

# 重力レンズ効果による画像変形のシミュレーション

大阪工業大学 情報科学部 情報ネットワーク学科

西田大輝

平成 29 年 2 月 8 日

# 目次

## 1 序論

### 1.1 背景

### 1.2 本研究の目的

### 1.3 本論文の構成

## 2 準備

### 2.1 Newton の万有引力ポテンシャルと擬ニュートンポテンシャル

### 2.2 Runge-Kutta 法

### 2.3 テスト計算

### 2.4 使用した環境

### 2.5 使用したデータ

### 2.6 物体への重力の影響のシミュレーションプログラム

### 2.7 画像を半分加工するプログラム

### 2.8 重力により光の影響を画像に反映させるプログラム

### 2.9 レイトレーシング法

### 2.10 レイトレーシング法を用い改良したプログラム

### 2.11 複数の重力源がある場合のプログラム

### 2.12 光の明るさの求め方

### 2.13 四つの点からなる四角形の面積の求め方

## 3 重力レンズ効果による光の明るさの変化

### 3.1 光の明るさの可視化

### 3.2 2つの変化する重力による明るさへの影響

### 3.3 ニュートンポテンシャルと擬ニュートンポテンシャルの光量の違い

## 4 結論

## 5 参考文献

# 1 序論

## 1.1 背景

重力は様々な物質に影響を与える。ニュートン力学では物質が互いに互いを引き合う万有引力として考えるが、一般相対性理論では重力を時空の歪みという形で説明する。一般相対性理論での重力による影響は光にも及ぶ。この重力がレンズのような役割を果たし、光をゆがめる現象を重力レンズ効果という。重力レンズ効果は1924年にオレスト・ダニーロヴィッチ・フヴォリソンに発表されたが話題にならなかった。後に、1936年にアルベルト・アインシュタインが、「アインシュタインリング」と呼ばれる、光源、重力源、観測者が一直線上に並んだ際に起こる現象がありえることを発表したことで有名になった。この重力レンズ効果は現在では様々な形で利用され研究されている。具体的にはダークマターの観測や太陽系外惑星の発見である。

前者は、ダークマターは質量を持つ以外はどのような性質があるか分かっておらず、目にすることも直接観測することもできないが、重力レンズ効果は質量があれば、光がその影響を受け観測者にゆがんだ像として写したり、明るさを変える。この歪み方から実際に観測できる物質では説明できない質量をダークマターとして間接的に観測できる。

後者は、太陽系外惑星は光源である恒星と異なり光を放たないためほぼ直接観測することはできない。トランジット法という光源の恒星の周囲を公転する惑星が、恒星の前を横切る際に一部分だけが暗くなることから観測する方法がある。

重力レンズ効果では、トランジット法とは逆に重力マイクロレンズ効果という、恒星から放たれた光がその恒星の前を横切る惑星の重力による重力レンズ効果で影響を受け、光が集中することで明るくなる現象から観測することができる。

以上のように重力レンズ効果は様々な面で研究され利用されている。

## 1.2 本研究の目的

この研究では、重力レンズ効果をシミュレーションし、その影響を画像ファイルという形で出力することで、重力レンズ効果が、実際に光にどのように影響を与え、観測者の目に映るかを再現することが目的である。

また、重力マイクロレンズ効果を用いた学習効果 [1] によれば、上記の重力マイクロレンズ効果の中には、連星となったブラックホールを重力源とした場合、光の明るさが発散する場合がありますとされる。そのため、光の明るさについても再現する。

## 1.3 本論文の構成

まず、2章ではシミュレーションに使用したものを記載する。使った式やコンピュータの環境、使用したデータなどである。次に、3章では目的である、シミュレーションを実際に完成させるまでに作った様々なプログラムの、その概要と結果を記載し、考察する。その後、4章では完成したシミュレーションプログラムとその結果から、シミュレーション結果の結論を述べる。

## 2 基礎となる式について

### 2.1 Newtonの万有引力ポテンシャルと擬ニュートンポテンシャル

Newtonの運動方程式は $x, y, z$ の3つの座標を添え字 $i$ を用いて表すと、物体の質量を $m$ として、

$$m \frac{dv^i}{dt} = -\frac{\partial}{\partial x^i} U_N \quad (1)$$

となる。また、 $U_N$ はポテンシャルを表し、質量 $M$ の重力源が原点にあるとすれば、万有引力定数を $G$ として、以下のような式で求める。

$$U_N(x, y, z) = -G \frac{Mm}{r} = -G \frac{Mm}{\sqrt{x^2 + y^2 + z^2}} \quad (2)$$

この式でポテンシャルを求めることができるが、これは相対性理論のポテンシャルや実際の重力による影響を考えると、小さい。そこで擬ニュートンポテンシャルを用いる。これはらくらく相対論 [2] [3] を参考にした。

