

Blenderによる科学的可視化入門

外田 慎太郎¹, 高田 智史², 松本 侑士³,

(要旨) 惑星科学を始めとした様々な分野において、離散要素法や粒子法といった粒子を用いたシミュレーションが行われている。その結果を可視化する際にはParaViewやPOV-Rayなどが用いられている。特にPOV-Rayは使いこなすことができれば現実顔負けのフォトリアルな可視化を行うことができるものの、テキストベースでの可視化のため、カメラ角度設定などの操作に困難が伴う。そこで本稿ではそれに代わるツールとして3DCGソフトウェアのBlenderを紹介する。Blenderを紹介する記事や書籍などは多数存在するが、そのほとんどはゼロから対象を作成するものであり、シミュレーション結果を可視化する手法を解説するものはほとんどない。特にBlenderを使用したことがない人でもフォトリアルな可視化を行えるような手順を紹介したい。

1. はじめに

惑星科学において重力 N 体計算などの離散要素法やSPHをはじめとした粒子法などの計算は重要である。得られた結果の可視化には、従来gnuplotやMatplotlib, POV-Ray[1]などが用いられてきた。特にPOV-Rayはテキストベースで可視化を行えるフリーのソフトウェアであり、使いこなせば現実顔負けのフォトリアルな画像を作成することができる。しかし実行にはテキストファイルによる記述を用いるため、Oosawa[2]のようなフロントエンドツールを用いなければカメラの角度などの設定操作に困難が伴う。それに対し本稿では、これらに代わる可視化ツールとしてBlender[3]を紹介したい。Blenderはオープンソースのソフトウェアであり、3DCGの製作や2Dアニメーション製作、動画編集など多種多様な作業を行うことができ、惑星科学の分野でも国

立天文台の N 体学校[4]で紹介されている。実際、Blenderの操作を取り扱った書籍などは数多く出版されており、YouTubeでも「Blender」と検索すると様々な動画がヒットする。しかしそれらは自分でゼロから対象を作成することの解説がほとんどであり、科学的可視化において重要であるシミュレーション結果をフォトリアルに表現する手法を解説したものは少ない。本稿では、ともにフリーのソフトウェアであるParaView[5]とBlender[3]を併用することで、図1のようなフォトリアルな可視化を簡単に行えることを紹介していく。なお、ページの問題もあるため、離散的な粒子を用いたシミュレーション結果を可視化することのみ紹介する。ただし、3Dモデル化できる結果が手元があれば基本的にこの手法で流体などの可視化も行うことができる。

大まかな流れとしては、各自でシミュレーションを行い、(a) ParaViewで読み取り可能なファイル形式で結果を出力する、(b) ファイルをParaViewで読み込み、一度ParaView上で可視化をする、(c) それをx3d形式で出力する、(d) Blenderでx3dファイルを読み込み、それを再度可視化する、

1.東京農工大学大学院産業技術専攻

2.東京農工大学大学院工学研究院先端機械システム部門

3.神戸大学大学院理学研究科惑星学専攻

takada@go.tuat.ac.jp



図1: LIGGGHTS [6] のサンプルスクリプトを実行して得られた結果をParaView [5] とBlender [3] を用いて作成した可視化の例. 背景画像としてPoly Haven [7] からダウンロードした画像を使用している.

という流れである. なお, ParaViewで読み取り可能なファイル形式は数多くあるものの, ここではVTK(Visualization Toolkit) ファイル形式を仮定する. 理由としては, 例えば LAMMPS [8] や LIGGGHTS [6] を用いて粒子シミュレーションをしている研究者にとっては, dumpコマンドでVTKを指定するだけで簡単にVTKファイルを出力できるからである.

2. VTK ファイルの準備

今回使用したParaviewのバージョンは5.11.0, Blenderのバージョンは4.4.0である. 以下では説明のため, 下記のようなVTKファイル [9] を用いることにする. これは9つの粒子が系の中に存在している場合のファイルである. なお, 各行一番左の数字は行番号を表すためにつけたものであり, 実際のVTK ファイルでは用いないことに注意が必要である.

```

1  # vtk DataFile Version 3.0
2  Data.vtk
3  ASCII
4  DATASET UNSTRUCTURED_GRID
5
6  POINTS 9 float
7  -1.0 -1.0 -1.0
8  1.0 -1.0 -1.0
9  -1.0 1.0 -1.0
10 1.0 1.0 -1.0
11 -1.0 -1.0 1.0
12 1.0 -1.0 1.0
13 -1.0 1.0 1.0

