# 3D Planets and Celestial Bodies

*MATLAB Implementation*

Tamas Kis | kis@stanford.edu

TAMAS KIS
https://github.com/tamaskis

# Contents

# 1 Download and Installation

## 1.1 Download from MATLAB Central's File Exchange

The `planet3D` function is available for download on MATLAB® Central's File Exchange at `test`.

## 1.2 Download from GitHub

The `planet3D` function is available for download on GitHub® at `https://github.com/tamaskis/planet3D-MATLAB`.

## 1.3 Files Included With Download

There are five files and one folder (storing the images) included in the downloaded `zip` file:

1. `3D Planets and Celestial Bodies - MATLAB Implementation - MATLAB Implementation.pdf` – *this PDF*
2. `EXAMPLES.M` – *examples for using the* `planet3D` *function*
3. **`Images` – *Folder that stores images needed to render celestial bodies. DO NOT DELETE. KEEP IN THE SAME FOLDER AS* `planet3D.m`**
4. `LICENSE` – *license for the* `planet3D` *function*
5. **`planet3D.m` – *MATLAB function for drawing planets (and Sun, Moon, and Pluto)***
6. `README.md` – *markdown file for GitHub documentation*

## 1.4 Accessing the `planet3D` Function in a MATLAB Script

There are **three** options for accessing the `planet3D` function in a MATLAB script:

1. Copy the `planet3D` function to the *end* of your MATLAB script.
2. Place the `planet3D.m` file in the same folder as the MATLAB script.
3. Place the `planet3D.m` file into whatever folder you want, and then use the `addpath(folderName)` command[1] where the `folderName` parameter is a string that stores the filepath of the folder that `planet3D.m` is in *relative to* the folder that your script is in.

**In all three cases, the `Images` folder *must* be in the same folder as `planet3D.m`.**

[1] `https://www.mathworks.com/help/matlab/ref/addpath.html`

# 2 planet3D

Creates high-resolution renderings of the Earth and the major celestial bodies in our solar system for space mechanics applications.

## Syntax

```
planet3D(planet)
planet3D(planet,position)
planet3D(planet,position,units)
planet3D(planet,position,units,background)
```

**NOTE:** If you don't want to specify `position` or `units`, for example, but *do* want to specify `background`, then you would use the syntax `planet3D(planet,[],[],background)`. The empty brackets are needed as placeholders, because `background` can only be read correctly if it is the *fourth* passed parameter. However, we don't need placeholders if we aren't "skipping over" parameters. For example, if we wanted to specify just the `position`, we could use the syntax `planet3D(planet,position)`.

## Description

`planet3D(planet,position,units,background)` draws a celestial body.

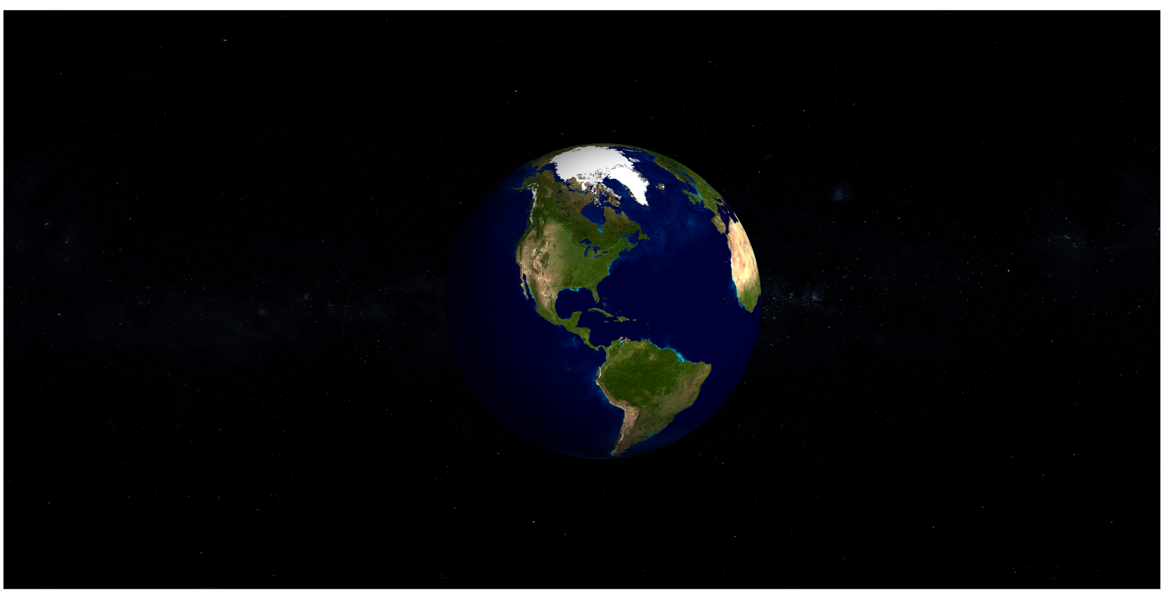
| | |
|---|---|
| `planet` | `'Sun'`, `'Moon'`, `'Mercury'`, `'Venus'`, `'Earth'`, `'Earth Cloudy'`, `'Earth Night'`, `'Earth Night Cloudy'`, `'Mars'`, `'Jupiter'`, `'Saturn'`, `'Uranus'`, `'Neptune'`, or `'Pluto'`. |
| `position` *(optional)* | Specifies the position of the celestial body. If `position` is not specified, the function defaults to $(0, 0, 0)$. NOTE: If you are also specifying `units`, make sure you input `position` in the correct units (i.e. in the units you intend to use). |
| `units` *(optional)* | Specifies the units the celestial body should be drawn in. Units available are `'km'`, `'AU'`, `'m'`, `'ft'`, `'mi'`, and `'nmi'`. |
| `background` *(optional)* | Specifies the background to use. Backgrounds available are `'Stars'`, `'Milky Way'`, and `'Black'`. |

