

目で視る相対論 3. 歪む宇宙

福 江 純*

1. はじめに

光速に近い速度で飛翔する宇宙船から見ると、宇宙はどう見えるだろうか？ という話を2回にわたって紹介した。相対論的現象を目で視るという観点からすれば、本来なら、もっといろいろと特殊相対論的現象の視覚化（例えば、準光速宇宙船と地球との時間の違い、準光速宇宙船の時空航路図、相対論的宇宙ジェットの見え方など）を紹介しなければならないところだろうが、なにせ材料の手持ちがない。そこでいきなりブラックホール周辺の一般相対論的現象へと話を進めさせていただく。

今回はまず手始めに、ブラックホール周辺の曲がった空間を視覚化することに挑戦してみよう。

なお相対論やプログラミングに関して：

①初心者の方……3節へ跳んで下さい。記事を読んでなんとなく分かったような気になればしめたものです。パソコン(PC 9801 F2 コンパチ)が手近にあれば、プログラムを入力して、パラメータをいじってみて下さい。

②初心者+αの方……2節も読んで下さい。もっと詳しくは適当な相対論の入門書を見て下さい。

③中級者の方……自分でもっとよいプログラムを作成して筆者まで送って下さるようお願いいたします。

④上級者の方……本号の次の記事へ進んで下さい。

2. 曲がった空間

数立方パーセク内には水素原子一つないような空間に（これはものごととで、少しぐらい希薄なガスがあっても構わないが）、質量 M の球対称の天体を置いてみよう。太陽でも中性子星でもいいが、一応ブラックホールだとしよう。

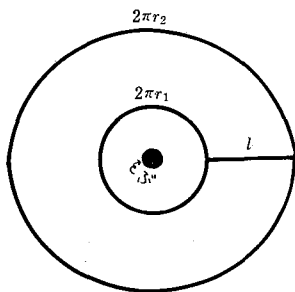


図1 ブラックホール周辺の距離 l 。

さて、このブラックホールの近くで、ブラックホールの中心を中心とする円の円周の長さを計ると $2\pi r_1$ 、少し外側で計ると $2\pi r_2$ だったとする（図1）。

もし空間が平らであれば、2つの円周の間の距離 l は単純に $r_2 - r_1$ である。わざわざ積分を使えば、距離 l は、半径方向の微小な長さ dr を r_1 から r_2 まで積分したものだから

$$l = \int dr = r_2 - r_1 \quad (1)$$

と表すこともできる。

しかしブラックホールの周辺では空間が曲がっており、図1のような計り方をした場合には半径方向に伸びているために、距離 l は $r_2 - r_1$ より長くなる。どのくらい伸びるかという、周長の長さが $2\pi r$ の場所での半径方向の微小な長さが、空間が平坦な場合に比べ、 $1/\sqrt{1-r_g/r}$ の割合で伸びる。あるいは、積分範囲を r_1 から r_2 までとして、距離 l は

$$l = \int_{r_1}^{r_2} \frac{dr}{\sqrt{1-r_g/r}} > r_2 - r_1 \quad (2)$$

となる。ただし(2)式で r_g は、ブラックホールの半径で、ブラックホールの質量 M 、万有引力定数 G 、光速 c を用いて

$$r_g = 2GM/c^2 \quad (3)$$

と表され、俗にシュバルツシルト半径と呼ばれる。

ここでの目標は、(2)式で表される曲がった空間を視覚化することだが、よくあるように、空間の次元数を1つ減らして、3次元ユークリッド空間内の曲面を考えることにしよう（図2）。ただしこの曲面上では、ブラックホールの周りの曲がった空間と同じ幾何学が成り立つとする。すなわち中心（図2の z 軸）の周りの2つの円周の間の距離（図2の曲面に沿って半径方向に計った長さ）が、(2)式になるような曲面だとする。

