

JAERI-Research  
98-008



量子分子動力学シミュレーションコードの並列化

1998年2月

加藤 香<sup>\*1</sup>・功刀資彰・芝原正彦<sup>\*2</sup>・小竹 進<sup>\*3</sup>

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、日本原子力研究所研究情報部研究情報課（〒319-1195 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（〒319-1195 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 Research Information Division, Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, 319-1195, Japan.

© Japan Atomic Energy Research Institute, 1998

編集兼発行 日本原子力研究所  
印 刷 いばらき印刷(株)

量子分子動力学シミュレーションコードの並列化

日本原子力研究所関西研究所光量子科学センター

加藤 香<sup>\*1</sup>・功刀 資彰・芝原 正彦<sup>\*2</sup>

小竹 進<sup>\*3</sup>

(1998年1月26日受理)

日本原子力研究所関西研究所では、物質における光熱変換機構を解析するために量子分子動力学コードを開発している。このコードをスカラー型超並列計算機Intel Paragon XP/S75とベクトル型並列計算機Fujitsu VPP300/12 上で並列化した。粒子群を分割し各プロセッサユニットへ割り付けることにより、台数分の効果を両並列計算機で得た。スカラー型超並列計算機Intel Paragon XP/Sでは、各粒子の計算を構成する演算の分割を粒子群の分割に加えて行うことにより、量子分子動力学コードの高並列化を達成した。

---

日本原子力研究所(東海駐在) : 〒319-1195 茨城県那珂郡東海村白方白根2-4

\*1 高エネルギー加速器研究機構

\*2 東洋大学

\*3 大阪大学

Parallelization of Quantum Molecular Dynamics Simulation Code

Kaori KATO<sup>\*1</sup>, Tomoaki KUNUGI, Masahiko SHIBAHARA<sup>\*2</sup> and Susumu KOTAKE<sup>\*3</sup>

Advanced Photon Research Center  
(Tokai Site)  
Kansai Research Establishment  
Japan Atomic Energy Research Institute  
Tokai-mura, Naka-gun, Ibaraki-ken

(Received January 26, 1998)

A quantum molecular dynamics simulation code has been developed for the analysis of the thermalization of photon energies in the molecule or materials in Kansai Research Establishment. The simulation code is parallelized for both Scalar massively parallel computer (Intel Paragon XP/S75) and Vector parallel computer (Fujitsu VPP300/12). Scalable speed-up has been obtained with a distribution to processor units by division of particle group in both parallel computers. As a result of distribution to processor units not only by particle group but also by the particles calculation that is constructed with fine calculations, highly parallelization performance is achieved in Intel Paragon XP/S75.

Keywords: Quantum Molecular Dynamics, Numerical Analysis, Parallelization, Scalar Massively Parallel Computer, Vector Parallel Computer, Scalability, Parallelization Performance

---

\*1 High Energy Accelerator Research Organization

\*2 Toyo University

\*3 Osaka University

## 目 次

1. はじめに .....	1
2. 量子分子動力学コードの概要 .....	2
3. 並列計算機 (Fujitsu VPP300/12, Intel Paragon XP/S75)の特徴 .....	3
3.1 Fujitsu VPP300/12 .....	3
3.2 Intel Paragon XP/S75 .....	3
4. 量子分子動力学コード並列化の方針 .....	3
5. Fujitsu VPP300/12 における量子分子動力学コードの並列化 .....	4
5.1 並列処理環境VPP Fortran .....	4
5.2 QD計算部分 .....	4
5.3 MD計算部分 .....	4
6. Intel Paragon XP/S75における量子分子動力学コードの並列化 .....	5
6.1 QD計算部分 .....	5
6.2 MD計算部分 .....	6
6.2.1 通信負荷の軽減 .....	7
6.2.2 イオン-電子間の力・ポテンシャル計算部分の並列化 .....	8
6.3 ノード空間への射影 .....	9
6.3.1 ノードグループ .....	9
6.3.2 通信の最適化 .....	9
6.4 nxライブラリ .....	10
6.5 マルチプロセッサ最適化機能の適用 .....	10
7. 並列化の結果と考察 .....	11
7.1 2機種 of 並列計算機上における粒子分割を適用した並列化 .....	11
7.2 拡張粒子分割 .....	11
7.3 大規模計算可能性の検討 .....	11
8. まとめ .....	12
参考文献 .....	12
付録 グループ内総和演算用サブルーチンの開発 .....	23

## Contents

1. Introduction .....	1
2. Quantum Molecular Dynamics (QMD) Simulation Code .....	2
3. Feature of Two Parallel Computers .....	3
3.1 Fujitsu VPP300/12 .....	3
3.2 Intel Paragon XP/S75 .....	3
4. Parallelization Scope of QMD Simulation Code .....	3
5. Parallelization for Vector Parallel Computer (VPP300) .....	4
5.1 VPP Fortran .....	4
5.2 QD Part .....	4
5.3 MD Part .....	4
6. Parallelization for Massively Parallel Computer (Paragon XP/S) .....	5
6.1 QD Part .....	5
6.2 MDPart .....	6
6.2.1 Reduction of Communication Cost .....	7
6.2.2 Parallelization for Ion-electron Force and Potential Part .....	8
6.3 Copy Parallel Execution Code to Nodes Space .....	9
6.3.1 Nodes Group .....	9
6.3.2 Optimization of Communication .....	9
6.4 nx Library .....	10
6.5 Optimization of Multi Processor .....	10
7. Result of Parallelization of QMD Simulation .....	11
7.1 Distributing Method by Particle .....	11
7.2 Distributing Method by Advanced Particle .....	11
7.3 Implementation of Supercomputing .....	11
8. Summary .....	12
References .....	12
Appendix Development of Subroutine for Summation in Nodes Group .....	23

## 1. はじめに

近年のレーザーを応用した材料加工の制御あるいはレーザー冷却などの光技術の進歩により、光と物質の相互作用に関する極短時間 (fs, ns, ps) における非定常現象の解析・解明が重要となってきている。日本原子力研究所関西研究所においても X線レーザーを含む極短パルス超高出力レーザーの開発が開始され、レーザービームを導光するミラー等の光学素子開発や将来に光量子源を材料表面改質や物質創製への応用する際の光量子-物質相互作用の解析・解明は極めて重要となっている [1]。このため、熱エネルギーレベルの現象であっても、電子、分子の運動というミクロな視点から光と物質の相互作用を理解し、光による加熱や冷却の過程を考える必要がある。分子・原子レベルでの光-物質相互作用の解析手法としては、マクロモデルとして光モンテカルロ-分子動力学法 [2] 及び、ミクロモデルとして量子分子動力学法 [3] がある。

関西研究所では、これらの光量子-物質相互作用シミュレーションコード群を開発しており、この研究のために 2 機種 of 並列計算機 (Intel Paragon XP/S75, Fujitsu VPP300/12) を導入している。Intel Paragon XP/S は 834 台のプロセッサユニット (PU) を持つ分散メモリスカラー型超並列計算機、Fujitsu VPP300/12 は 12 台の PU を持つ分散メモリベクトル型並列計算機である。並列計算機の性能を十分に引き出すためには、ベクトル型計算機のための計算手法 [4] が開発されたのと同様に、並列計算機に適した計算手法ならびに並列化アルゴリズムの開発が重要な課題である [5]。すでに我々は、光量子-物質相互作用シミュレーションコードである、光モンテカルロ-分子動力学法 [6] とプラズマ粒子シミュレーションの並列化を行った。

本論文では、よりミクロな立場で光と物質の熱的相互作用解析のための量子分子動力学シミュレーションの並列化について報告する。第 2 章で、量子分子動力学コードについて説明する。第 3 章では 2 機種 of 並列計算機の特徴、第 4 章では並列化の方針、第 5、6 章では各計算機での並列化、第 7 章では並列化の結果を述べる。第 8 章は、本論文で提案した並列化手法の有用性について総括する。

## 2. 量子分子動力学コードの概要

芝原, 小竹らは, 光照射による物質の光熱変換機構を解析するための量子分子動力学コードを開発した[7]. このシミュレーションでは, 計算モデル系を量子力学的な部分と古典力学的な部分に分けて取り扱う. すなわち, 光が作用する物質系を構成する原子を, イオン(原子)と最外殻の電子に分けて取り扱い, 電子の運動は量子力学的に取り扱うための時間依存のシュレディンガー方程式を解き, イオンの運動は質点粒子としてニュートン方程式を解く. 光の作用は, シュレディンガー方程式, ニュートン方程式に含まれるポテンシャル場の光電場による変化としてモデル化している. すなわち, 電子の状態変化の時間スケールはイオンの時間スケールに比べ非常に短いため, 電子の状態変化においてはイオンの状態は変化せず, イオンの状態変化に対しては電子状態の平均値が寄与するものとする. ただし, 系全体の時間に依存するシュレディンガー方程式は様々な仮定, 近似を用いて導出されており, その適用範囲・条件について十分な注意が必要である. 以降では, 自由電子に関する量子力学計算部分をQD計算部分, イオンに関する分子動力学計算部分をMD計算部分と記し, その概要を手短に述べる.

### (1) QD計算部分

各電子の運動は, 時間依存のシュレディンガー方程式で記述され, Split operator法[8], 高速フーリエ変換法(FFT)を組み合わせた手法で計算される. 電子は $32 \times 32$ の2次元格子上的波動関数として記述する. 各電子の状態変化は, 電子-電子間ポテンシャルおよび電子-イオン間ポテンシャルから計算される. 各電子の状態変化を計算し, 全電子状態の平均値を求める. この情報をMD計算部分に渡す.

### (2) MD計算部分

各イオンの運動は, 質点運動として記述されており, ある時間ステップにおけるイオンの位置・速度の変化は, 光の電場, QD計算部分で求められた電子状態の平均値から計算される. 各イオンの位置・速度から, イオン-電子間のポテンシャルを求め, QD計算部分に渡す. この場合, イオンの速度変化をなめらかにするためにQD計算で格子間補間を行っている.

### (3) プログラムの構造

プログラムの構造を Fig.1 に示す. 1時間ステップの計算は, 複数回繰り返されるQD計算と1回のMD計算の逐次処理から構成される. 1回のQD計算は, 電子(波動関数)の識別番号に関するループ, 波動関数のグリッドに関するループから構成される. 1回のMD計算は, イオンの識別番号に関するループ, 波動関数の補間グリッドに関するループから構成される.

### (4) 計算所要時間と必要メモリ量

典型的な問題として7原子からなる系の500時間ステップを計算する. Table.1にこの場合の計算パラメータを示す. 1時間ステップは, 20回のQD計算・1回のMD計算より構成される. Paragon XP/S (1PU)での逐次実行では, 計算所要時間は約84時間, 必要メモリは28.7MBである. VPP 300 (1PU)での逐次実行では, 計算所要時間は約20分, 必要メモリは134.7MBである. 計算に必要なメモリ量の違いは, プログラムがベクトル計算仕様か, スカラー計算仕様かの違いによる. 1PU上での逐次実行では, VPP300の方が約241倍高速である.



