

JAERI-Data/Code

96-008



直接シミュレーションモンテカルロコード
Pstc-2dの並列化

1996年3月

岸田則生*・横川三津夫・渡辺 正・蕪木英雄

日本原子力研究所
Japan Atomic Energy Research Institute

本レポートは、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。
入手の問い合わせは、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、お申し越してください。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）で複写による実費頒布をおこなっております。

This report is issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division, Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1996

編集兼発行 日本原子力研究所
印 刷 いばらき印刷㈱

直接シミュレーションモンテカルロコードPstc_2d の並列化

日本原子力研究所計算科学技術推進センター

岸田 則生*・横川三津夫・渡辺 正・蕪木 英雄

(1996年2月8日受理)

2次元 Rayleigh-Bénard 流れを直接シミュレーションモンテカルロ(DSMC)法でシミュレートするPstc_2d コードが Intel Paragon XP/S15-256 上で並列化された。並列化には Paragon 固有のメッセージパッシングインターフェイスであるNXライブラリが使用された。DSMC法はプロセッサ数 n に対して $O(n^2)$ のメッセージ転送回数を要求するので、 n が大きい時の並列化効率 はあまり良くないことが判明した。計算精度をある程度犠牲にするがメッセージ転送回数が $O(n)$ である新たなメッセージ転送アルゴリズムが示唆される。PVMによる並列化とNXライブラリによる並列化が比較され、後者の方が約3倍速いことが判明した。2次元 Rayleigh-Bénard 対流に対し Paragon の大メモリ空間を利用することで、アスペクト比が8の計算が可能になった。得られたロールの数は連続流モデルから期待される結果とよく一致した。この結果から、DSMC法をかなり大きな体系の連続流のシミュレーションに適用可能なことが判明した。

Parallelization of Direct Simulation Monte Carlo Code Pstc_2d

Norio KISHIDA*, Mitsuo YOKOKAWA, Tadashi WATANABE and Hideo KABURAKI

Center for Promotion of Computational Science and Engineering
Japan Atomic Energy Research Institute
Nakameguro, Meguro-ku, Tokyo

(Received February 8, 1996)

Pstc_2d code which simulates the two-dimensional Rayleigh-Bénard flow using the direct simulation Monte Carlo (DSMC) method has been parallelized on Intel Paragon XP/S15-256. The parallelization was carried out by employing message passing method. It has been made clear that parallelization efficiency is considerably worse when the number of processor elements becomes large. It is because the number of message passing increases in $O(n^2)$. A new algorithm which decreases the number of message to $O(n)$ is proposed, and it gives a little affect on physical accuracy of calculated results. It was found that parallelization using the NX library is three times faster than parallelization using the PVM. Utilizing a large memory space of the Paragon, the Rayleigh-Bénard system in the cavity of 44.8 mm in width and 5.6 mm in height with 1.28 million particles has been computed. The number of rolls agrees with that obtained by macroscopic hydrodynamic models. From this result, it is found that the DSMC method is applicable to simulate a continuum flow in a large scale system.

Keywords: Parallel Programming, Paragon, DSMC, Rayleigh-Bénard Flow, Monte Carlo Simulation, Calculation, Pstc_2d

* on loan to CRC Research Institute Inc.

目 次

1. 序 論	1
2. Pstc_2d コード	1
3. 並列化	3
3.1 並列化手法	3
3.2 並列アルゴリズム	4
3.3 計算条件および計算結果	5
3.4 並列化性能	6
4. まとめ	7
謝 辞	8
参考文献	8
付録1 Pstc_2d 入力マニュアル	19
付録2 Pstc_2d データ入力サブルーチン	20

Contents

1. Introduction	1
2. Pstc_2d Code	1
3. Parallelization	3
3.1 Parallelizing Methods	3
3.2 Parallel Algorithms	4
3.3 Calculated Results	5
3.4 Performance of Parallel Program	6
4. Summary	7
Acknowledgments	8
References	8
Appendix 1 Input Manual of Pstc_2d	19
Appendix 2 Data Input Subroutine of Pstc_2d	20

1. 序論

1960年代に Bird によって導入された直接シミュレーションモンテカルロ (Direct Simulation Monte Carlo : DSMC) 法¹⁾は希薄流から遷移流までを微視的にシミュレートする有力な方法として広く利用されている。最近、Kaburaki and Yokokawa²⁾と Watanabe ら³⁾らは DSMC 法をキャビティ流れや Rayleigh-Bénard 流れに適用して、DSMC 法が Navier-Stokes 方程式で記述される連続流に対しても適応可能であることを示した。DSMC 法は Boltzmann 方程式の解法の一つであるが、この方法は差分法とは異なり空間・時間を離散化しない解法である。すなわち、DSMC 法は流れの挙動をその流れを構成する物質粒子 (原子・分子) の非常に多くの集まりの運動として捉える。実際は Boltzmann 方程式を導入する際に使用した運動法則に従って、多くの粒子をモンテカルロ法で追跡することで流れの物理量の変化を捉える。DSMC 法の長所は計算手順が非常に簡単なことである。しかし、連続流や希薄流でも実体系のような大きな装置の流れをシミュレートする場合は、莫大な粒子数を用いた長時間計算を行わねばならないという短所もある。このことが DSMC 法による大規模解析があまり実行されない一つの原因となっている。

我々は 2 次元 Rayleigh-Bénard 対流の発展過程を微視的に解明するために Pstc_2d コードを開発し、アスペクト比 (縦横の長さの比) が 2 の矩形領域について多くのシミュレーションを行ってきた^{3, 4)}。その結果、DSMC 法で得られた速度場や温度場は連続流体モデルから得られるものと良く一致することを確認した。しかし、必要とする計算機資源 (メモリ空間、計算時間) の制約から高アスペクト比の Rayleigh-Bénard 対流の計算は実行できなかった。DSMC 法が連続流体のシミュレーションにおいて有効な手法であることを確立するためには、小さな体系ばかりでなく大きな体系においても連続流モデルと一致した結果が得られることを確かめる必要がある。

今回、我々はスカラ並列計算機 Intel Paragon XP/S15-256 上で動作する 2 次元 Rayleigh-Bénard 系をシミュレートする並列版 Pstc_2d コードを開発して、従来困難であったアスペクト比が 8 のシミュレーションを実行した。その結果から、DSMC 法は大規模な連続流に対しても有力なシミュレーション手法の一つとなり得ることが判明した。また、並列計算効率が使用プロセッサエレメント (Processor Element: PE) 数や問題の大きさに対してどの様に変化するかを調べた。さらに、Paragon 固有の NX ライブラリと一般に広く使用されている PVM (Parallel Virtual Machine)⁵⁾ による並列化を同時に行ない、両者の性能比較も行ったので、その結果も併せて報告する。

2. Pstc_2d コード

Pstc_2d コードは 2 次元 Rayleigh-Bénard 系を DSMC 法でシミュレートする計算コードである。Rayleigh-Bénard 系とは一様重力場中に置かれた上面と下面をそれぞれ一定温度で冷却および加熱された、無限の広がりを持つ水平流体層である。この系は上下面の温度差がある値を越えるとロールと呼ばれる渦が周期的に発生する体系として知られている⁶⁾。DSMC 法によるシミュレーションでは無限系の計算は実行不能なので、計算は左右の側面に適当な境界条件を課した矩形領域で行われる。

1. 序論

1960年代に Bird によって導入された直接シミュレーションモンテカルロ (Direct Simulation Monte Carlo : DSMC) 法¹⁾は希薄流から遷移流までを微視的にシミュレートする有力な方法として広く利用されている。最近、Kaburaki and Yokokawa²⁾と Watanabeら³⁾らは DSMC 法をキャビティ流れや Rayleigh-Bénard 流れに適用して、DSMC 法が Navier-Stokes 方程式で記述される連続流に対しても適応可能であることを示した。DSMC 法は Boltzmann 方程式の解法の一つであるが、この方法は差分法とは異なり空間・時間を離散化しない解法である。すなわち、DSMC 法は流れの挙動をその流れを構成する物質粒子 (原子・分子) の非常に多くの集まりの運動として捉える。実際は Boltzmann 方程式を導入する際に使用した運動法則に従って、多くの粒子をモンテカルロ法で追跡することで流れの物理量の変化を捉える。DSMC 法の長所は計算手順が非常に簡単なことである。しかし、連続流や希薄流でも実体系のような大きな装置の流れをシミュレートする場合は、莫大な粒子数を用いた長時間計算を行わねばならないという短所もある。このことが DSMC 法による大規模解析があまり実行されない一つの原因となっている。

