

プラズマ電磁粒子コードの並列化手法と速度向上率の評価

村田 健史^{†1} 上岡 功治^{†1} 高橋 誠治^{†2}
 岡田 雅樹^{†3} 上田 裕子^{†4}
 大村 善治^{†5} 松本 紘^{†5}

本稿では、プラズマ電磁粒子コードの並列化手法の1つとして、ハイブリッド分割法を考える。ハイブリッド分割法では、粒子計算サブルーチンを粒子分割法で、電磁場および流体量のサブルーチンを領域分割法で並列化する。並列化数は、サブルーチンごとに、独立して設定する。本研究では、まず、MPI 関数の通信速度特性と内部通信ロジックを考慮して、各サブルーチンの並列化数を変数とした速度向上率の定式化を行った。これにより、最大速度向上率の予測値と、各サブルーチンの並列化数を求めることができる。さらに、この手法を、プラズマ電磁粒子コードの1つである、プラズマ粒子流体混成コードに適用した。その結果、粒子計算サブルーチン、流体計算サブルーチン、および電流計算サブルーチンを最大 PE 数で並列化し、電磁場サブルーチンを逐次計算と設定することにより、16 並列で予測値が 12.96、実測値が 14.45 という、高い速度向上率を得た。

A Parallel Computing Method for Plasma Particle Simulation Code

KEN T. MURATA,^{†1} KOJI UEOKA,^{†1} SEIJI TAKAHASHI,^{†2}
 MASAKI OKADA,^{†3} HIROKO O. UEDA,^{†4} YOSHIHARU OMURA^{†5}
 and HIROSHI MATSUMOTO^{†5}

In this paper, we discuss a parallelization method for plasma electromagnetic particle simulation codes. Plasma particle codes are composed of particle subroutines, electromagnetic field subroutines, and fluid subroutines. The present method is based on combination between data division method for particles and area division method for fields and fluid. The parallelization number is independently set in each subroutine so that the code achieves best computing performance. With estimation of MPI communication and calculation duration time, we establish a formula to evaluate the speed-up. We applied this method to a plasma particle code: a hybrid code. High parallelization performance, the speed-up is as high as 13.59 as a predicted value and 14.45 as an evaluated value in 16 PE system, is achieved.

1. はじめに

現在、高速演算や大容量メモリを必要とするさまざまな分野で、並列計算機の実用的利用が始まっている^{1),2)}。時空間での物理現象を調べる計算機シミュレーションは、並列計算が有効な分野の1つである。

物理化学の計算機シミュレーションのうち、さまざまな媒質の振舞いを調べる手法としては、媒質の運動をつかさどる微積分方程式を数値的に解く、流体シミュレーションや粒子シミュレーションが有効である。流体シミュレーションは、媒質を流体としてとらえる。粒子シミュレーションは、個々の粒子に働く力を求め、粒子の運動を数値的に解く。これにより、粒子の軌道を求め、媒質の運動論的現象を解析する。

流体シミュレーションでは、媒質のすべての物理量は格子点上で定義される。したがって、空間を PE 個数で均等に分割し、各 PE が担当する領域のみを計算する、均等領域分割法を用いた並列化が有効である。この手法は、気体や液体の流れのシミュレーションや、熱伝導のシミュレーションなどで、広く用いられてきた。プラズマ流体シミュレーションにおいても、均等

†1 愛媛大学工学部

Faculty of Engineering, Ehime University

†2 NTT 先端技術総合研究所

NTT Science and Core Technology Laboratory Group

†3 国立極地研究所情報科学センター

Information Science Center, National Institute of Polar Region

†4 宇宙開発事業団技術研究本部システム解析ソフトウェア研究開発センター

NASDA System Analysis and Software Laboratory

†5 京都大学宇宙電波科学研究センター

Radio Science Center for Space and Atmosphere, Kyoto University

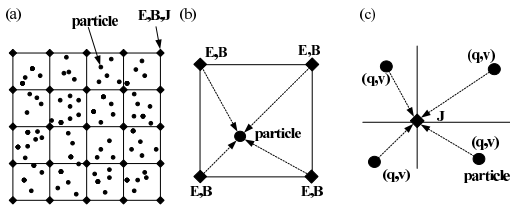


図1 プラズマ粒子コードのPIC法の概要：(a) 粒子は任意の位置，電磁場と電流は格子点上で定義される。(b) 粒子位置の電磁場は，近傍格子点から求められる。(c) 粒子量は，格子点上に足しこまれる

Fig.1 A schematic picture of PIC (Particle In Cell) method. (a) Particles located at any position in the region. (b) Electromagnetic fields at a particle. (c) Field and current values defined on grid points.

領域分割法は有効である。一方，プラズマ電磁粒子シミュレーションは，流体計算と比較して，一般に計算規模が大きい。そのため，コードの並列化による高速計算が，特に必要である。

これまで，効率的に粒子軌道を計算するための並列化手法が，さまざまな分野の粒子シミュレーションにおいて提案されてきた。これらの手法は，粒子分割法^{3),4)}と領域分割法^{5)~9)}に大別される。本研究では，プラズマ電磁粒子コードの並列化手法として，粒子分割法と領域分割法を組み合わせた，ハイブリッド分割法について考える。ハイブリッド分割法による計算時間を定式化し，速度向上率の予想値と並列計算機による実測値を比較する。さらに，最高の速度向上率を達成するための方法について調べる。

2. ハイブリッド分割法の提案

2.1 プラズマ電磁粒子シミュレーションコード

一般に，プラズマ電磁粒子コードは，粒子計算と流体計算，および電磁場の計算より成る。粒子計算および流体計算は，各媒質の運動(量)方程式や連続の式，エネルギーの式などを解く。一方，電磁場の計算は Maxwell 方程式を解く。

図1(a)に示すように，プラズマ電磁粒子コードでは，粒子は全計算領域を自由に運動できる。一方，電磁場および流体量は，格子点上で定義される¹⁰⁾。

プラズマ粒子シミュレーションでは，空間を離散的な格子に分割する。図1(b)のように，格子内の粒子は，近傍の格子点上で定義された電磁気力のみを受け，粒子間力は考えない。一方，電荷密度や電流密度は，図1(c)に示すように，個々の粒子の電荷や速度を格子点上で積分して求められる。粒子より求められた電

荷密度や電流密度により，Maxwell 方程式を解き，電磁場を時間方向に解き進める。

2.2 電磁粒子コードの並列化の目的

プラズマ電磁粒子シミュレーションコードの基礎方程式は，Vlasov 方程式

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{r}} + \mathbf{a} \cdot \frac{\partial f}{\partial \mathbf{v}} = 0 \quad (1)$$

の，0次，1次，2次のモーメントをとることにより得られる。なお， f は速度分布関数で，時刻 t ，位置 \mathbf{r} ，および速度 \mathbf{v} の関数である。 \mathbf{a} は，加速度 $d\mathbf{v}/dt$ である。

プラズマ電磁粒子シミュレーションでは，速度分布関数が，位置空間および速度空間において，十分に滑らかでなくてはならない。図2に，(a) 64個，(b) 160個，(c) 320個，(d) 640個，(e) 1,000個の粒子が作る速度分布関数を示す。図より，滑らかな速度分布のためには，格子点あたり粒子数が640以上であることが望ましい。

このような大規模粒子シミュレーションのためには，一般に，数 GB 以上の主記憶容量を必要とする。大規模主記憶容量は，単一ノードから構成される計算機では，実現が容易ではない。近年，発達してきた並列計算機を用いることにより，このような大規模計算が可能となる。

2.3 ハイブリッド分割法の概要

過去，プラズマ電磁粒子シミュレーションでは，負荷分散のための，動的領域分割法についての研究¹²⁾や，粒子分割法^{4),5)}の研究が進んでいる。動的領域分割法では，図3(a)に示すように，領域に含まれる粒子数が均等になるよう，領域を動的に分割する。そのため，電磁場および流体の計算では，負荷のばらつきが生じる。

ハイブリッド分割法では，図3(b)に示すように，粒子計算を均等データ分割し，電磁場および流体計算を均等領域分割する。そのため，ほぼ完全な負荷分散を実現することができる。ハイブリッド分割法では，粒子計算サブルーチンと電磁場および流体計算サブルーチンで処理が異なる。粒子計算については2.4節で，電磁場および流体計算については2.5節で説明する。

2.4 粒子計算部の並列化

一般に，プラズマ電磁粒子コードでは，全粒子数は全格子点数の数十から数百倍である。したがって，粒子計算が全体に占める割合は，電磁場計算や流体計算と比較して大きい。粒子計算を効率的に並列化するこ

