

# シリコン系のシミュレーションプログラム集簡易マニュアル

2003-01-01 東京大学大学院工学系研究科機械工学専攻 泉 聡志

[izumi@fml.t.u-tokyo.ac.jp](mailto:izumi@fml.t.u-tokyo.ac.jp)

公開用の Tersoff potential の MD 計算プログラムの簡易マニュアルです。コンパイル方法は

```
%make -f si.mak (sih.mak - SiH 用)
```

です。Object ファイル格納用に必ず obj という dir を作成しておいてください。Alpha, Intel の Linux 用コンパイラで動作確認しています。かなりいいかげんな公開プログラムなのでバグ・不明な点はお問い合わせください。

<b>初期データ・設定</b> .....	3
初期データ.....	3
設定(para.inc).....	4
実行方法.....	4
原子が持つ物性.....	4
基本的パラメータ.....	4
配列宣言.....	5
無次元化.....	5
出力ファイルに関する説明.....	5
<b>ファイルの出入力に関するプログラム</b> .....	5
input.f ファイル入力プログラム.....	5
output.f ファイル出力プログラム.....	6
fopen.f ファイル設定プログラム.....	6
fileout.f 結果の出力.....	6
monitor.f モニター用サブルーチン 温度・エネルギー・応力・book のモニター.....	6
out.f アニメーション用座標出力.....	7
<b>book-keeping に関するプログラム</b> .....	7
book_period.f 周期境界用 book-keeping プログラム (少数原子用).....	7
prbook.f book 前処理プログラム、原子の領域を判定する.....	7
book_mlt.f マルチタイムステップ用 book-keeping プログラム (通常 MD 向け).....	8
<b>差分法のプログラム</b> .....	9
verlet.f verlet 差分法.....	9
verlet_npt.f Parinello-Rahman 法対応の verlet 差分法.....	9
verlet_mlt.f マルチタイムステップ対応 verlet 差分法プログラム.....	10
<b>ポテンシャルのプログラム</b> .....	11
tersoff_sih.f Tersoff ポテンシャル(Si-H 系)算出.....	11
tersoff_sih 用プログラム.....	11
tersoff.f Tersoff ポテンシャル(Si 系)算出.....	11
<b>物性値算出プログラム</b> .....	11
press.f 応力整理用プログラム.....	11
energy.f 特定領域の温度・ポテンシャルエネルギー算出.....	12
average.f 物性値平均用サブルーチン.....	12
<b>系の制御プログラム</b> .....	13
temp_adj.f 温度制御サブルーチン.....	13
balance.f 並進速度補正プログラム クラスタ計算等に有効.....	13
<b>初期設定プログラム</b> .....	14
set_atom_bond.f 原子・結合情報の管理(Si-H 系のみ).....	14
set_cut.f book-keeping のカットオフ距離の設定.....	14
matcal.f 行列予備計算 1-1-16.....	14

constant.f	ポテンシャル用パラメータ予備計算(Si Tersoff, Si-H Izumi potential) .....	14
set_npt.f	Parinello-Rahman 法のセッティング .....	15

## 初期データ・設定

### 初期データ

- posi.dat

xyz 形式。1 行目に原子個数、2 行目にコメント、3 行目から原子情報。Fortran でのフォーマットは(I2,6e21.13) で、原子種、原子座標(x,y,z)、原子速度(vx,vy,vz)が並ぶ。

原子は、各設定領域に属する原子の順に並べる。

結晶成長の場合、最初にクラスター領域、温度制御なしの基板領域、温度制御ありの基板領域、固定領域という順序で並ぶ。

865

```
1 0.1874516248681E+02 0.1820409496284E+02 0.2215589106094E+01 0.4526965403462E-10 .....
```

```
14 0.7270666884799E+00 -0.6658020625970E+00 -0.1963357604483E+00 -0.7288080985816E-01 .....
```

- lattice.dat

現在の格子の大きさ、基準となる格子の大きさを 3×3 で入力する。

- number.dat

```
1.0000000E-02 7.0000000E+02 0.0000000E+00 0.0000000E+00
```

```
1000 0 0 20 10 10 999
```

```
986 4 50.7000 1.4
```

```
50 648 144 144
```

```
1 1 1 0
```

```
0 0 1 0
```

```
51 986 1
```

```
0 0 0
```

1.step 2.temp 3.Wall weight 4.Ex.Press

5.total steps 6.Tadj 7.init 8.book-keep 9.anime out 10.mon. out 11. book out

12. all atoms 13. Number of region 14. ztop 15.cutdeg

16. Number of region 1-iatms (cluster atoms, sub. Atoms., temp. atoms , fixed atoms)

.....

