Objetivo

El objetivo principal es generar un procedimiento para facilitar el diseño /dimensionamiento del sistema de potencia de los paneles solares de un satélite según los detalles de cada misión. Para ello se dividió en tres subobjetivos que definen las tres etapas consecutivas:

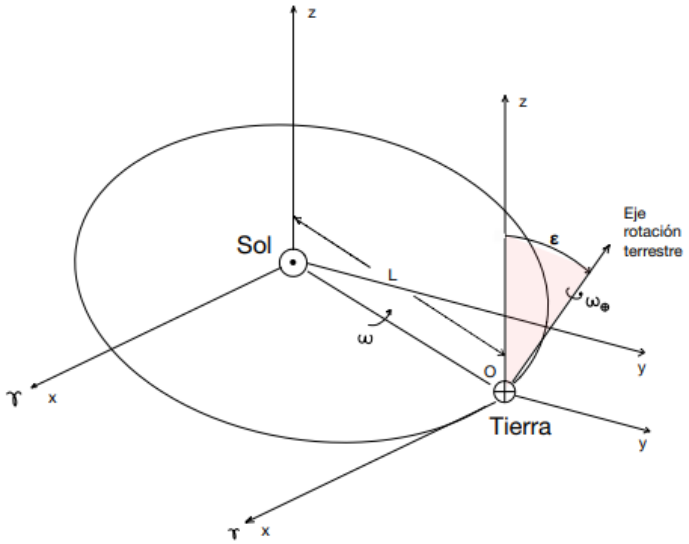
1. Conocer la **mecánica orbital** de la actitud y de la órbita, con las que se define el ángulo de la normal al panel solar y el sol, y así la radiación solar disponible.
2. Lograr simular, según las **propiedades de la celda solar** que se utilizará, para poder definir en BOL (Beginning Of Life) la cantidad de módulos, cadenas y celdas necesarias.
3. Analizar las distintas **degradaciones** que determinarán la eficiencia en EOL (End Of Life) que pueden ocurrir como resultado de las condiciones ambientales de la órbita, la temperatura y las pérdidas de eficiencia intrínsecas a la electrónica de las celdas.

En el presente informe se describe el procedimiento de la primera etapa, donde el objetivo fue simular la órbita y actitud analizada y obtener información sobre el el coseno entre la normal al panel solar y el sol durante la misión.

Mecánica orbital

Sistemas de referencia

Existen distintos sistemas de referencia para la navegación espacial, cada uno adecuado para distintas aplicaciones. Uno de los más importantes es el Heliocéntrico (centrado en el sol) con el plano OXY contenido en el plano de la eclíptica del sol y manteniendo su eje X apuntando siempre al equinoccio vernal (de Primavera), que también recibe el nombre de Primer Punto de Aries y se denomina con el símbolo γ. Este es utilizado principalmente para viajes interplanetarios dentro del Sistema Solar. Para el seguimiento de satélites que orbitan la Tierra el más adecuado es el Earth-centered inertial (ECI). El mismo se determina trasladando el sistema de referencia Heliocéntrico al centro de la Tierra.
De todos los sistemas de referencia existen distintas variantes. En este caso se utiliza el sistema *EarthMJ2000Eq*.



La determinación de la posición de un satélite en el sistema ECI se consigue conociendo 6 parámetros de posición y velocidad que pueden ser dados en distintos sistemas de coordenadas usando las ecuaciones de las órbitas de las leyes de Kepler. Algunos ejemplos de conjuntos de parámetros son:

- las condiciones iniciales en coordenadas cartesianas de posición (X, Y, Z) y velocidad (Vx, Vy, Vz)
- las condiciones iniciales en coordenadas esféricas de posición (Rmag/radio, RA/Ascención recta, DEC/declinación) y velocidad (Vmag/radial, Vy, Vz)

Elements		
RMAG	7218.032973047435	km
RA	0	deg
DEC	10.37584492005105	deg
VMAG	7.417715281675348	km/s
AZI	82.37742168155043	deg
FPA	88.60870365370448	deg

Con estas variables y matemática se pueden obtener los elementos orbitales Keplerianos que son los principales elementos que definen la órbita de un objeto en el sistema solar. Estos son los más utilizados para comunicar la posición de los satélites. Estos elementos incluyen:

- Semi-eje mayor (a, SMA): Es la distancia entre el centro de la órbita y el punto más lejano del objeto.
- Excentricidad (e, ECC): La excentricidad es una medida de la elipse de la órbita, es decir, qué tan ovalada está la órbita.
- Inclinación (i, INC): Es el ángulo entre la órbita de un objeto y el plano de la eclíptica.
- Ascensión recta (Ω, RAAN): Es el ángulo entre el punto vernal y la línea de nodos (nodo ascendente).
- Argumento del perihelio (ω, AOP): Es el ángulo entre el punto vernal y el punto de la órbita más cercano al Sol.
- Anomalía verdadera (ν, TA): fracción de un período orbital que ha transcurrido, expresada como ángulo

Elements		
SMA	7191.938817629013	km
ECC	0.02454974900598137	
INC	12.85008005658097	deg
RAAN	306.6148021947984	deg
AOP	314.1905515359921	deg
TA	99.8877493320488	deg

Por último, para poder definir la trayectoria (posición en función del tiempo) de un satélite, es necesario conocer el *Epoch* correspondiente a los parámetros orbitales. El *Epoch* es la fecha en que el satélite se encuentra en la posición respecto de la Tierra indicada por los parámetros orbitales anteriores. El mismo puede ser expresado en distintos formatos, y cómo el más usado es el *UTCGregorian*, razón por la cual se decidió utilizarla en este proyecto.

Epoch Format	UTCGregorian
Epoch	01 Jan 2000 11:59:28.000
Coordinate System	EarthMJ2000Eq
State Type	Keplerian

Actitud

El sistema de referencia más usado para la ubicación de satélites en LEO es el sistema *Earth-centered inertial* (ECI). Este tiene su origen fijo a la Tierra y su eje X ($\widehat{i}_1$ en la Fig. 6), fijo a la dirección del Equinoccio Vernal.

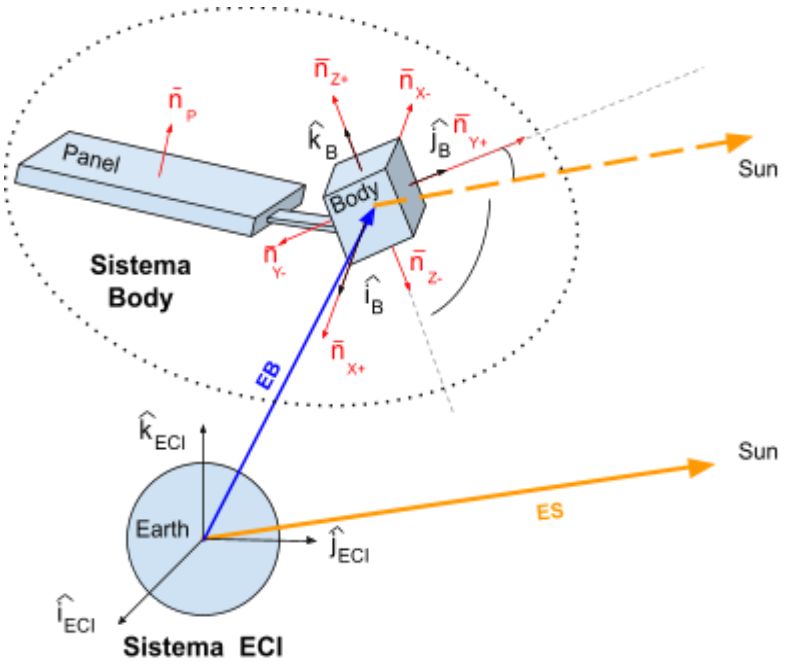


Figura 6. Sistemas de referencia usados en el programa. ECI: Earth-Centered Inertial. B: Body

Luego si los ejes de la base de este sistema de referencia son los siguientes

$$\widehat{i_{ECI}} = [1,\ 0,\ 0]$$
$$\widehat{j_{ECI}} = [0,\ 1,\ 0]$$
$$\widehat{k_{ECI}} = [0,\ 0,\ 1]$$

