Parallel MC/MD algorithms

이인호
한국표준과학연구원 (KRISS)

2nd KIAS CAC Summer School on Parallel Computing

고등과학원 2011년 6월30일-7월2일

MPI/FORTRAN 90

RS/6000 SP: Practical MPI Programming

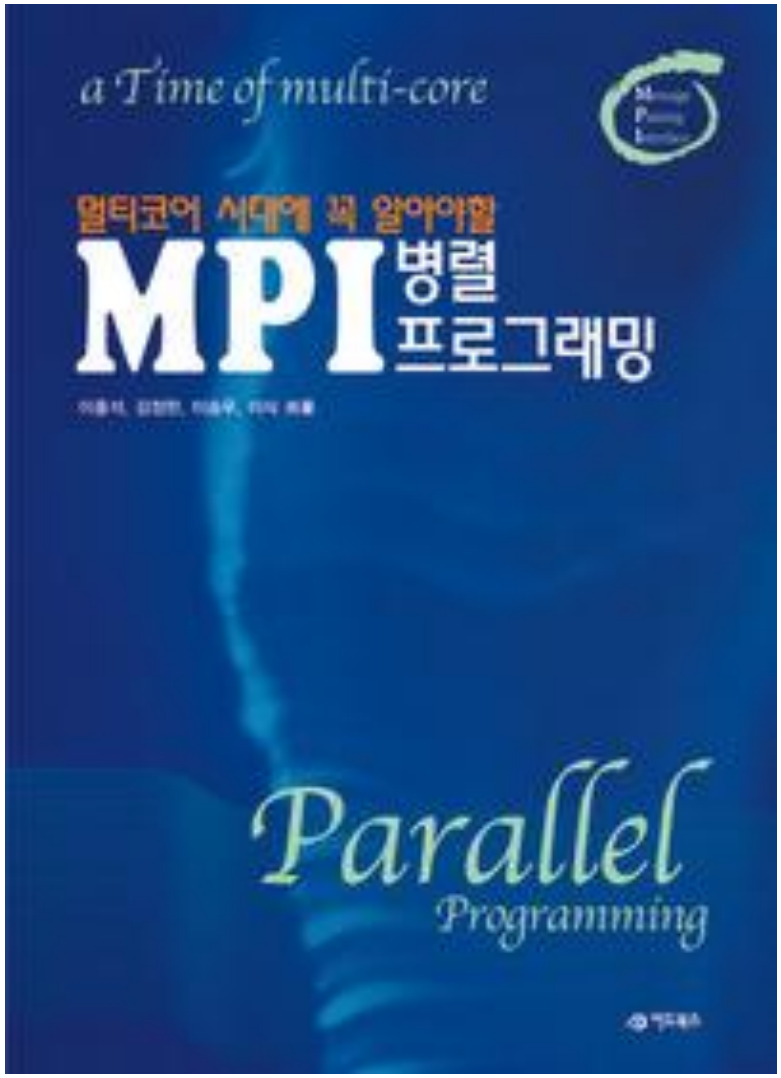
An IBM Redbooks publication

Yukiya Aoyama and Jun Nakano

http://www.redbooks.ibm.com/abstracts/sg245380.html

포트란 및 **MPI** 레퍼런스, 인터넷

KRISS



이홍석, 김정한, 이승우, 이식

KRISS

FORTRAN 90

Fortran 90

A Conversion Course for Fortran 77 Programmers (The University of Manchester)

The University of Liverpool

Fortran 90 course notes

두 개의 포트란 90 레퍼런스, 인터넷

KRISS

http://incredible.egloos.com

KRISS

Monte Carlo/molecular dynamics

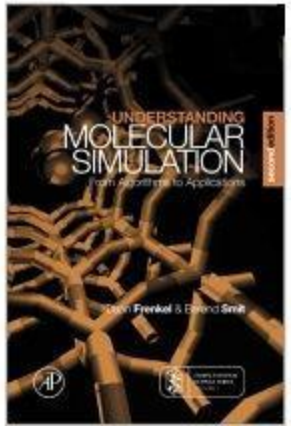
Understanding Molecular Simulation
From Algorithms to Applications

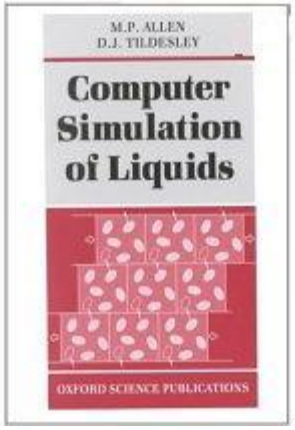
Daan Frenkel and Berend Smit
Academic Press 2002

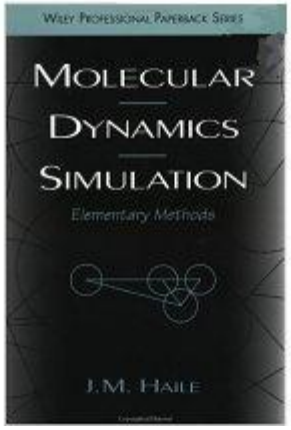
References

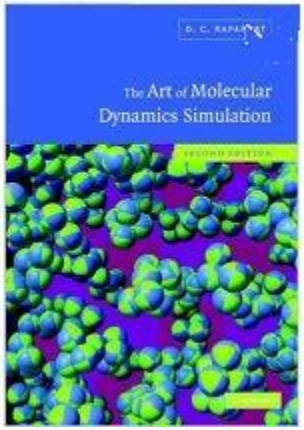
- Allen and Tildesley, *Computer simulation of liquids*
- Haile, *Molecular dynamics simulation*
- Frenkel and Smit, *Understanding molecular simulation*
- Landau and Binder, *A guide to Monte Carlo simulations in statistical physics*
- Rapaport, *The art of molecular dynamics simulation*
- Newman and Barkema, *Monte Carlo methods in statistical physics*
- http://dasher.wustl.edu/tinker
- http://www.ccp5.ac.uk/
- http://www.expasy.org/swissmod/course/course-index.htm

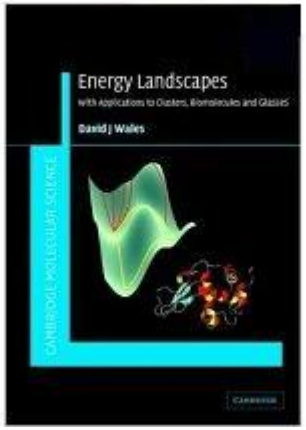
KRISS

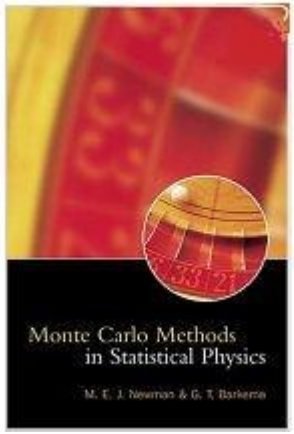

UNDERSTANDING
MOLECULAR
SIMULATION
second edition


M.P. ALLEN
D.J. TILDESLEY
Computer
Simulation
of Liquids
OXFORD SCIENCE PUBLICATIONS


WILEY PROFESSIONAL PAPERBACK SERIES
MOLECULAR
DYNAMICS
SIMULATION
Elementary Methods
J.M. HAILE


The Art of Molecular
Dynamics Simulation

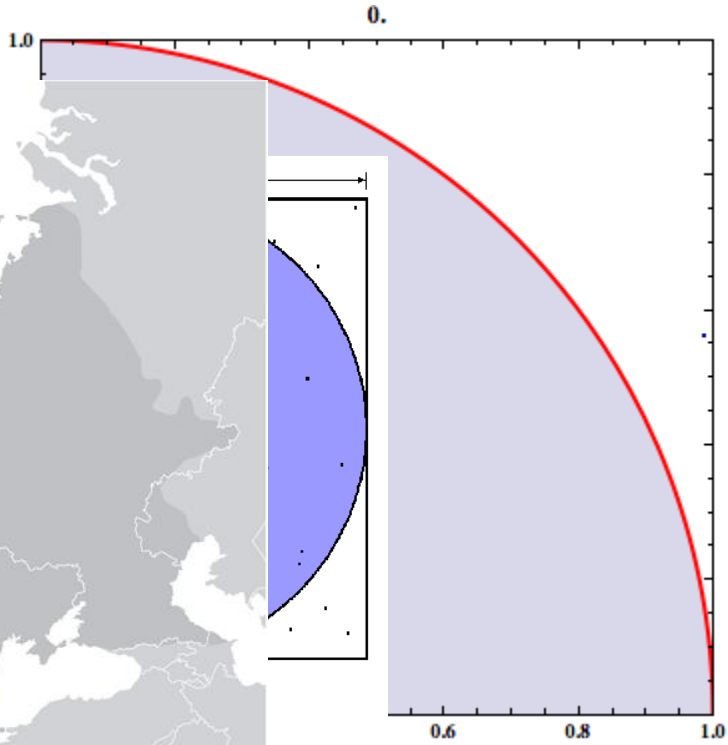

Energy Landscapes
David J Wales


Monte Carlo Methods
in Statistical Physics
M. E. J. Newman & G. T. Barkema

KRISS

컴퓨터 시뮬레이션

Manhattan Project (1942-1946)
Monte Carlo simulation



Monaco
Grace Patricia Kelly

KRISS

Monte Carlo

- http://en.wikipedia.org/wiki/Monte_Carlo

neutron diffusion

Metropolis method

Metropolis, Rosenbluth, Teller (1953) method:

- Move from s to s' with probability $T(s \rightarrow s') =$ constant
- Accept with move with probability:

$$a(s \rightarrow s') = \min [\ 1\ ,\ \exp \{\ -\ (E(s')-E(s))/(k_B T)\ \}\ ]$$

- Repeat many times

Given ergodicity, the distribution of s will be the canonical distribution:
$\Pi(s) = \exp(-E(s)/k_B T)/Z$

Parallel random number

Parallel

KRISS

Metropolis Algorithm

There are several Markov chain Monte Carlo algorithms. One of the most popular and simple algorithms, applicable to a wide class of problems, is the *Metropolis algorithm*.

In summary, new solutions are generated near the oldest best solution. The new solution becomes the best solution if it scores better than the old best solution, or a random function accepts it. At large temperatures, the random function is more forgiving and less-then-better solutions can be accepted as the next best solution.

Move/sampling

```
Program mc


Do icycle=1,ncycle
    call mc_move
    call sample
Enddo


End Program mc
```

pseudocode

```
Subroutine mc_move

  i=int(ranmar()*nparticles)+1
  call x_energy(x(i),e1)
  xnew=x(i)+(ranmar()-0.5)*deltax
  call x_energy(xnew,e2)
  If(ranmar() <= min(1.,exp(-beta*(e2-e1))) x(i)=xnew

End Subroutine mc_move
```

script

maximization of F = minimization of (-F)

```
j=int(ranmar()*nparticle)+1
```

```
if( ranmar() < min(1.d0, exp(-dell) ))
```

```
      subroutine rmarin(ij,kl)
!  This subroutine and the next function generate random numbers. See
!  the comments for SA for more information. The only changes from the
!  orginal code is that (1) the test to make sure that RMARIN runs first
!  was taken out since SA assures that this is done (this test didn't
!  compile under IBM's VS Fortran) and (2) typing ivec as integer was
!  taken out since ivec isn't used. With these exceptions, all following
!  lines are original.

! This is the initialization routine for the random number generator
!    RANMAR()
! NOTE: The seed variables can have values between:    0 <= IJ <= 31328
!                                                      0 <= KL <= 30081
      real u(97), c, cd, cm
      integer i97, j97
      common /raset1/ u, c, cd, cm, i97, j97
      if( ij .lt. 0  .or.  ij .gt. 31328  .or.  &
         kl .lt. 0  .or.  kl .gt. 30081 ) then
         print '(a)', ' The first random number seed must have a value  &
     & between 0 and 31328'
         print '(a)',' The second seed must have a value between 0 and  &
     & 30081'
           stop
      endif
      i = mod(ij/177, 177) + 2
      j = mod(ij    , 177) + 2
      k = mod(kl/169, 178) + 1
      l = mod(kl,     169)
      do 2 ii = 1, 97
        s = 0.0
        t = 0.5
        do 3 jj = 1, 24
          m = mod(mod(i*j, 179)*k, 179)
          i = j
          j = k
          k = m
          l = mod(53*l+1, 169)
          if (mod(l*m, 64) .ge. 32) then
            s = s + t
          endif
          t = 0.5 * t
3       continue
        u(ii) = s
2     continue
      c = 362436.0 / 16777216.0
      cd = 7654321.0 / 16777216.0
      cm = 16777213.0 /16777216.0
      i97 = 97
      j97 = 33
      return
      end
```

```
function ranmar()
    real u(97), c, cd, cm
    integer i97, j97
    common /raset1/ u, c, cd, cm, i97, j97
      uni = u(i97) - u(j97)
      if( uni .lt. 0.0 ) uni = uni + 1.0
      u(i97) = uni
      i97 = i97 - 1
      if(i97 .eq. 0) i97 = 97
      j97 = j97 - 1
      if(j97 .eq. 0) j97 = 97
      c = c - cd
      if( c .lt. 0.0 ) c = c + cm
      uni = uni - c
      if( uni .lt. 0.0 ) uni = uni + 1.0
      ranmar = uni
    return
    end
```

Simulated annealing

- (SA) is a generic probabilistic metaheuristic for the global optimization problem of locating a good approximation to the global optimum of a given function in a large search space.

simulated annealing

Definition: A technique to find a good solution to an optimization problem by trying random variations of the current solution. A worse variation is accepted as the new solution with a probability that decreases as the computation proceeds. The slower the cooling schedule, or rate of decrease, the more likely the algorithm is to find an optimal or near-optimal solution.

Formally, simulated annealing is a *Markov Chain Monte Carlo* method.

That is, the probability of jumping from one point to an other depends only on the last point and not on the entire previous history (this is the peculiar property of a Markov chain).

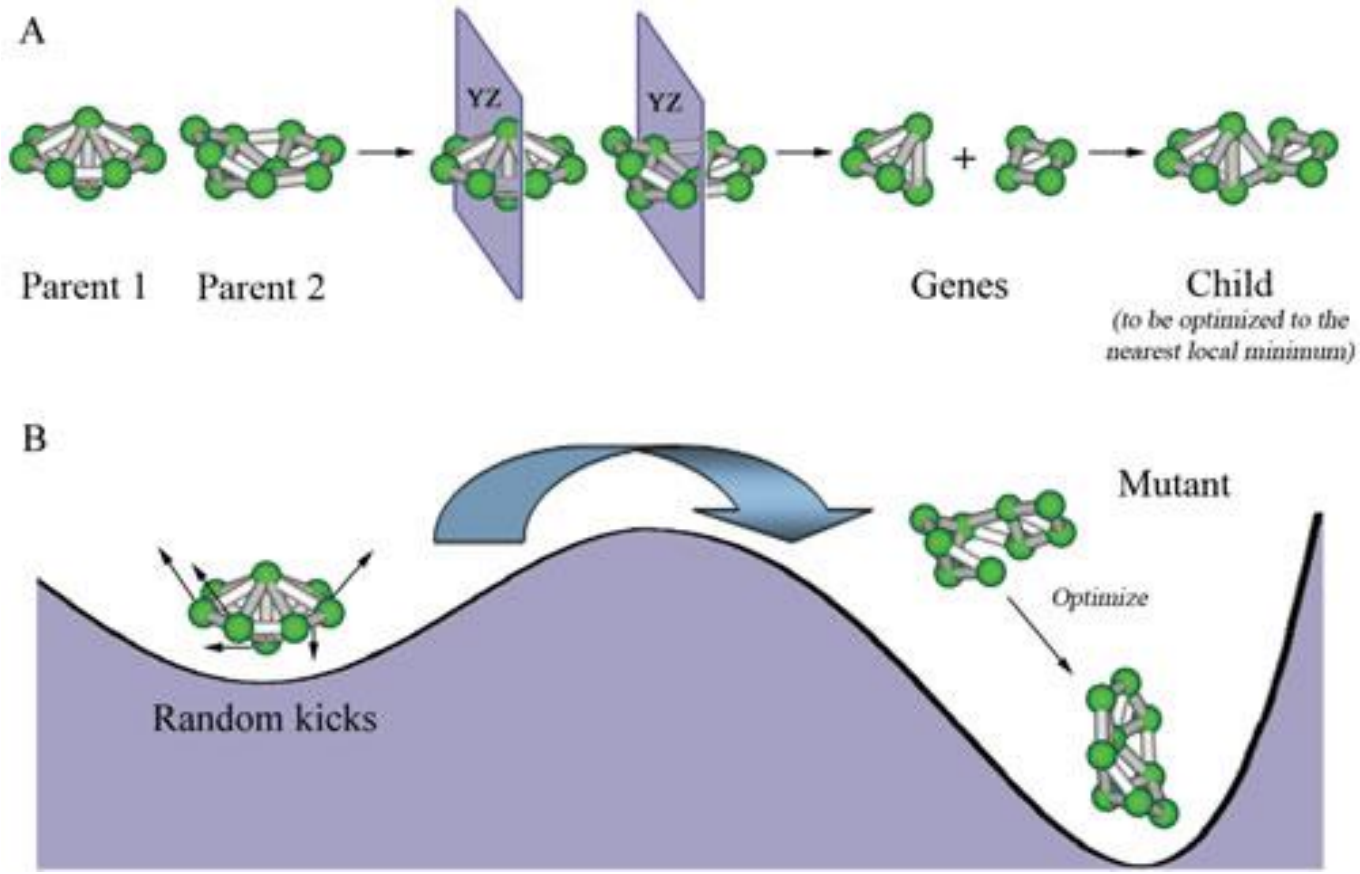
SA run; example

Intermediate results before next temperature reduction

Current T: 0.9990E-3
Min function value so far: 5.856412345678
Total moves : 51200
Downhill : 20456
Accepted uphill : 5005
Rejected uphill : 25739
Trials out of bounds: 166
New minima this temperature : 0

Genetic algorithm
Parents/children

A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution.