1 タイムステップ 0.01 0.54[fs]

2 温度[K]

3 壁の重さの逆数 ..... Parinello-Rahman 法対応 0.0d0 を入れると NVT 法になる。

4 無効パラメータ 改行

5 計算ステップ数

6 温度制御の有無(0:無制御、1:制御、2:Simulated Anneal)

7 継続計算設定(0:新規、1:継続) 1 の場合 pforce.dat に力の情報が要ります。

8 book-keeping のステップ間隔の設定

9 アニメーションのステップ間隔の設定

10 物理量(温度等)のモニターのステップ間隔の設定

11 book の出力のステップ間隔の設定 改行

12 総原子個数(Ar は除く) iall

13 領域の数 ir\_atms

14 z 領域の最大値

15 領域分割の大きさの設定 ( 最大カットオフ長(bcut\_r) × cutdeg になる )

16 各領域の原子数 iatm(1) ~ iatm(ir\_atms)

17 各領域の原子が動くか否か( 1→moving ) imov(1) ~ imov(ir\_atms)

- 18 各領域の温度調整の有無 itemp(1) ~ itemp(ir\_atms)
- 19 バルクの最初の原子(is\_b)、最後の原子(ie\_b)、周期境界条件フラグ(1→周期 BC, 2→自由 BC)
- 20 1ならマルチステップ法、0なら通常のMD
- 21 1なら応力の計算なし、0なら応力を計算
- 22 0なら分子動力学(固定)

#### 設定(para.inc)

- 1 ICODE  
ICODE=1 で通常モード、0 でデバックモード。結合情報がカレントに出力される
- 2 zplot  
アニメーションの出力範囲を指定。2.0 で半分の領域が出力。3.0 は 1/3。
- 3 iregion (=10) 区別する領域の最大個数  
区別する領域の原子数、初期温度、温度、エネルギー  
iatm(iregion), tf(iregion), tnw(iregion), uk(iregion)
- 4 iave (=15) 01-03-08  
平均する物性値の最大個数 avr\_sc(iave), avr\_at(iave,iall)
- 5 indiv (=4) ndivn(indiv), ndiv2n(indiv)の大きさ
- 6 nbinfo(=5) nbrall(nbinfo, iall\*ncut)の大きさ
- 7 nval(=17) ..... メモリする物性値の最大個数 outmon(ndim\_mon,nval)
- 8 np8(=20),ncut(=20) 最大近接数(同じ?)
- 9 mltstep(=10) マルチタイムステップの分割
- 10 ibalance(=0) 並進速度補正を行なうかどうか(使うなら 1, 使わないなら 0)
- 11 mnok(=20) Max. # of kind of atom(mnok) = 20 (1:Si,2:H,3:C.....) for set\_atom\_bond.f
- 12 mnoa(=50) Max. atomic number(mnoa) = 50 ( for example As=33) for set\_atom\_bond.f
- 13 nbk(=20) Max. # of kind of bond(nbk) = 20 (1:Si-Si,2:Si-H,.....) for set\_atom\_bond.f
- 14 iout\_ex=0 xy.dat ( アニメーションの出力フォーマットの選択 )
- 15 i\_mi=1 Si-H ポテンシャルの選択 0:Marty, 1:izumi

#### 実行方法

実行の前に実行ファイルがあるディレクトリの下に入力・出力ファイル用のディレクトリを作り、入力ファイルを置く必要がある。実行ファイル名の後に、入力・出力ファイルが入るディレクトリ名を入力。ディレクトリ名を省略すると files というディレクトリ名になる。

%md si64

#### 原子が持つ物性

- x(i), y(i), z(i), xi(i), yi(i), zi(i) ..... 原子の座標 ( 仮想座標 )
- vx(i), vy(i), vz(i), vxi(i), vyi(i), vzi(i) ..... 原子の速度 ( 仮想速度 )
- fx(i), fy(i), fz(i) ..... 原子に働く力
- ddx2(i), ddy2(i), ddz2(i) .....規格化された原子に働く力 ( 継続計算の場合初期値が必要 )
- nkind(i) ..... 原子の種類 wg(i),wginv(i) ..... 原子の質量、質量の逆数

#### 測定するもの

- ekin(i), vij(i), stress(6,i) ..... 原子の運動エネルギー、ポテンシャルエネルギー、原子応力

#### 基本的パラメータ

- ntsteps ..... 総ステップ数 dt ..... タイムステップ nstep ..... 現在のステップ
- cutdeg ..... 領域の広さ nbstep ..... book-keeping の間隔ステップ
- itemp ..... 領域の温度制御情報 ntemp ..... 温度制御の有無

