

JAERI-M
92-080

分子動力学シミュレーションコードISISの開発

1992年6月

神林 奨

JAERI-Mレポートは、日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合わせは、日本原子力研究所技術情報部情報資料課（〒319-11 茨城県那珂郡東海村）あて、
お申し込みください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11 茨城県那珂郡
東海村日本原子力研究所内）で複写による実費領布をおこなっております。

JAERI-M reports are issued irregularly.
Inquiries about availability of the reports should be addressed to Information Division Department
of Technical Information, Japan Atomic Energy Research Institute, Tokaimura, Naka-gun, Ibaraki-
ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1992

編集兼発行 日本原子力研究所
印刷 ニッセイエプロ株式会社

分子動力学シミュレーションコードISISの開発

日本原子力研究所東海研究所情報システムセンター

神林 奨

(1992年5月12日受理)

分子動力学法を用いたコンピュータシミュレーションは、物理、化学、生物学など広範な分野において、実験測定や理論計算と相補的な研究手法として定着しつつある。分子動力学法は、ニュートンの運動方程式を数値的に解くことにより、多数の古典粒子からなる体系の運動の様子を数値的に再現する手法である。1980年代に入り、分子動力学法の技術的改良がなされ、実験測定との比較に便利な温度や圧力が一定という条件でのシミュレーションが可能となった。

本報告では、まず分子動力学法の概要をまとめ、次にソフト球モデルを対象として新規に開発した分子動力学シミュレーションコードISIS (isokinetic simulation of soft-spheres) について述べる。ISISコードは、一般的なマイクロカノニカルアンサンブルのシミュレーションとともに束縛法を用いた温度一定のシミュレーションを可能としている。また、運動方程式の逐次積分に第5次の差分式を用いて数値誤差を小さく抑え、粒子間力の計算時間を短縮するために簿記法を用いている。ISISコードのベクトル計算機に対する適合性は極めてよく、VP2600/10上で約16~24倍の速度向上率を得た。

A Molecular Dynamics Simulation Code ISIS

Shaw KAMBAYASHI

Computing and Information Systems Center

Tokai Research Establishment

Japan Atomic Energy Research Institute

Tokai-mura, Naka-gun, Ibaraki-ken

(Received May 12, 1992)

Computer simulation based on the molecular dynamics (MD) method has become an important tool complementary to experiments and theoretical calculations in a wide range of scientific fields such as physics, chemistry, biology, and so on. In the MD method, the Newtonian equations-of-motion of classical particles are integrated numerically to reproduce a phase-space trajectory of the system. In the 1980's, several new techniques have been developed for simulation at constant-temperature and/or constant-pressure in convenient to compare result of computer simulation with experimental results.

We first summarize the MD method for both microcanonical and canonical simulations. Then, we present an overview of a newly developed ISIS (Isokinetic Simulation of Soft-spheres) code and its performance on various computers including vector processors. The ISIS code has a capability to make a MD simulation under constant-temperature condition by using the isokinetic constraint method. The equations-of-motion is integrated by a very accurate fifth-order finite differential algorithm. The bookkeeping method is also utilized to reduce the computational time. Furthermore, the ISIS code is well adopted for vector processing: Speedup ratio ranged from 16 to 24 times is obtained on a VP2600/10 vector processor.

Keywords: Molecular Dynamics Method, Computer Simulation, Vector Processing, Bookkeeping Method, Isokinetic Constraint Method

目 次

1. はじめに	1
2. 分子動力学法	3
2.1 運動方程式と粒子間相互作用	3
2.2 境界条件のおき方	4
2.3 初期条件の設定	6
2.4 数値アルゴリズム	7
2.5 物理量の計算	8
3. 等温分子動力学法	13
3.1 乱数を用いる方法	13
3.2 束縛法	14
3.3 拡張系の方法	15
4. 分子動力学シミュレーションコードISIS	17
4.1 ソフト球モデルと無次元化運動方程式	17
4.2 数値アルゴリズム	20
4.3 簿記法	23
4.4 計算できる物理量	24
4.5 プログラムの構成	25
5. 計算時間の測定と評価	33
5.1 カーネルプログラムの場合	33
5.2 ISISコード全体の場合	36
6. まとめ	43
謝 辞	44
参考文献	44

Contents

1. Introduction	1
2. Molecular Dynamics Method	3
2.1 Equation of Motion and Molecular Interaction	3
2.2 Boundary Condition	4
2.3 Initial Condition	6
2.4 Numerical Algorithm	7
2.5 Calculation of Physical Quantities	8
3. Constant Temperature Methods for Molecular Dynamics Simulation	13
3.1 Stochastic Method	13
3.2 Constraint Method	14
3.3 Extended System Method	15
4. ISIS Code for Molecular Dynamics Simulation	17
4.1 Soft-sphere Model and Dimensionless Equation of Motion	17
4.2 Numerical Algorithm	20
4.3 Bookkeeping Method	23
4.4 Available Physical Quantities	24
4.5 Configuration of Computer Programmes	25
5. Measurement and Evaluation of Computational Time	33
5.1 Case Study of Kernel Programme	33
5.2 Case Study of Complete ISIS Code	36
6. Summary	43
Acknowledgements	44
Reference	44

1. はじめに

物質は、温度・圧力といった熱力学パラメータの変化によって、気体、液体、固体の3相の間を移り変わる。この3相における物質の性質を制御しているものは、外的条件によって決まる熱力学パラメータ(温度、圧力あるいは密度)を別にすれば、物質を構成している粒子(原子、分子、イオンなど)の間に働く相互作用である。したがって、全ポテンシャルエネルギーを $U_N(\mathbf{r}_1, \dots, \mathbf{r}_N)$ と仮定した N 個のモデル粒子からなる体系を考え、そのハミルトニアンを

$$H_N(\mathbf{r}_1, \dots, \mathbf{r}_N, \mathbf{p}_1, \dots, \mathbf{p}_N) = \frac{1}{2m} \sum_{i=1}^N p_i^2 + U_N(\mathbf{r}_1, \dots, \mathbf{r}_N), \quad (1.1)$$

と書けば、このモデル物質の統計力学的性質は、位相空間の平衡確率密度

$$f_0^{(N)}(\mathbf{r}_1, \dots, \mathbf{r}_N, \mathbf{p}_1, \dots, \mathbf{p}_N) = \frac{\exp[-\beta H_N]}{h^{3N} N! Q_N(V, T)}, \quad (1.2)$$

$$Q_N(V, T) = \frac{1}{h^{3N} N!} \int \cdots \int \exp[-\beta H_N] d\mathbf{r}_1 \cdots d\mathbf{r}_N d\mathbf{p}_1 \cdots d\mathbf{p}_N, \quad (1.3)$$

を使って完全に決定できるはずである。ここで、 Q_N は分配関数であり、 $\beta = (k_B T)^{-1}$ である。 Q_N の積分のなかで、 p_i に関する部分はガウス積分なので直ちに求めることができる。しかしながら、 r_i に関する部分は、 $U_N(\mathbf{r}_1, \dots, \mathbf{r}_N)$ が特別の場合(例えば、理想気体、すなわち $U_N(\mathbf{r}_1, \dots, \mathbf{r}_N) = 0$)を除いて、近似的にしか求めることができない。

式(1.2)や(1.3)を直接計算するかわりに、何らかの方法で確率密度 $f_0^{(N)}$ に従った構成粒子の配置 $\mathbf{r}_1, \dots, \mathbf{r}_N$ を再現することができれば、モデル物質の性質を求めることができる。計算機の発達とともに、分子動力学(molecular dynamics: MD)法とモンテカルロ(Monte Carlo: MC)法の2つ手法によって、モデル粒子系の運動の様子を厳密に(数値的に)再現することが可能となった。MD法では、与えられた粒子間相互作用を持つ有限個の古典粒子系について、ニュートンの運動方程式を数値的に解き、構成粒子の運動のシミュレーションを行う。他方、MC法では、確率 $f_0^{(0)}$ にしたがう粒子配置を乱数を使って生成する。いずれの方法でも、シミュレーションによって得られた位相空間上の軌跡から、モデル物質の熱物性や静的構造を正確に求めることができる。この2つの手法の中で、MD法は平衡状態の物理量ばかりでなく、粒子拡散など粒子の運動の時間依存性に関する量も求められることが大きな特徴である。この点を重視して、本報告ではMD法を使ったコンピュータシミュレーションに焦点をあてる。

MDシミュレーションでは、(1)取り扱うことのできる粒子数はアボガドロ数($\approx 6 \times 10^{23}$ 個)と比べそれほど多くなく、現在の計算機の速度・容量では、たかだか $10^3 \sim 10^4$ 個程度である、(2)シミュレーションできる時間は、物理的時間に換算すると、 $10^{-12} \sim 10^{-11}$ 秒程度である、といった制約がある。しかしながら、これまでのシミュレーションの結果によると、数十個ないし数百個の粒子数でも、多くの物理量は事実上無限大の体系とそれほど違わないことが明らかになっている。MDシミュレーション手法の利点は、次の4点に要約できる。

- (1) 与えられた粒子間相互作用を持つモデル物質の性質が正確にわかる。したがって、シミュレーション結果と実験値とを比較することから、現実物質の相互作用の形に関する知見を得ることができる。
- (2) 従来の近似理論では取り扱えないような体系のミクロな性質に関する詳細な情報が得られる。したがって、近似理論の検討あるいはその改良の手助けとなり得る。
- (3) 欠陥、不純物などの余分な相互作用を除去した理想的な物質の性質を知り得る。
- (4) 現実実現できないような、高温・高圧状態などの極限状態の物性を知り得る。

MD シミュレーション手法を用いた物性研究の先駆けは、剛体球分子からなるモデル物質を使った固相と液相間の相転移に関する研究である [1]。このシミュレーションは、斥力のみで相互作用する簡単なモデル物質でも高密度では結晶化することを示した点で、画期的なものであった。一方、現実の物質に対するシミュレーションとして、レナード・ジョーンズ (Lennard-Jones) ポテンシャルを使った液体アルゴンの研究 [2, 3]、そして長距離振動型 (long-range oscillatory: LRO) ポテンシャルを使ったナトリウム (Na) の研究 [4] をあげることができる。

上に述べた MD シミュレーションでは、外場のないニュートンの運動方程式を用いている。外場のかかっていない体系では、エネルギーの保存則によって体系の全エネルギーが一定に保たれる (ミクロカノニカルアンサンブル)。これは、平衡状態において、体系の温度とポテンシャルエネルギーがある平均値のまわりを揺らいでおり、温度ないし圧力はシミュレーションの終了後に平均値として初めて与えられることを示している。しかしながら、通常の実験測定は温度一定ないし圧力一定という条件のもとで行われるため、このような体系のシミュレーションは実験測定との比較には不便である。

1980 年代に入り、MD 手法の技術的改良がなされ [5]、密度と温度が一定 (カノニカルアンサンブル)、密度と圧力が一定、圧力と温度が一定といった条件を満たす MD シミュレーションが可能となった。これらの改良された MD シミュレーション手法では、(1) ニュートンの運動方程式に束縛条件を付加する [6, 7]、(2) 仮想的な熱浴 (温度一定) あるいはピストン (圧力一定) を体系に付加する [8, 9]、(3) 揺動散逸定理に従ってランダムな力を体系に与える [10, 11]、という方法でニュートンの運動方程式が拡張されている。このような新しい MD シミュレーション手法は、平衡状態の物性研究に限らず、対象としている体系に外界から急激な変化を与えた際に起こる過渡現象の研究に応用されている。

以下、2章ではミクロカノニカルアンサンブルの MD シミュレーションを例にとって MD 法の概要をまとめ、3章で密度と温度一定の MD 法を説明する。4章では、新規に開発した MD シミュレーションコード ISIS について述べる。ISIS コードは、ミクロカノニカルアンサンブルあるいはカノニカルアンサンブルのどちらの条件でも MD シミュレーションを行うことが可能である。さらに、運動方程式の数値積分に高精度のアルゴリズムを使用しており、非常に多くの積分ステップを安定に実現することができる。また 5章に示すように、ISIS コードはベクトル計算機を利用することにより、飛躍的に計算時間を短縮することができる。VP2600/10 ベクトル計算機では、スカラー計算に比べて約 16 ~ 24 倍の速度向上を得ている。これにより、多数 ($\sim 10^4$) 個の粒子系のシミュレーションが可能となった。

2. 分子動力学法

物質は多数の粒子（原子，分子，イオンなど）から構成されている。このような多粒子系の性質を左右するのは，(a) 構成粒子の特徴（粒子の質量と粒子間に働く相互作用），(b) 外的条件によって定まる熱力学パラメータ（温度，圧力あるいは密度），(c) 構成粒子の運動の様子，という3つの要素である。

分子動力学（molecular dynamics: MD）法は，物質の構成粒子に対して適切なモデルを仮定し，指定した熱力学的状态におけるモデル体系の運動状態を，ニュートンの運動方程式から数値的に求める方法である。モデル物質の様々な性質は，時々刻々得られる“観測量”（粒子の空間配置や速度の関数）の平均操作によって計算することができる。MD法を使った粒子シミュレーション（MDシミュレーション）は，入力パラメータの変更によって様々な物質に対応でき，計算対象の粒子数が有限であることを除けば数値的に厳密な結果を与えるので，物性を研究する道具として極めて汎用的かつ有効である。

本章では，ミクロカノニカルアンサンブルのMDシミュレーションを例にとり，(1) 運動方程式と粒子間相互作用，(2) 境界条件のおき方，(3) 初期条件の設定，(4) 数値アルゴリズム，(5) 物理量の計算，の5点について説明する。MD法の詳細については，文献[12, 13, 14, 15]を参照されたい。

2.1 運動方程式と粒子間相互作用

説明のための簡単な体系として，質量 m で内部自由度を持たない同種粒子 N 個が1辺の長さ L の立方体の容器に入っている古典系を考える。この体系を構成する粒子の位置座標 $r_i(t)$ および速度 $v_i(t)$ の時間発展は，ニュートンの運動方程式

$$m \frac{dv_i(t)}{dt} = F_i(t), \quad v_i(t) = \frac{dr_i(t)}{dt}, \quad (2.1)$$

によって定められる。ここで， $F_i(t)$ は時刻 t に i 番目の粒子に働く力である。

各粒子に外力が働かない場合，体系の全エネルギーは保存する。このような体系の位相空間上の軌跡は，統計力学におけるミクロカノニカルアンサンブルに相当する。ミクロカノニカルアンサンブルの熱力学変数は，粒子数 N ，体積 $V = L^3$ ，そして全エネルギー E の3つであり，これは熱力学極限（数密度 N/V が一定値 ρ の条件で， $N \rightarrow \infty$ ， $V \rightarrow \infty$ とする操作）における2つの変数，すなわち密度 ρ と E に対応する。

さて，運動方程式(2.1)を解くには，力 F_i に対するモデル，すなわちモデルポテンシャルを与える必要がある。多くのMDシミュレーションでは，「系の全ポテンシャルエネルギーが任意の2個の粒子の間に働く2体ポテンシャルの和で与えられる」とする2体相互作用の近似を用いる。さらに単原子分子の体系では，2体ポテンシャルを，球対称の関数 $u(r)$ と与

えることが多い。この場合、 i 番目の粒子に働く力は

$$F_i = - \sum_{j \neq i} \frac{\partial u(r_{ij})}{\partial r_i}, \quad (2.2)$$

で与えられる。ここで、 $r_{ij} = |r_i - r_j|$ は i 番目の粒子と j 番目の粒子との距離 $|r_i - r_j|$ である。

MD シミュレーションに用いるモデルポテンシャルは、原理的に電子論に基づいて求めることが可能である。しかしながら、高速計算機の発達した現在でも、第 1 原理から定量的に信頼し得るポテンシャルを求めることは、小数の簡単な場合を除いて、一般に容易ではない。そこで半経験的なモデルポテンシャルとして、(1) パラメータを含む関数型を仮定する、(2) 計算を単純化 (近似) した第 1 原理を用いて定める、といった方法が用いられる。MD シミュレーションによって詳しく調べられているモデルポテンシャルとして次のものがある。

i. Lennard-Jones ポテンシャル

$$u(r) = \epsilon \left[\left(\frac{\sigma}{r} \right)^n - \left(\frac{\sigma}{r} \right)^m \right], \quad (\epsilon > 0). \quad (2.3)$$

ϵ, σ, n, m はパラメータである。

ii. 剛体球ポテンシャル

$$u(r) = \begin{cases} +\infty & (r < \sigma) \\ 0 & (r \geq \sigma) \end{cases}. \quad (2.4)$$

σ は剛体球の直径を表す。

iii. 逆べきポテンシャル

$$u(r) = \epsilon \left(\frac{\sigma}{r} \right)^n, \quad (\epsilon > 0). \quad (2.5)$$

