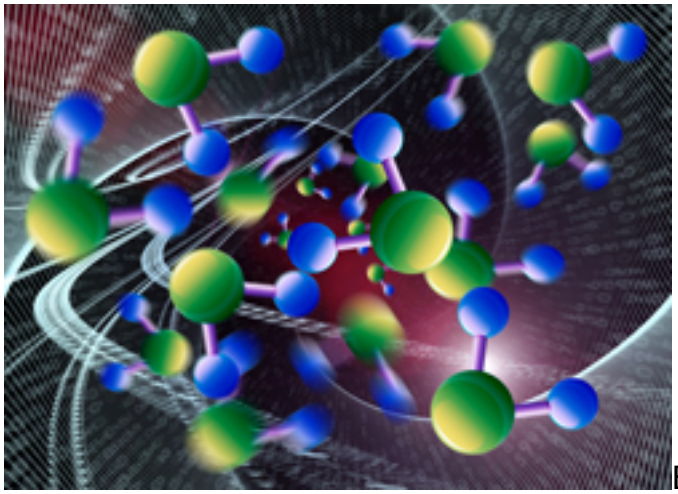
How computers push on the molecules they simulate



Because modern computers have to depict the real world with digital representations of numbers instead of physical analogues, to simulate the continuous passage of time they have to digitize time into small slices. This kind of simulation is essential in disciplines from medical and biological research, to new materials, to fundamental considerations of quantum mechanics, and the fact that it inevitably introduces errors is an ongoing problem for scientists.

Scientists at the U.S. Department of Energy (DOE)'s Lawrence Berkeley National Laboratory (Berkeley Lab) have now identified and characterized the source of tenacious errors and come up with a way to separate the realistic aspects of a simulation from the artifacts of the computer method. The research was done by David Sivak and his advisor Gavin Crooks in Berkeley Lab's Physical Biosciences Division and John Chodera, a colleague at the California Institute of Quantitative Biosciences (QB3) at the University of California at Berkeley. The three report their results in *Physical Review X*.

"Our group uses a theoretical method called nonequilibrium statistical mechanics to study molecular machines, the protein complexes essential to processes like photosynthesis and DNA repair," says Sivak. "But when we applied common algorithms to model the behavior in biological molecules, we found persistent, significant errors in the simulation results."

Systems in equilibrium are relatively easy to simulate, but natural systems are often driven far from equilibrium by absorbing light, burning energy-dense chemical fuel, or other driving forces. Sivak, who recently joined the University of California at San Francisco as a Systems Biology Fellow, describes nonequilibrium statistical mechanics as "a way of understanding situations where conditions change abruptly and the system has to play catch-up," a kind of problem in which there are few exact analytical results.

How biological molecules move is hardly the only field where computer simulations of molecular-scale motion are essential. The need to use computers to test theories and model experiments that can't be done on a lab bench is ubiquitous, and the problems that Sivak and his colleagues encountered weren't new.

"A simulation of a physical process on a computer cannot use the exact, continuous equations of motion; the calculations must use approximations over discrete intervals of time," says Sivak. "It's well known that standard algorithms that use discrete time steps don't conserve energy exactly in these calculations."

One workhorse method for modeling molecular systems is Langevin dynamics, based on equations first developed by the French physicist Paul Langevin over a century ago to model Brownian motion. Brownian motion is the random movement of particles in a fluid (originally pollen grains on water) as they collide with the fluid's molecules—particle paths resembling a "drunkard's walk," which Albert Einstein had used just a few years earlier to establish the reality of atoms and molecules. Instead of impractical-to-calculate velocity, momentum, and acceleration for every molecule in the fluid, Langevin's method substituted an effective friction to damp the motion of the particle, plus a series of random jolts.

When Sivak and his colleagues used Langevin dynamics to model the behavior of molecular machines, they saw significant differences between what their exact theories predicted and what their simulations produced. They tried to come up with a physical picture of what it would take to produce these wrong answers.

"It was as if extra work were being done to push our molecules around," Sivak says. "In the real world, this would be a driven physical process, but it existed only in the simulation, so we called it 'shadow work.' It took exactly the form of a nonequilibrium driving force."

They first tested this insight with "toy" models having only a single degree of freedom, and found that when they ignored the shadow work, the calculations were systematically biased. But when they accounted for the shadow work, accurate calculations could be recovered.

"Next we looked at systems with hundreds or thousands of simple molecules," says Sivak. Using models of water molecules in a box, they simulated the state of the system over time, starting from a given thermal energy but with no "pushing" from outside. "We wanted to know how far the water simulation would be pushed by the shadow work alone."

The result confirmed that even in the absence of an explicit driving force, the finite-time-step Langevin dynamics simulation acted by itself as a driving nonequilibrium process. Systematic errors resulted from failing to separate this shadow work from the actual "protocol work" that they explicitly modeled in their simulations. For the first time, Sivak and his colleagues were able to quantify the magnitude of the deviations in various test systems.

Such simulation errors can be reduced in several ways, for example by dividing the

evolution of the system into ever-finer time steps, because the shadow work is larger when the discrete time steps are larger. But doing so increases the computational expense.

The better approach is to use a correction factor that isolates the shadow work from the physically meaningful work, says Sivak. “We can apply results from our calculation in a meaningful way to characterize the error and correct for it, separating the physically realistic aspects of the simulation from the artifacts of the computer method.”

Source: Lawrence Berkeley National Laboratory [1]

Source URL (retrieved on *05/24/2013 - 5:35pm*):
http://www.rdmag.com/news/2013/01/how-computers-push-molecules-they-simulate

Links:
[1] http://newscenter.lbl.gov/news-releases/2013/01/03/computers-push-molecules/