Usando estos vectores se puede definir la posición del satélite

$$\overline{u_{Sat}} = X_{Sat}.\widehat{i_{ECI}} + Y_{Sat}.\widehat{j_{ECI}} + Z_{Sat}.\widehat{k_{ECI}}$$

donde X_{Sat}, Y_{Sat}, Z_{Sat} son… . El sistema body se define genéricamente para un satélite cúbico o un CubeSat como se ve en la Fig 6. Los ejes de este sistema están fijos al satélite y sus componentes en la base del sistema ECI se obtienen rotando $\widehat{i_{ECI}}$, $\widehat{j_{ECI}}$ y $\widehat{k_{ECI}}$ según la actitud del mismo. La actitud se define en este programa con los ángulos de Euler $(\omega_1, \omega_2, \omega_3)$ correspondientes a la rotación R_{321}

$$R_1(\phi)R_2(\theta)R_3(\psi) = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ -c_\phi s_\psi + s_\phi s_\theta c_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & s_\phi c_\theta \\ s_\phi s_\psi + c_\phi s_\theta c_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix}$$

donde $\phi = \omega_3$ (`Sat.EulerAngle3`), $\theta = \omega_2$ (`Sat.EulerAngle2`) y $\psi = \omega_1$ (`Sat.EulerAngle1`). Luego se obtienen los 3 ejes del sistema body según

$$\widehat{i_B} = \widehat{i_{ECI}}.R_{321} \quad ; \quad \widehat{j_B} = \widehat{j_{ECI}}.R_{321} \quad ; \quad \widehat{k_B} = \widehat{k_{ECI}}.R_{321}$$

Por otro lado, se pueden definir en el sistema body las normales de las caras del satélite y de los paneles solares desplegables de la forma

$$\overline{n_f} = a_{f1}.\widehat{i_B} + a_{f2}.\widehat{j_B} + a_{f3}.\widehat{k_B} \quad \text{con}$$
$$\widehat{i_B} = [1,\ 0,\ 0]$$
$$\widehat{j_B} = [0,\ 1,\ 0]$$
$$\widehat{k_B} = [0,\ 0,\ 1]$$

con $f = X^+, X^-, Y^+, Y^-, Z^+, Z^-, P$ las distintas caras del satélite y a_{fn} siendo $n = 1, 2, 3$, las componentes de las normales a cada cara en el sistema body.

GMAT

- Spacecraft: genero uno y le seteo lo siguiente
    - Orbit
        - Epoch format: UTCGregorian
        - Coordinate System: EarthMJ2000Eq
        - State type: Keplerian
            - SMA: semi mayor axis
            - ECC: eccentricity
            - INC: inclination
            - RAAN
            - AOP
            - TA
    - Attitude
        - Attitude model: NadirPointing
            - Euler angle sequence: (En el reporte entrega los angulos de euler respecto del sistema de coordenadas elegido)
                - Ej: sequence 321 → w1: rotación del eje z, w2: del eje y, w3: del eje x
            - Attitude reference body (Earth, Sun): cuerpo celeste al que quiero que apunte (elegir el nadir)
                - Body alignment vector: vector del eje de coordenadas del satélite que quiero que mire al nadir (cuerpo celeste)
            - Attitude Constraint Type: vector que quiero usar de referencia para mantener fijo a un eje del satélite (normal de la órbita o velocidad del satélite)
                - Body constraint vector: vector del eje de coordenada del satélite que quiero que intente apuntar siempre al vector del constraint type (se minimiza su diferencia)
        - Attitude model: Three Axis Kinematic
            - Euler angle sequence: (En el reporte entrega los angulos de euler respecto del sistema de coordenadas elegido)
                - Ej: sequence 321 → w1: rotación del eje z, w2: del eje y, w3: del eje x
            - Attitude reference body (Earth, Sun): cuerpo celeste al que quiero que apunte (elegir el nadir)
                - Body alignment vector: vector del eje de coordenadas del satélite que quiero que mire al nadir (cuerpo celeste)
            - Attitude Constraint Type: vector que quiero usar de referencia para mantener fijo a un eje del satélite (normal de la órbita o velocidad del satélite)
                - Body constraint vector: vector del eje de coordenada del satélite que quiero que intente apuntar siempre al vector del constraint type (se minimiza su diferencia)
    - Visualization (.3ds): Ingreso el dibujo CAD del satélite en formato .3ds
- Propagators
    - Prop:
        - Force model
        - Drag model
        - Solar radiation

