

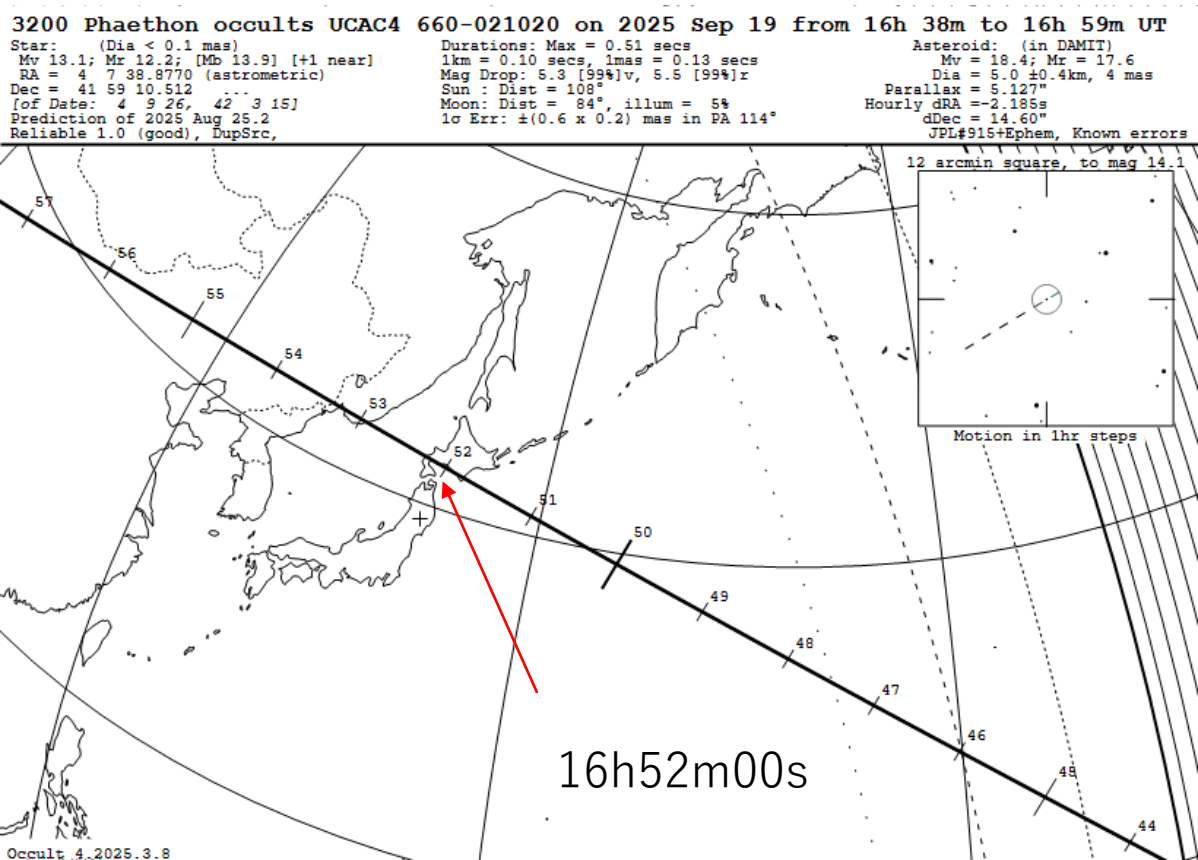
pythonベース掩蔽予報・解析 ツールSORAの簡単な紹介

2025/9/19 14h15m-14h35m

野田寛大

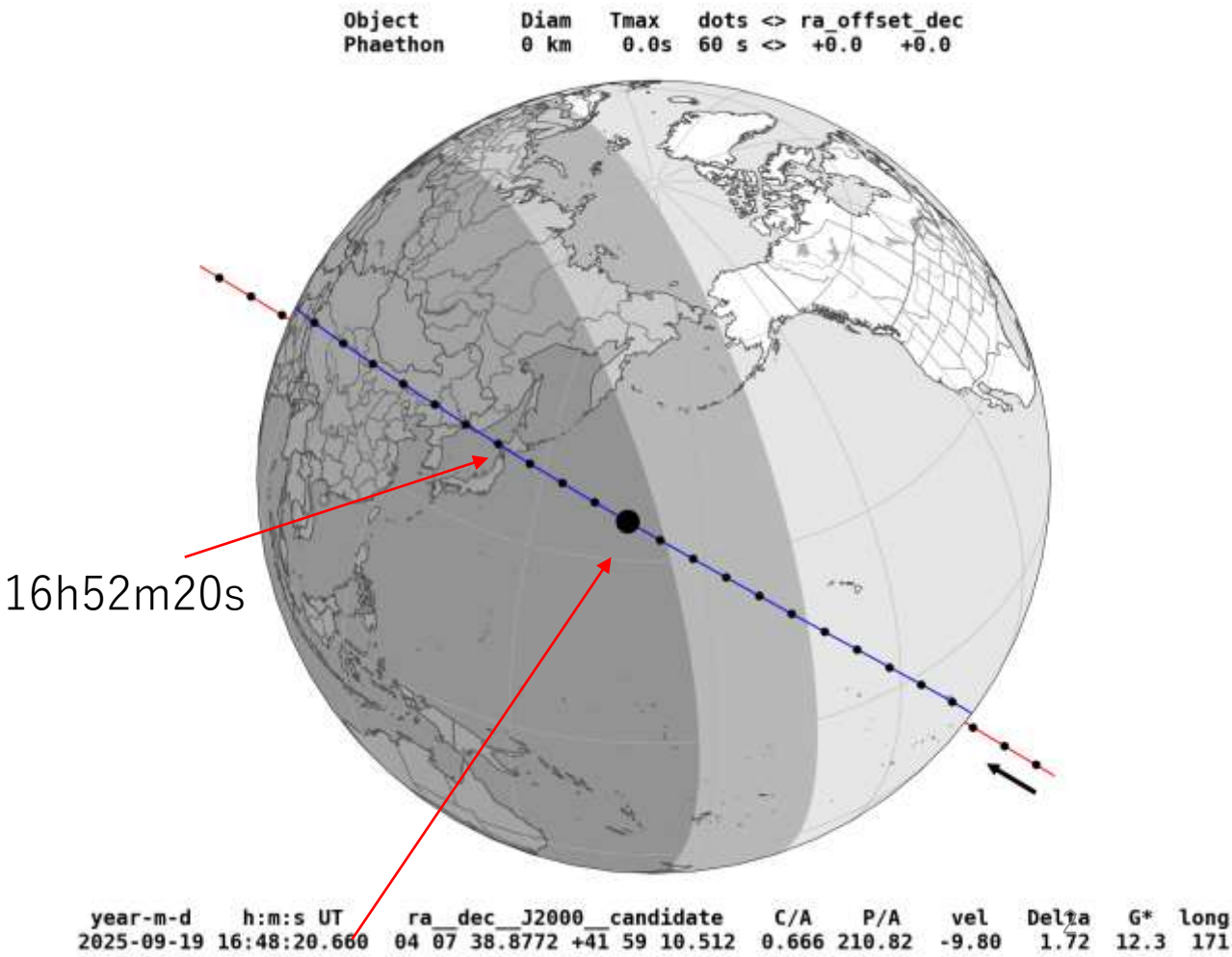
国立天文台RISE月惑星探査プロジェクト
掩蔽観測ワークショップ#5 於 北海道伊達市

Occult4との比較



SORA出力

点の間隔は1分間 zoom=1 (全球図)



伝えたいこと

- 配布するものの説明
- 情報へのポインタ（ウェブ）
- 「SORAはpythonで動くツールで、簡単なコードを自分で書く必要があります」
- 「コードの例はウェブにあります」
- インストールの例
- コードの例、出力の例

自力で作業するための配布資料

- [wslでのインストール.txt](#)
- [addnasajplperm.txt](#) [request.getで証明書のエラーが出る場合に証明書に追加するテキスト]
- [Phaethon_20250919.ipynb](#) jupyter notebook形式ファイル
今回の現象を予報するコード

SORA

- Open Source Software (OSS)
- Pythonで書かれている
- マニュアルがしかりしている
- 論文に謝辞を



SORA: Stellar occultation reduction and analysis

A. R. Gomes-Júnior^{1,2★} B. E. Morgado^{1,2,3,4★} G. Benedetti-Rossi^{1,2,3★} R. C. Boufleur,²
F. L. Rommel,^{2,4} M. V. Banda-Huarcá,^{2,4} Y. Kilic,^{5,6} F. Braga-Ribas^{2,3,4,7} and B. Sicardy³

¹UNESP - São Paulo State University, Grupo de Dinâmica Orbital e Planetologia, CEP 12516-410, Guaratinguetá, SP 12516-410, Brazil

²Laboratório Interinstitucional de e-Astronomia - LIneA and INCT do e-Universo, Rua Gal. José Cristino 77, Rio de Janeiro, RJ 20921-400, Brazil

³LESIA, Observatoire de Paris, Université PSL, CNRS, Sorbonne Université, Univ. Paris Diderot, Sorbonne Paris Cité, 5 place Jules Janssen, F-92195 Meudon, France

