

12. Sampling methods
why not use Boltzmann
sampling?
umbrella sampling

SAMPLING METHODS

Simple sampling - worst, completely unbiased.
Importance sampling - general technique of choosing random numbers from a distribution that allows the function evaluation to be concentrated in the regions of function space that make important contributions.

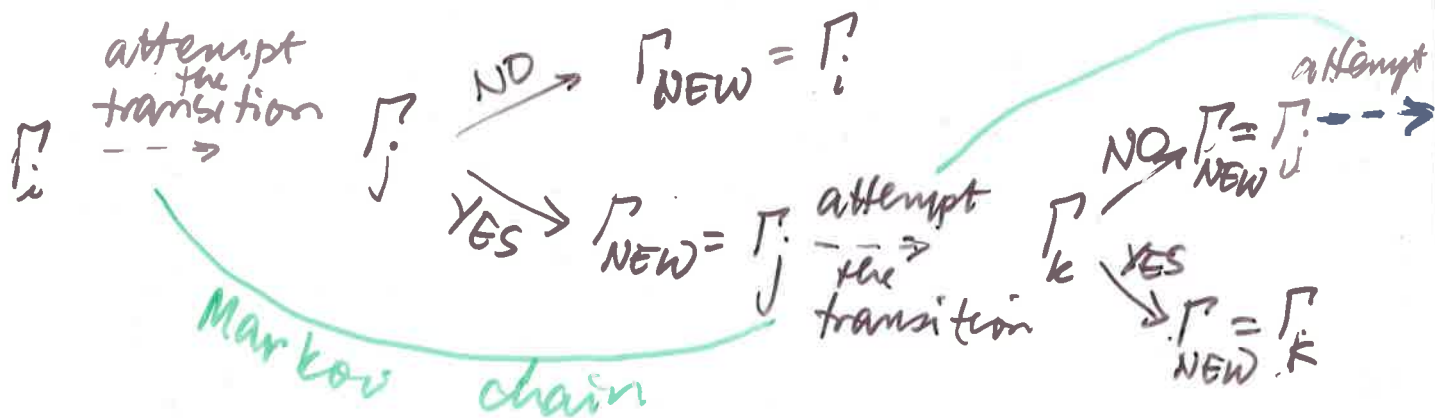
But the hard part is to find a method of generating a sequence of random states so that by the end of the simulation each state has occurred with the appropriate probability.

Metropolis's solution to this difficulty is to set up a Markov chain of states of the system. The chain is so constructed so that it has a limiting distribution that is the appropriate one.

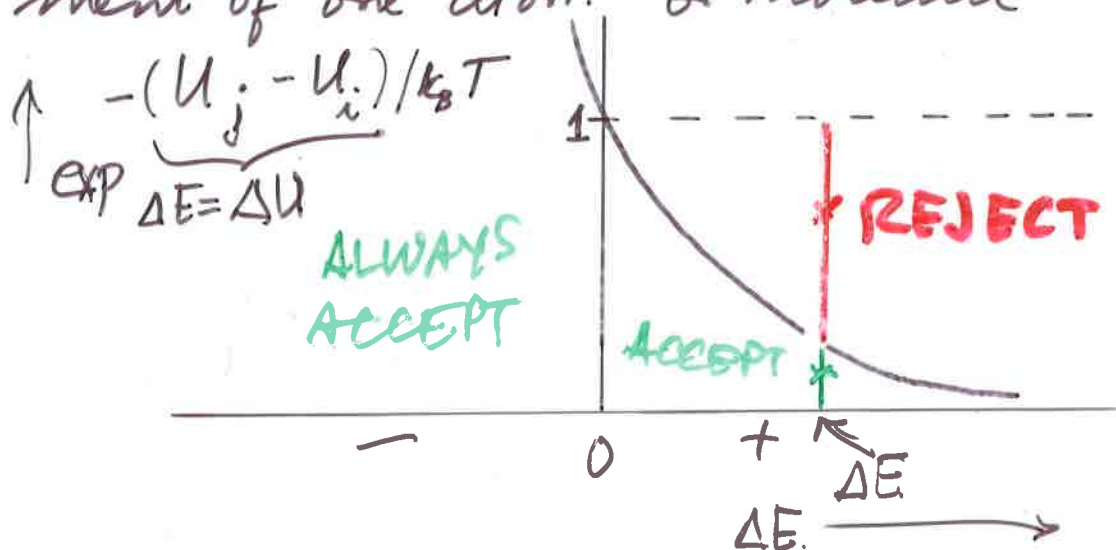
What is a Markov chain?

A Markov chain is a sequence of trials (attempts) such that

- the outcome of each trial belongs to a finite set of outcomes $\{\Gamma_1, \Gamma_2, \Gamma_3, \dots\}$. This set is called the state space, each one is called a system quantum state or a state configuration.
- the outcome of each trial depends only on the outcome of the trial that immediately precedes it.



where Γ_i and Γ_j are neighboring states which are related (for example) by a displacement of one atom or molecule



In the original **Metropolis** method, one randomly chosen atom is moved to generate a new state.

Variations on this original scheme:

- several or all atoms are moved simultaneously, accept or reject using normal criteria

$$P_{\text{accept}} = \begin{cases} \min [1, \exp(-\Delta E/k_B T)] & \frac{\Delta E}{k_B T} \leq \text{largest exponent (machine limit)} \\ 0 & (\Delta E/k_B T) > \text{largest exponent (machine limit)} \end{cases}$$

- Grand canonical MC, not only displace atoms but create/annihilate (Norman-Filinov)

P_{accept} = same as original Metropolis criteria

but

$$\Delta E = \Delta U = U_j - U_i \quad \text{displace}$$

$$\Delta E = \Delta U + k_B T \ln \left(\frac{N+1}{V \rho^0} \right) - \mu \quad \text{create}$$

$$\Delta E = \Delta U - k_B T \ln \left(\frac{N}{V \rho^0} \right) + \mu \quad \text{annihilate}$$

So why do we need other sampling methods?

- (1) When the density of molecules is high, the commonly used version of Metropolis' method is inefficient since most moves are REJECTED!
- (2) Metropolis' MC is designed to sample regions of configuration space in which potential energy is negative or small positive, BUT some properties require more substantial sampling over higher energy configurations, i.e., requires NON-BOLTZMANN sampling.

To overcome these two problems, we need some additional bias beyond what conventional Metropolis' sampling provides.

To introduce the bias, one effectively introduces a weighting function that weights Π_i and Π_j ; the weighting function is chosen to encourage the system to explore regions of phase space not frequently sampled by conventional Metropolis'. The ^{biased} results are then CORRECTED for this bias at the end of the simulation, so as to give the desired ensemble averages.

Some methods to overcome problem (1):

(a) CBGCMC — Cavity-biased grand canonical Monte Carlo

(b) FBMC — Force-bias Monte Carlo

(c) SMC — "Smart Monte Carlo"

A method to overcome problem (2) is "Umbrella sampling"

CAVITY-BIASED GCMC

Problem (1) in GCMC: At high densities creation and destruction attempts can have unacceptably low acceptance ratios.

$$\hookrightarrow \frac{\# \text{ moves accepted}}{\# \text{ moves attempted}}$$

Solution: First, look for cavities of a suitable radius r_{hole} . Then, choose one among the found cavities, attempt to create a particle there.

This method introduces a bias: we only attempt to insert a particle in not a randomly chosen location, but in a location randomly chosen from among those which have a high probability of accepting an insertion.

The probabilities of accepting a creation or annihilation attempt are modified by an additional factor $P_N =$ the probability of finding a cavity of radius r_{hole} or greater in a fluid of N molecules.

$$P_{\text{accept}}^{\text{creation}} = \min\left[1, \exp\left\{\frac{-1}{k_B T} [\Delta U + k_B T \ln\left(\frac{N+1}{V_{\text{pot}}}\right) - \mu]\right\}\right]$$

P_N

$$P_{\text{accept}}^{\text{annihilation}} = \min\left[1, \exp\left\{\frac{-1}{k_B T} [\Delta U - k_B T \ln\left(\frac{N}{V_{\text{pot}}}\right) + \mu]\right\}\right]$$

P_{N-1}

that is, where volume appears, we now have

$$P_N V \quad \text{or else} \quad P_{N-1} V$$

Furthermore, $P_N = \frac{\text{number of found cavities}}{\text{number of test points}}$
 Since the random insertion occurs with probability $(1 - P_N)^{N_{\text{TEST}}}$ by the nature of the process, then, annihilation acceptance will have to be chosen with probability $(1 - P_N)^{N_{\text{TEST}}}$

Based on all of the above considerations the proposed algorithm for the (T, V, μ) Monte Carlo simulations consists of the following steps:

(1) Initialize N, r^N .

