# 空間ダイナミックレンジが大きい AMR シミュレーションのシームレスな可視化

松本倫明

## 1. はじめに

適合格子細分化法（AMR 法）は宇宙流体シミュレーションで急速に普及した計算手法である。AMR 法では解像度が必要な領域を自動的に選び出し、その領域の格子を細分化して高い解像度を得る。その結果すべての領域を細かい一様格子で覆うよりも少ない計算量で、局所的に高い解像度を得ることができる。すなわち現実的な計算資源で、解像度を何桁も向上することができる。

AMR 法は、もともとは衝撃波を鋭く捉えるために考えられたものである（Berger & Colella 1989）。衝撃波の移動に合わせて高解像領域も移動させ、実行的に高い解像度で衝撃波を捕捉する。一方、宇宙流体のシミュレーションでは、自己重力系における天体形成で AMR 法が用いられることが多い。星形成や銀河形成では、大きなガス雲から小さな天体が形成する。この様子を再現するためには、母体となる大きなガス雲と形成する小さな天体の間の何桁にも渡る広い空間ダイナミックレンジをカバーする必要があり、AMR 法は有効な方法である（e.g., Offner et al. 2009; Kim et al. 2014; Matsumoto et al. 2017）。

数値計算法として AMR 法は有効な方法であるが、計算結果の可視化には工夫が必要である。広い空間ダイナミックレンジのため、注目する領域を拡大するような可視化がしばしば用いられる。図 1 の例では初期に半径 0.14 pc、中心密度 $10^{-19}$ g cm$^{-3}$ の分子雲コアが収縮し、星周円盤と渦状腕をともなう三個のガス塊（原始星の前段階）が形成する（松本 2007）。ガス塊の半径は数 au 程度であり、密度は $10^{-8}$ g cm$^{-3}$ に達する。ガス塊の大きさは母体の分子雲コアの 30 万分の 1 と極端に小さい。AMR 格子の細分化の回数を表す AMR レベルは 14 まで達し、分解能を表す格子の幅はもっとも粗いグリッドで 1,000 au であるのに対し、もっとも細かいグリッドで 0.06 au と $2^{-14}$ 倍も小さい。このように、天体の大きさや密度に広いダイナミックレンジがあるのは、自己重力系の典型的な空間スケールがジーンズ長で決まり、ジーンズ長は密度の平方根の反比例するという性質に由来する。

図 1 に示す可視化の例は特定の 4 個の空間スケールで 5 個の断面図を表示したものである。空間スケールを不連続に変えた可視化である。これとは別に、広いダイナミックレンジを表現する方法として、動画上で対象を連続的に拡大・縮小よる可視化がしばしば行われる。連続的な拡大・縮小による可視化を用いた例として、古典的には Powers of Ten（1977）が有名である。また 3 次元計算の可視化には図 1 のような断面図ではなく、3 次元のボリュームレンダリングや等値面による可視化が好まれることが多い。

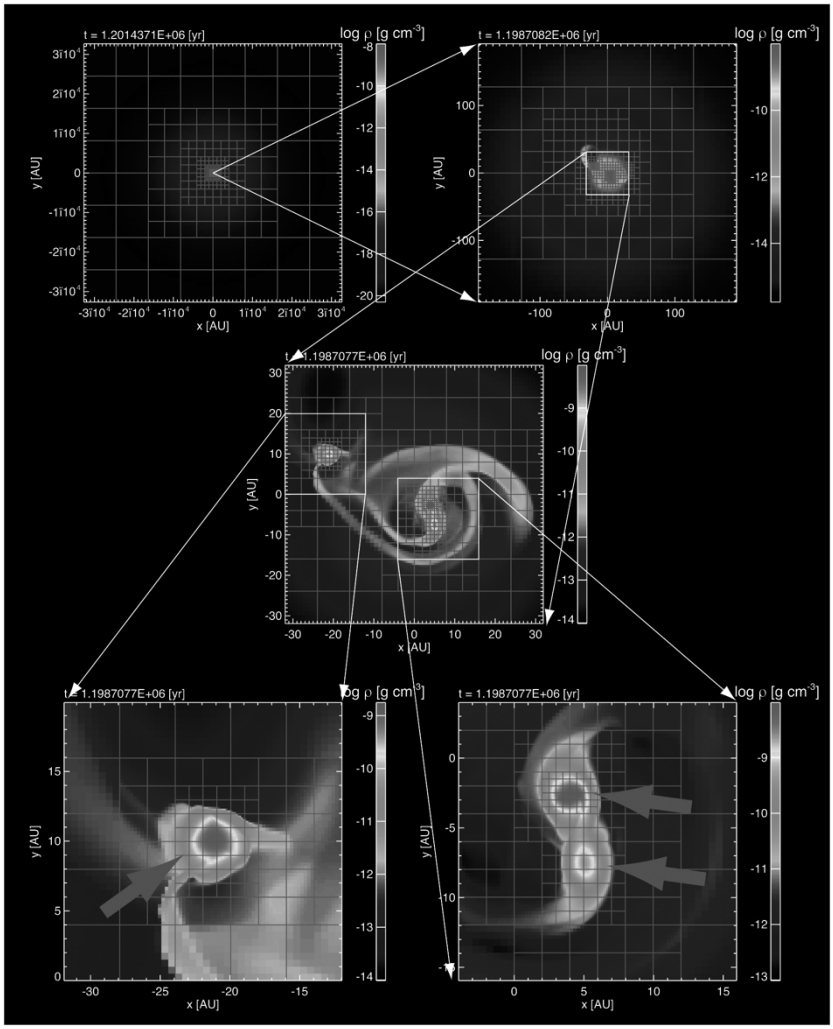


図 1　分子雲コアにおける三重星の形成を AMR 法で可視化した例（松本 2007）。図の各領域の断面図を拡大して表示している。色はガスの密度の対数を示す。矩形は AMR ブロックを表し、この例では 1 個のブロックに $8^3$ 個のセルが含まれる。矢印は 3 個の高密度ガス塊を示し、これらのガス塊はそれぞれ原始星へ進化すると考えられる。

