

5次元ブラソフコードによる

無衝突プラズマシミュレーションの性能評価

梅田 隆行

名古屋大学太陽地球環境研究所

深沢 圭一郎

九州大学情報基盤研究開発センター

1. はじめに

我々が住む宇宙の99.99%以上の体積はプラズマと呼ばれる電離気体で占められている。宇宙空間に存在するプラズマの大部分は密度が非常に小さく無衝突状態にあり、宇宙プラズマ（無衝突プラズマ）を理解することは、宇宙の本質的な理解につながる。地球周辺の宇宙環境は、太陽から放出された高速のプラズマ流である“太陽風”及び、太陽風が運ぶ惑星間空間磁場（太陽の固有磁場）と、地球の固有磁場との相互作用によって複雑な磁気圏構造を形成している。プラズマ放出現象をはじめとする太陽の様々な変動により、人工衛星の故障や宇宙飛行士の被曝に繋がる地球磁気圏の環境変動が引き起こされ、これを宇宙天気と呼ぶ。近年の国際宇宙ステーションでの活動や人工衛星打ち上げなど、日本においても宇宙利用が現実的になってきており、宇宙天気の子報・予測に繋がる宇宙プラズマ研究は非常に重要である。

地球磁気圏内には、プラズマの密度や温度などの物理パラメータが異なる様々な領域が生じる。その領域間の境界層で現れる不安定性（平衡状態の破れ）は、磁気圏の変動に大きな影響を与えていると考えられている。グローバル磁気圏構造に対して、境界層不安定性は中間（メゾ）スケール現象と呼ばれる。これらのグローバル及び中間スケールの現象は、ブラソフ（無衝突ボルツマン）方程式の0次・1次・2次のモーメントを取ることによって求められる磁気流体力学（MHD）方程式によって記述される。しかし、近年の科学衛星による高精度な「その場」観測では、中間スケールの不安定性は、MHD方程式で記述できるマクロ物理過程と粒子の運動論方程式によって記述できるミクロ物理過程が結合していることを示唆している。これらのマルチスケールの磁気圏変動である宇宙天気を真に理解するためには、全てのスケールをシームレスに扱える運動論方程式（第一原理）によるシミュレーションが本質的である。

プラズマの運動論シミュレーションには2つの手法がある。1つは、プラズマ粒子であるイオンや電子などの個々の荷電粒子の運動を、ニュートン-ローレンツ方程式により解き進めるPIC（Particle-In-Cell）法である。格子（Cell）上に定義された電磁場中を粒子が動きまわることから、このように呼ばれている。宇宙空間に存在する膨大な数の荷電粒子を有限の計算機資源で扱うことは不可能であるため、ある程度まとまった数の荷電粒子の集団を1つの“超”粒子として扱う。PIC法はその数値解法の完成度が高く、プラズマ科学分野では広く用いられている。しかし、超粒子として扱うことにより熱雑音が大きくなること、電荷密度や電流密度などの荷電粒子の運動に起因する場の量を格子上に割り振る際に生じる高波数モードが数値誤差として蓄積すること、さらに並列化の際に負荷のバランス（各プロセス内の粒子数の均一性）を

保つために特殊なデータ分割が必要になることなどの欠点がある。

一方もう1つの運動論的手法であるブラソフ法は、位置-速度位相空間に定義されたプラズマ粒子の分布関数の時間発展を、ブラソフ方程式により直接解き進める方法である。格子上に定義された分布関数は熱雑音を持たず、また流体シミュレーションと同様に領域分割による並列計算も容易である。しかし、ブラソフ方程式は実空間3次元及び速度空間3次元の計6次元を扱う方程式であり、コンピュータで解くには膨大なリソースを必要とする。このため、その手法の開発はあまり進んでいない。実際、ここ数年のHPCプロジェクトによる計算機環境の飛躍的な向上によって手法の開発が進み、実空間2次元及び速度空間3次元の5次元テストシミュレーションがようやく実行できるようになった段階である。