これは，プラズマ粒子を超粒子として考えるためである¹¹⁾。

3.1 節の，プラズマ粒子流体混成コードも同様である。

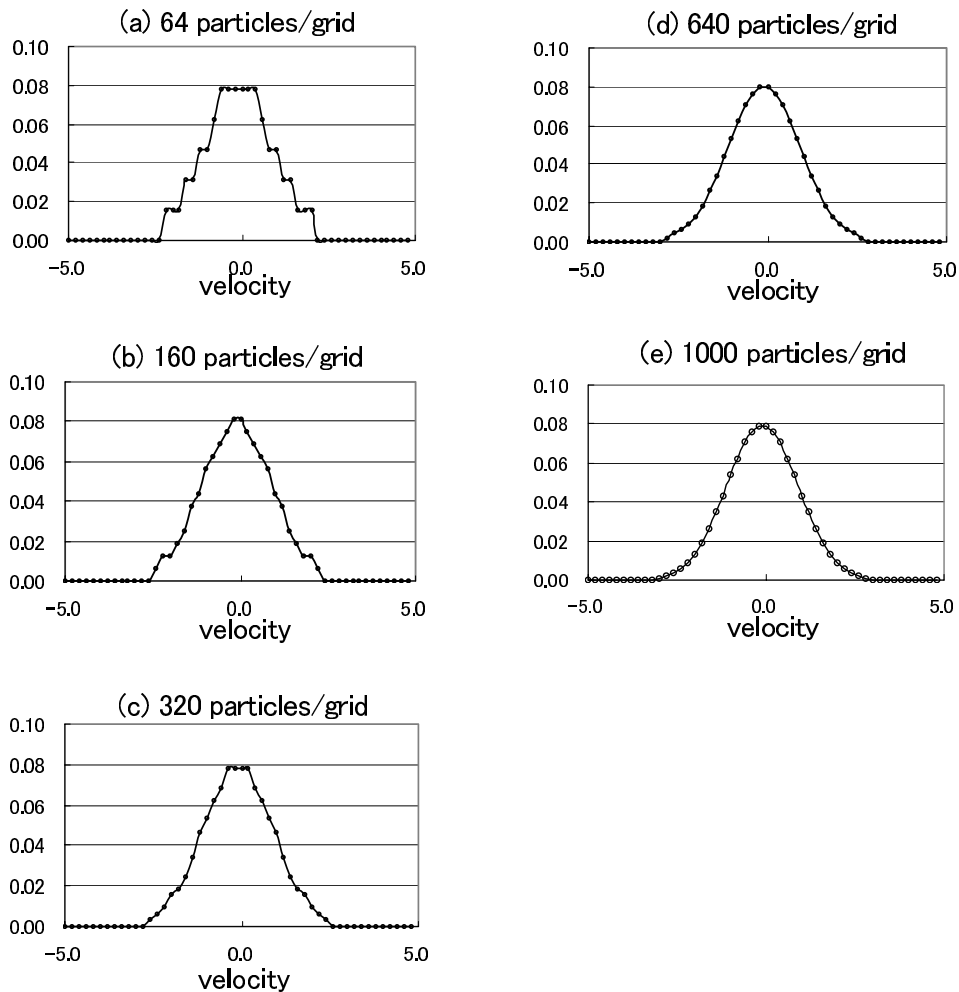


図2 プラズマ粒子による速度分布関数の実現：粒子数は (a) 64 個, (b) 160 個, (c) 320 個, (d) 640 個, (e) 1,000 個から構成されている．熱速度は 1.0 とした．速度空間を 50 分割して表示した

Fig.2 Particle distribution functions realized by (a) 64, (b) 160, (c) 320, (d) 640, and (e) 1,000 particles. Thermal velocity is 1.0.

とが、全計算効率向上には重要である。

プラズマ電磁粒子コードの粒子計算は、粒子の位置および速度を時間的に進める運動計算部と、粒子の位置と速度を各格子点上に足し込み、モーメント量（運動量と電荷密度）を求めるモーメント計算部から成る。

運動計算部の手法としては、Bunemann-Boris法¹³⁾が最もよく使用される。Bunemann-Boris法では、粒子の位置と速度を時間的に進めるために、図1(b)に示すように、粒子位置における電磁場の値を、近傍格子点上から内挿する。プラズマ電磁シミュレーションコードでは、すべての粒子は、シミュレーション空間全体に移動することが可能である。そのため、並列計算機においては、各PEは、全空間の電磁場のデータを有する必要がある。ただし、運動計算部でPE間の

通信が必要ない。そのため、計算に要する時間は、並列化数が増えるごとに、反比例に短縮される。

一方、各粒子の持つ速度や電荷は、モーメント計算部において、図1(c)に示すように、モーメント量（密度や運動量）として、粒子近傍の格子点に足し込まれる。各PEのモーメント量を合計することにより、全粒子によるモーメント量を得る。全PEのモーメント値を合計する計算には、PE間通信が必要となる。

2.5 電磁場および流体計算部の並列化

電磁場および流体計算に必要な計算量は、プラズマ電磁粒子コード全体の計算量の数%から数十%程度である。しかし、アムダールの法則によると、少量の逐次処理も、全体の並列処理の効率を妨げる¹⁴⁾。そのため、ハイブリッド分割法では、電磁場や流体の計算

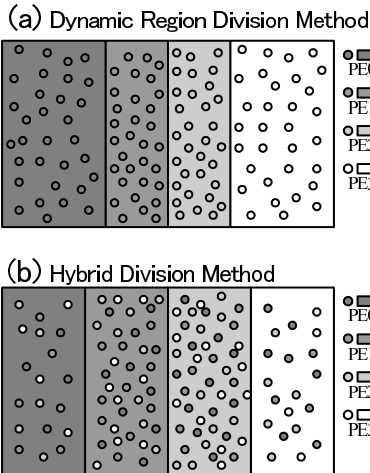


図 3 (a) 動的領域分割法と (b) ハイブリッド分割法の模式図
 Fig. 3 Schematic picture of (a) Dynamic region division method and (b) Hybrid division method.

に関しても、より高い効率を得るための並列化を行う。

電磁場および流体計算の並列化には、計算する領域を実空間上で均等に分割し、各領域の計算を各 PE に割り当てる均等領域分割法を採用する。ただし、電磁場計算と流体計算では、PE 間のデータ通信は、異なる通信となる。

各 PE での粒子計算には、全空間の電磁場値が必要である。すなわち、電磁場計算では、各 PE は担当領域の計算後、担当領域のデータをすべての PE 間で交換する。一方、流体量は粒子計算に不要であるので、分割したまま計算を進めることができる。したがって、流体計算では、担当計算領域の境界部（のりしろ部）のデータのみ、交換すればよい。

2.6 ハイブリッド分割法の手順

プラズマ電磁粒子コードでは、基礎方程式は共通であっても、方程式の差分法やアルゴリズム、また、各サブルーチンの計算手順などは、コードによって異なる。したがって、ハイブリッド分割法の実装は各コードによって異なり、一般化は難しい。一般的には、以下のような手順で、各計算部のサブルーチンを実装する。

- (1) タイムチャートを作成し、各サブルーチンの計算順序を決める。
- (2) 電磁場計算サブルーチンおよび流体計算サブルーチンを、領域分割法により実装する。粒子計算サブルーチンを、粒子分割法により実装する。
- (3) 各サブルーチン間の関係を調べ、それに応じた通信関数をサブルーチンに実装する。

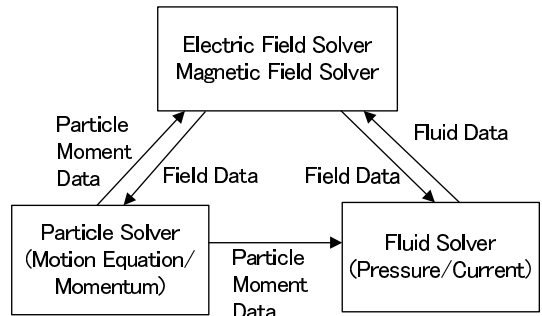


図 4 プラズマ電磁粒子コードの構成（電磁粒子流体混成コードの場合）：電磁場計算・粒子計算・流体計算の関係
 Fig. 4 Construction of an EM plasma particle simulation code (a plasma hybrid code): Relationship between EM field, particle and fluid subroutines.