A. N. Alexandrova, A. I. Boldyrev. J. Chem. Theory and Comput. 2005, 1, 566-580

KRISS

Conformational Space Annealing (CSA)

- Based on genetic algorithm(GA) → maintain a pool of conformations (bank) and new conformations are obtained by mating and mutation (small perturbation)
- Consider only conformational space of local minima (as in MCM)
- Diversity of the bank is directly controlled by introducing the concept of distance between the conformations
- Narrows the search to smaller region while maintaining diversity of sampling
- Key idea : GA + annealing in conformational space

Phys. Rev. Lett. **91**, 080201 (2003)

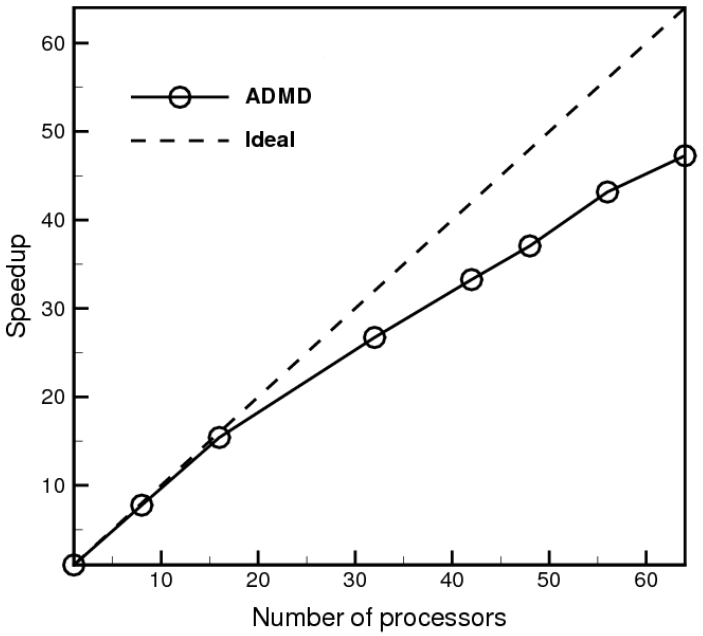
speedup

$$S_p = \frac{T_1}{T_p}$$

***p* is the number of processors.**

***T*$_p$ is the execution time of the algorithm with *p* processors.**

Not CPU time, but wall-clock time reference



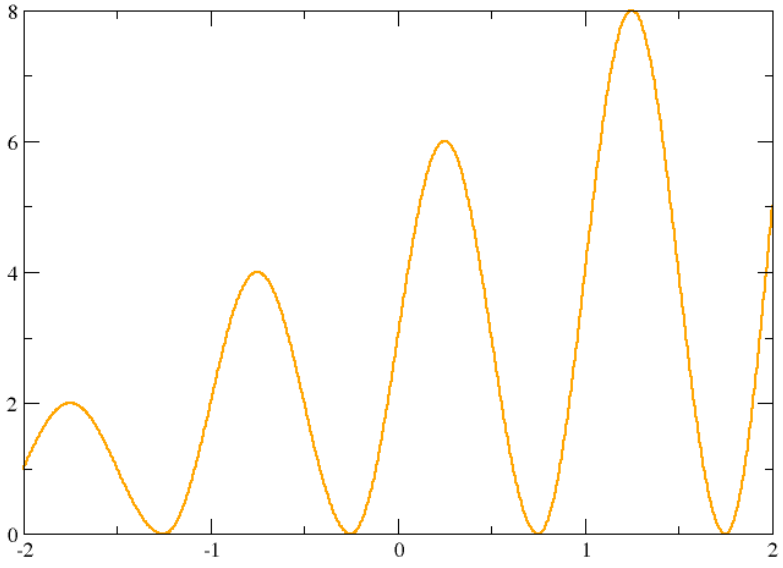
http://en.wikipedia.org/wiki/Speedup

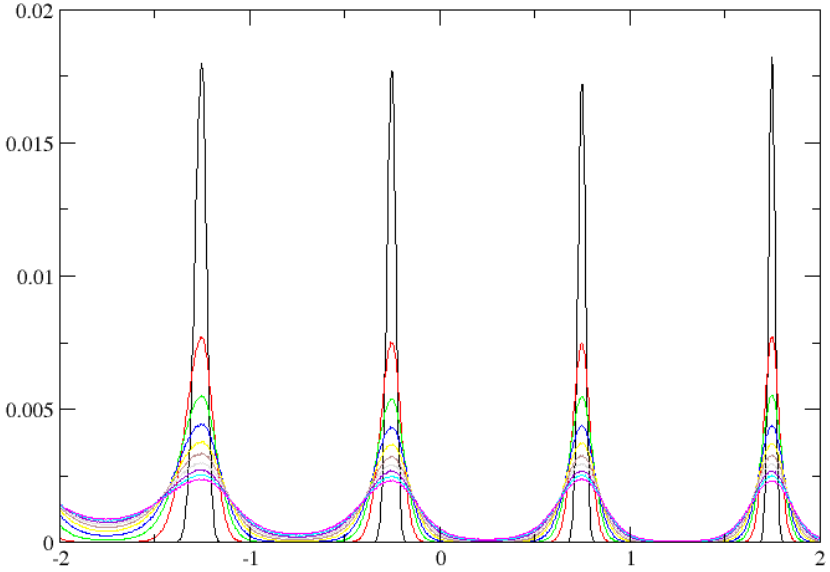
KRISS

replica

http://en.wikipedia.org/wiki/File:Xavi_Hern%C3%A1ndez_-_002.jpg

REMC





Frenkel and Smit, p. 391

min(1, exp((b_i-b_j)(U_i-U_j))), typo in text

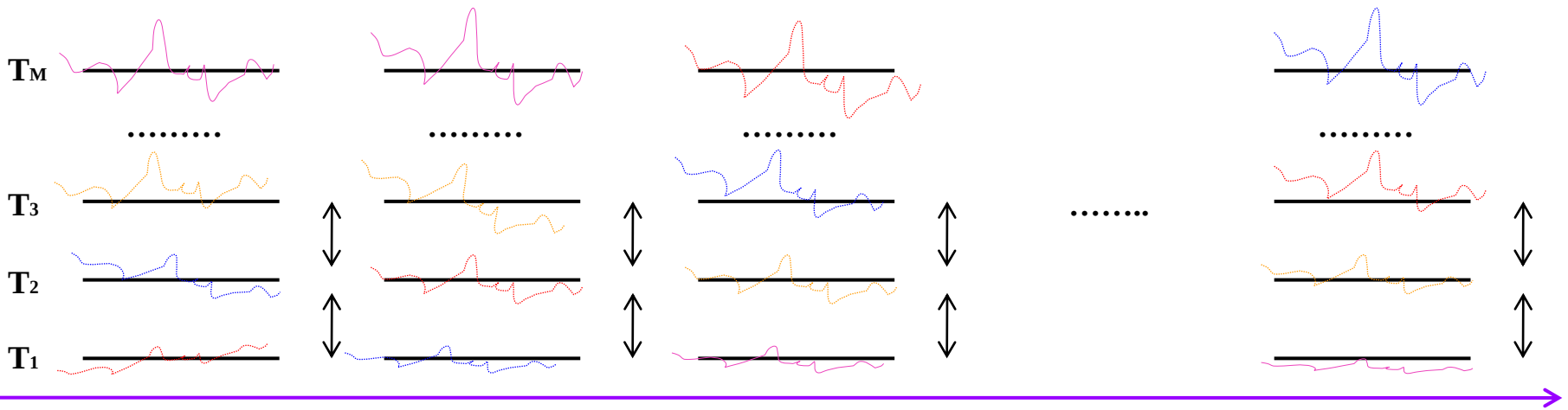
KRISS

Replica-exchange Monte Carlo

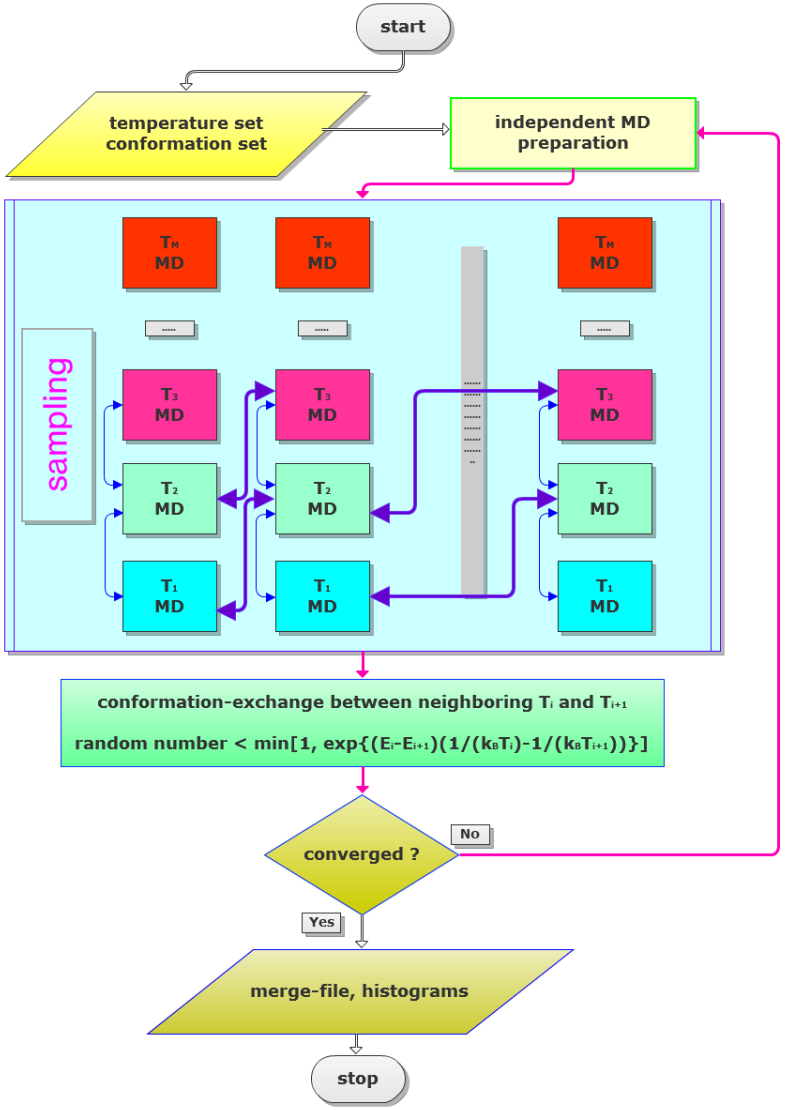
$$p = min\left\{1, \frac{\exp\{-E_j/(k_B T_i) - E_i/(k_B T_j)\}}{\exp\{-E_i/(k_B T_i) - E_j/(k_B T_j)\}}\right\}$$
$$= min\left\{1, \exp\{(1/(k_B T_i) - 1/(k_B T_j))(E_i - E_j)\}\right\}$$

***Chemical Physics Letters* 314, 141 (1999).** ***Biophys. J.* 84, 775 (2003).**

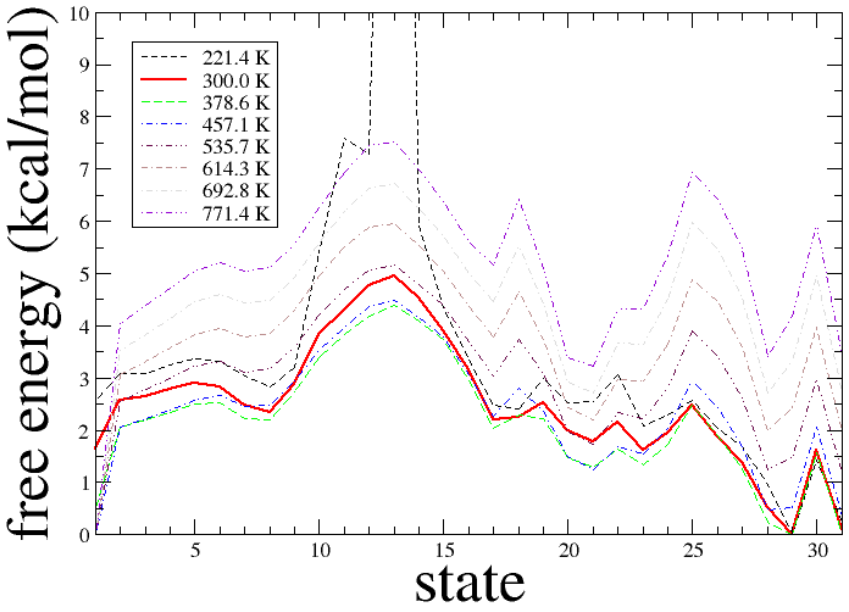
Exchanges of conformations are tried periodically.

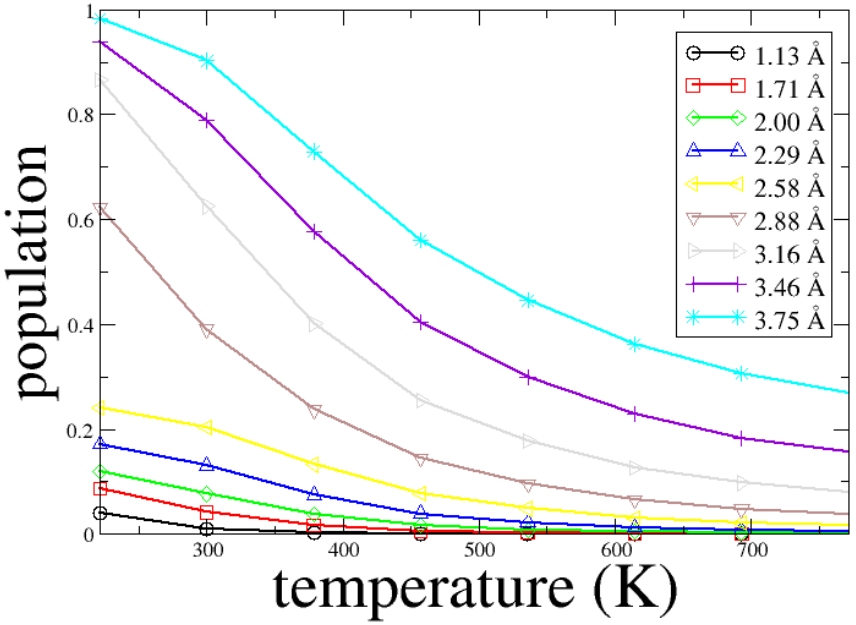


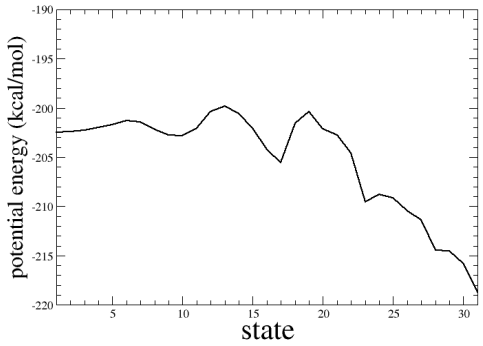
Replica, multiple simulations, and temperatures



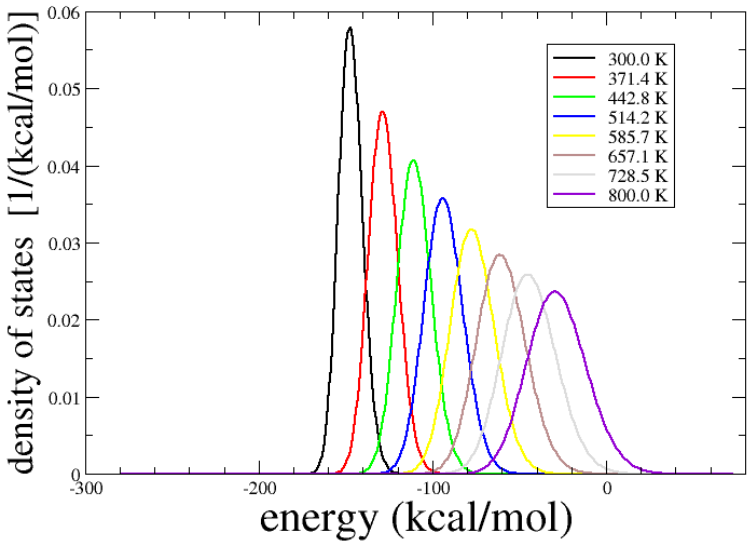
met-enkephalin

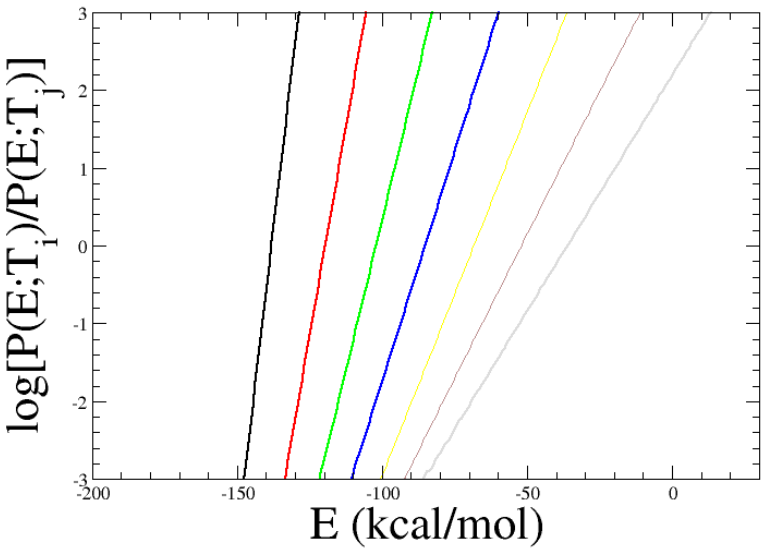






KRISS


density of states [1/(kcal/mol)]
energy (kcal/mol)
300.0 K
371.4 K
442.8 K
514.2 K
585.7 K
657.1 K
728.5 K
800.0 K
0.06
0.05
0.04
0.03
0.02
0.01
0
-300
-200
-100
0


log[P(E;T_i)/P(E;T_j)]
E (kcal/mol)
3
2
1
0
-1
-2
-3
-200
-150
-100
-50
0

KRISS

script

```
      subroutine equal_load(n1,n2,nproc,myid,istart,ifinish)
!     Written by In-Ho Lee, KRISS, September (2006)
      implicit none
      integer nproc,myid,istart,ifinish,n1,n2
      integer iw1,iw2
      iw1=(n2-n1+1)/nproc ; iw2=mod(n2-n1+1,nproc)
      istart=myid*iw1+n1+min(myid,iw2)
      ifinish=istart+iw1-1 ; if(iw2 > myid) ifinish=ifinish+1
!     print*, n1,n2,myid,nproc,istart,ifinish
      if(n2 < istart) ifinish=istart-1
      return
      end
```

```
      n1=1 ; n2=ntemper*mltpl
      call equal_load(n1,n2,nproc,myid,jstart,jfinish)
```

Embarrassingly parallel

- little or no communication of results between tasks

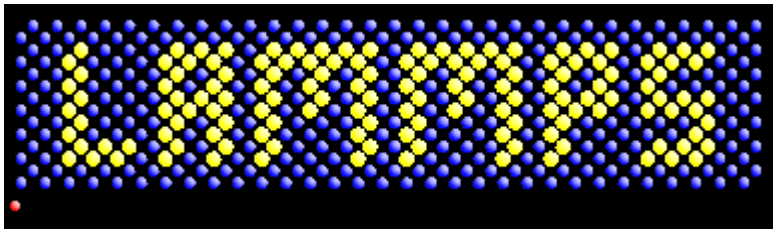
Molecular dynamics

- (MD) is a computer simulation method for investigating the physical movements of atoms and molecules.

