

2011 年度

修士論文

宇宙天気モデリング用球データ可視化ソフトウェア

神戸大学大学院システム情報学研究科
計算科学専攻

吉田 真人

指導教員		陰山	聡	教授
審査教員	主査	臼井	英之	教授
	副査	陰山	聡	教授
	副査	田中	成典	教授

2012 年 2 月 6 日

Visualization Software of Spherical Data for Space Weather Modeling

Masato Yoshida

Abstract

Computer simulation of space weather is important since the modern technological infrastructure is affected by the space environment. Among various research on space weather, Joint Project for Space Weather Modeling (JPSWM) is a Japanese project in which simulation codes for the dynamics of solar corona, solar wind, magnetosphere, and ionosphere are combined. Among these component codes, the simulation code for the solar corona is the most important one, since the coronal activity is the driver of all other dynamical events in the solar system. We have developed an interactive visualization software, “SV4” for JPSWM project. One of characteristic features of SV4 is that it visualizes data defined on a spherical grid system named Yin-Yang grid. Although SV4 is developed as a part of JPSWM project, it can be used in other projects that make use of Yin-Yang grid. Since Yin-Yang grid is now widely used in various fields in geophysics and astrophysics, SV4 would be useful for them, too. Another characteristic feature of SV4 as a visualization software is its base programming language. SV4 is coded in Fortran90 with a graphics library f90gl which is a Fortran interface for OpenGL and GLUT. Having found that most researchers involving JPSWM project use Fortran90/95 and they are not familiar with C/C++, which are standard languages used in computer graphics, we have decided to develop a fully Fortran-based visualization tool and provide the source codes in order that the simulation researchers can customize them. SV4 displays magnetic field lines with their releasing points from which magnetic field lines are integrated. The user can control a releasing point by the mouse drag. The magnetic field line is calculated in real time and shown on the screen. SV4 can visualize scalar fields by the isosurface method. The isosurface level is interactively controlled by the mouse drag. One can rotate, translate, and zoom in/out the visualized 3-dimensional objects in the display window. Another visualization method for scalar data implemented in SV4 is the ortho-slicer in which distribution of the scalar is shown by color contours on x-y, y-z,

and z-x planes.

宇宙天気モデリング用球データ可視化ソフトウェア

吉田真人

要旨

現代文明を支える技術的社会基盤は宇宙環境に大きな影響を受けることが知られている。GPS (Global Positioning System)、衛星通信システム、大規模な送電網、電離層での反射を利用した長距離電波通信などは宇宙環境の激しい変化によって大きな障害を受け、時には機能停止を強いられることがある。そこで、宇宙環境の変化とその影響を予測する宇宙天気予報が近年さかんに研究されている。本研究では、宇宙天気予報プロジェクトの一つである宇宙天気モデリング共同プロジェクト (JPSWM) において出力されるシミュレーションデータを可視化するための可視化ソフトウェア “SV4” を開発した。JPSWM では Yin-Yang 格子と呼ばれる球ジオメトリでの数値計算に適した計算格子を採用している。この Yin-Yang 格子上で定義されたスカラー場及びベクトル場のデータを、Yin-Yang 格子の上で直接可視化処理し、表示することが SV4 の特徴である。また、Fortran 言語で書いた可視化プログラムから直接 OpenGL に基づく高速グラフィック処理を呼び出すことも SV4 の特徴である。SV4 は磁力線の表示、等値面、物理量分布の断面図表示など様々な可視化手法を実装している。これらの可視化手法は磁力線計算の初期点、等値面の値や断面図の位置などをマウスで操作でき、リアルタイムで再描画することができる。また可視化した CG オブジェクトは、マウスやキーボードを用いて任意の位置、角度から観察することができる。SV4 は様々な可視化手法と対話的可視化によりシミュレーションデータのより効率的な解析を可能にする。SV4 は汎用性を意識して開発した。ソースコードも公開する予定なので、Yin-Yang 格子を使った JPSWM 以外のシミュレーションデータの可視化にも活用されることが期待される。

目次

1	緒論	1
1.1	研究の背景	1
1.2	研究の目的	3
1.3	研究の概要	4
1.4	技術的背景	5
2	Yin-Yang 格子	7
3	磁力線の可視化	15
4	等値面の可視化	17
5	断面図可視化と太陽表面分布可視化	20
6	メニューシステム	22
7	典型的な可視化手続き	27
8	まとめ	31
9	謝辞	31

1 緒論

1.1 研究の背景

現代文明は地球規模に広がった技術基盤に支えられている。GPS (Global Positioning System)、衛星通信システム、長距離送電網、電離層での反射を利用した電波通信など、グローバルなスケールで情報やエネルギーを伝えるためのシステムを人類は構築してきた。我々の文明を支えるこのような技術基盤は実は地球外の環境 – 宇宙環境 – の安定性に強く依存し、宇宙環境の激しい環境の変化に対して脆い構造になっている。

宇宙環境とは直接には地球大気圏の外部、人工衛星の軌道付近を指すことが多いが、広い意味で使う時には、太陽から (地球を含めた) 惑星まで、太陽系スケールの環境を指す。地球から 1 億 5000 万 km も離れた太陽の状態が、地球の技術システムに直接的な影響を及ぼすということは、太陽がその光 (輻射) を通じて地球の自然環境を定めていることを考えれば、驚くべきことではないかもしれない。しかし、先に述べた様々な技術的社会基盤が太陽を起因する宇宙規模の変化に、場合によっては破壊されうるほどの大きな影響を受ける理由は、太陽の幅射の変化によるものではない。(太陽の輻射は比較的安定している)

太陽の表面で極めてダイナミックに変化しているものは太陽表面の磁場である。太陽フレアやプロミネンス、CME (Coronal Mass Ejection) 等として知られる太陽表面の爆発現象は現在ではそのエネルギー源が太陽磁場にあるということが分かっている。大きな太陽フレアは $10 \times 10^{22} \sim 10 \times 10^{25}$ ジュールのエネルギーを放出する極めて激しい爆発現象である。これは水素爆弾の数億個分にも相当する。

太陽表面で開放されたこのような爆発的エネルギーの一部は高エネルギー粒子となって地球を襲う。高エネルギーの粒子が半導体材料に照射されることで電子機器は誤作動や故障を起こす。電子機器と同様に高エネルギー粒子 (つまり放射線) は生物にも深刻な悪影響を及ぼす。宇宙ステーションに滞在中の宇宙飛行士は高エネルギー粒子の到達が予想される時には船外活動は避け、船内の放射線防御を施された部屋に居ることが望ましいとされる。電子機器も同様にあらかじめ高エネルギー粒子の到達が予測できれば通常の運転モードから緊急退避モードに切り替えることでハードウェアの防御や誤作動の防止が可能である。

太陽表面の磁気の急激な環境の変化あるいは爆発現象が地球の技術システムに悪影響を与えるルートはもう一つある。それは太陽フレア等の爆発現象により放出された高エネル

ギーのプラズマ流が強い磁場を伴って地球に到達するというルートである。太陽からは太陽風と呼ばれる水素プラズマの“風”が常に吹き出しており、その動圧のため地球磁場が変形され、地球磁気圏と呼ばれる構造を作っている（Fig. 1）。太陽の状態が変化するとこの太陽プラズマの流れや磁場等が急激に変化することになる。地球磁気圏の形状が急激に変化すること、地球周辺の磁場が変化すること、に他ならない。このような地磁気の大変化は磁気嵐と呼ばれる。

大きな磁気嵐が起きると地上の様々な電気システムに悪影響を与えることが知られている。例えば 1989 年にカナダで起きた磁気嵐では大規模な停電が発生した。これは磁気嵐に伴う磁場の変化が電磁誘導を引き起こしたためである。更に石油輸送用のパイプラインも磁気嵐に伴う電磁誘導により生じた電流のために損傷を受けた。また、太陽フレアなどは地球表面近くの電離層にも影響を与えることがある。デリンジャー現象と呼ばれる現象は太陽フレアから発生した X 線や紫外線が電離層の電子密度を増加させ、電離層で反射するはずの短波が吸収される現象でこれが起きると長距離通信に深刻な影響を与える。

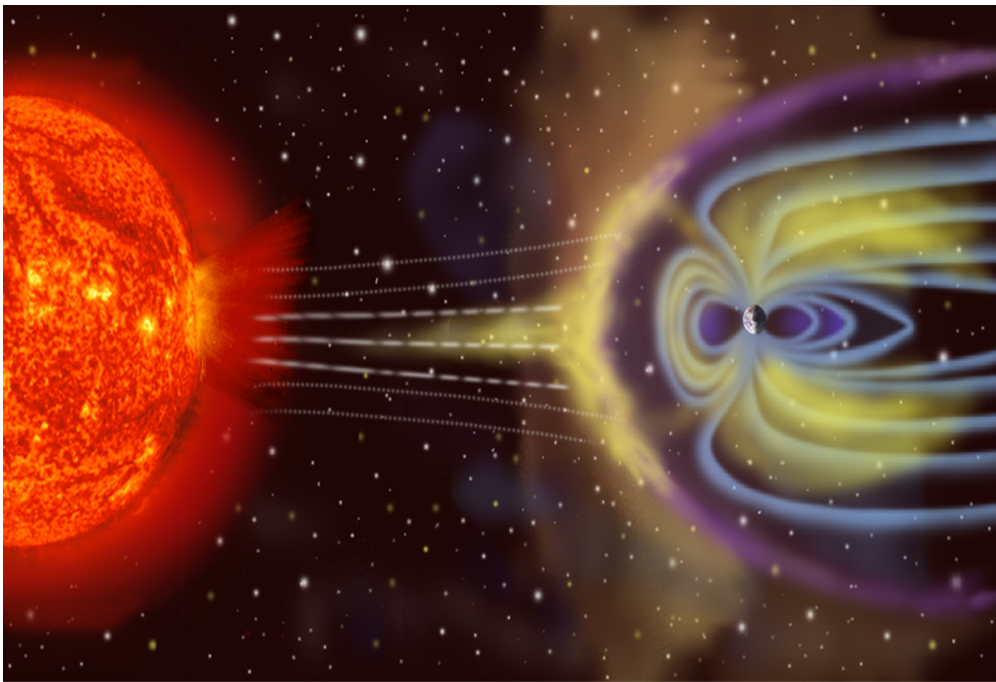


Fig. 1 Solar wind and earth's magnetosphere. [1]

