WILLIAM GREEN: So let's get started. This is my last lecture of the class, and I want to thank you guys. This has been a really good class. I really enjoyed the good questions on the forum especially. So I don't know if you guys enjoyed it, but I liked it anyway. So I wanted to give-- my lecture here is going to be a wrap up on the stochastic methods, and then professor Swan's going to give a review on Wednesday. It will be the last lecture of this class. And then the final is a week from today, I think. Morning? Yeah, morning. All right.

All right. So as you did in the homework and we talked about, there is a lot of multidimensional integrals we'd like to be able to evaluate. And a lot of them have this form that is a probability density time some f that we're trying to evaluate. That's the integrand. And then I didn't draw in the millions of integral symbols around it. But usually x is a very high dimensional thing, so you have to integrate a lot of things.

And often we actually don't really know p. We know a waiting factor, w. So I just drew three of the integrals there, but there might be 10 to the 23rd integrals there. So w is a lot easier for us to deal with than anything with all those integral signs. So for example, the Boltzmann distribution is one of these things. And also, just recall back earlier, we were talking about models versus data.

The Bayesian analysis of experiments says that you take your likelihood function, the likelihood that you would have observed the data you did if the parameters were true for certain value, theta. And then you multiply that times the prior knowledge of the parameter values, and that gives you the weighting factor really for any integrals involving parameters, because it's really giving you the w of data given everything you know. Prior information, and you also know the new data you measured. And put them all together, you get the weighting factor for the parameters.

And so you can evaluate all kinds of multidimensional integrals over all the parameters in the problem. So very often you might do experiments where you maybe have four or five or six

adjustable parameters, theta, some of which you know something about ahead of time, and you want to put them together this way. And that should be a summary of everything you know. But it's sort of a goofball summary, because it's a multidimensional function, right, a function with a lot of variables.

But now you know how to do integrals of functions with lots of variables. So you can compute things like some linear combination of-- what's the expectation value of some function of those parameters. So if f, f here was a function of theta-- why is it bouncing? OK. I wonder why you guys are always look like this.

All right. So f of theta, of any function you want, you can evaluate-- it's a function of the parameters, like prediction of what will happen in some new experiment that depends on the parameters. You'd have it as f of theta, and then you could integrate it over w of theta to get the expected value of the result of the new experiment. And also actually the distribution you have to get that way.

Does this make any sense? I see one person thinks it makes sense. OK. So just because we did Monte Carlo using Boltzmann, which is a common one to use, there's other weighting factors. I guess that's my comment. And anytime you end up something that has some integrals with weighting factors, then you should think, oh, I can use Metropolis Monte Carlo. And also, anytime you have experiments, a lot of times you think of the, OK, I want to get the best fit least squares approach, which is what you already knew before I taught you anything.

But it's very important to remember that you can also write the actual probability density, the shape of the probability density the parameters. And that way you can get all the correlations between the parameters and include all the previous knowledge of the parameters and stuff like that. So it's a very important thing to remember from this course is this Bayesian formula.

All right. But Metropolis Monte Carlo, as you saw in the homework, is not always that easy. So you need to choose the step length in every dimension. It's sort of the factor you're going to multiply times random numbers to figure out how far to step from one state to the next. And it's not so obvious how to choose that before you start because you may not know that much about the shape of the integrand, right? Actually, even with the strength of the p in the probability distribution, you might not be that sure about it.

And so if you accidentally choose it too large, what will happen is it tries to take big steps away from a probable state and ends up at really improbable states, and then most of the times it

won't accept that transition, right? Metropolis Monte Carlo will just repeat the same state over and over again. So if you see you just keep sitting there getting the same value over and over and over again, then that's warning you that you must have chose your stepsize too large.

Alternatively, if you choose it too small, all your steps will be accepted because you're not moving anywhere. So you're basically staying at the same point over and over again, except it's slightly different, because you took a little tiny step. But that way you won't necessarily sample the whole range. And you can see that by plotting. They say the distance from the initial state position. You initiated the Monte Carlo Markov chain at some state.

And if you plot probability distribution of density, the distance from all the new states to the original state, and if that number is super teeny tiny, then you might be a little concerned that you really didn't cover the whole space. But again, you have to know what is super teeny tiny, so you have to have a range in mind. Yeah?

AUDIENCE: Can you do this adaptively?

WILLIAM GREEN: Yeah, yeah, yeah. So good algorithms for this try to pick it up that there's a problem, like if it keeps on either not accepting any states, it would try to shrink the step, the delta. Or if it's always excepting the states, then it might try to increase it. You want accept actually most of the transitions because it makes it more efficient. But you don't accept them all. So you want to do some-- I don't know-- keep it to 0.9 probability or something like that, or 0.8. And there is probably a whole journal paper about what's the optimal way to do the adaptive stepsizing to get to convert the rest. Yeah?

