Low-precision Ephemeris of the Sun, Moon and Planets

This document describes a suite of MATLAB functions that compute low-precision ephemerides of the sun, moon, and planets. The MATLAB functions that compute an ephemeris for the Sun, Moon and all planets except Pluto are based on the algorithms described in *Low-Precision Formulae for Planetary Positions*, T. C. Van Flandern and K. F. Pulkkinen, *The Astrophysical Journal Supplement Series*, **41**:391-411, November 1979. To the precision of implemented algorithm (one arc minute), these coordinates can be considered to be true-of-date.

Each algorithm uses time arguments given by

$$t = JD - 2451545$$

$$T = t/36525 + 1$$

where JD is the Julian date.

Each function uses fundamental trigonometric arguments (in revolutions) of the following form:

$$G_s = 0.993126 + 0.00273777850t$$

$$G_2 = 0.140023 + 0.00445036173t$$

$$G_4 = 0.053856 + 0.00145561327t$$

$$G_5 = 0.056531 + 0.00023080893t$$

$$F_4 = 0.849694 + 0.00145569465t$$

$$L_4 = 0.987353 + 0.00145575328t$$

The heliocentric, ecliptic longitude λ, latitude β and distance r are computed from series involving these arguments. These series are of the form

$$\lambda = L_4 + 38451\sin G_4 + 2238\sin(2G_4) + 181\sin(3G_4) + \ldots$$

$$\beta = 6603\sin F_4 + 622\sin(G_4 - F_4) + 615\sin(G_4 + F_4) + \ldots$$

$$r = 1.53031 - 0.1417\cos G_4 - 0.0066\cos(2G_4) + \ldots$$

where the unit of r is Astronomical Units (AU). In these MATLAB functions, the value of AU is equal to `149597870.691` kilometers.

The heliocentric, ecliptic position vector of the planet is determined from

$$\mathbf{r} = r\begin{Bmatrix} \cos\beta\cos\lambda \\ \cos\beta\sin\lambda \\ \sin\beta \end{Bmatrix}$$

sun1.m – solar ephemeris

This function calculates the true-of-date geocentric right ascension, declination and position vector of the Sun. The syntax of this MATLAB function is

```
function [rasc, decl, rsun] = sun1 (jdate)

% solar ephemeris

% input

%  jdate = Julian date

% output

%  rasc = right ascension of the sun (radians)
%         (0 <= rasc <= 2 pi)
%  decl = declination of the sun (radians)
%         (-pi/2 <= decl <= pi/2)
%  rsun = eci position vector of the sun (km)
```

moon.m – lunar ephemeris

This function calculates the true-of-date geocentric right ascension, declination and position vector of the Moon. The syntax of this MATLAB function is

```
function [rasc, decl, rmoon] = moon (jdate)

% lunar ephemeris

% input

%  jdate = julian date

% output

%  rasc  = right ascension of the moon (radians)
%          (0 <= rasc <= 2 pi)
%  decl  = declination of the moon (radians)
%          (-pi/2 <= decl <= pi/2)
%  rmoon = eci position vector of the moon (km)
```

mercury.m – Mercury ephemeris

This function calculates the true-of-date heliocentric ecliptic position vector of Mercury. The syntax of this MATLAB function is

```
function rmercury = mercury (jdate)

% true-of-date heliocentric, ecliptic
% position vector of Mercury

% input

%  jdate = julian date
```

```
% output

%  rmercury = position vector of Mercury (km)
```

venus.m – Venus ephemeris

This function calculates the true-of-date heliocentric ecliptic position vector of Venus. The syntax of this MATLAB function is

```
function rvenus = venus (jdate)

% true-of-date heliocentric, ecliptic
% position vector of venus

% input

%  jdate = Julian date

% output

%  rvenus = position vector of Venus (km)
```

earth.m – Earth ephemeris

This function calculates the true-of-date heliocentric ecliptic position vector of the Earth. The syntax of this MATLAB function is

```
function rearth = earth (jdate)

% true-of-date heliocentric, ecliptic
% position vector of the Earth

% input

%  jdate = Julian date

% output

%  rearth = position vector of the Earth (km)
```

mars.m – Mars ephemeris

This function calculates the true-of-date heliocentric ecliptic position vector of Mars. The syntax of this MATLAB function is

```
function rmars = mars (jdate)

% true-of-date heliocentric, ecliptic
% position vector of Mars

% input

%  jdate = Julian date

% output
```

```
%  rmars = position vector of Mars (km)
```