```

```

14 1.0 1.0 1.0
15 0.0 0.0 0.0
16
17 CELL_TYPES 9
18 1
19 1
20 1
21 1
22 1
23 1
24 1
25 1
26 1
27
28 POINT_DATA 9
29 SCALARS diameter float
30 LOOKUP_TABLE default
31 1.0
32 1.0
33 1.0
34 1.0
35 1.0
36 1.0
37 1.0
38 1.0
39 2.0

```

ここで簡単にファイルの中身の説明を行う.

● 第1ブロック

これはいわゆる「おまじない」である. ファイル名が「Data.vtk」であることや, 文字コードがASCIIであること, ファイルの形式が「Un-structured Grid」形式であることを示している.

● 第2ブロック

ここには粒子の座標データを記述していく. 6行目の「9」はポイント数が9点, つまり粒子が9個であることを示す. 「float」は記述する型がfloat型であることを示す. 7行目から15行目に各粒子の座標(x, y, z)を列挙している. なお, 必ずしも1粒子ごとに改行する必要はない. ここでは見やすさの観点から改行を行っている.

● 第3ブロック

17行目は9個の粒子のタイプを述べている. 18行目から26行目の「1」は各データが1点から構成されることを表している. 粒子はそれぞれ独立で重心座標1点のみで記述できるため, ここでは「1」を個数分列挙すればよい.

● 第4 ブロック

ここでは各粒子の大きさを示している。29行目でスカラー量の直径をfloat型で記述することが宣言されている。またスカラー量の場合はルックアップテーブルを指定する必要があるが、ここでは用いないので「default」とであると指定する。31行目から39行目に各粒子に対応する直径を記述している。ここで7行目から15行目に記述した粒子と同じ順番で直径を対応させる必要がある。

3. ParaViewによる準備

前章で準備したVTKファイルをParaViewで読み込んでいく。ファイルの読み込みはParaViewを開き、ドラッグアンドドロップを行うか、もしくは「File」→「Open」で参照するファイルを選択する。読み込んだファイルを「適用(Apply)」すると、「Layout」内に表示される。ただし粒子数やファイル形式によってはこれだけでは表示されないこともある。

ここからは可視化のために粒子メッシュを作成する。読み込んだファイルを選択した状態で、「Filter」→「Alphabetical」→「Glyph」もしくは図2内で赤くハイライトされている🔍のアイコンから Glyphを追加する。ここで図3内で赤くハイライトされている項目の「メッシュ形状」、「大きさ」、「表示数」を設定する。メッシュ形状は「Glyph Type」で選択する。矢印や立方体など様々な形状が用意されているが、今回は「Sphere」を指定する。

大きさは「Scale Array」と「Scale Factor」を指定して設定する。「Scale Array」は読み取ったファイル内で定義した係数であり、今回は第4ブロックの頭で定義した「diameter」を選択する。「Scale Factor」は「Glyph Type」の寸法と「Scale Array」の数値を結ぶ値であり、直径の場合は1、半径の

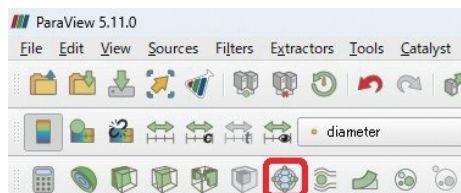


図2: 粒子のデータをGlyphで可視化する方法。アイコンが並んでいる上部から🔍を探し出し、それをクリックする。

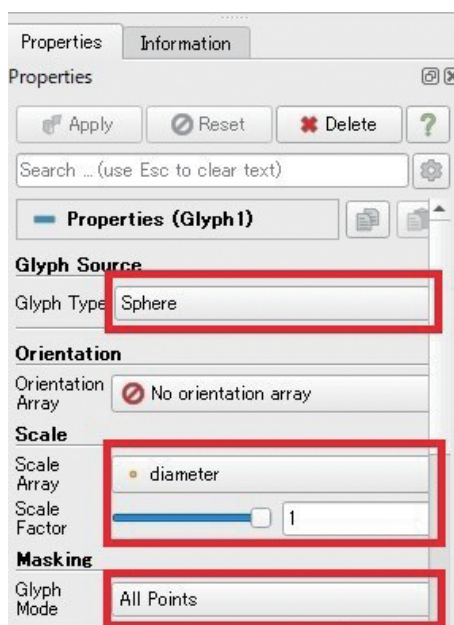


図3: Glyphで粒子の直径を反映させる方法。Sphereを選んでからScaleを「diameter」、Scale Arrayを「1」にし、その後、Maskingをすべての粒子に反映させるために「All Points」を選ぶ。

