samplings, you tend to do typically more than  $\frac{2^n}{2}$  to the  $n$  passes for a reasonable random number generator. You will actually hit up on that.

I want to come back to what the quantity is, what essentially the Hamiltonian is very briefly. So if you did a lattice model where you can actually change the overall spin-- so you're flipping up spin and down spin-- you would typically have a field term in the Hamiltonian. So you have the interaction term, and then you have a field term. You can think of it as a chemical potential for the spin which try to essentially force the spin to have a given average value. And you would put that full in some almost Legendre transform of the Hamiltonian into the exponential. And I showed you last time how you actually get the quantity that goes into the exponential. For example, if you were to work at constant temperature, pressure, and number of particles, your Hamiltonian, the exponential, would be the energy  $pV$  or minus  $pV$ .

OK. What I will do now is tell you a little about the problem we're going to give you in the lab. I think the lab is still like more than a week away because of scheduling. We had to move it back. But I want to show you, introduce you to, the problem we're going to solve there.

Essentially, what we're going to do is an absorption model for hydrogen on a palladium 1 0 0 surface. And palladium is an FCC metal. So the 1 0 0 surface is essentially a centered square lattice. Remember, it's if you take the 0 0 1 plane of the FCC cube, it's the centered square lattice. Which if you rotate it 45 degrees, it's a simple square lattice.

So that's going to make this at least topologically a very simple model. We're going to have a square lattice, and we're going to look at hydrogen absorption on that. So the question is really, at every lattice site-- so I can't really draw on the lattice sites anymore-- is there a hydrogen there or not? Typically, you'll see that written in what's called a lattice Hamiltonian. And a lattice Hamiltonian has occupation variables which are either 1 or 0. So 1 when the site is occupied by hydrogen, 0 when the site is not occupied.

So only when you have two occupations across a bond length, say  $i$  and  $j$ , will you actually get interaction energy between the two hydrogens. So you can write your Hamiltonian as something like sum over  $i j$  bonds. And  $i j$ , you could have the restriction here, say, that  $i j$  are nearest neighbor. So then you just have a near neighbor interaction  $W_{ij}$ . So that gives you the interaction between the hydrogen.

And then if you do this is an open system, you have a chemical potential-like term, which you think of as the absorption energy for hydrogen but which is essentially the difference between absorption energy and whatever chemical potential you have in your source of hydrogen. But in the end it's just a number times-- this is the total amount of hydrogen on your surface-- some  $\sum_i p_i$ . You can transform this to a lattice model Hamiltonian. I don't know if everybody's familiar with that.

Essentially we like to work with lattice model Hamiltonians, so we do the spin transformation.  $\sigma_i$  is-- how does it go again--  $2 p_i - 1$ . So if  $p_i$  is 1, then  $\sigma_i$  is 1. And if  $p_i$  is 0, then  $\sigma_i$  is minus 1. So you transform the variables.

And if you substitute-- so you invert this, that means that  $p_i$  is  $\frac{1 + \sigma_i}{2}$ . If you substitute that in here, you transform your lattice model Hamiltonian to a spin Hamiltonian. And what you'll see is the interactions, the relation between the interactions  $V_{ij}$  in the spin model. And in the lattice model, the interaction of the spin model is a factor of 4 smaller than the interaction with the lattice model. And the chemical potential in the spin model also relate to the absorption energy in the lattice model.

It's useful to keep these transformations in mind, because this one, a spin model is often mathematically much easier to deal with than a lattice model. But the lattice model, it's often easier to interpret the quantities in a physical sense. And the reason is that here you only have interaction when  $i$  and  $j$  are occupied by hydrogen. So  $W$  is sort of a true hydrogen-hydrogen interaction on the surface.

In a spin model, you have terms here for any occupation of  $i$  and  $j$ . When  $i$  and  $j$  are hydrogen, then you have the interaction energy  $V_{ij}$ . When  $i$  and  $j$  are both vacant, so they're minus 1 minus 1, you also have plus 1. So you also have the interaction  $V_{ij}$ . And then when you just have a hydrogen at either  $i$  or  $j$ , so you have plus 1 and minus 1, then you get minus  $V_{ij}$ .

So it's harder to interpret these constants physically. We always tend to use spin models though, because they're mathematically much more elegant. And the reason is that the variable is symmetric. The variable is symmetric around 0, plus 1 and minus 1, which gives you a lot of useful properties. OK.

I must draw too much. OK, here we go. Like I mentioned already before, to equilibrate the system-- essentially what your assignment is going to be for a Hamiltonian that's a little more complicated than the one I've shown you-- it's actually going to be a first and second neighbor Hamiltonian, and the interactions were calculated on paper from electronic structure-- I want you to essentially get the phase diagram for hydrogen absorption on palladium 100. So that means, as a function of coverage, how much of the sites are current temperature, what the phase diagram looks like.

And again, to equilibrate that, you have to keep in mind that it's much faster to do grand canonical ensemble than canonical ensemble. So an obvious way that you may think of doing this is if you plot the hydrogen coverage versus temperature. You may think of like initializing a system at a given hydrogen coverage and then just sort of heating it and looking at it.

