直接シミュレーションモンテカルロ法の強制ベクトル計算†

宇佐美 勝†† 加藤征三††

気体，特に希薄気体の流れの解析に，直接シミュレーションモンテカルロ法（DSMC 法）が用いられている．この手法は，数万から数十万個の分子の位置座標と速度成分を計算機に記憶し，分子の物理空間内での移動と分子間衝突の計算を繰り返すことによって流れ場の状態を算出するものである．DSMC 法の最大の問題点は計算時間にあるので，今回，ベクトル計算によるシミュレーションの高速化を検討した．従来のプログラムの一部にはベクトル化の不適な DO ループを含む計算ルーチンが存在する．そこをスカラ計算で行うと，その部分の負担のために計算速度が十分には増加しない．一方，その DO ループを強制ベクトル化して計算すると，結果として数パーセントの演算欠落を生じるが，適切な後処理を行えば修正が可能でルーチンは正常に完了し，全体としてベクトル化の効果も得られる．このような強制ベクトル化とそれを補う後処理は，他の計算の高速化にも応用できる可能性がある．

1. まえがき

気体力学における実際的な問題は，ボルツマン方程式の強い非線形性のために，解析的手法によってはほとんど解くことができない．一方，Bird[1),2)] を中心にして発展してきた直接シミュレーションモンテカルロ法（DSMC 法）は，計算機技術の進歩とあいまって最近数多くの実用的な問題に適用されている[3)]．その適用範囲は，半導体製造技術等における希薄気体中での諸問題に限らず，代表長さが極めて小さい微小な流れ場の問題にも広がりつつある．DSMC 法の最大の問題点は計算時間にあり，流れ場の気体密度が増加するにつれて深刻となる．それを解消するために，ベクトル計算機による計算の高速化を検討した．DSMC 法における分子移動の計算ルーチンでは，分子一個一個が独立に移動するため並列処理が可能であるし，分子間衝突のルーチンでも空間を細分してセルごとに独立に計算することから，基本的には，ベクトル化による高速化が期待できる．しかし，セルごとに分子間衝突を計算するのに先立って，すべての分子を，分子の属するセルの順序で並べ直す処理が必要であり，これが現段階のベクトル計算機では根本的にベクトル化の不可能な計算である．そこで，この計算ルーチンに関して，特殊な算法を用いてベクトル化を図った．その算法とは，本来はベクトル化が不適な DO ループに対し，強制ベクトル化オプションを付加してベクトル演算を強制実行し，その結果として一部に演算欠落*の生じたものを，後続の計算処理により修正を行って正常化し，全体として速度増加を図ろうとするものである．著者らは以前から，スリットあるいは円筒孔を通過する気体流に DSMC 法を適用してきたが[4),5)]，今回，上述のような処理を含めて，従来のプログラムをベクトル化し計算速度の増加を達成することができた．

2. 直接シミュレーションモンテカルロ法（DSMC 法）の概要

DSMC 法では，対象とする物理空間の大きさ，あるいは，気体密度に応じて数万から数十万個の分子を同時に取り扱う．まず分子に初期座標と初期速度成分とを与え，対象となる物理空間に配置する．その後，微少時間ステップ Δt ごとに，分子の移動計算と分子間衝突の計算を分離し，それらの計算を交互に繰り返すことによって流れ場の状態を計算していく．この手法の主たる部分は以下の四つの計算ルーチンであり，これらの繰り返しによりこのシミュレーションの大部分が実行される（**図1**の流れ図参照）．

（1）個々の分子の物理空間上での移動ルーチン（境界での分子の反射あるいは流入流出を含む）

（2）分子が，どのセル（空間を分割する微小体積要素）に属するかを算出するルーチン

（3）cross-reference テーブル上での分子の並べ替

† Forced Vectorization in the Direct Simulation Monte-Carlo Method by MASARU USAMI and SEIZO KATO (Department of Mechanical Engineering, Faculty of Engineering, Mie University).

†† 三重大学工学部機械工学科

* ベクトル要素間に依存関係があったとき，それをベクトル化すると，プログラムの意図に反する結果の生じる可能性がある．「演算欠落」という言葉は，このように「本来ベクトル化してはならない DO ループの計算」を「ベクトル計算」してしまったために生じた「一部計算の誤り」という意味で用いている．

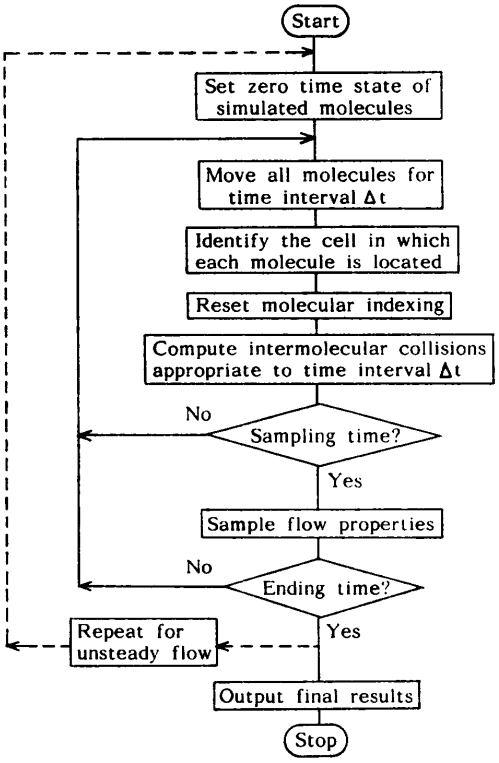

図 1 直接シミュレーションモンテカルロ法に関する一般的な流れ図
Fig. 1 Flow chart for application of the direct simulation Monte-Carlo method.