## 配列宣言

```
include 'poten.inc' ポテンシャルパラメータの情報 ( SW を使う場合は poten_sw.inc )
include 'cell.inc' MD セルに関する情報
include 'set_kind.inc' 原子・結合の種類を管理
include 'atom.inc' ! atomic information 原子情報
include 'book.inc' ! for book-keeping method Tersoff 用 book-keeping 用配列
include 'multi_step.inc' ! for multi-step method マルチタイムステップ用配列
上記の配列に対する allocate
include 'atom_alloc.inc'
include 'book_alloc.inc'
include 'multi_step_alloc.inc'
```

## 無次元化

- 距離  $x = x * 1.0 \times 10^{-10} [m]$
- 温度  $T = T * 1.1604858 \times 10^4 [K]$
- 重さ  $m = m * 4.6643445 \times 10^{-26} [kg]$
- エネルギー  $E = E * 1.60219 \times 10^{-19} [J]=[N \cdot m]$
- 時間  $t = t * 5.3955819 \times 10^{-14} [s]$
- 速度  $v = v * 1.853368 \times 10^3 [m/s]$
- 力  $F = F * 1.60219 \times 10^{-9} [N]$
- 応力  $= * 1.60219 \times 10^{11} [Pa] = * 160.219 [GPa]$
- 加速度  $a = a * 13.43497 \times 10^{16} [m/s^2]$

## 出力ファイルに関する説明

- ave\_atm.dat 原子の運動エネルギー・ポテンシャルエネルギー・位置(x,y,z)の平均値。average.f で設定
- ave\_sc.dat 領域毎の運動エネルギーと全エネルギー、応力値の平均値。average.f で設定
- error.dat エラー情報
- warning.dat warning 用法
- init\_data.dat 最終位置 posi.dat に対応
- init\_f.dat 最終の力情報 pforce.dat に対応
- init\_lat.dat 最終の格子サイズ lattice.dat に対応
- nc.dat book-keeping 情報 詳細は monitor.f を参照
- str\_sub.dat 領域毎の応力情報
- tmp-ene.dat 領域毎の温度・エネルギー情報
- xy.dat アニメーション用 xyz ファイル

## ファイルの出入力に関するプログラム

input.f ファイル入力プログラム

subroutine input

(iall,nstat,nkind ,x,y,z,vx,vy,vz,xi,yi,zi,vxi,vyi,vzi,fx,fy,fz,ddx2,ddy2,ddz2,wg,wginv,hmat,hmat0,hmati,vol,hmat0i,vol0)

入力)

nstat, iall

出力)

- posi.dat より nkind, x, y, z, vx, vy, vz
  - lattice.dat より hmat, hmat0
  - pforce.dat より ( nstat=1 のとき ) fx, fy, fz, ddx2, ddy2, ddz2
- wg, wginv,

hmati, vol を計算して、

- 1 . xi = h0-1 x より、規格化座標を計算
- 2 . x = h xi より実座標を計算 ( h0 h の時は変形が起こるように対応！ )

xi, yi, zi, vxi, vyi, vzi を計算する。

output.f ファイル出力プログラム

subroutine output(iall,nkind,x,y,z,vx,vy,vz,fx,fy,fz,ddx2,ddy2,ddz2,hmat,hmat0)

入力) 全パラメータ

出力)

init\_data.dat ..... 座標・速度

init\_lat.dat ..... セルの形状マトリックス

init\_f.dat ..... 力

fopen.f ファイル設定プログラム

subroutine fopen(arg,f\_number,ifile,index)

入力)

arg ..... ディレクトリ名

ifile ..... f\_number の次元

index ..... arg の文字数

出力)

f\_number ..... ファイル名

ファイルを開放する

implantation 用のみ posi.dat を追加

fileout.f 結果の出力

subroutine fileout(iall,iatms,ndim\_mon,ndim\_out,outmon,outxy,imon,ioutxy)

入力)

iall, iatms, ndim\_mon, ndim\_out,

imon, ioutxy ..... outmon に格納された個数、outxy に格納された個数

outmon(ndim\_mon,17), outxy(ndim\_out,4,iall0+1)

出力)

ファイルに出力 36→ tmp-ene.dat, 24→ str\_sub.dat, 21→ xy.dat

monitor.f モニター用サブルーチン 温度・エネルギー・応力・book のモニター

subroutine monitor(iall,is\_b,ie\_b,iatm,iatms,stress,tnw,uk,outmon,nc,nbr,npbx,npby,npbz,nbrp,rtest,nstep,nout\_nc,ndim\_mon,imon,files)