- (4) 各サブルーチンの計算時間と通信時間を、測定または予測より求める。
- (5) 各サブルーチンの並列化数を決定する。

ハイブリッド分割法の、プラズマ電磁粒子コードへの応用例を、3 章で議論する。

3. ハイブリッド分割法のプラズマ電磁粒子コードへの適用

3.1 プラズマ粒子流体混成コードについて

前章のハイブリッド分割法を、既存のプラズマ電磁粒子シミュレーションコードに適用する。本稿では、プラズマ電磁粒子シミュレーションコードの 1 つである、プラズマ粒子流体混成コードを用いる（図 4）。

プラズマ粒子流体混成コードは、電荷の準中性条件を仮定し、イオンを超粒子¹⁰⁾、電子を無質量の流体として扱うもので、低周波のイオン粒子運動の解析に適したコードである。図 1 (a) に示すように、イオン粒子はシミュレーション空間内にランダムに配置され、電磁場および流体量は、格子点上で定義される。

プラズマ粒子流体混成コードで用いられる基本方程式は、以下のとおりである。

- s 種のイオンの運動方程式（超粒子として扱う）

$$m_s \frac{dv}{dt} = q_s (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \tag{2}$$

$$\frac{d\mathbf{r}}{dt} = \mathbf{v} \tag{3}$$

- 電荷の準中性条件式

$$-en_e + \sum_s q_s n_s = 0 \tag{4}$$

プラズマ電磁ハイブリッドコードとも呼ばれる¹¹⁾。本稿では、提案するハイブリッド分割法との混乱を避けるため、プラズマ粒子流体混成コードと記述する。

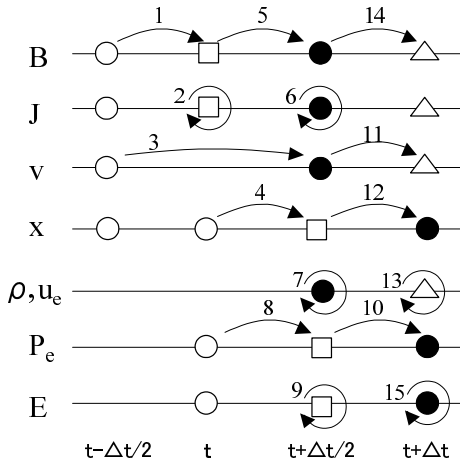


図5 プラズマ粒子流体混成コードのタイムチャート(変数は3.1節に準じる).○は初期値,●は最終値,□は中間値,△は予測値である

Fig. 5 The timechart of the present plasma hybrid code. The parameters are given in Section 3.1: ○: initial value, ●: destination value, □: intermediate value and △: predicted value.

- 電子の運動方程式(無質量の流体として扱う)

$$n_e m_e \frac{d\mathbf{u}_e}{dt} = 0 = -en_e(\mathbf{E} + \mathbf{u}_e \times \mathbf{B}) + \eta \mathbf{J} \quad (5)$$

- 電子のエネルギー式

$$\left(\frac{\partial}{\partial t} + \mathbf{u}_e \cdot \nabla\right) P_e = -\gamma_e P_e \nabla \cdot \mathbf{u}_e + \eta J^2 \quad (6)$$

- マックスウェル方程式(ダーウィン近似)

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \quad (7)$$

$$\mathbf{J} = \frac{1}{\mu_0} \nabla \times \mathbf{B} \quad (8)$$

- 電流密度の定義式

$$\mathbf{J} = -en_e \mathbf{u}_e + \sum_s q_s n_s \mathbf{u}_s \quad (9)$$

$\mathbf{E}, \mathbf{B}, \mathbf{J}, P_e$ は,それぞれ,電場,磁場,電流密度,電子圧力を表す.また, $m_s, q_s, n_s, \mathbf{v}, \mathbf{r}$ は第 s 種イオンの質量,電荷量,数密度,速度,位置となる.さらに, $m_e, e, n_e, \mathbf{u}_e$ は,それぞれ,電子の質量,電荷量,数密度,流速である.比熱比 γ_e および電気抵抗 η は,定数で与えられる.

プラズマ粒子流体混成コードでは,これらの基本式を差分化する.式(2)~(9)はサブルーチン化されるが,サブルーチン順序はコードに依存する.本研究で用いる粒子流体混成コードのタイムチャートを,図5に示す.2.6節の手順(1)タイムチャートの作成に対応する.図中の各サブルーチンの並列化については,

3.2節において議論する.

3.2 ハイブリッド分割法による計算全体の流れ

本研究で用いたプラズマ粒子流体混成コードの,並列化の流れについて説明する.本研究では,PE間のデータ通信に,MPI-2(Message-Passing Interface)関数を用いる.MPIは,現在,多くの並列計算機で標準化されている,代表的な通信ライブラリである.

プラズマ粒子流体混成コードの並列化の結果を,図6に示す.図は,4PEで並列化を行った例である.2.6節の手順(2)領域分割法と粒子分割法の適用,および,手順(3)各サブルーチンのMPI通信関数の決定が,これにあたる.

図6の(1),(5),(9),(14),(15)は,電磁場計算である.(2),(6),(7)(8),(10),(13)は電流および電子流体計算,(3)(4),(11)(12)が粒子計算を表している.以上のステップ番号は,図5のステップ番号と一致している.

(1)で,各PEは,担当する領域の磁場計算を行う.粒子計算のため,磁場の全データをすべてのPEが保持している必要がある.そのため,MPI_BCASTを用いてデータ通信を行う.次に(2)で,各PEが担当する領域の電流の計算を行う.電流の計算は磁場のデータを必要とする.(3)(4)では,粒子の位置,速度を時間的に進める.次に(5),(6)で,再度,磁場および電流値を求める.(7)では,各PEが担当する粒子のモーメント量を求める.MPI_ALLREDUCEを用いて,全領域のモーメント量の和をとる.(8)で,各PEは,担当する領域の電子流体計算を行う.(9)では,各PEが担当する領域の電場の計算を行う.電場のデータは磁場のデータと同様に,粒子計算を行う際に使用される.そのため,電場のデータをMPI_BCASTを用いて,すべてのPEにコピーする.(10)以降は,(1)~(9)と同様である.

4. 並列化による計算時間の評価式の導出

4.1 処理時間測定環境

本節では,2.6節において示した並列化手順の最後,すなわち,手順(4)各サブルーチンの計算および通信時間の評価,および,手順(5)各サブルーチンの並列化数決定について述べる.後に示すように,ハイブリッド分割法では,すべてのサブルーチンを,最大のPE数で並列化しないほうが速度向上率が良い場合がある.したがって,各サブルーチンの並列化数を決定するための,速度向上率の評価式が必要となる.本稿では,例として,3.1節のプラズマ粒子流体混成コードにおける定式化を行う.

