RA_DEC

February 29, 2020

[1]:
```
import numpy as np
import pandas as pd
import skyfield
from skyfield.api import Loader
skyfield_load = Loader('../data/skyfield')
from skyfield.positionlib import ICRF, Barycentric
from skyfield.toposlib import Topos
import astropy
from astropy.units import deg, au, km, meter, day, minute, second
from astropy.coordinates import SkyCoord, ICRS, GCRS, BarycentricMeanEcliptic,␣
↪HeliocentricMeanEcliptic, EarthLocation
from scipy.interpolate import CubicSpline
import os
import matplotlib.pyplot as plt

# MSE imports
import astro_utils
from astro_utils import jd_to_mjd, mjd_to_jd
from ra_dec import radec2dir, dir2radec, site2geoloc, qv2dir, direction_diff,␣
↪radec_diff, skyfield_observe
from horizons_files import load_pos_jpl, load_ast_jpl, load_obs_jpl,␣
↪load_obs_ast_jpl
from horizons_files import obs_add_interp_qv, obs_ast_add_interp_qv,␣
↪obs_add_calc_dir, obs_add_radec, obs_direction_diff
import asteroid_integrate
from asteroid_data import make_data_one_file, get_earth_pos
from utils import range_inc
```

0.0.1 Observation of Earth and Mars according to JPL

[2]:
```
# Data directories
dir_name = '../data/jpl/testing/hourly'
dir_name_daily = '../data/jpl/testing/daily'

# Build DataFrame for earth and mars position at 3 hour frequency
df_earth = load_pos_jpl(body_name='earth', dir_name=dir_name)
df_mars = load_pos_jpl(body_name='mars', dir_name=dir_name)
```

```
# Earth at daily frequency
df_earth_daily = load_pos_jpl(body_name='earth', dir_name=dir_name_daily)
# df_mars_daily = load_pos_jpl(body_name='mars', dir_name=dir_name_daily)
```

[3]:
```
# Display the earth dataframe
df_earth
```

[3]:
```
             mjd   JulianDate  time_key         X         Y         Z  \
0      55197.000  2455197.500   1324728 -0.179765  0.970347 -0.000017
1      55197.125  2455197.625   1324731 -0.181915  0.969951 -0.000017
2      55197.250  2455197.750   1324734 -0.184064  0.969551 -0.000017
3      55197.375  2455197.875   1324737 -0.186212  0.969145 -0.000017
4      55197.500  2455198.000   1324740 -0.188359  0.968736 -0.000017
…            …            …         …         …         …         …
29212  58848.500  2458849.000   1412364 -0.161514  0.978014 -0.000019
29213  58848.625  2458849.125   1412367 -0.163673  0.977658 -0.000018
29214  58848.750  2458849.250   1412370 -0.165831  0.977297 -0.000018
29215  58848.875  2458849.375   1412373 -0.167988  0.976931 -0.000018
29216  58849.000  2458849.500   1412376 -0.170144  0.976560 -0.000018

             VX        VY            VZ        LT        RG        RR
0     -0.017202 -0.003148  8.961125e-07  0.005700  0.986858  0.000038
1     -0.017195 -0.003186  8.995828e-07  0.005700  0.986863  0.000039
2     -0.017188 -0.003223  9.023349e-07  0.005700  0.986868  0.000039
3     -0.017181 -0.003260  9.043645e-07  0.005700  0.986873  0.000040
4     -0.017174 -0.003298  9.056684e-07  0.005700  0.986878  0.000041
…           …         …             …         …         …         …
29212 -0.017273 -0.002832  7.120095e-07  0.005725  0.991261  0.000020
29213 -0.017267 -0.002870  6.982338e-07  0.005725  0.991264  0.000020
29214 -0.017261 -0.002908  6.842707e-07  0.005725  0.991266  0.000021
29215 -0.017254 -0.002946  6.701300e-07  0.005725  0.991269  0.000021
29216 -0.017248 -0.002984  6.558216e-07  0.005725  0.991271  0.000021

[29217 rows x 12 columns]
```

[4]:
```
# Display the mars dataframe
# df_mars
```

[5]:
```
# Load the JPL observations of Mars
df_obs_mars_geo = load_obs_jpl(body_name='mars', observer_name='geocenter',␣
 ↪dir_name=dir_name)
df_obs_mars_pal = load_obs_jpl(body_name='mars', observer_name='palomar',␣
 ↪dir_name=dir_name)

# Display the dataframe
df_obs_mars_geo
```

[5]:

```
               mjd   JulianDate  time_key      RA_jpl    DEC_jpl    ux_jpl  \
0      55197.000  2455197.500   1324728  142.327061  18.799029 -0.749289
1      55197.125  2455197.625   1324731  142.309336  18.809991 -0.749061
2      55197.250  2455197.750   1324734  142.291395  18.821012 -0.748831
3      55197.375  2455197.875   1324737  142.273237  18.832093 -0.748598
4      55197.500  2455198.000   1324740  142.254862  18.843232 -0.748362
…            …            …       …           …          …         …
29212  58848.500  2458849.000   1412364  235.600679 -19.305260 -0.533190
29213  58848.625  2458849.125   1412367  235.687152 -19.325324 -0.531949
29214  58848.750  2458849.250   1412370  235.773651 -19.345349 -0.530706
29215  58848.875  2458849.375   1412373  235.860176 -19.365335 -0.529463
29216  58849.000  2458849.500   1412376  235.946727 -19.385281 -0.528218

         uy_jpl    uz_jpl  RA_apparent  DEC_apparent     delta  delta_dot  \
0      0.658994  0.065524   142.475968     18.752304  0.738832  -8.970408
1      0.659244  0.065613   142.458277     18.763272  0.738185  -8.940011
2      0.659497  0.065702   142.440369     18.774299  0.737541  -8.909502
3      0.659752  0.065791   142.422245     18.785386  0.736899  -8.878881
4      0.660010  0.065879   142.403904     18.796533  0.736259  -8.848146
…           …         …          …            …         …          …
29212 -0.845971  0.006438   235.880046    -19.365556  2.188001 -12.410093
29213 -0.846752  0.006417   235.966592    -19.385488  2.187105 -12.417570
29214 -0.847532  0.006395   236.053165    -19.405380  2.186208 -12.425025
29215 -0.848309  0.006373   236.139763    -19.425233  2.185311 -12.432457
29216 -0.849085  0.006352   236.226387    -19.445046  2.184413 -12.439868

       light_time
0        6.144676
1        6.139300
2        6.133941
3        6.128601
4        6.123279
…             …
29212   18.197048
29213   18.189595
29214   18.182137
29215   18.174675
29216   18.167208

[29217 rows x 13 columns]
```

[6]:

```python
# Extract position and velocity of earth from df_earth
q_earth_jpl = np.array([df_earth.X.values, df_earth.Y.values, df_earth.Z.
 ↪values]) * au
v_earth_jpl = np.array([df_earth.VX.values, df_earth.VY.values, df_earth.VZ.
 ↪values]) * au / day
```

```
# Extract position of mars from df_mars
q_mars_jpl = np.array([df_mars.X.values, df_mars.Y.values, df_mars.Z.values]) *␣
↪au
v_mars_jpl = np.array([df_mars.VX.values, df_mars.VY.values, df_mars.VZ.
↪values]) * au / day

# Extract obstime_jd, ra, and dec from DataFrame with geocentric observations
obstime_mars_geo_jd = df_obs_mars_geo.JulianDate.values
# ra_mars_geo_jpl = df_obs_mars_geo.RA.values * deg
# dec_mars_geo_jpl = df_obs_mars_geo.DEC.values * deg

# Observation times for palomar observations
obstime_mars_pal_jd = df_obs_mars_pal.JulianDate.values

# Vector of observation times in MJD format
obstime_mars_geo_mjd = jd_to_mjd(obstime_mars_geo_jd)
obstime_mars_pal_mjd = jd_to_mjd(obstime_mars_pal_jd)

# Alias to obstime_mars_mjd because they are the same
obstime_mars_jd = obstime_mars_geo_jd
obstime_mars_mjd = obstime_mars_geo_mjd
```

