分子シミュレーション 夏の学校　　2002 年 7 月 27 日(土)-29 日(月)

東京工業大学 地球惑星科学専攻
河村 雄行

<1．はじめに>

<2．原子・分子間相互作用の形式>

希ガス分子間相互作用から、イオン結合、共有結合に対応する、
原子・分子間相互作用モデルの形式を議論する。
３体相互作用、シェルモデル、ブリージングモデル等にも言及する。
また水の各種モデルを概観する。

<3．経験的パラメータの最適化>

結晶構造（格子定数と原子座標)、温度－密度、圧力－密度などの状態方程式、
赤外吸収・ラマン散乱などのスペクトル、などの実験データを用いた、相互作用
モデル関数のパラメータの最適化を議論する。

<4．半経験的パラメータの最適化>

分子軌道法を用いた、原子・分子間相互作用モデル関数のパラメータの
最適化を議論する。２原子分子を中心にする（固体結晶は土屋による)。

<5．モデルポテンシャルの評価・適用（無機-有機複合系のために)>

吸着、固溶体、粘性、誘電率等のさまざまな性質を通して、
モデルポテンシャルの正当性と適用限界を検討する。

<付録：MPI 並列計算による分子動力学法>

多相系、多結晶物質などの、大規模複雑系の MD 計算のための
並列計算システムとして、MPI 並列のソフトとハードを概観する。
プログラミングについても、実例を示し、解説する。

<参考文献>

「分子間力と表面力(第 2 版)」 J.N.イスラエルアチヴィリ、 朝倉書店
「固体論の基礎」G.バーンズ、東海大学出版会

<1．はじめに>

分子シミュレーション（分子動力学法(ＭＤ)とモンテカルロ法(ＭＣ)、さらに分子力学法(MM)と格子力学法(ＬＤ)）を実行するためには、原子間や分子間に作用する力・エネルギーを規定しておかなければならない。その規定の方法に関して、分子シミュレーション法を次のように分類できる。

分子動力学法(ＭＤ)とモンテカルロ法(ＭＣ)について,
原子・分子間相互作用の取り扱いに関する分類

相互作用の取り扱い	内容
経験的	構造，物性などの実験値を再現するように，分子シミュレーションを用いて試行錯誤でパラメータを決定する
準経験的	分子軌道法による構造ーエネルギー面を再現するようにパラメータを最適化する.
半経験的	経験的ないし半経験的な量子力学の方法を用いて,力とエネルギーをその場で計算する.
第一原理的	非経験的な量子力学の方法を用いて，波動方程式を解くことにより，力とエネルギーを直接計算する

無機溶融塩やセラミックスなどの共有結合性ないしイオン結合性固体や液体についての経験的ないし準経験的分子シミュレーションでは，通常個々の原子と原子の間に働く相互作用（原子間相互作用）を考える。有機分子の集合体では分子間相互作用のみを与える（剛体分子モデル）か、分子間と分子内を区別して考える場合が多い.

最近接原子間にはたいてい価電子のやり取りがあり，化学結合を形成している．これは本質的には多体的なものである．多体的相互作用を忠実に表すためには，量子力学を使わなければならない（-> 第一原理的ＭＤ）．しかしそれには大変な計算の手間がかかり，現在のところ有効に扱える物質系と系の大きさは非常に限られる.

したがって，経験的ないし準経験的ＭＤ法とＭＣ法では，しばしば２体中心力項のみ，ないし２体中心力項＋３体項で近似される相互作用モデルが用いられる．そのような，有効相互作用関数とモデルパラメータを用いて非常に精密なシミュレーションが、大規模な系に対して適用可能である.

<2．原子・分子間相互作用の形式>

[希ガス＝レーナード・ジョーンズ型]

２個の原子間にのみ相互作用(引力または斥力)が働き、多数の原子集団系の相互作用は２原子間の相互作用の足し合わせで表すことができる場合、その相互作用形式を２体中心力型であると言う。この**２体中心力**型原子間相互作用モデルの最も簡単な形式の一つはレナード・ジョーンズ(L-J)型である：

$$u_{ij}(r_{ij}) = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

希ガスにおいて，σとεは気相のビリアル係数と粘性から与えられる．

右辺第１項は経験的短距離斥力項であり，Pauli の排他律によるものであり、距離の-12乗である必然性はない。このような 12-6 型と呼ばれる形式は計算しやすさにより選ばれたものであると思われるが、今日では、その意味でこれでなければならない理由はない。ボルンとマイヤーにより提案された指数関数型の近接反発形式、u_{rep}＝b exp（－r_{ij}/ρ）を用いることも一般的である（「固体論の基礎」1.4)。

第２項は時間空間平均された**双極子－誘起双極子**引力項である．すなわち、近くにある双極子により誘起された双極子が、エネルギー的に安定な双極子間相対配置をとりやすいために、時間平均すると引力となるものである。これによるエネルギーや力の計算の観点では、距離の-6 乗という関数は、短距離力とは言えず、アルゴン液体で、1.5 nm のカットオフ距離を採用した場合には、全エネルギー（-5.4 kJ/mol 程度）で-0.016 kJ/mol、圧力で-0.2 MPa 程度の補正が必要である。

このような形式に静電相互作用項、すなわちイオンの点電荷間の相互作用を加えればNaCl などのイオン性化合物に適用できるものとなる．すなわち理想的なイオン結合性液体や固体の中の原子間相互作用は非結合反発相互作用、分子間力、および静電相互作用で記述できる。したがって、塩や酸化物と同じ形式の近接反発項の関数を採用することは、相互作用モデルの汎用性を確保する上で重要である。すなわち次のような形式である：

$$u_{ij}(r_{ij}) = b\exp(-r_{ij}/\rho) - c_{ij}/r_{ij}^{6}$$

一方，分子集団系については分子内結合の共有結合については，本来方向性が必要となるが，多くの場合分子内相互作用は考慮せず（剛体分子），上記の形の関数を分子間に適用することで，分子集団系の計算がしばしば行われる．