The problem there occurs when you're in two-phase regions. If you're in a canonical system and you're in a two-phase region, your system has to phase separate within your cell. So for example, that means it may make a region with a lot of hydrogen and a region with less hydrogen. And so what happens is that if you're in a two-phase region you need a boundary in your cell, essentially a phase boundary.

And the phase boundary, a boundary carries energy with it. And because the system here is really small, maybe you'll do a 40 by 40 cell, or something, or even 100 by 100 cell-- this is really small. So the boundary has a non-negligible effect on your total energy. Whereas in reality, in real systems, the patches that coexist would be much bigger, and so the boundary would have, proportionately, a much smaller energy contribution.

So if you do canonical and you're in these two-phase regions, your energy will depend somewhat on your system size. Because the system size-- with system size, I mean the size of the Monte Carlo box-- determines the relative proportion of boundary to perfect system. And so you'll get a different energy. So it's much better to do canonical simulations, where you essentially assume some hydrogen chemical potential and use as the Hamiltonian some interactions, some  $i, j$ ,  $V_{ij}$ ,  $\sigma_i \sigma_j$ , so you have an interaction term, and then you just control the amount of hydrogen with a chemical potential term,  $\mu_H$ , some  $i$ ,  $\sigma_i$ .

So how will you see phase boundaries then? Well, essentially at a given temperature you just tweak  $\mu_H$ , the chemical potential. This is just like in an experiment where you would control the gas pressure. And you may start-- if you start with very high  $\mu_H$ -- this actually makes a little more sense, I guess, if I use a minus. So if I start with very low  $\mu_H$ , so very negative  $\mu_H$ , then I want this to be low. So then the spin is going to drift to minus 1.

And depending on what you've defined to be hydrogen, if you define minus 1 to be a vacant site, a plus 1 hydrogen, you're going to drift-- your system is going to tend towards having no hydrogen. And as you crank up your chemical potential, you'll get more and more hydrogen in your system. But the nice thing is that if you have two-phase regions, you'll essentially jump across them.

So let me give you an example. Let's say that this phase diagram simply looked like this. But I'll tell you already, it doesn't. But it's a sort of simple example. Let's say it was just sort of a miscibility gap. So at low temperature you have states-- this is essentially the state here with no hydrogen, and this is full coverage. And so in here you have coexistence between patches of full coverage and no hydrogen.

If you do this by cranking up the chemical potential, you will essentially sort of move from here to here. From no hydrogen, you'll move to this phase boundary, and then you'll jump across to this phase boundary. And you will never live in the two-phase region. So let me actually plot that.

If you were to plot the hydrogen coverage as a function of the chemical potential, you'll essentially go up very small. And then at some critical chemical, potential you'll jump up and then you'll sort of go to 1. And this discontinuity here, that's the width of your two-phase region.

So you never live in a two-phase region when you grand canonical, because you work only with intensive variables in your thermodynamics. So you essentially jump across the first-order transitions, rather than living in the discontinuity that defines them. So it tends to be much more efficient to get phase diagrams.

OK. One of the nice aspects of Monte Carlo simulation is that because it's a simulation method with essentially no approximations in the approach-- there are essentially no approximations in the Metropolis algorithm-- so you can get the answer as accurately as you want. Or as accurately as you can afford is probably a better way to say it. So where do inaccuracies come from when you do Monte Carlo sampling? So we're going to talk about the inaccuracies of the integration of the ensemble averaging. So we're not going to talk about the inaccuracy of the Hamiltonian you start with, that's something you take.

You're going to ensemble average some properties defined by Hamilton. If the Hamiltonian is not accurate, well, we're not going to deal with that. But the question is, how accurate is the sampling? Well, you get sources from two things-- the fact that you take a finite time sample or a finite sample, so you don't really collect the whole ensemble, and you tend to work with finite size. Finite size is a bit of a misnomer. Unfortunately, we all use it.

Because if you use periodic boundary conditions, of course your system is always infinite. It's just that you have imposed a periodicity on it. So the error really comes from imposed periodicity. But but we just tend to call them finite size effects. If you set up a 20 by 20 lattice, you can't get any things in there that have periodicity 35. It won't fit in there.

You can't get any truly random things. In anything with periodic boundary conditions, you don't have truly random things, because you force that that unit repeats. And to get in more and more random, you can make the box bigger so the repeat period becomes longer. But in the end, it's all non-periodic. So you can't truly simulate a random liquid, say.

Notice that you also impose symmetry from your periodic boundary condition. If you take, let's say, in 3D, a cubic cell and you simulate a liquid in there, what's the symmetry of that system? What's the symmetry of a liquid, a real liquid? It has a perfect spherical symmetry actually. It's point group, it has perfect spherical symmetry. Every direction is identical. So it's higher than cubic symmetry actually.

But what do you do? You impose cubic symmetry. Because you essentially have a cubic lattice of a repeat unit that you displace. For a lot of properties, this won't matter at all. But it's just something to keep in mind. So you get errors from your sample time and from your finite or your periodic boundary conditions-- put it like that way.