0.0.2 Position & Observation of Earth and Mars according to Skyfield

[7]:
```
# Manually load planetary positions using de435
# planets_sf = skyfield_load('../data/jpl/ephemeris/de435.bsp')
planets_sf = skyfield_load('de435.bsp')
earth_sf = planets_sf['earth']
mars_sf = planets_sf['mars barycenter']

# load timescale
ts_sf = skyfield_load.timescale()

# Generate vector of observation times in Skyfield format
obstime_mars_sf = ts_sf.tt_jd(obstime_mars_jd)
```

[8]:
```
# Observe mars from earth geocenter with Skyfield
ra_mars_geo_sf, dec_mars_geo_sf, delta_mars_geo_sf = \
    skyfield_observe(observer_sf=earth_sf, body_sf=mars_sf,␣
↪obstime_sf=obstime_mars_sf)
```

[9]:
```
# location of palomar as a Skyfield topos object
geoloc_pal = site2geoloc('palomar', verbose=True)
lon, lat, height = geoloc_pal.geodetic
palomar_topos = skyfield.toposlib.Topos(latitude_degrees=lat.value,␣
↪longitude_degrees=lon.value, elevation_m=height.value)
```

```
palomar_topos
```

```
Geolocation of palomar:
cartesian = (-2410346.78217658, -4758666.82504051, 3487942.97502457) m
geodetic  = GeodeticLocation(lon=<Longitude -116.863 deg>, lat=<Latitude 33.356
deg>, height=<Quantity 1706. m>)
```

[9]:
```
<Topos 33deg 21' 21.6" N -116deg 51' 46.8" E>
```

[10]:
```
# Observe mars from palomar with Skyfield
palomar_sf = earth_sf + palomar_topos
ra_mars_pal_sf, dec_mars_pal_sf, delta_mars_pal_sf = \
    skyfield_observe(observer_sf=palomar_sf, body_sf=mars_sf,␣
 ↪obstime_sf=obstime_mars_sf)
```

[11]:
```
# Load planetary positions and velocities by querying the Skyfield JPL␣
 ↪ephemeris interface
# Create them as arrays with bundled astropy units of au and km / second

# Earth
q_earth_sf = earth_sf.at(obstime_mars_sf).ecliptic_position().au * au
v_earth_sf = earth_sf.at(obstime_mars_sf).ecliptic_velocity().km_per_s * km /␣
 ↪second

# Palomar
q_palomar_sf = palomar_sf.at(obstime_mars_sf).ecliptic_position().au * au
v_palomar_sf = palomar_sf.at(obstime_mars_sf).ecliptic_velocity().km_per_s * km␣
 ↪/ second

# Mars
q_mars_sf = mars_sf.at(obstime_mars_sf).ecliptic_position().au * au
v_mars_sf = mars_sf.at(obstime_mars_sf).ecliptic_velocity().km_per_s * km /␣
 ↪second
```

[12]:
```
# Demonstrate that q_earth_sf is the same as q_earth_jpl
q_earth_eps = np.mean(np.linalg.norm(q_earth_sf - q_earth_jpl, axis=0))
v_earth_eps = np.mean(np.linalg.norm(v_earth_sf - v_earth_jpl, axis=0))
q_mars_eps = np.mean(np.linalg.norm(q_mars_sf - q_mars_jpl, axis=0))
v_mars_eps = np.mean(np.linalg.norm(v_mars_sf - v_mars_jpl, axis=0))

# Report
print('Difference between Skyfield (JPL ephem) and Horizons download:')
print(f'q_earth : {q_earth_eps:5.3e} au')
print(f'v_earth : {v_earth_eps:5.3e} au / day')
print(f'q_mars  : {q_mars_eps:5.3e} au')
print(f'v_mars  : {v_mars_eps:5.3e} au / day')
```

```
Difference between Skyfield (JPL ephem) and Horizons download:
q_earth : 1.406e-09 AU au
v_earth : 1.626e-08 km / s au / day
q_mars  : 1.565e-09 AU au
v_mars  : 3.758e-08 km / s au / day
```

Conclusion Skyfield is essentially identical to JPL in coordinates of Earth and Mars. Difference is on the order of 10^{-9} AU.

0.0.3 Compare Skyfield vs JPL on RA/DEC of Mars

[13]:
```
# Convert SkyField RA/DEC to directions
# u_mars_geo_sf = radec2dir(ra=ra_mars_geo_sf, dec=dec_mars_geo_sf,␣
↪obstime_mjd=obstime_mars_mjd)
# u_mars_pal_sf = radec2dir(ra=ra_mars_pal_sf, dec=dec_mars_pal_sf,␣
↪obstime_mjd=obstime_mars_mjd)
```

[14]:
```
# Add RA/DEC and direction from Skyfield to mars observation frames
obs_add_radec(df_obs=df_obs_mars_geo, ra=ra_mars_geo_sf, dec=dec_mars_geo_sf,␣
↪source='sf')
obs_add_radec(df_obs=df_obs_mars_pal, ra=ra_mars_pal_sf, dec=dec_mars_pal_sf,␣
↪source='sf')
```

[15]:
```
# Report difference between JPL and Skyfield from geocenter
print(f'Comparing direction of Mars from Geocenter: Skyfield vs. JPL')
print(f'(1) Direction according to JPL: radec2dir applied to JPL RA/DEC')
print(f'(2) Direction according to Skyfield: radec2dir applied to Skyfield RA/
↪DEC (from observe)\n')
diff_geo_jpl_sf = obs_direction_diff(df_obs=df_obs_mars_geo, src1='jpl',␣
↪src2='sf', verbose=True)
```

```
Comparing direction of Mars from Geocenter: Skyfield vs. JPL
(1) Direction according to JPL: radec2dir applied to JPL RA/DEC
(2) Direction according to Skyfield: radec2dir applied to Skyfield RA/DEC (from
observe)

Angle Difference: sf vs. jpl
Mean  :   0.000444 deg (   1.598 seconds)
Median:   0.000506 deg (   1.822 seconds)
Max   :   0.000615 deg (   2.213 seconds)
```

[16]:
```
# Report difference between JPL and Skyfield from palomar
print(f'Comparing direction of Mars from Palomar: Skyfield vs. JPL')
print(f'(1) Direction according to JPL: radec2dir applied to JPL RA/DEC')
print(f'(2) Direction according to Skyfield: radec2dir applied to Skyfield RA/
↪DEC (from observe)\n')
diff_geo_pal_sf = obs_direction_diff(df_obs=df_obs_mars_pal, src1='jpl',␣
↪src2='sf', verbose=True)
```

```
Comparing direction of Mars from Palomar: Skyfield vs. JPL
(1) Direction according to JPL: radec2dir applied to JPL RA/DEC
(2) Direction according to Skyfield: radec2dir applied to Skyfield RA/DEC (from
observe)

Angle Difference: sf vs. jpl
Mean  :   0.000444 deg (   1.599 seconds)
Median:   0.000506 deg (   1.820 seconds)
Max   :   0.000619 deg (   2.228 seconds)
```

[17]:

```python
# Review observation frame with additional columns
df_obs_mars_geo
```