我々は 2次元 Rayleigh-Bénard 対流の発展過程を微視的に解明するために Pstc_2d コードを開発し、アスペクト比 (縦横の長さの比) が 2 の矩形領域について多くのシミュレーションを行ってきた^{3, 4)}。その結果、DSMC 法で得られた速度場や温度場は連続流体モデルから得られるものと良く一致することを確認した。しかし、必要とする計算機資源 (メモリ空間、計算時間) の制約から高アスペクト比の Rayleigh-Bénard 対流の計算は実行できなかった。DSMC 法が連続流体のシミュレーションにおいて有効な手法であることを確立するためには、小さな体系ばかりでなく大きな体系においても連続流モデルと一致した結果が得られることを確かめる必要がある。

今回、我々はスカラ並列計算機 Intel Paragon XP/S15-256 上で動作する 2次元 Rayleigh-Bénard 系をシミュレートする並列版 Pstc_2d コードを開発して、従来困難であったアスペクト比が 8 のシミュレーションを実行した。その結果から、DSMC 法は大規模な連続流に対しても有力なシミュレーション手法の一つとなり得ることが判明した。また、並列計算効率が使用プロセッサエレメント (Processor Element: PE) 数や問題の大きさに対してどの様に変化するかを調べた。さらに、Paragon 固有の NX ライブラリと一般に広く使用されている PVM (Parallel Virtual Machine)⁵⁾ による並列化を同時に行ない、両者の性能比較も行ったので、その結果も併せて報告する。

2. Pstc_2d コード

Pstc_2d コードは 2次元 Rayleigh-Bénard 系を DSMC 法でシミュレートする計算コードである。Rayleigh-Bénard 系とは一様重力場中に置かれた上面と下面をそれぞれ一定温度で冷却および加熱された、無限の広がりを持つ水平流体層である。この系は上下面の温度差がある値を越えるとロールと呼ばれる渦が周期的に発生する体系として知られている⁶⁾。DSMC 法によるシミュレーションでは無限系の計算は実行不能なので、計算は左右の側面に適当な境界条件を課した矩形領域で行われる。

DSMC 法の計算手順を簡単に記すと以下のようなになる。

1. 計算パラメータを読み込む。
2. 計算領域を多数の衝突セルに分割する。
3. サンプル粒子を各衝突セルに配置する。
4. サンプル粒子を微小時間の間、各粒子の速度に従って移動させる。
5. 計算領域外に出た粒子を各境界条件に従って再配置する。
6. 各衝突セル内でモンテカルロ法により粒子衝突をシミュレートする。
7. 流れの巨視的物理量を求めファイルに出力する。
8. 流れの定常性を調べ、定常状態に達したら計算を終了し、非定常状態であればステップ 4 からの計算を繰り返す。

ステップ 2 のセル分割において、セルの大きさはセル中での流体密度が一定と見なせる程度に取られねばならない。一方、ステップ 6 における粒子衝突の計算では、自己が属するセル内の他の粒子とのみ衝突が起これば DSMC 法では仮定している。従って、セルの大きさは高々平均自由行程の程度に取られなければならない。ところで、流体中に進行衝撃波が発生している場合などでは、局所的に密度が高い場所、言い換えれば平均自由行程が短い場所が時間と共に移動する。このような場合、セルの大きさが全計算領域で同じで計算中に大きさを変えられないコードでは、密度が高い場所でのセル分割の条件に合わせて全領域のセル分割を行わねばならず、セルの大きさがかなり小さいためセル数が非常に大きくなり計算資源を無駄に使用することになる。そこで、Pstc_2d ではセルの大きさは流体の局所的な密度に応じて自動的に変化するようにコーディングされている。Pstc_2d コードでは物理量の平均値を求めるセルをサンプリングセル、自動分割されて実際に衝突計算が実行されるセルを衝突セルと呼んでいる。ステップ 3 の粒子の配置では、位置については完全にランダム、速度については Maxwell の速度分布関数に従うように配置される。ステップ 4 で粒子を移動した後、各粒子がどのセル中に存在しているかを記憶しているインデックス配列の更新を行う。ステップ 5 の境界条件の処理では上下面は壁温に応じた拡散反射条件を、側面では鏡面反射もしくは周期境界条件を課している。ステップ 6 では密度一様の下で Boltzmann 方程式から導かれる Kac 方程式⁷⁾に従って粒子衝突の計算がモンテカルロ法により実行される⁸⁾。ステップ 7 ではサンプリングセル平均の流体の速度、温度等が計算される。また後のグラフ化等が必要となる量をファイルに書き出す。ステップ 8 で計算の終了を判定し、一連の計算を終える。

3. 並列化

3.1 並列化手法

Pstc_2d コードを並列化する場合、分散メモリ型スカラ並列計算機の Intel Paragon XP/S15-256 か、分散メモリ型ベクトル並列計算機の Fujitsu VPP500/42 の利用が可能である。DSMC 法の計算で最も CPU 時間を消費するのは分子衝突をモンテカルロ法によってシミュレートする部分である。しかし、Pstc_2d コードは衝突計算アルゴリズムに Bird 法を採用している関係で、この部分のベクトル化は出来ない⁸⁾*。VPP500/24 は並列計算機の観点からすると超並列計算機ではなく、ベクトル計算機を少台数結合した並列計算機である。VPP500/24 では、90 %程度のベクトル化率を達成できないと、並列化だけで大幅なスピードアップを望むのは困難である。⁹⁾したがって、VPP500/24 は Pstc_2d コードの並列化には向いていないので、今回の並列化は Paragon 上のみで行うことにした。

Paragon は分散メモリ型の MIMD(Multiple Instructions and Multiple Data Streams) 並列計算機である。ただし、コンパイラ自身による自動並列化機能は提供されていないし、分散メモリを仮想共有メモリに見せかけるライブラリも提供されない。各 PE は自分自身が持つ局所メモリ以外のデータは参照できない。他 PE が持つデータの参照はデータ転送関数ライブラリである NX ライブラリを用いて行われる。Paragon の各 PE は格子状に結合されていて、各 PE は 200 MB/s でデータを送受信できる。以上のことから、Paragon における並列化はいわゆるメッセージパッシング手法を用いて行うことになる。

DSMC 法の並列化では、

1. サンプル粒子を粒子番号に従って使用 PE 数のグループに分割して計算を実行する粒子分割法、
2. 計算領域を使用 PE 数で等分割して、各 PE がそれぞれの分割領域の計算を実行する領域分割法、
3. 衝突セルをセル番号に従って使用 PE 数のグループに分割して計算を実行するセル分割法

の採用が可能である。分割領域はセルの集合から構成されるので、領域分割法とセル分割法は分割領域の形とセル番号の振り方が少し異なるだけのアルゴリズム的にはほとんど同等の並列化手法である。

DSMC 法での粒子衝突の計算は各セル毎に独立に実行されるので、Pstc_2d コードの最内側ループは粒子番号ではなくセル番号のループになっている。これは DO ループ分割による並列化を考えると、粒子分割では自然な並列化とはならず大幅なアルゴリズムの変更をしなければ並列化出来ないことを意味する。Paragon はグローバルメモリ機構を持ってないので、他 PE 上にある粒子情報を参照しようとすれば、メッセージパッシングによる PE 間データ転送を行わなければならない

*南部法⁸⁾を採用すればベクトル化可能であるが、これはわずかではあるけれども計算の物理的な内容を変更することになるので、今回の並列化では別の衝突計算法への変更は行わないことにした。