jupiter.m – Jupiter ephemeris

This function calculates the true-of-date heliocentric ecliptic position vector of Jupiter. The syntax of this MATLAB function is

```
function rjupiter = jupiter (jdate)

% true-of-date heliocentric, ecliptic
% position vector of Jupiter

% input

%  jdate = Julian date

% output

%  rjupiter = position vector of Jupiter (km)
```

saturn.m – Saturn ephemeris

This function calculates the true-of-date heliocentric ecliptic position vector of Saturn. The syntax of this MATLAB function is

```
function rsaturn = saturn (jdate)

% true-of-date heliocentric, ecliptic
% position vector of Saturn

% input

%  jdate = Julian date

% output

%  rsaturn = position vector of Saturn (km)
```

uranus.m – Uranus ephemeris

This function calculates the true-of-date heliocentric ecliptic position vector of Uranus. The syntax of this MATLAB function is

```
function ruranus = uranus (jdate)

% true-of-date heliocentric, ecliptic
% position vector of Uranus

% input

%  jdate = Julian date

% output

%  ruranus = position vector of Uranus (km)
```

neptune.m – Neptune ephemeris

This function calculates the true-of-date heliocentric ecliptic position vector of Neptune. The syntax of this MATLAB function is

```
function rneptune = neptune (jdate)

% true-of-date heliocentric, ecliptic
% position vector of Neptune

% input

%  jdate = Julian date

% output

%  rneptune = position vector of Neptune (km)
```

pluto.m – Pluto ephemeris

This MATLAB function calculates the heliocentric position vector of Pluto relative to the ecliptic and equinox of J2000. This algorithm is based on the method described in Chapter 36 of *Astronomical Algorithms* by Jean Meeus.

The fundamental time argument for this method is a function of the Julian Ephemeris Date *JED* as follows:

$$T = \frac{JED - 2451545}{36525}$$

The heliocentric ecliptic coordinates of Pluto are computed from series of the form

$$\lambda = \lambda^m + \sum_{i=1}^{43} A \sin\alpha + B\cos\alpha$$

where

$\lambda^m =$ coordinate mean value

$\alpha = iJ + jS + kP$

$J, S, P =$ mean longitudes of Jupiter, Saturn and Pluto

$i, j, k =$ integer constants

$A, B =$ coefficients of periodic term

The syntax of this MATLAB function is

```
function rpluto = pluto(jdate)

% heliocentric coordinates of pluto

% heliocentric position vector of Pluto
% ecliptic and equinox of J2000
```

```
% input

%  jdate = Julian date

% output

%  rpluto = position vector of Pluto (km)
```

The following is the MATLAB source code that performs the revolution to radians calculation required in these ephemeris functions.

```
function y = r2r (x)

% revolutions to radians function

% input

%  x = argument (revolutions; 0 <= x <= 1)

% output

%  y = equivalent x (radians; 0 <= y <= 2 pi)

% Orbital Mechanics with Matlab

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

y = 2.0 * pi * (x - fix(x));
```

This software suite also includes a simple script called `demo_lpe.m` that demonstrates how to interact with these MATLAB functions. The following is a typical user interaction with this script.

```
demo_lpe - demonstrates how to use the low-precision ephemeris functions

please input the calendar date
(1 <= month <= 12, 1 <= day <= 31, year = all digits!)
? 11,25,2012

    celestial body menu

  <1> Mercury
  <2> Venus
  <3> Earth
  <4> Mars
  <5> Jupiter
  <6> Saturn
  <7> Uranus
  <8> Neptune
  <9> Pluto
  <10> Moon
  <11> Sun

please select the celestial body
? 4


program demo_lpe

celestial body        'Mars'


UTC calendar date       25-Nov-2012
```

```
UTC Julian date         2456256.500000

heliocentric, ecliptic position vector and magnitude

        rx (km)                ry (km)                rz (km)                rmag (km)
 +1.00010670219172e+008  -1.85067063915505e+008  -6.34030746242878e+006  +2.10457007017376e+008
```

Here’s the screen display for the Moon on the same calendar date. Note that the lunar and solar ephemeris calculations are geocentric and equatorial.

```
program demo_lpe

celestial body        'Moon'


UTC calendar date       25-Nov-2012

UTC Julian date         2456256.500000

geocentric declination          11.880497 degrees

geocentric right ascension      20.994086 degrees

geocentric, equatorial position vector and magnitude

        rx (km)                ry (km)                rz (km)                rmag (km)
 +3.66912826199245e+005  +1.40801188411522e+005  +8.26786524412326e+004  +4.01603979384059e+005
```