$n, \epsilon^{1/n} \sigma$ はパラメータである。 $n \rightarrow \infty$ とすると剛体球ポテンシャルになる。

式 (2.4) および (2.5) に示されたモデルポテンシャルは、引力を含まず気体から液体への凝縮現象も与えない現実離れしたものである。しかしながら、これらは、有限の大きさを持つ粒子間の相互作用の最も基本的な特徴である排除体積効果を取り込んでおり、体系を容器に入れて密度を一定にした状態において、液体など密度の大きい状態の時空構造の概略を代表するモデルとして有効であることがわかってきた [16]。モデルポテンシャルの妥当性は、現実の物質に対する (1) X 線や中性子散乱、(2) 融解現象、(3) 結晶の物性、などの実験値との比較によって行う。

2.2 境界条件のおき方

MD シミュレーションで取り扱うことのできる粒子数は、現在の大型計算機を使ってもたかだか $10^3 \sim 10^4$ のオーダーであり、アボガドロ数 ($\approx 6 \times 10^{23}$) と比較して極めて少ない。このような少数個の粒子からなる体系のシミュレーション結果を、できるだけ巨視的な体系の振る舞いに近づけるために、周期境界条件が用いられる。

周期境界条件とは、 N 個の粒子を入れた容器（これを基本セルと呼ぶ）の周囲に、粒子の配置が基本セルのそれと全く同じ虚像（イメージセル）を積み重ね、基本セル内の粒子の運動方程式を解く際にイメージセルに含まれる粒子から受ける力も考慮する（Fig. 2.1 を参照、図では簡単のため 2次元系を考えている）。このとき、容器の壁は全くの素通しにして、基本セルの粒子が壁を通り抜けイメージセルに進入するとき、それと入れ替わりにイメージセルの粒子が反対の面から基本セルの中に入るものとする。したがってこの条件においては、基本セルに含まれる粒子の数は不変である。このような境界条件によって、粒子系を閉じこめた箱の表面からの影響を取り除こうというのである。これまでのシミュレーションの結果によると、得られる物理量は高々数 100 個、場合によっては数 10 個の粒子数でも事実上無限大の系の値とそれほど変わらないことがわかっている。

周期境界条件を課した場合、基本セルに含まれる N 個の粒子全体のポテンシャルエネルギー U は、

$$U = \frac{1}{2} \sum_{\nu} \sum_{ij} u(|L\nu + r_{ij}|), \quad (2.6)$$

となる。ここで、 $\nu = (\nu_x, \nu_y, \nu_z)$ はイメージセルの方向を表すベクトルであり、 ν_x, ν_y, ν_z は整数である。このとき、 $(\nu_x, \nu_y, \nu_z) = (0, 0, 0)$ は基本セルを表す。また、 r_i, r_j はすべて基本セルの内に位置する。式 (2.6) を用いると、 i 番目の粒子に働く力は、

$$F_i = -\frac{\partial}{\partial r_i} \left[\sum_{\nu} \sum_{j \neq i} u(|L\nu + r_{ij}|) \right], \quad (2.7)$$

と書ける。

実際の計算では、式 (2.7) の ν についての和をどこまでもとるわけにはいかないで、2 体ポテンシャル $u(r)$ がある距離 r_c (カットオフ半径と呼ぶ) で消滅するとおく近似が用いられる。 r_c を選ぶ際、(1) r_c が小さすぎると実際の $u(r)$ とは違ったポテンシャルを使うことになる、(2) r_c が大きすぎるとシミュレーションに要する計算時間が増大する、という 2 つの条件を考慮する。

$u(r)$ が短距離の相互作用のとき、 $r_c \leq L/2$ と選ぶ方法を、minimum image distance 法という。このとき、式 (2.7) において、粒子対 (i, j) に対して $|L\nu + r_{ij}| < r_c$ を満足する ν が 1 つ以下になることが容易にわかる。したがって、式 (2.7) は、

$$F_i = -\frac{\partial}{\partial r_i} \left[\sum_{j \neq i} u(r_{ij}) \right], \quad \text{ただし } r_{ij} = \min_{\nu} |L\nu + r_i - r_j|, \quad (2.8)$$

という表現になる。式 (2.8) に現れる r_{ij} の計算アルゴリズムは、

$$r_{ij} = \sqrt{x_{ij}^2 + y_{ij}^2 + z_{ij}^2}, \quad (2.9)$$

$$x_{ij} = \begin{cases} x_i - x_j & (|x_i - x_j| \leq L/2) \\ x_i - x_j + L/2 & (x_i - x_j < -L/2) \\ x_i - x_j - L/2 & (x_i - x_j > L/2) \end{cases}, \quad (y_{ij}, z_{ij} \text{ も同様}) \quad (2.10)$$

となり、式 (2.7) の計算と比べはるかに簡単である。

2.3 初期条件の設定

ニュートンの運動方程式 (2.1) を数値的に解く (逐次積分する) 際、適切な初期条件 (各粒子の初期配置 $r_i(0)$ および初期速度 $v_i(0)$) が必要である。

多くの MD シミュレーションでは、シミュレートする体系の状態 (すなわち、気相、液相、固相のいずれか) に関わらず、粒子の初期配置として指定する密度の結晶構造を与える。この初期配置は、無秩序な粒子配置を持つ気相あるいは液相のシミュレーションに対して不自然に見えるが、次の技術的な観点から考えると妥当である。仮に、乱数を用いて無秩序な配置を決めたとき、たまたま 2 つの粒子がかなり接近した配置をとると、粒子間に非常に強い斥力が働き、次の瞬間にこれらの粒子は非物理的な移動を行う。このような不都合を避けるには、結晶構造の初期配置をとることが最も容易なのである。

粒子配置の初期条件として結晶構造を用いるには、与える結晶構造と 2.2 節で述べた周期境界条件とが整合性を持っていなければならない。この整合性を実現するには、粒子数 N を結晶構造の種類によって制限する。よく使われる体心立方格子 (bcc) や面心立方格子の場合の粒子数は、 p を正整数として $N = 2p^3$ (bcc), $N = 4p^3$ (fcc) なる整数に限定される。Fig. 2.2 に、 $p = 2$ に対応する fcc 配置を示した。一般に、結晶構造は、シミュレーションの対象となる物質が固相においてとる構造を選ぶ。例えば、アルゴンのシミュレーションでは fcc を用い、アルカリ金属では bcc を用いる。

一方、各粒子に与える初期速度 $v_i(0)$ は、目標とする温度 T_0 におけるボルツマン分布、

$$f(v) = \sqrt{\frac{m}{2\pi k_B T}} \exp\left[-\frac{mv^2}{2k_B T_0}\right], \quad (2.11)$$

にしたがう乱数 (正規分布の乱数) を使って無秩序に選ぶ。ここで、 k_B はボルツマン定数である。さらに、 $v_i(0)$ の条件として、

$$\sum_i v_i(0) = (0, 0, 0), \quad (2.12)$$

を付け加える。これは、(1) 体系の重心を静止させた方が後の解析に都合がよい、(2) 数値誤差の蓄積をできるだけ避ける、という 2 つの理由による。また、(2) の条件と相補的に、基本セルの重心を $(0, 0, 0)$ ととり、粒子の初期配置に、

$$\sum_i r_i(0) = (0, 0, 0), \quad (2.13)$$

なる条件を与えるとよい。

粒子の初期配置と初期速度を与えると、粒子系の全エネルギー E が決まる。ミクロカノニカルアンサンブルの MD シミュレーションでは、 E が時間によらず一定値をとる。一般に、上に述べた初期条件は体系の平衡状態からずれており、シミュレーションを開始すると、 $E =$ 一定の条件で、運動エネルギーからポテンシャルエネルギーへエネルギーの移動が起こる。すなわち、体系が平衡状態に近づくにつれ温度が減少する。そこで、指定する温度にお

ける平衡状態を実現するため、次に示すように各粒子の速度を調節する必要がある。

$$v_i^*(t) = v_i(t) \left[1 + \alpha \left(\sqrt{\frac{T_0}{T}} - 1 \right) \right], \quad T = \frac{m \sum_i v_i(t)^2}{3k_B N}. \quad (2.14)$$

ここで、 $v_i^*(t)$ は調節後の速度である。また、 α ($0 \leq \alpha \leq 1$) は速度調節の度合いを指定するパラメータである。 $\alpha = 0$ は速度調節を全くしないことになり、 α が 1 に近づくほど急激な調節になる。このような速度調整を繰り返すことによって、指定した温度に近づけることができる。温度の変化が小さくなったら、以後の計算では温度調節をすることなくシミュレーションを続ける。なお、3章で述べるカノニカルアンサンブルの MD シミュレーションでは、この速度調節の操作を自動的に実現することができる。

2.4 数値アルゴリズム

MD 法で解く運動方程式は、 $3N$ 元の連立常微分方程式である。数値解法を選択にあたっては、計算精度と各粒子に働く力の計算にかかる時間とのかね合いを考慮することになる。ここでは、MD 法でよく用いられる Verlet の方法 [3] を説明する。

粒子間相互作用が連続的なポテンシャルで表される場合、各粒子の位置座標と速度の時間発展は、時間 t を時間メッシュ h で等間隔に分割したグリッド上の時系列データとして表現される。時刻 $t \pm h$ における i 番目の粒子の座標は、 t のまわりのテーラー展開を使って、

$$r_i(t \pm h) = r_i(t) \pm r_i^{(1)}(t)h + \frac{1}{2}r_i^{(2)}(t)h^2 \pm \frac{1}{6}r_i^{(3)}(t)h^3 + \frac{1}{12}r_i^{(4)}(t)h^4 \pm \dots, \quad (2.15)$$

と表現できる。式 (2.15) を使うと、時刻 $t+h$ における位置座標と時刻 t における速度は、

$$r_i(t+h) = 2r_i(t) - r_i(t-h) + h^2 \frac{F_i(t)}{m} + O(h^4), \quad (2.16)$$

$$v_i(t) = \frac{1}{2h} [r_i(t+h) - r_i(t-h)] + O(h^2), \quad (2.17)$$

となる。これが、Verlet の方法である。この方法の特徴は、位置座標の差分式に速度が入らないため、体系のポテンシャルエネルギーの時間発展が粒子の速度とは独立に求められることである。なお、初期値 $r_i(0), v_i(0)$ から $r_i(h)$ を計算するには、

$$r_i(h) = r_i(0) + hv_i(0) + \frac{h^2}{2} \cdot \frac{F_i(0)}{m}, \quad (2.18)$$

を用いる。

実際に、差分式 (2.16) と (2.17) を用いてプログラムを作成するには、桁落ちによって発生する誤差の蓄積を避けるため、次のような方法をとる。まず、時刻 t から $t+h$ の間に生じる粒子 i の位置座標の変位

$$\begin{aligned} R_i(t+h) &= r_i(t+h) - r_i(t) \\ &= R_i(t) + h^2 \frac{F_i(t)}{m}, \end{aligned} \quad (2.19)$$

を定義し、局所的な桁落ちを防ぐ。そして、位置座標と速度を

$$\mathbf{r}_i(t+h) = \mathbf{r}_i(t) + \mathbf{R}_i(t+h), \quad (2.20)$$

$$\mathbf{v}_i(t) = \frac{1}{2h}[\mathbf{R}_i(t+h) + \mathbf{R}_i(t)], \quad (2.21)$$

のように計算する。ここで、式 (2.20) の計算は、

$$\mathbf{r}_i(nh) = \mathbf{r}_i(0) + \sum_{s=1}^n \mathbf{R}_i(sh), \quad (2.22)$$

とすることで、誤差の蓄積を防ぐ。

さて、式 (2.19)~(2.22) で表される Verlet の方法は、数値的に安定である。この方法の難点は、位置座標と速度を求める時刻が、時間メッシュ h だけずれていることである。この点を改良した方法 [12, 14] は、

$$\mathbf{r}_i(t+h) = \mathbf{r}_i(t) + h\mathbf{v}_i(t) + \frac{h^2}{2m}\mathbf{F}_i(t) + O(h^3), \quad (2.23)$$

$$\mathbf{v}_i(t+h) = \mathbf{v}_i(t) + \frac{h}{2m}[\mathbf{F}_i(t+h) + \mathbf{F}_i(t)] + O(h^3), \quad (2.24)$$

である。この方法は、2.3節で述べた MD シミュレーションの初期段階に必要な温度調節 (式 (2.14)) を行うのに都合がよい。

2.5 物理量の計算

MD シミュレーションでは、ニュートンの運動方程式から、 $\{\mathbf{r}_i(t), \mathbf{v}_i(t), i=1 \sim N\}$ の時系列を生成する。シミュレーションを進めていくと、力学量はほぼ一定値のまわりでゆらぐようになる。これをもって、シミュレーションの対象が、熱平衡状態に入ったものとみなす。熱平衡状態にある体系のさまざまな性質は、それぞれに対応する力学量の時系列データを、時間平均したもので与えられる。つまり、ある物理量 A に対応する力学量 $a[\mathbf{r}(t), \mathbf{v}(t)]$ を使って

$$A = \lim_{t \rightarrow \infty} \int_0^t a[\mathbf{r}(s), \mathbf{v}(s)] ds, \quad (2.25)$$

で定義される。実際には、離散データとして力学量 $a[\mathbf{r}(t), \mathbf{v}(t)]$ が与えられるので、

$$A = \frac{1}{N_s} \sum_{n=0}^{N_s-1} a[\mathbf{r}(t_0 + n\tau), \mathbf{v}(t_0 + n\tau)], \quad (2.26)$$

を使って計算する。ここで、 τ は時間メッシュ h の整数倍ととり、 N_s は時間平均に用いるサンプル数を表す。もし、力学量が 1 粒子あたりの量の場合には、さらに粒子数で平均することができ、

$$A = \frac{1}{N_s N} \sum_{n=0}^{N_s-1} \sum_{i=1}^N a_i[\mathbf{r}(t_0 + n\tau), \mathbf{v}(t_0 + n\tau)], \quad (2.27)$$

となる。この際、平均操作全体のサンプル数は、 $N_s N$ となる。平均操作における相対誤差は、大まかに (サンプル数) $^{-1/2}$ と見積もることができる。このため、1 粒子あたりの平均量の方

がよい精度で求められる。以下、シミュレーションを進める中で、特に重要な物理量についてまとめる。詳細については、文献 [12, 16, 17] を参照されたい。

2.5.1 運動エネルギーとポテンシャルエネルギー

2.1節で触れたように、ミクロカノニカルアンサンブルのMDシミュレーションでは、体系の全エネルギー E が保存する。したがって、体系の運動エネルギー K (すなわち温度 T) とポテンシャルエネルギー U は、時間とともにゆらいている。1粒子あたりの K および U の平均値は、次式で与えられる。

$$\frac{\langle K \rangle_t}{N} = \frac{1}{N} \sum_{i=1}^N \left\langle \frac{mv_i^2(t)}{2} \right\rangle_t, \quad (2.28)$$

$$\frac{\langle U \rangle_t}{N} = \frac{1}{N} \sum_{i=1}^N \left\langle \frac{1}{2} \sum_{j \neq i} u(r_{ij}(t)) \right\rangle_t. \quad (2.29)$$

ここで、時間平均操作を $\langle \dots \rangle_t$ で示した。したがって、1粒子あたりの内部エネルギー E/N は、 $(\langle K \rangle_t + \langle U \rangle_t)/N$ で与えられる。一方、温度 T は、求めた運動エネルギーの値から、

$$T = \frac{2}{3k_B} \cdot \frac{\langle K \rangle_t}{N}, \quad (2.30)$$

と計算する。MDシミュレーションの精度は、 E/N の保存の程度を調べることで吟味する。これは、 E/N の t に対する傾きを計算することで系統的に行うことができる。これと相補的に、体系の全運動エネルギーと重心の保存の様子を調べるとよい。これら結果を使って、シミュレーションの精度を決定するパラメータ、すなわち時間メッシュ h の適否を判断する。

2.5.2 圧力

圧力 P は、密度や温度とともに、シミュレーションから得られた物理量と実験の測定値とを比較・検討する上で、重要な熱力学量である。 P は、次に示すビリアル定理 (virial theorem) から計算することができる。

$$\begin{aligned} \frac{PV}{Nk_B T} &= 1 + \frac{1}{3Nk_B T} \sum_{i=1}^N \langle r_i(t) \cdot F_i(t) \rangle_t \\ &= 1 - \frac{1}{6Nk_B T} \sum_{i=1}^N \left\langle \sum_{j \neq i} r_{ij}(t) \frac{\partial u(r_{ij}(t))}{\partial r_{ij}} \right\rangle_t. \end{aligned} \quad (2.31)$$

この圧力の計算は、式 (2.29) に示したポテンシャルエネルギーの計算とともに、粒子に働く力の計算 (式 (2.7)) と同じ計算ブロックで行うことよい。

2.5.3 定積比熱

統計力学では、平均値のまわりのゆらぎ (分散) が重要な役割を持つ。その端的な例が、比熱である。ミクロカノニカルアンサンブルにおける定積比熱 C_V は、

$$C_V = \left(\frac{\partial E}{\partial T} \right)_V$$

$$\begin{aligned}
&= \left(\frac{\partial T}{\partial E} \right)_V^{-1} \\
&= k_B \left[1 - \left(1 - \frac{2}{3N} \right) \langle K \rangle_t \left\langle \frac{1}{K} \right\rangle_t \right]^{-1}, \quad (2.32)
\end{aligned}$$