ない。従って、粒子分割法で並列化するとするならば、セルループ中で非常に複雑な PE 間通信を行い衝突候補粒子の情報を他 PE から転送して来なければならず、効率の良いアルゴリズムとはならない。これに対してセル分割法は自然な DO ループ分割による並列化になっており、有力な並列化手法の一つである。ところで、DSMC 法では粒子がどのセルに属するかのインデックス情報を持っている必要がある。この情報は粒子が時間ステップごとに移動するため、ステップ毎に更新する必要がある。領域分割法では分割領域を常に矩形に出来るので、粒子がどの PE のどのセルに移動するかを判断するのは極めて容易である。それに対して、セル分割法は使用 PE 数によっては分割領域が矩形とはならず、粒子の属する PE とセルを求める部分のアルゴリズムが少しだけ複雑になる。Figure 1 に 30 セルの矩形領域を 5 台の PE を用いて領域分割とセル分割を行なった例を示す。領域分割法は確かに自然な DO ループ分割にはなっていないが、問題領域の分割という面ではセル分割より自然である。分割された各領域毎に 1PE を割り当てる並列化を採用した場合は、セル分割でも領域分割でもメッセージパッシングの回数およびデータ転送量は全く同じである。並列化に要する工数は両分割法ともほぼ同等と見積もられるので、今回は粒子の属するセル判定コストがわずかだけ少なくなる、領域分割法で並列化することにした。また、並列化プログラムのコーディングは異なるデータに対し同一プログラムを各 PE で走らせる SPMD (Single Program and Multiple Data Stream) 方式を採用した。この理由は、並列プログラムの大きさはマスタ・スレーブ方式の方が少し小さくなるが、SPMD 方式の方がソースコードの可読性や保守性が高く管理し易いためである。

3.2 並列アルゴリズム

並列化版 Pstc_2d コードの計算手順を簡単に示すと次のようになる。

1. 標準入力に接続されている PE (ゼロ番 PE) から計算パラメータを読み込み領域分割を行うと共に、パラメータ値を他の PE に転送する。
2. 各 PE が受け持つ計算領域を多数の衝突セルに分割する。
3. サンプル粒子を各衝突セルに配置する。
4. サンプル粒子を微小時間その速度の従って移動させる。
5. 自分に割り当てられた計算領域外に出ていった粒子の情報を移動先の PE に送信し、逆に自分の計算領域に入ってきた粒子の情報を受信する。
6. シミュレーション境界外に出た粒子を各境界条件に従って再配置する。
7. 各衝突セル内でモンテカルロ法を用いて粒子衝突をシミュレートする。
8. 流れの巨視的物理量を求めた後、それをゼロ番 PE に転送してファイルに出力する。

9. 流れの定常性を調べ、定常状態に達したら計算を終了し、非定常状態であればステップ 4 からの計算を繰り返す。

逐次版との最も大きな違いは、並列版ではステップ 5 の操作が加わることである。

DSMC 法では衝突セルの大きさは第 2 章で述べたように、本来粒子の平均自由行程と同程度に取られねばならない。すなわち、粒子の移動距離は高々セルの幅程度なので、粒子情報のやりとりは物理的には隣接セル間のみで行うだけでよい。つまり、PE 間の通信は隣接分割領域間でのみ行えば良いはずである。例えば Fig. 2 のような 5 分割の場合なら、PE2 は PE1 と PE3 との間でのみメッセージパッシングを行えばよく、PE0 と PE4 との間ではメッセージパッシングする必要はない。ところで、平均自由行程 l の粒子が自由行程 s を持つ確率は $\exp\{-s/l\}$ で与えられるから、わずかな確率ではあるが衝突後の粒子はシミュレーション領域中のどのセルにも原理的には移動可能である。一方、逐次版は粒子の全領域への移動を可能にするコーディングが採用されている。そこで、今回の並列化では逐次版との計算結果のコンシステンスを保持するために、全ての PE 間で粒子情報を交換するようコーディングすることにした。

3.3 計算条件および計算結果

Rayleigh-Bénard 系の計算領域は四辺を平坦な壁面で囲まれた 2 次元矩形領域である。高さは 5.6 mm、幅はアスペクト比が 2、4、8 の場合それぞれ 11.2 mm、22.4 mm、44.8 mm であり、サンプリングセルの形状は一辺 0.28 mm の正方形である。従って、サンプリングセル数はそれぞれ 800、1600、3200 である。衝突セルは計算開始時において 1 サンプリングセルにつき 25 とした。

初期温度は上下面とも 80 K とし、初期圧力は 20 Pa とした。サンプル粒子には、空気の平均分子を想定した直径 3.7 Å、質量 4.8×10^{-26} Kg の剛体球分子を使用し、初期状態において 1 サンプリングセル当たり 400 個の粒子を Maxwell 分布に従って配置した。以上の条件においてクヌーセン数を計算するとおおよそ 0.016 となり、この系はほぼ連続流体と見做すことが出来る。

計算は時間ゼロで底面をレイリー数 4000 に対応する 358 K に設定し、粒子の時間変化を追跡した。タイムステップは平均自由時間の 0.9 倍とした。境界条件は、上下面では拡散反射とし、左右面では鏡面反射もしくは周期境界条件を用いる。また、仮想的な重力加速度を用いて対流の成長を加速した。

逐次版と並列化版の結果が一致することを確かめるために、アスペクト比が 2 の矩形領域における定常状態に達した速度場をまず計算した。それぞれ結果を Fig. 3(a) と (b) に示す。並列計算は 2 台の PE を使用して行われた。両者は良く一致しており、並列化が誤り無く行われているのが分かる。

次に、側面の境界条件が正しく反映されているかを調べるために、アスペクト比が 4 の場合において鏡面反射と周期境界条件を課して計算を行った。得られた定常状態の速度場の様子をそれぞれ Fig. 4(a) と (b) に示す。この計算は PE を 128 台使用して行われた。Figure 4(b) の周期境界条件では渦の中心が鏡面反射の場合に比べてずれているのが、また左側壁から流出した流れが右

側壁から流入しているのがみてとれる。これは、周期境界条件が正しく計算に反映されていることを示している。Rayleigh-Bénard 系の連続流体モデルによる理論解析では、ほとんどの場合周期境界条件が用いられるので、周期境界条件が正しく課されているか確かめることは重要である。

アスペクト比が 8 の矩形領域で周期境界条件を課した場合の速度場の発達の様子を Fig. 5 に示す。それぞれの図は 200 タイムステップにわたるサンプリングの時間平均より得られたものである。200 ステップでは下面の温度上昇による上向きと重力による下向きの速度ベクトルが見られる。200 ステップから 600 ステップでは渦が形成される過程が見て取れる。1600 ステップで 10 個の対流渦が形成され、2000 ステップではほぼ安定な渦となる。この状態は以後安定で 10000 ステップにいたっても変わることがない。

アスペクト比が 8 の矩形領域で反射境界条件を課した場合の速度場の発達の様子を Fig. 6 に示す。600 ステップまでは周期境界条件の場合とほぼ同じように渦は成長しているが、全領域に渦が形成されるのは極めて遅く 4000 ステップになってからである。また渦の数も 10 個であり、やはり周期境界条件の場合と異なる。これらの渦は安定ではなく 9000 ステップあたりから右から 3 番目の渦が消えはじめ 10600 ステップにおける 9 個の対流渦形成の中間状態を経由して、最終的には 11400 ステップで現れる 8 個の対流渦状態に落ち着き、これは 20000 ステップに至るまで安定であった。初期乱数を変えて同じ計算を行うと最初から 8 個の対流渦状態が出現することもあり、この過程がカオス的であることが示唆される。

以上の結果は連続流体モデルから得られる結論と一致しており、DSMC 法が連続流体に近い流れにも適応可能なシミュレーション手法であることを示す結果である。これらのアスペクト比が 8 の計算は Paragon の大きなユーザメモリ空間 (6 GB) が利用可能になったために実行できたものであり、大規模 DSMC 解析には並列計算機が必須であることがわかる。

3.4 並列化性能

まず最初に使用 PE 数を増加させたときの並列化性能について調べる。Table 1 に、アスペクト比を 2 に固定しプロセッサ数を変化させた時の 100 ステップまでの実行時間、スピードアップ (S_n)、並列化効率 (U_n) を示す。ただし、 $S_n = T_1/T_n$ 、 $U_n = (T_1/n)/T_n$ である。ここで T_1 は逐次版の実行時間を、 T_n は n 台の PE を使用したときの実行時間を表す。並列化性能を調べるこの計算においては、物理量のサンプリング間隔を 10 タイムステップ毎に行っているため、前章のシミュレーション計算とは計算条件は異なっている。Figure 7 に得られたスピードアップの変化を示す。破線は並列化が理想的になされた場合に得られるスピードアップ直線である。並列化効率は既に 4 PE 使用時に 90 % を下回っていて、32 PE では約 30 % にまで低下してしまう。これは DSMC 法のコードでは原理的に全ての PE 間の組合せについてデータ転送が必要なため、通信回数が $n(n-1)/2$ すなわち $O(n^2)$ で増加するためであると考えられる。

