Molecular Dynamics Simulations

* Purely deterministic method.

* Follow time evolution of systems by solving Newton's equation of motion:

$$m\frac{d^2r}{dt^2} = -F(r) \qquad \text{where } F(r) = -\nabla V(r)$$

* Properties of statistical ensemble are calculated as time averages.

* Microcanonical ensemble (N, V, E) is the common choice + other constraints:

eg. momentum M =====> (N, V, E, M) = MD ensemble.

* For N particles, problem reduces to solving systems of 6N 1st order ODE:

$$\begin{cases} \dot{\vec{r}}_i = \vec{v}_i \\ m\dot{\vec{v}}_i = \vec{F}_i \end{cases}; \qquad F_i = -\nabla_{r_{ij}} \sum_{i \neq j} V(r_{ij}); \quad \nabla_{r_{ij}} = \hat{i}\frac{\partial}{\partial x_{ij}} + \hat{j}\frac{\partial}{\partial y_{ij}} + \hat{k}\frac{\partial}{\partial z_{ij}}$$

Potential Functionals

* **2-body interactions**: $\Phi\,(1,\ldots,N) = \Sigma_{i<j} V2(i,j)$

- Lennard-Jones (LJ):

$$V_2(r) = 4\varepsilon\,[x^{-12} - x^{-6}]$$

ε - maximal well-depth of potential.

$x = r/\sigma$ - distance in units of σ, where σ is effective atomic radius.

works fine for rare gases, still used for gas – metal interactions in thin film growth simulations

- Yukawa potentials:

Simple: $V_2(r) = (E/x)\,\exp[-a(x-1)]$

Double: $V_2(r) = (E/x)\,\{\exp[-a(x-1)] - \exp[-b(x-1)]\}$

x just as for Lennard-Jones, a and b are parameters.

advantages over soft-spheres and LJ to perform “analytical” investigations in thermodynamics and variational free-energy studies.

- Morse:

$$V_2(r) = E\ \{\exp[-2a(r - r_0)] - 2\exp[-a(r - r_0)]\}$$

similar properties to LJ.

allows for more intermediate interactions.

behaves more like a bonding-type potential.

- Hard-Sphere:

$$V(r) = \infty; r < \sigma \quad \text{and} \quad V(r) = 0; r > \sigma$$

used initially in first MD simulations

"wall-like" potential, still useful in some contexts.

- Soft-Sphere:

V(r) = x – n, with x as for LJ and n can vary

n = 1, yields soft interaction.

the larger n, the harder the repulsion.

- Other potentials: Square-Well, Ionic

Potential Functionals

* **3-body interactions**: $\Phi\,(1,\ldots,N) = \Sigma_{i<j}V_2(i,j) + \Sigma_{i<j<k}V_3(i,j,k)$

- Stillinger – Weber:

$V_2(r) = A\,(Br^{-p} – r^{-q})\,\exp[(r\text{-}a) - 1]$; where A, B, p, q, a – params.

$V_3(r) = h\,(r_{ij}, r_{ik}, \theta_{jik}) + h\,(r_{ji}, r_{jk}, \theta_{ijk}) + h\,(r_{ki}, r_{kj}, \theta_{ikj})$.

$$E = \sum_{i}\sum_{j>i}\Phi_2(r_{ij}) + \sum_{i}\sum_{j\neq i}\sum_{k>j}\Phi_3(r_{ij}, r_{ik}, \theta_{ijk})$$

$$\Phi_2(r_{ij}) = A_{ij}\varepsilon_{ij}\left[B_{ij}\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{p_{ij}} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{q_{ij}}\right]\exp\left(\frac{\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}}\right)$$

$$\Phi_3(r_{ij}, r_{ik}, \theta_{ijk}) = \lambda_{ijk}\varepsilon_{ijk}\left[\cos\theta_{ijk} - \cos\theta_{0ijk}\right]^2 \exp\left(\frac{\gamma_{ij}\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}}\right)\exp\left(\frac{\gamma_{ik}\sigma_{ik}}{r_{ik} - a_{ik}\sigma_{ik}}\right)$$

- Tersoff:

$V_2(r) = f_C(r) + [f_R(r) + bf_A(r)]$; $b = b(\theta_{ijk})$

Tersoff functional

- the 2 & 3 body interactions require extensive CPU time.

$$E = \sum_{i}^{N} \Phi_i ; \qquad \Phi_i = \frac{1}{2} \sum_{j(\neq i)}^{N} f_c(r_{ij})[V_R(r_{ij}) - b_{ij} V_A(r_{ij})]$$

where

$$f_c(r_{ij}) = \left\{ \begin{array}{ccc} 1 & if & r_{ij} < R - D \\ \frac{1}{2} - \frac{1}{2}\sin\left[\frac{\pi}{2}(r_{ij} - D)/D\right] & if & R - D < r_{ij} < R + D \\ 0 & if & r_{ij} > R + D \end{array} \right\}$$

3-body part given by:

$$V_R(r_{ij}) = A\exp(-\lambda_1 r_{ij}); \;\; V_A(r_{ij}) = B\exp(-\lambda_2 r_{ij}); \;\; b_{ij} = (1 + \beta^n \xi_{ij}^n)^{-1/2n}$$

$$\xi_{ij} = \sum_{k(\neq i,j)}^{N} f_c(r_{ik}) g(\theta_{ijk}) \exp[\lambda_3^3 (r_{ij} - r_{ik})^3]$$

$$g(\theta_{ijk}) = 1 + c^2/d^2 - c^2/[d^2 + (h - \cos\theta_{ijk})^2]$$

A, B, λ_1, λ_2, λ_3,β, n, c, d, h – given parameters; R, D [Å] – interaction range