[17]:

```
             mjd   JulianDate  time_key      RA_jpl    DEC_jpl    ux_jpl  \
0      55197.000  2455197.500   1324728  142.327061  18.799029 -0.749289
1      55197.125  2455197.625   1324731  142.309336  18.809991 -0.749061
2      55197.250  2455197.750   1324734  142.291395  18.821012 -0.748831
3      55197.375  2455197.875   1324737  142.273237  18.832093 -0.748598
4      55197.500  2455198.000   1324740  142.254862  18.843232 -0.748362
…            …            …        …           …          …         …
29212  58848.500  2458849.000   1412364  235.600679 -19.305260 -0.533190
29213  58848.625  2458849.125   1412367  235.687152 -19.325324 -0.531949
29214  58848.750  2458849.250   1412370  235.773651 -19.345349 -0.530706
29215  58848.875  2458849.375   1412373  235.860176 -19.365335 -0.529463
29216  58849.000  2458849.500   1412376  235.946727 -19.385281 -0.528218

         uy_jpl    uz_jpl  RA_apparent  DEC_apparent     delta  delta_dot  \
0      0.658994  0.065524   142.475968     18.752304  0.738832  -8.970408
1      0.659244  0.065613   142.458277     18.763272  0.738185  -8.940011
2      0.659497  0.065702   142.440369     18.774299  0.737541  -8.909502
3      0.659752  0.065791   142.422245     18.785386  0.736899  -8.878881
4      0.660010  0.065879   142.403904     18.796533  0.736259  -8.848146
…             …         …            …             …         …          …
29212 -0.845971  0.006438   235.880046    -19.365556  2.188001 -12.410093
29213 -0.846752  0.006417   235.966592    -19.385488  2.187105 -12.417570
29214 -0.847532  0.006395   236.053165    -19.405380  2.186208 -12.425025
29215 -0.848309  0.006373   236.139763    -19.425233  2.185311 -12.432457
29216 -0.849085  0.006352   236.226387    -19.445046  2.184413 -12.439868

       light_time       RA_sf     DEC_sf     ux_sf     uy_sf     uz_sf
0        6.144676  142.327169  18.798962 -0.749290  0.658992  0.065524
1        6.139300  142.309446  18.809924 -0.749062  0.659242  0.065612
2        6.133941  142.291505  18.820944 -0.748832  0.659495  0.065701
3        6.128601  142.273349  18.832025 -0.748599  0.659751  0.065790
4        6.123279  142.254975  18.843164 -0.748364  0.660009  0.065879
…               …           …          …         …         …         …
29212   18.197048  235.600125 -19.305132 -0.533198 -0.845966  0.006438
```

```
29213   18.189595  235.686598 -19.325196 -0.531957 -0.846747  0.006417
29214   18.182137  235.773097 -19.345221 -0.530714 -0.847527  0.006395
29215   18.174675  235.859622 -19.365207 -0.529471 -0.848304  0.006374
29216   18.167208  235.946173 -19.385153 -0.528226 -0.849080  0.006352

[29217 rows x 18 columns]
```

Conclusion Skyfield is very close to JPL on observation of Mars. Mean difference is **1.60 arc seconds** from both Geocenter and Palomar. The difference between Geocenter and Palomar is about 3.78 arc seconds on average for Mars. Results appear to be consistent between Geocenter and Palomar.

0.0.4 Estimate Importance of Including Observatory Location

[18]:
```
# Difference between palomar and geocenter according to JPL
radec_diff(name1='geocenter-JPL', name2='palomar-JPL',
           ra1=df_obs_mars_geo.RA_jpl.values*deg, dec1=df_obs_mars_geo.DEC_jpl.
 ↪values*deg,
           ra2=df_obs_mars_pal.RA_jpl.values*deg, dec2=df_obs_mars_pal.DEC_jpl.
 ↪values*deg,
           obstime_mjd=df_obs_mars_geo.mjd.values, verbose=True)
```

```
Angle Difference: palomar-JPL vs. geocenter-JPL
Mean  :   0.001409 deg (   5.073 seconds)
Median:   0.001050 deg (   3.780 seconds)
Max   :   0.006340 deg (  22.824 seconds)
```

[18]:
```
5.073474626406522
```

[19]:
```
# Difference between palomar and geocenter according to Skyfield
radec_diff(name1='geocenter-Skyfield', name2='palomar-Skyfield',
           ra1=df_obs_mars_geo.RA_sf.values*deg, dec1=df_obs_mars_geo.DEC_sf.
 ↪values*deg,
           ra2=df_obs_mars_pal.RA_sf.values*deg, dec2=df_obs_mars_pal.DEC_sf.
 ↪values*deg,
           obstime_mjd=df_obs_mars_geo.mjd.values, verbose=True)
```

```
Angle Difference: palomar-Skyfield vs. geocenter-Skyfield
Mean  :   0.001409 deg (   5.074 seconds)
Median:   0.001050 deg (   3.780 seconds)
Max   :   0.006340 deg (  22.825 seconds)
```

[19]:
```
5.07352264809586
```

Conclusion Ignoring the observatory location would introduce an error of about **3.8 arc seconds** for Mars. This effect is important enough that we should certainly try to model it.

0.0.5 Calculate Direction from Earth to Mars with qv2dir() and JPL Position / Velocity

[20]:
```
# Add interpolated JPL Positions to observation DataFrames
obs_add_interp_qv(df_obs=df_obs_mars_geo, df_body=df_mars, df_earth=df_earth,␣
 ↪source_name='jpl')
obs_add_interp_qv(df_obs=df_obs_mars_pal, df_body=df_mars, df_earth=df_earth,␣
 ↪source_name='jpl')
```

[21]:
```
# Display augmented df_obs_mars
df_obs_mars_geo
```

[21]:

	mjd	JulianDate	time_key	RA_jpl	DEC_jpl	ux_jpl
0	55197.000	2455197.500	1324728	142.327061	18.799029	-0.749289
1	55197.125	2455197.625	1324731	142.309336	18.809991	-0.749061
2	55197.250	2455197.750	1324734	142.291395	18.821012	-0.748831
3	55197.375	2455197.875	1324737	142.273237	18.832093	-0.748598
4	55197.500	2455198.000	1324740	142.254862	18.843232	-0.748362
…	…	…	…	…	…	…
29212	58848.500	2458849.000	1412364	235.600679	-19.305260	-0.533190
29213	58848.625	2458849.125	1412367	235.687152	-19.325324	-0.531949
29214	58848.750	2458849.250	1412370	235.773651	-19.345349	-0.530706
29215	58848.875	2458849.375	1412373	235.860176	-19.365335	-0.529463
29216	58849.000	2458849.500	1412376	235.946727	-19.385281	-0.528218

	uy_jpl	uz_jpl	RA_apparent	DEC_apparent	…	uz_sf
0	0.658994	0.065524	142.475968	18.752304	…	0.065524
1	0.659244	0.065613	142.458277	18.763272	…	0.065612
2	0.659497	0.065702	142.440369	18.774299	…	0.065701
3	0.659752	0.065791	142.422245	18.785386	…	0.065790
4	0.660010	0.065879	142.403904	18.796533	…	0.065879
…	…	…	…	…	…	…
29212	-0.845971	0.006438	235.880046	-19.365556	…	0.006438
29213	-0.846752	0.006417	235.966592	-19.385488	…	0.006417
29214	-0.847532	0.006395	236.053165	-19.405380	…	0.006395
29215	-0.848309	0.006373	236.139763	-19.425233	…	0.006374
29216	-0.849085	0.006352	236.226387	-19.445046	…	0.006352

	body_x_jpl	body_y_jpl	body_z_jpl	body_vx_jpl	body_vy_jpl
0	-0.733418	1.457212	0.048394	-0.011980	-0.005093
1	-0.734916	1.456575	0.048418	-0.011974	-0.005105
2	-0.736412	1.455936	0.048441	-0.011968	-0.005117
3	-0.737908	1.455296	0.048464	-0.011961	-0.005130
4	-0.739402	1.454654	0.048488	-0.011955	-0.005142
…	…	…	…	…	…
29212	-1.328050	-0.873097	0.014063	0.008264	-0.010457
29213	-1.327017	-0.874404	0.014010	0.008276	-0.010449

```
29214   -1.325981   -0.875709    0.013958    0.008288   -0.010441
29215   -1.324945   -0.877014    0.013905    0.008300   -0.010433
29216   -1.323906   -0.878318    0.013852    0.008313   -0.010425

       body_vz_jpl  earth_x_jpl  earth_y_jpl  earth_z_jpl
0         0.000188    -0.179765     0.970347    -0.000017
1         0.000187    -0.181915     0.969951    -0.000017
2         0.000187    -0.184064     0.969551    -0.000017
3         0.000186    -0.186212     0.969145    -0.000017
4         0.000186    -0.188359     0.968736    -0.000017
…              …           …           …           …
29212    -0.000422    -0.161514     0.978014    -0.000019
29213    -0.000422    -0.163673     0.977658    -0.000018
29214    -0.000422    -0.165831     0.977297    -0.000018
29215    -0.000422    -0.167988     0.976931    -0.000018
29216    -0.000422    -0.170144     0.976560    -0.000018

[29217 rows x 27 columns]
```

