

Python を利用した月食図の描画

江 越 航 *

概 要

月食は日食と並び天文現象の中でも分かりやすく印象深い現象であるため、一般市民の関心も高い。月食の時間経過を説明するために、しばしば月食図が用いられる。この図の作成には月と太陽の座標を求めるために専門的な天文計算が必要となることから、NASA や国立天文台が作成したものを使用することが多い。しかし月食図自体は単純な図のため、天体の座標さえ分かれば自分で描くことができる。

近年、Python のライブラリが充実してきたことから、これを利用して月食図を描くことを試みた。本稿ではそのための手法について述べる。

1. はじめに

Python は最近広く使用されている汎用のプログラミング言語である。科学技術計算が得意で、線形代数や微積分、フーリエ解析、統計解析など関連するライブラリも多くそろっている。特に近年は、機械学習のプログラムを書く際の言語として知られており、ディープラーニングによる人工知能の技術開発のために広く用いられている。

Python は機械学習以外にも様々な分野のライブラリが豊富に作成されている。天文計算専用のライブラリについても既に高精度のものが作成されており、筆者はこれを利用して日の出・日の入り等の暦の計算や、恒星・惑星の位置、国際宇宙ステーションの軌道の計算例、日食の計算方法の例を示した[1][2]。

天文現象の中でも月食は、日食と並び印象深い現象であるため、一般市民の関心も高い。この月食の計算を行うことは、天文学に関心のある層の人々にとって興味ある分野である。そこで本稿では Python を利用して、月食図を描く方法について述べる。

2. 月食図

2021 年は日本全国で、5 月 26 日に皆既月食、11 月 19 日に部分月食の 2 回の月食を観測することができた。

各月食の状況を説明するために、主に 2 つの説明

用の図が用いられる。

1 つ目の図は、月食の経過を示すため、時間の経過とともに月が地球の半影、および本影をどのように移動していくか、位置関係を示す図である。この図を見れば、何時に月が欠け始めて、何時に終わるのか、また月食の欠け具合や継続時間などの全体像をつかむことができる。

もう一つの図は、世界地図上に、各地で月食が見えるかどうかを示した図である。月食は月が出ていれば、世界中同じ時刻に見ることができるが、実際に月食が見えるかどうかは、その時間に月が昇っているかどうかによる。そこで、欠け始め、最大食、欠け終わりなど月食の各段階で、見える地域を示した地図が作られている。

月食図としてよく用いられるのは、NASA Eclipse Web Site のページに掲載されている図である(図 1) [3]。このページには、紀元前 2000 年から西暦 3000 年までに起こる、すべての月食の月食図が掲載されている。

図 1 において、上側の図が地球の影を通過する月の経路を示している。また、下側の図が月食の各段階で観測可能な地球上の領域を示している。

上側の経路図では、欠け始めや最大食の瞬間など、特徴的な時刻の月の位置と地球の影の位置関係が描かれている。しかし、1 時間ごとの定期的な月の移動の様子は描かれていない。

また下の世界地図は、世界各地での月食観測が可能かどうか、全体像を見られるように広い範囲の地図が描かれている。しかし、例えば日本で月出帯食にな

*大阪市立科学館学芸課
e-mail:egoshi@sci-museum.jp

るような場合には、日本付近のみを拡大して、さらに詳細な図が欲しい。そうした場合は、自分で月食図を描く必要がある。

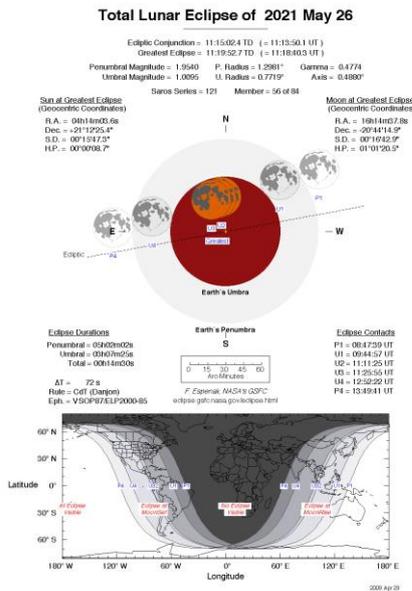


図1 NASA Eclipse Web Site の月食図

3. 地球の本影・半影の視半径

月食図を描くためには、地球から見た時の月の見かけの大きさ、および地球の影の大きさを求める必要がある。月の見かけの大きさは、月の直径を月までの距離で割れば求めることができるので、ここでは地球の影の大きさを求める方法について述べる。

