Numerical Prediction of Orbital Events

This document describes an interactive MATLAB script called `npoe.m` that can model important orbital events of satellites in Earth orbits. This script implements a special perturbation solution of orbital motion using a variable step size Runge-Kutta-Fehlberg (RKF78) integration method to numerically integrate Cowell's form of the system of differential equations. Orbital events are predicted using Brent's method for finding the root of a single nonlinear equation. The user can control both the integration and root-finding convergence criteria.

This script can accurately predict the time and orbit characteristics for any *physically realizable* user-defined value of the following orbit parameters:

- geodetic altitude
- geodetic latitude
- east longitude
- geocentric declination
- true anomaly
- argument of latitude
- flight path angle
- orbital speed
- right ascension

The software can be easily modified to include other orbital events and more sophisticated equations of orbital motion.

The `npoe.m` script can model one or more of the following types of orbit perturbations during the event prediction process:

- Earth gravity - user-defined degree and order
- Solar gravity - point mass
- Lunar gravity - point mass
- Atmospheric drag with a U.S. Standard 1976 atmosphere model
- Solar radiation pressure

The user can provide the initial orbital elements for this script interactively or by specifying the name of a data file. The following is a typical data file for this application. When creating this type of ASCII text file the user can change the numeric data and annotation, but do not change the number of lines in the file or the line location of the data. Please note the units and valid range for each data item.

```
semimajor axis (kilometers)
(semimajor axis > 0)
6878.14

orbital eccentricity (non-dimensional)
(0 <= eccentricity < 1)
.0125

orbital inclination (degrees)
(0 <= inclination <= 180)
28.5

argument of perigee (degrees)
(0 <= argument of perigee <= 360)
270

right ascension of the ascending node (degrees)
(0 <= RAAN <= 360)
45

true anomaly (degrees)
(0 <= true anomaly <= 360)
0
```

The following is a typical user interaction with this MATLAB script. Please note the units and valid range for each input. Smaller integration and root-finding error tolerances will predict the orbit and events more accurately at the expense of longer run time.

```
                program npoe

< numerical prediction of orbital events >


initial calendar date and time

please input the calendar date
(1 <= month <= 12, 1 <= day <= 31, year = all digits!)
? 1,1,2001

please input the universal time
(0 <= hours <= 24, 0 <= minutes <= 60, 0 <= seconds <= 60)
? 0,0,0


please input the simulation period (days)
? 5


algorithm control parameters


please input the integration error tolerance
(a value of 1.0e-8 is recommended)
? 1e-8
```

```
please input the root-finding error tolerance
(a value between 1.0e-4 and 1.0e-6 is recommended)
? 1e-4


gravity model inputs


please input the degree of the gravity model (zonals)
(0 <= zonals <= 18)
? 2

please input the order of the gravity model (tesserals)
(0 <= tesserals <= 18)
? 0


orbital perturbations


would you like to include solar perturbations (y = yes, n = no)
? n

would you like to include lunar perturbations (y = yes, n = no)
? n

would you like to include drag perturbations (y = yes, n = no)
? y

would you like to include srp perturbations (y = yes, n = no)
? n


 orbital elements menu

   <1> user input

   <2> data file

? 1

please input the semimajor axis (kilometers)
(semimajor axis > 0)
? 8000

please input the orbital eccentricity (non-dimensional)
(0 <= eccentricity < 1)
? .025

please input the orbital inclination (degrees)
(0 <= inclination <= 180)
? 45

please input the argument of perigee (degrees)
(0 <= argument of perigee <= 360)
? 200
```

```
please input the right ascension of the ascending node (degrees)
(0 <= raan <= 360)
? 100

please input the true anomaly (degrees)
(0 <= true anomaly <= 360)
? 45


aerodynamic drag inputs


please input the drag coefficient (non-dimensional)
? 2

please input the cross-sectional area (square meters)
? 10

please input the spacecraft mass (kilograms)
? 2000

please select the orbital element to predict

   <1> geodetic altitude
   <2> geodetic latitude
   <3> east longitude
   <4> geocentric declination
   <5> true anomaly
   <6> argument of latitude
   <7> flight path angle
   <8> orbital speed
   <9> right ascension

? 2

please input the geodetic latitude (degrees)
(-90 <= latitude <= +90)
? 20
```