⁴Observatório Nacional/MCTIC, R. General José Cristino 77, Rio de Janeiro, RJ 20.921-400, Brazil

⁵Akdeniz University, Faculty of Sciences, Department of Space Sciences and Technologies, 07058 Antalya, Turkey

⁶TÜBİTAK National Observatory, Akdeniz University Campus, 07058 Antalya, Turkey

⁷Federal University of Technology-Paraná (UTFPR / DAFIS), Curitiba, Brazil

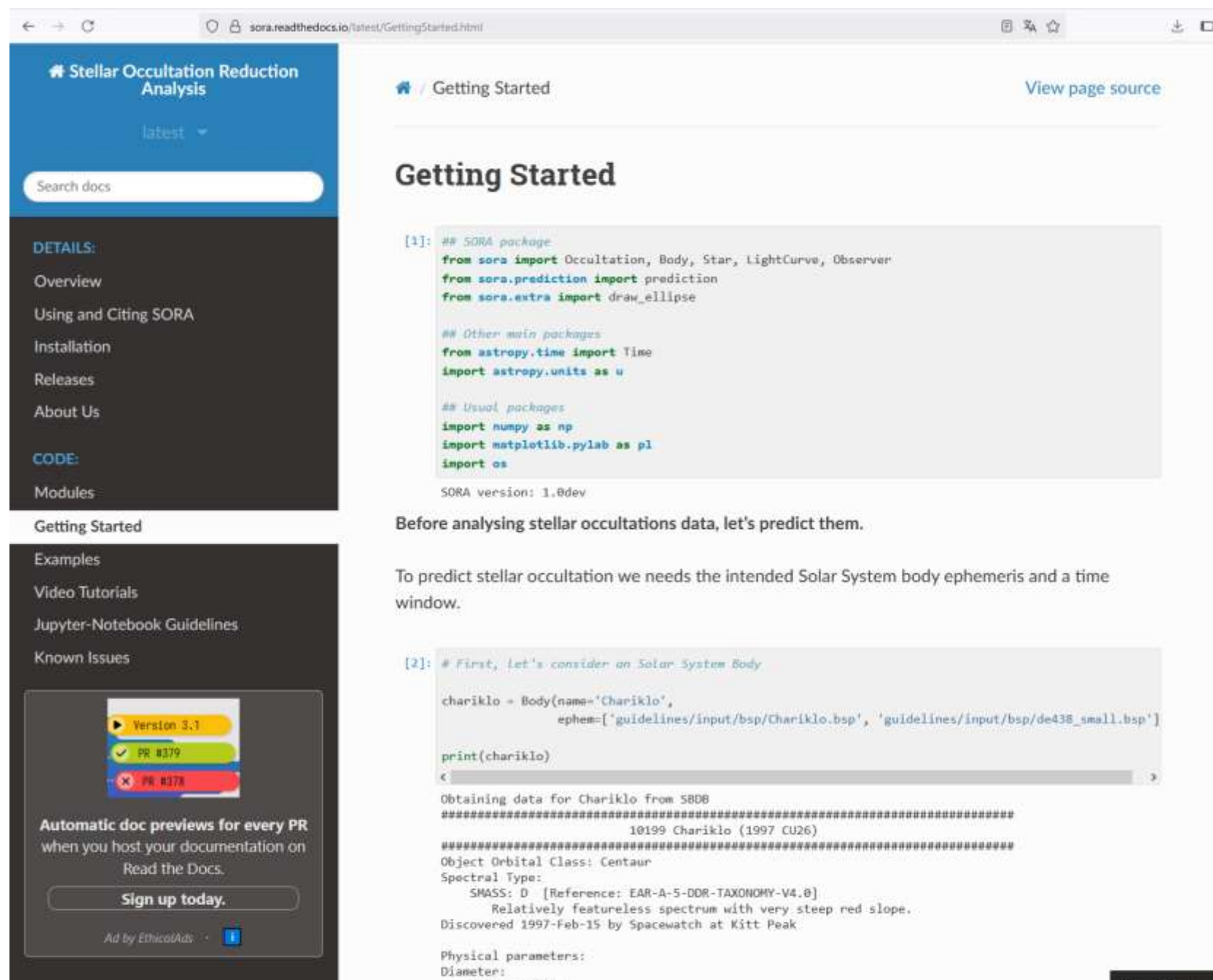
Accepted 2022 January 4. Received 2022 January 4; in original form 2021 August 25

ABSTRACT

The stellar occultation technique provides competitive accuracy in determining the sizes, shapes, astrometry, etc., of the occulting body, comparable to in-situ observations by spacecraft. With the increase in the number of known Solar system objects expected from the LSST, the highly precise astrometric catalogs, such as Gaia, and the improvement of ephemerides, occultations observations will become more common with a higher number of chords in each observation. In the context of the Big Data era, we developed SORA, an open-source python library to reduce and analyse stellar occultation data efficiently. It includes routines from predicting such events up to the determination of Solar system bodies' sizes, shapes, and positions.