Number of refereed papers mentioning "molecular dynamics"

1960	1970	1980	1990	2000	2006
1	16	114	484	3597	6253

http://lammps.sandia.gov/

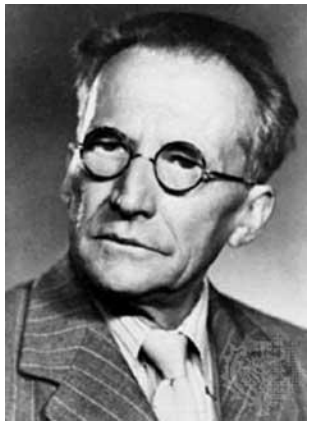


LAMMPS is a classical molecular dynamics code, and an acronym for Large-scale Atomic/Molecular Massively Parallel Simulator.

http://code.google.com/p/gpulammps/

Born-Oppenheimer approximation

Hydrogen?
electronic degrees of freedom



In 1923, Born and Oppenheimer noted that nuclei are much heavier than electrons, and move on a time scale which is about two order of magnitude longer than that of the electrons.

Force-field

$$V(\{\vec{R}_i\}) = Min_{\{\psi_j(\vec{r})\}}\Phi(\{\psi_j(\vec{r})\},\{\vec{R}_i\})$$

$$m_i\ddot{\vec{R}} = -\frac{\partial V(\{\vec{R}_i\})}{\partial \vec{R}_i}$$

Newton (1642-1727)

KRISS

Similar situation in *ab initio* MD

$$L = \sum_{i=1}^{N} \frac{m_i}{2} \dot{\vec{R}}_i^{\,2} - V_{DFT}(\{\vec{R}_i\}, \{\psi_j(\vec{r})\}) + \frac{\mu}{2} \sum | \dot{\psi}_j(\vec{r}) |^2 + \sum \lambda_{mn} \{ \langle \psi_m(\vec{r}) \mid \psi_n(\vec{r}) \rangle - \delta_{mn} \}$$

holonomic

$$\psi_j(t+\Delta) = 2\psi_j(t) - \psi_j(t-\Delta) - \frac{\Delta^2}{\mu}(H\psi_j)(t) + \sum_m \lambda_{jm}\psi_m(t)$$
$$\equiv \tilde{\psi}_j(t+\Delta) + \sum_m \lambda_{jm}\psi_m(t)$$

Iterative orthogonalization procedure ←SHAKE

Cf. Gram-Schmidt orthogonalization

Phys. Rev. Lett. **55**, 2471 (1985)

KRISS

Molecular dynamics

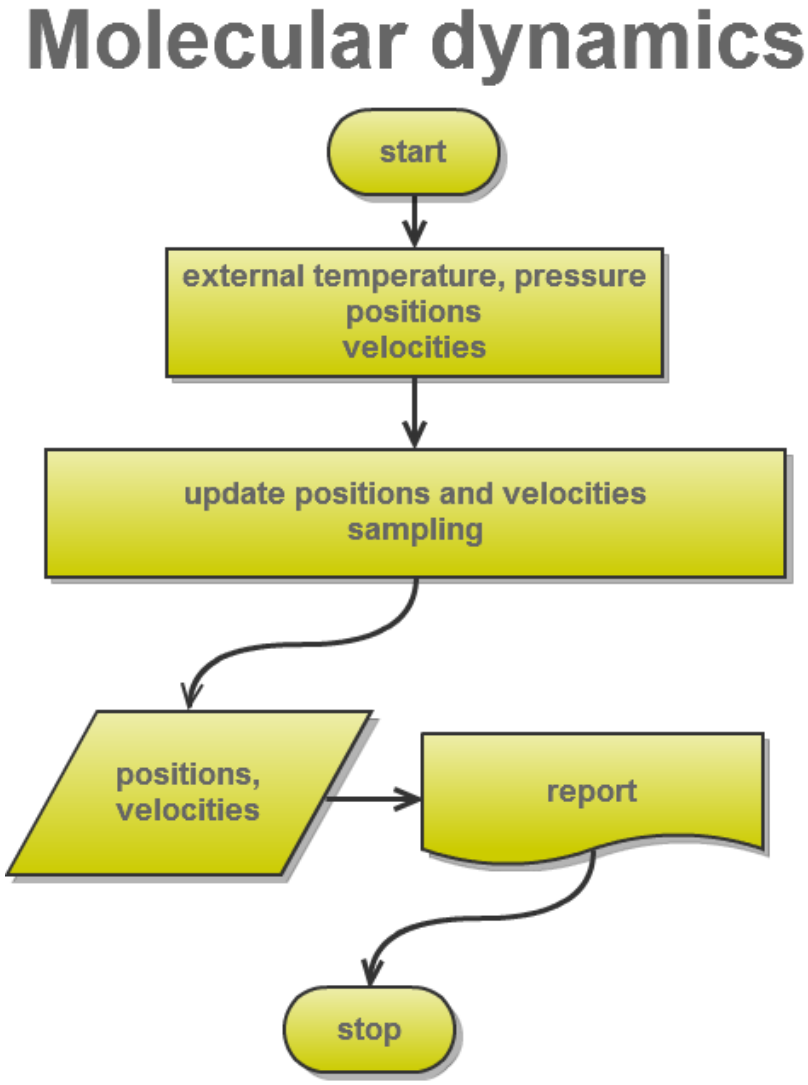
$$m_i \ddot{\vec{R}}_i = -\frac{\partial V(\{\vec{R}_i\})}{\partial \vec{R}_i}$$

$$\{\vec{R}_i(0)\}, \quad \{\dot{\vec{R}}_i(0)\}$$

Initial value problem : unknown!

Maxwell-Boltzmann

Integration



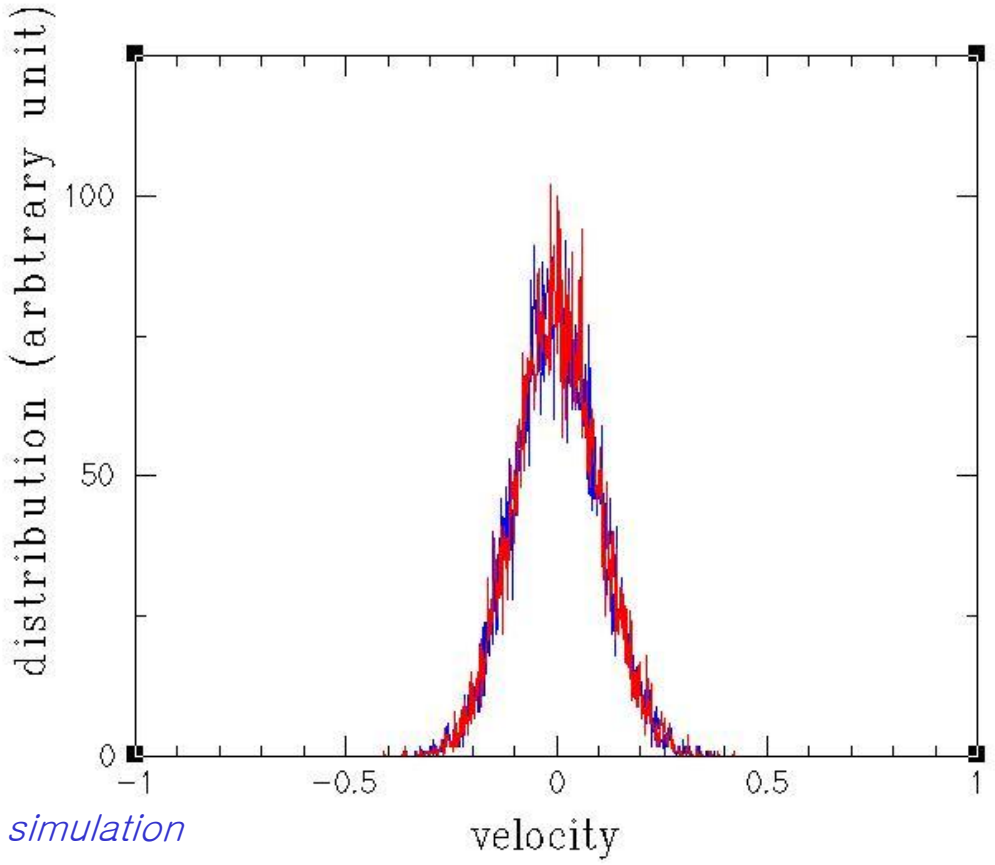
KRISS

Gaussian distribution

Two independent experiments
Average=0, sigma=0.1
10000 points

```
xl0=0.0, sigma=sqrt(temperature/mass)
subroutine gauss(sigma,xl0,xl)
implicit none
real*8 sigma,xl0,xl
real*8 tmp
real*8 r,v1,v2
r=2.d0
do while (r >= 1.d0)
call random_number(tmp)
v1=2.*tmp-1.d0
call random_number(tmp)
v2=2.*tmp-1.d0
r=v1**2+v2**2
enddo
xl=v1*sqrt(-2.d0*log(r)/r)
xl=xl0+sigma*xl
return
end
```

tmp is a uniform random number within [0,1]



Frenkel and Smit, *Understanding molecular simulation*

KRISS

pseudocode

```
Program md
  call init
Do while(t <= tmax)
  call force(f,e)
  call integrate(f,e)
  t=t+deltat
  call sample
Enddo
End program md
```

Verlet algorithm

$$\vec{r}(t+\Delta) = \vec{r}(t) + \vec{v}(t)\Delta + \frac{1}{2}\vec{a}(t)\Delta^2 + \frac{1}{6}\vec{b}\Delta^3 + O(\Delta^4)$$

$$\vec{r}(t-\Delta) = \vec{r}(t) - \vec{v}(t)\Delta + \frac{1}{2}\vec{a}(t)\Delta^2 - \frac{1}{6}\vec{b}\Delta^3 + O(\Delta^4)$$

$$\vec{r}(t+\Delta) = 2\vec{r}(t) - \vec{r}(t-\Delta) + \vec{a}(t)\Delta^2$$

$$\vec{a}(t) = -\frac{1}{m}\frac{\partial V(\{\vec{r}\})}{\partial \vec{r}(t)}$$

$$\vec{r}(t+\Delta) = 2\vec{r}(t) - \vec{r}(t-\Delta) - \frac{\Delta^2}{m}\frac{\partial V(\{\vec{r}\})}{\partial \vec{r}(t)}$$

Shadow property

Time reversal symmetry

$$\vec{v}(t) = \frac{\vec{r}(t+\Delta) - \vec{r}(t-\Delta)}{2\Delta}$$

Phys. Rev. 159, 98 (1967)

An estimation

$\{q(t), p(t)\} \rightarrow \{q(t+\Delta), p(t+\Delta)\}$

Jacobian = 1; area preserving

KRISS

Velocity Verlet algorithm

$$\vec{r}(t+\Delta) = \vec{r}(t) + \vec{v}(t)\Delta + \frac{\vec{f}(t)}{2m}\Delta^2$$

$$\vec{v}(t+\Delta) = \vec{v}(t) + \frac{\vec{f}(t)+\vec{f}(t+\Delta)}{2m}\Delta$$

$$\vec{r}(t+2\Delta) = \vec{r}(t+\Delta) + \vec{v}(t+\Delta)\Delta + \frac{\vec{f}(t+\Delta)}{2m}\Delta^2$$

$$\vec{r}(t) = \vec{r}(t+\Delta) - \vec{v}(t)\Delta - \frac{\vec{f}(t)}{2m}\Delta^2$$

$$\vec{r}(t+2\Delta) + \vec{r}(t) = 2\vec{r}(t+\Delta) + \{\vec{v}(t+\Delta) - \vec{v}(t)\}\Delta + \frac{\vec{f}(t+\Delta) - \vec{f}(t)}{2m}\Delta^2$$

$$\vec{r}(t+2\Delta) + \vec{r}(t) = 2\vec{r}(t+\Delta) + \frac{\vec{f}(t+\Delta)}{m}\Delta^2$$ Verlet algorithm

KRISS

```
program md_andersen

call initial(temperature)
call force(f,energy)
t=0.d0
do while (t < t_max)
   call integrate(1,f,energy,temperature)
   call force(f,energy)
   call integrate(2,f,energy,temperature)
    t=t+dt
    call sample
end do
stop
end program md_andersen
```

```
subroutine integrate(switch,f,energy,temperature)

If (switch == 1) then
  x(:)=x(:)+ dt*v(:)+dt*dt*f(:)/2.
  v(:)=v(:)+dt*f(:)/2.

else if (switch == 2)then
  v(:)=v(:)+dt*f(:)/2.
  tempa=dot_product(v,v)/(nparticle*3./mass)
  sigma=sqrt(temp)
  do i=1,nparticle
     If(ranf() < nu*dt)then
     v(i)=gauss(sigma)
     end if
  end do
end if
return
end
```

zero mean and standard deviation sigma

velocity Verlet algorithm

ranf() is a uniform random number within [0,1]

Frenkel and Smit, *Understanding molecular simulation*

Where P(t;ν)dt is the probability that the next collision will take place in the interval [t,t+dt].

Start with an initial set of positions and velocities and integrate the equations of motion for a time Δt.
A number of particles are selected to undergo a collision with the heat bath.
The probability that a particle is selected in a time step of length Δt is νΔt.
If particle *i* has been selected to undergo a collision, its new velocity will be drawn from a Maxwell - Boltzmann distribution corresponding to the desired temperature T. All other particles are unaffected by this collision.

KRISS

Andersen thermostat

$$P_{MB}(p) = \left(\frac{1}{2\pi m k_B T}\right)^{3/2} \exp\left(-\frac{p^2}{2mk_BT}\right)$$

$$P(t;\nu) = \nu \exp(-\nu t)$$

Stochastic event

Frenkel and Smit, *Understanding molecular simulation*

KRISS

```
program md_andersen


call initial(temperature)
call force(f,energy)
t=0.
do while (t < t_max)
   call integrate(1,f,energy,temperature)
   call force(f,energy)
   call integrate(2,f,energy,temperature)
    t=t+dt
    call sample
end do
stop
end
```

```
subroutine integrate(switch,f,energy,temperature)


If (switch == 1) then
  x(:)=x(:)+ dt*v(:)+dt*dt*f(:)/2.
  v(:)=v(:)+dt*f(:)/2.

else if (switch == 2)then
  v(:)=v(:)+dt*f(:)/2.
  tempa=dot_product(v,v)/(nparticle*3./mass)
  sigma=sqrt(temp)
  do i=1,nparticle
     If(ranf() < nu*dt)then
     v(i)=gauss(sigma)
     end if
  end do
end if
return
end
```

Frenkel and Smit, *Understanding molecular simulation*

KRISS

Langevin Dynamics Simulation

The Langevin equation is a stochastic differential equation in which two force terms have been added to Newton's second law to approximate the effects of neglected degrees of freedom. One term represents a frictional force, the other a random force.

Langevin dynamics represents a coupling of the system to a heatbath and thus provides a means to control and maintain temperatures.