場合は2を指定する。今回は「diameter」の値であるため「Scale Factor」は1に設定する。(例えばLAMMPSなどでシミュレーション結果をVTK出力する際は半径の場合も多いので注意が必要である。)

最後の表示数は「Glyph mode」で設定する。デフォルトでは一部の粒子のみ表示する設定となっているため、これを「All Points」に変更する。

最後に「適用(Apply)」すると、「Layout」に粒子が表示される。この操作で表示されない場合、👤👤👤👤👤👤👤のアイコンからカメラ方向を設定すると、物体が中心に映るようにカメラが移動する。またカメラの基本操作は、「左クリック+マウス操作」で方向の変更、「ホイールクリック+マウス操作」で位置の変更である。上記に加えて「Glyph」の断面の表示や色分け等も可能ではあるが、本稿では省略する。

ここからは、作成した3Dモデルをx3d形式で出力する方法について説明する。事前準備として、点データの非表示、Glyph解像度の変更を行う。この段階

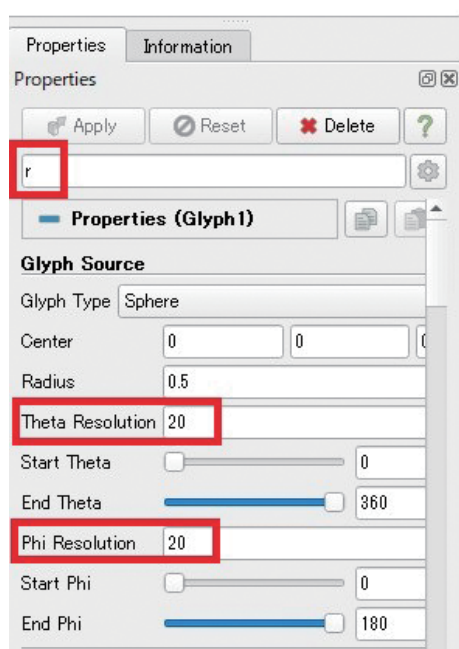


図4: Glyph の解像度を設定する画面。検索窓に「r」を入力するまでは下半分は表示されない。Glyph Sourcesが表示されたのちに「Theta Resolution」と「Phi Resolution」が表示される(赤四角で囲まれたところ)。

では左上の「Pipeline Browser」にVTKファイルとGlyphが表示されており、それぞれの左に開いた眼のアイコンが表示されている。このアイコンをクリックすることで表示/非表示を切り替えることができる。VTKファイルの点データが3Dモデルに含まないように、VTKファイルを非表示にしておく。

次にGlyphの解像度の設定を行う。「Pipeline Browser」の「Glyph」を選択し、「Properties」の「検索ボックス(Search)」に「r」と入力すると、「Theta Resolution」と「Phi Resolution」が候補に表示される。これらの値を増やすことで各Glyphの解像度を上げることができる。当然ながら解像度を高めると3Dデータが重くなってしまうため、値としては12から16程度を推奨する。実際の編集画面を図4に示す。

最後にBlenderで読み取り可能なx3d形式で出力する。「File」→「Export Scene」を選択し、「ファイル形式(Files of type)」は「X3D Files(*.x3d)」を選択し、任意の名前で保存する。また「Export Color Legend」の画面が出るが、今回は使用しな

いのでそのまま「OK」で良い。

Paraview内で行う操作は以上である。次章でBlender側の操作を示す。

4. Blender による可視化

前章で準備したx3dファイルをBlenderで読み込み、実際に可視化を行っていく。実際に行う操作としては最低限、ファイルの読み取り、環境光の導入、カメラの位置および向きの設定、画像の出力が必要である。それに加えてマテリアルの設定やライティングの追加を行うことで質を高めることが可能であるが、本稿では割愛する。Blenderを起動後、まずは言語を日本語に設定する。以降の説明は言語設定を日本語にした前提で行う。初回起動時以外は「Edit」→「Preferences」→「Interface」→「Translation」から変更できる。またデフォルトで設置されている立方体、ライトを削除しておく、カメラは後々使うため残しておく。右上の「Collection」で右クリックのメニューから削除できる。