通信量を減らし並列化効率を上げるためには、平均自由行程 (mfp) を指標にした切断距離を導入して、2 台の PE が受け持つ計算領域間の最近接距離がこの切断距離以上であれば、メッセージパッシングを行わないような新たなアルゴリズムを開発する必要がある。例えば、切断距離とし

て 4.6 mfp を選ぶと、この距離以上に移動する可能性のある粒子数は全体の $\exp\{-4.6l/l\} \cong 0.01$ だから、この修正が最終結果に与える影響は高々 1 % であろうし、6.9 mfp なら 0.1 % であろう。この方法は確かに $O(n)$ でしか通信量は増加しない。しかし、以上のようにこの新しい方法は本来の DSMC 法の近似計算である。従って、この変更によってシミュレーション結果がほとんど影響を受けない切断距離はシミュレーションを実行する物理系毎に慎重に見いださなければならないと考えられる。

次に問題の大きさを変化させた時の並列化性能を調べる。Table 2 にアスペクト比が 2、4、8 の場合にアスペクト比に比例した PE 数を用いた時の 1 タイムステップ当たりの計算に要する時間を使用した PE 数と共に示す。問題の大きさを 2 倍にした時、PE 数も 2 倍にしているので、PE 間通信回数は $\{2n(2n-1)/2\}/\{n(n-1)/2\}$ 倍、すなわち n が大きければ約 4 倍になる。ところが、各 PE が受け持つ初期粒子数は PE 数が 2 倍に増加しているので不変である。すなわち、メッセージパッシングを除いたシリアル部分の計算量は増加しない。Table 2 を見ると PE 数が 2 倍、4 倍に増えても、計算時間は 10 % 以下しか増加していない。Table 2 の結果からすると 32 個の PE 使用時の通信コストはかなり大きいので、当然計算時間は大幅に増えることが予想される。しかし、同程度の計算時間しかかからないのは、データ転送量がゼロの通信が大部分を占めていて、実際の全データ転送量がほとんど変化しないため考えられる。これは、今回扱っているような連続流体の場合、隣接セル間でのみ粒子のやりとりが生起していることに基づく。実際、PE 間のデータ転送量を調べてみると、隣接していない PE 間ではゼロであった。今回の領域分割では一つ隣の領域まで 25 衝突セル、すなわち平均自由行程の 25 倍離れているのでこれは当然の結果である。このことから並列計算用の新たな DSMC 法のアルゴリズムの開発が望まれる。

最後にメッセージパッシングを用いた並列計算で標準的に利用されている PVM⁵⁾ と Paragon 固有の NX ライブラリの計算時間の比較を示す。Table 3 に PE 数が 2、4、8 の場合の実行時間を示す。これから、PVM 版は NX ライブラリ版の約 3 倍の実行時間を要することが分かる。Casanova ら¹⁰⁾により、データ送受信のためのメッセージパッシング関数同士の速度の比較では、データ転送速度は NX ライブラリの方が PVM より 6 倍から 7 倍速いことが分かっている。我々の結果はこれほどの速度差は無い。しかし、実際の並列コードでは各種のメッセージパッシング関数を使用されているので、単なる関数同士を比較した場合の性能が出ないのは当然である。我々の結果は PVM で書かれた応用プログラムを NX ライブラリで書き直すとの程度速度が改善されるかを見積もる際の貴重なデータとなろう。

4. まとめ

本作業を通じて DSMC 法の計算コードを効率よく並列化するには新たなデータ転送アルゴリズムを導入する必要があることが明らかになった。並列化によって利用可能になった大きなメモリ空間を使用した Rayleigh-Bénard 系のシミュレーション計算から、DSMC 法は大きな体系の連続流体に対しても連続流モデルから得られる結果と一致する事が判明した。これにより Rayleigh-Bénard 系の微視的立場からの研究の進展が促進されると考えられる。また、DSMC 法は連続流体の解析

て 4.6 mfp を選ぶと、この距離以上に移動する可能性のある粒子数は全体の $\exp\{-4.6/l\} \cong 0.01$ だから、この修正が最終結果に与える影響は高々 1 % であろうし、6.9 mfp なら 0.1 % であろう。この方法は確かに $O(n)$ でしか通信量は増加しない。しかし、以上のようにこの新しい方法は本来の DSMC 法の近似計算である。従って、この変更によってシミュレーション結果がほとんど影響を受けない切断距離はシミュレーションを実行する物理系毎に慎重に見いださなければならないと考えられる。

次に問題の大きさを変化させた時の並列化性能を調べる。Table 2 にアスペクト比が 2、4、8 の場合にアスペクト比に比例した PE 数を用いた時の 1 タイムステップ当たりの計算に要する時間を使用した PE 数と共に示す。問題の大きさを 2 倍にした時、PE 数も 2 倍にしているので、PE 間通信回数は $\{2n(2n-1)/2\}/\{n(n-1)/2\}$ 倍、すなわち n が大きければ約 4 倍になる。ところが、各 PE が受け持つ初期粒子数は PE 数が 2 倍に増加しているので不変である。すなわち、メッセージパッシングを除いたシリアル部分の計算量は増加しない。Table 2 を見ると PE 数が 2 倍、4 倍に増えても、計算時間は 10 % 以下しか増加していない。Table 2 の結果からすると 32 個の PE 使用時の通信コストはかなり大きいので、当然計算時間は大幅に増えることが予想される。しかし、同程度の計算時間しかかからないのは、データ転送量がゼロの通信が大部分を占めていて、実際の全データ転送量がほとんど変化しないため考えられる。これは、今回扱っているような連続流体の場合、隣接セル間でのみ粒子のやりとりが生起していることに基づく。実際、PE 間のデータ転送量を調べてみると、隣接していない PE 間ではゼロであった。今回の領域分割では一つ隣の領域まで 25 衝突セル、すなわち平均自由行程の 25 倍離れているのでこれは当然の結果である。このことから並列計算用の新たな DSMC 法のアルゴリズムの開発が望まれる。

最後にメッセージパッシングを用いた並列計算で標準的に利用されている PVM⁵⁾ と Paragon 固有の NX ライブラリの計算時間の比較を示す。Table 3 に PE 数が 2、4、8 の場合の実行時間を示す。これから、PVM 版は NX ライブラリ版の約 3 倍の実行時間を要することが分かる。Casanova ら¹⁰⁾により、データ送受信のためのメッセージパッシング関数同士の速度の比較では、データ転送速度は NX ライブラリの方が PVM より 6 倍から 7 倍速いことが分かっている。我々の結果はこれほどの速度差は無い。しかし、実際の並列コードでは各種のメッセージパッシング関数が使用されているので、単なる関数同士を比較した場合の性能が出ないのは当然である。我々の結果は PVM で書かれた応用プログラムを NX ライブラリで書き直すとの程度速度が改善されるかを見積もる際の貴重なデータとなろう。

4. まとめ

本作業を通じて DSMC 法の計算コードを効率よく並列化するには新たなデータ転送アルゴリズムを導入する必要があることが明らかになった。並列化によって利用可能になった大きなメモリ空間を使用した Rayleigh-Bénard 系のシミュレーション計算から、DSMC 法は大きな体系の連続流体に対しても連続流モデルから得られる結果と一致する事が判明した。これにより Rayleigh-Bénard 系の微視的立場からの研究の進展が促進されると考えられる。また、DSMC 法は連続流体の解析