Boltzmann distribution

Collision frequency Simulates drags of the motion of the solute through the solvent.

$$m_i \ddot{\vec{R}}_i = \vec{f}_i - \gamma_i \dot{\vec{R}}_i + \vec{\xi}_i$$

$$\langle \vec{\xi} \rangle = \vec{0}$$

$$\langle \vec{\xi}_i(t) \cdot \vec{\xi}_i(s) \rangle = 6 k_B T \gamma_i \delta(t-s)$$

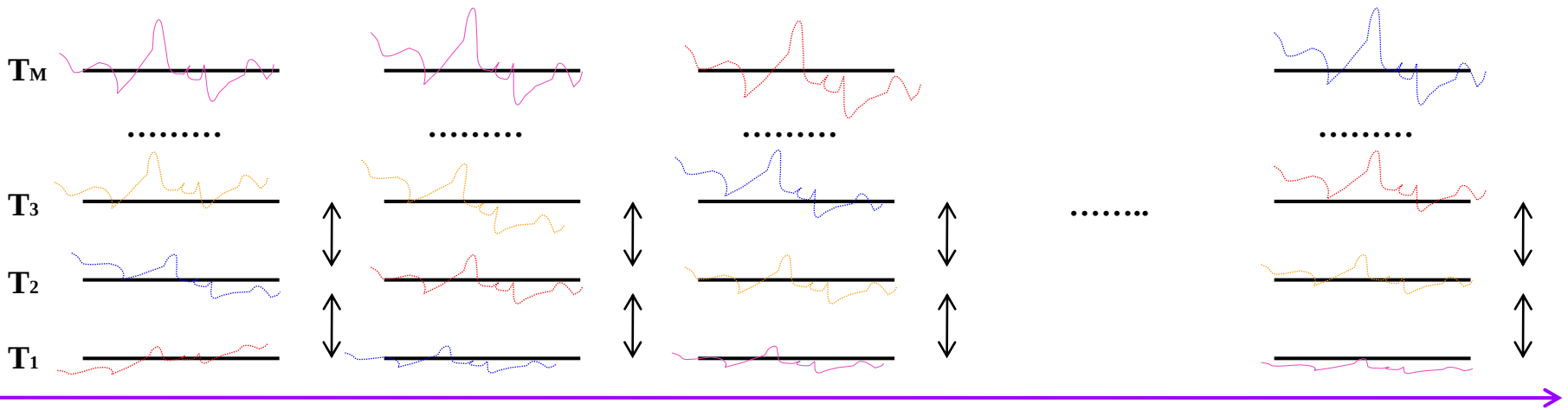
KRISS

Replica-exchange molecular dynamics

$$p = min\left\{1, \frac{\exp\{-E_j/(k_B T_i) - E_i/(k_B T_j)\}}{\exp\{-E_i/(k_B T_i) - E_j/(k_B T_j)\}}\right\}$$
$$= min\left\{1, \exp\{(1/(k_B T_i) - 1/(k_B T_j))(E_i - E_j)\}\right\}$$

***Chemical Physics Letters* 314, 141 (1999).** ***Biophys. J.* 84, 775 (2003).**

Exchanges of conformations are tried periodically.



Folding@Home Distributed Computing - Microsoft Internet Explorer

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

뒤로 검색 즐겨찾기 미디어

주소(D) http://folding.stanford.edu/ 이동 연결

Folding@home distributed computing

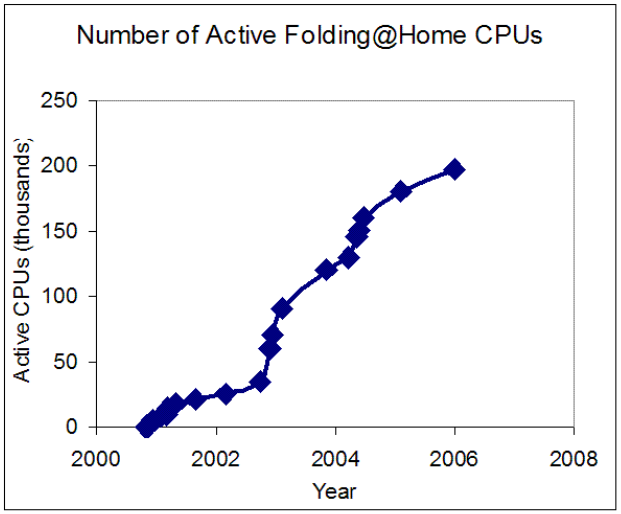
En Español

- Home
- Download
- FAQ
- Forum
- Help!
- Education
- News
- Stats
- Science
- Results
- Papers

Our goal: to understand protein folding, protein aggregation, and related diseases

What are proteins and why do they "fold"? Proteins a biology's workhorses -- its "**nanomachines**." Before prote can carry out their biochemical function, they remarkably assemble themselves, or "**fold**." The process of protein folding, while critical and fundamental to virtually all of biolo remains a mystery. Moreover, perhaps not surprisingly, whe proteins do not fold correctly (i.e. "misfold"), there can be serious effects, including many well known **diseases**, such Alzheimer's, Mad Cow (BSE), CJD, ALS, and Parkinson's disease.

What does Folding@Home do? Folding@Home is a distributed computing project which studies **protein foldin** misfolding, aggregation, and **related diseases**. We use nc



The points are milestones recorded in the news section. The line is meant to guide the eye -- there have been fluctuations in the number active of active CPUs inbetween milestones and they are not shown.

KRIS

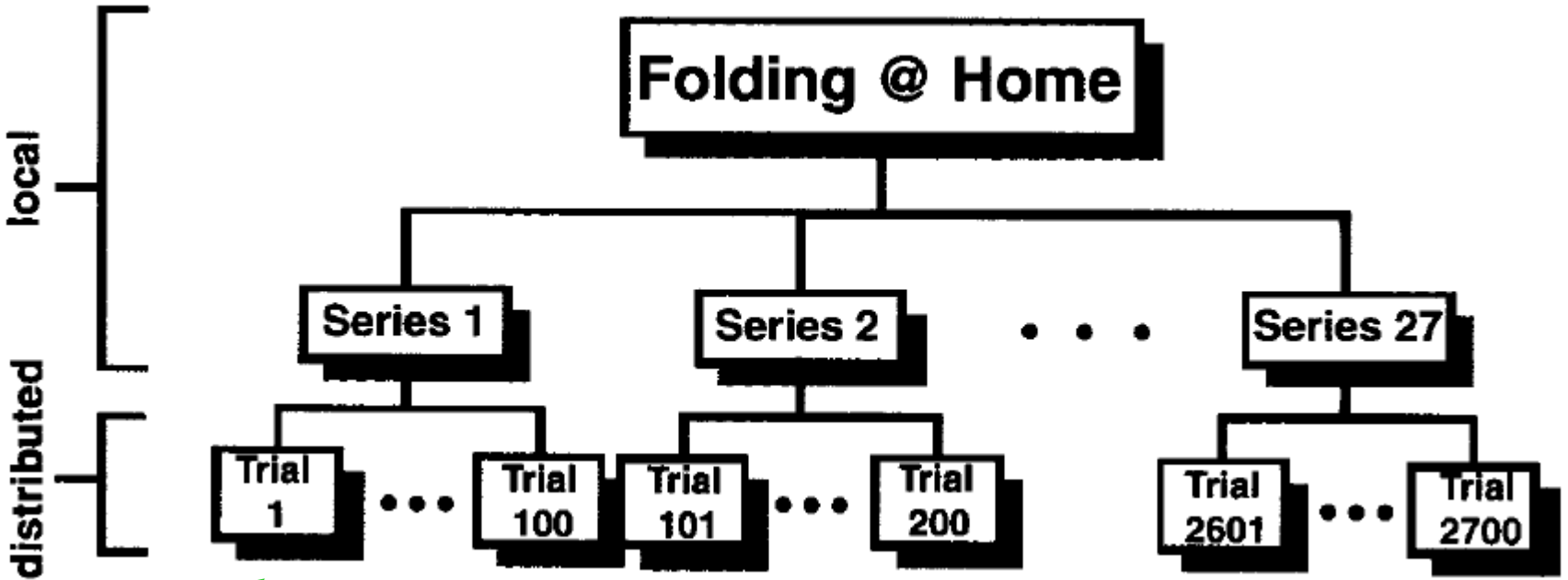
Search for Extra-Terrestrial Intelligence

$$\tau = \frac{N_{total}t}{N_{folded}}$$

$$= \frac{2700 \times 14 \text{ ns}}{8}$$

$$= 4725 \text{ ns}$$ Exp. 6000 ns

27 x 100 independent trial simulations 8/2700=0.003

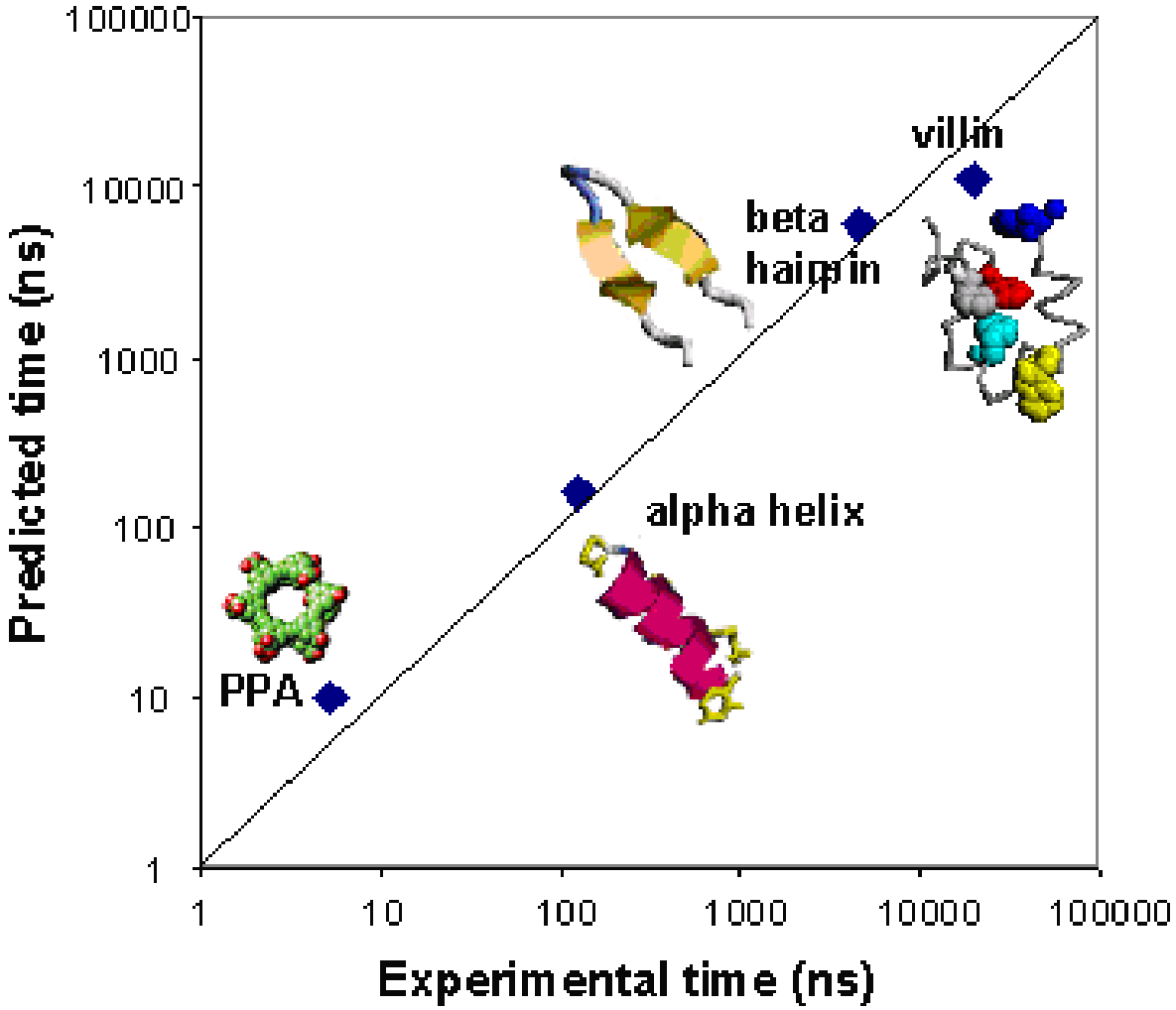


J. Mol. Biol. **313**, 151 (2001)

14 ns

KRISS

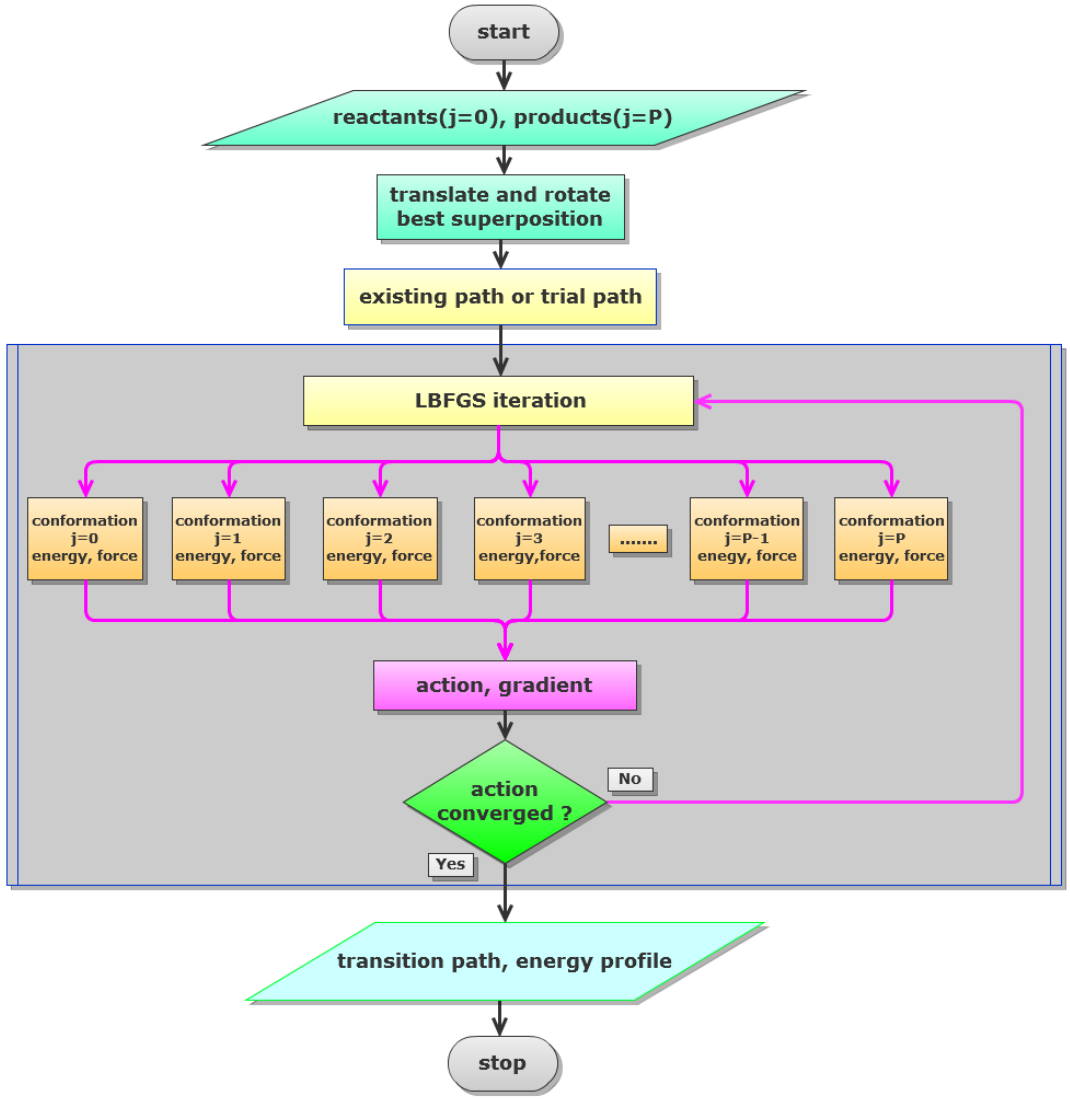
Accurate prediction of folding rates for many small proteins



Pande et al.

http://folding.stanford.edu/results.html

KRISS


start
reactants(j=0), products(j=P)
translate and rotate
best superposition
existing path or trial path
LBFGS iteration
conformation
j=0
energy, force
conformation
j=1
energy, force
conformation
j=2
energy, force
conformation
j=3
energy,force
.....
conformation
j=P-1
enegy, force
conformation
j=P
energy, force
action, gradient
action
converged ?
No
Yes
transition path, energy profile
stop

KRISS

Hamilton's principle

$$\delta \int_0^\tau dt L(\{q\},\{\dot{q}\}) = 0$$

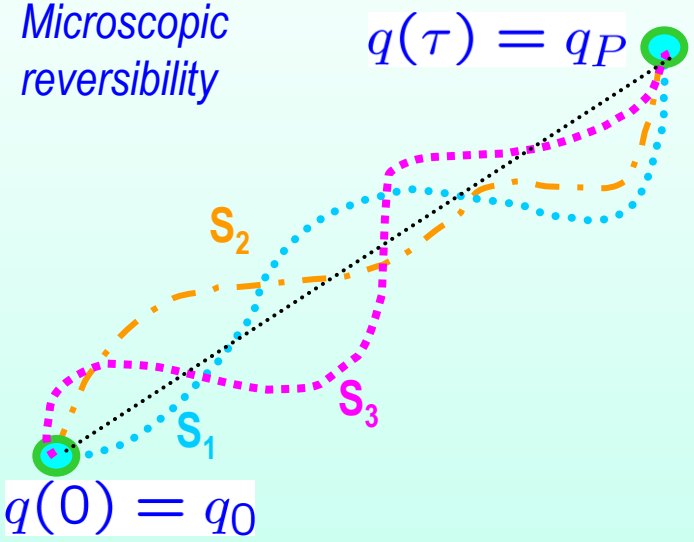
$\Leftrightarrow$

Lagrange's equation

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = 0$$

$$L(\{q\},\{\dot{q}\}) = T(\{\dot{q}\}) - V(\{q\})$$

Microscopic reversibility



$$S = \Delta \sum_{j=0}^{P-1} \left\{ \frac{M}{2}\left(\frac{q_j - q_{j+1}}{\Delta}\right)^2 - V(\{q_j\}) \right\}$$

$$q_j = q_0 + \frac{j\Delta}{\tau}(q_P - q_0) + \sum_{k=1}^{P-1} a_k \sin\left(\frac{k\pi j\Delta}{\tau}\right)$$

$$j = 0,1,2,3,\ldots,P; \quad \Delta = \tau/P$$

Fast sine transformation
{q} ↔ {a}
$O(P^2) \rightarrow O(P \log P)$

http://gams.nist.gov/serve.cgi/Module/SLATEC/SINT/7394/

Saddle? $\{a_k\} \Leftrightarrow S$

Complexity → $3N_{atom}$ (P-1)