本研究の最終的な目的は、プラズマシミュレーションとしては「次々々」世代の技術にあたる6次元第一原理ブラソフシミュレーション手法を世界に先駆けて確立し、プラズマ科学に基づいた宇宙天気の実現に貢献することにある。そのために本プロジェクトでは、5次元問題を扱うブラソフコードの性能評価を、国内で現存する超並列計算機の中でも特にコア数の多い東京大学情報基盤センターのFX10上で行う。本研究グループではこれまで、東京大学、名古屋大学、九州大学等のスーパーコンピュータシステムにおいて全ノード計測を行ってきており、また磁気リコネクション、ケルビン-ヘルムホルツ不安定性、太陽風-非磁化天体相互作用などの物理計算への適用に、世界で唯一成功している[Umeda et al., 2010a,b; Umeda et al., 2011; Umeda, 2012; Umeda et al., 2012a]。2012年5月には、東京大学情報基盤センターの大規模HPCチャレンジにおいてFX10を用いた計測を行った（電磁流体コードによる惑星磁気圏シミュレーション性能測定、代表：深沢）。表1に富士通製システムを用いてこれまでに計測を行ってきたブラソフコードの単コア性能を示す。

本プロジェクトに先立ち、これらの全ノード計測において既知となった問題点を以下に示す。

- FX1とFX10で、実効効率に大きな差がある。FX1の15%に対して、FX10では11.6%しか出ていない（5/22の計測）。
- CX400上において、富士通製コンパイラとインテル社製コンパイラで性能差がある。インテルコンパイラを用いた場合の22.5%に対して、富士通コンパイラを用いた場合では14.4%しか出ていない。
- FX10では、1440ノードまでの計測には成功したが、2400ノード以上は実行できなかった。本プロジェクトでは、以上の問題点を克服し、4800ノードの計測を行うことを目的とする。

表1：ブラソフコードの単コア性能。

格子点数 40x20x30x30x30/core の弱いスケールリングによる計測結果。時間ステップ数は5。

システム	経過時間	実効性能	実効効率
FX10（東大）5/22	227.7sec	1.71GFlops	11.6%
FX10（東大）10/25（本プロジェクト）	200.0sec	2.54GFlops	17.2%
FX1（名大）	290.0sec	1.50GFlops	15.0%
HX600（名大）	250.1sec	1.74GFlops	17.4%
RX200（九大）	167.1sec	2.60GFlops	22.2%
CX400（九大）富士通コンパイラ	153.2sec	3.11GFlops	14.4%
CX400（九大）インテルコンパイラ	97.7sec	4.87GFlops	22.5%

2. 計算手法の概要

無衝突プラズマの振る舞いは、以下のブラソフ（無衝突ボルツマン）方程式によって記述される。

$$\frac{\partial f_s}{\partial t} + \vec{v} \cdot \frac{\partial f_s}{\partial \vec{r}} + \frac{q_s}{m_s} (\vec{E} + \vec{v} \times \vec{B}) \cdot \frac{\partial f_s}{\partial \vec{v}} = 0 \quad (1)$$

ここで \vec{E} , \vec{B} , \vec{r} 及び \vec{v} はそれぞれ電場, 磁場, 位置, 速度を表す。また, $f_s(\vec{r}, \vec{v}, t)$ は位置-速度位相空間におけるプラズマ粒子の分布関数であり, s はイオンや電子など種類を示す。 q_s と m_s はそれぞれ電荷と質量を表す。

プラズマ粒子の分布関数は、電磁場によって変形する。電磁場の時空間発展は以下のマックスウェル方程式によって記述される。

$$\nabla \times \vec{B} = \mu_0 \vec{J} + \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t} \quad (2.1)$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (2.2)$$

$$\nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0} \quad (2.3)$$

$$\nabla \cdot \vec{B} = 0 \quad (2.4)$$

ここで \vec{J} は電流密度, ρ は電荷密度, μ_0 は真空中の透磁率, ϵ_0 は真空中の誘電率, c は光速を示す。ブラソフ方程式(1)を速度空間で積分すると、以下の電荷保存則が得られる。