Embedded-Atom Method (EAM) – Johnson’s functional

- very useful for transition metals interactions (this is the fcc analytical form);
  - there are also separate forms for bcc and hcp structures.

$$E_t = \sum_i F(\rho_i) + \frac{1}{2}\sum_{i,j}{}' \Phi(r_{ij}); \qquad \rho_i = \sum_j{}' f(r_{ij})$$

where:
- E_t = total internal energy.
- ρ_i = the electron density at atom ***i*** due to all other atoms.
- f (r_{ij}) = the electron density at atom ***i*** due to atom ***j*** as a fct. of distance.
- r_{ij} = separation distance between atoms ***i*** and ***j***.
- F(ρ_i) = the energy to embed atom ***i*** in an electron density ρ_i.
- Φ(r_{ij}) = two-body potential between atoms ***i*** and ***j***.

- can also be used in the following form for all alloy potentials:

$$\Phi^{ab}(r) = \frac{1}{2}\left(\frac{f^b(r)}{f^a(r)}\Phi^{aa}(r) + \frac{f^a(r)}{f^b(r)}\Phi^{bb}(r) \right)$$

- electron density and two-body potential taken as decreasing exponentials:

$$f(r) = f_e \exp[-\beta(r/r_e - 1)];\ \ \Phi(\mathrm{r}) = \Phi_\mathrm{e} \exp[-\gamma(\mathrm{r}/\mathrm{r}_\mathrm{e} - 1)].$$

-the embedding function is determined by fitting to a universal eq. of state in the form of a Rydberg fct.:

$$F(\rho) = -E_c(1 - \ln x)x - 6\Phi_e y$$

where: $x = (\rho/\rho_e)^{\alpha/\beta};\ y = (\rho/\rho_e)^{\gamma/\beta};\quad \alpha = 3(\Omega B/E_c)^{1/2};\ \gamma = 15(\Omega G/\beta E_{UF})$

==

	Inputs					**Model parameters**				
Atom	Ω	E_c	E_{UF}	ΩB	ΩG	f_e	Φ_e	α	β	γ
Cu	11.81	3.54	1.30	10.17	4.05	0.30	0.59	5.09	5.85	8.00
Ag	17.10	2.85	1.10	11.10	3.61	0.17	0.48	5.92	5.96	8.26
Au	16.98	3.93	0.90	17.70	3.29	0.23	0.65	6.37	6.67	8.20
Ni	10.90	4.45	1.70	12.28	6.45	0.41	0.74	4.98	6.41	8.86
Pd	14.72	3.91	1.54	17.92	4.99	0.27	0.65	6.42	5.91	8.23
Pt	15.06	5.77	1.60	26.60	6.12	0.38	0.95	6.44	6.69	8.57

Modified Embedded Atom Method (MEAM, 2000-2010)

Main idea is to incorporate most fcc, hcp, bcc transition metals, as well as some gases (N, O) under the same analytical “umbrella”.

$$E_{total} = \sum_{i} \left[\sum_{j<i} \Phi_{ij}(r_{ij}) + F_i(\rho_i) \right] \quad \text{where} \quad F_i(\rho_i) = AE_c \frac{\rho_i}{\bar{\rho}_i} \ln \frac{\rho_i}{\bar{\rho}_i}$$

F_i is the embedding function, ρ_i is the background e^- density at lattice site i, and $\bar{\rho}_i$ is the background e^- density for a reference structure.

The improvement/novelty, as well as the difficulty, comes from considering the directionality in bonding through a combination of partial electron density terms, $\rho_i^{(h)}$, h = 1-3, for different angular contributions with weight factors $t^{(h)}$.

$$\left(\rho_i^{(0)}\right)^2 = \left[\sum_{j(\neq i)} S_{ij} \rho_j^{a(0)}(r_{ij}) \right]^2 \qquad \left(\rho_i^{(1)}\right)^2 = \sum_{\alpha} \left[\sum_{j(\neq i)} \frac{r_{ij}^{\alpha}}{r_{ij}} S_{ij} \rho_j^{a(1)}(r_{ij}) \right]^2$$

$$\left(\rho_i^{(2)}\right)^2 = \sum_{\alpha,\beta}\left[\sum_{j(\neq i)} \frac{r_{ij}^{\alpha}}{r_{ij}}\frac{r_{ij}^{\beta}}{r_{ij}} S_{ij}\rho_j^{a(2)}(r_{ij})\right]^2 - \frac{1}{3}\left[\sum_{j(\neq i)} S_{ij}\rho_j^{a(2)}(r_{ij})\right]^2$$

$$\left(\rho_i^{(3)}\right)^2 = \sum_{\alpha,\beta,\gamma}\left[\sum_{j(\neq i)} \frac{r_{ij}^{\alpha}}{r_{ij}}\frac{r_{ij}^{\beta}}{r_{ij}}\frac{r_{ij}^{\beta}}{r_{ij}} S_{ij}\rho_j^{a(3)}(r_{ij})\right]^2 - \frac{3}{5}\sum_{\alpha}\left[\sum_{j(\neq i)} \frac{r_{ij}^{\alpha}}{r_{ij}} S_{ij}\rho_j^{a(2)}(r_{ij})\right]^2$$