このように、太陽表面で起こる現象は電磁的環境－宇宙環境－に大きな影響を与え、その宇宙環境の変化が我々の生活を支える様々技術的社会基盤に大きな影響、時には壊滅的なダメージを与える。したがって宇宙環境を予測することは天気予報と同様に重要な社

会的要請である。

宇宙天気予報を実現するためのプロジェクトが現在世界各地で進められている。その中で日本国内においては名古屋大学の草野完也教授を中心として宇宙天気モデリング共同プロジェクト (JPSWM) が進められている [2]。JPSWM プロジェクトは、観測衛星などの観測機器によって得られた太陽のデータに基づいて大規模シミュレーションを行うことにより宇宙天気予報の実現を目指すプロジェクトである。

JPSWM プロジェクトの特徴の一つは太陽コロナから、太陽風、地球磁気圏、電離層まで様々な領域を対象とした複数のシミュレーションコードを連結していることである。それぞれの領域のコードはそれぞれの研究チームが開発と計算実行をしており、地理的にも名古屋、横浜、京都、広島と分散している。

この種の大規模シミュレーションでは大量の数値データが生産される。そのデータを解析するために、数値データを画像化するいわゆる可視化のプロセスは必要不可欠である。しかし、この様に分散したプロジェクトでは、統一的な可視化ツールを共有することは難しい、実際、宇宙天気共同プロジェクトではそのような共通可視化ツールはこれまで存在しなかった。

JPSWM プロジェクトにおいて可視化ツールの共有が難しい理由は二つある。一つ目は、高価な可視化ソフトウェアをプロジェクトに参加する研究者全員に配布することはコスト的に難しいという点である。二つ目の理由は JPSWM プロジェクトでは太陽コロナ等のシミュレーションに Yin-Yang 格子と呼ばれる最近考案された計算格子用いられており、この Yin-Yang 格子上のシミュレーションデータを可視化できるツールが限られることである。

1.2 研究の目的

上述のように JPSWM プロジェクトに参加する研究者に全員に配布できる様に、フリーで提供できる可視化ツールが望まれている。

現在、一般的なシミュレーションデータの可視化を目的とした“可視化ソフト”と呼ばれる市販あるいは無償のソフトウェアが手に入る。例えば充実した可視化機能と洗練されたユーザーインターフェースを備えた可視化ソフトの一つとして、AVS/Express が挙げられる。しかし AVS/Express 等は、高価であるだけでなく、宇宙天気予報プロジェクトには不要な機能が多すぎるためかえって使いにくい。

Yin-Yang 格子上で定義されたデータを可視化できるソフトウェアとして、大野と陰山 [4] によって開発されたプログラムがある。しかしこのプログラムはレイキャスティング

[5] という計算コストのかかるレンダリング方式を採用しているので、対話的な可視化をすることができない。

JPSWM プロジェクト用の可視化ソフトウェアに求められる性質として、もう一つ、プロジェクトに参加している研究者がソースコードを自由に改変し、解析に活用できる様にする必要があるという点が挙げられる。JPSWM プロジェクトに参加している研究者の専門は太陽コロナ、太陽風、地球磁気圏、電離層など幅広いそのため、それぞれのデータ解析で必要となる可視化機能は、それを使う研究者が自分で調整するのが一番望ましい。

シミュレーション研究者が可視化ソフトウェアをゼロから書き上げることは通常期待できないが、出来上がった可視化ソフトのフレームワークにいくつか独自の可視化機能を追加したり、改良することは十分可能である。このとき、JPSWM プロジェクトに参加している研究者は Fortran 言語を“母語”としているので通常可視化ソフトウェアで用いられる C/C++ 言語にはなじみが少ないということを考慮に入れなければならない。

以上に述べた背景の下、本研究では Yin-Yang 格子上で定義された太陽コロナデータ [6] を対話的に可視化するプログラムを Fortran 言語で開発した。SV4 (Spherical data Visualizer in Fortran) と名付けたこのプログラムはドキュメントと共に既に JPSWM プロジェクトに提供した。

1.3 研究の概要

通常の可視化ソフトと同様に SV4 ではシミュレーションの数値データを OpenGL を用いて可視化する。OpenGL はコンピューターグラフィックス (CG) の業界標準 API である。OpenGL は 3 次元物体の生成と証明、カメラの設定というレンダリングの基本機能を提供する。しかし、ウインドウの制御やマウスを通じたユーザーインターフェース等は提供しない。この部分は OpenGL を補助するライブラリを利用する。そのような補助ライブラリとして良く用いられるものとして GLUT がある。本研究では OpenGL と GLUT の関数を用いる

しかし、OpenGL や GLUT は C 言語用の実装されているため使うことはできない。そこで本研究では Fortran 言語で SV4 を実装するため、CG 用 API として f90gl を採用した。f90gl は OpenGL、GLUT の関数を Fortran 言語用の実装した無償で提供されている API である。

Yin-Yang 格子上のシミュレーションデータを直接 f90gl で可視化することはできない。これは OpenGL、GLUT の関数はカーテシアン座標で入出力しているため、Yin-Yang 座標を使うことはできないためである。SV4 はカーテシアン座標を Yin-Yang 座標に座

標変換するアルゴリズムを用いることでこの問題を解決した。SV4 は f90gl と座標変換アルゴリズムを用いて、読み込んだ Yin-Yang 格子上のシミュレーションデータを直接可視化する。

JPSWM プロジェクトで必要とされる可視化機能を決めるにあたり、このプロジェクトに参加している研究者と検討した上で以下の可視化機能に焦点を当てることにした。

- ユーザーが指定する任意の初期点からの磁力線の計算と表示
- 等値面、断面図、太陽表面の物理量分布の可視化
- 対話的操作による可視化したオブジェクトの回転、移動、拡大縮小機能

磁力線は読み込んだ磁場データを元にユーザーが指定した点から数値積分を行うことで計算する。計算して求めた点列を CG 化することで磁力線の可視化を実装した。等値面はマーチングキューブアルゴリズムを実装した。断面図、太陽表面の物理量分布の可視化は面における物理量を色分布で表現する。この様に SV4 は JPSWM プロジェクトのシミュレーションデータを解析するために必要な複数の可視化手法を実装している。また、GLUT の関数を用いたグラフィカルユーザーインターフェースを備えているので、マウスやキーボードを用いて視点の移動や磁力線初期点の移動などを対話的に行える。SV4 は Yin-Yang 格子上のデータを直接可視化するだけでなく、様々な可視化手法や対話的操作により、ユーザーがシミュレーションデータを解析しやすい可視化を可能にする。

1.4 技術的背景

SV4 ではグラフィックス用 API として OpenGL と OpenGL Utility Toolkit (GLUT) を使用する。この節では OpenGL と GLUT について説明する。

OpenGL の描画プロセスではプリミティブと呼ばれる最小形状単位を用いる。プリミティブの形状とは点、線分、もしくはポリゴン（主に三角形）である。これらプリミティブは頂点座標を指定することで定義される。点であればその点の座標、線分であれば始点と終点の座標、三角形であれば三つの頂点の座標である。また、ポリゴンのそれぞれの頂点に対して、色と法線ベクトルを定義することができる。OpenGL は色や法線ベクトルの定義を元にポリゴンの色付けや陰影付けを行う。このポリゴンを組み合わせることで表示したい三次元構造物を作る。

次に視点と光源を設定する。視点の設定は視点の位置の座標、視線の目標位置の座標、視野の上方向のベクトルを定めることで完了する。光源の設定は光源の位置と光源の色を定めることで完了する。三次元構造物、視点、光源の情報が揃ったので、指定した視点か

ら三次元構造物を含む空間を見た視野と風景が決まる。次に射影平面を設定し、指定した視点と視野に基づいてその平面に映像を射影する。射影の際、視点から同じ方向に複数の物体が重なって見える位置にある場合には、OpenGL が奥行きの自動的に判定し、視点に最も近いポリゴンだけを射影平面に投影する。奥行き距離の判定には z バッファと呼ばれるハードウェアを利用することで高速処理される。

以上に述べた OpenGL のプロセスはレンダリングと呼ばれる。レンダリングの処理をまとめると、OpenGL は仮想空間にポリゴンなどのプリミティブ形状で 3 次元物体を形成し、指定した視点と照明の下でその空間を見た風景を平面に投影することで 2 次元画像を作成する。OpenGL は物理的な位置を指定するだけで画像化されるので、可視化プログラムと相性が良い。また、ハードウェアに近い低次元ライブラリなので、柔軟なプログラミングが可能である。

OpenGL 以外の CG 用 API として DirectX がある。DirectX は主に Windows 系 OS でしか使えない。それに対して、OpenGL はハードウェアに依存しないように設計されている。OpenGL は UNIX ワークステーション、Windows、MacOSX など広く使えるクロスプラットフォームの API である。この点において、JPSWM プロジェクトの研究者に提供する SV4 には OpenGL が適している。

OpenGL はレンダリングコマンドがハードウェアに依存しないように設計されている。そのため OpenGL は特定のウインドウシステムやオペレーティングシステムに対するコマンドを備えていない。つまり OpenGL はウインドウマネージメントができない設計になっている。対話的な可視化プログラムを実現するにはマウスイベントやキーボードイベントの取得も不可欠である。

SV4 はウインドウシステムやインターフェース操作を実装するために GLUT(OpenGL Utility Toolkit) を用いた。GLUT はハードウェアに依存しないレンダリングライブラリである OpenGL に、ウインドウシステム関数やグラフィックユーザーインターフェース (GUI) 関数を拡張したライブラリである。

SV4 で主に用いた GLUT の機能を以下に挙げる。

- ディスプレイ上で CG 表示用のウインドウを作成
- マウスのクリック、マウスカーソルの座標などのマウスの入力やキーボードの入力を取得
- 入力をきっかけとしてウインドウ内の CG を更新する (アニメーション)
- 右クリックによる GUI 機能を備えたメニューの表示