えルーチン（分子間衝突の計算に先立って，すべての分子をその分子の属するセルの順序で並べる処理が必要であり，そのための計算ルーチン．cross-reference テーブルは，テーブル上の，あるセルに属する任意の番号「座席番号」から，分子番号を参照するための表である．）

（4）セル単位での分子間衝突の計算ルーチン

これら四つの計算ルーチンのうち，「移動ルーチン」の高速化には，まず，すべての分子を Δt 時間だけ境界に関係なく直線移動させ，その後，単純な直線運動でない分子だけを「データ圧縮」により集め（ここまでの計算はベクトル化される），集められた少数の分子についてだけ，スカラ計算によって複雑な移動処理および境界との干渉計算を実行するという算法を用いる．また，「分子の属するセルを決定するルーチン」においては，実際には複雑なセル形状でも，セルを決定する際にはあたかも等間隔セルとして計算できる「座標変換」[6] の手法を利用し，ベクトル化による高速化を行う．しかし，「cross-reference テーブル上の分子の並べ替えルーチン」については，通常の方法ではベクトル化はできない．第3章では，このルーチンに関するベクトル化について詳細に論じる．また第4章では，「分子間衝突ルーチン」に関するベクトル化手法について簡単に説明する．

なお，本論文では，計算機に HITAC-S 810/10 を用いたので，以下の記述はその計算機におけるコンパイラ（FORT 77/HAP）の特性を基本としている．一般的なベクトル計算機の性質も考慮したが，必ずしもそのままあてはまらない記述もある．また，第3章以降の文中に出てくる大文字の記号は，FORTRAN プログラムの変数を表したものであり，第3章に掲げた図のプログラムの変数記号と同じ意味を持つ．主な FORTRAN 変数を**表 1** に示した．

表 1 FORTRAN 変数の一覧表
Table 1 Variables for the FORTRAN program.

ICHE	cross-reference テーブル上で分子が重複する位置を占めた場合にこの値が1となる
K	DO ループの制御変数（cross-reference テーブルの座席番号を変化させる）
L	作業変数
M	DO ループの制御変数（分子番号を変化させる）
N	DO ループの制御変数（セル番号を変化させる）
NC	セル数
NM	総分子数
NM2	cross-reference テーブルの座席番号の最大値
ICO (N)	cross-reference テーブル上の，セルNの最後尾座席番号
IC1 (N)	セルNのセル内分子数（各セルの分子数頻度表）
IC2 (N)	cross-reference テーブル上の，セルNの先頭座席から1を引いた値(累積頻度表的な意味をもつ)
IP (M)	分子Mの属するセル番号
IW ( )	作業変数
KK ( )	作業変数
LCR (K)	cross-reference テーブル
NOM (M)	添字Mと同じ値をもつ配列(NOM(M)=M, ベクトル化された DO ループの高速化のために用いる)

3. cross-reference テーブル上の分子の並べ替え

3.1 従来の計算法

分子間衝突は同一のセル内に存在する分子の間で行われる．同一セルに存在する任意の二つの分子ペアを選択するには，cross-reference テーブルを利用する．**図 2** に示すように，cross-reference テーブル LCR (K) の添字Kは，分子がテーブル上で位置を占める座席の番号であり，そのKに対し，LCR (K) にはその座席に位置する分子の番号が記憶される．すなわち空間内の各セルは，分子移動の計算が終了した直後にセル

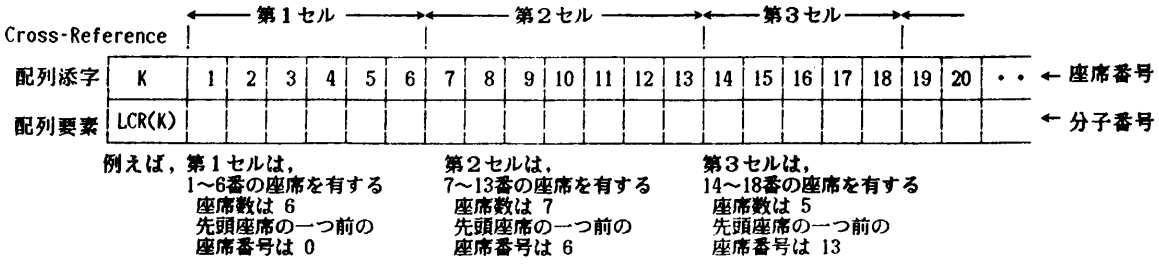

図 2 Cross-Reference テーブル上の各セルの座席番号と分子番号との関係
（分子の速度成分等の情報は別の配列に記憶されている）

Fig. 2 Relation between molecule numbers and seat numbers for each cell in the cross-referencing table. Informations of velocity components and position coordinates on each molecule are stored in the other arrays.

内分子数だけの連続した番号列（座席番号）を順にあてがわれ，それらの座席番号Kに対する cross-reference 配列要素 LCR (K) に，そのセル内に存在する分子の分子番号を記憶する．衝突を行う二つの分子を選択するためには，そのセルに属する座席番号列から二つの番号を選び出せばよく，得られた座席番号から分子番号を参照し，個々の分子情報を記憶している配列変数からその分子番号に対応する速度成分等の情報を得ることができる．