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \vec{J} = 0 \quad (3)$$

電磁場を変化させるための電流密度 \vec{J} はプラズマの運動によって生じるが、これはブラソフ方程式(1)の第二項にある実空間の流束 $\vec{v} f_s$ を速度空間で積分することによって求まり、電流密度 \vec{J} が電荷保存則(3)を満足する限り、ポアソン方程式(2.3)は自動的に満たされる。以上の方程式は、ブラソフコードにおいて解いているプラズマ粒子の運動論方程式であり、無衝突プラズマの第一原理と呼ぶ。

ブラソフ方程式は4次元以上の「超多次元」を扱う方程式であり、そのままの形で多次元数値積分を行うのは非常に困難であるため、演算子分離（operator splitting）法が古くから用いられてきた[e.g., Cheng & Knorr, 1976]。過去の研究では、各次元それぞれを1次元移流方程式に分解する方法が採用されていたが、本研究では、以下のように実空間移流、速度空間移流、速度空間回転の3つの物理的な演算子に分離する手法を開発した[Umeda et al., 2009]。

$$\frac{\partial f_s}{\partial t} + \vec{v} \cdot \frac{\partial f_s}{\partial \vec{r}} = 0 \quad (4.1)$$

$$\frac{\partial f_s}{\partial t} + \frac{q_s}{m_s} \vec{E} \cdot \frac{\partial f_s}{\partial \vec{v}} = 0 \quad (4.2)$$

$$\frac{\partial f_s}{\partial t} + \frac{q_s}{m_s} (\vec{v} \times \vec{B}) \cdot \frac{\partial f_s}{\partial \vec{v}} = 0 \quad (4.3)$$

この演算子分離は、PIC法においてニュートン-ローレンツ式（荷電粒子の運動方程式）を時間2次精度で解く手法として広く用いられている Boris アルゴリズム[Boris, 1970]に基づいている。

式(4.1)及び(4.2)は多次元の移流方程式であり、 \vec{v} 及び \vec{E} がそれぞれ \vec{r} 及び \vec{v} に依存しない線形方程式である。ブラソフ方程式の数値積分では、古くからセミラグランジュ法が採用されて

きた。即ち、格子点上に与えられている分布関数の値を更新する際に、次の時刻にその格子点上に来るべき分布関数の値の座標をバックトレースし、隣接する格子点より値を内挿する。2000年代前半までは、スプライン補間やCIP (Constraint Interpolation Profile) 法などの高次補間法を用いたセミラグランジュ法が用いられた、実空間 1 次元、速度空間 1 次元の、2 次元位相空間シミュレーションが主流であった。本研究をはじめとする近年の研究では保存型セミラグランジュ法を採用し、超多次元シミュレーションに成功している。これは、保存則であるブラソフ方程式において、プラズマの数密度の保存を数値的に保証し、数値誤差を軽減しようとするものである。PIC シミュレーションでは、電荷保存則(3)を厳密に満たすことにより数値誤差の軽減を行っており、ブラソフシミュレーションでは保存型解法を用いることにより、式(3)が自動的に満たされるため、保存型解法を採用するメリットは大きい。また式(4.1)及び(4.2)の多次元の線形移流方程式を解く上で、計算精度を上げることは非常に重要であるが、2 次精度以上の手法を用いた際に現れる数値振動を抑制するためのリミッタを保存型解法に導入するのは比較的容易である。

本研究では、演算子分離による数値拡散を抑制するために、多次元の線形移流方程式に対する演算子非分離 (unsplitting) 法を新たに開発している[Umeda et al., 2009]。また本研究では、無振動性及び正值性を保証するリミッタを新たに開発し、数値振動の抑制を行っている[Umeda, 2008; Umeda et al., 2012b]。ここで無振動スキームとは、ある区間において新たな極値 (極大, 極小) を生じず、既に存在する極地を (できるだけ) 減衰させないスキームである。本研究で行ったベンチマークテストでは、開発途上の 5 次多項式を用いた補間をテスト的に採用している。