ニュートンの重力方程式は、アインシュタインの一般相対性理論に比べて、シュバルツシルト半径の周辺での物体の振る舞いが大きく異なるが、擬ニュートンポテンシャルによってブラックホール周辺の実際の振る舞いに近づけるようにしたものができる。これは上記のポテンシャルの式に、質量 $M$ を持つ物体のシュバルツシルト半径 $a$ を用いて表すことができる。光速を $c$ として、シュバルツシルト半径は以下の式で表せる。

$$a = \frac{2GM}{c^2} \quad (3)$$

この $a$ を用いた擬ニュートンポテンシャルの式は以下のとおりである。

$$U_{PN}(x, y, z) = -G \frac{Mm}{r - a} = -G \frac{Mm}{\sqrt{x^2 + y^2 + z^2} - a} \quad (4)$$

この式は、容易にニュートンポテンシャルを精度の高いものにするができるが、シュバルツシルト半径に近づけば近づくほど精度が下がってしまう。しかし、この研究ではシュバルツシルト半径より内側に入った光の粒子は考慮しないので、無視できる。

## 2.2 Runge-Kutta 法

Runge-Kutta 法は常微分方程式の数値計算法のひとつである。ここでは4次の Runge-Kutta 法を用いた。

$t$  を時間、 $f(t, y)$  を常微分方程式、 $h$  を  $t$  方向の刻み幅とすると、解くべき方程式は以下のとおりである。

$$y' = f(t, y) \quad (5)$$

$$y(t_0) = y_0 \quad (6)$$

$$k_1 = f(t_n, y_n) \quad (7)$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \quad (8)$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \quad (9)$$

$$k_4 = f(t_n + h, y_n + hk_3) \quad (10)$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (11)$$

$$t_{n+1} = t_n + h \quad (12)$$

方程式の中で、 $y_n = y(t_n)$  がわかるなら、十分に小さい刻み幅  $h$  を与えることで、時刻  $t_{n+1}$  の近似値  $y_{n+1}$  を求めることができる。

## 2.3 テスト計算

§2.1 の運動方程式とポテンシャルの方程式について §2.2 の Runge-Kutta 法で近似することで、ひとつの粒子がどのように重力の影響を受けるかのシミュレーションを行うプログラムを作成した。このプログラムは中心に重力源を置き、その影響により、ひとつの質量  $m$  の粒子がどのように軌道を変更されるかを表すプログラムである。

粒子が円運動を行っており、重力源を中心に公転していることから、恒星とその周りを公転する惑星が再現できたとともに、この結果は正しい。このプログラムをもとに拡張修正などを行い、研究の準備を進めていく。

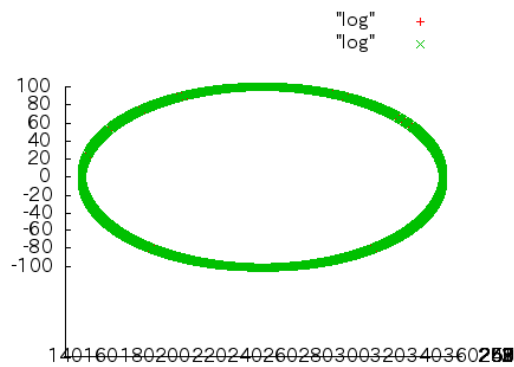


図 1: 結果

## 2.4 使用した環境

主に、ubuntu と Windows 8 の OS のコンピュータを用いた。プログラミングに使用した言語はすべて C 言語である。

## 2.5 使用したデータ

研究中に使用したデータは以下の画像ファイルである。この画像ファイルはフリー素材である。URL は以下のとおりである。

<http://free-photos.gatag.net/2013/07/04/040000.html>



図 2: kao.ppm

## 2.6 物体への重力の影響のシミュレーションプログラム

§2.3で記載したように、重力の影響を受け光の粒子がどのように重力の影響を受けるかのシミュレーションプログラムが完成した。その実行結果が以下のとおりである。

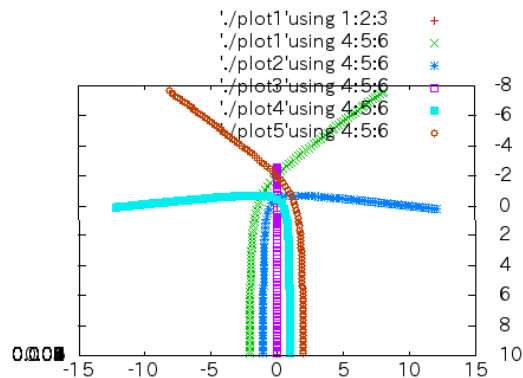


図 3: 等間隔に 5 つ置いた光の粒子の軌跡

このシミュレーションプログラムでは 5 つの等間隔に並べられた  $z$  軸方向に進行する光の粒子が、重力源によって、どのように重力の影響を受けるかを確認するとともに、このプログラムが誤っていないかを判断するためのものである。実際には Newton 重力では光は曲がらないが、ここでは曲がる粒子の軌跡を光の軌跡とみなして出力している。この出力結果は、実際のデータが左右対称であり、誤った軌道を描いているわけではないため、適切であると判断できる。