入力)

iall, iend, iatm, iatms, stress, tnw, uk, nc, nbr, npbx, npby, npbz, rtest,

imon ..... 配列 outmon の順番

nout\_nc ..... book を出力する間隔 ( 十分大きく！ )

ndim\_mon ..... 配列 outmon のサイズ、総ステップと nout\_mon から決められる。

files ..... book が入るファイル名

is\_b ..... バルク領域が始まる初期番号

ie\_b ..... バルク領域が終わる番号 この間の平均応力を計算する。

出力)

outmon(ndim\_mon,17)..... 17 個まで変数を収納できる配列 fileout.f で出力される

温度が有効領域数(iatms)個、全エネルギーが有効領域数個、応力が 6 個

(例) 結晶成長 (iatm=3) × 2+6 = 12, imon でいくつかの個数が入っているかを管理する。

out.f アニメーション用座標出力

*subroutine out(iall,iall0,istr,iend,ndim\_out,nkind,ioutxy,x,y,z,outxy)*

入力)

iall, iall0, nkind, x, y, z

ndim\_out ..... アニメーションのコマ数 総ステップと nout\_out から決められる。

出力) istr – iend の範囲!

ioutxy .....現在のコマ番号

outxy(ndim\_out,4,iall0+1)..... アニメーションの格納配列 outxy(ioutxy,1,1)に原子個数が入る

## book-keeping に関するプログラム

book\_period.f 周期境界用 book-keeping プログラム (少数原子用)

*subroutine book\_period*

*(idel,istr\_b,iend\_b,iall,ndivxn,ndivyn,ndivzn,ndivx,ndivy,ndivz,nkind,nc,nbr,npbx,npby,npbz,nbrall,nbrp,nbrmax,iddm,nbstep,nstep,x,y,z,xi,yi,zi,vx,vy,vz,gmat,hmat,*

*rtest,rx,ry,rz,rbook,cutdeg,dt,bc\_si,dmarg1,dmarg2,bcut\_r)*

入力)

iall, nkind(iall), x,y,z,xi,yi,zi,vx,vy,vz,gmat(3,3),hmat(3,3), cutdeg, dt, nbstep, nstep,

dmarg1, dmarg2 .....相対速度 book-keeping のマージンパラメータ

bc\_si ...未使用, bcut\_r .....領域分割長さ

idel .....分割 book-keeping に使用。1 なら book を初期化しない (0 で初期化)

istr\_b, iend\_b ..... Book を作る原子の範囲

出力)

nc(i) ..... 原子 i のまわりの近接原子数

nbr(nci,i) .....原子 i の nci 番目の近接原子の番号

nbrp(nci,i) .....原子 i の nci 番目の近接原子との結合番号

nbrmax ..... 総結合番号数

npbx(nci,i), npby(.), npbz(.)...原子 i の nci 番目の近接原子との x 方向周期境界条件情報 (全 0)

nbrall(1,n) ..... n 結合番号の i 原子

nbrall(2,n) ..... n 結合番号の j 原子

nbrall(3,n) n 結合番号が i 原子にとって、何番目の結合か?

nbrall(4,n) n 結合番号が j 原子にとって、何番目の結合か?

nbrall(5,n) n 結合番号の種類 (1-7) 1:Si-Si, 2:Si-H, 3:H-H, 4:C-C 5: C-C(T), 6:Si-C, 7:C-H

ndivxn, ndivyn, ndivzn ..... 領域分割数 x,y,z 方向

ndivx(i), ndivy(i), ndivz(i) ..... i 原子が属する領域

計算用パラメータ)

iddm(iall), rx(iall), ry(iall), rz(iall), rbook(iall), rtest(iall\*ncut)

原子結合の見分けを kbnd = npairbond(nk1,nk2) で行なう。

prbook.f book 前処理プログラム、原子の領域を判定する。

*subroutine prbook(iall,ndivx,ndivy,ndivz,ndivxn,ndivyn,ndivzn,x,y,z,xi,yi,zi,hmat,cutdeg,bcut\_r)*