そこで本稿では AMR 法を用いたシミュレーションデータの可視化について述べる。広いダイナミックレンジを持つ 3 次元データを連続的に拡大して可視化する方法について述べる。広いダイナミックレンジは、空間スケールだけではなく密度も有しており、これらの情報をシームレスに可視化する。可視化ではガスの密度のボリュームレンダリングを用いて立体的に表現する。

本稿の構成は次の通りである。2 章では可視化事例とサンプルコードのアクセス方法を述べる。3 章と 4 章ではそれぞれ可視化に用いるシミュレーションとそのデータを紹介する。5 章では実際の可視化と動画の作成方法について述べる。各手順においける検討するべきこ

とがらについてもその都度議論する。6 章で本論文をまとめる。

## 2. 可視化事例とサンプルファイル

本稿の方法で可視化した事例を YouTube で公開している。以下の URL からアクセス可能である。

`https://youtu.be/MLHLYDmz6J8`

この動画の前半では、分子雲コアから出発して原始星と星周円盤にズームインする様子が可視化されている。動画の後半は回転しながらズームアウトするカメラによる可視化で、本稿の方法を応用したものである。

可視化に用いたスクリプトを GitHub で公開している。以下の URL からアクセス可能である。

`https://github.com/t0m0-tomo/zoom_in_movie`

これらのスクリプトは参考事例であり、一般的に記述されていない。読者が可視化したい問題に合わせて修正することを想定している。とくに SFUMATO ユーザが各自のデータを可視化するときにこれらのスクリプトは可視化に役立つであろう。

## 3. AMR シミュレーションの概要

本稿で可視化に用いるデータは分子雲コアから原始星が形成するシミュレーションによって計算された。数値モデルは Matsumoto et al. (2017)を発展させたものである。Matsumoto et al. (2017)では原始星が形成してから 1000 年後までしか計算できなかった。この原因はモデルに磁場を取り入れたために、原始星のモデルであるシンク粒子付近でアルフヴェン速度が極端に大きくなり、時間ステップ幅も極端に短くなったためである。その後 Matsumoto et al. (2019)は新しい MHD スキーム「Boris-HLLD 法」を開発し、MHD シミュレーションにおける時間ステップ幅を比較的大きくとれるように改善した。その結果、原始星が形成してから 10,000 年後までを計算することができた。本稿では原始星形成後 9,518 年のデータを可視化する。

シミュレーションに用いた初期条件は Matsumoto et al. (2017)のモデル M1B01 と同一である。まず分子雲コアのモデルとして、臨界 Bonnor-Ebert 球（Bonnor 1956; Ebert 1955）の密度分布を持つガス雲を考えた。このガス雲を重力収縮に対して不安定にするために長さのスケールを $2^{1/2}$ 倍拡張した。これは密度を 2 倍増やすことと等価である。ガス雲の中心におけるガスの密度を $10^{-8}\,\mathrm{g\,cm^{-3}}$（個数密度 $2.61\times10^{5}\,\mathrm{cm^{-3}}$ に対応）とし、半径を $R_c = 0.0614$ pc、ガスの温度を 10 K（音速は 0.19 km $\mathrm{s^{-1}}$ に対応）とした。このガス雲を大きさ $x, y, z \in [-2R_c, 2R_c]^3$ の計算領域の中央に配置した。ガス雲の外側には臨界 Bonner-Ebert 球の外縁部の密度と同じ密度のガスを置いた。初期の速度場として乱流場を仮定した。乱流場は等方的で、マッハ数の計算領域全体わたる平均値を 1 とし、Larson (1981)のスケーリング則と

整合的なパワースペクトル$P(k) \propto k^{-4}$を持つ。初期の磁場は $z$ 方向に一様な強度25.6 $\mu$Gを仮定した。これは臨界磁場強度の 0.1 倍に相当する。以上が初期条件である。

ガスの状態方程式としてバロトロピックな関係式$P(\rho) = c_s^2\,\rho + \kappa\rho^{7/5}$を仮定した。ここで$P$と$\rho$は圧力と密度を表し、$\kappa = c_s^2\rho_{cr}^{-2/5}$である。状態方程式が等温から断熱に変化する臨界密度を$\rho_{cr} = 10^{-13}$ g cm$^{-3}$(個数密度$n_{cr} = 2.62 \times 10^{10}$ cm$^{-3}$に対応)とした。また Machida et al. (2007)と同様のオーム散逸を仮定した。密度が$\rho_{\rm sink} = 1 \times 10^{-10}$ g cm$^{-3}$に達し、いくつかの条件（Matsumoto et al. 2015）が満たされるとシンク粒子が生成される。シンク粒子は原始星を表す簡易的なモデルで、ガスを降着して質量を増し、質量に応じて点源重力を発生させる。シンク粒子の半径は 1.55 au であり、実際の原始星の半径よりも 100 倍以上大きい。大きなシンク粒子を仮定することにより、AMR の細分化が過度に進むのを抑制し、現実的な計算時間で原始星周辺構造の進化を再現する。

シミュレーションでは、筆者が開発した AMR コード SFUMATO を用いた（Matsumoto 2007)。AMR のもっとも粗いグリッドレベルを 0 とし、このレベルのセル幅を 198 au、セル数を $256^3$ 個とした。もっとも細かいグリッドレベルは 9 であり、セル幅は 0.386 au である。もっとも粗いセル幅ともっとも細かいセル幅の比は、最大グリッドレベルの 2 のべき乗に等しい（198 au / 0.386 au = $2^9$)。これは原始星周辺を解像する分解能は、初期に分子雲コアを置いたグリッドよりも 512 倍高いことを意味する。

以上の設定に基づいてシミュレーションを遂行した。分子雲コアは乱流によって乱されつつ、重力収縮によって中心部の密度が増加し、原始星（シンク粒子）が形成する。シンク粒子の周囲には星周円盤が形成する。本稿では原始星形成後 9,518 年のデータを可視化する。

## 4. 可視化データの形式

SFUMATO は 2 種類のデータ形式の出力をサポートしている。一つ目はジョブのリスタートに用いるファイルで、ある時刻におけるシミュレーションのすべての情報を含んでいる。二つ目は図 2 のように任意の領域を一様格子に切り出したファイルで、一様格子の分解能は AMR のグリッドレベルを単位として与えられる（以後、分解能レベルと呼ぶ)。本稿では後者の一様格子のファイルを用いる。多くの 3 次元可視化ソフトウェアには AMR の階層格子を直接可視化できるものもあるが、使用できる描画方法が限られるほか、階層格子の配置情報が複雑になるなどの問題がある。また 3 次元の可視化では細分化した AMR 格子を認識できるほどヒトの目の解像度は良くない。したがって AMR 階層格子を一様格子にリマップしたデータを可視化する方が便利である。