The following is the program output for this example

```
                     program npoe

    < numerical prediction of orbital events >

degree of gravity model          2.0000

order of gravity model           0.0000

integration tolerance       1.0000e-008

root-finding tolerance      1.0000e-004

simulation includes drag perturbations
```

```
calendar date       01-Jan-2001

universal time        00:48:11

      sma (km)        eccentricity     inclination (deg)    argper (deg)
 8.0046872515e+003  2.4673077710e-002  4.5017240386e+001  1.9943354822e+002

     raan (deg)     true anomaly (deg)   arglat (deg)       period (min)
 9.9896464322e+001  1.8933803861e+002  2.8771586830e+001  1.1878900675e+002

     lan (deg)           energy           fpa (deg)           rasc (deg)
 3.4710599597e+002  -2.4897943385e+001  -2.3510205531e-001  1.2111077845e+002

  declination (deg)   latitude (deg)    longitude (deg)      altitude (km)
 1.9904071046e+001  2.0000000000e+001  8.3203101057e+000  1.8237851183e+003

   r-perigee (km)     r-apogee (km)     v-perigee (kps)     v-apogee (kps)
 7.8071869809e+003  8.2021875221e+003  7.2329304996e+000  6.8846073901e+000

   h-perigee (km)     h-apogee (km)    lat-perigee (deg)    lat-apogee (deg)
 1.4302355707e+003  1.8252358168e+003  -1.3683384750e+001  1.3679899680e+001

 mean motion (dps)     manom (deg)         eanom (deg)
 3.2364115037e+006  1.8980531206e+002  1.8957028025e+002

   < please press any key to continue >


calendar date       01-Jan-2001

universal time        01:29:05

      sma (km)        eccentricity     inclination (deg)    argper (deg)
 8.0051311556e+003  2.5342296778e-002  4.5017869159e+001  1.9894940453e+002

     raan (deg)     true anomaly (deg)   arglat (deg)       period (min)
 9.9772686198e+001  3.1228552300e+002  1.5123492753e+002  1.1879888815e+002

     lan (deg)           energy           fpa (deg)           rasc (deg)
 3.3672863502e+002  -2.4896562733e+001  -1.0560675477e+000  2.5856379278e+002

  declination (deg)   latitude (deg)    longitude (deg)      altitude (km)
 1.9900002960e+001  1.9999999928e+001  1.3551974161e+002  1.4902153206e+003

   r-perigee (km)     r-apogee (km)     v-perigee (kps)     v-apogee (kps)
 7.8022627461e+003  8.2079995651e+003  7.2375748868e+000  6.8798079805e+000

   h-perigee (km)     h-apogee (km)    lat-perigee (deg)    lat-apogee (deg)
 1.4252550418e+003  1.8309915712e+003  -1.3349423662e+001  1.3345924950e+001

 mean motion (dps)     manom (deg)         eanom (deg)
 3.2365012409e+006  3.1440662541e+002  3.1335077747e+002
```

Technical Discussion

Cowell's method is a *special perturbation* technique that numerically integrates the vector system of second-order, nonlinear differential equations of motion of a satellite given by

$$\mathbf{a}(\mathbf{r},\mathbf{v},t) = \ddot{\mathbf{r}}(\mathbf{r},\dot{\mathbf{r}},t) = \mathbf{a}_g(\mathbf{r}) + \mathbf{a}_d(\mathbf{r},\mathbf{v},t) + \mathbf{a}_{sm}(\mathbf{r},t) + \mathbf{a}_{srp}(\mathbf{r},t)$$

where

$t =$ Universal time

$\mathbf{r} =$ inertial position vector of the satellite

$\mathbf{v} =$ inertial velocity vector of the satellite

$\mathbf{a}_g =$ acceleration due to gravity

$\mathbf{a}_d =$ acceleration due to atmospheric drag

$\mathbf{a}_{sm} =$ acceleration due to the Sun and Moon

$\mathbf{a}_{srp} =$ acceleration due to solar radiation pressure

The satellite's orbital motion is modeled with respect to a *true-of-date* Earth-centered-inertial (ECI) coordinate system. The origin of this system is the center of the Earth and the fundamental plane is the Earth's equator. The *x*-axis is aligned with the true-of-date Vernal Equinox, the *z*-axis is aligned with the Earth's spin axis, and the *y*-axis completes this orthogonal, right-handed coordinate system.