と表される。通常、十分大きな N について（熱力学極限において）成り立つ近似式、

$$\left\langle \frac{1}{K} \right\rangle_t = \frac{1}{\langle K \rangle_t} \left(1 + \frac{\langle K^2 \rangle_t - \langle K \rangle_t^2}{\langle K \rangle_t} \right) + O\left(\frac{1}{N}\right), \quad (2.33)$$

を使って、

$$C_V = k_B \left(\frac{2}{3N} - \frac{\langle K^2 \rangle_t - \langle K \rangle_t^2}{\langle K \rangle_t^2} \right)^{-1} + O\left(\frac{1}{N}\right), \quad (2.34)$$

と書き直した式が計算に用いられる。比熱の概略値がわかっているならば、式(2.34)を使って温度ゆらぎの度合いを逆算することができ、目標温度設定のための速度スケージングに便利である。なお、比熱の計算では、よい統計精度を得るのが難しいため、長時間のシミュレーションが必要である。

2.5.4 2体分布関数

2体分布関数 $g(r, r')$ は、1個の粒子が r に存在するとき、位置 r' の体積要素 dr' 内に存在する平均粒子数を $\rho g(r, r')$ に等しいとおくことで定義される。均質等方的な体系において、 $g(r, r')$ は、 $|r - r'| \equiv r$ のみの関数 $g(r)$ となる。

2体分布関数は、MDシミュレーションによって得られた粒子の位置座標のデータから次のように求められる。 $n_i(r, t)$ を時刻 t に、粒子 i を中心とした半径 $r - \Delta r/2$ と $r + \Delta r/2$ の2球面ではさまれた球殻中に含まれる粒子数とする。 $g(r)$ は、

$$g(r) = \frac{1}{4\pi\rho r^2\Delta r} \cdot \frac{1}{N_s N} \sum_{i=1}^N \langle n_i(r, t) \rangle_t, \quad (2.35)$$

から得られる。ただし、距離 r は、基本セルの大きさにしたがって $r < L/2$ の範囲に限定する。長距離にわたって $g(r)$ を求めるには、粒子数を多くとる必要がある。また、 Δr は平均粒子間距離の $1/50$ 程度にとるとよい。

固体（結晶）の原子は規則的格子配置をとるので、 $g(r)$ は格子点に対応する距離で鋭いピークを持ち、このようなピークは長距離にわたって減衰しない。一方、液体の場合には、平均の粒子間距離の $3 \sim 4$ 倍程度の距離で粒子配置の相関が消え、 $g(r) \sim 1$ となる。また、気体においては、粒子配置の相関はほとんどなく、粒子の芯の部分を除いて $g(r) \sim 1$ となる。このように、2体分布関数 $g(r)$ を使って、シミュレーションの対象が気相、液相、固相のどの相に対応するか判断することができる。

2.5.5 平均2乗変位

平均2乗変位は、粒子がある定められた時刻 t_0 から、時間 t の間にどれだけ変位（自己拡散）するかを調べる量である。

平均 2 乗変位 $R_2(t)$ は,

$$R_2(t) = \frac{1}{N} \sum_{i=1}^N \langle |r_i(t+t_0) - r_i(t_0)|^2 \rangle_t, \quad (2.36)$$

によって定義される。ブラウン運動の理論からよく知られているように、大きな t について、気体や液体における $R_2(t)$ は t に比例するようになり、

$$R_2(t) = 6Dt + C \quad (2.37)$$

と書ける。ここで、 D は自己拡散係数であり、 C は定数である。固体の中の原子は、勝手に拡散することはないので、 $D=0$ である。そこで、 $R_2(t)$ の振る舞いを調べることで、シミュレーションの対象が固相と流動相（液相あるいは気相）のいずれに対応するかを区別することができる。

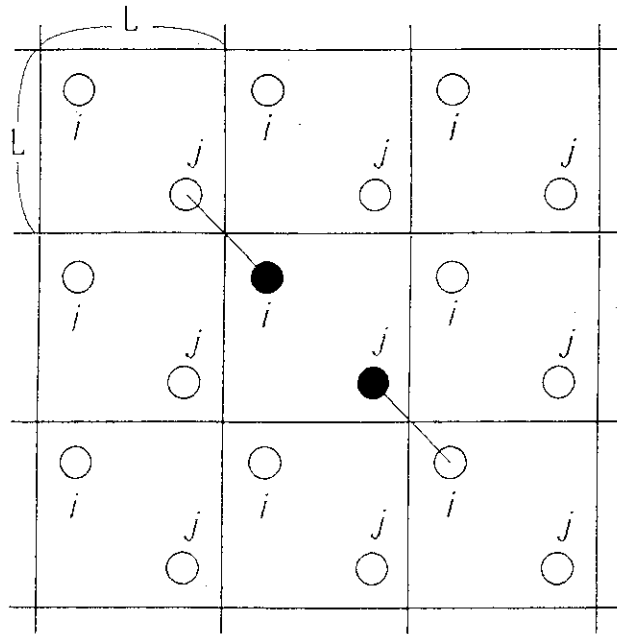


Fig. 2.1 Periodic boundary condition for two dimensional system. The central square corresponds to the fundamental cell. The other squares corresponds to the image cells. The nearest distance between i -th and j -th particles is shown by solid line.

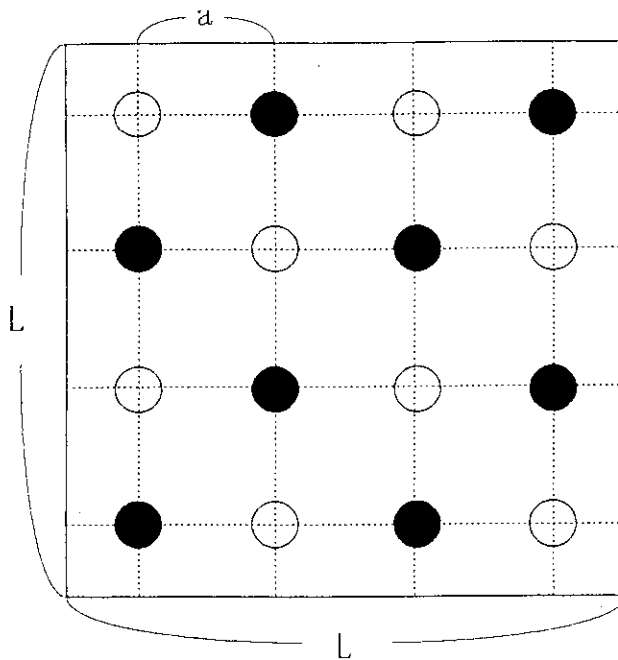


Fig. 2.2 Configuration of 32 particles face-centered lattice. a is the lattice constant. \bigcirc denotes a lattice point on the paper. \bullet denotes a lattice point on plains which perpendicularly separate from the paper by distance a .

3. 等温分子動力学法

2章で述べたMD法では、体系の全エネルギー E 、体積 V 、粒子数 N が一定に保たれていた。この NVE 一定のMD法は、熱平衡状態のシミュレーションの基礎となる方法である。しかしながら、温度 T が平均値（期待値）として与えられること、そして、自由自在に T を制御するのが難しい、といった問題がある。1980年代に入り、温度をパラメータとして入力すれば、その値へ向かって系が自動的に緩和するような力学系が考案された [5]。

温度一定の統計集団（カノニカルアンサンブル）は、温度 T の熱浴と熱平衡にある体系の集団である。それぞれの体系は、熱浴とエネルギーをやりとりすることができ、これによって体系の温度が一定に保たれる。もし熱浴に接している効果を仮想的な力 $F_i^C(t)$ で表現できたとすると、拡張されたニュートンの運動方程式

$$m \frac{d^2 \mathbf{r}_i(t)}{dt^2} = \mathbf{F}_i(t) + \mathbf{F}_i^C(t), \quad (3.1)$$

を使って、温度一定のシミュレーションが可能になると考えられる。ここで、 $\mathbf{F}_i(t)$ は粒子間相互作用（2体ポテンシャル $u(r)$ ）によって生じる力である。 $\mathbf{F}_i^C(t)$ のとり方は、大きく分けて3つの方法がある [5]。

- (1) 乱数を用いる方法 [10, 11]: 熱浴が仮想的な粒子から構成されていると考え、減衰力と熱浴構成粒子から受けるランダムな力の和を \mathbf{F}_i^C として用いる。
- (2) 束縛法 [6, 7]: 運動エネルギー（温度）の値が、時間変動せず、常に $K = 3Nk_B T/2$ を満足するような束縛条件（束縛力）を、 \mathbf{F}_i^C として用いる。
- (3) 拡張系の方法 [8]: 小数の自由度を持つ仮想的体系を熱浴として導入し、粒子系と仮想的体系を合わせた運動方程式をつくる。 \mathbf{F}_i^C は、目標の温度と時々刻々の“温度”とのずれに基づくフィードバックとして与えられる。

以下、上にあげた3つの方法についてまとめる。

3.1 乱数を用いる方法

この方法では、 \mathbf{F}_i^C として減衰力とランダムな力（揺動力）を使ったランジュヴァン型の運動方程式

$$m \frac{d^2 \mathbf{r}_i(t)}{dt^2} = \mathbf{F}_i(t) - \zeta m \frac{d\mathbf{r}_i(t)}{dt} + \boldsymbol{\eta}_i(t), \quad (3.2)$$

を用いる。ここで、 ζ (≥ 0) は抵抗係数であり、右辺の第2項と第3項は、それぞれ、減衰力とランダムな力を表す。これらの力と温度とは、揺動散逸定理

$$\langle \eta_{i\alpha}(t_1) \eta_{j\beta}(t_2) \rangle = 2mk_B T \zeta \delta_{ij} \delta_{\alpha\beta} \delta(t_1 - t_2), \quad (3.3)$$

によって関係づけられている。ここで、 α, β は x, y, z 成分のいずれかを表し、 $\delta_{ij}, \delta_{\alpha\beta}$ はクロネッカーデルタであり、そして $\delta(t_1 - t_2)$ はデルタ関数である。また、 $\langle \dots \rangle$ はカノニカル平均を表す。

MDシミュレーションで用いる時間メッシュを h とすれば、式 (3.3) を満足するランダムな力は、

$$\eta_{i\alpha}(t) = \frac{1}{h} \sqrt{\frac{2mk_B T \zeta h}{p}} \gamma_{i\alpha}(t) \theta(p - a_{i\alpha}), \quad (3.4)$$

となる。ただし、 p は時刻 t でランダムな力が働く確率であり、 $\gamma_{i\alpha}$ は分散が 1 のガウス分布乱数であり、 $a_{i\alpha}$ は 0 と 1 の間に分布する一様乱数であり、そして、 $\theta(x)$ はステップ関数である。式 (3.2) および (3.4) から明らかなように、 $\zeta \rightarrow 0$ とすると、通常のニュートンの運動方程式が得られる。

この方法を使ったシミュレーションには、ミクロカノニカルアンサンブルの MD シミュレーションに現れない特別な入力パラメータとして、 p, ζ が必要となる。特に、 ζ の値の選び方には、注意を要する。 ζ が大きすぎると粒子配置が平衡状態に近づきにくくなり、逆に ζ が小さすぎると目標の温度が得られにくくなる。そこで、 p, ζ の値は、試験的なシミュレーションを数回繰り返して、経験的に定める。実際に行われたソフト球モデルの MD シミュレーション [18] では、 $h = 0.01\tau$ としたとき (τ は熱速度で粒子間の平均距離を移動する時間である)、 $p = 0.3$ 、 $\zeta = 0.02/h$ なる値が用いられている。

ニュートンの運動方程式 ($F_i^c = 0$) では、各粒子に働く力が内力だけなので、体系の重心が保存される。ランダムな力を含む運動方程式 (3.2) の場合、重心が保存するか否かは、実際の計算で用いる疑似乱数 $\eta_{i\alpha}, a_{i\alpha}$ の質に依存する。このため、シミュレーションに先立って、疑似乱数の検定を十分行うことが大切である。

3.2 束縛法

温度が時間によらず一定をとるための束縛条件は、

$$C(\mathbf{r}_1, \dots, \mathbf{r}_N, \mathbf{v}_1, \dots, \mathbf{v}_N) = \sum_i^N \frac{mv_i^2}{2} - \frac{g}{2} k_B T = 0, \quad (3.5)$$

である。ここで、 g は体系の自由度の数である。束縛法は、ニュートンの運動方程式に、式 (3.5) を満たすような束縛力 F_i^c を付け加える方法である。

束縛条件から束縛力を求める方法の説明のために、簡単な束縛条件

$$C(x, \dot{x}, t) = 0, \quad (3.6)$$

を考えよう。この束縛条件が時間によらず常に満たされるには

$$\frac{dC}{dt} = \ddot{x} \cdot \frac{\partial C}{\partial \dot{x}} + \dot{x} \cdot \frac{\partial C}{\partial x} + \frac{\partial C}{\partial t} = 0, \quad (3.7)$$

が成り立たなければならない。ガウスの最小作用の原理によると、 $\partial C / \partial \dot{x}$ に平行な方向に束縛力をとると、その束縛力は最小なものになる。温度一定の束縛条件では、 $\partial C / \partial \dot{x}$ の方向は、各粒子の速度の方向に一致する。そこで、束縛法における運動方程式は、

$$\frac{d\mathbf{r}_i(t)}{dt} = \mathbf{v}_i(t), \quad (3.8)$$

$$m \frac{d^2\mathbf{r}_i(t)}{dt^2} = \mathbf{F}_i(t) - \zeta(t) m \mathbf{v}_i(t), \quad (3.9)$$

と書き表せる。ここで、 $\zeta(t)$ は、未定変数である。

式 (3.9) に含まれる未定変数 $\zeta(t)$ は、式 (3.5) で示した束縛条件を使って次のように求める。式 (3.5) の両辺を時間について微分すると、

$$\sum_{i=1}^N m \mathbf{v}_i(t) \cdot \dot{\mathbf{v}}_i(t) = \sum_{i=1}^N \mathbf{v}_i(t) \cdot (\mathbf{F}_i(t) - \zeta(t) m \mathbf{v}_i(t)) = 0, \quad (3.10)$$

となる。この式から、

$$\zeta(t) = -\frac{\sum_i \mathbf{v}_i(t) \cdot \mathbf{F}_i(t)}{g k_B T}, \quad (3.11)$$

が得られる。ここで、体系の自由度 g は、束縛条件を課したので、 $3N$ から 1 を減じた、 $g = 3N - 1$ である。また、 ζ はポテンシャルエネルギー U の時間微分と

$$\frac{dU(t)}{dt} = -\sum_i \mathbf{v}_i(t) \cdot \mathbf{F}_i(t) = -\zeta(t) g k_B T, \quad (3.12)$$

なる関係を持つ。熱平衡状態では、ポテンシャルエネルギーの平均値が一定となる。したがって、ポテンシャルエネルギーの時間微分の平均値はゼロとなる。つまり、シミュレーションの対象が熱平衡状態に入ったかどうかは [6]、

$$\langle \zeta(t) \rangle_t = 0, \quad (3.13)$$

が達成されたかどうかを調べればよい。なお、式 (3.12) は、ミクロカノニカルアンサンブルの MD シミュレーションにおける全エネルギーの保存則と同様に、シミュレーションの精度の検定に用いる。

式 (3.8), (3.9), そして (3.11) で示される束縛法は、(1) 経験的に決めるパラメータをいっさい含まない、(2) 粒子配置がカノニカル分布にしたがう、(3) 全運動量がゼロの初期条件で重心の保存が保証される、(4) 非平衡状態のシミュレーションへの拡張が容易である、という利点を持つ。

3.3 拡張系の方法

この方法では、 N 個の粒子系とただ 1 つの自由度を持つ仮想的な熱浴をあわせた $3N + 1$ 自由度の体系（拡張系）を考える。

拡張系における運動方程式は、

$$\frac{d\mathbf{r}_i(t)}{dt} = \mathbf{v}_i(t), \quad (3.14)$$

$$\frac{d^2\mathbf{r}_i(t)}{dt^2} = \mathbf{F}_i(t) - \zeta(t) m \mathbf{v}_i(t), \quad (3.15)$$

$$\frac{d\zeta(t)}{dt} = \frac{1}{Q} \left(\sum_{i=1}^N m \mathbf{v}_i(t)^2 - 3N k_B T \right), \quad (3.16)$$

$$\frac{d \ln s(t)}{dt} = \zeta(t), \quad (3.17)$$

と表現される。ここで、 Q 、 $s(t)$ および ζ は、それぞれ、仮想的な熱浴の“質量”（[エネルギー][時間²]の次元），“位置座標”（無次元）と“速度”（[時間⁻¹]の次元）である。この連立微分方程式にしたがう体系では、次の“ハミルトニアン”が保存量となる。

$$\bar{H} = \sum_{i=1}^N \frac{mv_i(t)^2}{2} + U(\mathbf{r}_1, \dots, \mathbf{r}_N) + \frac{1}{2}Q\zeta(t)^2 + 3Nk_B T \ln s(t). \quad (3.18)$$

