2011年度 卒業論文

球ジオメトリにおける マルチグリッド法

神戸大学工学部情報知能工学科 西田 泰大

指導教員 政田 洋平 助教 陰山 聡 教授

2012年2月22日

球ジオメトリにおけるマルチグリッド法

西田 泰大

要旨

過去数十年にわたる太陽活動の観測によって、その活動性の源が磁場であるということがわかってきた。しかし、磁場の増幅機構、すなわち太陽ダイナモについては多くの謎が残されたままである。本研究の目的は、低マッハ数近似とインヤン格子を用いた新しいダイナモシミュレーションコードを開発することである。開発したコードを太陽に応用し、スーパーコンピュータを使って太陽内部のプラズマの運動を数値的に解くことで、太陽ダイナモ機構を解明することが本研究の最終目標である。

開発を進めるシミュレーションコードでは、低マッハ数近似磁気流体方程式を 高速かつ高精度で解く必要がある。コードの高速化の最大の障害となるのが、方 程式の中に含まれる速度場のポアソン方程式である。今回我々は、このポアソン 方程式を高速に解くためのマルチグリッドソルバーを開発し、インヤン格子への 適用実験を行った。

マルチグリッド法とは、微分方程式の陰的反復計算をグリッド間隔の異なる複数階層で行うことで、様々な波長を持つ誤差を効率的に減少させる手法のことである。この方法を用いることで、ポアソン方程式を数値的に解く際の行列計算にかかる時間を、劇的に減少させることができる。

開発したマルチグリッドソルバーの性能を調べるために、星の表面から出て無限遠でゼロに漸近する磁場を記述するポアソン方程式を、インヤン格子上で数値的に解くことを試みた。本研究の結果、マルチグリッド法を用いたヤコビ法でポアソン方程式を解くことで、双極型、四重極型、八重極型、そして観測的に得られた地球磁場や海王星、天王星の磁場を数値的に再現することに成功した。また、マルチグリッド法を用いることで、ヤコビ法のみで計算した場合と比べて、計算速度が6~29倍に向上することを確認した。

開発したポアソン方程式のマルチグリッドソルバーを、インヤン格子を用いた 低マッハ数近似ダイナモシミュレーションコードに適用することが今後の目標で ある。

目 次

		の活動とダイナモ理論
1.1		f動とその物理的要因
1.2		デイナモ機構と MHD 方程式
1.3		バイナモシミュレーションの問題点と低マッハ数近似 MHD 方
	程式	
第 2		复法とマルチグリッド法
2.1		<u> </u>
	2.1.1	ヤコビ法....................................
	2.1.2	重み付きヤコビ法
		Gauss-Seidel 法
		Red-Black Gauss-Seidel 法
2.2		△較
2.3	異なる	5周波数モードの収束率
2.4	マルチ	- グリッド法
2.5	Interp	olation & Restriction
	2.5.1	Interpolation
	2.5.2	Restriction
2.6	V-Cyc	ele マルチグリッド
2.7	マルチ	ーグリッド法の性能比較
第3	章 マノ	レチグリッド法のインヤン格子への応用
第3 3.1	-	レチグリッド法のインヤン格子への応用 ペテンシャルとインヤン格子
-	磁場が	ペテンシャルとインヤン格子
3.1	磁場が	
3.1	磁場式 厳密解	ポテンシャルとインヤン格子
3.1	磁場 厳密解 3.2.1	ポテンシャルとインヤン格子
3.1	磁場ボ 厳密解 3.2.1 3.2.2 3.2.3	ポテンシャルとインヤン格子 解と数値解の比較 双極子 (dipole) 磁場 四重極子 (quadrupole) 磁場 八重極子 (octupole) 磁場
3.1 3.2	磁場 厳密解 3.2.1 3.2.2 3.2.3 天体~	ポテンシャルとインヤン格子
3.1 3.2	磁場ポ 厳密解 3.2.1 3.2.2 3.2.3 天体へ 3.3.1	ポテンシャルとインヤン格子 異と数値解の比較 双極子 (dipole) 磁場 四重極子 (quadrupole) 磁場 八重極子 (octupole) 磁場
3.1 3.2	磁場が 厳密解 3.2.1 3.2.2 3.2.3 天体へ 3.3.1 3.3.2	ポテンシャルとインヤン格子 解と数値解の比較 双極子 (dipole) 磁場 四重極子 (quadrupole) 磁場 八重極子 (octupole) 磁場 いの応用 地球
3.1 3.2 3.3	磁場が 厳密解 3.2.1 3.2.2 3.2.3 天体へ 3.3.1 3.3.2 計算性	ポテンシャルとインヤン格子 We と数値解の比較 双極子 (dipole) 磁場 四重極子 (quadrupole) 磁場 八重極子 (octupole) 磁場 いの応用 地球 天王星、海王星

1 序論 太陽の活動とダイナモ理論

1.1 太陽活動とその物理的要因

太陽は太陽系の中心天体であり、その活動は我々の住む地球に様々な影響を及ぼしている。例えば、太陽活動の中長期的変動は温暖化や寒冷化の直接的な原因となり、数百年から数億年スケールの全地球規模の気候変動をもたらす。また、太陽活動が11年という比較的短いサイクルで静穏期と活動期を繰り返すことも知られており、活動期に頻発する巨大な太陽フレアやそれに付随したコロナ質量放出(CME)は、宇宙飛行士を被爆の危険にさらすだけでなく、人工衛星に搭載された電子機器や電力システムの破壊の原因にもなる。

過去数十年にわたる太陽活動の精密観測により、太陽の活動性の源は「磁場」であることがわかってきた。太陽内部で増幅された磁気エネルギーが表面に供給され、蓄積された磁気エネルギーの爆発的解放によって太陽フレアなど様々な太陽活動が引き起こされている。太陽の磁気活動性の指標の1つとして知られるのが、太陽表面に現れる黒点である。太陽表面に存在する黒点の総量(太陽黒点数)とその出現緯度を時間軸(横軸)に沿ってプロットしたのがFig.1のバタフライダイアグラムである。11年周期で黒点数やその出現緯度が変化していることがわかる。

太陽の活動性についての理解が深まる一方で、その原因である磁場の生成・増幅過程、すなわち太陽ダイナモ過程については未だに多くの謎が残されている。太陽ダイナモ機構の解明は、太陽活動やその変動の原因の理解に直結するだけでなく、地球の気候が今後どのように変化していくかを予見するためにも極めて重要である。

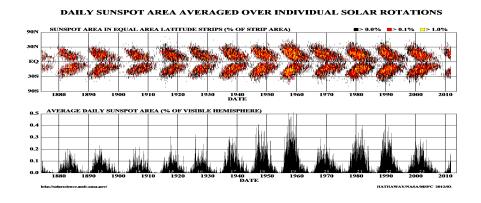


Fig. 1: Butterfly diagram (Citation from [1])

1.2 太陽ダイナモ機構と MHD 方程式

Fig.2 に示すように太陽は多層構造であり、中心から中心核・放射層・対流層・(光球)・彩層・コロナの順に並んでいる。太陽は極めて高温で、その主成分である水素やヘリウムなどの物質はプラズマ状態になっている。光球以下の電気伝導性の高い高温プラズマの運動にともない大局的な電流が流れることで、太陽の磁場が生成・増幅されると考えられている。この太陽内部での大規模な磁場生成・増幅の仕組みを、発電機(ダイナモ)になぞらえて太陽ダイナモと呼ぶ。同様の現象は地球内部にも見られるが、地球の場合は外核に存在する液体鉄内を流れる電流が磁場の起源である。ダイナモ機構で生じる地球の磁力線は、しばしば北極から南極へ向かってまるで地球が一つの棒磁石であるかのような秩序だった磁気双極子構造で描かれる。太陽も磁気双極子成分は持つが、黒点等の強く時間変動する磁気的活動領域が赤道近傍に存在するため、多くの場合、その磁場は単純な双極子では記述できない複雑な構造になっている。

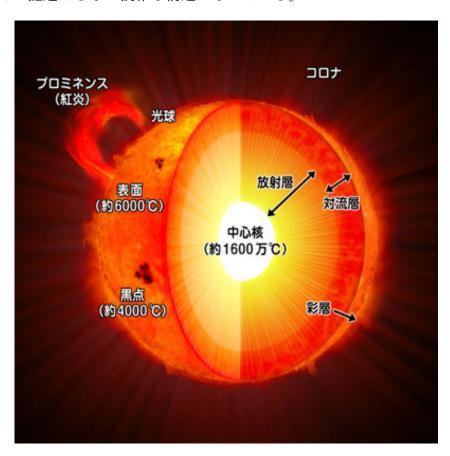


Fig. 2: Internal Structure of the Sun (Citation from[2])