$\rho_j^{a(h)}(r_{ij})$ is the atomic e$^-$ density from atom j to site i, r_{ij}^{α} is the component of the distance vector between atoms j and i, with α = 1, 2, 3 representing x, y, z, and Sij is the screening function between atoms j and i.

$$\rho_j^{a(h)}(r) = f_{cutoff}(r)\exp\left[-\beta_j^h\left(\frac{r}{r_{j,e}} - 1\right)\right]$$

with $\beta_j^{(h)}$ pot. parameters and $r_{j,e}$ nearest equilibrium distance in reference structure.

$$f_{cutoff}(r) = \begin{cases} 1, & r \le r_c - \Delta r \\ \left[1 - \left(1 - \frac{r_c - r}{\Delta r}\right)^4\right]^2, & r_c - \Delta r < r < r_c \\ 0, & r_c \le r \end{cases}$$

Here r_c is the cutoff distance and Δr gives the cutoff region

Electron density is defined as:

$$\rho_j = \rho_j^{(0)} G(\Gamma); \quad \text{with} \quad G(\Gamma) = \frac{2}{1 + \exp(-\Gamma)}; \quad \text{and} \quad \Gamma = \sum_h t_j^{(h)} \left[\frac{\rho_j^{(h)}}{\rho_j^{(0)}} \right]^2$$

where $t_j^{(h)}$ are adjustable parameters.

The screening function S_{ij} is defined as the product of the product of the screening factor, $S_{ijk}(x_{ijk})$, due to all neighboring k atoms.

All summations are carried out over a finite range, typically including only 1st, or 1st and 2nd neighbors only.

Method is still evolving. Works reasonably well for most transition metals and a selection of gases. However, to push is to extend its transferability to binary and ternary compounds.

Difficulties arise from the fact that these materials have complex bonding, such as combinations of ionic-covalent-metallic-type bonds.

Latest important additions to MEAM family: TiN, TiC.

New Parameterization of the **Modified Embedded Atom Potential** for Large Scale Simulations of **TiN Thin Films Growth**

D.G. Sangiovanni, V. Chirita & L. Hultman

Thin Film Physics, Linköping University, Sweden

AVS 2010, Albuquerque, USA

TiN MEAM parameterization*

Fitted to bulk properties

- Lattice parameter
- Cohesive energy
- Elastic constants & moduli
- Thermal linear expansion

Surface properties & related phenomena to be verified

- Surface energies
- Surface relaxation

Critical tests for thin film growth simulations

- Adatom diffusion & Ehrlich barriers

* YM Kim & BJ Lee, Acta Materialia **56**, 3481 (2008)

Original parameters

	Single element system													
	E_c	r_e	B	A	$\beta^{(0)}$	$\beta^{(1)}$	$\beta^{(2)}$	$\beta^{(3)}$	$t^{(1)}$	$t^{(2)}$	$t^{(3)}$	C_{min}	C_{max}	d
Ti	4.87	2.92	110	0.66	2.70	1.0	3.0	1.0	6.80	-2.0	-12.0	1.00	1.44	0.00
N	4.88	1.10	5.96[#]	1.80	2.75	4.0	4.0	4.0	0.05	1.0	0.0	2.00	2.80	0.00

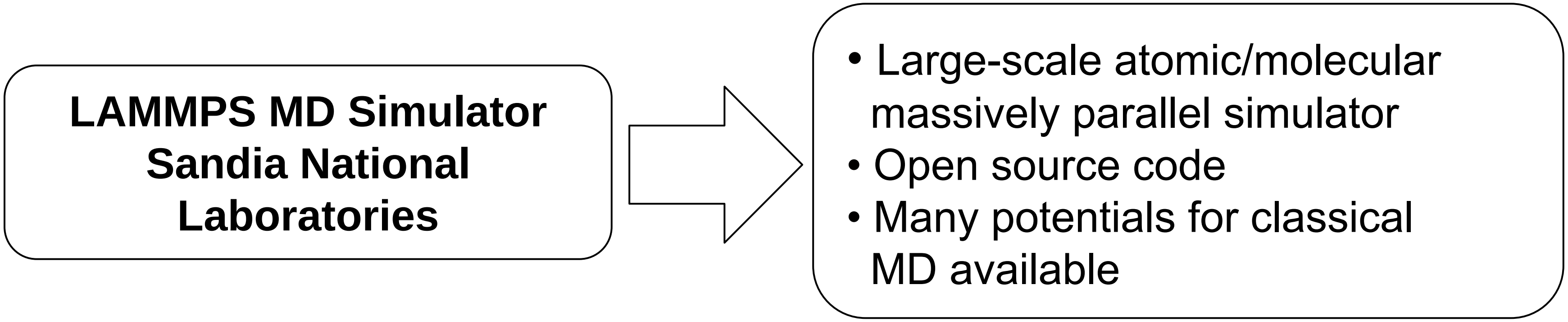
	Binary system											
	E_c	r_e	B	ρ_0 (N/Ti)	C_{min} (Ti,N,Ti)	C_{min} (N,Ti,N)	C_{min} (Ti,Ti,N)	C_{min} (Ti,N,N)	C_{max} (Ti,N,Ti)	C_{max} (N,Ti,N)	C_{max} (Ti,Ti,N)	C_{max} (Ti,N,N)	d
TiN	6.615	2.121	320	18	0.16	0.09	0.81	1.46	2.80	2.80	1.44	2.80	0.00

Units: $\boldsymbol{E_c}$ (eV), $\boldsymbol{r_e}$ (Å), ***B*** (GPa)