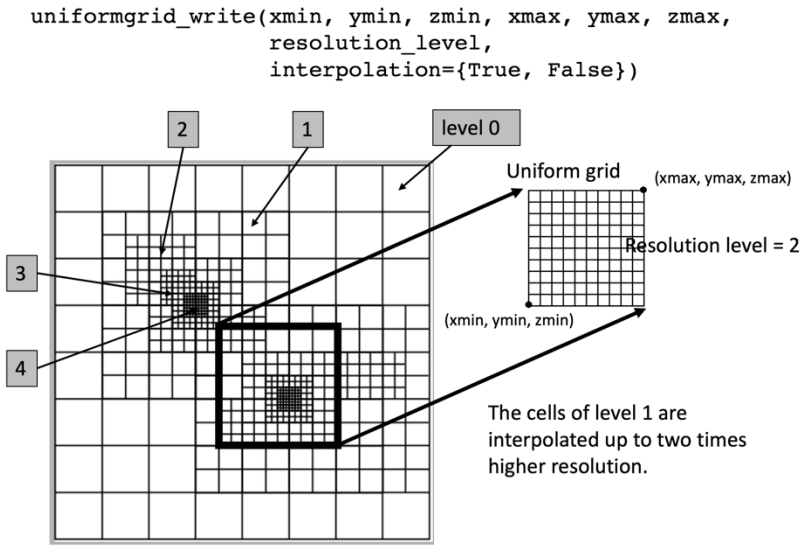


図 2　AMR 格子から一様格子を出力する模式図。サブルーチン `uniformgrid_write` は座標（`xmin, ymin, zmin, xmax, ymax, zmax`）で指定された領域を指定された分解能（`resolution_level`）で切り出す。指定する分解能は AMR のグリッドレベルを単位とし、これを分解能レベルと呼ぶ。図の例では分解能レベルとして 2 を指定している。図は簡単化のために 2 次元で図示しているが、実際は 3 次元である。

今回可視化で用いるデータは原始星形成後 9,518 年の時点におけるスナップショットである。分解能レベル 0 から 9 の 10 段階の分解能に応じて 10 個の一様格子のファイルを出力した（表 1 参照)。分解レベルが 1 以上のすべてのデータはシンク粒子を中心としており、解像度によらずセル数を $256^3$ 程度とした。したがって、分解能レベルが大きい（解像度が高い）ほど、小さな領域のデータを持つ（図 3 参照)。表 1 に可視化に用いたファイルのファイル名とそれぞれの一様格子が持つ物理的な領域の大きさ（領域の一辺の半値幅と体積)、分解能レベル、一様格子が持つセル数を示す。ファイル ug1765000.0.xdmf は計算領域全体を表す一様格子で、計算領域の物理的な大きさは $(0.2456\ \mathrm{pc})^3$ である。分解能レベルはもっとも粗いレベル 0 である。これは AMR のグリッドレベルが 0 の分解能と同じ分解能を意味する。セル数は $256^3$ 個である。ファイル 50au.1.1765000.9.xdmf は AMR のグリッドレベル 9 相当のもっとも細かい解像度を持ち、データはシンク粒子を中心とした体積（100 au)$^3$ のほぼ立方体に近い直方体である。シンク粒子から直方体の面までの距離は箱サイズ $h$ に等しい。セル数は x, y, z 方向にそれぞれ 259, 258, 259 個並ぶ。3 方向でセル数が異なるのは、シンク粒子がセルの配置とは無関係の位置に存在するために、実数の箱サイズが整数のセル数に丸められるためである。

表 1 に掲載したファイルはすべて XDMF 形式である。XDMF 形式はシミュレーション等のデータを保存するための汎用的な形式であり、XML 形式の拡張になっている。XDMF

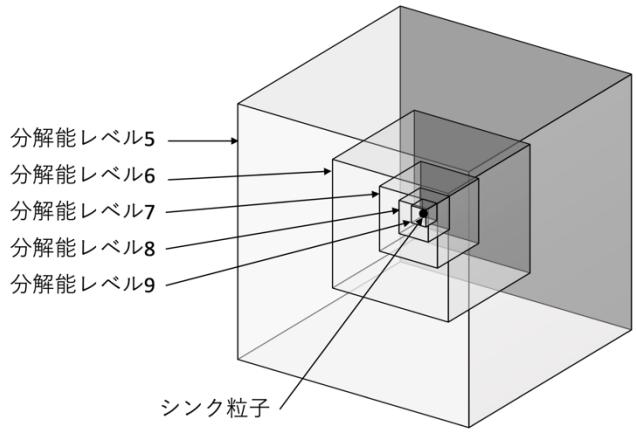


図 3　AMR 格子から切り出された一様格子の空間的な配置を示す模式図。分解能レベルが 0 以外のすべての一様格子はシンク粒子を中心として入れ子状に配置され、分解能レベルが大きいほど（分解能が高いほど）小さな領域を覆う。

ファイル自体にデータを含めることも可能であるが、本研究では別途用意したバイナリデータへのアクセス方法を XDMF ファイルに記載した。

表 1　一様格子のファイル名とデータの領域

| ファイル名 | 箱サイズ $h$（半値幅） | 領域 | 分解能レベル | セル数 |
|---|---|---|---|---|
| ug1765000.0.xdmf | 25,329 au | $(0.2456\ \mathrm{pc})^3$ $=(50{,}658\ \mathrm{au})^3$ | 0 | (256, 256, 256) |
| 16000au.1.1765000.1.xdmf | 16,000 au | $(32{,}000\ \mathrm{au})^3$ | 1 | (323, 324, 324) |
| 8000au.1.1765000.2.xdmf | 8,000 au | $(16{,}000\ \mathrm{au})^3$ | 2 | (324, 324, 324) |
| 4000au.1.1765000.3.xdmf | 4,000 au | $(8{,}000\ \mathrm{au})^3$ | 3 | (323, 324, 324) |
| 2000au.1.1765000.4.xdmf | 2,000 au | $(4{,}000\ \mathrm{au})^3$ | 4 | (324, 323, 324) |
| 1000au.1.1765000.5.xdmf | 1,000 au | $(2{,}000\ \mathrm{au})^3$ | 5 | (324, 324, 323) |
| 400au.1.1765000.6.xdmf | 400 au | $(800\ \mathrm{au})^3$ | 6 | (259, 258, 259) |
| 200au.1.1765000.7.xdmf | 200 au | $(400\ \mathrm{au})^3$ | 7 | (259, 258, 259) |
| 100au.1.1765000.8.xdmf | 100 au | $(200\ \mathrm{au})^3$ | 8 | (259, 258, 259) |
| 50au.1.1765000.9.xdmf | 50 au | $(100\ \mathrm{au})^3$ | 9 | (259, 258, 259) |