The sampling error is the easiest one to deal with, because you can quantify it if you really want to work on it. Having said that, let me tell you, most people who run [INAUDIBLE] just run it long enough and hope it's long enough. Rarely ever do you see this quantitative analysis. But I want to show you that you can actually do it if you really wanted to.

It's essentially a very basic statistic. So we're only going to talk about averages so far. Because the errors on quantities that are not available, like heat capacity, and free energy, and so on, entropy, we'll come to later. So I'm only going to talk about quantities that exist in the microscopic state, and the macroscopic value is just the average over the microscopic defined variables-- so things like a volume or an energy or a concentration.

So if you want to the average for some property  $A$ , I'm going to use different terminology here. When I write  $\bar{A}$  with a bar on top, I mean the true average of  $A$  in the ensemble. So the ensemble gives you some distribution of  $A$ 's.  $\bar{A}$  is the true average, and  $A$  between these brackets is the average I actually collect in my finite sample. And the question is, how different are those?

Well, that's basic statistics. You know that  $A$  fluctuates with some spread around the true average. So you have some standard deviation, which is the standard deviation of  $A$  in the distribution. For example, if the property here was the energy we were looking at, that standard deviation is essentially the heat capacity. Heat capacity is a measure of the standard deviation in the ensemble.

So if you take a finite sample in this distribution, you will not get the true average, but you'll get some other average. If you do that a lot of times, you can actually see how the average you sample is distributed. And if the sample you take is uncorrelated, then you know that the standard deviation on the average is essentially the standard deviation on the distribution divided by the square root of  $M$ .

So I've written it here with squares. The square of the average, the standard deviation on the average, goes like the square on the standard deviation of the distribution divide by  $M$ , which  $M$  is the sample size. Is everybody familiar with that? OK. Essentially, if you come you pick averages from a distribution-- this could be some population-- that distribution itself has a standard deviation. Not everybody sits at the mean.

So if you only take say three or four-- let's say you sample with only three or four members-- you're going to get some average for that distribution. But you won't be right. So if you do that many times-- you take many times a sample of four-- you're going to get different values of the average. And if you plot how that average is distributed, that also tends to be a Gaussian.

If your original distribution is a Gaussian, your average is a Gaussian. And you can look at the root mean square distribution at the standard deviation for your average. And it turns out that if your sample is uncorrelated, they relate like this.

Now, why is that useful? Because you actually have all these quantities in your simulation. In your simulation, you actually get some idea about the standard deviation on the distribution, because you take a very large sample. You can average it by essentially the standard deviation in your sample.

So you essentially say that the standard deviation I get in my sample is going to be pretty close to the standard deviation of the distribution. And then if the sample size, you have an idea of the error on your average. So this you get in your simulation, because you can easily-- this is the average of  $A$  squared. This is what you get in your distribution. But you can also keep track of  $A$  squared average.

There is one caveat. So this is true if the samples you take are uncorrelated. So it means that if you really randomly pick from the ensemble. The problem is that in a Markov chain that we set up through a Metropolis algorithm, the samples are not uncorrelated. Because every step in a Markov chain comes from the previous one.

And you can sort of see why that is. Let's say I did Monte Carlo on a liquid and I did really small perturbations. So I just moved the atom within a range of a tenth of an Angstrom or something like that. So that way I will almost always accept the moves. But obviously my states are very correlated. One state in a Markov chain just has one atom displaced 0.1 Angstrom from the other one. So I have a high degree of correlation.

Correlation reduces the quality of your sampling. Actually, the extreme limit is if you do perturbations with very high energy in a Markov chain, they will almost never be accepted. And then your Markov chain just has identical states for a long time. Because if you don't accept the state, you actually keep the old one in the Markov chain. So you have a high degree of correlation.

So the quality of a sample gets polluted by what's called the correlation time. And the correlation time is, again, something you can calculate essentially with this equation. And what does it measure?

What is correlation time, in case you're not familiar with it. If I look at a quantity, let's say as a function of time  $A$   $T$ , and time could be here sample length, the distance in my Markov chain. Let's say I sort of get a bunch of points scattered like this. You're essentially trying to measure if what comes at any time  $T$  that can be predicted from what you had at some  $T$  here, where this is  $t$  minus  $\tau$ . If it can be, then what happens at  $T$  and  $T$  minus  $\tau$  are correlated, and they're correlated over a distance  $\tau$ .

And so that's what you're doing here. What you're doing here is you're looking at whether-- this is the perturbation away from equilibrium. This is how far  $A$  is away from equilibrium at time  $T$ . Let's not call it equilibrium. Let's call it the average-- sorry.

And this is how far  $A$  is away from the average at time  $T$  plus  $\tau$ . And so you're seeing if these are correlated. You're looking at how this fluctuation looks like when you're normalizing by the overall fluctuations in the system.