[#]alpha value

Reference structures: hcp (Ti), dimer (N), cubic-B1 (TiN)

LAMMPS & MEAM



LAMMPS MD Simulator Sandia National Laboratories

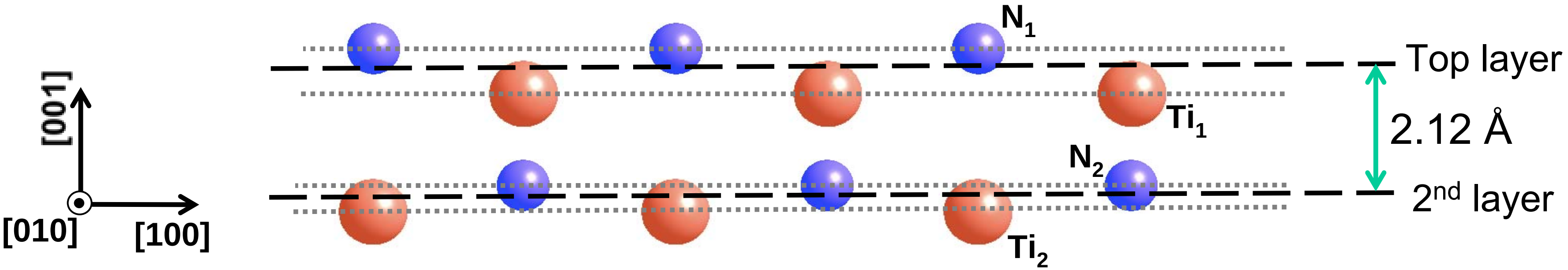
- Large-scale atomic/molecular massively parallel simulator
- Open source code
- Many potentials for classical MD available

	TiN Bulk				
	MEAM Original work	**Experimental**[1]	**DFT**[2]	**LAMMPS** Original param.	**LAMMPS** Own param.
$\boldsymbol{E_c}$ (eV)	-6.615			-6.608	-6.611
$\boldsymbol{a}$ (Å)	4.242	4.240	4.254	4.243	4.241
$\boldsymbol{B}$ (GPa)	320	318	290	319	294
$\boldsymbol{C_{11}}$ (GPa)	659	625	640	736	659
$\boldsymbol{C_{12}}$ (GPa)	150	165	115	101	112
$\boldsymbol{C_{44}}$ (GPa)	183	163	159	179	171

[1] Kim et al. J. App. Phys., **72** (1992)
[2] Sangiovanni et al. Phys. Rev. B, **81** (2010)

TiN surface results

	MEAM Original work	**DFT**[1,2]	**LAMMPS** Original param.	**LAMMPS** Own param.
E_{surf} (100)				
Not relaxed (meV/Å²)	111	105	95	105
Relaxed (meV/Å²)	81	86	67	84
E_{surf} (111)				
Not relaxed (meV/Å²)	245	311	232	236
Relaxed (meV/Å²)	228	226	211	217
Surf Relaxation (001)				
Δ dist. N_1-Ti_2 (%)	7.7	2.1	5.7	5.3
Δ dist. Ti_1-N_2 (%)	-5.0	-6.3	-5.5	-5.1
z dist. N_1-Ti_1 (Å)	0.26	0.18	0.24	0.21



Surface relaxation

[1] Marlo et al. Phys. Rev. B, **62** (2000)
[2] Dudiy et al. Phys. Rev. B, **69** (2004)

Diffusion energy barriers, previous results

	Energy Barrier (eV)
	Experimental[1,2]
TiN (100) Surface	
N adatom	1.1 - 1.4*****
Ti adatom	1.1 - 1.4*****
Ti-N_x admolecules	1.1 - 1.4*****
TiN (111) Surface	
N adatom	-
Ti adatom	-
Ti-N_x admolecules	-

	Energy Barrier (eV)	
	DFT[3]	**DFT**[4]
TiN (100) Surface		
N adatom	0.95	0.95
Ti adatom	0.35	0.43
Ti-N dimer	>0.88	-
TiN (111) Surface		
N adatom	-	-
Ti adatom	1.74	-
Ti-N dimer	-	-

***** 1.1 eV for f_{N2} = 0.1, 1.4 eV for f_{N2} = 1.0

[1] MA Wall et al., Phys. Rev. B, **70** (2004)

[2] MA Wall et al., Surf. Sci., **581** (2005)

[3] D Gall et al., J. App. Phys., **93** (2003)

[4] Present work, VASP, 15 x 15 x 1 **k** points, 5 layers of 16 atoms, PAW+GGA

Diffusion energy barriers, original parameterization

	Experimental	DFT	DFT Present work	LAMMPS Original Param.
N on TiN(001) (eV)	1.1 - 1.4*	0.95	0.95	0.4
Ti on TiN(001) (eV)	1.1 - 1.4*	0.35	0.43	0.6

	Single element system													
	E_c	r_e	B	A	$\beta^{(0)}$	$\beta^{(1)}$	$\beta^{(2)}$	$\beta^{(3)}$	$t^{(1)}$	$t^{(2)}$	$t^{(3)}$	C_{min}	C_{max}	d
Ti	4.87	2.92	110	0.66	2.70	1.0	3.0	1.0	6.80	-2.0	-12.0	1.00	1.44	0.00
N	4.88	1.10	5.96#	1.80	2.75	4.0	4.0	4.0	0.05	1.0	0.0	2.00	2.80	0.00

