

Manual de uso del programa de cálculo de ángulos de incidencia solar en paneles solares espaciales- SolarAng1

1. Objetivo	1
2. Alcance	1
3. Descripción de Programa	2
a. "Simulación larga": Caso duration_run > 1 día	2
i. Simulación 1: un año completo	2
ii. Simulación 2: día que contiene al eclipse más largo	3
iii. Periodo WC: la órbita completa que contiene al peor caso	3
b. "Simulación Corta": Caso duration_run < 1 día	3
i. Simulación	3
ii. Periodo	4
4. Requerimientos para utilizar SolarAng1:	4
a. Python 3.7	4
b. GMAT	5
c. SolarAng1	9
5. Programa SolarAng1	9
a. solar_ang.py (main)	10
b. gmat_set_orbit.py (paquete/librería para el control de GMAT)	11
c. orbital_mec.py (paquete/librería para la mecánica orbital)	11
6. Uso del programa	12
a. Paso a paso	12
b. Notificaciones específicas de cada simulación	15
c. Errores posibles	17
7. Resultados obtenidos	19
8. Apéndice	22
a. Apéndice A: Definiciones de mecánica orbital	22

1. Objetivo

El presente documento describe el programa que sirve para conocer el ángulo de incidencia del sol (o más precisamente, el coseno de este) sobre los paneles solares para una dada órbita y actitud del satélite de interés.

2. Alcance

Este documento alcanza a todos los posibles usuarios del programa SolarAng1

3. Descripción de Programa

El programa de cálculo de ángulos y cosenos permite conocer el coseno entre la normal a las caras del satélite (o un plano referido a este) y el sol para distintas órbitas y así poder diseñar y dimensionar el sistema de potencia.

Para resolver este problema de mecánica orbital se utiliza el programa libre de la NASA: General Mission Analysis Tool (GMAT), el cual entre otras cosas, puede simular la trayectoria de la órbita y actitud deseadas. El mismo se controla utilizando su api para Python 3.7.

El programa SolarAng1 consiste en una serie de scripts de Python y un script de GMAT. En el script de GMAT se definen los parámetros de la órbita y la actitud del satélite que se desea simular. El mismo tiene una configuración de modelos de mecánica orbital (gravedad, drag, etc.) predefinida, a los que consideraremos como “opciones avanzadas” ya que no son relevantes para el cálculo de potencia. Además, en él, también se configuran los reportes con los resultados de la simulación. Las características de la órbita y actitud que hay que definir para cada simulación (inputs) se muestran a la derecha y se explican en la Tabla 1. Por otro lado, los scripts de Python consisten en dos paquetes o librerías con funciones definidas para utilizar en un tercer script de Python que las ejecuta. Este último contiene las rutinas del programa.

El proceso que lleva a cabo la rutina del programa tiene el fin de simular la órbita deseada con GMAT para un período de tiempo con cierto paso definidos por el usuario (se recomienda 1 año para un análisis completo de la órbita, y posibles eclipses, con un paso de 12 horas). Luego define el eclipse más largo del año (el peor caso en cuanto a generación energética) y realiza otra simulación con GMAT, con un paso más corto (30 segundos) que contenga a dicho eclipse, para conocer la variación de los ángulos con más detalle.

Para ambas simulaciones el programa grafica los ángulos, sus cosenos y realiza una ilustración de la orientación del satélite en algunos puntos del espacio.

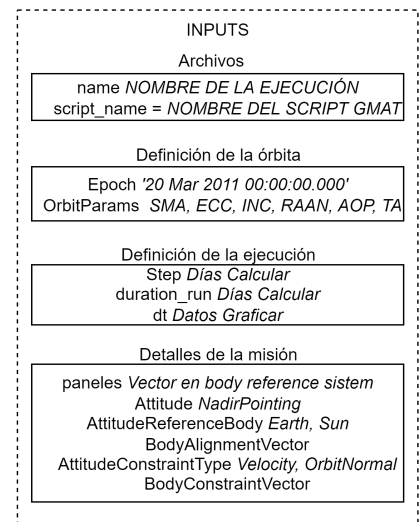
Los diferentes procesos del programa difieren dependiendo de la duración de la simulación principal. Para ejecuciones con `duration_run < 1 día`, el programa procede como se muestra a la izquierda de la Fig. 1. Para ejecuciones con `duration_run > 1 día`, el programa procede como se muestra a la derecha.

a. “Simulación larga”: Caso `duration_run > 1 día`

A continuación se describe el proceso para el caso `duration_run > 1 día`; específicamente para el caso del análisis completo de una órbita, es decir, el análisis de un año.

i. Simulación 1: un año completo

En primer lugar realiza una simulación de 1 año (más 2 días para visualizar la periodicidad anual, es decir, que empiece y termine en las mismas condiciones) con un intervalo de tiempo relativamente grande ($\text{step} = 0.5$ días) para reducir la duración de la corrida del programa. Se estudia solo un año ya que se consideran despreciables las variaciones año a año en comparación con el rango de estudio (máximo decenas de años).