太陽内部のプラズマの運動は以下の圧縮性 MHD (Magnet-Hydro Dynamics) 方程式で記述される。

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \boldsymbol{f} \;, \tag{1}$$

$$\frac{\partial \boldsymbol{f}}{\partial t} = -\nabla \cdot (\boldsymbol{v}\boldsymbol{f}) - \nabla p + \boldsymbol{j} \times \boldsymbol{B} + \rho \boldsymbol{g} + 2\rho \boldsymbol{v} \times \boldsymbol{\Omega} + \mu \left[\nabla^2 \boldsymbol{v} + \frac{1}{3} \nabla (\nabla \cdot \boldsymbol{v}) \right] , \quad (2)$$

$$\frac{\partial p}{\partial t} = -\boldsymbol{v} \cdot \nabla p - \gamma p \nabla \cdot \boldsymbol{v} + (\gamma - 1)\kappa \nabla^2 T + (\gamma - 1)\eta \boldsymbol{j}^2 + (\gamma - 1)\boldsymbol{\Phi} , \qquad (3)$$

$$\frac{\partial \mathbf{A}}{\partial t} = \mathbf{v} \times \mathbf{B} + \eta \nabla^2 \mathbf{A} , \qquad (4)$$

where

$$p = \rho T , \; \boldsymbol{B} = \nabla \times \boldsymbol{A} , \; \boldsymbol{j} = \nabla \times \boldsymbol{B} , \; \boldsymbol{g} = -g_0/r^2 \hat{\boldsymbol{r}} ,$$

$$\nabla \cdot \boldsymbol{B} = 0 , \; \boldsymbol{\Phi} = 2\mu \left[e_{ij} e_{ij} - \frac{1}{3} (\nabla \cdot \boldsymbol{v})^2 \right] , \; e_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_i} + \frac{\partial v_j}{\partial x_i} \right) .$$

ここで、 ρ は質量密度、pは圧力、vは速度場、Bは磁場、fは質量流速密度、Aは磁場のベクトルポテンシャル、jは電流密度、 γ は比熱比、 μ は粘性率、 κ は熱伝導率、 η は電気抵抗率である。MHD 方程式は8変数の非線形連立偏微分方程式であり、近似無しに厳密解を求めることは極めて難しく、現在の人類の数学的能力ではほぼ不可能と言っても過言ではない。このような厳密解を求めることのできない複雑な問題に対して威力を発揮するのが、計算機を使って数値的に近似解を求める手法、シミュレーションである。太陽内部のプラズマの運動を理解し太陽ダイナモ機構を解明するためには、大型計算機を使った MHD 方程式の非線形シミュレーションが不可欠である。

1.3 太陽ダイナモシミュレーションの問題点と低マッハ数近似 MHD 方程式

MHD 方程式を陽的に離散化する際、時間刻み幅に対する上限を与えるのが、以下の CFL 条件(Courant-Friedrichs-Lewy condition)である:

$$\frac{\Delta x}{\Delta t} > C_s \ . \tag{5}$$

ここで C_s は音速、 Δt は計算の時間刻み幅、 Δx は計算格子の幅である。これは シミュレーションにおいて「情報が伝播する速さ」を「実際の現象で波や物理量 が伝搬する速さ」よりも早くしなければならないという制約からくる条件である。 この条件を破ると数値発散が生じ、物理的に意味のない解を得ることになる。実 は、太陽ダイナモのシミュレーション研究のボトルネックになっているのがこの CFL条件である。太陽内部のプラズマの運動速度は音速に比べて桁違いに小さい (3~4桁) ことが知られている [3]。CFL条件は (MHD) 流体シミュレーションの 時間刻み幅を音速で制限するため、音速よりも極めて速度の遅いプラズマの運動 にともなう物理現象の時間発展を物理的に意味のある時刻まで追うためには膨大な数の時間積分が必要になる。現状では、どんなに高性能な計算機を持ってしても、圧縮性 MHD 方程式の下では現実的な太陽ダイナモ計算は実現不可能である。太陽ダイナモのシミュレーション研究を推進するために、プラズマの運動と音波の間に存在する数桁の速度の隔たりを解消する必要がある。

音速とプラズマの運動速度の間に横たわる速度のギャップを解消するための一つの手法が低マッハ数近似である。これは、音速 C_s に比べてプラズマの運動速度 V が十分に小さいという仮定のもと、マッハ数 $M \equiv V/C_s$ の高次の項を落とすことで、ダイナミクスに対する音波の寄与が無視できる形に MHD 方程式を近似 する手法である。低マッハ数近似を施した MHD 方程式を使用することで、系の時間発展を音速ではなくプラズマの運動速度で決まる CFL 条件で解くことができるようになる。低マッハ数近似を施した流体の方程式は以下のように記述される [4]。

$$\frac{\partial \mathbf{f}}{\partial t} = -\nabla \cdot (\mathbf{v}\mathbf{f}) - \nabla \pi + \rho \mathbf{g} , \qquad (6)$$

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\rho \boldsymbol{v} h) = \nabla \cdot \kappa \nabla T , \qquad (7)$$

$$\nabla \cdot \boldsymbol{v} = \frac{1}{\rho \frac{\partial p}{\partial \rho}} \left[\frac{1}{\rho c_p} \frac{\partial p}{\partial T} \left(\nabla \cdot \kappa \nabla T \right) \right] \equiv S . \tag{8}$$

ここで π は、圧力pをマッハ数で展開したときの、マッハ数の2乗に比例した圧力摂動量である。

$$p(x,t) = p_0(t) + Mp_1(t) + M^2\pi(x,t) .$$

この低マッハ数近似を MHD 方程式に施した『低マッハ数近似 MHD 方程式』を安定かつ高精度に解く計算コードを開発することが、太陽ダイナモのシミュレーション研究にとって必要不可欠である。

本研究の目的は、式(8)の中に現れる速度場のポアソン方程式を高速に解くためのマルチグリッドソルバーを開発し、インヤン格子を用いた MHD シミュレーションコードに実装することである。最終的には、開発した新しいインヤン格子 MHD コードを使って世界最高精度の太陽ダイナモシミュレーションを行い、太陽の磁場の起源についての新しい知見を得ることを目標とする。

第2章 反復法とマルチグリッド法 2

反復法 2.1

MHD 方程式のような複雑な微分方程式の階を数値的に得るためには、問題を 離散化した後、行列形式の方程式 Au=f を解く必要がある。その一般的な方法 として直接法と反復法がある。直接法とは逆行列 A-1 を求める方法であり、丸め 誤差以外の誤差なしに厳密に方程式を解くことが出来るが、行列の階数が上がる につれて計算量が増加するため今回のような大規模計算には不適である。反復法 とは、近似解vを求めその近似解を方程式 Av = f に代入して、その誤差から近似 解を修正して厳密解に近づけていく方法である。計算ステップは有限回でなけれ ばならないので、この方法では丸め誤差以外にも打ち切り誤差が発生するが、直 接法に比べて計算速度が速いのが特徴である。本研究ではポアソン方程式を解く ための手法として反復法を用いる。

反復法にも様々な種類がある。本章では一般的な幾つかの反復解法をまとめる。 反復法を適応する問題は以下の1次元ポアソン方程式である。

$$u''(x) = f(x), \ u(0) = u(1) = 0, \ (0 \le x \le 1)$$
 (9)

u''(x)を中央差分近似で差分化すると、

$$v_i = u(x_i) , f_i = f(x_i) , v_0 = v_n = 0 , h = \frac{1}{n} ,$$
 (10)

$$v_i'' = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} \,, \tag{11}$$

$$v_i'' = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} , \qquad (11)$$

$$\frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} = f_i , (1 \le i \le n - 1) , \qquad (12)$$

となり、行列形式では、

$$A\mathbf{v} = \mathbf{f} \,, \tag{13}$$

$$A = \frac{1}{h^2} \begin{bmatrix} 2 & -1 & 0 & \cdots & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & \cdots & \vdots \\ 0 & -1 & 2 & -1 & 0 & \cdots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \cdots & 0 & -1 & 2 & -1 & 0 \\ \vdots & \cdots & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & \cdots & \cdots & 0 & -1 & 2 \end{bmatrix}, \boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ \vdots \\ \vdots \\ v_{n-2} \\ v_{n-1} \end{bmatrix}, \boldsymbol{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{n-2} \\ f_{n-1} \end{bmatrix},$$

と書ける。uが厳密解、vが近似解であり、e = u - vを誤差 (error)、r = f - Avを残差 (residual) と呼ぶ。

2.1.1 ヤコビ法

まずヤコビ法についてまとめる。(12) 式の両辺に h^2 をかけると

$$-v_{i-1} + 2v_i - v_{i+1} = h^2 f_i , (1 \le i \le n - 1)$$
 (14)

$$u_0 = u_n = 0 ,$$

と書き直すことができる。次に初期値 $v^{(0)}$ を決め、次ステップのi番目の値を以下のように更新する。

$$v_i^{\text{(new)}} = \frac{1}{2} (v_{i-1}^{\text{(old)}} + v_{i+1}^{\text{(old)}} + h^2 f_i) , (1 \le i \le n - 1)$$
 (15)

ここで現在の近似値を $v^{(\text{old})}$ 、更新した近似値を $v^{(\text{new})}$ と表記している。全ての点において $v^{(\text{new})}$ の更新が行われると、その点が次の計算ステップの $v^{(\text{old})}$ となる。この反復計算を誤差が収束するまで繰り返す。この方法をヤコビ法と呼ぶ。