CAVITY-BIASED
method

(2) Perform a particle displacement according to the canonical ensemble Metropolis method.

(MEZEI)

(3) Perform an insertion/deletion step as follows:

(3.1) With probability $1/2$ go to step 3.3.

(3.2) Insertion:

Generate N_t points.

Find the number of points that are in a cavity.

Update the estimate of P_c^N .

If no point was found to be in a cavity, go to step 3.2.2.

(3.2.1) Cavity-biased insertion:

Insert one particle at one of the points which was found to be in a cavity.

Accept the insertion with probability P_i^{CB} .

Go to step 4.

(3.2.2) Random insertion:

Insert a particle at a randomly selected point.

Accept the insertion with probability P_i^R .

Go to step 4.

(3.3) Deletion:

Choose a particle randomly.

With probability $(1 - P_c^{N-1})^{N_t}$ go to step 3.3.2.

(3.3.1) Cavity-biased deletion:

Delete the chosen particle with probability P_d^{CB} .

(Note, that in case $N-1$ was not sampled yet in the calculation, P_c^{N-1} is to be obtained by extrapolating from P_c^N, P_c^{N+1}, \dots , or is to be set equal to P_c^N if N is the only value sampled. In the calculations reported here, linear extrapolation was used.)

Go to step 4.

(3.3.2) Random deletion:

Delete the chosen particle with probability P_d^R .

(4) Accumulate ensemble averages.

(5) Repeat from step 2 until acceptable convergence limits have been reached.

CAVITY-BIASED

OCMC

~/mixture/cbgemix.f

```

C *****
C FLUID1 CREATION STEP
C *****

c-----
c Now ready to create particle 1
c Check to see if there are any holes to put it in
  itest = 0
c Generate NTEST test points uniformly
  do 95 i=1,NTEST
    xt = g05daf(0.0d0,1.0d0)*size
    yt = g05daf(0.0d0,1.0d0)*size
    zt = g05daf(0.0d0,1.0d0)*size
c Find out if the test point is in a beta cage
  DO 9 ib=1,IBCAGE
    DISTX = Xt - BCAGE(ib,1)
    DISTY = Yt - BCAGE(ib,2)
    DISTZ = Zt - BCAGE(ib,3)
    DISTX = DISTX - SIZE * ANINT(DISTX/SIZE)
    DISTY = DISTY - SIZE * ANINT(DISTY/SIZE)
    DISTZ = DISTZ - SIZE * ANINT(DISTZ/SIZE)
    DIST2 = DISTX**2 + DISTY**2 + DISTZ**2
    IF (DIST2.LT.SEPKAG) go to 95
  9 CONTINUE
c Not close to any zeolite atoms?
c rhole = (sigco*rf)**2
c rhole = COMIN*1.05d0
c rhole = COMIN
  DO 76 m=1,NO
    distx = Xt - O(m,1)
    disty = Yt - O(m,2)
    distz = Zt - O(m,3)
    distx = distx - SIZE*ANINT(distx/SIZE)
    disty = disty - SIZE*ANINT(disty/SIZE)
    distz = distz - SIZE*ANINT(distz/SIZE)
    dist2 = distx**2 + disty**2 + distz**2
    if (dist2.lt.rhole) goto 95
  76 CONTINUE
c rhole = (sigcs*rf)**2
c rhole = CSMIN*1.05d0
c rhole = CSMIN
  DO 77 m=1,NS
    distx = Xt - S(m,1)
    disty = Yt - S(m,2)
    distz = Zt - S(m,3)
    distx = distx - SIZE*ANINT(distx/SIZE)
    disty = disty - SIZE*ANINT(disty/SIZE)
    distz = distz - SIZE*ANINT(distz/SIZE)
    dist2 = distx**2 + disty**2 + distz**2
    if (dist2.lt.rhole) goto 95
  77 continue

c rhole= (sig11*rf)**2
c rhole = CCMIN*1.05d0
c rhole = CCMIN
c see if the test point intersects Ar atoms
  do 92 m=1,nmole1
    distx = x1(m)-xt
    disty = y1(m)-yt
    distz = z1(m)-zt

```

```

distx = distx -size*anint(distx/size)
disty = disty -size*anint(disty/size)
distz = distz -size*anint(distz/size)
dist2 = distx**2 + disty**2 + distz**2
if (dist2.lt.rhole) goto 95
92 continue
c did not find any overlaps with Ar, now check for Xe
c rhole = (sig12*rf)**2
c rhole = CNMIN*1.05d0
c rhole = CNMIN
do 94 m=1,nmole2
distx = x2(m)-xt
disty = y2(m)-yt
distz = z2(m)-zt
distx = distx -size*anint(distx/size)
disty = disty -size*anint(disty/size)
distz = distz -size*anint(distz/size)
dist2 = distx**2 + disty**2 + distz**2
if (dist2.lt.rhole) goto 95
94 continue

```

```

c found a hole!
itest = itest+1
xhole(itest) = xt
yhole(itest) = yt
zhole(itest) = zt
95 continue

```

add one to the number of holes found

done generating NTEST test points

```

c update the cavity biased probability
nt1(nmole1,nmole2) = nt1(nmole1,nmole2) + 1
frac = dfloat(itest)/dfloat(nctest)
frac1(nmole1,nmole2) = frac1(nmole1,nmole2) + frac
frac1(nmole1,nmole2) = frac1(nmole1,nmole2)
+ /dfloat(nt1(nmole1,nmole2))
c if no holes found attempt to create at a random location anyway
if (itest.eq.0) goto 103
c now choose one among the found holes
ih = g05daf(0.0d0,1.0d0)*itest +1
nmakeatt1 = nmakeatt1 + 1
x1new = xhole(ih)
y1new = yhole(ih)
z1new = zhole(ih)

```

continuously updated probabilities PN

```

DO 96 J=1,NO
RXCO(J) = X1NEW - O(J,1)
RYCO(J) = Y1NEW - O(J,2)
RZCO(J) = Z1NEW - O(J,3)
RXCO(J) = RXCO(J) - SIZE*ANINT(RXCO(J)/SIZE)
RYCO(J) = RYCO(J) - SIZE*ANINT(RYCO(J)/SIZE)
RZCO(J) = RZCO(J) - SIZE*ANINT(RZCO(J)/SIZE)
96 CONTINUE

```

etc. calculate energy changes

```

c-----
IF (ITEST.EQ.0) DETOTAL=DEa1/UGC/TEMP+LOG (DFLOAT(NMOLE1+1) /VOLUME)
+ -RTLACT1/UGC/TEMP
IF (ITEST.NE.0) DETOTAL=DEa1/UGC/TEMP+LOG (DFLOAT(NMOLE1+1) /VOLUME)
+ /frac1(nmole1,nmole2) -RTLACT1/UGC/TEMP

```



```

C *****
C REJECT/ACCEPT NEW CONFIGURATION BASED ON METROPOLIS SCHEME
C *****

```

```

PACC = 1.0D0
IF (DETOTAL.GT.EXPLIM) DETOTAL = EXPLIM
IF (DETOTAL.GT.0.0D0) PACC = EXP(-DETOTAL)
IF (G05DAF(0.0D0,1.0D0).LT.PACC) THEN

```

```

C UPDATE THE ENERGY (IN kJ PER MOLE)

```

```

E = E + DEa1
sh = sh + dstotal
E1 = E1 + DEa1
E2 = E2 + EF1F2NEW
EFF = EFF + EF1F1NEW + EF1F2NEW
sff = sff + sff12new
EF1F1 = EF1F1 + EF1F1NEW
EF1F2 = EF1F2 + EF1F2NEW
sff12 = sff12 + sff12new
EFW = EFW + EFWNEW1
EFW1 = EFW1 + EFWNEW1
.... etc. updating sums