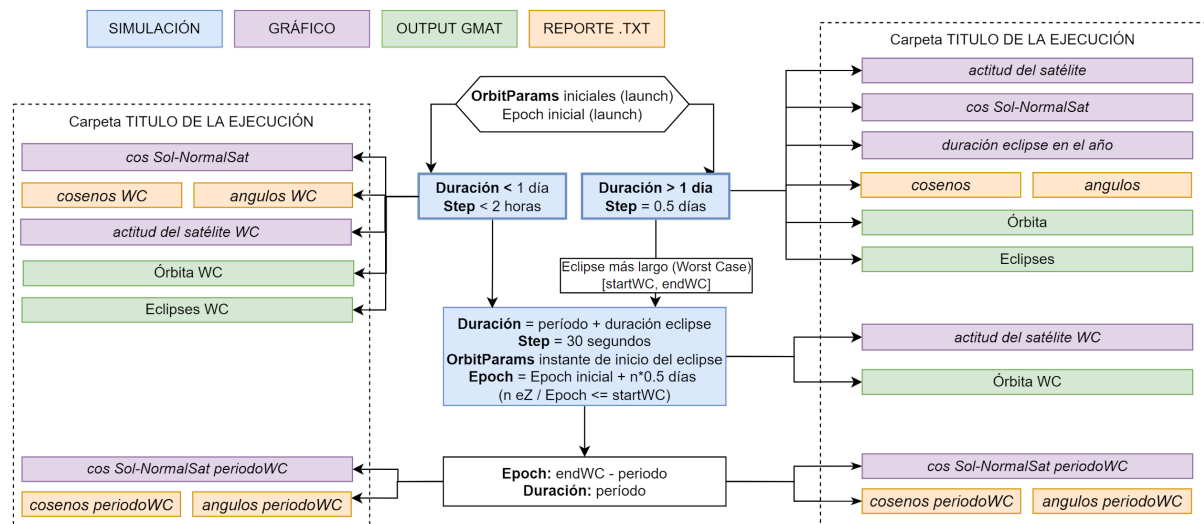


Figura 1. Proceso del programa. A la izquierda, los inputs o variables a definir inicialmente. En el centro, los 3 procesos principales indicando a su derecha los outputs de cada parte.

En la Fig. 1 se muestran los archivos generados de esta primera simulación: por un lado los archivos de texto plano “Órbita” con datos de la posición del satélite, la órbita y el sol durante la simulación y “Eclipse” con los datos de los eclipses que ocurren durante el período de la simulación que son reportes generados por GMAT. Por otro lado, en una subcarpeta llamada *angulos*, un archivo de texto plano con los ángulos y otro con los cosenos calculados, el gráfico ilustrativo de la actitud del satélite, los ángulos y cosenos en función de la fecha del año y la duración de cada eclipse en función de la fecha del año.

ii. Simulación 2: día que contiene al eclipse más largo

El siguiente paso del programa consiste en definir el Peor Caso (PC) o Worst Case (WC), es decir la órbita con el eclipse más largo y hacer otra simulación que incluya el período de dicho eclipse, pero con un menor intervalo de tiempo entre cada toma de datos (step = 30 segundos). Esto es importante ya que el sistema de potencia debe dimensionarse tal que pueda abastecer la misión durante el peor período (de menor luz solar).

En la Fig. 1 se muestran también los archivos generados con los resultados de la simulación: “Órbita” y “Eclipse” generados por GMAT, el gráfico ilustrativo de la actitud y el reporte de los ángulos y cosenos en texto plano.

iii. Periodo WC: la órbita completa que contiene al peor caso

Dado el eclipse más largo, selecciona el período que lo contiene dentro del reporte de angulos y cosenos de la simulación 2 y lo grafica. El Epoch de inicio de este período se elige tal que el eclipse quede al final. En la Fig. 1 se muestran los archivos generados para esta selección de datos que incluye un gráfico de los cosenos y un reporte .txt de ellos.

b. “Simulación Corta”: Caso $duration_run < 1$ día

i. Simulación

La simulación, en este caso, tiene una duración suficientemente corta como para elegir un step menor al orden del período de la órbita manteniendo un tiempo de ejecución aceptable. Debido a esto, no es necesario realizar una segunda simulación. En la Fig. 1 se muestran los archivos generados en esta parte: un gráfico de los ángulos y cosenos de todo el período de tiempo simulado, dos archivos de texto plano para guardar los cosenos y los ángulos y los archivos “Órbita WC” y “Eclipse WC” (en este caso se los refiere como WC solo por corresponderse con una simulación corta).

ii. Periodo

Idem punto 3 del caso `duration_run > 1` día.

4. Requerimientos para utilizar SolarAng1:

Para utilizar este software se necesitan dos programas: GMAT y Spyder (IDE de Python). El orden de instalación es el sugerido a continuación. Sin embargo, la primera parte de la instalación de GMAT es independiente de Python, a diferencia de la segunda.

a. Python 3.7

Para poder controlar GMAT con código, es necesario trabajar en Python 3.7 (un requerimiento de dicho programa).

A continuación se detallan los pasos necesarios para la instalación de este programa en Windows 10:

i. Descarga de Miniconda [Miniconda — conda documentation](#)

1. Descargar la versión del instalador de 64 bits.

Latest Miniconda Installer Links

Latest - Conda 22.11.1 Python 3.10.8 released December 22, 2022

Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	2e3886630fa3fae7636432a954be530c88d0705fce497120d56e0f5d865b0d51
	Miniconda3 Windows 32-bit	4fb64e6c9c28b88beab16994bfba4829110ea3145baa60bda5344174ab65d462

2. Al ejecutar el instalador, seguir los pasos.

ii. Crear un *environnement* con la versión de Python que se necesita (además siempre es recomendable trabajar en un *environment base* (ambiente principal, el que viene por default).

1. Buscar el prompt de Anaconda en el buscador de windows (“Anaconda Prompt (miniconda3)”)

2. Ingresar los siguientes comandos:

```
conda create -n py307 python=3.7
conda activate py307
```

3. En paréntesis al inicio de la línea deberá figurar “(py307)” como se muestra en la siguiente imagen:

```
(base) C:\Users\Karen>conda activate py307
(py307) C:\Users\Karen>
```

iii. Dentro del nuevo *environnement* hay que instalar el editor y las librerías:

```
conda install spyder numpy matplotlib pandas scipy
```

Advertencia 1: este paso puede tardar y al final muestra todas las descargas completadas y “(more hidden)” por lo que no se va a poder seguir el avance de la descarga en su totalidad. Si tarda demasiado se puede optar por instalar de a un paquete con el comando `conda install <nombre_paquete>` donde `nombre_paquete` es el nombre del paquete que se desea instalar.

iv. Cerrar la terminal del prompt de Anaconda

v. Abrir el editor Spyder (buscar en el Inicio de Windows)

Con estos pasos se puede empezar a programar con Python en Spyder (editor IDE).

b. GMAT

El programa utiliza GMAT para simular las trayectorias del satélite, su actitud y la posición del sol. Para su instalación y configuración se deben seguir los siguientes pasos:

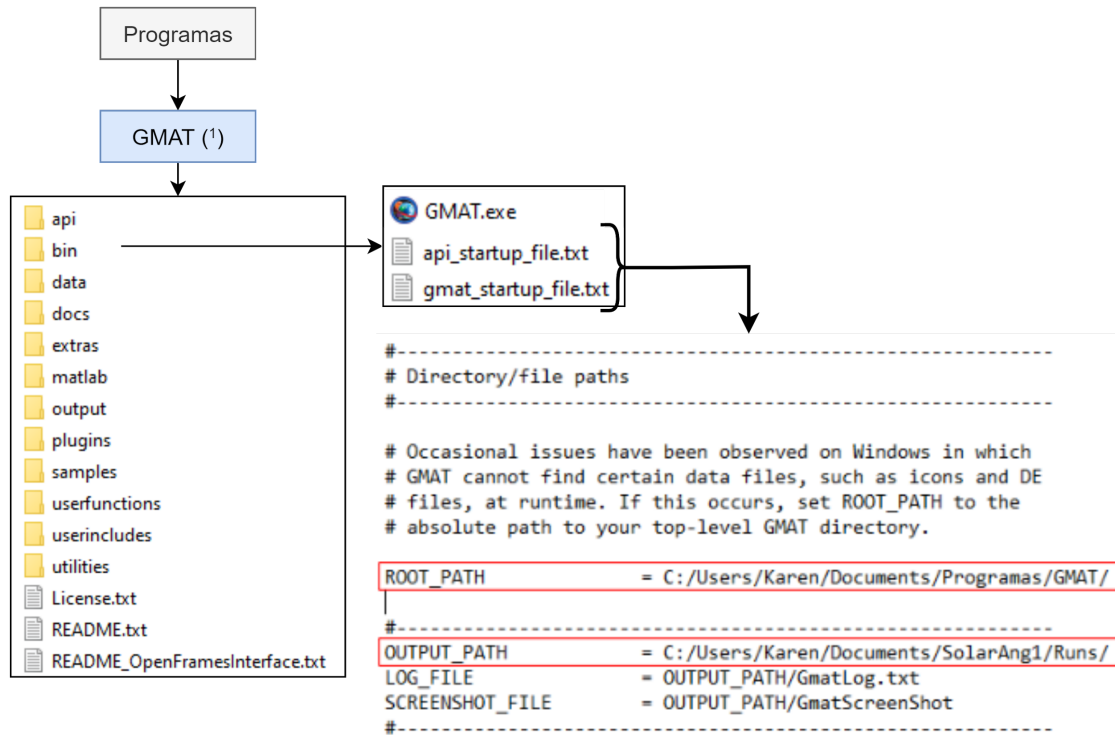


Figura 2. Distribución de carpetas final, luego de la instalación y con la configuración de los path del programa principal y el api.

i. Primera parte:

1. Descargar GMAT desde la página [GMAT download | SourceForge.net](#).
2. Descomprimir la carpeta gmat-win-R2020a.zip, que contiene las carpetas y archivos mostrados en la Fig. 1.
3. Crear una carpeta llamada Runs donde se guardaran todos los outputs de las simulaciones. Se recomienda ubicarla dentro de una carpeta llamada SolarAng, donde también se ubicarán los archivos del programa.
4. Ejecutar GMAT.exe, dentro de GMAT > bin para iniciar el programa.
5. Configurar los path en el archivo GMAT > bin > gmat_startup_file.txt:
 - a. ROOT_PATH: Path absoluto de la ubicación de la carpeta de máximo nivel de GMAT. Referencia (¹) en la Fig. 2.

6. OUTPUT_PATH: Path absoluto de la carpeta Runs .

```
#-----  
# Directory/file paths  
#-----  
  
# Occasional issues have been observed on Windows in which  
# GMAT cannot find certain data files, such as icons and DE  
# files, at runtime. If this occurs, set ROOT_PATH to the  
# absolute path to your top-level GMAT directory.  
  
ROOT_PATH          = C:/Users/Karen/Documents/Programas/GMAT/  
|  
#-----  
OUTPUT_PATH        = C:/Users/Karen/Documents/SolarAng1/Runs/  
LOG_FILE           = OUTPUT_PATH/GmatLog.txt  
SCREENSHOT_FILE     = OUTPUT_PATH/GmatScreenShot  
#-----
```

ii. Segunda parte: Para poder manipular GMAT desde Python, hay que configurar la *api* mediante los siguientes pasos:

1. Desde el prompt de Anaconda, en el *environnement* creado anteriormente py307, ir al directorio GMAT > api y ejecutar el comando <python> BuildApiStartupFile.py

Los comandos a utilizar son: `conda activate py307` para trabajar en el *environnement* que tiene la versión de Python que GMAT acepta; `cd <path_relativo>` que sirve para cambiar de carpeta usando el path relativo de la carpeta objetivo y el comando <python> BuildApiStartupFile.py donde BuildApiStartupFile.py es el nombre del script de python que se va a ejecutar y <python> es el comando que llama a python. Estos comandos se utilizan de la siguiente forma:

Para moverse a la carpeta bin desde el cmd hay que conocer la ubicación relativa a la ubicación actual de dicha carpeta, la llamaremos `path_api` para referirnos a ella. En el caso de la figura de abajo el path relativo entre la carpeta en la que estás es `C:\Users\Karen`, por lo que `path_api` es `Documents\Programas\GMAT\api`. Con el comando `dir` se puede ver una lista de los archivos que están dentro de la ubicación actual.

```

C:\Users\Karen>cd Documents\Programas\GMAT\api
Ubicación actual path_api
C:\Users\Karen\Documents\Programas\GMAT\api>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 60A7-3116

Directorio de C:\Users\Karen\Documents\Programas\GMAT\api

06/12/2022  14:29    <DIR>          .
06/12/2022  14:29    <DIR>          ..
15/12/2022  15:46    <DIR>          .ipynb_checkpoints
05/12/2022  08:39             2.947 API_README.txt
05/12/2022  08:39             1.791 BuildApiStartupFile.py
05/12/2022  08:39             1.839 Ex_R2020a_BasicFM.m
05/12/2022  08:39             1.611 Ex_R2020a_BasicFM.py
05/12/2022  08:39             2.430 Ex_R2020a_BasicForceModel.m
05/12/2022  08:39             2.444 Ex_R2020a_BasicForceModel.py
05/12/2022  08:39             3.451 Ex_R2020a_CompleteForceModel.m
05/12/2022  08:39             3.259 Ex_R2020a_CompleteForceModel.py
05/12/2022  08:39             2.302 Ex_R2020a_FindTheMoon.m
05/12/2022  08:39             2.371 Ex_R2020a_FindTheMoon.py
05/12/2022  08:39             1.242 Ex_R2020a_PropagationLoop.m
05/12/2022  08:39             1.412 Ex_R2020a_PropagationLoop.py
05/12/2022  08:39             1.193 Ex_R2020a_PropagationStep.m
05/12/2022  08:39             1.215 Ex_R2020a_PropagationStep.py
05/12/2022  08:39             4.615 Ex_R2020a_RangeMeasurement.m
05/12/2022  08:39             4.772 Ex_R2020a_RangeMeasurement.py
05/12/2022  08:39             6.147 Ex_R2020a_ToLuna.script
05/12/2022  08:39    <DIR>          Jupyter
06/12/2022  14:02             717 load_gmat.py
                18 archivos             45.758 bytes
                4 dirs 455.584.555.008 bytes libres

```

A continuación, dentro de la carpeta api, correr el comando `python BuildApiStartupFile.py`

```

C:\Users\Karen\Documents\Programas\GMAT\api>python BuildApiStartupFile.py

```

Esto creará el archivo `api_startup_file.txt` (Fig. 2) en la ubicación `GMAT > bin`. Para verlo se puede usar el comando `cd ../bin` donde `..` indica subir una carpeta.


```

C:\Users\Karen\Documents\Programas\GMAT\api>cd ../bin

C:\Users\Karen\Documents\Programas\GMAT\bin>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 60A7-3116

Directorio de C:\Users\Karen\Documents\Programas\GMAT\bin

06/12/2022  14:02    <DIR>          .
06/12/2022  14:02    <DIR>          ..
06/12/2022  14:02    <DIR>          .ipynb_checkpoints
15/12/2022  15:58             11.551 api_startup_file.txt
05/12/2022  08:39           332.336 condrv140.dll
05/12/2022  08:39           182.784 CsaltCWrapper.dll
05/12/2022  08:39           377.856 CSALTTester.exe
06/12/2022  11:23              870 GMAT - Acceso directo.lnk
05/12/2022  08:39          8.167.424 GMAT.exe
05/12/2022  08:39           11.277 GMAT.ini
05/12/2022  08:39          691.348 gmat.jar
05/12/2022  08:39           1.040 GMATAPI.m
06/12/2022  14:24    <DIR>          gmatpy
05/12/2022  08:39          4.423.680 gmat_java.dll
15/12/2022  14:56           10.552 gmat_startup_file.txt
05/12/2022  08:39           179.200 libCInterface.dll
05/12/2022  08:39           844.800 libCSALT.dll

```

iii. Configurar/verificar los path en el archivo GMAT > bin > api_startup_file.txt:

1. ROOT_PATH: Path absoluto de la ubicación de la carpeta de máximo nivel de GMAT. Referencia ⁽¹⁾ en la Fig. 2.
2. OUTPUT_PATH: Path de la carpeta Runs donde se guardaran todos los outputs de las simulaciones. Referencia ⁽²⁾ en la Fig. 2

```

#-----
# Directory/file paths
#-----

# Occasional issues have been observed on Windows in which
# GMAT cannot find certain data files, such as icons and DE
# files, at runtime. If this occurs, set ROOT_PATH to the
# absolute path to your top-level GMAT directory.

ROOT_PATH          = C:/Users/Karen/Documents/Programas/GMAT/

#-----
OUTPUT_PATH        = C:/Users/Karen/Documents/GMATdocs/Runs/
LOG_FILE           = OUTPUT_PATH/GmatLog.txt
SCREENSHOT_FILE     = OUTPUT_PATH/GmatScreenShot
#-----
MEASUREMENT_PATH   = OUTPUT_PATH/
#-----

```

iv. Abrir el archivo GMAT > api > load_gmat.py y modificar GmatInstall con el path de la carpeta de máximo nivel de GMAT.

Nota: El archivo `load_gmat.py` puede ser abierto con un editor de texto o con el editor de Python Spider:

```
import sys
from os import path

apistartup = "api_startup_file.txt"
GmatInstall = "C:/Users/Karen/Documents/Programas/GMAT"
GmatBinPath = GmatInstall + "/bin"
Startup = GmatBinPath + "/" + apistartup

if path.exists(Startup):

    sys.path.insert(1, GmatBinPath)

    import gmatpy as gmat
    gmat.Setup(Startup)

else:
    print("Cannot find ", Startup)
    print()
    print("Please set up a GMAT startup file named ", apistartup, " in the ",
          GmatBinPath, " folder.")
```

- v. Copiar `load_gmat.py` en la carpeta donde se ubicarán los archivos `.py` que conforman el programa SolarAng1 (Fig. 3) y la carpeta `OUTPUT_PATH`.

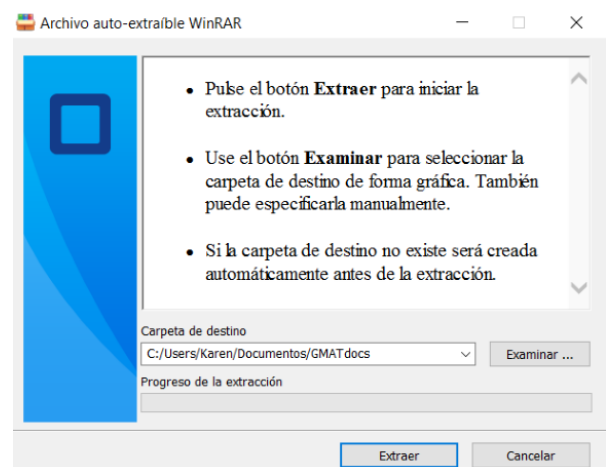
c. SolarAng1

Por último, hay que descargar el archivo del programa desde [SolarAng1 - DES - Google Drive](#) [No funciona! usar: [este link](#)]. Este es un archivo autoextraíble del tipo "rar". Una vez descargado haciendo doble click le pedirá elegir el lugar donde colocar los archivos, estos deben archivar de forma que queden en la misma ubicación que `Runs(OUTPUT_PATH)` (ver figura 3).