AUDIENCE: How did you prevent your initial guess [INAUDIBLE]?

WILLIAM GREEN: Well, to me, I don't think that throwing them out is a good idea. I don't know. People do it. I think it's that your really just need to make enough steps that you are sampling everywhere, and then it shouldn't matter. And a good thing to do is actually start from a completely different initial state and make sure you get the same value of the integral, and that will give you a lot more confidence that you're not sensitive to it. If you're sensitive to it, you're in trouble. Because you really won't know how many to throw away.

And you don't know if you actually even achieved the real integral. I think in the hydrogen peroxide example, actually it was symmetrical, so it was a little bit didn't matter too much. But it I had given you an asymmetrical one, there's like-- here's like the dihedral angle. It has to do

with the orientation of the h, like sort of out of the plane. And it should be symmetrical that you can sort of plus and minus, and there's a high probability region over here and a high probability region over here.

And you start one of them, say, over here, and you just sample around here, you get a lot of points all have basically the same value dihedral angle. And you never see any of these, then it's no good, right? Now this particular case, you might still get the right answer by luck, because they're exactly symmetrical. If this one is a little bit asymmetrical, then you'd get the wrong answer. Does that make sense? So you really want to make sure you're really sampling over all the physically accessible region.

But again, this is a common problem for us all the time, is that when you're doing a calculation, you want to know what the answer is before you start. This is a absolutely critical thing. Because how the heck you going to know if you have a bug? Right? The computer's going to give you some number at the end. You have to know actually what the answer is before you start. I know this seems weird, right? But anyway, it's the truth.

If you're going to calculate anything, you really want to know what the answer is roughly. What the units are, what the order of magnitude is. Some things about it. And ideally you should try to think of some simple test you can do to try to check whether your code's right, whether things are reasonable. The reasonableness test, is it reasonable that what which you get. So this is the same kind of thing. Like you draw the H2O2 and say, oh, I think the dipole moment should be about such and such, because I know the typical charges on an H atom and O atom.

And if you get some number that's way off from that, then obviously you made a mistake somewhere, right? It could be a mistake in your code, or it could just be you just didn't sample correctly because your delta was wrong, for example. All right?

Now I know I've told this to students every time, and every time I tell this to students, they totally reject this idea that you should know the answer before you start. But it's absolutely critical. I mean, my whole job as a professor is I tell my students, please calculate this. But before I tell them that, I know what the answer is, right, roughly. And so that way when they come back and show it to me, I can say, oh, it's reasonable, OK, I believe them. Yeah.

So I think the number should be 20 and they come back with 14.3, I say, 20, 14.3, they're pretty close. OK, so I believe them. And when I think it's 20 and they come back, it's 10 to the

minus 6, then I'm pretty confident, I tell them, I think you must have made a mistake. And they're like, no, no, no, I did a great job. I'm sure my calculus is perfect. And then I'm like, now we're really probing, and I think I'm like-- I don't know-- I have it in for them. But no, actually I knew what the answer is, so obviously it can't be right. So they must have made a mistake.

Anyway, you should be like it, too. You want to be able to be self-critical about what do you expect your answers to be before you do them. All right.

What else is problematic with Monte Carlo? This is very problematic. If you want to achieve high accuracy, you need a really large number of states. And it's funny. It's like you can get a pretty good accuracy with not very many samples because of the behavior of one over square root of n. So it has very big changes, that small values of n. And just a relatively small number of samples, you get something that's halfway reasonable. But then if you have three significant figures and you want to get the fourth one, it's a killer because you have to do 100 times more effort, 100 times more samples to get one more significant figure.

And so you've already done 100,000, and now all the sudden you have to do 10 million samples. And then you only get one more [INAUDIBLE] of that. Now you have to do a billion samples, and it's like, this gets really out of hand. So that's a really unfortunate thing about it.

But the nice behavior at the beginning is you get roughly-- if you only want a few significant figures, you can get them pretty darn fast with Monte Carlo, certainly a lot better than trying to do the trapezoid rule in multidimensions or something like that. All right.

So you guys just did Monte Carlo. Are there things I should tell you about or we should talk about, problems you ran into? Now you see, when I set that homework problem up for you, I helped you out a lot. I don't know if you noticed that. So the original problem had, it's four atoms, they each have three xyz positions. So there's 12 coordinates. And then I did clever tricks to get it down to, I think, six, maybe. OK. So going from six dimensions to 12 dimensions, that's actually a pretty big deal.