次にx3dファイルをインポートする。まずBlenderでx3dファイルをインポートするためのアドオンを導入する。「編集」→「プリファレンス」→「エクステンションを入手」を開く。検索欄に「x3d」と入力し、「Web3D X3D/VRML2 format」をインストールす

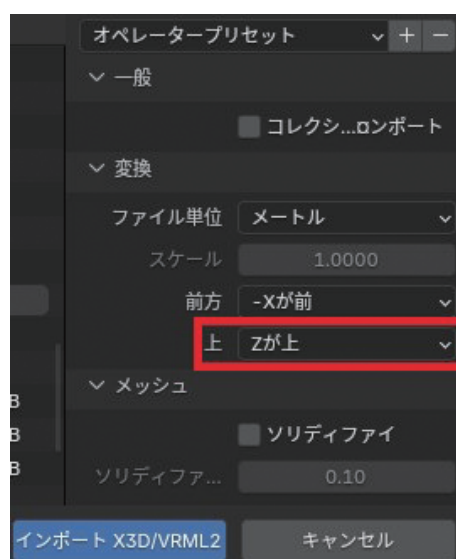


図5: Blenderでx3dファイルをインポートする際の上方向の設定。

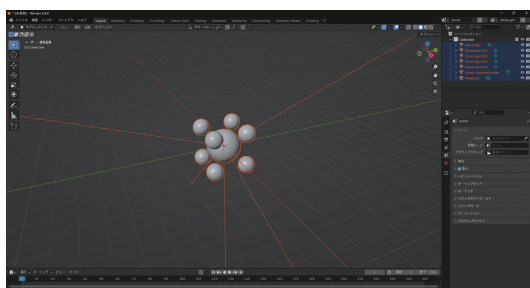


図6: オブジェクトモードにおいて粒子が描画されている状態。

る。これにより、「ファイル」→「インポート」→「X3D Extensible 3D (.x3d/.wrl)」からx3dファイルを読み込むことができる。この時、デフォルトではY軸正方向を上として読み取ってしまうため、インポートする際「変換」の「上」を「Zが上」に変更することを推奨する(図5参照)。

インポートすると図6に示すように「オブジェクト (Shape IndexedFaceSet)」と「ライト (DirectLight)」が5つ、「カメラ(Viewpoint)」が設置される。ライトと新しいカメラは不要であるため全て削除する。この際デフォルトのカメラは削除しないようにする。

次に環境光の導入を行う。これは物体を照らす明かりを設置する操作であるが、手作業で設定するには非常に労力がかかる。そこで今回は、無料で公開されている環境テクスチャを利用する。これはある空間の全方位からの光情報をまとめたテクスチャであり、適用するだけでその空間の光情報を再現することができる。具体的には「Poly Haven」というサイト[7]を利用する。この内「HDRIs」というカテゴリが環境テクスチャであるため、任意の背景を選択しEXR形式でダウンロードする。Blenderへの導入は、右側の地球儀のマークの「ワールド」タブから行う。「カラー」のすぐ右の黄色い丸アイコンをクリックし、「テクスチャ」内の「環境テクスチャ」を選択する(図7参照)。その後「開く」から先ほどダウンロードした環境テクスチャを選択することでBlender内でも適用される。適用された結果は右上の「ビューポートシェーディング」を「ソリッドモード」から「レンダープレビュー」に切り替えることで確認できる。図8右上の4つの内、左から2番目が「ソリッドモード」、4番目が「レンダープレビュー」である。視点の操作方法は「マウス