## 2.7 重力による光の影響を画像に反映させるプログラム

§2.6のプログラムを完成させたのち、重力による光の影響を画像に反映させるプログラムを作成した。このプログラムは§2.6のプログラムの重力レンズ効果のシミュレーションによって、画像を加工するものである。

p p m形式の画像はピクセルという色のデータの集まりで、一つ一つのピクセルが集まってひとつの画像になる。このピクセルのデータを光の粒子とみなし、重力レンズ効果の影響を与え、観測者にどのような像が見えるかをシミュレーションするプログラムである。

重力源を100、光の開始位置を-100、重力源の位置を0、観測者の位置を100、光の速さを1として設定した。質量は10である。このシミュレーションで扱う画像を画像、加工した画像を画像2とする。

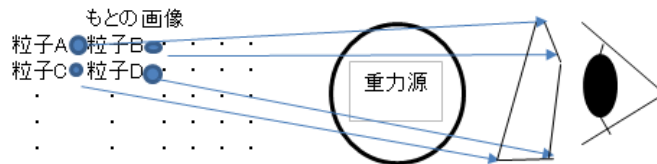


図 4: 画像変形のイメージ図

中心に向かうにつれ、画像がゆがんでいるが、中心の座標には重力源がある。プログラム 1 の出力結果にあるように、中心に行くほど光の軌道は歪み、中心に近すぎると外側にはじかれ、届かない粒子ができる。以上のことから、中心の黒い穴は重力レンズ効果により、光が届かず、画像に上書きされなかったと判断できる。この黒い穴は中心に置いた重力源が大きいほど大きくなる。この結果からこのプログラムも適切であると判断できる。



図 5: 顔の画像 1



図 6: 変形された顔の画像 2

## 2.8 レイトレーシング法

§2.7のプログラムには、入力画像、出力画像の2つの画像の格納データがあり、入力画像から光の粒子を飛ばし、出力画像のそれぞれのピクセルデータの位置に光の粒子が届いた場合、その光のデータを出力画像のデータに上書きするというものであった。しかし、このプログラムでは10万個以上の光の粒子を飛ばしているが、同じ出力先のデータに複数の粒子が届いてしまう場合がある。この場合に行う例外処理などの関係でプログラムが複雑になってしまい、さらに計算量が膨大になってしまう。

そこで、上記のように粒子を入力先から飛ばし、どのような光が届いたかで出力画像を決定するのではなく、出力先から粒子を飛ばし、どのような光が届くのかで出力画像を決定できるレイトレーシング法という技法を用いることにした。

通常、光は、光源から放たれ、反射したり、透過したりを経て、我々観測者の目に届く。実際、これまでのプログラムでは、光源から光を放ち、それが重力によってどのように軌道を変え、観測者に届くかをシミュレーションしていた。しかし、レイトレーシング法では、それとは逆に観測者の目から光をたどる。光線追跡法とも呼ばれるこの手法はCGなどの技術でも扱われているが、光の反射による演算量が膨大である。しかし、この課題では、光が通る場所は宇宙空間であり、光が反射することはまずないため、これを無視できる。

今までの技法とは異なり、この技法では、ピクセルの一つ一つにどのようにものが見えるかをシミュレーションするため、一つのピクセルに複数の光が重複することはない。このことから、プログラムの計算量を引き下げることが可能にする。

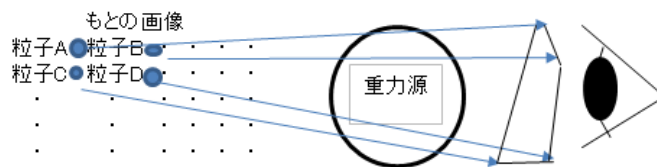


図7: レイトレーシングのイメージ画像

## 2.9 レイトレーシング法を用いた改良したプログラム

レイトレーシング法を用いたプログラムに変更した後、プログラム1のような出力結果ではなく、複数のデータから上下左右で対称であることを証明するデータを出力した。このデータは下が出力先のデータで、上が入力元のデータである。100×100の範囲で光の粒子を飛ばし、その結果を確認するために、縦、横、斜めからみたデータがそれぞれ出力結果1、出力結果2、出力結果3である。