次に、ヤコビ法を行列形式で表す。まず行列 A を A = D - L - U に分解する。ここで D は A の対角成分、-L と -U は A の下三角成分と上三角成分であり、以下のように書き下せる。

$$D = \begin{bmatrix} 2 & 0 & \cdots & \cdots & 0 \\ 0 & 2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 2 & 0 \\ 0 & \cdots & \cdots & 0 & 2 \end{bmatrix}, \ L = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & 0 \\ 0 & \cdots & \cdots & 1 & 0 \end{bmatrix}, \ U = \begin{bmatrix} 0 & 1 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 1 \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix},$$

これを用いると

$$(D - L - U)\boldsymbol{v} = h^2 \boldsymbol{f} , \qquad (16)$$

$$\Leftrightarrow D\mathbf{v} = (L+U)\mathbf{v} + h^2\mathbf{f} , \qquad (17)$$

$$\Leftrightarrow \boldsymbol{v} = D^{-1}(L+U)\boldsymbol{v} + D^{-1}h^2\boldsymbol{f} , \qquad (18)$$

となる。 $R_I = D^{-1}(L+U)$ と定義すると、ヤコビ法は行列形式で

$$\boldsymbol{v}^{(\text{new})} = R_J \boldsymbol{v}^{(\text{old})} + D^{-1} h^2 \boldsymbol{f} ,$$

と書き表すことができる。

2.1.2 重み付きヤコビ法

次に、ヤコビ法に重みを付けて修正する重み付きヤコビ法についてまとめる。 まず v^* をヤコビ法と同様にして、

$$v_i^* = \frac{1}{2} (v_{i-1}^{\text{(old)}} + v_{i+1}^{\text{(old)}} + h^2 f_i) , \qquad (19)$$

と表す。この v* を用いて次ステップの近似解を以下のように決める。

$$v_i^{\text{(new)}} = (1 - \omega)v_i^{\text{(old)}} + \omega v_i^* . \tag{20}$$

これが重み付きヤコビ法である。 ω の値を調節すると、ヤコビ法よりも効率よく解を求めることが出来る。これを行列形式で表すと、

$$v_i^{\text{(new)}} = (1 - \omega)v_i^{\text{(old)}} + \frac{\omega}{2}(v_{i-1}^{\text{(old)}} + v_{i+1}^{\text{(old)}} + h^2 f_i), \qquad (21)$$

$$\boldsymbol{v}^{(\text{new})} = \left[(1 - \omega)I + \omega D^{-1}(L + U) \right] \boldsymbol{v}^{(\text{old})} + \omega h^2 D^{-1} \boldsymbol{f} , \qquad (22)$$

ここで、 $R_{\omega} = [(1-\omega)I + \omega D^{-1}(L+U)] = (1-\omega)I + \omega R_J$ とすると、

$$\boldsymbol{v}^{(\text{new})} = R_{\omega} \boldsymbol{v}^{(\text{old})} + \omega h^2 D^{-1} \boldsymbol{f} , \qquad (23)$$

となる。

2.1.3 Gauss-Seidel 法

Gauss-Seidel 法 (以下 GS 法) のアルゴリズムは、ヤコビ法に簡単な変更を加えるだけであり、その時の $v_i^{(\text{new})}$ は

$$v_i^{\text{(new)}} = \frac{1}{2} (v_{i-1}^{\text{(new)}} + v_{i+1}^{\text{(old)}} + h^2 f_i) , \qquad (24)$$

となる。変更点は、 $v_{i-1}^{(\text{old})}$ が $v_{i-1}^{(\text{new})}$ となっていることであり、ヤコビ法では全てのi で計算を終えてから値を更新するのに対し、GS 法では計算が終わった値はすぐに更新され直後の計算に利用される。この改良により収束速度が上がる。またヤコビ法の場合には、グリッド数n の格子点の場合「2n」個分のメモリ容量が必要であったのに対し、GS 法では「n」個分の容量で済むというメリットもある。

GS法を行列形式で表すと、

$$A = (D - L - U) ,$$

$$(D - L)\boldsymbol{u} = U\boldsymbol{u} + \boldsymbol{f} ,$$

$$\boldsymbol{u} = (D - L)^{-1}U\boldsymbol{u} + (D - L)^{-1}\boldsymbol{f} ,$$

ここで、 $R_G = (D-L)^{-1}U$ とおくと、

$$\boldsymbol{v}^{(\text{new})} = R_G \boldsymbol{v}^{(\text{old})} + (D - L)^{-1} \boldsymbol{f} , \qquad (25)$$

となる。

2.1.4 Red-Black Gauss-Seidel 法

GS 法はヤコビ法よりも計算効率も良くメモリ使用も少なくてすむが、値を更新する順番が重要で、安易に変更出来ない。そのため、GS 法は並列計算に不適であるという欠点を持つ。

そこで GS 法の発展版として、Red-Black Gauss-Seidel 法 (以下 RBGS 法) が提案されている。これはグリッドを偶数番目と奇数番目に分割し、先に全ての偶数 (Red) グリッドの値を更新した後に奇数 (Black) グリッドの値を更新するというアルゴリズムである。

$$Red \ part : v_{2i}^{\text{(new)}} = \frac{1}{2} (v_{2i-1}^{\text{(old)}} + v_{2i+1}^{\text{(old)}} + h^2 f_{2i}) , \qquad (26)$$

Black part:
$$v_{2i+1}^{\text{(new)}} = \frac{1}{2} (v_{2i}^{\text{(new)}} + v_{2i+2}^{\text{(new)}} + h^2 f_{2i+1})$$
. (27)

この修正により更新の順序に制限がなくなり、あらゆる順番で計算することが可能になる。つまり複数のの独立なプロセッサに値の更新を分散する並列計算が可能になる。

2.2 性能比較

実際にヤコビ法、重み付きヤコビ法、GS 法、RBGS 法の 4 つの方法で簡単な例題を解き、その性能を比較する。 例題として $A\mathbf{u}=0$ 、つまり式 (13) の \mathbf{f} を 0 と置いた問題を考える。この問題の厳密解は $\mathbf{u}=0$ なので誤差の評価がしやすい。つまり誤差 $\mathbf{e}\dot{\mathbf{m}}-\mathbf{v}$ となる。

各反復解法の漸化式をまとめる:

$$Jacobi: v_j^{(n+1)} = \frac{1}{2}(v_{j-1}^{(n)} + v_{j+1}^{(n)}).$$
 (28)

Weighted Jacobi: $v_j^{(*)} = \frac{1}{2}(v_{j-1}^{(n)} + v_{j+1}^{(n)}),$

$$v_j^{(n+1)} = (1-\omega)v_j^{(n)} + v_j^{(*)}. \tag{29}$$

Gauss Seidel:
$$v_j^{(n+1)} = \frac{1}{2} (v_{j-1}^{(n+1)} + v_{j+1}^{(n)})$$
. (30)

Red Black Gauss Seidel:
$$v_{2j}^{(n+1)} = \frac{1}{2}(v_{2j-1}^{(n)} + v_{2j+1}^{(n)})$$
,

$$v_{2j+1}^{(n+1)} = \frac{1}{2} (v_{2j}^{(n+1)} + v_{2j+2}^{(n+1)}).$$
 (31)

初期値には、

$$v_j^{(0)} = \sin\left(\frac{jk\pi}{n}\right) , \quad (0 \le j \le n , \ 1 \le k \le n-1)$$
 (32)

のフーリエモードを採用する。ここでjはベクトルvの成分、kは波の波数を表す。 グリッド分割数 n は n=64、反復回数 (iteration) は最大で 100 回とする。初期値 に k=1、k=3、k=6 を代入した時のそれぞれの結果を Fig.3 \sim Fig.6 に示す。 いずれの方法でも、反復回数の増加にともない誤差は減少している。 また波数 k が大きいほど減少率は大きくなっている。

一般的な問題の初期値は 1 つの波数 (モード) だけで構成されている訳ではなく、複数のモードが混在する波になる。よって次に k=1、k=6、k=32 の 3 つのモードが混在した波を考える。初期値を以下のように設定する。

$$v_j^{(0)} = \frac{1}{3} \left[\sin \left(\frac{j\pi}{n} \right) + \sin \left(\frac{6j\pi}{n} \right) + \sin \left(\frac{32j\pi}{n} \right) \right] . \tag{33}$$

この初期値を、重み付きヤコビ法で解いた結果がFig.7である。この図より、最初の数ステップで急激に誤差が減少し、その後は減少率が小さくなることがわかる。これは、最初の数ステップで短波長モードの大半が減少する一方、長波長モードの誤差がなかなか減少せずに存在し続けるためである。

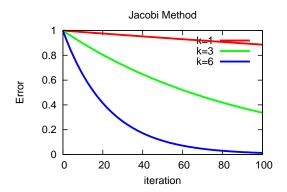
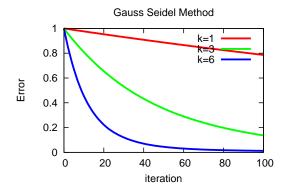


Fig. 3: Jacobi method (Vertical axis indicates the error and the horizontal one is the iteration number. Different colors denote the different initial wave numbers).