### 3. 並列計算機 (Fujitsu VPP300/12, Intel Paragon XP/S75) の特徴

本量子分子動力学コードの並列化のために使用した, 2機種 of 並列計算機 (Fujitsu VPP300/12, Intel Paragon XP/S75) の主な特徴を述べる. この2機種はともに, 各PUが独立にメモリを持つ分散メモリ型並列計算機である. 今回の並列計算での各計算機のピーク性能は, Intel Paragon XP/S では, 最大224 PU を使用するため, 34.6 GFLOPS (=75 MFLOPS×2×224 PU), Fujitsu VPP300 では最大7PUを使用するため, 15.4 GFLOPS (=2.2 GFLOPS×7 PU) である. 以降は, VPP300 を構成するPUをPE, Paragon XP/S を構成するPUをノードと記す.

#### 3.1 Fujitsu VPP300/12

ピーク性能 2.2GFLOPS のベクトル型プロセッサを搭載し, PEの最高並列度は 12 である. 1 PEの主記憶容量は512MB (システム使用分を除く主記憶容量384MB), システム全体の主記憶容量は6GB (=512MB×12, システム使用分を除く主記憶容量4.6GB=384MB×12) である. 並列処理環境として VPP Fortran を備えている.

#### 3.2 Intel Paragon XP/S75

ピーク性能 75MFLOPS のスカラー型プロセッサを搭載し, ノードの最高並列度は 834 である. また1ノードは, 2個の計算用プロセッサと1個の通信用プロセッサから構成される. 1ノードの主記憶容量は 128MB, システム全体の主記憶容量は 100GB (=128MB×800) である. 並列処理環境として, nx ライブラリを備えている.

## 4. 量子分子動力学コード並列化の方針

量子分子動力学コードを2機種 of 並列計算機 Fujitsu VPP300/12, Intel Paragon XP/S75 上で並列化する. 並列化による高速化は, 演算の分割によって1PUの演算負荷を小さくすることである. このとき分割された演算が, もとのプログラム演算に占める割合を  $R$ , 使用PU数を  $PU$  とすれば, 並列プログラムの性能向上度  $\alpha$  ( $\alpha$  倍) は, アムダールの法則を用いれば次のように求められる [9].

$$\alpha = \frac{1}{1 - R + \frac{R}{PU} + \beta}$$

ここで,  $\beta$  は並列化によって新たに生じる, 通信や同期待ち合わせ等の負荷を示す.

この法則は, 元のプログラムに対して, 例えば逐次計算部分が1%残された ( $1-R=0.01$ ) 場合, もしくは並列化によって新たに通信や待ち等の負荷が 1% 発生した場合,  $\alpha < 100$  となり, このプログラムに対してPUをどれほど投資しても, 最大 100 倍程度の性能向上しか見込めないことを示している. また, 分散メモリ型並列計算機での並列化に伴い発生するPU間通信が負荷となることを示している.

すなわち, 2機種が持つ共通の課題は, (1) 各PUが持つローカルメモリ上へのデータアロケーション, (2) PU間通信の最適化, (3) 逐次実行部分の最小化の3点である. また, 2機種 of ハードウェア構成の違いより, VPP300 ではベクトル並列化, Paragon XP/S では超並列化を目標とする.

## 5. Fujitsu VPP300/12 における量子分子動力学コードの並列化

VPP300 はベクトル型の並列計算機であり、プログラムのベクトル長の保持が特に重要となる。また、VPP300/12 は最大並列度が8と低いため、プログラムを構成する複数演算をある程度まとめて、これを複数PEで並列に処理する。本コードのベクトル化のポイントは、QD 計算部分における1電子波動関数のポテンシャル項の計算にある。各粒子を独立に計算しても、このポテンシャル項計算部分のベクトル長は保持される。このため、粒子分割を適用したベクトル並列化を行う。また、並列計算においては逐次処理部分の最小化も重要である。MD計算部分とQD計算部分は逐次処理であるため、QD部分の並列化だけではMD計算部分は逐次処理として残される。したがって、MD計算部分も粒子分割による並列化を行う。次に、使用する並列処理環境 (VPP Fortran), QD計算部分, MD計算部分について述べる。なお、粒子分割を適用した並列プログラムの構造を Fig.2 に示す。

### 5.1 並列処理環境 VPP Fortran

VPP Fortran[10] を使用して、1 PE が処理するプログラムに複製指示を追加した並列実行用プログラムを作成する。processor 文を用いた使用 PE 数の定義, parallel region 文を用いたリージョンの定義を行う。このプログラムを起動すると、指定されたリージョンのPE数個の複製と、各複製の各PE上への割り当てが行われ、各リージョンが並列実行される。このとき、各リージョンの計算する粒子数を1, リージョン数を粒子数, 各リージョン識別番号と電子識別番号を一致させれば、本コードの粒子分割による並列実行が可能である。ただし1 PEが更新するのは1粒子分のデータであるから、他の粒子に関する情報が必要な場合はPE間通信を行う。なお、この1粒子分のデータは重複ローカル配列として各PEにアロケートさせ、各粒子に関する情報の通信には各PEに用意した作業用重複ローカル配列と spread do 文を用いた総和演算を利用する。また、リージョンの識別番号 (粒子識別番号) は、システムコール idvproc() から取得するPE識別番号に一致させる。

### 5.2 QD 計算部分

QD 計算部分では各電子の状態変化を計算し、全電子状態の平均値を求める。1時間ステップを構成する演算で最も負荷の高い部分は、電子の状態変化の計算である (Fig.1 QD part 参照) ため、この部分の並列化を行う。1電子に関する一連の演算を1リージョンと定義する。このとき、各リージョンの演算負荷は同じで、同数のリージョンを各PEに分散させることで演算負荷バランスを均一化できる。QD 計算部分で使用される配列は、配列添字に電子識別番号を持つ配列を基本とする。これらの配列を電子識別番号ごとに分割し、各PEのローカルメモリにアロケートする。電子と他の粒子の相互作用ポテンシャルの計算部分は他の粒子の情報を必要とするため、PE間通信を行い各PEのローカルメモリに全粒子の情報を用意する。

### 5.3 MD 計算部分

MD 計算部分では、各イオンの時間ステップにおける位置・速度の計算、イオン-電子間のポテンシャルの計算を行う。本計算部分では、各イオンに関する計算を並列化する。1イオンに関する計算を1リージョンに選び、各リージョン識別番号はイオン識別番号と同じにとる。本計算部分で使用される配列は、配列添字にイオンの番号を持つ。これらの配列をイオンごとに分割して、各PEのローカルメモリにアロケートする。イオン-電子間のポテンシャル計算部分は全電子の情報を必要とするが、QD 計算部分の並列化によって各電子の情報は分割されているため、各PEのローカルメモリに全電子の情報を用意するにはPE間通信が必要となる。MD部分の通信負荷の軽減に関しては、6章に記述する。

## 6. Intel Paragon XP/S75 における量子分子動力学コードの並列化

Paragon XP/S はスカラー型超並列計算機である。一般に超並列計算機での並列化では、プログラムから高い並列性を抽出することが特に重要となる。ここでは、7原子系を対象とした量子分子動力学コードの Paragon XP/S 上での並列化、すなわち粒子数に対してノード数が十分多数である場合の並列化手法を検討する。

本コードでは、1粒子の計算が、高負荷の複数演算から構成されること、内部に分岐を持たないことに着目し、粒子分割に加えて各粒子の演算をさらに並列化する。以降、この手法を「拡張粒子分割」と記す。粒子分割に関しては、VPP300 での並列化と同様の方法を用いる。拡張粒子分割においては、ノードをグループ分けし、グループごとの通信制御が必要となる。この通信を最適化するために、使用するノード集合に仮想的な配置を与え、総和演算用のライブラリの開発を行う。各粒子の演算をさらに並列化すると、演算負荷は小さくなりまた新たな通信が発生し、相対的に通信負荷が大きくなる。このため、並列化後の演算負荷と通信負荷との見積もりを行い、通信負荷の増加により高速化が望めない場合は逐次実行を行う。また Paragon XP/S は1ノードに計算専用の2CPUを搭載している。この2CPUの同時使用、すなわち自動並列化も試みる。

ここで、以降での説明に使う用語の定義を行う。ある一連の演算を「プロセス」、プロセスを構成する演算を「サブプロセス」と定義する。また、プロセス・サブプロセス同士が行う情報交換を「コミュニケーション」と記す。逐次実行において直列に並んでいたプロセス・サブプロセスを分解し並列に並べ替えることを「演算の再構成」と記す。再構成した演算をノード集合へ割り当てることを「再構成した演算のノード空間への射影」と記す。以降、演算の再構成について、QD 計算部分、MD 計算部分、再構成された演算のノード空間への射影、Paragon XP/S75 における並列処理環境 (nx ライブラリの使用)、1ノードに搭載される計算専用の 2CPU の同時使用の順に述べる。再構成したプログラムの構造を Fig.3 に示す。

### 6.1 QD 計算部分

Paragon XP/S での並列化においても、VPP300 と同様に各電子を並列に計算する。同時に、各電子 (波動関数) 計算を構成する演算も並列化する。1電子の計算において、最も負荷の高い演算は、各電子が他の電子から受けるクーロンポテンシャルの計算である。ここでは、QD 計算部分を再構成して拡張粒子分割を行うため、(1)プログラムの構造、(2)使用される配列のデータ構造、(3)サブプロセスの定義、(4)各ノードへのデータアロケーションを決定、(5)通信負荷の軽減、(6)FFT計算部分の逐次実行について述べる。

#### (1) プログラムの構造

1電子 (ie) に関するクーロンポテンシャルの計算は次のように行われる。「電子 (ie) と他の1電子 (ie' ≠ ie) のグリッド間のクーロン力  $eecl02(2n-1, 2n-1, 14)$ 」と「ie' ≠ ieの電子存在密度  $vclo2(n, n, ie')$ 」の積が、ie, ie'間のポテンシャルである。このポテンシャルをie'に関して足し込む計算を繰り返し、1電子 (ie) が他の電子から受けるクーロンポテンシャル  $vwcons(n, n, ie)$  を求める。

#### (2) 使用される配列のデータ構造

使用される配列は、配列添字に波数空間グリッドの格子点番号を持つ  $n \times n$  の2次元配列 ( $n=32$ ) を基本としている。ところが、グリッド間クーロンポテンシャルを表す配列のデータ構造は、 $eecl02(2n-1, 2n-1, 14)$  と例外的な形である。この配列は時間ステップで更新されないが、参照される要素は時間ステップに依存する。この配列を分割すると、配列要素のダイナミックアロケーションは非常に複雑なものとなる。したがって、この配列の分割は行わない。

## (3) サブプロセスの定義

各 $ie$  に関する計算をプロセス  $ie$  が  $ie'$  の部分空間から受けるクーロンポテンシャル作成を1サブプロセスとする。

## (4) 各ノードへのデータアロケーション

サブプロセス入力的全配列要素  $eeclo2, vc1ot$  を各ノードのローカルメモリに重複してアロケートする。このようにすれば、サブプロセスは自分が起動されたノードのローカルメモリから任意に入力を抽出でき、配列  $eeclo2$  のダイナミックアロケーションを避けることができる。各サブプロセスの演算負荷が同一であるため、同数のサブプロセスを各ノードに分散させ、各ノードの演算負荷を均一にする。

## (5) 通信負荷の軽減

サブプロセス出力は、 $ie, ie'$  間のクーロンポテンシャルであり、遠隔相互作用が重要であるため、 $ie$  が  $ie'$  の全空間から受けるポテンシャルを作成する必要がある。この通信の負荷を最小限にするため、先に  $ie'$  を部分空間に分割したまま全ての  $ie'$  に関するポテンシャル計算を行う。次にこの部分空間からのポテンシャルを全空間からのものに合成する。