This MATLAB script uses a *spherical harmonic* representation of the Earth's geopotential function given by

$$\Phi(r,\phi,\lambda) = \frac{\mu}{r} + \frac{\mu}{r}\sum_{n=1}^{\infty} C_n^0 \left(\frac{R}{r}\right)^n P_n^0(u) + \frac{\mu}{r}\sum_{n=1}^{\infty}\sum_{m=1}^{n}\left(\frac{R}{r}\right)^n P_n^m(u)\left[S_n^m \sin m\lambda + C_n^m \cos m\lambda\right]$$

where ϕ is the geocentric latitude of the satellite, λ is the geocentric east longitude of the satellite and $r = |\mathbf{r}| = \sqrt{x^2 + y^2 + z^2}$ is the geocentric distance of the satellite. In this expression the *S's* and *C's* are *unnormalized* harmonic coefficients of the geopotential, and the *P's* are associated Legendre polynomials of degree *n* and order *m* with argument $u = \sin\phi$.

The software calculates the satellite's acceleration due to the Earth's gravity field with a vector equation derived from the gradient of the potential function expressed as

$$\mathbf{a}_g(\mathbf{r},t) = \nabla\Phi(\mathbf{r},t)$$

This acceleration vector is a combination of pure two-body or *point mass* gravity acceleration and the gravitational acceleration due to higher order nonspherical terms in the Earth's geopotential. In terms of the Earth's geopotential Φ, the inertial rectangular cartesian components of the satellite's acceleration vector are as follows:

$$\ddot{x} = \left(\frac{1}{r}\frac{\partial \Phi}{\partial r} - \frac{z}{r^2\sqrt{x^2+y^2}}\frac{\partial \Phi}{\partial \phi} \right) x - \left(\frac{1}{x^2+y^2}\frac{\partial \Phi}{\partial \lambda} \right) y$$

$$\ddot{y} = \left(\frac{1}{r}\frac{\partial \Phi}{\partial r} - \frac{z}{r^2\sqrt{x^2+y^2}}\frac{\partial \Phi}{\partial \phi} \right) y + \left(\frac{1}{x^2+y^2}\frac{\partial \Phi}{\partial \lambda} \right) x$$

$$\ddot{z} = \left(\frac{1}{r}\frac{\partial \Phi}{\partial r} \right) z + \left(\frac{\sqrt{x^2+y^2}}{r^2}\frac{\partial \Phi}{\partial \phi} \right)$$

The three partial derivatives of the geopotential with respect to r,ϕ,λ are given by

$$\frac{\partial \Phi}{\partial r} = -\frac{1}{r}\left(\frac{\mu}{r}\right)\sum_{n=2}^{N}\left(\frac{R}{r}\right)^n (n+1)\sum_{m=0}^{n}\left(C_n^m \cos m\lambda + S_n^m \sin m\lambda\right)P_n^m\left(\sin\phi\right)$$

$$\frac{\partial \Phi}{\partial \phi} = \left(\frac{\mu}{r}\right)\sum_{n=2}^{N}\left(\frac{R}{r}\right)^n \sum_{m=0}^{n}\left(C_n^m \cos m\lambda + S_n^m \sin m\lambda\right)\left[P_n^{m+1}\left(\sin\phi\right) - m\tan\phi\, P_n^m\left(\sin\phi\right)\right]$$

$$\frac{\partial \Phi}{\partial \lambda} = \left(\frac{\mu}{r}\right)\sum_{n=2}^{N}\left(\frac{R}{r}\right)^n \sum_{m=0}^{n} m\left(S_n^m \cos m\lambda - C_n^m \sin m\lambda\right)P_n^m\left(\sin\phi\right)$$

where

$R =$ radius of the Earth

$r =$ geocentric distance of the satellite

$S_n^m, C_n^m =$ harmonic coefficients

$\phi =$ geocentric latitude of the satellite $= \arcsin\left(\dfrac{z}{r}\right)$

$\lambda =$ longitude of the satellite $= \alpha - \alpha_g$

$\alpha =$ right ascension of the satellite $= \arctan\left(\dfrac{y}{x}\right)$

$\alpha_g =$ right ascension of Greenwich

Right ascension is measure positive east of the vernal equinox, longitude is measured positive east of Greenwich, and latitude is positive above the Earth's equator and negative below.