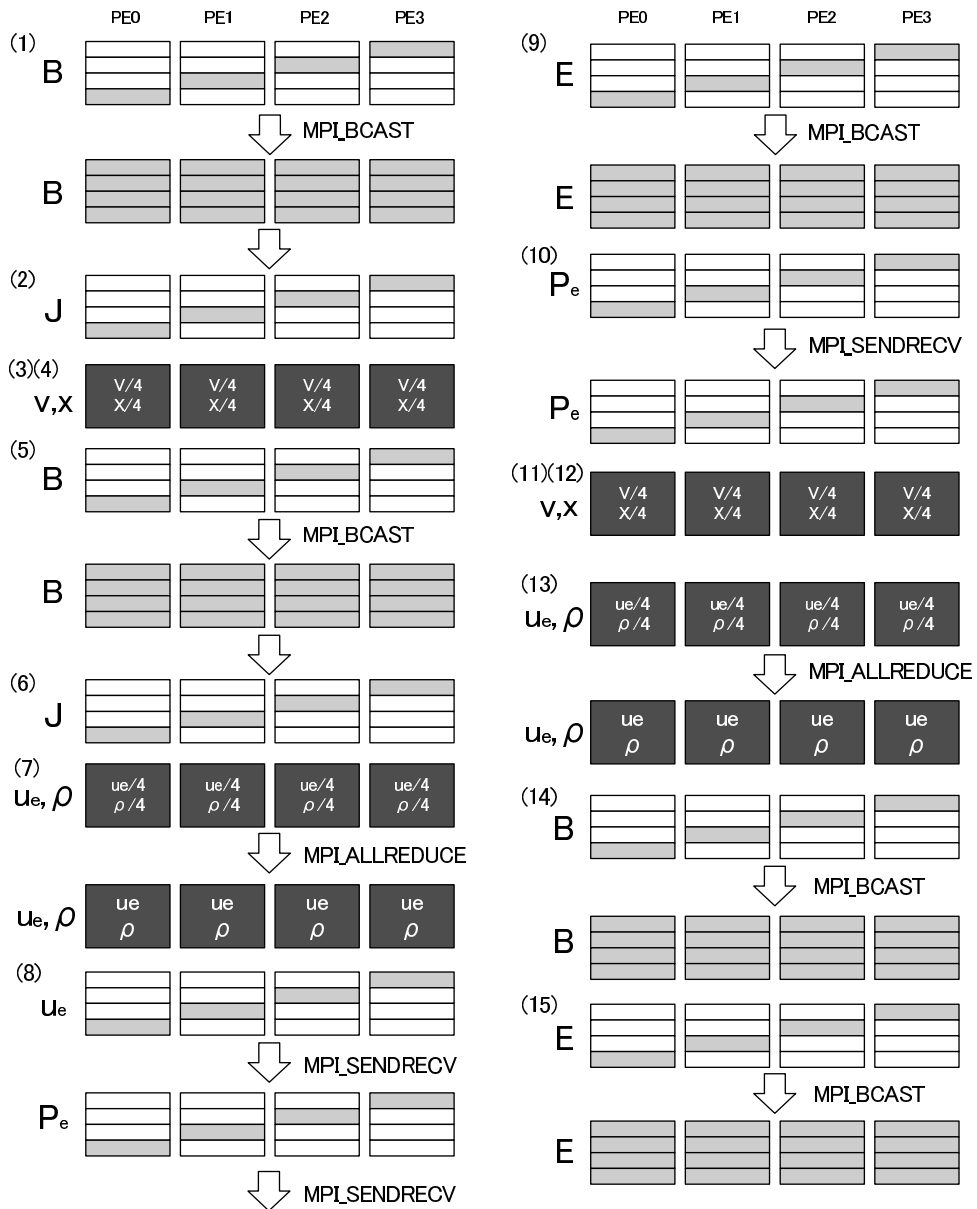


図6 プラズマ粒子流体混成コードの並列化処理の流れ (4PEの場合)
 Fig.6 Process of parallelization of a hybrid code (4PE).

本研究でハイブリッド分割法評価のために用いた並列計算機は、富士通社の並列計算機 AP3000 (以下 AP3000) である。用いたコンパイラは mpifrt で、最適化等は行っていない。AP3000 は、1つのノードが2つのプロセッサから構成される共有分散メモリ型の並列計算機である。本研究で用いた AP3000 は、それぞれのノードに 640 Mbyte のメモリが備わっている。本研究では、実行時のオプション設定により、16PE の分散メモリ型並列計算機として設定し、測定

を行った。実行コマンドは、mpirun である。

本稿で用いる MPI 通信の評価式は、ベンチマークプログラムにより確認する。ベンチマークプログラムとして、Pallas 社の b_eff benchmark (version 3.3)⁹⁾ を用いる。b_eff では、1 byte から 4 Mbyte までのデータ長の、通信速度特性を評価することができる。

4.2 全処理時間の定式化

ハイブリッド分割法における各サブルーチンの並列化数を n_i 、サブルーチンの全処理時間を $T_i(n_i)$ とす

る．ここで， i は，サブルーチンを識別する引数である．電場計算，磁場計算，粒子計算，電子流体計算，および電流計算の各サブルーチンを，それぞれ E, B, P, e ，および J と表す．最大並列化数 n の場合の，計算全処理時間 $T(n)$ は，次式で表される．

$$T(n) = \sum_{i \in \{E, B, P, e, J\}} T_i(n_i) \quad (10)$$

また，速度向上率 $S(n)$ は

$$S(n) = \sum_{i \in \{E, B, P, e, J\}} T_i(1) / \sum_{i \in \{E, B, P, e, J\}} T_i(n_i) \quad (11)$$

となる．

サブルーチン i の全処理時間は，実計算時間 $T_i^{\text{cal}}(n_i)$ と，PE 間でのデータ通信時間 $T_i^{\text{com}}(n_i)$ ，の和で与えられる．

$$T_i(n_i) = T_i^{\text{cal}}(n_i) + T_i^{\text{com}}(n_i) \quad (12)$$

本稿では，各サブルーチンの逐次計算時間，および 2 並列の場合の MPI 通信時間を，それぞれ既知とする．すなわち，各サブルーチン i の $T^{\text{cal}}(1)$ ， $T^{\text{SR}}(2)$ ， $T^{\text{BC}}(2)$ ， $T^{\text{AR}}(2)$ ，を既知として， n_i 並列における全処理時間の評価を行う．

4.3 実計算時間の評価式

プラズマ電磁粒子シミュレーションでは，各サブルーチン内での実計算のほとんどがループ処理になっており，逐次計算部分はわずかである．したがって， n_i 並列のサブルーチン i の実計算時間 $T_i^{\text{cal}}(n_i)$ は，

$$T_i^{\text{cal}}(n_i) = \frac{T_i^{\text{cal}}(1)}{n_i} \quad (n_i \geq 1) \quad (13)$$

となる．

4.4 MPI 通信時間の定式化

MPISENRECV の内部通信ロジック¹⁾を，図 7(a) に示す．通信に要する時間 $T^{\text{SR}}(n)$ は，次式で与えられる．

$$T^{\text{SR}}(n) = T^{\text{SR}}(2) \quad (n \geq 2) \quad (14)$$

MPLBCAST の内部通信ロジック¹⁾を，図 7(b) に示す．図より， n 並列（ただし n は 2 のべき乗）の場合， $\log_2 n$ 回の通信が必要となることが分かる．一方，一回の通信量は，2 並列の場合と比較して， $2/n$ 倍となり，全通信時間は $n/2$ 倍となる．以上より，通信に要する時間 $T^{\text{BC}}(n)$ は，次式により与えられる．

$$T^{\text{BC}}(n) = T^{\text{BC}}(2) \cdot \log_2(n) \quad (n \geq 2) \quad (15)$$

MPLALLREDUCE 通信の内部通信ロジック¹⁾を，図 7(c) に示す．通信に要する時間 $T^{\text{AR}}(n)$ は，次式により与えられると考えられる．

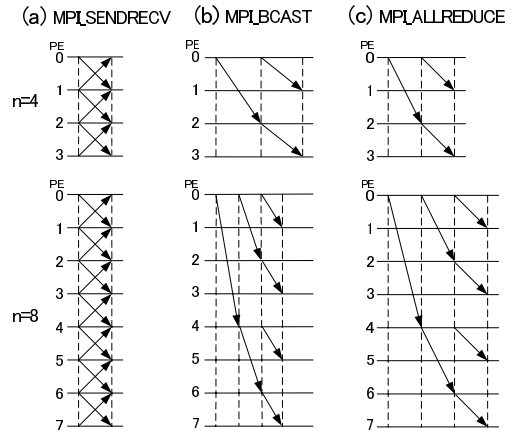


図 7 MPI 関数の内部通信ロジック（4 並列および 8 並列の場合）
(a) MPISENRECV, (b) MPLBCAST, (c) MPLALLREDUCE

Fig. 7 Internal logics for MPI functions (4 and 8 parallelization): (a) MPISENRECV, (b) MPLBCAST and (c) MPLALLREDUCE.

$$T^{\text{AR}}(n) = T^{\text{AR}}(2) \cdot \log_2 n \quad (n \geq 2) \quad (16)$$

4.5 速度向上率の評価式

本節では，4.3 節と 4.4 節の結果を用いて，プラズマ粒子流体混成コードの速度向上率の定式化を行う．図 6 より，粒子計算における通信には，MPLALLREDUCE を用いる．式 (12)，式 (16) より，粒子の全計算時間は，

$$T_P(n_P) = \frac{T_P^{\text{cal}}(1)}{n_P} + T_P^{\text{com}}(2) \cdot \log_2 n_P \quad (17)$$

となる．

図 6 より，電場計算サブルーチンは，MPLBCAST を用いる．電場全処理時間は，式 (12)，式 (15) より，

$$T_E(n_E) = \frac{T_E^{\text{cal}}(1)}{n_E} + T_E^{\text{com}}(2) \cdot \log_2 n_E \quad (18)$$

となる．磁場についても同様であるので，

$$T_B(n_B) = \frac{T_B^{\text{cal}}(1)}{n_B} + T_B^{\text{com}}(2) \cdot \log_2 n_B \quad (19)$$

