

# 最近の粒子、流体の可視化事情

滝脇 知也(国立天文台)

# 可視化の例

4D2U

# なぜ可視化するのか？

- A) 専門家むけ(1D,2D)  
背後の物理を理解するため
- B) 非専門家むけ(3D)  
セットアップやダイナミクスを簡単に紹介するため
- C) 一般人むけ(3D)  
研究の魅力を分かってもらうため  
予算獲得のため

用途によりツールや必要となる知識・技法も異なってくる。  
どの用途に興味がありますか？

# 今回の発表の目的

以下に関する情報共有、問題発見

1. **どんなツールがあるのか？**
2. **どんな問題があるのか？**
3. **どんな勉強法があるのか？**

# どんなツールがあるのか？

A) 専門家むけ

matplotlib.lib

gnuplot

IDL

etc...

B) 非専門家むけ

VisIt

Paraview

AVS(有料)

KVS

etc...

C) 一般人むけ

POV-Ray

Unity

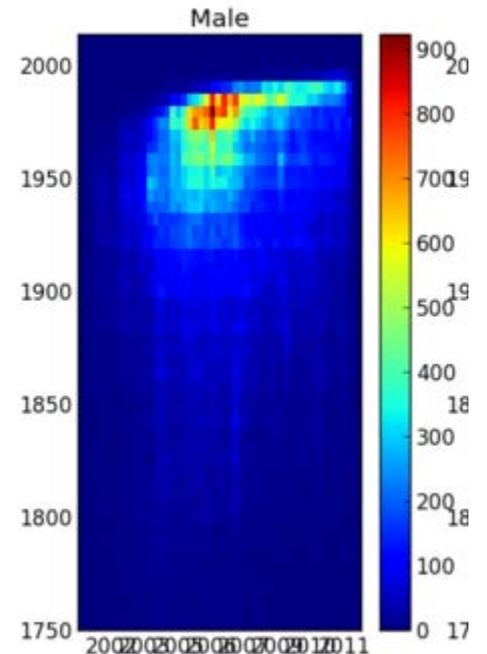
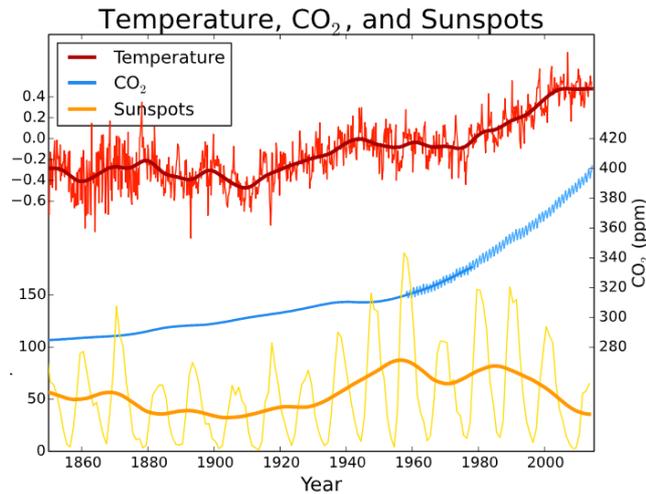
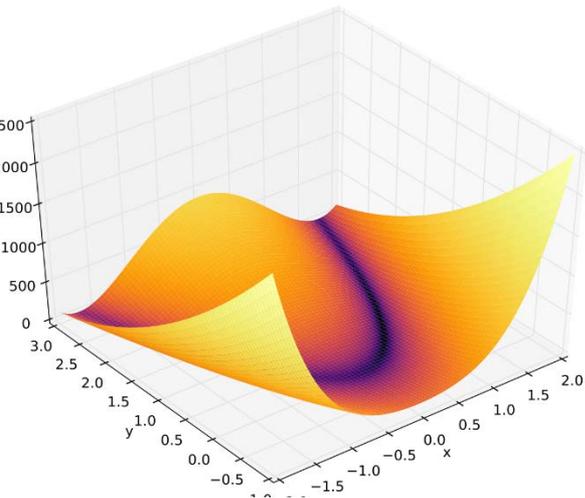
Zindaiji

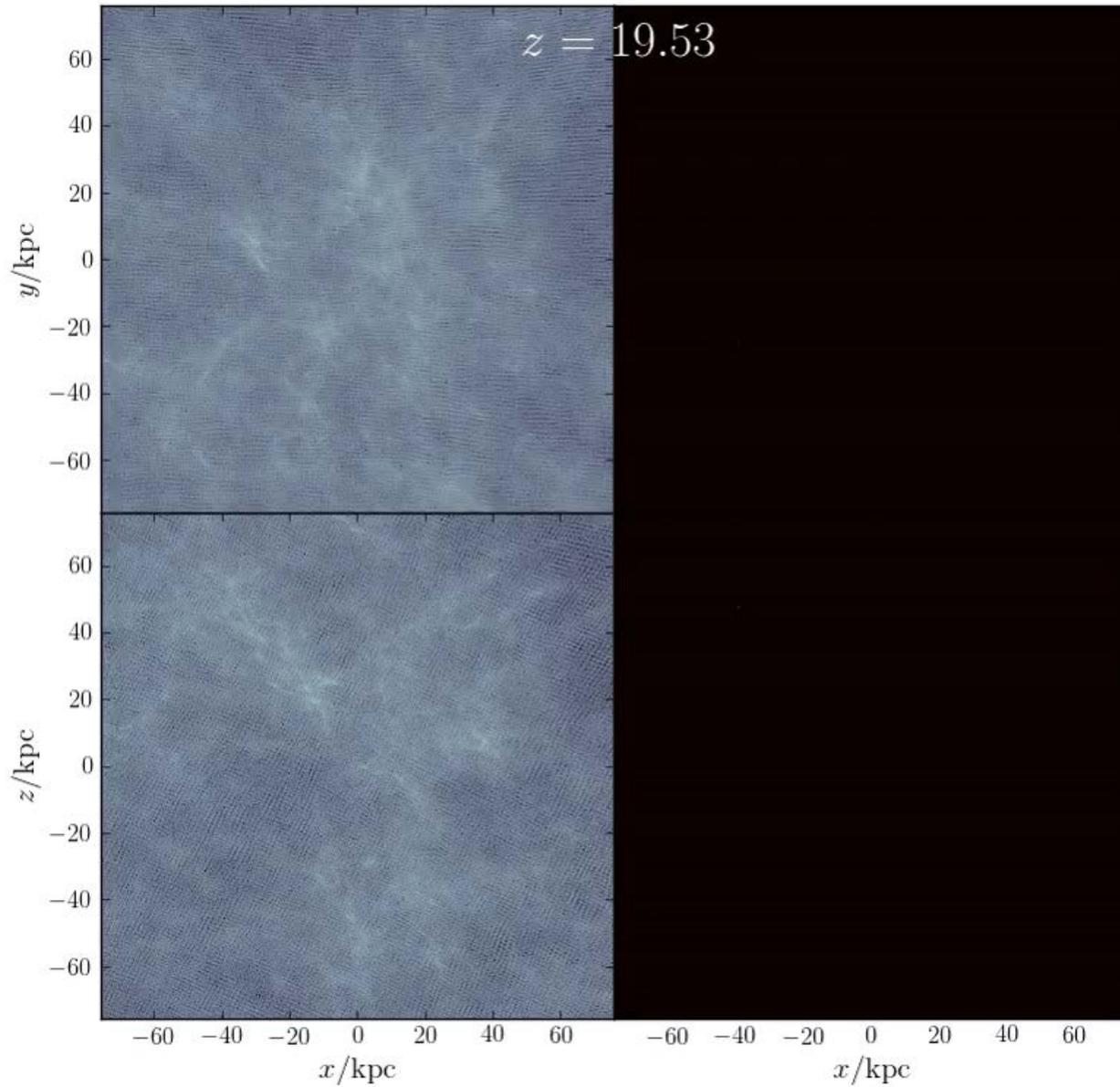
etc...