# 3 Example Plots

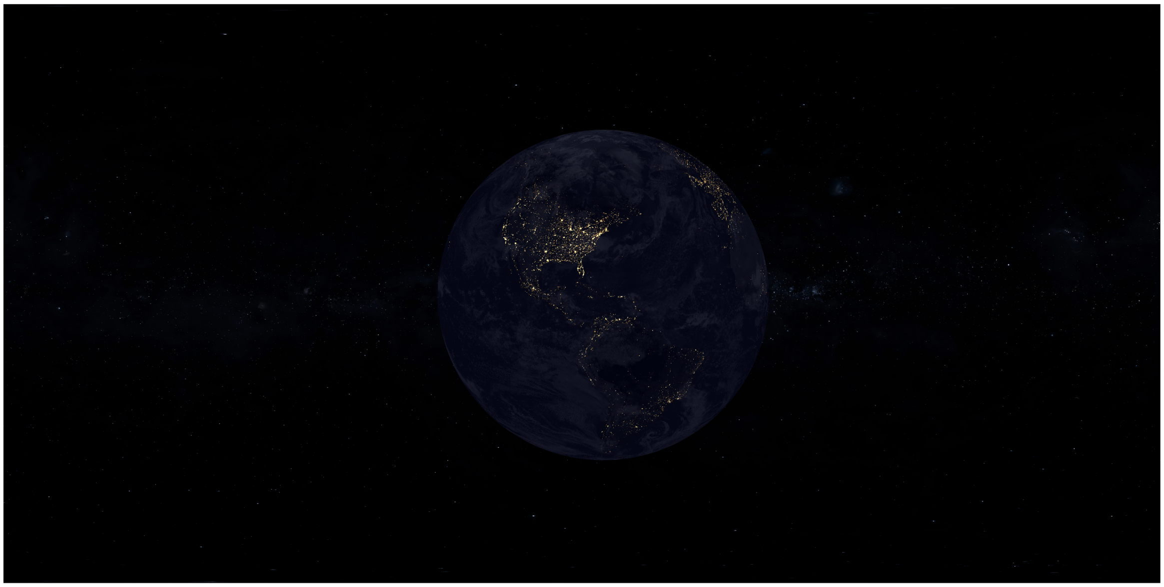
## 3.1 Earth

### 3.1.1 | Earth (With Clouds)

### 3.1.2 | Earth (No Clouds)