にも十分使用可能なシミュレーション技法であることも明らかになった。

謝辞

(株)CRC 総合研究所の野口英樹氏に PVM 版 Pstc_2d コードを Paragon 上に実装して戴いたことに関し深く感謝いたします。

参考文献

- 1) Bird, G. A. : "Molecular Gas Dynamics," Cralendon Press, Oxford, (1976).
- 2) Kaburaki, H. and Yokokawa, M. : Molecular Simulation, **12**, 441 (1994).
- 3) Watanabe, T., Kaburaki, H. and Yokokawa, M. : Phy. Rev. E, **49**, 4060 (1994).
- 4) Watanabe, T., Kaburaki, H., Machida, M. and Yokokawa, M. : Phy. Rev. E, **52**, 1601 (1995).
- 5) Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. and Sunderam, V. : ORNL/TM-12187, 1 (1994).
- 6) Cross, M. C. and Hohenberg. P. C. : Revs. Mod. Phys., **65**, 851 (1993).
- 7) Nanbu, K. : J. Phys. Soc. Jpn., **52**, 4160 (1983).
- 8) 保原 充、大宮司 久明 編 : "数値流体力学," 東京大学出版会, 東京, 287 (1992).
- 9) 富士通 : Private Communication.
- 10) Casanova, H., Dongarra, J. and Jiang, W. : "The Performance of PVM on MPP Syatems," Univ. of Tennessee Technical Report, CS-95-301, 1 (1995)

にも十分使用可能なシミュレーション技法であることも明らかになった。

謝辞

(株)CRC 総合研究所の野口英樹氏に PVM 版 Pstc_2d コードを Paragon 上に実装して戴いたことに関し深く感謝いたします。

参考文献

- 1) Bird, G. A. : "Molecular Gas Dynamics," Cralendon Press, Oxford, (1976).
- 2) Kaburaki, H. and Yokokawa, M. : Molecular Simulation, **12**, 441 (1994).
- 3) Watanabe, T., Kaburaki, H. and Yokokawa, M. : Phy. Rev. E, **49**, 4060 (1994).
- 4) Watanabe, T., Kaburaki, H., Machida, M. and Yokokawa, M. : Phy. Rev. E, **52**, 1601 (1995).
- 5) Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. and Sunderam, V. : ORNL/TM-12187, 1 (1994).
- 6) Cross, M. C. and Hohenberg. P. C. : Revs. Mod. Phys., **65**, 851 (1993).
- 7) Nanbu, K. : J. Phys. Soc. Jpn., **52**, 4160 (1983).
- 8) 保原 充、大宮司 久明 編 : "数値流体力学," 東京大学出版会, 東京, 287 (1992).
- 9) 富士通 : Private Communication.
- 10) Casanova, H., Dongarra, J. and Jiang, W. : "The Performance of PVM on MPP Syatems," Univ. of Tennessee Technical Report, CS-95-301, 1 (1995)

にも十分使用可能なシミュレーション技法であることも明らかになった。

謝辞

(株)CRC 総合研究所の野口英樹氏に PVM 版 Pstc_2d コードを Paragon 上に実装して戴いたことに関し深く感謝いたします。

参考文献

- 1) Bird, G. A. : "Molecular Gas Dynamics," Cralendon Press, Oxford, (1976).
- 2) Kaburaki, H. and Yokokawa, M. : Molecular Simulation, **12**, 441 (1994).
- 3) Watanabe, T., Kaburaki, H. and Yokokawa, M. : Phy. Rev. E, **49**, 4060 (1994).
- 4) Watanabe, T., Kaburaki, H., Machida, M. and Yokokawa, M. : Phy. Rev. E, **52**, 1601 (1995).
- 5) Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. and Sunderam, V. : ORNL/TM-12187, 1 (1994).
- 6) Cross, M. C. and Hohenberg. P. C. : Revs. Mod. Phys., **65**, 851 (1993).
- 7) Nanbu, K. : J. Phys. Soc. Jpn., **52**, 4160 (1983).
- 8) 保原 充、大宮司 久明 編 : "数値流体力学," 東京大学出版会, 東京, 287 (1992).
- 9) 富士通 : Private Communication.
- 10) Casanova, H., Dongarra, J. and Jiang, W. : "The Performance of PVM on MPP Syatems," Univ. of Tennessee Technical Report, CS-95-301, 1 (1995)

Table 1 Execution time on Paragon.

Number of PE's	Execution time (s)	Speedup Ratio : S_n	Efficiency : U_n (%)
1	920.2	1.00	100.0
2	497.5	1.85	92.5
4	260.5	3.53	88.3
8	142.6	6.45	80.6
16	114.9	8.00	50.0
32	97.4	9.45	29.5

Table 2 Execution time for every time step with various aspect ratios.

Aspect ratio	Number of PE's	Execution time (s)
2	32	3.36
4	64	3.60
8	128	3.63

Table 3 Execution time ratios of PVM to NX library for every 50 time steps.

Number of PE's	NX (s)	PVM (s)	PVM/NX
2	497.5	1407.1	2.82
4	260.5	745.0	2.86
8	142.6	390.7	2.74

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

(a) A domain decomposition

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

(b) A cell decomposition

Fig. 1 Schematic diagrams of a domain decomposition (a) and a cell decomposition (b). A simulated area is divided into 30 cells and five PE's are used. Each PE calculates a flow in an area surrounded by the thick solid lines.

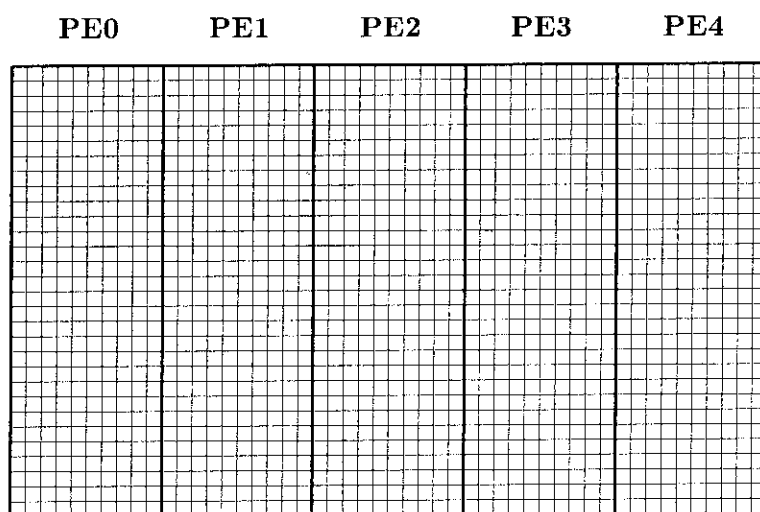
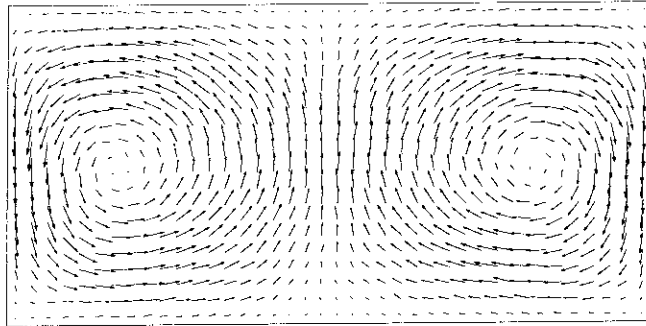
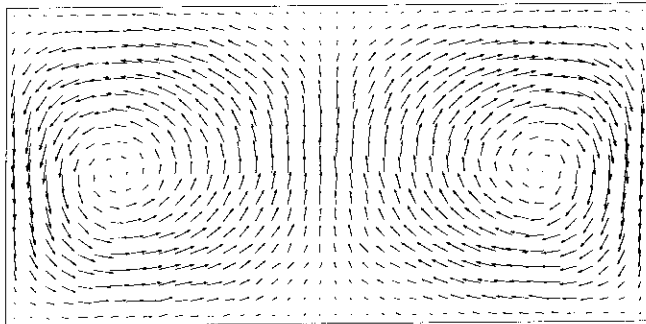


Fig. 2 Small rectangles represent sampling cells. Large rectangles surrounded by thick solid lines represent domain decomposition areas. From the principle of DSMC method, data transfers between PE's are needed because a particle in a cell can move to any cell after a collision. However data transfers between two neighboring PE's are only needed, because a particle can move within a mean free path at each time step.