入力)

iall, x, y, z, xi, yi, zi, hmat, cutdeg,

bcut\_r

出力)

ndivxn, ndivyn, ndivzn ..... 領域分割数

ndivx, ndivy, ndivz(iall) ..... 各原子の領域番号

book\_mlt.f マルチタイムステップ用 book-keeping プログラム ( 通常 MD 向け )

subroutine book\_mlt

(iall, istr\_b, iend\_b, iatm, iperiod,

nkind, nc, nbr, npbx, npby, npbz, nbrall, nbrp, nbrmax, rtest, nkind\_m, nc\_m, nbr\_m, npbx\_m, npby\_m, npbz\_m, nbrall\_m, nbrp\_m, nbrmax\_m, mltf, imltf, imltf2, rtest\_m, mltn, mltfn, cutdeg, xi, yi, zi, gmat)

multmeth=1 の時動作

入力 )

iall, nkind, nc, nbr, npbx, npby, npbz, nbrall, nbrp, nbrmax, rtest, xi, yi, zi, gmat, cutdeg

出力 )

mltn ..... マルチタイムステップ原子数

mltn ..... マルチタイムステップ原子用力領域

mltf(i) ..... i 番目の力領域原子番号(1~mltn はマルチステップ領域)

imltf(j) ..... j 原子は何番目のマルチタイム力領域の原子になるか ?

imltf2(j) ..... j 原子がマルチタイム力 ? 領域に含まれていれば 1, それ以外は 0

nkind\_m, nc\_m, nbr\_m, npbx\_m, npby\_m, npbz\_m, nbrall\_m, nbrp\_m, nbrmax\_m, rtest\_m

マルチステップ領域の book 情報 ( 全体の book から作る )

para.inc の ICODE=1 で、nc\_m.dat に力領域全体の book を出力

マルチステップを行う領域(マルチ領域)の選定(水素のみ登録)、マルチステップ領域の力を計算するための領域(マルチ力領域)の選定 ( cutdeg を使って範囲を設定 )。マルチ力領域の局所原子情報・book 作成。周期境界条件対応

系に応じてその都度設定しなければならない!!!

例えば、クラスタ領域(iatm(1))を登録、それ以外の領域の水素を登録 と設定する等



## 差分法のプログラム

verlet.f verlet 差分法

subroutine verlet(*ifstr,iall,istr,iend,nkind,nc,nbr,npbx,npby,npbz,nbrall,nbrp,nbrmax,dt,ddx2,ddy2,ddz2,hmat,hmati,fx,fy,fz,dx,dy,dz,x,y,z,xi,yi,zi,vx,vy,vz,vxi,vyi,vzi,stress,vij,wginv,wg,ekin*)

入力)

*ifstr* ..... 応力計算フラグ(0 なら計算する)

*istr* ..... verlet 適用開始原子

*iend* ..... verlet 適用最終原子 (*istart* ~ *iend* 以外の原子は動かさない)

*dt* ..... 時間ステップ

*ddx2,ddy2,ddz2,fx,.,fy,fz,x,y,z,xi,yi,zi,vx,vy,vz,vxi,vyi,vzi,stress,vij,wginv,wg* ..... 原子データ

*hmat(3,3), hmati(3,3)* 格子マトリクス

*nkind,nc,nbr,npbx,npby,npbz,nbrall,nbrp,nbrmax* ..... tersoff 用 book データ(*book\_r.f* より)

*contx,conty,contz* ..... 原子拘束情報

*ekin* ..... 原子の運動エネルギー (not used)

出力)

*x,y,z,xi,yi,zi,vx,vy,vz,vxi,vyi,vzi, ddx2,ddy2,ddz2,fx,.,fy,fz* 原子間力・速度・座標

*vij, stress* 原子エネルギー・応力

*ddx2, ddy2, ddz2* .....  $h^{-1}$  F/m

*calc\_force.f* と *press.f* を含む

verlet\_npt.f Parinello-Rahman 法対応の verlet 差分法

subroutine verlet\_npt(*ifstr,iall,istr,iend,*

*cccc for tersoff\_h.f*

*nkind,nc,nbr,npbx,npby,npbz,nbrall,nbrp,nbrmax,*

*dt,ddx2,ddy2,ddz2,hmat,hmati,fx,fy,fz,dx,dy,dz,x,y,z,xi,yi,zi,vx,vy,vz,vxi,vyi,vzi,stress,vij,wginv,wg,*

*cccc for verlet\_npt.f*

*ddx1,ddy1,ddz1,smat,hmatt,hmatti,gmat,gmati,sigma,apst0,vol,vgmat,vhmat,stm1,stm2,hmat0,wm*)

*wm* 0.0d0 で実行される。

verlet.f との相違点

入力)

必ず *set\_npt.f* が必要。

*smat(3,3)* ..... 設定応力テンソル

*wm* ..... 壁の重さの逆数

*ddx1(iall), ddy1(iall), ddz1(iall)* .....  $G^{-1} dG/dt$

*stm1(3,3)* .....  $Vh^{-1}$

*stm2(3,3)* .....  $h$  初期値 0 を引き渡す。

出力)