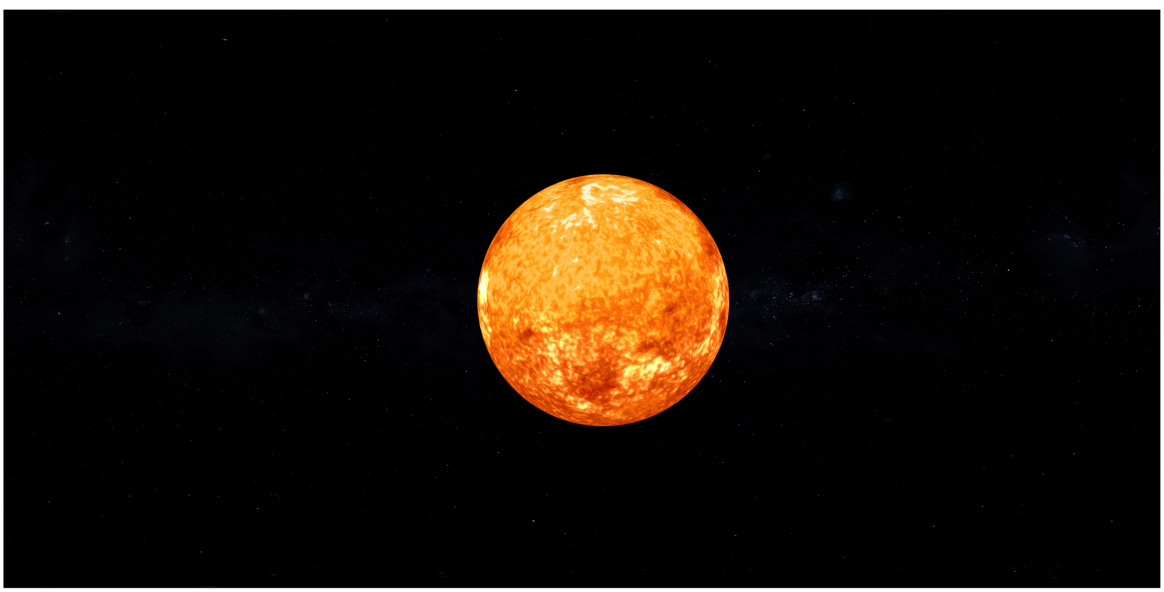
### 3.1.3 | Earth (Night, With Clouds)



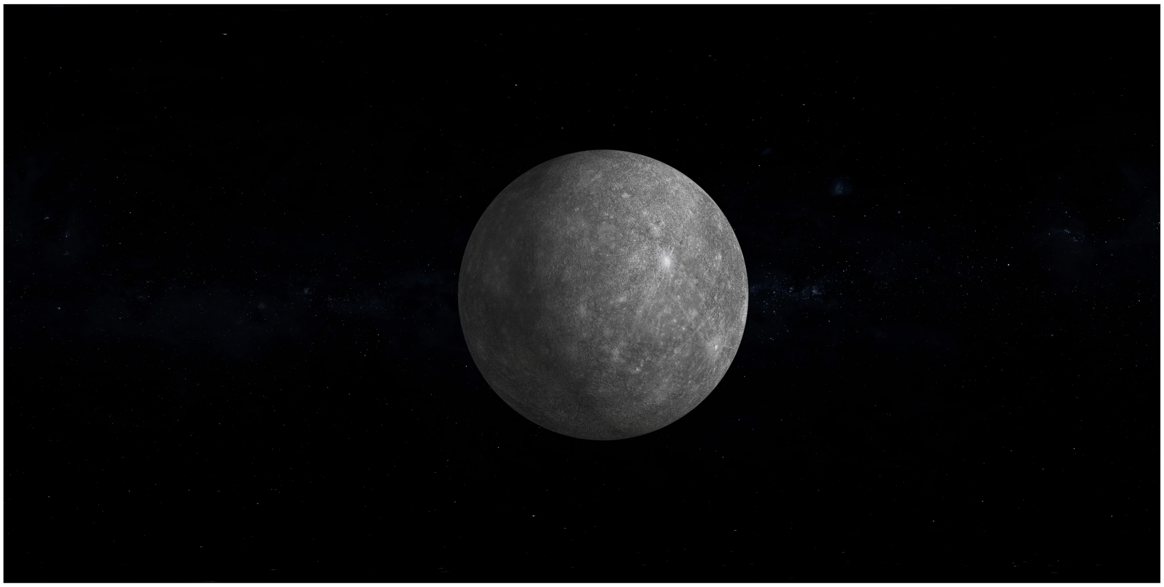
### 3.1.4 | Earth (Night, No Clouds)