滝脇が経験のある赤字のやつだけ紹介

# matplotlib

Python baseのライブラリ。  
データ解析もプロットも同時にできる。  
凡例等にTeXの表現も使える。





matplotlib.lib

岡本さん提供

# matplotlib

スクリプトの例。雰囲気伝えるのみで実際には動きません。  
オブジェクト指向のプログラミングスタイル。

```
#!/usr/bin/python
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
vr[i][j]=.....
```

```
X,Y = np.meshgrid(phi,the) <=座標の情報
```

```
Z = vr <=プロットする変数
```

```
myfig = plt.figure()
```

```
..... <=図のタイトル  
myax.set_title(title)
```

```
myax.set_xlabel(r'$\phi$') <=図のラベル
```

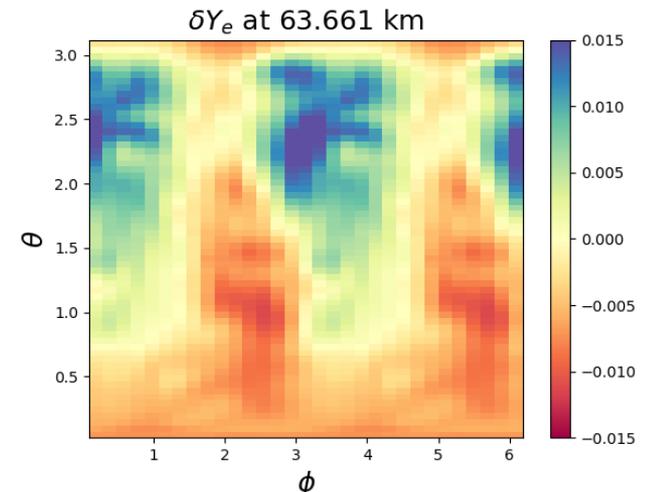
```
myax.set_ylabel(r'$\theta$')
```

```
myim = myax.contourf(X,Y,Z,cmap='bwr',interpolation='spline16') <=コンタワーと色と補間法
```

```
myim = myax.pcolor(X,Y,Z,cmap="bwr") <=pseudo colorと色マップと補間法
```

```
myim.set_clim(-0.015,0.015) <=描画するレンジ
```

```
myfig.savefig('vr.png') <=ファイルの保存
```

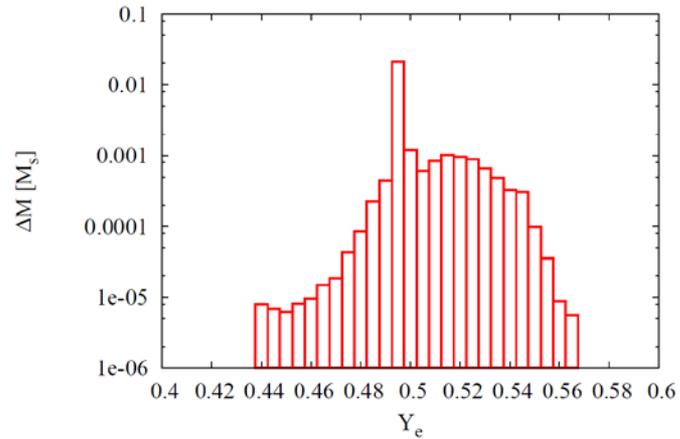
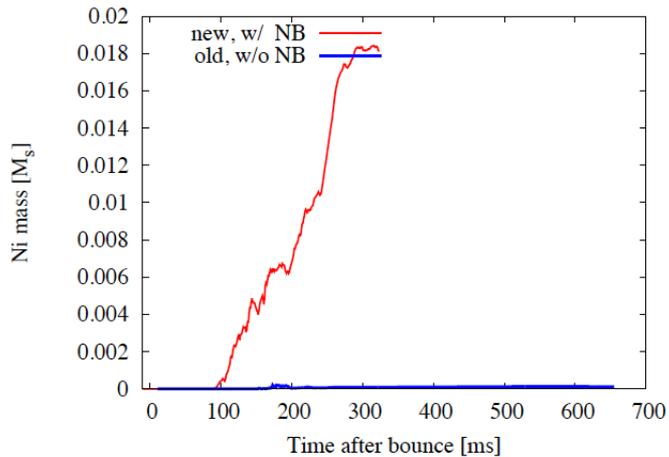


# gnuplot

シンプルなソフトウェア。

ただし、凝ると複雑なこともできる。

データ解析もできなくはないがやらないほうが良い。



# gnuplot

スクリプトの例。雰囲気伝えるのみで実際には動きません。

基本的には直観的。ただし、結構暗黙のトリックも多く、習熟度によって出来に差が出る

```
set xlabel "X [km]"  
set ylabel "Z [km]"  
set cbrange [1:30]
```

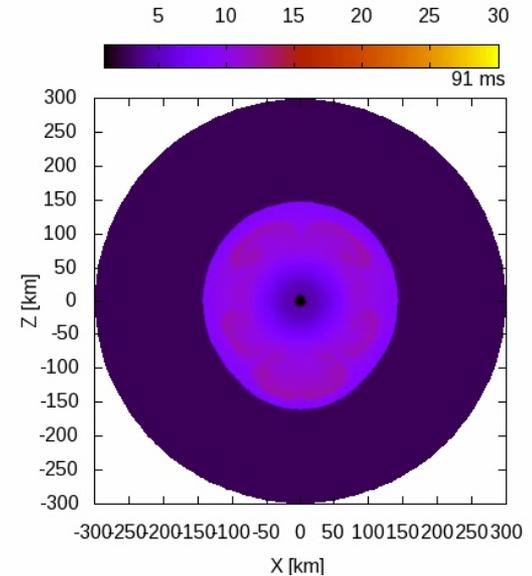
```
splot [-srange:srange][-srange:srange] ¥
```

```
  ifnames u ( $2*sin($1)):( $2*cos($1)):( $2<srange?$4:NaN) w pm3d ¥  
, ifnames u ( -$2*sin($1)):( $2*cos($1)):( $2<srange?$4:NaN) w pm3d ¥
```

2コラム目 x sin(1コラム目)をX軸に

2コラム目があるレンジ以内だったら  
4コラム目をプロットそうじゃないときはNaN

データファイル名。2次元配列で1次元目が終わったところで空行をはさむ



# どんなツールがあるのか？

A) 専門家むけ

matplotlib.lib

gnuplot

IDL

etc...

B) 非専門家むけ

VisIt

Paraview

AVS(有料)

KVS

etc...

C) 一般人むけ

POV-Ray

Unity

Zindaiji

etc...

滝脇が経験のある赤字のやつだけ紹介



# VisIt

VisIt 2.0.0

File Controls Options Windows PlotAtts OpAtts Help

Global

Active window 1  Auto apply

Selected Files

- localhost
  - /afs/ipp-garching.mpg.de/home/m/mjr
  - data2/mjrdata
    - 201: /HOTB/data/b0123dDZ\_\*.silo database
      - b0123dDZ\_0492.silo cycle 98398 time 352
      - b0123dDZ\_0650.silo cycle 0650 time ?
      - b0123dDZ\_0655.silo cycle 0655 time ?**
      - b0123dDZ\_0656.silo cycle 0656 time ?
      - b0123dDZ\_0658.silo cycle 0658 time ?

