Repeating Ground Track Orbit Design

This document describes four MATLAB scripts that can be used to design and analyze repeating ground track orbits. Scripts are provided for both preliminary and high fidelity orbit design.

These types of orbits are useful for remote sensing and other special applications since they overfly the same points on the Earth's surface every repeat cycle. Repeating ground track orbits are usually specified by an integer number of days *N* and integer number of orbits *K* in the repeat cycle.

The repeating ground track design equation is

$$\frac{\left(\dfrac{N}{K}\right)}{\omega_e - \dot{\Omega}} - \frac{1}{\dot{\omega} + \dot{M}} = 0$$

where

K = integer number of orbits in repeat cycle

N = integer number of days in repeat cycle

ω_e = inertial rotation rate of the Earth

$\dot{\Omega}$ = RAAN perturbation

ω = argument of perigee perturbation

$\dot{M}$ = mean anomaly perturbation

The perturbations of the mean argument of perigee and mean anomaly due to J_2 are given by the next two equations:

$$\dot{M} = \tilde{n} = \frac{dM}{dt} = n\left\{1 + \frac{3}{2} J_2 \left(\frac{r_{eq}}{p}\right)^2 \sqrt{1-e^2}\left(1 - \frac{3}{2}\sin^2 i\right)\right\}$$

$$\dot{\omega} = \frac{d\omega}{dt} = \frac{3}{2} J_2 \tilde{n} \left(\frac{r_{eq}}{p}\right)^2 \left(2 - \frac{5}{2}\sin^2 i\right)$$

where

$n = \sqrt{\mu / a^3}$ = mean motion

$p = a\left(1 - e^2\right)$ = semiparameter

a = semimajor axis

e = orbital eccentricity

i = orbital inclination

J_2 = second gravity harmonic of the Earth

r_{eq} = equatorial radius of the Earth

repeat1.m – time to repeat ground track – Kozai orbit propagation

This MATLAB script estimates the time required for an Earth satellite to repeat its ground track. The mean orbital elements are propagated using Kozai's algorithm and the user can select a closure tolerance for the ground track.

The algorithm begins by initializing the Earth-relative longitude of the ascending node and the total number of days according to

$$\lambda_{an} = 0 \qquad ndays = 0$$

The nodal period is computed using the expression $\tau_n = \dfrac{2\pi}{\left(\tilde{n} + \dot{\omega}\right)}$ where $\tilde{n}$ is the "perturbed" mean motion and $\dot{\omega}$ is the perturbation of the argument of perigee due to Earth oblateness.

The delta-longitude at the ascending node per nodal period is given by

$$\Delta\lambda = \tau_n \left(\omega_e - \dot{\Omega}\right)$$

where ω_e is the inertial rotation rate of the Earth and $\dot{\Omega}$ is the perturbation of the right ascension of the ascending node due to Earth oblateness.

The current Earth relative longitude and number of orbits are incremented according to

$$\lambda_{i+1} = \lambda_i + \Delta\lambda$$

$$norbits = norbits + 1$$

After each increment, convergence is checked. If $\left|\lambda - 2\pi\right| \le \varepsilon$ or $\left|\lambda\right| \le \varepsilon$ the method has satisfied the user-defined closure tolerance ε. The total number of *solar* days to repeat the ground track is determined from $ndays = norbits\,\tau_n / 86400$.

The following is a typical user interaction with this script. Please note that the semimajor axis, orbital eccentricity and inclination are Kozai mean orbital elements. User input is in bold font.

```
              program repeat1

     < time to repeat ground track - analytic solution >


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

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

   please input the orbital inclination (degrees)
   (0 <= inclination <= 180)
   ? 28.5
```

```
please input the closure tolerance (degrees)
(a value between 0.1 and 0.5 degrees is recommended)
? 0.1
```

The following is the output screen generated with this application.

```
                    program repeat1

  < time to repeat ground track - analytic solution >

mean semimajor axis                 8000.000000  kilometers
mean eccentricity (nd)                 0.000000
mean inclination                      28.500000  degrees
mean argument of perigee               0.000000  degrees
mean raan                              0.000000  degrees

number of orbits to repeat          2075.000000
number of solar days to repeat       170.653126

Keplerian period                     118.684684  minutes
nodal period                         118.429158  minutes

length of nodal day                 1420.466169  minutes
fundamental interval                  30.014440  degrees

closure tolerance                      0.100000  degrees
actual closure                         0.036832  degrees
```

repeat2.m – time to repeat ground track - numerical integration

This application estimates the time required for a satellite to repeat its ground track. The satellite's orbit is propagated using numerical integration and the user can select a closure tolerance.