## (6) FFT計算部分

フーリエ変換部分は逐次実行部分として残す。これは他の優先的に並列化を行うプロセスがフーリエ変換される配列の分割を要請しないこと、および演算負荷が小さいため通信負荷との和の見積もりからは並列化による高速化が期待できないことの2つの理由による。

Paragon XP/Sでは、1回の通信データ量が500KB以下である場合は通信効率が低い。FFT計算部分の並列化を行うと、変換方向の変更時に通信が必要となる。本コードでは各FFTサブルーチン内で大きさ50KBの配列を2つ使用する。配列2つを一度に通信しても最大100KBである。このため、データ分割方法の工夫による通信効率の差はそれほどないと予想される。

## 6.2 MD 計算部分

VPP300での並列化と同様に各イオンを並列に計算し、さらに各イオンを構成する演算も並列化する。またMD計算部分において、最も負荷の高い演算は、イオン-電子間の力・ポテンシャル計算部分である。また、通信負荷の軽減が必要である。ここでは、通信負荷の軽減および、拡張粒子分割のために行うMD計算部分再構成についてイオン-電子間のポテンシャル計算部分の並列化を述べる。

### 6.2.1 通信負荷の軽減

イオンごとに異なる3つの総和値（1イオン-全電子間のクーロン引力のx方向成分、およびy方向成分、クーロンポテンシャルの大きさ）の作成には、「補間グリッド上の全電子の存在密度」と「各イオンの位置」が必要である。QD計算部分が並列化されたため、電子の存在密度を表す配列は、分割されて各ノードにアロケートされている。この総和値作成時に、各ノードが「補間グリッド上の全電子の存在密度」をダイナミックアロケーションするため発生する通信は、もっとも負荷の高い通信となる。（1）プログラムの構造解析を行い、通信負荷を軽減するために、イオン-全電子間の力・ポテンシャル計算部分の（2）データアロケーションの変更，（3）プログラム構造の変更を行う。

#### (1) プログラムの構造解析

イオン (io) が補間グリッド上の電子存在密度から受ける力・ポテンシャルは、力を例に取れば、次のように求められる。ポテンシャルに関しても同様である。

1イオン (io) が1電子 (ie) から受ける力は、1電子の補間グリッド各点上の存在密度  $vaf(ix, iy, ie)$  から受ける力を、全グリッド点 ( $ix=1, 311, iy=1, 311$ ) に関して加算したものである。この力を全電子 ( $ie=1, 7$ ) に関して総和して、1イオンが全電子から受ける力を決定する。すなわち、イオンが全電子から受ける力・ポテンシャルの計算には、各イオンごとに全電子に関する情報が必要である。

#### (2) データアロケーション

「全イオンの位置」を重複し、「補間グリッド上の各電子存在密度」を分割して、各ノードにアロケートしておく。本コードでは、「補間グリッド上の各電子の存在密度」は、時間ステップで更新される大きさ5.2MBの3配列であり、グリッド補間計算部分でのみ参照される配列である。「イオンの位置」は、時間ステップで更新される大きさ56Bの2配列であり、「部分和」は倍精度実数型の3変数である。したがって「補間グリッド上の各電子の存在密度」の通信よりも、「部分和」の交換、「イオンの位置」の更新のほうが回数は増えるが負荷は小さい。

#### (3) プログラム構造の変更

QD計算部分の並列化により、電子の存在密度を表す配列は分割された。全電子の存在密度から、イオンが受ける力・ポテンシャルを次のように計算し、電子の存在密度を表す配列のダイナミックアロケーションを回避する。

各ノード上で、1電子の存在密度から1イオンが受ける力・ポテンシャルを並列に計算する。この結果、各ノード上には1イオンが各電子の存在密度から受ける力・ポテンシャル、すなわち1イオンが全電子の存在密度から受ける力・ポテンシャルの部分和が作成される。したがって、この部分和を総和して、1イオンが全電子の存在密度から受ける力・ポテンシャルを求めればよい。

## 6.2.2 イオン-電子間の力・ポテンシャル計算部分の並列化

MD 計算部分において最も負荷の高い演算は、イオンごとに異なる3つの総和値（1イオン-全電子間のクーロン引力 $x$ 方向成分および $y$ 方向成分、クーロンポテンシャルの大きさ）の計算である。ここでは、この総和演算をさらに並列化する。特に、力・ポテンシャル計算部分の並列化検討として（1）サブプロセスの定義、（2）使用される配列のデータ構造、（3）各ノードへのデータアロケーションと通信負荷の軽減について述べる。

### （1）サブプロセスの定義

「補間グリッド上の1電子（ie）の存在密度と1イオン（io）間の力・ポテンシャル」は、補間グリッド各点上の1電子  $vaf(ix, iy, ie)$  の存在密度と io 間の力を、全グリッド点に関して総和したものである。ix 方向の総和演算を1サブプロセスとして、補間グリッド2方向（ix, iy）の総和演算を iy 方向に分解する。このようにサブプロセスをとると、プロセス数は iy 方向の総グリッド数となる。各サブプロセス識別番号はグリッド識別番号と同じとする。各プロセスの演算負荷は同一であるがプロセス数が総グリッド数となるため、各サブプロセスを循環的に各ノードに割り当てることで演算負荷の均一化を図る。

### （2）使用される配列のデータ構造

この計算で使用される各電子の存在密度を表す配列は、配列添字に補間後の $x$ 方向、 $y$ 方向総グリッド数を持つ  $((n-1)*9+n) \times ((n-1)*9+n)$  の2次元配列であり、時間ステップごとに更新される。ここで、サブプロセスの割り当てを循環的に行えば、「補間グリッド上の各電子の存在密度」の配列要素は、補間グリッド（ix, iy）の iy に関して循環的に抽出できる。

### （3）各ノードへのデータアロケーションと通信負荷の軽減

本並列化ではこのサブプロセスが、自分が起動されたノードのローカルメモリから任意に入力を抽出できるように、「補間グリッド上の各電子の存在密度」の各電子の全空間に関するデータを、各ノードのローカルメモリに重複してアロケートする。

またサブプロセス出力、すなわち $y$ 方向に分割された部分空間での総和値から、全空間での総和値を作成する必要がある。通信回数を最小限にするため、 $y$ 方向の部分和のまま $x$ 方向の総和演算を進め、この計算が終了した後全空間における総和値を作成する。

### 6.3 ノード空間への射影

再構成された演算をノード空間へ射影する。このときのプロセス・サブプロセスと、これらが起動されるノード間通信の並列化検討として、1ノードグループおよび、通信の最適化について述べる。再構成した演算のノード空間への射影を Fig.4 に示す。

#### 6.3.1 ノードグループ

1粒子の計算を並列実行するために、使用する全ノードをグループ分けする。特定のノードで構成されるノード集合を「グループ」とし、粒子識別番号をグループ識別番号に一致させ、プロセス（1粒子に関する計算プログラム）を1グループ上で実行する。各サブプロセスは、グループ内の各ノードで並列実行する。

#### 6.3.2 通信の最適化

サブプロセス出力は、プロセス出力の部分集合である。つまり、1サブプロセスは、1粒子に関する部分的なデータを更新する。1粒子に関する全データが必要な場合は、サブプロセス間コミュニケーションが必要となる。またプロセス出力は、プログラム出力の部分集合である。つまり、1プロセスは、1粒子に関するデータを更新する。全粒子に関するデータが必要な場合は、プロセス間コミュニケーションが必要である。このようなプロセス・サブプロセス間のコミュニケーションを効率的に行うために、ノード集合に仮想的に2次元の配置を与え、これに再構成した演算を射影する手法として（1）ノードの仮想的な配置、（2）再構成した演算のノード空間への射影、（3）コミュニケーションの規則性、（4）通信時におけるノード動作の決定について述べる。

##### （1）ノードの仮想的な配置

ここで、プロセス数を  $P$ 、サブプロセス数を  $SP$  と表す。  $SP \times P$  個のノードを用意する。各ノードを  $x$  方向に  $SP$  個、  $y$  方向に  $P$  個並べ、位置を2つの座標値  $(x, y)$  で示す。このようにして、  $SP \times P$  個のノードに2次元の仮想的な配置を与える。

##### （2）再構成した演算のノード空間への射影

各ノードの  $y$  座標値とプロセスの識別番号、各ノードの  $x$  座標値とサブプロセス識別番号が一致するように、再構成したプログラムをこのノード空間に射影する。

##### （3）コミュニケーションの規則性

サブプロセス間コミュニケーションは、ノード間の  $x$  方向通信となる。この  $x$  方向通信の動作、つまり同じグループ内のノードのコミュニケーションは、1粒子に関する部分的なデータから全データの作成を行うためである。

1粒子に関する計算は、グループ内で並列実行される場合と冗長実行される場合がある。冗長実行時には、グループ内の全ノードが1粒子に関するデータを重複して所有する必要がある。このため、グループ内のノードが作成した1粒子に関する全データは、グループ内の全ノードに返還する。すなわち、  $x$  方向通信の動作を「  $y$  座標値の等しいノードがアロケートするデータは、同じ内容に更新される」ように定める。このようにすれば、プロセス間コミュニケーションは  $x$  座標値の等しいノード間の  $y$  方向通信とできる。

$y$  方向通信の場合も、「  $x$  座標値の等しいノードがアロケートするデータは、同じ内容に更新される」ようにすれば、時間ステップの更新時には、全ノードがアロケートする情報は同じ内容に更新される。このように通信の動作を定めれば、時間ステップ・識別番号によらずプロセス間のコミュニケーションは  $x$  方向、サブプロセス間のコミュニケーションは  $y$  方向となる。

#### (4) 通信時におけるノード動作の決定

部分集合から集合を作成するもしくは、部分和から総和を作成する際に、各方向の通信が発生する。ここでは、部分集合を部分和の形に置き直し、いずれの場合も総和演算を採用する。このようにして、x方向通信では「y座標の等しいノード集合をグループとしたグループ内総和演算」、y方向通信では「x座標の等しいノード集合をグループとしたグループ内総和演算」を行う。また、作成した総和値が部分和を上書きするように、部分和を提供したノードに返還する。

### 6.4 nx ライブラリ

FORTRAN77 による計算内容の記述に、nx ライブラリ[11]の fortran system call によるノード動作の記述（主にノード間通信の指示）を追加して並列実行用プログラムを作成する。nx ライブラリには使用するノード集合をグループに分けるためのシステムコール、グループ内総和演算を行うためのシステムコールが用意されていない。このため、プログラムにグループ数とグループ内ノード数を指定するパラメータを与え、この2つのパラメータとシステム全体のノード識別番号から、グループ識別番号、グループ内ノード識別番号を検出する。また、ノード間通信のためにグループ内総和演算用サブルーチンを開発する。開発したサブルーチンの動作・プログラムリストについては Appendix A に記した。

### 6.5 マルチプロセッサ最適化機能の適用

本コードに関しては、マルチプロセッサ最適化機能[12]の適用によって、本コードで最も高負荷である波動関数のポテンシャル項計算部分が約2倍高速化できた。よって、本並列化ではこの機能を利用する。高速化の理由は、この手続きでのキャッシュヒット率の向上であると考えられる。これは、ポテンシャル項計算部分のループ分割がなされていないこと、使用キャッシュサイズを半分に指定した場合の実行時間が、使用キャッシュサイズを最大に指定した場合より遅いことから推測される。