And so if I have given you the original 12-dimensional problem, you could still compute it and actually use the same kind of code as you wrote. But the number samples you'd need to get a good number gets really wildly different. Right, and so again, this has to do with knowing the answer ahead of time. You know, I knew, OK, that the magnitude of the dipole moment doesn't depend on the orientation of the molecule, so therefore, I'm going to get rid of all the rotational degrees of freedom.

I know it doesn't depend on the transitional position of the molecule, so therefore, I get rid of all those. And so before I even did any calculation, I can tell you I can get rid of six degrees of freedom, so that'll help you a lot. But it's a similar kind of thing. You want to do that all the time yourself, is try to think what can I do that's easier than just doing the brute force problem? Yeah?

AUDIENCE: So on that problem, changing the max stepsize, you could change the percentage of step-- or [INAUDIBLE].

WILLIAM GREEN: Yeah.

AUDIENCE: And I think the mean state roughly around the same, but the shape of distribution actually changed a little bit.

WILLIAM GREEN: OK.

AUDIENCE: So how is--

WILLIAM GREEN: So I think it might be this kind of thing is one part of it. It's like if you only integrate one lobe of the dihedral, you actually get a dipole moment value that's pretty reasonable, even if you totally missed this other lobe. But if you did a different stepsize, you might start getting some samples over here, too. So the distribution would look a lot different, but you still end up getting roughly the mean.

But also it's partly that the-- the good thing about the Monte Carlo method is that, the first few samples, no matter what they are, give you something that's on the order of magnitude of what the value of the thing is. So even a really lousy sampling at the beginning still gives you something that's halfway reasonable for the average.

AUDIENCE: So if you're trying to recreate the overall distribution histogram, then how do you know which max stepsize to choose, because [INAUDIBLE].

WILLIAM GREEN: Yeah, so the rule of thumb I've heard is that people try to get an acceptance ratio between 0.2 and 0.8. So it means you want-- Yeah. But I really don't know. I'm not an expert in this field. But I'm sure you read papers and people have big discussions about what the acceptance ratio should be in order to taking big enough steps that you have a chance to get the weird stuff. In this kind of one, it might be a problem if you're doing steps, say, in the dihedral, you

may have to step quite a long distance to find the other lobe. So sometimes having some pre-knowledge of what you think the shape of the things are is a big help.

Yeah? No? Sorry, maybe I didn't answer your question. Is that--

AUDIENCE: Yeah.

WILLIAM GREEN: Yeah. All right. All right, so that's Monte Carlo. To Then we talked about the more difficult problem of where the probability distribution is not stationary, and in fact, we don't even have the w written out. All we have is a differential equation that divides the probability distribution. And we did it here for the case of discrete states, and that's the right one to use, for example, for the kinetics equation, if you want to get them exactly right.

Now that differential equation looks really easy, right? Just a the times p. It's a linear differential equation system. You knew how to do that one probably in your second semester of undergraduate taking differential equations class, right? They did dialyze the matrix. Yeah? Remember this?

So that looks really simple. But the problem is that the number of states is so big. So have any of you tried problem two on the optional homework? You haven't tried? Has anyone even looked at it? I know somebody looked at it because I got some question about it. All right. At least one person looked at it. Let me tell you guys briefly what this problem is. It's a really relevant problem. If you go work for Professor Jensen, Professor Braatz, maybe Professor Roman, you might ed up doing this calculation. This is not a crazy calculation. It's very simple.

All they have is like a chess board of sites on a catalytic surface. And they some number of sites. And this is what-- if you take a crystalline catalyst and cleave it, you'll have a whole repetitive pattern on the surface, and somewhere along there is some active site that does something. You can tell that experimentally because you stick that catalyst in the presence of some gases, you make products that you didn't make when it wasn't there. So it did something.

So the way people model this is they say, OK, there's some active sites on here, and there's probably one of them per unit cell maybe. And the site can either be empty, like if I ultra high vacuum, I pump on it and heat it, should be nothing there. And then if I expose it to a little bit of some gas A, some of the sites might have A molecules sticking on them now. And I also exposed to some sites gas B. Maybe one of these sites might have a B molecule on it, too.

And if A and B can react and make my product, C, then maybe they're sitting next to each other and they might react with each other. OK, so this is the whole way you look at this problem. And so all I want to know is, if I have some A's and B's sitting on the surface, sort of how often are they going to react? And the problem is a little bit more complicated. It says, well, suppose I also know if I put a whole lot of B on the surface, what I end up seeing is coke. My whole surface gets totally coked up.