```

```

C *****
C FLUID1 ANNIHILATION STEP
C *****

```

```

C RANDOMLY SELECT AN ATOM TO TRY AND DESTROY (EXCHANGE WITH BULK)

```

```

110 IF (NMOLE1.EQ.0) GOTO 100
NDEADATT1 = NDEADATT1 + 1
NKILL = NINT(G05DAF(0.5D0, (DFLOAT(NMOLE1)+0.5D0)))
DEa1 = -EZE01(NKILL)
dstotal = 0.0d0
EFWNEW1 = -EZE01(NKILL)
EF1F1NEW = 0.0D0
EF1F2NEW = 0.0D0
sff12new = 0.0d0
DO 230 J = 1,NMOLE1
DEa1 = DEa1 - EIJ11(J,NKILL)
EF1F1NEW = EF1F1NEW - EIJ11(J,NKILL)

```

```

230 CONTINUE

```

```

DO 232 J = 1,NMOLE2
DEa1 = DEa1 - EIJ12(NKILL,J)
dstotal = dstotal -sij12(nkill,j)
EF1F2NEW = EF1F2NEW - EIJ12(NKILL,J)
sff12new = sff12new - sij12(nkill,j)

```

```

232 CONTINUE

```

```

C-----
c with probability (1-fra1)^NTEST use random point probability
c else choose cavity biased probability
pann = (1.0d0 - fra1(nmole1-1,nmole2))*NTEST
IF (G05DAF(0.0D0,1.0D0).LT.PANN) THEN
DETOTAL=DEa1/UGC/TEMP-LOG (DFLOAT(NMOLE1)/VOLUME)
+
+RTLACT1/UGC/TEMP
ELSE
DETOTAL=DEa1/UGC/TEMP-LOG (DFLOAT(NMOLE1)/VOLUME
+ /fra1(nmole1-1,nmole2)) +RTLACT1/UGC/TEMP
end if

```



P_{N-1}

```

C *****
C REJECT/ACCEPT NEW CONFIGURATION BASED ON METROPOLIS SCHEME
C *****

```

```
PACC = 1.0D0
IF (DETOTAL.GT.EXPLIM) DETOTAL = EXPLIM
IF (DETOTAL.GT.0.0D0) PACC = EXP(-DETOTAL)
IF (G05DAF(0.0D0,1.0D0).LT.PACC) THEN
```

```
C ACCEPT THE NEW CONFIGURATION
```

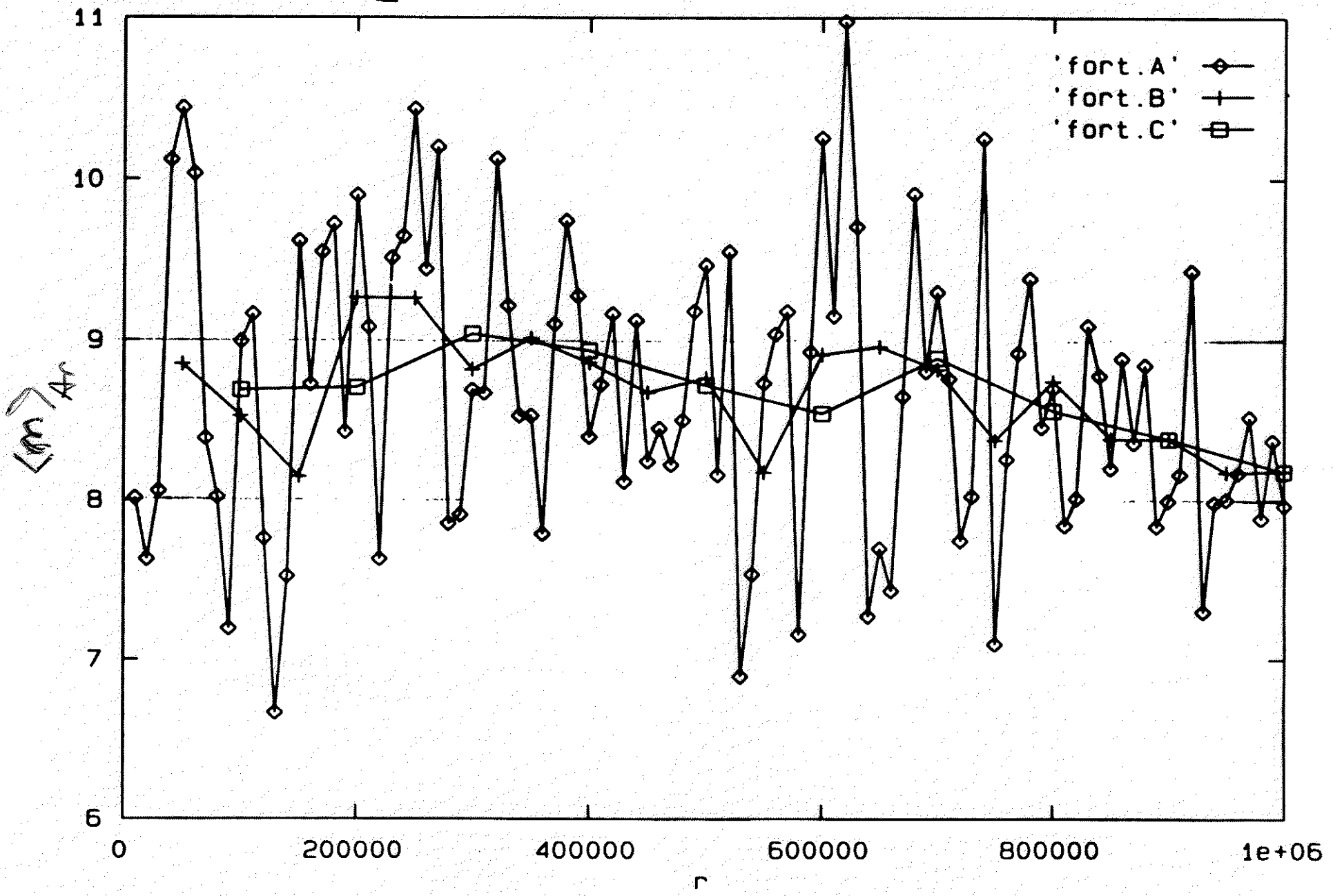
```
NDEAD1 = NDEAD1 + 1
E       = E       + DEa1
sh      = sh      + dstotal
E1      = E1      + DEa1
E2      = E2      + EF1F2NEW
EFW     = EFW     + EFWNEW1
EFW1    = EFW1    + EFWNEW1
EFF     = EFF     + EF1F1NEW + EF1F2NEW
sff     = sff     + sff12new
EF1F1   = EF1F1   + EF1F1NEW
EF1F2   = EF1F2   + EF1F2NEW
sff12   = sff12   + sff12new
....etc. update the sums that are being accumulated ...
```

new arrays used

```
dimension xhole(ntest),yhole(ntest), zhole(ntest),
+ fra1(0:max,0:max),fra2(0:max,0:max),frac1(0:max,0:max),
+ frac2(0:max,0:max), nt1(0:max,0:max),nt2(0:max,0:max)
```