さて図2の xy 平面内での中心からの距離 r

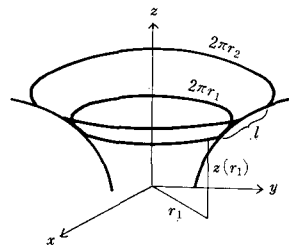


図2 平坦な3次元空間に埋め込まれた、曲がった2次元空間。

* 大阪教育大 Jun Fukue: Visual Relativity 3. Warped Space

($=\sqrt{x^2+y^2}$) に関して、求めたい曲面の方程式を $z=z(r)$ とすると、曲線の長さの公式から、曲面に沿って r 方向に計った長さは

$$l = \int \sqrt{1 + (dz/dr)^2} dr \quad (4)$$

となる。したがって、(2) 式と (4) 式を比較して

$$\sqrt{1 + (dz/dr)^2} = 1/\sqrt{1-r_0/r} \quad (5)$$

が成り立てばよいことが分かる。

この (5) 式を (dz/dr) について解くと

$$dz/dr = \pm \sqrt{r_0/(r-r_0)} \quad (6)$$

となり、(6) 式を積分して最終的に

$$z = \pm 2\sqrt{r_0(r-r_0)} \quad (7)$$

が得られる。ただし積分定数は $r=r_0$ で $z=0$ となるように決めた。また符号は+のみ考えれば十分である。

3. プログラムの目的・状況設定・実行

今回紹介するプログラムは、上で述べたように、3次元ユークリッド空間の中に埋め込まれた、ブラックホールの周囲の時空と同じ幾何学の成り立つ回転曲面を、空間内の任意の位置から投影視するものである。

ただし投影面の中心は3次元ユークリッド空間の原点に固定してあり、また曲面についてもワイヤフレームで表すだけで、陰線消去のような高度な処理は施していない(というよりできない)。

前節の結果より、シュバルツシルト半径 r_0 を長さの単位とすれば ($r_0=1$ とする)、曲面の方程式は

$$z = \pm 2\sqrt{r-1} \quad (8)$$

となる。ただし $r = \sqrt{x^2+y^2}$ である。

観測者は、 xz 平面内で、中心から距離 R_0 、 z 軸から角度 θ_0 の場所において、そこから中心方向を眺めているとする(図3)。投影面は視点と原点を結ぶ直線に垂直だとする。観測者の視野の半頂角は α である。

プログラムを実行すると、以上述べたようなことを簡単に説明した文章が画面に出てくる。何かキーを押せばパラメータの入力を要求してくるので、観測者の視点の位置 (R_0, θ_0)、視野の半頂角 α を入力する(角度の単位は度)。さらに続けて座標格子の描き方もきいてくる

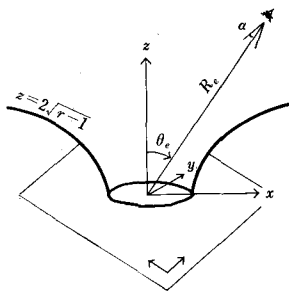


図 3 全体の位置関係。

ので適当に選択する。なおリターンキーを押していけば自動的に標準値が設定されるようになっている。

実行例を図4から図8にいくつか示しておく。図4(表紙上)を例にして、簡単に説明しておく。まず画面の右側には、視点の位置が極座標 (R_0, θ_0) と直角座標 (X_0, Z_0) で表示される。視野の半頂角 α も表示される。四角い枠の周りの数字は、シュバルツシルト半径を単位とした投影面の長さを表している。日付が表示されるのはおまけ。