月軌道上での地球の影の大きさは、太陽・地球・月の大きさと、太陽-地球間、地球-月間の距離から、幾何学的に求めることができる[4]。

本影の視半径の大きさは図2を参照して、

$$(\text{本影視半径}) =$$

$$(\text{月視差}) - (\text{太陽視半径}) + (\text{太陽視差})$$

で計算できる。同様に半影視半径の大きさは図3より

$$(\text{半影視半径}) =$$

$$(\text{月視差}) + (\text{太陽視半径}) + (\text{太陽視差})$$

で計算できる。

一方、上式における太陽の視半径は、太陽の大きさと太陽までの距離から

$$(\text{太陽視半径}) =$$

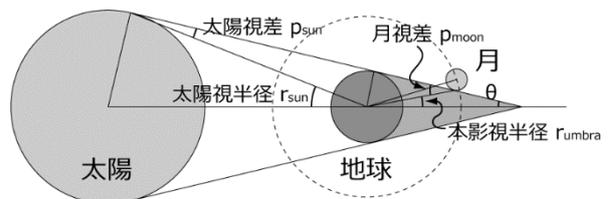
$$(\text{太陽半径}) / (\text{太陽までの距離})$$

さらに太陽・月の視差(太陽・月を地球表面で見た位置と、地球中心で見た位置の差)は

$$(\text{月視差}) = (\text{地球半径}) / (\text{月までの距離})$$

$$(\text{太陽視差}) = (\text{地球半径}) / (\text{太陽までの距離})$$

で計算できることから、太陽・地球・月の半径、および太陽までの距離と月までの距離が分かれば、幾何学的に本影・半影の視半径の大きさが求められる。

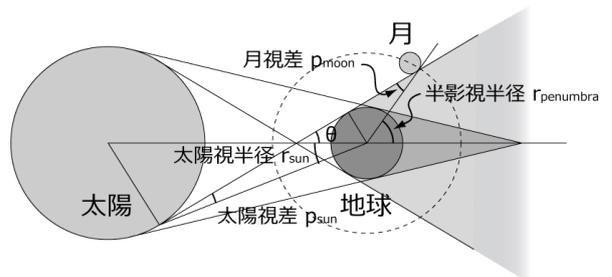


本影視半径 r_{umbra}

$$= \text{月視差 } p_{moon} - \theta$$

$$= \text{月視差 } p_{moon} - (\text{太陽視半径 } r_{sun} - \text{太陽視差 } p_{sun})$$

図2 本影視半径の大きさ



半影視半径 $r_{penumbra}$

$$= \text{月視差 } p_{moon} + \theta$$

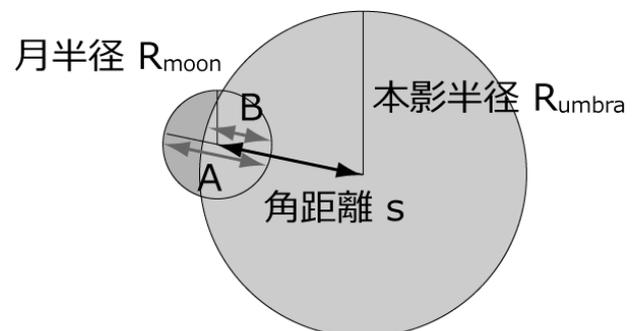
$$= \text{月視差 } p_{moon} + (\text{太陽視半径 } r_{sun} + \text{太陽視差 } p_{sun})$$

図3 半影視半径の大きさ

なお地球の影の視半径の大きさは、大気による屈折の影響で約 1/50 大きくなることから、実際に計算する際は、さらに 51/50 を掛ける必要がある。

4. 月食の食分計算

月食の進行状況を知るには、太陽・月の座標から、本影と月の重なり具合である食分を求める必要がある。食分の値は図4に示す通り、本影の視半径と月の見かけの大きさ、および本影中心と月中心の角距離から計算できる。



$$\text{食分} = \frac{B}{A} = \frac{R_{umbra} + R_{moon} - s}{2R_{moon}}$$

図4 食分の計算

月と地球の本影の角距離は、天球上での月と太陽の座標が分かれば計算できる。本影は太陽と反対側

になるので、太陽と月の角距離を求めて180度引くことで、月と地球の本影の角距離が求められる。

月と太陽の位置を求めるには、専門的な天文計算が必要となるが、Pythonの天文計算ライブラリの1つであるPyEphem、あるいはSkyfieldを用いれば、簡単に任意の時刻の太陽・月の座標を得ることができる。ここでは、Skyfieldを用いて計算する例を示す。