C-----

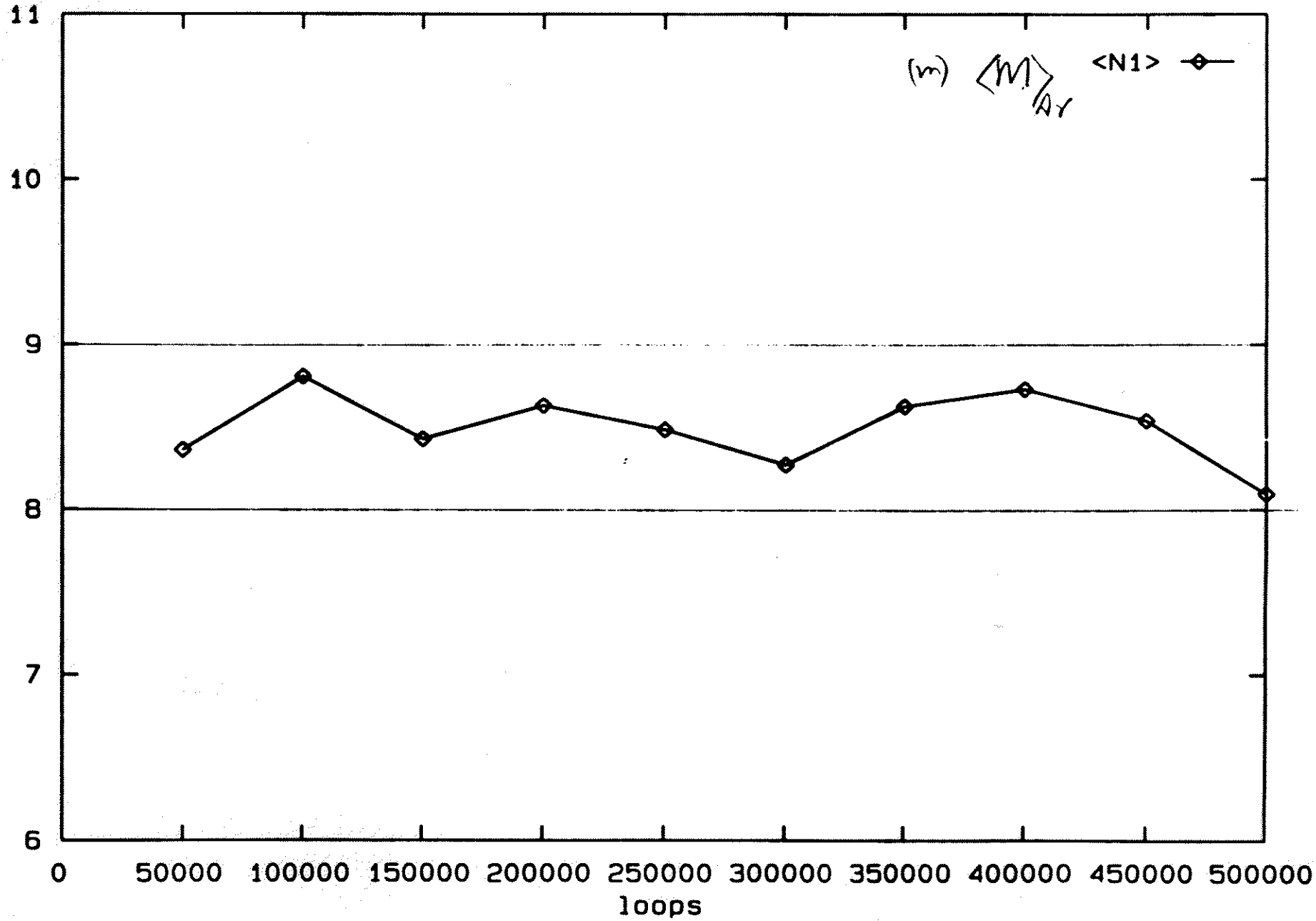
GCMC $\mu_1=20772$ $\mu_2=26330$ $y_{Ar}=0.90$ $T=300K$



CBGCMC

$\mu_1=20772, \mu_2=26330, t=300, \gamma=0.90$

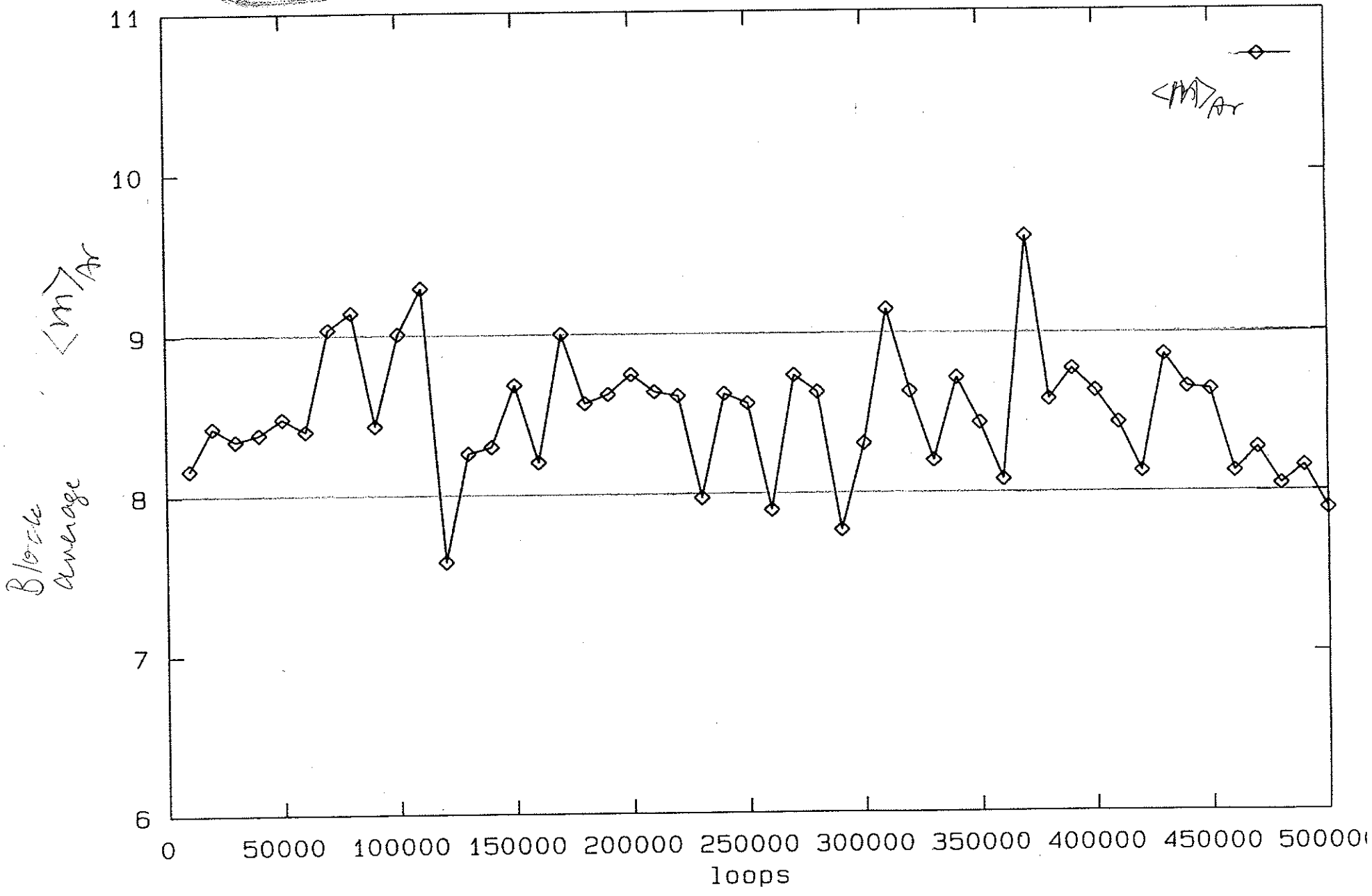
Block average $\langle M \rangle_M$
each block is 50,000 loops