図 3 は，「cross-reference テーブル上の分子の並べ替え」に関する従来の計算ルーチンを示したもので，これはビンソート（バケットソート）と呼ばれるソーティング法の一種である．図中，IP (M) は，分子Mの属するセル名を表す変数である．まず DO_4 のループで，すべてのセルのセル内分子数 IC 1 (N) をゼロにした後，全分子にわたる DO_5 のループによって各分子の属するセル名を調べ，セル内分子数 IC 1 (N) をカウントしていく．次の DO_6 のループでは，得られたセル内分子数を順に累積していくことにより，cross-reference テーブル上の各セルの「先頭座席 (starting address) から 1 を引いた値」IC 2 (N) を順に計算する．この際，次の操作のために，すでに累積計算に用いられたセル内分子数 IC 1 (N) は再びゼロにされる．最後に，DO_7 のループでもう一度，各々の分子についてそれの属するセルを調べ，IC 1 (N) をカウントしながら，それにそのセルの「先頭座席 − 1」すなわち IC 2 (N) を加えることによって分子が入るべき座席番号Kを算出していく．最終的に，一つのセルに割り当てられた一連の座席番号列には，そのセルに属する分子が分子番号の小さい順に並ぶことになる．

```
c---------Molecular Indexing (1)-----------
c  LCR( ) is the cross-referencing array.
c  NC is the total number of cells.
c  NM is the total number of molecules.
c  IC1(N) is number of molecules in cell N.
c  IC2(N) is "Starting Address - 1" for
c  cell N in LCR( ).
c  IP(M) is the cell number in which
c  molecule M lies.
c
      DO 4 N=1,NC
    4 IC1(N)=0
c
      DO 5 M=1,NM
      N=IP(M)
    5 IC1(N)=IC1(N)+1
c
      L=0
      DO 6 N=1,NC
      IC2(N)=L
      L=L+IC1(N)
    6 IC1(N)=0
c
      DO 7 M=1,NM
      N=IP(M)
      IC1(N)=IC1(N)+1
      K=IC1(N)+IC2(N)
    7 LCR(K)=M
c-----------------------------------------
```

図 3 従来用いられている cross-reference テーブル上の分子並べ替えルーチン

Fig. 3 Conventional routine for molecular indexing in the cross-referencing array.

ところで，この「分子の並べ替え」の計算は，現段階のベクトル計算機では，根本的にベクトル化が不可能な計算である．すなわち，DO_5, DO_6, DO_7 の三つのループは通常いずれもベクトル化できない（DO_6 については，ベクトル化できるコンパイラもある）．特に，DO_5 と DO_7 のループは総分子数 NM 回の繰り返しであるので，高速化のための処理が望まれる．セル内分子数の変動が少ない特殊な条件の下では，DO_5 のループを省略する算法も考えられるが，以下では，そのような特殊

な条件に限定されない，強制ベクトル化を利用した高速化のための算法を論じる．

```
c--------Molecular Indexing (2)--------
c   NOM(N) is set equal to N.
c   IW( ) and KK( ) are working arrays.
c
      DO 4 N=1,NC
    4 IC1(N)=0
c
*VOPTION VEC
      DO 5 M=1,NM
      N=IP(M)
    5 IC1(N)=IC1(N)+1

      DO 15 N=1,NC
      IF(IC1(N).LE.10) THEN
      IC1(N)=20
      ELSE
      IC1(N)=IC1(N)*2
      ENDIF
   15 CONTINUE
c
  100 L=0
      DO 6 N=1,NC
      IC2(N)=L
      L=L+IC1(N)
    6 IC1(N)=0
c
      NM2=L
      DO 26 K=1,NM2
   26 LCR(K)=0
c
*VOPTION VEC
      DO 27 M=1,NM
      IW(M)=1
      N=IP(M)
      IC1(N)=IC1(N)+1
      K=IC1(N)+IC2(N)
   27 LCR(K)=NOM(M)
c
      DO 28 K=1,NM2
      M=LCR(K)
      IF(M.EQ.0) GO TO 28
      IW(M)=0
   28 CONTINUE
c
      L=0
      DO 29 M=1,NM
      IF(IW(M).EQ.0) GO TO 29
      L=L+1
      KK(L)=M
   29 CONTINUE
c
      DO 30 N99=1,L
      M=KK(N99)
      N=IP(M)
      IC1(N)=IC1(N)+1
      K=IC1(N)+IC2(N)
   30 LCR(K)=M
c---------------------------------------
```

図 4 強制ベクトル化を用いた分子並べ替えの計算ルーチン
Fig. 4 Routine of the molecular indexing using the forced vectorization with the appropriate post-processing for the unvectorizable DO-loops.

3.2 強制ベクトル化を用いた高速化

3.2.1 DO_5 に関する強制ベクトル化

図3において「DO_5 と DO_7」のループがベクトル化できないのは，このループの中に，セル内分子数をカウントしていく計算「IC 1 (N)＝IC 1 (N)＋1」があるためであるが，いま，「DO_5」のループに強制ベクトル化オプションを付加し，強制的にベクトル計算を行うことを考えてみる（**図 4** の DO-5）．