So the correlation function essentially measures over what time you keep correlation. So for example in a liquid where the system over a long time  $A$  is purely randomly, you'll have of course short correlation time. Probably in a real liquid after a picosecond the system still has a lot of memory from time 0. But after a whole second, maybe it doesn't anymore. So you'll have a finite correlation time. And this is what you'll typically see.

If you plot correlation time for systems with some degree of disorder, it'll start out at 1, because that's the way this function is normalized, and then sort of go down. There are systems where the correlation time is infinite. If you have a perfect harmonic oscillator and you look at, say, the velocity of a harmonic oscillator versus  $T$ , since this is a periodic system, it just sort of oscillates back and forth. And if the period, if the harmonic oscillator's perfect, then once I know what happens at one time, I can exactly predict what happens at another time. So that system has an infinite correlation time.

Essentially, what happens is that the quality of the average gets diluted by the correlation time. If you have correlation in your sample, the standard deviation on the average is what we had before. This is the standard deviation on the uncorrelated sample, but it gets multiplied by essentially a factor that depends on the correlation time. And the correlation time is the integral of the correlation function. So it's sort of the integrated time over which you have correlation.

So these are all things that are relatively easy to calculate. It actually costs you no more computing time to keep track of all these things essentially. This can be done by post-processing, this correlation time. If you output the trajectory  $A_T$ , you can do this simply by post-processing, and it's a fairly fast operation. So if you are interested, you can look at the quality of your averages. Like I said, most people just kind of eyeball it.

I want to show you a few correlation times in this lattice model, spin model. This is actually from your homework assignment, from the lab assignment. So it's actually correlation time in this hydrogen on palladium system. Because I kind of learned a lesson from it myself that I hadn't figured out until I started doing this homework problem.

And I should tell you something up front. Random number generation is expensive in a computer. It takes time, because the sort of fraction algorithm that calculates it does quite a few operations, quite a few multiplications. So especially if your energy calculation is fast-- like if you did this lattice model, this magnetic spin model, 95% of the time would be spent in the random number generator. The rest is all so fast.

So people often try to do shortcuts to reduce the number of random numbers you have to take. Because think about it, you have to take quite a few. Let's say you do a three-dimensional spin model. First of all, you have to randomly pick a spin you're trying to perturb. That takes three random numbers. So you have to randomize three,  $x$ ,  $y$ ,  $z$ .

And then you need a fourth random number to compare with the exponential of minus beta delta  $E$ . So that's four and the numbers per pass. So that's not what we usually do. Here's what we usually do-- we don't randomly pick spins in the lattice, we run through the latter sequentially. Because it kills three random number operations. So you just kind of go. And most of the time that works out fine, except this time.

So here's the correlation time in a lattice model. This is actually the hydrogen on palladium at different temperatures. And the scale down here is the number of Monte Carlo passes. So this is correlation time in units of Monte Carlo time. Which is 1 unit of time is one pass through the whole lattice. So if you have a 10 by 10, it's 100 perturbations. If you have a 100 by 100, it'd be 10,000 perturbations. That's one unit of time.

And to give you an idea, this is a system, at these conditions it undergoes a second-order transition near about somewhere  $T$ 's eight or nine or something like that. So below that you're in an ordered state, above that you're in a disordered state. So let's first look at the low temperature.

So you're in an order state-- you actually have very short correlation time. It goes down very quickly to 0. And then there's this spurious oscillations-- I'll show you in a second where to come from. which is really weird, because you wouldn't you don't expect large correlation times at either low or high temperature. Basically at either low temperature you barely have any fluctuations, and at very high temperature everything is so random that it's uncorrelated. It's at intermediate temperature that you have the longest correlation and in particular near-phase transitions.

You sort of see the same thing a  $T$  equals 0. Correlation time is a little longer. Remember, the correlation time is the integral under this curve. And then you got these fluctuations.

And then at  $T$  equals 8, you're very close to the transition. You see now you have a fairly high degree of fairly long correlation time. I mean, if you sort of eyeball the integration here, we'd almost have a triangle from 1 to 15. So we'd have, what, about 7 and 1/2 under this integral?

So that means that the quality of your averaging here has almost gone down by a factor of 10 from what you would get if you did a random sample. It's actually more. Because, remember, the quality of the average goes down by a factor 1 plus 2 times the correlation time. So if we have 7 and 1/2, we have a factor of 16 there. So the quality of our average is pretty poor near the second-order transition. And then at high temperature, it goes-- you really have a short correlation time.

Now, the reason I keep this bad data up here, because it's a mistake I made myself. The spurious fluctuations, at first I thought they were noise. But they're obviously way too big to be noise. And they don't go away when you take a longer sample. So if they're noise, then I think, well, I got an order of magnitude longer. They don't go away.

Those fluctuations stay, and they can't be real. At  $T$  equals 2, you can't have correlation over 20. And why would you have correlation time over at 20 but not 15?

And so these actually come from sequentially stepping through the lattice. What actually happens is that your lattice starts locking in with the random number generator. So your random number generator has a period to it. And if you sequentially step through the lattice, you're visiting every site at a given periodicity of time. And at some point, that periodicity of time started locking in with the periodicity of the random number generator. Which really what it's saying is that certain spins are not randomly flipped anymore, because they occur at a certain time scale in the random number generator. And all random number generators have a period in them, and that's what causes this.