定常的な時系列では、時間についての1次微分量の平均値は0である。したがって、式(3.16)により、熱平衡状態において、仮想的な熱浴を除いた N 個の粒子系の運動エネルギーの平均値は $3Nk_B T/2$ であることが保証される。なお、 $Q \rightarrow 0$ の極限で、拡張系の方法は、3.2節で述べた束縛法に一致する。拡張系の方法では、束縛法と同様に体系の重心が保存する。束縛法との相違点は、粒子配置だけでなく、粒子速度もカノニカル分布にしたがう点である。

拡張系の方法に現れる特別な入力パラメータ Q は、粒子数が多いほど大きくとる。粒子間相互作用が式(2.3)で示したレナードジョーンズポテンシャルで与えられる場合、 m 、 σ を質量、長さの単位にとると、仮想熱浴の無次元化された質量 $Q' = m\sigma^2 Q$ を定義できる。粒子数 $N = 108$ の系では $Q' \simeq 4$ でよいといわれている [12]。そこで、この値を参考にして、ほぼ粒子数に比例して Q' の値を決めるとよい。

4. 分子動力学シミュレーションコード ISIS

ISIS コードは、ソフト球モデルを対象として、新規に開発した MD シミュレーションコードである。ISIS コードを作成したきっかけは、筆者がウィーン工科大学 (Technische Universität Wien) へ留学中に行った液体アルカリ金属の MD シミュレーション (マイクロカノニカルアンサンブル, 10^5 ステップ) で、非常に多くの積分ステップを精度よく行うことの困難さ、そして、マイクロカノニカルアンサンブルにおける温度制御の難しさを実感したところにある。ISIS コード開発の目的は、(1) MD シミュレーション手法の習得、(2) 高精度の数値アルゴリズムを用いたシミュレーションの信頼性向上、そして、(3) 等温 MD 法を使った簡単な温度調節の実現、の 3 点である。(2) および (3) を実現するために、ISIS コードは、Verlet のアルゴリズムを拡張した第 5 次の差分式、そして、経験的パラメータを含まない等温 MD 法である束縛法を採用した。ISIS コードの特徴は、

- (a) 同種粒子からなる体系のみならず 2 種類の粒子からなる体系のシミュレーションができること
- (b) 簿記法 (bookkeeping method) の利用によって計算時間を短縮したので取り扱える粒子数が多いこと
- (c) カノニカルアンサンブルとマイクロカノニカルアンサンブルのいずれかを選択してシミュレーションを行うことができる
- (d) ベクトル計算機の使用によって飛躍的に計算時間が短縮されること

の 4 点である。計算時間の評価については 5 章で詳しく述べることにして、以下、ソフト球モデルの概要、数値アルゴリズム、簿記法、計算できる物理量、プログラムの構成、の 5 点についてまとめる。

4.1 ソフト球モデルと無次元化運動方程式

ソフト球モデルは、逆べきポテンシャル (式 (2.5) 参照) で粒子間相互作用が与えられるモデル物質である。逆べきポテンシャルは、引力部分を全く含まないため、見た目には現実離れた相互作用である。しかしながら、逆べきポテンシャルに Kac ポテンシャルと呼ばれる引力を付け加えた粒子間相互作用

$$u(r) = \epsilon \left(\frac{\sigma}{r} \right)^n - \alpha \gamma^3 \exp[-\gamma r], \quad (\alpha \geq 0), \quad (4.1)$$

を考え、熱力学的極限をとった後で $\gamma \rightarrow 0$ とすると、

$$\frac{P\beta}{\rho} = \frac{P_0\beta}{\rho} - 4\pi\alpha\rho\beta, \quad (4.2)$$

なる van der Waals 型の状態方程式となることが示される [16, 21, 22]。ここで、 P_0 は $\alpha = 0$ すなわちソフト球モデルの圧力である。つまり、式 (4.1) で与えられるモデル物質 (理想 3

相モデルと呼ばれる)では, 斥力部分(逆べきポテンシャル)と引力部分(Kacポテンシャル)の独立な取り扱いが許されるのである。そこで, 式(4.2)を通して, ソフト球モデルと現実の物質との対応関係を吟味することができる。実際に理想3相モデルは, 適切な n をとることで, 現実的な物質の性質をよく再現することがわかっている [16, 21, 22]。

さて, ソフト球モデルの重要な利点は, 逆べきポテンシャルのスケーリングの性質によって,

$$\Gamma = \rho \sigma^3 (\epsilon \beta)^{3/n}, \quad (4.3)$$

なる1つの結合定数で状態を表すことができる点である。これは, スケーリングの性質を持たないモデル(例えば Lennard-Jones ポテンシャル)が2つの変数(例えば温度 T と圧力 P)を必要とするのと比べて非常に便利である。スケーリングの性質は, 次のように導くことができる [16]。

N 個のソフト球が体積 $V = L^3$ を占めている体系を考える。この系のハミルトニアン H は,

$$H = \sum_{i=1}^N \frac{mv_i^2}{2} + \sum_{i>j} \epsilon \left(\frac{\sigma}{r_{ij}} \right)^n, \quad (4.4)$$

で与えられる。長さの単位 l と時間の単位 τ を

$$l = \rho^{-1/3}, \quad (4.5)$$

$$\tau = l \sqrt{m\beta}, \quad (4.6)$$

のようにとる。ここで, l の物理的な意味は平均の粒子間距離であり, τ は熱速度によって l だけ移動する時間である。 l と τ によってスケールされた無次元の“位置座標” r^* , “時間” t^* , そして“速度” v^* を

$$r^* = l^{-1} r, \quad (4.7)$$

$$t^* = \tau^{-1} t, \quad (4.8)$$

$$v^* = \tau l^{-1} v, \quad (4.9)$$

と定義する。これらを用いると, 式(4.4)のハミルトニアンは,

$$\begin{aligned} H &= m\tau^{-2}l^2 \sum_{i=1}^N \frac{v_i^{*2}}{2} + \epsilon \sigma^n l^{-n} \sum_{i>j} \left(\frac{1}{r_{ij}^*} \right)^n \\ &= \frac{1}{\beta} \left[\frac{1}{2} \sum_{i=1}^N v_i^{*2} + \Gamma^{n/3} \sum_{i>j} \left(\frac{1}{r_{ij}^*} \right)^n \right], \end{aligned} \quad (4.10)$$

と書き直される。したがって, 温度 T におけるカノニカル分布は,

$$\exp[-\beta H] = \exp \left[-\frac{1}{2} \sum_{i=1}^N v_i^{*2} - \Gamma^{n/3} \sum_{i>j} \left(\frac{1}{r_{ij}^*} \right)^n \right], \quad (4.11)$$

となり, 系の熱力学的状態は Γ のみで指定できることがわかる。

次に、無次元化された運動方程式を導こう。式 (3.8), (3.9), および (3.11) で与えられる束縛法の運動方程式は,

$$\frac{dr_i^*(t)}{dt} = v_i^*(t), \quad (4.12)$$

$$\frac{d^2r_i^*(t)}{dt^{*2}} = F_i^*(t) - \zeta^*(t)v_i^*(t), \quad (4.13)$$

$$\zeta^*(t) = \tau\zeta(t) = -\frac{\sum_i v_i^*(t) \cdot F_i(t)}{3N-1}, \quad (4.14)$$

$$F_i^*(t) = n\Gamma^{n/3} \sum_{i \neq j} \frac{r_{ij}^*(t)}{r_{ij}^*(t)^{n+2}}, \quad (4.15)$$

と無次元の形式に変形される。また、式 (4.14) のかわりに $\zeta^*(t) = 0$ とすれば、式 (4.12), (4.13), そして、(4.15) は、無次元化されたニュートンの運動方程式となる。この際、次の注意が必要である。ミクロカノニカルアンサンブルでは、温度 T が平均値として定義されるため、式 (4.3) で与えられる結合定数 Γ も同様にシミュレーションの後に得られる量である。そこで、目標の温度 T_0 に対応して Γ_0 を定め、

$$\frac{dr_i^*(t)}{dt} = v_i^*(t), \quad (4.16)$$

$$\frac{d^2r_i^*(t)}{dt^{*2}} = n\Gamma_0^{n/3} \sum_{i \neq j} \frac{r_{ij}^*(t)}{r_{ij}^*(t)^{n+2}}, \quad (4.17)$$

$$\tau = l\sqrt{m\beta_0}, \quad (4.18)$$

なる無次元化された運動方程式を用いる。ここで、 $\beta_0 = (k_B T_0)^{-1}$ である。平均値としての Γ は観測された温度

$$T = \frac{1}{3k_B N} \sum_{i=1}^N \langle v_i^*(t)^2 \rangle_t, \quad (4.19)$$

を使い

$$\Gamma = \Gamma_0 \left(\frac{T_0}{T} \right)^{3/n}, \quad (4.20)$$

と計算する。

さて、ここまで同種粒子からなる体系のシミュレーションに説明を限定してきたが、述べてきた手法を数種類の粒子から構成される系へ拡張するのは容易である。“直径”が σ_1, σ_2 , 質量が m_1, m_2 , なる2種類の粒子を、それぞれ N_1, N_2 個含む体系を考える ($N = N_1 + N_2$)。また、粒子間ポテンシャルが

$$u_{ij}(r) = \epsilon \left(\frac{\sigma_{ij}}{r} \right)^n, \quad \sigma_{ij} = \frac{\sigma_1 + \sigma_2}{2}, \quad (4.21)$$

で与えられるとする。長さの単位 l を式 (4.5) で定義し、時間の単位 τ を

$$\tau = l\sqrt{m_1\beta}, \quad (4.22)$$

で定義する。このとき、式 (4.15) のかわりに、次の無次元化された力

$$F_i^*(t) = \frac{nm_1}{m_i} \Gamma_{\text{eff}}^{n/3} \sum_{i \neq j} \left(\frac{\sigma_{ij}}{\sigma_{\text{eff}}} \right)^n \frac{r_{ij}^*(t)}{r_{ij}^*(t)^{n+2}}, \quad (4.23)$$

と、式 (4.12), (4.13), (4.14) を組み合わせれば 2 成分系の無次元化された束縛法の運動方程式が得られる。ここで、

$$\Gamma_{\text{eff}} = \rho \sigma_{\text{eff}}^3 (\epsilon \beta)^{3/n}, \quad \sigma_{\text{eff}} = \sum_i \sum_j x_i x_j \sigma_{ij}^3, \quad x_1 = 1 - x_2 = \frac{N_1}{N}, \quad (4.24)$$

である。2 成分のソフト球モデルも、スケーリングの性質を持つので、熱力学変数は 2 つでよい（通常、2 成分系は、温度、圧力、濃度 x_1 の 3 つの変数を持つ）。1 成分モデルとの対応をつけやすいので、普通、 Γ_{eff} と x_1 を変数として用いる。

ここで、長さの単位 l を使った際の、粒子数とシミュレーションに用いる基本セルの大きさとの関係について触れよう。基本セルの 1 辺の長さ L は、

$$L = V^{1/3} = N^{1/3} \left(\frac{V}{N} \right)^{1/3}, \quad (4.25)$$

で与えられるので、 l を使って無次元化すると、

$$L^* = l^{-1} L = N^{1/3} \quad (4.26)$$

となる。2.3 節で述べたように、周期境界条件を使ってシミュレーションを行う場合、結晶構造と整合性をもった粒子数をとる場合が多い。例えば、fcc 格子では $N = 4p^3$, bcc 格子では $N = 2p^3$ なる値である。このとき L^* は、

$$L^* = \begin{cases} \sqrt[3]{4}p, & \text{fcc} \\ \sqrt[3]{2}p, & \text{bcc} \end{cases}, \quad (4.27)$$

なる値をとる。この関係は、 l を長さの単位にとれば 2 体ポテンシャルのとり方に依存せず常に成り立つので、プログラミングが簡単になるのみならず、異なる物質のシミュレーション結果を比較するのに好都合である。

4.2 数値アルゴリズム

ISIS コードで用いる運動方程式の積分アルゴリズムには、多くの積分ステップを安定して行うことができ、しかも、ニュートンの運動方程式だけでなく束縛法の運動方程式も扱えることが要求される。3 章でふれた束縛法などの等温 MD 法では、時々刻々の粒子速度を通じて体系の温度制御を行うので、正確な速度の予測がきわめて重要となる。2.4 節で説明した Verlet のアルゴリズムは数値的に安定な方法であるが、速度に対する局所誤差が $O(h^2)$ なので、束縛法の運動方程式を解くには不向きである。そこで、ISIS コードでは Bernu の積分アルゴリズムを採用した。

4.2.1 ミクロカノニカルアンサンブルの場合

Bernu のアルゴリズムでは、Verlet のアルゴリズムで無視した座標の高階微分の項を取り入れることで、座標、速度に対する局所誤差をそれぞれ $O(h^6)$, $O(h^5)$ としている。ミク

ロカノニカルアンサンブルの MD 法に対する無次元化された差分式は、

$$r_i^*(t+h) = r_i^*(t) + R_i^*(t), \quad (4.28)$$

$$hv_i^*(t+h) = R_i^*(t) + h^2 Z_i^*(t), \quad (4.29)$$

$$R_i^*(t) = R_i^*(t-h) + F_i^*(t)h^2 + \frac{1}{12}F_i^{*(2)}(t)h^4 + O(h^6), \quad (4.30)$$

$$Z_i^*(t) = \frac{1}{2}F_i^*(t) + \frac{2}{3!}F_i^*(t)h + \frac{3}{4!}F_i^{*(1)}(t)h^2 + \frac{4}{5!}F_i^{*(2)}(t)h^3 + O(h^4), \quad (4.31)$$

である。この差分式を評価する際に問題となるのは、力の高階微分の取扱いである。Bernu のアルゴリズムでは、 $F_i^{*(2)}(t)$ と $Z_i^*(t)$ を、次のように表現する。

$$h^2 F_i^{*(2)}(t) = F_i^*(t+h) - 2F_i^*(t) + F_i^*(t-h), \quad (4.32)$$

$$Z_i^*(t) = \frac{97}{360}F_i^*(t+h) + \frac{19}{60}F_i^*(t) - \frac{13}{90}F_i^*(t-h) + \frac{1}{45}F_i^*(t-2h). \quad (4.33)$$

ここで、式 (4.32), (4.33) に時間 $t+h$ における力が現れることに注意しよう。粒子に働く力は、同時刻における粒子の座標を知らなければ計算することができない。そこで、

$$h^2 F_i^{*(2)}(t) = 2F_i^*(t) - 5F_i^*(t-h) + 4F_i^*(t-2h) - F_i^*(t-3h), \quad (4.34)$$

なる表現を使って $r_i^*(t+h)$ の予測を行ったのち、式 (4.32) を用いて予測値の修正を行う。この修正を多く行うほど計算精度が高まる。しかし、修正のたびに粒子間力 F_i^* の計算が必要となり計算時間の増大を招くので注意が必要である

4.2.2 束縛法のための変更点

等温 MD 法の運動方程式では、粒子の位置座標 r_i^* の 2 階微分に粒子間力 F_i^* のほかに粒子速度を含む項 $-\zeta^* v_i^*$ がつけ加わる。このような運動方程式は、通常、1 階の連立微分方程式に書き直して積分する。しかし、制御係数 ζ^* が陽に与えられる束縛法は、Bernu のアルゴリズムを使って解くことができる。式 (4.29) に式 (4.33) を代入した差分式を、

$$hv_i^*(t+h) = R_i^*(t) + h^2[C_0 F_i^*(t+h) + F_i^*], \quad (4.35)$$

と書こう。ここで、

$$F_i^*(t) = F_i^*(t) - \zeta^* v_i^*(t),$$

$$F_i^* = C_1 F_i^*(t) + C_2 F_i^*(t-h) + C_3 F_i^*(t-2),$$

であり、 C_0, C_1, C_2, C_3 は定数である。式 (4.35) から、

$$\begin{aligned} X_i &= R_i^*(t) + h^2[C_0 F_i^*(t+h) + F_i^*] \\ &= h(1 + hC_0 \zeta^*) v_i^*(t+h), \end{aligned} \quad (4.36)$$

なる X_i を定義する。 X_i^2 の和は

$$\begin{aligned} \sum_{i=1}^N X_i^2 &= h^2(1 + hC_0 \zeta^*)^2 \sum_{i=1}^N v_i^*(t+h)^2 \\ &= (3N-1)(1 + hC_0 \zeta^*)^2 h^2, \end{aligned} \quad (4.37)$$

と計算される。したがって、

$$\mathbf{v}_i^*(t+h) = \sqrt{\frac{\sum \mathbf{X}_i^2}{3N-1}} \mathbf{X}_i, \quad (4.38)$$

