**Meeus Solar Position Calculations**

This notebook explores some of the calculations that underpin the [NOAA Solar Calculator](https://observablehq.com/@danleesmith/noaa-solar-calculations) notebook. To the best of my knowledge the NOAA calculations are generally based on the equations detailed in the book [*Astronomical Algorithms*](https://www.willbell.com/math/mc1.htm) by [Jean Meeus](https://en.wikipedia.org/wiki/Jean_Meeus). They are presented here as a way for me to familiarise myself with how they are implemented.

A helpful artcile that describes many of these calculations is presented by James Still on his blog post ["Astronomical Calculations: Solar Coordinates"](https://squarewidget.com/solar-coordinates/).

*Please note: this notebook is incomplete and there are several errors in the calculations that I am slowly working through.*

**Inputs**

latitude = -37.819982

longitude = 144.983431

Øverst i skjemaet

Date

 2020-06-21

Nederst i skjemaet

Øverst i skjemaet

Time

Nederst i skjemaet

Øverst i skjemaet

UTC Offset



Nederst i skjemaet

**Datetime**

The mean sidreal time at Greenwich or Universal time

datetime = Moment {\_isAMomentObject: true, \_i: "2020-06-21T10:00:00+10:00", \_f: "YYYY-MM-DDTHH:mm:ssZ", \_tzm: 600, \_isUTC: true, \_pf: Object, \_locale: Locale, \_d: 2020-06-21T10:00Z, \_isValid: true, \_offset: 600}

EPOCH\_DAY = Moment {\_isAMomentObject: true, \_i: "2020-06-21T10:00:00+10:00", \_f: "YYYY-MM-DDTHH:mm:ssZ", \_tzm: 600, \_isUTC: true, \_pf: Object, \_locale: Locale, \_d: 2020-06-21, \_isValid: true, \_offset: 600}

**Hail, Ceasar!**

**Julian Day**

The [Julian Day](https://en.wikipedia.org/wiki/Julian_day) is a unit of time used in astronomy that counts the number of days, and fractions thereof, that have elapsed since 12:00:00 [UTC](https://en.wikipedia.org/wiki/Coordinated_Universal_Time) on the 1st of January, 4713 BCE. This date marks the start of the Julian Period which is a 7980 year cycle, or epoch, that was invented in 1583 CE by [Joseph Justus Scaliger](https://en.wikipedia.org/wiki/Joseph_Justus_Scaliger) in order make it easy to reconcile the difference between the [Julian](https://en.wikipedia.org/wiki/Julian_calendar) and [Gregorian](https://en.wikipedia.org/wiki/Gregorian_calendar) calendars.

As an example: The Julian Day equivalent for 12:00:00 PM on the 21st of June, 2020 CE is **2,459,022.00** which is the count of the number of days that have elapsed since the 1st of January, 4713 BCE being the start of the Julian Period epoch.

Before we can develop a framework to calculate the Julian Day from a given timestamp we need to undestand another time system: [Unix time](https://en.wikipedia.org/wiki/Unix_time). Unix time is a measure of the number of milliseconds that have elapsed since 00:00:00 UTC on the 1st of January 1970 CE and importantly is what is returned from the [Date.getTime()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date/getTime) function that we will employ here.

So to get the Julian Day from this Unix timestamp we need to convert the returned milliseconds into a fractional count of the number of days that have elapsed since the 1st of January, 1970 CE until that timestamp and then add to this the fractional count of number of days between the start of the Julian Period epoch and the Unix epoch.

We know there are **86,400,000** milliseconds in a day and between the start of the two epochs **2,440,587.5** days have passed. Therefore the equation to calculate the Julian Day for a given Unix timestamp becomes:

JD = 2459021.0833333335

A great resource that helped me understand how to calculate the Julian Day from UNIX time came from [this thead](https://stackoverflow.com/questions/11759992/calculating-jdayjulian-day-in-javascript) on stackoverflow.

**Julian Century**

For the purposes of our calculations the Julian Day is just a means to an end. What we will actually be using is another member of the Julio time keeping dynasty: the Julian Century.