KRISS

The Verlet algorithm

$$q(t+\Delta) = q(t) + v(t)\Delta + \frac{1}{2}a(t)\Delta^2 + \frac{1}{6}b(t)\Delta^3 + O(\Delta^4)$$

$$q(t-\Delta) = q(t) - v(t)\Delta + \frac{1}{2}a(t)\Delta^2 - \frac{1}{6}b(t)\Delta^3 + O(\Delta^4)$$

+

$$q(t+\Delta) = 2q(t) - q(t-\Delta) + a(t)\Delta^2 + O(\Delta^4)$$

$$j = 1, 2, 3, \ldots, P-1 \qquad \Delta = \tau/P$$

$$2q_j - q_{j+1} - q_{j-1} - \frac{\Delta^2}{M}\frac{\partial V(\{q_j\})}{\partial q_j} \approx 0$$

$$O = \sum_{j=1}^{P-1}\left(2q_j - q_{j+1} - q_{j-1} - \frac{\Delta^2}{M}\frac{\partial V(\{q_j\})}{\partial q_j}\right)^2$$

"path quality" measure

Adv. Chem. Phys. **126**, 93 (2003)

Elber *et al.* (1996)

aka Onsager-Machlup

KRISS

improved path quality : new object function

$$O = \sum_{I=1}^{N} \sum_{j=1}^{P-1} \left(2\vec{R}_{I,j} - \vec{R}_{I,j-1} - \vec{R}_{I,j+1} - \frac{\Delta^2}{M_I} \frac{\partial V(\{\vec{R}_{I,j}\})}{\partial \vec{R}_{I,j}} \right)^2$$

Onsager-Machlup

Chem. Phys. Lett. **105,** 9299 (1996)

$$S = \Delta \sum_{j=0}^{P-1} \left\{ \sum_{I=1}^{N} \frac{M_I}{2} \left(\frac{\vec{R}_{I,j} - \vec{R}_{I,j+1}}{\Delta} \right)^2 - V(\{\vec{R}_{I,j}\}) \right\}$$

$$\Theta(\{\vec{R}_{I,j}\}; E) = S + \mu \sum_{j=0}^{P-1} \left(E_j - E \right)^2$$

Target energy

Passerone-Parrinello

Phys. Rev. Lett. **87,** 10832 (2001)

Fictitious T

$$\tilde{\Theta}(\{\vec{R}_{I,j}\}; E; T) = S + \mu \sum_{j=0}^{P-1} (E_j - E)^2 + \nu \sum_{I=1}^{N} \left(\langle K_I \rangle - \frac{3k_B T}{2} \right)^2$$

$$\mu > \mu^*, \nu > \nu^*$$

Phys. Rev. Lett. **90**, 065501 (2003)
Phys. Rev. B **68,** 064303 (2003)
J. Chem. Phys. **120**, 4672 (2004)

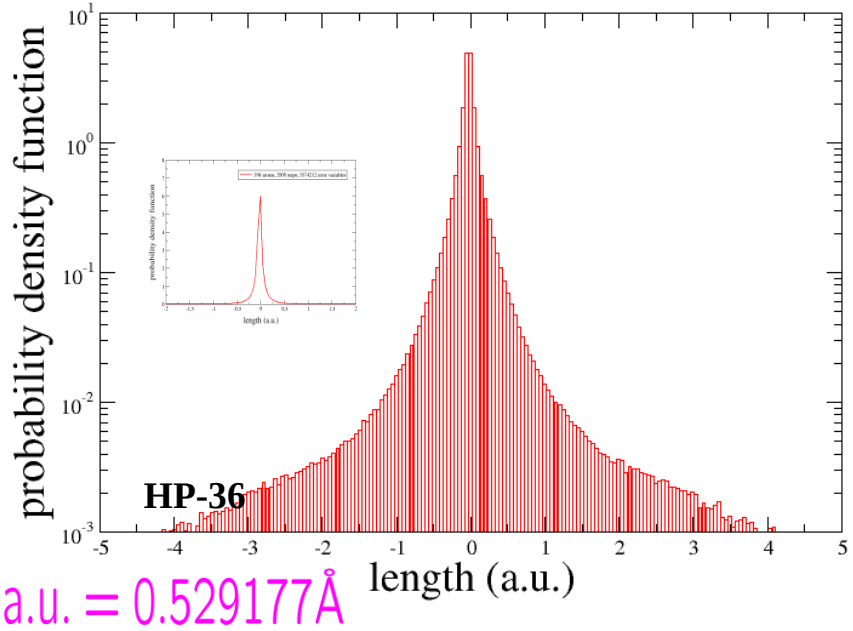
KRISS

Typical distributions of $\{\epsilon_{I,j}\}$ $C(j)$

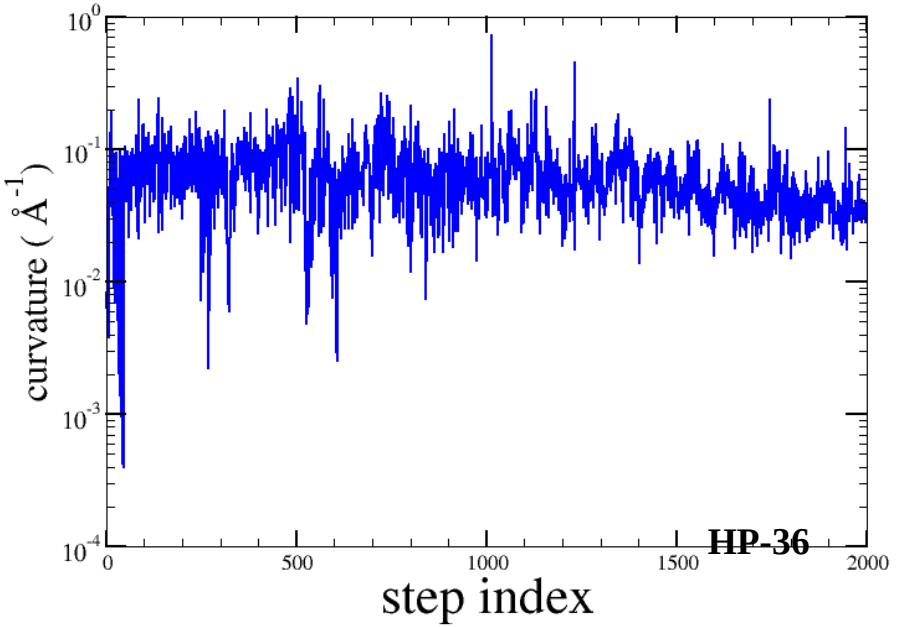
$$\vec{\epsilon}_{I,j} = 2\vec{R}_{I,j} - \vec{R}_{I,j-1} - \vec{R}_{I,j+1} - \frac{\Delta^2}{M_I}\frac{\partial V(\{\vec{R}_{I,j}\})}{\partial \vec{R}_{I,j}}$$

$$C(j) = \frac{|\ \vec{R}_{j+1} - 2\vec{R}_j + \vec{R}_{j-1}\ |}{|\ \vec{R}_{j+1} - \vec{R}_j\ |^2}$$

I : 1, 2, 3,…., N
j : 1, 2, 3,….,P-1



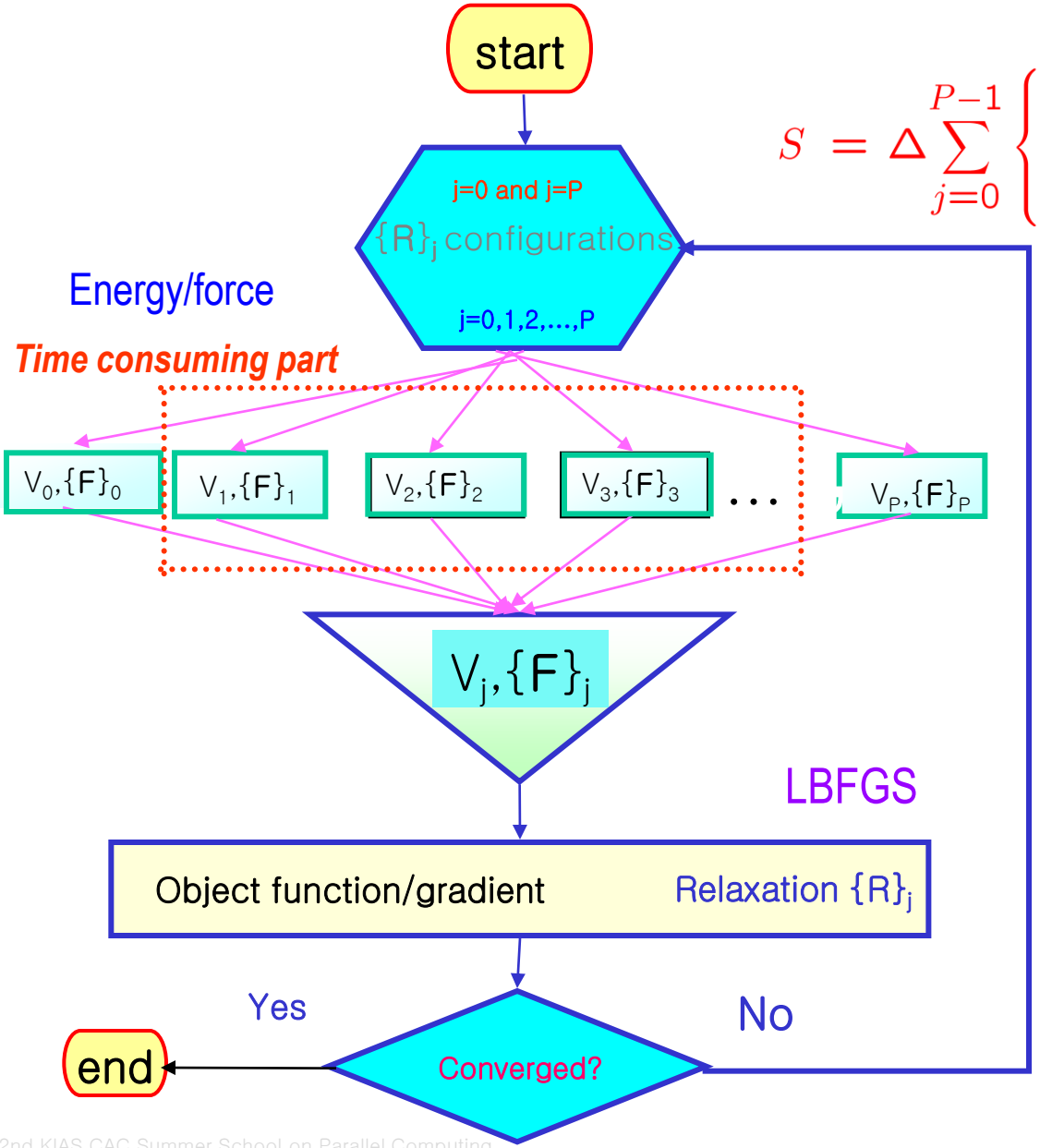
a.u. = 0.529177Å



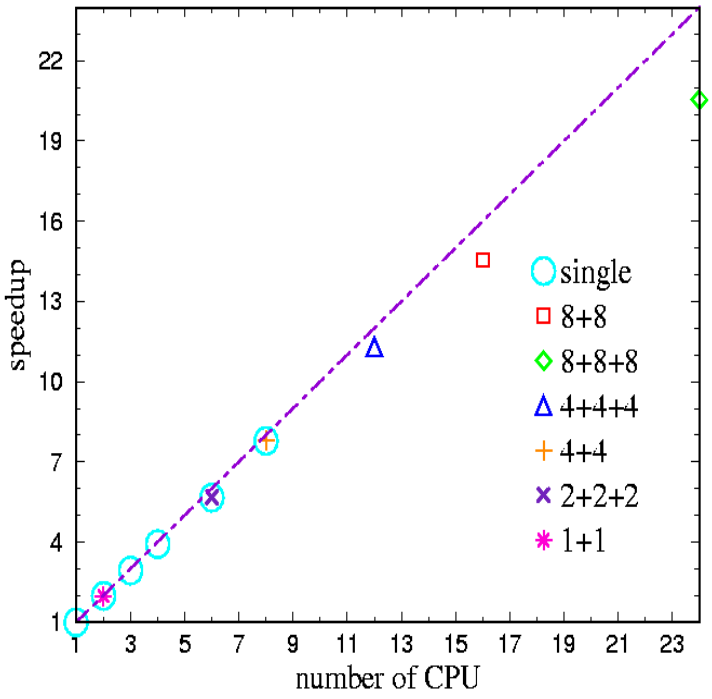
Complexity → 3N(P-1),
N = the number of atoms,
P = the number of steps

KRISS

Parallel ADMD



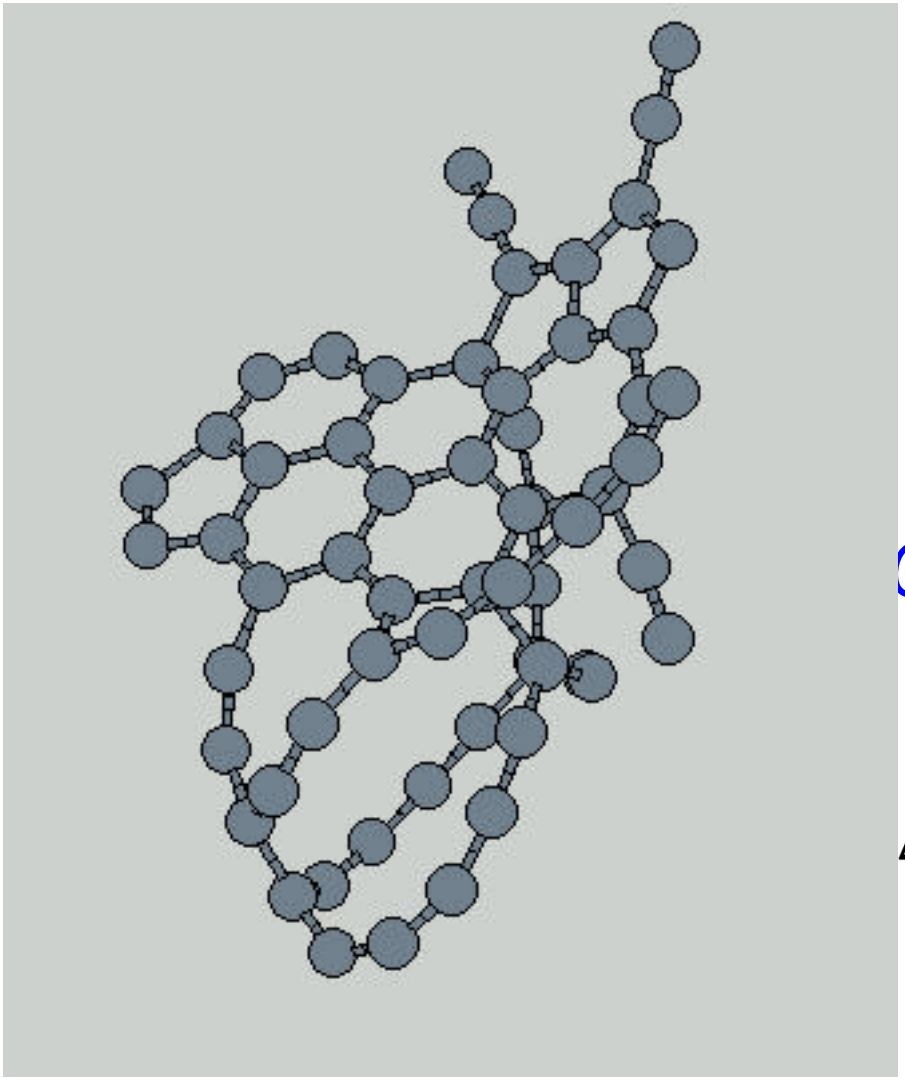
$$S = \Delta \sum_{j=0}^{P-1} \left\{ \sum_{I=1}^{N} \frac{M_I}{2} \left(\frac{\vec{R}_I^{\,j} - \vec{R}_I^{\,j+1}}{\Delta} \right)^2 - V(\{\vec{R}_I^{\,j}\}) \right\}$$