The main nonlinear equation solved by this script is given by

$$f\left(t\right)=\lambda_{an}^{0}-\lambda_{an}^{p}=0$$

where λ_{an}^{0} is the initial east longitude of the ascending nodal crossing and λ_{an}^{p} is the ascending node east longitude predicted by the software. By solving the *f(t)* equation, we are trying to minimize the difference between the initial east longitude of the ascending node and the ascending node east longitude predicted by the software for the user-defined number of integer orbits in the repeat cycle.

The solution of this nonlinear equation uses a Runge-Kutta-Fehlberg 7(8) algorithm imbedded within Brent's one dimensional root-finding method.

The east longitude of the ascending node can be determined from the x and y components of the Earth-centered-fixed (ECF) *unit* position vector at the ascending node as follows:

$$\lambda=\tan^{-1}\left(r_{y_{ecf}},r_{x_{ecf}}\right)$$

These two position components can be determined from the x and y components of the Earth-centered-inertial (ECI) position vector and the Greenwich apparent sidereal time θ_g , both evaluated at the ascending node, according to

$$r_{x_{ecf}} = r_{x_{eci}} \cos\theta_g + r_{y_{eci}} \sin\theta_g$$

$$r_{y_{ecf}} = r_{y_{eci}} \cos\theta_g - r_{x_{eci}} \sin\theta_g$$

The Greenwich sidereal time is given by $\theta_g = \theta_{g_0} + \omega_e t$ where θ_{g_0} is the Greenwich sidereal time of the initial ascending node crossing, ω_e is the inertial rotation rate of the Earth, and *t* is the time of the (future) ascending node crossing.

While solving for the root of *f(t)*, the algorithm also propagates the orbit to each ascending node crossing by solving the following nonlinear equation

$$g(t) = r_z = 0$$

subject to the mission constraint $v_z > 0$ which ensures that the orbit is ascending as it crosses the node. The solution of this nonlinear equation also uses a Runge-Kutta-Fehlberg 7(8) algorithm imbedded within Brent's one dimensional root-finding method.

The following is a typical user interaction with this script. The initial epoch and osculating orbital elements correspond to the first ascending node crossing. User input is in bold font.

```
                      program repeat2

  < time to repeat ground track - integrated solution >

 please input the calendar date
 (1 <= month <= 12, 1 <= day <= 31, year = all digits!)
 ? 9,5,2013

 please input the Universal Coordinated Time (UTC)
 (0 <= hours <= 24, 0 <= minutes <= 60, 0 <= seconds <= 60)
 ? 10,20,30

 please input the semimajor axis (kilometers)
 (semimajor axis > 0)
 ? 7200.439089

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

 please input the orbital inclination (degrees)
 (0 <= inclination <= 180)
 ? 108

 please input the right ascension of the ascending node (degrees)
 (0 <= raan <= 360)
 ? 200
```

```
please input the degree of the Earth gravity model (zonals)
(2 <= zonals <= 18)
? 8

please input the order of the Earth gravity model (tesserals)
(0 <= tesserals <= 18)
? 8

would you like to include the point-mass gravity of the Sun (y = yes, n = no)
? y


please input the closure tolerance (degrees)
(a value between 0.1 and 0.5 degrees is recommended)
? 0.1
```

The output created by the script for this example is as follows:

```
              program repeat2

< time to repeat ground track - integrated solution >

semimajor axis              7200.439089  kilometers

eccentricity                  0.00000000

inclination                 108.000000  degrees

argument of perigee           0.000000  degrees

raan                        200.000000  degrees

true anomaly                  0.000000  degrees

Keplerian period            101.344037  minutes

average nodal period        101.250333  minutes

final delta-longitude         0.000001  degrees

solar days to repeat         19.054750  days

number of orbits to repeat      271

degree of gravity model         8

order of gravity model          8

simulation includes the point-mass gravity of the Sun
```

In this screen display, `average nodal period` is the total simulation time divided by the user-defined `number of orbits to repeat`, and `solar days to repeat` is the total simulation time, in seconds, divided by 86400 seconds per solar day. The entry `final delta-longitude` is the difference between the initial east longitude of the ascending node and the final east longitude of the ascending node predicted by the MATLAB script.