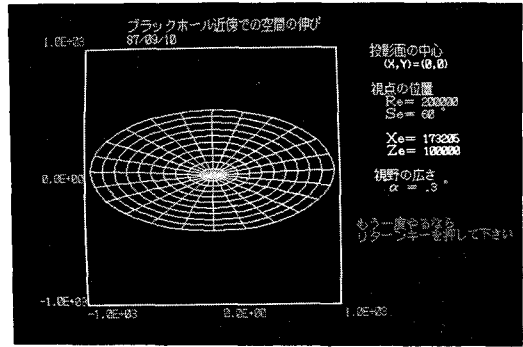


図 6 遠方から広い範囲を見た場合(投影面の一边はシュバルツシルト半径の2000倍)。空間の曲がりが見えない。

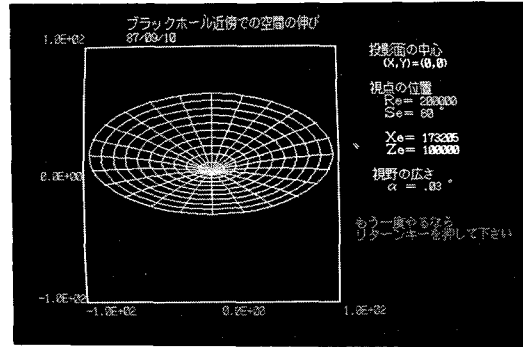


図 7 図6と同じ場所から、図6の10分の1の領域を見ている。中心付近が少し曲がってきたのが分かる。

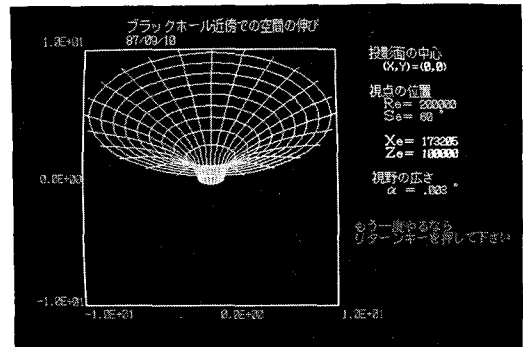


図 8 さらに図7の10分の1で、中心から $10r_0$ くらい領域を見ている。投影面の一边はシュバルツシルト半径の20倍。

座標格子の描き方で直角座標を選択すると、図5（表紙下）のようなメッシュを描くようになっている。図4と図5のパラメータは同じであるがメッシュの描き方によって感じがかなり違ってくる。

ワイヤフレームで曲がった空間を表示し終わったら、もう一度やるか尋ねてくる。

4. コメント

わりとたあいもないプログラムなのだが、例えば図6から図8のように、視野の範囲を変えていくと、空間の曲がりのようすがよく分かる。すなわち図6では投影面の一边は $2000r_0$ で、時空はほとんど平坦に見えるが、見る範囲を1桁小さくすると（図7）、少し曲がってくる。さらに中心から $10r_0$ ほどの領域では（図8）、空間の曲がりが無視できないことが視感できる。

今回から紹介するいくつかのプログラムは、筆者が1985年と1986年に大阪府科学教育センターで、現職の先生方に対してブラックホールの話を講義させてもらった際に、理解の一助にでもなればと思って作成したものです。そのような非常に勉強になる機会を筆者に与えて下さった、科学教育センターの小林英輔氏にこの場を借りて感謝します。

参 考 文 献

ランダウとリフシツ『場の古典論』（東京図書1978年）§100.

付録1 プログラムリスト

（プログラムの本質的でない部分は一部省略した）。

付録2 プログラムの説明と設定値の変更

*INITIALIZE のブロックの1130行と1140行で定義している関数 FNX と FNY は、視点 (E が付いた変数) から見て空間内の点 (X, Y, Z) を投影面に投影したときの、投影面上での X 座標 (1130行) と Y 座

標 (1140行) を与えるものである。

*DOCUMENT (1160-1460) は、説明文を表示する部分である。(ミンコフスキー空間とあるのは、ユークリッド空間が正しい。)

*PARA (1470-1710) は、パラメータを入力し、さらにサブルーチン *GRAPH (3060-3360) を呼ぶ部分である。入力パラメータの標準値を変更したい場合は、1530行、1550行、1610行、1660行をいじればよい。なお視野の広さを入力するとき参考とするため、投影面の一边の長さの半分がシュバルツシルト半径の10倍になるような角度を計算して、“適当な角度”として表示するようになっている。これを変えるためには、1500行の変数 AREA (投影面の一边の長さの半分) を変更すればよい。

*MESSAGE (1720-1840) では、画面右側にパラメータの値を表示する。

*PLOT (2430-3050) が、画面上にメッシュをプロットするメインの部分である。座標格子の描き方で、直角座標以外を選ぶと2500行から2630行が、直角座標を選択すると2650行から2840行が実行される。図4などで10ヶ描いている等半径円(投影した)の個数を変えたければ、2500行と2510行をいじればよい。等半径の円は1または10または100置きに描くようにしているが、2470行の変更で変えられる。動径方向の24本の曲線の本数の変更は2550行と2560行の修正でできる。直角座標の場合のメッシュの本数を変更するのは、2670行を修正すればよい。