{Communication}/{CPU} ratio

KRISS

C_{60} formation ADMD simulation



Tangled polycyclic

Chain/hexagon

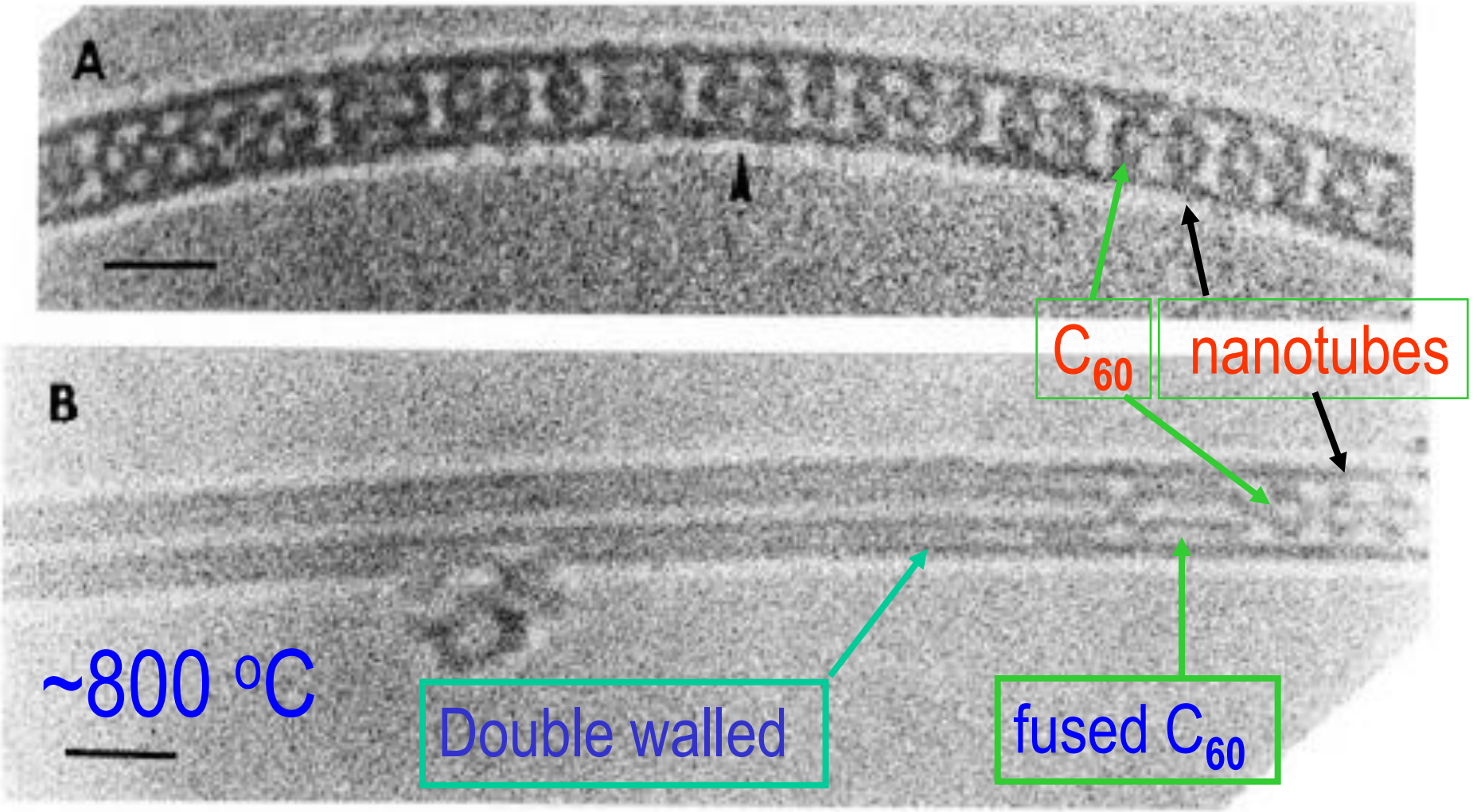
$E_a = 3.3$ eV

$\Delta E = 45.7$ eV

J. Chem. Phys. **120**, 4672 (2004)

KRISS

C_{60} fusion



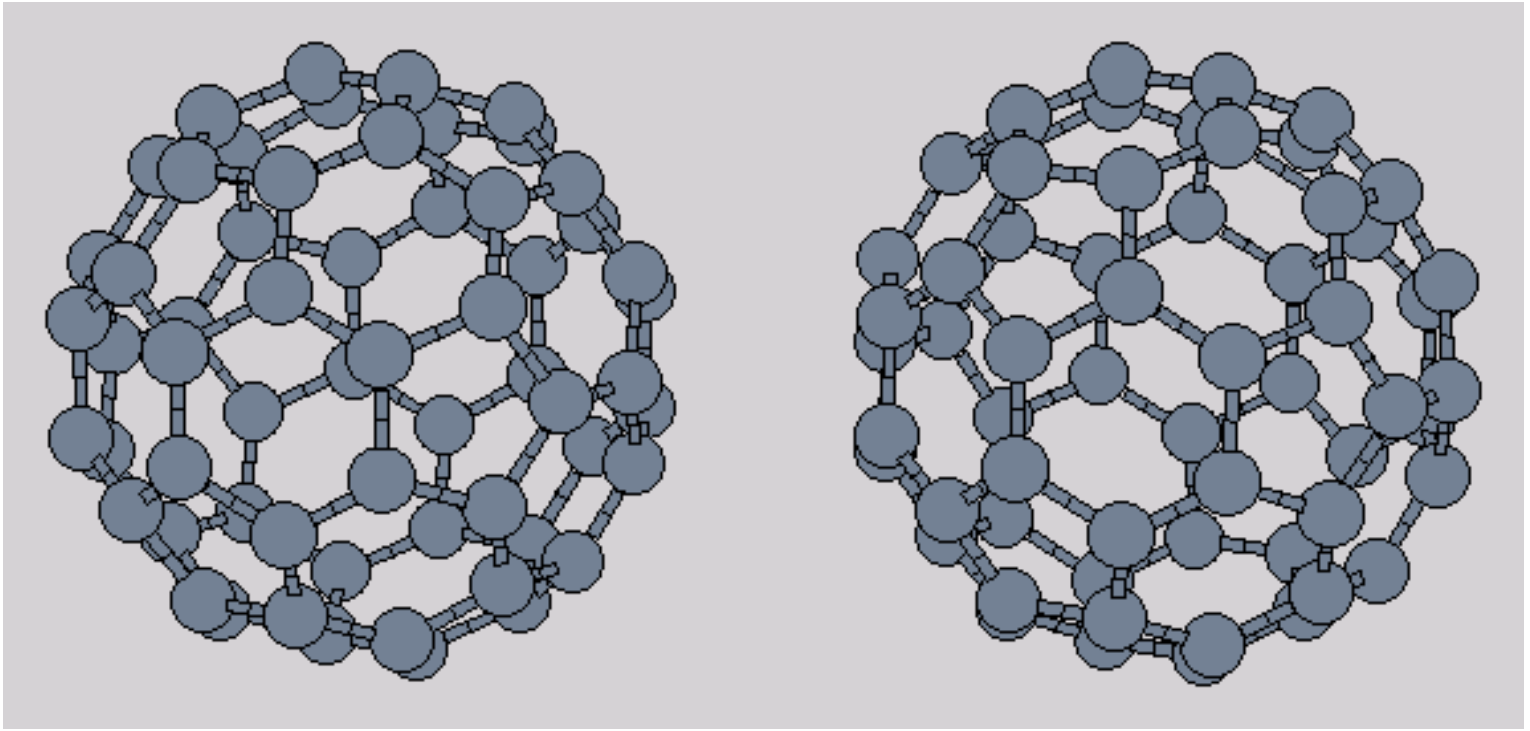
Chem. Phys. Lett. **315**, 31 (1999)

Chem. Phys. Lett. **384**, 320 (2004)

$$\text{rate} \propto 10^{12} \times \exp(-E_a/(k_B T)) \text{ sec}^{-1}$$
$$\sim 10^{-17} \text{ sec}^{-1}$$

KRISS

C_{60} fusion ADMD simulation

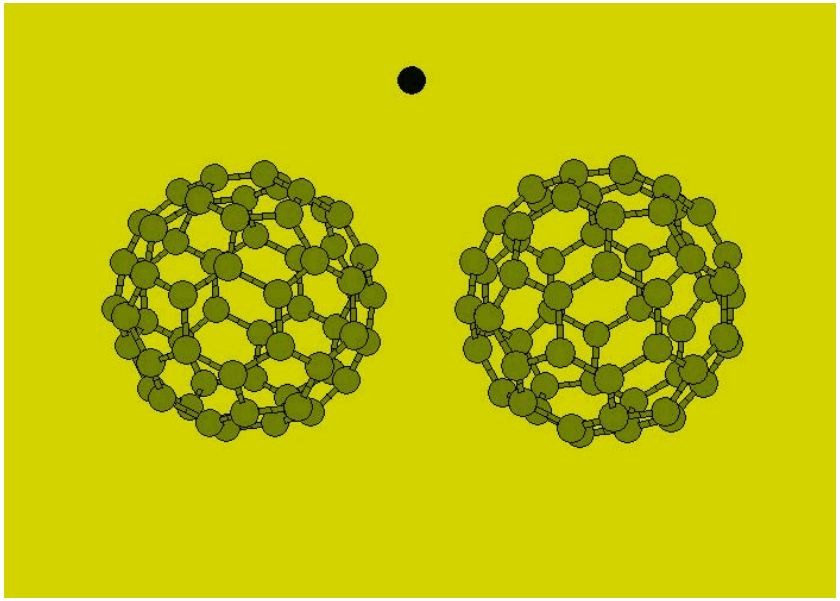


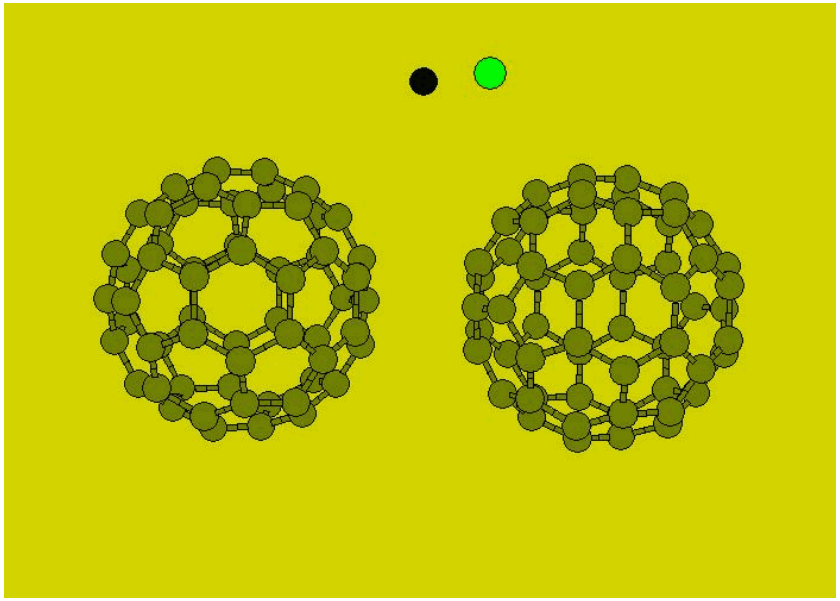
$C_{60} + C_{60} \rightarrow$ (5,5)-capsule

Phys. Rev. Lett. **90**, 065501 (2003)

KRISS

C_{60} fusion →capsule (5,5) chirality





$E_a = 1.4$ eV

An extra carbon atom

$E_a = 0.0$ eV

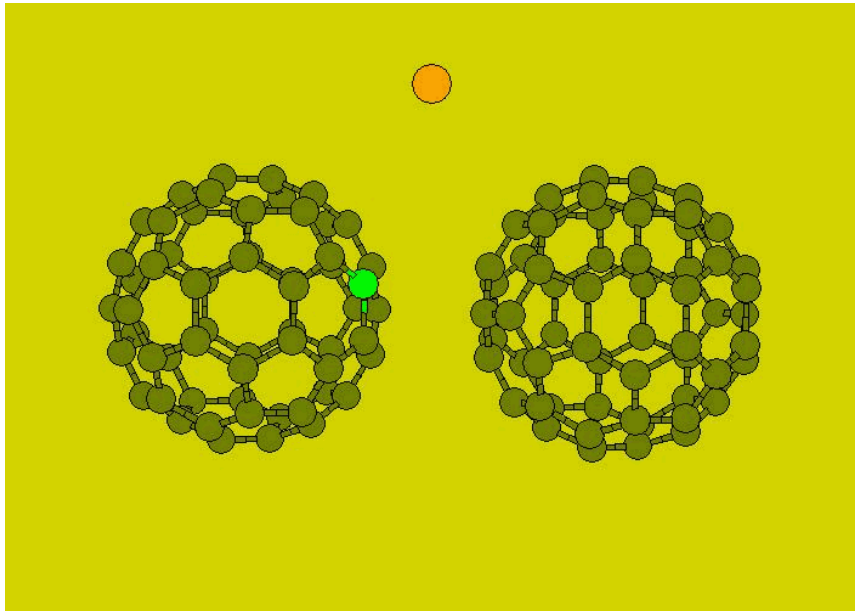
Two extra carbon atoms

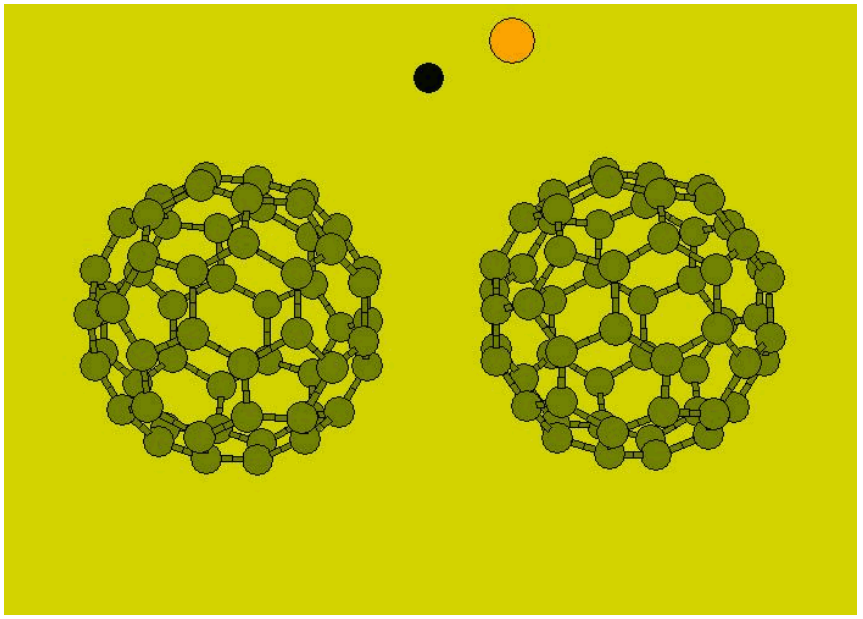
catalyst

Appl. Phys. Lett. **88**, 011913 (2006)

KRISS

Single/double exchange





$E_a = 1.1$ eV

$E_a = 0.0$ eV

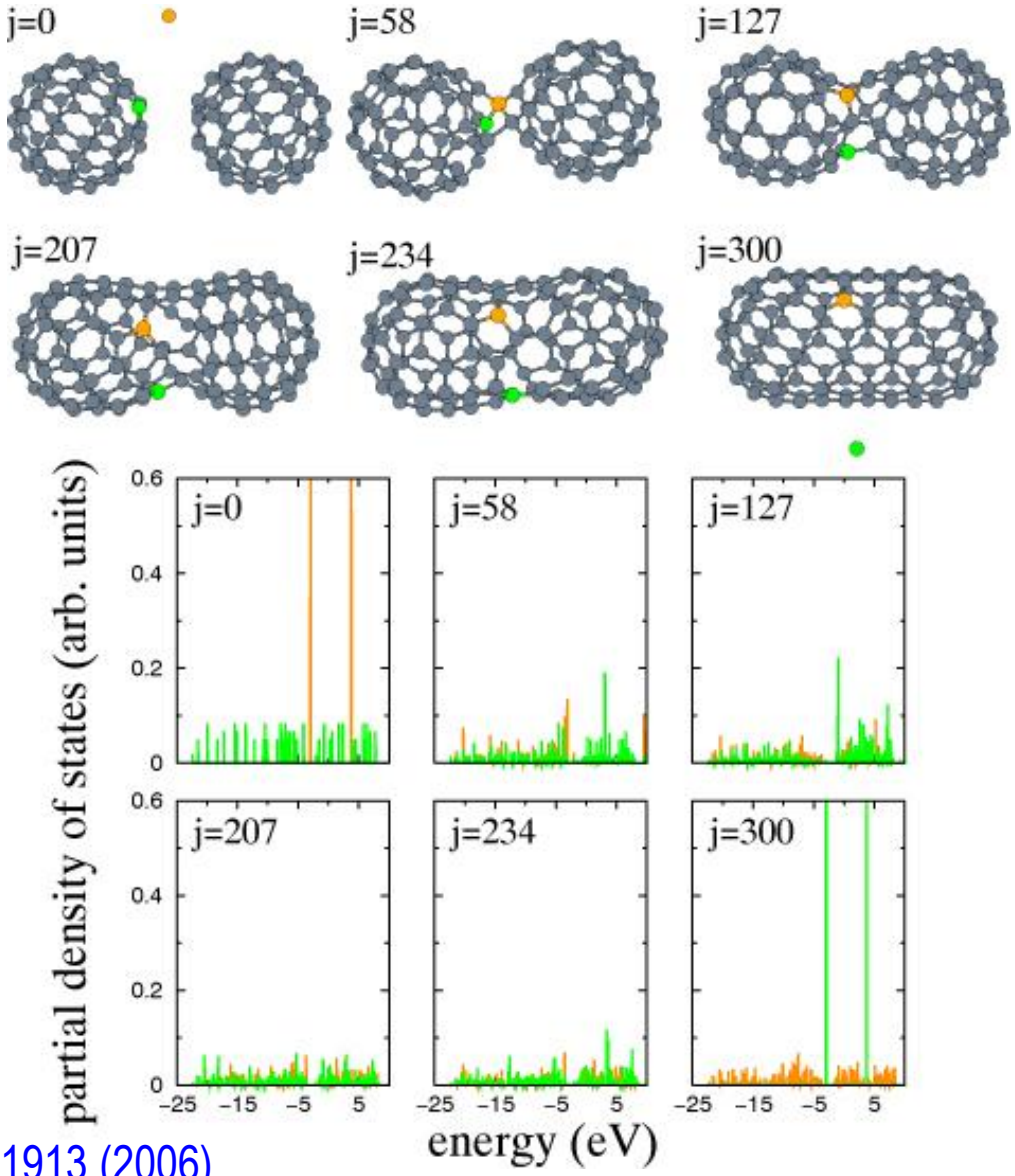
$C_{60} + C_{60} + C \rightarrow C_{120} + C$