Fig. 4: Weighted Jacobi method.



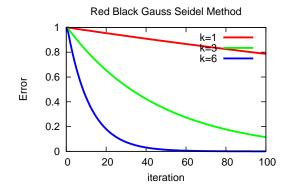


Fig. 5: Gauss-Seidel method.

Fig. 6: Red-Black Gauss-Seidel method.

Weighted Jacobi Method.ω=2/3.mixed initial guess.(k=1,k=6,k=32)

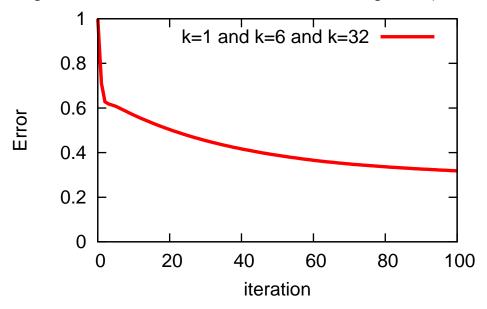


Fig. 7: Weighted Jacobi iteration for the mixture mode.

2.3 異なる周波数モードの収束率

前節でまとめた反復解法は全て

$$\boldsymbol{v}^{(\text{new})} = R\boldsymbol{v}^{(\text{old})} + \boldsymbol{g} ,$$
 (34)

という形で表される。ここで R はこれまで求めた各反復行列 (iteration matrix) である。当然これは厳密解に対しても成り立つので、

$$\boldsymbol{u} = R\boldsymbol{u} + \boldsymbol{g} \;, \tag{35}$$

も成立する。(35)式から(34)式を引くと

$$\boldsymbol{u} - \boldsymbol{v}^{(\text{new})} = R \left(\boldsymbol{u} - \boldsymbol{v}^{(\text{old})} \right) ,$$
 (36)

$$e^{\text{(new)}} = Re^{\text{(old)}}, \qquad (37)$$

となる。よってm回目の反復後の誤差 $e^{(m)}$ は、

$$\boldsymbol{e}^{(m)} = R^m \boldsymbol{e}^{(0)} , \qquad (38)$$

と書ける。収束するためには誤差が0に近づけばよいので、

$$\|e^{(m)}\| = \|R^m e^{(0)}\| \le \|R\|^m \|e^{(0)}\|,$$
 (39)

より、|| R ||<1を満たせばよいことがわかる。

この条件を満たすためには、行列 R の固有値 $\lambda(R)$ の最大値 $\rho(R)=\max|\lambda(R)|$ が $\rho(R)<1$ である必要がある。

まず具体例として重み付きヤコビ法を考える。重み付きヤコビ法の反復行列 R_{ω} は $R_{\omega}=(1-\omega)I+\omega R_{J}$ で表される。これを変形すると、

$$R_{\omega} = I - \frac{\omega}{2} \begin{bmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & \cdots & \vdots \\ 0 & -1 & 2 & -1 & 0 & \cdots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \cdots & 0 & -1 & 2 & -1 & 0 \\ \vdots & \cdots & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & \cdots & \cdots & 0 & -1 & 2 \end{bmatrix}, \tag{40}$$

となる。この時、固有値 $\lambda(R_{\omega})$ と $\lambda(A)$ は

$$\lambda(R_{\omega}) = 1 - \frac{\omega}{2}\lambda(A) , \qquad (41)$$

という関係にあり、行列 R_{ω} と A の固有ベクトルが同じになることがわかる。 また行列 A の固有値は

$$\lambda_k(A) = 4\sin^2\left(\frac{k\pi}{2n}\right), \ (1 \le k \le n-1), \tag{42}$$

固有ベクトルは

$$w_{k,j} = \sin\left(\frac{jk\pi}{n}\right), \ (1 \le k \le n-1, \ 0 \le j \le n),$$
 (43)

行列 R_{ω} の固有値は

$$\lambda_k(R_\omega) = 1 - 2\omega \sin^2\left(\frac{k\pi}{2n}\right), \ (1 \le k \le n - 1) \ , \tag{44}$$

である。つまり ω の値が $0 < \omega \le 1$ の時、 $\rho(R_{\omega}) < 1$ が成立し、誤差が収束する。一方、行列Aの固有ベクトルは重要な性質を持つ。それは初期の誤差 $e^{(0)}$ が行列Aの固有ベクトルwで展開出来るというものである。

$$e^{(0)} = \sum_{k=1}^{n-1} c_k \mathbf{w_k} . (45)$$

ここで、 $c_k \in \mathbf{R}$ はそれぞれのモードの誤差を測る指標となる係数である。m 回目の反復後の誤差 $\mathbf{e}^{(m)}$ は

$$e^{(m)} = R_{\omega}^{m} e^{(0)} = \sum_{k=1}^{n-1} c_{k} R_{\omega}^{m} w_{k} = \sum_{k=1}^{n-1} c_{k} \lambda_{k}^{m} (R_{\omega}) w_{k} , \qquad (46)$$

と評価される。最後の変形は行列 A と行列 R_{ω} の固有ベクトルが同じことから、 $R_{\omega} \boldsymbol{w_k} = \lambda_k (R_{\omega}) \boldsymbol{w_k}$ を用いた。

この変形より、誤差はそれぞれのモードで展開でき、(重み付き)ヤコビ法を用いた場合はその収束速度が各モードの固有値に比例することがわかる。

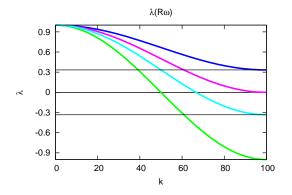
Fig.8 に $\lambda(R_\omega)=1-2\omega\sin^2\left(\frac{k\pi}{2n}\right)$ の波数依存性を示す。横軸は波数 k, $(1\leq k\leq 99)$ 、縦軸は固有値 λ である。 ω は $\omega=1$ (ヤコビ法), $\frac{1}{3}$, $\frac{1}{2}$, $\frac{2}{3}$ の 4 パターンを調べた。収束速度を上げるには $|\lambda(R_\omega)|$ の値をなるべく 0 に近づければ良い。ここでは便宜上、 $1\leq k<\frac{n}{2}$ を低周波、 $\frac{n}{2}\leq k\leq n-1$ を高周波と呼ぶ。k=1 の固有値は、

$$\lambda_1 = 1 - 2\omega \sin^2\left(\frac{\pi}{2n}\right) = 1 - 2\omega \sin^2\left(\frac{\pi h}{2}\right) \approx 1 - \frac{\omega \pi^2 h^2}{2}, \quad \left(h = \frac{1}{n}\right) , \quad (47)$$

となり、グリッドが細かくなるにつれ λ_1 の値が1に近づくことがわかる。これはヤコビ法では低周波の波の収束率が低いことを意味する。一方、高周波の波は、 $\omega=2/3$ を用いると高周波部分の全てのkで $|\lambda|<\frac{1}{3}$ が成り立つ。この時、効率的に解が収束していることがわかる。

Fig.9 は Au=0の解を求める問題において初期値に w_k を与え、様々なモードでの収束までの反復回数を調べたものである。グリッド数は n=64 とし、 $\omega=1$ (ヤコビ法) と $\omega=2/3$ の場合について比較している。図より、ヤコビ法では k が n-1 に近づくにつれて収束速度が遅くなっているが、重み付きヤコビ法ではそれが改善されていることがわかる。しかし、波数が小さな部分はどちらの手法でも収束速度は遅い。

Fig.10 は初期値に k=2 と k=16 のモードの波を混ぜて、 $\omega=\frac{2}{3}$ の重み付きヤコビ法で 10 回反復した結果である。この図からわかるように、高周波の波はすぐに減少するが、低周波の滑らかな波はあまり減少しない。このように、グリッド数が増えるほど低周波の波の収束が悪くなるのは、反復法に共通した欠点である。グリッド数の大きな高精度シミュレーションで反復法を用いるためには、低周波の波をより効率的に収束させる解法を実装することが不可欠である。



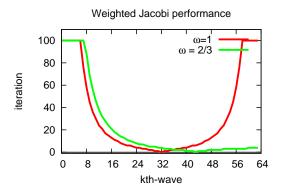


Fig. 8: Eigenvalue of R_{ω} (Blue curve is $\omega = 1/2$, red, skyblue, and green curves correspond to the eigen values of $\omega = 1/3, \omega = 2/3$ and $\omega = 1$).

Fig. 9: The comparison of the converging performance of the Weighted Jacobi method between the cases with $\omega=1$ and $\omega=2/3$

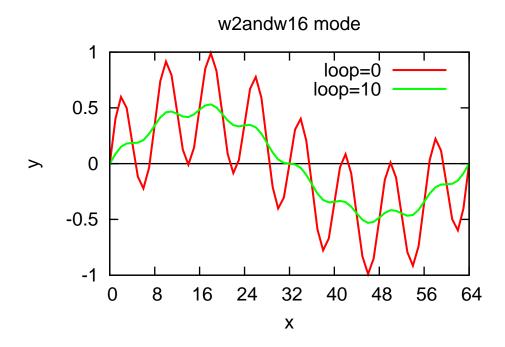


Fig. 10: The profile of mixture mode with k=2 and k=16 before and after the iteration.

