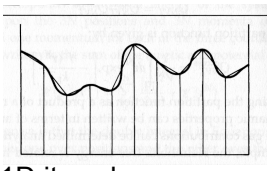
Monte Carlo

- Note that for estimating π (area under circle) a simple Simpson's trapezoidal rule would be faster

Simple MC: π for 10^7 trials = 3.14169

Simpson: π for 10^4 function evaluations = 3.141593



- Word of caution: while in 1D it scales as n

$$\int_a^b f(x)dx = \frac{1}{2}d(f(a)+2f(a+d)+\ldots+f(b))$$

- In 2D it scales as n^2

$$\int_a^b \int_c^d f(x,y)dx\,dy$$

Monte Carlo

- N particles, 3N dimensional integral that would involve m^{3N} function evaluations where m is no. points in the grid in each dimension.
- e.g. N=50, m=3 (very low), 3^{150} (~10^{71}) evaluations are needed...!
- Can MC be much better than straight numerical integration?
  - well, we cannot use typical methods
- BTW, how can we obtain thermodynamics from only positions?

Partition function depend on all 6N degrees of freedom! (p and r)

Monte Carlo

- Thus, some sort of MC is the only choice.
- Need to calculate

$$\langle V(\mathbf{r}^N)\rangle = \int d\mathbf{r}^N V(\mathbf{r}^N)\rho(\mathbf{r}^N)$$

- Where

$$\rho(\mathbf{r}^N) = \frac{exp[-V(\mathbf{r}^N)/k_BT]}{Z}$$

and

$$Z = \int d\mathbf{r}^N exp\left(-\frac{V(\mathbf{r}^N)}{k_BT}\right)$$

Monte Carlo

- Root of the problem with simple MC:
  - Obtain one configuration by randomly generating 3N coordinates
  - Calculate $V(\mathbf{r}^N)$
  - Calculate $exp(-V(\mathbf{r}^N)/k_BT)$ Boltzmann factor
  - Add this one to a sum of other Boltzmann factors
  - After N_{trials}, do

$$\langle V(\mathbf{r}^N)\rangle = \frac{\sum_{i=1}^{N_{trial}} V_i(\mathbf{r}^N)exp[-V_i(\mathbf{r}^N)/k_BT]}{\sum_{i=1}^{N_{trial}} exp[-V_i(\mathbf{r}^N)/k_BT]}$$

Not practical due to the large number of configurations with small (or negligible) Boltzmann Factor!

Metropolis MC

- One way out is to use only those configurations that make a contribution
- Use a specific probability density distribution that chooses those “heavy weights”

Monte Carlo

- In formal terms, we want to evaluate

$$F = \int_{x_1}^{x_2} dx f(x)$$

- That can be written

$$F = \int_{x_1}^{x_2} dx \left(\frac{f(x)}{\rho(x)}\right)\rho(x)$$

ρ(x) is an arbitrary prob. density

Monte Carlo

- We then consider a number of trials τ that involve random numbers drawn from ρ(x) between x_1 and x_2

$$F = \int_{x_1}^{x_2} dx f(x)$$

- Then

$$F = \left\langle \frac{f(\zeta_\tau)}{\rho(\zeta_\tau)} \right\rangle_{trials}$$

Monte Carlo

- For the simple MC, we chose a uniform prob. distribution

$$\rho(x) = \frac{1}{(x_2 - x_1)}$$

- So that

$$F \approx \frac{(x_2 - x_1)}{\tau_{max}} \sum_{\tau=1}^{\tau_{max}} f(\zeta_\tau)$$

Metropolis MC

- For thermodynamics, use

$$\rho(\mathbf{r}^N) = \exp[-V(\mathbf{r}^N)/k_B T]$$

- to generate states, and then count them equally.

$$\text{Now: } f(x) = A\rho_{NVT} \qquad \langle A \rangle_{NVT} = \langle A\rho_{NVT}/\rho \rangle_{trials}$$

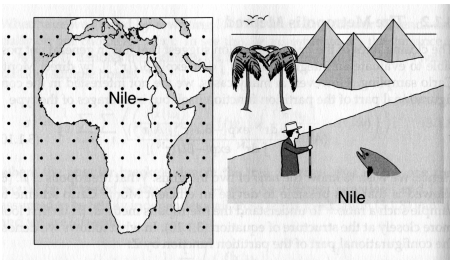
$$if: \ \rho = \rho_{NVT} \qquad \langle A \rangle_{NVT} = \langle A \rangle_{trials}$$

- In contrast,simple MC generates states with equal prob. But weighs them by

$$\exp[-V(\mathbf{r}^N)/k_B T]$$

Metropolis MC

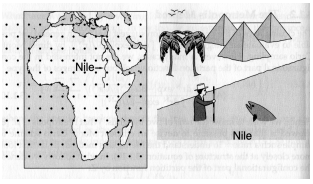
- Example: measure the depth of the Nile,



Simple MC Metropolis MC

Metropolis MC

- Simple MC uses predetermined points
- Metropolis MC generates a random walk through the region of space where contributions are greatest
  - Trial moves are rejected if they go "out of the water" (don't contribute)
  - Unweighted average of these gives and estimate of the depth



Metropolis MC

- Metropolis MC then:
  - Generates a random walk along phase space that generates sequences of random states
  - Each random states occurs with the appropriate probability
  - Trick is to generate such a random walk
- Solution: generate a Markov chain of states
  - Outcome of each trial belongs to a finite set of outcomes
  - Outcomes of each trial depends on the outcome of the previous state
  - This solution saves the trouble of even calculating the partition function!