        - Integrator (algoritmo para estimar la trayectoria entre puntos discretos calculados, ej: runge Kutta)
        - Corrección relativista
        - Gravity models (cuerpos a tener en cuenta)
        - Atmosphere model
- Output
    - OrbitView
    - GroundTrack
    - ReportFile: elijo las variables que quiero que imprima en un .txt
    - XYplot
- Event Locators
    - EclipseLocator: Reporta inicio, fin y duración de penumbra antiubra y umbra

Programa de cosenos

De GMAT obtengo un archivo .txt que reporta los siguientes datos

Sat.UTCGregorian	Sat.ElapsedSecs	Sat.Earth.LST	Sat.EulerAngle1	Sat.EulerAngle2	Sat.EulerAngle3					
Sat.EarthMJ2000Eq.X	Sat.EarthMJ2000Eq.Y	Sat.EarthMJ2000Eq.Z	Sat.Earth.Altitude	Sat.Earth.BetaAngle	Sun.EarthMJ2000Eq.X					
Sun.EarthMJ2000Eq.Y	Sun.EarthMJ2000Eq.Z	Sat.EarthMJ2000Eq.AOP	Sat.EarthMJ2000Eq.RAAN		Sat.EarthMJ2000Eq.INC	Sat.Earth.ECC				
Sat.Earth.SMA	Sat.Earth.TA	Sat.Earth.OrbitPeriod								
20 Mar 2011 12:00:00.000	43200	265.97	85.863	29.87	-9.2271	-441.69	-6106.6	3516.3	687.65	-81.691	1.4895e+08
-1.4913e+06	-6.4736e+05	261.7	270.49	97.993	0.0026747	7048.8	128.5	5889.5			
21 Mar 2011 00:00:00.000	86400	95.851	-84.33	30.172	-170.75	-602.94	6072.4	3547.6	685.84	-81.881	1.4898e+08
-3.0663e+05	-1.3378e+05	271.44	270.99	97.992	0.0026546	7048.7	238.06	5889.4			
21 Mar 2011 12:00:00.000	1.296e+05	3.4034	-176.72	-81.996	-91.791	976.86	56.069	-6958.2	669.5	-82.064	1.49e+08
8.7803e+05	3.7982e+05	254.92	271.48	98.001	0.0012309	7035	14.827	5872.3			
22 Mar 2011 00:00:00.000	1.728e+05	267.5	87.395	29.563	-9.1978	-279.18	-6136.6	3484.5	689.45	-82.263	1.4901e+08
2.0627e+06	8.934e+05	255	271.96	97.992	0.0027254	7048.8	134.88	5889.6			

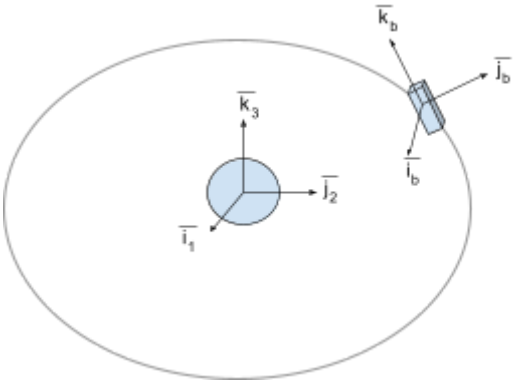
Los ángulos de Euler, Sat.EulerAngle1 (w_1), Sat.EulerAngle2 (w_2) y Sat.EulerAngle3 (w_3) son propagados en la simulación con una secuencia (definida previamente en GMAT) rotando los vectores del origen de coordenadas (también definido en GMAT). Con la secuencia 321, la rotación R_{321} queda:

$$R_1(\phi)R_2(\theta)R_3(\psi) = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ -c_\phi s_\psi + s_\phi s_\theta c_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & s_\phi c_\theta \\ s_\phi s_\psi + c_\phi s_\theta c_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix}$$

```
def rotEuler(x,w3,w2,w1):
    "Rotación 321"
    w1 *= np.pi/180
    w2 *= np.pi/180
    w3 *= np.pi/180
    c1 = mt.cos(w1)
    c2 = mt.cos(w2)
    c3 = mt.cos(w3)
    s1 = mt.sin(w1)
    s2 = mt.sin(w2)
    s3 = mt.sin(w3)
```

```
    R321 = np.array([[c2*c3, c2*s3, -s2], [-c1*s3+s1*s2*c3, c1*c3+s1*s2*s3, s1*c2], [s1*s3+c1*s2*c3,
-s1*c3+c1*s2*s3, c1*c2]])
    return np.matmul(x,R321)
```