	Binary system												
	E_c	r_e	B	ρ_0 (N/Ti)	C_{min} (Ti,N,Ti)	C_{min} (N,Ti,N)	C_{min} (Ti,Ti,N)	C_{min} (Ti,N,N)	C_{max} (Ti,N,Ti)	C_{max} (N,Ti,N)	C_{max} (Ti,Ti,N)	C_{max} (Ti,N,N)	d
TiN	6.615	2.121	294	18	0.16	0.09	0.81	1.46	2.80	2.80	1.44	2.80	0.00

Diffusion energy barriers, own parameterization

	Experimental	**DFT**	**DFT** Present work	**LAMMPS** Orig. Param.	**LAMMPS** Own Param.
N on TiN(001) (eV)	1.1 - 1.4	0.95	0.95	0.4	0.7
Ti on TiN(001) (eV)	1.1 - 1.4	0.35	0.43	0.6	0.9

- Present work DFT results agree with existing DFT values
- MEAM diffusion barriers higher than DFT values
- MEAM values considerably closer to experimental barriers
- Qualitative agreement with experimental findings – no preferred diffusing species on (001)

Recommended CDIO project potential functionals

1. Lennard – Jones (LJ)

$$\Phi(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

- well suited for inert gases (metals), mostly for gas-metals interactions.

Values of LJ parameters

ε [10^{-16} erg]; **σ** [Å]

	He	**Ne**	**Ar**	**Kr**	**Xe**	**Ni**	**Ag**	**Rh**
ε	14	50	167	225	320	0.66 [eV]	0.34 [eV]	0.67 [eV]
σ	2.56	2.74	3.40	3.65	3.98	2.24	2.65	2.46

2. Morse potential

$$\Phi(r) = D\left(e^{-2\alpha(r-r_0)} - 2e^{-\alpha(r-r_0)}\right)$$

- D, r_0, α determined by fitting to reproduce bulk properties (lattice constant, cohesive energy, compressibility).

- used mostly for gas-metals (metals) interactions.

Values of parameters for Morse potential

D [eV]; r_0 [Å]; α [$Å^{-1}$]

	D	**r_0**	**α**
Ag	0.325	3.13	1.36
Rh	0.665	2.87	1.54

MD integrators

Selection criteria: ***simplicity, efficiency, stability, reliability, parallelization.***

Most integrators are based on Taylor expansion of particle coordinates r(t) around time, for small increments Δt, which is the MD time step.

Euler Algorithm

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2!m}\Delta t^2 + \frac{\Delta t^3}{3!}\dddot{r} + O(\Delta t^4) + ...$$

Algorithm is obtained by truncating the expansion beyond the Δt^2 term.

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

Similar to the Verlet algorithm but in fact much worse on all counts.

- it is not time-reversible
- introduces catastrophic energy drifts

Definitely NOT recommended.

Verlet Algorithm (Verlet, Störmer, 1967)

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2!m}\Delta t^2 + \frac{\Delta t^3}{3!}\dddot{r} + O(\Delta t^4) + \ldots.$$

$$r(t-\Delta t) = r(t) - v(t)\Delta t + \frac{f(t)}{2!m}\Delta t^2 - \frac{\Delta t^3}{3!}\dddot{r} + O(\Delta t^4) + \ldots.$$

Summing the above equations, one obtains:

$$r(t+\Delta t) + r(t-\Delta t) = 2r(t) + \frac{f(t)}{m}\Delta t^2 + O(\Delta t^4)$$

The equation for advancing the positions in Verlet algorithm becomes:

$$r(t+\Delta t) = 2r(t) - r(t-\Delta t) + \frac{f(t)}{m}\Delta t^2 \qquad (1)$$

The errors in the estimations of new positions are of order Δt^4.

r(t + Δt) and r(t - Δt) play symmetrical roles, the algorithm time-reversible.

Velocities are not required to compute trajectories.

Yet, velocities are needed to directly calculate kinetic energy and temperature.

Velocities can be computed from:

$$r(t+\Delta t) - r(t-\Delta t) = 2v(t)\Delta t + O(\Delta t^3)$$

and

$$v(t) = \frac{r(t+\Delta t) - r(t-\Delta t)}{2\Delta t} + O(\Delta t^2) \qquad (2)$$

The errors in velocities estimations are of order Δt^2.

Though algorithm is stable, one can improve this accuracy.

Leap-frog Algorithm (Hockney, 1970; Potter, 1972)

Derived from the Verlet algorithm, evaluates velocities at half-step:

$$v(t-\Delta t/2) = \frac{r(t) - r(t-\Delta t)}{\Delta t} \qquad \text{and} \qquad v(t+\Delta t/2) = \frac{r(t+\Delta t) - r(t)}{\Delta t}$$

New positions can be obtained as:

$$r(t+\Delta t) = r(t) + \Delta t v(t+\Delta t/2) \qquad (3)$$

Velocities can be computed from:

$$v(t+\Delta t/2) = v(t-\Delta t/2) + \Delta t\,\frac{f(t)}{m} \qquad (4)$$

Stored quantities are positions, r(t), and accelerations, a(t) = f(t)/m.
Equation (4) is implemented first, and once v(t + Δt/2) are obtained, velocities v(t) at current time step are calculated to permit the computation of quantities which require r(t) and v(t):

$$v(t) = \frac{1}{2}\left[v(t+\Delta t/2) + v(t-\Delta t/2)\right]$$

Upon completion of the half-step above, eq, (3) is used to caculate r(t + Δt).
Accelerations a(t) are also evaluated now and stored for next step.