が得られる。一方、制御係数 ζ^* は、次のようにして求める。 \mathbf{v}_i^* と \mathbf{F}_i^* の内積の和

$$\begin{aligned} \sum_{i=1}^N \mathbf{X}_i \cdot \mathbf{F}_i^*(t+h) &= h(1+hC_0\zeta^*) \sum_{i=1}^N \mathbf{v}_i^*(t+h) \cdot \mathbf{F}_i^*(t+h) \\ &= (3N-1)\zeta^*(1+hC_0\zeta^*)h, \end{aligned} \quad (4.39)$$

と式 (4.37) を使えば、

$$\begin{aligned} \zeta^* &= \frac{1}{h(1+hC_0\zeta^*)} \cdot \frac{\sum \mathbf{X}_i \cdot \mathbf{F}_i^*(t+h)}{3N-1} \\ &= \frac{\sum \mathbf{X}_i \cdot \mathbf{F}_i^*(t+h)}{\sqrt{(3N-1) \sum \mathbf{X}_i^2}}, \end{aligned} \quad (4.40)$$

となり、 ζ^* が得られる。

4.2.3 数値誤差の評価

運動方程式を積分する際の数値誤差は、各種保存量をモニターすることで評価する。ミクロカノニカルアンサンブルの MD 法では、全エネルギー E が保存される。したがって、シミュレーションによって得られた時系列 $E(t)$ を

$$E(t) \simeq at + \langle E(t) \rangle_t, \quad (4.41)$$

と 1 次近似したとき、傾き a が誤差の目安となる。また、束縛法を用いたシミュレーションの場合は、式 (3.12) からのずれを示す。

$$\Delta = \left| \beta U(0) - \beta U(t) + (3N-1) \int_0^t \zeta^*(s) ds \right|, \quad (4.42)$$

なる Δ を計算し、 $\Delta = 0$ が要求される精度の範囲内で成立するか否かを評価する。ミクロカノニカルアンサンブルの MD 法と束縛法のどちらも全運動量が保存するので、上に述べた数値誤差の目安とともにモニターする必要がある。

ここで、実際に ISIS コードを実行させ、上記の方法で評価した数値誤差を示す。シミュレーションの対象は 1 成分のソフト球モデル ($n = 12$) で、粒子数 $N = 500$ 、結合定数 $\Gamma = 0.8$ (液体)、時間メッシュ $h = 0.0025\tau$ 、なるパラメータを使った。 10^4 ステップのシミュレーションのデータから、ミクロカノニカルアンサンブルのシミュレーションと束縛法を用いたシミュレーションの双方に対して、それぞれ、

$$\frac{|a|}{E} < 0.00001\%, \quad \frac{\Delta}{\beta U} < 0.005\%, \quad (4.43)$$

という極めて良好な結果を得た。

4.3 簿記法

粒子間力の計算は MD シミュレーションで最も計算時間のかかる部分である。2 体ポテンシャルによって相互作用する N 粒子系において、すべての粒子対に対して力を求めるのに必要な計算時間は $N(N-1)/2$ に比例して増加する。この事情は、ポテンシャルのカットオフ半径 r_c を設定しても変わらない。したがって、大きな N を用いたシミュレーションを行うには、計算時間短縮のための工夫が必要である。ISIS コードでは、簿記法 (bookkeeping method) の改良版を用いている。

簿記法は、Verlet が 864 個の Lennard-Jones 系の MD シミュレーションで導入した方法であり、次のようなものである。

- (1) すべての粒子対の距離 r_{ij} を求める。
- (2) 各粒子について、その粒子から $r_M (> r_c)$ 以内にある粒子の表をつくる。
- (3) n ステップの間、(2) で得られた表に含まれる粒子対を使って力を計算する。
- (4) (1) に戻る。

この方法を用いると、(1) の段階では $N(N-1)/2$ 組の粒子対の計算が必要であるが、(3) の段階に含まれる粒子の組み合わせは $N_M(N-1)/2$ であり単に N に比例したものとなる。ここで、 N_M は半径 r_M の球内に含まれる平均の粒子数である。

簿記法に現れるパラメータ n および r_M は、(2) の段階で作成した表に含まれない粒子が (3) の過程で半径 r_c の球内に入り込まないようにとる (Fig. 4.1)。 r_M と n は数密度や温度に依存し、経験的に決めざるをえない。1つの目安は、

$$r_M - r_c \gtrsim \bar{v}nh, \quad (4.44)$$

である。ここで、 $\bar{v} = \sqrt{\frac{k_B T}{3m}}$ である。さて、粒子対の格納された表を、固定された n ステップごとでなく、自動的に更新することができれば、最適化するパラメータの数が1つになり非常に便利である。ISIS コードでは、次に示す方法を用いて表の自動的な更新を実現している。まず、最新の更新時点 s_0 から s ステップ経過後の粒子の 2 乗変位

$$R_{2i}(sh) = |r_i(s_0 + sh) - r_i(s_0)|^2, \quad (4.45)$$

を計算する。そして、

$$\alpha(r_M - r_c)^2 \leq \max[R_{2i}(sh)], \quad (4.46)$$

なる条件を満足した時点で、表の更新を行う。通常、 α は、2つの粒子が向き合って、変位の最大値 $\max[R_{2i}(sh)]$ だけ近づく場合を考え、 $\alpha = \frac{1}{4}$ ととればよい。もし保存量に変化が現れるようであれば、 α の値を小さくする。

ここで、粒子対の登録方法について簡単に述べる。簿記法を使ったプログラム例を、Fig. 4.2 に示す。このプログラムは、ISIS コードに現れるものを、説明のために 1 成分 1 次元系へ書き換えたものである。粒子対を登録した表は、リスト配列 IPAIR とポインター配列 NDX から構成されている。粒子 i に対して、 $r_{ij}^2 < r_M^2$ を満たす粒子 j を順に取り出し、その粒子番号列を配列 IPAIR に格納する。そして、グループ i に登録される粒子数が、配列 NDX に格納される。サブプログラム TABLE および FORCE のいずれも、DO ループ 20 をベクトル処理

することができる。なお、配列 IPAIR の大きさが r_M^3 に比例するので、大きな r_c ほど多くの記憶容量が必要である。

4.4 計算できる物理量

ISIS コードを用いた MD シミュレーションから計算できる物理量を Table 4.1 にまとめる。また、2成分系のシミュレーションでの拡張点を Table 4.2 にまとめた。

通常、カノニカルアンサンブルあるいはミクロカノニカルアンサンブルの MD シミュレーションでは、いくつかの密度・温度の組についてシミュレーションを行わなければ、等温圧縮率 χ_T 、熱膨張係数 α_P 、比熱比 γ 、等温音速 v_T 、断熱音速 v_S を計算することができない。しかし、ソフト球モデルは逆べきポテンシャルのスケーリングの性質から、

$$\rho \left(\frac{\partial}{\partial \rho} \right)_T = \Gamma \frac{\partial}{\partial \Gamma}, \quad \beta \left(\frac{\partial}{\partial \beta} \right)_V = \frac{3}{n} \cdot \Gamma \frac{\partial}{\partial \Gamma}, \quad (4.47)$$

なる関係が成立するので、1つの Γ に対するシミュレーションからこれらの物理量を求めることができるのである。

2体分布関数 $g(r)$ は統計力学で重要な役割をはたしており、様々な物理量を計算できる。ISIS コードでは、その中で配位数 N_C とアインシュタイン振動数 Ω_0 を計算する。 N_C は、ある粒子の最近接粒子数を表し、

$$N_C = 4\pi\rho \int_0^{r_m} g(r)r^2 dr, \quad (4.48)$$

と定義される。ここで、 r_m は $g(r)$ の第1極小点の現れる位置である。 N_C の値の大小により、粒子配置の粗密の程度を評価することができる。また、結晶における N_C の値は、格子の種類によって決まっている。一方、 Ω_0 は、

$$\Omega_0^2 = \frac{\rho}{3m} \int \nabla^2 u(r) g(r) dr, \quad (4.49)$$

によって定義される。 Ω_0 の物理的な意味は、ある1つの粒子に着目し、他のすべての粒子が平衡状態における平均位置に静止したと考えると得られるポテンシャル井戸で、その粒子が局所振動するときの振動数である。また、 Ω_0^2 は、速度相関関数 $Z(t)$ のテーラー展開の2次の係数として現れる。

物質中の多体相関の時空的様相を示す2つのスペクトル関数 $S(k, \omega)$ 、 $C_t(k, \omega)$ の中で、 $S(k, \omega)$ は中性子散乱実験から測定可能なものである。しかし、 $C_t(k, \omega)$ は、実験から測定できない量であり、MD シミュレーションが唯一の観測手段である。 $S(k, \omega)$ 、 $C_t(k, \omega)$ を計算できる波数の領域は、周期境界条件を課すため、次のものに限られる。

$$\mathbf{k} = k_0(n_x, n_y, n_z), \quad k_0 = \frac{2\pi}{L}. \quad (4.50)$$

ここで、 n_x, n_y, n_z は整数である。波数が大きくなるほど波数ベクトル \mathbf{k} の方向が増加する。ISIS コードでは、データ量の削減と統計精度の均一化のために、

$$\mathbf{k} = k_0(n, 0, 0), \quad \mathbf{k} = k_0(0, n, 0), \quad \mathbf{k} = k_0(0, 0, n), \quad (4.51)$$

のように各波数について3方向のみ考慮している。したがって、実現できる波数は、 k_0 の整数倍に限定される。 $S(k, \omega)$, $C_i(k, \omega)$ の計算では、有限の時系列データに対するフーリエ変換を行うため、ノイズの発生が避けられない。このノイズを取り除くため、次のガウス型フィルターを用いて観測値を平滑化する。

$$f(\omega) = \frac{2}{\omega_0} \sqrt{\frac{\log 2}{\pi}} \exp \left[- \left(\frac{2\omega}{\omega_0} \right)^2 \log 2 \right], \quad (4.52)$$

$$S(k, \omega) = \int_{-\infty}^{\infty} S^{\text{obs}}(k, \omega') f(\omega - \omega') d\omega'. \quad (4.53)$$

ここで、 ω_0 はフィルターの半値幅であり、 S^{obs} はノイズを含んだ状態の動的構造因子である。半値幅 ω_0 を大きくとるほどスペクトルは滑らかになるが、高次の周波数モーメントの保存が悪くなるので注意が必要である。

4.5 プログラムの構成

MDシミュレーションに必要なプログラムは、(1)シミュレーションの実行とマイクロなデータのサンプリング、(2)サンプリングデータの平均操作と解析、という2つの機能に分類される。ISISコードは(1)の機能を受け持ち、いくつかの解析プログラムが(2)の機能を受け持つ。Table 4.3と4.4に、ISISコードを構成するプログラムと解析プログラムをまとめ、Fig. 4.3に積分ステップのフローチャートを示す。また、Table 4.5にISISコードへの入力パラメータの形式および意味をまとめた。

ここで、ISISコードを使ったMDシミュレーションの進め方を簡単に述べる。シミュレーションは、次の3つの段階に分けられる。

- | | |
|-----------|--|
| 初期配置の生成 | 結晶配置から出発し、ポテンシャルエネルギーがある一定値のまわりを揺らぎ始めるまで(1000ステップ程度)シミュレーションする。(束縛法) |
| 温度調節と平衡化 | 目標の温度へ徐々に移行させる。目標の温度に達したら、体系が平衡状態に落ちつくまでシミュレーションする。(束縛法) |
| プロダクションラン | 温度調節を全くせず、マイクロなデータをサンプリングしながらシミュレーションする。(束縛法、マイクロカノニカルアンサンブル) |

どの段階を実行するかは、入力パラメータJOBによって指定する。なお、体系が平衡状態に落ちつくまでに、通常1000~5000ステップ程度のシミュレーションが必要である。

Table 4.1 Available physical properties of soft-sphere system from a MD simulation with the ISIS code. The specific heat in the microrcanonical ensemble is defined by Eq. (2.34).

性質あるいは関数	定義
ポテンシャルエネルギー	$u^* = \frac{\beta U}{N} = \frac{\beta}{N} \langle \sum_{i \neq j} u(r_{ij}) \rangle$
圧力	$P^* = \frac{P\beta}{\rho} = 1 + \frac{n\beta U}{3N}$
定積比熱 (束縛法)	$c_V^* = \frac{C_V}{k_B N} = \frac{3}{2} + \frac{1}{N} \left[\langle \sum_{i \neq j} \beta u(r_{ij}) ^2 \rangle - \langle \sum_{i \neq j} \beta u(r_{ij}) \rangle^2 \right]$
等温圧縮率	$\chi_T^* = \frac{\rho \chi_T}{\beta} = \frac{3}{n} \left[\left(1 + \frac{n}{3}\right) P^* - \frac{n}{3} c_V^* - 1 + \frac{n}{2} \right]^{-1}$
熱膨張係数	$\alpha_P^* = \alpha_P T = \chi_T^* \left(\frac{n}{3} c_V^* + 1 - \frac{n}{2} \right)$
比熱比	$\gamma = C_P / C_V = 1 + \frac{\alpha_P^{*2}}{c_V^* \chi_T^*}$
等温音速	$v_T^* = v_T \sqrt{m\beta} = (m\rho \chi_T)^{-1/2}$
断熱音速	$v_S^* = v_S \sqrt{m\beta} = \sqrt{\gamma} v_T^*$
2体分布関数	$g(r) = \frac{1}{N} \langle \sum_{i \neq j} \delta(r + \mathbf{r}_j - \mathbf{r}_i) \rangle$
平均2乗変位	$R_2^*(t) = \langle \mathbf{r}_i^*(t) - \mathbf{r}_i^*(0) ^2 \rangle$
自己拡散係数	$D^* = D l^{-2} \tau = \lim_{t \rightarrow \infty} \frac{\tau}{6t} R_2^*(t)$
速度相関関数	$Z(t) = \frac{m\beta}{3} \langle \mathbf{v}_i(t) \cdot \mathbf{v}_i(0) \rangle$
Z(t) のスペクトル	$Z^*(\omega) = \tau^{-1} Z(\omega) = \frac{1}{2\pi\tau} \int_{-\infty}^{\infty} Z(t) \exp[i\omega t] dt$
動的構造因子	$S^*(k, \omega) = \frac{S(k, \omega)}{\tau S(k)} = \frac{1}{2\pi\tau N S(k)} \lim_{T \rightarrow \infty} \frac{1}{T} \left \int_0^T \exp[i\omega t] \rho(\mathbf{k}, t) dt \right ^2$
流れ相関関数のスペクトル (直行要素)	$C_i^*(k, \omega) = \frac{m\beta C_i(k, \omega)}{\tau k^2} = \frac{m\beta}{4\pi\tau k^2 N} \sum_{\alpha=1}^3 \lim_{T \rightarrow \infty} \frac{1}{T} \left \int_0^T \exp[i\omega t] (\mathbf{k} \times \mathbf{j}(\mathbf{k}, t))_{\alpha} dt \right ^2$
注)	$\rho(\mathbf{k}, t) = \sum_{i=1}^N \exp[\mathbf{k} \cdot (\mathbf{r} - \mathbf{r}_i(t))], \mathbf{j}(\mathbf{k}, t) = \sum_{i=1}^N \mathbf{v}_i(t) \exp[\mathbf{k} \cdot (\mathbf{r} - \mathbf{r}_i(t))]$

Table 4.2 Available physical properties of binary soft-sphere system from a MD simulation with the ISIS code. The indices a and b denote the species of particle.

性質あるいは関数	定義
部分2体分布関数	$g_{aa}(r) = \frac{1}{N_a} \left\langle \sum_{i \neq j}^{N_a} \delta(\mathbf{r} + \mathbf{r}_{aj} - \mathbf{r}_{ai}) \right\rangle$ $g_{ab}(r) = \frac{1}{\sqrt{N_a N_b}} \left\langle \sum_i^{N_a} \sum_j^{N_b} \delta(\mathbf{r} + \mathbf{r}_{bj} - \mathbf{r}_{ai}) \right\rangle$
平均2乗変位	$R_{2a}^*(t) = \langle \mathbf{r}_{ai}^*(t) - \mathbf{r}_{ai}^*(0) ^2 \rangle$
自己拡散係数	$D_a^* = D_a l^{-2} \tau = \lim_{t \rightarrow \infty} \frac{\tau}{6t} R_{2a}^*(t)$
速度相関関数	$Z_a(t) = \frac{m\beta N}{3N_a} \langle \mathbf{v}_{ai}(t) \cdot \mathbf{v}_{ai}(0) \rangle$
$Z_a(t)$ のスペクトル	$Z^*(\omega) = \tau^{-1} Z(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} Z(t) \exp[i\omega t] dt$
動的構造因子	$S_{ab}^*(k, \omega) = \frac{S_{ab}(k, \omega)}{\tau S_{ab}(k)} = \frac{1}{2\pi\tau N S_{ab}(k)} \lim_{T \rightarrow \infty} \frac{1}{T} \left[\int_0^T \exp[i\omega t] \rho_a(\mathbf{k}, t) dt \right. \\ \left. \times \int_0^T \exp[-i\omega t] \rho_b(-\mathbf{k}, t) dt \right]$
流れ相関関数のスペクトル (直行要素)	$C_{iab}^*(k, \omega) = \frac{m\beta C_{iab}(k, \omega)}{\tau k^2} = \frac{m\beta}{4\pi\tau k^2 N} \sum_{\alpha=1}^3 \lim_{T \rightarrow \infty} \frac{1}{T} \left[\int_0^T \exp[i\omega t] (\mathbf{k} \times \mathbf{j}_a(\mathbf{k}, t))_{\alpha} dt \right. \\ \left. \times \int_0^T \exp[-i\omega t] (\mathbf{k} \times \mathbf{j}_b(\mathbf{k}, t))_{\alpha} dt \right]$
注)	$\rho_a(\mathbf{k}, t) = \sum_{i=1}^{N_a} \exp[\mathbf{k} \cdot (\mathbf{r} - \mathbf{r}_{ai}(t))], \mathbf{j}_a(\mathbf{k}, t) = \sum_{i=1}^{N_a} \mathbf{v}_{ai}(t) \exp[\mathbf{k} \cdot (\mathbf{r} - \mathbf{r}_{ai}(t))]$

Table 4.3 Programmes of the ISIS code. The number appears in the INPUT/OUTPUT column represents unit identifier. i and o denote input and output, respectively. b and f denote binary- and formatted-streams, respectively.