donde $\phi = w_3$, $\theta = w_2$ y $\psi = w_3$. Luego para obtener la actitud del satélite roto cada vector unitario del sistema inercial elegido. En el caso del sistema EarthMJ2000Eq los vectores unitarios son:



$$\overline{i_1} = [1,\ 0,\ 0]$$
$$\overline{j_2} = [0,\ 1,\ 0]$$
$$\overline{k_3} = [0,\ 0,\ 1]$$

```
def sist_ref(scale, ax1 = 1,  graph = False, graph_equinox = False, color = "grey"):
    "Vectores unitarios positivos y negativos → Sistema de referencia ECI: x = Vernal equinox"
    i_s = np.array([[0,0,0],[1,0,0]]) * scale
    j_s = np.array([[0,0,0],[0,1,0]]) * scale
    k_s = np.array([[0,0,0],[0,0,1]]) * scale
    negi_s = np.array([[0,0,0],[-1,0,0]]) * scale
    negj_s = np.array([[0,0,0],[0,-1,0]]) * scale
    negk_s = np.array([[0,0,0],[0,0,-1]]) * scale
    if graph_equinox == True:
        i_sun = cp.copy(i_s)*15
        ax1.plot_wireframe(np.array([i_sun[:,0]]), np.array([i_sun[:,1]]), np.array([i_sun[:,2]]), color =
"yellow")
    if graph == True:
        for l in [i_s,j_s,k_s]:
            ax1.plot_wireframe(np.array([l[:,0]]), np.array([l[:,1]]), np.array([l[:,2]]), color = color)
    return i_s, j_s, k_s, negi_s, negj_s, negk_s
```

que al rotarlos se tienen los vectores unitarios fijos al cuerpo con la actitud del Epoch correspondiente

$$\overline{i}\,.\,R_{321} = \overline{i_b} = [i_{b1},\ i_{b2},\ i_{b3}]$$
$$\overline{j}\,.\,R_{321} = \overline{j_b} = [j_{b1},\ j_{b2},\ j_{b3}]$$
$$\overline{k}\,.\,R_{321} = \overline{k_b} = [k_{b1},\ k_{b2},\ k_{b3}]$$

```
def get_axbody(g, l, att, orb, ax1 = 1, graph = False, ax_colors = ["lightblue", "blue", "darkblue"]):
    "l: vector unitario a modificar, att: vector de angulos de Euler, orb = vector de posicion en la
orbita"
    ax_body = np.array([l[0] , rotEuler(l[1]-l[0], *att)])  # Rotación body según actitud en orbit[m]
    ax_body_rot = ax_body[1]
    if graph == True:                                   # Grafico el sistema de coordenadas rotado
según la attitude (solo los positivos g<3)
```

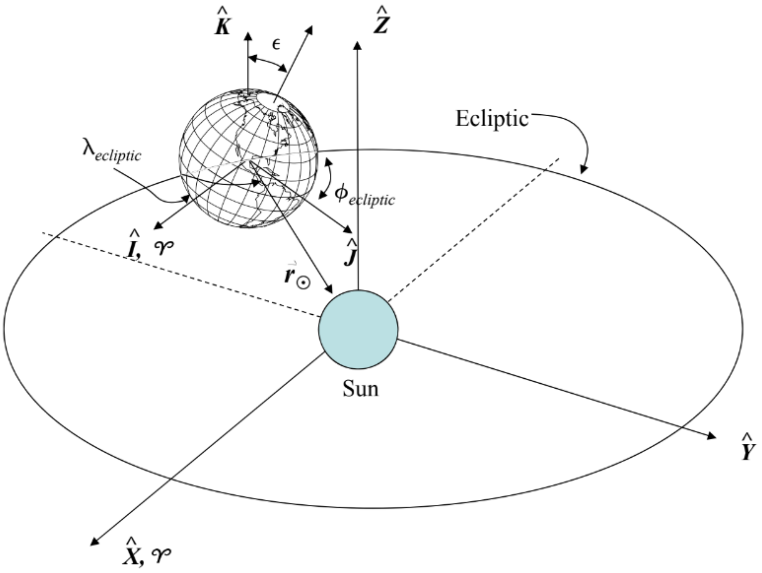
```
        ax_body = np.add(ax_body, orb)                          # Translación body según orbita
        if g>5:
            ax1.plot_wireframe(np.array([ax_body[:,0]]), np.array([ax_body[:,1]]),
np.array([ax_body[:,2]]), color = ax_colors)
        elif g<3:
            ax1.plot_wireframe(np.array([ax_body[:,0]]), np.array([ax_body[:,1]]),
np.array([ax_body[:,2]]), color = ax_colors[g])
    return ax_body, ax_body_rot
```