観測地での太陽、月の座標は、あらかじめ定義した地球のオブジェクトearthと、時刻tを与えることで、以下の式から得ることができる。なお、太陽、月の座標は、地球中心から観測した値となる。

```
sun_app =
    earth.at(t).observe(sun).apparent()
moon_app =
    earth.at(t).observe(moon).apparent()
```

太陽の見かけの大きさは、太陽の半径をr_sunで与えておき、太陽までの距離はskyfieldのライブラリからkm単位で得られるので、2つの値の比をとればよい。月の見かけの大きさも、同様に計算できる。

```
r_sun = 696000
sun_dist = sun_app.distance().km
sun_rad = np.arctan2(r_sun, sun_dist)
```

次に太陽と月の視差を計算する。視差は(地球半径) / (太陽または月までの距離)で計算することができる。

```
r_earth = 6378
parallax_sun = r_earth / sun_dist
parallax_moon = r_earth / moon_dist
```

以上で求めた太陽の視半径、および太陽・月の視差の値から、3節で解説した通り本影視半径の大きさが求められる。

```
umbra =
    (parallax_moon - sun_rad + parallax_sun)
    * 51/50
```

太陽と月の角距離は、先に求めた太陽、月の座標に対して、separation_from関数を用いることで、計算される。本影は太陽の反対側になるので、太陽と月の角距離を求めて、180度引けばよい。

```
app_sep =
    abs(sun_app.separation_from(moon_app).
        radians - np.deg2rad(180))
```

以上より食分の値は

```
percent_eclipse = (umbra + moon_rad -
    app_sep) / (moon_rad * 2)
```

で計算できる。

この計算を、月食が起こると見込まれる時間帯において、適当な時間間隔、例えば1秒ごとに行い、それぞれの時点での食分の値を求める。すると最初に食分が0以上になった時刻が欠け始め(第一接触)、再び食分が0になった時刻が食の終わり(第四接触)となる。その間の食分の最大値が、その月食の最大食となり、その時の時刻が最大食の時刻となる。

5. 月食図の描画

月食の際に、月が地球の本影・半影内を通過していく様子を示すために、図1のような月食図が用いられる。

この上側の図では、地球の本影・半影を中心に固定し、時刻の経過に応じて月の位置が移動している。図の座標系は、星図と同様に上が北で、時計周りに北-西-南-東となっている。xy軸の目盛りの単位は角度である。

この図を描くために必要な本影・半影の視半径の大きさ、および月の見かけの大きさは既に4節により計算されている。また、本影の中心と月の角距離についても計算されている。

あとは、本影の中心に対して月の中心がどの方位角に位置するかが求めればよい。

2天体のなす角度はskyfieldに専用のライブラリがあり、これを使用して求めることができる。ここでは、本影の代わりに太陽に対する月の方位角を求めれば、結局求める結果を得ることができる。

```
position_angle =
    position_angle_of(moon_app.radec(),
        sun_app.radec()).radians
```

図の中心を原点として、見かけの大きさの月を、本影の中心と求められた角距離だけ離れた位置で、上記で求めた方位角の方向にプロットすることで、月食の経過を描くことができる。

例えば、2021年11月19日の部分月食について、1時間ごとに月の位置を描くと、図5のような図を描くことができる。この月食は、ほとんど皆既の部分月食と言われたが、月食図を見れば、月が本影のぎりぎり端を通り過ぎていたことが分かる。