Leap-from algorithm yields identical trajectories to those obtained with Verlet.

pros – significantly smaller numerical errors in velocities evaluation.

cons – positions and velocities are not evaluated at the same time step.

Velocity Verlet Algorithm (Swope, anderson et. al. 1982)

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 = r(t) + v(t)\Delta t + \frac{\Delta t^2}{2}a(t) \qquad (5)$$

$$v(t+\Delta t) = v(t) + \frac{f(t)+f(t+\Delta t)}{2m}\Delta t = v(t) + \frac{\Delta t}{2}\left[a(t) + a(t+\Delta t)\right] \qquad (6)$$

Implemented to store r(t), v(t) and a(t) **all** at the same time.

Similar to predictor-corrector algorithms, but equivalent to Verlet:

a: using (5) $$r(t+2\Delta t) = r(t+\Delta t) + v(t+\Delta t)\Delta t + \frac{f(t+\Delta t)}{2m}\Delta t^2$$

b: from (5) $$r(t) = r(t+\Delta t) - v(t)\Delta t - \frac{f(t)}{2m}\Delta t^2$$

a+b = c $$r(t+2\Delta t) + r(t) = 2r(t+\Delta t) + \left[v(t+\Delta t) - v(t)\right]\Delta t + \frac{f(t+\Delta t) - f(t)}{2m}\Delta t^2$$

(6) in c: $$r(t+2\Delta t) + r(t) = 2r(t+\Delta t) + \frac{f(t+\Delta t)}{m}\Delta t^2 \equiv \text{positions in Verlet alg.}$$

Velocity Verlet Algorithm

Requires storage of r(t), v(t), a(t).

Consists of two stages and a force evaluation in between:

Stage 1: r(t + Δt) are evaluated using (5) and v(t + Δt/2) using:

$$v(t+\Delta t/2) = v(t) + \frac{\Delta t}{2} a(t)$$

Intermediate stage: f(t + Δt) and a(t + Δt) are computed.

Stage 2: velocity move is completed.

$$v(t+\Delta t) = v(t+\Delta t/2) + \frac{\Delta t}{2} a(t+\Delta t)$$

Attractive trade-off between numerical stability, convenience, simplicity.

Ensures excellent energy conservation, solid trajectories evolution.

Nearly universal choice in available MD codes, includingQMD

Beeman Algorithm (1976)

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{2}{3}a(t)\Delta t^2 - \frac{1}{6}a(t-\Delta t)\Delta t^2 \qquad (7)$$

$$v(t+\Delta t) = v(t) + \frac{1}{3}a(t+\Delta t)\Delta t + \frac{5}{6}a(t)\Delta t - \frac{1}{6}a(t-\Delta t)\Delta t \qquad (8)$$

This scheme requires storage of r(t), v(t), a(t) and a(t - Δt).

Additional requirement to store a(t - Δt) compensated by improved energy conservation.

However, the algorithm is not time-reversible.

Velocity-Corrected Verlet Algorithm

Errors in predicted positions and velocities are both of order Δt^4.

Based on Taylor expansions for r(t + 2Δt), r(t + Δt), r(t - Δt) and r(t - 2Δt)

$$r(t+2\Delta t) = r(t) + 2v(t)\Delta t + \dot{v}(t)(2\Delta t)^2 \big/ 2! + \ddot{v}(t)(2\Delta t)^3 \big/ 3! + \ldots$$

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \dot{v}(t)\Delta t^2 \big/ 2! + \ddot{v}(t)\Delta t^3 \big/ 3! + \ldots$$

$$r(t-\Delta t) = r(t) - v(t)\Delta t + \dot{v}(t)\Delta t^2 \big/ 2! - \ddot{v}(t)\Delta t^3 \big/ 3! + \ldots$$

$$r(t-2\Delta t) = r(t) - 2v(t)\Delta t + \dot{v}(t)(2\Delta t)^2 \big/ 2! - \ddot{v}(t)(2\Delta t)^3 \big/ 3! + \ldots$$

By combining the above equations one obtains:

$$12v(t)\Delta t = 8\left[r(t+\Delta t) - r(t-\Delta t)\right] - \left[r(t+2\Delta t) - r(t-2\Delta t)\right] + O(\Delta t^4)$$

or

$$v(t) = \frac{v(t+\Delta t/2) + v(t-\Delta t/2)}{2} + \frac{\Delta t}{12}\left[\dot{v}(t-\Delta t) - \dot{v}(t+\Delta t)\right] + O(\Delta t^4) \qquad (9)$$

Significant improvement in errors but requires complex variables storage and calculations.