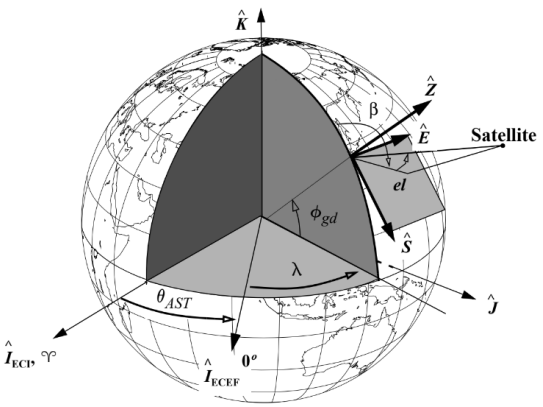
con los cuales se genera la matriz actitud $\overline{A}$

$$\overline{A} = \begin{matrix}[i_{b1},\ i_{b2},\ i_{b3}] \\ [j_{b1},\ j_{b2},\ j_{b3}] \\ [k_{b1},\ k_{b2},\ k_{b3}]\end{matrix}$$

Ángulo de incidencia de la radiación del sol

Calculo el vector Tierra - Sol para cada instante en cada punto de la órbita como lo indica en el capítulo Fundamentals of astrodynamics and applications - David Vallado - Sec 5.1.1.





ϕ = latitude

λ = longitude

1) Julian Date: Date = 01 Jun 2000 11:59:00.000 → yr, mo, d, h, min, s →

$$JD = 367(yr) - \text{INT}\left\{\frac{7\left\{yr + \text{INT}\left(\frac{mo+9}{12}\right)\right\}}{4}\right\} + \text{INT}\left(\frac{275\,mo}{9}\right) + d + 1{,}721{,}013.5 + \frac{\frac{\left(\frac{s}{60*} + min\right)}{60} + h}{24}$$

2) Number of Julian Centuries $$T_{UT1} = \frac{JD_{UT1} - 2{,}451{,}545.0}{36{,}525}$$

```
def time_params(UTC):
    "INPUT example string: 21 Mar 2000 11:59:28.000"
x = datetime.strptime(fecha, '%d %b %Y %H:%M:%S.%f') year = x.year mon = x.month day = x.day hr = x.hour
min = x.minute secs = x.second return year, mon, day, hr, min, secs
    if sec==60:
        sec=59
    return yr, mon, day, hs, min, sec
```

3) Mean Longitude of the Sun $$\lambda_{M_\odot} = 280.460\,° + 36{,}000.771\;T_{UT1}$$ Approximation:

$$\text{LET } T_{TDB} \cong T_{UT1}$$

4) Mean Anomaly for the Sun $$M_\odot = 357.529\,109\,2° + 35{,}999.050\,34\,T_{TDB}$$

5) Mean Logitude of the ecleptic

$$\lambda_{ecliptic} = \lambda_{M_\odot} + 1.914\,666\,471°\,\text{SIN}(M_\odot) + 0.019\,994\,643\,\text{SIN}(2M_\odot)$$

6) Distance of the Earth to the Sun (Unidades atronómicas)

$$r_\odot = 1.000\,140\,612 - 0.016\,708\,617\,\text{COS}(M_\odot) - 0.000\,139\,589\,\text{COS}(2M_\odot)$$