## 7. 並列化の結果と考察

Table.1 に示したように、7原子の系、1時間ステップが20回のQD計算・1回のMD計算より構成される500時間ステップの並列処理を行った。このときVPP300では7PE、Paragon XP/Sでは最大224ノードを使用して、PUの増加とともに計算所要時間がどれだけ短くなるか、その並列化による加速率を測定した。

### 7.1 2機種種の並列計算機上における粒子分割を適用した並列化

VPP300とParagon XP/S上で、粒子分割を行った結果を比較すると、VPP300でのMD部分の通信負荷が非常に高い。VPP300では、MD部分のベクトル化がなされており、スカラー計算実行時よりも演算負荷が小さい。このため並列実行時に発生する通信負荷が無視できない大きさになる。しかしこの通信負荷は、Paragon XP/Sでの拡張粒子分割と同様の方法による軽減が可能であると予想される。

VPP300での粒子分割による並列化前後の計算所要時間、計算所要時間の比をTable.2に示す。負荷内訳をFig.5に示す。Paragon XP/Sでの粒子分割による並列化前後の計算所要時間、計算所要時間の比をTable.3に示す。負荷内訳をFig.6に示す。

### 7.2 拡張粒子分割

#### (1) 使用ノード数と、実行時間・加速率の関係

使用ノード数と、実行時間・加速率の関係をFig.7に示す。拡張粒子分割法も56並列程度までは並列化の効率は良好であるが、112並列程度ではほぼ飽和する。

#### (2) 拡張粒子分割における負荷内訳

拡張粒子分割における負荷内訳をFig.8に示す。性能飽和の原因は、逐次処理部分であることがわかる。通信負荷が小さいことから、この手法に対するノード集合への仮想的な配置の導入、開発したライブラリは有効であった。

### 7.3 大規模計算可能性の検討

量子分子動力学シミュレーションで行われている計算は、原子(電子)が数個から10数個の系を対象としている。粒子数は、使用する計算機による制約(計算時間・メモリ量)が大きき理由である。並列化を行うことで、より高速で大容量メモリを必要とする計算が可能となった。VPP300、Paragon XP/Sでの粒子分割による並列計算の結果から、(1)系の粒子数の増加と必要メモリ量、(2)計算時間の見積もりを行う。

#### (1) 粒子数と必要メモリ量の見積もり

粒子分割で並列化を行った場合、計算可能な粒子数と計算に必要なメモリ量についての見積もりをFig.9にまとめる。これより、VPP300では1000程度までの計算が可能となり、Paragon XP/Sでは10000程度の計算が可能となる。

#### (2) 粒子数と計算時間の見積もり

粒子分割時のプログラム性能の実測結果から、粒子分割で並列化を行う場合の、粒子数と計算所要時間についての見積もりをFig.10にまとめる。本コードにおける計算時間は、粒子間相互作用の計算を全粒子について行うので、粒子数 $N$ の2乗に比例する。このため、粒子数の増加とともに計算時間は膨大に増加する。しかし、近接粒子間の力を正確に計算し、遠くの粒子間の相互作用を適当に近似することで、全粒子に関する計算の必要はなくなる。このため、時間の増加スケールを、粒子数 $N < 100$ までは $N$ の2乗、 $N \geq 100$ からは $10 \cdot N$ と見積もった。

## 8. まとめ

現在の量子分子動力学シミュレーションは、数個から数10個の原子を対象としており、粒子を独立に計算しただけでは並列度の高い処理ができない。そこで本報告では、このようなシミュレーションを高速化するための並列化手法を提案した。

ベクトル計算仕様プログラムに対しては、並列実行時においてもそのベクトル長を保持できるように、粒子分割を適用したベクトル並列化を行った。また、粒子数に対してPU数が十分多数である場合には、ノードグループの導入、グループごとの通信制御を行い、1粒子を構成する演算（サブプロセス）まで並列化を行う手法を開発した。この手法を用いることによって、プログラムから粒子分割以上の並列性を抽出し、並列度の高い処理を行うことが可能となった。プログラム性能の実測結果から、通信負荷の最小化は十分であることが示された。これは、PU集合への仮想的な配置の導入、開発したライブラリの有用性を示しているものと考えられる。

また、実際に VPP300, および Paragon XP/S の2機種 of 並列計算機上での並列化手法として、粒子分割を適用した場合のプログラム性能の実測結果から計算所要時間の見積もりと、計算可能となる粒子数の見積もりを行った。本シミュレーションでの1粒子に関する計算は、ベクトル化が可能であり最も高速に計算できている。このため、少数粒子を対象とする場合はベクトル並列化による計算が適しているといえる。また、粒子数が10000程度の大規模計算を行う場合は、システムとして巨大メモリを有する Paragon XP/S での粒子分割による並列処理が適している。Paragon XP/Sでのピーク性能での計算が可能となれば、10000粒子500ステップ計算の時間（見積もり125時間程度）が、8時間程度で計算できる可能性がある。これは現実の大規模計算に十分耐えうる性能であると考えられる。

## 参考文献

- [1] 谷啓二：プラズマ核融合学会誌 **72**, 935 (1996)。
- [2] 功刀資彰他：機論B. **61**, 1826 (1995)。
- [3] S. Kotake, Thermal Sci. and Eng., vol3, no.3, 21 (1995)
- [4] 日本物理学会編：スーパーコンピュータ, 培風館 (1985); 島崎眞昭：スーパーコンピュータとプログラミング, 共立出版 (1989)。
- [5] 徳田伸二：プラズマ核融合学会誌 **72**, 916 (1996); M.L. Norman, Physics Today vol, 42 (1996)。
- [6] 加藤香：RIST news, (1996)。
- [7] 芝原正彦, 博士論文, 東京大学, 1996年, 12月。
- [8] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, 丹慶勝市, 奥村晴彦, 佐藤俊郎, 小林誠 訳; NUMERICAL RECIPES in C [日本語版], 技術評論社, 平成5年。
- [9] 柴山潔, コンピュータアーキテクチャ, オーム社 (1997年)。
- [10] 富士通(株)UXP/M VPP FORTRAN77 EX/vpp 使用手引書 V12用, 1994年1月。
- [11] PARAGON<sup>TM</sup> Fortran System Calls Reference Manual.
- [12] PARAGON<sup>TM</sup> Fortran Compiler User's Guide.

Table.1 Computing parameter.

原子数	7
1時間ステップにおけるQD計算回数	20
1時間ステップにおけるMD計算回数	1
時間ステップ数	500

Table.2 Real time and speed up ratio on VPP300.

使用PE数	計算所要時間	計算所要時間の比
1	20分07秒	1
7	3分35秒	5.62

Table.3 Real time and speed up ratio on Paragon XP/S.

使用ノード数	計算所要時間	計算所要時間の比
1	84時間20分	1
7	12時間20分	6.84

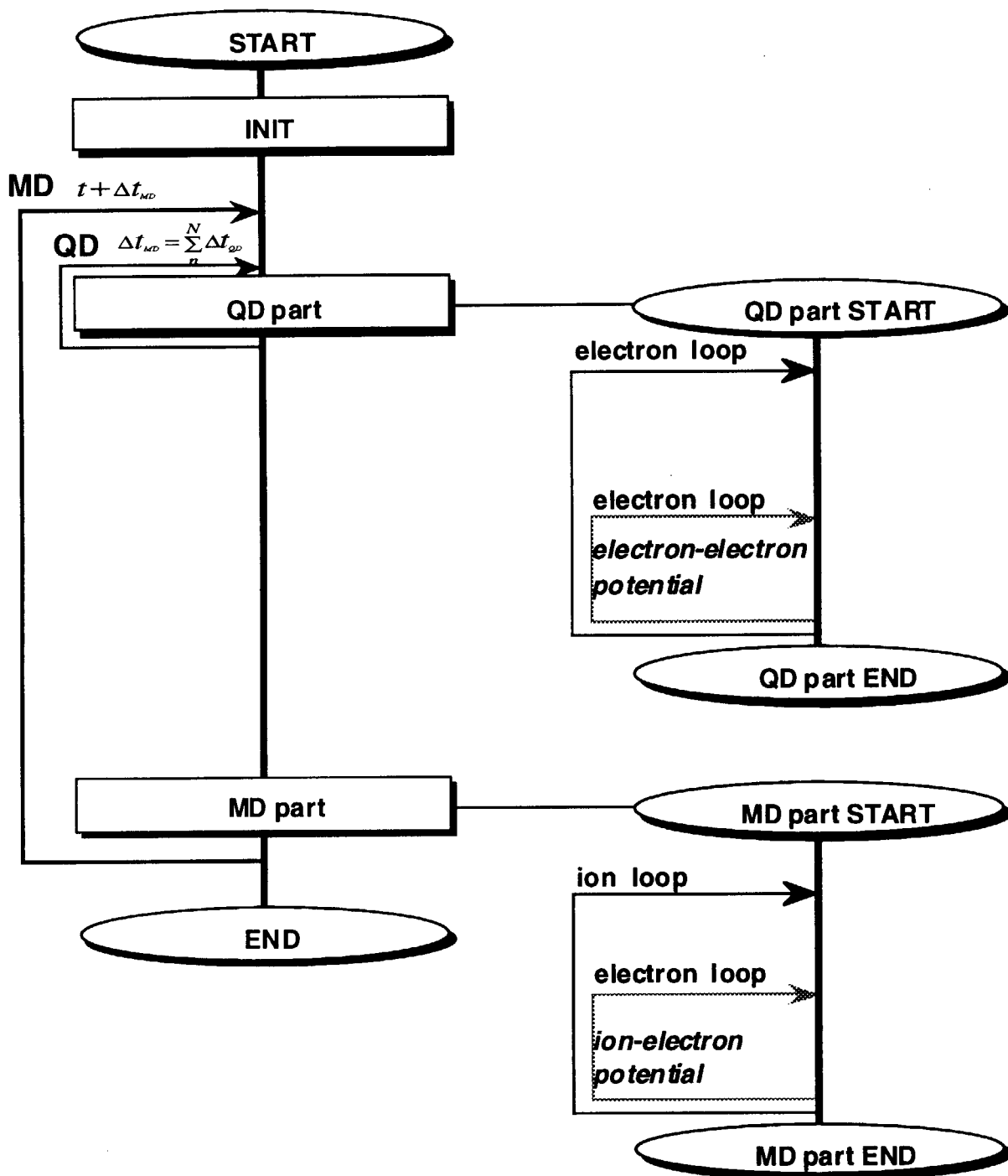


Fig.1 Flow chart of serial execution code.