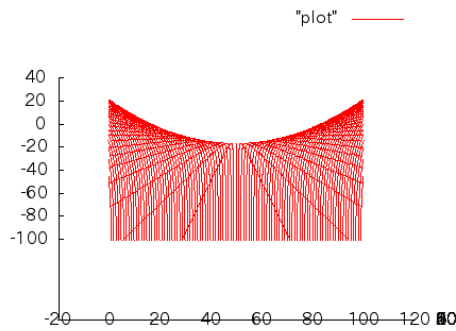


図 8: 縦方向の断面図の出力結果 1

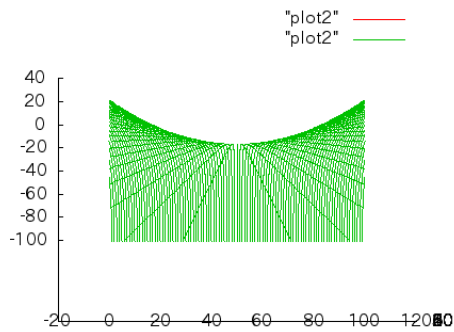


図 9: 横方向の断面図の出力結果 2

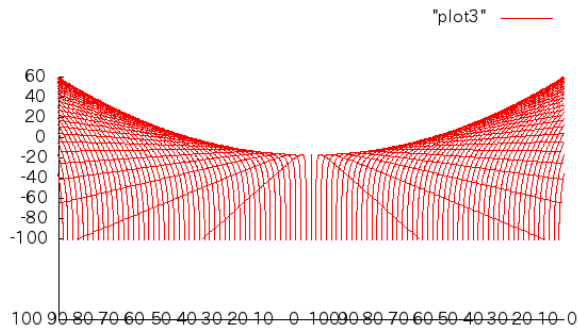


図 10: 斜め方向の断面図の出力結果 3

このような結果のうち、出力結果 1 と 2 はそれぞれ中心に向け、縦と横からきりとった断面図であり、対称なデータになる。出力結果 1, 2 は細かな座標データも完全に一致し、グラフ上のデータも同様である。よって適切な結果である。対して出力結果 3 は 1, 2 と縮尺が異なるが、斜めは縦、横に比べ  $\sqrt{2}$  倍になるため、縮尺が異なる。しかし、縦、横の結果とほぼ同様の形をしていることから、これも同様に適切な結果であるといえる。

## 2.10 複数の重力源がある場合のプログラム

次に、目的の項で述べたように、連星のような複数の重力源の場合をシミュレーションする。プログラムに入力する画像は以前までのような画像ではどのように影響しているかが判別しづらいため、オリジナルの方眼紙状のデータを作った。入力元の画像を加工前の図、重力源が一つの場合の画像を重力源が 1 つの図、重力源が 2 つの場合の画像を重力源が 2 つの図とする。また、2 つの重力源は同じ大きさで重さ 10、観測者、重力源、光源は等間隔、1 つの重力源の場合は X 座標 500、Y 座標 250 に、2 つの重力源の場合は 2 つ目を 1 つ目の X 座標に 50、Y 座標に 50 ずらした位置に置く。

図 1 は §2.7 の画像 2 と同様の穴が中心にあることから、この画像が適切に加工されたことがわかる。図 3 はその穴と同様の穴が 2 つできている。また 2 つの穴の中心が引き伸ばされていることから、その間に大きな重力によって引っ張り合っていることが見てとれる