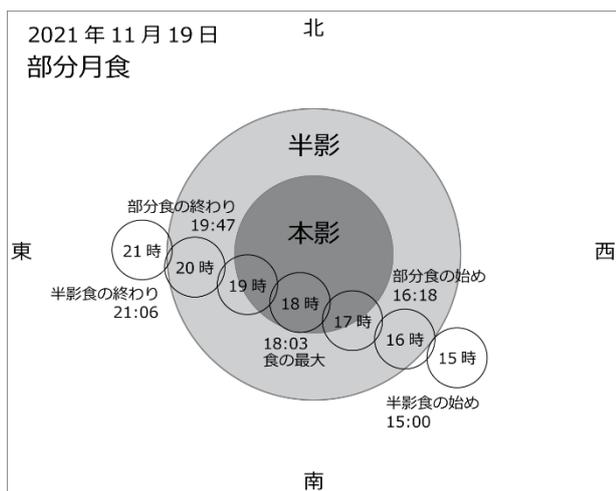


図5 月食図の描画

6. 月食世界地図の描画

月食は月が出ていれば、世界中で同じ時刻に見ることができる。ただし実際に月食が見えるかどうかは、その時間に月が昇っているかどうかによる。そこで月食が見える範囲を世界地図上に示した図もよく用いられる。

この図を描くためには、基準となる時刻に関して、月の出、または月の入りの同時刻線を描けばよい。日の出の同時刻線を描く方法については、以前の研究報告で紹介した[1]。これと同じやり方で描くことができる。

日の出同時刻線では切りのいい値での時刻に区切って目盛りをつけていたが、月食の時刻線に関しては月食の始め、最大、終わりなどの時刻についての、月の出、または月の入りの同時刻線を描くことになる。こうして作成したのが、図6の月食世界図である。こちらも、2021年11月19日の部分月食の図である。

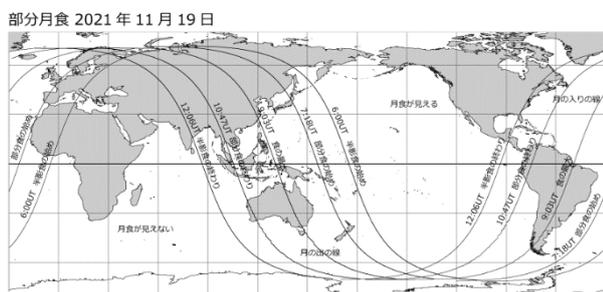


図6 月食世界図の描画

この月食は、関東より西では、既に欠けた状態の月が昇ってくる月出帯食であった。そこでさらに日本付近

を拡大したのが図7である。自身で拡大図を描くことができれば、どの地域で欠け始めからの月を見ることができるのかも詳しく知ることが可能となる。

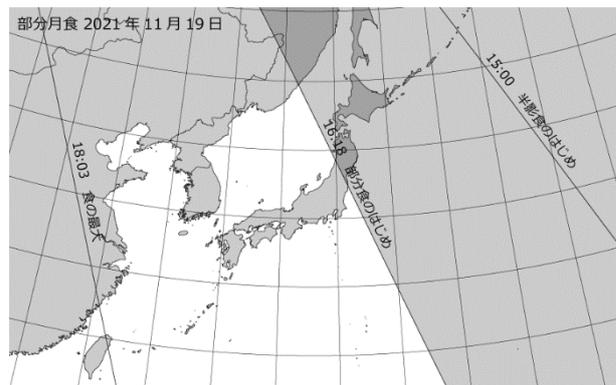


図7 日本付近の月食図

7. おわりに

今回、プログラミング言語 Python を用いて月食図を描画する方法について述べた。昨年の月食の際に、月食の経過を説明するためには、NASA や国立天文台が作成した月食図を引用するケースが多かったが、自分自身で描くことができれば、より目的に応じた図を作成することが可能となる。

例えば2021年度に見られた月出帯食の場合には、日本付近を拡大した月食図を描くことで、より各地の見え方の詳細を説明することができる。また遠い将来や過去の月食で、適当な月食図がないものであっても、自分で解説図を作ることが可能になる。

Python はほかにも天文情報を取り扱う上で、多くの可能性を秘めている。今後も更なる活用方法を検討していきたい。

参考文献

- [1]江越 航, 大阪市立科学館研究報告, 29, p25-30(2019)
- [2]江越 航, 大阪市立科学館研究報告, 31, p5-8(2021)
- [3]NASA Eclipse Web Site, <https://eclipse.gsfc.nasa.gov/lunar.html>
- [4]齊田 博, 「天文の計算教室」, 地人書館, (1977)