分子数を 50000，セル数を 2500 として，各セルが 20 個の分子を含むとき，分子番号をその分子の属するセルの順に与えた場合（この場合，小さなセル番号のセルに含まれる分子ほど，小さな分子番号を持つ．ただし，このような分子の番号付けの方法は DSMC 法において本質的なものではない）には，DO_5 の計算に要する時間はスカラ計算の約 0.63 倍となるが，強制ベクトル化に伴い約 95% もの演算欠落を生じる．その演算欠落とは，本来，順次的にスカラ計算すれば，IC 1 (N) は 1 個ずつカウントされていくものが，パイプライン上で並列計算を行うために，同じセルに属する二つ以上の分子が，同じ IC 1 (N) の値を用いる可能性が生じ，それゆえに発生するものである．すなわち，重なりの生じた場合には，そのうちの一つだけしかセル内分子数としてカウントされない．しかし，分子の番号付けはどのように行ってもよいので，分子番号をセルの順に与えた後に，二つの分子をランダムに選んで，それらの分子番号を入れ換えてみると，入れ換えた分子ペアの個数によって DO_5 に要する計算時間と演算欠落の数（総分子数の減少数）は変化してくる．**図 5** は，その様子を示したものであるが，全分子の分子番号を所属セルに無関係にランダムに与えた場合には，計算時間はスカラ計算の約 0.14 倍となり，演算欠落は総分子数の 5% 以下になる（計算時間は，状況によりある程度変動する）．DO_5 のルーチンの目的は，cross-reference テーブル上で各セルに割り当てる座席数を決めることにあるが，演算欠落の割合が小さければ，近似的に計算したセル内分子数の二倍の座席数を確保すれば，このルーチンの目的は達成できる（ただし，必要な配列要素数は倍増する）．あるいは，別の分子番号付けの方法として，分子番号をセルの順に与えた後に，いくつかの分子をランダムに選び，それらの分子の属するセル番号をランダムに与え直すという操作を行っても，やはり，与え直した分子の数が増加するほど演算欠落発生率は減少し，図5と同様な結果が得られる（この場合，

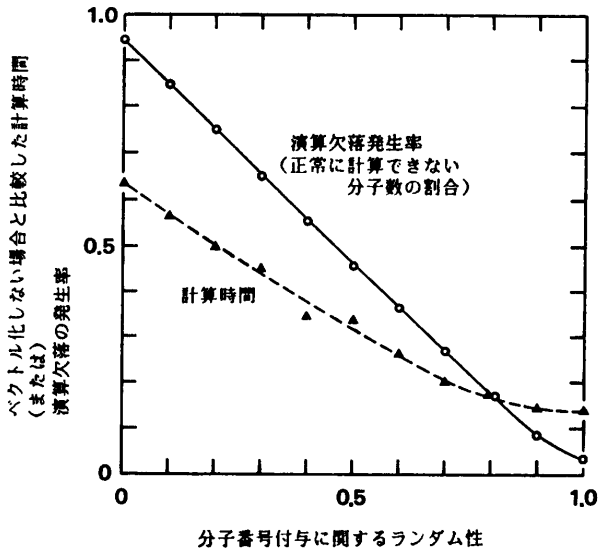

図 5 分子番号の付け方の違いによって生じる，強制ベクトル化された DO_5 のループに要する計算時間（ベクトル化しない場合との比）と演算欠落の発生率

Fig. 5 Ratio of the computer time for 'DO_5' with forced vectorization to that without vectorization and the rate of error on the count of number of molecules vary with how to number the molecules.

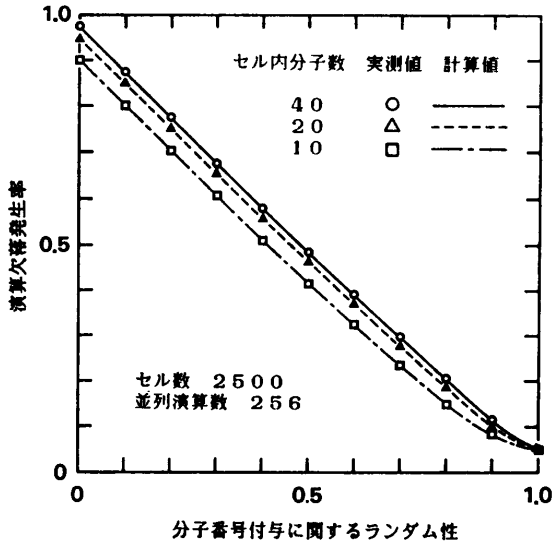

図 6 演算欠落発生率に及ぼすセル内分子数の影響

Fig. 6 Influence of the number of molecules in a cell on the rate of calculation error.

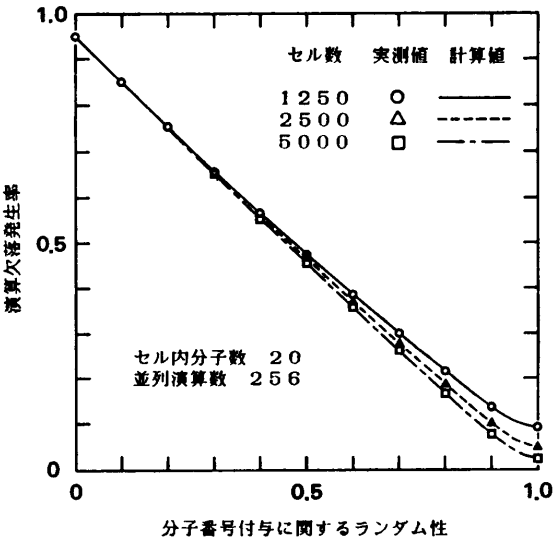

図 7 演算欠落発生率に及ぼすセル数の影響

Fig. 7 Influence of the number of cells on the rate of calculation error.