$N_{EQUIL} = 10,000$ $N_{SUB} = 50,000$

CBGCMC

$\mu_1=20772, \mu_2=26330, t=300, \text{var}=0.90$

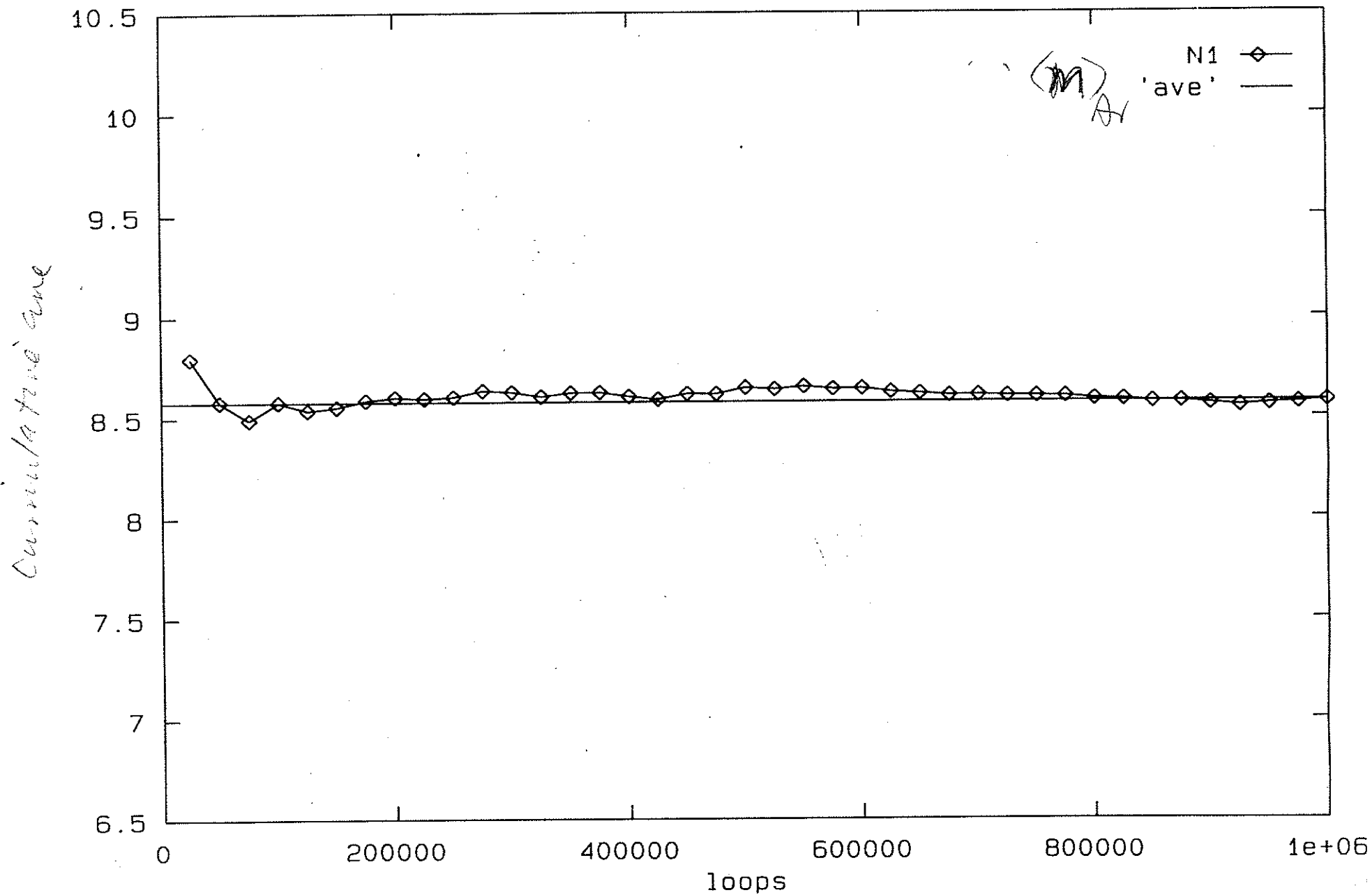


NEQUIL = 10,000

NSUB = 10,000

CBGEMC NEQUIL = 100,000

Ar/Xe/NaA yar = .900, $\langle n \rangle_{ar} = 1.072$ $\langle n \rangle_{Xe} = 2.967$ 11.05atm



Umbrella Sampling

BOLTZMANN WEIGHTING

dr stands for $dr^N d\Omega^N$
(over all positional and orientational coordinates)

ENSEMBLE AVERAGE of a property A

$$\langle A \rangle_{NVT} = \frac{\int dr A(r) \exp^{-U(r)/kT}}{\int dr e^{-U(r)/kT}}$$

$$= \frac{\sum_{i=1}^{\text{all configs}} A_i(r) \exp^{-U_i(r)/kT}}{\sum_{i=1} \exp^{-U_i(r)/kT}}$$

Metropolis:

Choose a Boltzmann distribution of configurations

$$P(r) = \frac{\exp^{-U_i(r)/kT}}{\int dr \exp^{-U_i(r)/kT}}$$

$$\langle A \rangle_{NVT} = \frac{1}{N\text{STEPS}} \sum_{i=1}^{N\text{STEPS}} A_i$$

provided the configurations i are sampled at random from the distribution P appropriate to NVT

NON-BOLTZMANN WEIGHTING

ENSEMBLE AVERAGE of a property A

$$\langle A \rangle_{NVT} = \frac{\int dr \frac{A(r)}{W(r)} W(r) e^{-U(r)/kT}}{\int dr \frac{1}{W(r)} W(r) e^{-U(r)/kT}}$$

Patey + Valleau (1975)

Choose a non-Boltzmann distribution of configurations

$$P_W(r) = \frac{W(r) \exp^{-U_i(r)/kT}}{\int dr W(r) \exp^{-U_i(r)/kT}}$$

$$\langle A \rangle_{NVT} = \frac{\sum_{i=1}^{N\text{STEPS}} \left(\frac{A_i(r)}{W(r)} \right)}{\sum_{i=1}^{N\text{STEPS}} \left(\frac{1}{W_i(r)} \right)}$$

provided the configurations i are sampled at random from the appropriate distribution

What is the limiting distribution?

$$\frac{P_j}{P_i} = \frac{\exp^{-U_j/kT}}{\exp^{-U_i/kT}}$$

reference
distribution

How to end up with this limiting
distribution?

ALGORITHM:

Form a sequence of trials that
accept or not accept neighboring
configurations according to the
criterion:

$$P_{\text{accept}} = \min\left[1, \exp^{-\frac{(U_j - U_i)}{kT}}\right]$$

Using this sequence of trials:

$$\langle A \rangle = \frac{1}{N_{\text{TRIALS}}} \sum_j A_j \stackrel{\text{accumulate}}{=} \langle A \rangle_{\text{trials}}$$

What is the relation of P_W to the
reference distribution?

$$\frac{P_j}{P_i} = \frac{P_{W(U_j - U_i)} / W(U_j - U_i)}{\langle 1 / W(U_j - U_i) \rangle}$$

ALGORITHM:

Form a sequence of trials that accept
or not accept neighboring configura-
tions according to the
criterion:

$$P_{\text{accept}} = \min\left[1, \frac{W_j}{W_i} \exp^{-\frac{(U_j - U_i)}{kT}}\right]$$

where $W_j \neq W_i$ (i.e., different
additional weighting factors for
different configurations)

$$\langle A \rangle = \frac{\langle A/W \rangle_{\text{trials}}}{\langle 1/W \rangle_{\text{trials}}}$$

PROGRAMMING:

Using the above criterion for P_{accept} , accumulate A over a large number of Monte Carlo trials, then divide by the number of trials to get

$$\langle A \rangle_{NVT}$$

PROGRAMMING:

Using the above criterion for P_{accept} , accumulate A/W over a large number of Monte Carlo trials.

At the same time, accumulate $1/W$.

Divide the accumulated $\sum^{\text{accum.}} A/W$ by the accumulated $\sum^{\text{accum.}} 1/W$

to get $\langle A \rangle_{NVT}$

- Why is it called umbrella sampling?

P_W is chosen as wide and uniform as possible, forming an umbrella over the configurations, that is, extending the range of energies sampled in a conventional Monte Carlo by a factor of 3 or more, thereby allowing more accurate calculation of much smaller p values.

where Boltzmann gives very little discrimination between ≈ 0 and ≈ 0

- What are the problems in doing umbrella sampling?

(a) There is no a priori recipe for finding a weighting function W

(b) Subjective element in choosing a suitable weighting function

(c) Trial and error method of adjusting W until P_W is as wide and uniform as possible.

EXAMPLE

Interionic forces

Large distances : Primitive model
(used in Debye-Hückel theory) works well

Assume the effect of the solvent on the interionic interactions can be represented by treating the solvent as a continuum dielectric with a dielectric constant ϵ .