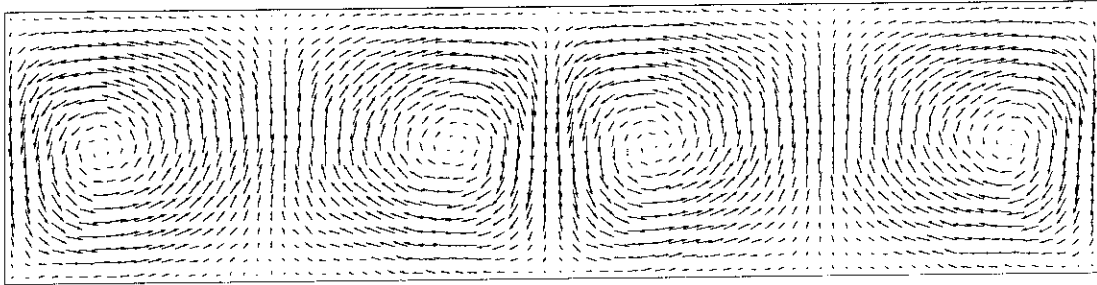


(a) Serial

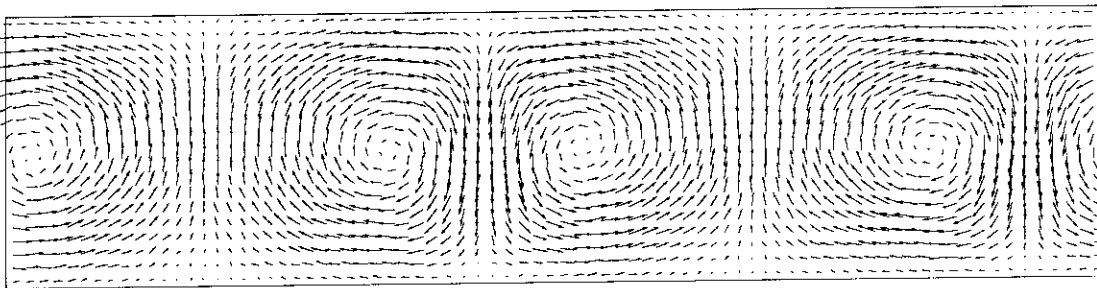


(b) Parallel

Fig. 3 Evolved velocity fields of the two dimensional Rayleigh-Bénard flow with aspect ratio of two. The upper figure shows a result of a serial programming and the lower figure shows a result of a parallel programming. Both results are in good agreement.



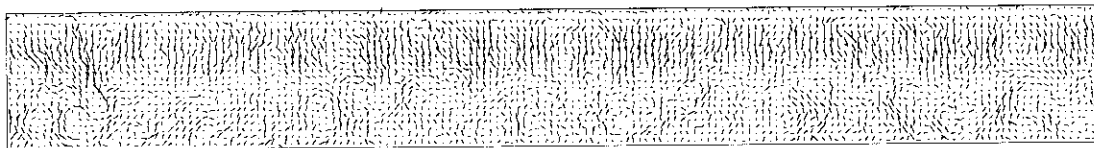
(a) Reflecting boundary conditons



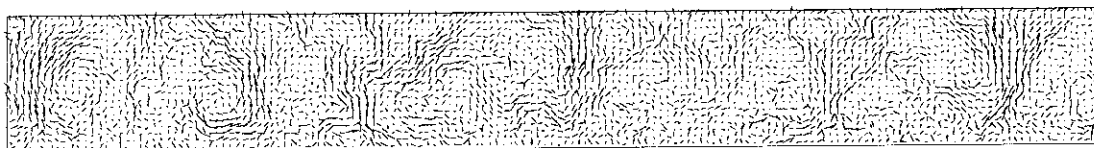
(b) Cyclic boundary conditions

Fig. 4 Evolved velocity fields of the two dimensional Rayleigh-Bénard flow with aspect ratio of four. The upper figure shows a result of the reflecting boudary conditions and the lower figure shows a result of the cyclic boundary conditions. The positions of the roles are different.

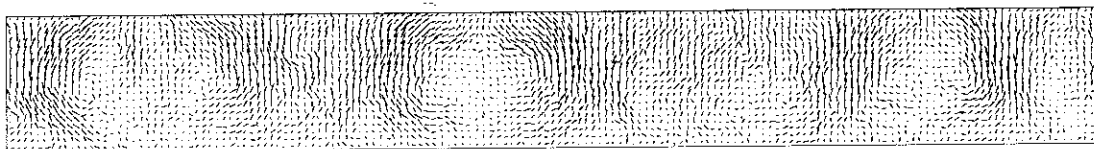
Sampling from 0 to 200 time step



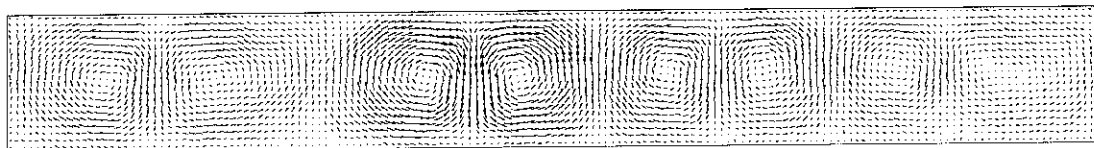
Sampling from 200 to 400 time step



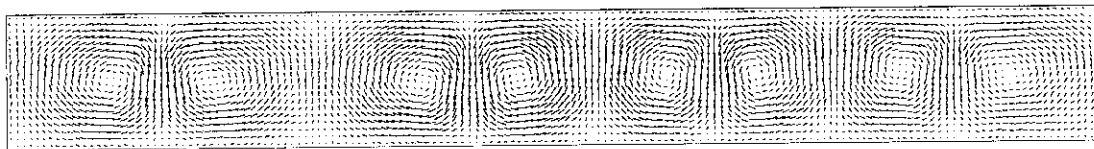
Sampling from 400 to 600 time step



Sampling from 1400 to 1600 time step



Sampling from 1800 to 2000 time step



Sampling from 9800 to 10000 time step

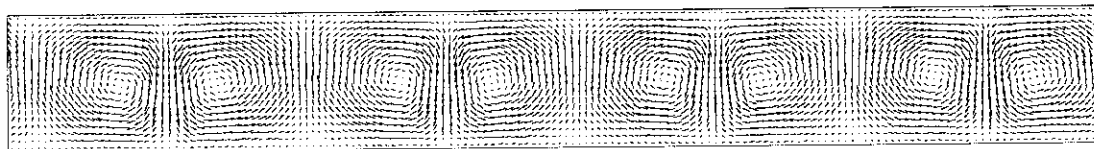
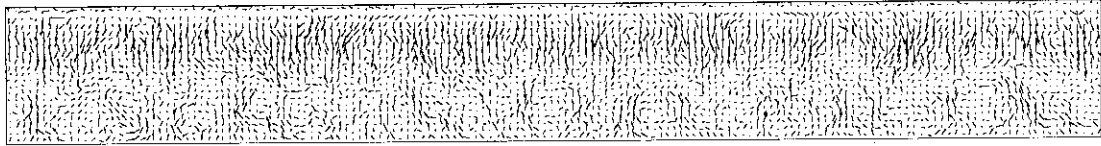
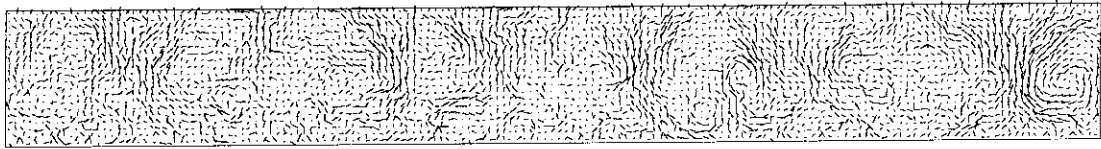


Fig. 5 Time evolution of velocity fields of the two dimensional Rayleigh-Bénard flow with aspect ratio of eight. The cyclic boundary conditions were used. Each velocity field is obtained by sampling during 200 time steps.