名前	機能	入出力
MAIN	主プログラム。 入出力の制御。 各サブプログラムの制御。	5(f/i) 入力パラメータ 6(f/i) モニター情報 1(b/i) ジョブ継続情報 2(b/o) ジョブ継続情報 3(b/o) RDF ヘッダ 4(b/o) MSD ヘッダ 7(b/o) VACF ヘッダ 10~39(b/o) DCOPE ヘッダ ($\rho(k, t)$) 30~69(b/o) DCOPE ヘッダ ($j(k, t)$)
RDFSMP	2体分布関数のサンプリング。	
RDF	2体分布関数の中間的な平均。	3(b/o)
MSD	平均2乗変位のサンプリング。	4(b/o)
VACF	速度相関関数のサンプリング。	7(b/o)
DCOPE	$\rho(k, t)$ および $j(k, t)$ のサンプリング。	10~39(b/o) $\rho(k, t)$ 40~69(b/o) $j(k, t)$
TABLE	簿記法に現れる粒子対の表の作成。	
CHECK	粒子対の表の更新条件の判定。	
FORCE	粒子間力とポテンシャルエネルギーの計算。	
STEPVE	Verlet のアルゴリズムによるステップ。	
STEPPC	Bernu のアルゴリズムによるステップ (束縛法)。	
STEPPM	Bernu のアルゴリズムによるステップ (マイクロカノニカルアンサンブル)。	
XSORT	予測座標・変位と修正座標・変位の入れ替え。	
INIT	初期座標および速度の設定。	
RANU	一様乱数の生成。	
RANG	正規分布乱数の生成。	
SCALE	結合定数 Γ の調節。	
COM	重心の計算。	
MOM	全運動量の計算。	
MELF	融解度の計算。	

Table 4.4 Programmes for data-analyses. The number appears in the INPUT/OUTPUT column represents unit identifier. i and o denote input and output, respectively. b and f denote binary- and formatted-streams, respectively.

名前	機能	入出力
AMSD	平均 2 乗変位の平均. 自己拡散係数の計算.	1(b/i), 6(f/o)
ARDF	2 体分布関数の最終的な平均.	1(b/i), 6(f/o)
AVACF	配位数, アインシュタイン振動数の計算. 速度相関関数の平均. スペクトルの計算.	1(b/i), 6(f/o)
SKW	$S(k, \omega)$ の計算.	1(b/i), 2(b/o)
CTKW	$C_t(k, \omega)$ の計算.	1(b/i), 2(b/o)
AVES	$S(k, \omega)$ の平滑化.	5(f/i), 10(b/i), 6(f/o)
AVE	$C_t(k, \omega)$ の平滑化.	5(f/i), 10(b/i), 6(f/o)

Table 4.5 Format and meaning of input parameters for the ISIS code.

パラメータの並び					
JOB, MSTEPA, MSTEPB, MCOR					
ISWP, ISWM, ISWR, ISWV, ISWO					
IAVP, IAVM, IAVR, IAVV					
GS, GE, DR, RSP, RPCT					
POW, M, MCC, DT, RHO, RC, X1, RS, RM					
NORM					
(AK(I), I=1, NORM)					
パラメータの意味					
JOB	0	結晶配置からスタート	IAVP		モニター情報の平均操作の間隔
	1	温度調節	IAVM		$R_2(t)$ の時間原点の間隔
	2	束縛法でプロダクションラン	IAVR		$g(r)$ の平均操作の間隔
	3	マイクロノニカルアンサンプルで プロダクションラン	IAVV		$Z(t)$ の時間原点の間隔
MSTEPA		温度調節するステップ数	GS		スタート時の結合定数
MSTEPB		温度調節無しのステップ数	GE		目標の結合定数
MCOR	> 0	ステップの修正回数	DR		$g(r)$ のメッシュ幅
ISWP		モニター情報の出力間隔	RSP		$r_M - r_c$
ISWM	$\neq 0$	$R_2(t)$ のサンプリング間隔	RPCT	≤ 0.25	簿記法の安全パラメータ α
	0	サンプリングなし	POW		ポテンシャルパラメータ n
ISWR	$\neq 0$	$g(r)$ のサンプリング間隔	M		$\sqrt[3]{N/4}$ あるいは $\sqrt[3]{N/2}$
	0	サンプリングなし	MCC	0	FCC 格子
ISWV	$\neq 0$	$Z(t)$ のサンプリング間隔		1	BCC 格子
	0	サンプリングなし	DT		時間メッシュ
ISWO	$\neq 0$	$\rho(k, t), j(k, t)$ のサンプリング間隔	RHO		数密度
	0	サンプリングなし	RC		カットオフ半径
NORM		波数の総数	X1		種類 1 の数濃度
AK(I)		k/k_0 の列	RS		σ_2/σ_1
			RM		m_2/m_1

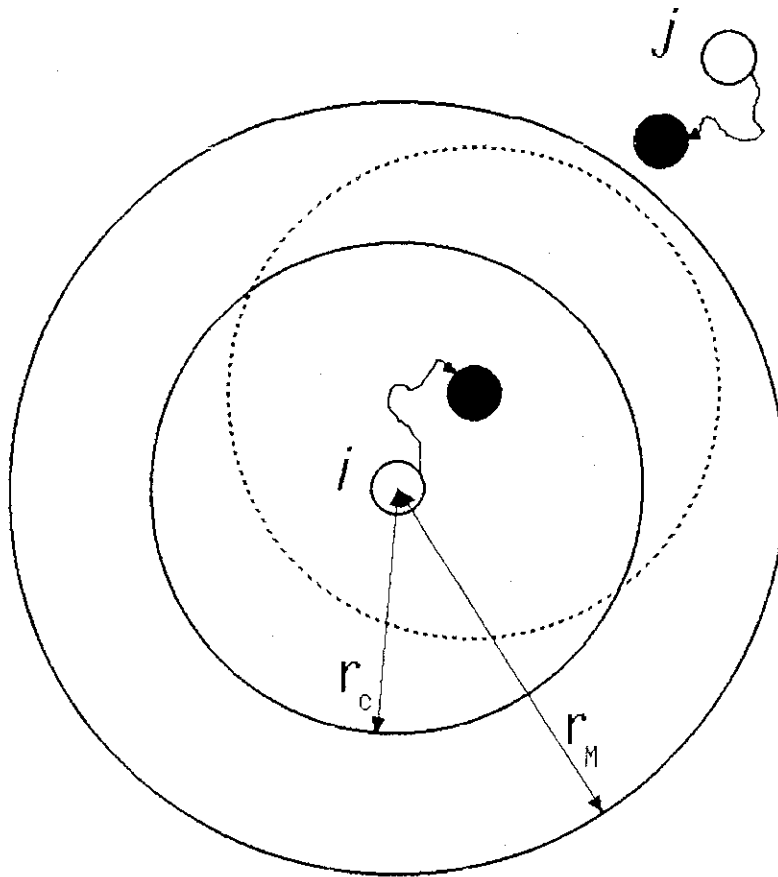


Fig. 4.1 Schematic diagram of the bookkeeping method for two dimensional system. The solid circles represent the cut-off "sphere" with radius r_c and the "sphere" with radius r_M to book particles. The dotted circle denotes the cut-off "sphere" after movement of the particle i .

```

(a)
C
C UPDATE NEIGHBOUR TABLE
C
  SUBROUTINE TABLE(X,IPAIR,NDX,N,MPAIR
+   ,SIDEL,SID2L,RM2,NUP,NCNT)
  REAL X(N)
  INTEGER IPAIR(N/2*MPAIR),NDX(0:N-1)
C
  IF (NUP.EQ.0) THEN
    RETURN
  END IF
  NCNT=NCNT+1
C
  NDX(0)=0
  K=0
  DO 10 I=1,N-1
    DO 20 J=I+1,N
      XIJ=X(I)-X(J)
      IF(XIJ.GE.SID2L) THEN
        XIJ=XIJ-SIDEL
      ELSE IF (XIJ.LT.-SID2L) THEN
        XIJ=XIJ+SIDEL
      END IF
      RIJ2=XIJ**2
      IF (RIJ2.LT.RM2) THEN
        K=K+1
        IPAIR(K)=J
      END IF
20    CONTINUE
      NDX(I)=K
10  CONTINUE
C
  RETURN
  END

(b)
C
C CALCULATION OF FORCE AND POTENTIAL-ENERGY
C
  SUBROUTINE FORCE(X,FX,IPAIR,NDX,MPAIR
+   ,EP,N,Z,POW,SIDEL,SID2L,RC2)
  REAL X(N),FX(N)
  INTEGER IPAIR(N/2*MPAIR),NDX(0:N-1)
+   ,POW
C
  NP1=-(POW+2)
  NP2=-POW
  PZ=POW*Z
C
  EP=0.
  DO 5 I=1,N
    FX(I)=0.
  5  CONTINUE
C
  DO 10 I=1,N-1
    *VOCL LOOP,NOVREC(FX)
    DO 20 JDX=NDX(I-1)+1,NDX(I)
      J=IPAIR(JDX)
      XIJ=X(I)-X(J)
      IF(XIJ.GE.SID2L) THEN
        XIJ=XIJ-SIDEL
      ELSE IF (XIJ.LT.-SID2L) THEN
        XIJ=XIJ+SIDEL
      END IF
      RIJ2=XIJ**2
      IF (RIJ2.LT.RC2) THEN
        RIJ2=SQRT(RIJ2)
        DUM=RIJ2**NP1
        EP=EP + RIJ2**NP2
        FX(I)=FX(I)+DUM*XIJ
        FX(J)=FX(J)-DUM*XIJ
      END IF
20    CONTINUE
10  CONTINUE
    EP=EP*Z/N
    DO 100 I=1,N
      FX(I)=FX(I)*PZ
100  CONTINUE
C
  RETURN
  END

```

Fig. 4.2 Computer programmes to calculate the molecular interaction by using the book-keeping method. These programmes are simplified to one dimensional system. (a) TABLE programme to book particles. (b) FORCE programme to calculate intermolecular force and potential energy.

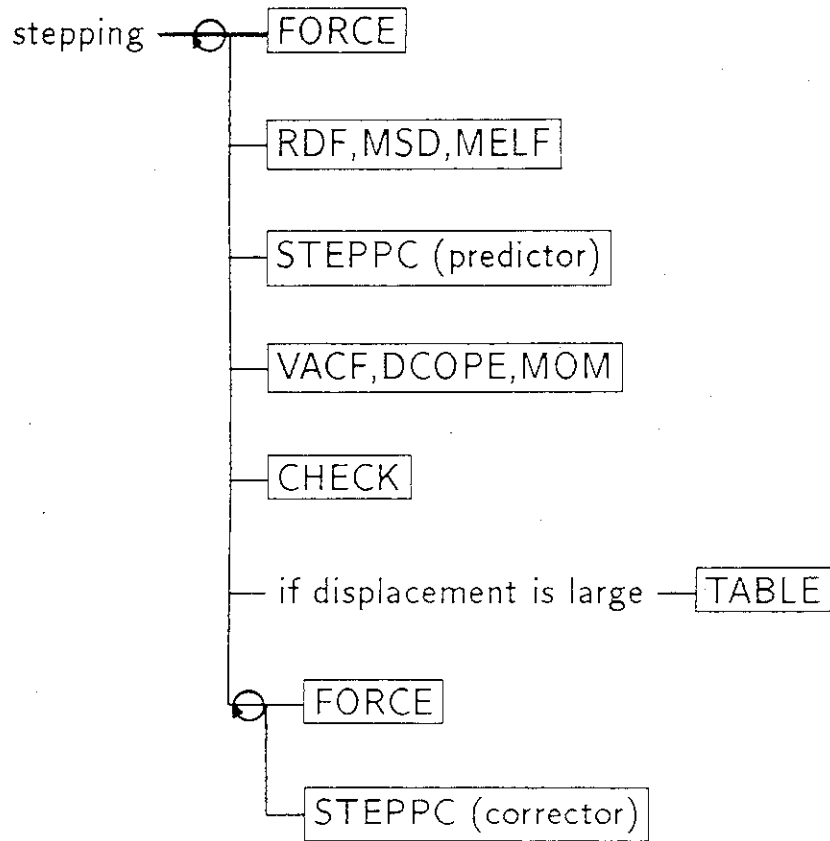


Fig. 4.3 Flow diagram of stepping of the discrete equation-of-motion in the ISIS code.

5. 計算時間の測定と評価

MD シミュレーションを実行する際に、もっとも計算時間のかかるところは、粒子間距離を求めて力を計算する部分である。粒子間力の計算時間を短縮することは、次の2つの観点から非常に重要である。

- (1) 粒子数を多くとることができる。したがって、基本セルの1辺が長くなり、相関距離の長い現象が観測可能となる。例えば、 $g(r)$ の遠距離での振る舞いや短波長領域の $S(k, \omega)$ を計算することができる。
- (2) 非常に多くのステップ数にわたるシミュレーションが可能となる。これによって、粒子数で平均をとることのできない物理量、例えば $S(k, \omega)$ を精度よく計算することができる。

計算時間短縮の鍵は、計算機の演算形式とプログラムに用いる計算アルゴリズムの2つである。ここでは、ISIS コードの粒子間相互作用計算プログラムおよびコード全体に対する計算時間の測定結果から、ベクトル計算機の利用、そして、簿記法の利用によって、どれだけ計算時間が短縮されるのかを考察する。

5.1 カーネルプログラムの場合

まず、ISIS コードを用いた MD シミュレーションにおいて、粒子間力の計算がどの程度計算時間を占めるのか示そう。Table 5.1 に、fcc 構造に配置した 500 個のソフト球粒子 ($n=12$) の融解現象のシミュレーション (1000 ステップ) を、FORTUNE で解析した結果をまとめる。粒子間力の計算を行うサブプログラム FORCE において、もっとも大きな計算コストを要しており、粒子対を登録するサブプログラム TABLE と合わせると、コスト全体の 98% を占めることがわかる。そこで、ISIS コードから3つのサブプログラム FORCE, TABLE, INIT, を抜き出して2つの時間計測用のプログラムを作成した。1つは簿記法を使わないように FORCE を改造した FORCEO を使ったもの (PGMa) であり、もう一方は簿記法を使うもの (PGMb) である。時間測定に用いたパラメータは Table 5.2 に示したものである。以下、簿記法を使わない場合と使った場合のそれぞれについて、FORCEO と FORCE に要する計算時間の測定結果をまとめる。

5.1.1 簿記法を使わない場合

PGMa の時間測定に使用した計算機は、ウィーン工科大学・情報処理センターの CDC860 と VP50EX, そして、原研・情報システムセンターの M780/20 と VP2600/10 である。使用した FORTRAN コンパイラは、NOS/VE FTN 1.7 (CDC860), FORTRAN77 V10L31 (FACOM, スカラー翻訳版), そして FORTRAN77/VP V10L30 (FACOM, ベクトル翻訳版) である。なお、後で述べる解析のために、 $r_c^* = 0.1$ に対する計測も併せて行った。サ

プログラム FORCEO に要した計算時間の測定結果を, Fig. 5.1, 5.2, および 5.3 にまとめた. また, ベクトル計算機におけるスカラー翻訳版とベクトル翻訳版との計算時間の比を, Fig. 5.4 に示す. ベクトル処理による処理速度の向上は, VP50EX について 5~8 倍であり, VP2600/10 について 10~20 倍である.