Modeling the orbital motion

The `repeat2` MATLAB script implements a *special perturbation* technique which solves the vector system of second-order, nonlinear differential equations of orbital motion given by

$$\mathbf{a}(\mathbf{r},t)=\ddot{\mathbf{r}}(\mathbf{r},t)=\mathbf{a}_g(\mathbf{r},t)+\mathbf{a}_s(\mathbf{r},t)$$

where

$t =$ dynamical time

$\mathbf{r} =$ inertial position vector

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

$\mathbf{a}_s =$ acceleration due to the Sun

Geocentric acceleration due to non-spherical Earth gravity

The `sunsync3` MATLAB script implements 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, λ is the geocentric east longitude and $r=|\vec{r}|=\sqrt{x^2+y^2+z^2}$ is the geocentric distance. In this expression the *S's* and *C's* are 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 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 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

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

$\phi =$ geocentric declination $= \sin^{-1}\left(r_z / r\right)$

$\lambda =$ longitude $= \alpha - \alpha_g$

$\alpha =$ right ascension $= \tan^{-1}\left(r_y / r_z\right)$

$\alpha_g =$ right ascension of Greenwich

The right ascension is measure positive east of the vernal equinox, longitude is measured positive east of Greenwich, and declination 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 $n > m \ne 0$ the coefficients are called *tesseral* terms.

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

$$P_n^0\left(\sin\phi\right) = \frac{1}{n}\left[\left(2n-1\right)\sin\phi\, P_{n-1}^0\left(\sin\phi\right) - \left(n-1\right)P_{n-2}^0\left(\sin\phi\right)\right]$$

$$P_n^n\left(\sin\phi\right) = \left(2n-1\right)\cos\phi\, P_{n-1}^{n-1}\left(\sin\phi\right), \qquad m \ne 0,\ m < n$$

$$P_n^m\left(\sin\phi\right) = P_{n-2}^m\left(\sin\phi\right) + \left(2n-1\right)\cos\phi\, P_{n-1}^{m-1}\left(\sin\phi\right), \qquad m \ne 0,\ m = n$$

where the first few associated Legendre functions are given by

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

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

The trigonometric arguments are determined from expansions given by

$$\sin m\lambda = 2\cos\lambda\sin(m-1)\lambda - \sin(m-2)\lambda$$
$$\cos m\lambda = 2\cos\lambda\cos(m-1)\lambda - \cos(m-2)\lambda$$
$$m\tan\phi = (m-1)\tan\phi + \tan\phi$$

These gravity model data files are simple space delimited ASCII data files. The following is a portion of a typical gravity model data file. In this file, column one is the degree index, column two is the model order index, and columns three and four are the corresponding *un-normalized* gravity coefficients (zonals and tesserals, respectively).

```
 2     0    -1.08262668355E-003    0.00000000000E+000
 3     0     2.53265648533E-006    0.00000000000E+000
 4     0     1.61962159137E-006    0.00000000000E+000
 5     0     2.27296082869E-007    0.00000000000E+000
 6     0    -5.40681239107E-007    0.00000000000E+000
 7     0     3.52359908418E-007    0.00000000000E+000
 8     0     2.04799466985E-007    0.00000000000E+000
 9     0     1.20616967365E-007    0.00000000000E+000
10     0     2.41145438626E-007    0.00000000000E+000
11     0    -2.44402148325E-007    0.00000000000E+000
12     0     1.88626318279E-007    0.00000000000E+000
```

Gravity model coefficients are often published in *normalized* form. The relationship between normalized $\bar{C}_{l,m}, \bar{S}_{l,m}$ and un-normalized gravity coefficients $C_{l,m}, S_{l,m}$ is given by the following expression:

$$\begin{Bmatrix} \bar{C}_{l,m} \\ \bar{S}_{l,m} \end{Bmatrix} = \left[\frac{1}{(2-\delta_{m0})(2l+1)} \frac{(l+m)!}{(l-m)!} \right]^{1/2} \begin{Bmatrix} C_{l,m} \\ S_{l,m} \end{Bmatrix}$$

where δ_{m0} is equal to 1 if *m* is zero and equal to zero if *m* is greater than zero.

This MATLAB script is "hard-wired" to use an Earth gravity model data file named `egm96.dat`. The user can use a different data file by editing the following line of the source code.

```
% read gravity model coefficients

[ccoef, scoef] = readgm('egm96.dat');
```

Geocentric acceleration due to the point-mass gravity of the Sun

The acceleration contribution of the Sun represented by a *point mass* is given by