Metropolis MC

- Have states given by $\{\Gamma_1, \Gamma_2, \ldots \Gamma_m, \Gamma_n, \ldots \Gamma_M\}$
- And the system is described by the probability $\rho(\Gamma)$ which is a vector
- Two states *m* and *n* are linked by a transition probability π_{mn} (probability to go from state *m* to state *n*)
- Given the probability of state *m* to be ρ_m, the probability to go from *m* to *n* is $\rho_m \pi_{mn}$
- An infinite application of π yields the limiting probability ρ_{NVT}

Taking a walk in configurational space

- Transition probability matrix π will take the system from one state to the next
- From state *m* to state *n*
  - $\rho^{(n)} = \pi\, \rho^{(m)}$
- Limiting distribution
  - $\rho = lim_{\tau\to\infty}\, \pi^{\tau}\, \rho^{(1)}$

example

Taking a walk in configurational space

- Means that it will satisfy the eigenvalue equation

$$\rho = \pi\, \rho$$

- With the property

$$\sum_m \rho_m\, \pi_{mn} = \rho_n$$

- and

$$\sum_n \pi_{mn} = 1$$

Metropolis MC

- When searching for the right π it is convenient to add reversibility (detailed balance)

$$\rho_m \pi_{mn} = \rho_n \pi_{nm}$$

overly strong (and not necessary) condition, but guarantees correct sampling

- Possible solution (Metropolis *et al*, 1953)

$$\begin{aligned} \pi_{mn} &= \alpha_{mn} && \rho_n \geq \rho_m \quad m \neq n \\ \pi_{mn} &= \alpha_{mn}(\rho_n/\rho_m) && \rho_n < \rho_m \quad m \neq n \end{aligned}$$

- α_{mn} is rather arbitrary except for satisfying

$\alpha_{mn} = \alpha_{nm}$

Metropolis MC

- A comment on π_{mn} and α_{mn} :
  - π_{mn} is the transition matrix
  - α_{mn} is the *stochastic* matrix (*underlying matrix* in the Markov chain)
  - Given the probability to accept a trial move p_{mn}, then

$$\pi_{mn} = \alpha_{mn}\, p_{mn}$$

  - If $\rho_n \geq \rho_m$ then $p_{mn} = 1$
  - If $\rho_n < \rho_m$ p_{mn} is the ratio of the Boltzmann probabilities, thus recovering the previous expressions

Metropolis MC

- For convenience, use

$$\begin{aligned} \alpha_{mn} &= 1/N_{\Re} && \mathbf{r}_i^n \in \Re \\ \alpha_{mn} &= 0 && \mathbf{r}_i^n \notin \Re \end{aligned}$$

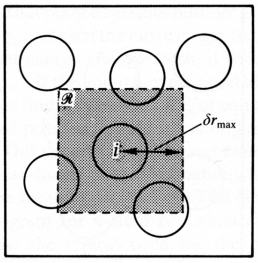
- Assumes there are N_R new positions in the gray square
- If trial move decreases energy, (m to n)

$$\rho_n \geq \rho_m$$

then accept move that *by construction* has probability 1/N

- If trial move increases energy, accept with probability

$$\rho_n/\rho_m$$



Metropolis MC

- That is

$$\frac{\rho_n}{\rho_m} = \frac{exp(-\beta V_n)/Z}{exp(-\beta V_m)/Z} = exp(-\beta \delta V_{nm})$$

- where

$$\delta V_{nm} = \left(\sum_{j=1}^{N} V(r_{ij}^n) - \sum_{j=1}^{N} V(r_{ij}^m) \right)$$

Sums exclude atom i

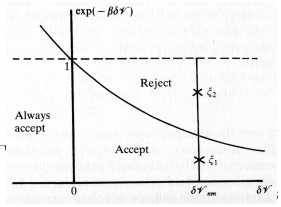
Metropolis MC

- Implementation: pick an atom at random
  - Generate random position (ς is rnd in [0,1])

$$\begin{aligned} x_{new} &= x_{old} + (2\zeta - 1)\delta r_{max} \\ y_{new} &= y_{old} + (2\zeta - 1)\delta r_{max} \\ z_{new} &= z_{old} + (2\zeta - 1)\delta r_{max} \end{aligned}$$

  - Calculate energy difference δV_{nm}
  - If get < 0, accept
  - If get > 0, flip a coin

$$rand(0,1) \leq exp(-\delta V(\mathbf{r}^N))/k_B T$$



Metropolis MC

- Acceptance ratio depends on the size of

$$\delta r_{max}$$

- This is an adjustable parameter.
  - If too small: many accepted moves, but these states will be very similar – slow exploration of phase space
  - If too big: many rejected moves because of overlaps
- Adjust such that get about 50% of acceptance
- Can get fancier: dynamically adjust!