ReOpen Replace Overlay

Time

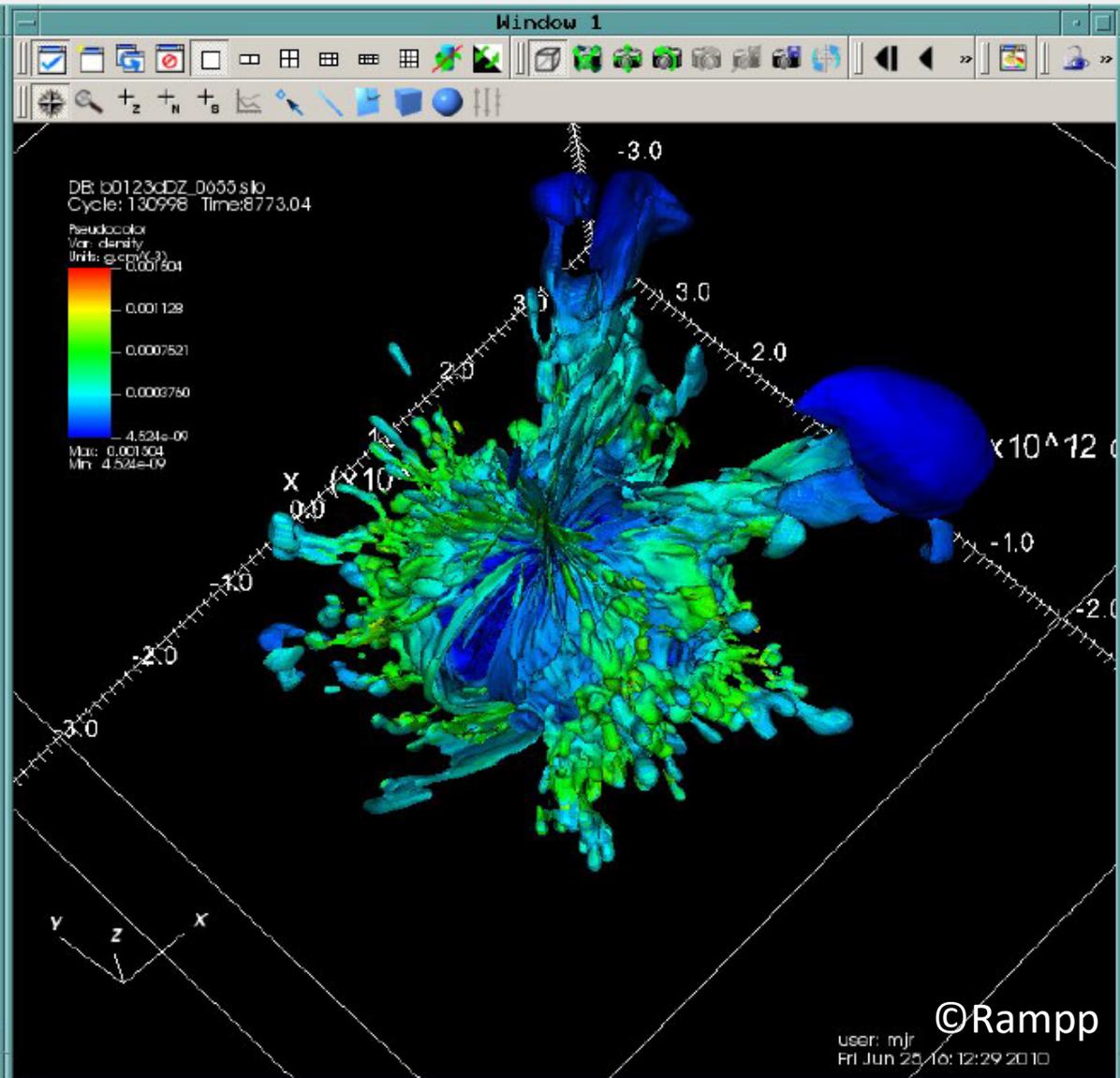
0655

Plots

Add Operators Delete Hide/Show Draw Variables

- 201: b0123dDZ\_\*.silo database: density
  - Transform
  - Isosurface**
  - Pseudocolor
- 201: Mesh - Transform(grid) (hidden)

Apply operators /  selection to all plots



# Visit python

PythonからVisitを呼ぶこともできる。

```
import visit
```

```
visit.Launch()
```

```
visit.AddPlot("Pseudocolor", "Entropy")
```

```
visit.AddOperator("Transform")
```

```
c = visit.TransformAttributes()
```

```
c.transformType = c.Coordinate
```

```
c.inputCoordSys = c.Spherical
```

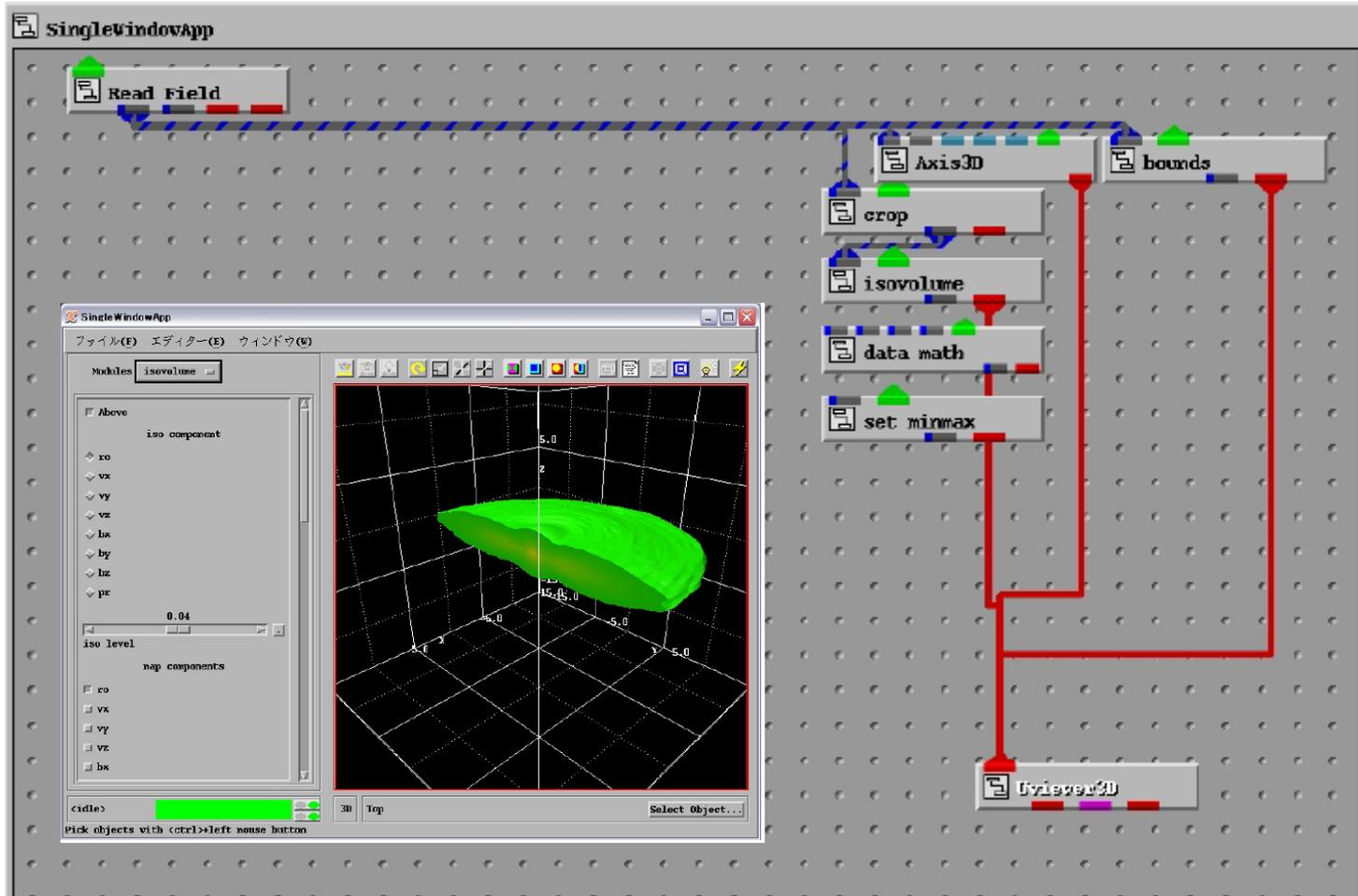
```
c.outputCoordSys = c.Cartesian
```

```
visit.SetOperatorOptions(c)
```

*AVS*

**From 1D to 3D**

# AVS



# VisIt/Paraview/AVS

- たぶんできることはだいたい同じ
- AVSは開発が止まっている。  
滝脇はVisItを薦める。  
Pythonから呼べるのも便利

# どんなツールがあるのか？

A) 専門家むけ

matplotlib.lib

gnuplot

IDL

etc...