OpenGL と GLUT はもともと C 言語系の言語用に開発されているので Fortran 言語

では使えない。そこで SV4 では f90gl[7] を採用した。f90gl は OpenGL と GLUT の機能を Fortran 言語の API として無償で提供されている。Fortran 言語で OpenGL を使う他の方法として Visual Fortran コンパイラを使う方法があるが、開発環境やコンパイラに制限を受けるので SV4 の汎用性を減ずることになるので採用しなかった。一方、f90gl は UNIX ワークステーション、Windows、MacOSX などでも広く使用することができるので元の OpenGL と GLUT 同じ様に汎用性を保つことができる。

f90gl の関数の機能は元の OpenGL や GLUT の関数と全く同じである。またポインタなどの概念が違ふ Fortran 言語においても、f90gl の関数の引数は OpenGL や GLUT の関数の引数とほぼ同じになる様に工夫がされている。この f90gl を用いることで C 言語で OpenGL や GLUT の関数を使用する方法と同じ様に Fortran90 言語で OpenGL や GLUT の関数を使用できる。SV4 で使用している OpenGL と GLUT の機能はすべて f90gl の関数を用いて実装している。

2 Yin-Yang 格子

SV4 は Yin-Yang 格子上で定義されたデータを可視化する。そこで、この章では Yin-Yang 格子について解説する。

コンピューターシミュレーションでは解くべき基礎方程式（太陽コロナシミュレーションの場合は MHD 方程式）を離散化する。離散化の方法は有限差分法、有限体積法、有限要素法、境界要素法、スペクトル法等、数多く存在する。その中で、JPSWM プロジェクトでは有限体積法または有限差分法を採用しているコードが多い。有限差分法とは、微分方程式で定式化された問題を計算機で解くにあたり、微分を差分で近似する方法である。そのために空間を順序付けされた格子点で離散化する。格子点全体を計算格子と呼ぶ。計算格子のデザインはシミュレーションの精度と速度を決める重要な要因である。計算結果としての数値データは格子点上で値を持つ。

球状のジオメトリの中での現象を対象とした計算機シミュレーションでは、球座標に基づいて空間を離散化した球座標格子（緯度経度格子）がしばしば用いられているが、この球座標格子には二つの問題がある。

一つ目は余緯度が 0 度もしくは 180 度の点（即ち北極点と南極点）が座標特異点になっているという問題である。二つ目は北極点と南極点近傍で格子点の分布が密になっているという問題である（Fig. 2）。一つ目の問題はロピタルの定理を用いることで解消できる。一方、二つ目の問題を解決するのはそれほど簡単ではない。格子間隔が密になることで積分タイムステップに大きな制約を受ける。その解決法の一つとしてローパスフィルターを

かける方法がある。しかしその方法を用いると計算コストもかかり、また並列化の効率を下げるという問題がある。極座標格子の格子間隔が極近くで密なる問題点を解決するために様々な球ジオメトリ用の格子系が考案された。その内の一つが Yin-Yang 格子である。次に Yin-Yang 格子構成法について述べる。

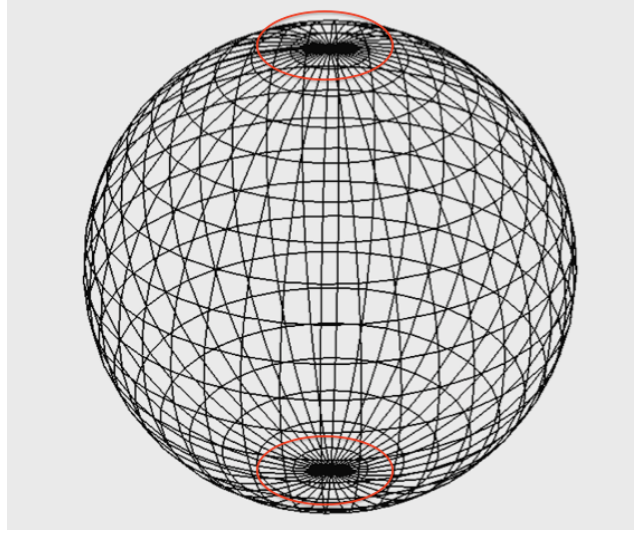


Fig. 2 Numerical difficulties of the spherical polar coordinate.

極座標格子から余緯度 θ が $\frac{\pi}{4} \leq \theta \leq \frac{3\pi}{4}$, 且つ経度 ϕ が $-\frac{3\pi}{4} \leq \phi \leq \frac{3\pi}{4}$ の格子部分を抜き出す。この部分ではで格子間隔の分布がほぼ均一となる。そしてその格子部分と合同な格子もう一つ用意する。この二つの部分格子において、極座標と同じ θ 、 ϕ の方向性を持っている格子を Yin 格子と呼び、もう一方を Yang 格子と呼ぶ (Fig. 3)。これら二つの格子を相補的に組み合わせ、球全体を覆う格子が Yin-Yang 格子である (Fig. 4)。

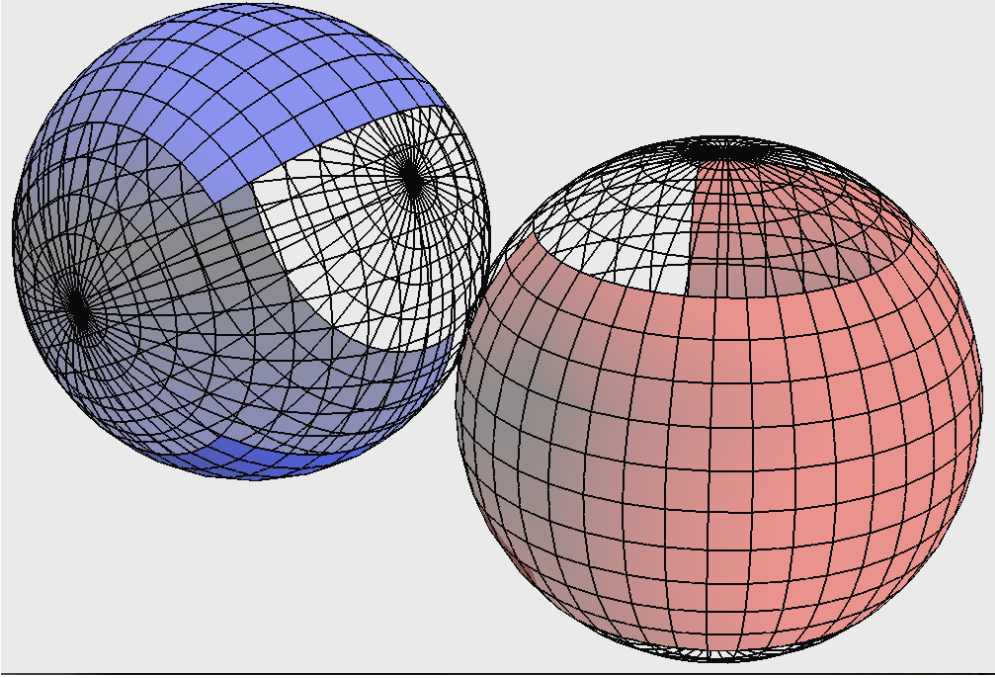


Fig. 3 “Yin” and “Yang” components. These components are congruent extractions from $\frac{\pi}{4} \leq \theta \leq \frac{3\pi}{4}$ and $-\frac{3\pi}{4} \leq \phi \leq \frac{3\pi}{4}$ in the spherical polar coordinates. The red component which has the same directions as the spherical polar coordinate system is called “Yin”. The blue component is called “Yang”.

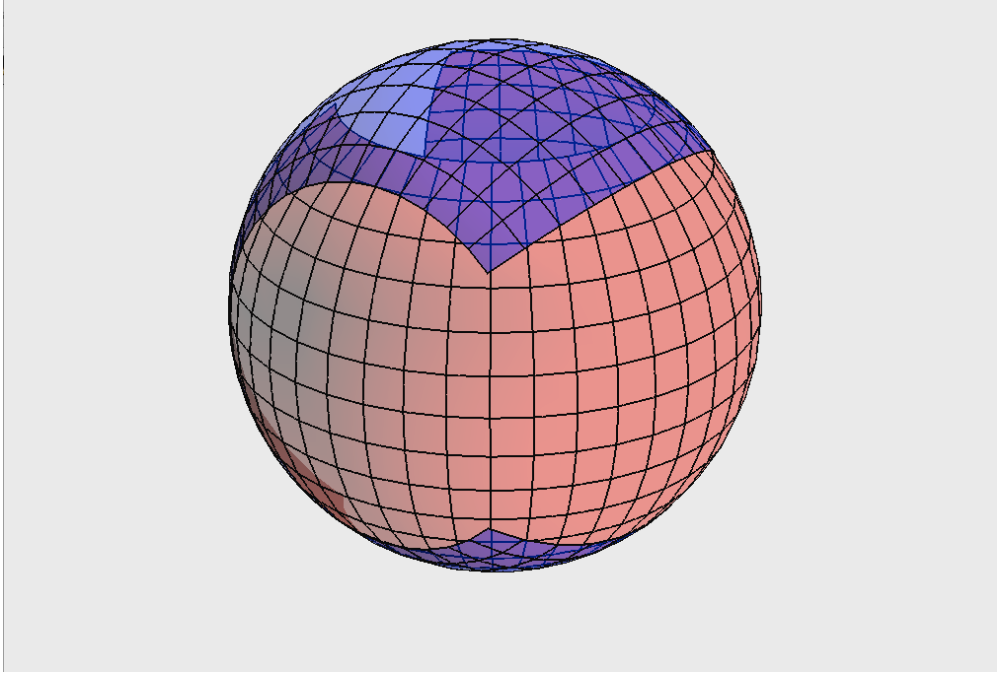


Fig. 4 Yin-Yang grid. The “Yin” and “Yang” components are combined in a complementary way to cover the spherical surface.