式(4.3)は荷電粒子の速度が磁力線により運動エネルギーを保ったまま変化する回転方程式を表す。直交座標系における回転方程式は剛体回転問題と等価であり、線形移流問題と同様に、数値計算において最も基本的であるが、計算精度が重要となる問題である。本研究で採用している back-substitution 法[Schmitz & Grauer, 2006]では、Boris アルゴリズム[Boris, 1970]に基づいて速度空間での粒子の軌道をバックトレースし、 v_x , v_y , v_z 方向それぞれの演算子を分離して回転運動を解いている。剛体回転問題では、系の外側 (速度空間では速度が速い成分) になればなるほど移動量が大きくなり、クーラン条件の影響を受けやすくなる点に注意が必要であり、今後、陰解法や演算子非分離法の開発が必要である。

以上のように、ブラソフ方程式の数値解法は未だ発展途上である。この大きな原因は、ブラソフコードで扱う次元が多いためであり、開発やデバッグのために大容量の共有メモリ環境が必要となるからである。

一方、マックスウェル方程式(2.1)及び(2.2)は、FDTD (Finite Difference Time Domain) 法と呼ばれる電磁場解析法を用いて解く。FDTD 法では、Yee 格子[Yee, 1966]と呼ばれる staggered 格子を用いており、式(2.4)が自動的に満たされるように物理量が配置されている。また leap-frog アルゴリズムに基づいて電場と磁場の時刻も半タイムステップずらしており、空間精度と時間精度は共に 2 次である。

ブラソフシミュレーションでは非常に多くのメモリを必要とするため、並列計算が必須となる。ブラソフコードで使用する物理量は全て格子点上で与えられており、並列化においては領域分割法が有効である。図 2 は実空間 2 次元及び速度空間 3 次元を使用するブラソフコードにおける並列化の概念を示す。我々の目は 4 次元以上の空間を認識できないが、2 次元実空間の

各格子に 3 次元速度空間（速度分布関数）が定義されていると考えると分かりやすい。本研究では図 1 のように実空間（x-y 平面）においてのみ領域分割を行い、速度空間の領域分割は行わない[Umeda et al., 2012a]。これは、電荷密度や電流密度などのモーメント量を計算する際に必要な速度空間の積分において、各実空間でのリダクション処理を行わないようにするためである。またオプションとして、プロセスエレメント（PE）毎に OpenMP によるスレッド並列化も許可している。

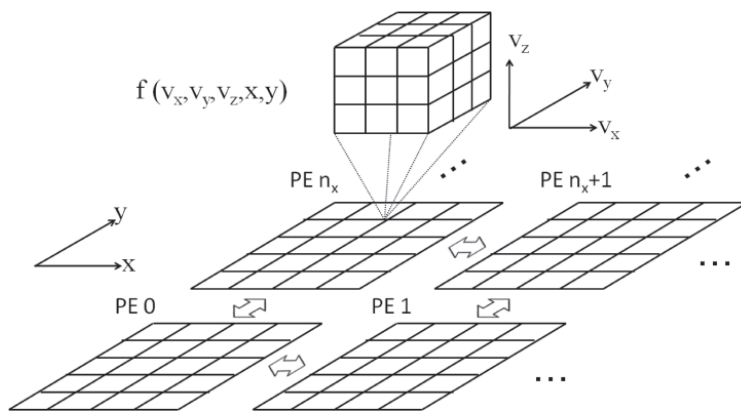


図 1：5 次元ブラソフコードにおける空間領域分割[Umeda et al., 2012a]。

3. 性能評価

3.1 システム間及びコンパイラ間の性能差について

本プロジェクトで用いるブラソフコードでは、様々な組み込み関数を多用しているため、まず、FX10, FX1 及び CX400 上において、表 2 に示す組み込み関数の性能評価を行った。図 2 に示すような簡単なプログラムを使用し、1 秒あたりの呼び出し回数 (Mcall/sec) を測定した結果を表 2 に示す。ここで、FX10, FX1 及び CX400 上での富士通社製コンパイラでは

```
firt -Kfast
```