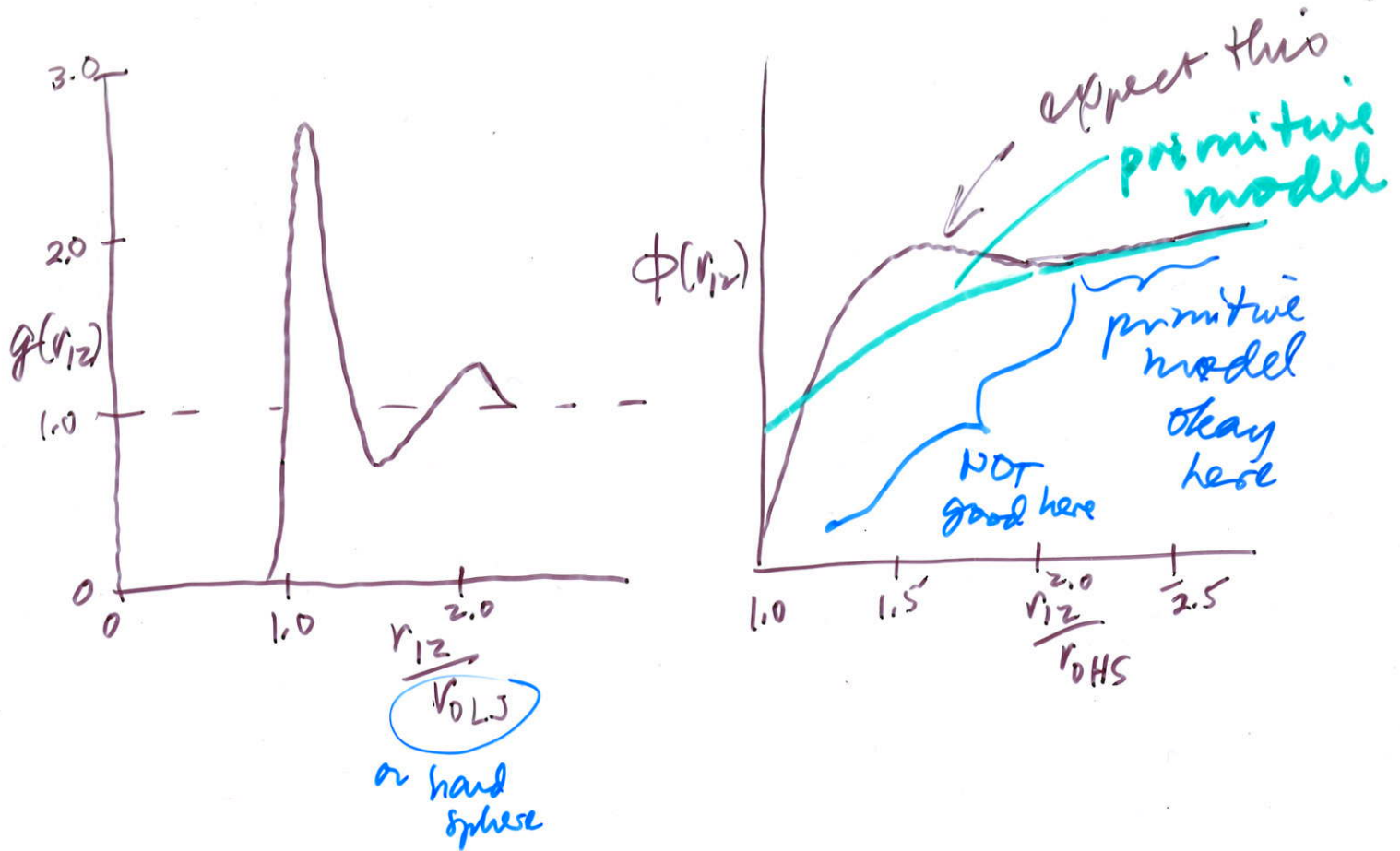
$$\phi_{\text{primitive model}}(r_{12}) = U_{\text{hard sphere}}(r_{12}) + \frac{q_1 q_2}{\epsilon r_{12}}$$

Pair potential energy

Why can't we use the primitive model?

From the standpoint of electrochemistry, the magnitude of the enhancement of $g(r_{12})$ near contact, and its r_{12} dependence are of some interest.

Expect a minimum in the radial distribution function $4\pi r_{12}^2 g(r_{12})$ and a peak near contact



The phenomenon is often termed "ion pairing"

It plays an important part in the discussions of both the thermodynamic and transport properties of electrolytes.

Primitive model does not describe the ion pairing region well.

Therefore, use Monte Carlo simulations of a pair of ions with a large number of water molecules.

Why not use straightforward Boltzmann sampling?

Boltzmann sampling would ONLY explore the small separation regions whereas a wider region of ionic separations are of interest up to the regions where the primitive model works or is expected to work

∴ use NON-BOLTZMANN sampling as shown.

Why not use a weighting function (found by trial and error) so that the desired non-Boltzmann sampling can be done over the entire range of r_{12} values?

Tedious trial & error method may not zero in on a form that is applicable through-out.

∴ use different weighting functions for different overlapping ranges of r_{12}

$$W_{(r_{12}, \text{Range})} = \exp \left[\frac{A}{r_{12}} + \frac{B}{r_{12}^2} \right] / U(r_{12}, \text{Range})$$

adjustable parameters
↓

Introduced
umbrella
sampling
and use of
overlapping
ranges

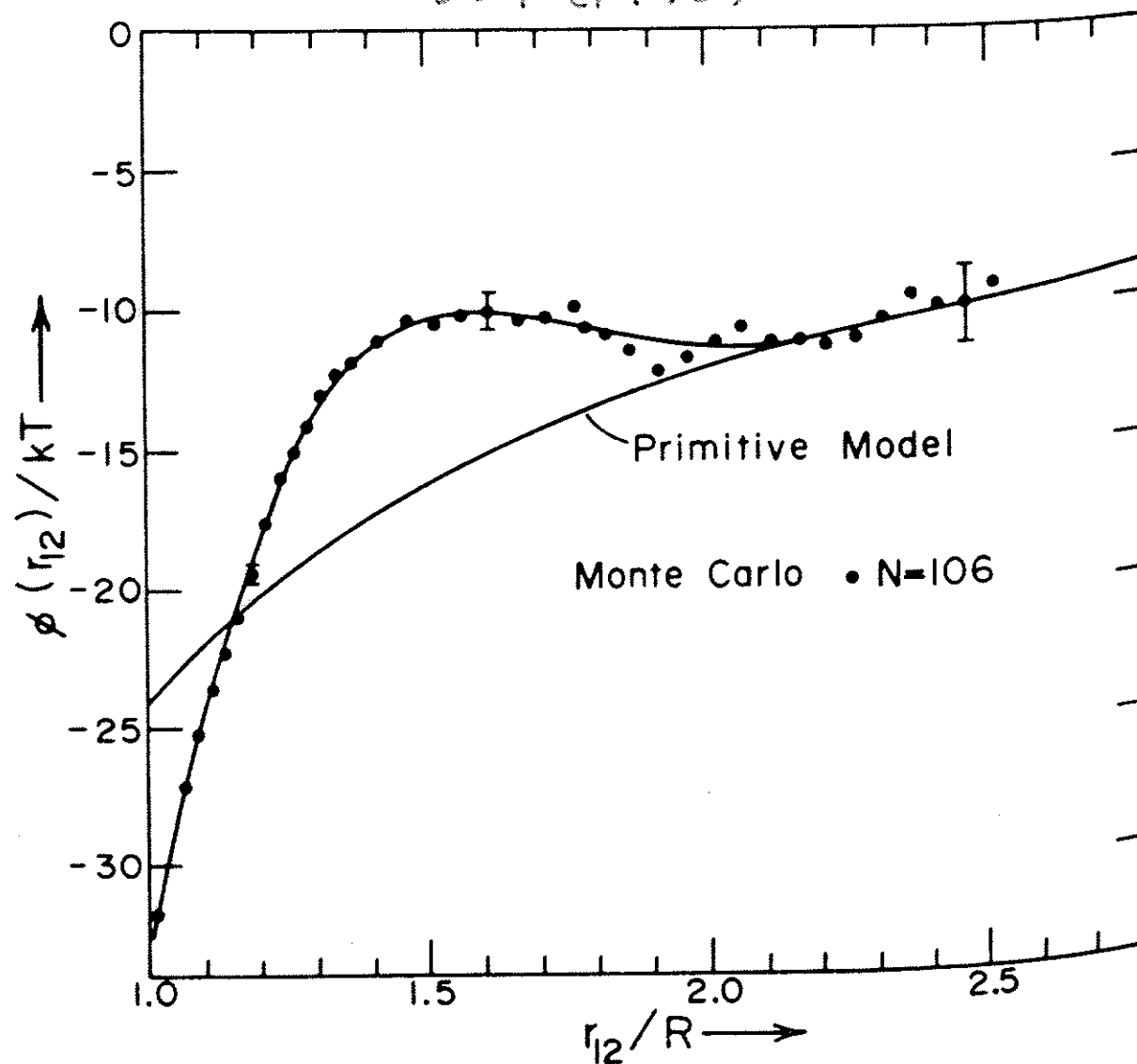
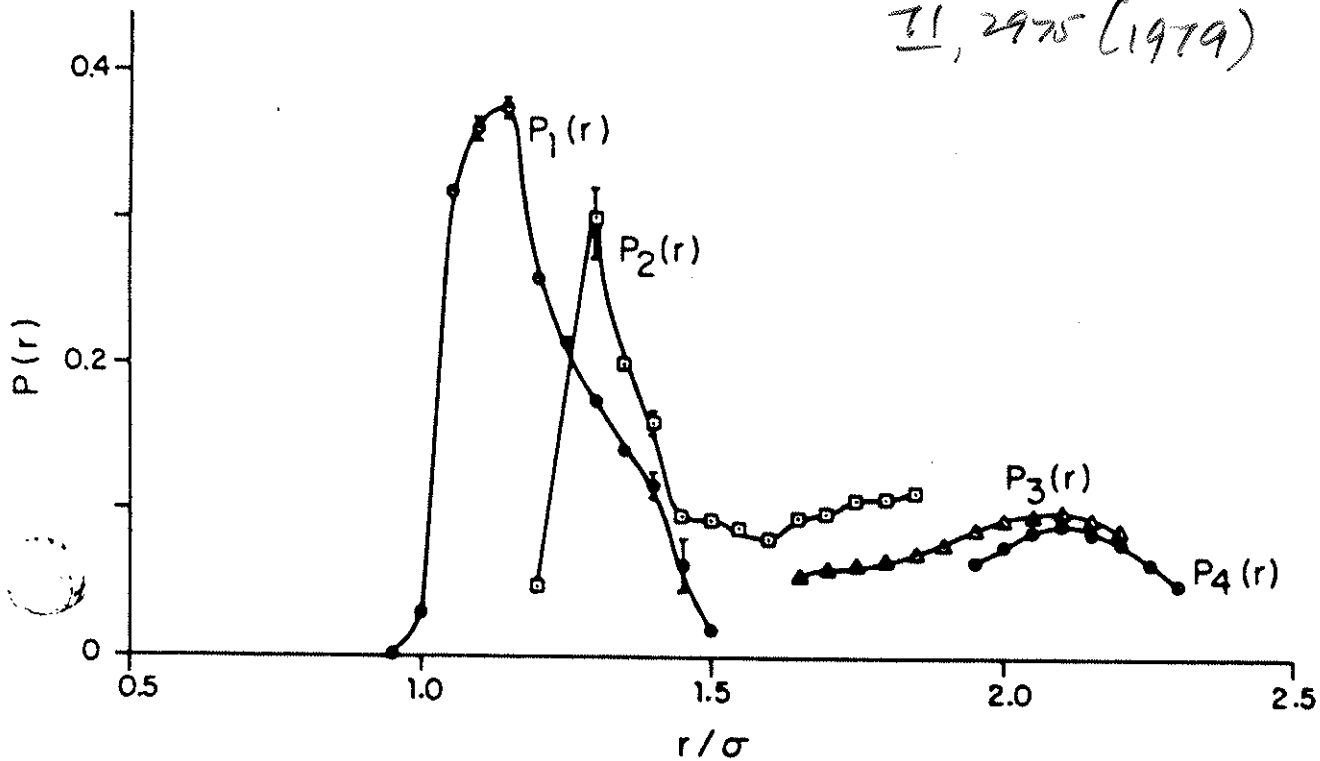