である．

図 6 より，電子流体の MPI 通信は MPISENRECV を用いる．式 (12)，式 (14) より，電子流体全処理時間は，

$$T_e(n_e) = \frac{T_e^{\text{cal}}(1)}{n_e} + T_e^{\text{com}}(2) \quad (20)$$

となる．

電流計算は，MPI 通信が不要である．したがって，電流全処理時間は，

$$T_J(n_J) = \frac{T_J^{\text{cal}}(1)}{n_J} \quad (21)$$

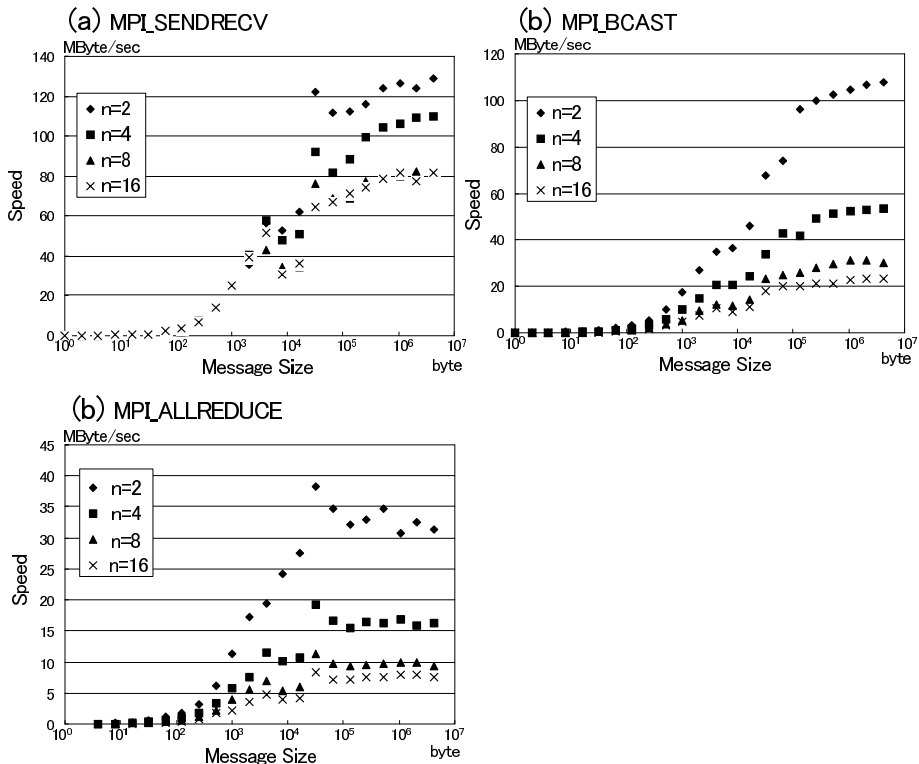


図 8 Pallas 社 b_{eff} (ver. 3.3)¹⁹⁾ によるベンチマーク測定結果 . (a) MPI_SENDRECV , (b) MPI_BCAST , (c) MPI_ALLREDUCE の通信速度
 Fig. 8 MPI benchmark test estimated by b_{eff} (ver. 3.3)¹⁹⁾: for (a) MPI_SENDRECV, (b) MPI_BCAST and (c) MPI_ALLREDUCE.

である .

以上より, 式 (17), 式 (18), 式 (19), 式 (20), 式 (21) を用いると, 粒子流体混成コードの全処理時間 $T(n)$ を, 式 (10) から求めることができる .

4.6 MPI 関数評価式の検証

図 8 (a) に, b_{eff} による MPI_SENDRECV を用いたデータ通信速度のベンチマーク測定結果を示す . それぞれ, 2, 4, 8, 16 並列におけるデータ通信速度特性を表している . 通信データ長は, 1 byte から 4 Mbyte である . データ長が約 1 Mbyte 以上において, ピークバンド幅を達成している .

図 9 (a) に, 図 8 (a) のベンチマークの結果と式 (14) の比較を示す . 図は, 並列化数ごとの通信時間を示す . データ通信時間は, PE 数の増加にともない, 上昇する傾向がある . これは, MPI_SENDRECV 関数の開始および終了処理時間が, 実通信時間と比較して無視できず, かつ, PE 数増加にともない増大するためであると考えられる . しかし, 図 9 (a) で示される MPI_SENDRECV の通信時間は, 後述の図 8 (b) や図 8 (c) の通信関数と比べ, 十分に短い . したがって,

本稿では, 式 (14) を仮定する . すなわち, 2PE での MPI_SENDRECV の通信時間をあらかじめ測定することにより, 任意の PE 数での通信時間が予測できる .

図 8 (b) に, b_{eff} による通信速度の測定結果を示す . それぞれ, 2, 4, 8, 16 並列における, データ通信速度特性を表している . 通信データ長は 1 byte から 4 Mbyte である . ただし, MPI_BCAST では, 並列化数によってデータ長が異なるため, 図 8 (b) の横軸は, 2 並列時のデータ長で示す . 通信データ長が約 500 Kbyte 以上で, ピークバンド幅を達成している .

図 9 (b) に, 図 8 (b) のベンチマーク結果と, 式 (15) との比較を示す . 並列化数やデータ長にかかわらず, ベンチマーク測定結果と式 (15) は, よい一致を示すことが分かる . すなわち, 2PE での MPI_BCAST の通信時間をあらかじめ測定することにより, 任意の PE 数での通信時間が予測できる . ただし, 通信量が 128 Kbyte 以上では, ベンチマーク値は, 式 (15) をわずかに上回る傾向がある .

図 8 (c) に, b_{eff} による通信速度の測定結果を示す . それぞれ, 2, 4, 8, 16 並列における, データ通信速

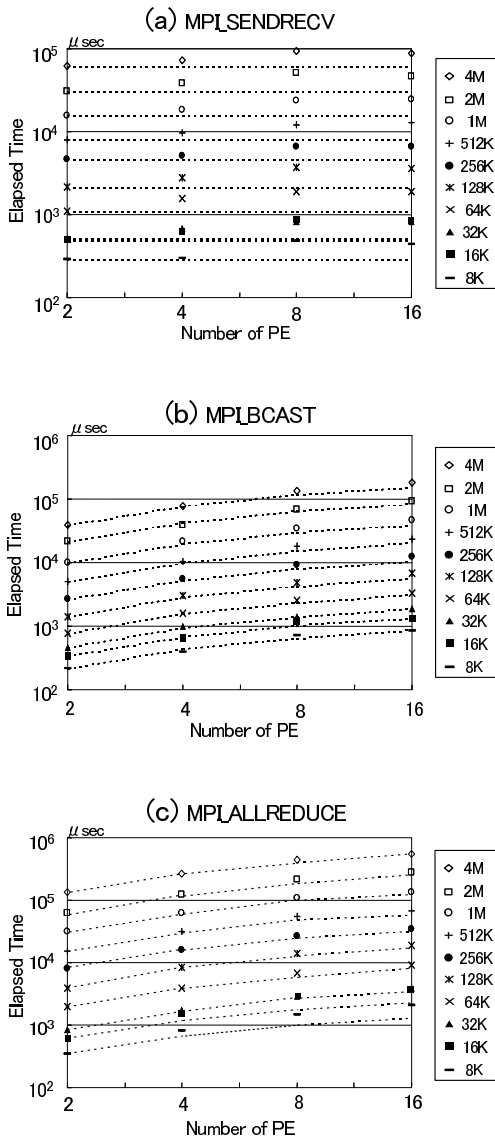


図9 MPI関数のベンチマーク実測値と理論式(14), (15), (16)の比較。点線が理論値を表す。(a) MPLSENDRECV, (b) MPLBCAST, (c) MPLALLREDUCEの測定時間

Fig. 9 Computational and theoretical period of MPI communications. The dashed lines are theoretical curves: (a) MPLSENDRECV, (b) MPLBCAST and (c) MPLALLREDUCE.

度特性を表している。通信データ長は、4 byte から 4 Mbyte である。ばらつきはあるが、約 100 Kbyte 以上で、ピークバンド幅を達成している。

図 9(c) に、図 8(c) のベンチマーク結果と式 (16) との比較を示す。図より、64 Kbyte 以上のデータ長では、ベンチマーク測定時間が、式 (16) とよい一致を示すことが分かる。すなわち、MPLBCAST と同様、2 並列時での MPLALLREDUCE の通信時間をあら

表 1 プラズマ粒子流体混成コードの各サブルーチンの逐次実計算時間と 2 並列時の MPI 通信時間 (AP3000 による)。粒子数は、(a) 640/grid, (b) 320/grid, (c) 160/grid

Table 1 Calculation and MPI communication times in 2 PE for each subroutine: Particle numbers are (a) 640/grid, (b) 320/grid, and (c) 160/grid, respectively.