を、CX400 上でのインテル社製コンパイラでは

```
ifort -ip -ipo -O3 -xAVX
```

を指定した。

まず、FX1/FX10 上では CX400 とのクロック周波数比 (FX10 : 1.848GHz, CX400 : 2.7GHz) に対して、整数に関連した関数の性能が悪くなるのが分かった。特に、floor 関数の性能が極端に悪く、FX10 に至っては FX1 よりも性能が悪いことが分かる。また CX400 においても、floor 関数の性能において、富士通コンパイラとインテルコンパイラで差が出た。これに関しては、floor 関数強化オプション (-W0, -zmpa=Ailfunc-for-floor:非公式。なお、次回のコンパイラバージョンアップで、オプションなしでも floor 関数が強化される予定) を指定することによってある程度改善することが分かったが、FX10/FX1 上では、アドレス計算に出来るだけ整数の組み込み関数を用いないようにした方が、性能が向上することが分かった。本プロジェクトで用いるブラソフコードでは、整数でアドレス計算している部分を全て、型変換を含む倍精度で計算するように変更し、性能が向上した (表 1 参照)。

```

real(kind=8) :: din(nx)
integer(kind=4) :: nout(nx)
real(kind=8) :: t1, t2
integer(kind=8) :: ii

call cpu_time(t1)
do ii = 1, nx
    nout(ii) = floor(din(ii))
end do
call cpu_time(t2)
print*, dble(nx/1000000)/(t2-t1), 'M calls/sec'

```

図 2：関数性能測定プログラムの例。

ここでは、floor 関数（引数は倍精度実数、戻り値は単精度整数）の場合を示している。

表 2：各システムにおける、組み込み関数呼び出しの単コア性能（単位はメガ回/秒）。

整数関数は単精度、実数関数は倍精度を使用した。

	FX10	FX1	CX400 (Fujitsu)	CX400 (Intel)
RAND	15.8M/s	19.6M/s	61.8M/s	97.7M/s
AINT	384.6M/s	286.8M/s	721.3M/s	724.7M/s
ANINT	714.3M/s	224.6M/s	721.3M/s	714.4M/s
ABS	833.3M/s	427.9M/s	728.3M/s	724.7M/s
SIGN	833.3M/s	435.6M/s	728.3M/s	724.8M/s
FLOOR	42.0M/s ⇒ 416.7M/s	92.9M/s	824.3M/s	1041.8M/s
NINT	416.7M/s	139.1M/s	903.8M/s	926.1M/s
INT	416.7M/s	318.9M/s	1000.2M/s	1020.6M/s

3.2 前回に全ノード計測を失敗した原因

5月22日に実施した大規模 HPC チャレンジでは、1440 ノード (23040 コア) までの計測には成功したが、2400 ノード (38400 コア) 以上は実行できず、また原因の特定には至らなかった。今回 (10月25日) は、2400 ノードジョブの実行データを分析し、初期状態で既に値が NaN になっている部分があることを突き止めた。

本プロジェクトで用いるブラソフコードでは、初期状態でハリス電流層と呼ばれる磁化プラズマの平衡解[Harris, 1962]を設定しており、その中で用いている $1/\cosh^2(x)$ において、NaN が検出されていることを突き止めた。即ち、 $1.0E+154$ 以上の値の 2 乗で Inf が戻り、その逆数として NaN が戻る。これは FX10 上の富士通コンパイラ特有の現象だと思われるが (CX400 では起こらないことは確認済)、計算領域が大きくなり引数が大きくなったことで生じた、大規模計算によって初めて分かった問題である。

3.3 全ノード計測結果

問題点の修正を行い、コードの性能測定を行った。4次元以上の超多次元問題のシミュレーションには非常に多くのメモリ容量を必要とするため、コアあたりあるいはノードあたりのメモリ容量を適切に設定する必要がある。現存する計算機上で実際にシミュレーションを実行する際には、 30^3 — 60^3 程度の速度空間を使用している。これは、コアあたりのメモリ容量として1GB—4GBを想定しており、速度空間の格子数と実空間の解像度を固定したまま、実空間の格子数(計算領域)を増やす、弱いスケールリングにより実際の計算を行っていることを意味する。本プロジェクトでは、コアあたりの格子数を実空間では 40×20 、速度空間では $30 \times 30 \times 30$ に設定する。これはメモリ容量で約1GBに相当し、実際の計算で使用している格子数とほぼ同じである。