So you can easily solve it if you just randomly pick spins. Another way you can do it if you want to sort of have it both ways, the things we do now sometimes is that you can just once in a while randomly jump ahead in the random number generator. Or you can randomly jump ahead in the spins you touch in the lattice. Rather than sequentially, you just sort of stop and you go to another one.

OK. Size effects I'm not going to say a lot about, because they're kind of obvious. There's the obvious part of size effect that if your system has some periodicity that doesn't fit in your box, you're kind of messing up your system essentially. You're forcing on a periodicity which it doesn't have. And so if it's a solid state system, often what's going to do is make some kind of phase boundaries to try to fit into it. So that's an obvious one, that if your system has real periodicity your box size should be commensurate with that periodicity. It's kind of the obvious thing.

The other obvious thing-- if your system is really way too small, you can start seeing interaction between the perturbation in your Monte Carlo box and its image coming from the periodic boundary conditions. But usually you've got to have pretty small systems to actually run into that. And then there's a sort of classic problem of near second-order transitions. Near second-order transitions, the fluctuations in a system go towards infinity. The correlation length of the fluctuation becomes infinite, and at the transition it's actually infinite.

So for example, if you do this ferromagnet, it spins all up, and it was a second-order transition, there will be whole patches of spins kind of flipping over. And the spatial correlation over distance over which they do that becomes infinite at the second-order transition. That means there will be a point where your cell is by definition not large enough to capture those fluctuations. And essentially what happens is that if you work with a finite cell, you remove all fluctuations below a certain wavelength. Any wavelength that's larger than your box, those fluctuations are gone essentially from your system.

So that does things to the heat capacity. Remember that the heat capacity is a measure of the fluctuations. It's  $E^2$  average minus  $E$  average squared essentially over  $kT$  squared. So this is the really funny thing. The heat capacity at a second-order transition depends on the system size even when normalized.

So you calculate the heat capacity of your system, you normalize it by the number of particles or spins in your system, you think that's a constant quantity-- it's not. That depends on your system size. And that's what I've shown here.

This is actually not the heat capacity, it's the susceptibility variant of it. So it's the fluctuation of the spin rather than the energy. But look at it. This is for a ferromagnet. This is a small size. This is a 5 by 5 lattice, the green thing.

This blue thing here is a 10 by 10, 20 by 20, 30 by 30, 50 by 50. And so what you see is the bigger your system, the bigger your heat capacity. And I don't know-- if we did it right, it should go like the square root of  $n$  essentially, where  $n$  is the number of particles in your system.

This is a great thing to remember. If you see phase transitions in your system, if you want to know whether their first or second-order, this is the test to do if you absolutely want to know. We'll talk later in more detail about how to see phase transitions. But if you want to be absolutely sure, you'll often see humps in susceptibilities so susceptibilities I mean second-order derivatives of free energy-- source heat capacities magnetic susceptibilities chemical susceptibilities, all those things tend to show peak behavior near phase transitions. But you'll often see peaks in them even when you're not out at a phase transition. And the way to know that it's a second or transition is often just to look at the scaling with system size.

I think I talked about this last time-- keep in mind that if you do Monte Carlo, you can pretty much choose any types of moves. And I talked last time about diffusion-like moves and exchange moves. One thing I wanted to kind of point out-- let me see if I still-- I'll do it here.

Sometimes looking carefully at your system can really help you do moves. I'll give you an example. A while ago I was studying oxygen transport on platinum surfaces. And it's sort of the same. Like in hydrogen on palladium, in this case it's a platinum 1 1 1, So it's a triangular lattice. But you have oxygen sitting on particular lattice sites.

But because of some reason, the oxygen wanted to form strong triplet clusters. So that means on different place in the labs you all had these triplet clusters. So I'm drawing on a lattice. As you can't see it, but anyway.

If your perturbation mechanism is moving one oxygen or flipping one oxygen into a vacant site, the dynamics of these kind of systems becomes really poor. And the reason is-- remember, if you anneal the system, what it's going to do its form these triplet clusters. But then what you'll find in your Monte Carlo is you can't move them anymore. Because these clusters then will have to arrange in some pattern, but you're going to have really poor convergence in your Monte Carlo.

And what's the reason? The only way that you can displace this thing is by breaking it up. Because the perturbation you assign is I take this, and I make it into a vacant site. But now you break up the triplet or I want to start a new one here. But it's the triplet clusters that are very stable, then all these perturbations of breaking one up or starting a new one and going at them are very high in energy.

So what essentially happens is that your Monte Carlo doesn't move anymore. Because any excitation you've given it is very high in energy. Even though the system may have low energy excitations, it may be moving this whole thing over. It sort of keeps the triplet together. And if there's a weak energy interaction with them, that's a low energy excitation.

So sometimes, especially at low temperature, what you should do is look at your system. And especially if there are stable ordered units that form-- for example, those could be molecular units if you're doing an off lattice for molecular simulation-- sometimes it then pays to move those things as a whole. It can be a much more efficient perturbation. It becomes, of course, more complicated to do those kind of moves.