双極子－誘起双極子相互作用を、その実体に即して取り扱う手法として **Shell モデル**がある。すなわち原子を[核+内殻電子]＋[外殻電子]の２個の点電荷としてとらえ、反発相互

作用の中心は[外殻電子]に置く。これら２個の電荷の間はバネで連結され、平衡位置は間隔がゼロのところにあるとし（２個の電荷の間には静電相互作用は働かない)、調和振動子を適用する。粒子（原子）の運動としては、[核＋内殻電子]のみを扱い、[外殻電子]の位置は力学的つりあいの式から求める。このモデルにより、時間平均された引力ではなく、時間とともに変化する双極子を扱えるだけでなく、点電荷ー誘起双極子間の相互作用も明確に取り入れられることになる。

[分子間結合性－イオン結合性－共有結合性汎用相互作用モデル]

上で述べた様に酸化物ではしばしば BMH（Born=Mayer=Huggins 型：逆べきのかわりに指数関数を用いる）＋静電ポテンシャルを用いることが多い．一般の無機凝集化合物の相互作用を表すために，たとえば以下のような，２体中心力項 ＋ ３体項の形式 8)が考えられる．BMH-静電ポテンシャルはこの２体中心力項の第３項までに相当する．

２体中心力項：

$$u_{ij}(r_{ij}) = \frac{z_i z_j e^2}{4\pi\varepsilon_0 r_{ij}} \qquad \text{クーロン}$$

$$-\frac{c_i c_j}{r_{ij}^{\ 6}} \qquad \text{分子間}$$

$$+ f_0(b_i + b_j)\exp\left(\frac{a_i + a_j - r_{ij}}{b_i + b_j}\right) \qquad \text{近接反発}$$

$$+ D_{1ij}\exp(-\beta_{1ij} r_{ij}) + D_{2ij}\exp(-\beta_{2ij} r_{ij}) + D_{3ij}\exp\left[-\beta_{3ij}(r_{ij} - r_{3ij})^2\right] \qquad \text{共有(動径)}$$

３体項（共有結合の角度部分）：

$$u_{jij}(\theta_{jij}, r_{ij}) = -f_k\left\{\cos\left[2(\theta_{jij} - \theta_0)\right] - 1\right\}\sqrt{k_1 k_2}$$

$$k_1 = \frac{1}{\exp\left[g_r(r_{ij1} - r_m)\right] + 1}$$

（ε_0:真空の誘電率、e:電気素量、r_{ij}:粒子間距離、θ_{jik}:粒子間角度、f_0:単位調整）

ここで，ほとんどイオン結合で表現できるようなイオン性の程度の高い系、たとえば KCl では共有結合項は用いない。原子が中性と見なせる場合（希ガス）には静電項は用いない。分極しにくい原子・イオン、すなわち半径が小さく価数の大きな原子・イオンでは分子間力項を用いない（たとえば、ケイ酸塩中のシリコン原子)。これらの式の中のパラメータは分子軌道法によるエネルギー面への最適化や実験による物性値などに合わせるよ

うに経験的に決められる（後述）。

３体項は混成軌道の角度部分を表しているもので、特に対称性の低い分子，たとえば H_2O や NH_3 などにおいて必要（必須）である．CH_4 や CO_2 のような対称性の高い分子では３体項なしにこれらの分子を構成することも可能ではある．H_2O 系においては，分子内と水素結合のすべての組み合わせの H－O－H 角にこの３体相互作用を適用している．k_j は３体相互作用の到達範囲を規定するものであり、分子内結合のO－H間ではほぼ１、水素結合距離で 0.05 程度である。

H2O を例にとると、下のように、独立な 17 個のパラメータを決めなければならない。

	z	c	a	b
O	○	○	○	○
H	△	×	○	○

	D_1	β_1	D_2	β_2	D_3	β_3	r_3
O-O	×	×	×	×	×	×	×
O-H	○	○	○	○	○	○	○
H-H	×	×	×	×	×	×	×

	f_k	θ_0	g_r	r_m	
H-O-H	○	○	○	○	（水素結合も含む）

[多体相互作用モデル]

これまで示してきた相互作用モデルはシミュレーション計算の最初に与えられた関数がそのパラメータも含めて最後まで一定のものとして与えられる。これに対して、イオン電荷はイオンの周りの原子配置により電価移動量が変化するものであり、共有結合電子についても同様である。このような化学結合の時間的空間的変化に対応できるモデルとして、現在までにいくつかのものが提案されているが、その代表的なものとして、電荷平衡化法(ＱＥｑ)と結合次数を含む Tersoff モデルを紹介する。

<u>電荷平衡化法</u>は、系を構成している全原子を通じて原子の電子系の化学ポテンシャルは等しいとして、各原子の電子系の化学ポテンシャル(μ_i)を原子の電気陰性度(χ_{i0})と自己静電反発(η_{i0}、電子が同じ軌道に存在する自分自身から受ける静電反発)からなる簡単な様式で表現し、

$$\mu_i = \frac{\partial E}{\partial q_i} = \chi_{i0} + 2\eta_{i0} q_i + \sum_{j(\neq i)} \frac{q_j}{r_{ij}}$$

次の連立方程式を解くことにより、個々の原子の電荷(q_i)を求めるものである：

$$\mu_1 = \mu_2 = \cdots = \mu_N, \quad \sum_i q_i = 0$$

ＭＤあるいはＭＣの各ステップにおいてＮ次の連立方程式を解く必要があるため、原子数が大きくなると、分子シミュレーションにおける力の計算より多くの計算時間を要するが、