2.4 マルチグリッド法

前節で示したように、細かいグリッドの反復計算では低周波の波は収束が悪くなる。この解決法の一つとして、解を求めたいグリッド長よりも長いグリッド長を持つ(グリッド数が少ない)粗いグリッドを用意する。解くべき方程式を粗いグリッド上に移植し(Restriction)、その粗いグリッド上で計算した結果を元の細かいグリッドに移植する(Interpolation)。これを再帰的に繰り返して様々なグリッド長を持つグリッド上で計算を行い、そのグリッド長に対応するモードの誤差を効率良く減らしていく計算方法がある。この方法をマルチグリッド法という[5]。Fig.11のように、細かい(fine)グリッドで見るとなめらかな低周波の波でも、粗い(coarse)グリッドに射影するとギザギザな高周波の波として見ることが出来る。このように低周波のモードの誤差は粗いグリッドを用いて効率よく減らすことで、収束までのスピードを加速させることが出来る。

今後、 Ω^h のような表記をするが、これはグリッド間隔がhの格子を用いるという意味である。

2.5 Interpolation & Restriction

マルチグリッド法において重要な、粗いグリッドから細かいグリッドへの移動 (Interpolation) と細かいグリッドから粗いグリッドへの移動 (Restriction) のアルゴリズムをまとめる。

2.5.1 Interpolation

粗い (coarse) グリッドから細かい (fine) グリッドへの変換アルゴリズムは

$$I^h_{2h} : \Omega^{2h} o \Omega^h (ext{Interpolation})$$

$$I^h_{2h} oldsymbol{v}^{2h} = oldsymbol{v}^h \; ,$$

と表現される。ここで v^h 、 v^{2h} はそれぞれ Ω^h 、 Ω^{2h} での近似値vであり、 I^h_{2h} は $2h \to h$ の変換行列である。具体的には

$$\begin{cases} v_{2j}^h = v_j^{2h} \\ v_{2j+1}^h = \frac{1}{2} (v_j^{2h} + v_{j+1}^{2h}) \end{cases}, \tag{48}$$

という形で変換を行う。Fig.12 に l 次元の Interpolation の概念図をまとめる。 2次元の場合は、

$$\begin{cases}
v_{2i,2j}^{h} = v_{i,j}^{2h} \\
v_{2i,2j+1}^{h} = \frac{1}{2}(v_{i,j}^{2h} + v_{i,j+1}^{2h}) \\
v_{2i+1,2j}^{h} = \frac{1}{2}(v_{i,j}^{2h} + v_{i+1,j}^{2h}) \\
v_{2i+1,2j+1}^{h} = \frac{1}{4}(v_{i,j}^{2h} + v_{i+1,j}^{2h} + v_{i,j+1}^{2h} + v_{i+1,j+1}^{2h}),
\end{cases} (49)$$

となる。Fig.13 に 2 次元の Interpolation の概念図をまとめる。

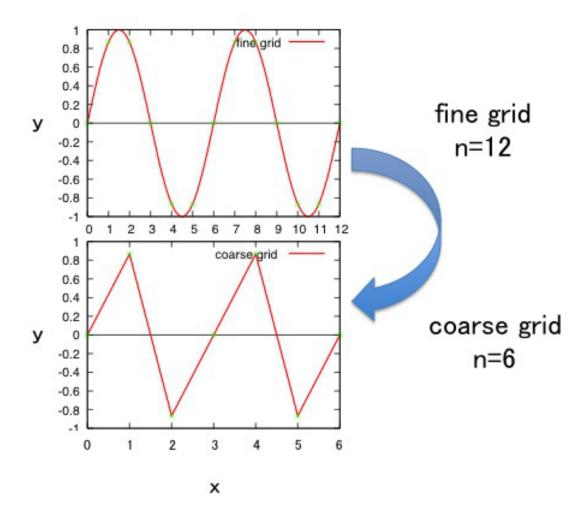


Fig. 11: Sinusoidal wave with k = 4 on fine and coarse grids. The fine grid has a grid number of n = 12 and the coarse grid is n = 6.

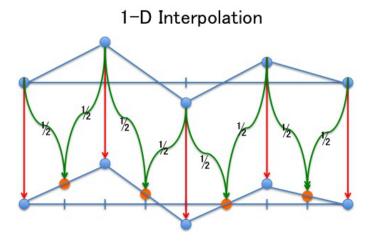


Fig. 12: Schematic view of 1-D Interpolation.

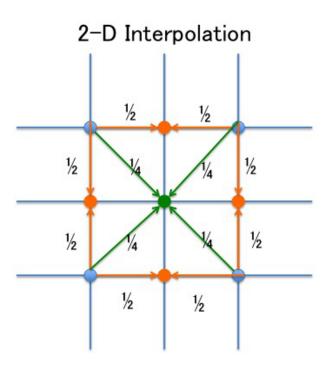


Fig. 13: Schematic view of 2-D Interpolation.

簡単な例として、1次元のグリッド数 n=8 での Interpolation を考える。このとき、変換行列 I_{2h}^h は

$$I_{2h}^{h} \boldsymbol{v}^{2h} = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{1}^{2h} \\ v_{2}^{2h} \\ v_{3}^{2h} \end{bmatrix} = \begin{bmatrix} v_{1}^{h} \\ v_{2}^{h} \\ v_{3}^{h} \\ v_{4}^{h} \\ v_{5}^{h} \\ v_{6}^{h} \\ v_{7}^{h} \end{bmatrix} = \boldsymbol{v}^{h},$$
 (50)

となる。

2.5.2 Restriction

細かい (fine) グリッドから粗い (coarse) グリッドへの変換は以下のアルゴリズムを用いる。

$$I_h^{2h} : \Omega^h \to \Omega^{2h}(\text{Restriction})$$

$$I_h^{2h} \boldsymbol{v}^h = \boldsymbol{v}^{2h} .$$

ここで v_i^{2h} は、

$$v_j^{2h} = \frac{1}{4} (v_{2j-1}^h + 2v_j^h + v_{2j+1}^h) , \qquad (51)$$

である。Fig.14に1次元のRestrictionの概念図をまとめる。2次元の場合は、

$$v_{i,j}^{2h} = \frac{1}{16} (v_{2i-1,2j-1}^h + v_{2i+1,2j-1}^h + v_{2i-1,2j+1}^h + v_{2i+1,2j+1}^h) + \frac{1}{8} (v_{2i,2j-1}^h + v_{2i,2j+1}^h + v_{2i-1,2j}^h + v_{2i+1,2j}^h) + \frac{1}{4} v_{2i,2j}^h,$$
(52)

となる。Fig.15 に 2 次元の Restriction の概念図をまとめる。

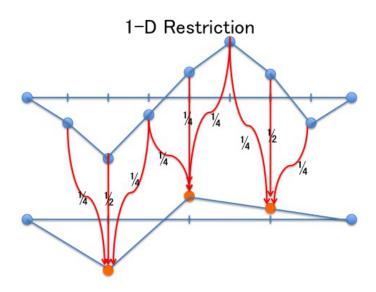


Fig. 14: Schematic view of 1-D Restriction.

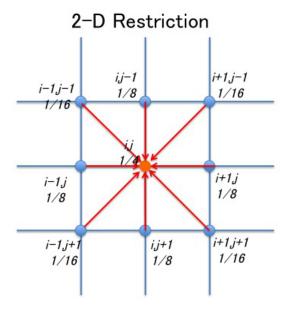


Fig. 15: Schematic view of 2-D Restriction.

グリッド数 n=8 の 1 次元の例で考えると Restriction は、

$$I_{h}^{2h}\boldsymbol{v}^{h} = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} v_{1}^{h} \\ v_{2}^{h} \\ v_{3}^{h} \\ v_{4}^{h} \\ v_{5}^{h} \\ v_{6}^{h} \\ v_{7}^{h} \end{bmatrix} = \begin{bmatrix} v_{1}^{2h} \\ v_{2}^{2h} \\ v_{3}^{2h} \end{bmatrix} = \boldsymbol{v}^{2h} , \qquad (53)$$

となる。これらの変換行列を用いて多段階のグリッドレベルの格子を移動しながら効率良く誤差を収束させていく。

2.6 V-Cycle マルチグリッド

多段階のマルチグリッドでは、様々なグリッドレベルの移動のスケジューリングがある、その中から本研究で用いている V-cycle マルチグリッドについてまとめる。アルゴリズムは再帰的になり、具体的には以下のようにまとめられる。

$V-Cycle\ Scheme$

$$\boldsymbol{v}^h \leftarrow V^h(\boldsymbol{v}^h, \boldsymbol{f}^h)$$

- $(1)A^h \mathbf{u}^h = \mathbf{f}^h$ を与えられた初期値 \mathbf{v}^h で α_1 回計算
- (2) Ω^h が一番粗いグリッドならば (4) へ、それ以外ならば

$$egin{aligned} oldsymbol{f}^{2h} &\leftarrow I_h^{2h} oldsymbol{f}^h \;, \ oldsymbol{v}^{2h} &\leftarrow I_h^{2h} oldsymbol{v}^h \;, \ oldsymbol{v}^{2h} &\leftarrow V^{2h} (oldsymbol{v}^{2h}, oldsymbol{f}^{2h}) \;. \end{aligned}$$