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

where

μ_s = gravitational constant of the Sun

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

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

Order reduction

The first-order system of equations required by this computer program can be created from the second-order system by the method of *order reduction*. With the following definitions,

$$
\begin{array}{lll}
y_1 = r_x & y_2 = r_y & y_3 = r_z \\
y_4 = v_x & y_5 = v_y & y_6 = v_z
\end{array}
$$

where v_x, v_y, v_z are the velocity vector components, the first-order system of differential equations is given by

$$\dot{y}_1 = v_x \qquad \dot{y}_2 = v_y \qquad \dot{y}_3 = v_z$$

$$\dot{y}_4 = a_{x-e} + a_{x-s} \qquad \dot{y}_5 = a_{y-e} + a_{y-s} \qquad \dot{y}_6 = a_{z-e} + a_{z-s}$$

In these equations, a_{x-e}, a_{y-e} and a_{z-e} are the *x*, *y* and *z* gravitational contributions due to non-spherical Earth gravity, and a_{x-s}, a_{y-s} and a_{z-s} are the *x*, *y* and *z* gravitational contributions of the point-mass gravity of the Sun.

repeat3.m – required mean semimajor axis - Wagner's algorithm

This MATLAB script calculates the mean semimajor axis required for a repeating ground track orbit using an algorithm devised by Carl Wagner. This iterative numerical method is described in "A Prograde Geosat Exact Repeat Mission?", *The Journal of the Astronautical Sciences*, Vol. 39, No. 3, July-September 1991, pp. 313-326.

This algorithm starts with the following initial guess for the required mean semimajor axis

$$a_0 = \mu^{1/3}\left[\left(\frac{R}{D}\right)\omega_e\right]^{-2/3}$$

and iteratively improves this guess with the following update:

$$a_{i+1} = \mu^{1/3}\left[\left(\frac{R}{D}\right)\omega_e\right]^{-2/3}\left[1-\frac{3}{2}J_2\left(\frac{r_e}{a_i}\right)^2\left(1-\frac{3}{2}\sin^2 i\right)\right]^{2/3}\left[1+J_2\left(\frac{r_e}{a_i}\right)^2\left\{\frac{3}{2}\left(\frac{R}{D}\right)\cos i-\frac{3}{4}\left(5\cos^2 i-1\right)\right\}\right]^{2/3}$$

where

$R =$ integer number of orbits

$D =$ integer number of synodic (or nodal) days

$J_2 =$ second gravity coefficient

$\omega_e =$ inertial rotation rate of the Earth

$r_{eq} =$ equatorial radius of the Earth

$i =$ orbital inclination

$\mu =$ gravitational constant of the Earth

The equation for the length of the *nodal* day is given by

$$T_N = \frac{2\pi}{\omega_e - \dot{\Omega}}$$

where $\dot{\Omega}$ is the perturbation of the right ascension of the ascending node. The relationship between R and D is as follows:

$$\frac{R}{D} = \frac{\tilde{n} + \dot{\omega}}{\omega_e - \dot{\Omega}}$$

where $\tilde{n}$ is the perturbed mean motion and $\dot{\omega}$ is the perturbation in argument of perigee. Both perturbation calculations are based on Kozai's mean orbit theory.

The following is a typical user interaction with this script. Please note that the semimajor axis, orbital eccentricity and inclination are mean orbital elements. User input is in bold font.

```
                  program repeat3

    < repeating ground track - Wagner's method >


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

  please input the orbital inclination (degrees)
  (0 <= inclination <= 180)
  ? 108


  please input the number of orbits in the repeat cycle
  ? 271

  please input the number of nodal days in the repeat cycle
  ? 19
```

The following is the solution calculated by this script.

```
                   program repeat3

       < repeating ground track - Wagner's method >

  mean semimajor axis                7192.231056  kilometers

  mean eccentricity                     0.000000
  mean inclination                    108.000000  degrees
  mean argument of perigee              0.000000  degrees
  mean raan                             0.000000  degrees

  number of orbits to repeat          271.000000
  number of solar days to repeat       19.054818

  Keplerian period                    101.170791  minutes
  nodal period                        101.250693  minutes
```

```
length of nodal day                  1444.154622  minutes
fundamental interval                   25.239852  degrees
```

repeat4.m – required osculating semimajor axis - numerical integration solution

This MATLAB script calculates the osculating semimajor axis required for a repeating ground track orbit. The script will ask the user for a semimajor axis initial guess and the orbit is propagated using numerical integration during the solution process. This script is similar to `repeat2` with the time to each ascending node determined by a Runge-Kutta-Fehlberg algorithm imbedded within Brent's root-finding method. The orbital equations of motion are the same as those documented in the `repeat2` technical discussion.

The bracketing interval for Brent's method, in kilometers, for the required osculating semimajor axis a is equal to

$$a_0 - 100 \le a \le a_0 + 100$$

where a_0 is the initial semimajor axis guess provided by the user. The script also sets the initial true anomaly to the ascending node $(\theta = -\omega)$.