*hmat(3,3), vhmat(3,3)* ..... 格子マトリクス・格子マトリクスの速度

*vol* .....  $V$ , *hmati(3,3)* .....  $h^{-1}$ , *hmatt(3,3)* .....  $h^1$ , *hmatti(3,3)* .....  $h^{-1}$ , *gmat(3,3)* .....  $G=h^1h$ ,

*gmati(3,3)* .....  $G^{-1}$ , *sigma(3,3)* .....  $A=Vh^{-1}$ , *apst0(3,3)* .....  $Vh^{-1} h^{-1}$ , *vgmat(3,3)* .....  $dG/dt$ ,

*ggmat1(3,3)* .....  $G^{-1} dG/dt$

*tersoff\_h.f, press.f, matcal.f* を含む。

verlet\_mlt.f マルチタイムステップ対応 verlet 差分法プログラム

subroutine verlet\_mlt(ifstr,iall,istr,iend,

#for tersoff\_h.f (same as verlet.f)

nkind,nc,nbr,npbx,npby,npbz,nbrall,nbrp,nbrmax,

#for verlet\_mlt.f (same as verlet.f)

dt,ddx2,ddy2,ddz2,hmat,hmati,fx,fy,fz,dx,dy,dz,x,y,z,xi,yi,zi,vx,vy,vz,vxi,vyi,vzi,stress,vij,wginv,wg,

#for multistep

nkind\_m,nc\_m,nbr\_m,npbx\_m,npby\_m,npbz\_m,nbrall\_m,nbrp\_m,nbrmax\_m,

mltf,imltf,imltf2,rtest\_m,mltn,mltn,

ddx2\_m,ddy2\_m,ddz2\_m,x\_m,y\_m,z\_m,vx\_m,vy\_m,vz\_m,xi\_m,yi\_m,zi\_m,vxi\_m,vyi\_m,vzi\_m,fx\_m,fy\_m,fz\_m,dx\_m,dy\_m,dz\_m,wginv\_m)

入力)

ifstr ..... 応力計算フラグ(0 なら計算する)

istr ..... verlet 適用開始原子

iend ..... verlet 適用最終原子 (istart ~ iend 以外の原子は動かさない)

dt ..... 時間ステップ

ddx2,ddy2,ddz2,fx.,fy,fz,x,y,z,xi,yi,zi,vx,vy,vz,vxi,vyi,vzi,stress,vij,wginv ,wg ..... 原子データ

hmat(3,3), hmati(3,3) 格子マトリクス

nkind,nc,nbr,npbx,npby,npbz,nbrall,nbrp,nbrmax ..... tersoff 用 book データ(book\_r.f より)

・ book\_mltr.f から

mltn, mltfn, mltf(iall), imltf(iall), imltf2(iall),

nkind\_m, nc\_m, nbr\_m, npbx\_m, npby\_m, npbz\_m, nbrall\_m, nbrp\_m, nbrmax\_m, rtest\_m

出力)

x,y,z,xi,yi,zi,vx,vy,vz,vxi,vyi,vzi, ddx2,ddy2,ddz2,fx.,fy,fz 原子間力・速度・座標

vij, stress 原子エネルギー・応力

計算用パラメータ)

全原子データから力領域データを抜き出し、計算後全原子データへ埋め込む。

ddx2\_m,ddy2\_m,ddz2\_m,x\_m,y\_m,z\_m,vx\_m,vy\_m,vz\_m,xi\_m,yi\_m,zi\_m,vxi\_m,vyi\_m,vzi\_m,fx\_m,fy\_m,fz\_m,dx\_m,dy\_m,dz\_m,wginv\_m

プログラム解説)

マルチステップ対応の verlet 差分法 (必要のないものも含まれている)

1. マルチ領域以外の速度の更新 + 実空間への写像
2. マルチ力領域の局所原子情報・book 作成
3. マルチタイムステップ開始
4. マルチ領域以外の座標の更新(速度は一定)
5. マルチ領域の座標更新(含むイオン)
6. マルチ力領域の座標更新(マルチ領域の力の計算にのみ使用)
7. マルチ領域の速度の更新
8. マルチ力領域の座標の実空間への写像
9. マルチ力領域の力の計算(tersoff) (molire)
10. マルチ領域の加速度計算
11. マルチ領域の速度の更新
12. マルチタイムステップ終了 (3)へ
13. マルチ力領域の速度と座標の実空間への写像
14. 全領域の座標の実空間への写像