- $(3)\boldsymbol{v}^h \leftarrow I_{2h}^h \boldsymbol{v}^{2h}$
- $(4)A^h u^h = f^h$ を低レベルグリッドで修正した v^h を初期値とし α_2 回計算

模式的に表すと Fig.16 のようになる。その他にも W-Cycle や Full Multigrid V-Cycle など様々なスケジューリングが提案されている。V-Cycle 以外の方法については参考文献 [5] を参照されたい。

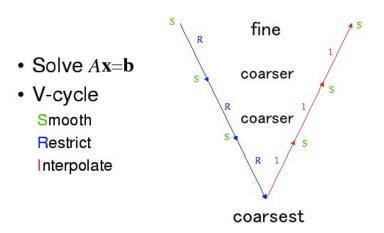


Fig. 16: Scheduling of V-Cycle (Citation from[6])

2.7 マルチグリッド法の性能比較

マルチグリッド法の性能を調べるために、グリッド数 n=256、初期値 $v^{(0)}=\frac{1}{2}\left[\sin\left(\frac{16j\pi}{n}\right)+\sin\left(\frac{40j\pi}{n}\right)\right]$ で重み付きヤコビ法のみで計算した結果と、重み付きヤコビ法にマルチグリッド法を用いて計算した結果を Fig.17 で比較する。マルチグリッドは 7 段階のグリッドレベルで V-Cycle 行い、 $\alpha_1=10$ 、 $\alpha_2=10$ を用いている。また、反復回数はマルチグリッド法を用いた方法で 10 回、用いない方法で 140 回行った。さらに初期値を $v^{(0)}=\frac{1}{4}\left[\sin\left(\frac{4j\pi}{n}\right)+\sin\left(\frac{8j\pi}{n}\right)+\sin\left(\frac{16j\pi}{n}\right)+\sin\left(\frac{32j\pi}{n}\right)\right]$ に変更して、同様の計算を行った結果が Fig.18 である。赤線が初期値、青線が重み付きヤコビ法のみで用いた結果、緑線がマルチグリッド法を用いた結果である。 どちらの図からもマルチグリッド法を用いることで、収束が遅いなめらかなモードの誤差が収束していることがわかる。

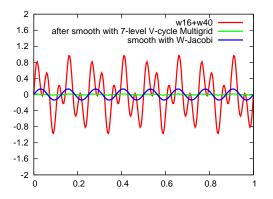


Fig. 17: The comparison of converging performance (The red curve denotes the initial wave profile. The blue and green curves are the results of the Weighted Jacobi and MultiGrid iterations respectively).

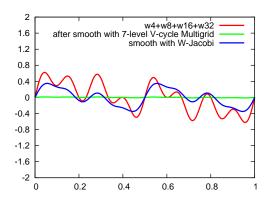


Fig. 18: The comparison of converging performance.

3 第3章 マルチグリッド法のインヤン格子への応用

3.1 磁場ポテンシャルとインヤン格子

本研究の最終目的はマルチグリッド法を用いた低マッハ数近似 MHD 方程式コードを開発することである。その実現へ向けた第一歩として今回はマルチグリッド法のインヤン格子への適用実験を行う。インヤン格子を使ったマルチグリッド法の応用例として太陽や地球など磁場を持つ星の表面から無限遠に向かって伸びるポテンシャル磁場の解を反復解法を使って求める。

星の外側の磁場 \mathbf{B} (r>1) は星の内部の MHD 方程式の磁場と連続的になめらかに接続しなければならないため、以下の条件を満たす必要がある。

$$\nabla \times \boldsymbol{B}_v = 0, \ (r > 1) \ , \tag{54}$$

B を磁場ポテンシャル ψ で表すと、

$$\mathbf{B} = -\nabla \psi, \ (r > 1) \ , \tag{55}$$

となるので、 $\nabla \cdot \mathbf{B} = 0$ より、ラプラス方程式

$$\nabla^2 \psi = \left[\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2}{\partial \phi^2} \right] \psi = 0, \ (r \ge 1)$$
(56)

が得られる。

ここで次のような変数変換を行う。

$$\zeta = 1/r \,, \tag{57}$$

この変換によって式 (56) は以下のような ψ の2階偏微分方程式に書き換えられる。

$$\left[\zeta^2 \frac{\partial^2}{\partial \zeta^2} + \frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta}\right) + \frac{1}{\sin^2 \theta} \frac{\partial^2}{\partial \phi^2}\right] \psi = 0, \ (0 \le \zeta \le 1) \ . \tag{58}$$

これによって、球面から無限遠までの問題を、単位球の内部の有限範囲内の問題に焼き直すことが出来る。式 (58) の原点 $\zeta=0$ での値は、 ポテンシャル磁場の条件 $\psi(r=\infty)=0$ より $\psi(\zeta=0)=0$ となる [8]。

今回の実験ではこの単位球での 2 階偏微分方程式を、単純な球座標による格子ではなく、インヤン格子を用いて解く。インヤン格子とは、本研究室の陰山教授が考案した新しい球座標キメラ格子のアイデアで、イン格子とヤン格子と呼ばれる 2 つの格子が互いに部分的な重なりを許しながら球全体を覆う形となる。この格子を用いることで通常の緯度経度格子法において問題となる座標特異点と、格子間隔の極端な不均一性という二つの問題を解消することができる。具体的には、普通の球座標では高緯度部分において、グリッド間隔が極端に小さくなり、グリッドが密集してしまう。そのため (5) 式の CFL 条件の Δx が小さくなり、それに伴って Δt の条件が厳しくなってしまう。一方インヤン格子ではグリッドが密集する高緯度部分が無く、その部分はもう片方の格子の低緯度部分で補われているため、グリッドが極端に密集するという球座標特有の問題点が解消されるのである。

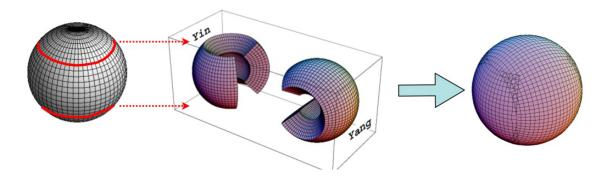


Fig. 19: Yin-Yang grid (Citation from [7]).

イン格子とヤン格子それぞれの境界上の点は、それぞれ互いの格子の中に入り込んでいる。そのため境界上の点は、互いの格子を用いて相互補間を行うことで計算を進める[8]。つまり、マルチグリッド法でグリッドレベルを移動する前に、

イン格子とヤン格子で相互補間を行う必要があり、これは本研究で解決すべき 1 つの問題点である。以下では、インヤン格子にマルチグリッド法を適用して式 (58) を反復法によって解き、星のポテンシャル磁場を求める。

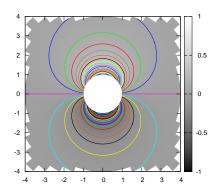
3.2 厳密解と数値解の比較

3.2.1 双極子 (dipole) 磁場

まず最も簡単な例として、双極子型の磁場ポテンシャルを数値計算で求める。 双極子型の磁場ポテンシャルは解析的に

$$\psi_{\text{exact}}(r,\theta,\phi) = c \frac{\cos\theta}{r^2} ,$$
(59)

と表される。c は比例定数であり、今回は c=1 を用いている。この磁場ポテンシャルの厳密解を用いて境界上 (r=1) の値を与え、数値解 ψ と厳密解 $\psi_{\rm exact}$ を比較する。なお、今回の計算で使用した格子数は $(r,\theta,\phi)=65\times99\times291$ である。 Fig 20、21 は磁場ポテンシャルの数値解と厳密解の $\phi=0$ での断面図である。数値解の断面図では計算領域を灰色で塗りつぶしている。これより半径が 1 以上の箇所のみ計算されていることがわかる。



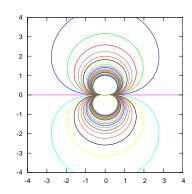


Fig. 20: Numerical solution of dipole field, ψ .

Fig. 21: Exact solution of dipole field, $\psi_{\rm exact}$.

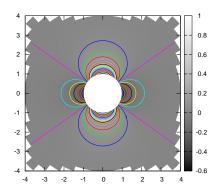
2つの図の比較から、数値計算によって得られた磁場は双極子磁場の厳密解と 高い精度で一致していることがわかる。

3.2.2 四重極子 (quadrupole) 磁場

次に、四重極子型の磁場ポテンシャルを考える。厳密解は解析的に

$$\psi_{\text{exact}}(r,\theta,\phi) = c \frac{3\cos^2\theta - 1}{2r^3} , \qquad (60)$$

と表される。Fig 22、23 は同様に $\phi=0$ での断面図である。四重極の場合も高い精度で近似解と厳密解が一致していることがわかる。



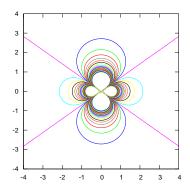


Fig. 22: ψ of quadrupole field.