次に, 計算時間の粒子数 N とカットオフ半径 r_c に対する依存性について述べる. FORCEO の主ループは,

```

DO 10 I=1,N-1
  DO 20 J=I+1,N
    a  粒子間距離  $r_{ij}$  を計算し,
        カットオフ半径  $r_c$  と  $r_{ij}$  の大小を判定する
    b  もし,  $r_{ij} < r_c$  ならば,
         $i, j$  間の力を計算し, 配列に足し込む
  20  CONTINUE
10  CONTINUE

```

なる 2 重ループである (Fig. 4.2(b) 参照). したがって, スカラー処理における計算コストは, 粒子対の組み合わせの数から,

$$\begin{aligned} \text{cost} &= A_a \frac{N(N-1)}{2} + A_b \frac{N_c N}{2} \\ &\simeq B_a N^2 + B_b N_c N, \end{aligned} \quad (5.1)$$

と記述できる. ここで, 添え字 a, b のついた A は, プログラム中に現れる計算ブロックに対応した時定数である (ここで, $B = A/2$ である). N_c は, ある粒子を中心とした半径 r_c の球内にある平均の粒子数から 1 を減じた数である. fcc 格子では, $r_c^* = 0.1, 3, 5$ のそれぞれについて, $N_c = 0, 134, 530$ なる値をとる. また, 流動相 (気相, 液相) では,

$$N_c = \frac{4\pi r_c^3 \rho}{3} - 1, \quad (5.2)$$

で見積もることができる. 一方, ベクトル処理における計算コストは, 式 (5.1) を拡張した

$$\text{cost} \simeq B_a N^\alpha + B_b N_c^\beta N^\gamma, \quad (5.3)$$

なる表現を仮定する. Table 5.3 に, 計測データを式 (5.1) と (5.3) に適応して得られたパラメータをまとめた. VP2600/10 のベクトル翻訳版に対する β, γ の結果に, r_c^* つまり N_c のとり方でばらつきが生じている. VP50EX の場合にこのようなばらつきがみられないことから, これはベクトル演算機器の構成 (パイプラインのセット数) に起因するものと考えられる.

さて, 計算ブロック a と b のそれぞれに必要な計算コストは, ある粒子数 N_i で等しくなり, N_i を境にして計算時間の粒子数依存性が変化することになる. 式 (5.1) と (5.3) を使くと,

$$N_i = \begin{cases} \frac{B_b N_c}{B_a} & (\text{scalar}) \\ \left(\frac{B_b}{B_a}\right)^{\frac{1}{\alpha-\gamma}} N_c^{\frac{\beta}{\alpha-\gamma}} & (\text{vector}) \end{cases}, \quad (5.4)$$

が得られる。Table 5.4 に、式 (5.4) から求めた N_t の値を示す。スカラー処理では、比較的少ない粒子数で $\text{cost} \propto N$ から $\text{cost} \propto N^2$ の振る舞いへ移行することがわかる。一方、VP2600/10 のベクトル翻訳版では、他の計算機と比べて、格段に N_t が大きくなっている。この結果は、Figs. 5.2, 5.3 の大きな N の領域において、ベクトル翻訳版の VP2600/10 上での計算時間の曲線が、他のものと比べ小さな傾きであることとよく対応している。

5.1.2 簿記法を使った場合

PGMb の時間測定に使用した計算機は、原研・情報システムセンターの VP2600/10 である。サブプログラム FORCE および TABLE に要した計算時間の測定結果を、Fig. 5.5 および 5.6 にまとめた。また、スカラー翻訳版とベクトル翻訳版との計算時間の比を、Fig. 5.7 に示す。ベクトル処理による処理速度の向上は、TABLE について 40～50 倍であり、FORCE について 15～16 倍である。

PGMa と PGMb の測定結果を比較すると、サブプログラム FORCE に要する計算時間が、極めて短くかつほぼ粒子数に比例していることがわかる。ベクトル翻訳版の FORCE0 と FORCE の計算時間を比較すると、後者は $N = 4000$ で約 1/5、そして、 $N = 13500$ で約 1/10 に短縮されている。5.1.1 項と同様に、簿記法を使った場合の計算時間を、

$$\begin{aligned} \text{cost} &= A_t \frac{N(N-1)}{2} + A_f \frac{N_M N}{2} \\ &\simeq B_t N^2 + B_f N_M N, \end{aligned} \quad (5.5)$$

と仮定する。 A_t と A_f は、それぞれ TABLE と FORCE における時定数である ($B=A/2$)。 r_M は、半径 r_M の球内に含まれる平均の粒子数から 1 を減じた数である。fcc 格子では、 $r_m^* = 4$, 6 のそれぞれについて、 $N_M = 248$, 934 なる値をとる。式 (5.5) は、スカラー処理とベクトル処理の両方に適用することができる。Table 5.5 に、計測データを式 (5.5) に適応して得られたパラメータをまとめた。Table 5.3 と 5.5 の値を比較すると、 $B_t \simeq B_a$ (スカラー翻訳版及びベクトル翻訳版) と $B_f \simeq B_a + B_b$ (スカラー翻訳版) が成り立ち、これは PGMa と PGMb の計算内容を考えると妥当である。なお、ベクトル翻訳版に対する B_f の値に、 r_M^* つまり N_M のとり方ではらつきが生じている。これは、FORCE0 の場合と同様に、ベクトル演算機器の構成 (パイプラインのセット数) に起因するものと考えられる。

実際に簿記法を使って流動相の MD シミュレーションを行う場合、TABLE はほぼ一定のステップ間隔で実行される。Verlet の差分式のように FORCE を 1 ステップ 1 回実行する場合、1 ステップごとに粒子間力の計算に必要な計算コストは、

$$\text{cost} \simeq \frac{1}{n_{\text{update}}} B_t N^2 + B_f N_M N, \quad (5.6)$$

で与えられる。ここで、 n_{update} は TABLE を実行する平均のステップ間隔である。したがって、5.1.1 項と同様に、計算コストの粒子数依存性が $\text{cost} \propto N$ から $\text{cost} \propto N^2$ へ移行する粒子数 N_t が存在し、

$$N_t = \frac{n_{\text{update}} B_f N_M}{B_t}, \quad (5.7)$$

で与えられる。Table 5.5 に、 $n_{\text{update}} = 50$ として式 (5.7) から求めた N_t の値を示す。スカラ翻訳版とベクトル翻訳版のいずれも、PGMa の場合と比較して、 N_t が 1 桁大きく、簿記法の利用が多く粒子数の MD シミュレーションに極めて有効であることがわかる。なお、Bernu の差分式を使う場合、1 ステップあたり最低 2 回 FORCE を実行するので、 N_t は Table 5.5 に示した値の半分以下になる。

5.2 ISIS コード全体の場合

5.1.1 節で、粒子間力の計算時間が、ベクトル計算機の利用、そして簿記法の利用によって格段に短縮されることを示した。ここでは、fcc 格子に配置したソフト球モデルの融解現象の MD シミュレーションを例にとって、ISIS コード全体に必要な計算時間の測定結果を簡単に述べる。シミュレーションは、Table 5.2 に準じた入力パラメータを用い、ISIS コードのベクトル翻訳版を使って、FACOM VP2600/10 上で 1000 ステップ行った。また、比較のために簿記法を使わないシミュレーションも行った。計算時間の測定結果を、Table 5.6 に示す。この結果は、他の MD コードと比較しても、遜色のないものと思われる。例えば、Lennard-Jones 系で $r_c = 2.5\sigma$ とした場合、簿記法よりも効率がよいとされているリンク・セル (link cell) 法の計測結果は CRAY X-MP/416 (936MFLOPS) 上で 1 ステップあたり 0.299sec ($N = 6912$) であり、簿記法を用いたものでは CRAY Y-MP/432 (2GFLOPS) 上で 1 ステップあたり 0.327sec ($N = 6912$)、0.57sec ($N = 10976$) である。これらの計測結果は力の計算を 1 ステップあたり 1 回行うアルゴリズム (Fig. 4.3 の予測子 (predictor) の前の力の計算を省略するもの) なので、ISIS コードの結果を半分にした値、0.125sec ($N = 6912$)、0.204sec ($N = 10976$) が比較対象となる。いずれの場合も ISIS コードの方が計算時間が短いことがわかる。もちろん、比較対象の計算機の処理性能が異なるので、計算手法の良否を結論するには、同一計算機上での時間測定が必要である。

Table 5.1 Result of cost-analysis of the ISIS code using a FACOM M780 scalar processor.

名前	実行回数	コスト	割合 (%)
CHECK	1003	0.101192×10^8	0.1
TABLE	1005	0.695488×10^8	0.6
FORCE	2005	0.113910×10^{11}	97.8
STEPPC	2000	0.156166×10^9	1.3
STEPPM	0		
STEPVE	3	0.162757×10^6	0.0
XSORT	1000	0.704600×10^7	0.1
MAIN	1	0.793262×10^6	0.0
INIT	1	0.559070×10^5	0.0
RANG	1500	0.113699×10^6	0.0
RANU	2942	0.254240×10^6	0.0
SCALE	1	0.521000×10^3	0.0
COM	2	0.811600×10^4	0.0
MOM	105	0.798735×10^6	0.0
MELF	103	0.110764×10^8	0.1
MSD	1000	0.460000×10^5	0.0
VACF	1000	0.420000×10^5	0.0
RDF	1001	0.310310×10^5	0.0
RDFSMP	1000	0.380000×10^5	0.0
DCOPE	1000	0.390000×10^5	0.0
合計	16672	0.106788×10^{11}	100.0

Table 5.2 Parameters used for the test of kernel programmes of the ISIS code.

粒子数	$N = 4p^3$ ($p = 2 \sim 15$)
粒子配置	fcc
カットオフ半径	$r_c^* = 3, 5$
ポテンシャルパラメータ	$n = 12$
結合定数	$\Gamma = 0.8$
数密度	$\rho\sigma^3 = 0.8$
簿記法のパラメータ	$r_M^* - r_c^* = 1, \alpha = 0.25$

Table 5.3 Parameters for Eq. (5.1) obtained by fitting the measurements on various computers. S and V appeared in the mode field denote scalar-compilation and vector compilation, respectively.

計算機	モード	A_a (μs)	A_b (μs)	α	β	γ
CDC860	S	4.58	26.3	2	1	1
VP50EX	V	0.15	0.78	2	0.5	1.4
VP50EX	S	0.77	3.3	2	1	1
M780/20	S	0.49	1.38	2	1	1
VP2600/10	V	0.01	0.24	2	$\left\{ \begin{array}{l} 0.5 \\ 0.3 \end{array} \right.$	1.4 ($r_c^* = 3$)
VP2600/10	S	0.30	0.89	2		1.5 ($r_c^* = 5$)

Table 5.4 Estimated turning number N_t for various computers. S and V appeared in the mode field denote scalar-compilation and vector compilation, respectively.

計算機	モード	N_t ($r_c^* = 3$)	N_t ($r_c^* = 5$)
CDC860	S	770	3000
VP50EX	V	930	2900
VP50EX	S	930	2300
M780/20	S	380	1500
VP2600/10	V	12000	25000
VP2600/10	S	400	1600

Table 5.5 Parameters for Eq. (5.4) obtained by fitting the measurements on a FACOM VP2600/10 computer and estimated turning number N_t . S and V appeared in the mode field denote scalar-compilation and vector compilation, respectively.

モード	B_t (μs)	B_f (μs)	N_t	r_c
V	0.01	0.08	99000	3
S	0.25	1.25	62000	3
V	0.01	0.06	280000	5
S	0.29	1.20	190000	5

Table 5.6 Total computational time spent for MD simulation of soft-sphere fluid over 1000 time-steps by using ISIS code. (a) using the bookkeeping method. N_{update} denotes the total update count of neighbour table. (b) without the bookkeeping method.

(a)				
N	CPU (sec)	VU (sec)	VU/CPU (%)	N_{update}
500	17.63	15.76	89.4	19
2048	71.59	65.12	91.0	22
4000	142.10	129.79	91.3	22
6912	250.08	229.27	91.7	24
10976	406.46	373.80	92.0	25
13500	506.92	466.97	92.1	25
(b)				
N	CPU (sec)	VU (sec)	VU/CPU (%)	$\text{CPU}_{(b)}/\text{CPU}_{(a)}$
500	26.96	25.56	94.8	1.53
2048	368.74	358.76	97.3	5.15
4000	1360.99	1330.11	97.7	9.58

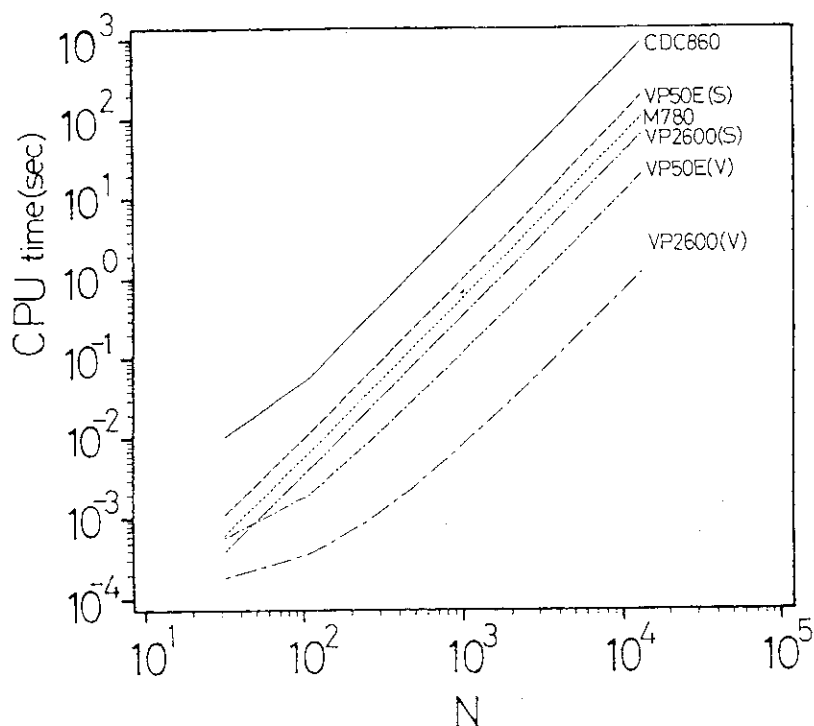


Fig. 5.1 Computational time spent for FORCE0 on various computers ($r_c^* = 0.1$).

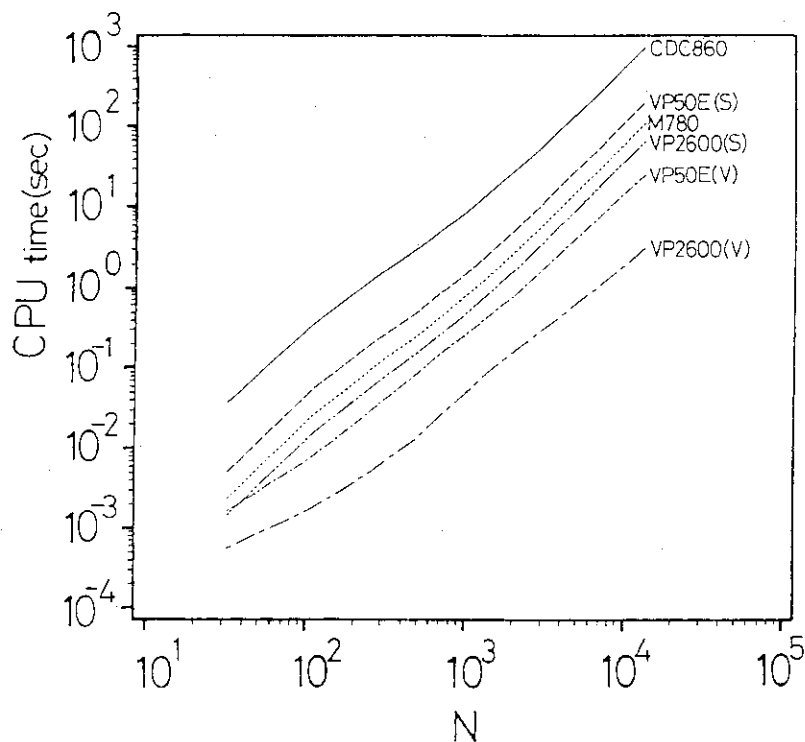


Fig. 5.2 Computational time spent for FORCE0 on various computers ($r_c^* = 3$).