それでも第一原理的な計算よりはるかに少ない計算時間で、より現実的なシミュレーションが可能である。このようなモデルにより、例えば架橋酸素、非架橋酸素、およびO^{2-}イオンの間の化学的区別がより明確になされるものと期待できる。

<u>Tersoff モデル</u>では原子間相互作用ポテンシャルは次のように表される：

$$u_{ij}(\{r_i\}) = f_c(r_i)\left[A_{ij}\exp(-\lambda_{ij}r_{ij}) - b_{ij}(\{r_i\}, f_c(r_{ij}))B_{ij}\exp(-\mu_{ij}r_{ij})\right]$$

f_c は相互作用をスムーズに打ち切るための関数である。右辺第 1 項は近接反発相互作用、第２項は共有結合の引力である。$b_{ij}(\{r_{ij}\}, f_c(r_{ij}))$は結合次数に相当し、２体および３体のやや複雑な関数で表されている。このモデルは炭素やシリコンの結晶やクラスターに適用され、一定の成果が得られている。すなわち、ダイヤモンド構造では１電子対共有結合の結合長と結合強度になり、グラファイトでは 1.33 電子対の共有結合のそれらになる。このモデルをそのままSiO_2 などに適用するのは困難であるが、結合次数の考え方を採り入れることは重要であろう。（Tersoff(1988) Phys.Rev.B. 37,6991-7000）

<３．経験的パラメータの最適化>

上記の相互作用における規定しなければならないパラメータは SiO2 や H2O などの２種の元素からなる系でも８個程度（簡単な２体中心力項のみの場合）から１７個程度（３体項が１種類の場合）にもなる．純粋なイオン結合で取り扱う NaCl では７個である。分子軌道法などの量子力学計算から非経験的にパラメータを決定することがしばしば試みられる。適当な小規模の原子集団からなるクラスターを考え、その原子配置を変化させた各構造での分子軌道法計算から各構造の全エネルギーが得られ、それらを総合すると構造に対するエネルギー面が得られる。それに対し、上で述べたような原子間相互作用関数のパラメータを最適化する。このようなパラメータ決定が望ましいものではあるが，クラスターの有限性などのため、中・長距離力の評価が十分ではなく、現時点では経験的な決定法の方が有効であるといえる．

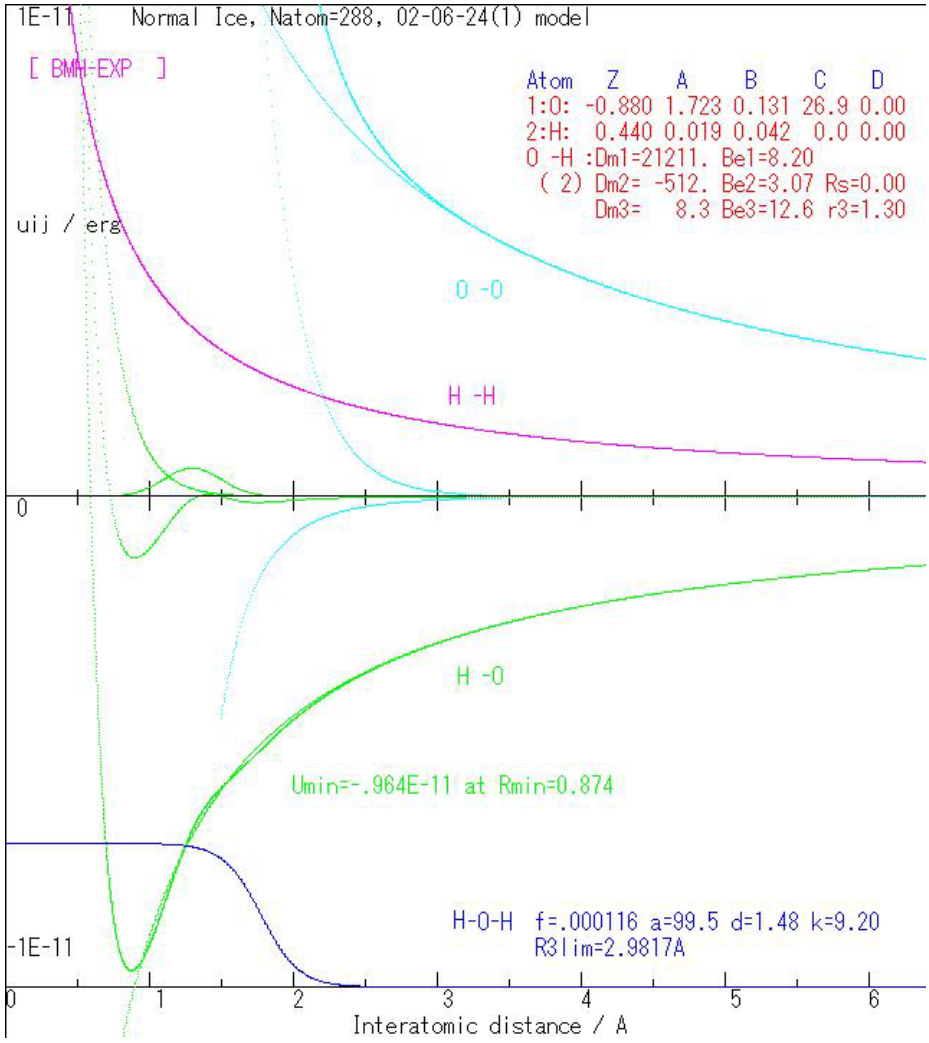
経験的なパラメータ決定は分子シミュレーション計算を用いて，試行錯誤的に，実験値を再現するようにパラメータを最適化するものである．実験値としては，