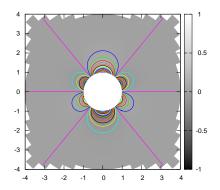
Fig. 23: ψ_{exact} of quadrupole field.

3.2.3 八重極子 (octupole) 磁場

八重極子型の磁場ポテンシャルを計算する。厳密解は解析的に

$$\psi_{\text{exact}}(r,\theta,\phi) = c \frac{5\cos^3\theta - 3\cos\theta}{2r^4} , \qquad (61)$$

と表される。同じく $\phi=0$ での断面図を Fig24、25 に示す。これも同様に高い精度で厳密解が再現されている。



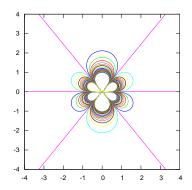


Fig. 24: ψ of octupole field.

Fig. 25: ψ_{exact} of octupole field.

3.3 天体への応用

3.3.1 地球

双極子~八重極子型の磁場ポテンシャルの数値実験で、マルチグリッド法をインヤン格子に応用しても反復計算により高い精度で解が得られることがわかった。 最後に、実際の天体(地球、天王星、海王星)の磁場構造を、インヤン格子上での 反復計算にマルチグリッド法を適用して、数値的に再現する。

初期値には地球の磁場ポテンシャル値を与える。式は以下の形になる[9]。

$$\psi(\theta, \phi, t) = \sum_{n=1}^{n_{\text{max}}} \sum_{m=0}^{n} \left[g_n^m(t) \cos(m\phi) + h_n^m(t) \sin(m\phi) \right] S_n^m(\cos \theta) .$$
 (62)

 $g_n^m \ h_n^m$ は観測から得られたデータ値であり、今回用いた値を Table.1 にまとめる。 S_n^m とは以下の式で定義されるシュミットの半規格化ルジャンドル陪関数である。

$$S_n^m(x) = \begin{cases} (-1)^m \sqrt{\frac{2(n-m)!}{(n+m)!}} P_n^m(x) & (m>0) \\ P_n(x) & (m=0) \end{cases},$$
 (63)

ここで

$$P_n^m(x) = (-1)^m (1 - x^2)^{m/2} \frac{d^m}{dx^m} P_n(x) , \qquad (64)$$

$$P_n(x) = \frac{1}{2^n n!} \left[\frac{d^n}{dx^n} (x^2 - 1)^n \right] . {(65)}$$

インヤン格子で空間離散化された星の表面に式 (62) を用いて磁場ポテンシャルを与え、無限遠で 0 に接続する磁場構造をマルチグリッド法で求めた結果が Fig. 26 である。Fig. 25 までと同様 $\phi=0$ での断面図の磁場ポテンシャルの等高線を表示している。双極型の磁場成分の寄与が強いが、中心部分は完全な双極型ではない。また、磁軸がわずかに斜めに傾いているのが見て取れる。

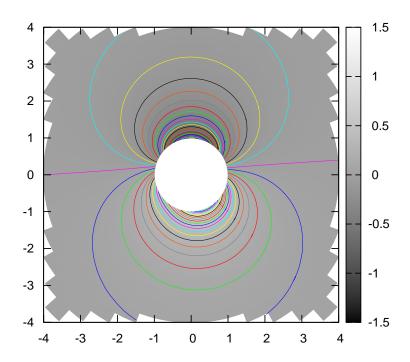


Fig. 26: Geomagnetic field obtained on Yin-Yang grid by using MultiGrid iteration.

Table. 1: The parameters adopted for reproducing geomagnetic field : $g_n^m, h_n^m(\text{Citation from}[9])$

g/h	n	m	パラメータ (nT)		
g	1	0	-29556.8		
g	1	1	-1671.8		
h	1 1 5		5080.0		
g	2	0	-2340.5		
g	g 2		3047.0		
h	2	1	-2594.9		
g	2	2	1656.9		
h	2	2	-516.7		
g	3	0	1335.7		
g	3	1	-2305.3		
h	3	1	-200.4		
g	3	2	1246.8		
h	3	2	269.3		
g	3	3	674.4		
h	3	3	-524.5		
g	4	0	919.8		
g	4	1	798.2		
h	4	1	281.4		
g	g 4		211.5		
h	h 4 2 -225.8		-225.8		
g	4	3	-379.5		
h	4	4 3 145.7			
g	4	4	100.2		
h	4	4	-304.7		

3.3.2 天王星、海王星

次に極めて複雑な磁場構造を持つことで知られる天王星と海王星の磁場をマルチグリッドインヤン格子を使った数値実験で再現する。天王星、海王星の磁場ポテンシャルは地球と同じ形 [式 (62)] で表されるが、定数 g_n^m, h_n^m (観測量) の値が異なる。本研究で用いた n=3 までの値を Table.2 にまとめた。反復計算の結果得られた天王星と海王星の磁場構造が Fig.27,28 である。いずれの磁場構造も、双極子構造からはかけ離れた多重極成分の重ね合わせから成っていることがわかる。以上の数値実験から本研究で開発したマルチグリッド法によるポアソン方程式ソルバーを使って、複雑な磁場構造の数値解をインヤン格子上で効率的に導出することに成功したと言える。

Table. 2: The parameters for the magnetic fields of Uranus and Neptune: $g_n^m, h_n^m(\text{Citation from}[10])$

<u>0])</u>							
g/h	n	m	Uranus	Neptune			
g	1	0	11855	10336			
g	1	1	11507	3359			
h	1	1	-15812	-9772			
g	2	0	-5877	8566			
g	2	1	-13085	-406			
h	2	1	5851	11139			
g	2	2	-605	4644			
h	2	2	4185	-743			
g	3	0	4183	-5749			
g	3	1	-1336	11632			
h	3	1	-5817	-3905			
g	3	2	-6776	-1889			
h	3	2	-357	903			
g	3	3	-4021	-2920			
h	3	3	-2265	-245			

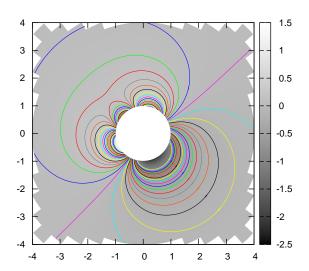


Fig. 27: The magnetic field lines of Uranus.

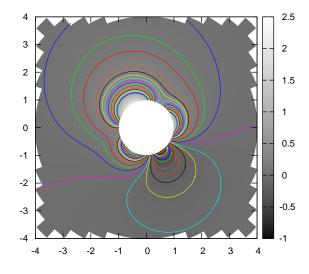


Fig. 28: the magnetic field lines of Neptune

3.4 計算性能比較

最後にTable.3、Fig.29に双極子型~地球磁場をヤコビ法のみで計算した場合にかかった時間と、ヤコビ法にマルチグリッド法を適用して計算した場合にかかった時間をまとめ、比較する。

Table. 3: The comparison of convergence performance ([sec]).

	双極	四重極	八重極	地球磁場型
Jacobi	18110	11899	8636	19363
MultiGrid	3041	494	297	3203
Jacobi/MutiGrid	5.9552779	24.087045	29.077441	6.0452701

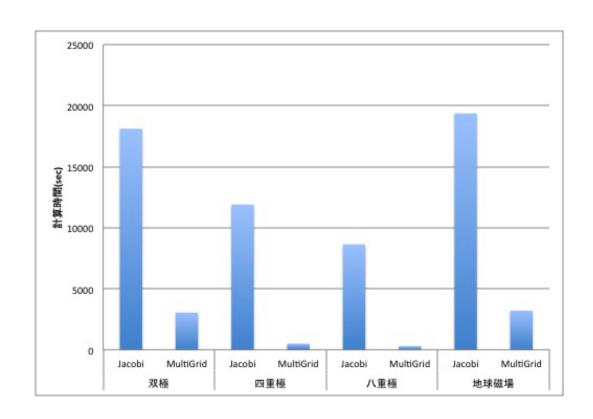


Fig. 29: The comparison of convergence performance.

