Vol.2014-HPC-146 No.20
2014/10/3

OpenMP と MPI を用いたハイブリッド並列ブラソフコードの性能測定

梅田 隆行[†1]　　深沢 圭一郎[†2,†3]

ブラソフコードは宇宙空間を満たす無衝突プラズマの第一原理シミュレーション手法である．ブラソフシミュレーションでは，位置及び速度で与えられる超多次元位相空間における荷電粒子の分布関数の時間発展を，運動論方程式により直接解き進めている．4 次元以上の空間を扱うシミュレーションでは，ノードあたり，あるいはコアあたりに使用できるメモリ容量の制限から，数値解法や性能チューニングにおいて様々な工夫が必要である．本研究グループはこれまでに様々な HPC 関連プロジェクトと通じて，ブラソフコードの性能チューニングを行ってきた．本論文では特に，OpenMP の COLLAPSE ディレクティブを用いたハイブリッド並列ブラソフコードについて，京コンピュータ，Fujitsu FX10 及び Fujitsu CX400 において行った性能測定の結果について議論する．

Performance measurement of Vlasov code hybrid-parallelized with OpenMP and MPI

TAKAYUKI UMEDA[†1]　　KEIICHIRO FUKAZAWA[†2,†3]

Vlasov code is a first-principle simulation method for collisionless space plasma. The Vlasov code solves the time development of phase-space distribution functions of charged particles in hyper-dimensions based on fully kinetic equations. Since the distribution functions are defined in more than four dimensions, the Vlasov code requires high-resolution and high-performance numerical schemes which should work in limited computational memory per node or per core. Our Vlasov code has been made performance tuning on various scalar CPU architectures under Japanese HPC projects. In the present study, the COLLAPSE directive of OpenMP is used for hybrid-parallelization of our Vlasov code. The results of the performance measurement on the K-computer, Fujitsu FX10 and Fujitsu CX400 are discussed.

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．

我々が住む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，宇宙飛行士の被曝，人工衛星の故障や通信障害に繋がる地球磁気圏・電離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の打ち上げなど，日本においても宇宙利用が現実的になってきており，宇宙天気の予報・予測に繋がる宇宙プラズマ研究は極めて重要である．

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，粒子運動論を扱う方程式であるブラソフ（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において MHD 方程式で記述できる物理過程と粒子の運動論方程式によって記述できる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，ニュートンーローレンツ方程式により解き進める PIC（Particle-In-Cell）法である．格子点（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

†1 名古屋大学太陽地球環境研究所
Solar-Terrestrial Environment Laboratory, Nagoya University
†2 九州大学情報基盤研究開発センター
Research Institute for Information and Technology, Kyushu University
†3 京都大学学術情報メディアセンター
Academic Center for Computing and Media Studies, Kyoto University

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，プラズマを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の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．格子点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に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 次元シミュレーションがようやく実用の域に達しつつある段階である．

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．そのための準備として，現存する超並列計算機上のおける 5 次元ブラソフコードの性能評価及び性能チューニングを行っている．

これまでの京コンピュータを含む超並列計算機での大規模シミュレーションの経験より，京コンピュータや Fujitsu FX10 などの SPARC 系システムにおいては，MPI と OpenMP／自動並列を併用したハイブリッド並列が MPI のみのフラット MPI よりも演算性能が高く，x86 系システムではフラット MPI よりも演算性能が高いことが分かっている．また SPARC 系システム上でのハイブリッド並列においても，スレッド数が 2-4 の場合に演算性能が高くなり，ノードあたりのプロセス数を 1 にすると極端に性能が劣化する現象が見られた．一方で，プロセス数が増えると，全体通信の時間の増加や，出力ファイル数の増加などにより，プロセス数をできるだけ減らしたほうが利点は大きい．そこで本研究では，多重ループのスレッド化を行う OpenMP の COLLAPSE ディレクティブに着目し，ノード内のプロセス数(=ノード内のコア数/スレッド数)を変えたときの演算性能の測定を行った．

2. 計算手法の概要

2.1 基礎方程式

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．

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

ここで $\vec{E}$，$\vec{B}$，$\vec{r}$ と $\vec{v}$ 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．