	T	Time (sec)
E	$T_E^{cal}(1)$	0.13
	$T_E^{com}(2)$	0.08
B	$T_B^{cal}(1)$	0.17
	$T_B^{com}(2)$	0.22
P(a)	$T_P^{cal}(1)$	137.86
	$T_P^{com}(2)$	0.42
P(b)	$T_P^{cal}(1)$	68.90
	$T_P^{com}(2)$	0.19
P(c)	$T_P^{cal}(1)$	34.53
	$T_P^{com}(2)$	0.16
e	$T_e^{cal}(1)$	0.43
	$T_e^{com}(2)$	0.01
J	$T_J^{cal}(1)$	0.11
	$T_J^{com}(2)$	0.00

かじめ測定することにより、任意の並列化数での通信時間が予測できる。

5. プラズマ粒子流体混成コードの速度向上率の評価

5.1 計算実測環境

本節では、プラズマ粒子コードの速度向上率を実測し、4 章で求めた評価式と比較する。実験は、4 章と同様、プラズマ粒子流体混成コードを用いる。特に、与えられた格子点数や粒子数などのシミュレーションパラメータに対して、最高の速度向上率を得ることを目標とする。用いるシミュレーションパラメータは、格子点数 128×128 の 2 次元である。粒子数は、2.2 節の図 2 を考慮して、格子点あたり 160, 320, 640 個とする。測定実験には、前節同様、AP3000 を用いる。なお、実測値は、すべて 8 回の計算の合計値である。

5.2 速度向上率の予測

本節では、4 章で求めた式 (10) および式 (11) より、 n 並列時の、速度向上率の予測値を求める。4.4 節によると、式 (10) の各項を求めるためには、各サブルーチンの逐次計算の実計算時間と、2 並列時の通信時間の実測値が必要である。これらの測定結果を、表 1 に示す。格子点あたりの粒子数は (a) 640 個、(b) 320 個、(c) 160 個で、格子点数は 128×128 である。

表 1 の基準値を用いて、速度向上率の予測値を、式 (10)、式 (11) を用いて求めることができる。5.2 節では、すべてのサブルーチンを、同じ並列化数で並列化

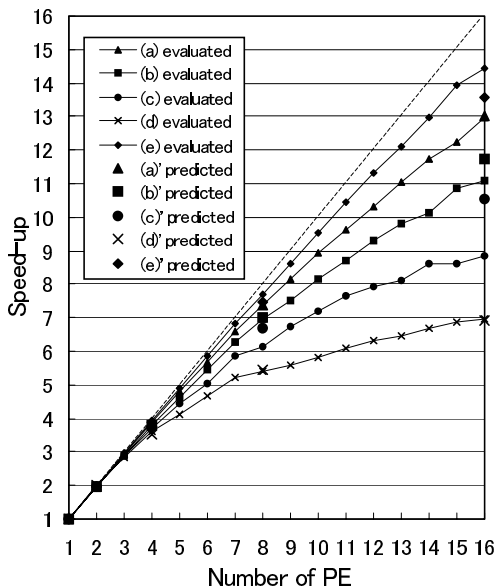


図 10 プラズマ粒子流体混成コードの速度向上率の予測値と実測値 (AP3000 による). 粒子数は, (a) 640/grid, (b) 320/grid, (c) 160/grid, (d) 40/grid, (e) 640/grid. 格子点数は, (a), (b), (c), (e) が 128×128 , (d) が 512×512 . 電磁場サブルーチンの並列化数は, (a), (b), (c), (d) が最大並列化数, (e) は逐次計算. (a)', (b)', (c)', (d)', (e)' は予測値, (a), (b), (c), (d), (e) は実測値

Fig. 10 Predicted and evaluated speed-up of present hybrid code via AP3000: Particle number is (a) 640/grid, (b) 320/grid, (c) 160/grid, (d) 40/grid and (e) 640/grid. Grid number is 128×128 in (a), (b), (c) and (e), and 512×512 in (d). Parallelization number of EM Field Routine corresponds to maximum PE number in (a), (b), (c) and (d), and no parallelization in (e). (a), (b), (c), (d) and (e) are evaluated values respectively, and those with dash are predicted values.

した場合について考える. 図 10 に, 2, 4, 8, 16 並列時の, (a), (b), (c) の速度向上率の予測値を, それぞれ (a)', (b)', (c)' として示す. なお, 16 並列では, 予測値は, (a)' 13.04, (b)' 11.75, (c)' 10.54 となった.

5.3 速度向上率の実測評価

本節では, プラズマ粒子流体混成コードを用いた, 16 並列までの速度向上率の実測を行う. 5.3.1 項では, すべてのサブルーチンを, 最大並列化数で並列化した場合の評価を行う. 5.3.2 項では, 各サブルーチンごとの処理時間を調べ, 最も高い速度向上率を得るために, サブルーチンごとに並列化数を設定する.

5.3.1 すべてのサブルーチンで並列化数が等しい場合

まず, すべてのサブルーチンを, 等しい並列化数 n (=最大並列化数) で並列化する場合について考える.

5.2 節 (a), (b), (c) の 3 つのパラメータについて, 16 並列までの速度向上率を実測した結果を, 図 10 (a), (b), (c) に示す. 16 並列では, (a) 12.96, (b) 11.09, (c) 8.84 であった. (a) は予測値とよい一致を示しているが, (b) と (c) は, 予測値よりも低くなる. この理由については, 5.4 節において議論する.

図 10 より, 速度向上率は, 格子点数が等しい場合には, 粒子数が多いほど向上することが読み取れる. これは, 粒子計算サブルーチンと比較して, 電磁場計算サブルーチンの並列化効率が悪いからである. 図 11 に, 5.2 節 (a) の, サブルーチンごとの実計算時間と通信時間を示す. (c) 粒子計算, (d) 電子流体計算, および (e) 電流計算サブルーチンは, 通信時間に比して実計算時間が大きいため, スケーラビリティが高い. 一方, (a) と (b) の電磁場サブルーチンは計算量が小さく, また, スケーラビリティの低い MPLBCAST を通信関数として用いているため, 並列化数が大きいほど, 全処理時間が増加する. (c) の全処理時間が短くなると, 相対的に, (a) や (b) の効果が大きくなる.

速度向上率の低下は, 粒子数が等しい場合には, 格子点数が多いほど, 顕著になる. 比較のため, 粒子数が 5.2 節 (a) と等しく, 格子点数が 512×512 のパラメータによる速度向上率を調べた. その結果を, 図 10 (d) に示す. 同図 (a) と比較して, 速度向上率の低下が顕著に見られる.

5.3.2 サブルーチンごとに並列化数を設定する場合

図 11 では, 電磁場サブルーチンにおいて, 並列化数が大きいほど, 全処理時間が増加する傾向が見られた. この結果は, サブルーチンごとに並列化数を変更することにより, さらに速度向上率を向上できることを示唆している. 本項では, サブルーチンごとの並列化数を個別に設定することにより, 図 10 (a) の速度向上率の向上を試みる.

図 11 より, 16PE システムで全処理時間が最小となるのは, 粒子, 電子流体, および電流サブルーチンが, それぞれ, 16, 16, 15 並列の場合である. 一方, 電磁場サブルーチンでは, PE 数にかかわらず, 逐次計算が最も全処理時間が小さい. このように, 並列化数をサブルーチンごとに個別に設定して, 5.2 節 (a) のパラメータによって求めた, 16 並列までの速度向上率の実測値と予測値を, 図 10 (e) および (e)' に示す. 16 並列時の速度向上率は, 予測値 (e)' が 13.59, 実測値 (e) が 14.45 となった. したがって, 同図 (a) の実測値 12.96 と比べ, 速度向上率は約 12% 改善された. なお, 実測値 (e) が予測値 (e)' よりも高い速度向上率を示している理由については, 5.4 節で議論する.