[22]:

```
# Build geolocation of theoretical observer at geocenter
geoloc_geo = site2geoloc(site_name='geocenter', verbose=False)

# Build geolocation of observer at Palomar
geoloc_pal = site2geoloc(site_name='palomar', verbose=True)
```

```
Geolocation of palomar:
cartesian = (-2410346.78217658, -4758666.82504051, 3487942.97502457) m
geodetic  = GeodeticLocation(lon=<Longitude -116.863 deg>, lat=<Latitude 33.356
deg>, height=<Quantity 1706. m>)
```

[23]:

```
df_obs = df_obs_mars_geo
src = 'jpl'
site_name='geocenter'

# Columns for position and velocity from this source
q_body_cols = [f'body_x_{src}', f'body_y_{src}', f'body_z_{src}']
v_body_cols = [f'body_vx_{src}', f'body_vy_{src}', f'body_vz_{src}']
q_earth_cols = [f'earth_x_{src}', f'earth_y_{src}', f'earth_z_{src}']

# Extract position and velocity of space body; build as Nx3 array with astropy␣
 ↪units
q_body = df_obs[q_body_cols].values * au
v_body = df_obs[v_body_cols].values * au / day

# Extract position of earth; build as Nx3 array with astropy units
q_earth = df_obs[q_earth_cols].values * au
```

```python
# Observation times and geolocation of this site
obstime_mjd = df_obs.mjd.values
obsgeoloc = site2geoloc(site_name=site_name, verbose=False)
```

[24]:
```python
# Compute the directions qv2dir() accounting for observer location; save them␣
↪to the DataFrame of mars observations
obs_add_calc_dir(df_obs=df_obs_mars_geo, site_name='geocenter',␣
↪source_name='jpl')
obs_add_calc_dir(df_obs=df_obs_mars_pal, site_name='palomar', source_name='jpl')
```

[25]:
```python
# Review added columns
df_obs_mars_geo
```

[25]:

```
             mjd   JulianDate  time_key      RA_jpl    DEC_jpl    ux_jpl  \
0      55197.000  2455197.500   1324728  142.327061  18.799029 -0.749289
1      55197.125  2455197.625   1324731  142.309336  18.809991 -0.749061
2      55197.250  2455197.750   1324734  142.291395  18.821012 -0.748831
3      55197.375  2455197.875   1324737  142.273237  18.832093 -0.748598
4      55197.500  2455198.000   1324740  142.254862  18.843232 -0.748362
…            …            …        …           …          …        …
29212  58848.500  2458849.000   1412364  235.600679 -19.305260 -0.533190
29213  58848.625  2458849.125   1412367  235.687152 -19.325324 -0.531949
29214  58848.750  2458849.250   1412370  235.773651 -19.345349 -0.530706
29215  58848.875  2458849.375   1412373  235.860176 -19.365335 -0.529463
29216  58849.000  2458849.500   1412376  235.946727 -19.385281 -0.528218

         uy_jpl    uz_jpl  RA_apparent  DEC_apparent  …  body_vy_jpl  \
0      0.658994  0.065524   142.475968     18.752304  …    -0.005093
1      0.659244  0.065613   142.458277     18.763272  …    -0.005105
2      0.659497  0.065702   142.440369     18.774299  …    -0.005117
3      0.659752  0.065791   142.422245     18.785386  …    -0.005130
4      0.660010  0.065879   142.403904     18.796533  …    -0.005142
…           …         …            …           …  …  …          …
29212 -0.845971  0.006438   235.880046    -19.365556  …    -0.010457
29213 -0.846752  0.006417   235.966592    -19.385488  …    -0.010449
29214 -0.847532  0.006395   236.053165    -19.405380  …    -0.010441
29215 -0.848309  0.006373   236.139763    -19.425233  …    -0.010433
29216 -0.849085  0.006352   236.226387    -19.445046  …    -0.010425

       body_vz_jpl  earth_x_jpl  earth_y_jpl  earth_z_jpl  RA_calc_jpl  \
0         0.000188    -0.179765     0.970347    -0.000017   142.327163
1         0.000187    -0.181915     0.969951    -0.000017   142.309440
2         0.000187    -0.184064     0.969551    -0.000017   142.291500
3         0.000186    -0.186212     0.969145    -0.000017   142.273343
4         0.000186    -0.188359     0.968736    -0.000017   142.254969
…              …            …            …            …            …
29212    -0.000422    -0.161514     0.978014    -0.000019   235.600124
```

```
29213    -0.000422    -0.163673     0.977658    -0.000018   235.686597
29214    -0.000422    -0.165831     0.977297    -0.000018   235.773096
29215    -0.000422    -0.167988     0.976931    -0.000018   235.859621
29216    -0.000422    -0.170144     0.976560    -0.000018   235.946172

       DEC_calc_jpl  ux_calc_jpl  uy_calc_jpl  uz_calc_jpl
0         18.798953    -0.749290     0.658992     0.065523
1         18.809914    -0.749062     0.659242     0.065612
2         18.820935    -0.748832     0.659495     0.065701
3         18.832015    -0.748599     0.659751     0.065790
4         18.843154    -0.748364     0.660009     0.065879
…               …          …           …          …
29212    -19.305126    -0.533198    -0.845966     0.006438
29213    -19.325190    -0.531957    -0.846747     0.006417
29214    -19.345215    -0.530714    -0.847527     0.006395
29215    -19.365201    -0.529471    -0.848304     0.006374
29216    -19.385148    -0.528226    -0.849080     0.006352

[29217 rows x 32 columns]
```

[26]:
```python
# Review all the columns that are now on the observation DataFrame
df_obs_mars_geo.columns
```

[26]:
```
Index(['mjd', 'JulianDate', 'time_key', 'RA_jpl', 'DEC_jpl', 'ux_jpl',
       'uy_jpl', 'uz_jpl', 'RA_apparent', 'DEC_apparent', 'delta', 'delta_dot',
       'light_time', 'RA_sf', 'DEC_sf', 'ux_sf', 'uy_sf', 'uz_sf',
       'body_x_jpl', 'body_y_jpl', 'body_z_jpl', 'body_vx_jpl', 'body_vy_jpl',
       'body_vz_jpl', 'earth_x_jpl', 'earth_y_jpl', 'earth_z_jpl',
       'RA_calc_jpl', 'DEC_calc_jpl', 'ux_calc_jpl', 'uy_calc_jpl',
       'uz_calc_jpl'],
      dtype='object')
```

0.0.6 Direction from Geocenter to Mars: Compare JPL, Skyfield, and qv2dir(JPL position)

[27]:
```python
# Report difference for Mars from Geocenter between Skyfiled and MSE calculated
print(f'Comparing direction of Mars from Geocenter: Skyfield vs. MSE calc from␣
 ↪JPL positions:')
print(f'(1) Direction according to Skyfield: radec2dir applied to Skyfield RA/
 ↪DEC')
print(f'(2) Direction according to MSE: qv2dir applied to JPL positions &␣
 ↪velocities\n')
diff_geo_sf_calc_jpl = obs_direction_diff(df_obs=df_obs_mars_geo, src1='sf',␣
 ↪src2='calc_jpl', verbose=True)
```

```
Comparing direction of Mars from Geocenter: Skyfield vs. MSE calc from JPL
positions:
```

```
(1) Direction according to Skyfield: radec2dir applied to Skyfield RA/DEC
(2) Direction according to MSE: qv2dir applied to JPL positions & velocities

Angle Difference: calc_jpl vs. sf
Mean  :   0.000008 deg (   0.027 seconds)
Median:   0.000008 deg (   0.029 seconds)
Max   :   0.000012 deg (   0.042 seconds)
```