次に SV4 が可視化する Yin-Yang 格子上データの構造とその利用法を説明する。

SV4 では Yin-Yang 格子上のシミュレーションによる出力データを 4 次元配列に格納し、その配列を用いて可視化処理をする。その 4 次元配列の構造は $(N_r, N_\theta, N_\phi, 2)$ になっている。ここで N_r 、 N_θ 、 N_ϕ はそれぞれ半径 r 、余緯度 θ 、経度 ϕ 方向の格子点数である。空間全体の格子点数は Yin-Yang 格子では Yin 格子と Yang 格子があるので $N_r * N_\theta * N_\phi * 2$ となる。シミュレーションにおいて、格子点数を多くとればとるほど格子の目は細くなるのでより精密なシミュレーションが可能になるが、逆に計算コストは大きくなり、出力されるデータサイズも大きくなる。よってシミュレーションによって N_r 、 N_θ 、 N_ϕ は変わるので、可視化するシミュレーションに合わせた 4 次元配列を用意しなければならない。SV4 はコンフィギュレーションファイルを編集することにより、 N_r 、 N_θ 、 N_ϕ を可視化するシミュレーションに合わせた設定にする。次にその 4 次元配列の構造について説明する。上記の N_r 、 N_θ 、 N_ϕ を用いて、半径 r 、余緯度 θ 、経度 ϕ のそれぞれ格子間隔 Δr 、 $\Delta \theta$ 、 $\Delta \phi$ は以下の式で定義できる。

$$\Delta r = \frac{r_{max} - r_{min}}{N_r - 1} \quad (1)$$

$$\Delta\theta = \frac{\theta_{max} - \theta_{min}}{N_\theta - 1} = \frac{\frac{3\pi}{4} - \frac{\pi}{4}}{N_\theta - 1} = \frac{\frac{\pi}{2}}{N_\theta - 1} \quad (2)$$

$$\Delta\phi = \frac{\phi_{max} - \phi_{min}}{N_\phi - 1} = \frac{\frac{3\pi}{4} - (-\frac{3\pi}{4})}{N_\phi - 1} = \frac{\frac{3\pi}{2}}{N_\phi - 1} \quad (3)$$

求めた格子間隔 Δr 、 $\Delta\theta$ 、 $\Delta\phi$ を用いて、部分極座標格子の格子点 r_i 、 θ_j 、 ϕ_k を以下の式で定義できる。

$$r_i = r_{min} + \Delta r * i \quad (0 \leq i < N_r) \quad (4)$$

$$\theta_j = \theta_{min} + \Delta\theta * j \quad (0 \leq j < N_\theta) \quad (5)$$

$$\phi_k = \phi_{min} + \Delta\phi * k \quad (0 \leq k < N_\phi) \quad (6)$$

次に以下のような Yin 格子と Yang 格子を識別する識別子 l を定義する。

$$l = \begin{cases} 0 & (\text{yin 領域のとき}) \\ 1 & (\text{yang 領域のとき}) \end{cases} \quad (7)$$

これら r_i 、 θ_j 、 ϕ_k 、 l を使うことで、格子点 (i, j, k, l) は識別子 l の領域における半径 r_i 、余緯度 θ_j 、経度 ϕ_k の格子点を一意に求めることができる。このようにして Yin-Yang 格子のすべての格子点を定義することができ、4次元配列と格子点を対応付けることができる。SV4 では Yin-Yang 格子上のシミュレーションで得られた各格子点上の様々な物理量をそれぞれの物理量ごとに四次元配列として読み込み可視化する。具体例を出すと、配列 pr の配列要素 $pr(i, j, k, l)$ は格子点 (i, j, k, l) における圧力の値を保持している。

しかし、データ読み込んだ四次元配列を用いて直接可視化処理することはできない。なぜなら SV4 は可視化処理のために Fortran 言語用の OpenGL 及び GLUT である f90gl を用いており、f90gl の API 関数群はカーテシアン座標により入出力を行っているので Yin-Yang 座標系のデータを直接参照して可視化処理することはできないからである。よって Yin-Yang 格子上のシミュレーションデータを可視化するためにカーテシアン座標から Yin-Yang 座標への座標変換をする必要がある。一方、Yin-Yang 座標からカーテシアン座標への座標変換は必要としない。これは f90gl の関数にとってカーテシアン座標で位置指定し、その位置の物理量が分かれば可視化処理が可能だからである。つまり SV4 はカーテシアン座標で指定された位置を Yin-Yang 座標へ座標変換することで Yin-Yang 座標格子上データへの参照を可能にしている。カーテシアン座標から Yin-Yang 座標への座標変換を以下に説明する。

カーテシアン座標 (x, y, z) から極座標 (r, θ, ϕ) への座標変換は以下の式で変換できる。

$$r = \sqrt{x^2 + y^2 + z^2} \quad (8)$$

$$\theta^i = \tan^{-1} \frac{\sqrt{x^2 + y^2}}{z} \quad (9)$$

$$\phi^i = \tan^{-1} \frac{y}{x} \quad (10)$$

カーテシアン座標で指定した点 (x, y, z) を上記の変換式を用いて変換した θ と ϕ が $\frac{\pi}{4} \leq \theta \leq \frac{3\pi}{4}$ 且つ $-\frac{3\pi}{4} \leq \phi \leq \frac{3\pi}{4}$ の範囲にある場合、その点は極座標と同じ方向性を持つ Yin 格子領域にあることを意味するので、求めた r 、 θ 、 ϕ と識別子 $l = 0$ を用いて Yin-Yang 座標 $(r, \theta, \phi, 0)$ に変換できる。一方範囲外の場合、そのまま求めた θ 、 ϕ を Yin-Yang 座標に組み込むことはできない。それは Yang 格子と極座標系の方向性が違うからである。よって Yang 格子における θ 、 ϕ を Yin 格子と Yang 格子の関係性を用いて求める。Yin 格子における x 、 y 、 z を x^i 、 y^i 、 z^i とし、Yang 格子における x 、 y 、 z を x^a 、 y^a 、 z^a とした時、可 Yin 格子と Yang 格子の x 、 y 、 z の方向は本研究では Fig. 5 の様に定義した。なお、この定義はオリジナルの Yin-Yang 格子の定義とは符号のとり方が違うが、本質的には同じものである。

$$\begin{bmatrix} x^i \\ y^i \\ z^i \end{bmatrix} = \begin{bmatrix} -x^a \\ -y^a \\ -z^a \end{bmatrix} \quad (11)$$

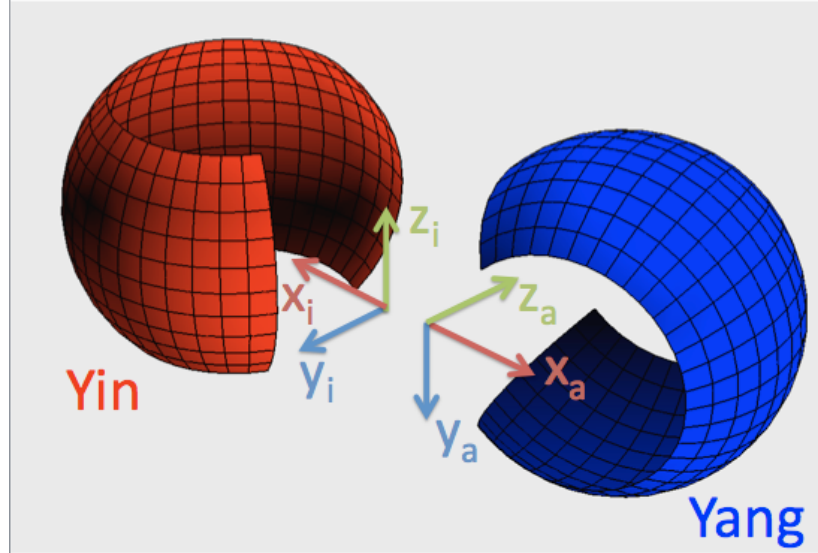


Fig. 5 The relation between the coordinate axes of “Yin” and “Yang”.

カーテシアン座標から極座標に変換する関数を $TF(x, y, z)$ とすると、上記の関係より引数を入れ替えた $TF(-x, -z, -y)$ により求まる r 、 θ 、 ϕ が Yang 格子領域における r 、

θ 、 ϕ となる。求めた r 、 θ 、 ϕ と $l = 1$ を用いて Yin-Yang 座標 $(r, \theta, \phi, 1)$ に変換できる。以上の様に場合分け行うことで、SV4 はカーテシアン座標から Yin-Yang 座標に座標変換する。

次に変換した Yin-Yang 座標を元に指定した位置を囲む 8 個の格子点を見つける。シミュレーションデータは格子点上に物理量を定義しているので、その 8 個の格子点上の物理量を線形補間することで物理量を求める。指定した位置の Yin-Yang 座標が (r, θ, ϕ, l) とすると、以下の式を満たす (i, j, k) が定まる。

$$r_i \leq r < r_{i+1} \quad , \quad \theta_i \leq \theta < \theta_{i+1} \quad , \quad \phi_i \leq \phi < \phi_{i+1} \quad (12)$$