So let's model that by saying, well, if suppose two B's are next to each other, then some probably reacting to turn to some coke product that sticks there permanently and just poisons the catalyst. OK? And this is real life, too, right? So if you're trying to run a catalytic process, a lot of times, you have to run with one reagent in great excess to keep the surface kind of clean, keep it covered by the unharmful reactant.

And then you let little bits of the other reactant come down and react with the A really quickly. But you don't let too much of the second reactant come down, because it might cause a problem like dimerize or coke or something. OK, so that's the model. Some of these guys have A's, some of them have B's. And in the unlikely case that two of these guys react, they both turn around and then turn them into S's. Coke. Soot. And then that part's dead forever after that. OK, so that's the model.

So in the case that they have, they had 100 sites. I think it was 10 by 10 if I remember correctly. So it's 100 sites, and each site can either be empty, or have an A molecule, or have a B molecule, or have a coke molecule, S. So there's four different states on each of 100 different sites. So how many states are there altogether? Its' four to the what? 100.

OK. So that's a pretty big number. So that's how many states. It's about 10 to the 60th, I think. All right. So this is a very large number. This is bigger than Avogadro's number. This is really a lot of states. You wouldn't have thought that just a stupid little problem with just a 10 by 10 piece of catalyst, and three different things that can stick there, would give you so many numbers, but it does.

So now I have a problem that if I have-- my p vector is now a probability that has a number for each of the 10 to the 60th possible states. And then the matrix m is that number squared, right? Dimension. So it's 10 to the 120th power elements inside the matrix m. So even though the matrix m is very sparse, this might still be a problem, because 10 to the 120th is really big, OK? And your computer can only hold 10 to the 9th or 10 to the 10th numbers in it, so this is

going to be a problem.

And also that's not just what p is. That's what p is at this instant, say, 1 millisecond after t naught. And then if I wait to 1.1 milliseconds after t naught, p will be different. So I have 10 to the 60th numbers that change with time in some way that I don't know. OK? So this is actually a really hard problem to solve it.

And so everybody uses Kinetic Monte Carlo to do it, because there's no way I can possibly even list all the elements of p. And in fact, if I sample, if there's 10 to 60th states, no matter how good my sampling algorithm is, I'm never going to sample 10 to the 60th states. So there's a lot of states I'm never going to get-- not even get one sample of. I'm just never going to encounter them at all, because there's just too many states.

But anyway, people do it anyway. So we go ahead and we'll do the Gillespie algorithm, which you guys-- we talked about in class. Now to compute a Gillespie trajectory, we have to compute two random numbers, right? We compute one random number, it tells us how long we wait until the next thing happens. And then we compute a second random number that tells us which of the many possible things that happened actually happened.

And so if I have a case like this where I had the-- to the original case. What can happen, the A could react with the B, the B could react to the B. The A could come back off. The A can move to the next adjacent site. The B can move to the next adjacent site. A lot of things can happen, right? So quite a variety of things can happen from this initial state. And so you have to, if you're solving this problem, you have to write that down. Yeah, John Paul?

AUDIENCE: So the special waiting times are [INAUDIBLE] process where we haven't decided there's going to be an arrival, [INAUDIBLE] arrival times by that [INAUDIBLE].

WILLIAM GREEN: Yeah.

AUDIENCE: And then we make something happen. But I mean I feel like you could run the exact same simulation without counting the times. Get the same answer and then after the fact come and [INAUDIBLE] into the system. [INAUDIBLE] doesn't seem to be [INAUDIBLE]

WILLIAM GREEN: Yes, that's right. Yeah, that's true. So you only do the time calculation because you care about the time. If you don't care about the time, you only care about the steady-state solution, then you can probably do it some other way.

AUDIENCE: You don't need to calculate the times inside that. You could calculate the entire state space trajectory without any reference to the time, yeah?

WILLIAM GREEN: Yes. Yeah, so you can get the sequence of all things that happened if it didn't give you the time, because it doesn't matter, right? For the time. It just means the sequence of what happened.

AUDIENCE: Yeah.

WILLIAM GREEN: The sequence of states is all that matters. But if you want the kinetic information, then you also want the time, because it might be it sat in one state for a million years, and the other state--

AUDIENCE: [INAUDIBLE]

WILLIAM GREEN: Yeah. Yeah, that's right. Let's go ahead. OK, so maybe make this cheaper. Make it one random number for j. No? OK.

All right. So things to just keep in mind about this. So the cost here is to compute in random numbers. We have to compute at least-- well, some number of random numbers. I don't know how many. And that number depends on the length of time I want to simulate and what my delta t is. Where the delta t is sort of like the average time between the events happening. It's like one over a in the [INAUDIBLE].