[28]:
```
# Report difference for Mars from Geocenter between JPL and MSE calculated
print(f'Comparing direction of Mars from Geocenter: JPL vs. MSE calc from JPL␣
 ↪positions')
print(f'(1) Direction according to JPL: radec2dir applied to JPL RA/DEC')
print(f'(2) Direction according to MSE: qv2dir applied to JPL positions &␣
 ↪velocities\n')
diff_geo_jpl_jpl_calc_jpl = obs_direction_diff(df_obs=df_obs_mars_geo,␣
 ↪src1='jpl', src2='calc_jpl', verbose=True)
```

```
Comparing direction of Mars from Geocenter: JPL vs. MSE calc from JPL positions
(1) Direction according to JPL: radec2dir applied to JPL RA/DEC
(2) Direction according to MSE: qv2dir applied to JPL positions & velocities

Angle Difference: calc_jpl vs. jpl
Mean  :   0.000446 deg (   1.604 seconds)
Median:   0.000508 deg (   1.830 seconds)
Max   :   0.000616 deg (   2.219 seconds)
```

Conclusion My calculations are almost identical to Skyfield; accurate to **0.027 arc seconds** Both Skyfield and I are off from JPL by **1.60 arc seconds**

0.0.7 Direction from Palomar to Mars: Compare JPL, Skyfield, and qv2dir(JPL position)

[29]:
```
# Calculate direction from palomar using JPL positions
obs_add_calc_dir(df_obs=df_obs_mars_pal, site_name='palomar', source_name='jpl')
```

[30]:
```
# Report difference for Mars from Palomar between Skyfield and MSE calculated
print(f'Comparing direction of Mars from Palomar: Skyfield vs. MSE calc from␣
 ↪JPL positions:')
print(f'(1) Direction according to Skyfield: radec2dir applied to Skyfield RA/
 ↪DEC')
print(f'(2) Direction according to MSE: qv2dir applied to JPL positions &␣
 ↪velocities\n')
diff_pal_sf_calc_jpl = obs_direction_diff(df_obs=df_obs_mars_pal, src1='sf',␣
 ↪src2='calc_jpl', verbose=True)
```

```
Comparing direction of Mars from Palomar: Skyfield vs. MSE calc from JPL
positions:
(1) Direction according to Skyfield: radec2dir applied to Skyfield RA/DEC
```

```
(2) Direction according to MSE: qv2dir applied to JPL positions & velocities

Angle Difference: calc_jpl vs. sf
Mean  :   0.000008 deg (   0.027 seconds)
Median:   0.000008 deg (   0.029 seconds)
Max   :   0.000012 deg (   0.042 seconds)
```

[31]:
```python
# Report difference for Mars from Geocenter between JPL and MSE calculated
print(f'Comparing direction of Mars from Geocenter: JPL vs. MSE calc from JPL␣
 ↪positions')
print(f'(1) Direction according to JPL: radec2dir applied to JPL RA/DEC')
print(f'(2) Direction according to MSE: qv2dir applied to JPL positions &␣
 ↪velocities\n')
diff_pal_jpl_calc_jpl = obs_direction_diff(df_obs=df_obs_mars_pal, src1='jpl',␣
 ↪src2='calc_jpl', verbose=True)
```

```
Comparing direction of Mars from Geocenter: JPL vs. MSE calc from JPL positions
(1) Direction according to JPL: radec2dir applied to JPL RA/DEC
(2) Direction according to MSE: qv2dir applied to JPL positions & velocities

Angle Difference: calc_jpl vs. jpl
Mean  :   0.000446 deg (   1.604 seconds)
Median:   0.000508 deg (   1.828 seconds)
Max   :   0.000621 deg (   2.234 seconds)
```

0.0.8 Vectors of First 16 Asteroids from JPL

[32]:
```python
# Load the asteroid position and velocity from JPL
df_ast = load_ast_jpl(ast_num0=1, ast_num1=16, dir_name=dir_name)
df_ast_daily = load_ast_jpl(ast_num0=1, ast_num1=16, dir_name=dir_name_daily)
```

[33]:
```python
df_ast
```

[33]:

	asteroid_num	mjd	JulianDate	time_key	X	Y \
0	1	55197.000	2455197.500	1324728	-1.660333	-2.123236
1	1	55197.125	2455197.625	1324731	-1.659381	-2.124130
2	1	55197.250	2455197.750	1324734	-1.658429	-2.125023
3	1	55197.375	2455197.875	1324737	-1.657476	-2.125915
4	1	55197.500	2455198.000	1324740	-1.656522	-2.126807
…	…	…	…	…	…	…
29212	16	58848.500	2458849.000	1412364	2.522975	-0.473790
29213	16	58848.625	2458849.125	1412367	2.523154	-0.472385
29214	16	58848.750	2458849.250	1412370	2.523332	-0.470980
29215	16	58848.875	2458849.375	1412373	2.523510	-0.469574
29216	16	58849.000	2458849.500	1412376	2.523687	-0.468169

Z	VX	VY	VZ	LT	RG	RR

```
0       0.238962  0.007615 -0.007150 -0.001627  0.015628  2.705909  0.000794
1       0.238759  0.007618 -0.007146 -0.001627  0.015629  2.706008  0.000794
2       0.238555  0.007622 -0.007142 -0.001628  0.015629  2.706107  0.000794
3       0.238352  0.007625 -0.007138 -0.001628  0.015630  2.706206  0.000794
4       0.238148  0.007628 -0.007134 -0.001628  0.015630  2.706306  0.000794
…            …         …         …         …         …         …         …
29212 -0.045683  0.001437  0.011240 -0.000566  0.014829  2.567482 -0.000652
29213 -0.045754  0.001431  0.011242 -0.000566  0.014828  2.567401 -0.000652
29214 -0.045825  0.001426  0.011243 -0.000566  0.014828  2.567319 -0.000651
29215 -0.045895  0.001420  0.011244 -0.000565  0.014827  2.567238 -0.000650
29216 -0.045966  0.001415  0.011245 -0.000565  0.014827  2.567157 -0.000650

[467472 rows x 13 columns]
```

0.0.9 Observations of First 16 Asteroids from JPL

[34]:
```python
# Load the asteroid observations from JPL
df_obs = load_obs_ast_jpl(ast_num0=1, ast_num1=16, observer_name='palomar',␣
 ↪dir_name=dir_name)
df_obs_daily = load_obs_ast_jpl(ast_num0=1, ast_num1=16,␣
 ↪observer_name='geocenter', dir_name=dir_name_daily)
```

[35]:
```python
df_obs
```

[35]:
```
       asteroid_num        mjd   JulianDate  time_key      RA_jpl    DEC_jpl  \
0                 1  55197.000  2455197.500   1324728  243.214830 -17.106252
1                 1  55197.125  2455197.625   1324731  243.266342 -17.117471
2                 1  55197.250  2455197.750   1324734  243.318051 -17.128756
3                 1  55197.375  2455197.875   1324737  243.369682 -17.140123
4                 1  55197.500  2455198.000   1324740  243.420999 -17.151530
…                 …          …            …         …           …          …
29212            16  58848.500  2458849.000   1412364  333.918987 -11.709509
29213            16  58848.625  2458849.125   1412367  333.967430 -11.692635
29214            16  58848.750  2458849.250   1412370  334.015569 -11.675789
29215            16  58848.875  2458849.375   1412373  334.063332 -11.658913
29216            16  58849.000  2458849.500   1412376  334.110877 -11.641955

         ux_jpl    uy_jpl    uz_jpl  RA_apparent  DEC_apparent     delta  \
0     -0.430710 -0.899809  0.069515   243.357955    -17.132205  3.437890
1     -0.429917 -0.900191  0.069477   243.409452    -17.143375  3.437011
2     -0.429121 -0.900574  0.069438   243.461181    -17.154603  3.436114
3     -0.428325 -0.900956  0.069397   243.512878    -17.165916  3.435195
4     -0.427533 -0.901335  0.069354   243.564287    -17.177279  3.434263
…            …         …         …            …             …         …
29212  0.879482 -0.475698 -0.014963   334.177362    -11.612170  3.052380
29213  0.879899 -0.474925 -0.014984   334.225801    -11.595248  3.053587
29214  0.880313 -0.474156 -0.015003   334.273964    -11.578362  3.054780
```

```
29215  0.880724 -0.473391 -0.015020   334.321743    -11.561454  3.055976
29216  0.881133 -0.472629 -0.015035   334.369269    -11.544470  3.057186

       delta_dot  light_time
0     -12.102732   28.592060
1     -12.282233   28.584749
2     -12.585595   28.577286
3     -12.849298   28.569644
4     -12.933136   28.561893
…            …          …
29212  16.859446   25.385873
29213  16.591743   25.395910
29214  16.504155   25.405834
29215  16.637661   25.415774
29216  16.903294   25.425841

[467472 rows x 14 columns]
```