Si ya existen las carpetas `Runs` y/o `Scripts_GMAT`, el sistema le preguntará si desea reemplazar/combinar las carpetas. Aceptar esta opción ya que dentro de la carpeta `Runs` hay ejemplos de resultados luego del uso del programa.

Una vez realizado este paso ya se puede utilizar el programa SolarAng1 desde la consola Spyder. Ver apartado "Uso del programa".

Ejemplos: SAOCOM 1-A



5. Programa SolarAng1

Los archivos que constituyen el programa SolarAng1 deben estar en la misma ubicación que `Runs(OUTPUT_PATH)`, `Scripts_GMAT` y `load_gmat.py` como se ve en la Fig. 3.

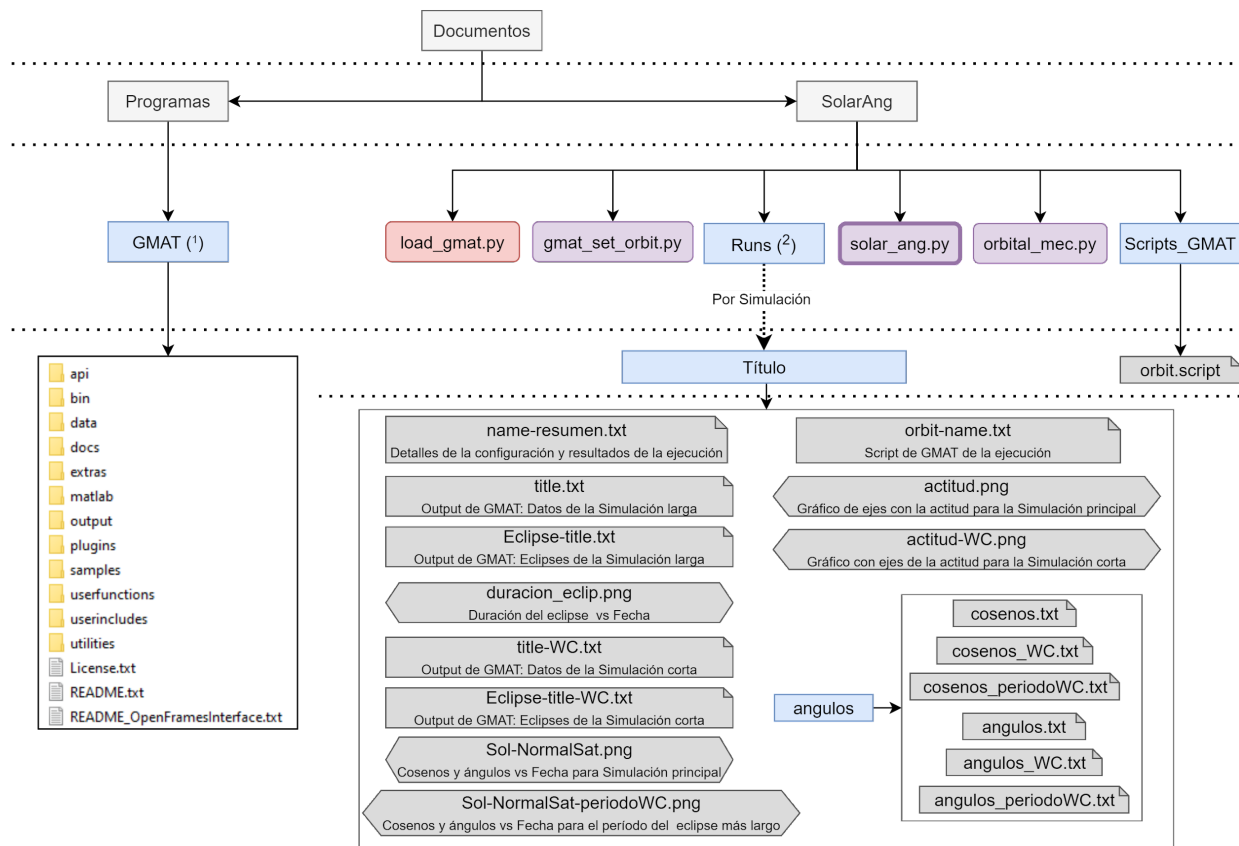


Figura 3. Distribución de las carpetas y archivos del programa completo. Dentro de SolarAng1 hay dos carpetas: Scripts_GMAT que contiene los scripts de GMAT base a utilizar y Runs que archiva cada carpeta creada en cada simulación. Además, SolarAng1 contiene los 3 scripts de Python del programa (solar_ang.py, gmat_set_orbit.py, orbital_mec.py).

Estos son scripts de python (en violeta en la Fig. 3) y se describe cada uno a continuación:

a. solar_ang.py (main)

Script principal del programa SolarAng en el que se llevan a cabo las rutinas que conforman las acciones del programa. Para las rutinas se utilizan las funciones definidas en gmat_set_orbit.py y orbital_mec.py.

Rutinas:

- i. Configuración de las variables de la simulación.
- ii. Importación de librerías y ejecución de scripts.
- iii. Modificación del script de GMAT con las definidas en el punto 1. Si ya existe una simulación con la misma órbita y actitud, pregunta si se quiere continuar. (**set_script**)
- iv. Ejecución de la simulación de GMAT. (**run_gmat**)
- v. Extracción de datos útiles de los output de GMAT (**data_orbit**)
- vi. Cálculo de cosenos con los datos extraídos (**cosenos**)
- vii. Extracción de datos útiles de los output de GMAT (**data_eclip**)
- viii. Aplicar las sombras a los cosenos calculados (**add_eclip**)
(No se agregan los eclipses ya que el step de 0.5 días es mayor a la duración de un eclipse. GMAT determina por separado la información de la órbita de los eclipses presentes).
- ix. Generar gráfico de coseno y ángulos en función del tiempo. (**graph_cos**)
- x. Generación de un reporte con los cosenos calculados (**report_cos**)
- xi. [Si el período de la simulación posee eclipses] Gráfico de duración de eclipses vs. fecha (**eclip_duration**)