セル内分子数は各セル同数には保たれないが，この方が，実際の DSMC 法の計算には近い状態といえる）．

ベクトル計算機で生じるこのような分子番号付けのランダムの程度と演算欠落発生率の変化の様子は，次の想定の下にスカラ計算機上で近似的にシミュレーションできる．〈想定：ベクトル計算機において並列実行される演算数を一定とし，分子番号の順にその並列演算数だけの演算が同時に進行すると考える．また，同時進行の演算の中に，同一セルに含まれる分子があった場合，その中の一個の分子以外はすべて演算欠落になると考える〉．このような想定の下で行ったスカラ計算によるシミュレーションと，実際にベクトル計算機（本論文で使用した HITAC-S 810/10 においては 256 個の演算が並列的に実行される）で実測された演算欠落発生率の比較が**図 6, 7** に描かれている．図 6, 7 共に，計算値と実測値は良く一致している．図 6 は，セル内分子数をパラメータにして示してあるが，セル内分子数が多くなるほど演算欠落発生率は増加する．ただし，分子番号付けのランダム性が 100% になるとセル内分子数の違いによる差は見られなくなる．また図 7 は，セル数をパラメータにとった場合であるが，分子番号付けのランダム性が 100% に近づくにつれて，セル数が少なくなるほど演算欠落発生率は大きくなる傾向にある．さらに，演算欠落発生率のシミュレーションでは並列実行される演算数を変化させることもできる．**図 8** はその様子を示したものであるが，並列演算数が増加するほど演算欠落発生率は増加する．

一方，分子番号付けが完全に（100%）ランダムに行われた場合の演算欠落発生率は，近似的に計算式を導くことができる．いま，並列実行される演算数を r，平均のセル内分子数を m，セル数を n とする（ただ

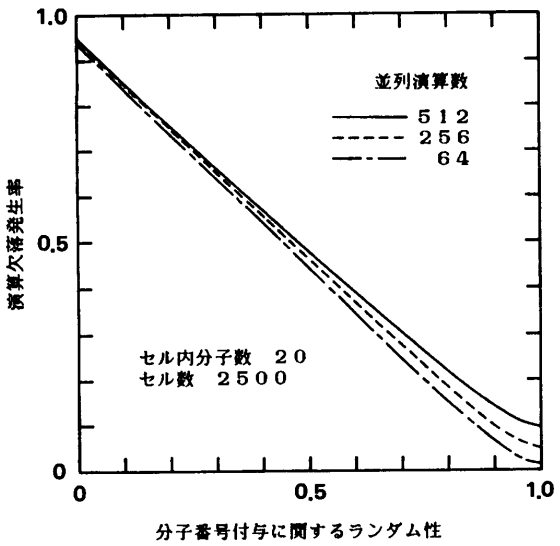

図 8 演算欠落発生率に及ぼす並列演算数の影響
Fig. 8 Influence of the number of parallel processings on the rate of calculation error.

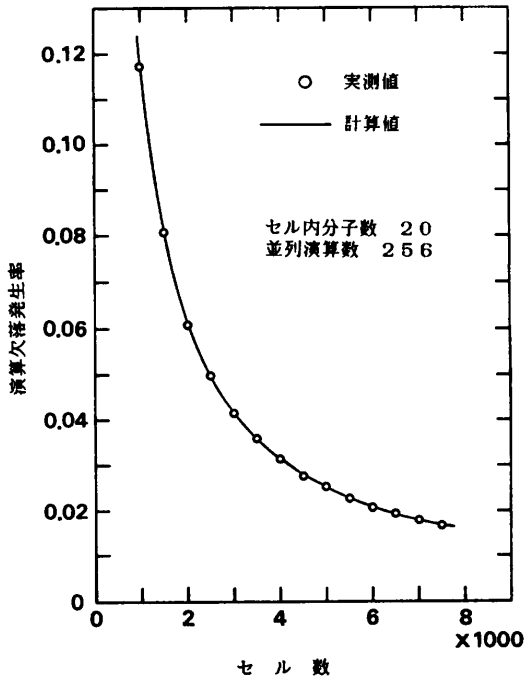

図 9 演算欠落発生率に及ぼすセル数の影響
(分子番号がセルに無関係に全くランダムに与えられた場合)
Fig. 9 Influence of the number of cells on the rate of calculation error. Molecular number is independent of the cell where the molecule lies.

し，簡単のために，全分子数 mn は，r の倍数であるとし，r の数だけの並列演算が mn/r 回繰り返されるとする)．一回の r 並列演算において，ある一つのセルに所属する分子が k 個その演算の中に含まれる確率は，二項分布

$$ {}_rC_k\left(\frac{1}{n}\right)^k\left(\frac{n-1}{n}\right)^{r-k},\quad 0\leqq k\leqq r $$

に従うと考えられる．いま，$k\geqq 2$ のとき，$k-1$ 回の演算欠落が生じるとすれば，すべてのセルについての演算欠落の期待値は次のようになる．

$$ n\left\{\sum_{k=2}^{r}(k-1){}_rC_k\left(\frac{1}{n}\right)^k\left(\frac{n-1}{n}\right)^{r-k}\right\} $$

このような並列演算が mn/r 回繰り返され，それぞれの場合において分子の所属するセル番号は全くランダムに与えられているとすれば，mn/r 回の各事象は独立と考えられるから，全体としての演算欠落発生率 p は式(1)となる．

$$ p=\frac{n}{r}\left\{\sum_{k=2}^{r}(k-1){}_rC_k\left(\frac{1}{n}\right)^k\left(\frac{n-1}{n}\right)^{r-k}\right\} \qquad (1) $$

図 9 は，式(1)により得られた演算欠落発生率 p と，実際にベクトル計算機において生じた欠落発生率を，セル内分子数を 20 に固定し，セル数を 1000～7500 で変化させて比較したものである．両者が良い一致を示していることから，近似的に導いた式(1)が有効であることがわかる．なお，セル数 n は並列演算数 r に比べて十分大きい ($n>>r>>1$) ことが一般的であるが，その場合には式(1)を簡略化して式(2)を得る．

$$ p=\frac{r}{2n} \qquad (2) $$

すなわち，演算欠落発生率 p は，セル数 n に反比例し並列演算数 r に比例する．一方，セル内分子数 m には依存しない．

実際の DSMC 法のシミュレーションでは，最初に，全分子に分子番号を（セルに関係なく）ランダムに与え，計算途中で境界から流入する分子の分子番号も注意して与えれば，計算中の分子はセルに関係なくほぼランダムな分子番号を持つと考えられる．そこで図 4 に示すように，DO_5 に強制ベクトル化オプションを付加して計算した後，得られたセル内分子数の二倍（図 4 の DO_15, DO_6 参照；演算欠落の数を考慮して二倍を与えているが，セル内分子数が 10 以下のときはさらに余裕をみて 20 を与えている）の値を用いて「先頭座席－1」IC 2 (N) を算出すればこのルーチンに関して高速化が達成できる．なお，最初に全分子に分子番号をランダムに与える方法としては，以下に示すような二つの方法がある．