0.0.10 Calculate Asteroid Direction with qv2dir() and JPL Position / Velocity

[36]:
```
# Add interpolated JPL position to asteroid observations
obs_ast_add_interp_qv(df_obs=df_obs, df_ast=df_ast, df_earth=df_earth,␣
 ↪source_name='jpl')
```

[37]:
```
# Add computed directions from the JPL positions
obs_add_calc_dir(df_obs=df_obs, site_name='palomar', source_name='jpl')
```

[38]:
```
# Review asteroid observations with additional columns: interpolated positions␣
 ↪& calculated RA/DEC, direction
df_obs
```

[38]:
```
       asteroid_num        mjd   JulianDate  time_key      RA_jpl    DEC_jpl  \
0                 1  55197.000  2455197.500   1324728  243.214830 -17.106252
1                 1  55197.125  2455197.625   1324731  243.266342 -17.117471
2                 1  55197.250  2455197.750   1324734  243.318051 -17.128756
3                 1  55197.375  2455197.875   1324737  243.369682 -17.140123
4                 1  55197.500  2455198.000   1324740  243.420999 -17.151530
…               …        …          …        …        …          …
29212            16  58848.500  2458849.000   1412364  333.918987 -11.709509
29213            16  58848.625  2458849.125   1412367  333.967430 -11.692635
29214            16  58848.750  2458849.250   1412370  334.015569 -11.675789
29215            16  58848.875  2458849.375   1412373  334.063332 -11.658913
29216            16  58849.000  2458849.500   1412376  334.110877 -11.641955

         ux_jpl    uy_jpl    uz_jpl  RA_apparent  …  body_vy_jpl  \
0     -0.430710 -0.899809  0.069515   243.357955  …    -0.007150
```

```
1      -0.429917 -0.900191  0.069477   243.409452  …    -0.007146
2      -0.429121 -0.900574  0.069438   243.461181  …    -0.007142
3      -0.428325 -0.900956  0.069397   243.512878  …    -0.007138
4      -0.427533 -0.901335  0.069354   243.564287  …    -0.007134
…          …        …        …          …   …          …
29212   0.879482 -0.475698 -0.014963   334.177362  …     0.011240
29213   0.879899 -0.474925 -0.014984   334.225801  …     0.011242
29214   0.880313 -0.474156 -0.015003   334.273964  …     0.011243
29215   0.880724 -0.473391 -0.015020   334.321743  …     0.011244
29216   0.881133 -0.472629 -0.015035   334.369269  …     0.011245

       body_vz_jpl  earth_x_jpl  earth_y_jpl  earth_z_jpl  RA_calc_jpl  \
0        -0.001627    -0.179765     0.970347    -0.000017   243.214514
1        -0.001627    -0.181915     0.969951    -0.000017   243.266024
2        -0.001628    -0.184064     0.969551    -0.000017   243.317732
3        -0.001628    -0.186212     0.969145    -0.000017   243.369364
4        -0.001628    -0.188359     0.968736    -0.000017   243.420684
…              …           …            …            …           …
29212    -0.000566    -0.161514     0.978014    -0.000019   333.918671
29213    -0.000566    -0.163673     0.977658    -0.000018   333.967115
29214    -0.000566    -0.165831     0.977297    -0.000018   334.015257
29215    -0.000565    -0.167988     0.976931    -0.000018   334.063021
29216    -0.000565    -0.170144     0.976560    -0.000018   334.110567

       DEC_calc_jpl  ux_calc_jpl  uy_calc_jpl  uz_calc_jpl
0        -17.106177    -0.430715    -0.899807     0.069515
1        -17.117396    -0.429922    -0.900189     0.069477
2        -17.128680    -0.429125    -0.900572     0.069438
3        -17.140047    -0.428330    -0.900954     0.069397
4        -17.151453    -0.427538    -0.901333     0.069354
…               …           …            …            …
29212    -11.709609     0.879479    -0.475703    -0.014963
29213    -11.692734     0.879896    -0.474930    -0.014984
29214    -11.675888     0.880311    -0.474161    -0.015003
29215    -11.659013     0.880722    -0.473396    -0.015020
29216    -11.642055     0.881130    -0.472634    -0.015035

[467472 rows x 28 columns]
```

[39]:
```python
# Report difference for between JPL and MSE calculated
print(f'Comparing direction of Asteroids from Palomar: JPL vs. MSE calc from␣
↪JPL positions')
print(f'(1) Direction according to JPL: radec2dir applied to JPL RA/DEC')
print(f'(2) Direction according to MSE: qv2dir applied to JPL positions &␣
↪velocities\n')
diff_ast_jpl_calc_jpl = obs_direction_diff(df_obs=df_obs, src1='jpl',␣
↪src2='calc_jpl', verbose=True)
```

```
Comparing direction of Asteroids from Palomar: JPL vs. MSE calc from JPL
positions
(1) Direction according to JPL: radec2dir applied to JPL RA/DEC
(2) Direction according to MSE: qv2dir applied to JPL positions & velocities

Angle Difference: calc_jpl vs. jpl
Mean  :   0.000243 deg (   0.873 seconds)
Median:   0.000255 deg (   0.918 seconds)
Max   :   0.000521 deg (   1.876 seconds)
```

Conclusion: qv2dir() is highly accurate in computing a right ascension and declination from position and velocity in the barycentric ecliptic plane. Errors are on the order of **0.87 arc seconds**. Differences with JPL are due to using a linear approximation to the adjustment of the space body's position due to light lag. The JPL calculation is iteratively solving for the position of the body on its true orbit at the instant photons leaving it hit the earth at print time. This simplified calculation is applying an adjustment of the form

```
r = norm(q_body - q_earth)
light_time = r / light_speed
dq = v_body * light_time
```

[40]:
```python
# Check that round trip between RA/DEC and direction is accurate

# extract u_jpl from df_obs
u_jpl = df_obs[['ux_jpl', 'uy_jpl', 'uz_jpl']].values
# extract RA, DEC and obstime_mjd
ra_jpl = df_obs.RA_jpl.values * deg
dec_jpl = df_obs.DEC_jpl.values * deg
obstime_mjd = df_obs.mjd.values

# Compute RA and DEC from direction computed by radec2dir() on the JPL data
ra_jpl2, dec_jpl2 = dir2radec(u_jpl, obstime_mjd)
```

[41]:
```python
# Compute difference in angles
diff_rt = radec_diff('JPL', 'MSE', ra1=ra_jpl, dec1=dec_jpl, ra2=ra_jpl2,␣
↪dec2=dec_jpl2,
                     obstime_mjd=obstime_mjd, verbose=True)
print(f'\nMean difference on round trip = {diff_rt:8.2e} arc seconds.')
```

```
Angle Difference: MSE vs. JPL
Mean  :   0.000000 deg (   0.000 seconds)
Median:   0.000000 deg (   0.000 seconds)
Max   :   0.000000 deg (   0.000 seconds)

Mean difference on round trip = 4.89e-11 arc seconds.
```

Conclusion: The round trip of radec2dir() and dir2radec() is accurate on the order of double precision. In the test, a direction was computed from the RA and DEC provided by JPL. This was then converted back to a RA and DEC. Errors are on the order of **5.8E-11 arc seconds**.