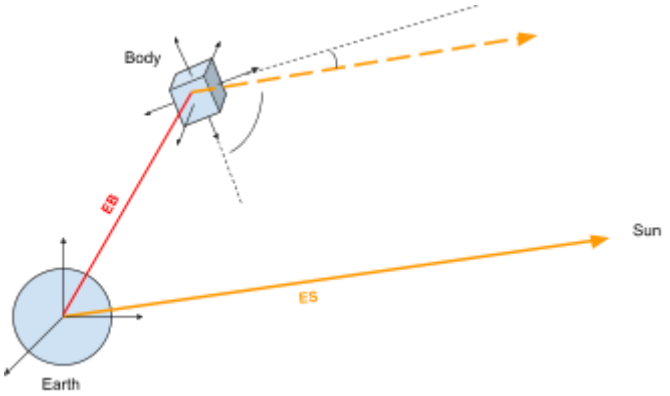
7) Obliquity of the ecliptic $$\epsilon = 23.439\,291° - 0.013\,004\,2\,T_{TDB}$$

$$\vec{r}_\odot = \begin{bmatrix} r_\odot\,\text{COS}(\lambda_{ecliptic}) \\ r_\odot\,\text{COS}(\epsilon)\,\text{SIN}(\lambda_{ecliptic}) \\ r_\odot\,\text{SIN}(\epsilon)\,\text{SIN}(\lambda_{ecliptic}) \end{bmatrix} \text{AU}$$

Se calcula el coseno entre el vector tierra sol con cada uno de los vectores unitarios fijos al cuerpo en dirección positiva $\overline{i_b}, \overline{j_b}, \overline{k_b}$ y en dirección negativa $\overline{i\,*_b}, \overline{j\,*_b}, \overline{k\,*_b}$ usando la ecuación:

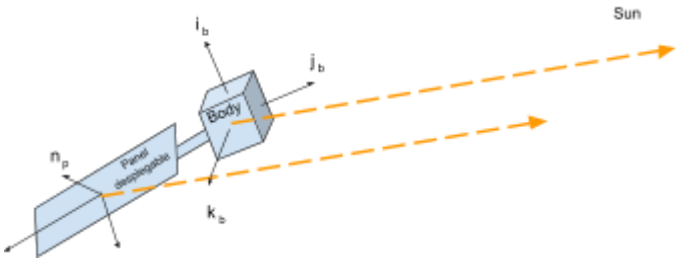
$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}||\vec{b}|}$$

donde $\overline{a}$ es un vector unitario fijo al cuerpo y $\overline{b}$ el vector al entre la Tierra y el Sol.
En este cálculo se está tomando la aproximación de que el Sol está muy lejos y el ángulo entre el vector Earth-Sun y Body-Sun, son paralelos.
(EB ~ 0)



Paneles solares desplegables:

Si los paneles solares no están en las caras del satélite sino que se adicionan como paneles desplegables con normales $\overline{n_p}$ en coordenadas del sistema body, hay dos opciones:



1) Tratar a cada $\overline{n_p}$ como a un vector unitario de sistema body. Es decir, rotarlo con los ángulos de Euler según el sistema de actitud y calcular su ángulo con el vector Sol $\overline{S}$.

2) Con los cosenos resultantes entre ($\overline{i_b}$, $\overline{j_b}$, $\overline{k_b}$) y $\overline{S}$, que es el output del código anterior, se pueden obtener los cosenos entre el $\overline{n_p}$ y $\overline{S}$ calculando la matriz de cosenos directores entre el sistema body y $\overline{n_p}$.
Los ángulos entre el sistema body y el panel (bp) son

$$\alpha_{bp} = arcsin\left(\frac{\overline{i_b}\cdot\overline{n_p}}{|\overline{i_b}|\,|\overline{n_p}|}\right);\ \beta_{bp} = arcsin\left(\frac{\overline{j_b}\cdot\overline{n_p}}{|\overline{j_b}|\,|\overline{n_p}|}\right);\ \gamma_{bp} = arcsin\left(\frac{\overline{k_b}\cdot\overline{n_p}}{|\overline{k_b}|\,|\overline{n_p}|}\right)$$

a los cuales, al sumarles el ángulo entre ($\overline{i_b}$, $\overline{j_b}$, $\overline{k_b}$) y $\overline{S}$ (bs), se obtienen los ángulos buscados, sus respectivos cosenos. Eso es, para los cosenos del eje $\overline{i_b}$

$$\alpha_{bp} + \alpha_{bs} = \alpha_{ps} \rightarrow cos\left(\alpha_{ps}\right)$$

y análogamente para $\overline{j_b}$ y $\overline{k_b}$.