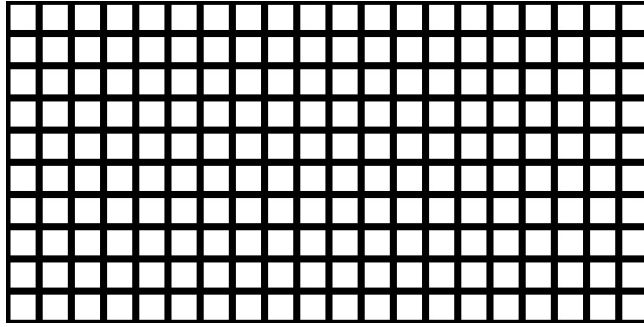


図 11: 加工前の画像

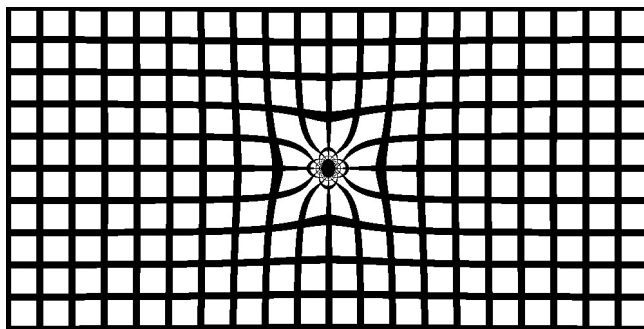


図 12: 重力源が1つの画像

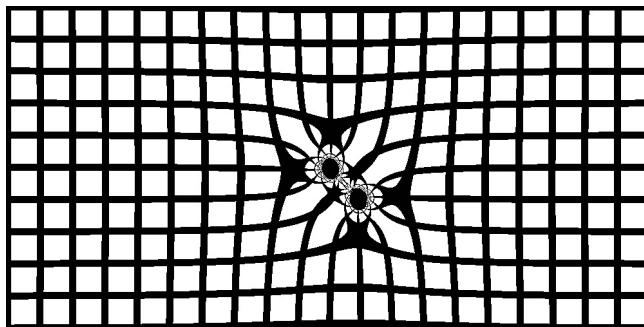


図 13: 重力源が2つの画像

## 2.11 光の明るさの求め方

目的の項にも記したように、本研究では、光の明るさについても扱う。光の明るさを表す方法として、この研究では光の到達地点の面積を用いた。シミュレーションプログラムでは、画像のひとつのピクセルをひとつの光として飛ばしているが、実際には一つ一つのピクセルの間にも、プログラム上では扱いきれないほどの光の粒子が存在するはずである。つまり、隣り合うピクセルとピクセルからそれぞれ飛ばされた光の粒子の、それぞれの到達した場所の距離が短かければ、その小さな範囲に光が集まっており、長ければ、その広い範囲に光が散らばっていると考えられる。

これを利用し、上下、左右、斜めに隣り合う点4つからなる四角形の面積を求め、その面積に応じて光の明るさを決める手法に決めた。ただし、このシミュレーションではレイトレーシング法を用いており、上記とは逆の結果になる。つまり面積が広いほど明るく、面積が小さいほど暗い。

## 2.12 四つの点からなる四角形の面積の求め方

複数の座標から、その座標を頂点に持った多角形の面積を求める公式の中にヘロンの公式というものがある。これは3つの座標を頂点にもつ三角形の面積を求める式であり、3辺の長さをそれぞれa,b,cとすると三角形の面積Tは以下の式であらわされる。

$$s = \frac{a + b + c}{2} \quad (13)$$

$$T = \sqrt{s(s-a)(s-b)(s-c)} \quad (14)$$

また、座標から辺の長さZは2点の座標をそれぞれ

$$(x_1, y_1)(x_2, y_2) \quad (15)$$

とすると以下の式で求められる。

$$Z = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (16)$$

この式を応用して、四角形の面積を求める式にしたものがブラーマグプタの公式である。同様に4辺の長さをそれぞれ  $a, b, c,$  とすると、四角形の面積  $T$  は以下の式であらわされる。

$$s = \frac{a + b + c + d}{2} \quad (17)$$

$$T = \sqrt{(s - a)(s - b)(s - c)(s - d)} \quad (18)$$

ただし、この式は4つの点がこの四角形に内接する円の円周上にある場合に限られる式である。この式をさらに一般化した式がブレートシュナイダーの公式である。この式の4辺をそれぞれ  $a, b, c, d,$  周の長さの半分を  $T,$  対角となる角度2つをそれぞれ  $A, C,$  面積を  $S$  とすると、以下のような式になる。

$$T = \frac{a + b + c + d}{2} \quad (19)$$

$$S = \sqrt{(T - a)(T - b)(T - c)(T - d) - abcd \cos^2 \frac{A + C}{2}} \quad (20)$$

また、角度  $C$  はその周りの座標3つをそれぞれ

$$(x_1, y_1)(x_2, y_2)(x_3, y_3) \quad (21)$$

とすると、以下のように求められる。角度  $A$  も同様である。

$$\cos C = \frac{(x_2 - x_1)(x_3 - x_2) + (y_2 - y_1)(y_3 - y_2)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}} \quad (22)$$

これにより角度が求められる。



### 3 重力レンズ効果による光の明るさの変化

#### 3.1 光の明るさの可視化

§2.11、§2.12の項目の内容を用いて、光の明るさを可視化した。黒い部分程暗く、白い部分程明るい。重力源を画像の中心に置き、観測者、重力源、光を一行に、等間隔に並べた。単位は、重さは10、間隔は100で、光の早さは1である。明るさは、重力による影響がない場合は一様に1の値の明るさを持つ。このグラフからはわかりにくいですが、実際のデータを確認すると、外側は光の明るさは1に近い値になっており、中心付近は2~5000程度、中心は光が届いていないため0の明るさを持っている。この結果から、重力源に近いほど光は明るくなり、一定の距離近づくと、光が届かなくなる。