B) 非専門家むけ

VisIt

Paraview

AVS(有料)

KVS

etc...

C) 一般人むけ

POV-Ray

Unity

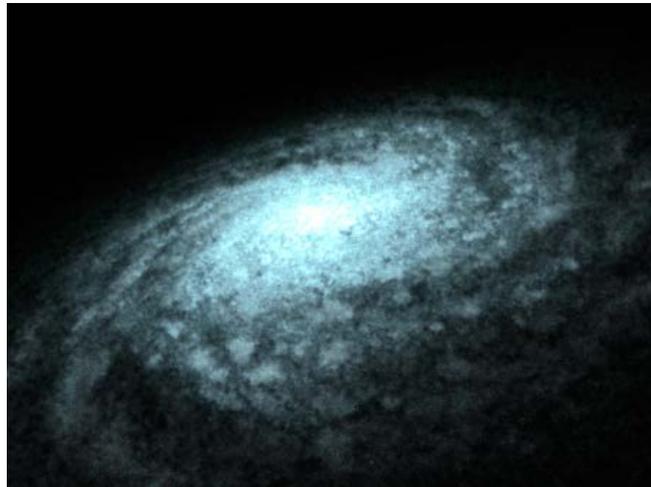
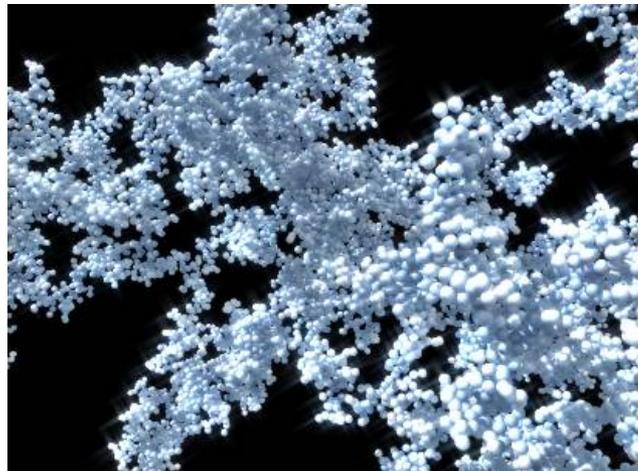
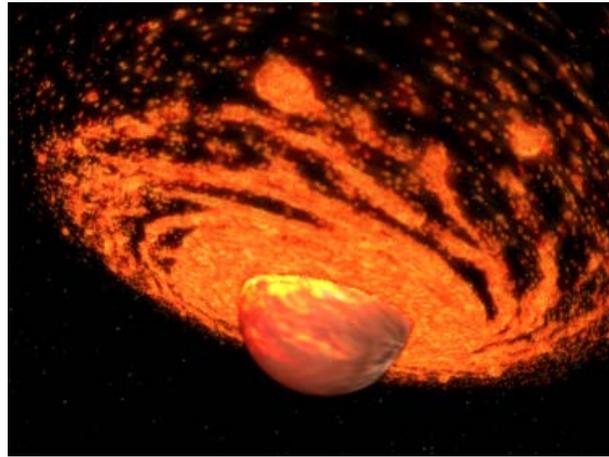
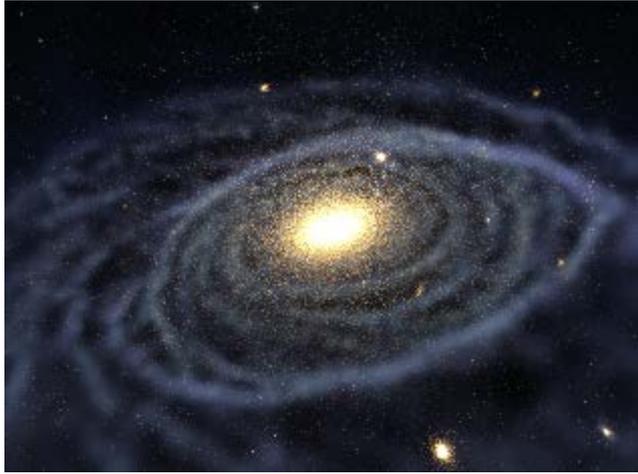
Zindaiji

etc...