This is by far the biggest problem I see in complicated systems, equilibrating complex systems. They tend to have multiple energy scales. And as soon as you freeze out one of them, it becomes hard to even equilibrate the lower energy scale. OK.

So let me sort of do a very different system now, to give you an idea of how you can do-- what kind of perturbations you can pick. So now, instead of Ising magnets-- Ising magnet is one with two states up and down-- we'll look at a Heisenberg magnet. A Heisenberg magnet is one in which you have a fixed spin, but it can rotate continuously in space. So it can take any direction in space not discretized.

So the spin vector has three components,  $x$ ,  $y$ , and  $z$ . The Hamiltonian is very similar to an Ising Hamiltonian. You can define the interaction energy through the dot product. So again, the dot product of the spins are parallel, the dot product is 1. If they're antiparallel, the dot product is minus 1. If they're orthogonal, it's 0. And then you can have anything in between.

So again, the  $j$  has a very similar meaning. Positive  $j$  will give you ferromagnets, negative  $j$  will give you antiferromagnets. And you can have a field term or a chemical potential term to control the amount of spin. So how would we do Monte Carlo perturbations on this system?

Well, first we need a coordinate system. And rather than  $x$ ,  $y$ ,  $z$ , if we can serve the amount spin, we can just use a set of spherical coordinates,  $\theta$ ,  $\phi$ , and  $R$ . so essentially you need three coordinates to give the direction of a vector. And that's all we need, since the norm, the length of the vector, is conserved.

And this is one of these ones where maybe if you didn't think carefully right away, you'd pick  $\theta$  and  $\phi$  randomly. So remember,  $\theta$  can go anywhere from-- so what's the range of  $\theta$ -- let's say 0 to 180, because the vector can go all the way down. And  $\phi$  has a range of 0 to 360. This one can go all the way around.

Now, if you picked  $\theta$  and  $\phi$  randomly in that interval, you'd make a big mistake. And the reason is, if you think about it, when you're at  $\theta$  equals 0, then all the states for  $\phi$ , from 0 to 360, map onto the same state. So you'd very often end up there. Whereas if  $\theta$  equals 90 degrees, so you're in this plane, then all the states for  $\phi$ , from 0 to 360 are different states.

So essentially, when the density with which you sample  $\theta$  should depend on its value. Let me show that in another slide. For a given  $\theta$ , the number of states that have that angle  $\theta$  is proportional to the slice of the sphere. So it's proportional to  $2\pi \sin \theta$ . This should be a  $d\theta$ .

So it's proportional to the area that you curve out from the sphere. So really what you need to do is you need to pick  $\theta$  proportional to  $\sin \theta d\theta$  or proportional to  $\sin \theta$  really. Because  $\sin \theta$  is 0 and  $\theta$  is 0, so there are very few states here. And there's maximum amount of states around the sphere when you're at  $\theta$  equals 90 degrees.

So how do you do that? You have to go from a homogeneous distribution of random numbers, which I've taken for convenience here as going from minus 1 to 1, rather than from 0 to 1, and you want to go to a random variable now that's distributed with a frequency. You're going to go to a random distribution between 0 and 180 if you're working for an angle. But you want it to be distributed like the sine of  $\theta$ . And I've taken  $1/2 \sin \theta$  so that the integral under that curve is 1.

So what you need to do is find some relation between  $\theta$  and  $x$ , since  $x$  is what you're going to pick from your random number generator and  $\theta$  is what you sample, so that  $\theta$  is distributed like that. And the way you can do that is you can actually write it out. I've written down the solution. But the way you find that is by saying that in a given interval around  $\theta$ , you should have the same status as a given interval around  $x$ .

So that means that  $p(\theta) d\theta$  has to equal  $p(x) dx$ , and this is  $1/2 \sin \theta d\theta$ .  $p(x)$  is  $1/2$ , because it's normalized there as  $1/2 dx$ . So I lose this, I lose this. So  $d\theta dx$  is-- sorry. Cross this out.  $dx d\theta$  is the sine of  $\theta$ . That means that  $x$  is minus the cosine of  $\theta$ .

And that means that  $\theta$  is the arccosine of minus  $x$ . So it's the cosine minus 1. So this is how you can go from a flat homogeneous distribution to some other distribution. So if you pick the angles, with this function you'll actually get a properly distributed value of  $\theta$ . And then, of course,  $\phi$ , the other angle-- what do call that again, I always forget get the names of these-- you can pick randomly from 0 to 360.

Oh, come on. OK. Let me actually, before I show, go to a result. What I've done here is set up a ferromagnetic Heisenberg model and looked at the heat capacity. I'm actually not sure this is ferromagnetic, but it's a Heisenberg model.

But this is the temperature axis, and this is the heat capacity. And I'm going to tell you in a second-- this is the way it's supposed to look, the bottom thing. But this is actually how it looks. At lower temperature get higher heat capacity, and then it kind of bends over. And this piece is wrong.