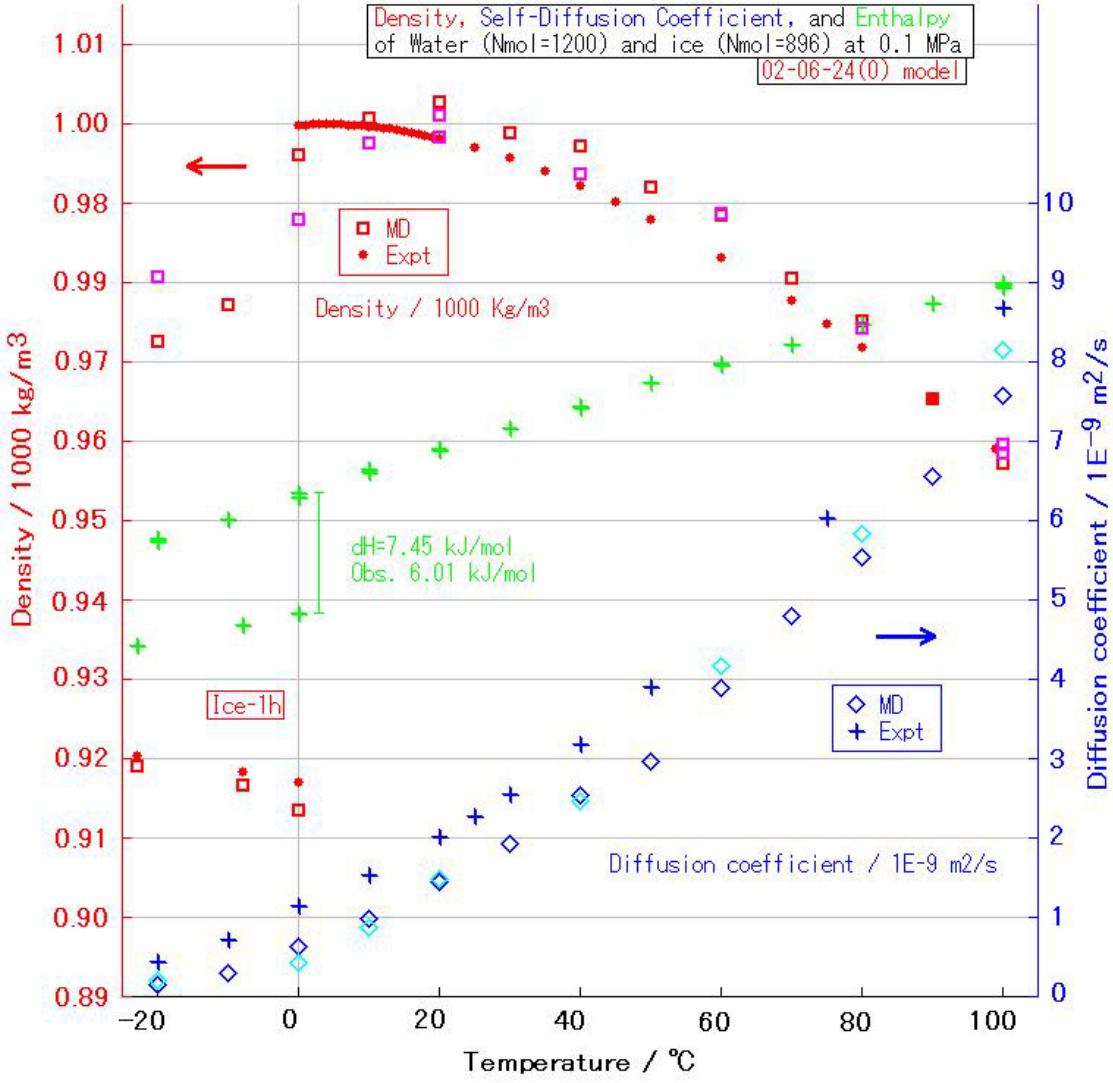
密度
結晶構造（格子定数、対称性、原子座標）
状態方程式（温度ー体積、圧力ー体積）
熱膨張率
圧縮率

弾性定数
分子振動スペクトル
拡散係数
粘性係数
誘電率
相転移（転移温度、転移圧力）
エンタルピー
その他

などがあり，相互作用を強く反映した物性を選ぶ．これらの温度・圧力の依存性も重要な観測量である。このようなモデルは、計算対象の化学組成と物理化学的条件はやや限定されるが、比較的単純な計算で有効な結果を得ることができ、特に空間的な大規模計算には今後も欠かせないであろう。