サブルーチン *GRAPH (3060-3360) では、グラフィック画面の初期設定、表題と日付の表示、枠と目盛りの表示などを行う。サブルーチン *BEEP (3370-3400) は、プログラムの実行中にブザーを鳴らす部分で、1420行のように呼び出す。

```

1000 '***** SPACE.BAS *****
1010 '
1020 '      Ver.1   1985 0914-0915   original version
1030 '      Ver.2   1986 0422-0423   Japaneseize
1040 '      Ver.3   1987 0812       minor change
1050 '
1060 '*****
1070 '
1080 *INITIALIZE '*****
1090 '
1100 CONSOLE 0,25,0,1,0 : WIDTH 80,25 : SCREEN 3,0,0,1 : CLS 3
1110 PAI=3.141592653589793# : TEN.L=LOG(10#) : PAI.DEG=PAI/180
1120 '
1130 DEF FNX(X,Y,Z)=RE*Y*RE/(RE*RE-XE*X-ZE*Z)
1140 DEF FNY(X,Y,Z)=(XE*Z-ZE*X)*RE/(RE*RE-XE*X-ZE*Z)
1150 '
1160 *DOCUMENT : 'プログラムノセツメイ *****
1170 '
1180 COLOR 7
1190 PRINT "1頁"
1200 COLOR 5
1210 PRINT "          プログラム名 < SPACEJ >"
1220 PRINT
1230 PRINT "          Ver. 1.1 1985 0916   by Jun Fukue" : COLOR 7