Key words: methods: data analysis – software: data analysis – occultations .

最初のページ



The screenshot displays the 'Getting Started' page of the SORA documentation. The page is divided into a sidebar on the left and a main content area on the right.

Sidebar:

- Stellar Occultation Reduction Analysis
- latest
- Search docs
- DETAILS:
 - Overview
 - Using and Citing SORA
 - Installation
 - Releases
 - About Us
- CODE:
 - Modules
- Getting Started
- Examples
- Video Tutorials
- Jupyter-Notebook Guidelines
- Known Issues

Main Content Area:

- Getting Started
- View page source
- ## Getting Started
- [1]:

```
## SORA package
from sora import Occultation, Body, Star, LightCurve, Observer
from sora.prediction import prediction
from sora.extra import draw_ellipse

## Other main packages
from astropy.time import Time
import astropy.units as u

## Usual packages
import numpy as np
import matplotlib.pyplot as plt
import os
```

SORA version: 1.0dev

Before analysing stellar occultations data, let's predict them.

To predict stellar occultation we need the intended Solar System body ephemeris and a time window.

- [2]:

```
# First, let's consider an Solar System Body

chariklo = Body(name='Chariklo',
                ephem=['guidelines/input/bsp/Chariklo.bsp', 'guidelines/input/bsp/de438_small.bsp'])

print(chariklo)
```

Obtaining data for Chariklo from 5808
=====

10199 Chariklo (1997 CU26)
=====

Object Orbital Class: Centaur
Spectral Type:
SMASS: D [Reference: EAR-A-5-DDR-TAXONOMY-V4.0]
Relatively featureless spectrum with very steep red slope.
Discovered 1997-Feb-15 by Spacewatch at Kitt Peak

Physical parameters:
Diameter:

出来る事

- 予報
- 別の方法で得たライトカーブを解析してD,R時刻を求める
- 整約図の作成、フィッティング
- (obj形式の3Dモデルを重ね書き)

得失

利点？

- linux, macでも動く（はず）
- コードの中身が見られる
 - しばらくは正しく保守されると予想
- マニュアル（英語）がある
- 自分でコードを自由に書け、図の作成も比較的自由度が高い
- occult4にないものも軌道データを用意すれば予報可能
- **そこそこ自動化が可能**
- プロが論文作成に使うようだ

