

GPGPU による 2 次元超粒子磁化プラズマ・シミュレーション Two-dimensional Magnetized Super-Particle in Cell Plasma Simulation by GPGPU Method

○向中野徹¹, 長峰康雄², 相澤正満²*Toru Mukainakano¹, Yasuo Nagamine², Masamitsu Aizawa²

We have studied the two-dimensional magnetized self-consistent super particle in cell (PIC) simulation. To reduce the execution time for these simulations, we have applied the so-called GPGPU method. We describe GPGPU method briefly, and report its effectiveness to the PIC simulation.

1. 目的

プラズマの粒子シミュレーションでは、計算量が膨大になる上に、相互作用のある系をセルフコンシステントに扱わなくてはならない。従来の PC では困難であった大規模プラズマ粒子シミュレーションを、GPGPU 技術により並列化し実行する。その際、実行効率について検証する。

2. 概要^[1]

GPGPU(General - Purpose computing on Graphics Processing Units)とは、画像処理を専門とする補助演算装置である GPU を、汎用的な数値計算に用いる技術のことである。現在の GPU が 3D 描画をするためには、一度の命令で多数のデータを処理しなくてはならない。そのため、GPU は多数の計算コアを積んだ並列計算機となっている。その並列性が注目され、GPU を汎用演算に使用する GPGPU 技術が研究されている。

GPU の処理は、CPU 用のメモリから GPU 用のメモリにデータをコピーして、CPU が GPU へ処理を指示し、処理が終わると CPU 用のメモリへデータを返すという流れである。(fig.1)

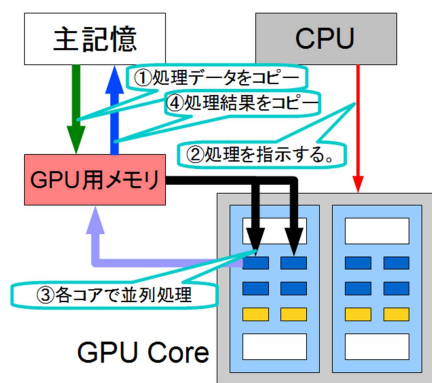


fig.1 GPU の処理の流れ

3. 超粒子磁化プラズマ・シミュレーション

実用的な数値計算の例として、プラズマシミュレーションを行うにあたり、そのベースモデルとして ES1 (Electro Static 1-dimensional)^{[2][4]} を 2次元に拡張したモデルを用いた。ES1 とは 1次元の静電場における非線形な多粒子系のモデルである。2次元に拡張するにあたり、ES1 では考慮していなかった磁場を組み込み、外部磁場として与えた。このモデルを、以下 ES2(Electro Static 2-dimensional)とする。

ES2 の特徴として、有限サイズの粒子 (超粒子) は 2次元空間を移動し、電場や電荷密度などの物理量は空間の格子点上で値を与える。セルフコンシステントなモデルなので、粒子が移動すると周囲の電場が変化する。そのため、各時間ステップごとに多くの計算が必要となる。電場の計算と粒子の移動計算を並列化することにより、計算時間の短縮をめざす。

シミュレーションの主な流れは、システム内に初速度を持った荷電粒子を配置した後、fig.2 に示すようなサイクルに入る。このサイクルは、荷電粒子の位置をもとに電荷密度を求めて電場を計算し、この新しい電場をもとに粒子を微小距離移動させて、新しい位置に更新するというものである。

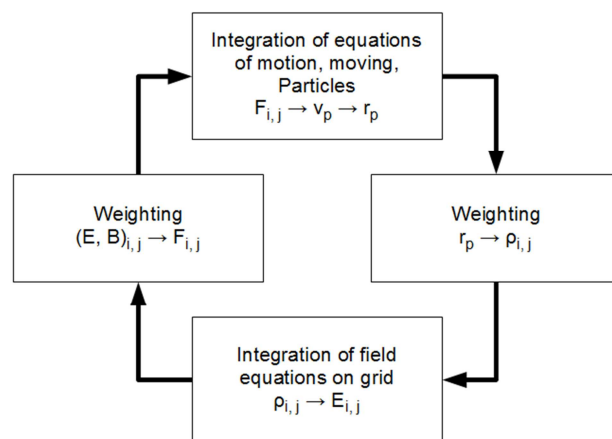


fig.2 ES2 での主要な計算サイクル

1 : 日本大学・院・量子 2 : 日本大学・教員・量子

各サイクルのうち最も計算量が多いのが粒子の加速・移動であり、以下の差分運動方程式を、ラーモア運動を正確に再現する Boris の方法^[3]により解いた。

$$\frac{v^{n+1/2} - v^{n-1/2}}{\Delta t} = \frac{e}{m_e} \left(\mathbf{E}^n + \frac{v^{n+1/2} + v^{n-1/2}}{2} \times \mathbf{B}^n \right) \quad (1)$$

$$\frac{r^{n+1} - r^n}{\Delta t} = v^{n+1/2} \quad (2)$$

ただし、 r, v は粒子の位置、速度、 \mathbf{E}, \mathbf{B} は粒子の位置での電場、磁場を表し、 n は現在のステップ数(時間)を表す。CPU による計算時間と、GPU による計算時間を比較して、セルフコンシステントなシステムで GPGPU が有効かどうかを検証した。

