

Un satélite geoestacionario orbitando la Tierra

Geosynchronous Equatorial Orbit (GEO)

A centripetal force is a force that makes a body follow a curved path. Its direction is always **orthogonal** to the motion of the body and towards the fixed point of the instantaneous center of curvature of the path. In Newtonian mechanics, gravity provides the centripetal force responsible for astronomical orbits.

$$f_c = G \frac{m_T m_s}{d_s^2}$$

Masas puntuales. La Tierra en el origen (inicialmente). El satélite tiene una masa de 20 toneladas.

$$y = [r; v]$$

$$y' = [v; a]$$

$$\frac{dy}{dt} = f(t, y)$$

Los vectores r y v tienen los datos [Tierra, satélite].

Inicialización: variables globales, posiciones y velocidades iniciales, número de puntos.

```
global G m;  
G = 6.67428E-11;           % m3 kg-1 s-2  
mT = 5.9736E+24;          % kg - masa de la Tierra  
ms = 20000;               % kg - masa del satélite  
m = [mT, ms];  
  
rT = 6378e3;              % radio de la Tierra m  
hs = 35786000;            % altura satélite geoestacionario m  
ds = rT + hs;             % distancia entre centros de masa  
r0 = [ [0;0;0], [ds;0;0] ]; % [rT, rs], rT y rs son vectores columna  
ss = sqrt(G * m(1)/r0(1,2)); % velocidad tangencial (speed) a=ss^2/ds  
v0 = [ [0;0;0], [0;ss;0] ]; % [vT, vs]  
  
ws = ss/ds;              % velocidad angular  
T = (2*pi)/ws;           % en segundos = 1 día  
h = 60;                  % 60 s = 1 minuto  
nv = 4;                  % número de vueltas  
Nk = nv*ceil(T/h);       % número de puntos [r;v] a calcular
```

Ciclo iterativo de cálculo de posiciones. Se utiliza el método de Runge-Kutta 4 para resolver numéricamente la ecuación diferencial vectorial de segundo orden.

```
pos = zeros(3*2,Nk+1);  
pos(:,1) = [r0(:,1);r0(:,2)];  
y = [r0;v0];  
for t=2:Nk+1  
    s1 = f(t,y);
```

```

s2 = f(t+h/2,y+(h/2)*s1);
s3 = f(t+h/2,y+(h/2)*s2);
s4 = f(t+h, y+h*s3);
phi = (s1+2*s2+2*s3+s4)/6;
y = y + phi*h;

r = y(1:3,:);
pos(:,t) = [r(:,1);r(:,2)];
end

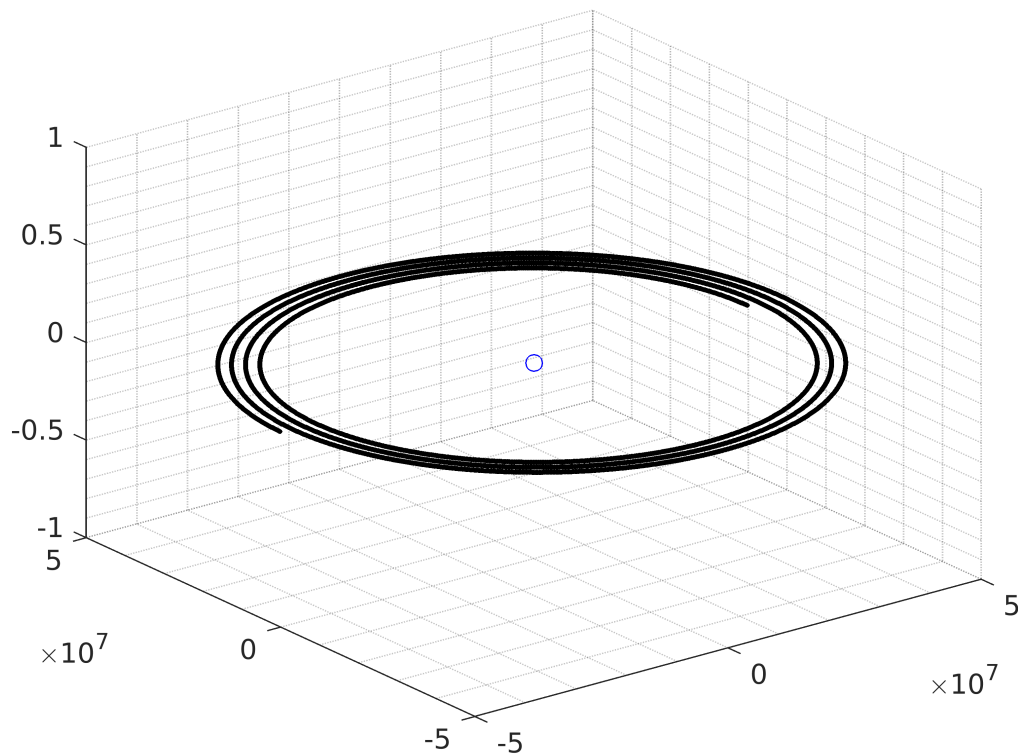
```

Graficación de las posiciones.

```

plot3(pos(1,:),pos(2,:),pos(3,:), 'ob');
hold on;
plot3(pos(4,:),pos(5,:),pos(6,:), '.k');
grid minor;

```



Función que calcula la aceleración a de cada cuerpo de acuerdo a la ley de gravitación universal (forma vectorial).

```

function ydot = f(~,y)
    global G m;
    r = y(1:3,:);
    v = y(4:6,:);
    dr = r(:,2) - r(:,1);
    u = dr / norm(dr);

    % vectores de posición rT y rS
    % dr va de la Tierra al satélite

```

```

dr2 = norm(dr)^2;
F = (G*m(1)*m(2)/dr2) * u;
a = [F/m(1), -F/m(2)]; % par acción - reacción; tercera ley de Newton
ydot = [v;a];
end

```