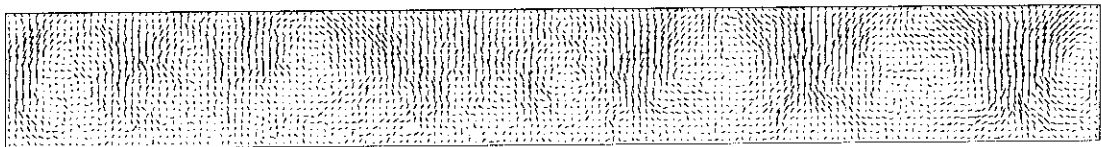
Sampling from 0 to 200 time step



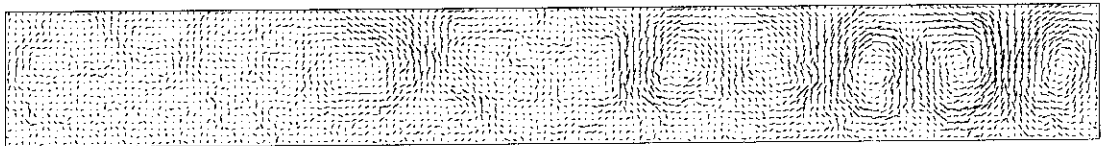
Sampling from 200 to 400 time step



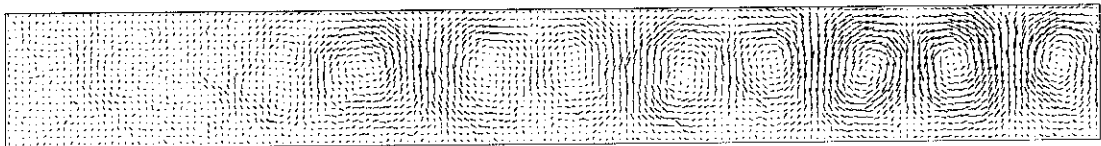
Sampling from 400 to 600 time step



Sampling from 600 to 800 time step



Sampling from 1000 to 1200 time step



Sampling from 2600 to 2800 time step

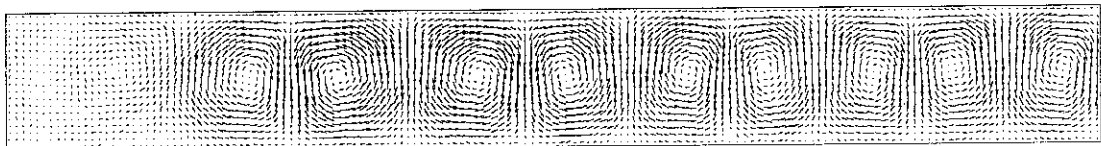
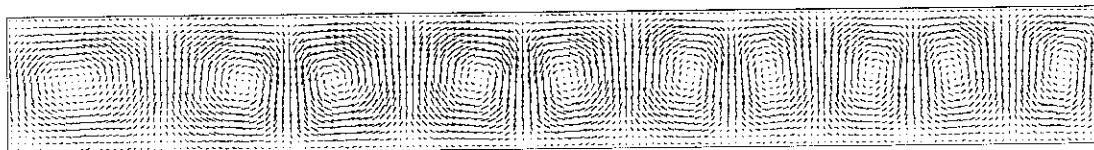
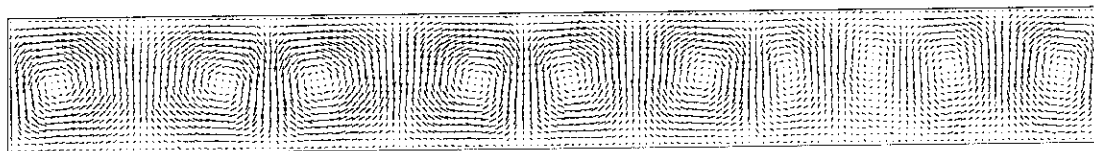


Fig. 6(a) Time evolution of velocity fields (0 to 2800 time steps) of the two dimensional Rayleigh-Bénard flow with aspect ratio of eight. The reflecting boundary conditions were used. Each velocity field is obtained by sampling during 200 time steps.

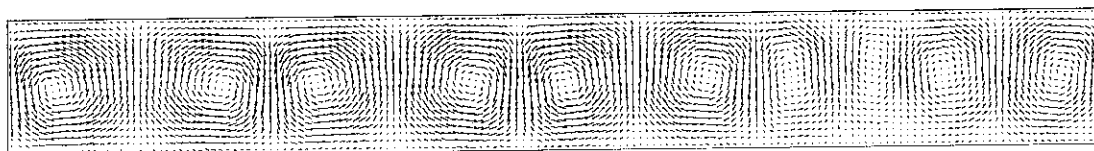
Sampling from 3800 to 4000 time step



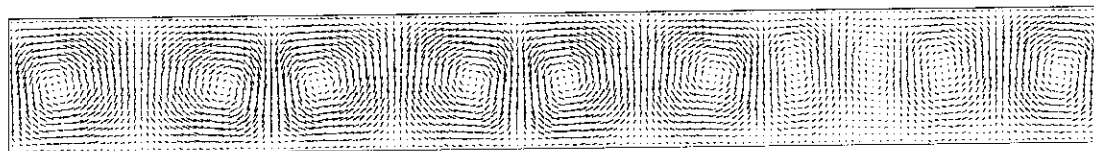
Sampling from 8800 to 9000 time step



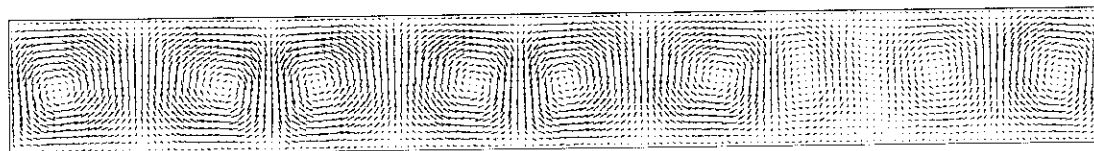
Sampling from 9400 to 9600 time step



Sampling from 9800 to 10000 time step



Sampling from 10200 to 10400 time step



Sampling from 10400 to 10600 time step

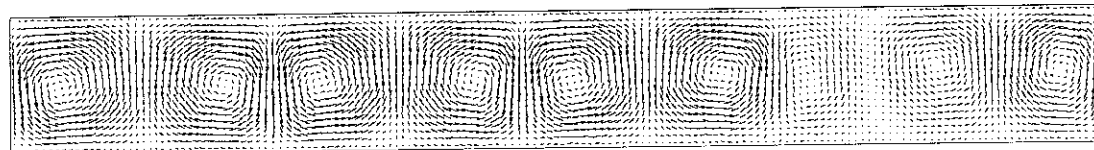
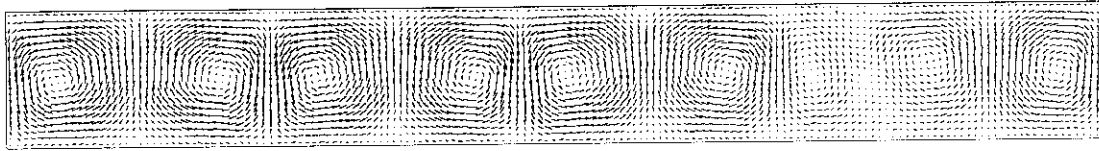
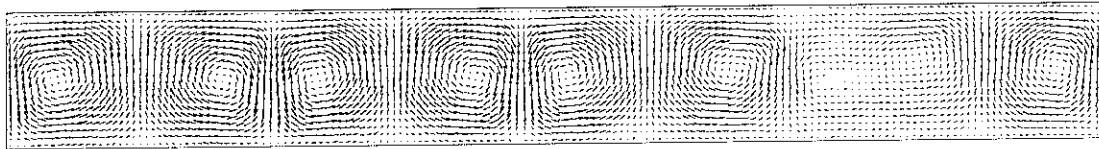


Fig. 6(b) Time evolution of velocity fields (3800 to 10600 time steps) of the two dimensional Rayleigh-Bénard flow with aspect ratio of eight. The reflecting boundary conditions were used. Each velocity field is obtained by sampling during 200 time steps.