欠点？

- インストールするのに色々知識が必要
- **pythonで自分で書く必要がある**
- 予報では逐一ファイルをダウンロードするので**ネットワークが必要**

おすすめ？開発環境

Windowsの場合

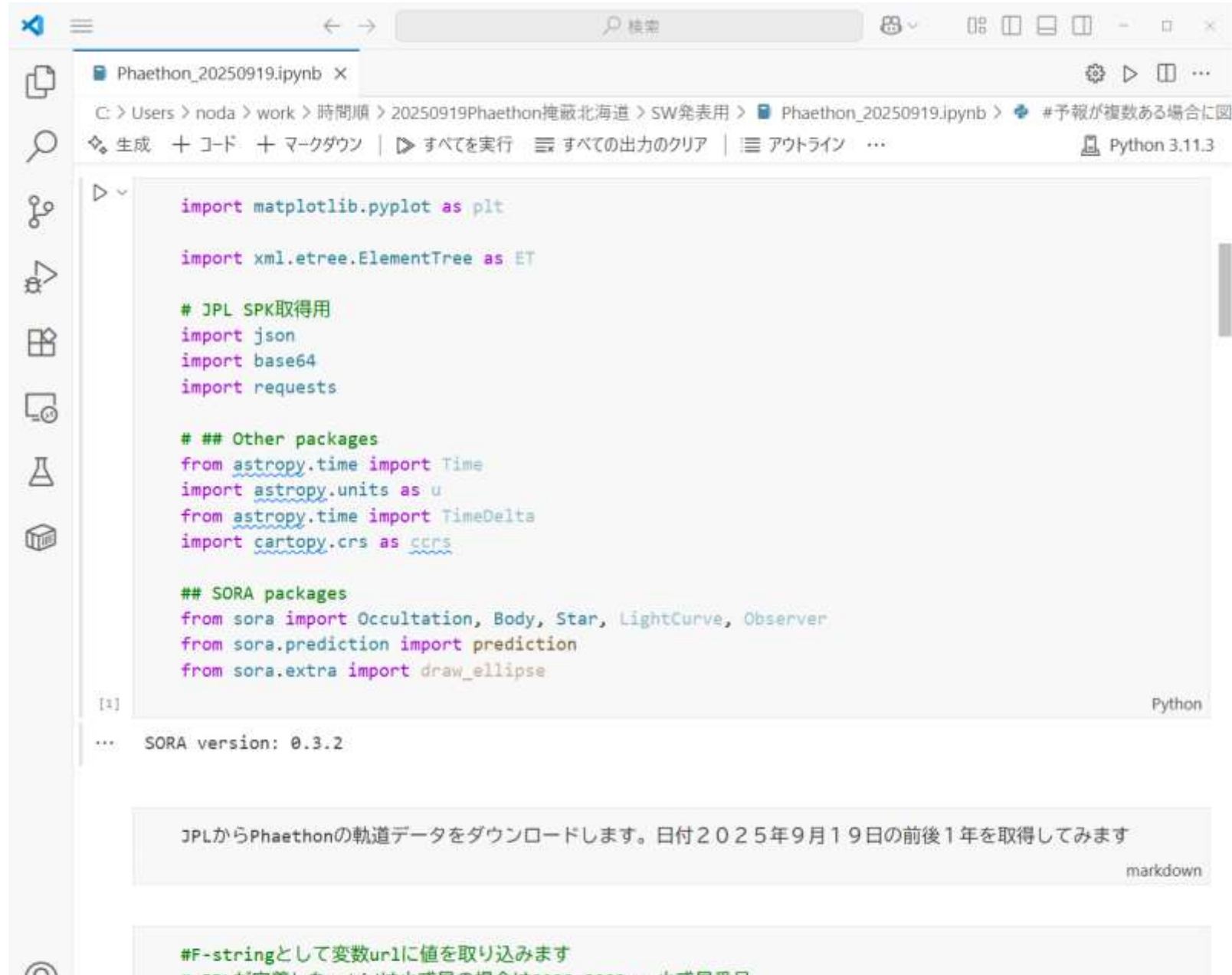
- WSL (windows subsystem for Linux)の仮想環境を利用
- jupyter notebook形式ファイルをVS code上で動かす
- 「[wslでのインストール.txt](#)」 参考
 - 註：JPL horizonsに接続するところでエラーが出る場合は手順⑤参照
- de442s.bsp等の一般的なJPL暦を以下から取得
https://naif.jpl.nasa.gov/pub/naif/generic_kernels/spk/planets
 - JPLから取得する *.bspファイル(SPK)の中身を確認したい場合はNAIF toolkitをwslにインストール (C、Fortran)

Macの場合 ??? (AIに聞いてみてください)

例：今回のPhaethon予報

VS codeでの表示例

- 配布物
Phaethon_20250919.ipynb
 - jupyter notebook形式
- pythonコードにコピーしてpython hoge.py等としても実行可能



```
import matplotlib.pyplot as plt

import xml.etree.ElementTree as ET

# JPL SPK取得用
import json
import base64
import requests

## Other packages
from astropy.time import Time
import astropy.units as u
from astropy.time import TimeDelta
import cartopy.crs as ccrs

## SORA packages
from sora import Occultation, Body, Star, LightCurve, Observer
from sora.prediction import prediction
from sora.extra import draw_ellipse
```

[1] Python

... SORA version: 0.3.2

JPLからPhaethonの軌道データをダウンロードします。日付2025年9月19日の前後1年を取得してみます

markdown

```
#F-stringとして変数urlに値を取り込みます
```

予報の流れ

- ①python modulesをインポート
- ②JPLから小天体（3200 Phaethon）軌道データをダウンロード
 - ※request.getでエラーが出る場合は証明書を追加する
- ③オブジェクト4つ作成
 - 小天体phaethon
 - 予報値pred
 - 恒星star_occ
 - 掩蔽occ
- ④予報図作成

流れ①import

①暦取得

```
import matplotlib.pyplot as plt

# JPL SPK取得用
import json
import base64
import requests

# ## Other packages
from astropy.time import Time
import astropy.units as u
from astropy.time import TimeDelta
import cartopy.crs as ccrs

## SORA packages
from sora import Occultation, Body, Star,
LightCurve, Observer
from sora.prediction import prediction
from sora.extra import draw_ellipse
```

#F-stringとして変数urlに値を取り込みます
JPLが定義したSPK ID(spkid)は小惑星の場合は2000,0000 + 小惑星番号

```
spkid = 20003200
start = "2024-09-19"
stop = "2026-09-19"
```