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

$$\nabla\times\vec{E}=-\frac{\partial\vec{B}}{\partial t} \tag{2.2}$$

$$\nabla\cdot\vec{E}=\frac{\rho}{\varepsilon_0} \tag{2.3}$$

$$\nabla\cdot\vec{B}=0 \tag{2.4}$$

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．

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

マックスウェル方程式(2.1)に含まれる電流密度 $\vec{J}$ はプラズマの運動によって生じ，これにより電磁場が変化する．電流密度 $\vec{J}$ はブラソフ方程式(1)の第二項にあたる実空間の流束 $\vec{v}f_s$ 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．

2.2 数値解放の概要

ブラソフ方程式は 4 次元以上の「超次元」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[1]．過去の研究では，各次元(x, y, z, v_x, v_y, v_z)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用いている[2]．

$$\frac{\partial f_s}{\partial t}+\vec{v}\frac{\partial f_s}{\partial \vec{r}}=0 \tag{4.1}$$

$$\frac{\partial f_s}{\partial t}+\frac{q_s}{m_s}\vec{E}\frac{\partial f_s}{\partial \vec{v}}=0 \tag{4.2}$$

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

この演算子分離は，PIC 法においてニュートンーローレンツ式（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[3]に基づいている．

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（unspliting）法を新たに開発している[2]. また本研究では，無振動性及び正値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[4][5]. 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，既に存在する極値は（できるだけ）減衰させないスキームであり，ENO/WENO 法はこれに該当するが，TVD 法は極地を鈍らせるために該当しない．

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[6]では，Boris アルゴリズム[3]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，即ち速度空間において速度が速くなればなるほど移動量（加速）は大きくなり，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，開発やデバッグに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[7]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を半タイムステップずらしており，時空間精度は 2 次である．

3. ハイブリッド並列

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 1 は実空間 2 次元及び速度空間 3 次元を使用する 5 次元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上に 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[4]．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．

5 次元ブラソフコードでは，OpenMP によるスレッド並列も併用している．スレッド並列はそのオーバーヘッドの大きさから，できるだけより外側のループで行うのが効率的である．x86 系 CPU においてはこれまで，スレッド並列を用いないフラット MPI 並列のほうが，スレッド並列と MPI プロセス並列を併用したハイブリッド並列よりも効率的であった．しかし，FX10 や京コンピュータなどの近年の計算機においては，ハイブリッド並列のほうが効率的になる場合がある．また，本研究グループにおける京コンピュータ 6144 ノードの実利用経験より，IO 処理や分散ファイルのデータ解析などの観点からプロセス数をできるだけ減らしたほうが利点は大きい．

ブラソフモデルは 4 次元以上の超次元を扱い，メモリ使用量が非常に多いため，速度空間の格子点を 30^3－60^3 に固定してコアあたりのメモリ使用量 1-4GB に設定しつつ，使用ノード数を増やして計算領域（実空間の格子数）を拡張していくのが実際の超並列計算機の利用方法である．しかし，近年の計算機においては，ノード内の共有メモリの容量は増えずにコア数のみが増加していく傾向にあるため，単一のループのみをスレッド化する単純な方法には限界がある．

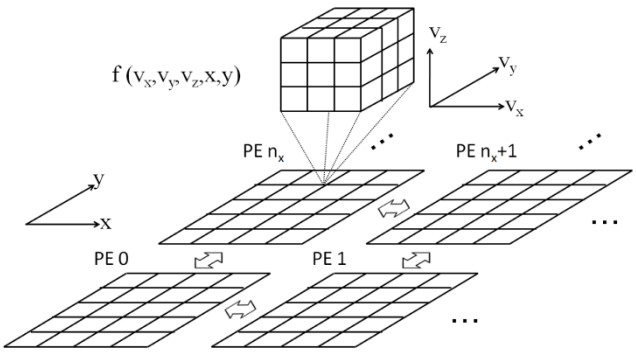

図 1　5 次元ブラソフコードにおける空間領域分割[8].
Figure 1　The domain decomposition in the configuration space for the five-dimensional Vlasov code [8].