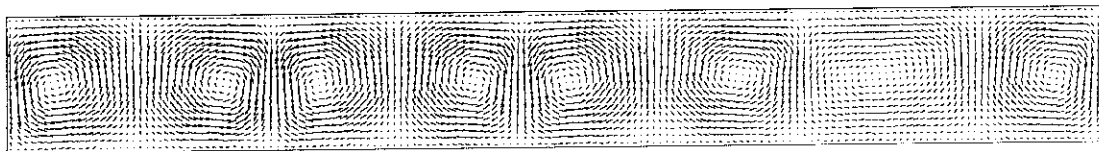
Sampling from 10600 to 10800 time step



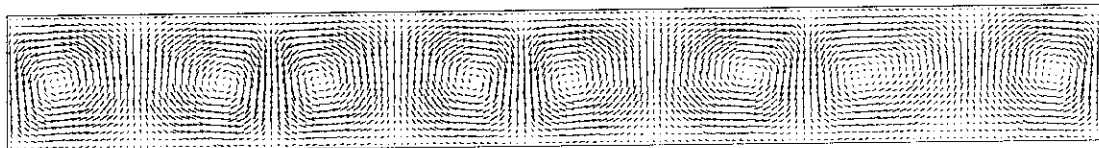
Sampling from 10800 to 11000 time step



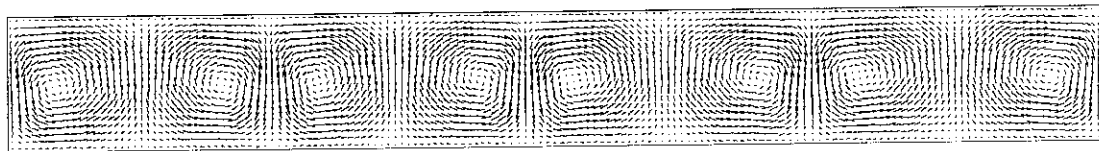
Sampling from 11000 to 11200 time step



Sampling from 11200 to 11400 time step



Sampling from 12200 to 12400 time step



Sampling from 19800 to 20000 time step

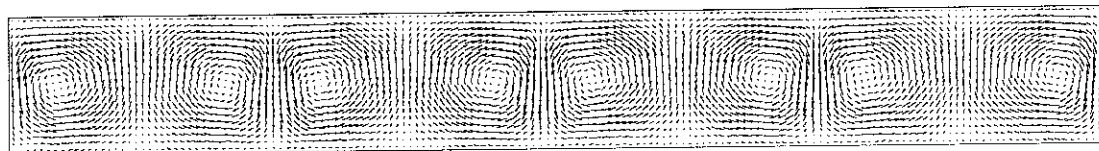


Fig. 6(c) Time evolution of velocity fields (10600 to 20000 time steps) of the two dimensional Rayleigh-Bénard flow with aspect ratio of eight. The reflecting boundary conditions were used. Each velocity field is obtained by sampling during 200 time steps.

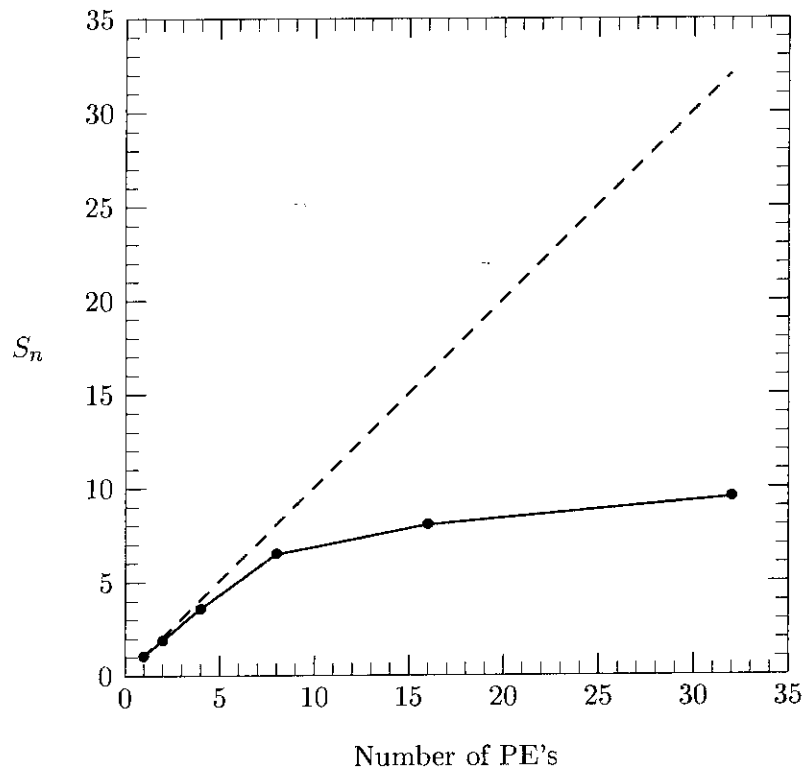


Fig. 7 Speedup ratio vs. number of PE's is shown by the solid circles and solid line. The broken line shows the theoretical maximum speedup ratio.

付録1 Pstc_2d 入力マニュアル

付録2にPstc_2dコードのデータ入力サブプログラムを示す。標準入力から設定可能な変数の初期値はフリーフォーマットのread文中のnpex、npey、irstrt、lstbgn、lstend、ran、aspect、dirの8変数のみである。Rayleigh-Bénard系を規定するその他の変数の変更はデータ入力サブルーチンdatain中の変数をエディターで直接書き換えることで行なうようになっている。以下にRayleigh-Bénard系を規定する変数の説明を行なう。

- npex : 計算領域の横方向分割数 (= 横方向 PE 数)
- npey : 計算領域の縦方向分割数 (= 縦方向 PE 数)
- irstrt : = 0。一回目のラン。
: ≥ 1。再スタートファイルからデータを読み込んで計算。
- lstbgn : 粒子情報ファイル stats.?????の開始番号。
louter×linner タイムステップ毎に+1される。
?????=00001より開始するのが望ましい。
- lstend : 粒子情報ファイル stats.?????の終了番号。
lstend=?????に達すると計算終了。
- ran : Rayleigh 数
- aspect : 計算領域のアスペクト比
- dir : 計算結果及びリスタートファイルを出力するディレクトリ名
- ny : 縦方向のサンプリングセル数
- ylen : 計算領域の縦方向の長さ (m)
- npcini : サンプリングセル内の初期粒子数
- preini : 初期圧力 (Pa)
- tmpini : 初期温度 (K)
- tmfctr : 平均自由時間に対する乗率
- louter : louter×linner タイムステップ毎に粒子情報をファイルに出力する。
- linner : 粒子情報の時間平均を求めるタイムステップ数。

付録2 Pstc_2d データ入力サブルーチン : datain

```

subroutine datain
include './cmdsmc.kis'
character parms(7)*16

c Input the data
read( 5, * ) npex, npey, irstrt, lstbgn, lstend, ran, aspect, dir
ny      = 20
nx      = ny*aspect
ylen    = 0.0056
xlen    = ylen*aspect
dx      = xlen/real(nx)
dy      = ylen/real(ny)

npcini  = 400
ntrack  = 100

preini  = 20.0
tmpini  = 80.0

tmfctr  = 0.9

louter  = 100
linner  = 2
lequil  = 0

c Output the data
write(6,6100) nx, ny, ran, aspect, lstbgn,
&          lstend, louter, linner, lequil, npcini,
&          tmpini, preini, tmfctr, irstrt
6100 format(/ / 5x,'=== ; Input Data ; ===',52('='),//
&          10x,'Number of Sampling Cells      : ',
&          i5,'*',i5//
&          10x,'Rayleigh Number                : ',f15.2/
&          10x,'Aspect Ratio                    : ',f15.2//
&          10x,'Statisticss file starting number : ',i10/
&          10x,'Statisticss file ending number  : ',i10/
&          10x,'Number of Outer Iterations     : ',i10/
&          10x,'Number of Inner Iterations     : ',i10/
&          10x,'Number of Iterations of Void Loop : ',i10//
&          10x,'Number of Particles in a Cell  : ',i10//
&          10x,'Initial Temperature [K]       : ',e15.5/
&          10x,'Initial Pressure [Pa]        : ',e15.5/
&          10x,'Time Step Factor              : ',e15.5//
&          10x,'Restart or Not ( 1 : Restart ) : ',i10//
&          5x,75('=')//)

c Data check
ngcell  = nx*ny
npe     = npex*npey

ncellp  = ngcell/npe
nomols  = ncellp*npcini

if( ncellp .gt. maxcel ) call errorp( 1001, ncellp )
if( nomols .gt. maxmol ) call errorp( 1002, nomols )
if( npe .gt. maxpe )    call errorp( 1003, npe )

return
end

```