- xii. Definición del eclipse más largo y selección de variables para la configuración de la simulación 2, más corta (1 step anterior de 0.5 días + 1 período) y con un menor step. (`subrunWC_params`)
- xiii. Repetición de los pasos 3, 4, 5 y 6 para esa nueva simulación. (`set_script`, `run_gmat`, `data_orbit`, `cosenos`)
- xiv. Obtención de los puntos en que el satélite está en eclipse. (`getpos_eclip`)
- xv. Aplicar las sombras y eclipses a los cosenos calculados. (`add_eclip`)
- xvi. Generación de un reporte con los cosenos calculados (`report_cos`)
- xvii. Selección de los datos que se quieren graficar del reporte creado en el paso 16. Los datos serán los que correspondan al período [endWC-período, endWC] (`select_timedelta`)
- xviii. Graficar el período completo que contiene al eclipse más largo, ubicando el mismo al final (`graph_cos`)
- xix. Generar el reporte de los cosenos de dicho período. (`report_cos`)
- xx. Generación de un archivo .txt con datos de la configuración de la simulación, y algunos resultados de la misma. (`print_report`)

Si el período inicial no tiene eclipses, se omiten los pasos 8 a 12 y el código imprime “El período seleccionado no tiene eclipses”.

b. gmat_set_orbit.py (paquete/librería para el control de GMAT)

Definición de funciones utilizadas para configurar el script de GMAT y controlarlo. Se le asignó la abreviación `gso.` para el uso de sus funciones.

- i. `set_script`: modifica la configuración definida para la simulación (parámetros orbitales, de sampleo, etc.) en el script de GMAT
 - 1. `set_prop`: modifica el valor de una propiedad en el script de GMAT según un elemento (satélite, Tierra, Sol, etc.), una propiedad (SMA, Epoch, etc.) y su valor correspondiente
 - 2. `set_duration`: modifica la duración de la simulación en el script de GMAT (en días)
 - 3. `set_step`: modifica el step de la simulación en el script de GMAT (en días)
- ii. `print_report`: genera un reporte con las propiedades de la simulación
- iii. `run_gmat`: Ejecuta el script de GMAT y mueve los resultados desde `OUTPUT_PATH` hasta la carpeta donde se ubican todos los resultados de la simulación.

c. orbital_mec.py (paquete/librería para la mecánica orbital)

Definición de funciones utilizadas para extraer, analizar y generar gráficos y reportes de los outputs de las simulaciones de GMAT. Se le asignó la abreviación `om.` para el uso de sus funciones.

- i. `ang_vec`: calcula en ángulo y el coseno entre dos vectores
- ii. `time_params`: recibe un string “21 Mar 2000 11:59:28.000” y devuelve → yr, mon, day, hs, mins, sec
- iii. `data_orbit`: lee el archivo de la órbita y entrega listas con la información obtenida por la simulación (tiempos, posiciones, ángulos, parámetros orbitales keplerianos, período, número de órbita, posición del sol)
- iv. `data_eclip`: lee el archivo de eclipses de la corrida indicada y genera listas de tiempos de inicio, finalización y duración de cada eclipse
- v. `getpos_eclip`: con los datos de `data_eclip` y `data_orbit` genera listas de los index y posición XYZ de todos los puntos en los que el satélite se encuentra eclipsado.
- vi. `graph_earth`: grafica la Tierra en la figura `actitud.png`
- vii. `sist_ref`: genera los ejes unitarios del sistema de referencia ECI (y sus versiones negativas)
- viii. `add_eclip`: anula todos los cosenos de los puntos de la órbita donde el satélite se encuentra eclipsado y anula cualquier coseno negativo (*agrega las sombras*)
- ix. `graph_cos`: grafica ángulos y cosenos vs fecha

- x. `report_cos`: genera un reporte .txt con los cosenos de cada cara y panel desplegable, la distancia al sol y la fecha de cada punto de la órbita
- xi. `paneles_desplegables`: Calcula el coseno entre el vector normal del panel y los del sistema body, después suma los ángulos panel-body y body-sun y entrega sus cosenos en `panelsun_cos` para calcularlos una vez realizada la ejecución del código cosenos.
 - 1. `cos_dir`: calcula el vector de cosenos directores
- xii. `num_period`: Entrega una lista que indica a qué número de periodo pertenece cada punto de la órbita
- xiii. `eclip_duration`: Grafica la duración de cada eclipse en función de la fecha
- xiv. `cosenos`: Calcula los cosenos entre la cara del satélite y el sol, y dibuja ilustrativamente la orientación del satélite.
 - 1. `cos_sun`: calcula el coseno del ángulo entre el vector sol (Tierra-Sol) y la normal de cada cara de un satélite y su distancia al sol en un punto específico de su órbita
 - a. `get_axbody`: genera los vectores rotados con los ángulos de Euler y opcionalmente lo grafica en la posición del satélite correspondiente
 - i. `rotEuler`: rota un vector con los ángulos de Euler ingresados
 - b. si `calc_sun = True`
 - i. `get_sunvect`: calcula y grafica el vector entre la posición del sol y la posición del satélite
 - 1. `sun_vector`: calcula el vector entre la posición del sol y la posición del satélite con una aproximación que posee un error de 2.4° .
 - c. si `calc_sun = False`
 - i. Toma el vector posición sol de GMAT.
 - 2. `add_paneles`: Agrega a los ejes del sistema de referencia, el vector normal a cada panel desplegable
- xv. `select_timedelta`: genera un reporte y un gráfico de los cosenos de un período de tiempo seleccionado
- xvi. `subrunWC_params`: con las fechas de inicio, finalización y duración de los eclipses selecciona el eclipse más largo y define los parámetros de la órbita medidos por la simulación anual para el Epoch de la nueva corrida. El Epoch también lo elige tal que incluya el eclipse más largo del año.

6. Uso del programa

a. Paso a paso

A continuación se explica paso a paso para usar el programa SolarAng1.