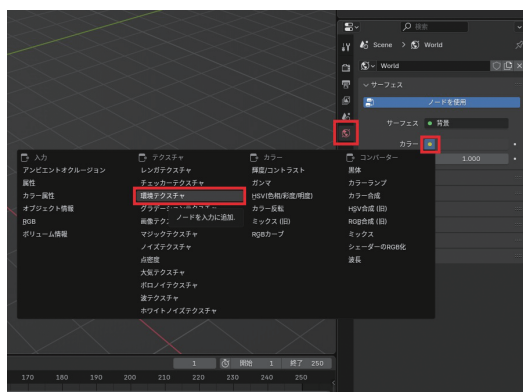


図7: 環境光の設定手順。

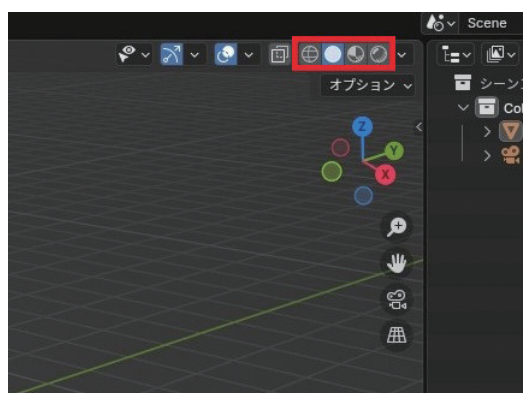


図8: レンダープレビューモードへの切り替え方法。画面右上のアイコンから選択する。

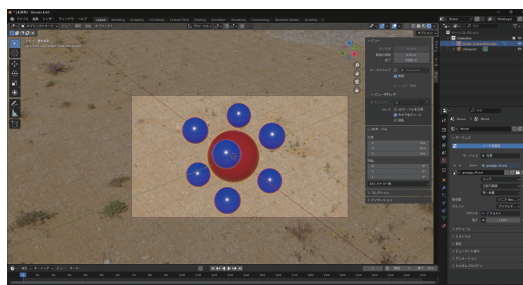


図9: レンダープレビューモードにおいて粒子が描画されている状態。明るく照らされている領域がレンダリングされる領域となっている。

ホイール押し込み+マウス操作」で視点の向きを、「マウスホイール押し込み+Shift+マウス操作」で視点の位置を、「マウスホイール回転」でズームイン・ズームアウトができる。

次にカメラ位置の設定を行う。視点を移動し最終的に出力したい向きからオブジェクトを視界に入れ

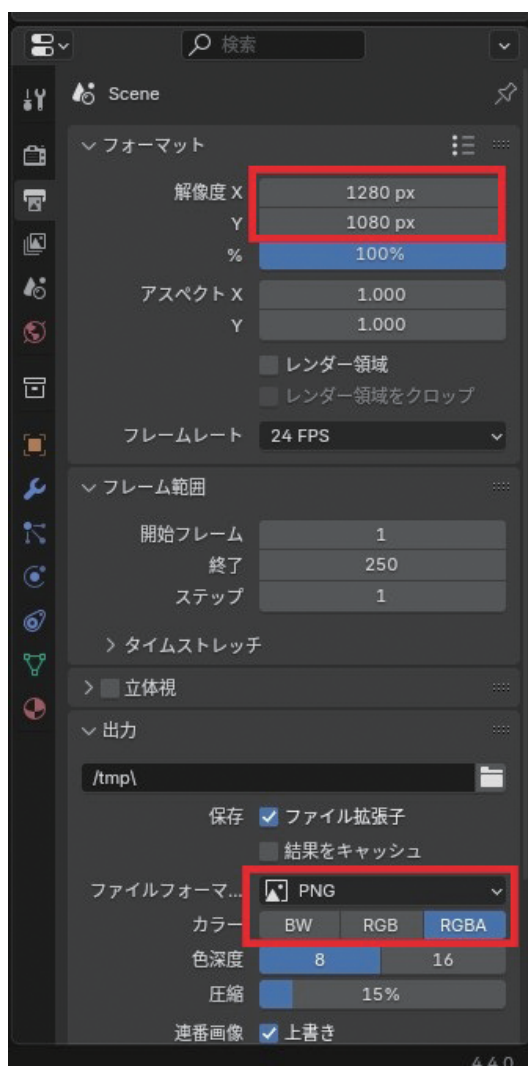


図10: 作成したい画像の解像度などの設定。

て、左上の「ビュー」→「視点を揃える」→「現在の視点にカメラを合わせる」を選択することでその画角にカメラを移動できる。この時画面中央の色が鮮やかなエリアが最終的に出力される範囲である(図9参照)。右上のサイドバーの「ビュー」→「ビューのロック」→「カメラをビューに」にチェックを入れることでそれ以降の視点移動にカメラが追従するようになるため、カメラ位置や向きの微調整はこの状態で行う。サイドバーはデフォルトでは最小化されているため、キーボード「N」キーで表示を切り替えることができる。