0.0.11 Calculate Asteroid Direction with MSE Position / Velocity

[42]:
```
ast_elt = asteroid_integrate.load_data()
# ast_elt.rename(mapper={'Num':'asteroid_num'}, axis='columns', inplace=True)
```

[43]:
```
# Range of asteroids to for data
ast_num_file_start: int = 1
ast_num_file_end: int = 1000
inputs, outputs = make_data_one_file(0, ast_num_file_end)
```

[44]:
```
ast_elt
```

[44]:
```
           Num       Name  epoch_mjd         a         e       inc     Omega  \
Num
1            1      Ceres    58600.0  2.769165  0.076009  0.184901  1.401596
2            2     Pallas    58600.0  2.772466  0.230337  0.608007  3.020817
3            3       Juno    58600.0  2.669150  0.256942  0.226699  2.964490
4            4      Vesta    58600.0  2.361418  0.088721  0.124647  1.811840
5            5    Astraea    58600.0  2.574249  0.191095  0.093672  2.470978
…          …         …          …       …         …         …         …
541124  541124  2018 RP23    58600.0  2.586399  0.289358  0.088749  2.000720
541125  541125  2018 RV23    58600.0  3.113036  0.213678  0.203046  0.544794
541126  541126  2018 RP24    58600.0  2.453880  0.176693  0.194504  2.649626
541127  541127  2018 RL26    58600.0  3.081248  0.081239  0.193310  2.381747
541128  541128  2018 RB27    58600.0  2.934075  0.064910  0.216796  0.496382

           omega         M      H     G      Ref         f            P  \
Num
1       1.284522  1.350398   3.34  0.12   JPL 46  1.501306  1683.145749
2       5.411373  1.041946   4.13  0.11   JPL 35  1.490912  1686.155979
3       4.330836  0.609557   5.33  0.32  JPL 108  0.996719  1592.787270
4       2.630709  1.673106   3.20  0.32   JPL 34 -4.436417  1325.432768
5       6.260280  4.928221   6.85  0.15  JPL 108 -1.738676  1508.600442
…            …         …      …     …        …         …
541124  3.913328  1.075531  17.30  0.15    JPL 7  1.654537  1519.293350
541125  0.242079  0.130760  16.10  0.15    JPL 8  0.206083  2006.201725
541126  3.695880  0.937231  17.30  0.15    JPL 6  1.258854  1404.036362
541127  3.426307  1.047446  16.00  0.15    JPL 6  1.195142  1975.551358
541128  5.871405  0.468771  15.50  0.15    JPL 4  0.531942  1835.714944

               n      long     theta    pomega      T_peri
Num
1       0.003733  4.036516  4.187424  2.686118 -361.745873
2       0.003726  3.190951  3.639917  2.149005 -279.616804
3       0.003945  1.621697  2.008860  1.012141 -154.522558
4       0.004740  6.115656  0.006132  4.442550 -352.940421
5       0.004165  1.093108  0.709396  2.448072  325.328481
```

```
...          ...        ...        ...        ...          ...
541124  0.004136  0.706394  1.285400 -0.369137 -260.066715
541125  0.003132  0.917633  0.992956  0.786873  -41.751113
541126  0.004475  0.999551  1.321174  0.062320 -209.433076
541127  0.003180  0.572315  0.720011 -0.475131 -329.336645
541128  0.003423  0.553374  0.616544  0.084602 -136.957715

[541073 rows x 19 columns]
```

[45]:
```
# The block of asteroid numbers to test (inclusive boundaries)
ast_num_min = 1
ast_num_max = 16

# The number of asteroids, times, and total rows we want to match
# Use daily data to match data file
N_ast = ast_num_max - ast_num_min + 1
N_t = df_earth_daily.mjd.size
N_row = N_ast * N_t
obstime_mjd = df_ast_daily.mjd.values

# Report data shape
print(f'Shape of data frames df_ast and df_obs:')
print(f'N_ast = {N_ast:5} asteroids')
print(f'N_t   = {N_t:5} observation times')
print(f'N_row = {N_row:5} rows in df_ast and df_obs')
```

```
Shape of data frames df_ast and df_obs:
N_ast =    16 asteroids
N_t   =  3653 observation times
N_row = 58448 rows in df_ast and df_obs
```

[46]:
```
# Filter for asteroid numbers
ast_num_file = np.arange(ast_num_file_start, ast_num_file_end, dtype=np.int64)
mask_ast = (ast_num_min <= ast_num_file) & (ast_num_file <= ast_num_max)

# MSE integrated times as one array
ts = inputs['ts'][0]

# Time range for JPL data
t_min = np.min(obstime_mjd)
t_max = np.max(obstime_mjd)

# Filter for MSE times that match
mask_t = (t_min <= ts) & (ts <= t_max)
```

[47]:
```
# Block of asteroid data
q_ast_all = outputs['q']
```

```python
v_ast_all = outputs['v']
u_ast_all = outputs['u']

# filter for selected asteroids only
q_ast_all_t = q_ast_all[mask_ast, :, :]
v_ast_all_t = v_ast_all[mask_ast, :, :]
u_ast_all_t = u_ast_all[mask_ast, :, :]

# filter for selected times only
q_ast_mse_3d = q_ast_all_t[:, mask_t, :]
v_ast_mse_3d = v_ast_all_t[:, mask_t, :]
u_ast_mse_3d = u_ast_all_t[:, mask_t, :]

# for some reason i don't understand, can't do these at once
# q_ast_mse = q_ast_all[mask_ast, mask_t, :]
q_ast_mse_3d.shape
```

[47]: (16, 3653, 3)

[48]:
```python
# Get position of Earth using utility function
q_earth_mse = get_earth_pos(obstime_mjd).transpose()

q_earth_mse.shape
```

[48]: (3, 58448)

[49]:
```python
# shape JPL positions to match q_ast_mse with three axes (asteroid_num,␣
↪time_idx, space_dim)
q_ast_jpl_3d = np.zeros((N_ast, N_t, 3))
q_ast_jpl_3d[:, :, 0] = df_ast_daily.X.values.reshape((N_ast, N_t))
q_ast_jpl_3d[:, :, 1] = df_ast_daily.Y.values.reshape((N_ast, N_t))
q_ast_jpl_3d[:, :, 2] = df_ast_daily.Z.values.reshape((N_ast, N_t))
q_ast_jpl_3d.shape
```

[49]: (16, 3653, 3)

[50]:
```python
# Reshape MSE asteroid data to match shape of DataFrame
q_ast_mse = np.zeros((3, N_row))
v_ast_mse = np.zeros((3, N_row))
u_ast_mse = np.zeros((3, N_row))
```

[51]:
```python
# Position
q_ast_mse[0, :] = q_ast_mse_3d[:, :, 0].reshape((-1))
q_ast_mse[1, :] = q_ast_mse_3d[:, :, 1].reshape((-1))
q_ast_mse[2, :] = q_ast_mse_3d[:, :, 2].reshape((-1))

# Velocity
```

```python
v_ast_mse[0, :] = v_ast_mse_3d[:, :, 0].reshape((-1))
v_ast_mse[1, :] = v_ast_mse_3d[:, :, 1].reshape((-1))
v_ast_mse[2, :] = v_ast_mse_3d[:, :, 2].reshape((-1))

# Direction from geocenter
u_ast_mse[0, :] = u_ast_mse_3d[:, :, 0].reshape((-1))
u_ast_mse[1, :] = u_ast_mse_3d[:, :, 1].reshape((-1))
u_ast_mse[2, :] = u_ast_mse_3d[:, :, 2].reshape((-1))
```

0.0.12 Check End to End Calculation of Direction to Geocenter in make_data_one_file()

[52]:
```python
# extract u_jpl from df_obs_daily
u_jpl = df_obs_daily[['ux_jpl', 'uy_jpl', 'uz_jpl']].values.T

# difference in directions as a vector
u_diff = u_ast_mse - u_jpl

# norm of difference, converted to arc seconds
u_diff_norm = np.linalg.norm(u_diff, axis=0)
angle_diff = np.rad2deg(u_diff_norm)*3600

# mean error in arc-seconds
mean_error = np.mean(angle_diff)
print(f'mean error: {mean_error:8.3f} arc seconds')
```

```
mean error:    0.974 arc seconds
```

Conclusion: The calculation encapsulated in make_data_one_file to earth geocenter is consistent with the manual calculation above. Tolerance to JPL is **0.97 arc seconds**