① セル内分子数が各セル同数という初期条件のある場合；1から総分子数までの整数を一周期内にすべて発生する整数一様乱数を用い，乱数をセル内分子数で割った商（整数部）に1を加えた値を，その乱数が発生された順番と等しい分子番号の分子の所属するセル番号とする．

② セル内分子数が同数である必要がない場合；0〜1まで分布する一様乱数を用い，乱数に総分子数を掛け，平均セル内分子数で割った商（整数部）に1を加えた値を，その乱数が発生された順番と等しい分子番号の分子のセル番号とする．

3.2.2 DO_7 に関する強制ベクトル化

次に，DO_7（図3）に関するベクトル化を考える．上述のように，このループにも強制ベクトル化オプションを付加すると（図4の DO_27），やはり，同じセルに属する二つ以上の分子が同じ IC 1 (N) の値を用いる可能性が生じる．その場合，それらの分子のうち，最後に計算を終えた分子がその IC 1 (N) に対応する cross-reference 配列 LCR (IC 1 (N)+IC 2 (N)) 上に名前を登録され，それより以前に同じ配列要素に記録された他の分子の名前は消去される．すなわち，重なり合った分子の一つだけしか生き残らないために，セル内分子数は，前述と同様，消去された分子の数だけ少ないものとなる．そこで，カウントされずに消去された分子だけを「データ圧縮」により収集し，それらの分子についてだけ，スカラ計算によって，セル内分子数をカウントしながら cross-reference 配列に分子番号を登録していけば，分子の並べ替えは正常に完了し，全体としてベクトル化による速度増加の効果も現れる．図4における DO_26〜DO_30 は上記の計算を実際のプログラムとして表現したものである．まず，DO_26 において cross-reference テーブル LCR (K) をすべてゼロにした後，強制ベクトル化された DO_27 のループにおいて LCR (K) 上に分子番号を記録していく．ただし，重複により消去された分子は LCR (K) 上に分子番号が記録されない．なお，DO_27 において，分子番号Mの代わりに配列 NOM (M)（NOM (M) にはMが入っている）を用いたのは速度増加のためであり，また，後の処理のためにすべての分子に対応する作業用配列要素 IW (M) には1を入れておく．DO_28 では，DO_27 のループで正常にカウントされた分子（LCR (K) 上に番号が記録された分子）に対応する作業配列要素 IW (M) をゼロにする．DO_29 では，IW (M) がゼロでない，すなわち，DO_27 のループで消去された分子番号を収集し，その総数をカウントする（データ圧縮）．最後の DO_30 で，データ圧縮された分子についてのみ，スカラ計算によって cross-reference テーブル上に分子番号を記録していく．これらの DO ループのうち，DO_30 以外はすべてベクトル化されている．図4のプログラムでは，作業用の配列を必要とするが，スーパコンピュータでは，拡張領域を用いて記憶容量をかなり大きくとれるので問題はない．

図4の DO_15 において，cross-reference テーブル上の各セルの座席数は余裕をもって確保されているが，低い確率ではあってもそのテーブル上で分子が重複した位置を占める可能性を否定できない．重複を避けるためには，これをチェックする計算を DO_30 の後に追加する方法が考えられる．**図10**は，そのためのルーチンを示したものである．変数 ICHE は，最初ゼロが与えられているが，重複が見つかると1に変化する．チェックのための二つの DO ループ（DO_80 と DO_90）はいずれもベクトル化できるので計算時間の増加はほとんどない．重複が見つかったときにはスカラ計算で DO_5 を実行した後，図4の行番号 100

```
c-------------------------------------
c   ICHE is initially zero.  If the
c   overlap occurs, it turns to one.
c   IC0(N) is the last address for
c   cell N in LCR( ).
c-------------------------------------
c
      IF(ICHE.NE.1) THEN
c
      DO 80 N=1,NC
   80 IC0(N)=IC1(N)+IC2(N)
c
      DO 90 N=1,NC-1
      IF(IC0(N).GT.IC2(N+1)) ICHE=1
   90 CONTINUE
c
      IF(ICHE.EQ.1) THEN
      DO 4 N=1,NC
    4 IC1(N)=0
      DO 5 M=1,NM
      N=IP(M)
    5 IC1(N)=IC1(N)+1
      GO TO 100
      ENDIF
c
      ELSE
c
      ICHE=0
c
      ENDIF
c-------------------------------------
```

図 10 cross-reference テーブル上の分子の重複をチェックする計算ルーチン

Fig. 10 Routine of the checkup for the overlap of molecules on the cross-referencing table.