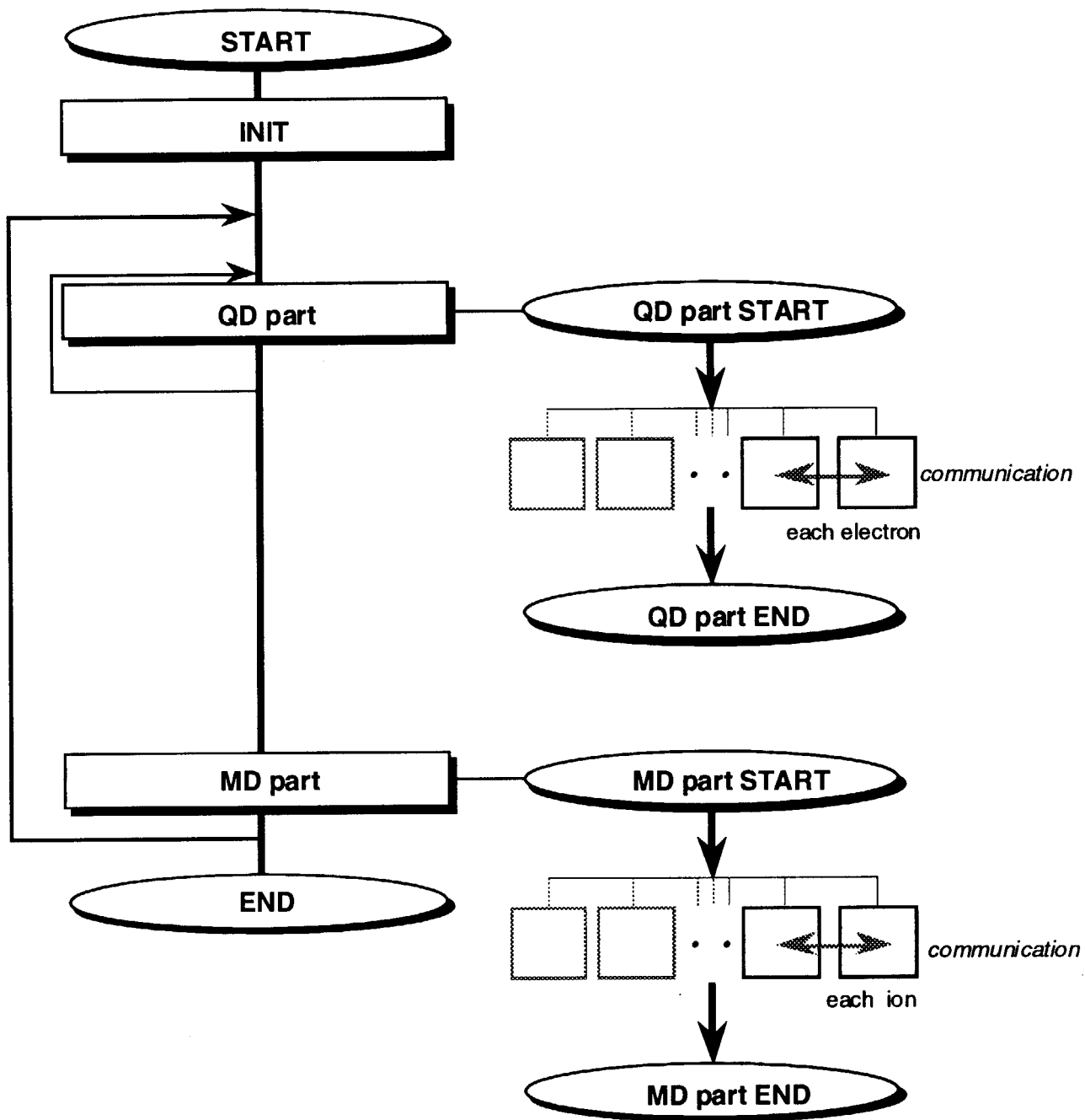


Fig.2 Construction of parallel execution code distributing method by particle.

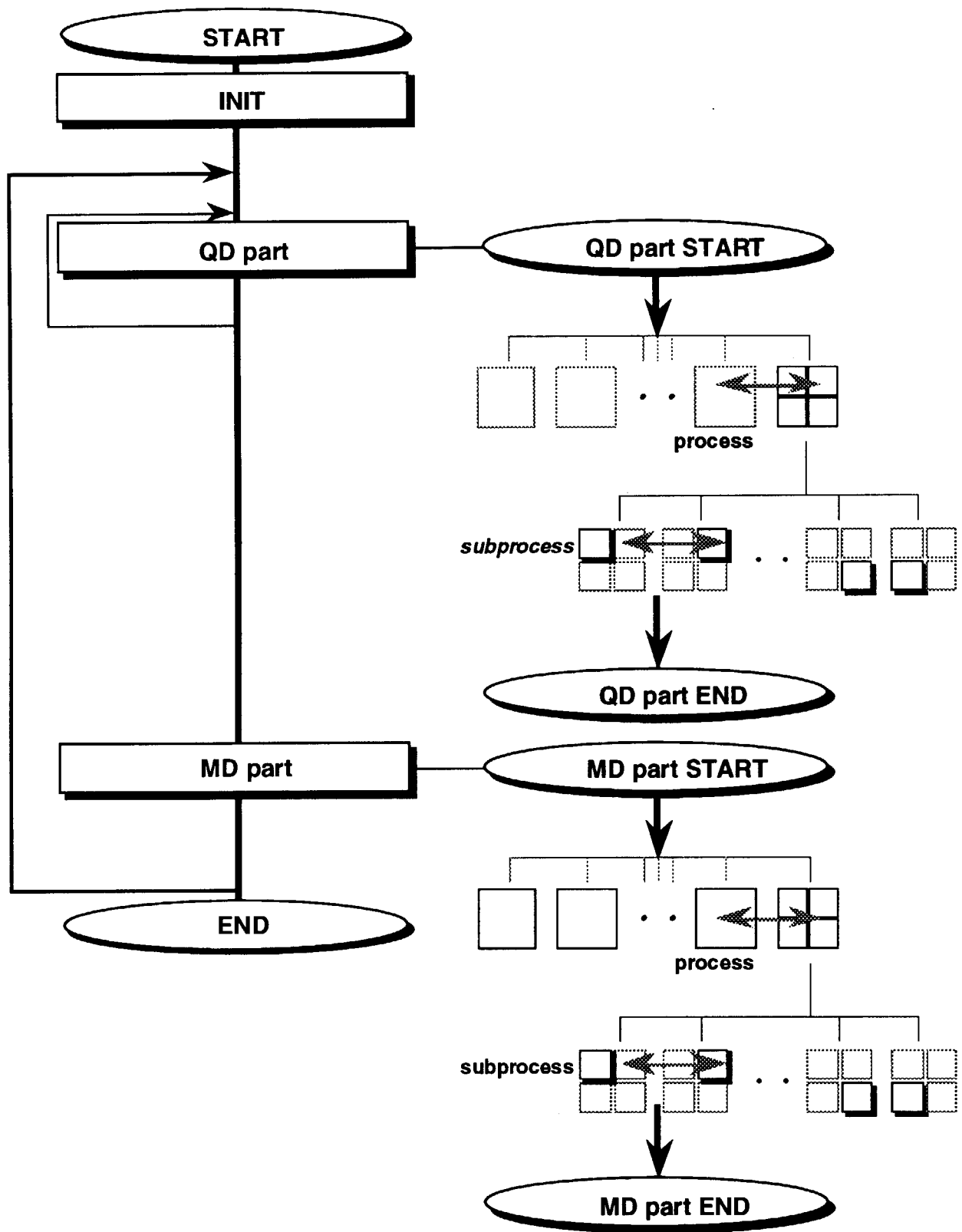


Fig.3 Construction of parallel execution code by distributing method by advanced particle.

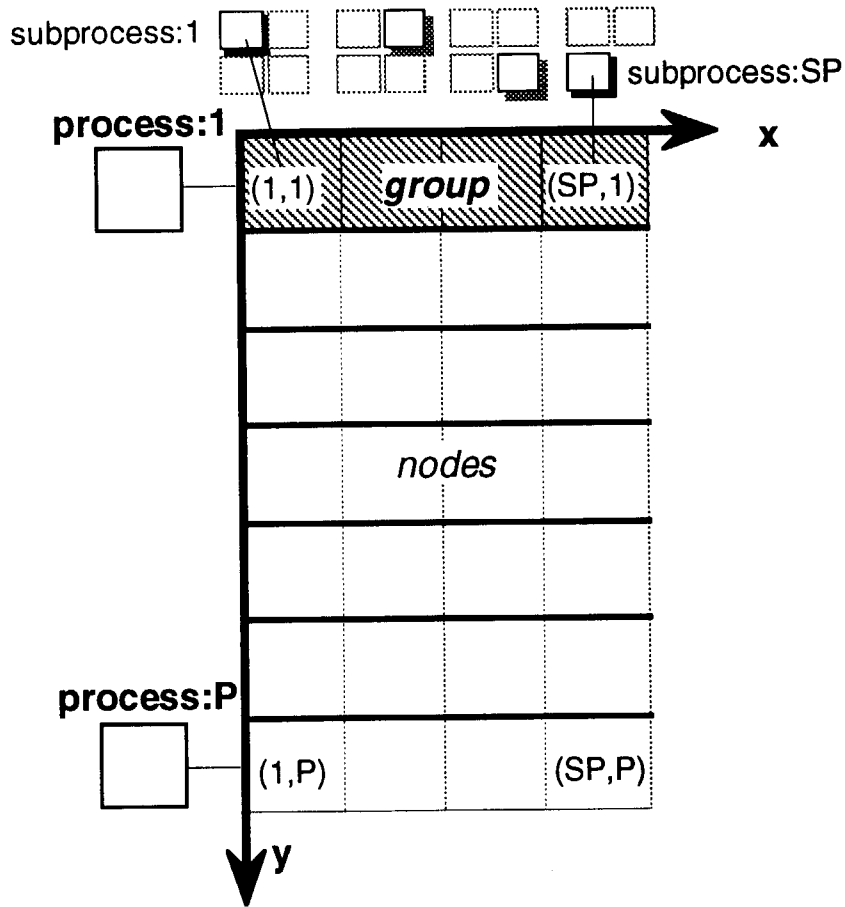


Fig.4 Projection of parallel execution code to nodes space.