求める物理量を Q とすると、上記の (i, j, k) と Yin-Yang 座標の l を用いて、指定した位置を囲む八つの点の Q は $Q(i, j, k, l)$ 、 $Q(i+1, j, k, l)$ 、 $Q(i, j+1, k, l)$ 、 $Q(i+1, j+1, k, l)$ 、 $Q(i, j, k+1, l)$ 、 $Q(i+1, j, k+1, l)$ 、 $Q(i, j+1, k+1, l)$ 、 $Q(i+1, j+1, k+1, l)$ と表せる。これら八つの Q を指定した位置に近い格子点の Q ほど重みが大い様に重み付けする。SV4 は以下の三次元線形補間を採用した。

$$f_{r1} = \frac{r_{i+1} - r}{r_{i+1} - r_i} \quad , \quad f_{r2} = 1 - f_{r1} \quad (13)$$

$$f_{\theta1} = \frac{\theta_{i+1} - \theta}{\theta_{i+1} - \theta_i} \quad , \quad f_{\theta2} = 1 - f_{\theta1} \quad (14)$$

$$f_{\phi1} = \frac{\phi_{i+1} - \phi}{\phi_{i+1} - \phi_i} \quad , \quad f_{\phi2} = 1 - f_{\phi1} \quad (15)$$

$$w_1 = f_{r1} * f_{\theta1} * f_{\phi1} \quad , \quad w_2 = f_{r2} * f_{\theta1} * f_{\phi1} \quad (16)$$

$$w_3 = f_{r1} * f_{\theta2} * f_{\phi1} \quad , \quad w_4 = f_{r2} * f_{\theta2} * f_{\phi1} \quad (17)$$

$$w_5 = f_{r1} * f_{\theta1} * f_{\phi2} \quad , \quad w_6 = f_{r2} * f_{\theta1} * f_{\phi2} \quad (18)$$

$$w_7 = f_{r1} * f_{\theta2} * f_{\phi2} \quad , \quad w_8 = f_{r2} * f_{\theta2} * f_{\phi2} \quad (19)$$

$$\begin{aligned} Q = & w_1 * Q(i, j, k, l) + w_2 * Q(i+1, j, k, l) + w_3 * Q(i, j+1, k, l) \\ & + w_4 * Q(i+1, j+1, k, l) + w_5 * Q(i, j, k+1, l) + w_6 * Q(i+1, j, k+1, l) \\ & + w_7 * Q(i, j+1, k+1, l) + w_8 * Q(i+1, j+1, k+1, l) \end{aligned} \quad (20)$$

以上の線形補間により指定した位置の物理量を求める。

まとめると、SV4 でカーテシアン座標で指定した位置を Yin-Yang 座標に変換し、読み込んだデータを参照、補間することにより物理量を求める。

3 磁力線の可視化

太陽コロナシミュレーションの解析において磁力線の可視化は本質的に重要である。この章では SV4 における磁力線可視化の実装について説明する。

磁力線は以下の式 (21) で定義される。

$$\frac{d\mathbf{x}(s)}{ds} = \frac{\mathbf{B}(\mathbf{x}(s))}{\|\mathbf{B}\|} \quad (21)$$

ここで $\mathbf{x}(s)$ は磁力線に沿った長さ s における空間中の位置ベクトルを表す。 $\mathbf{B}(\mathbf{x}(s))$ は位置 $\mathbf{x}(s)$ における磁場である。 $\|\mathbf{B}\|$ は磁場ベクトルの絶対値を表す。SV4 はユーザーが指定した初期点からルンゲ＝クッタ法で式 (21) を正方向と負方向の両方に数値積分することで磁力線の点列を求める。設定した積分ステップ数を超えるか、磁力線がシミュレーション空間から外れることで積分を終了する。この積分により、負方向の積分が終了した時の点を始点とし、正方向の積分が終了した時の点を終点とする点列が求まる。この点列は指定した初期点を貫く磁力線の点列を意味する。

求めた点列を磁力線として表示する方法として、点列を線で結んで磁力線を表示することも考えられるが、三次元空間上に線で磁力線を描くと磁力線の構造が分かりにくい画になる。よって、SV4 では求めた点列を軸とした円柱を繋げることにより、チューブ状の磁力線を描くことで可視化する。そのアルゴリズムを以下に説明する。

求めた点列の始点から次の点に向かうベクトルを $\mathbf{V}$ とし、 $\mathbf{V}$ を中心軸とした円柱を描く。まず $\mathbf{V}$ とは異なる任意のベクトルをとり、そのベクトルと $\mathbf{V}$ との外積を計算し、その方向へ向かう単位ベクトルを $\mathbf{U}$ とする。更に $\mathbf{V}$ と $\mathbf{U}$ の外積単位ベクトルを $\mathbf{W}$ とする。 $\mathbf{V}$ と $\mathbf{U}$ と $\mathbf{W}$ は互いに直交する。ここで初期点の位置ベクトルを $\mathbf{P}_1$ 、次の点の位置ベクトルを $\mathbf{P}_2$ としたとき

$$\mathbf{A}_1(\theta) = \mathbf{P}_1 + r(\sin \theta \mathbf{W} + \cos \theta \mathbf{U}) \quad (0 \leq \theta < 2\pi) \quad (22)$$

$$\mathbf{A}_2(\theta) = \mathbf{P}_2 + r(\sin \theta \mathbf{W} + \cos \theta \mathbf{U}) \quad (0 \leq \theta < 2\pi) \quad (23)$$

で決まる $\mathbf{A}_1(\theta)$ と $\mathbf{A}_2(\theta)$ は $\mathbf{V}$ に垂直な面上で $\mathbf{P}_1$ と $\mathbf{P}_2$ を中心とした半径 r の円周上の点である。 θ の値により円周上の任意の点をとることができる。 $\mathbf{P}_1(\theta)$ の点と $\mathbf{P}_2(\theta)$ においてそれぞれ 6 点 ($\theta = 0, \frac{\pi}{3}, \frac{2\pi}{3}, \pi, \frac{4\pi}{3}, \frac{5\pi}{3}$) の座標とその法線ベクトルである ($\sin \theta \mathbf{W} + \cos \theta \mathbf{U} : \theta = 0, \frac{\pi}{3}, \frac{2\pi}{3}, \pi, \frac{4\pi}{3}, \frac{5\pi}{3}$) を f90gl の関数に引数として渡す。その関数にそれらの点に結ぶ面を表示させることにより $\mathbf{V}$ を中心軸とした円柱が描くことができる (Fig. 6)。

P_2 の次の点を P_3 としたとき、 P_2 と P_3 において上記と同じ導出で $A_3(\theta)$ を求める。その $A_3(\theta)$ と前回の $A_2(\theta)$ を結ぶ面を表示することでその次の円柱を作る。これを点列の終点まで繰り返すことにより磁力線を表示する。ただし終点は円柱ではなく円錐にすることで磁力線の向きを示す。SV4 の磁力線可視化の様子を Fig. 7 に示す。

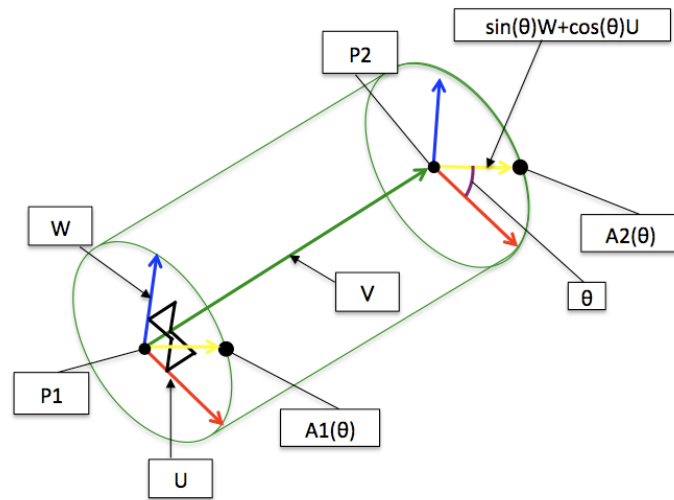


Fig. 6 Tube construction for a magnetic field line. $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{U}$ vectors are perpendicular to each other

SV4 では磁力線の初期点をファイルから読み込む他に、可視化中に初期点を示すボールのオブジェクトをドラッグすることで初期点を対話的に動かすことができるように設計した。初期点を動かすと同時に動かした初期点を貫く磁力線を再計算し、リアルタイムにその磁力線を表示する (Fig. 14)。このようにシミュレーションデータにおける特徴的な磁力線のを対話的に探せる機能が SV4 の特徴である。

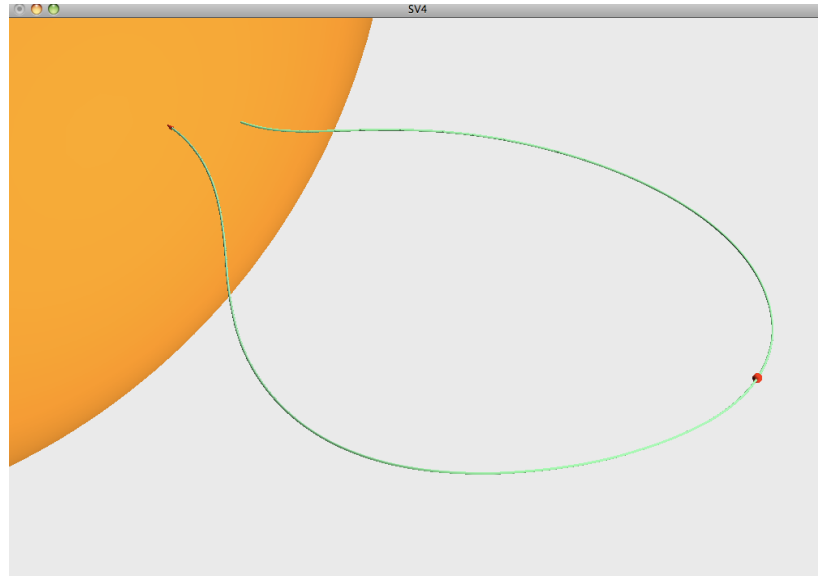


Fig. 7 An example of magnetic field line visualized by SV4. The red ball indicates the releasing point of the field line. The field line is integrated in both the positive and negative directions from the releasing point. Calculated magnetic field line is shown by a tube. A corn is placed at the end of the magnetic field line to indicate its direction.