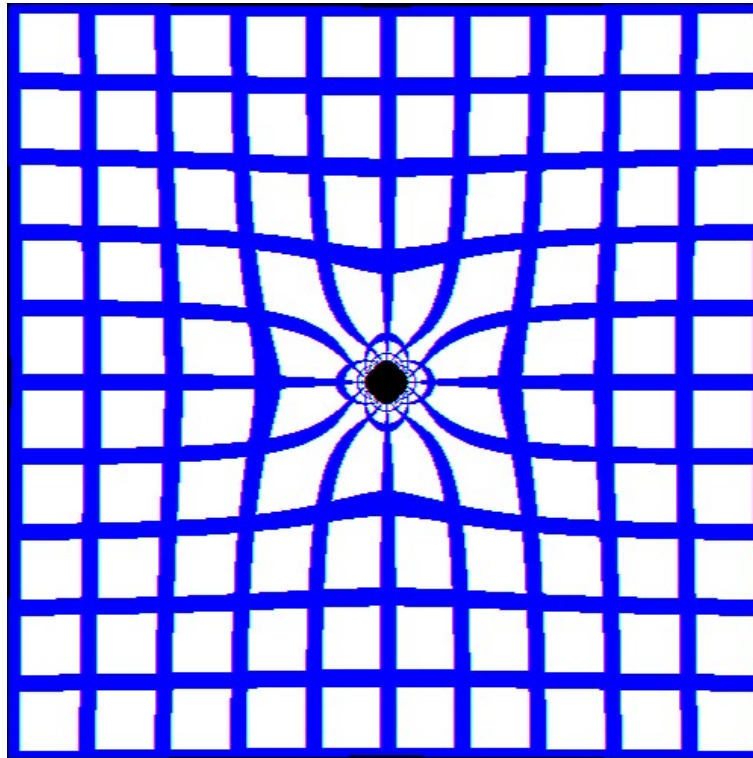


図 14: 実際の画像

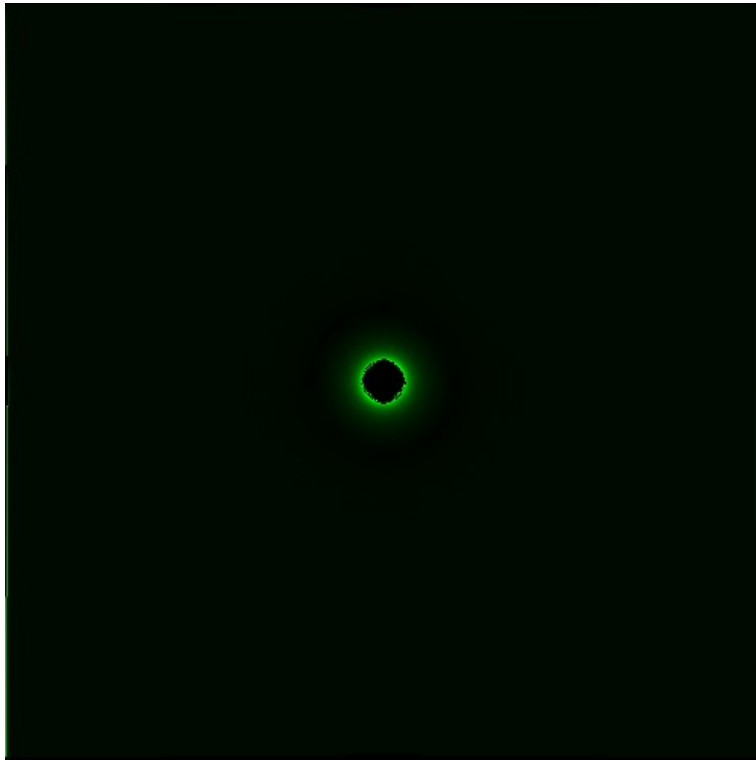


図 15: 明るさの画像

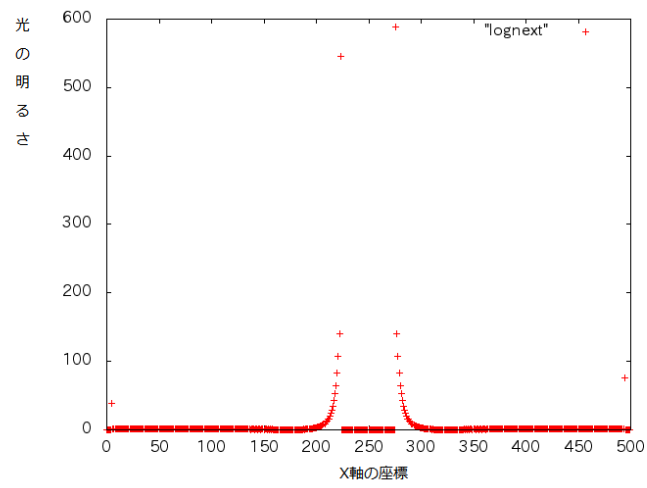


図 16: 明るさのグラフ

### 3.2 2つの変化する重力による明るさへの影響

§2.10の複数の重力源のプログラムを拡張し、大きな重力源と、その周りを公転する重力源を設定し、円形の光源の前を横切るような場合の明るさへの影響を調べた。この光源は一様な明るさを持ち、観測者、重力源の奥に配置される。大きな重力源は10、ちいさな重力源の大きさは1である。この場合の実際の光の明るさの変化は以下の通りである。