## 3.2 Moon

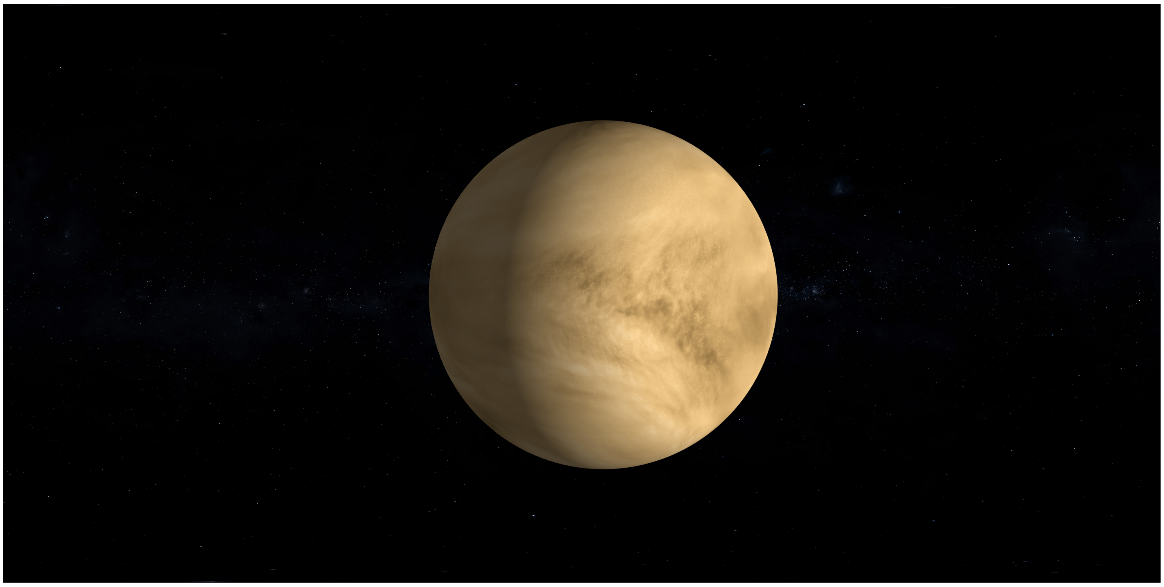
## 3.3 Sun



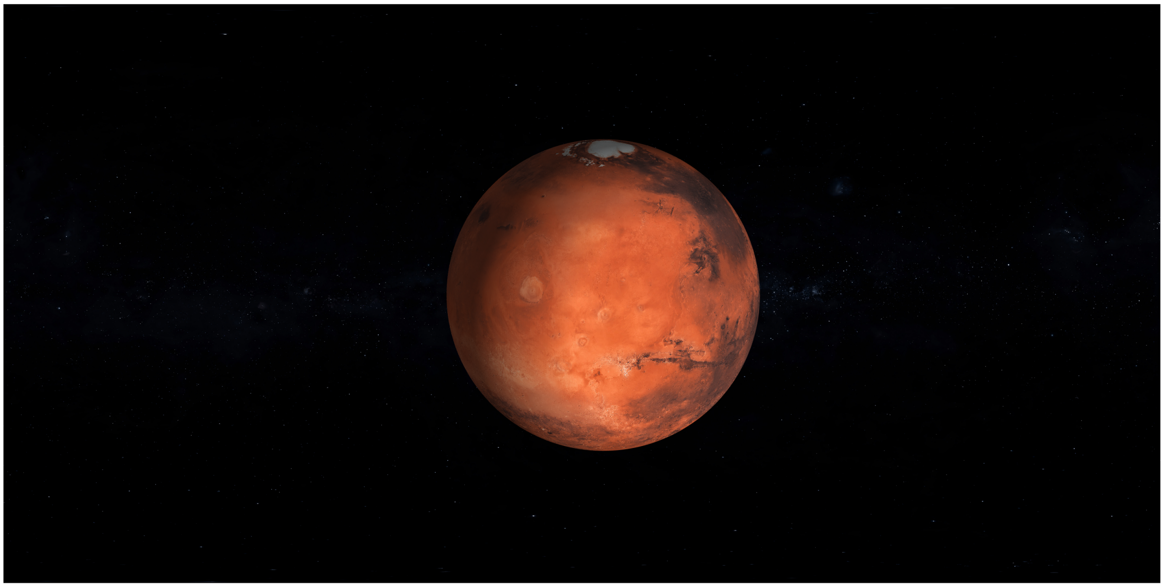
## 3.4 Mercury



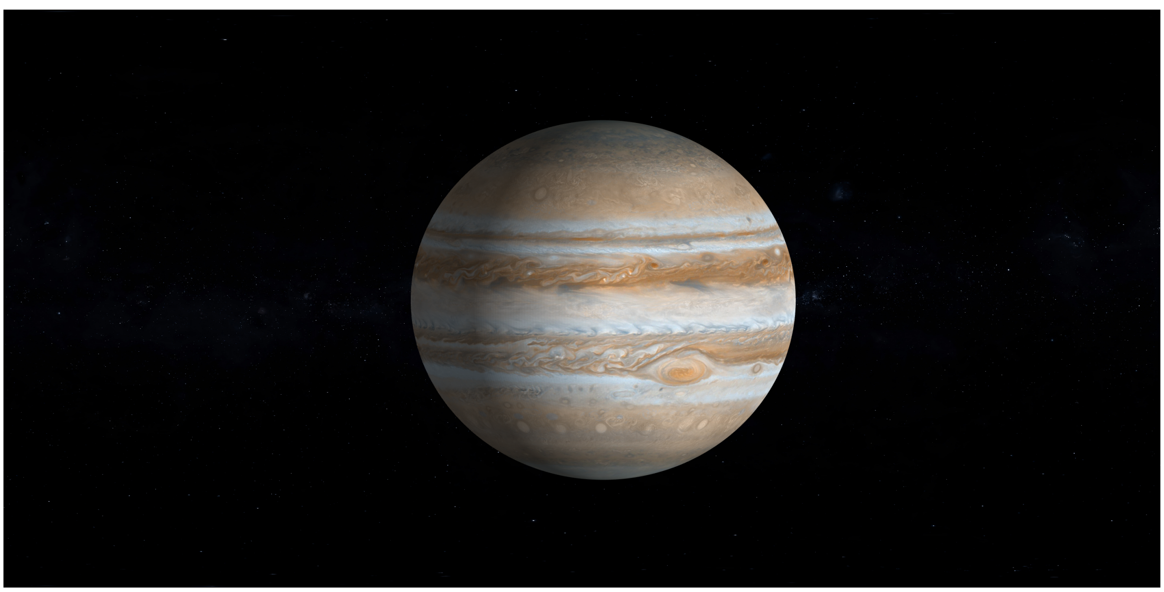
## 3.5 Venus



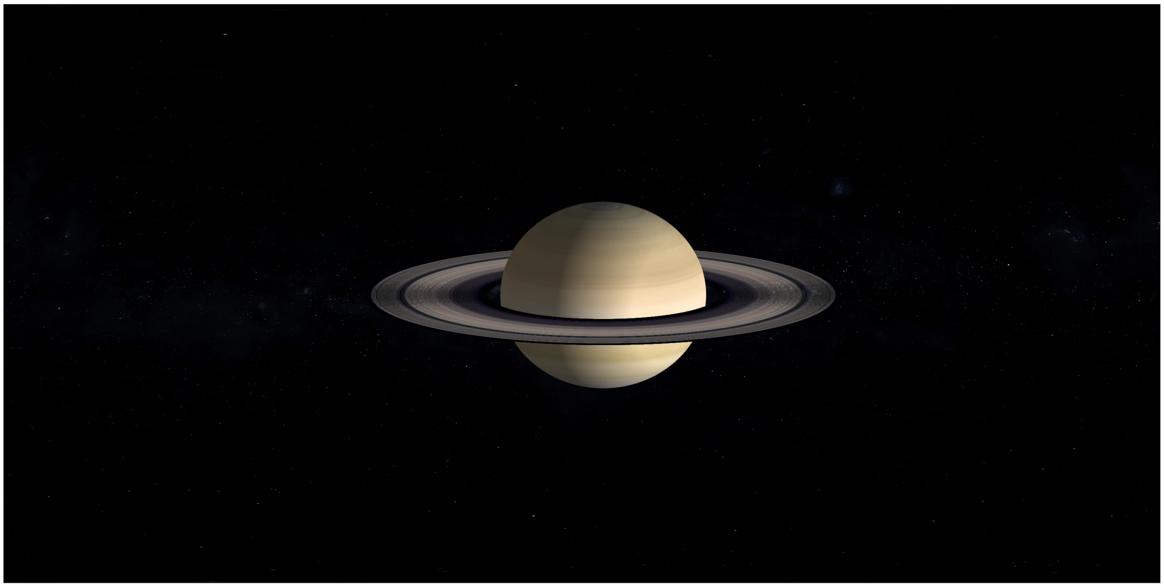
## 3.6 Mars



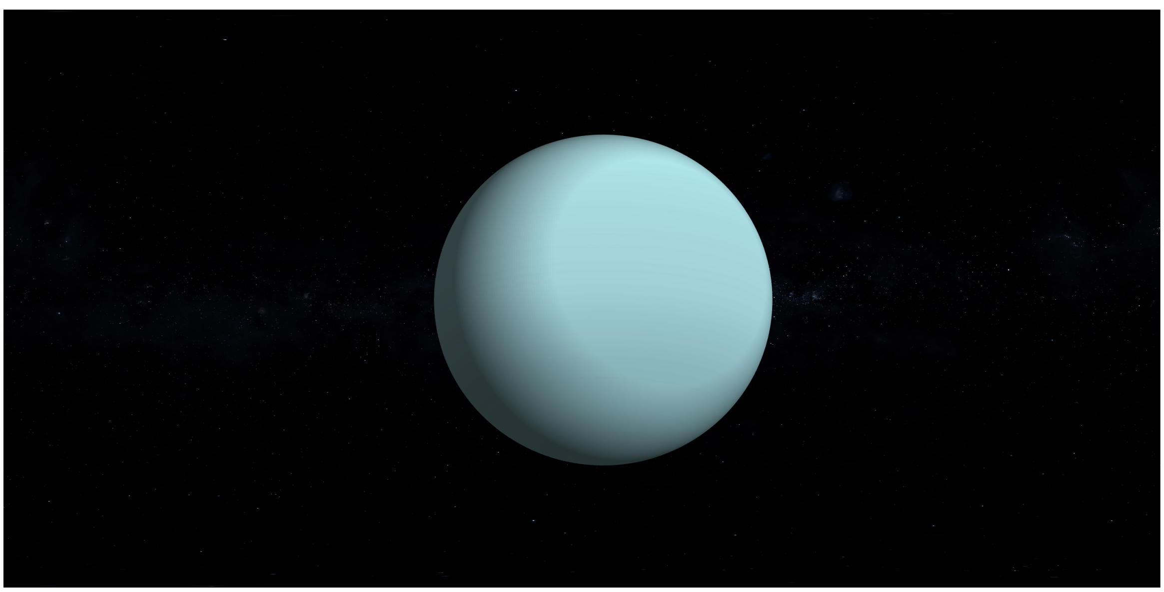
## 3.7 Jupiter



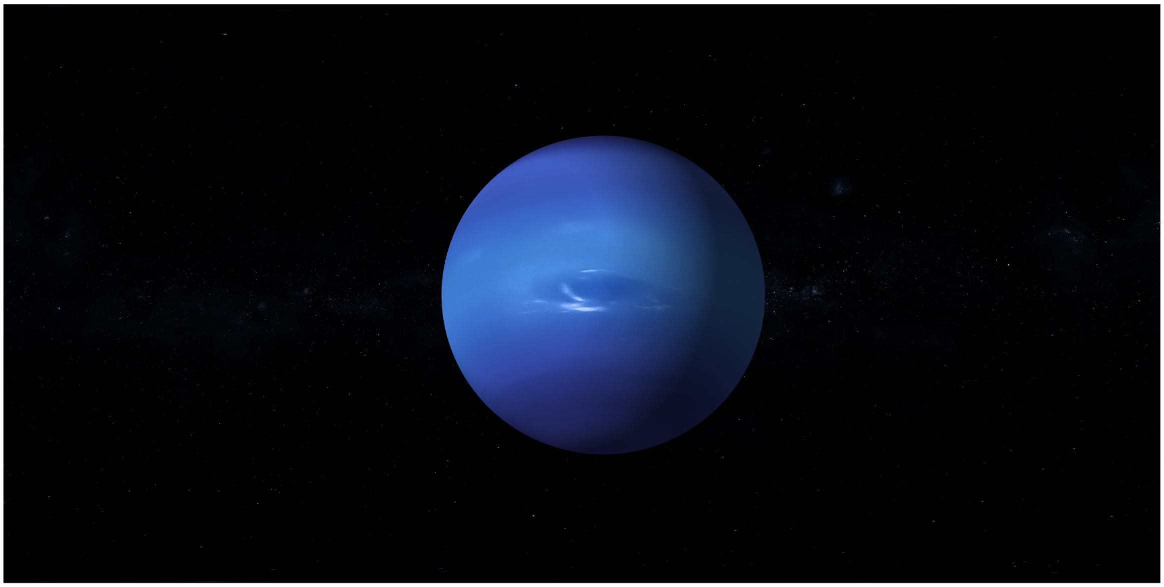
## 3.8 Saturn



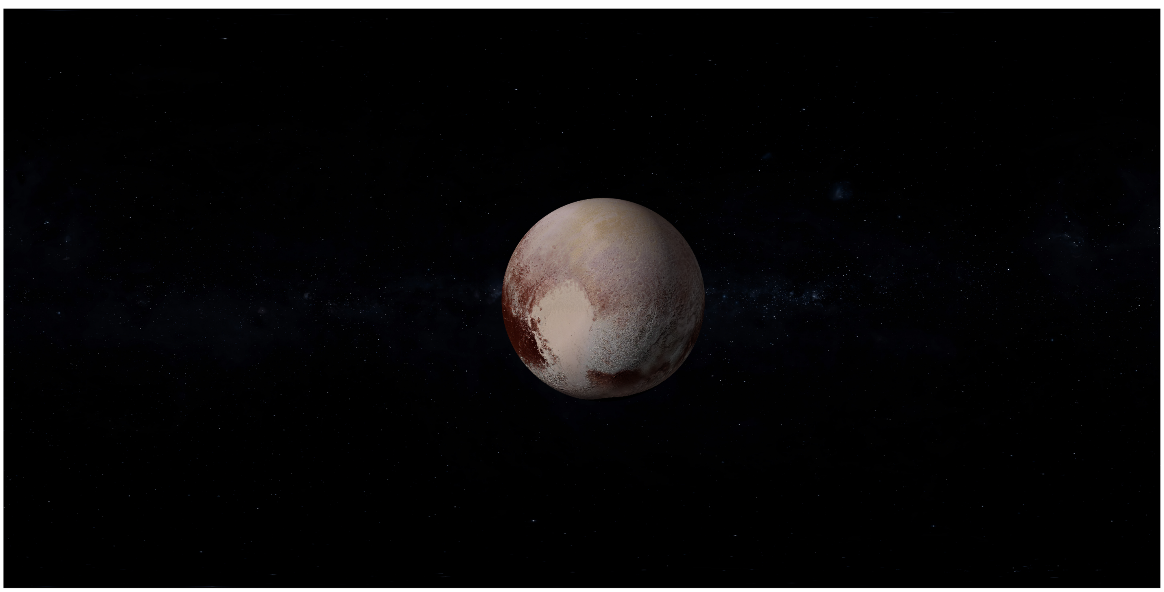
## 3.9 Uranus



## 3.10 Neptune



## 3.11 Pluto



# A Data and Constants

## A.1 Astronomical Data

| Planet/Body | Equatorial Radius | | Flattening | |
|---|---|---|---|---|
| | Value [km] | Source | Value | Source |