に飛び，DO_6 以下を再計算することになる．なお，そのようなチェックをしない場合に重複が生じると，異なるセルに属する分子同士で衝突の計算を行うという非現実的な計算の可能性が出てくる．ただ，重複の確率が非常に低いものならば，そのような非現実的な計算も計算誤差の中に埋もれてしまうものであり，計算が無限ループに陥るという性質のものではない．

図 11 は，図 4 の DO_6 以降の計算に要する時間を，従来のプログラム（図 3 の DO_6 と DO_7）に対する比として描いたものであるが，分子番号がセルに無関係に与えられていれば速度増加の得られることがわかる．図 11 には「分子の並べ替えルーチン」全体の計算時間（従来のものとの比）も描かれている．ただし，どちらの計算時間も状況によりある程度変動する．

図 12 は，セル内分子数を 20 に固定し，総分子数（すなわち総セル数）を 6 通りに変化させた場合のルーチン全体の計算時間(従来のプログラムとの比)

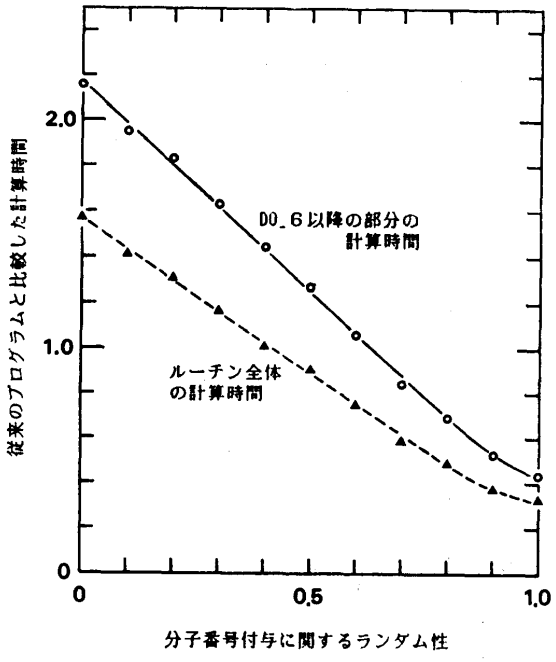

図 11 分子番号の付け方の違いによって生じる，図 4 の DO_6 以降の計算に要する時間および図 4 のプログラム全体に要する計算時間の変化（従来のプログラムとの比）

Fig. 11 Ratio of the computer time for the procedure after 'DO_15' in Fig. 4 to that after 'DO_5' in Fig. 3 and ratio of the computer time for the whole procedure in Fig. 4 to that in Fig. 3.

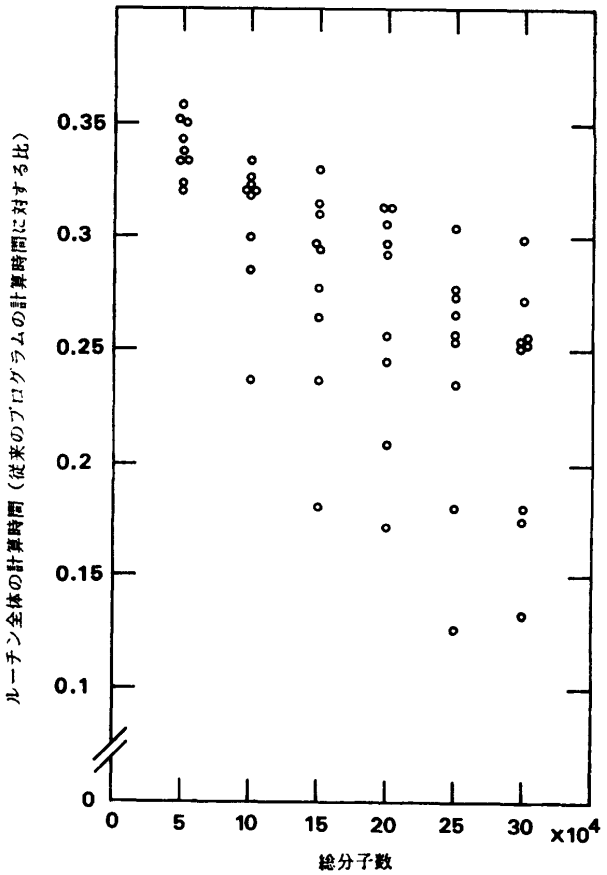

図 12 総分子数を変化させた場合の，分子並べ替えルーチン全体の計算に要する時間（従来のプログラムとの比）

Fig. 12 Ratio of the computer time for the procedure of molecular indexing with forced vectorization to the conventional one is investigated in several conditions of total number of molecules, where each cell contains twenty molecules.

を，それぞれ 9 回にわたって調べたものである．図から明らかなように，得られた計算時間にはかなりの変動が見られるが，全体的にみて，分子数が増加するほどベクトル化の効果は上がる傾向にある．

4. 分子間衝突の計算

本論文における DSMC 法では，分子間衝突の計算に Bird の方法[1] を採用している．Bird 法の特徴は，各衝突分子ペアに対して衝突時間を計算するところにある．各々のセルにおいて，分子間衝突が生じるごとにそれに応じた衝突時間が加算され，衝突時間の和が各セル Δt を超えるまで衝突の計算が行われる*．分

* 衝突時間 (collision time) の物理的意味については文献 1) を参照されたい．なお，Δt 時間内に，セル内分子のすべての組合せについて衝突が計算されるとは限らない．