必要に応じて書き換える

```
url = f"https://ssd.jpl.nasa.gov/api/horizons.api"
url += "?format=json&EPHEM_TYPE=SPK&OBJ_DATA=NO"
url += f"&COMMAND='DES%3D{spkid}%3B'&START_TIME='{start}'&STOP_TIME='{stop}'"
```

※このデータ要求書式の書き方については以下を参照
<https://ssd-api.jpl.nasa.gov/doc/horizons.html>

```
response = requests.get(url)
if response.status_code == 200:
    data = response.json()
    if "spk_file_id" in data:
        spk_filename = data["spk_file_id"] + ".bsp"
        with open(spk_filename, "wb") as f:
            f.write(base64.b64decode(data["spk"]))
        print(f"wrote SPK content to {spk_filename}")
    elif response.status_code == 400:
        data = response.json()
        print("MESSAGE: {}".format(data.get("message", json.dumps(data, indent=2))))
```



流れ②オブジェクト4つ作成

①天体オブジェクト："phaethon"

```
phaethon = Body(name='Phaethon', diameter=5, database=None,  
                spkid=20003200,  
                ephem=['20003200.bsp', 'de442s.bsp'])
```

②予報オブジェクト："pred"

```
pred = prediction(body=phaethon, time_beg='2025-09-19 00:00:00',  
                 time_end='2025-09-20 00:00:00', mag_lim=14)
```

③星オブジェクト：今回は予報が1つなので予報[0]を指定.

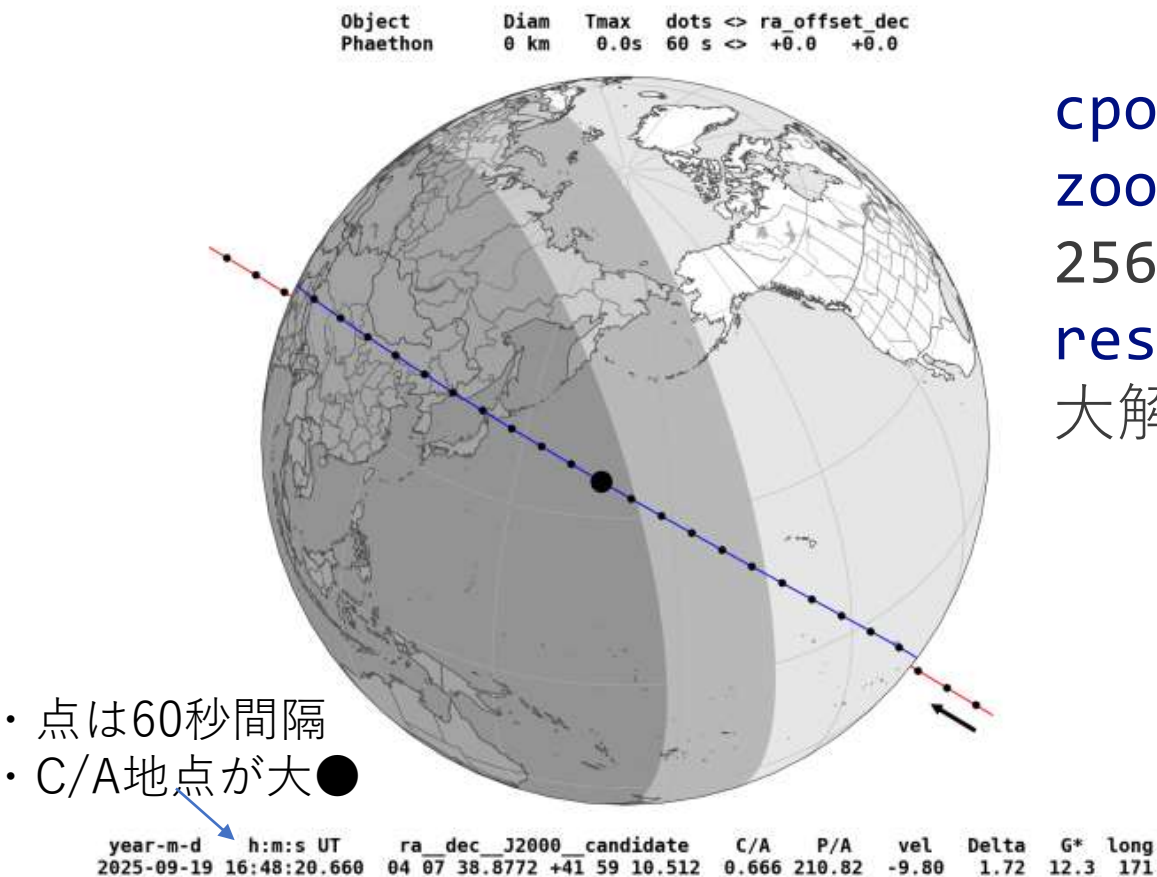
```
star_occ = Star(code= pred['GaiaDR3 Source ID'][0])
```