[53]:
```python
# df_obs_daily
```

0.0.13 Check Manual Calculation with qv2dir() and MSE Positions

[54]:
```python
# Compute direction from MSE position and velocity
u_mse, delta_mse = qv2dir(q_body=q_ast_mse*au, v_body=v_ast_mse*au/day,␣
 ↪q_earth=q_earth_mse*au)

# extract u_jpl from df_obs_daily
u_jpl = df_obs_daily[['ux_jpl', 'uy_jpl', 'uz_jpl']].values.T

# difference in directions as a vector
u_diff = u_mse - u_jpl

# norm of difference, converted to arc seconds
u_diff_norm = np.linalg.norm(u_diff, axis=0)
```

```python
angle_diff = np.rad2deg(u_diff_norm)*3600

# mean error in arc-seconds
mean_error = np.mean(angle_diff)
print(f'mean error: {mean_error:8.3f} arc seconds')
```

```
mean error:    0.974 arc seconds
```

Conclusion: My end to end calculation of astromentric RA and DEC are very close to those of JPL. In my calculation, I am only taking a single snapshot of planetary positions and velocities, plus orbital elements of the asteroids. Everything else is done by numerically integrating the system in rebound. I am computing an astrometric direction u on the unit sphere in the ecliptic frame, and comparing this to a direction from JPL. The JPL direction u_jpl is computed by applying radec2dir() on the quoted RA and DEC. Errors are on the order of **0.97 arc seconds**. I am guessing that one main source for the difference with JPL is that I used heliocentric rather than barycentric coordinates when saving the outputs of the rebound integration. I plan to switch to barycentric for the asteroid search. Of course there are also some other differences because these are completely separate calculations. JPL in particular is using many more massive bodies, and they are accounting for relativistic effects. Still, the bottom line is that an agreement of only 3.6 arc seconds is very tight and suggests that my methodology is basically sound.

0.0.14 Astrometric vs. Apparent Coordinates

[55]:
```python
df_obs
```

[55]:
```
       asteroid_num        mjd   JulianDate  time_key      RA_jpl    DEC_jpl  \
0                 1  55197.000  2455197.500   1324728  243.214830 -17.106252
1                 1  55197.125  2455197.625   1324731  243.266342 -17.117471
2                 1  55197.250  2455197.750   1324734  243.318051 -17.128756
3                 1  55197.375  2455197.875   1324737  243.369682 -17.140123
4                 1  55197.500  2455198.000   1324740  243.420999 -17.151530
…               …        …          …         …           …          …
29212            16  58848.500  2458849.000   1412364  333.918987 -11.709509
29213            16  58848.625  2458849.125   1412367  333.967430 -11.692635
29214            16  58848.750  2458849.250   1412370  334.015569 -11.675789
29215            16  58848.875  2458849.375   1412373  334.063332 -11.658913
29216            16  58849.000  2458849.500   1412376  334.110877 -11.641955

         ux_jpl    uy_jpl    uz_jpl  RA_apparent  …  body_vy_jpl  \
0     -0.430710 -0.899809  0.069515   243.357955  …    -0.007150
1     -0.429917 -0.900191  0.069477   243.409452  …    -0.007146
2     -0.429121 -0.900574  0.069438   243.461181  …    -0.007142
3     -0.428325 -0.900956  0.069397   243.512878  …    -0.007138
4     -0.427533 -0.901335  0.069354   243.564287  …    -0.007134
…           …         …         …            …  …            …
29212  0.879482 -0.475698 -0.014963   334.177362  …     0.011240
29213  0.879899 -0.474925 -0.014984   334.225801  …     0.011242
29214  0.880313 -0.474156 -0.015003   334.273964  …     0.011243
```

```
29215  0.880724 -0.473391 -0.015020   334.321743  …     0.011244
29216  0.881133 -0.472629 -0.015035   334.369269  …     0.011245

       body_vz_jpl  earth_x_jpl  earth_y_jpl  earth_z_jpl  RA_calc_jpl  \
0        -0.001627    -0.179765     0.970347    -0.000017   243.214514
1        -0.001627    -0.181915     0.969951    -0.000017   243.266024
2        -0.001628    -0.184064     0.969551    -0.000017   243.317732
3        -0.001628    -0.186212     0.969145    -0.000017   243.369364
4        -0.001628    -0.188359     0.968736    -0.000017   243.420684
…              …           …           …           …           …
29212    -0.000566    -0.161514     0.978014    -0.000019   333.918671
29213    -0.000566    -0.163673     0.977658    -0.000018   333.967115
29214    -0.000566    -0.165831     0.977297    -0.000018   334.015257
29215    -0.000565    -0.167988     0.976931    -0.000018   334.063021
29216    -0.000565    -0.170144     0.976560    -0.000018   334.110567

       DEC_calc_jpl  ux_calc_jpl  uy_calc_jpl  uz_calc_jpl
0        -17.106177    -0.430715    -0.899807     0.069515
1        -17.117396    -0.429922    -0.900189     0.069477
2        -17.128680    -0.429125    -0.900572     0.069438
3        -17.140047    -0.428330    -0.900954     0.069397
4        -17.151453    -0.427538    -0.901333     0.069354
…               …            …            …           …
29212    -11.709609     0.879479    -0.475703    -0.014963
29213    -11.692734     0.879896    -0.474930    -0.014984
29214    -11.675888     0.880311    -0.474161    -0.015003
29215    -11.659013     0.880722    -0.473396    -0.015020
29216    -11.642055     0.881130    -0.472634    -0.015035

[467472 rows x 28 columns]
```

JPL Definitions of Astrometric & Apparent RA/DEC

R.A.__________(ICRF)__________DEC = Astrometric right ascension and declination of the target center with respect to the observing site (coordinate origin) in the reference frame of the planetary ephemeris (ICRF). Compensated for down-leg light-time delay aberration.

Units: RA in decimal degrees (ddd.ffffffff) DEC in decimal degrees (sdd.ffffffff)

R.A.__________(airless-appar)__________DEC. = Airless apparent right ascension and declination of the target center with respect to an instantaneous reference frame defined by the Earth equator of-date (z-axis) and meridian containing the Earth equinox of-date (x-axis, IAU76/80). Compensated for down-leg light-time delay, gravitational deflection of light, stellar aberration, precession & nutation. Note: equinox (RA origin) is offset -53 mas from the of-date frame defined by the IAU06/00a P & N system.

Units: RA in decimal degrees (ddd.ffffffff) DEC in decimal degrees (sdd.ffffffff)

[56]:
```python
# alias the astrometric RA/DEC so the variable names look consistent
# these are the astrometric RA/DEC
ra_astro = ra_jpl
dec_astro = dec_jpl

# arrays of apparent asteroid angles from JPL
ra_appar = df_obs.RA_apparent.values * deg
dec_appar = df_obs.DEC_apparent.values * deg

# Compute difference in angles
diff_app = radec_diff('Astrometric', 'Apparent', ra1=ra_astro, dec1=dec_astro,␣
↪ra2=ra_appar, dec2=dec_appar,
                     obstime_mjd=obstime_mjd, verbose=False)
diff_mean = np.mean(diff_app)
diff_median = np.median(diff_app)
diff_max = np.max(diff_app)

# Report results
print(f'Mean Angle Difference: JPL astrometric vs. JPL apparent')
print(f'Mean  : {diff_mean:5.0f} seconds ({(diff_mean/3600):0.3f} degrees)')
print(f'Median: {diff_median:5.0f} seconds ({(diff_median/3600):0.3f} degrees)')
print(f'Max   : {diff_max:5.0f} seconds ({(diff_max/3600):0.3f} degrees)')
```

```
Mean Angle Difference: JPL astrometric vs. JPL apparent
Mean  :   743 seconds (0.206 degrees)
Median:   743 seconds (0.206 degrees)
Max   :   743 seconds (0.206 degrees)
```

Conclusion The difference between astrometric and apparent RA / DEC is really important! It's much more important than some of the other effects considered. It introduces errors on the order of **0.21 degrees / 743 arc seconds** We need to figure out the quotation basis of the ZTF data! Francisco from Alerce says he believes ZTF data "must be" astrometric. Hopefully this is correct!

[ ]:
```python

```