FIG. 1. The interionic potential of mean force at $\rho^* = 0.8$, $\mu^* = 1.0$ and $q^{*2} = 188.0$. The primitive model result assumes that $\epsilon = 7.7996$.

Example of use of overlapping ranges

C. Pangali, M. Rao, B.J. Berne, J. Chem. Phys. 71, 2975 (1979)



J. Chem. Phys., Vol. 71, No. 7, 1 October 1979

Monte Carlo simulation of the hydrophobic interaction

How do the overlapping ranges of r work?

- Use different weighting functions for each narrow range of r values
- Ranges should overlap
- When all done, piece the results for each window together somehow

after piecing together :

Pangali, Rao, and Berne: Monte Carlo :

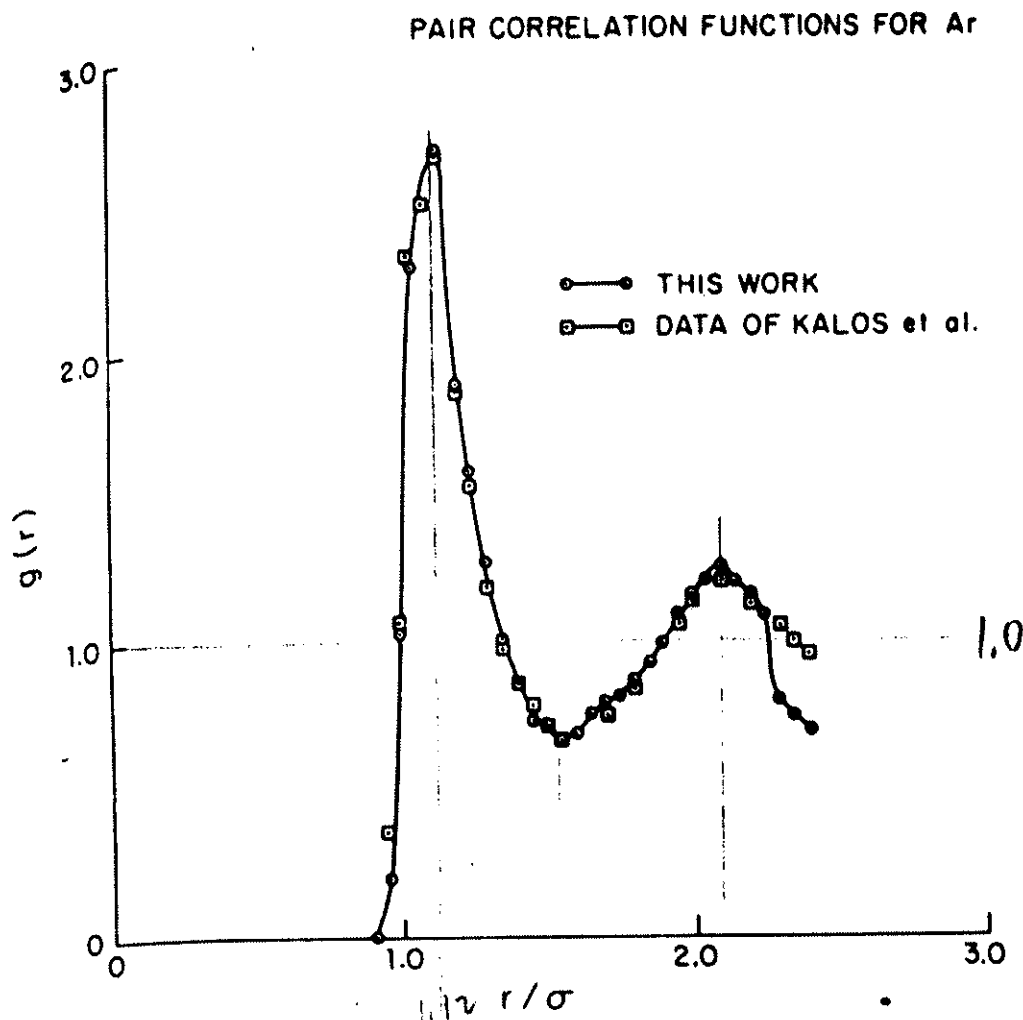


FIG. 2. The pair correlation function obtained by matching the $P_i(r)$'s of Fig. 1. For comparison with the exact result we show the data of Kalos *et al.* (Ref. 11).