15. マルチ領域 全領域へ原子情報の置き換え
16. 全領域の力の計算(tersoff) (molire)
17. 全領域の加速度計算
18. マルチ領域以外の速度の更新 + 実空間への写像

## ポテンシャルのプログラム

**ポテンシャルのプログラムは calc\_force というサブルーチン名に統一。元素対応はコンパイル時に行う！**

tersoff\_sih.f Tersoff ポテンシャル(Si-H 系)算出

*subroutine calc\_force*

(ifstr,iall,istr,iend,nkind,nc,nbr,npbx,npby,npbz,nbrall,nbrp,nbrmax,x,y,z,fx,fy,fz,hmat,stress,vij)

入力)

iall, ifstr 全原子数、応力計算フラグ(0 なら計算する)

nkind,nc,nbr,npbx,npby,npbz,nbrall,nbrp,nbrmax ..... tersoff 用 book データ

x,y,z,hmat 座標と格子マトリックス

出力)

fx(iall), fy(iall), fz(iall) ..... 全原子に働く力

vij(iall) ..... 原子エネルギー

stress(6,iall) ..... 原子応力 para.inc 中の ifstr=0 で計算、1 で非計算

tersoff\_sih 用プログラム

1. calc\_st.f 応力算出プログラム

*subroutine calc\_st(stress,rxn,ryn,rzn,ffx,ffz,I,j,iall)*

入力) rxn,ryn,rzn,ffx,ffz,I,j,iall 出力) stress(6,iall)

2. fspln.f スプラインプログラム Marty potential の F1(N),F2(N)を求める

*subroutine fspln(f1\_bnd,f2\_bnd,fn\_tot,fn\_tot2,nk1,nk2)*

入力) nk1,nk2,fn\_tot 出力) f1\_bnd,f2\_bnd

kbnd=1 で有効となり、それ以外では f1\_bnd=f2\_bnd=1.0d0 とする。

3. fspgh.f スプラインプログラム Marty potential の H(N)を求める

*subroutine fspgh(gh,fn\_tot,nk1,nk2,nk3)*

入力) nk1,nk2,nk3,fn\_tot 出力) gh

4. bparam3.f 3 体依存パラメータの設定

*subroutine bparam3(bpar,par\_ss,par\_sh1,par\_sh2,par\_hh,nk1,nk2,nk3)*

5. 2 体依存パラメータ用配列

*da1(nbk),db2(nbk),damd1(nbk),damd2(nbk),dcutr(nbk),dcutd(nbk),dbcut(nbk),dreq(nbk),denu(nbk),denu2(nbk)*

6. fssfunc.f Si-Si 二重結合補正パラメータ設定

入力) 原子対の種類 1:Si-Si, 2:Si-H, 3:H-H

出力) 各パラメータの値

tersoff.f Tersoff ポテンシャル(Si 系)算出

*subroutine calc\_force*

(ifstr,iall,istr,iend,,nkind,nc,nbr,npbx,npby,npbz,nbrall,nbrp,nbrmax,x,y,z,fx,fy,fz,hmat,stress,vij)

入力) tersoff\_sih.f と同じ

出力) tersoff\_sih.f と同じ

Si のみの系に対応

## 物性値算出プログラム

press.f 応力整理用プログラム

*subroutine press(iall,istr,iend,nkind,vx,vy,vz,stress,hmat,wg)*

入力) iall,istr,iend,nkind,vx,vy,vz,hmat,wg  
出力) stress(6,iall) para.inc 中の ifstr=0 で計算、1 で非計算  
原子応力算出手法を変更  
原子体積 = hmat(1,1)\*hmat(2,2)\*hmat(3,3)/float(iall)とした。

energy.f 特定領域の温度・ポテンシャルエネルギー算出  
subroutine energy(iall,istr,iend,iatm,iatms,vx,vy,vz,wg,vij,ekin,tnw,uk)

入力)  
iall, iend, vx, vy, vz, wg, vij, ekin,  
iatm(iregion) ..... 特定領域の原子数  
iatms ..... 特定領域の数  
iend ..... 計算対象原子数 (固定原子を除く)  
出力)  
tnw(iregion) ..... 特定領域の平均温度  
uk(iregion) ..... 特定領域の平均ポテンシャルエネルギー

average.f 物性値平均用サブルーチン  
subroutine average(iall,iall0, is\_b,ie\_b,istr,iend,iatm,iatms,x,y,z,avr\_sc,avr\_at,tnw,uk,vij,ekin,stress,navestep,nstep)