4 等値面の可視化

この章ではスカラーデータの可視化手法として最もよく用いられる手法の一つである等値面について説明する。

等値面とは三次元空間の物理量分布において同じ値の点の集合を面で表示したものである。これは二次元における等高線を三次元へ拡張した可視化手法である。SV4 はユーザーが指定した物理量の等値面を表示し、等値面の値を対話的に指定することが可能である。指定された値の等値面はリアルタイムで構成され、表示される。等値面を構成する方法としては古典的なマーチングキューブアルゴリズム [8] を採用した。ここでマーチングキューブアルゴリズムの概要を説明する。計算格子中の一つボクセルに注目した時、ボクセルの八つの頂点それぞれ対して等値面の値より大きい小さいかを調べることで $2^8 = 256$ 通りに場合分けできる。また一つのボクセルを横切る等値面のポリゴンパターンは 14 通りある (Fig. 8)。次に 256 通りのボクセルパターン対して 14 通りのポリゴンパターンを対応させた表を作る。この表を用いてボクセルの中に対応するポリゴンを描く、このルーチンを計算格子中の全てのボクセルに対して行うことで等値面を描く。SV4

の等値面可視化の様子を Fig. 9 に示す。

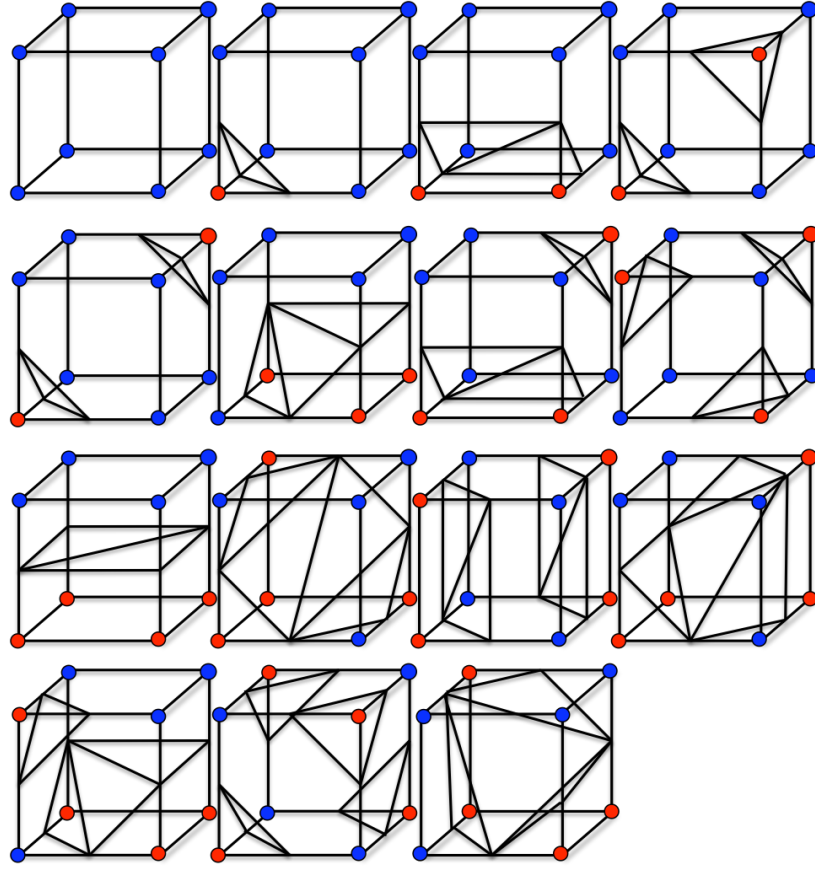


Fig. 8 Intersection patterns in the marching cubes algorithm. 14 intersection patterns of elemental isosurfaces in a voxel are shown. The red vertices have values higher than the target level of the isosurface. The blue ones have lower values.

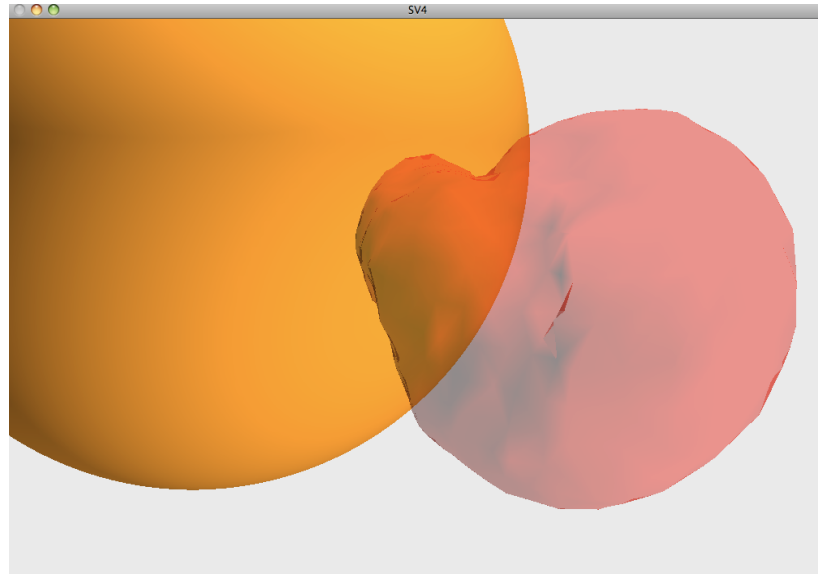


Fig. 9 An example of isosurface visualization by SV4. An isosurface of magnetic field intensity is shown. The level of the isosurface can be controlled by the mouse drag.

5 断面図可視化と太陽表面分布可視化

SV4 には前章で説明した磁力線の可視化と等値面の可視化機能に加えて座標軸に垂直な平面上のスカラー分布と太陽面上のスカラー分布を可視化する機能も実装した。Fig. 10 は z 軸に垂直な平面上の圧力スカラー場を可視化した例を示す。 xy 平面の z 座標、 yz 平面の x 座標、 xz 平面の y 座標はそれぞれマウスまたは矢印キーにより対話的に動かすことができる (Fig. 16)。球面上のスカラー分布も同様に表示される。この機能は太陽表面上の磁場分布を可視化する時に有効な機能である (Fig. 11)。

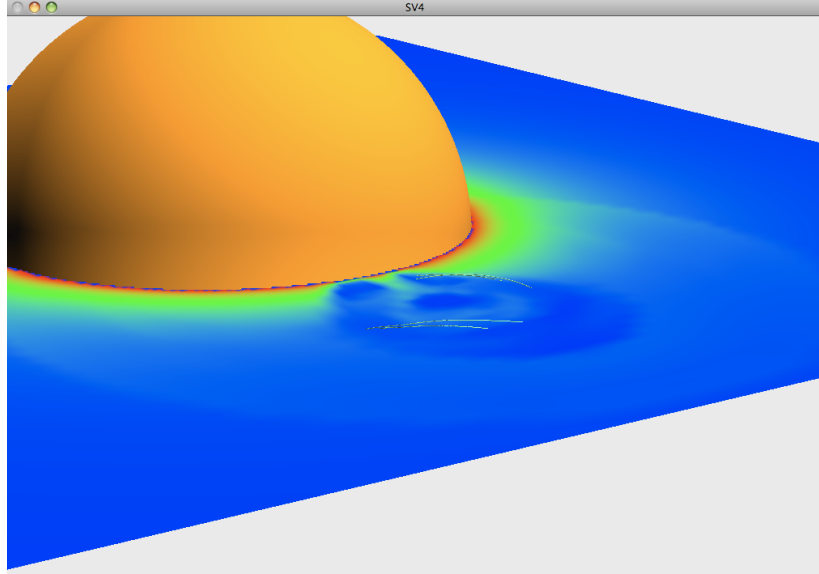


Fig. 10 Pressure distribution on the equatorial (x-y) plane. The z position of the plane can be controlled by the mouse drag. The color field is from blue to red. A shock is observed.

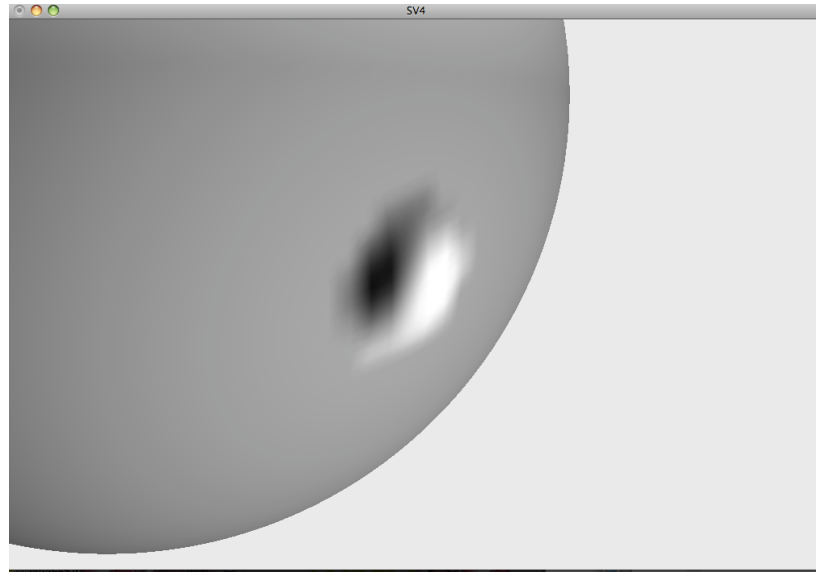


Fig. 11 Distribution of the radial component of the magnetic field on the solar surface. The polarity and intensity of the magnetic field is visualized by the gray scale.