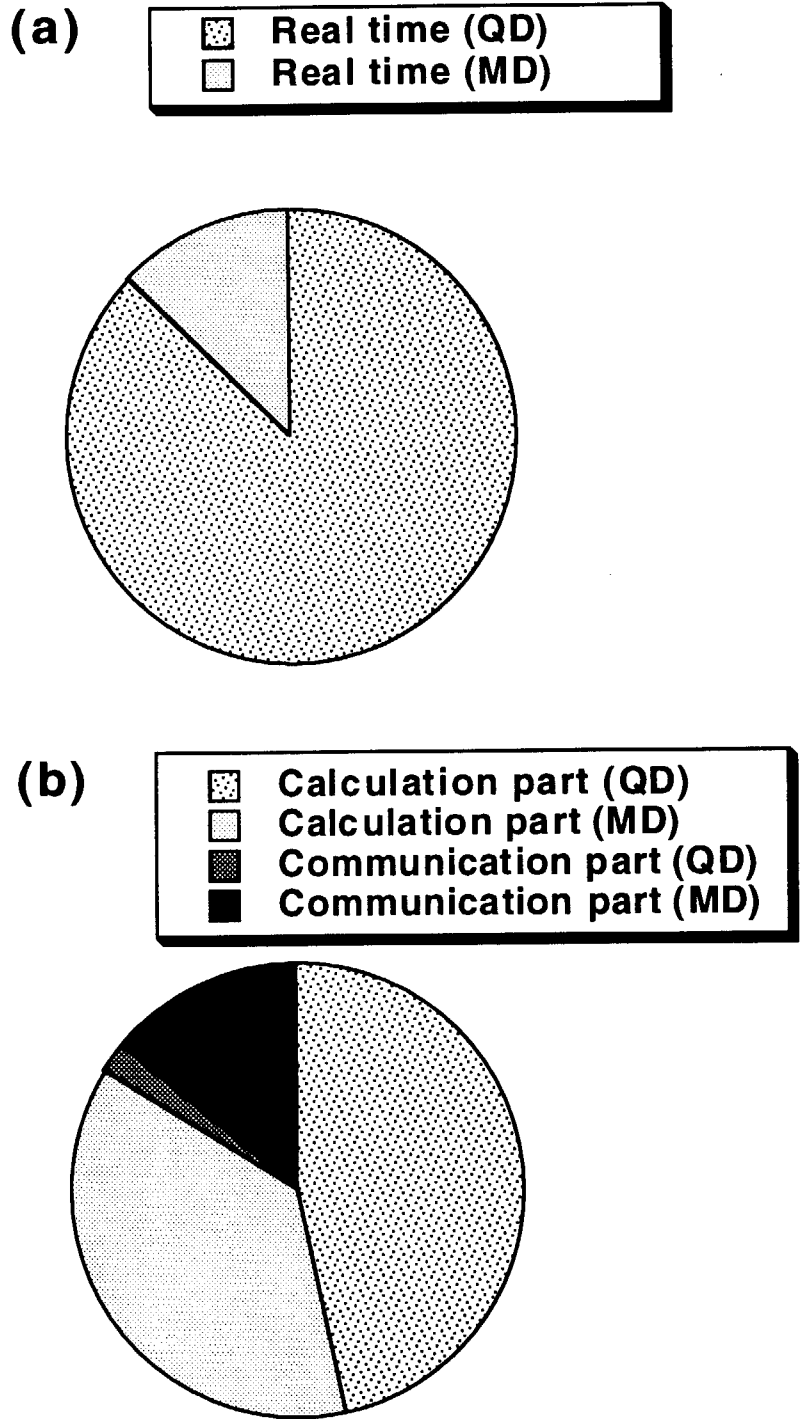


Fig.5 Cost distribution processing for VPP300. Figures (a) and (b) are in serial and parallel(7PE), respectively.



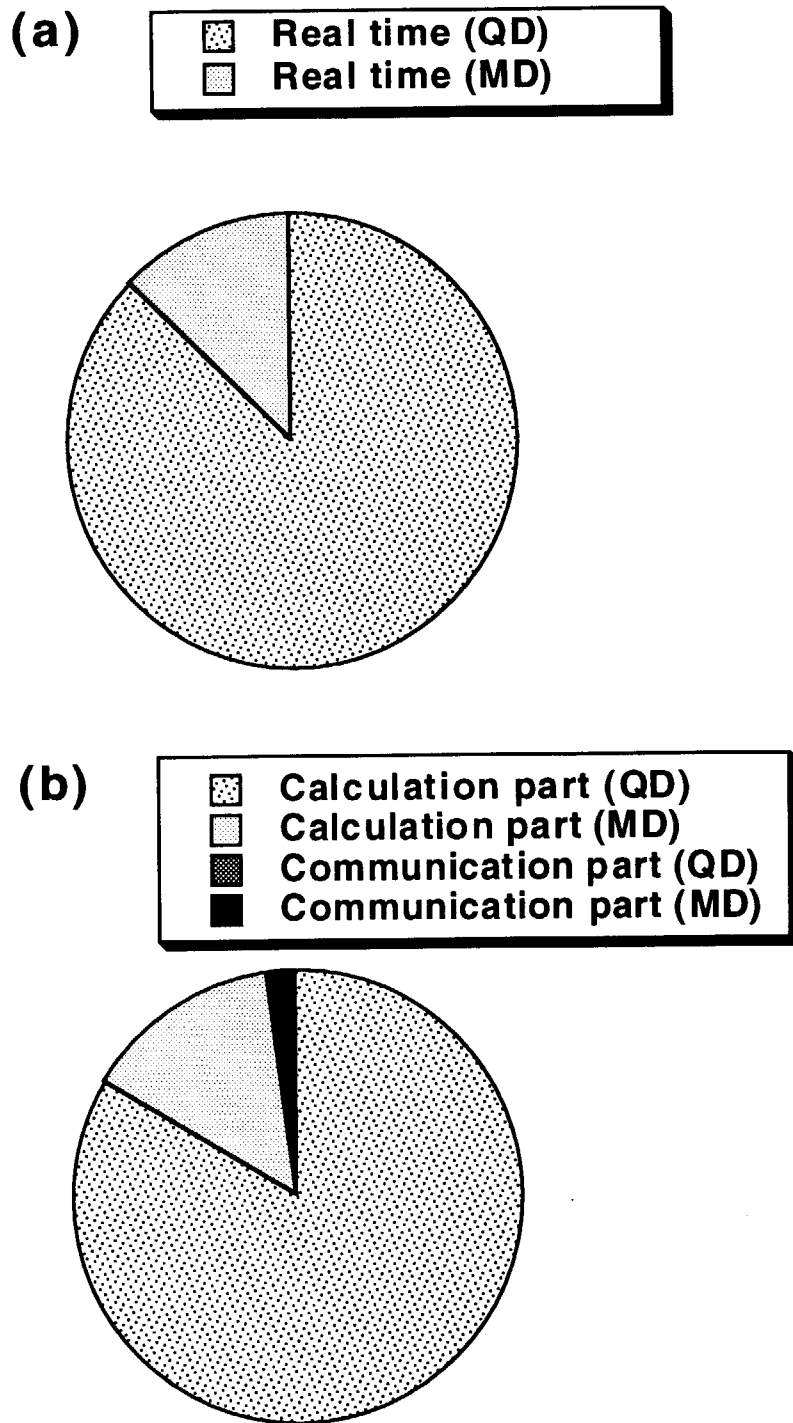


Fig.6 Cost distribution processing for Paragon XP/S. Figures (a) and (b) are in serial and parallel(7nodes), respectively.

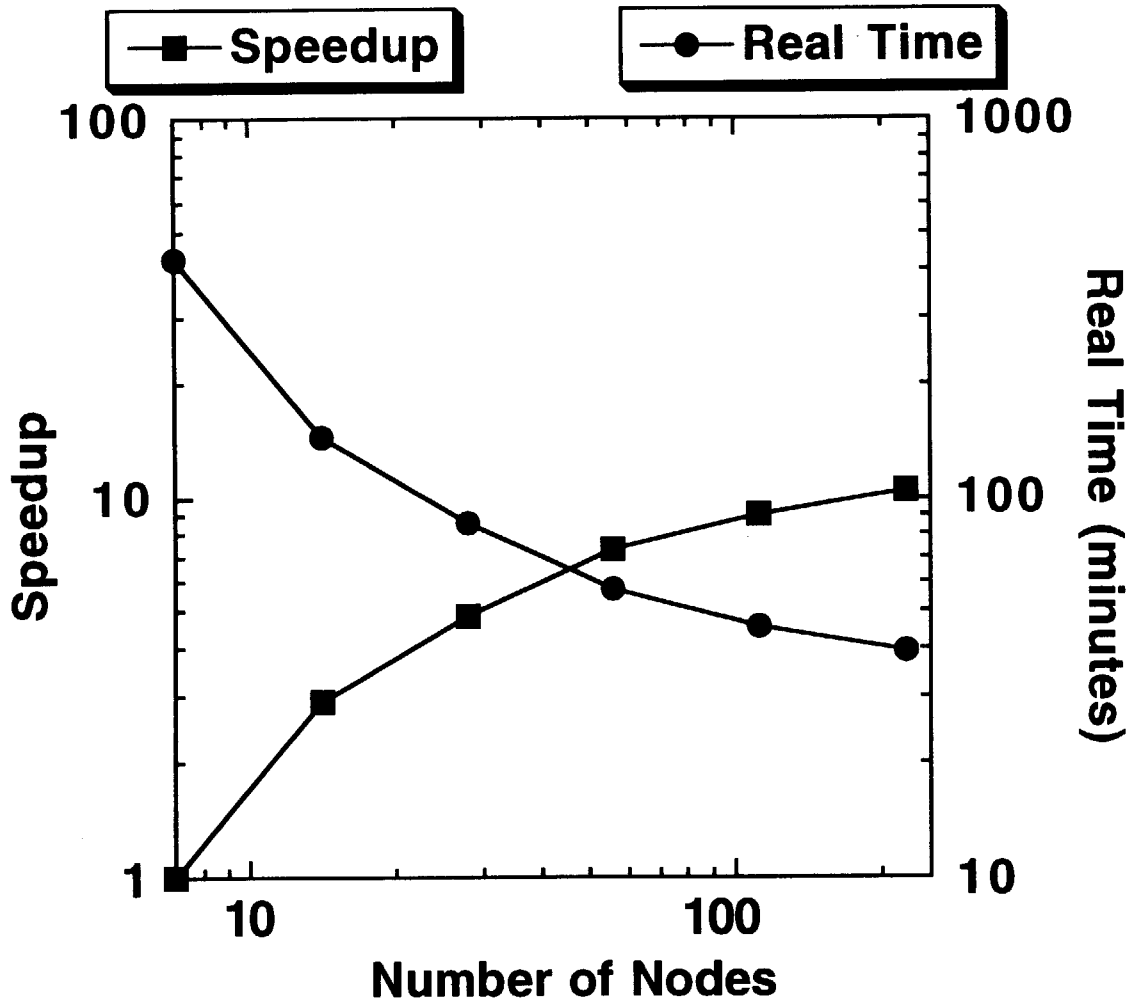


Fig.7 Speed up ratio and real time as a function of number of nodes in Paragon XP/S.