Nota: Cada advertencia que se encuentre en este paso a paso representa un mensaje de alerta que envía el programa.

- i. Desde Spyder, abrir el archivo `solar_ang.py`.
- ii. Introducir las variables de entrada:

	Variable	Descripción	Rangos	Ejemplo
FILE	Name	Nombre general de la simulación	string a elección (recomendación: lo más corto posible)	"SAC-D"
	script_name	Nombre del script de GMAT base a modificar	string exacto del nombre (todo lo anterior al punto)	"orbita"
ÓRBITA	Epoch	Fecha (Estampa temporal del inicio de la simulación)	string: "dd MMM yyyy hh:mm:ss.uuu" ¹	"20 Mar 2011 00:00:00.000" (²)
	SMA	Altura de la órbita (Semi Mayor Axis - R_{Tierra})	float (en km)	650
	ECC	Excentricidad	float: $0 > ECC > 1$	0.0012
	INC	Inclinación	float: $< 180^\circ$	98
	RAAN	Right Ascension of the Ascending Node	float: $0^\circ > RAAN > 360^\circ$	90
	AOP	Argument of Perigee	float: $0^\circ > AOP > 360^\circ$	90
	TA	True Anomaly	float: $0^\circ > TA > 360^\circ$	0
ACTITUD: NadirPointing ³	ARB AttitudeReferenceBody	Cuerpo al que se fija la orientación	string exacto del nombre de cuerpo en inglés ⁴	"Sun"
	BAV BodyAlignmentVector	Cara del satélite (o vector) que se orienta al cuerpo de referencia	vector de floats: [float, float, float]	[1,0,0]
	ACT AttitudeConstraintType	Tipo de orientación respecto de la órbita	string exacto del tipo de orientación según GMAT ⁴	"Velocity"
	BCV BodyConstraintVector	Cara del satélite (o vector al que se fija)	vector de floats: [float, float, float]	[0,1,1]
	paneles ⁵	Lista de vectores normales correspondientes a cada panel desplegable.	[[float, float, float], ..., [float, float, float]]	[[0,1,1], [1,1,0]]
	step	Intervalo de tiempo entre datos tomados	float (en días)	0.5 (sugerido)
	duration_run	Duración de la simulación	float (en días)	367 ² (sugerido)

Tabla 1. Settings o variables de entrada del programa del programa.

Estas variables se encuentran definidas luego de la importación de librerías en el script `solar_ang.py` como se ve en la Fig. 4.

¹ d: día - M: mes - y: año - h: hora - m: minuto - s: segundo - u: milisegundo

² Puede ser una buena práctica realizar una simulación inicial de 1 año completo más 2 días y elegir los parámetros orbitales para el Epoch con un día antes, así se observa en los resultados la continuidad periódica anual de los ángulos (que coincida el inicio con el final).

³ En esta versión del programa, el tipo de actitud Nadir Pointing debe mantenerse fijo ya que el programa se escribió para este caso específicamente.

⁴ Las abreviaciones se pueden encontrar en la documentación de GMAT.

⁵ Los vectores normales a los paneles hay que expresarlos en la base del sistema body. Ver apéndice.

```
# ----- Variables de entrada -----
R_Tierra = 6371 # km
name = "Test"
Epoch = '16 Jan 2023 19:24:16.000'
SMA = R_Tierra + 669 #7035
ECC = 0.00001839 #0.0012
INC = 98.0028 #98
RAAN = 27.5638 #90
AOP = 113.3397 #90
TA = 59.4917
ARB = "Earth" # AttitudeReferenceBody
BAV = [1,0,0] # BodyAlignmentVectorX, BodyAlignmentVectorY, BodyAlignmentVectorZ
ACT = "Velocity" # AttitudeConstraintType
BCV = [0,0,1] # BodyConstraintVectorX, BodyConstraintVectorY, BodyConstraintVectorZ
step = 0.0001 # 0.5 # Days Calcular
duration_run = 0.25 #30 # Days Calcular
script_name = "orbit"
paneles = []

# ----- Fin -----
```

Figura 4. Variables principales configurables para la definición de la órbita y la actitud del satélite en ella.

Debajo de las variables principales, se encuentran las variables de las opciones avanzadas (Fig. 5). Estas son:

- `force_overwrite` = Si se define `True`, en el caso de que se quiera hacer una ejecución con los mismos parámetros orbitales y de actitud, no se le preguntará si se desea sobre escribir los archivos. Si se define `False`, se preguntará.
- `Attitude` = Se define "NadirPointing" para todos los casos. Esta opción será modificable en versiones posteriores.
- `corte_subrun` = Cantidad mínima de días para las cuales se realizarán 2 simulaciones. Si `duration_run` es mayor, se realizarán dos simulaciones (la segunda con un `step` de 30 segundos). Si es menor, se realizará una sola con el `step` definido en las variables principales. Esta opción permite al usuario decidir cuánto detalle y tiempo de ejecución desea obtener.
- `model_file` = El programa contiene un modelo 3D de un CubeSat 3U con ejes salientes diferentes ((x,y,z) = (pirámide, cono chico, cono grande)). Al ejecutar una vez el código, el script de GMAT se actualizará para usar este modelo. Si se desea usar otro modelo, se puede elegir su path en esta variable.
- `scale_actitud` = Escala utilizada para graficar los ejes de coordenadas rotados según la actitud del satélite en el gráfico `actitud.png`.
- `scale_sol` = Escala utilizada para graficar los vectores que apuntan en dirección al sol en el gráfico `actitud.png`.
- `scale_panel` = Escala utilizada para graficar los vectores normales a los paneles solares (en paneles) según la actitud del satélite en el gráfico `actitud.png`.
- `lines` = `True` si se desean agregar líneas entre los puntos del gráfico de ángulos y cosenos para la simulación larga. `False` si no.

```
# ----- Opciones Avanzadas -----
force_overwrite = True
Attitude = "NadirPointing"
corte_subrun = 1 # días
model_file = "\"" + os.getcwd() + "\\CubeSat3U-salientes.3ds" + "\""
scale_actitud = 700
scale_sol = 2000
scale_panel = 2000
lines = True #Agregar líneas al gráfico de la simulación larga.
# -----
```

Figura 5. Variables configurables de las opciones avanzadas

iii. Ejecutar el código.