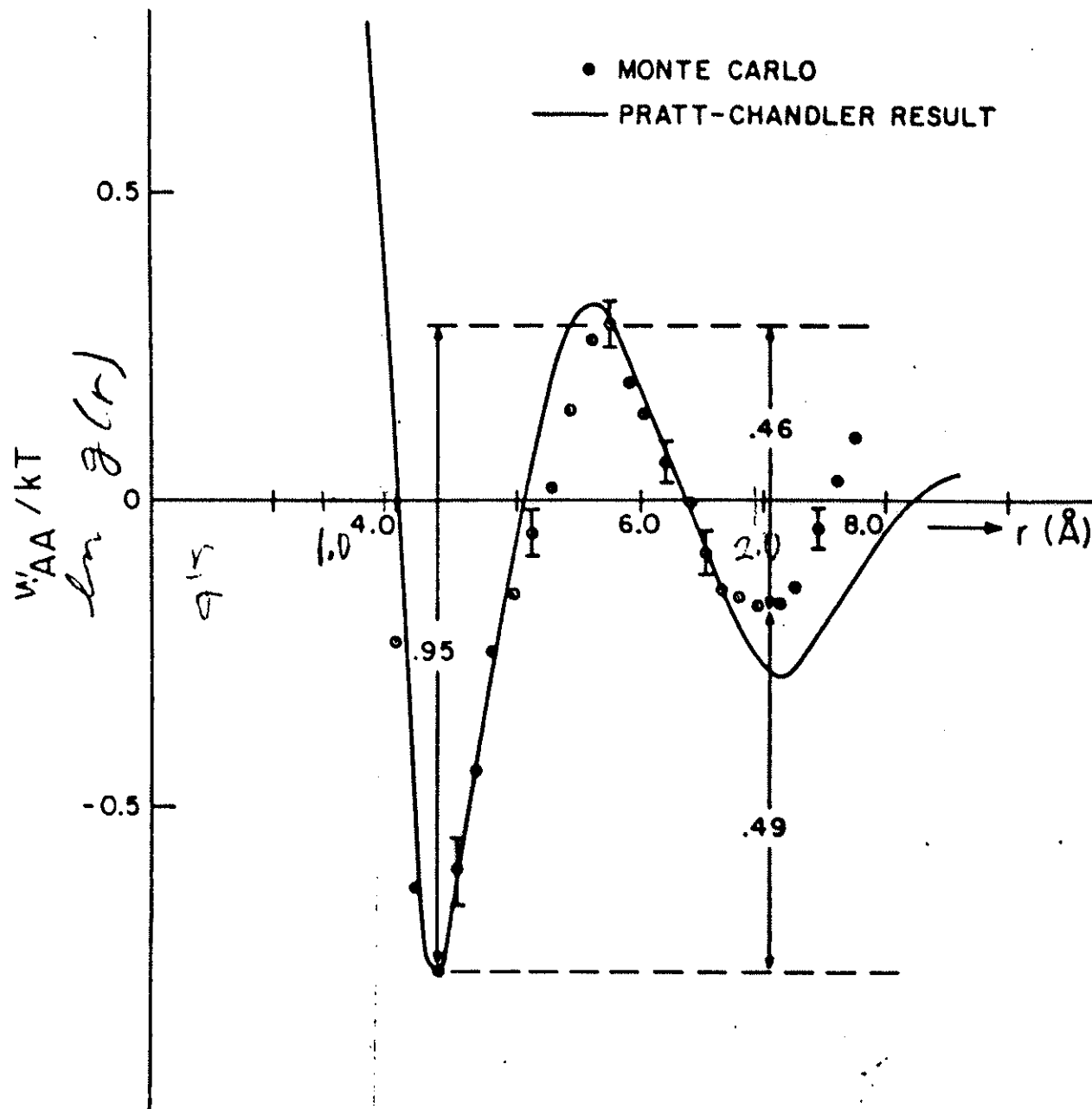


FIG. 3. The potential of mean force obtained from a simulation of two Lennard-Jones particles dissolved in 214 ST2 water molecules is shown in circles. Solid line shows the Pratt-Chandler result.

Pangali, Rao,
 and Berne

After piecing together:
H. Resat, M. Mezei, J. A. McCammon *J. Phys. Chem.*
100, 1426 (1996)

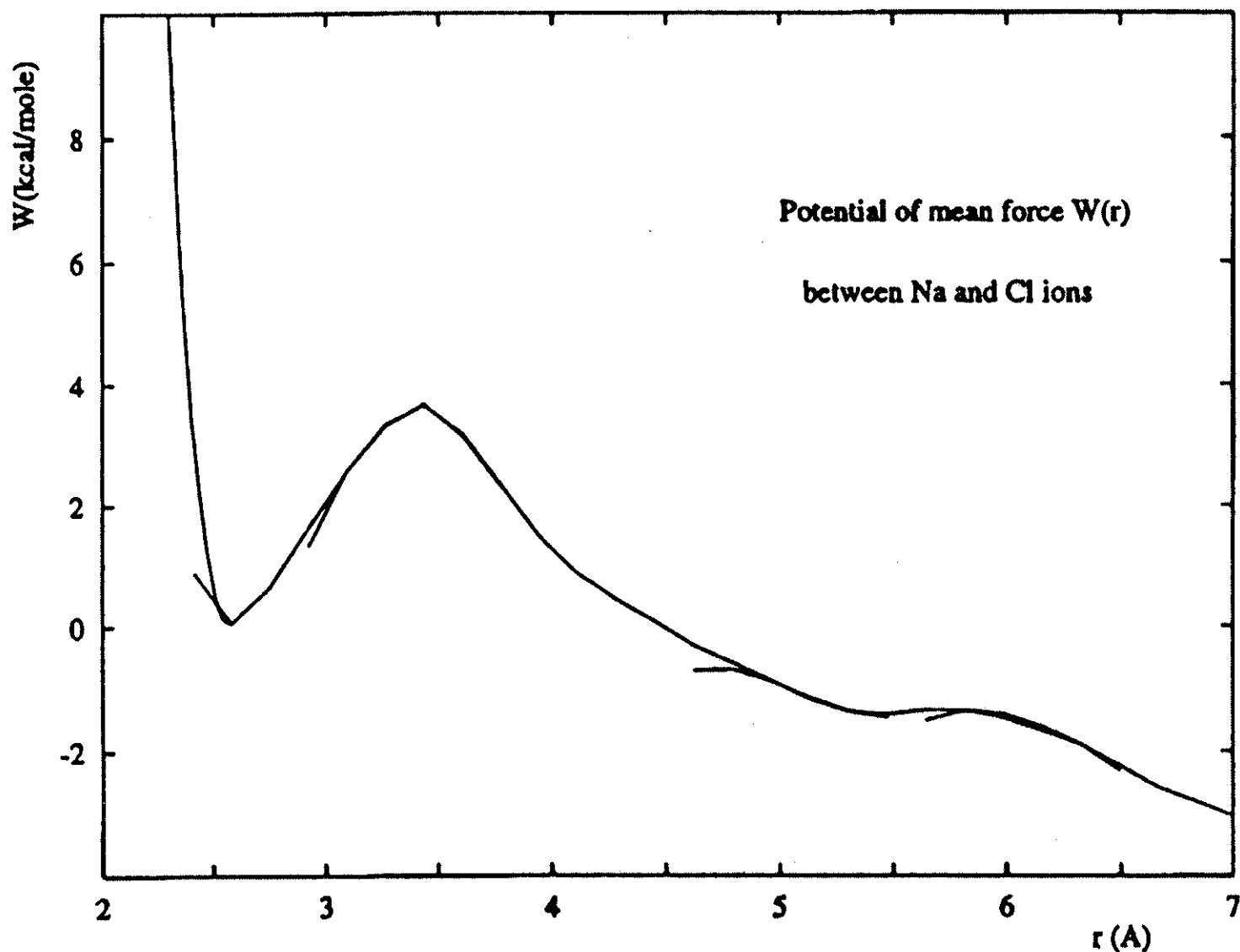


Figure 4. Potential of mean force as a function of the distance between sodium and chloride ions. Pmf and the distance are in kcal/mol and in angstroms, respectively.

Other weighting functions used:

$$W = e^{-B_W/kT}$$

where B_W is some functional form with adjustable parameters.

Find the form of W by trial and error:

(a) Assume a trial W . Run a short simulation and examine the regions sampled.

(b) Modify W to give larger weight to the undersampled or unsampled regions. Run a simulation with this new W . Examine the regions sampled.

(c) Keep doing (b) until satisfied.

How to match the different ranges or regions?

If $W = e^{-B_W/kT}$ form is used, with whatever functional form for B_W , this is equivalent to having a modified potential energy function that is changing with the range. Since only differences in potential energy are used in the sequence of trials, results from calculations in different ranges are undetermined up to a NORMALIZATION FACTOR. Determine the normalization factor in such a way as to obtain the best match in the P in overlapping regions.