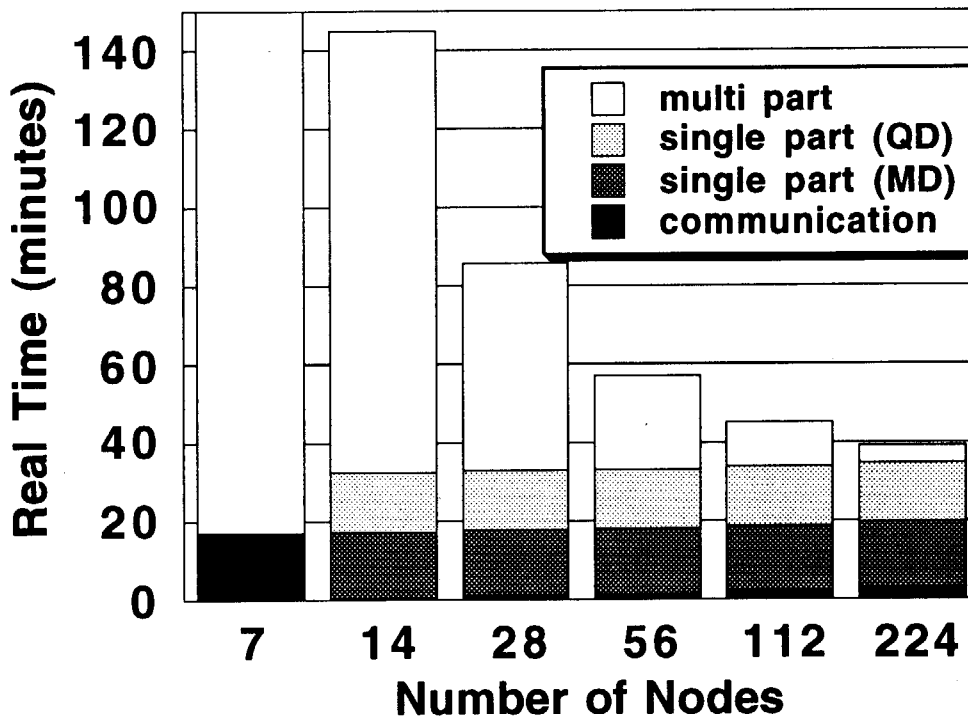
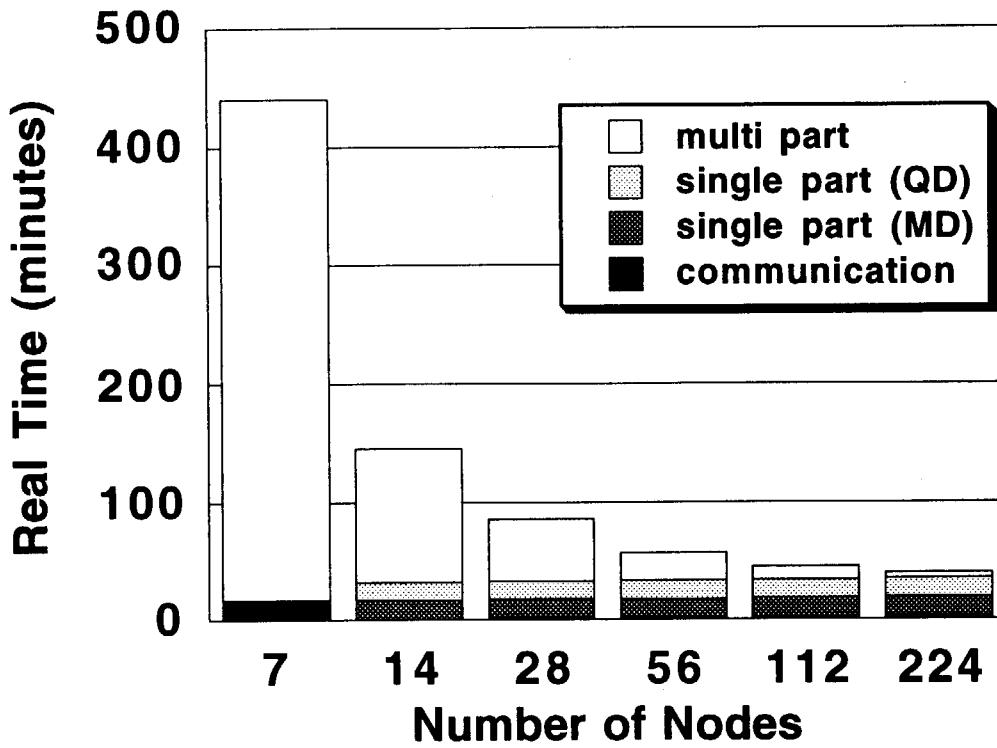


Fig.8 Cost distribution as a function of number of nodes in Paragon XP/S.

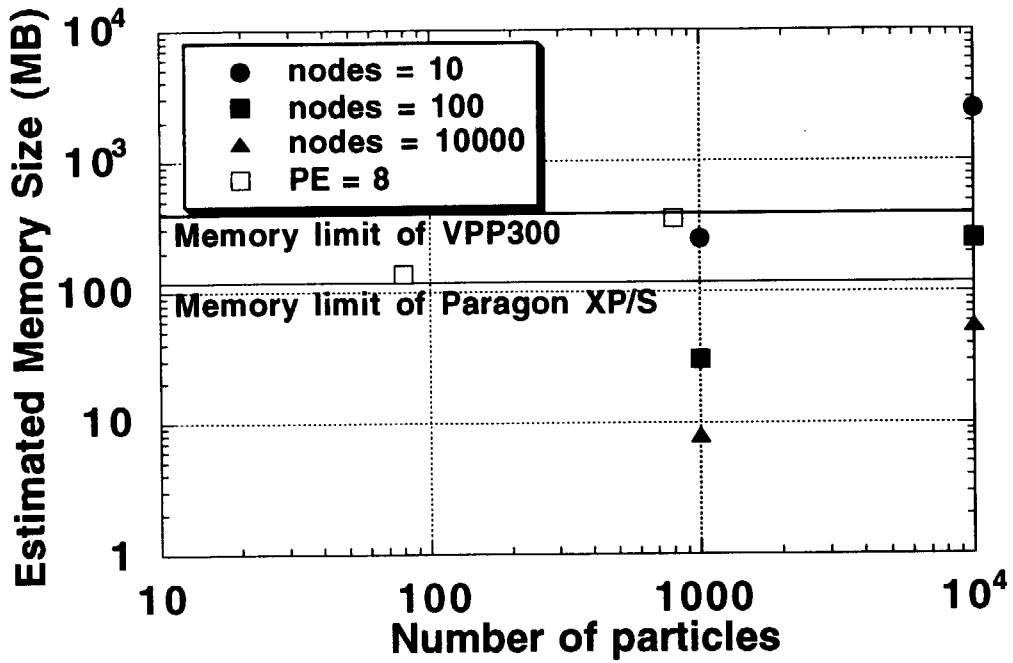


Fig.9 Estimated memory size as a function of number of particles.

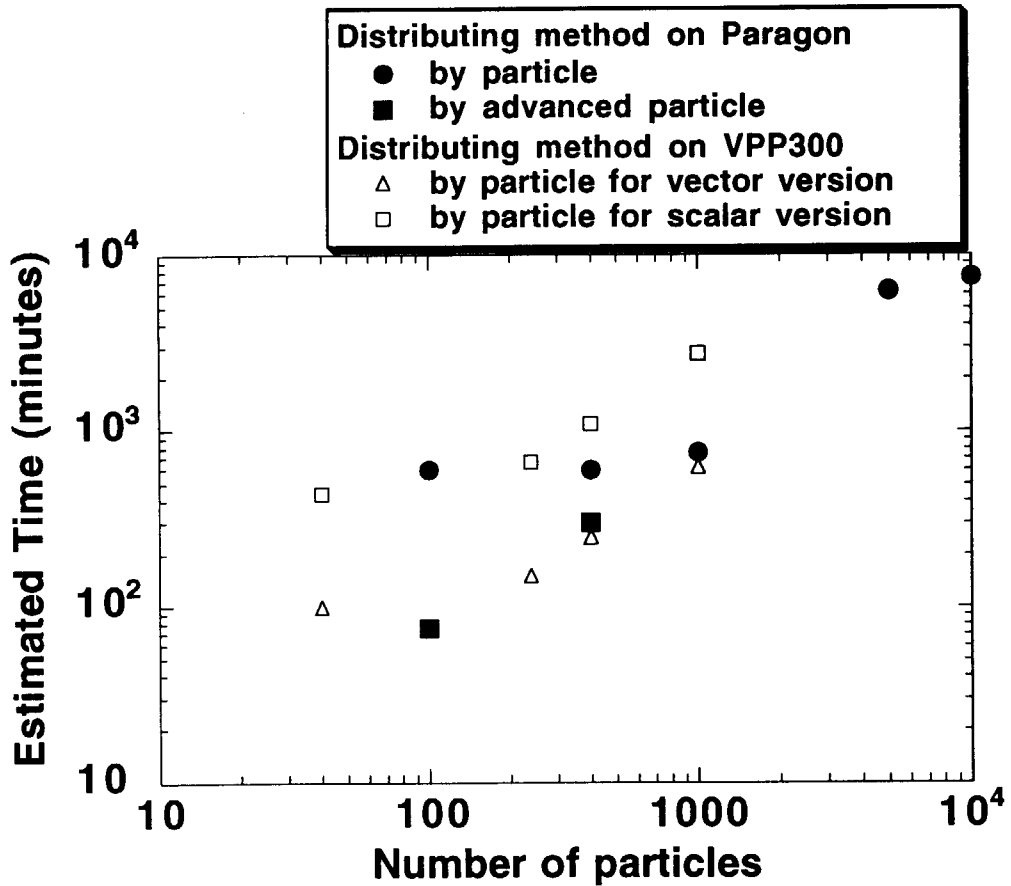


Fig.10 Estimated real time as a function of number of particles.

## 付録 グループ内総和演算用サブルーチンの開発

`nx` ライブラリには、グループ内総和演算を行うシステムコールは用意されていない。このため `nx` ライブラリを用いた、グループ内総和演算ルーチンの開発を行う。ここでは、グループ内総和演算用サブルーチンの動作条件と機能、総和値作成手順、`nx` ライブラリの適用について説明する。

### 1. サブルーチンの動作条件と機能

サブルーチンは、「 $x$ 方向・ $y$ 方向に2の $n$ 乗（特別な場合として7）個並ぶ2次元のノード空間での動作であることが条件で、与えられたグループをとる方向の指示と部分和から、各グループ内での総和値を出力する」ように機能する。

### 2. 総和値作成手順

グループ内のノード数が2の $n$ 乗である場合、次の手順で総和値を求めることができる。まず、異なる部分和を持つ2ノードを1対とする。互いのアロケートする部分和の交換を行い、受信した部分和と送信した部分和を和算し、部分和を上書きする。これを繰り返せば、部分和は総和値に上書きされる。次に、(1)どのようにノードを選択するか、(2)繰り返しが終了する条件について説明する。説明のため、部分和を重複してアロケートするノード集合をファミリーと記す。

#### (1) ノード対の検出

異なる部分和を持つ2ノードが1対となるには、異なるファミリーに属する相手を選択すること、同時に相手から自分が選択されるという条件を満たせばよい。このため、2ファミリーを1つに重ね合わせるように平行移動して、重なりあう2ノードを1対とする。

#### (2) 繰り返しの終了

グループ内の全ノードが同じファミリーに所属するとき、各ノードがアロケートする部分和は総和値となる。したがってグループ内のファミリー数が1となった場合に繰り返しの終了すればよい。

#### (3) グループ内ノード数が7である場合

グループ内ノード数が7である場合グループ内識別番号が0であるノードに、2ノード分のメッセージ送受信を行わせてグループ内ノード数が8であるように見せかける。

### 3. `nx` ライブラリの適用

プログラムの主な動作は、ファミリー番号の検出、送信先ノードの検出、部分和の交換である。この動作のために各ノードに与える識別番号は、グループ識別番号、グループ内ファミリー識別番号、グループ内ノード識別番号である。これらの識別番号は、`nx`ライブラリのシステムコールから得られるシステム全体のノード識別番号、各方向に配置するノード数、メッセージの交換回数から検出する。また部分和の交換には、`nx`ライブラリのシステムコール `csend`、`crecv` を用いる。

なお、本コードへの適用に関しては、実行時オプション `-mbf` による通信バッファのデフォルトサイズからの変更（5MBへの拡大）が必要であった。これは、`csend`、`crecv`の通信形式の特性によるものである。

サブルーチン test プログラムリスト

3月 2 13:48 1997 test.f Page 2

```

c----- summation
do 30 i=1, ndim
  dsenddata(i)=dsenddata(i)+dbuffer(i)
30 continue

call gsync()
goto 100
888 continue

call gsync()

return
end

c-----
c 1986.08.01 kaori@love.tokai.jaeri.go.jp
subroutine traintit(npex,npey)
c-----
implicit real*8 (a-h,o-z)
include 'fnx.h'
dimension-iclsr(2), ahead(2), na(2)
common/testcomx/
& iam(2)
& .isvch(2), nfmly(2,800)
& .ndsel(2,800), ndsel0(2,2,800)

c----- topology
nodes = numnodes()
na(1) = 1
na(2) = npex
iclsr(1)= npey
iclsr(2)= npey
iam(1) = mod(mynde(),npex)
iam(2) = int(mynde()/npex)
ahead(1) = iam(2)*npex
ahead(2) = iam(1)
isvch(1) = .mod(npex,2)
isvch(2) = .mod(npey,2)

id=0
2000 continue
id=id+1

c----- id change & topology reset
iam(1) = mod(mynde(),npex)
iam(2) = int(mynde()/npex)
if((isvch(id).eq.0 .and. (iam(id)+1).eq. iclsr(id)) iam(id)=-1

k=0
1000 continue
k=k+1
kk=2**k-1
nfmly(id,k)=iclsr(id)/kk