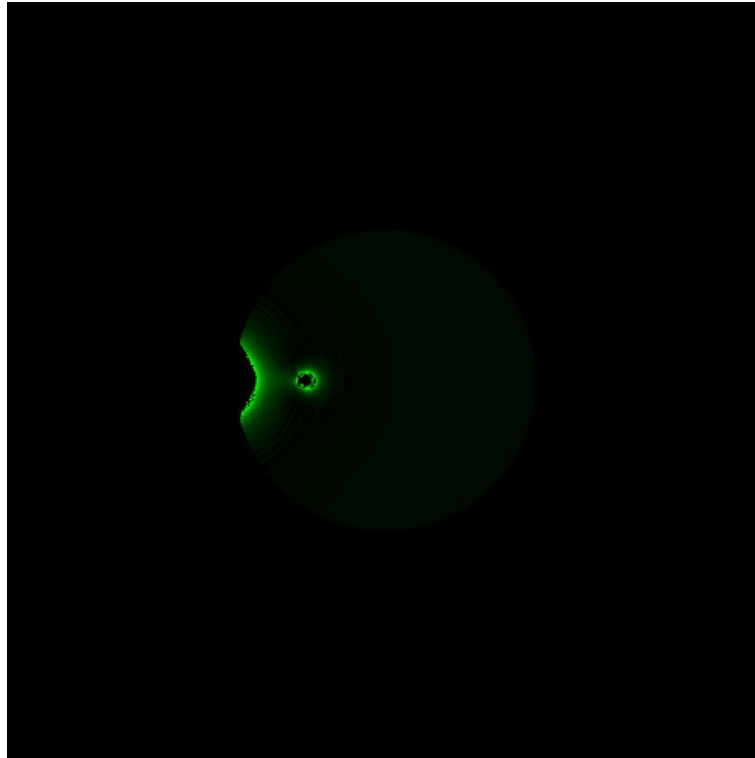


図 17: 変化する重力源が光源の前を横切る画像 1

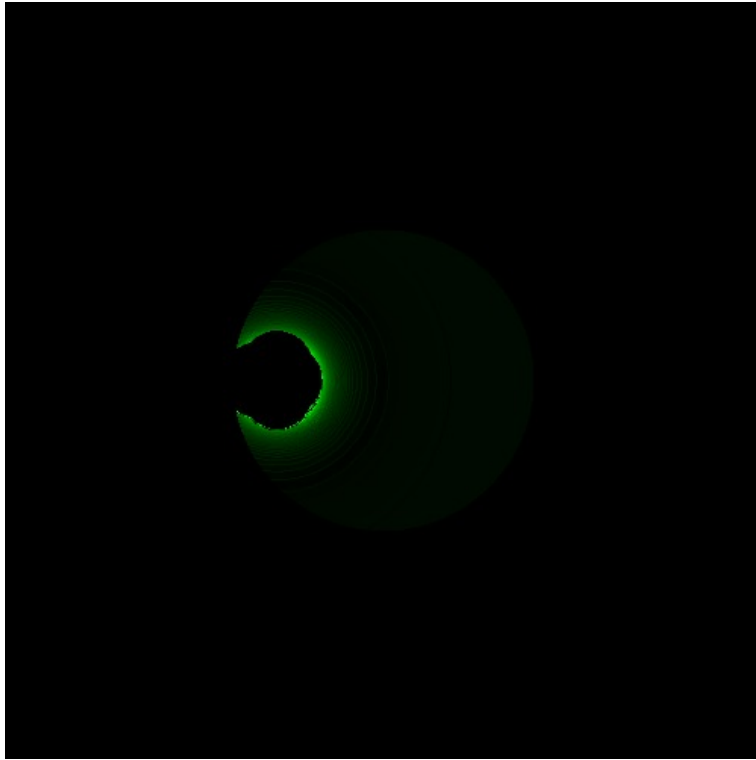


図 18: 変化する重力源が光源の前を横切る画像 2

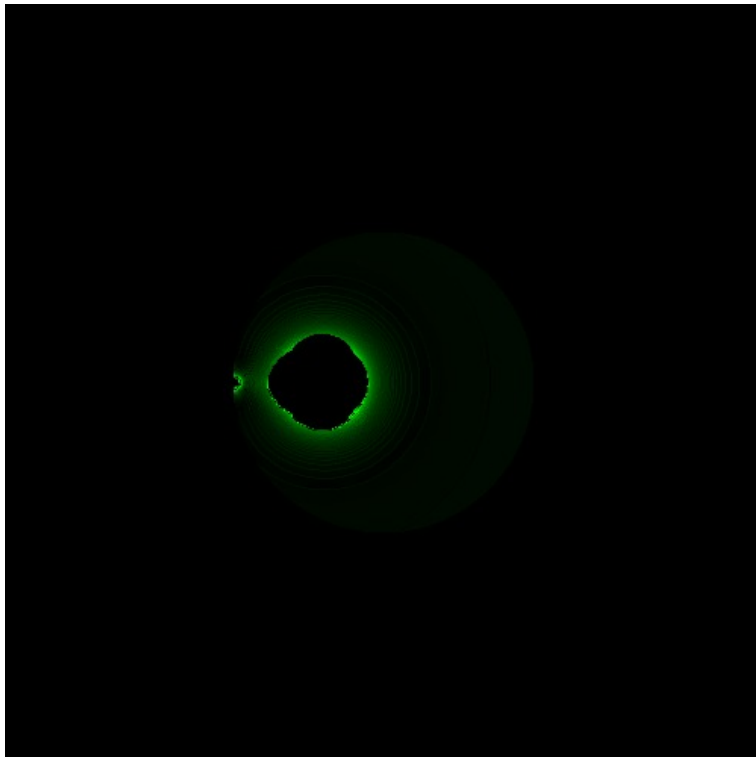


図 19: 変化する重力源が光源の前を横切る画像 3

### 3.3 ニュートンポテンシャルと擬ニュートンポテンシャルの光量の違い

§3.2の内容と同様の設定を用いて、その光量の変化を、ニュートンポテンシャル、擬ニュートンポテンシャルの両方で調べた。光量は実際の光源の明るさを1とする。縦軸は光の明るさの倍率、横軸は重力源の座標である。結果は以下のとおりである。

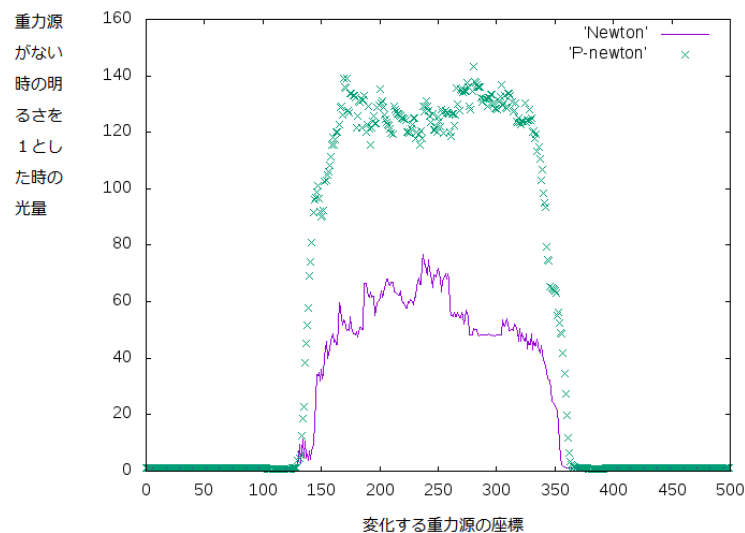


図 20: 光量の変化の比較

この画像は観測者からみて左から、変化する重力源を通過させた際の光量の変化を比べるグラフである。線のグラフはニュートンポテンシャル、点のグラフは擬ニュートンポテンシャルを表す。この結果から、重力の影響を強く受けている場合、擬ニュートンポテンシャルのほうが明るくなることがわかる。

## 4 結論

§3.3の結果から、光は重力の影響を強く受けるほど、明るくなることがわかった。また、ニュートンポテンシャルと擬ニュートンポテンシャルでは、擬ニュートンポテンシャルのほうが光量が明るくなっていることから、擬ニュートンポテンシャルのほうが強い重力を再現できることがわかった。

## 5 参考文献

### 参考文献

- [1] Colin S. Wallace Timothy G. Chambers Edward E. Prather and Gina Brisenden "Using graphical and pictorial representations to teach introductory astronomy students about the detection of extrasolar planets via gravitational microlensing "
- [2] 福江 純 「らくらく相対論入門 その1：擬ニュートンポテンシャルの特徴」天文月報
- [3] 福江 純 「らくらく相対論入門 その1：擬ニュートンポテンシャルへの補正」天文月報