本プロジェクトで用いるブラソフコードではMPIとOpenMP/自動並列のハイブリッド並列を採用しているため、まず、プロセス並列とスレッド並列のどの組み合わせが最適かを1ノード(16コア)で測定した。表3に示すとおり、2スレッドまたは4スレッドで実行した場合が最も速く、またフラットMPIの場合よりも高速であることが分かる。さらに表1との比較により、ハイブリッド並列の実効性能が単コアの実効効率を超えていることが分かった。以上の結果およびプロセス数の軽減より、以降は4スレッドのハイブリッド並列により性能計測を行った。

1コアあたり1GBにメモリ使用量を固定したときの、5次元ブラソフコードの弱いスケールリング性能を図3及び図4に示す。また参考値として、HA8000及びFX1での計測結果も示す。図3はコア数に対する実効速度を表し、図4はコア数に対する実効並列化率(Nコアの実効効率を1コアの実効効率で割った量)を表す。まず、どのシステムにおいても(ただし、東京大学HA8000を除く)、性能がほぼスケールしていることが分かる。HA8000上では、4096コアと8192コアで実効並列化効率がほぼ同じであり、これはノード間通信のトラフィックが飽和していると考えられる。FX1上では実効並列化率が90%以上であったが、全ノードを用いて計測した場合に極端に性能が劣化する現象が見られた(名大の3072コア:図3のFX1の実線データ、JAXAの12032コア:図3のFX1の点線データ)。一方FX10上では、256コアまでは100%を超えており(前述の通り、4スレッド化により単コアよりも性能向上が見られているため)、1000コア以上でも97%以上の実効並列化率でスケールしており、1PFlops以上でも十分にスケールすることが予想できる。また図4に見られるように並列性能に多少のブレがあるものの、全ノードを用いた場合でもFX1で見られたような性能劣化は全く起こらなかった。最終的な結果としては、76,800コア(4800ノード)を使用した際の実行性能及び実効効率はそれぞれ、192GFlopsと16.9%であり、目標としていた効率を上回った。

表3: ブラソフコードのFX10単ノード性能。

格子点数 $80 \times 40 \times 30 \times 30 \times 30$ の計測結果。時間ステップ数は5。

スレッド数	1	2	4	8	16
プロセス数	16	8	4	2	1
経過時間	199.8 sec	197.3 sec	197.3 sec	203.6 sec	205.9 sec

4. まとめ

H24年度第2回大規模 HPC チャレンジにおいて、宇宙プラズマの第一原理シミュレーション手法であるブラソフコードの5次元バージョンの性能評価を行った。30,000以上の並列計算を行えるシステムは日本では少なく、また大規模計算によって初めて露わになるコードのバグや、システム（コンパイラ）の障害もあるため、このような全ノード計測は HPC 分野にとって極めて重要であると言える。

本プロジェクトの成果として、まず、FX10は通信が非常に速く、京コンピュータクラスの並列数でも十分に性能がスケールする可能性があることが分かった。次に、FX10（SPARC64チップ）の特徴として、整数演算がインテル社製CPUに対して遅いことが挙げられ、整数演算を出来るだけ行わないようにチューニングすることが、高い実効性能を得るコツの1つだと言える。また、並列数を上げたときに急に動かなくなる場合、関数の引数が大きくなったことによるオーバーフローも原因となり得るので、デバッグ時の参考にして頂ければ幸いである。