```

```

1240 PRINT
1250 PRINT " このプログラムは、ブラックホール近傍での"; : COLOR 6
1260 PRINT "<空間の伸び>"; : COLOR 7 : PRINT "を、視覚化し表示するものです。シ
シュバルツシルト・ブラックホール、即ち球対称重力場中での"; : COLOR 5
1270 PRINT "線素"; : COLOR 7 : PRINT "の半径方向の成分 ds は次の式で表わされます
" : COLOR 4
1280 PRINT "          ds = dr / sqrt(1 - 1/r) > dr, " : COLOR 7
1290 PRINT " 但し、長さはシュバルツシルト半径 Rg = 2 GM / (c * c) によって無次元
化してあります。";
1300 PRINT "また、曲がった 2 次元の空間 (即ち曲面) を平坦な 3 次元ミンコフスキー空
間に埋め込んだものを考え、それを PERS の手法によってスクリーン上に投影して表現
します。";
1310 PRINT "従って 2 次元空間は重力場が無い時は XY 平面に帰着し、重力場による空間
の伸びは Z 方向の変化で示されます。
1320 PRINT "
1330 PRINT " 使用者はコンピューターから以下のパラメーターを入力するように要求さ
れます。(リターンキーを押せば、自動的に標準値が設定されます。)
1340 PRINT "          Re : 視点の位置の動径距離
1350 PRINT "          Se : 視点の位置の極角度 (°)
1360 PRINT "          α : 視野の広さ (角度°)
1370 PRINT "          mesh : 座標格子の選択 0 = 無地,
1380 PRINT "                               1 = 直角座標
1390 PRINT : COLOR 2
1400 PRINT "          説明を読んだらリターンキーを押して下さい。
1410 '
1420 BEP=1 : GOSUB *BEEP1
1430 DOC$=INKEY$ : IF DOC$="" THEN 1430
1440 IF ASC(DOC$)<>13 THEN 1430
1450 CLS 3
1460 '
1470 *PARA : 'ハ*ラメ*タ ル*チン *****
1480 '
1490 CONSOLE 15,8 : CLS : COLOR 7
1500 AREA=10
1510 PRINT "視点の位置を球座標 (Re, Se) で入力して下さい。
1520 PRINT " 中心からの距離はいくらにしますか? ([RET] = 20); Re=";
1530 BEP=1 : GOSUB *BEEP1 : INPUT RE : IF RE=0 THEN RE=20
1540 PRINT " 北極方向からの角度は何度ですか? ([RET] = 0); Se=";
1550 BEP=1 : GOSUB *BEEP1 : INPUT SE : PRINT
1560 SE=SE*PI/180
1570 RE=RE*SIN(SE) : ZE=RE*COS(SE)
1580 PRINT "視野の広さ α を角度で入力して下さい。( [RET] = 30)
1590 IF AREA<RE THEN ALPHA=ATN(AREA/RE/SQR(1-(AREA/RE)^2)) ELSE ALPHA=0
1600 PRINT " 適当な角度は"; ALPHA/PI/180; "° です。α=";
1610 BEP=1 : GOSUB *BEEP1 : INPUT ALPHA : IF ALPHA=0 THEN ALPHA=30
1620 ALPHA=ALPHA*PI/180
1630 SIZE =RE*TAN(ALPHA) : RMAX=SIZE
1640 PRINT : PRINT "座標格子を描きますか?"
1650 PRINT " [RET] = 描かない、1 = 直角座標、2 = 円筒座標、3 = 球座標 ";
1660 BEP=1 : GOSUB *BEEP1 : MESH=0 : INPUT MESH
1670 '
1680 XMIN=-SIZE : XMAX=SIZE : XNUM=3
1690 YMIN=-SIZE : YMAX=SIZE : YNUM=3
1700 GOSUB *GRAPH : BEP=1 : GOSUB *BEEP1
1710 '
1720 *MESSAGE '*****
1730 '
1740 COLOR 5 : LOCATE 55, 2 : PRINT "投影面の中心
1750 COLOR 7 : LOCATE 57, 3 : PRINT "(X,Y)=(0,0)
1760 COLOR 5 : LOCATE 55, 5 : PRINT "視点の位置
1770 COLOR 6 : LOCATE 57, 6 : PRINT "Re=";RE
1780 LOCATE 57, 7 : PRINT "Se=";SE/PI/180; "° "
1790 COLOR 7 : LOCATE 57, 9 : PRINT "Xe=";XE
1800 LOCATE 57,10 : PRINT "Ze=";ZE
1810 COLOR 5 : LOCATE 55,12 : PRINT "視野の広さ
1820 COLOR 6 : LOCATE 57,13 : PRINT "α =";ALPHA/PI/180; "° "
1830 BEP=1 : GOSUB *BEEP1
1840 '
2430 *PLOT : '*****
2440 '
2450 R=1 : Z=0
2460 C=2 : GOSUB *PLOT.SUB : PAINT(0,0),2,2
2470 RMAX0=10 : IF RMAX>1000 THEN RMAX0=1000 ELSE IF RMAX>100 THEN RMAX0=100
2480 '
2490 IF MESH<>1 THEN ELSE *PLOT1
2500 FOR I=1 TO 10
2510 R=RMAX0/10*I
2520 Z=2*SQR(R-1)
2530 C=5 : GOSUB *PLOT.SUB