4. 実行結果

粒子の数は $N = 262144 (= 2^{18})$ 、空間格子点数は $M_x = M_y = 256$ 、空間の大きさは $S_x = S_y = 256.0$ (つまり $\Delta \tilde{x} = \Delta \tilde{y} = 1.0$) 空間は x - y 方向ともに周期境界条件を適用した。時間刻み幅は $\Delta \tilde{t} = 0.03125$ (“ \sim ”は規格化により無次元化された量を表す) とした。初期条件は粒子の位置を $\tilde{y} = S_y/2 = 128.0$ とし、 x 方向はランダムに並べた。速度は y 成分は 0 とし、 $\tilde{v}_{x0} = \pm 30.0$ のどちらかになるような所謂 2 ビーム不安定性条件にした。磁場は z 方向を向き一定一様とし、初期速度でのラーモア半径 $\tilde{r}_L = 3.0$ とした。GPGPU の計算では、CUDA のブロック数を 12、ブロックあたりのスレッド数を 128 個の、合計スレッド数 16384 個で計算した。

ES2 の計算結果を fig.3 に示す。グラフは横軸が x 座標、縦軸が y 座標であり、赤点が $\tilde{v}_{x0} = +30.0$ 、緑点が $\tilde{v}_{x0} = -30.0$ の、実空間での粒子の位置を表している。

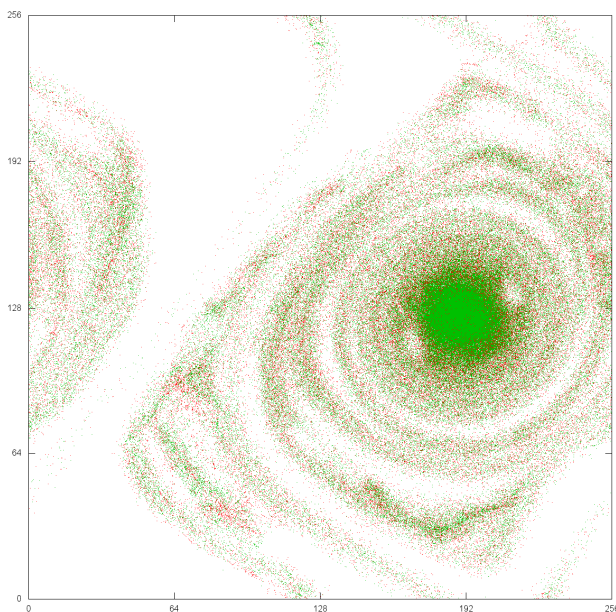


fig.3 ES2 の計算結果

($\tilde{t} = 128(4096step)$ 、横軸 : x 、縦軸 : y)

ステップごとに粒子の動きを追うと、多数の小さな渦を作りながら合体によって渦が成長していき、最終的には fig.3 が示す通り一つの大きな渦となった。

次に CPU と GPU の全体の計算時間のグラフを fig.4 に示す。計算する時間を $\tilde{t} = 128(4096 step)$ に固定し、粒子数を $N = 2^{10} \sim 2^{20}$ で 2 のべき乗で変化させた。また、空間格子点数を $M_x = M_y = 16$ に変更した。 N が十分に大きい ($N > 2^{13}$) 場合、CPU と GPU の両方で N と計算時間が比例関係にあり、増加量は CPU の方が大きい。 N が小さい ($N \leq 2^{13}$) 場合は GPU での計算時間が N にかかわらず一定になっているが、これは GPU のスレッド数と関係している。

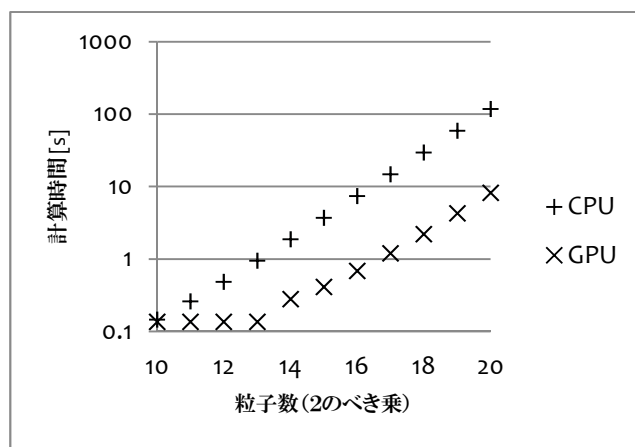


fig.4 CPU と GPU の計算時間(ES2)

5. 結論・課題

従来困難であった相互作用のあるプラズマシミュレーションを、GPGPU により並列化した。CPU のシングルプロセスプログラムと比べて、10 倍以上の高速化が実現できた。また時間が経つにつれて粒子が渦を巻く特徴的な動きを確認することができた。

今回のシミュレーションモデルでは磁場を一定値としたが、本来磁場は粒子の影響で時間変化する。より精度を高めるため、粒子自身が磁場に影響を与えるモデルを並列化することを今後の課題とする。

参考文献

- [1] 青木尊之・額田彰：「はじめての CUDA プログラミング(工学社)」, 2009 年
- [2] 石黒静児：「講座 プラズマ計算機シミュレーション入門 II 3. 静電粒子シミュレーション」, プラズマ・核融合学会誌, Vol.74, No.6, pp.591-597, 1998 年
- [3] C.K.Birdsall・A.B.Langdon：「Plasma Physics via Computer Simulation」, 1991 年
- [4] 内藤孝雄 2010 年度量子理工学専攻修士論文