Predictor-Corrector Algorithms

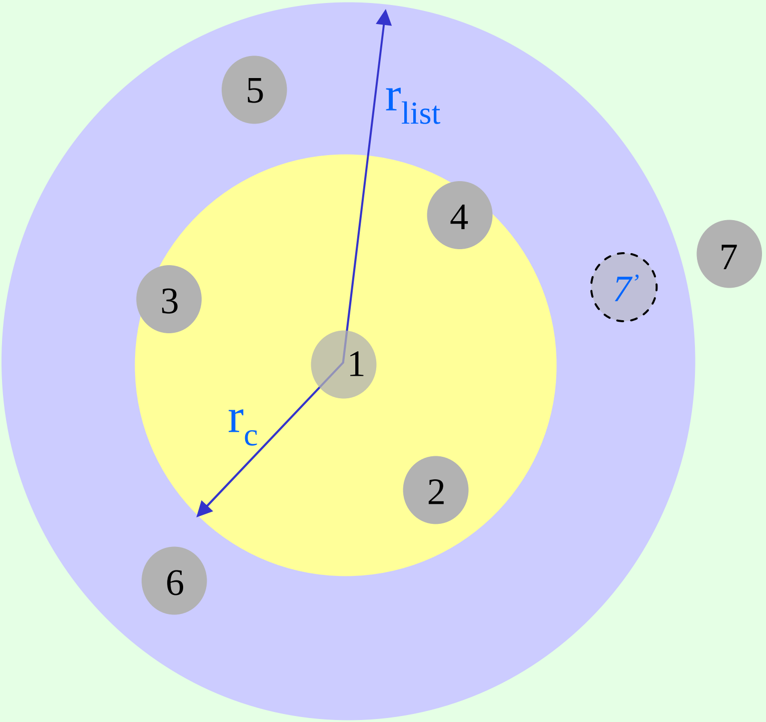
Several variants (Gear, etc), used mostly in conjunction with variable Δt.

Generally expensive in terms of CPU. More details in the book.

Neighbors Lists

Use of PBCs, potential truncation and MIC still leaves O(N2) pair interactions. Efficiency increases significantly if lists of neighbors are maintained & updated.

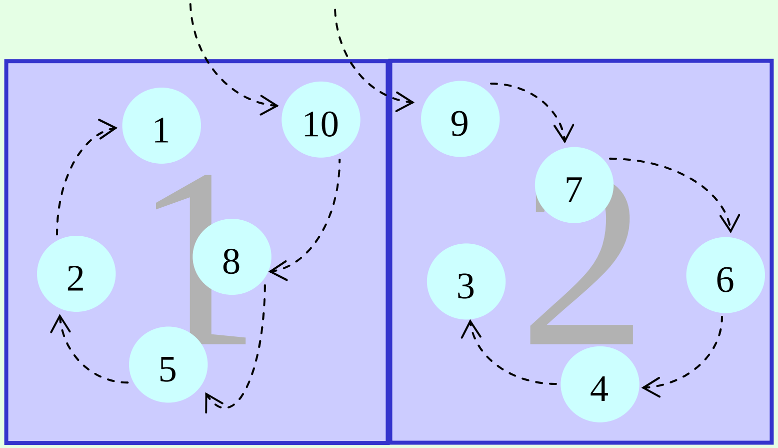
Verlet List



- r_{list} defines "*skin*" around cutoff sphere.
- only particles in cutoff & *skin* are kept.
- updating interval 10 – 100 time steps.
- larger *skin*, fewer updates, less efficient.
- method can reduce CPU time with 50%.
- one needs good balance between the size of the *skin* and updating interval.
- final choice dictated by studied problem, or phenomena.

Cells structures and linked cells

31	32	33	34	35	36
25	26	27	28	29	30
19	20	21	22	23	24
13	14	15	16	17	18
7	8	9	10	11	12
1	2	3	4	5	6



- primary cell divided in M x M x M, each cell has $N_c = N/M^3$ (in 3-dim).
- size of cells must be greater than cutoff distance.
- atoms in cell 14 only interact with those in cells 7, 8, 9, 13, 15, 19, 20 and 21.
- PBCs still apply to particles in cells 7, 13 and 19.
- two lists are made: HEAD-of-chain (cell id), and LIST of cell-head interactions.
- cells are linked. In can MD easily avoids double counting of ij interaction.
- number of interactions is reduced from N^2 (no MIC) and 1/2N(N-1) to $27NN_c$ in MC and $13.5NN_c$ in MD simulations.

Long range corrections

Can be added to compensate for potential truncation. For Lennard-Jones,

$V^{LJ}(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$ the average potential energy for any given atom i is:

$V_i(r) = \frac{1}{2}\int_0^\infty 4\pi r^2 \rho(r) V(r) dr,$ where $\rho(r)$ - average number density at r from i.

Tail contributions can be estimated as (if for $r \le r_c$, $\rho(r) \equiv \rho$):

$$V^{tail} = \frac{1}{2}\int_{r_c}^{\infty} 4\pi r^2 \rho(r) V(r) dr = \frac{1}{2} 4\pi\rho \int_{r_c}^{\infty} r^2 V(r) dr = \frac{8}{3}\pi\rho\varepsilon\sigma^3\left[\frac{1}{1}\left(\frac{\sigma}{r_c}\right)^9 - \left(\frac{\sigma}{r_c}\right)^6\right]$$

Reduced Units

Quite useful when dealing with systems containing one type of atoms.

Unit of mass = m, the mass of atoms in systems
For LJ potential: unit of length = σ; unit of energy = ε.