$C_{60} + C_{60} + C + C \rightarrow C_{120} + C + C$

Appl. Phys. Lett. **88**, 011913 (2006)

KRISS

Partial density of state



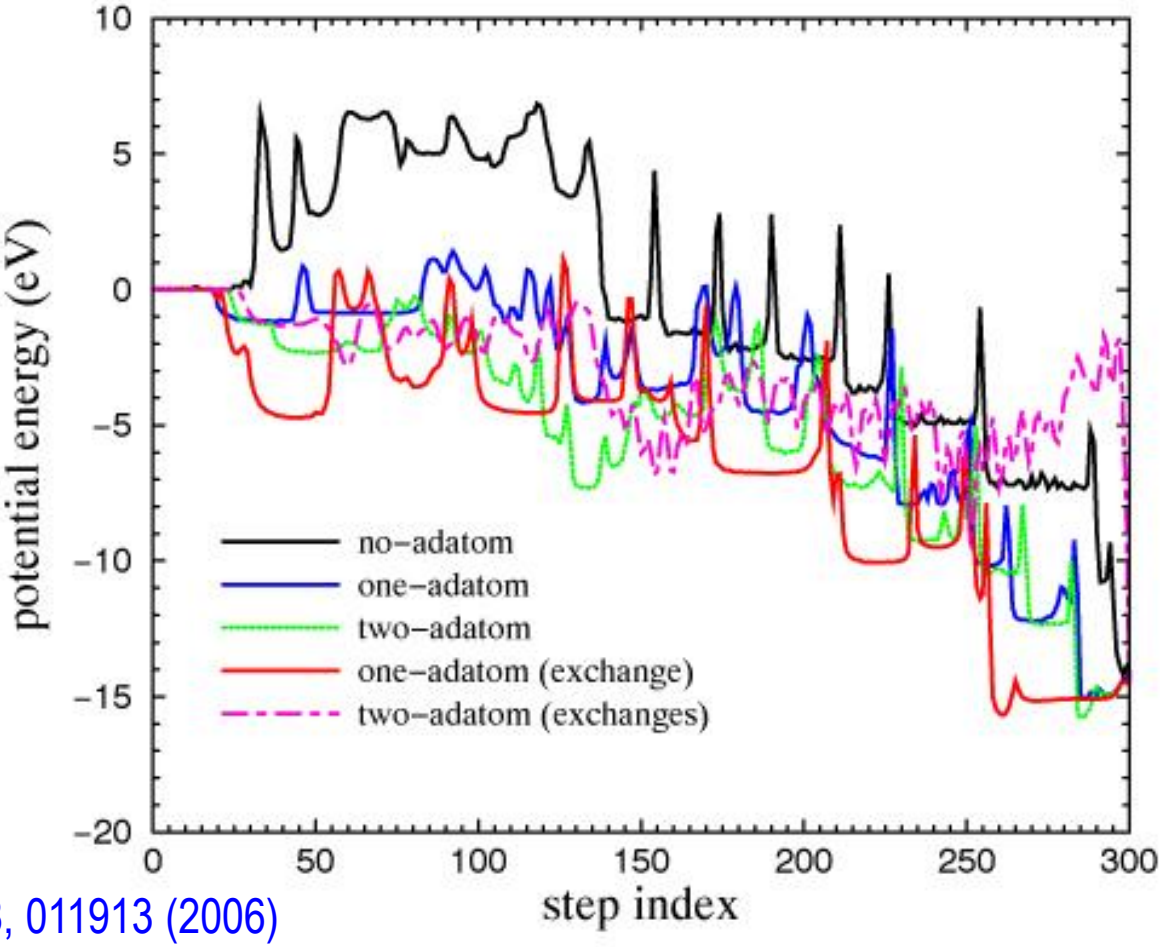
Appl. Phys. Lett. **88**, 011913 (2006)

KRISS

Potential energy profile along pathway

$C_{60} + C_{60} + C \rightarrow C_{120} + C$

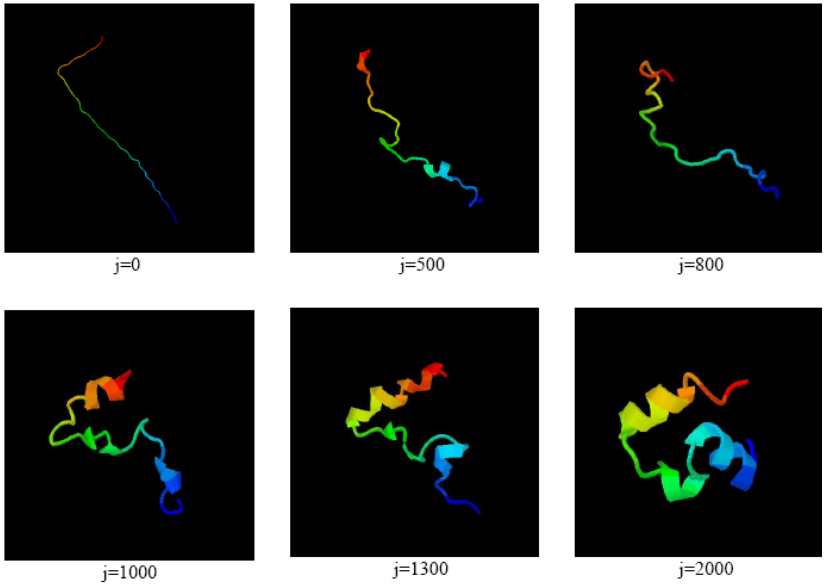
$C_{60} + C_{60} + C + C \rightarrow C_{120} + C + C$

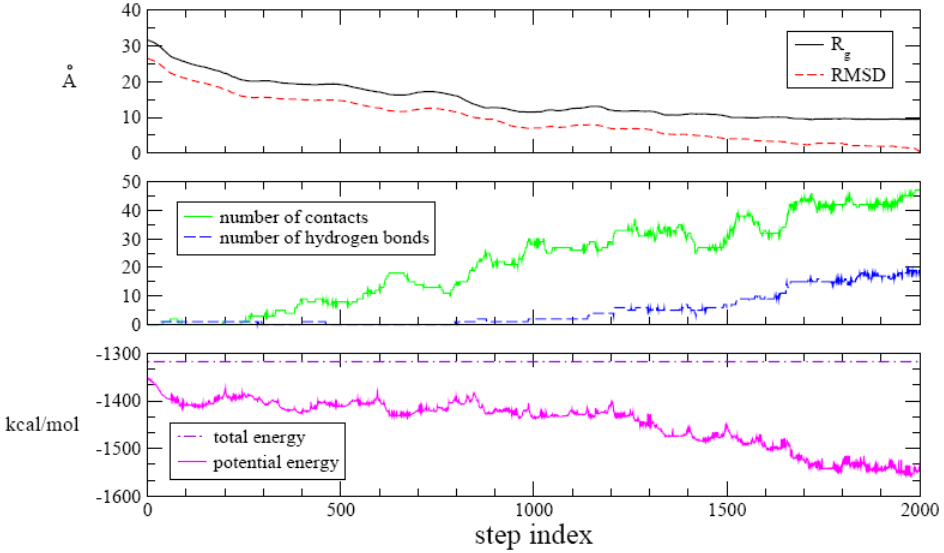


Appl. Phys. Lett. **88**, 011913 (2006)

KRISS

HP-36

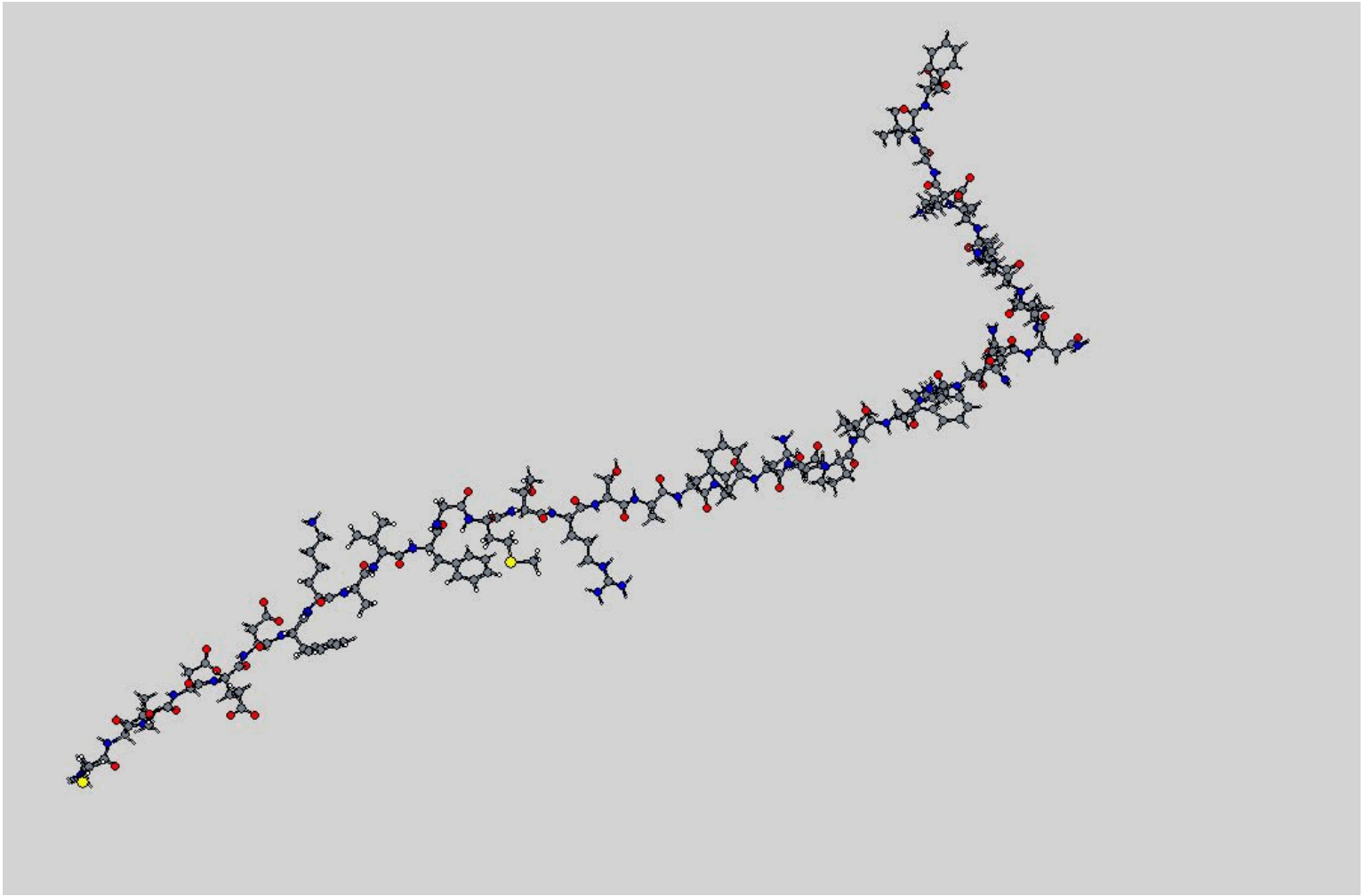




J. Comput. Chem. **31**, 57 (2009)

KRISS

HP-36



$3N(P-1) = 3 \times 596 \times 1999 = 3{,}574{,}212$

Amber94, GB/SA

36.28 fs

J. Comput. Chem. **31**, 57 (2009)

KRISS

Reaction coordinate(s) ←→ ADMD simulation

Reaction coordinate(s) → Markov state model

→ ***free energy profile***

→ ***time scale***

Markov Processes

- Stochastic process
  - movement through a series of well-defined states in a way that involves some element of randomness
  - for our purposes, "states" are microstates in the governing ensemble
- Markov process
  - stochastic process that has no memory
  - selection of next state depends only on current state, and not on prior states
  - process is fully defined by a set of transition probability matrix P_{ij}

    probability of selecting state *j* next, given that presently in state *i*.

Having the Markov property means that, *given the present state*, future states are independent of the past states. In other words, the description of the present state fully captures all the information that could influence the future evolution of the process.

http://en.wikipedia.org/wiki/Andrey_Markov

Predicting the weather

- very simple model (sunny-rainy model)
- sunny, rainy, K=2, 2x2 matrix P
- sunny → sunny : rainy = 0.9 : 0.1
- rainy → sunny : rainy = 0.5 : 0.5

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{x}^{(0)} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} P = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.9 & 0.1 \end{bmatrix}$$

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} P = \mathbf{x}^{(0)} P^2 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}^2 = \begin{bmatrix} 0.86 & 0.14 \end{bmatrix}$$

$$\Pr(X_{n+1} = x | X_1 = x_1, X_2 = x_2 \ldots, X_n = x_n) = \Pr(X_{n+1} = x | X_n = x_n).$$ **Markov model**

http://en.wikipedia.org/wiki/Examples_of_Markov_chains

KRISS

Steady state of the weather

```
      program steady_state_test
      implicit none
      integer k
      real*8 pp(2,2),vec1(2),vec2(2)

      pp(1,1)=0.9d0 ; pp(1,2)=0.1d0
      pp(2,1)=0.5d0 ; pp(2,2)=0.5d0

      vec1(1)=1.d0 ; vec1(2)=0.d0

      do k=1,1000
      vec2=matmul(vec1,pp)
      vec1=vec2
      end do
      write(6,*) vec2

      stop
      end
```

0.8333333333333345 0.1666666666666669
FORTRAN STOP

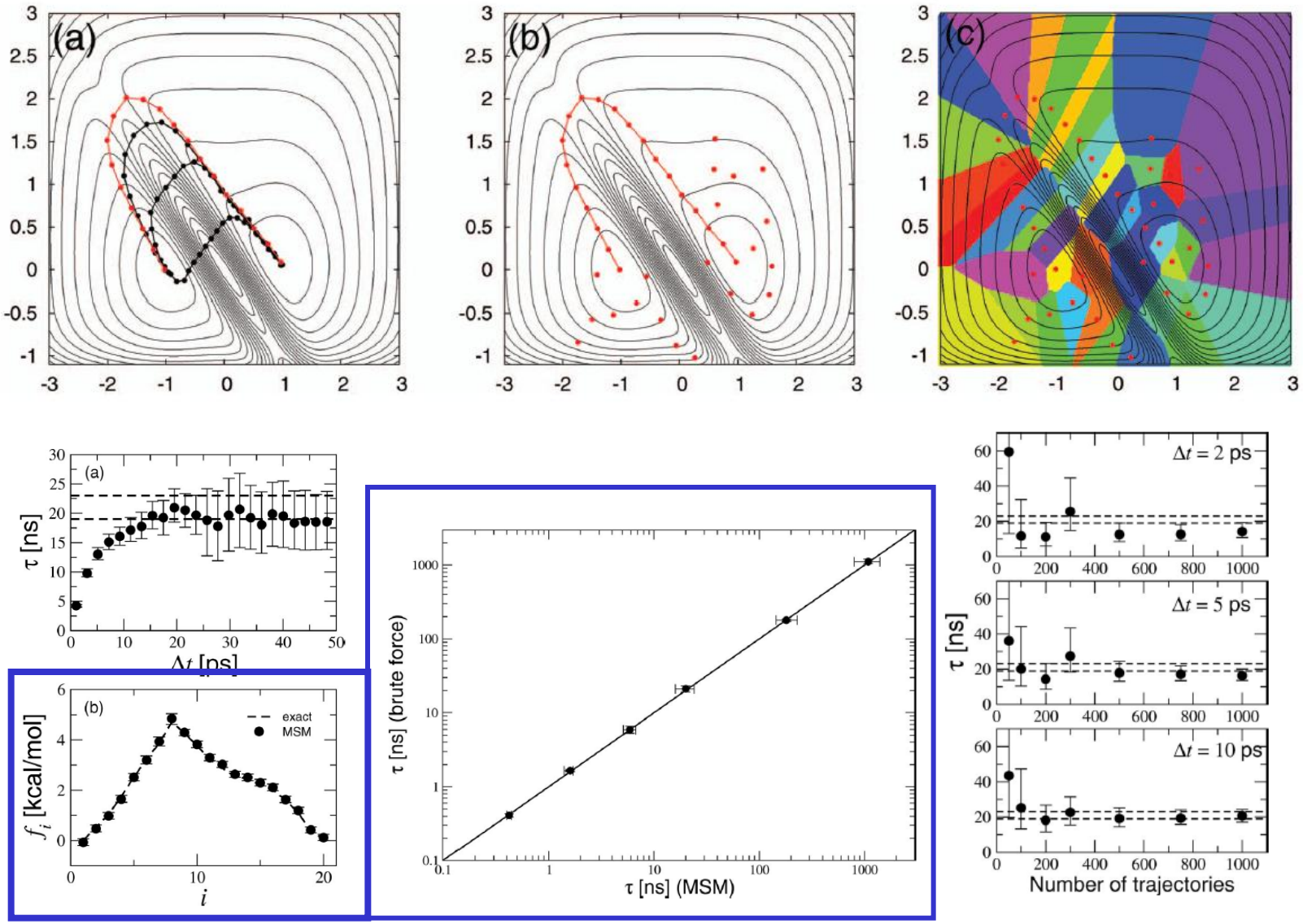
$$\lim_{k\to\infty} \mathbf{P}^k = \mathbf{1}\pi$$

vec2(1)*pp(1,2)-vec2(2)*pp(2,1)=0
Detailed balance

In conclusion, in the long term, 83% of days are sunny.