滝脇が経験のある赤字のやつだけ紹介

# Zindaiji

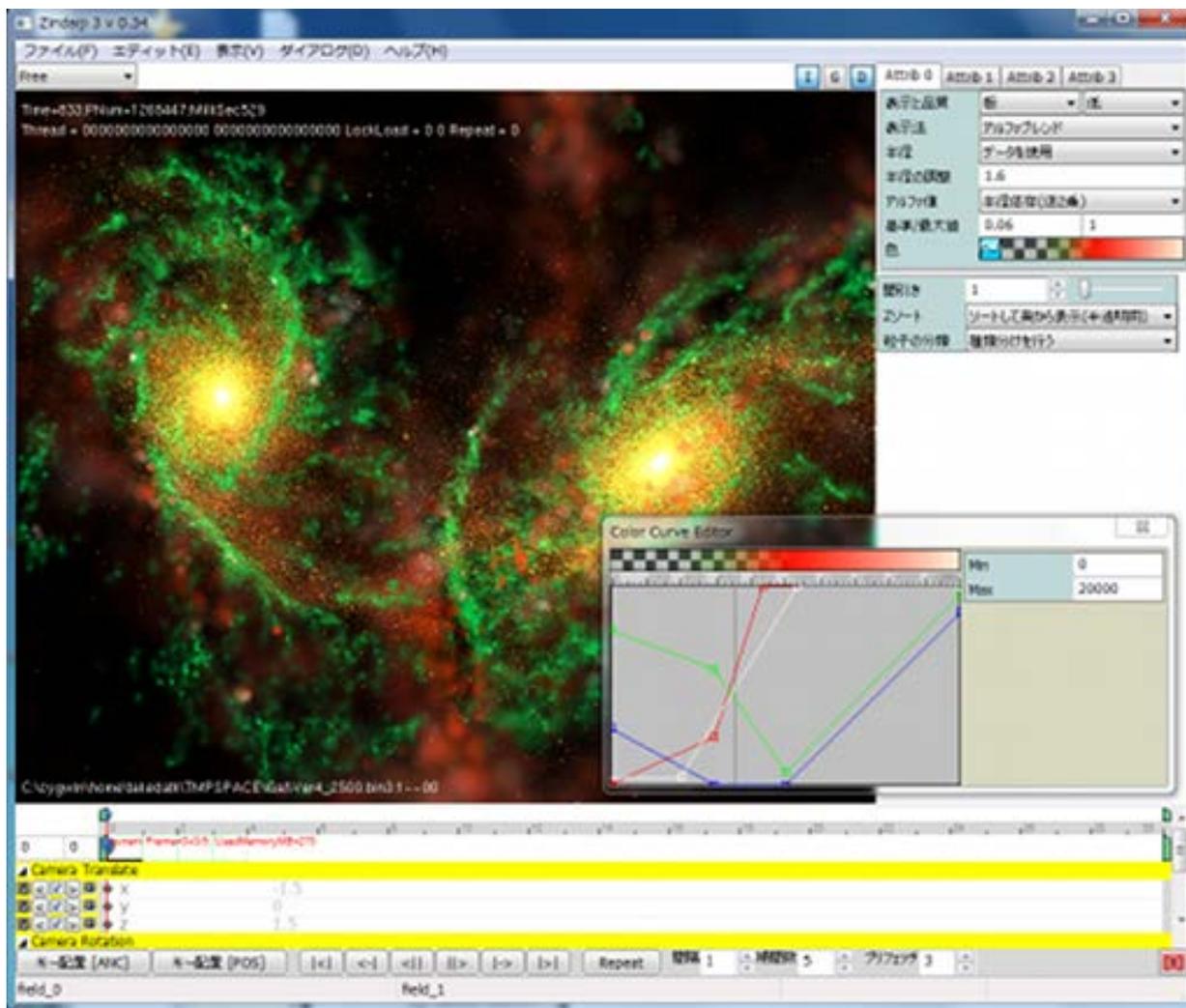
粒子系の可視化に



武田さん提供

# Zindaiji

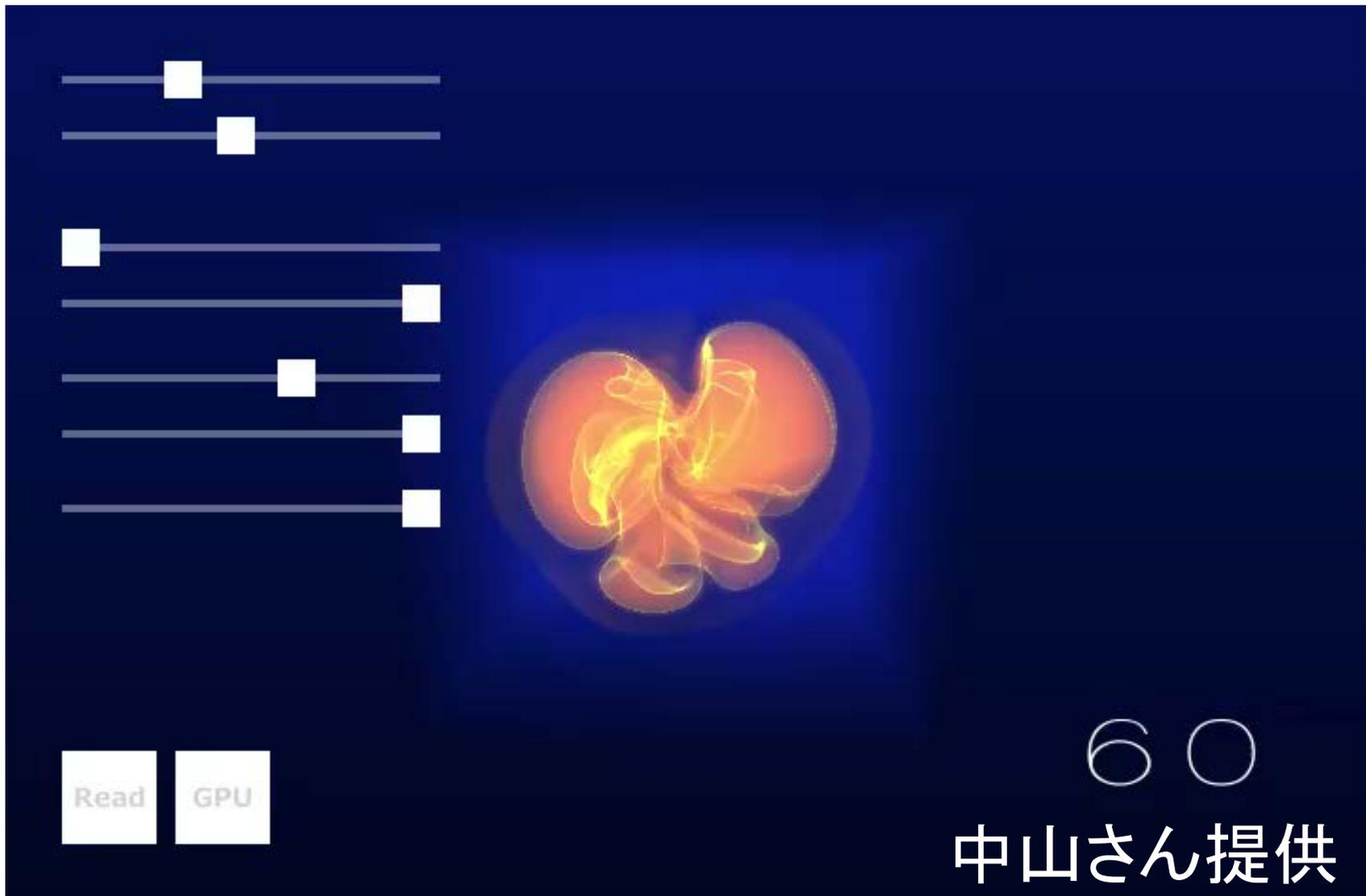
粒子系の可視化に



武田さん提供

# Unity

Unityはゲームを作る環境。Unityで可視化ソフトを作る感じのようだ。



# 今回の発表の目的

以下に関する情報共有、問題発見

1. どんなツールがあるのか？
2. **どんな問題があるのか？**
3. どんな勉強法があるのか？

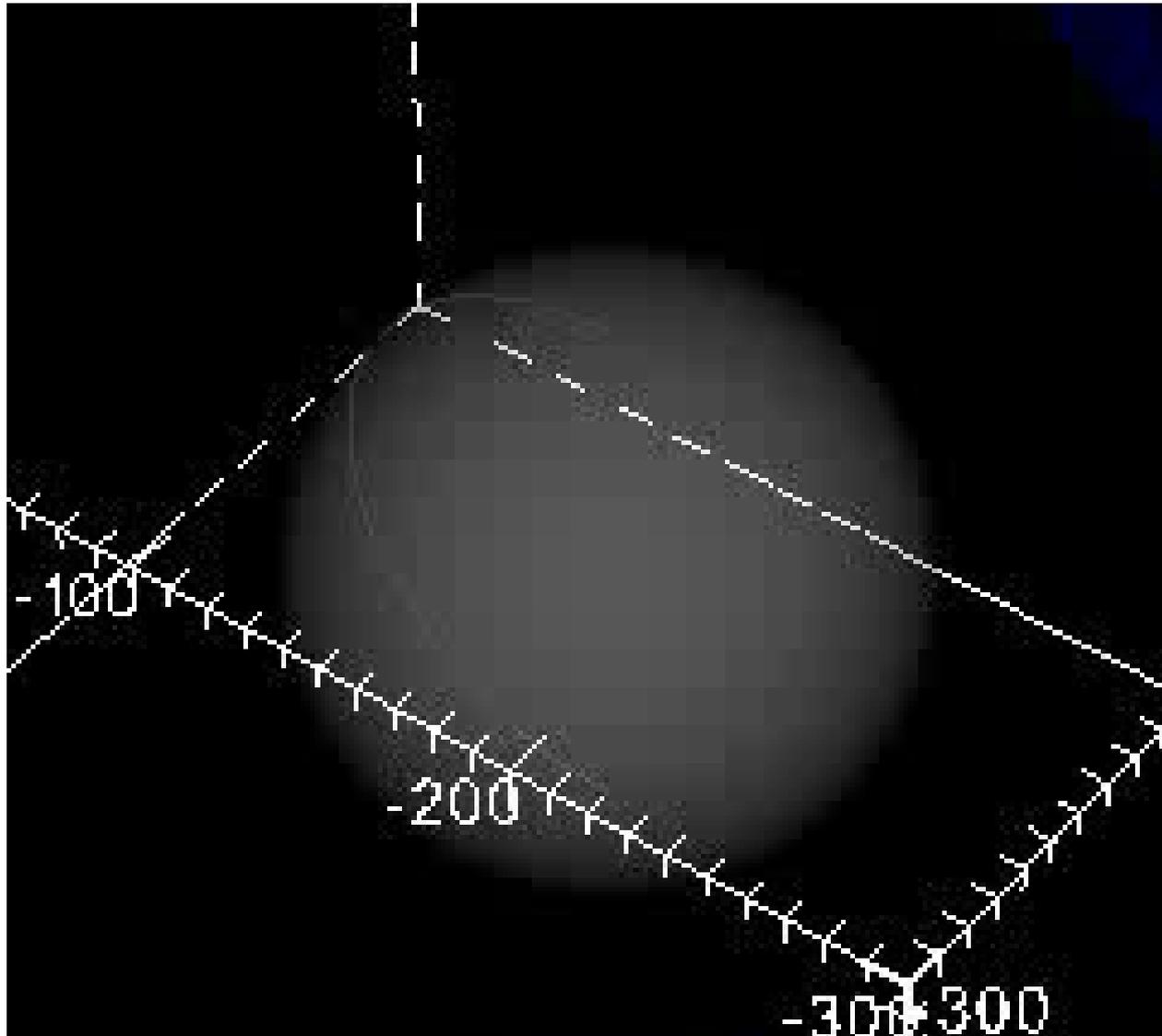
# どんな問題があるのか？

例えば...