density: $\rho^* = \rho\sigma^3$

temperature: $T^* = k_B T/\varepsilon$

energy: $E^* = E/\varepsilon$

pressure: $P^* = P\sigma^3/\varepsilon$

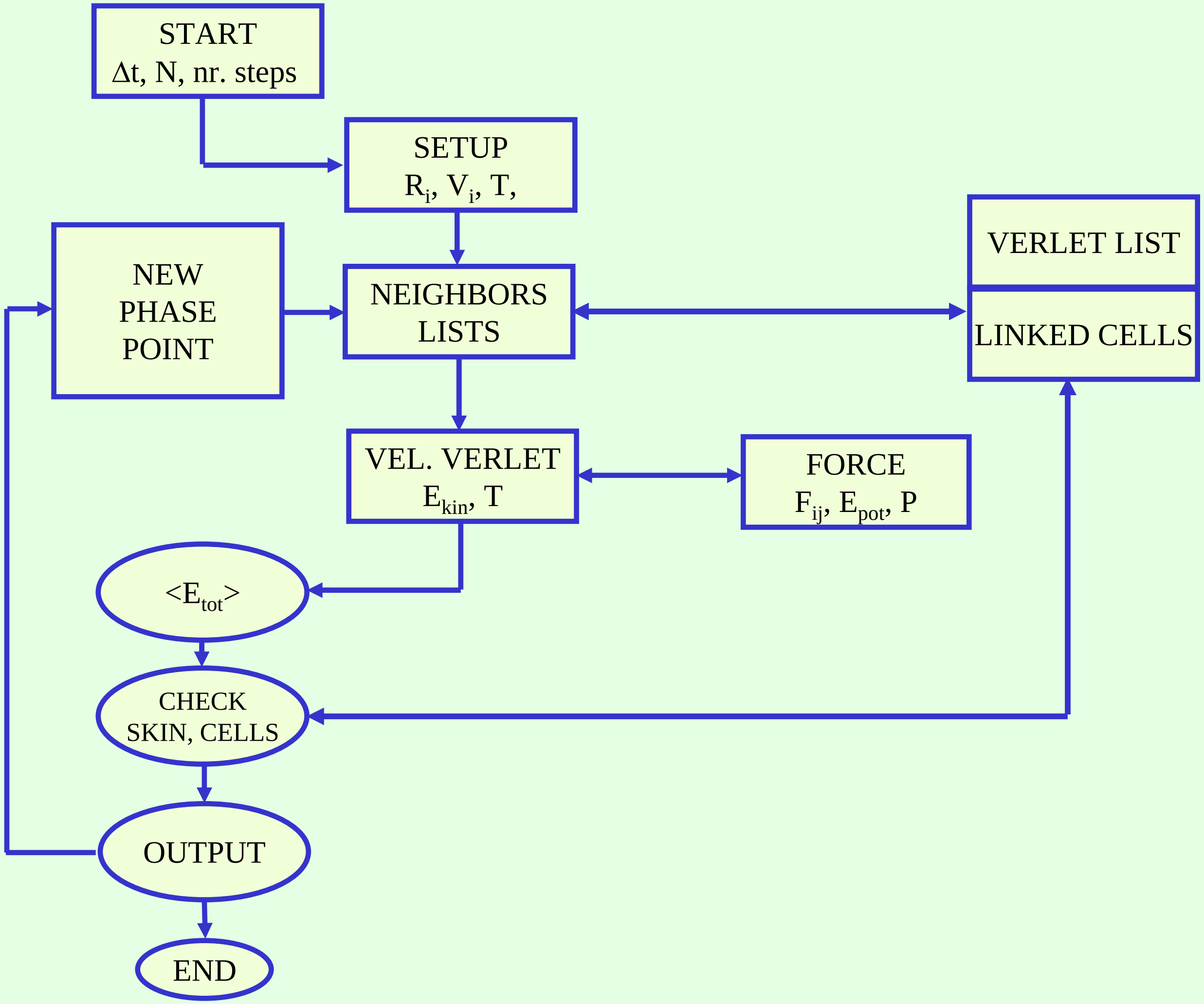
time: $t^* = \left(\varepsilon/m\sigma^3\right)^{1/2} t$

force: $F^* = F\sigma/\varepsilon$

$$V^{*LJ}(r) = 4\left[\left(\frac{1}{r^*}\right)^{12} - \left(\frac{1}{r^*}\right)^{6}\right]$$

OBS: Whatever units are used, one must be consistent!
Don’t mix eV with m, N with Å.

Molecular Dynamics Flow Diagram



How to setup MD

START

- Define MD box (X, Y, Z), density, MD step & number of steps (total time).

- Type of simulation: new **AND** old (useful for continuation runs).

- Starting temperature and potential parameters (sometimes useful here).

- Flags for storage of calculated physical quantities (required for time averages).

- Flags for specific requirements in MD runs:
  - thermostat or NO thermostat
  - heat sink or NO heat sink
  - interval for storage of coordinates (useful for visualization, movies)

- Flags for data OUTPUT (required at regular intervals but not as often as for visualization).

REMEMBER: IN/OUT statements slow done code execution!

INITIAL PHASE POINT

- Generate new AND/OR read existing atomic configurations (R_i, V_i).

- Provide for option to rescale velocities (required for simulations at new T).

- Calculate AND **zero** the total momentum (recommended).

DO THIS BEFORE THE MD LOOP

- Potential tabulation (if used) and potential parameters.

- Define heat sink parameters (number of fixed layers, rule for energy dissipation)

- Define the thermostat method/algorithm.

- Define (if used) any external action (eg. electric field, stress) on the system.

IMAGES/NEIGHBOURS LISTS

- Here one generates and stores the list of interactions to be used in the MD loop.

For good efficiency one must:

- Define and use Periodic Boundary Conditions (PBC)
    - **3-Dim** PBCs for **BULK** simulations
    - **2-Dim** PBCs for **SURFACE** simulations

- Provide for rapid counting of atoms (use Minimum Image Convention, Cells).

- Use potential truncation & shifting (based on potential range).

- **MAKE** and **STORE** the neighbors list (all valid PAIRS and TRIPLETS).

- Provide for possibility to update the list any time (every step OR every N steps).