## 5. 可視化

表 1 に示した解像度が異なる 10 個のファイルを 3 次元のボリュームレンダリングを用いて連続的に拡大する。カメラが分子雲コアに突入するような動きの動画を作成する。

はじめに、カメラを計算領域全体が見渡せる初期位置に置く。ここでは初期位置をシンク粒子から距離$d_{st} = 4h_0 =$ 100,904 auの点とした。ここで$h_0$は、分解能レベル 0 の一様格子の箱サイズ $h$ を表す（表 1 参照)。つぎに、カメラを初期位置から停止位置まで移動させながらシンク粒子方向にカメラを向けてガスを撮影する。ここでは停止位置をシンク粒子から距離$d_{ed} = 2h_9 = 100$ auの点とした。カメラをシンク粒子に近づけながら、可視化対象の一様格子を分解能レベル 0 から 9 のものまで 10 個を乗り継ぐ。一様格子を不連続に乗り継ぐと、動画の拡大シーンは不連続になる。連続的に拡大するためには、異なる分解能レベルの一様格子をなめらかに乗り継ぐ必要がある。本稿の可視化手法のポイントは、なめらかに乗り継ぐ方法である。

カメラの移動中に、シンク粒子からの距離の対数で等間隔に 150 フレーム撮影した。これはフレームレートが毎秒 15 フレームのときに 10 秒間に相当する。可視化ソフトェアにはParaview[1]を用いた。

### 5.1. 不透明度の調整

カメラの位置に応じてガスの不透明度を調整した。通常ガスの分布を、ボリュームレンダリングを用いて可視化するとき、ガスの密度が高いほど不透明度を高くしてガスの濃淡を表現する。カメラが高密度ガスの中に突入すると、カメラの目の前のガスが不透明のために奥の様子を見通すことができない。この状況を回避するために、カメラの周囲にあるガスの不透明度を下げるように調整を施した。表記を簡単にするためにガスの数密度を初期のガス雲の中心数密度で無次元化した対数を

$$\eta = \log \frac{n}{n_0}$$

と定義する。ここで$n_0 = 2.61 \times 10^5$ cm$^{-3}$は初期のガス雲の中心密度である。カメラの位置における基準となるガスの無次元化密度を

$$\eta_t = \min\left\{\eta_{st} + \log\left(\frac{d_{st}}{d}\right)^2 , \eta_{ed}\right\}$$

$$\eta_{st} = 6$$

$$\eta_{ed} = 30$$

とする。ここで等温ガス雲の性質$n \propto d^{-2}$の関係式を用いている。この基準となる数密度$\eta_t$の関数として不透明度$\kappa$を図 4 のように定めた。この関数の節点を

---

[1] 3 次元可視化ソフトウェア。オープンソースで開発されている。
https://www.paraview.org

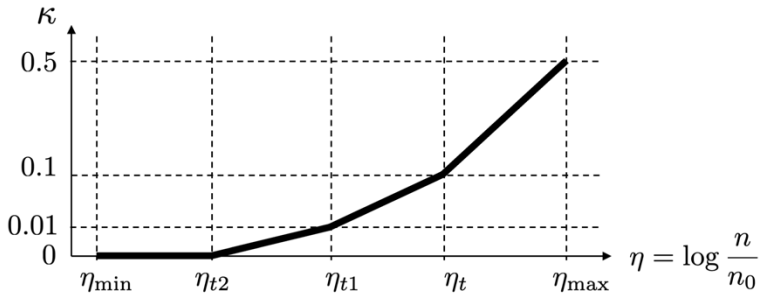


図 4 不透明度と数密度の関係。不透明度は数密度の対数の区分線形関数である。

$$\eta_{\min} = 4$$
$$\eta_{t2} = \max(\eta_t - 3, 4.3)$$
$$\eta_{t1} = \max(\eta_t - 1.6, 4.3)$$
$$\eta_{\max} = 13$$

とした。この関数を用いて、カメラの位置に応じてガスの不透明度を調整した。

表 2 に示すカラーパレットを用いた。$\eta_{\min} \leq \eta \leq 6.5$の範囲は既存のカラーパレット「Cool to Warm」を利用し、この範囲以上ではカラーパレットを自作した。

表 2 カラーパレット

| $\eta$ | R | G | B |
|---|---|---|---|
| $\eta_{\min}$ | 0.231373 | 0.298039 | 0.752941 |
| 4.4 | 0.231373 | 0.298039 | 0.752941 |
| 5.45 | 0.865003 | 0.865003 | 0.865003 |
| 6.5 | 0.705882 | 0.0156863 | 0.14902 |
| 8.5 | 0.992218 | 0.555217 | 0.236278 |
| 9.5 | 1 | 1 | 0 |
| 10 | 0 | 1 | 0 |
| 11 | 0 | 0.8 | 0.7 |
| 11.5 | 0 | 0.3 | 0.8 |
| 11.75 | 0.05 | 0.2 | 0.8 |
| 12 | 0.1 | 0.1 | 0.8 |
| $\eta_{\max}$ | 0.7 | 0.2 | 0.7 |