| Sun | 696000 | [13, Table D-5, p. 1043] | 0.000 009 | [11] |
| Moon | 1738.0 | [13, Table D-3, p. 1041] | 0.0012 | [7] |
| Mercury | 2439.0 | [13, Table D-3, p. 1041] | 0.0000 | [6] |
| Venus | 6052.0 | [13, Table D-3, p. 1041] | 0.000 | [14] |
| Earth | 6378.1363 | [13, Table D-3, p. 1041] | 0.003 352 813 1 | [13, Table D-3, p. 1041] |
| Mars | 3397.2 | [13, Table D-3, p. 1041] | 0.006 476 30 | [13, Table D-3, p. 1041] |
| Jupiter | 71492.0 | [13, Table D-4, p. 1042] | 0.064 874 4 | [13, Table D-4, p. 1042] |
| Saturn | 60268.0 | [13, Table D-4, p. 1042] | 0.097 962 4 | [13, Table D-4, p. 1042] |
| Uranus | 25559.0 | [13, Table D-4, p. 1042] | 0.022 927 3 | [13, Table D-4, p. 1042] |
| Neptune | 24764.0 | [13, Table D-4, p. 1042] | 0.0171 | [13, Table D-4, p. 1042] |
| Pluto | 1151.0 | [13, Table D-4, p. 1042] | .0.0 | [13, Table D-4, p. 1042] |

## A.2 Semi-Minor Axes

For MATLAB's `ellipsoid` function, we need the semi-minor axis, $b$, which can be calculated as

$$b = a(1 - f)$$

where $a$ is the semi-major axis (assumed to be the equatorial radius) and $f$ is the flattening [3, p. 7-4 (p. 73 in PDF)].

## A.3 Saturn's Rings

Saturn's rings range from $7000\ \mathrm{km}$ to $80000\ \mathrm{km}$ from the surface of the planet [9].

## A.4 Unit Conversions

### Kilometers to Astronomical Units [13]:

$$1\ \mathrm{AU} = 149597870\ \mathrm{km} \quad \rightarrow \quad \boxed{1\ \mathrm{km} = \frac{1}{149597870}\ \mathrm{AU}}$$

### Kilometers to Meters:

$$\boxed{1\ \mathrm{km} = 1000\ \mathrm{m}}$$

## Kilometers to Feet:

$$1\text{ km} = \left(\frac{1000\text{ m}}{1\text{ km}}\right)\left(\frac{100\text{ cm}}{1\text{ m}}\right)\left(\frac{1\text{ in}}{2.54\text{ cm}}\right)\left(\frac{1\text{ ft}}{12\text{ in}}\right) \quad \rightarrow \quad \boxed{1\text{ km} = \frac{100000}{30.48}\text{ ft}}$$