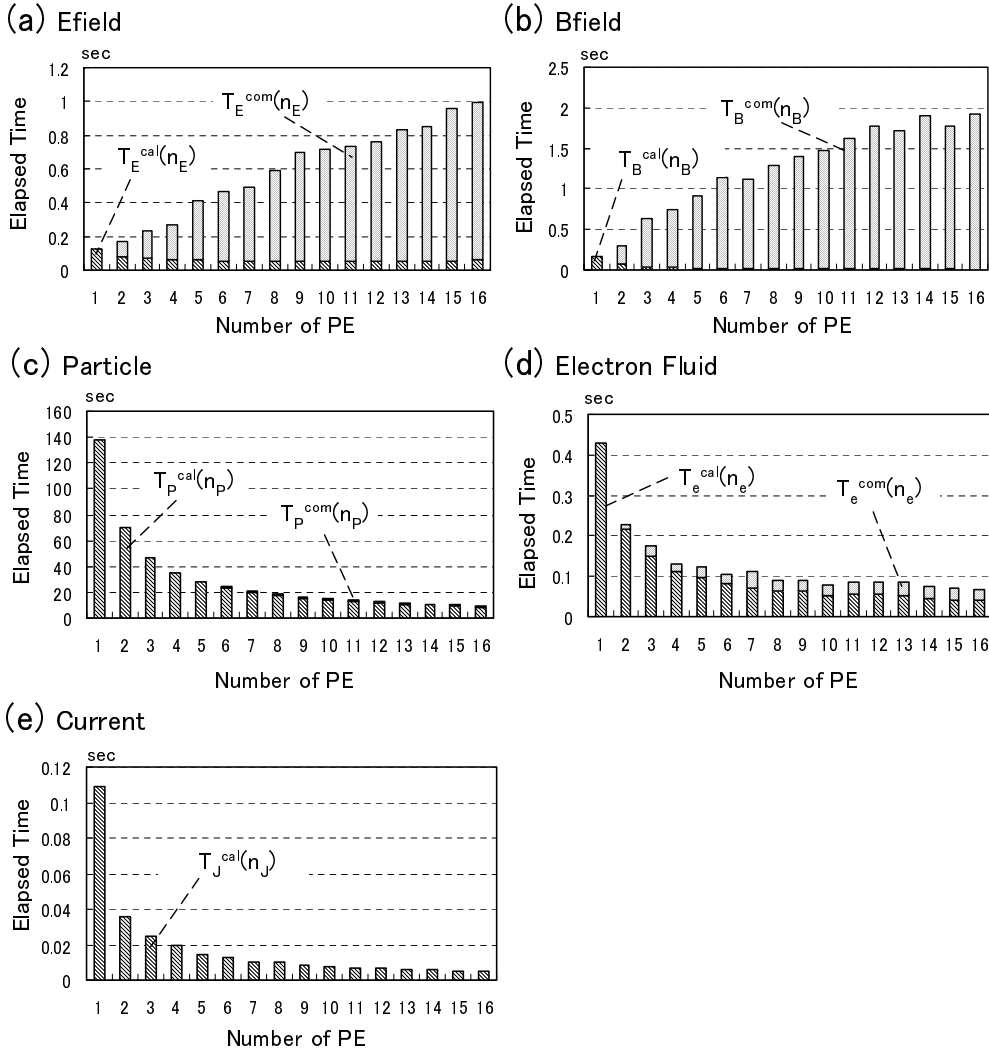


図 11 AP3000 によるハイブリッドコードの各サブルーチンの実計算時間とデータ通信時間：
 (a) 電場計算サブルーチン, (b) 磁場計算サブルーチン, (c) 粒子計算サブルーチン, (d)
 電子流体計算サブルーチン, (e) 電流計算サブルーチン

Fig. 11 Calculation duration and communication duration in hybrid code via AP3000: (a) Electric Field Solver Subroutine, (b) Magnetic Field Solver Subroutine, (c) Particle Solver Subroutine, (d) Electron Fluid Solver Subroutine and (e) Current Solver Subroutine.

図 10(e) では、電流計算サブルーチンは、15 並列の場合に処理時間が最小となるが、実測では 16 並列化した。その理由についても、5.4 節で述べる。

5.4 速度向上率測定の考察

本節では、5.3.1 項と 5.3.2 項の、16 並列時の速度向上率の、予測値と実測値の差について考察する。

図 10 では、16 並列時の速度向上率は、予測値 (b)', (c)' よりも実測値 (b), (c) の方が低かった。これは、MPLBCAST の影響である。図 9 (b) によると、MPLBCAST は、16 並列において、通信時間が理論

値よりも上回っている。すなわち、電場計算サブルーチンと磁場計算サブルーチンの全処理時間が予測値を上回るため、5.3.1 項の差が発生したと考えられる。この傾向は、格子点数に対する粒子数が少なくなるほど、顕著になる。

一方、図 10 (e) で実測値が予測値を上回る原因は、MPLALLREDUCE である。図 12 に、5.3.1 項の (b) と (e) の、MPLALLREDUCE 通信の実測値と予測値を示す。(b) および (e) が実測値、(b)' および (e)' が、式 (16) による予測値である。(b) と (e) より、デー

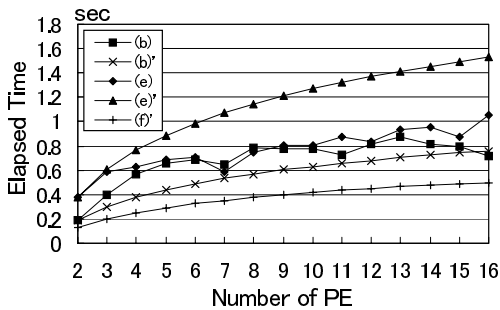


図 12 5.3.1 項 (b) と 5.3.2 項 (e) の, ALL-REDUCE の通信時間の実測値と予測値. (b) は 5.3.1 項の (b) の実測値, (b)' は (b) の予測値, (e) は 5.3.2 項の (e) の実測値, (e)' は (e) の予測値, (f) は b_{eff} より求めた予測値

Fig. 12 Evaluated and predicted duration time of ALL-REDUCE in (b) and (e). (b) is evaluated time of (b), (b)' is predicted time of (b), (d) is evaluated time of (e), (d)' is predicted time of (e) and (f) is predicted time via b_{eff} .

タ通信量が等しい場合には, 実測結果も, 比較的よい一致を示す. しかし, $n=2$ における実測値のばらつきが, $n=16$ において, (b)' と (e)' の差異を生じる. その結果, (e) では, 実測値が予測値を上回った. MPI-ALLREDUCE では, 通信の開始・終了処理時間が無視できない. 開始・終了処理時間はランダムに発生し, $n=2$ のばらつきの原因となる. したがって, MPI-ALLREDUCE を用いる場合には $n=2$ 以外の測定も行い, 評価式 (16) のフィッティングを, より正確に行うことが望ましい.

なお, 図 9 (c) で実測と予測がよい一致を示したのは, b_{eff} が, 開始・終了処理時間を除外して測定ができるためである. 比較のため, 同じ通信量の場合の, ベンチマークプログラム b_{eff} による予測値を, 図 12 (f) として示す.

次に, 5.3.2 項で, 電流計算サブルーチンを 16 並列化した理由を述べる. ハイブリッド分割法では, 最小の処理時間となる並列化数が, サブルーチンごとにばらつくことがある. 図 11 でも, 電子流体計算は 16 並列, 電流計算は 15 並列で, 全処理時間が最小となった. しかし, 図 6 によると, サブルーチンの並列化は他のサブルーチンとの関係に依存するため, 独立に設定できない. たとえば, 電流計算と電子流体計算のサブルーチンの並列化数を異なる値で設定すると, 両者の間に新しい通信が発生する. このとき, 速度向上率は, 逆に低下することになる.

電磁粒子シミュレーションコードでは, 一般に, 各サブルーチンの並列化条件が従属的である. したがって, ハイブリッド分割法の最適並列化数決定方法は, コードに依存し, 一般化が容易ではない. 今後の研究

が必要である.

6. おわりに

これまでの電磁粒子シミュレーションの並列化手法は, 領域分割法と粒子分割法に大別される. 本研究で扱う並列化手法は, この 2 つの手法を組み合わせた, ハイブリッド分割法である. ハイブリッド分割法は, 粒子計算サブルーチンを粒子分割し, 電磁場および流体計算サブルーチンを領域分割する. サブルーチン間でデータ交換をしながら, 計算を進める.

ハイブリッド分割法では, 各サブルーチンの実計算部の逐次処理時間と, 2 並列時の通信時間を既知として, 速度向上率を予測できる. 本研究では, 速度向上率評価のための定式化を行った. さらに, ベンチマークプログラムにより, 評価式が有効であることを確かめた.

次に, プラズマ電磁粒子コードの 1 つである, プラズマ粒子流体混成コードに, ハイブリッド分割法を適用した. MPI 関数を用いてプラズマ粒子流体混成コードを並列化し, 速度向上率を実測した. 代表的なパラメータにより, 並列計算機 AP3000 で実測したところ, 16 並列の場合, 予測値は 13.04, 実測値は 12.96 であった.

ハイブリッド分割法では, MPI 関数や通信データ量に応じて, 計算サブルーチンごとに, 並列化数を個別に設定できる. 一般に, 粒子計算のサブルーチンは, 計算時間が通信時間よりも十分に長い. そのため, 最大の PE 数を並列化数とすることで, 最高の速度向上率が得られる. 一方, 電磁場および流体計算のサブルーチンでは, 並列化数が PE 数となるとは限らない. たとえば, 本研究で 16PE を用いた場合に, 評価式により速度向上率を評価したところ, 電磁場計算の並列化数は 1, すなわち逐次計算の場合に, 速度向上率が最高となることが分かった.

そこで, ハイブリッド分割法で, サブルーチンごとに並列化数を設定して速度向上率を測定した. 電磁場サブルーチンを逐次計算した場合, 16 並列では, 予測値が 13.59, 実測値が 14.45 の速度向上率を得た. すなわち, 約 12% の速度向上率の改善がなされた.