And so if that delta t is very small, and my time I want to simulate is very large, that means that each trajectory is going to have a lot of events. And that means I'm going to have to generate a lot of random numbers. And so generate one trajectory is going to be really expensive. On the other hand, if the delta t were about the same size as the total time I'm simulating, then I might only get one or two events, and so I only have to generate four random numbers for each sample. All right?

And I just have to keep in mind that I'm going to have a lot of low probability states that I'm not going to sample at all. I might even have some high probability states I don't sample, because I just can't do enough samples. And I had the same exact problem I have with the Metropolis Monte Carlo and with all these stochastic methods that the scaling is one over square root of n. So I can get a rough idea pretty easily, but I don't get a really precise number of anything, then I'm going to have a problem because I'm going to have trouble to do enough samples to really refine the number.

The key thing in this is that with anything like this where I'm going to have so many states, I'm going to run a lot of trajectories to get at all-- sample all the things that can happen. And so I'm probably not going to be able to store all the trajectories I ran on my computer, because I'm going to run so many. And so I really want to compute things on the fly, which means I should decide ahead of time all the averages, all the f's I'm trying to compute.

So I might only compute f and f squared. And I don't know, whatever else I want to compute. The average value of the number of A's on the surface. I mean, there might be a lot of diagnostics I can compute to just check everything's OK. Think of that beforehand, coded to compute them as it runs, and then we only have to store the running averages of all those quantities, and I don't have to store anything about exactly which state sequence I hit. Is that OK? All right.

Now if you remember, in this equation, there was an initial condition of the right-hand side, which we totally ignored so far. But that's actually pretty important. And in a lot of real cases, from your macroscopic continuum view of things, you have an idea of the average value events. And what you don't know is about the discreteness and you don't know about the correlations. All you know is sort of an average. I expect to have two A's and three B's in the surface or something from the partial pressure of A and B and the binding constants of A and B. So I have some idea. So I think I know these averages.

And so what people usually do is they use a Poisson distribution for the probability of the exact number of A. So I suppose I think on average I'll have two A's. , Well, I might have one A, I might have three A's on the surface. And so I'll just use the Poisson distribution of n to get an estimate of sort of the expected width of distributions. And so that's the formula for the Poisson distribution. You have to know the average values of Na, Nb, Nc.

And then you can see that the N appears in the-- on average, N appears in the exponent and in the factorial [INAUDIBLE]. Right? And so really every time you start a new trajectory in Kinetic Monte Carlo, what you should do is sample from the Poisson distribution to get a new initial condition as well. OK? And that way you're sampling over both all the possible initial conditions and all the possible things that can happen to them to really get the real average of what you might want.

All right. Now we just realized how horrible this problem is because it has so many states. And so then we're going to have to think immediately, what can I do to make this faster and easier?

And so here are some things people do. One thing is that people try to figure out what are the really fast processes, and do I care about them. And some of the fast processes, you might say, well, like diffusion of A moving to here, and then it moves to here, and then it moves back to here, then it moves back to here.

That's of no interest to me. I don't care where the A is bound on the surface. All right? The only way I care about it is whether the reacts of the B when it's sitting next to it. In other respects, I really don't care about the diffusion. So having the time concept for the diffusion be the real time concept might not be that important.

So then you can do different things. One thing is you can assume it's infinitely fast, and you say, well, every time I look at a site, I assume that this A has a 1/10 chance of being here or here or here or here, all these different spots, as if it's equilibrated around all the empty sites. That would be one possible way to do it. That's the infinitely fast diffusion idea. Another idea is to say, well, let's slow down the fusion to make it slower just to help out our computation.

So I say, well, on average, I might get one reaction a millisecond. In reality, the diffusion time is a nanosecond. But I don't care about all that nanosecond stuff. Let's pretend that it's a tenth of a millisecond. So it'll still be pretty much equilibrated on the time scale on the reactions, but that way I'll be able to accelerate my calculation by seven orders of magnitude by going from a nanosecond time scale to a tenth of a millisecond time scale.

So there's a lot of different tricks like that. If you read a paper that does Kinetic Monte Carlo, you've got to read exactly what they did. But they usually do something like this, because it's just totally out of hand if you try to model everything perfectly. Also, the low probability events, the really low probability events, you're never going to sample.

So if I have a process that happens on my catalyst that takes 10 hours to happen, and my main reaction happens on a millisecond time scale, I'm never going to be able to run out to 10 hours. So I might as well just forget it. So if I know experimentally that the coking thing only happens on 10-hour time scale, I'd just take it out of there. And I don't have to clutter up my calculation with all these S's. I'm not going to form any of them in my time scale anyway.