最後に画像の出力を行う。出力設定は右の「レ

ンダー」と「出力」タブから行う。まずは「出力」タブの設定から行う。編集する項目を図10に示しておく。今回は静止画を出力するため、「フォーマット」の縦横の解像度と「出力」→「ファイルフォーマット」のファイル形式を任意に設定する。ここでは最終的に背景を透過した画像を作成したいため、ファイルフォーマットを「PNG」に、カラーを「RGBA」に設定している。また出力するディレクトリも設定できるが、出力結果を見てから保存するかを決めるためにデフォルトの「/tmp\」のままにしておくことを推奨する。

次に「レンダー」タブの設定を行う。「出力」タブと同様に編集する項目を図11に示しておく。背景を透過した画像を作成したい場合、「フィルム」→「透過」にチェックを入れることで環境テクスチャを透過した画像を出力できる。この場合環境テクスチャによる光自体は有効となる。また最も重要な項目として「レンダーエンジン」の設定がある。これは「Eevee」、「Workbench」、「Cycles」の3種があるが、基本的に「Eevee」が「Cycles」を使用する。簡単に説明すると「Eevee」は軽量、「Cycles」は高クオリティなエンジンであり、コンピュータの性能さえ問題がなければ、手軽に高クオリティの画像を出力できる「Cycles」を推奨する。ただしデフォルトの設定では出力にかなりの時間がかかってしまうため、軽くする方法を紹介する。

以下は「Cycles」を選択した際の設定である。まずコンピュータにGPU(グラフィックボード)が搭載されている場合、「デバイス」を「GPU 演算」に変更する。この時、左上の「編集」→「プリファレンス」→「システム」→「Cyclesレンダーデバイス」で使用するGPUにチェックが入っている必要がある。「なし」が選択されていた場合は「CUDA」が「OptiX」に変更することで上記の設定が可能となる。

次にサンプリングの設定を行う。「サンプリング」→「レンダー」→「最大サンプル数」で、レンダリング時のサンプル数を設定できる。デフォルトの4096は非常に時間がかかるため100~1000程度に設定しておく。出力結果にノイズ等の問題が発生する場合にサンプル数を増やすと良い。また「ビューポート」は出力前の現段階の画面出力の設定であり、視点移動に応じてリアルタイムで再計算を行うためこちらも値を小さくしておくといよい。