このようにして決められた、最新の H_2O のモデルとその再現性を示す。





H_2O の相互作用モデル、02-06-24(0)版による MD 計算と実験値との比較。

・水と氷の密度の温度変化（□●）

・水の拡散係数の温度変化（◇＋）

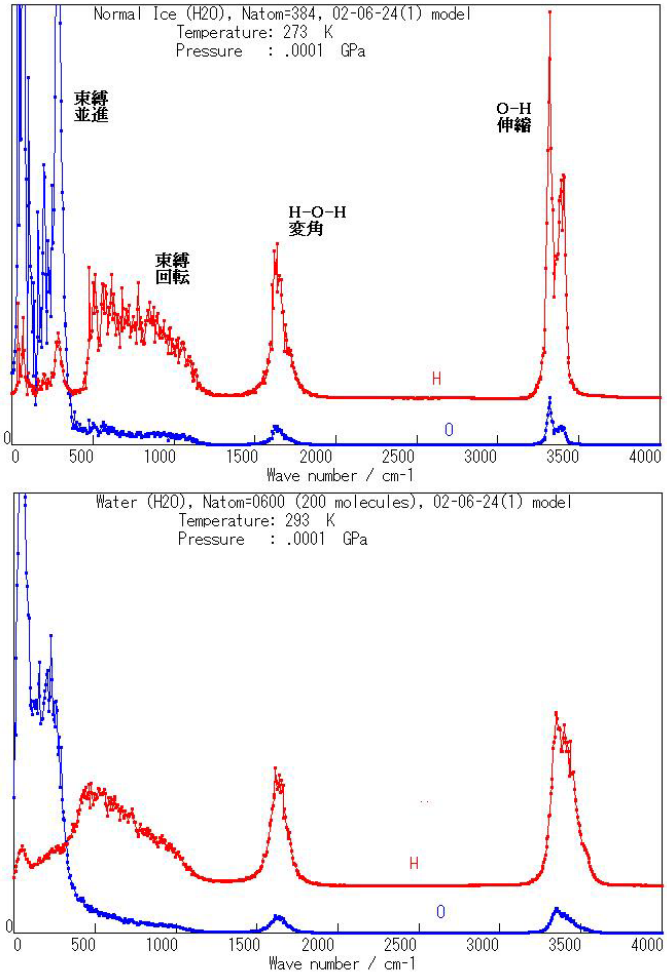
・水と氷のエンタルピーの温度変化（＋）と融解潜熱

水は 3600 原子系(1200 分子)、Ice-1h は 2688 原子系(896 分子)を用いた。

（一部、水 6000 原子(2000 分子)系もプロットしている）

このモデルは全ての原子の独立な運動の自由度を持っている（全自由度モデル）ので、伸縮および変角振動を含む全ての分子振動を計算できる（次ページ）。粘性係数や誘電率も定量的に再現できるものである。

00-06-24(1)モデルによる
氷（上）と水（下）の
全振動スペクトル。
0.4fs×20000 ステップ
の粒子速度データから
速度自己相関関数を計算し
それをフーリエ変換して
求めた。



<4．半経験的パラメータの最適化>

<u>クラスターの分子軌道法計算からの導出</u>

SiO_2 固体のような物質（分子性ではなく、化学結合がイオン性と共有性の中間）について、その構造より切りだしたクラスターを用いた分子軌道法計算により、パラメータを決めることがしばしば行われてきた（たとえば、Tsuneyuki（1996）Mol.Engineer. 6：1 57-182）。その場合には{H4SiO4}や[$e^{+}{}_{4}SiO_4$]、[$H_6Si_2O_7$]のような有限の大きさのクラスターについて、それに種々の変形を加えて、非経験的分子軌道法計算をおこない、その結果得られるエネルギー曲面のモデルポテンシャル関数のオパラメータをフィッテイングすることになる。

長距離の情報の不足からか、配位数の異なるクラスターを同時に精度良くすることが困難である。また、最適化後の微調整（？）がしばしば必要である。その後の発展はほとんどみられず、最近ではあまり用いられない。

２原子分子、O_2 や N_2 について、分子内と分子間の相互作用パラメータを、非経験的分子軌道法を用いて決定した例をしめす。分子ないは簡単な Morse 型ポテンシャルを用いた。分子間は近接反発項、分子間力項、および４極子相互作用を想定した。４極子相互作用は、分子を３点電荷で表すことによった（別の方法として、点の４重極子とし、４重極モーメントを与える方法がある。また５点電荷モデルもある）。

分子内相互作用

O_2、N_2 分子のそれぞれについて、孤立２原子系で、核間距離を変化させて、分子軌道法計算を行い、全エネルギーの曲線を得た。これに exp 項２個からなる関数をフィッティングした。パラメータは３ないし４個である。３パラメタ関数では表現力が不足している。採用したのは次の関数である。

$$u_{ij}(r_{ij}) = D_{1ij}\exp\left(-\beta_{1ij}r_{ij}\right) + D_{2ij}\exp\left(-\beta_{2ij}r_{ij}\right)$$

必然的に、理論計算の核間距離と伸縮振動を再現することが得られた。

分子間相互作用

２個の２原子分子を含む系（原子数は４個）について、分子間の相対配置を４通り（平行、ねじれ、垂直、直列）とし、それぞれについて分子間距離を変化させて分子軌道法計算を行った。そのエネルギー曲面（４曲線）に対して、原子－原子の分子間相互作用

$$u_{ij}(r_{ij}) = \frac{z_i z_j e^2}{4\pi\varepsilon_0 r_{ij}} - \frac{c_i c_j}{r_{ij}^{\,6}} + f_0\left(b_i + b_j\right)\exp\left(\frac{a_i + a_j - r_{ij}}{b_i + b_j}\right)$$

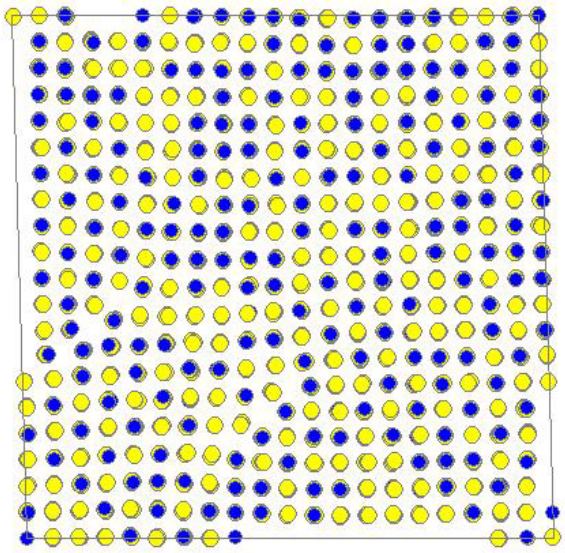
を適用し、パラメータの最適化を行った。全体的なフィッテイングの程度は必ずしもよくはない。これを用いて、O_2 と N_2 の液体と固体の分子動力学計算を行った結果、密度が系統的に小さいという結果を得た。すなわち、分子間力が過小評価されていることになる。c パラメータを調整することで、状態方程式の再現が改善された。