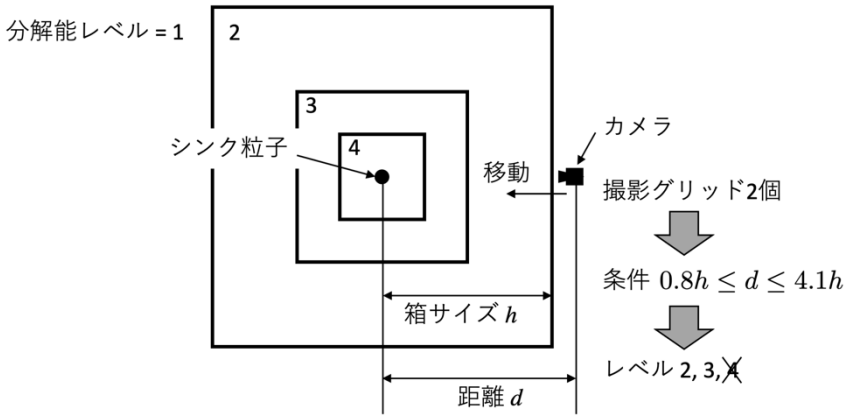


図 5 カメラ位置と撮影される一様格子（撮影グリッド）の関係を示す模式図。カメラはシンク粒子に近づきながら撮影する。カメラとシンク粒子間の距離 $d$ と箱サイズ $h$ の条件が満たされる一様格子のうち、分解能レベルが小さいものから 2 個の撮影グリッドが選ばれる。図の例では、分解能レベル $2, 3, 4$ の一様格子が条件を満たすので、撮影グリッドとして分解能レベル $2, 3$ の一様格子が選ばれる。

### 5.2. カメラ位置と一様格子の選択

図 5 に示すように、カメラとシンク粒子間の距離に応じて、撮影される（可視化される）一様格子が選ばれる。具体的にはカメラとシンク粒子間の距離が $d$ のとき、条件

$$0.8h \leq d \leq 4.1h$$

を満たす一様格子のうち、分解能レベルが小さいものから 2 個を選んだ。ある距離から一様格子を撮影するとき、分解能レベルが大きな一様格子は分解能が高く細密な可視化ができる反面、カメラから見た見込み角が小さいために一様格子の端が画角に入ってしまう。一方、分解能レベルが小さい一様格子は分解能が悪く粗い可視化になるが、広い領域を覆っているので一様格子の端が画角に入り込まず、カメラが分子雲コアに埋没したような画像が得られる。条件式はこれらのバランスを考慮している。

この処方箋に従うと、特定のカメラ位置で 2 枚の画像が生成される。5.4 節で述べる手法を用いて、この 2 枚の画像を合成し、解像度が異なる一様格子の可視化をなめらかに接続する。

### 5.3. 可視化のバッチ処理

可視化の処理を制御するために Python[2]スクリプトによるバッチ処理を行った。図 6 に概要を示す。Python スクリプト zoomseries.py はカメラ位置と撮影グリッドごとに設定ファイル inputParam.py を出力し、Paraview を子プロセスとして起動する。Paraview の子プロセスは `pvbatch` コマンドによって起動される。このコマンドでは Paraview の GUI のウ

[2] もっとも普及しているスクリプト言語のひとつ。https://www.python.org

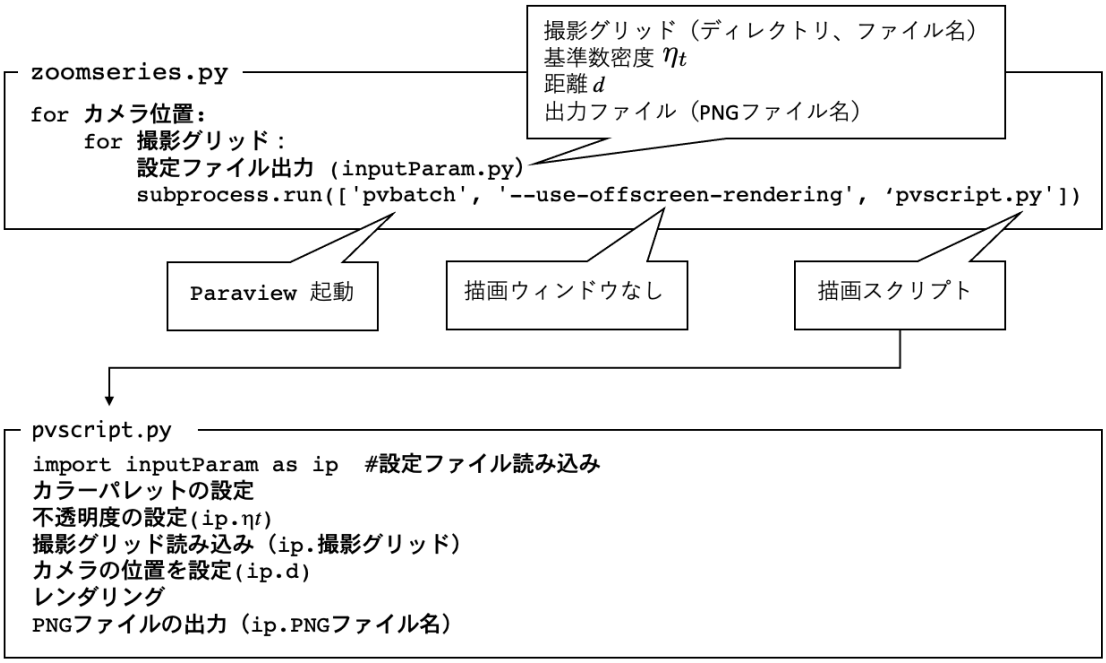


図 6 可視化の手続きと擬似コード。Python スクリプト zoomseries.py はカメラ位置と撮影グリッドごとに Paraview の子プロセスを起動する。設定ファイル inputParam.py を通じて各種パラメータを Paraview に渡す。

インドを表示しない。またコマンドラインオプション`--use-offscreen-rendering` を指定するとソフトウェアレンダリングが行われるので、バッチ処理には便利である。

Paraview のプロセスは、描画スクリプト pvscript.py を読み込む。このスクリプト内で、設定ファイルを読み込み、撮影グリッドを可視化し、画像ファイルを書き出す。これにより、カメラ位置と撮影グリッドごとに画像ファイルが作られる。ひとつのカメラ位置には最大 2 個の撮影グリッドがあるため、ひとつのカメラ位置に画像ファイルが最大 2 個生成される。