So I did exactly the algorithm that I told. I implemented the distribution for  $\theta$  correctly, and I sort of randomly picked angles to look for perturbations. And this is what I get. And the first time I saw this, I remember, ah, there must be a phase transition here, because there's a sort of maximum in heat capacity. But there actually isn't. Because I'll show in a second, if you do it right, you get the heat capacity slide on the bottom here.

Keep in mind that heat capacity is sampling fluctuations. So if you do something wrong and you don't allow your system to fluctuate enough, you will reduce its heat capacity artificially. And that's what's going on there. I'm basically not doing a good job about sampling certain fluctuations, and that's why the heat capacity, as I go towards 0, spuriously goes down.

And it's interesting, because these fluctuations that I'm missing have no counterpart in discrete models, like lattice models. In this model, what they actually are are spin waves. And that's the previous slide.

In a lattice model, the lowest energy fluctuation is a finite amount of energy above the ground state. If you think of your ferromagnetic model, what's the lowest energy fluctuation? Flipping one spin over. Remember, I told you that cost you  $8j$ -- that's the energy cost. It's a finite number above the ground state.

In a continuous model, you usually have excitations that are an infinitesimal amount above the ground state. That means these fluctuations are reachable at any temperature. Because if they're an infinitesimal amount above the ground state, you can get them as soon as  $T$  is infinitesimally amount above 0.

And in a Heisenberg model, then equivalent thing are spin waves. We may have thought that we have a ferromagnet, and the lowest order fluctuation is to take one spin and kind of randomly orient it. Well, they're not. The lowest energy fluctuation is a spin wave. It's literally sort of like the wave in a stadium, where everybody moves a little.

It's essentially where you move every spin just a little bit correspondent to the next one. And if you actually calculate the energy of that and you take the long wave length limit, you find that the energy is actually only an infinitesimal amount above the ground state-- so for the infinite wavelength. And as the wavelength gets shorter, the energy of the fluctuation goes up.

So you can see now that we don't get that in our sampling algorithm. And the reason is that we're just never efficient enough to perturb states where you have just these really small amounts of spin canting over a very long wavelength. Because what do we do? Think about it.

If we're in a ferromagnetic state, what are we going to do? With our algorithm, we're going to be picking a spin and taking a random new orientation. And we're never going to accept it. If you're a low enough temperature that's finite energy perturbations, you're not going to accept it. OK.

So how do you fix this problem? Well, if you understand the physics, you start building it in. You start taking other perturbations. And here's a very simple solution to it-- if you have a spin at a given orientation, rather than finding a random new orientation of the spin, you say, I'm going to randomize it just within a small cone. So I'm only allowing a certain amount of deviation from that angle. And as you make that smaller and smaller, you'll start getting spin waves. Because every spin will only be perturbed a very small amount of time. You're essentially just making your sampling algorithm a lot more efficient.

And if you do that, you get this heat capacity. So it's perfect. Notice that the heat capacity is finite at 0, did you expect that? I mean, heat capacity cannot be non-zero at 0 Kelvin. If you have finite heat capacity, so heat capacity that's not 0 at 0 degrees Kelvin, you have infinite entropy. Because remember, the entropy is the integral of  $C_V$  over  $T^2$ .  $S$  is heat capacity at any heat capacity essentially.

So if this goes to 0, this has to go to 0. Because otherwise this integral blows up. And heat capacity always goes to 0. So there's something wrong with the model. And what's wrong with it?

What's wrong with it is kind of what I pointed out, that it has these infinitesimally small excitation energies, because it's a continuum model. In any real system, remember that the states are quantized. That means that the first excitation is always a finite energy above the ground state. And you see, this is important. If you're a finite energy above the ground state, there will always be some temperature that's low enough to freeze out that excitation.

If your first excitation is 1 millielectron-volt above the ground state, you can do the math. When  $kT$  is about 1 millielectron-volt, below that temperature will start freezing out that excitation, and it won't contribute to the heat capacity anymore. And this is how heat capacities go to 0. It's because energy states are in reality quantized.

So any time you work with a continuum model-- let me restate that-- I mean, a model with continuous degrees of freedom-- this was one, the Heisenberg angle-- atoms moving, like a continuous set of coordinates in space, will give you a finite heat capacity at 0 Kelvin. And that's wrong. So how should you solve that. Well, that's a difficult problem.

The way to solve that is you will have to know what the quantized excitations are. And if you had atoms, there your continuous degree of freedom is a displacement. So what's the quantized degree of freedom there? Atoms in a solid.

Remember, if you do this on a solid, just displace the atoms small amounts, you'll get non-zero heat capacity. So how do you quantize the displacement degree of freedom in a solid? Those are called phonons. It's by phonon diagonalization that you find the eigenstates. So those are the quantum states, and they are a finite area above the ground state.

What are they in a Heisenberg magnet? I don't know, and I think nobody knows. Finding the eigenstates of a Heisenberg model seems to be a really difficult problem that to my understanding is not fully resolved. So we don't even know what the eigenstates are of the system if you wanted to do it fully quantized.