このモデルは O_2、N_2 のゼオライトへの吸着挙動を調べるために用いた。

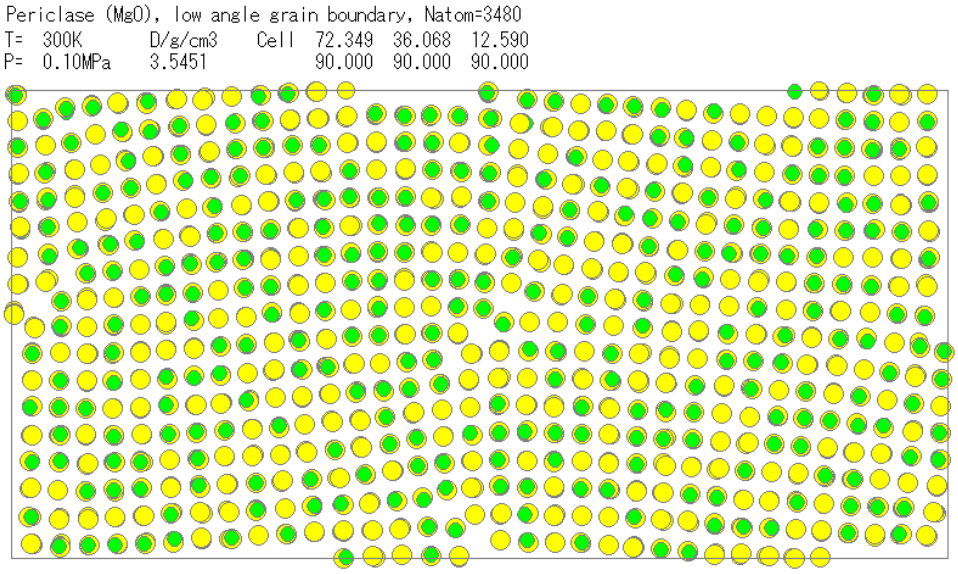
<５．モデルポテンシャルの評価・適用（無機-有機複合系のために)>

以上のようにして作成した相互作用モデルは、その作成に用いた単一相の物性等についてだけでなく、しばしば、その原子、原子間を含む種々の系で用いることができる必要がある（汎用性の確保)。すなわち、溶液系、多相系、多粒子系、その他の複雑大規模系などである。

転位を含む固体系
(CaO)