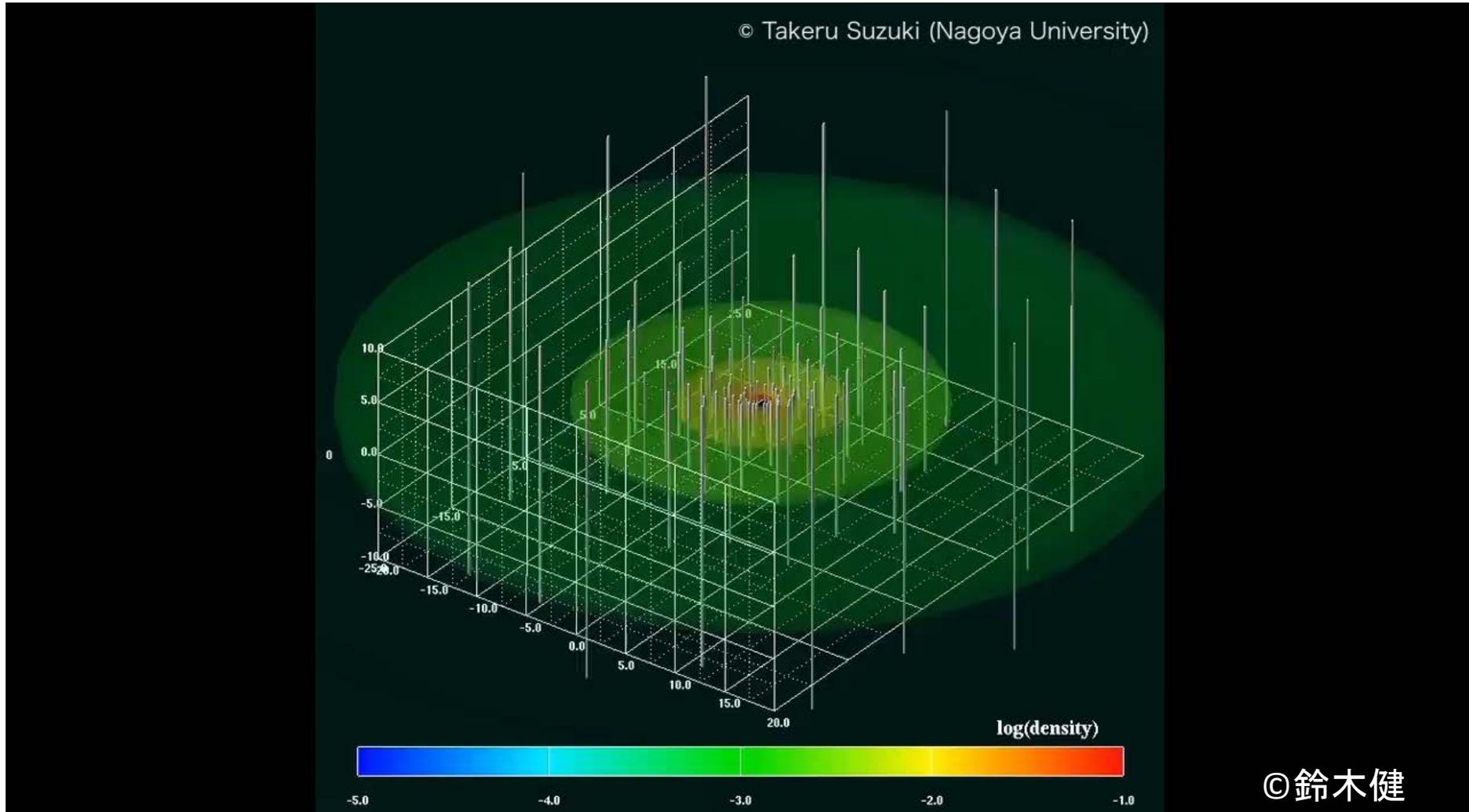
- 見た目が思い通りにならない
- 計算が重い
- メモリが大量に必要
- データの読み込みが難しい
- コンタワーが直感に反する問題
- テンサーやベクトルの可視化