6 メニューシステム

SV4 のウインドウ上で右クリックするとメニューが表示される (Fig. 12)。このメニューから SV4 の可視化設定やマウスや矢印キーの役割設定は行うことができる。マウスや矢印キーの役割設定はマウスもしくは矢印キーを使ってどのような操作をするかを設定する。特によく使う機能は、マウスにより可視化した CG オブジェクトの平行移動、回転移動、拡大縮小を行う機能である。この機能はメニューにより平行移動、回転移動、拡大縮小をマウスの左ボタン、中ボタン、矢印キーに自由に割り振ることができる。オブジェクトの平行移動、回転移動、拡大縮小以外に、マウスに値面の値の増減、 xy 、 yz 、 xz 断面図の移動機能を割り振ることもできる。

また、メニュー項目にはないマウスの機能として、3 章で述べたように、磁力線の初期点を表すボールのオブジェクトをドラッグすることにより初期点を対話的に変更する機能を備えている。この磁力線初期点の移動はマウスに割り振られた役割に関わらず、初期点をドラッグすることにより機能する。初期点を動かして見つけた磁力線の構造をメニュー項目の “save bline” により可視化空間上に保存することができる。

この様にメニューにより、SV4 の様々な機能を活用することができる。以下にメニュー

項目の詳細を示す。

- left mouse button : 左ボタンの役割を決定する。
 - rotate: マウスの左ボタンを押しながら左右に動かすことで z 軸を中心に回転、上下に動かすことで x 軸を中心に回転
 - pan: マウスの左ボタンを押しながら左右に動かすことで左右に視点平行移動、上下に動かすことで上下に視点平行移動
 - zoom: マウスの左ボタンを押しながら上に動かすことで拡大、下に動かすことで縮小
 - move xyslice: マウスの左ボタンを押しながら上に動かすことで xy 断面図を z 軸の正方向に、下に動かすことで負方向に動かす。
 - move yzslice: マウスの左ボタンを押しながら上に動かすことで yz 断面図を x 軸の正方向に、下に動かすことで負方向に動かす。
 - move zxslice: マウスの左ボタンを押しながら上に動かすことで xz 断面図を y 軸の正方向に、下に動かすことで負方向に動かす。
 - change isos: マウスの左ボタンを押しながら上に動かすことで等値面の値を大きくする、下に動かすことで等値面の値を小さくする。
- middle mouse button : 中ボタンの役割を決定する。
 - rotate: マウスの中ボタンを押しながら左右に動かすことで z 軸を中心に回転、上下に動かすことで x 軸を中心に回転
 - pan: マウスの中ボタンを押しながら左右に動かすことで左右に視点平行移動、上下に動かすことで上下に視点平行移動
 - zoom: マウスの左ボタンを押しながら上に動かすことで拡大、下に動かすことで縮小
 - move xyslice: マウスの中ボタンを押しながら上に動かすことで xy 断面図を z 軸の正方向に、下に動かすことで負方向に動かす。
 - move yzslice: マウスの中ボタンを押しながら上に動かすことで yz 断面図を x 軸の正方向に、下に動かすことで負方向に動かす。
 - move zxslice: マウスの中ボタンを押しながら上に動かすことで xz 断面図を y 軸の正方向に、下に動かすことで負方向に動かす。
 - change isos: マウスの中ボタンを押しながら上に動かすことで等値面の値を大きくする、下に動かすことで等値面の値を小さくする。
- arrow key: 矢印キーの役割を決定する。

- rotate: 矢印キーの左右キーで z 軸を中心に回転、上下キーで x 軸を中心に回転
- pan: 矢印キーの左右キーで左右に視点平行移動、上下キーで x 軸を中心に回転
- zoom: 矢印キーの上キーで拡大、下キーで縮小
- select bline: 矢印キーの上下で静的な磁力線を選択する（選択された磁力線は赤くなる）。矢印キーの左右で選択解除する。磁力線を選択している（磁力線が赤くなっている）状態でキーボードの D ボタンを押すと選択している磁力線を削除する。
- set value: 断面図や太陽表面、等値面に何を表示させるかを設定する。
 - xy slice: xy 断面図で表示する物理量を指定する。
 - yz slice: yz 断面図で表示する物理量を指定する。
 - xz slice: xz 断面図で表示する物理量を指定する。
 - solar: 太陽表面で表示する物理量を指定する。
 - isos: 等値面で表示する物理量を指定する。
 - color field: 断面図や太陽表面の分布色を選ぶ。black to white を選ぶと値が低い所から高くなるにつれて黒→白となる、color field を選ぶと青→緑→赤となる。
- on-off: それぞれのオブジェクトを表示するかどうかを設定する
 - solar face on: 太陽の表示をオンにする。
 - solar face off: 太陽の表示をオフにする。
 - isos on: 等値面の表示をオンにする。
 - isos off: 等値面の表示をオフにする。
 - xy slice on: xy 断面図の表示をオンにする。
 - xy slice off: xy 断面図の表示をオフにする。
 - yz slice on: yz 断面図の表示をオンにする。
 - yz slice off: yz 断面図の表示をオフにする。
 - xz slice on: xz 断面図の表示をオンにする。
 - xz slice off: xz 断面図の表示をオフにする。
 - all slice off: すべての断面図の表示をオフにする。
- save bline: このメニューを選んだ時の動的な磁力線（球を貫いている磁力線）の構造を静的な磁力線に追加する。
- next step: 時間ステップごとに複数のシミュレーションデータファイルを読み込

んでいる場合は、場のデータを次のステップに更新し、更に磁力線や等値面などすべてオブジェクトを再計算し更新する。最後のステップで next step を行うと最初のステップに戻る。

- reset to initial view: デフォルトの視点に戻る。
- reset to initial object: オブジェクトの設定をデフォルトに戻る。
- quit(save): 磁力線の初期点オブジェクトの位置、オブジェクトのタイプを外部ファイル保存し SV4 を終了する。
- quit: SV4 を終了する。ただし SV 4 は視点情報を変更するたびに外部ファイルに保存しているので、視点情報は保存される。

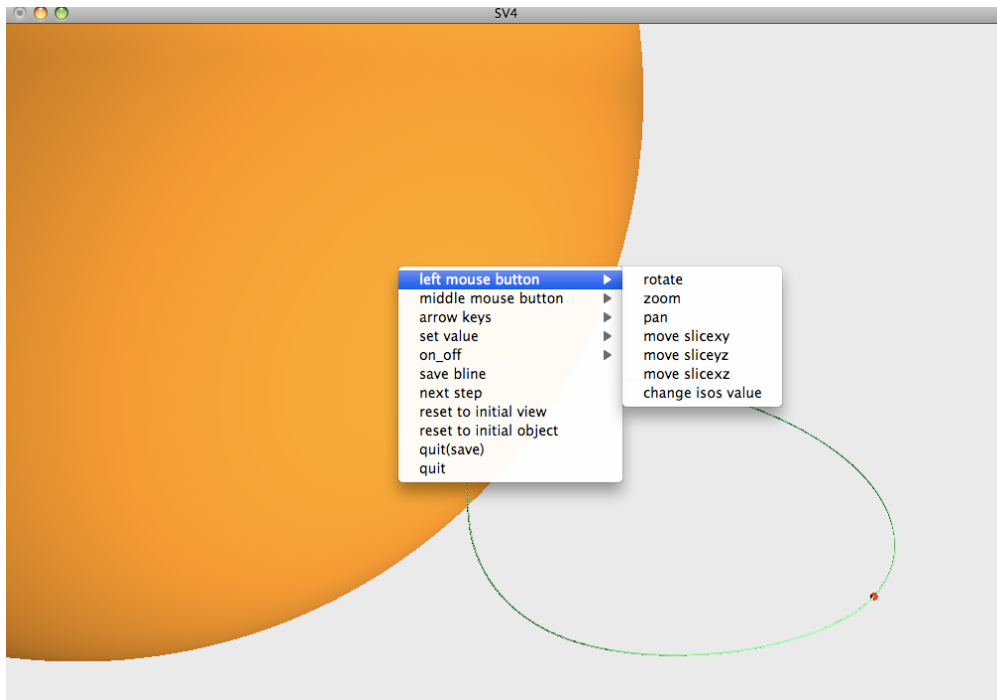


Fig. 12 A snapshot of the menu panels of SV4. The right mouse button is used to display the menu. The user can select the mouse function and the physical quantity to be visualized.

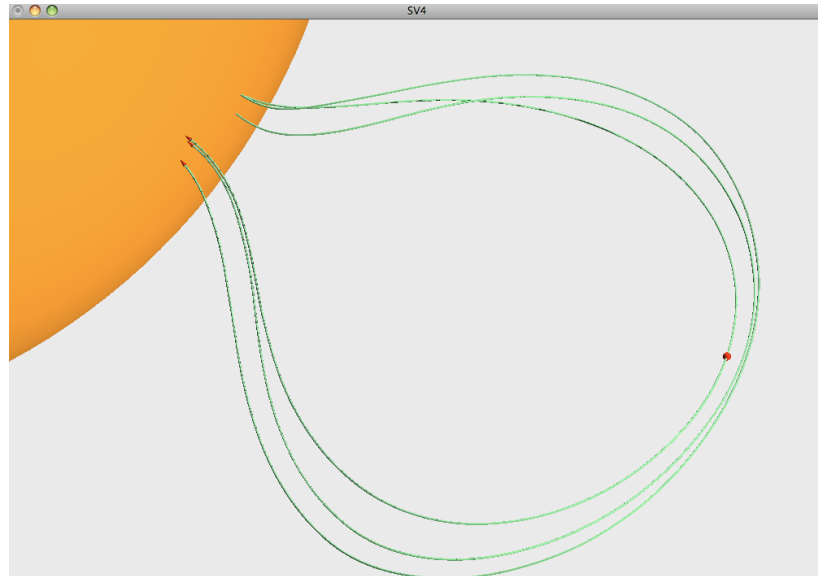


Fig. 13 An example snapshot of magnetic field lines. The lines without red ball are constructed in the previous session of analysis by SV4. The positions of the lines are saved in a file.