```

```

2540 NEXT I
2550 FOR I=0 TO 23
2560 PHI=PAI/12*I
2570 PSET(FNX(COS(PHI),SIN(PHI),0),-FNY(COS(PHI),SIN(PHI),0)),1
2580 FOR R=1 TO RMAX STEP RMAX/20
2590 X=R*COS(PHI) : Y=R*SIN(PHI) : Z=2*SQR(R-1)
2600 LINE -(FNX(X,Y,Z),-FNY(X,Y,Z)),5
2610 NEXT R
2620 NEXT I
2630 GOTO *PLOT.END
2640 '
2650 *PLOT1
2660 C=5
2670 IX=10 : JX=IX
2680 FOR I=-IX TO IX
2690 X=RMAXO/IX*I : Y=-RMAXO : R=SQR(X*X+Y*Y) : Z=2*SQR(R-1)
2700 PSET(FNX(X,Y,Z),-FNY(X,Y,Z)),C
2710 FOR Y=-RMAXO TO RMAXO+RMAXO/50 STEP RMAXO/20
2720 R=SQR(X*X+Y*Y) : IF R>1 THEN Z=2*SQR(R-1) ELSE Z=0
2730 LINE -(FNX(X,Y,Z),-FNY(X,Y,Z)),C
2740 NEXT Y
2750 NEXT I
2760 FOR J=-JX TO JX
2770 X=-RMAXO : Y=RMAXO/JX*J : R=SQR(X*X+Y*Y) : Z=2*SQR(R-1)
2780 PSET(FNX(X,Y,Z),-FNY(X,Y,Z)),C
2790 FOR X=-RMAXO TO RMAXO+RMAXO/50 STEP RMAXO/20
2800 R=SQR(X*X+Y*Y) : IF R>1 THEN Z=2*SQR(R-1) ELSE Z=0
2810 LINE -(FNX(X,Y,Z),-FNY(X,Y,Z)),C
2820 NEXT X
2830 NEXT J
2840 GOTO *PLOT.END
2850 '
2860 *PLOT.SUB
2870 PSET(FNX(R,0,Z),-FNY(R,0,Z)),C
2880 FOR PHI=0 TO 2*PAI+PAI/18 STEP PAI/18
2890 X=R*COS(PHI) : Y=R*SIN(PHI)
2900 LINE -(FNX(X,Y,Z),-FNY(X,Y,Z)),C
2910 NEXT PHI
2920 RETURN
2930 '
2940 *PLOT.END : BEP=3 : GOSUB *BEEP1 '*****
2950 '
2960 COLOR 2 : LOCATE 52,16 : PRINT "もう一度やるなら
2970 LOCATE 52,17 : PRINT "リターンキーを押して下さい
2990 WHILE 1
3000 LOOP$=INKEY$ : IF LOOP$="" THEN 3000
3010 IF ASC(LOOP$)=13 THEN *PARA
3020 BEP=1 : GOSUB *BEEP1
3030 WEND
3040 END
3050 '
3060 *GRAPH : 'グラフィック サブルーチン *****
3070 '
3080 'Initialize
3090 CONSOLE 0,25,0,1 : WIDTH 80,25 : SCREEN 3,0,0,1 : CLS 3
3100 XLEN = XMAX-XMIN : XDEL = XLEN/(XNUM-1)
3110 YLEN = YMAX-YMIN : YDEL = YLEN/(YNUM-1)
3120 XMIN = 65 : XMAX = 575 : YMIN = 35 : YMAX = 365 : XMAXS=395
3130 XDELS = 80*(XDEL/XLEN)*(XMAXS-XMIN)/640
3140 YDELS = 25*(YDEL/YLEN)*(YMAXS-YMIN)/400
3150 WINDOW(XMIN,-YMAX)-(XMAX,-YMIN) : VIEW(XMIN,YMIN)-(XMAXS,YMAXS)
3170 'Headings
3180 WHAT$="ブラックホール近傍での空間の伸び
3190 COLOR 6 : LOCATE 15,0 : PRINT WHAT$ : LOCATE 15,1 : PRINT DATE$
3220 'Axes
3230 LINE(XMIN,-YMIN)-(XMAX,-YMAX),7,B
3260 FOR I=1 TO XNUM STEP 1
3270 COLOR 4 : LOCATE XMIN*80/640+(I-1)*XDELS,23
3280 PRINT USING "##.#^"; XMIN+(I-1)*XDEL;
3290 NEXT I
3310 FOR J=1 TO YNUM STEP 1
3320 LOCATE 0, YMAXS*25/400-(J-1)*YDELS-1
3330 PRINT USING "##.#^"; YMIN+(J-1)*YDEL;
3340 NEXT J
3350 RETURN
3360 '
3370 *BEEP1:'*****
3380 FOR IBEEP1=1 TO BEP : FOR KBEEP1=1 TO 5 : BEEP 1 : FOR JBEEP1=1 TO 20
3390 NEXT JBEEP1 : BEEP 0 : FOR JBEEP1=1 TO 5 : NEXT JBEEP1 : NEXT KBEEP1
3400 FOR JBEEP1=1 TO 100 : NEXT JBEEP1 : NEXT IBEEP1 : RETURN

```