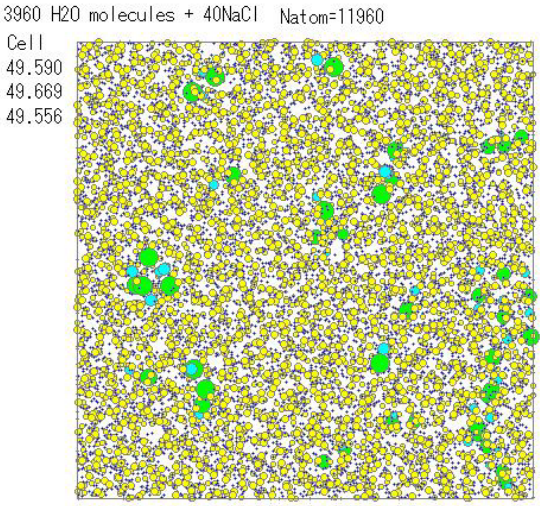


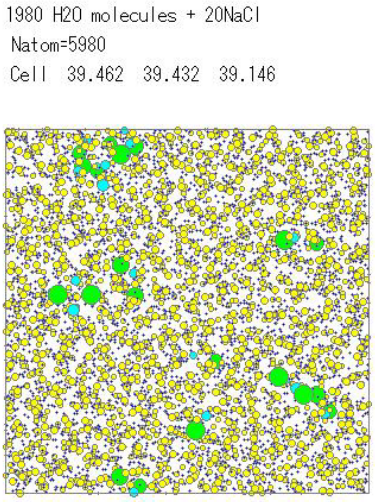
固体ー
固体
界面系
（MgO、
小傾角
界面）



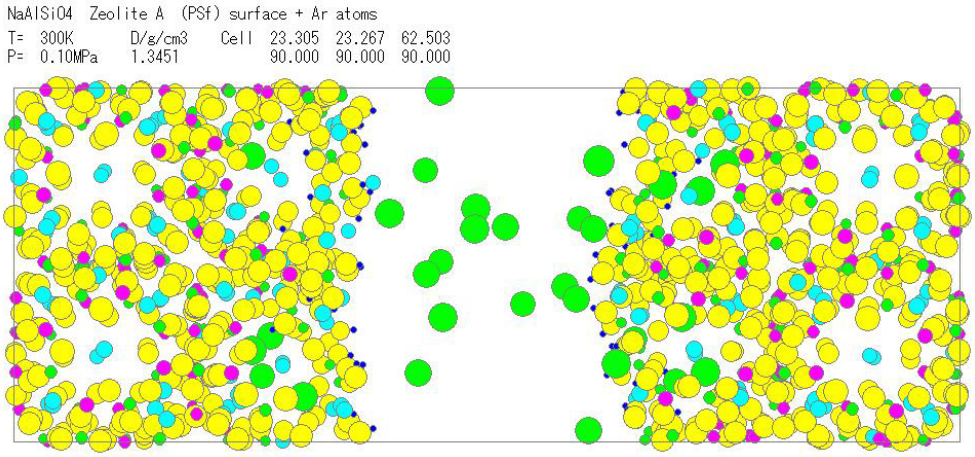
NaCl
水溶液系

（海水相
当濃度）

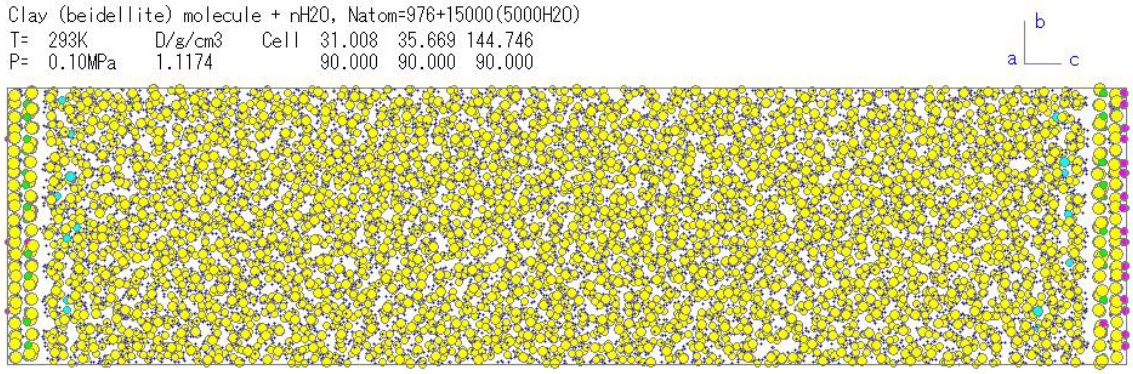




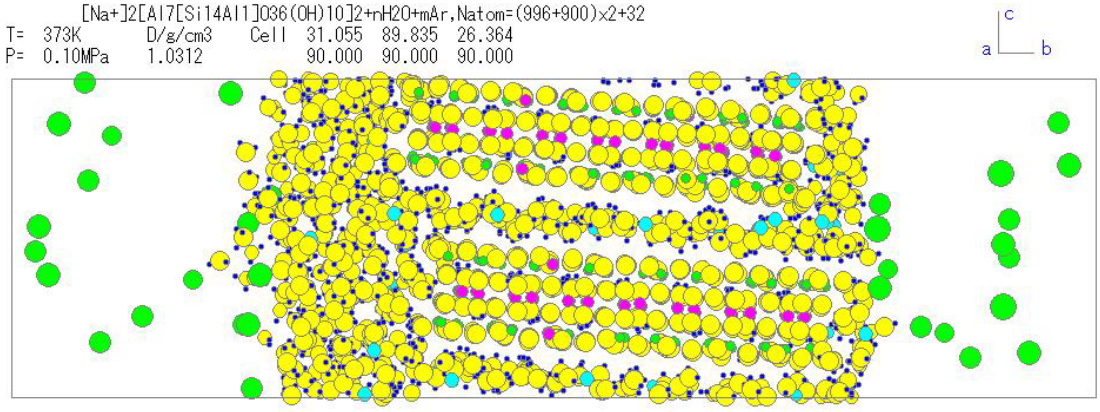
ゼオライト－雰囲気気体系



粘土分子－水・水溶液系



粘土粒子－付着水－気体系



これらの計算のためには、それぞれの単一相の相互作用モデルだけでなく、相互の原子間の相互作用モデルが必要となる。このような様々な複雑な系に適用可能で、定量的な予測計算の可能な原子間相互作用モデルの構築が望まれている。

<付録：MPI 並列計算による分子動力学法>

MPI 並列のプログラミングのサンプルとして、三角関数の積分を、区間を分割して、各プロセスが分担して計算を行うプログラムを後に示す。MD 計算においても、ここで用いたサブルーチン等のみで十分に効率的なプログラムを作成できる。

MPI は、ここでは MPI/Pro（MSTI 社、 8 プロセスまで無料ダウンロード可）を用いているが、プログラムは MPI-1 で記述されており、MPICH（MP-MPICH、NT-MPICH、WMPI）などの他の MPI でも同じである。