以下に，本研究課題において試した OpenMP スレッド並列化について解説する．5 次元ブラソフコードの計算エンジンは 5 重ループになっており，外側 2 ループ(i 及び j)は実空間(x, y)格子のインデックスを示す．また内側 3 ループ(l, m 及び n)は速度空間(vx, vy, vz)格子のインデックスを示す. OpenMP ディレクティブは最外側ループに挿入する（プログラム 1 参照）．この場合，最外側のループ(j)のみがスレッド並列化される．一方で，!$OMP DO ディレクティブの COLLAPSE オプションは，多重ループのスレッド化を行う機能であり，プログラム 2 の場合は j 及び i のループがスレッド並列化される．

なお, i, j のループと l, m, n のループでは演算やデータの並びが異なるため，COLLAPSE オプションは 2 を指定している．また，流体型の 3 次元コードでは i, j, k のループに対して COLLAPSE オプションを 3 に指定することも可能ではあるが，最内側ループはできるだけループ長を長く取ったほうが効率的であるため，COLLAPSE オプションは 2 のほうがよい．

4. 性能測定

4 次元以上の超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，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，速度空間の格子数は 30^3—60^3 程度に設定する必要がある．これは，コアあたりのメモリ容量を 1GB—4GB に想定している．本研究では，コアあたりの格子数を実空間では 40×20，速度空間では 30×30×30 に設定し，これはメモリ容量で約 1GB である．実際の計算でも，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，弱いスケーリングを採用している．

```
!$OMP PARALLEL DO
DO j = 1,ny
  DO i = 1,nx
    DO n = 1,nvz
      DO m = 1,nvy
        DO l = 1,nvx
           演算
        END DO
      END DO
    END DO
  END DO
END DO
!$OMP END DO
```

図 2　As-Is コード.
Figure 2　The as-is code.

単一ノードにおいて，プロセス数とスレッド数を変化させた時の各システム／コンパイラの性能を表 1 に示す. CX400 及び FX10 はノードあたり 16 コアであるため，表 1 の測定での実空間の格子数は 160×80 である．なお CX400 ではインテルコンパイラを用いたほうが富士通コンパイラを用いた時よりも約 1.5 倍高速であることから，表 1 の測定ではインテルコンパイラのみで行っており，プログラム 1 と 2（COLLAPSE オプション無・有）の比較を行っている．

まず，FX10 と CX400 において，2 倍近くの性能差がある．これは，クロック周波数及び実効効率においてそれぞれ約 1.4 倍の差があるためである．次に，FX10 では 4 スレッド･4 プロセスのハイブリッド並列が最速であり, CX400 では COLLAPSE オプション無の場合はフラット MPI が最速である．しかし COLLAPSE オプション有の場合は 4 スレッド・4 プロセスのハイブリッド並列が最速となった．特筆すべき点としては，COLLAPSE オプション有の場合にスレッド数を増やしていっても性能劣化が小さいことが挙げられる．これは，プロセス数を減らすという観点において重要である．一方で，スレッド数 1（フラット MPI）及びスレッド数 2 の場合において，ループの分割の観点からは全く同じにもかかわらず, FX10 と CX400 の両方において，COLLAPSE オプション無・有で性能に差が出た．

次に，CX400 において全ノード計測を行った結果を表 2 に示す．富士通コンパイラとイオンテルコンパイラの両方を使用して, COLLAPSE 無の As-Is コードでフラット MPI(1 スレッド・16 プロセス／ノード)の場合と，COLLAPSE 有のコードでノードあたり 4 スレッド・4 プロセスの場合を比較した. MPI_Init/Finalize の経過時間は，ジョブスクリプトの#PJM -m s オプションで送信されるジョブ統計情報(ELAPSE TIME (USE))から逆算した. 計測結果より，

```
!$OMP PARALLEL DO COLLAPSE(2)
DO j = 1,ny
  DO i = 1,nx
    DO n = 1,nvz
      DO m = 1,nvy
        DO l = 1,nvx
           演算
        END DO
      END DO
    END DO
  END DO
END DO
!$OMP END DO
```

図 3　COLLAPSE ディレクティブを用いたコード.
Figure 3　The program with COLLAPSE directive.