描画スクリプトの作成においては、スクラッチからスクリプトを書くのではなく、Paraview のトレース機能を用いて雛形を作り、雛形を改造するのが簡単な方法である。具体的には Paraview の Tools>Start Trace でトレースを開始し、モジュールやカメラを操作し、Tools>Stop Trace でトレースを終了する。トレース中の操作がトレース終了後に Python コードとして表示される。また、改造した Python スクリプトは Tools>Python Shell>Run Script で対話的に実行することができる。これはデバッグに役立つだろう。

描画スクリプト作成における注意点を述べる。動画を 1080p で作成するために以下のように描画サイズを指定する。

```
renderView1.ViewSize = [1920, 1080] #1080p
```

ボリュームレンダリングは不透明度を指定するパラメータ「オブジェクト名`.ScalarOpacityUnitDistance`」を用いる。このパラメータはガスが不透明になる物

理的な長さを与える。このオブジェクト名はボリュームレンダリングを行う対象を表す。このパラメータが大きいとガスが不透明になる長さが長くなり、ガスはより不透明になる。逆にこのパラメータが小さいとガスが不透明になる長さは短くなり、ガスはより透明になる。今回の可視化では、カメラが原始星に向かって移動している間にカメラと原始星の間のガスの不透明度がおおよそ一定になるようにしたい。そこで、このパラメータを距離 $d$（distance）に比例させて、

```
オブジェクト名.ScalarOpacityUnitDistance = distance/2/129
```

とした。この式の係数は実験的に求めた。またカメラの位置を

```
renderView1.CameraPosition
        = [renderView1.CameraFocalPoint[0],
           renderView1.CameraFocalPoint[1] - distance,
           renderView1.CameraFocalPoint[2]]
```

として距離に依存させた。同様に

```
renderView1.CameraParallelScale = distance/3.
```

として距離に比例させた。描画スクリプトの最後に画像ファイルを出力するコマンドを挿入する。例えば次のようなコマンドである。

```
SaveScreenshot(ip.pngfile, magnification=1, quality=100,
view=renderView1)
```

`ip.pngfile` は画像のファイル名で、`inputParam.py` からインポートされる。

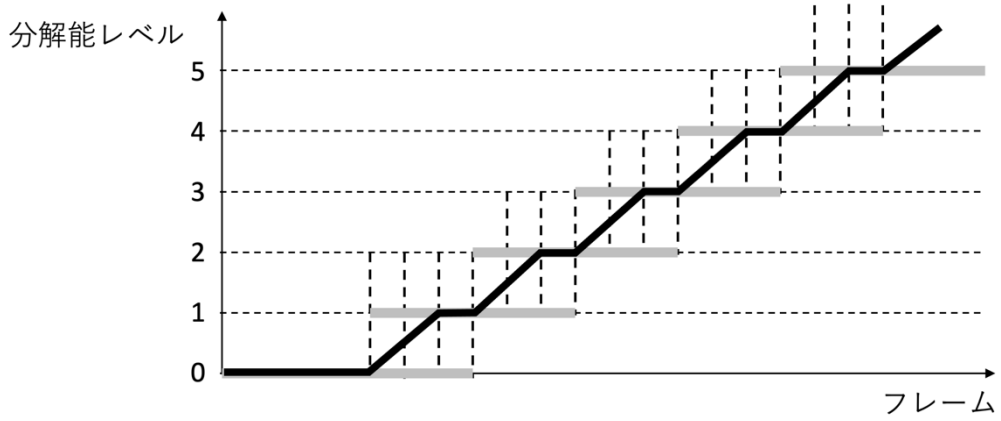


図 7 分解能レベルの画像の合成する処方箋を示す模式図。グレーの線は Paraview で出力された画像ファイルを示す。各フレームは最大 2 個の異なる分解能レベルの画像を持つ。黒線は異なる分解能レベルを線形補間するときの重みを表す。たとえば分解能レベルが 2 と 3 の中間にあるフレームでは、分解能レベル 2 と 3 の画像を線形補間する。補間の重みはグラフ上で分解能レベルまでの距離に比例する。

### 5.4. 異なる解像度画像の合成

前節のバッチ処理によってフレームごとに最大 2 個の異なる分解能レベルの画像ファイルが出力される。異なる分解能レベルをなめらかに乗り継ぐために、異なる分解能レベルの画像の線形補完を行う。

図 7 に合成の方法を示す。初期のフレームでは分解能レベルが 0 の画像だけが存在する。フレームが進むと分解能レベル 0 に加えて一段細かい解像度（分解能レベル 1）の画像が存在するようになる。この可視化事例ではフレーム 10 から 34 で分解能レベル 0 と 1 の 2 枚の画像が重複して存在する。重複する期間のはじめの 2/3 のフレームにおいて、2 枚の画像を線形補間によって合成する。線形補間の重みを線形に変化させて、分解能レベル 0 から 1 へ連続的に変化させる。重複する期間の残りのフレームは分解能レベル 1 を重み 100% として採用する。その後の分解能レベル 1 と 2 が重複するフレームでは、分解能レベル 1 と 2 について同様の線形補間を行う。この操作を順次繰り返し、最後のフレームまで線形補間による合成を行う。

重複する期間における重みの関数は、発見的に定められたものであり、可視化事例に依存する。今回の可視化事例では重複する期間のうちはじめの 2/3 で線形補間による合成を用いたが、重複する全期間に渡って線形補間を施すことも可能である。この場合にはすべてフ