In an Ising model, we know. They're actually trivial. The eigenstates are states with just one spin perturbed from say plus 1 to minus 1. So there we're actually getting the right excitations as we go to the ground state, as we go to 0.

In atoms in solids, we know what they are. They're phonons. Those are the eigenstates for the displacements of the system. In a model like this, we don't know what they are.

OK. Other things-- if you go through more complex geometries, the kind of perturbation algorithms in Monte Carlo get successively more complicated and tedious. And I'll give you an example of it. Let's say you do a long chain molecule, a polymer. This is sort of a simplified version of a polymer. It's to kind-- you remove all the degrees of freedom from the side group, so it's essentially a chain because of the fixed angle of a certain carbon-carbon bonds. It can only rotate along these bonds.

So often for a polymer, people don't use all the possible degrees of freedom. They say, well, an  $sp^2$  or an  $sp^3$  carbon bond has a certain angle. So all I can do is rotate around that angle so that I keep that bonding angle. So it essentially restricts the rotation degrees of freedom of every carbon joint along the polymer chain axis.

So what then you have to do is you have to sample space with those degrees of freedom. And I've given here an example. Let's say you want to start rotating this end of the chain from that carbon. So you would essentially keep this fixed and kind of swing the rest of the chain around.

But what you see is that these are going to be very inefficient algorithms. Because if this is a really long chain, if you just do a small  $\omega$ , you're kind of swinging that far end of the chain really far, and it's going to bump into other atoms. So people use these kind of algorithms. But if you read the literature of how to do Monte Carlo on polymers not on lattices or in real space, it's full of all kinds of tricks to find good perturbations that make these algorithms more efficient.

But in the end, it is a difficult problem. Because the problem with a polymer or any long biological molecule is that it's really hard to find excitations that sample all your degrees of freedom but that don't make a large piece of your chain bump into other things. Because, you see, the annoying thing about polymers is that, yes, your atoms can move freely a little bit, but they are all connected. And building in that constraint in the perturbation is not trivial.

Those of us who work with small molecules and atoms have it much easier. They are sort of-- the perturbations tend to be relatively trivial. OK, I got to move on.

Because of this, often people do polymers on a lattice. And you can do that reasonably well. I mean, I'll show you a trivial version of it, but people have much more sophisticated. A really simple way of putting a polymer is by snaking it on a lattice. So this may be every bead. So you say, well, this is sort of a polymer snaking around. And people have even built lattice models that are much denser, where every segment of the polymer-- where there's a lot more lattice points than segments of the polymer. So the density of lattice points-- there are much more lattice points per unit length of the polymer, so that you start to approach a continuum.

So the lattice algorithms are sort of an important class of Monte Carlo algorithms in polymer dynamics. So how would you, let's say, perturb-- let's say this is your polymer. How would you do Monte Carlo perturbations on that? How would you--

First of all, let me tell you an obvious way. You could rebuild that polymer. So what's the length there? 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11. So you could build essentially all self-avoiding random walks of length 11. And that is the ensemble of your polymer of chain length 11.

Now, because it's 11, that's actually quite doable. But, of course, if we go to 50, that's not doable anymore. I get to see all kinds of other things here on the TV. So you can't see that.

So how would you do an algorithm to sort of perturb chains like that? Remember, it doesn't have to be realistic as long as it samples your space. I'm just going to stand here until I get an answer. Look at that. How would you perturb that snake essentially?

Come on. You have to come up with at least one scheme. So there are three or four different types of perturbations you can do on that in sampled space. Remember, the idea is, how can you make this thing move on the lattice and take on different configurations? I think you've pointed-- I'm pretty sure that what you said is right.

**AUDIENCE:** Like, you just cut it and regrow it.

**GERBRAND**

**CEDER:**

Yeah. You can cut it and regrow it. There's definitely a way you do it. So again, rather than sampling the whole random walk of the long polymer, you can just cut it and regrow a part of it. When you do that, you've got to be really careful that you do your whole sampling statistics right for the parts that you keep.

But with these you can actually do simple things. Like here's an obvious one-- you can take the end, this end here, and fluctuate it. Like you could say, take this segment and make it go either here or here. You could slide the snake forward.

So you could say, I'm going to put a vector on this and move this forward one. So that means-- I randomly pick where I put the next segment. So I take this segment. So that means I can either slide it to here or to here, and I slide the whole snake forward. That's essentially the same as-- the way people often describe is, I take the end piece and I put it on the front with a random orientation.

You can do other things. You can do what's called the King jump. See this is whenever you have a sort of square corner, you can kink it over and do this. Again, that's a symmetric perturbation. You can just as easily go from here to here, then from here to here.

In 3D that's called a crank crankshaft jump. Because in 3D, if I can-- let's say you have a polymer like this. You can rotate this piece like a crankshaft. You can do those kind of perturbations. So you can pretty much choose the kind of things that you want to do. OK.

OK, let me stop here. This is kind of a new topic. So I'll pick that up-- today's Tuesday-- so I guess on Thursday then. So I'll you Thursday.