For $m = 0$ the coefficients are called *zonal* terms, when $m = n$ the coefficients are *sectorial* terms, and for $n > m \neq 0$ the coefficients are called *tesseral* terms.

The Legendre polynomials with argument $\sin\phi$ are computed using recursion relationships given by:

$$
\begin{aligned}
&P_n^0(\sin\phi) = \frac{1}{n}\left[(2n-1)\sin\phi\, P_{n-1}^0(\sin\phi) - (n-1)P_{n-2}^0(\sin\phi)\right] \\
&P_n^n(\sin\phi) = (2n-1)\cos\phi\, P_{n-1}^{n-1}(\sin\phi), \qquad\qquad m \ne 0,\ m < n \\
&P_n^m(\sin\phi) = P_{n-2}^m(\sin\phi) + (2n-1)\cos\phi\, P_{n-1}^{m-1}(\sin\phi), \quad m \ne 0,\ m = n
\end{aligned}
$$

where the first few associated Legendre functions are given by

$$
P_0^0(\sin\phi) = 1, \quad P_1^0(\sin\phi) = \sin\phi, \quad P_1^1(\sin\phi) = \cos\phi
$$

and $P_i^j = 0$ for $j > i$.

The trigonometric arguments are determined from expansions given by

$$
\begin{aligned}
\sin m\lambda &= 2\cos\lambda \sin(m-1)\lambda - \sin(m-2)\lambda \\
\cos m\lambda &= 2\cos\lambda \cos(m-1)\lambda - \cos(m-2)\lambda \\
m\tan\phi &= (m-1)\tan\phi + \tan\phi
\end{aligned}
$$

The acceleration experienced by the satellite due to atmospheric drag is computed using the following vector expression:

$$
\mathbf{a}_d(\mathbf{r},\mathbf{v},t) = -\frac{1}{2}\rho(\mathbf{r},t)\left|\mathbf{v}_r\right|\mathbf{v}_r \frac{C_d A}{m}
$$

where

$\mu =$ gravitational constant of the Earth
$\mathbf{v}_r =$ satellite velocity vector relative to the atmosphere
$\rho =$ atmospheric density
$C_d =$ drag coefficient of the satellite
$A =$ reference area of the satellite
$m =$ mass of the satellite

During orbit propagation the software uses constant values for the mass, drag coefficient and the reference area of the satellite. Please note that the reference area is measured perpendicular to the relative velocity vector.

The aerodynamic drag algorithm assumes that the atmosphere rotates at the same angular speed as the Earth. With this assumption the relative velocity vector is given by

$$
\mathbf{v}_r = \mathbf{v} - \mathbf{w} \times \mathbf{r}
$$

where $\vec{\omega}$ is the inertial rotation vector of the Earth. The angular velocity vector of the Earth is

$$\mathbf{w} = \omega_e \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

where $\omega_e = 7.2921151467\text{E-5}$ radians per second.

The cross product expansion of the previous equation gives the three components of the relative velocity vector as follows:

$$\mathbf{v}_r = \begin{bmatrix} v_x + \omega_e r_y \\ v_y - \omega_e r_x \\ v_z \end{bmatrix}$$

The calculation of atmospheric density in this script is based on the 1976 U.S. Standard atmosphere.

The acceleration contribution of the Sun and Moon represented by *point masses* is given by

$$\mathbf{a}_{sm}\left(\mathbf{r},t\right) = -\mu_m \left(\frac{\mathbf{r}_{m-b}}{\left|\mathbf{r}_{m-b}\right|^3} + \frac{\mathbf{r}_{e-m}}{\left|\mathbf{r}_{e-m}\right|^3} \right) - \mu_s \left(\frac{\mathbf{r}_{s-b}}{\left|\mathbf{r}_{s-b}\right|^3} + \frac{\mathbf{r}_{e-s}}{\left|\mathbf{r}_{e-s}\right|^3} \right)$$

where

$\mu_m =$ gravitational constant of the Moon

$\mu_s =$ gravitational constant of the Sun

$\mathbf{r}_{m-b} =$ position vector from the Moon to the satellite

$\mathbf{r}_{s-b} =$ position vector from the Sun to the satellite

$\mathbf{r}_{e-m} =$ position vector from the Earth to the Moon

$\mathbf{r}_{e-s} =$ position vector from the Earth to the Sun

The solar and lunar ephemerides used in this program are computer implementations of the numerical method 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. The values of the gravitational constants are as follows:

$$\mu_m = 4902.793 \frac{\text{km}^3}{\text{sec}^2} \qquad \mu_s = 132712438000 \frac{\text{km}^3}{\text{sec}^2}$$