入力)  
iall, iall0, iend, iatm, iatms, x, y, z, tnw, uk, vij, ekin, ,stress, navestep, nstep  
出力)  
iave=10(para.inc)  
avr\_sc(iave) スカラー平均 iatms 分の温度と総エネルギー ( iatms × 2 ) + 1 ~ iend の応力 6 成分  
avr\_at(iave,iall0) 原子平均 原子運動エネルギー、全エネルギー、 x, y, z 座標 ( 5 個 )  
1 ~ iend の有効原子のみの平均

## 系の制御プログラム

temp\_adj.f 温度制御サブルーチン

*subroutine temp\_adj(iall,iatm,iatms,itemp,vx,vy,vz,vxi,vyi,vzi,hmat,tf,tnw)*

入力)

iall, iatm, iatms, vxi, vyi, vzi, hmat, tnw

tf(iregion) ..... 特定領域の初期温度

itemp(iregion) ..... 特定領域の温度制御フラグ =1 で温度制御

出力)

vx, vy, vz, vxi, vyi, vzi ..... itemp で設定された領域のみ温度制御する。

balance.f 並進速度補正プログラム クラスタ計算等に有効

*subroutine balance(iall,istr,iend,vx,vy,vz,vxi,vyi,vzi,hmat)*

入力)

iall, istr, iend, vxi, vyi, vzi, hmat        istr, iend で補正する領域を指定

出力)

vxi, vyi, vzi, vx, vy, vz

バルクなら is\_b ~ is\_e を制御する。クラスタなら 1 ~ iall を制御

## 初期設定プログラム

set\_atom\_bond.f 原子・結合情報の管理(Si-H系のみ)

```
subroutine set_atom_bond
  include 'para.inc'
  include 'set_kind.inc'
cccccc atom mass --> atom number
  kind_atm = 0
  kind_atm(14) = 1; inkind(1) = 14 ! Si
  kind_atm(1) = 2; inkind(2) = 1 ! H
c
  simass = 28.09d0
  wg_si = 1.0d0 ! 28.09
  wg_h = 1.008d0/simass ! 1.008
c
  wg_kind(1) = wg_si
  wg_kind(2) = wg_h
cccccc i-j pair --> bond number
  npairbond=0
  npairbond(1,1) = 1 ! Si-Si bond
  npairbond(1,2) = 2 ! Si-H bond
  npairbond(2,1) = 2 !
  npairbond(2,2) = 3 ! H-H bond
  return
end
```

set\_cut.f book-keeping のカットオフ距離の設定

*subroutine set\_cut(multmeth,iall,nkind,tf\_sub,dt,nbstep,bc\_si,dmarg1,dmarg2,bcut\_r)*

入力)

multmeth, tfsb, dt, nbstep

出力)

bc\_si ..... シリコンのカットオフマージン (未使用)

bcut\_r ..... 領域分割の大きさ カットオフ距離が長いものを採用 (通常はシリコン dbcut(1) )

dmarg1 ..... 相対速度 book-keeping の接近原子のカットオフマージン

dmarg2 ..... 相対速度 book-keeping の解離原子のカットオフマージン

matcal.f 行列予備計算 1-1-16

*subroutine matcal*

*(hmat,smat,hmati,hmatt,hmatti,gmat,gmati,sigma,apst0,vol,vgmat,vhmat)*

入力)

hmat(3,3), smat(3,3) vhmat(3,3) 格子マトリクス h、外応力マトリクス s、格子マトリクス速度

出力)

vol ..... V, hmati(3,3) .....  $h^{-1}$ , hmatt(3,3) .....  $h^t$ , hmatti(3,3) .....  $h^{-t}$ , gmat(3,3) .....  $G=h^t h$ ,

gmatti(3,3) .....  $G^{-1}$ , sigma(3,3) .....  $A=Vh^{-t}$ , apst0(3,3) .....  $Vh^{-1} h^{-t}$ , vgmat(3,3) .....  $dG/dt$

constant.f ポテンシャル用パラメータ予備計算(Si Tersoff, Si-H Izumi potential)

*subroutine constant*

出力) poten.inc

set\_kind.inc (book-keeping に使用)

```
common /dpa2/ dbcut(nbk)
common /bc/ bcutss,bcutsh,bcuthh
&          ,bcutcc,bcutcc2,bcutcs,bcutch
```

set\_npt.f Parinello-Rahman 法のセッティング

subroutine set\_npt(iall,ddx1,ddy1,ddz1,stm1,stm2,smat,arg)

wm 0.0d0 で実行される。

入力)

iall, arg

出力)

ddx1, ddy1, ddz1, stm1, stm2 ..... 初期値ゼロを設定

'smat.dat'が存在すれば、smat(3,3)を読み込む。存在しなければ smat=0.0d