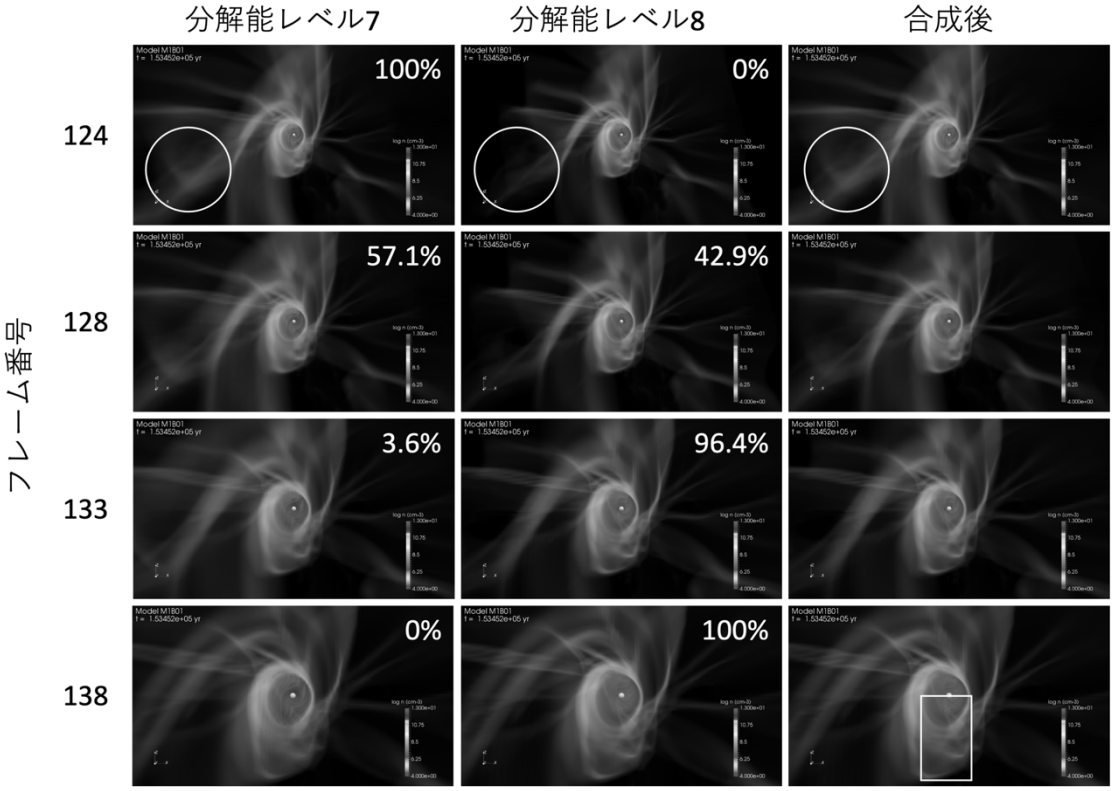


図 8 分解能レベルが 7 から 8 に移り変わるときの画像の合成の例。上から下にフレーム番号 124, 128, 133, 138 の画像を表示し、左から右に分解能レベルが 7 と 8 の画像とこれらを合成した画像を表示している。それぞれの画像右上の割合は合成における重みを示す。

レームで何らかの線形補間が施される。その結果、すべてのフレームに低解像度の画像の寄与が現れ、動画全体で画像のシャープさが低減すると考えられる。逆に、線形補間による合成の期間を短くすると、画像はシャープになる傾向にあるが、分解能レベル間の遷移が急になり、拡大のなめらかさが低減する。

この可視化事例では、線形補間の重みは画像全体にわたり空間的に一定としたが、重みを空間的に可変にすることもできる。たとえば、画像の中心領域は高解像度の画像の重みを高くして、周辺領域は低解像度の重みを高くする方法もある。これにより、原始星方向の構造は細密に描画でき、同時に周辺に広がるガスも描画することができる。より根源的な方法としては、一様格子のデータにより高解像なデータを合成してから可視化する方法がある。しかし本稿のように可視化後の画像を合成するほうが、一様格子を合成するよりも計算コストが低く、より簡単な方法である。

図 8 に分解能レベル 7 と 8 が重複するフレームにおける合成の例を示す。フレーム 124 から 138 には分解能レベル 7（低解像度で広い領域のデータを可視化した画像）と 8（高解像度で狭い領域のデータを可視化した画像）が 2 枚存在する。これらを合成することにより、分解能レベル 7 から 8 へなめらかに遷移する。

フレーム 124 では円で囲まれた領域のフィラメント状の構造が分解能レベル 7 の画像に写っているが、分解能レベル 8 の画像には写っていない。これはフィラメント状の構造が手前に伸びており、分解能レベル 8 では一様格子の領域の外にフレームアウトしてしまったためである。分解能レベル 7 でも中心部の解像度は動画目的としは十分である。分解能

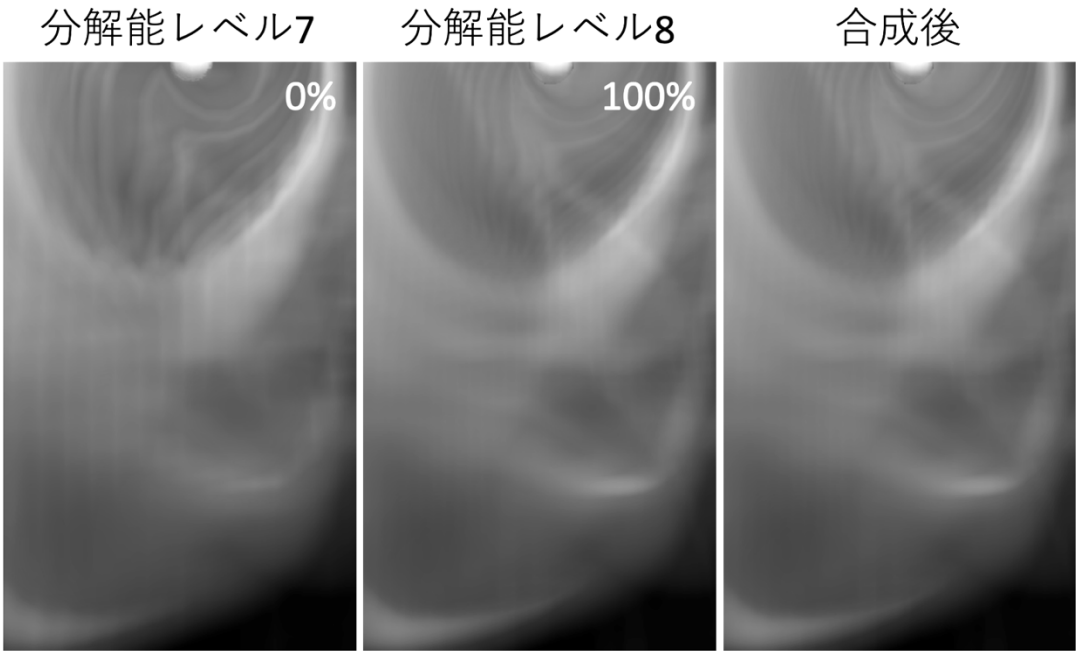


図 9 フレーム 138 の合成の拡大図。図 8 の矩形で囲まれた領域を拡大して表示している。図の右上の割合は線形補間で用いた重みである。

レベル 7 が 100%の重みなので、合成後の画像は分解能レベル 7 と同一になる。

フレーム 138 では分解能レベル 7 の重みが 0%で分解能レベル 8 の重みが 100%である。分解能レベル 7 の画像では低分解能のために格子の形状が目立ち、星周円盤外周部の細密な構造が表現できていない（図 9 を参照）。一方、高解像な分解能レベル 8 でも周辺部に広がったガスは描画されている。したがって、合成では分解能レベル 8 の重みが 100%であることが妥当である。