ハイブリッド分割法では, 粒子計算に必要な電磁場の情報を, すべての PE が保持していなければならないという, 主記憶容量上の不利点がある. しかし, 最近のプラズマ粒子シミュレーションでは, 格子点数に対する粒子数が増加する傾向にあり, 大きな不利点となりえない. 逆に, すべての配列変数が各 PE に均等に割り当てられるため, 並列計算効率の妨げとなる

PE間の同期²⁰⁾を最小限に抑えることができる。また、ハイブリッド分割法は、動的領域分割法と比較して、粒子データ通信の必要がない。そのため、データ通信量は、パラメータや問題に依存せず一定となる。ほぼ完全な負荷分散が実現できることが、ハイブリッド分割法の利点の1つである。

なお、流体量を持たないプラズマ電磁粒子コード(たとえば KEMPO コード¹¹⁾)でも、電磁場方程式は領域分割できるため、ハイブリッド分割法は適用可能である。すなわち、本手法は、ほとんどの標準的なプラズマ粒子コードに適用でき、汎用性は高いと言える。

粒子流体混成コードは、すでに、宇宙プラズマ^{15),16)}、核融合プラズマ^{17),18)}などの、さまざまなプラズマ研究分野で利用されている。本研究で用いた2次元粒子流体混成ハイブリッド並列コードは筆者らにより公開されており、今後も、さまざまなプラズマ研究分野において、利活用されることが期待される。

謝辞 本研究は、科学技術振興事業団計算科学技術活用型特定研究開発推進事業での成果である。

参 考 文 献

- 1) 天野英晴：並列コンピュータ，昭晃堂(1996)。
- 2) 湯浅太一，安村通晃，田中登志之：はじめての並列プログラミング，共立出版(1998)。
- 3) 松原正純，板倉憲一，朴 泰祐：超並列計算機 CP-PACS における大規模分子動力学法シミュレーション，情報処理学会論文誌，Vol.40, No.5, pp.2172-2182 (1999)。
- 4) Akiyama, Y., Misou, K., Omura, Y. and Matsumoto, H.: Parallelization of Space Plasma Particle Simulation, *High Performance Computing*, Polychronopoulos, et al. (Eds.), Lecture Notes in Computer Science 1336, pp.281-292, Springer (1997)。
- 5) 蔡 東正，安室秀則，李 堯亭，陸 全明，肖池階：Linux PC クラスタを用いた並列粒子シミュレーション，情報処理学会論文誌，Vol.42, No.12, pp.124-131 (2001)。
- 6) 林 亮子，堀口 進：固定分配セルを用いた動的負荷分散法による並列分子動力学法シミュレーション，情報処理学会論文誌，Vol.40, No.5, pp.2152-2162 (1999)。
- 7) Berger, M.J. and Collela, P.: Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations, *J. Comput. Phys.*, Vol.53, pp.484-512 (1984)。
- 8) Berger, M.J. and Collela, P.: Local Adaptive Mesh Refinement for Shock Hydrodynamics, *J. Comput. Phys.*, Vol.82, pp.64-84 (1989)。
- 9) 古山彰一，松澤照男：動的負荷分散を用いた適応格子法の並列計算，情報処理学会論文誌，Vol.38, No.9, pp.1869-1877 (1997)。
- 10) 矢部 孝，川田重夫，福田昌宏：シミュレーション物理入門—超粒子モデルの世界，朝倉書店。
- 11) Matsumoto, H. and Omura, Y.: *Computer Space Plasma Physics: Simulation Techniques and Software*, Terra Scientific Publishing Company, Tokyo (1993)。
- 12) Ferrard, R.D., Liewer, P.C. and Decyk, V.K.: Dynamic Load Balancing for a 2D Concurrent Plasma PIC Code, *J. Comput. Phys.*, Vol.109, pp.329-340 (1993)。
- 13) Bunemann, O.: Time reversible difference procedures, *J. Comput. Phys.* (1967)。
- 14) Amdahl, G.M.: Validity of the Single-Processor Approach to Archiving Large Scale Computing Capabilities, *Proc. Spring Joint Computer Conference*, Vol.30 (1967)。
- 15) Thomas, V.A., Winske, D., Thomsen, M.F. and Onsager, T.G.: Hybrid simulations of a hot flow anomaly, *J. Geophys. Res.*, Vol.96, p.11625 (1991)。
- 16) Winske, D. and Leroy, M.M.: Hybrid simulation techniques applied to the Earth's bow shock, *Computer Simulations of Space Plasmas*, Matsumoto, H. and Sato, T. (Eds.), p.255, D. Reidel/Terra Sci., Hingham MA (1984)。
- 17) Imamura and Tokuda: A hybrid computing by coupling different architectural machines, a case study for takamak plasma simulation, *Proc. Int. Conf. PDCS99* (1999)。
- 18) Tokuda, S., Naitou, H. and Lee, W.W.L.: A Particle-Fluid Hybrid Simulation Model Based on Nonlinear Gyrokinetics, *Journal of Plasma and Fusion Research*, Vol.74, No.1, pp.44-53 (1998)。
- 19) Pallas MPI Benchmarks—PMB, Part MPI. <http://www.pallas.com>
- 20) 岡本一晃，松岡浩司，廣野英雄，横田隆史，佐藤三久，坂井修一：超並列計算機のための同期処理機構とその評価，情報処理学会論文誌，Vol.40, No.3, pp.1245-1256 (1999)。

(平成 13 年 8 月 19 日受付)

(平成 13 年 10 月 23 日再受付)

(平成 13 年 12 月 11 日再々受付)

(平成 14 年 3 月 25 日採録)



村田 健史(正会員)

平成7年京都大学大学院工学研究科研究指導認定退学。同年愛媛大学工学部助手。平成8年愛媛大学工学部講師。宇宙プラズマ計算機実験、宇宙電波解析システムおよび無線ネットワークの研究に従事。博士(工学)。電子情報通信学会、応用数理学会、地球電磁気・地球惑星圏学会、アメリカ地球物理学会(AGU)各会員。



上岡 功治(正会員)

平成12年愛媛大学工学部情報工学科卒業。同年同大学大学院工学研究科博士前期課程在籍。プラズマ粒子シミュレーションの並列計算の研究に従事。



高橋 誠治(正会員)

平成10年愛媛大学大学院工学研究科博士前期課程修了。NTT サービスインテグレーション基盤研究所。セキュリティシステム開発に従事。



岡田 雅樹(正会員)

平成6年京都大学大学院工学研究科研究指導認定退学。同年文部省国立極地研究所助手。宇宙飛翔体電磁波環境シミュレーションおよび地球磁気圏物理学の研究に従事。博士(工学)。地球電磁気・地球惑星圏学会、アメリカ地球物理学会(AGU)各会員。



上田 裕子

平成4年電気通信大学大学院電気通信学研究科単位取得退学。同年千葉大学工学部助手。平成13年宇宙開発事業団技術研究本部副主任研究員。宇宙プラズマおよび電磁波動計算機シミュレーションの研究に従事。博士(工学)。電子情報通信学会、地球電磁気・地球惑星圏学会、アメリカ地球物理学会(AGU)各会員。



大村 善治

昭和60年京都大学大学院工学研究科博士後期課程電気工学第二専攻研究指導認定退学。同年京都大学工学部助手。昭和63年京都大学超高層電波研究センター助手。平成元年京都大学超高層電波研究センター助教授。平成12年京都大学宙空電波科学研究センター教授。専門は電子ビーム不安定性と静電孤立波の計算機実験および電波科学計算機実験システムの開発。アメリカ地球物理学会(AGU)、地球電磁気・地球惑星圏学会各会員。



松本 紘

昭和40年京都大学工学部電子科卒業。昭和42年同大学大学院工学研究科修士課程了。昭和52年同大超高層電波研究センター教授。平成4~10年同大超高層電波研究センター長。平成12年同大宙空電波科学研究センター教授。現在に至る。文部科学省宇宙科学研究所客員教授併任。文部科学省核融合科学研究所客員教授併任。独立行政法人通信総合研究所客員研究官およびアドバイザー。中国武漢大客座教授。工学博士。地球電磁気・地球惑星圏学会評議員。国際電波科学(URSI)会長。JGR アジア・太平洋地域編集長。河上記念財団作工学会賞。昭和50年日本電磁気学会田中館賞。平成5年。平成10年NASA Group Achievement Award。平成11年情報通信月間推進協議会志田林三郎賞。平成12年アンテナ・伝播国際シンポジウム論文賞受賞。アメリカ地球物理学会フェロー、IEEE シニア各会員。