7 典型的な可視化手続き

この章では SV4 の典型的な可視化手続きについて説明する。

SV4 を起動し、メニューの “set value” から可視化手法と物理量を選択することで、選択した手法で指定した物理量を可視化した CG オブジェクトを表示する。磁力線可視化手法を適用する場合はデフォルトの磁力線初期点をマウสดラッグにより移動させる (Fig. 14)。またメニュー “save bline” により磁力線を保存することで複数の磁力線可視化をすることができる。等値面の可視化手法の場合はマウスを用いて等値面の値を操作することで適当な等値面を可視化する (Fig. 15)。断面図手法の場合はマウスか矢印キーを用いて、物理量を観測したい位置まで断面図持ってくることで適当な断面図を可視化する (Fig. 16)。

これらの可視化は同時に表示することができる。例えば磁力線を可視化し、更に太陽表面の r 方向の磁場を可視化することで磁力線が r 方向の磁場が正である太陽表面から出てきて、 r 方向の磁場が負である太陽表面に戻る様子が可視化できる。可視化した CG オブジェクトはマウスや矢印キーによる平行移動、回転移動、拡大縮小機能を使うことでより詳しく観察できる (Fig. 17)。視点と CG オブジェクトの状態は SV4 を終了する時に保

存されるので、再開した時に終了した時と同じ状態から始められる。この様に SV4 は複数の可視化手法と対話的操作によって、よりユーザーが解析しやすい可視化を可能にしている。

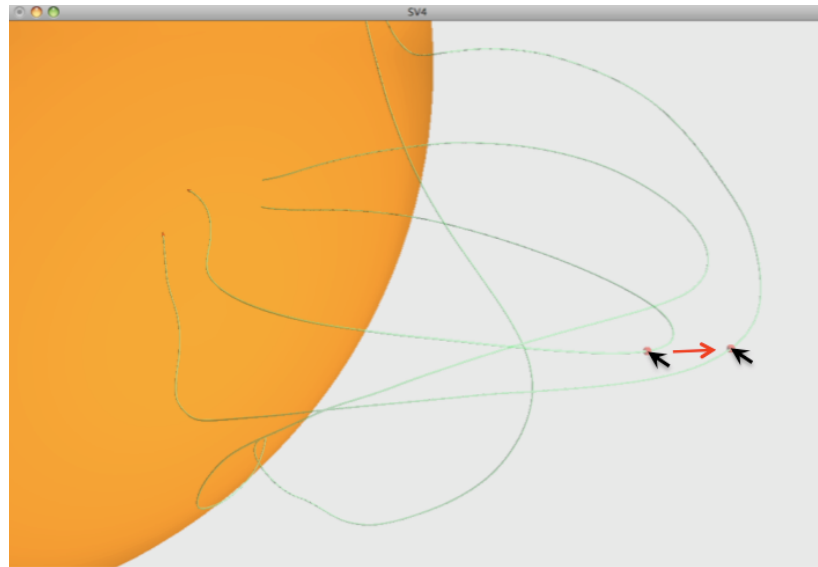


Fig. 14 Interactive change of the releasing point of the magnetic field line. As the user drags the red ball by mouse, the magnetic field line is recalculated in real time from the new position and re-displayed instantly.

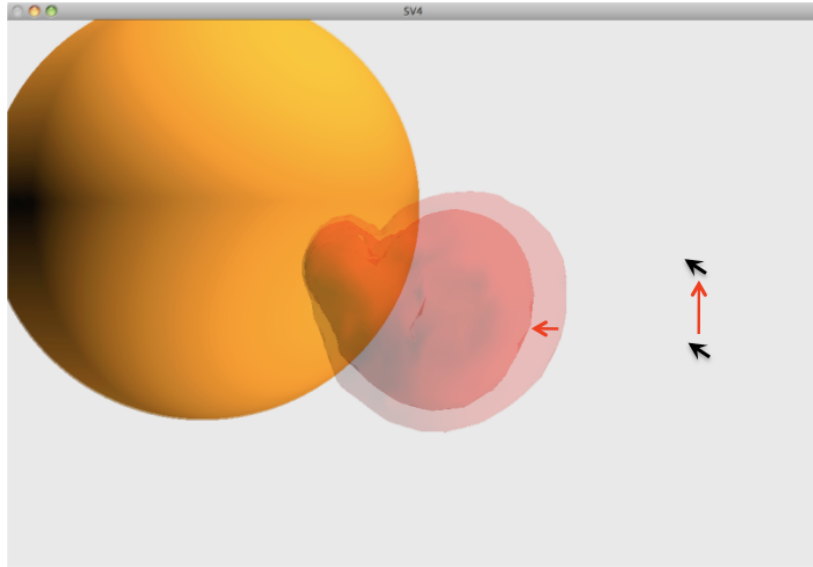


Fig. 15 Interactive control of the isosurface level. The user can change the level by the mouse drag.

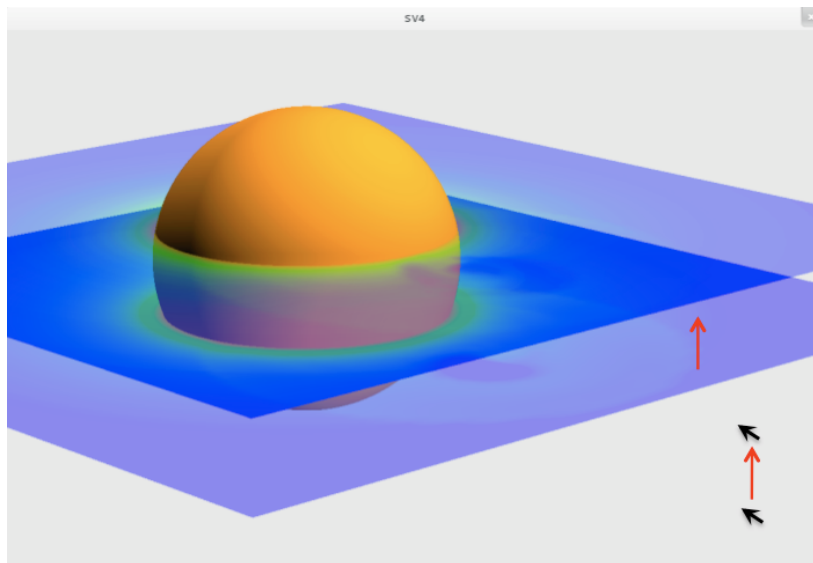


Fig. 16 Control of the slice plane. The slice plane is controlled by the mouse drag in vertical direction.

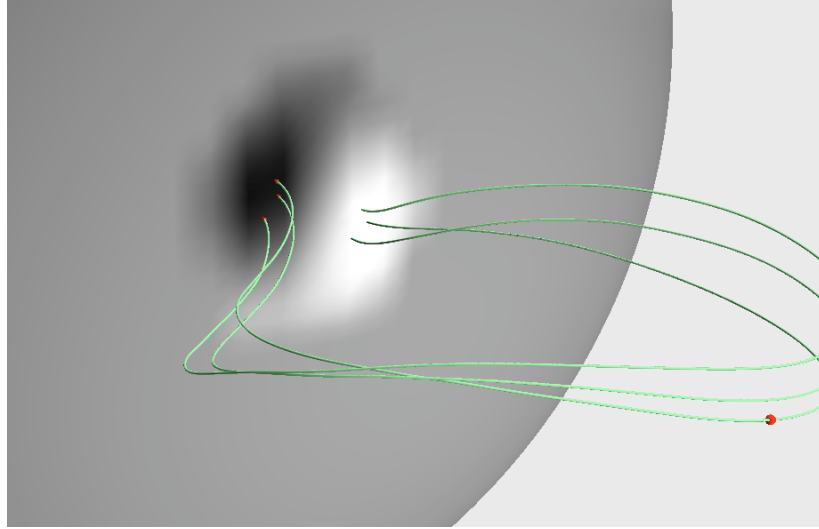


Fig. 17 The radial component of the magnetic field on the solar surface with magnetic field lines. The magnetic field lines run from positive magnetic field region of the solar surface (white) and go back to negative field region (black).

8 まとめ

宇宙天気モデリング共同プロジェクト (JPSWM) の一環として Yin-Yang 格子上で定義されたシミュレーションデータを可視化するプログラム SV4 を開発した。プロジェクトに参加する研究者は C 言語よりも Fortran90 言語になじみがあり、彼らが提供した SV4 を改良しやすくするために SV4 は Fortran90 言語で開発した。そのため SV4 では CG 用 API として Fortran90 言語用の OpenGL、GLUT である f90gl を利用した。

SV4 はカーテシアン座標から Yin-Yang 座標への座標変換を駆使することで Yin-Yang 格子上のデータを直接可視化する。可視化手法には磁力線、等値面、断面図など様々な手法を実装している。また f90gl の GLUT サブルーチンを用いたインタフェースを備えているので、マウスやキーボードを用いて視点の移動や磁力線初期点の移動などを対話的に行うことができる。

SV4 は Yin-Yang 格子上のデータを直接可視化できるだけでなく、様々な可視化手法や対話的操作により、ユーザーがシミュレーションデータをより解析しやすい可視化を可能にする。Yin-Yang 格子は最近、マントル対流や地球ダイナモシミュレーション、超新星爆発シミュレーションなど、JPSWM プロジェクト以外でも利用が広がっている。今後は、JPSWM プロジェクト以外のこのようなシミュレーションデータの可視化にも SV4 が活用されることを期待している。

9 謝辞

陰山聡教授、政田洋平助教には 2 年間を通して様々なご指導頂きましたことを深く感謝致します。名古屋大学、草野完也教授には JPSWM プロジェクトについて、ご教授頂きましたことを深く感謝致します。本研究で用いた太陽シミュレーションデータは理化学研究所の塩田大幸博士から提供して頂いたものです。データ及びシミュレーションコードについてご教授頂きましたことを深く感謝致します。

参考文献

- [1] Heliophysics. http://sec.gsfc.nasa.gov/sec_resources_imagegallery.htm
- [2] 宇宙モデリングプロジェクト, http://www.jamstec.go.jp/ifree/space_earth/jswm/ja/
- [3] Akira Kageyama and Tetsuya Sato. “Yin-Yang grid” An overset grid in spherical geometry. *Geochemistry Geophysics Geosystems*. (2004)

- [4] Akira Kageyama and Nobuki Ohno . Visualization of spherical data by Yin-Ynag grid. *Computer Physics Communicatons*. (2009)
- [5] R.A. Drebin, L.Carpenter, P.Hanrahan. Volume rendering, *CompuerGraphics, Proc.SIGGRAPH'88*. vol.22 pp.65-74(1988)
- [6] Daikou Shiot, Kanya Kusano, Takahiro Miyoshi, and Kazunari Shibata. MAGNETOHYDRO-DYNAMIC MODELING FOR A FORMATION PROCESS OF CORONAL MASS EJECTIONS:INTERACTION BETWEEN AN EJECTING FLUX ROPE AND AN AMBIENT FIELD. *TheAstrophysicalJournal*. Volume 718, Number 2,1305?1314(2010)
- [7] William F. Mitchell. f90gl. <http://math.nist.gov/f90gl/>
- [8] William E. Lorensen and Harvey E. Cline. MARCHING CUBES: A HIGH RESOLUTION 3D SURFACE CONSTRUCTION ALGORITHM. *ComputerGraphics*. Volume 21, Number 4, July 1987