### 5.5. 動画のエンコード

前節で合成した画像はフレームごとに 1 枚存在する。これらの画像を、ffmpeg[3]を用いて動画にエンコードした。フレームレートを毎秒 15 フレーム（FPS=15）としたので、150 フレームの画像は 10 秒間に動画になる。

動画形式（コンテナ）を MP4 とした。コーデクは H.264 である。

エンコードするスクリプトでは、映像のビットレートを 24Mbps と指定した。このビットレートはフレームレートが毎秒 15 フレームの 1080p の動画としては大きい。YouTube[4]によると、アップロードする動画のおすすめのビットレートは、1080p で毎秒 24〜30 フレームの場合でも映像ビットレートを 8〜10Mbps としている。一般に、シミュレーションの動画はビデオカメラで撮影した動画よりもコントラストが高く、高いビットレートを要求する。今回は比較的大きなビットレートを用いることで動画の劣化を低減し、動画を静止したときにも鮮明な映像になるようにした。実際には 2 パスエンコードを行ったため、エンコードした動画のビットレートは最終的に 5.26Mbsp まで減った。

## 6. まとめと議論

本論文では、空間的に広いダイナミックレンジを持つ AMR シミュレーションの可視化の方法を述べた。計算領域に注目する天体が 1 個存在するため、伝統的な可視化の表現であるズームインを採用した。分子雲コアから出発して、カメラが分子雲コアの内部に進入して、原始星と星周円盤の近くまで到達する様子を可視化した。

シミュレーションでは、分子雲コアが重力収縮して、内部に原始星を模したシンク粒子とその周囲に星周円盤が形成する様子を、AMR 法を用いて再現した。可視化するデータには AMR 格子そのものではなく、AMR 格子から切り出された一様格子を用いた。

可視化ソフトェアには Paraview を用いた。Paraview を制御するために Python スクリプトを作成した。この Python スクリプトは、各フレームにおけるカメラの位置や読み込みファイル名などの各種パラメータを Paraview に受け渡し、Paraview を起動して画像ファイル

---

[3] 動画をエンコードするためのオープンソースのソフトウェア。https://ffmpeg.org

[4] YouTube アップロードする動画におすすめのエンコード設定
https://support.google.com/youtube/answer/1722171?hl=ja

（PNG ファイル）をフレーム毎に出力する。ズームインにおいて異なる解像度の一様格子をなめらかに乗り継ぐために、１フレームにつき２個の異なる解像度の一様格子を可視化し、これらの画像を線形補間を用いて合成した。

本稿ではカメラを１箇所に向かって移動させて対象を拡大する動画を作成した。カメラの動きを逆転させると、対象から遠ざかる動画を作成できる。また、カメラの位置と向きを回転させると、対象が回転する様子を撮影することもできる。カメラの軌跡とカメラの方向をうまく指定すれば、計算領域を自在に撮影することがきる。

本稿ではシミュレーションのスナップショットを用いた。この方法を応用するとシミュレーションの時系列データを用いて、天体が時間進化しながらカメラが計算領域の内部を動き回る動画が作成できる。

## 謝辞

この研究は科研費（18H05436, 18H05437, 17K05394）によって一部補助された。数値シミュレーションには国立天文台天文シミュレーションプロジェクト XC50 を用いた。このシミュレーションの実施は、アメリカ歴史博物館のプラネタリウムショーWorlds Beyond Earth (2020)の製作に筆者が参加したことが契機であった。またこの可視化手法を開発した動機は、筆者が ALMA のプレスリリース[5]に動画を提供するためであった。

## 参考文献

Berger, M. J. & Colella, P. 1989, Journal of Computational Physics, 82, 64. doi:10.1016/0021-9991(89)90035-1

Bonnor, W. B. 1956, MNRAS, 116, 351. doi:10.1093/mnras/116.3.351

Ebert, R. 1955, ZA, 37, 217

Kim, J.-hoon., Abel, T., Agertz, O., et al. 2014, ApJS, 210, 14. doi:10.1088/0067-0049/210/1/14

Larson, R.~B. 1981, MNRAS, 194, 809. doi:10.1093/mnras/194.4.809

Machida, M. N., Inutsuka, S.-ichiro., & Matsumoto, T. 2007, ApJ, 670, 1198. doi:10.1086/521779

Matsumoto, T. 2007, PASJ, 59, 1243. doi:10.1093/pasj/59.6.1243

Matsumoto, T., Dobashi, K., & Shimoikura, T. 2015, ApJ, 801, 77. doi:10.1088/0004-637X/801/2/77

Matsumoto, T., Machida, M. N., & Inutsuka, S-ichiro. 2017, ApJ, 839, 69. doi:10.3847/1538-

[5] 星の卵の「国勢調査」―アルマ望遠鏡が追う星のヒナ誕生までの 10 万年 https://alma-telescope.jp/news/press/taurus-202008

4357/aa6a1c
Matsumoto, T., Miyoshi, T., & Takasao, S. 2019, ApJ, 874, 37. doi:10.3847/1538-4357/ab05cb
Offner, S. S. R., Klein, R. I., McKee, C. F., et al. 2009, ApJ, 703, 131. doi:10.1088/0004-637X/703/1/131
Powers of Ten 1977, EAMES OFFICE LLC
Worlds Beyond Earth 2020, American Museum of Natural History
松本倫明 2007 シリーズ現代の天文学 14 シミュレーション天文学 第 12 章, 富阪幸治、花輪知幸、牧野淳一郎 編