MD プログラムでの実際のプログラミングについて解説・議論する。また計算・通信効率、ハードウエアとソフトウエア（Windows のみ）の構築法についても言及する。

```
      PROGRAM  MPItest
C=========================================================I
C##                                                      ##
C##          Program  :  MPItest.for                      ##
C##                                                      ##
C##     2002-MAR-06  by  Katsuyuki Kawamura (TITECH)     ##
C##                                                      ##
C=========================================================I
C
      include  'mpif.h'                                          ! include file
      integer  status(MPI_STATUS_SIZE)                           !
c
      common /datai/ pi,range,delta,resulti,result0
           real *8  pi,range,delta,resulti,result0
C
C     ------------------------------------------------------------ MPI
      call  MPI_init  (ierr)                                     ! 初期化
      call  MPI_Comm_Rank  (MPI_COMM_WORLD, myrank, ierr)        ! 自分のランクの取得
      call  MPI_Comm_Size  (MPI_COMM_WORLD, mpsize, ierr)        ! 全体のプロセス数
c
      if (myrank.ne.0)  call proci (myrank, mpsize)              ! myrank が 0 でなければ
c                                                                ! 子サブルーチンへ行く
      CALL  KCLOCK  (IYEAR,IMONTH,IDAY,IHOUR,IMINUT,ISECND,I100TH)
      WRITE  (*,4001)  IHOUR,IMINUT,ISECND, IYEAR,IMONTH,IDAY
 4001 FORMAT (' Started  at ',I2,':',I2,':',I2,' on ',I2,'/',I2,'/',I2)
c
      write (6,*) 'Number of CPU''s =',mpsize
C
c     Integration of sin(x), x=0 to 2.pi
      pi    = 3.141592654D0
      range = 2*pi / mpsize
      delta = 0.00000001D0
c
      call MPI_Bcast(pi,  1,MPI_DOUBLE_PRECISION,0,MPI_COMM_WORLD,ierr)    ! πの値を知らせる
      call MPI_Bcast(range,1,MPI_DOUBLE_PRECISION,0,MPI_COMM_WORLD,ierr)    ! 積分範囲の通知
      call MPI_Bcast(delta,1,MPI_DOUBLE_PRECISION,0,MPI_COMM_WORLD,ierr)    ! 積分刻みの通知
c
      write (*,8801)
 8801 format ('   MPI_Bcast sent data to processes')
```

```
c
      call calc  (myrank, mpsize)
      result0 = resulti
      write (*,8901)  icp,result0
c
c     ------------------------------------- receive results from CPU's
      if (mpsize.gt.1)  then
      do  190  icp = 1, mpsize-1
         call MPI_Ssend (icp,1,MPI_INTEGER,icp,icp, MPI_COMM_WORLD,ierr)     ! 結果の要求
         call MPI_Recv  (resulti,1, MPI_DOUBLE_PRECISION, icp, icp,          ! 結果を受け取る
     *                                     MPI_COMM_WORLD,status,ierr)       !
         write (*,8901)  icp,resulti
 8901    format ('   MPI_Recv received results from ',i2,3x, f16.12)
         result0 = result0 + resulti
  190 continue
      write (6,*) 'Result=',result0
      end if
c
      call  MPI_Bcast (-9999, 1, MPI_INTEGER, 0, MPI_COMM_WORLD, ierr)      ! 計算終了の通知
      call  MPI_Finalize  (ierr)                                           ! 親プロセスの終処理
C
      CALL  KCLOCK  (IYEAR,IMONTH,IDAY,IHOUR,IMINUT,ISECND,I100TH)
      WRITE  (*,4002)  IHOUR,IMINUT,ISECND, IYEAR,IMONTH,IDAY
 4002 FORMAT (' Finished at ',I2,':',I2,':',I2,' on ',I2,'/',I2,'/',I2)
      STOP
      END
C
C
C                                                              =======
C============================================================ proci
      SUBROUTINE  proci  (myrank, mpsize)                                  ! 子プロセスのための
C                                                                          ! サブルーチン
      include  'mpif.h'
      integer  status(MPI_STATUS_SIZE)
C
      common /datai/ pi,range,delta,resulti,result0
           real *8  pi,range,delta,resulti,result0
c
      call MPI_Bcast(pi,   1,MPI_DOUBLE_PRECISION,0,MPI_COMM_WORLD,ierr)    ! π値を受け取る
      call MPI_Bcast(range,1,MPI_DOUBLE_PRECISION,0,MPI_COMM_WORLD,ierr)   ! 積分範囲を受け取る
      call MPI_Bcast(delta,1,MPI_DOUBLE_PRECISION,0,MPI_COMM_WORLD,ierr)   ! 積分刻みを受け取る
c      write (*,*)  myrank,pi,range,delta
C
      call calc  (myrank, mpsize)
C
      call MPI_Recv  (iii,  1, MPI_INTEGER, 0, myrank, MPI_COMM_WORLD,      ! 結果要求の受理
     *                                                  status,ierr)       !
      call MPI_Ssend (resulti,1, MPI_DOUBLE_PRECISION, 0, myrank,           ! 結果の通知
     *                                         MPI_COMM_WORLD,ierr)        !
c
      call  MPI_Finalize  (ierr)                                           ! 子プロセスの終処理
      STOP
      END
C
```

```
C
C                                                                 =======
C================================================================ calc
      subroutine  calc  (myrank, mpsize)                                  ! 積分計算の
C                                                                         ! サブルーチン
      common /datai/ pi,range,delta,resulti,result0
            real *8  pi,range,delta,resulti,result0
c
      real *8  x1,x2,x,y
c
      resulti = 0.0D0
                x1 = range * myrank
                x2 = range * (myrank +1) - delta
                write (6,*) 'range',myrank,x1,x2
      do 10  x = x1, x2, delta
         y=(sin(x)+sin(x+delta))*delta/2.0D0
         resulti = resulti + y
   10 continue
c      write (6,*)  myrank,resulti
      return
      end
C
C
C                                                                ========
C================================================================ KCLOCK
      SUBROUTINE  KCLOCK  (IYEAR,IMONTH,IDAY,IHOUR,IMINUT,ISECND,I100TH)   ! 時刻を取得する
      PARAMETER  (LNI=32109, LTB=2504, LEL=8, LEM=10,     LCT=2000000,     ! サブルーチン
     *                    LSR=1254, LEE=LEL*(LEL+1)/2, L50=LCT/50+1,
     *           LAA= 172,  LNV=9876, LEF=LEM*(LEM+1)/2, LST=32,
     *           LAT=LAA*4, LVA=24+LEM*2,               LRG=LNI*5   )
C
      COMMON /CHARAC/ TITLE(15),RUNOPT(53),ATOM(LEM),ATMNET(2),
     *               ATMXTL(LAA),FLNAME(19)
        CHARACTER *4  TITLE,ATOM,ATMNET,ATMXTL
        CHARACTER *10 RUNOPT
        CHARACTER *25 FLNAME
C
      integer*4  IYEAR,IMONTH,IDAY, IHOUR,IMINUT,ISECND,I100TH
      character  Adtval(3)*12
      integer    Idtval(8)
c
      Call DATE_AND_TIME(Adtval(1),Adtval(2),Adtval(3),Idtval)
             IYEAR  = mod(Idtval(1),100)
             IMONTH = Idtval(2)
             IDAY   = Idtval(3)
             IHOUR  = Idtval(5)
             IMINUT = Idtval(6)
             ISECND = Idtval(7)
             I100TH = Idtval(8)
      RETURN
      END
```