この勉強会のように問題を共有する取り組みは稀

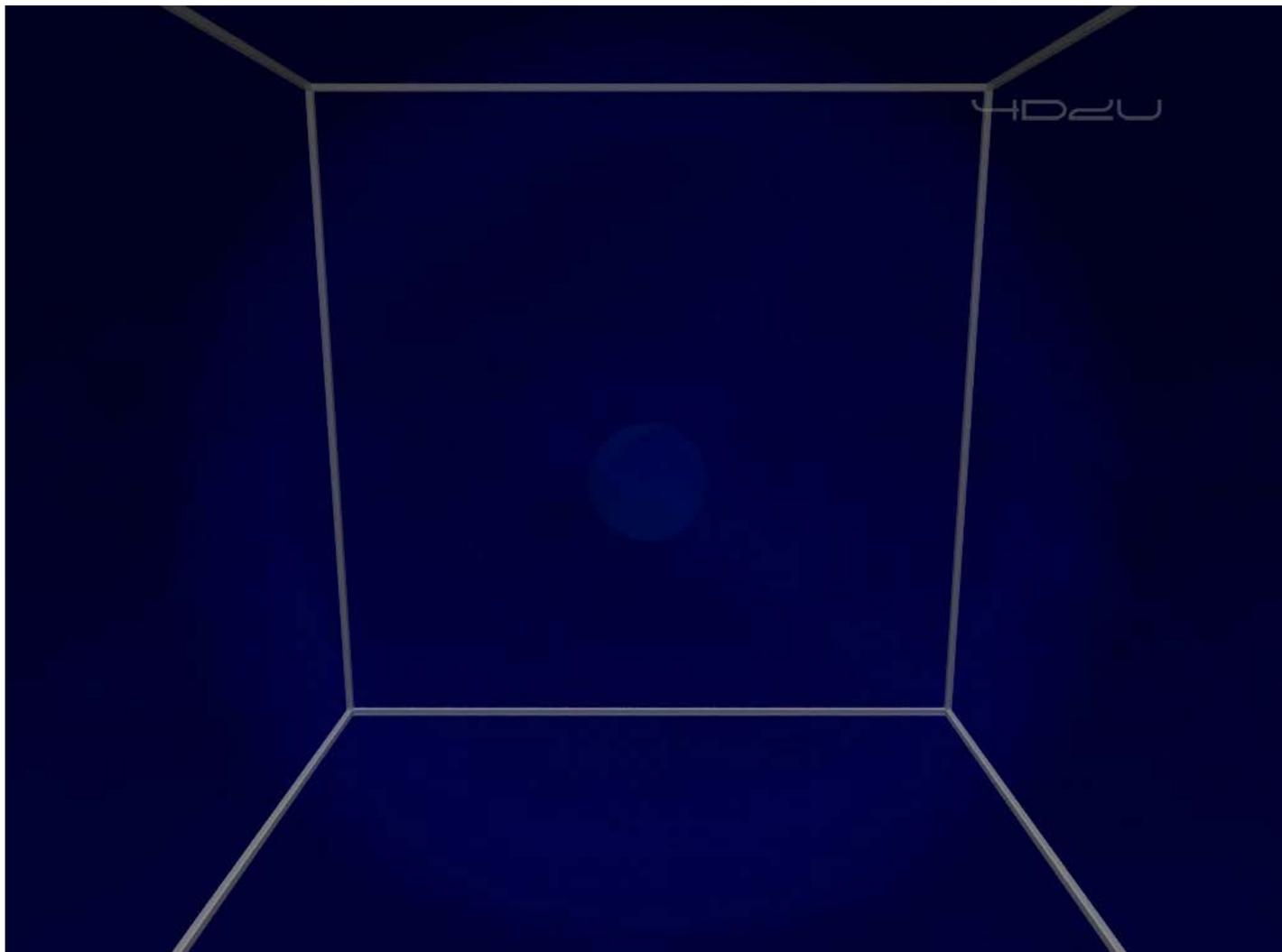
# 見た目が思い通りにならない 球座標のつなぎ目



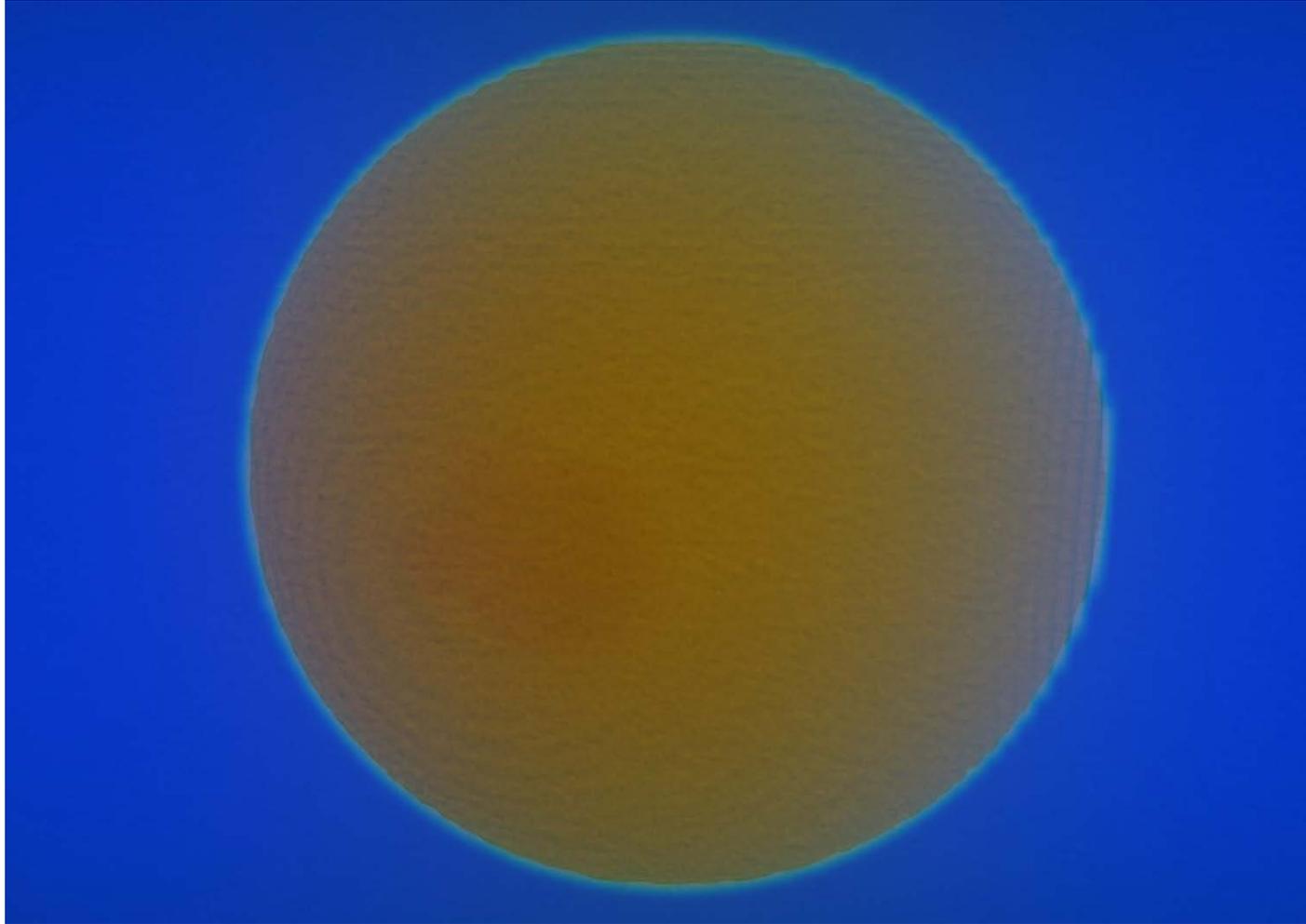
# 見た目が思い通りにならない 流線、磁力線のつけね問題



# コンタワーが直観に反する



# コンタワーが直観に反する



メッシュ法のデータを粒子で可視化？

# データの読み込みが難しい

バイナリデータでどこからどこまでを読んで3次元格子上のデータと認識するのか？

AVSはサブルーチンで計算した。手計算でもできるが、間違いがよく起こる。

VisItではhdf5形式をXDMPFで読み込む。

# Hdf5のデータの書き込み

```
rank=3
```

```
dims(1) = imax; dims(2) = jmax; dims(3) = kmax
```

```
dsetname="/rho"
```

```
call hdf5OUT(file_id,rank,dims,dsetname,V3DCENF(1,1,1,1))
```

```
  dsetname="Density"
```

```
  write(unitxmf,'(a)')    <Attribute Name="//trim(dsetname)
```

```
&      //" AttributeType="Scalar" Center="Cell">'
```

```
  write(DIM3D,"(I0,1x,I0,1x,I0)") kmax,jmax,imax
```

```
  write(unitxmf,'(a)')    ' '
```

```
&      //"<DataItem Dimensions="//trim(DIM3D)//"'
```

```
  write(unitxmf,'(a)')    ' //
```

```
&      'NumberType="Float" Precision="4" Format="HDF">'
```

```
  write(unitxmf,'(a)')    '//trim(fname)//"/rho"
```

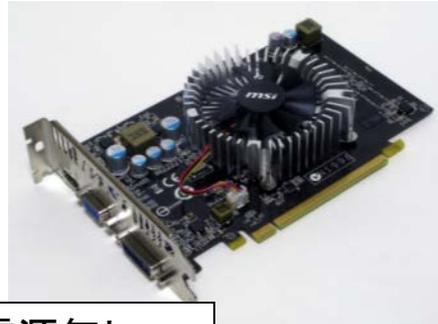
```
  write(unitxmf,'(a)')    </DataItem>'
```

```
  write(unitxmf,'(a)')    </Attribute>'
```

多粒子系の可視化は  
一般にメモリや計算時間との闘いに

# レンダリングの種類と描画時間(1)

## ■一秒当たりの描画粒子数目安



2010年ごろ補助電源無し  
の中で速いやつ(推奨300W)



2011年ごろ補助電源あり  
6pin x 2で速いやつ(推奨500W)