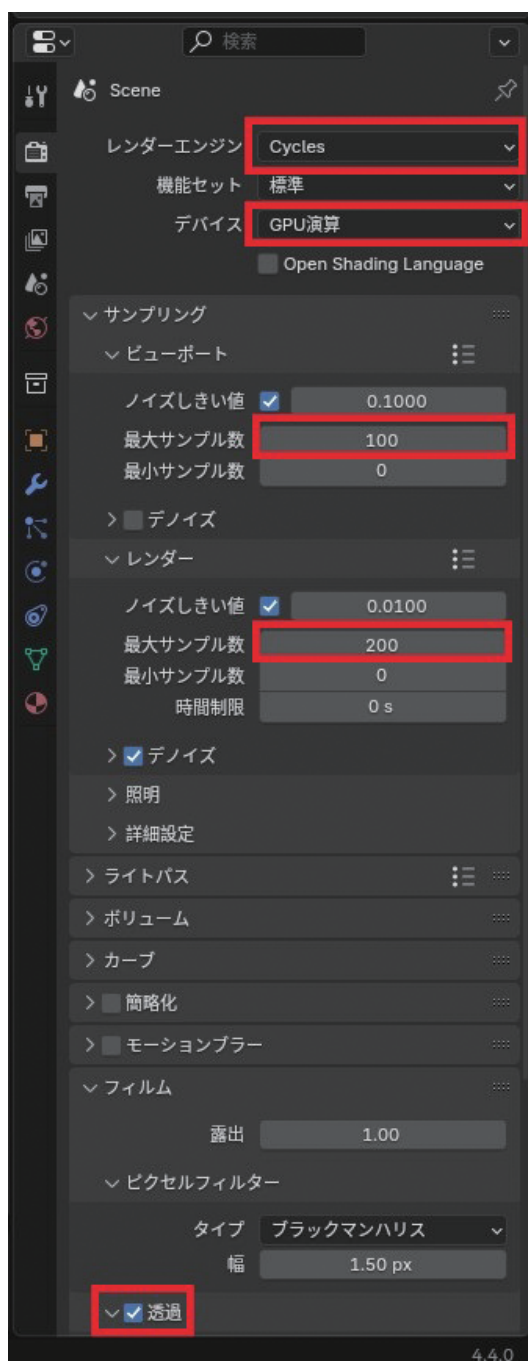


図11: レンダーエンジンの設定方法。デフォルトではEeveeが選択されていることが多いが、よりフォトリアルスティックな可視化を行うためにCyclesを選択する。またレンダリングする際の最大サンプル数の設定もここで行う。

以上で出力前の設定は完了である。最後に左上の「レンダー」→「画像をレンダリング」を選択することで

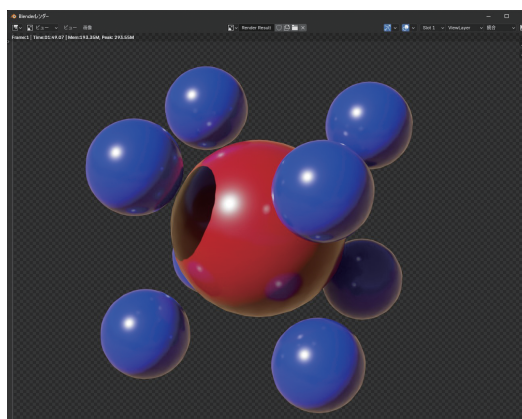


図12: レンダリングが終了したのちに出力される画像。これで問題なければこれを保存すれば終了である。

新しくウィンドウが作成され、出力が開始する。ウィンドウ上部に現在の進捗が表示されており、右の「～Remaining:～」や「～Sample～」が表示されなくなったら出力が終了である(図12 参照)。結果に問題がなければ「画像」→「名前を付けて保存」から保存して一連の流れは終了である。この際「名前を付けて保存」が選択できなければ出力途中であるため、終了したかはこちらでも確認できる。

5. おわりに

本稿ではBlenderを用いた科学的可視化の方法について簡単に紹介した。一度ParaViewで可視化し、x3dファイルに変換、それを再度Blenderで可視化するというステップを踏む必要があるものの、そこまで複雑ではない操作だけである程度の質が担保されたことは特筆に値するのではないかな。もちろん、筆者らもBlenderの機能をすべて使いこなせているわけではない。本稿を読んでBlenderによる可視化に興味を持っていただいた読者の方から、著者らにフィードバックを与えていただければ幸いである。なお、機会があればシミュレーション結果を動画にする方法などについても紹介したい。

References

- [1] <https://www.povray.org/>.
- [2] <https://qcganime.web.fc2.com/OOSAWA/oosa->

wa.html.

- [3] <https://www.blender.org/>.
- [4] <https://www.cfca.nao.ac.jp/content/n%E4%B-D%93%E5%AD%A6%E6%A0%A1>.
- [5] <https://www.paraview.org/>.
- [6] Kloss, C. et al., 2012, Prog. Comput. Fluid Dyn. An Int. J. 12, 140.
- [7] <https://polyhaven.com/>.
- [8] Thompson, A. P. et al., 2022, Comp. Phys. Comm. 271, 10817.
- [9] <https://docs.vtk.org/en/latest/>.

著者紹介

外田 慎太郎

東京農工大学大学院工学府産業技術専攻在学。2024年、東京農工大学工学部機械システム工学科卒業。専門は粉体工学。日本機械学会に所属。

高田 智史

東京農工大学大学院工学研究院先端機械システム部門准教授。2016年、京都大学大学院理学研究科物理学・宇宙物理学専攻博士後期課程修了。博士(理学)。京都大学大学院理学研究科特定研究員、東京大学地震研究所特任研究員、東京農工大学大学院工学研究院先端機械システム部門テニュアトラック准教授を経て、2024年3月より現職。専門は粉体物理学、材料力学。日本物理学会、日本機械学会などに所属。

松本 侑士

神戸大学大学院理学研究科惑星学専攻特命助教。2014年、東京工業大学大学院理工学研究科地球惑星科学専攻博士後期課程修了。博士(理学)。国立天文台、千葉工業大学などを経て、2025年より現職。専門は惑星形成。日本惑星科学会、日本天文学会などに所属。