The Julian Century is comprised of **36,525** days of 86,000 [SI](https://en.wikipedia.org/wiki/International_System_of_Units) seconds (or 86,4000,000 milliseconds) each. The Julian Centrury should not be confused with the Julian Year which is comprised of 365.25 days of 86,000 SI seconds or even the Julian Day itself which as we saw earlier is about uniquley specifing a place in time without, in the words of Wikipedia, ["becoming bogged down in date-in-month, week or year in any particular calendar"](https://en.wikipedia.org/wiki/Julian_year_%28astronomy%29).

In astronomy circles the Julian Century has another epoch we need to work with known as J2000 or *Julian Epoch J2000.0*. J2000 is shorthand for the epoch that starts at [12 noon-ish](https://en.wikipedia.org/wiki/Epoch_%28astronomy%29#Julian_Dates_and_J2000) on the 1st of January, 2000 CE in the Gregorian Calendar or Julian Day equivalent **2,451,545.0**.

With this information in hand we can now calculate the Julian Century equivalent of our Julian Day within the J2000 epoch as follows:

T = 0.20468400638832276

Reading Meeus, we need to ensure that the quantity for *T* is calculated with a sufficicent number of decimal places. This is because *T* is expressed in centuries. As a result, an error of 0.00001 in *T* corresponds to an error of 0.37 days in time.

**Solar Coordinates**

The following calculations are predominantly outlined in Chapter 25. Meeus considers them to be "low accuracy" see:

*"When an accuracy of 0.01 degree is sufficient, the geocentric position of the sun may be calculated by assuming a purely elliptical motion of the Earth; that is, the perturbations by the Moon and the planets may be neglected"*

This low accuracy calculaiton is what is presented here.

Geometric mean longitude of the Sun (pg.163 : 25.2)

L = 7649.248274570985

Mean anomaly of the Sun (pg.163 : 25.3)

M = 7725.958949538352

Eccentricity of the the Earth's orbit (pg.163 : 25.4)

e = 0.016700024390258222

The Sun's equation of center (pg.164)

C = 0.45506781341604896

The Sun's true longitude (pg. 164)

sol = 7649.703342384401

The Sun's true anomaly (pg.164)

v = 7726.414017351768

The Sun's radius vector, or the distance between the centres of the Sun and the Earth expressed in astronomical units (pg.164)

R = 1.0162181240094386

Apparent longitude of the sun, also known as the 'true equinox of the date' (p.164)

λ = 7649.692872906328

Mean obliquity of the ecliptic (the Earth's axial tilt) in decimal degrees with correction (pg.147 : 22.2)

ε = 23.436667193453506

The Sun's apparent right ascention in equitorial coordinates (pg.165 : 25.6 & 25.8)

α = 89.66525727747052

The Sun's apparent declination in equitorial coordinates (pg.165 : 25.7)

δ = 23.436310356549576

**Nutation**

Longitude of the ascending node of the Moon's mean orbit on the ecliptic, measured form the mean equinox of the date (pg. 144)

Ω = -270.8421520260651

Simplified nutation in longitude in arcseconds (pg. 144)

Δψ = -17.107524434687573

**Time**

**There is an error with the following calculations! To the best of my knowledge it is caused by an error in the way the Julian Day is being calculated: need to remedy**

Sidereal Time at Greenwich in degrees (pg.88 : 12.4). The mean sidereal time is the Greenwich hour angle of the mean vernal point (the intersection of the ecliptic of the date with the mean equator of the date). Accounting for the nutation in right ascention we can transform from the mean sidreal time to the apparent sidreal time at Greenwich

θ = 7470.288892820998

The local hour angle measured westwards from South (pg. 92)

H = 35.64020454352794

**Transformation of Coordinates**

Calculaiton of local horizontal coordinates (pg.93 : 13.5, 13.6)

az = 34.72406957148576

alt = 20.19014171493422

**Trig Functions**

sin = ƒ(x)

cos = ƒ(x)

tan = ƒ(x)

deg = ƒ(r)

rad = ƒ(d)

**Requirements**

import {[coordinates](https://observablehq.com/@jashkenas/inputs#coordinates), [date](https://observablehq.com/@jashkenas/inputs#date), [time](https://observablehq.com/@jashkenas/inputs#time), [select](https://observablehq.com/@jashkenas/inputs#select)} from ["@jashkenas/inputs"](https://observablehq.com/@jashkenas/inputs)

moment = ƒ()