子間衝突の計算は，セルごとに独立に行われるので，基本的にはベクトル化が可能であるが，各セルにおいて，衝突時間の和がタイムステップ Δt を超えるまで衝突計算を繰り返すという処理は，GO_TO 文による戻りが必要なために，そのままではベクトル化できない．そこで，分子間衝突のルーチンでは，セル数だけ繰り返す DO ループについてベクトル計算を行うこととし，そのベクトル化された DO ループの中では，各セル一対の分子間衝突しか計算しないものとする．すなわち，まず「データ圧縮」により，分子間衝突が可能なセル（分子が2個以上あり，衝突時間の和がタイムステップを超えていないセル）を収集した後，それらセルごとに，まず1回だけ分子間衝突を計算する．一時的に選ばれた分子ペアが実際に衝突するかどうかは，相対速度の大きさに依存するので，場合によっては，その衝突が棄却されることもある．その場合，そのセルは分子間衝突を計算しない．各セルごとに，一対の分子ペアについて衝突計算を終えた後，再び「データ圧縮」して，分子間衝突可能なセルを収集する．そして，それらのセルごとに，やはり1回だけ分子間衝突を計算する．これらの操作を繰り返し，すべてのセルの衝突時間の和がタイムステップを超えたら，分子間衝突の計算は終了する．

5. 実際のシミュレーションにおける従来のプログラムとの速度比較

二次元スリットを通過する希薄気体流に関して，比較的密度の高い状態（平均自由行程と代表長さの比で定義されるクヌッセン数の逆数が 1/Kn＝70 の状態，用いた分子数約50000個）のシミュレーションにおいて，従来用いていたプログラムによる計算速度と，前述のようなベクトル化を施したプログラムによる計算速度との比較を行った．**表2**は，四つのルーチンの計算に要した CPU 時間の小計を示したものであるが，それぞれの計算ルーチンでベクトル化の効果が現れており，全体的に見ると，約8倍の高速化が達成されている．今回詳述した「cross-reference テーブル上の分子の並べ替え」の計算ルーチン自体は，従来のものの約4倍の速度となっているが，この部分を従来どおりのスカラ計算で行うと，全体的には5倍程度の高速化にしかならないので，このルーチンのベクトル化の効果は大きい．

表 2　実際のシミュレーションにおける従来のプログラムとベクトル化されたプログラムとの計算時間の比較

Table 2　Comparison between the computer time of the simulation using the vectorized program and that using the conventional one.

	Conventional (sec)	Vectorized (sec)	Ratio
Molecular motion	272.3	42.7	6.38
Cell identification	503.4	54.2	9.29
Molecular indexing	154.7	36.8	4.20
Intermolecular collision	521.1	40.8	12.8
Sum total	1451.5	174.5	8.32

6. むすび

DSMC 法プログラムに関して，ベクトル計算による高速化を検討し満足できる結果を得た．プログラム中の，通常の方法ではベクトル化できない DO ループに対しては，「強制ベクトル化とそれを補う（演算欠落修正のための）後処理」という算法を見いだした．また，演算欠落発生率に及ぼす，セル数，セル内分子数および並列演算数の影響を明らかにした．この算法は，状況によって計算時間が変動する等の不確定な要素を持つが，このことに注意して用いれば，一般プログラムの高速化にも，十分有効に利用できる．従来，ベクトル化の不適な DO ループをベクトル化するためには，DO ループの直前でベクトル化に適する形に変形するというのが一般的な方法[7]であったが，今回述べたような算法は，今後，他の計算においても適用されるものと考える．

謝辞　本研究に関し，貴重なご助言をいただいた名古屋大学藤本哲夫教授に感謝の意を表する．また多くの有益なご指摘をいただいた査読者にお礼申し上げる．今回の計算には，岡崎国立共同研究機構分子科学研究所・電子計算機センターの HITAC-S 810/10 および M 680 を利用した．

参考文献

1) Bird, G. A.: *Molecular Gas Dynamics*, Oxford Univ. Press, London (1976).
2) Bird, G. A.: Monte-Carlo Simulation in an Engineering Context, *Proc. 12 th. Int. Symp. Rarefied Gas Dynamics, Progress in Astronautics and Aeronautics*, Vol. 74, Part 1, pp. 239-255 (1981).
3) 南部健一：希薄気体力学の新しい応用面—おもに成膜技術への応用，機械の研究，Vol. 39, No.

5, pp. 17-22 (1987).
4) 藤本哲夫，宇佐美勝：スリットを通過する希薄気体のモンテカルロシミュレーション（上流と下流との間で大きな圧力比をもつ場合），日本機械学会論文集，B編，Vol. 50, No. 459, pp. 2717-2722 (1984).
5) 藤本哲夫，宇佐美勝，加藤征三：円筒孔を通過する希薄気体の穴板温度上昇による流量抑制効果（非定常実験と直接シミュレーション），日本機械学会論文集，B編，Vol. 51, No. 466, pp. 1957-1961 (1985).
6) Merkle, C. L.: New Possibilities and Applications of Monte-Carlo Methods, *Proc. 13th Int. Symp. Rarefied Gas Dynamics*, Vol. 1, pp. 333-348 (1985).
7) 例えば，折居茂夫：ベクトル計算機における粒子コードの高速化手法，情報処理，Vol. 27, No. 11, pp. 1257-1263 (1986).

（昭和62年 6 月 11 日受付）
（昭和62年 10月 14 日採録）

宇佐美 勝（正会員）

昭和24年生．昭和48年名古屋工業大学機械工学科卒業．昭和50年大学院修士課程修了．同年(株)新潟鉄工所に入社．昭和51年三重大学工学部機械工学科助手．昭和62年同助教授．工学博士．希薄気体流のシミュレーションおよび実験的研究に従事．最近は，ニューラルネット，並列分散処理に興味をもつ．日本機械学会会員．

加藤 征三

昭和18年生．昭和42年名古屋大学工学部機械工学科卒業．昭和47年大学院博士課程満了．同年名古屋大学工学部機械工学科助手．昭和52年三重大学工学部機械工学科助教授．昭和59年同教授．工学博士．分子線解析，高速噴流のレーザーによる可視化，ホログラフィーによる温度場の解析等の研究に従事．日本機械学会会員．