Metropolis MC

- Tricks:
  - Remember that change in energy only depends on the moving particle

$$\delta V_{nm} = \left(\sum_{j=1}^{N} V(r_{ij}^n) - \sum_{j=1}^{N} V(r_{ij}^m) \right)$$

  - For short-range only need a neighborhood, r_c
  - Could account for the rest by effective term, but difference takes it away, so do not need it!
  - Don't need to choose atoms at random, sequential will do
  - Can move many atoms at a time

demo

MC vs MD

- MD:
  - Other ensembles: NVE (conventional), NVT, const-P
  - Time-dependent quantities such as transport coefficients
  - Good for large molecules
- MC:
  - In addition to the above, isobaric-isothermal, μVT
  - Lattice models
  - Rapid convergence on thermo properties of simple small molecular liquids

Other Ensembles

- Canonical (review) (NVT):
  - Partition function

$$Q(N,V,T) \equiv \frac{1}{\Lambda^{3N} N!} \int d\mathbf{r}^N exp[-\beta U(\mathbf{r}^N)]$$

  - Where

$$\Lambda = \sqrt{h^2/(2\pi m k_B T)}$$

  - Probability of finding configuration $\mathbf{r}^N$

$$N(\mathbf{r}^N) \propto exp[-\beta U(\mathbf{r}^N)]$$

Limiting distribution

  - implementation

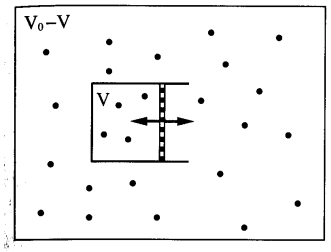
$$acc(o \rightarrow n) = min(1, exp\{-\beta[U(n) - U(o)]\})$$

Isobaric-Isothermal Ensemble

- (NPT) Similar to most experiments with constant P and T *
- NPT are convenient for equations of state. Does not need the virial (good for non-spherical particles, or where pot func is only available numerically).
- NPT good close to 1st order phase transitions:
  - Allows system to transform to other phase
  - Avoids phase separation typical in NVT

*One sweep in NPT can yield ρ for 1atm, while using MD for NVT would need many trials

NPT



- System can exchange volume with external system
- In the limit of a very big exterior system, change in P does not affect pressure outside

NPT

- Isobaric-Isothermal Ensemble (NPT)
  - Partition function ($L=V^{1/3}$, $\mathbf{r}_i = L\mathbf{s}_i$) Because of the scaled lengths, the ds integral is over a unit cube

$$Q(N,P,T) \equiv \frac{\beta P}{\Lambda^{3N} N!} \int dV V^N exp(-\beta PV) \int d\mathbf{s}^N exp[-\beta U(\mathbf{s}^N; L)]$$

  - Probability of finding configuration with s^N at a given V

$$N(V;s^N) \propto V^N exp(-\beta PV) exp[-\beta U(s^N;L)]$$
$$= exp\{-\beta[U(s^N,V) + PV - N\beta^{-1} \ln V]\}$$

Limiting distribution

  - Implementation (V is also tested as V')

$$acc(o \rightarrow n) = min(1, exp\{-\beta[U(s^N,V') - U(s^N,V) + P(V'-V) - N\beta^{-1} \ln(V'/V)]\})$$

NPT - implementation

- V is simply treated as an additional coordinate
- Trial moves in s_i
- Trial moves in V: V' = V + ΔV where ΔV random number uniformly distributed between
  - $[-\Delta V_{max}, \Delta v_{max}]$
- But may do volume trial less frequently (it is expensive) than coordinate trials
  - Once every cycle, but still measure 1/N to do or not

NPT

- Pseudo code
- Prob for vol change goes like 1/N

```
PROGRAM mc_npt                          basic NPT ensemble simulation

do icycl=1,ncycl                        perform ncycl MC cycles
  ran=ranf()*(npart+1)+1
  if (ran.le.npart) then
    call mcmove                         perform particle displacement
  else
    call mcvol                          perform volume change
  endif
  if (mod(icycl,nsamp).eq.0)
+   call sample                         sample averages
enddo
end
```


NPT

- However, vol change is cheap with potentials that are powers of distance:

$$U_n = \sum_{i<j} \varepsilon(\sigma/r_{ij})^n$$
$$= \sum_{i<j} \varepsilon[\sigma/(Ls_{ij})]^n$$

- thus

$$U_n(L') = \left(\frac{L}{L'}\right)^n U_n(L)$$

Can do volume trial as frequent as particle displacements

notes