Stationary distribution

Markov State Model: example



J. Chem. Phys. **129**, 064107 (2008).

A. C. Pan and B. Roux

Free energy profile and characteristic time scale

KRISS

Transition-count matrix

- $M_{ij} = M_{ij} + 1$, state i; 0 → state j; τ, lag time τ
- time-consumming part : many (relatively) short MD simulations : parallel job
- $M_{ij}/(\text{sum}_j\ M_{ij})$ → P_{ij} : finite sampling

- Posterior_distribution (P | M) ~ Likelihood(M | P) x Prior_distribution (P)
- Likelihood(M | P) ~ $P_{ij}^{\ M_{ij}}$
- Prior_distribution (P) ~ $P_{ij}^{\ \beta_{ij}-1}$
- Posterior_distribution (P | M) ~ $P_{ij}^{\ \alpha_{ij}-1}$, a Dirichlet distribution
- $\alpha_{ij}=\beta_{ij}+M_{ij}$
- $\beta_{ij}= ½$ or $\beta_{ij}= 1$ or $\beta_{ij}= 1/K$, [KxK matrices, M, P, α, β]
- Dirichlet distribution, ~ $P_{ij}^{\ \alpha_{ij}-1}$

- Bayesian Inference : error analysis

W. C. Swope , J. W. Pitera, and F. Suits, J. Phys. Chem. B **108**, 6571 (2004).

Transition matrix

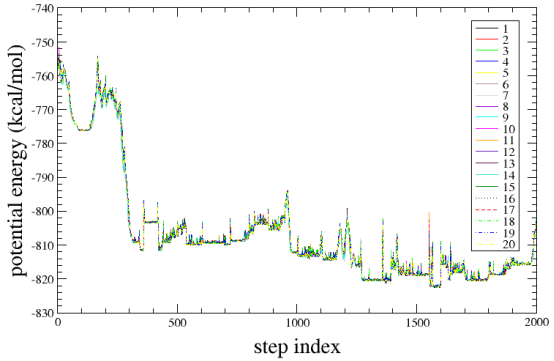
• If the state space is finite, the transition probability distribution can be represented by a matrix, called the **transition matrix**,
with the *ij* element of **P** equal to $P_{ij}= Pr(X_{n+1}=j| X_n=i)$.

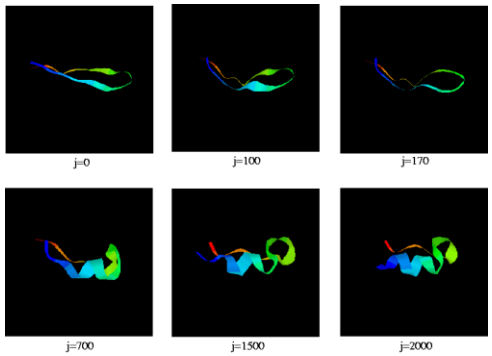
$$\pi = \pi P, \quad \pi: \text{left-eigenvector, eigenvalue } =1$$
$$\text{probability of the state i : } \pi_i \quad : \text{sum}_i\ \pi_i = 1$$

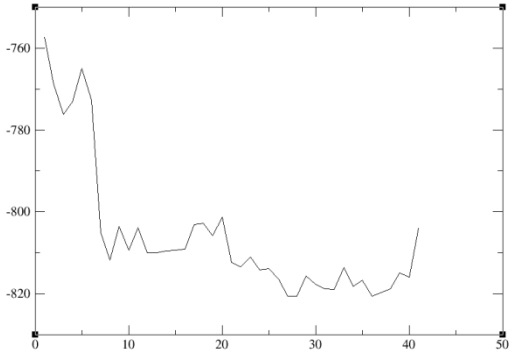
- a left-eigenvector of the transition matrix associated with the eigenvalue 1
- a right-eigenvector of the transition matrix associated with the eigenvalue 1
- single eigenvalue=1 for left-eigenvector and right-eigenvector
- a time-homogenous chain: Markov chain, a stationary distribution
- Perron-Frobenius theorem
- other eigenvalues, |eigenvalue| < 1
- $(\text{relaxation time scale})_s = -\tau/\ln(\lambda_s)$, largest eigenvalue of P matrix less than 1
- $\lambda_1,\lambda_2,\lambda_3,\lambda_4,\lambda_5,\ldots,\lambda_{K-1},\lambda_K=1$, (in ascending order), $\lambda_{K-1}=\lambda_s$

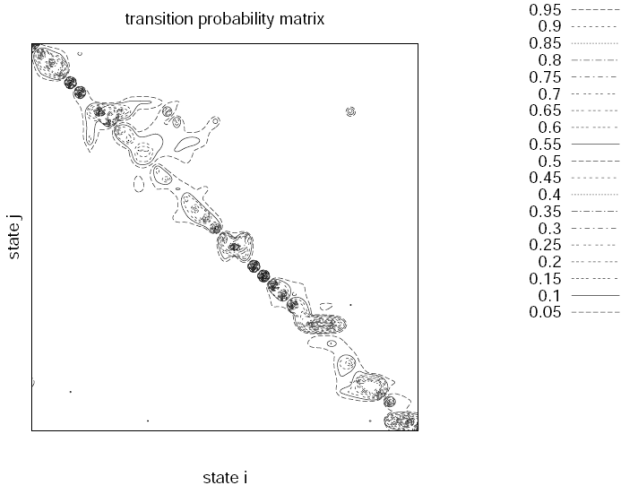
- $P_{ij} >=0$
- $\text{sum}_j\ P_{ij} =1$. (stochasticity) : sum of each row is unity
- $\pi_i\ P_{ij} = \pi_j\ P_{ji}$ (detailed balance) : $\pi = \pi P$
- $\text{sum}_i\ \pi_i P_{ij} = \text{sum}_i\ \pi_j P_{ji} = \pi_j\ \text{sum}_i\ P_{ji} = \pi_j$

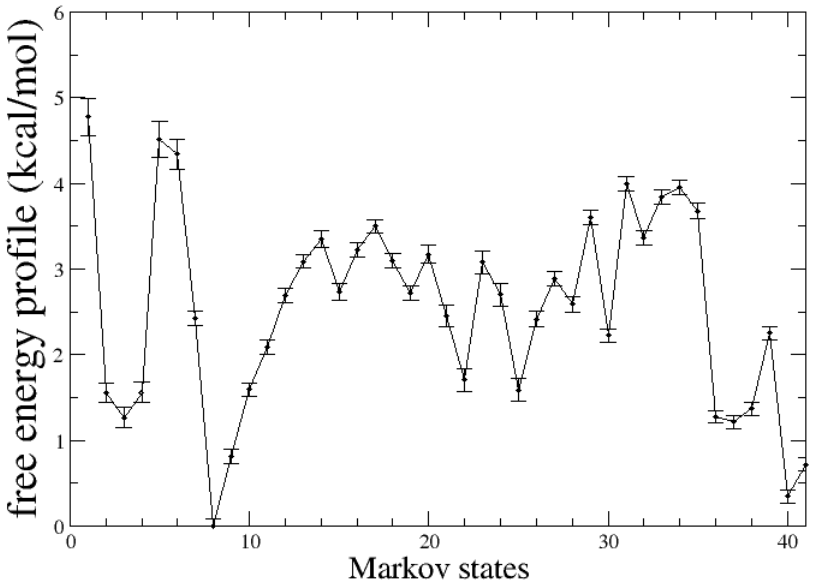
0.15 microsecond











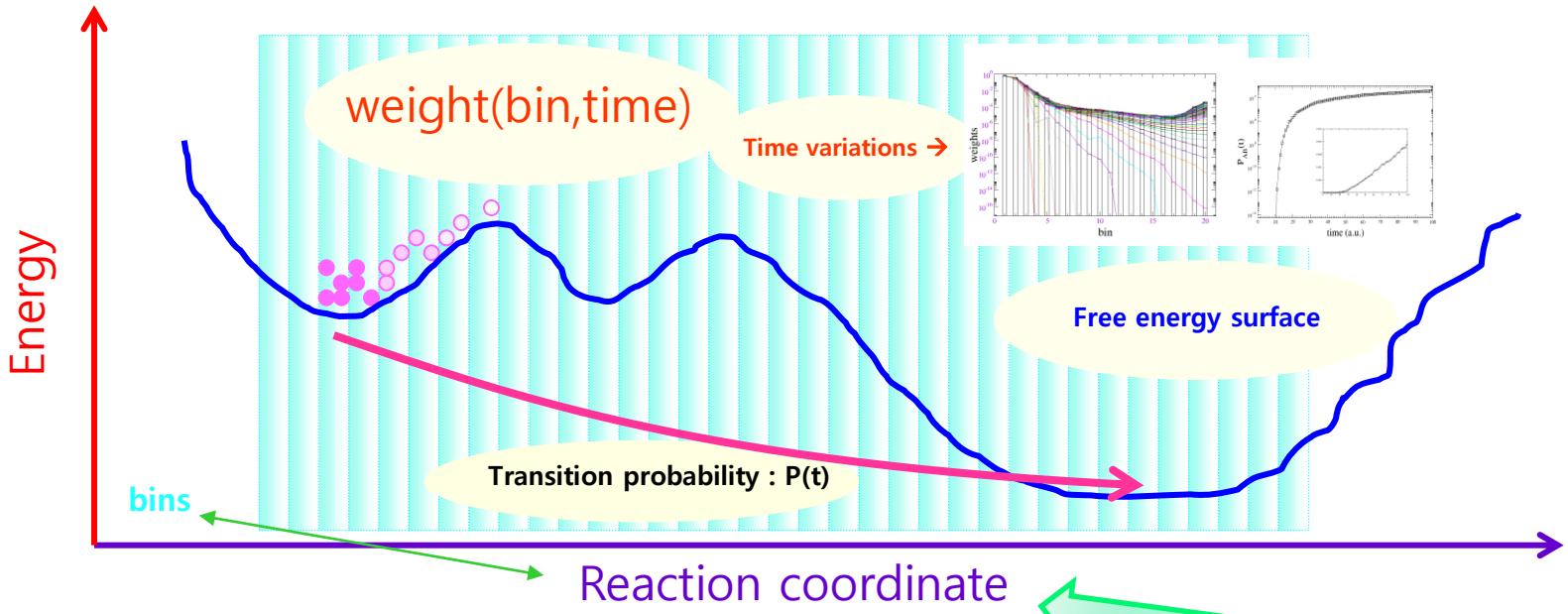
41x41 matrix

Langevin dynamics, 300 K, 91 ps

$$m\ddot{x} = -m\gamma\dot{x} - \frac{\partial V(x)}{\partial x} + \xi(t)$$

KRISS

Weighted-Ensemble

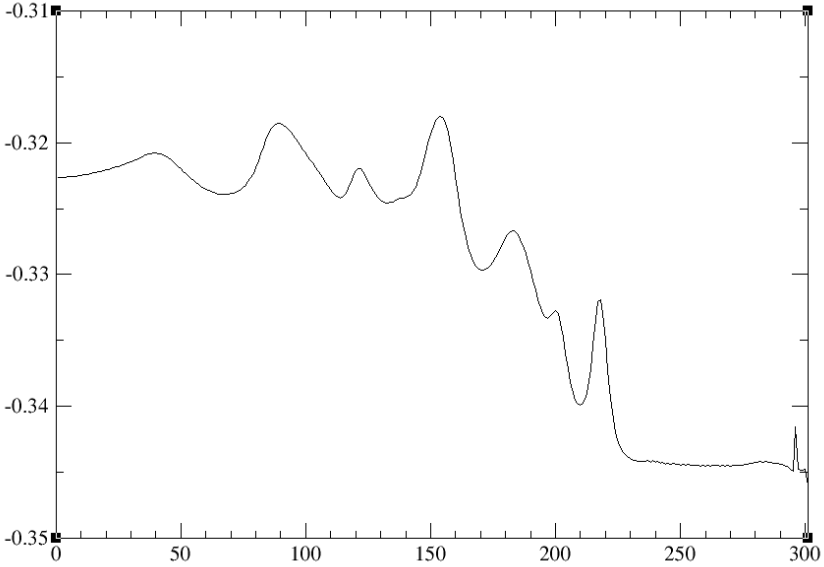


$$m\ddot{x} = -m\gamma\dot{x} - \frac{\partial V(x)}{\partial x} + \xi(t)$$

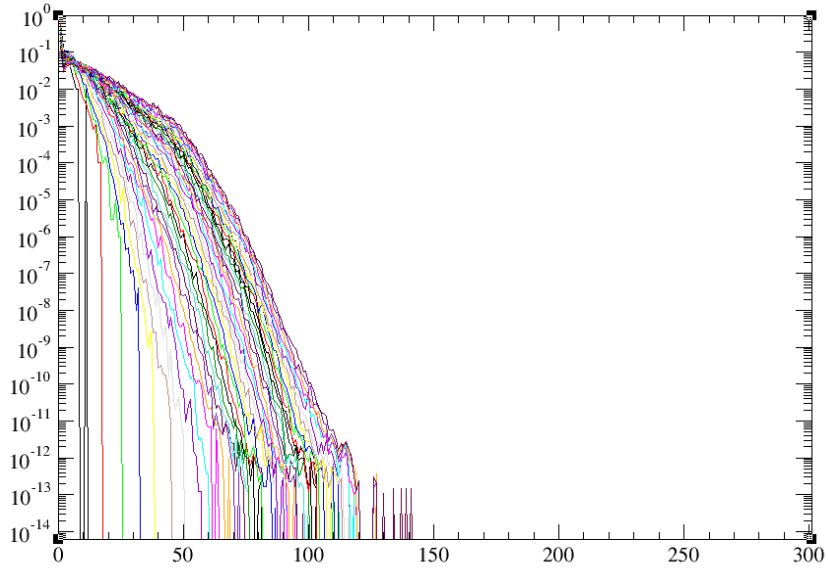
Weighted-ensemble Brownian dynamics simulations for protein association reactions
Biophysical Journal, Volume 70, Issue 1, Pages 97-110

KRISS

Example



Potential energy profile

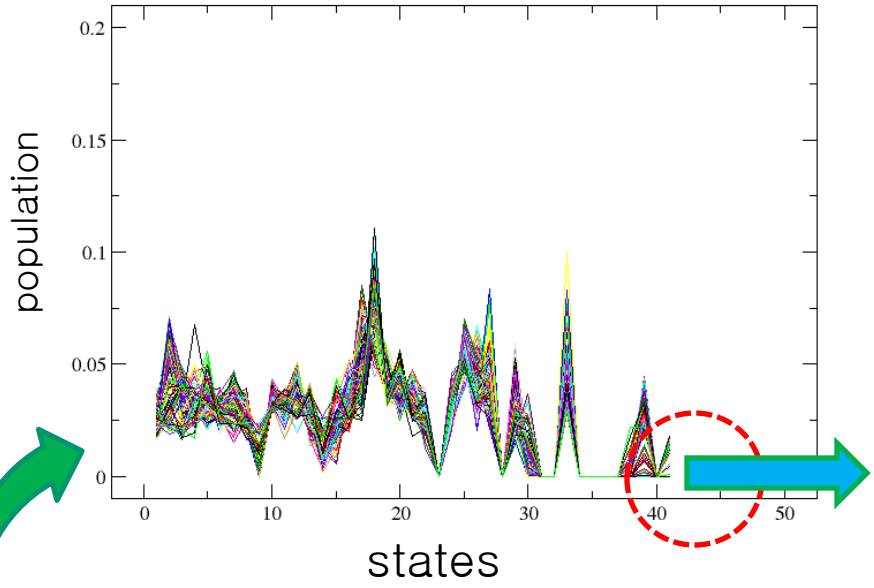


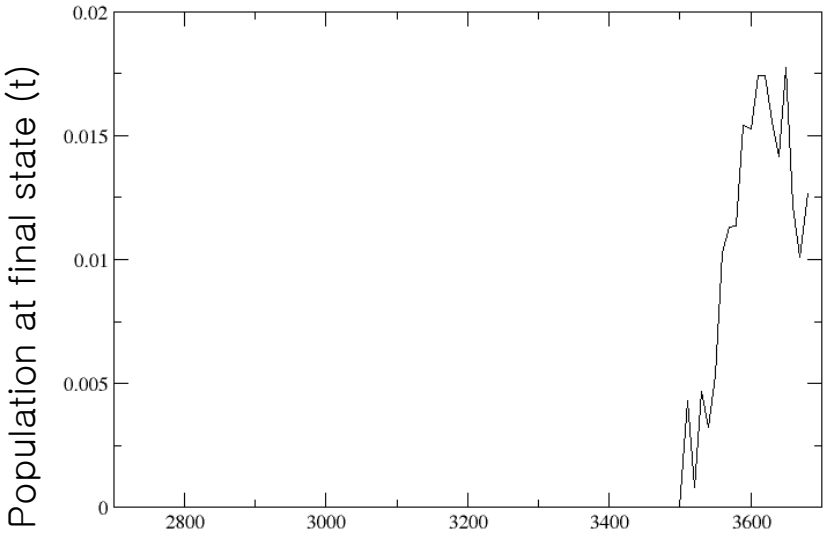
Population changes

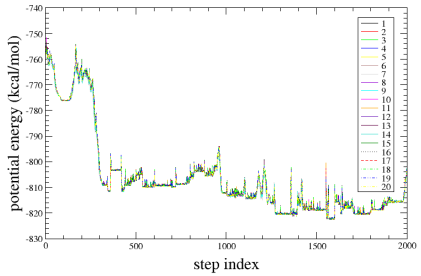
$$m\ddot{x} = -m\gamma\dot{x} - \frac{\partial V(x)}{\partial x} + \xi(t)$$

KRISS

1l2y







32 replicas/bin, 300 K, 91 ps^{-1}

$$m\ddot{x} = -m\gamma\dot{x} - \frac{\partial V(x)}{\partial x} + \xi(t)$$

KRISS

conclusions

* MC/MD algorithms
* Parallel versions of MC/MD
* speedup