```

3月 2 13:48 1997 test.f Page 1

```

c-----
c 1986.08.01 kaori@love.tokai.jaeri.go.jp
subroutine test
&(dsenddata,dbuffer,ndim,npex,npey,id)
c-----
implicit real*8 (a-h,o-z)
include 'fnx.h'
dimension dsenddata(ndim),dbuffer(ndim)
common /nvwstep/ nvstep
common/testcomx/
& iam(2)
& .isvch(2),nfmly(2,800)
& .ndsel(2,800),ndsel0(2,2,800)

call gsync()
if(nvwstep.eq.1) then
call traintit(npex,npey)
end if

lep = ndim*8
k=0
100 continue
k=k+1
c----- buffer clear
do 20 i=1,ndim
  dbuffer(i)=0.0d0
20 continue

c----- set break condition
if(isvch(id).eq.0 .and. nfmly(id,k).eq.1) goto 888
if(isvch(id).eq.1 .and. nfmly(id,k).eq.0) goto 888

c----- special node action [iclsr(id)=7&iam(id)=0]
if( isvch(id).eq.1 .and. iam(id).eq.0 ) then
if( k.eq.1 ) goto 188

c----- iii=1:dummy action (iii=0:standard action)
mid =
do 250 iii=1,2
  = mynde()
  ndsl = ndsel0(iii,id,k)
call csend(mid,dsenddata, len,ndsl,0)
250 continue
mid1=ndsel0(1,id,k)
mid2=ndsel0(2,id,k)
call crcv(mid1,dbuffer,lep)
call crcv(mid2,dbuffer,lep)
188 continue

c----- standard node action
else
  ndsl = ndsel(id,k)
  mid = mynde()
  write(*,*) mynde(),ndsl
  call csend(mid,dsenddata, len,ndsl,0)
  call crcv(ndsl,dbuffer,lep)
end if

```

3月 2 13:48 1987 test.f Page 3

```

c ----- Set break condition
  if(isvch(id).eq.0 .and. nfmly(id,k).eq.1 .or.
    & isvch(id).eq.1 .and. nfmly(id,k).eq.0) goto 1888
c ----- Am I spacial node ? (ictstr(id)=7&iam(id)=0)
  if( isvch(id).eq.1 .and. iam(id).eq.0 )
c    <----- Yes, I am. iij=1:dummy (iij=0:standard)
    iiam = nhead(id)-nn(id)
    do 1500 iij=1,2
      myfid = (iam(id)+1)/kk
      iflag = mod(myfid,2)
    intvl
    ndsel0(iij,id,k) = iiam + intvl
    iiam = iiam + nn(id)
1500    continue
c    <----- NO, I am not. standard
    else
      if(id.eq.1) iiam = nhead(1)+iam(1)
      if(id.eq.2) iiam = iam(2)+nn(2)+iam(1)
      myfid = (iam(id)+1)/kk
      iflag = mod(myfid,2)
    intvl
    ndsel(id,k) = iiam + intvl
c    ----- standard to standard
    if(isvch(id).eq.0) then
      if( ndsel(id,k).lt.nhead(id)) then
        if(id.eq.1) ndsel(id,k)=ndsel(id,k)+ic1str(id)
        if(id.eq.2) ndsel(id,k)=ndsel(id,k)+nodes
      end if
    end if
c    ----- standard to dummy
    if(isvch(id).eq.1) then
      if(ndsel(id,k).lt.nhead(id)) then
        if(isvch(id).eq.1) ndsel(id,k)=nhead(id)
        end if
      end if
    end if
c
    goto 1000
1888    continue
    if(id.eq.1) goto 2000
    return
    end

```

This is a blank page.



# 国際単位系 (SI) と換算表

表1 SI基本単位および補助単位

量	名称	記号
長さ	メートル	m
質量	キログラム	kg
時間	秒	s
電流	アンペア	A
熱力学温度	ケルビン	K
物質	モル	mol
光度	カンデラ	cd
平面角	ラジアン	rad
立体角	ステラジアン	sr

表2 SIと併用される単位

名称	記号
分, 時, 日	min, h, d
度, 分, 秒	°, ', "
リットル	l, L
トン	t
電子ボルト	eV
原子質量単位	u

1 eV = 1.60218 × 10<sup>-19</sup> J  
1 u = 1.66054 × 10<sup>-27</sup> kg

表5 SI接頭語

倍数	接頭語	記号
10 <sup>18</sup>	エクサ	E
10 <sup>15</sup>	ペタ	P
10 <sup>12</sup>	テラ	T
10 <sup>9</sup>	ギガ	G
10 <sup>6</sup>	メガ	M
10 <sup>3</sup>	キロ	k
10 <sup>2</sup>	ヘクト	h
10 <sup>1</sup>	デカ	da
10 <sup>-1</sup>	デシ	d
10 <sup>-2</sup>	センチ	c
10 <sup>-3</sup>	ミリ	m
10 <sup>-6</sup>	マイクロ	μ
10 <sup>-9</sup>	ナノ	n
10 <sup>-12</sup>	ピコ	p
10 <sup>-15</sup>	フェムト	f
10 <sup>-18</sup>	アト	a

表3 固有の名称をもつSI組立単位

量	名称	記号	他のSI単位による表現
周波数	ヘルツ	Hz	s <sup>-1</sup>
力	ニュートン	N	m·kg/s <sup>2</sup>
圧力, 応力	パスカル	Pa	N/m <sup>2</sup>
エネルギー, 仕事, 熱量	ジュール	J	N·m
工率, 放射束	ワット	W	J/s
電気量, 電荷	クーロン	C	A·s
電位, 電圧, 起電力	ボルト	V	W/A
静電容量	ファラド	F	C/V
電気抵抗	オーム	Ω	V/A
コンダクタンス	ジーメンズ	S	A/V
磁束	ウェーバ	Wb	V·s
磁束密度	テスラ	T	Wb/m <sup>2</sup>
インダクタンス	ヘンリー	H	Wb/A
セルシウス温度	セルシウス度	°C	
光束	ルーメン	lm	cd·sr
照射度	ルクス	lx	lm/m <sup>2</sup>
放射能	ベクレル	Bq	s <sup>-1</sup>
吸収線量	グレイ	Gy	J/kg
線量等量	シーベルト	Sv	J/kg

表4 SIと共に暫定的に維持される単位

名称	記号
オングストローム	Å
バーン	b
バル	bar
ガリ	Gal
キュリー	Ci
レントゲン	R
ラド	rad
レム	rem

1 Å = 0.1 nm = 10<sup>-10</sup> m  
1 b = 100 fm<sup>2</sup> = 10<sup>-28</sup> m<sup>2</sup>  
1 bar = 0.1 MPa = 10<sup>5</sup> Pa  
1 Gal = 1 cm/s<sup>2</sup> = 10<sup>-2</sup> m/s<sup>2</sup>  
1 Ci = 3.7 × 10<sup>10</sup> Bq  
1 R = 2.58 × 10<sup>-4</sup> C/kg  
1 rad = 1 cGy = 10<sup>-2</sup> Gy  
1 rem = 1 cSv = 10<sup>-2</sup> Sv

(注)

- 表1-5は「国際単位系」第5版, 国際度量衡局1985年刊行による。ただし, 1 eV および 1 u の値はCODATAの1986年推奨値によった。
- 表4には海里, ノット, アール, ヘクタールも含まれているが日常の単位なのでここでは省略した。
- bar は, JISでは流体の圧力を表わす場合に限り表2のカテゴリーに分類されている。
- EC閣僚理事会指令では bar, barn および「血圧の単位」mmHgを表2のカテゴリーに入れている。

## 換算表

力	N (=10 <sup>5</sup> dyn)	kgf	lbf
1		0.101972	0.224809
9.80665		1	2.20462
4.44822		0.453592	1

粘度 1 Pa·s (N·s/m<sup>2</sup>) = 10 P (ポアズ) (g/(cm·s))

動粘度 1 m<sup>2</sup>/s = 10<sup>4</sup> St (ストークス) (cm<sup>2</sup>/s)

圧	MPa (=10 bar)	kgf/cm <sup>2</sup>	atm	mmHg (Torr)	lbf/in <sup>2</sup> (psi)
1		10.1972	9.86923	7.50062 × 10 <sup>3</sup>	145.038
0.0980665		1	0.967841	735.559	14.2233
0.101325		1.03323	1	760	14.6959
1.33322 × 10 <sup>-4</sup>		1.35951 × 10 <sup>-3</sup>	1.31579 × 10 <sup>-3</sup>	1	1.93368 × 10 <sup>-2</sup>
6.89476 × 10 <sup>-3</sup>		7.03070 × 10 <sup>-2</sup>	6.80460 × 10 <sup>-2</sup>	51.7149	1

エネルギー・仕事・熱量	J (=10 <sup>7</sup> erg)	kgf·m	kW·h	cal (計量法)	Btu	ft·lbf	eV
1		0.101972	2.77778 × 10 <sup>-7</sup>	0.238889	9.47813 × 10 <sup>-4</sup>	0.737562	6.24150 × 10 <sup>18</sup>
9.80665		1	2.72407 × 10 <sup>-6</sup>	2.34270	9.29487 × 10 <sup>-3</sup>	7.23301	6.12082 × 10 <sup>19</sup>
3.6 × 10 <sup>5</sup>		3.67098 × 10 <sup>5</sup>	1	8.59999 × 10 <sup>5</sup>	3412.13	2.65522 × 10 <sup>6</sup>	2.24694 × 10 <sup>25</sup>
4.18605		0.426858	1.16279 × 10 <sup>-6</sup>	1	3.96759 × 10 <sup>-3</sup>	3.08747	2.61272 × 10 <sup>19</sup>
1055.06		107.586	2.93072 × 10 <sup>-4</sup>	252.042	1	778.172	6.58515 × 10 <sup>21</sup>
1.35582		0.138255	3.76616 × 10 <sup>-7</sup>	0.323890	1.28506 × 10 <sup>-3</sup>	1	8.46233 × 10 <sup>18</sup>
1.60218 × 10 <sup>19</sup>		1.63377 × 10 <sup>-20</sup>	4.45050 × 10 <sup>-26</sup>	3.82743 × 10 <sup>-20</sup>	1.51857 × 10 <sup>-22</sup>	1.18171 × 10 <sup>-19</sup>	1

1 cal = 4.18605 J (計量法)  
= 4.184 J (熱化学)  
= 4.1855 J (15°C)  
= 4.1868 J (国際蒸気表)  
仕事率 1 PS (馬力)  
= 75 kgf·m/s  
= 735.499 W

放射能	Bq	Ci
1		2.70270 × 10 <sup>-11</sup>
3.7 × 10 <sup>10</sup>		1

吸収線量	Gy	rad
1		100
0.01		1

照射線量	C/kg	R
1		3876
2.58 × 10 <sup>-4</sup>		1

線量当量	Sv	rem
1		100
0.01		1

量子分子動力学シミュレーションコードの並列化