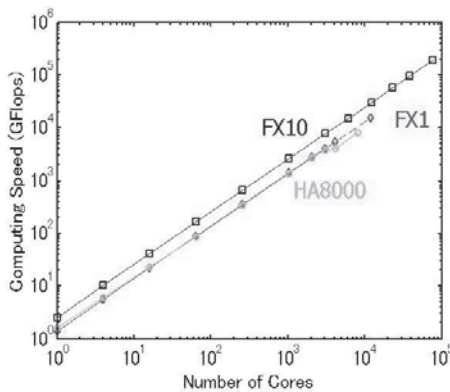


図3：1GB/core使用時の5次元ブラソフコードの弱いスケール性能(1)。

コア数に対する実効速度を示している。

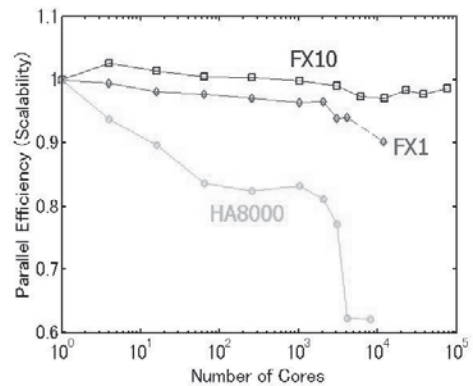


図4：1GB/core使用時の5次元ブラソフコードの弱いスケール性能(2)。

コア数に対する実効並列化率を示している。

参考文献

- Boris, J. P.: Relativistic plasma simulation-optimization of a hybrid code, Proc. Fourth Conf. Num. Sim. Plasmas, ed. by J. P. Boris and R. A. Shanny, pp.3—67, Naval Research Laboratory, Washington D. C. (Nov. 1970).
- Cheng, C. Z., Knorr, G.: The integration of the Vlasov equation in configuration space, J. Comput. Phys., Vol.22, No.3, 330—351 (1976).
- Harris, E. G., On a plasma sheath separating regions of oppositely directed magnetic fields, Nuovo Cimento, Vol.23, No.1, 115—121 (1962).
- Schmitz, H., R. Grauer, R.: Comparison of time splitting and backsubstitution methods for integrating Vlasov's equation with magnetic fields, Comput.Phys. Commun., Vol.175, No.2, 86—92 (2006).
- Umeda, T.: A conservative and non-oscillatory scheme for Vlasov code simulations, Earth Planets Space, Vol.60, No.7, 773—779 (2008).

- Umeda, T.: Effect of ion cyclotron motion on the structure of wakes: A Vlasov simulation, *Earth Planets Space*, Vol.64, No.2, 231—236 (2012).
- Umeda, T., Fukazawa, K., Nariyuki, Y., Ogino, T.: A scalable full electromagnetic Vlasov solver for cross-scale coupling in space plasma, *IEEE Trans. Plasma Sci.*, Vol.40, No.5, 1421—1428 (2012a).
- Umeda, T., Kimura, T., Togano, K., Fukazawa, K., Matsumoto, Y., Miyoshi, T., Terada, N., Nakamura, T. K. M., Ogino, T.: Vlasov simulation of the interaction between the solar wind and a dielectric body, *Phys. Plasmas*, Vol.18, No.1, 012908(7pp.) (2011).
- Umeda, T., Miwa, J., Matsumoto, Y., Nakamura, T. K. M., Togano, K., Fukazawa, K., Shinohara, I.: Full electromagnetic Vlasov code simulation of the Kelvin-Helmholtz instability, *Phys. Plasmas*, Vol.17, No.5, 052311(10pp.) (2010a).
- Umeda, T., Nariyuki, Y., Kariya, D.: A non-oscillatory and conservative semi-Lagrangian scheme with fourth-degree polynomial interpolation for solving the Vlasov equation, *Comput. Phys. Commun.*, Vol.183, No.5, 1094—1100 (2012b).
- Umeda, T., Togano, K., Ogino, T.: Two-dimensional full-electromagnetic Vlasov code with conservative scheme and its application to magnetic reconnection, *Comput. Phys. Commun.*, Vol.180, No.3, 365—374 (2009).
- Umeda, T., Togano, K., Ogino, T.: Structures of diffusion regions in collisionless magnetic reconnection, *Phys. Plasmas*, Vol.17, No.5, 052103(6pp.) (2010b).
- Yee, K. S., Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Trans. Antenn. Propagat.*, NOL.AP-14, No.3, 302—307 (1966).