④掩蔽オブジェクト：予報値で得られた時刻を指定

```
occ = Occultation(star=star_occ, body=phaethon, time=pred['Epoch'][0])
```

流れ③予報図

サイト指定sites=sites、
間隔指定cpoint=1 [s]

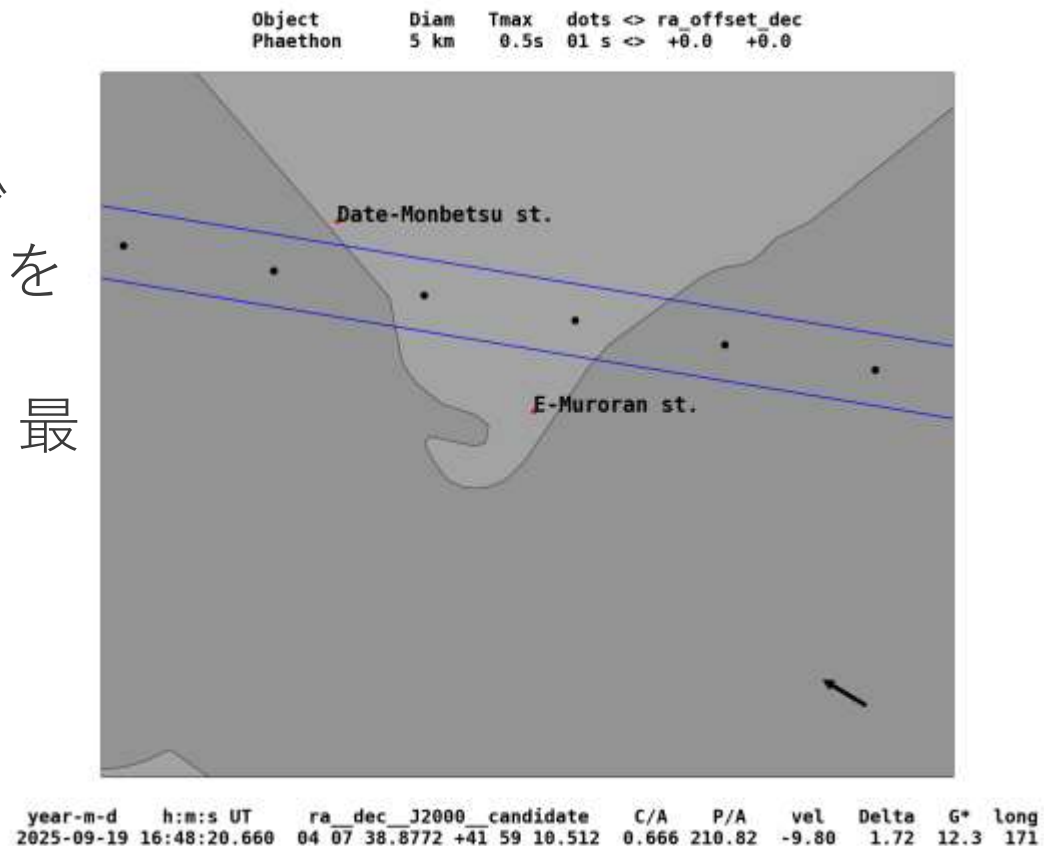


- 点は60秒間隔
- C/A地点が大●

occ.plot_occ_map(nameimg="occ_local2")

掩蔽(または予報) objectのメソッド

cpoints=1, 1秒
zoom=256, 地球を
256分割
resolution=1 最
大解像度



```
sites= {
    "Date-Monbetsu st.": [140.85633650665656, 42.46941118201754, 0, 0, "red", "."],
    "E-Muroran st.": [141.02528514543485, 42.34876141396795, 0, 0, "red", "."]
}
pred[0].plot_occ_map(centerproj=[141.02, 42.34],
    nameimg="pred_centerproj_local256_date",
    sites=sites, site_name=True, cpoints=1, zoom=256, resolution=1)
```