## Kilometers to Miles:

$$1\text{ km} = \left(\frac{100000/30.48\text{ ft}}{\text{km}}\right)\left(\frac{1\text{ mi}}{5280\text{ ft}}\right) \quad \rightarrow \quad \boxed{1\text{ km} = \frac{100000}{160934.4}\text{ mi}}$$

## Kilometers to Nautical Miles:

$$1\text{ nmi} = 1852\text{ m} = 1.852\text{ km} \quad \rightarrow \quad \boxed{1\text{ km} = \frac{1}{1.852}\text{ nmi}}$$

# B Image Sources

| Image | File Name | Source | Copyright/License |
|---|---|---|---|
| Sun | `Sun.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) [1, 10] |
| Moon | `Moon.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) [1, 10] |
| Mercury | `Mercury.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) [1, 10] |
| Venus | `Venus.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) [1, 10] |
| Earth (Day) | `Earth.png` | [12] | none [5, 12] |
| Earth (Night) | `Earth Night.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) [1, 10] |
| Clouds | `Clouds.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) [1, 10] |
| Mars | `Mars.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) [1, 10] |
| Jupiter | `Jupiter.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) [1, 10] |
| Saturn | `Saturn.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) [1, 10] |
| Saturn Rings | `Saturn Rings.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) [1, 10] |
| Uranus | `Uranus.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) [1, 10] |
| Neptune | `Neptune.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) [1, 10] |
| Pluto | `Pluto.png` | [8] | none [8] |
| Milky Way | `Milky Way.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) |
| Stars | `Stars.png` | [10] | CC Attribution 4.0 International (CC BY 4.0) |

# C References for Code

**3D Earth Example (`earth_example.m`) [4]:**

- Use of `ellipsoid` function to render the Earth.

**Earth-sized Sphere with Topography (`earth_sphere`) [2]:**

- Handling of unit conversions.

# References


[1] *Attribution 4.0 International (CC BY 4.0)*. creative commons. https://creativecommons.org/licenses/by/4.0/. (accessed: January 27, 2021).

[2] Will Campbell. *Earth-sized Sphere with Topography*. MATLAB CENTRAL FILE EXCHANGE. https://www.mathworks.com/matlabcentral/fileexchange/27123-earth-sized-sphere-with-topography. (accessed: January 22, 2021).

[3] *Department of Defense World Geodetic System 1984*. Tech. rep. NIMA TR8350.2. https://apps.dtic.mil/sti/pdfs/AD1000581.pdf. National Imagery and Mapping Agency, 2004.

[4] Ryan Gray. *3D Earth Example*. MATLAB CENTRAL FILE EXCHANGE. https://www.mathworks.com/matlabcentral/fileexchange/13823-3d-earth-example. (accessed: January 22, 2021).

[5] *Image Use Policy*. NASA visible earth. https://visibleearth.nasa.gov/image-use-policy. (accessed: January 23, 2021).

[6] *Mercury Fact Sheet*. NASA. https://nssdc.gsfc.nasa.gov/planetary/factsheet/mercuryfact.html. (accessed: January 22, 2021).

[7] *Moon*. Wikipedia. https://en.wikipedia.org/wiki/Moon. (accessed: January 22, 2021).

[8] *Pluto Color Map*. NASA Jet Propulsion Laboratory. https://www.jpl.nasa.gov/images/pluto-color-map/. (accessed: January 23, 2021).

[9] *Rings of Saturn*. Wikipedia. https://en.wikipedia.org/wiki/Rings_of_Saturn. (accessed: January 22, 2021).

[10] *Solar Textures*. Solar System Scope. https://www.solarsystemscope.com/textures/. (accessed: January 22, 2021).

[11] *Sun*. Wikipedia. https://en.wikipedia.org/wiki/Sun. (accessed: January 22, 2021).

[12] *The Blue Marble: Land Surface, Ocean Color and Sea Ice*. NASA visible earth. https://visibleearth.nasa.gov/images/57730/the-blue-marble-land-surface-ocean-color-and-sea-ice/57731l. (accessed: January 22, 2021).

[13] David A. Vallado. *Fundamentals of Astrodynamics and Applications*. 4th. Hawthorne, CA: Microcosm Press, 2013.

[14] *Venus Fact Sheet*. NASA. https://nssdc.gsfc.nasa.gov/planetary/factsheet/venusfact.html. (accessed: January 22, 2021).