The main nonlinear equation solved by this script is given by

$$f(t) = \lambda_{an}^{0} - \lambda_{an}^{p} = 0$$

where λ_{an}^{0} is the initial east longitude of the nodal crossing and λ_{an}^{p} is the ascending node east longitude predicted by the software. By solving the *f(t)* equation, we are trying to minimize the difference between the initial east longitude of the ascending node and the ascending node east longitude predicted by the software for the user-defined number of integer orbits in the repeat cycle.

The east longitude of the ascending node can be determined from the x and y components of the Earth-centered-fixed (ECF) *unit* position vector at the ascending node as follows:

$$\lambda = \tan^{-1}\left(r_{y_{ecf}}, r_{x_{ecf}}\right)$$

These two position components can be determined from the x and y components of the Earth-centered-inertial (ECI) position vector and the Greenwich apparent sidereal time θ_g, both evaluated at the ascending node, according to

$$r_{x_{ecf}} = r_{x_{eci}} \cos\theta_g + r_{y_{eci}} \sin\theta_g$$

$$r_{y_{ecf}} = r_{y_{eci}} \cos\theta_g - r_{x_{eci}} \sin\theta_g$$

The Greenwich sidereal time is given by $\theta_g = \theta_{g_0} + \omega_e t$ where θ_{g_0} is the Greenwich sidereal time of the initial ascending node crossing, ω_e is the inertial rotation rate of the Earth, and t is the time of the (future) ascending node crossing.

While solving for the root of *f(t)*, the algorithm also propagates the orbit to the next ascending node crossing by solving the following nonlinear equation

$$g(t) = r_z = 0$$

subject to the mission constraint $v_z > 0$ which ensures that the nodal crossing is ascending. The solution of this nonlinear equation also uses a Runge-Kutta-Fehlberg 7(8) algorithm imbedded within Brent's one dimensional root-finding method.

The following is an example user interaction with this script. The initial epoch and osculating orbital elements correspond to the first ascending node crossing. User input is in bold font.

```
                        program repeat4

< required osculating semimajor axis - integrated solution >

please input the calendar date
(1 <= month <= 12, 1 <= day <= 31, year = all digits!)
? 9,5,2013

please input the Universal Coordinated Time (UTC)
(0 <= hours <= 24, 0 <= minutes <= 60, 0 <= seconds <= 60)
? 10,20,30

please input the semimajor axis (kilometers)
(semimajor axis > 0)
? 7192

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

please input the orbital inclination (degrees)
(0 <= inclination <= 180)
? 108

please input the right ascension of the ascending node (degrees)
(0 <= raan <= 360)
? 200


please input the number of integer orbits in the repeat cycle
? 271


please input the degree of the Earth gravity model (zonals)
(2 <= zonals <= 18)
? 8

please input the order of the Earth gravity model (tesserals)
(0 <= tesserals <= 18)
? 8

would you like to include the point-mass gravity of the Sun (y = yes, n = no)
? y
```

The following is the screen output created by this MATLAB script for this repeating ground track orbit design example.

```
                         program repeat4

   < required osculating semimajor axis - integrated solution >

   semimajor axis              7200.439089  kilometers

   eccentricity                 0.00000000

   inclination                  108.000000  degrees

   argument of perigee            0.000000  degrees

   raan                         200.000000  degrees

   true anomaly                   0.000000  degrees

   Keplerian period             101.344037  minutes

   average nodal period         101.250333  minutes

   solar days to repeat          19.054750  days

   number of orbits to repeat      271

   degree of gravity model         8

   order of gravity model          8

   simulation includes the point-mass gravity of the Sun
```

In this screen display, `average nodal period` is the total simulation time divided by the user-defined `number of orbits to repeat`, and `solar days to repeat` is the total simulation time, in seconds, divided by 86400 seconds per solar day.

IMPORTANT NOTE

The `repeat2` and `repeat4` MATLAB scripts are based on an Earth orbit repeating its trajectory at the appropriate ascending node crossing. However, non-spherical Earth gravity effects and the point-mass gravity of the Sun will eventually change the orbital characteristics. Therefore, the Earth relative ground tracks will eventually drift away from the ground track associated with the user-defined initial orbit conditions.

For example, see "Perturbations of Repeating Groundtrack Satellites by Tesseral Harmonics in the Gravitational Potential", by M. P. Francis, G. S. Gedeon and B. C. Douglas, AIAA Journal, Vol. 4, No. 7, July 1966.