(参考)この後の観測データの解析 通常は以下の通り

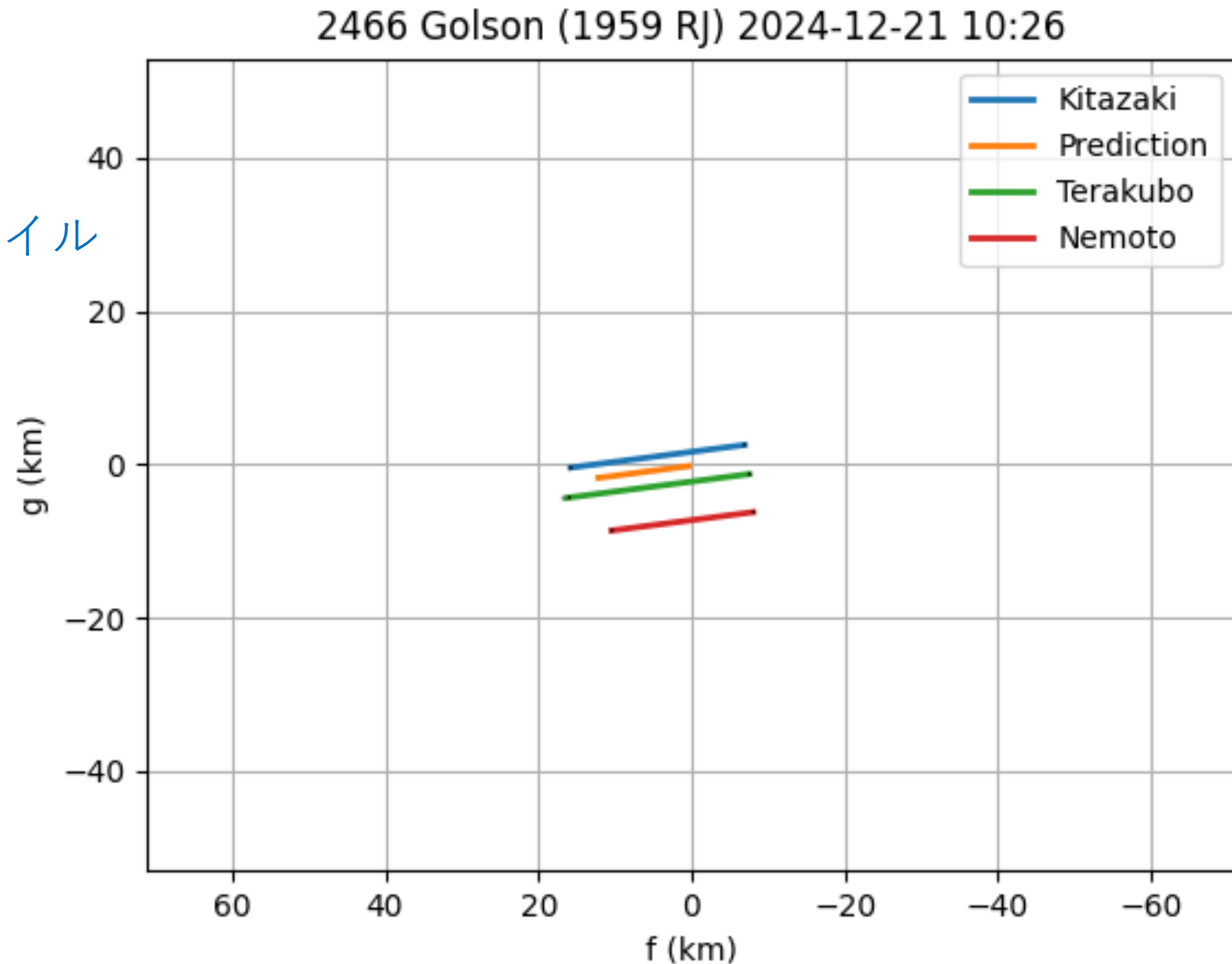
- オブジェクト2つ作成
 - オブザーバobs
 - ライトカーブlc
- 掩蔽オブジェクトoccに弦を追加 e.g.
`occ.chords.add_chord(observer=obs, lightcurve=lc)`
- ライトカーブ解析 `occ_lcfit()`, フィットのchi2プロット、フィットモデルプロット `lc.plot_lc()`
- 弦プロット `occ.chords.plot_chords()`
- 楕円フィット `occ.fit_ellipse()`, 楕円プロット `draw_ellipse()`

他にできる事

- SORA本体の機能でもobj形式の形状モデルを扱えるようだが、特定の天体に限られている模様。また自転軸、回転位相が指定できないかもしれない

Reduction例

XML化されたサマリデータファイル
から**自動的に**reduction plotを
作成した例



まとめ

- 3 枚目「伝えたいこと」ご参照
-

- 「SORAはpythonで動くツールで、簡単なコードを自分で書く必要があります」
 - 「コードの例はウェブにあります」
 - インストールの例
 - コードの例、出力の例
-