And so therefore, I cut the number of states. Instead of being 4 to the 100th, now it's 3 to the 100th because I got rid of all the coke. And so now I've drastically reduced the size of my problem. Not often you can do a reduction like this. That's a pretty big reduction in the size of

a problem. All right?

And then you have to have an idea of what adequate sampling is. So you're going to see some lower probability processes compared to some higher probability ones. And you have to know, when are these low ones so small that they're statistical noise. And this is the margin of error problem. Do you ever see the polling, like in the political polls, they always say, so-and-so many people believe in evolution plus or minus something, right?

Well, the way they get the plus or minus is from the square root of the number of samples with a positive result. So the margin of error on low probability things is much larger. So the number of people who believe that Professor Greene is God is a very small number. So if you find somebody like that, you have to figure they're within the margin of error, right? OK?

But the number of people who are going to attend 1034 this morning is a big enough number that the margin of error in that is maybe two or three people. It's not going to be 100 people. That make sense? All right. So the more likely the event is, the more likely you'll get a lot of samples of it if you count the errors roughly the square root of the number of samples you got of that event, then the statistical error in your sample of the high probability events won't be that large. Whereas for the low probability events, it could be really gigantic.

All right, so just for an example. In the main reaction, A plus B is really fast because it's a good catalyst. So I'm going to get a lot of trajectories going to show that reaction. So that's good. And I should get good samples [INAUDIBLE] But the coking reaction is really slow. At least I hope it's slow, otherwise the catalyst is no good. And so if it's really slow, I might not get very many of those guys.

And so even if I left it in the calculation, I might not be able to reach any conclusion because I might only see one when coking even out of all of the 100,000 trajectories I run. And so then I won't know whether to say anything about that or not.

And then if I let the diffusion in and it's too fast, then my delta t is going to be too large, which means that my CPU time to compute a single trajectory is going to be too large, which means that I won't be able to get good sampling because I won't be able to run very many trajectories. So I might want to do something to get rid of that.

All right. Now there's another method that a lot of people use. I think actually Professor Swan uses this sometimes. Is that correct? Yes. So this is another method. And what it is, is you

solve the equations of motion of the atoms or clumps of atoms typically using Newton's equations of motion. And typically people use it using force fields that were fitted to some experimental data, and maybe with some quantum chemistry data as well, to get some idea of the forces between the atoms and the force with which the molecules, when they bump into each other, how they interact.

And typically, it's done classically using Newton's equations of motions. But if you don't like that, you can put in the quantum mechanical effects in a couple of different ways, and my groups, worked them one way called RPMD. And you can get pretty good agreement with the quantum chemical results by doing this fancier version of molecular dynamics. So there's some different equations you solve, but basically the same. And so you can do it.

And there's a nice algorithm called the velocity of Verlet algorithm that almost everybody uses in this field. And what's nice about that one is that it can do a lot of steps. A lot of steps of moving the atoms around. And after you do a million steps like that, you compute the energy by adding up all the potential energies and kinetic energies of all the atoms. And it'll still be about the same energy as you started from.

Whereas a lot of methods, if you do the integral like that and you calculate the energy at the end, it won't be the same as energy you calculated because the methods have little round-off errors, and they kind of accumulate in a way that messes up the energy. And so this particular algorithm is a nice O to E solver method that is good for-- has a property that is good for conserving the energy.

And then a lot of people use a thermostat, because they care about samples that are in contact with a thermal bath. So you're sampling a few molecules, maybe 100 molecules or 1,000 molecules, but not 10 to the 23rd molecules. So you have your 100 molecules or your 1,000 molecules, and you pretend that they're in contact with some bath, and you're watching those 100 molecules wiggle around, and their energy is not exactly conserved because they're exchanging energy with a thermal bath.

And so there's different things called thermostats which are computer ways of adjusting the velocities of the atoms periodically as if they got a kick from the thermal bath that makes them go up or down. And that's very important trying to do, say, chemical kinetics, because their reactions are so slow that most of the time you'll watch the atoms wiggling around, nothing happens. You need the unusual case when you get a big kick somewhere that gives you

enough energy that you overcome a barrier to make a reaction happen.

All right. So that's what molecular dynamics is. And numerous people in the world and in this department and on the campus do these kind of calculations. And there's two ways that they're used. One way is used as an alternative to Metropolis Monte Carlo. So you're trying to compute basically multidimensional integrals, basically it's from statistical mechanics integrals, more or less. And instead of using Metropolis Monte Carlo, you decide to use molecular dynamics.