We can define a *solar radiation constant* for any satellite as a function of its size, mass and surface reflective properties according to the equation:

$$C_{srp} = \gamma\, P_s\, a^2 \frac{A}{m}$$

where

$\gamma =$ reflectivity constant
$P_s =$ solar radiation constant
$a =$ astronomical unit
$A =$ surface area normal to the incident radiation
$m =$ mass of the satellite

The reflectivity constant is a dimensionless number between 0 and 2. For a perfectly absorbent body $\gamma = 1$, for a perfectly reflective body $\gamma = 2$, and for a translucent body $\gamma < 1$. For example, the reflectivity constant for an aluminum surface is approximately 1.96.

The value of the solar radiation pressure on a perfectly absorbing satellite surface at a distance of one Astronomical Unit from the Sun is

$$P_s = 4.4 \times 10^{-3} \frac{kg}{km - \sec^2}$$

The acceleration vector of the satellite due to solar radiation pressure is given by:

$$\mathbf{a}_{srp} = C_{srp} \frac{\mathbf{r}_{sat-to-Sun}}{r^3_{sat-to-Sun}}$$

where

$\mathbf{r}_{sat-to-Sun} = \mathbf{r}_{sat} - \mathbf{r}_{Earth-to-Sun}$
$\mathbf{r}_{sat} =$ geocentric, inertial position vector of the satellite
$\mathbf{r}_{Earth-to-Sun} =$ geocentric, inertial position vector of the Sun

During the integration process, the software must determine if the satellite is in Earth shadow or sunlight. Obviously, there can be no solar radiation perturbation during Earth eclipse of the satellite orbit. The software makes use of a *shadow parameter* to determine eclipse conditions. This parameter is defined by the following expression:

$$\varphi = -\frac{\left|\mathbf{r}_{sc} \times \mathbf{r}_{es}\right|}{\left|\mathbf{r}_{es}\right|} sign(\mathbf{r}_{sc} \bullet \mathbf{r}_{es})$$

where $\mathbf{r}_{sc} =$ is the geocentric, inertial position vector of the satellite and $\mathbf{r}_{es} =$ is the geocentric, inertial position vector of the Sun relative to the satellite.

The *critical* values of the shadow parameter for the penumbra (subscript *p*) and umbra part (subscript *u*) of the shadow are given by:

$$\varphi_p = |\mathbf{r}_{sc}| \sin \psi_p$$

$$\varphi_u = |\mathbf{r}_{sc}| \sin \psi_u$$

The penumbra and umbra shadow angles are found from:

$$\psi_p = \eta + \theta_p$$

$$\psi_u = \eta - \theta_u$$

These are the angles between the geocentric anti-Sun vector and the vector to a satellite at the time of shadow entrance or exit.

If we represent the shadow as a cylinder, the shadow angle is given by:

$$\eta = \sin^{-1}\left(\frac{r_e}{r_{sc}}\right)$$

The corresponding penumbra and umbra *cone* angles are as follows:

$$\theta_p = \sin^{-1}\left(\frac{r_s + r_e}{r_{es}}\right)$$

$$\theta_u = \sin^{-1}\left(\frac{r_s - r_e}{r_{es}}\right)$$

where

$r_e =$ radius of the Earth
$r_s =$ radius of the Sun
$r_{es} =$ distance from the Earth to the Sun

If the condition $\varphi_u < \varphi \le \varphi_p$ is true, the satellite is in the penumbra part of the Earth's shadow, and if the inequality $0 \le \varphi \le \varphi_u$ is true, the satellite is in the umbra part of the shadow. If the absolute value of the shadow parameter is larger than the penumbra value, the satellite is in full sunlight. The shadow calculations also assume that the Earth's atmosphere increases the radius of the Earth by two percent.