武田さん提供

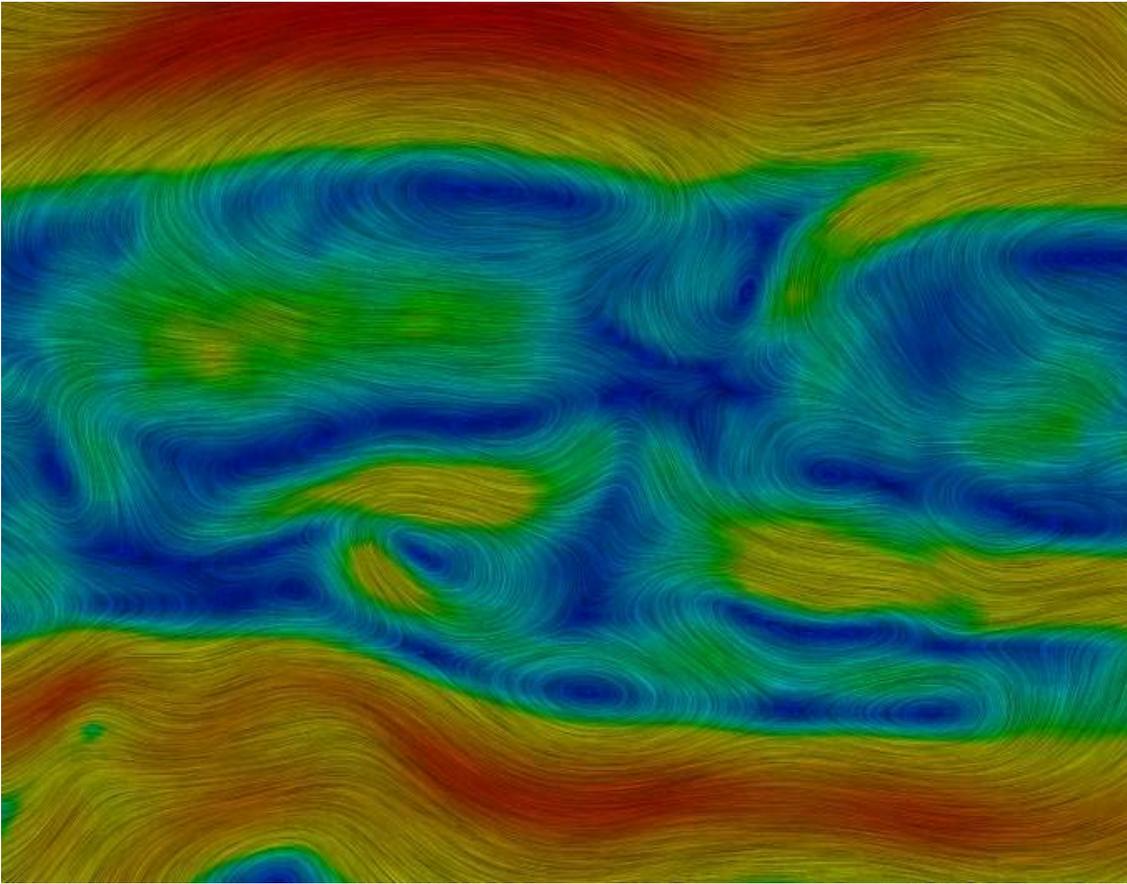
| 描画タイプ     | GT240 | GTX560 | 粒子数目安  |
|-----------|-------|--------|--------|
| GL_POINTS | 4千万   | 1億     | ~10億   |
| ビルボード     | 2000万 | 4000万  | ~4億    |
| SPH粒子と星   | 200万  | 200万   | ~2000万 |
| 球による表示    | 200万  | 300万   | ~3000万 |
| 影つきの表示    | 10万   | 10万    | ~100万  |
| 複数の影      | 5万    | 5万     | ~50万   |

注)シーンによって、  
数値は相当変動します

# ベクトルやテンソルの可視化

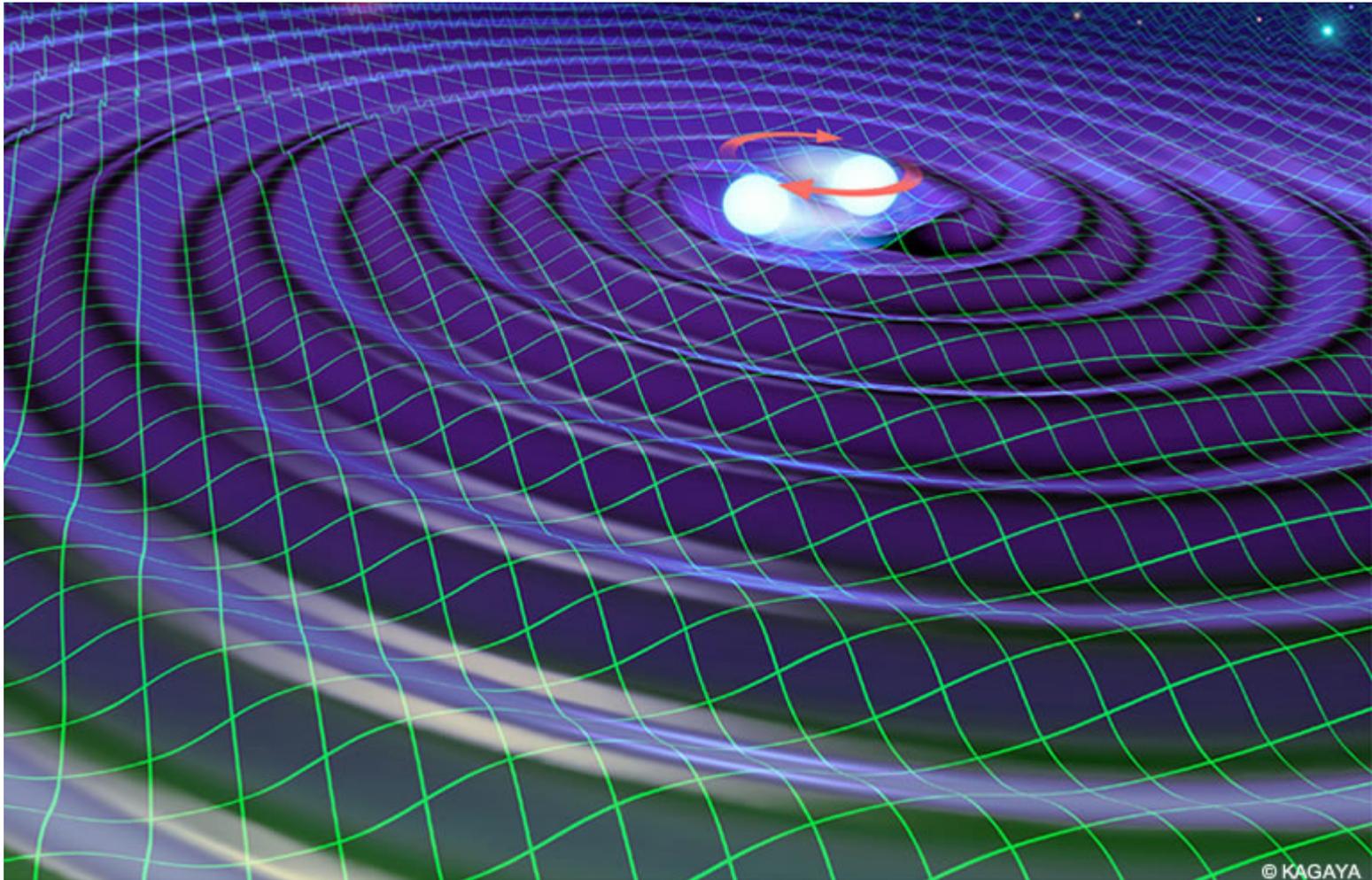
ベクトル、流線以外の方法は？

Line integral convolution



<http://www.zhanpingliu.org/research/flowvis/lic/BasicLIC/BasicLIC.htm>

# ベクトルやテンソルの可視化



重力波は本当はテンソル。このイメージで大丈夫？

# 今回の発表の目的

以下に関する情報共有、問題発見

1. どんなツールがあるのか？
2. どんな問題があるのか？
3. **どんな勉強法があるのか？**

# どんな勉強法がある？

- WEBの検索  
一番基本的だが、意外と目的のものが見つからない。
- メンバー関数の定義を読む  
オブジェクト指向のpythonとかVisItだと有効。
- 講習会  
CfCAでもやっているが、講師不足。業者ではなく、その分野の可視化をしている人が良い。
- 教科書  
ツールの勉強じゃなく、可視化の勉強があっても良い？
- 情報共有の場を作る  
今回のような試みは重要。ただし、継続性や情報の蓄積はどうする？
- 他の分野に学ぶ(産業界や、CGの学科等)  
学ぶことは多そう。  
ただし、物理よりもCGによる、スコープのずれがあるので調整が必要だろう。

# 何に関心があるのか？

- A) 専門家むけ
- B) 非専門家むけ
- C) 一般人むけ

## •どんなツールがある？

- A: matplotlib
- A: gnuplot
- A: IDL
- B: VisIt
- B: paraview
- B: AVS
- B: KVS
- C: povray
- C: Unity
- C: Zindaiji
- C: 立体視

## •どんな問題がある？

- 見た目が思い通りにならない
- 計算が重い
- メモリが大量に必要
- データの読み込みが難しい
- コントラールが直感に反する問題
- テンサーやベクトルの可視化

## •どんな勉強方法がある？

- WEBの検索
- メンバー関数の定義を読む
- 講習会
- 教科書
- 情報共有の場を作る
- 他の分野に学ぶ(産業界等)