This is a tricky choice between those two options. Nice thing about the molecular dynamics is I didn't have to choose any stepsize, basically, right? it's like the time scale was set by the real physical motions of the atoms. And if I don't want to think about it, I can just put in, what's the real physical time scale of the vibration of some atoms. And I don't have to think about it at all and just run it.

But something should be a little bit thinking about is the molecular dynamics equations we're solving, it's really a time accurate method. We actually get real time dependences. We're using the real physical time, which turns out to be pretty darn fast. So molecular vibrations or time scales are like tens of femtoseconds. And so that's really fast. And so you have a really tiny delta t.

But if you're trying to actually compute a steady state property, then maybe this isn't-- it's not necessarily the best way to do it, OK? Because you're doing something where you're doing extra effort to keep the time accuracy. It's sort of along the lines of John Paul's question. If you're doing Kinetic Monte Carlo, if you didn't really care about the time, then you don't have spend time computing the time, right?

And the same thing here is if you don't really care about the time, then you might want to use the Metropolis Monte Carlo instead, because if time's not really in your problem, then you can tailor it, take steps that don't really have to be physical or related to physical amounts of time, and you can still get the right integral. Whereas here, it's going to necessarily do things that are exactly the physical amount of time. And some processes in the real world are pretty darn slow, at least compared to 100 femtoseconds. So you might have problems trying to compute by a time accurate method.

Now on the other hand, sometimes you want to compute time dependent properties. And this is more or less an exact simulation of what the molecules really do on the time scale that

you're simulating. So if that's what you care about, like what's happening on the time scale of picoseconds and nanoseconds, this might be exactly right, because you're actually simulating with a tool that's time accurate on the time scale of what you're really trying to measure. And so some chemical reactions, some kinds of energy transfer processes, like Professor Tisdale's group has exciton transfers that are happening on picosecond time scales. It's tailored to that kind of problem.

OK, but this is the limitation. So you have to use a very small delta t. Therefore, your total time is typically limited to nanoseconds as far as you can integrate, because you have to-- if you just count how many time steps you would need. So if you're trying to determine some kind of static equilibrium property, you start from some initial gas at the positions of the atoms and the initial velocities, and then physically, it takes some time scale for that initial gas to relax to the real equilibrium, because you don't know the real equilibrium situation of the system.

And that time scale, if it's longer than nanoseconds, you're in trouble, because it's not going to be done before you've done the calculation. And you never have even achieved the equilibrium situation. Also, if you have a situation like that hydrogen peroxide case we talked about, suppose it takes a millisecond for the hydrogen peroxide to change confirmation from the dihedral angle being one way and to be the other way.

If I can only say, well, simulate for a nanosecond, then I'm never going to see that happen. So I'm never going to jump from the one conformer to the other. On the other hand, if I care what happens inside the one conformer, everything inside there probably happens on nanosecond time scales, and so I'll get the really good sampling of what's happening inside the one conformer.

And similarly for the dynamic processes, if you have a process that happens on nanosecond time scale, this is really the ideal way to do it. If you have a process that's happening on millisecond or seconds or hours time scales, then this is not really the way to do it at all.

And then there's an initial condition problem with this. It's kind of related to the Kinetic Monte Carlo initial condition problem. So in the Kinetic Monte Carlo, we had to sample over all the possible initial conditions. We did it with the Poisson distribution. There is a similar issue here, is how do I start the initial conditions? Where do I arrange the atoms to be to start out? I really want to sample over all the different possible ways the molecules could be arranged.

And particularly if I have some-- say I have a protein. And the protein has some conformation

it likes to sit-in and then it can unfold and go to some other conformation. Maybe there's two or three conformations like this. The time scales for those changes, again, might be milliseconds. I'm not going to be able to follow them with the molecular dynamic, so I need to have some sampling method to set me up in each of the different conformations that I want to sample. And then I can follow very accurately what would happen over a couple nanoseconds after it's in that confirmation. But I'll hardly ever see it actually achieve the other confirmation on the time scale.

So I guess what I want to say to you guys is these are different tools for really different purposes. The Metropolis Monte Carlo, the Kinetic Monte Carlo, and the molecular dynamics, they're all good for some kinds of problems. But none of them is good for all problems. And I often get journal papers where somebody uses the wrong method for the wrong problem. And so I have to reject it.

I've often had people come that want to do postdocs for me. And they're talking about their thesis, and I see the poor kid has spent five years of his life using the wrong tool for the problem he's doing. It's very sad. So don't that be you, OK? So don't just use a tool because you know it. Say, does this tool work for this problem? If not, I've got to find a new tool. And just make sure you're using tools that match the problems you want that you're trying to solve. I think that's all I got. So I have 10 more minutes left in class. Any questions you guys have? Yeah?