いずれもマルチグリッド法を用いたほうが計算時間が短くなっていることがわかる。また計算対象によって減少率に差が現れている。その差は八重極型で最大で、ヤコビ法のみの場合と比べておおよそ29倍の加速が実現している。また、双極型、地球磁場型の計算では他の二つに比べるとあまり計算速度が向上していな

いこともわかる。これは双極型の初期値 (cos θ) が低周波な波であったため、最も粗いグリッドで計算しても、マルチグリッドを使った効果を最大に得られることが出来なかったためであると推測される。いずれにせよ、マルチグリッド法を用いることで、ヤコビ法では長い時間を費やさなければならないポアソン方程式を解く問題の計算時間を格段に減少させることが出来ることが数値実験からわかった。

3.5 AVS による可視化

3.4 節までに得られた様々なタイプの磁場構造は、3次元的な構造を観察し、比較することでその違いがよりはっきり区別できる。Fig.30~35 に、汎用可視化ツールである AVS[11] を使った双極型、四重極型、八重極型、地球磁場型、天王星磁場型、海王星磁場型の可視化結果を掲載する。今後は、マルチグリッド法を使った太陽ダイナモシミュレーションコードの開発や太陽ダイナモの研究とともに、太陽ダイナモを詳しく調べることの出来るような新しい可視化手法の開発にも力を注ぎたい。

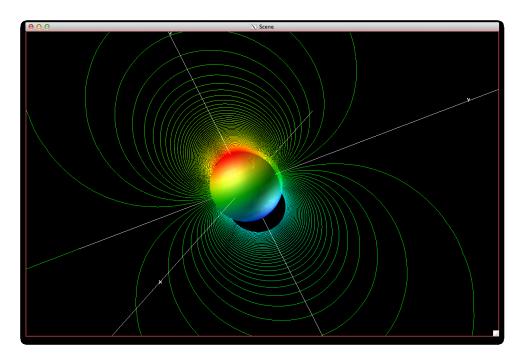


Fig. 30: 3D-Visualization of dipole field obtained on Yin-Yang grid by using MultiGrid poisson solver.

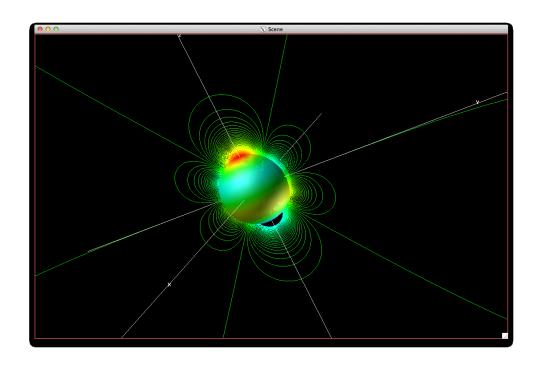


Fig. 31: 3D-Visualization of quadrupole field.

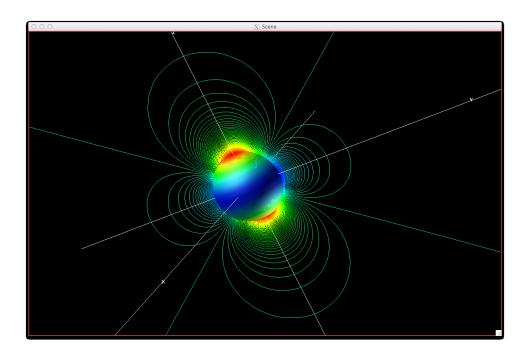


Fig. 32: 3D-Visualization of octupole field.

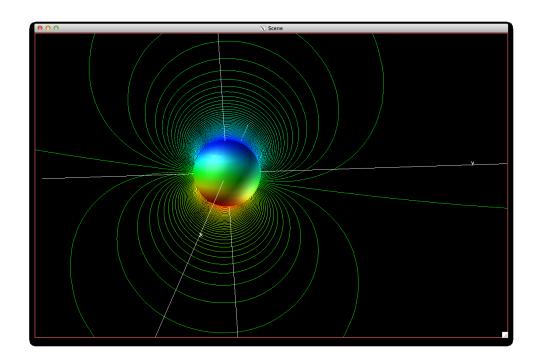


Fig. 33: 3D-Visualization of Geomagnetic field.

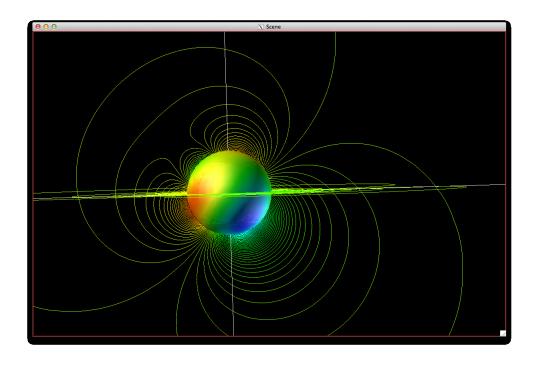


Fig. 34: 3D-Visualization of magnetic field of Uranus.

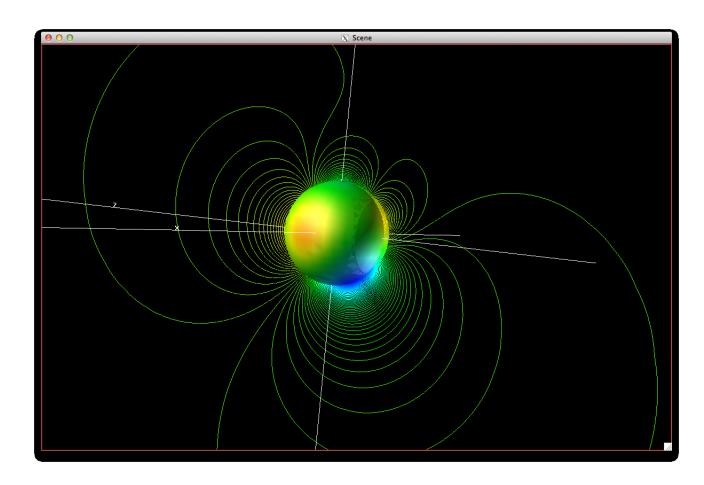


Fig. 35: 3D-Visualization of magnetic field of Neptune.

4 まとめ

磁気流体(MHD)シミュレーションの時間刻み幅は、CFL条件より音速によって制限される。太陽ダイナモシミュレーションのボトルネックとなるのがこの時間刻み幅の制限であり、現実的な時間積分回数で太陽ダイナモ現象を解像するためには、太陽内部の流体速度と音速の間にある3-4桁の速度のギャップを埋める必要がある。つまり、音波の流体ダイナミクスへの寄与を無視し、流体速度で決まる長い時間刻み幅で時間積分できるような新しいダイナモ計算コードの開発が不可欠である。

音波のダイナミクスへの寄与を無視した近似方程式として、低マッハ数近似 MHD 方程式が知られている。現在我々は、当研究室の陰山聡教授が開発したインヤン格子を用いて低マッハ数近似 MHD 方程式を解く、新しいダイナモシミュレーションコードの開発を進めている。本研究では、低マッハ数近似 MHD 方程式の中に現れる速度場のポアソン方程式を高速に解くためのマルチグリッドソルバーを開発し、インヤン格子への適用実験を行った。

マルチグリッド法とは、微分方程式の陰的反復計算をグリッド間隔の異なる複数階層で行うことで、様々な波長を持つ誤差を効率的に減少させる手法のことである。この方法を用いることで、ポアソン方程式などの微分方程式を数値的に解く際の行列計算にかかる時間を、劇的に減少させることができる。

今回行ったマルチグリッド法の適用実験では、星の表面から出て無限遠でゼロに漸近するポテンシャル磁場を記述するポアソン方程式を、インヤン格子上で数値的に解くことを試みた。本研究の結果、マルチグリッド法を用いたヤコビ法で解くことで、双極型、四重極型、八重極型、そして観測的に得られた地球磁場や海王星、天王星の磁場を数値的に再現することに成功した。また、ヤコビ法のみでポテンシャル磁場を計算した場合と比べて、計算速度がおおよそ6~29倍に向上することを確認した。

今後は、今回開発したマルチグリッドソルバーを大規模計算機上で実用化するために、MPIを用いた並列化を進める必要がある。最終的には、並列化*最適化されたマルチグリッドソルバーを、インヤン格子を用いた低マッハ数近似太陽ダイナモシミュレーションコードに適用し、太陽内部の流体の長時間進化を数値的に調べることで、太陽磁場の起源についての新たな知見を得ることが目標である。

謝辞

本研究にあたり、熱心にご指導して頂き様々なアドバイスや的確な助言を下さった政田洋平助教、陰山聡教授に深く感謝致します。また同研究室の山浦優気氏には研究に関する様々な助言を頂きました。

参考文献

- [1] http://solarscience.msfc.nasa.gov/SunspotCycle.shtml
- [2] http://www.gekkou.or.jp/g-3/ht3-07/event-g-sun.html
- [3] Carolus J. Schrijver, George L. Siscoe "Heliophysics" (2010)
- [4] J.B.Bell , M.S.Day , C.A.Rendleman , S.E.Woosley and M.A.Zingale "Adaptive low Mach number simulations of nuclear flame microphysics" , pp. 679-681 (2004)
- [5] William L. Briggs, Van Emden Henson, Steve F. McCormick "A Multigrid Tutorial Second Edition" siam.
- [6] http://www.cs.berkeley.edu/%7Erichie/cs267/mg/report/node2.html
- [7] http://www.research.kobe-u.ac.jp/csi-viz/research/index.ja.html
- [8] Akira Kageyama and Masaki Yoshida "Geodynamo and mantle convection simulations on the Earth Simulator using the Yin-Yang grid ",pp. 333-336 (2005)
- [9] Susan Macmillan and Stefan Maus "International Geomagnetic Reference Field the tenth generation", pp.1135-1137(2005)
- [10] Sabine Stanley, Jeremy Bloxham "Numerical dynamo models of Uranus' and Neptune's magnetic fields", pp. 556-557
- [11] http://www.cybernet.co.jp/avs/