表 1　ブラソフコードの単一ノード性能と COLLAPSE ディレクティブ有・無の比較
Table 1　Comparison of the performance of Vlasov code on a single node with/without COLLAPSE directive.

スレッド数／プロセス数	FX10 As-Is ／ COLLAPSE	CX400 As-Is ／ COLLAPSE
1／16	206.64 sec ／ 206.59 sec	107.03 sec ／ 107.82 sec
2／8	199.68 sec ／ 200.16 sec	113.16 sec ／ 107.48 sec
4／4	199.47 sec ／ 199.53 sec	107.75 sec ／ 106.07 sec
8／2	201.08 sec ／ 200.77 sec	113.16 sec ／ 109.02 sec
16／1	201.90 sec ／ 200.83 sec	113.39 sec ／ 108.89 sec

表 2　CX400 におけるブラソフコードの全ノード性能
Table 2　Performance of Vlasov code on the entire nodes.

各セクションの 経過時間	1 スレッド per ノード(As-Is) Fujitsu ／ Intel	4 スレッド per ノード(COLLAPSE) Fujitsu ／ Intel
MPI_Init / Finalize	17.04 sec ／ 1153.53 sec	15.04 sec／ 402.45 sec
初期化(t2-t1)	135.33 sec ／ 706.59 sec	13.63 sec ／ 21.42 sec
メインループ(t3-t2)	166.35 sec ／ 124.88 sec	162.33 sec ／ 120.13 sec

インテルコンパイラを用いた場合に主に `MPI_Init` に大量の時間を費やしていることが判明した．この現象は富士通コンパイラを用いた場合はほとんど発生せず，富士通のジョブスケジューリングシステムとインテルコンパイラ環境の相性の悪さが原因であると考えられる．また，As-Is コードを用いてフラット MPI で計測した場合，初期化ルーチン，特に電磁場を平衡解に収束させるために用いている `MPI_Allreduce` において時間がかかっており，特にインテルコンパイラを用いた場合に顕著であった．重要な結果として，`MPI_Init` や `MPI_Allreduce` の経過時間はハイブリッド並列化によりプロセス数を減らすことによって大幅に削減できることが挙げられる．またメインループの自体も，プロセス数を減らすことによって計算効率の向上が若干見られた．

5. おわりに

ブラソフコードは，宇宙空間に広く存在する無衝突プラズマの第一原理シミュレーション手法である．プラズマは位置－速度位相空間における分布関数として定義され，超多次元関数として与えられる．ブラソフシミュレーションは計算負荷が非常に高く，その手法の開発やデバッグが困難であるため，計算手法は未だ発展途上にある．本研究では，2 次元実空間及び 3 次元速度空間を扱う 5 次元ブラソフコードについて，ハイブリッド並列を採用した場合の性能評価を行った．OpenMP の COLLAPSE ディレクティブをもちいることにより，フラット MPI と同等以上の性能が得られることが分かり，これはプロセス数の削減に有効な手段の１つと言える．

謝辞　本研究は，科学研究費補助金・挑戦的萌芽研究 No.25610144 によりサポートを受けている．ベンチマークテストに使用したスーパーコンピュータシステムの計算リソースは，九州大学先端的計算科学研究プロジェクト及び HPCI システム利用研究(hp120092, hp140064, hp140081)により提供された．また性能チューニングに際し，サイエンティフィック・システムズ研究会マルチコア性能 WG，富士通及び理研 AICS の似鳥啓吾氏に助言を頂いた．

参考文献

1. Cheng, C. Z., Knorr, G. : The integration of the Vlasov equation in configuration space, *J. Comput. Phys.*, Vol.22, No.3, 330—351 (1976).
2. 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).
3. 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).
4. Umeda, T.: A conservative and non-oscillatory scheme for Vlasov code simulations, *Earth Planets Space*, Vol.60, No.7, 773—779 (2008).
5. 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 (2012).
6. 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).
7. 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).
8. 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 (2012).