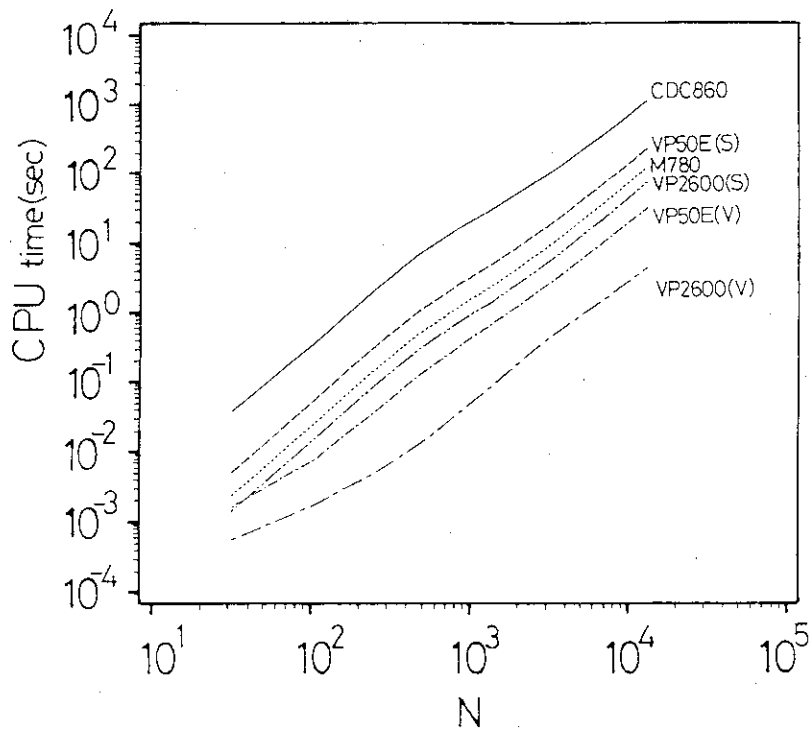


Fig. 5.3 Computational time spent for FORCE0 on various computers ($r_c^* = 5$).

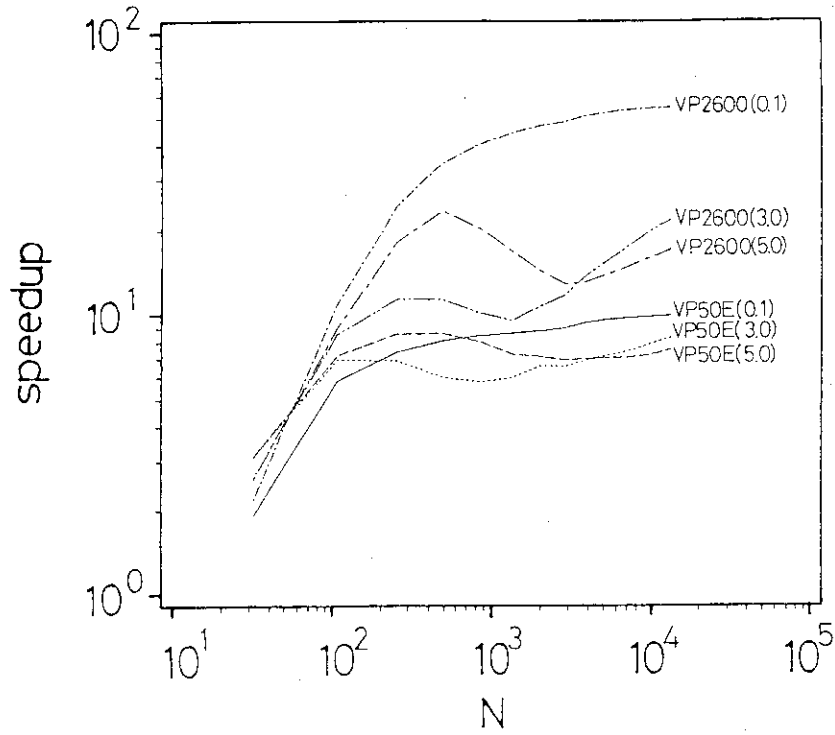


Fig. 5.4 Speedup of FORCE0 obtained by vector processing on VP50EX and VP2600/10.

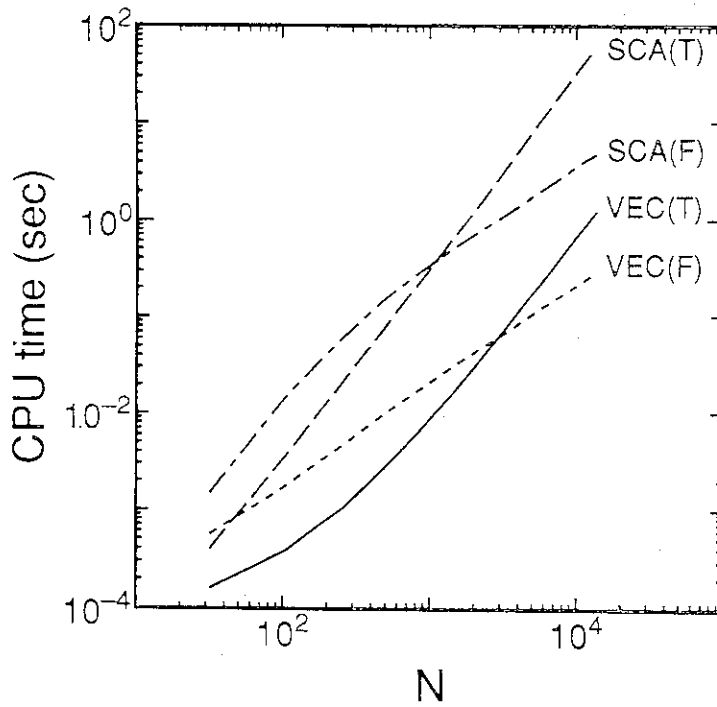


Fig. 5.5 Computational time spent for the TABLE and FORCE programmes on a FACOM VP2600/10 computer ($r_c^* = 3$).

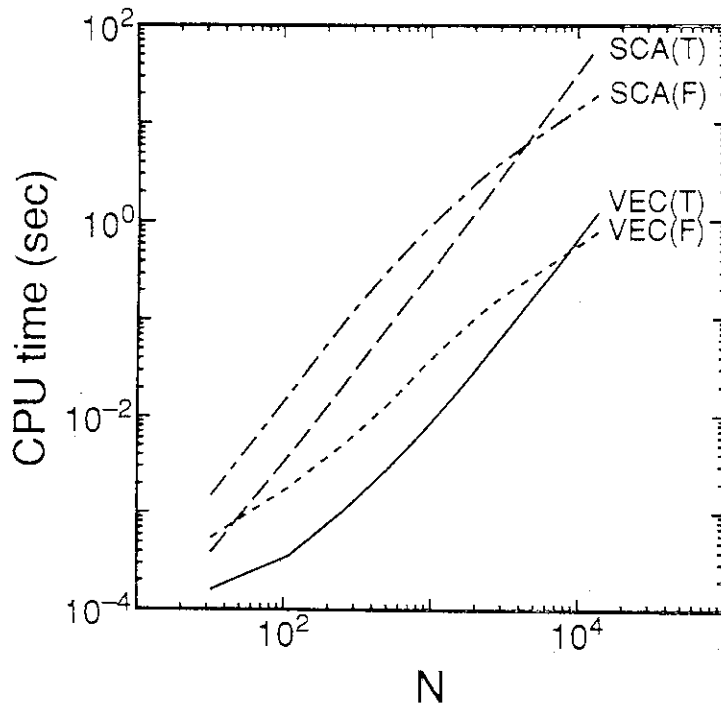


Fig. 5.6 Computational time spent for the TABLE and FORCE programmes on a FACOM VP2600/10 computer ($r_c^* = 5$).

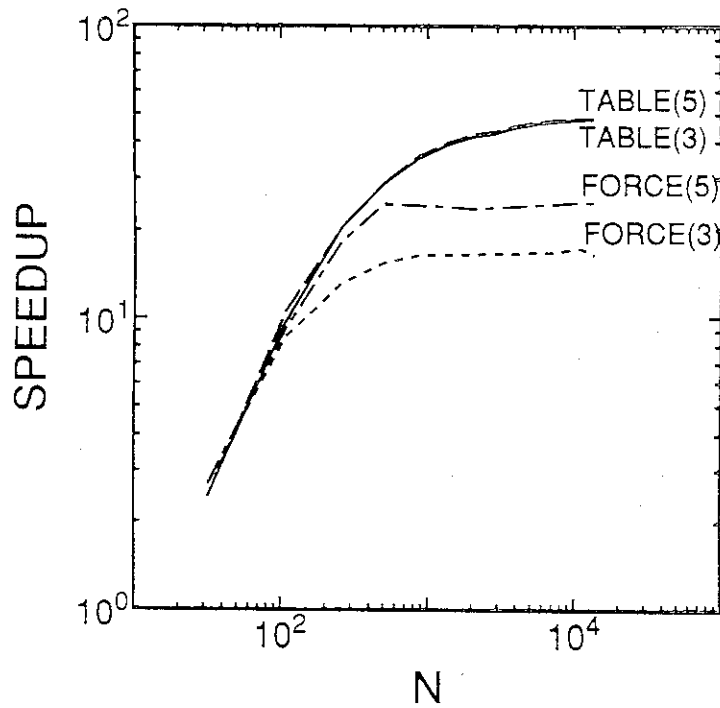


Fig. 5.7 Speedup of TABLE and FORCE obtained by vector processing on VP2600/10.

6. ま と め

本報告では、まず物性研究の新しい流れである MD 法によるコンピュータシミュレーションの概要をまとめ、温度一定のシミュレーションを可能にする拡張された運動方程式について説明した。さらに、ソフト球モデルの MD シミュレーションを対象として新規に開発した ISIS コードのあらましと各種計算機上で行った計算時間の測定結果について述べた。

MD シミュレーション手法は、古典力学の成立する範囲内において物質の構成粒子の振る舞いを厳密に（数値的に）再現する。古典力学では、(1) 粒子の質量、(2) 粒子に作用する力、(3) 境界条件、(4) 初期条件、の 4 つが基本的な入力パラメータである。したがって MD 法はこれらのパラメータを変更することによって、多種多様の物理現象に応用することができる。ISIS コードを例にとれば、粒子間ポテンシャルの入力を表形式の数値データとして与えるよう改造することによって、他の物質（例えば金属など）のシミュレーションを行うことができる。

ISIS コードで温度調節のために用いている束縛法は、温度のゆらぎが存在する乱数を使う方法や拡張系の方法と比較して、体系の温度が全くゆらぐことなく一定値をとるので物理的イメージを描くことが難しい。しかしながら、束縛法には経験的パラメータが全く含まれず、系全体の温度を一定に保つ以外にも適応できることが大きな利点である。例えば、体系に温度勾配を与えた MD シミュレーションへの利用が考えられる。これは、温度勾配のプロファイルを $G(r_i)$ として、

$$C(r_1, \dots, r_N, v_1, \dots, v_N) = \sum_i^N \left(\frac{mv_i^2}{2} - \frac{g}{2} k_B T G(r_i) \right) = 0, \quad (6.1)$$

なる束縛条件にしたがう束縛力を導くことで実現できる。

MD シミュレーションでは、5章で述べたとおり、粒子間力の計算に、非常に大きなコストがかかる。これは、粒子間力の計算が、粒子間の組み合わせの数 $N(N-1)/2$ に比例することに起因する。ISIS コードでは、簿記法とベクトル計算機の使用によって、計算時間を著しく短縮した。これによって、大規模なシミュレーション、非常に多くの粒子からなる体系のシミュレーションが可能になるとともに、シミュレーションの観測時間を長くとることができる。また、ISIS コードで採用した差分式は局所誤差が小さく、各種保存量を多くの積分ステップにわたって維持することに役だっている。

謝 辞

この研究を行うにあたり, Österreichische Forschungsfonds (P7618-TEC) の補助を受けた。筆者の留学中にお世話になったウィーン工科大学・理論物理学科の G. Kahl 助教授そしてウィーン工科大学・情報処理センターの W. Weisz 博士に厚くお礼申し上げます。そして、筆者の留学を支えてくださった情報システムセンターの方々に心から感謝します。

This work was supported by the Österreichische Forschungsfonds under project number P7618-TEC. The author would like to thank Assistant Professor G. Kahl and Dr. Weisz for valuable discussions and for kind hospitality during his stay in TU Wien. The author also thanks all colleagues in Computing and Information Systems Center for their help during his absence.

参考文献

- [1] B. J. Alder and T. E. Wainwright, *J. Chem. Phys.* **27**, 1208 (1957).
- [2] A. Rahman, *Phys. Rev.* **136**, A405 (1964).
- [3] L. Verlet, *Phys. Rev.* **159**, 98 (1967).
- [4] A. Paskin and A. Rahman, *Phys. Rev. Lett.* **16**, 300 (1966).
- [5] S. Nosé, *Prog. Theor. Phys. Supplement* **103**, 1 (1991).
- [6] W. G. Hoover, A. J. C. Ladd, B. Moran, *Phys. Rev. Lett.* **48**, 1818 (1982); W. G. Hoover, *Physica* **118A**, 111 (1983); W. G. Hoover, *Molecular Dynamics* (Springer, Berlin, 1986).
- [7] D. J. Evans and G. P. Morriss, *Comput. Phys. Rep.* **1**, 297 (1984).
- [8] S. Nosé, *Mol. Phys.* **52**, 255 (1984); *J. Chem. Phys.* **81**, 511 (1984).
- [9] H. C. Andersen, *J. Chem. Phys.* **72**, 2384 (1980).
- [10] R. H. Morf and E. P. Stoll, *Numerical Analysis*, 3083 (Birkhäuser, Basel, 1977).
- [11] Y. Hiwatari, E. P. Stoll, and T. Schneider, *J. Chem. Phys.* **68**, 3401 (1978).
- [12] 上田 顯, コンピュータシミュレーション(朝倉書店, 1990)。
- [13] 田中 實, 山本 良一 編, 計算物理学と計算化学(海文堂出版, 1988)。

謝 辞

この研究を行うにあたり、Österreichische Forschungsfonds (P7618-TEC) の補助を受けた。筆者の留学中にお世話になったウィーン工科大学・理論物理学科の G. Kahl 助教授そしてウィーン工科大学・情報処理センターの W. Weisz 博士に厚くお礼申し上げます。そして、筆者の留学を支えてくださった情報システムセンターの方々に心から感謝します。

This work was supported by the Österreichische Forschungsfonds under project number P7618-TEC. The author would like to thank Assistant Professor G. Kahl and Dr. Weisz for valuable discussions and for kind hospitality during his stay in TU Wien. The author also thanks all colleagues in Computing and Information Systems Center for their help during his absence.

参考文献

- [1] B. J. Alder and T. E. Wainwright, *J. Chem. Phys.* **27**, 1208 (1957).
- [2] A. Rahman, *Phys. Rev.* **136**, A405 (1964).
- [3] L. Verlet, *Phys. Rev.* **159**, 98 (1967).
- [4] A. Paskin and A. Rahman, *Phys. Rev. Lett.* **16**, 300 (1966).
- [5] S. Nosé, *Prog. Theor. Phys. Supplement* **103**, 1 (1991).
- [6] W. G. Hoover, A. J. C. Ladd, B. Moran, *Phys. Rev. Lett.* **48**, 1818 (1982); W. G. Hoover, *Physica* **118A**, 111 (1983); W. G. Hoover, *Molecular Dynamics* (Springer, Berlin, 1986).
- [7] D. J. Evans and G. P. Morriss, *Comput. Phys. Rep.* **1**, 297 (1984).
- [8] S. Nosé, *Mol. Phys.* **52**, 255 (1984); *J. Chem. Phys.* **81**, 511 (1984).
- [9] H. C. Andersen, *J. Chem. Phys.* **72**, 2384 (1980).
- [10] R. H. Morf and E. P. Stoll, *Numerical Analysis*, 3083 (Birkhäuser, Basel, 1977).
- [11] Y. Hiwatari, E. P. Stoll, and T. Schneider, *J. Chem. Phys.* **68**, 3401 (1978).
- [12] 上田 顯, コンピュータシミュレーション(朝倉書店, 1990).
- [13] 田中 實, 山本 良一 編, 計算物理学と計算化学(海文堂出版, 1988).

- [14] D. W. Heermann, *Computer Simulation Methods in Theoretical Physics* (Springer-Verlag, Berlin, 1986).
- [15] 樋渡 保秋, 固体物理 **17**, 141 (1982); **17**, 197 (1982); **17**, 317 (1982); **17**, 452 (1982); **18**, 144 (1983).
- [16] 戸田 盛和, 松田 博嗣, 樋渡 保秋, 和達 三樹, 液体の構造と性質 (岩波書店, 1976).
- [17] J. P. Hansen and I. R. McDonald, *Theory of Simple Liquids*, 2nd ed. (Academic Press, London, 1986).
- [18] Y. Hiwatari, J. Phys. Soc. Jpn. **47**, 733 (1979).
- [19] B. Bernu, Physica **122A**, 129 (1983).
- [20] S. Toxvaerd, J. Comput. Phys. **47**, 444 (1982).
- [21] Y. Hiwatari and H. Matsuda, Prog. Theor. Phys. **47**, 741 (1972).
- [22] H. Matsuda and Y. Hiwatari, in *Cooperative Phenomena*, edited by H. Haken and M. Wagner (Springer, Berlin, 1973), p. 250.
- [23] G. S. Grest, B. Dünweg, and K. Kremer, Comput. Phys. Commun. **55**, 269 (1989).
- [24] A. A. Chialvo and P. G. Debenedetti, Comput. Phys. Commun. **64**, 15 (1991).