AUDIENCE: Just have one on weighting functions for the Monte Carlo simulations. I'm still a little shaky on how you always determine them. It seems like it was given to us on the problem we did on the homework and then in the example in class, you just set it to an even distribution, or a uniform distribution?

WILLIAM GREEN: Yeah, yeah, yeah.

AUDIENCE: How do you choose what's best, how do you know [INAUDIBLE]?

WILLIAM GREEN: Yeah. Well, yes. So you get some integral. This integral of g of x something. All right? Where this is a lot. And then you have to figure out, what am I going to do so I can solve this thing? And the clever thing is to figure out, hmm, can I rewrite this as p of x times f of x, because I know how to solve those kind. And actually, I don't even-- Yeah, and even if I don't know what p is really, maybe I can write it as w of x over some number of integrals here. That will be my p

of x.

So most of these things you can do with uniform distributions if you want. But then your sampling can be extremely inefficient, because you'll sample a lot of regions that have very low probability of being really there. But this is like a cleverness thing about can I figure out what's that p times f that's equal to g that's going to work the best for me. And work the best, that's a good question. So one way of working the best is, if f is most constant, because if f is perfectly constant, then I get the right answer in the first sample. OK?

So that's one thing is trying to figure out f to be really pretty constant. The second one is to try to figure out p that's very sharply focused. So I don't need to sample a lot of x values to get it. Now these two are kind of at odds with each other, the sharp p and the flat f. Probably again, probably whole people wrote their PhD thesis in applied mathematics about what's the optimal choice of p and f.

A lot of times, I'm not that smart, so I just like, if I have a problem in stat mech, I just-- I'll always do a Boltzmann factor. It may not really the best thing to do, but that's what I would do, right? And if I was doing a Bayesian problem, it's sort of like the w is given to you, right? Where's that? That's the formula I know, so I'm going to use that w. Maybe there's a more clever way to do it, but that's what I normally would do.

Actually, one thing that I don't know if we've ever talked about explicitly, but very important to know, is that this kind of formula, these formulas have w of theta. That's the joint probability that theta one is something, and theta two is something, and theta three is something, and theta four is something. It's that probably density. But a lot of times, you don't care about that level of detail. Like you may have only a few of those parameters that matter to you.

So you really care, you would like to know p of-- I don't know, theta one, theta two. And you'd like to get rid of all the rest of the thetas, because these are two thetas you really care about. Maybe these are the ones you think you're controlling in your experiment, you're trying to determine. In the other ones, somebody already measured the mass of the proton, so you really don't want to determine the mass of proton again, and you're not going to really say anything about it even if you did.

If your calculation says that it made the mass of proton a little bit different than what the standard book says, you might believe that in your heart that it's true, but you're probably not going to say it, because you're like, I better double check before I do that. But I'm sure that it

determines the length of my reactor, because I measured that with a meter stick, and nobody else knows that number, so I'm sure that's the theta I got that I'm really going to be in control of here.

So oftentimes you do this. You want theta one, theta two, but you actually know theta one, theta two, theta three, quite a few of them from my formula. And so you can do what's called a marginal integral. Suppose I have-- I don't know-- theta three, theta four. I could do D theta three, D theta four. And this is like integrating out these degrees of freedom to get the probability density that I want, all right?

So if you have some case where you all you care about is the variance of something, and you don't care about all the rest, you can kind of integrate them out. That's a very handy trick to do.

Well, you can do similar things with the Boltzmann things, like for example, a lot of the Boltzmann distributions, we don't actually care about the momenta, because what we measure, say, is like a crystallography thing. We see positions of atoms. We don't see velocities anyway. So a lot of times, people will just integrate all the velocities out of the problem right at the beginning. It depends on what you want, right? You're in charge of your problem. Any more questions? Yes, Kristen.

AUDIENCE: OK, well, I just have an announcement.

WILLIAM GREEN: OK.

AUDIENCE: [INAUDIBLE] today at office hours, we'll talk about KMC, probably from about 5:00 to 5:30, but come any other time [INAUDIBLE] questions. And we just posted a poll for the final review session. It's either going to be Wednesday evening or Friday morning, so if you could vote on that as soon as possible, that would be [INAUDIBLE].

WILLIAM GREEN: OK, got that? So final review is either Wednesday evening or Friday morning, you have to vote. And if you come today, 5:00 to 5:30, it'll be all about Kinetic Monte Carlo. And other times today are about anything you want to talk about. And the homework solution will be posted shortly, I think, for the last homework that was graded. Anything else? All right, good luck as you study and good luck on the exam.