1. Durante la ejecución, a medida que el programa avanza con los procesos, muestra mensajes por la consola informando en qué etapa se encuentra. A continuación se muestra un ejemplo para los dos posibles casos:

```
In [67]: runfile('C:/Users/Karen/Documents/SolarCalc/SolarAng/solar_ang.py', wdir='C:/Users/Karen/Documents/SolarCalc/SolarAng')
Reloaded modules: gmat_set_orbit, load_gmat, gmatpy, gmatpy.gmat_py, gmatpy._gmat_py, gmatpy.station_py, gmatpy._station_py, gmatpy.navigation_py, gmatpy._navigation_py, orbital_mec
---- Periodo de tiempo de órbita simulada mayor a 1 día(s) → Se realizaran dos simulaciones.
Configurando el script de GMAT.
Iniciando simulacion de GMAT.
Fin de la simulacion de GMAT (Duracion = 191.640625 segundos)
Creando graficos (case = 1).
Configurando el script de GMAT.
Iniciando simulacion de GMAT.
Fin de la simulacion de GMAT (Duracion = 1.109375 segundos)
Creando graficos (case = periodoWC).
<Figure size 720x360 with 0 Axes>
```

Estos mensajes significan lo siguiente:

- a. Si existe una carpeta de una simulación con los mismos parámetros de órbita y actitud, el programa preguntará si se quieren pisar los resultados. Tener en cuenta que algunos parámetros como la duración de la simulación o los paneles desplegables no se incluyen en el título y generan este caso; si se desean guardar ambas simulaciones una recomendación es modificar la variable `name`.
- b. Configura el script de GMAT con los parámetros definidos para la simulación.
- c. Inicia la primera simulación en GMAT
- d. Finaliza la primera simulación en GMAT e informa su duración.
- e. Inicia a generar los gráficos para el caso más general (`case = 1`).
- f. Configura el script de GMAT con los parámetros definidos para la segunda simulación (worst case)
- g. Inicia la segunda simulación en GMAT
- h. Finaliza la segunda simulación en GMAT e informa su duración.
- i. Inicia a generar los gráficos para el período del peor caso (`case = periodoWC`).
- j. El mensaje `<Figure size 720x360 with 0 Axes>` puede ser ignorado.

b. Notificaciones específicas de cada simulación

El programa envía mensajes informativos o de advertencia durante la ejecución. Las mismas se denotan con los guiones “----”. A continuación se explican las advertencias posibles y se aclara en qué función se detectan.

- “Ya existe una simulación con esta órbita y actitud. Está segura de que quiere hacer esta corrida? Se borrarán los resultados anteriores. S: Sí, N: Cualquier tecla”


```
(gso.set_script())
```

Si existe una carpeta de una simulación con los mismos parámetros de órbita y actitud, el programa preguntará si se quieren pisar los resultados. Esta advertencia requiere la acción del usuario: escribiendo `s` o `S` en la consola, se acepta la sobreescritura de los resultados de alguna ejecución para la misma órbita y actitud anterior. Con cualquier otra tecla, el programa finaliza.

Tener en cuenta que algunos parámetros como la duración de la simulación o los paneles desplegados no se incluyen en el título y generan este caso; si se desean guardar ambas simulaciones una recomendación es modificar la variable `name`.

Esta parte se puede omitir, forzando al programa a sobre escribir, eligiendo en la función `set_script(force_overwrite = True)`.

- “El eclipse más largo es el último de la simulación.”

```
(om.subrunWC_params())
```

Esto podría significar que más adelante en el tiempo, la órbita puede tener eclipses aún más largos. Solo para Simulación larga.

- “El eclipse más largo es el primero de la simulación.”

```
(om.subrunWC_params())
```

Esto puede significar que más atrás en el tiempo, la órbita puede tener eclipses aún más largos. Solo para Simulación larga.

- “Período de tiempo de órbita simulada menor a 1 día(s) → Se realiza una simulación.”

```
(solar_ang.py)
```

Si `duration_run` es menor a 1 día (o el tiempo de corte `corte_subrun` definido), avisa al usuario que se realizará una sola simulación: “Simulación corta”.

- “Período de tiempo de órbita simulada mayor a {corte_subrun} día(s) → Se realizaran dos simulaciones.”

```
(solar_ang.py)
```

Si `duration_run` es mayor a 1 día (o el tiempo de corte `corte_subrun` definido), avisa al usuario que se realizarán dos simulaciones: “Simulación larga”.

- “El periodo seleccionado no tiene eclipses”

```
(om.data_eclip())
```

Cuando el programa envía este mensaje procede de distintas formas según el tipo de simulación:

- Simulación larga: El programa finaliza.
- Simulación corta: El programa continúa omitiendo los pasos donde se extraen los datos necesarios sobre los eclipses y el paso del período WC.

En el siguiente ejemplo aparecen algunos de estos mensajes:

```

In [84]: runfile('C:/Users/Karen/Documents/SolarCalc/SolarAng/solar_ang.py', wdir='C:/
Users/Karen/Documents/SolarCalc/SolarAng')
Reloaded modules: gmat_set_orbit, load_gmat, gmatpy, gmatpy.gmat_py, gmatpy._gmat_py,
gmatpy.station_py, gmatpy._station_py, gmatpy.navigation_py, gmatpy._navigation_py,
orbital_mec
Período de tiempo de órbita simulada mayor a 1 día(s) → Se realizaran dos simulaciones.
Configurando el script de GMAT.
---- Ya existe una simulacion con esta orbita y actitud. Esta segura de que quiere hacer
esta corrida? Se borrarán los resultados anteriores. S: Si, N: No
S
Iniciando simulacion de GMAT.
Fin de la simulacion de GMAT (Duracion = 15.015625 segundos)
Creando graficos (case = 1).
---- El eclipse mas largo es el ultimo del periodo de tiempo seleccionado.
Configurando el script de GMAT.
Iniciando simulacion de GMAT.
Fin de la simulacion de GMAT (Duracion = 1.09375 segundos)
Creando graficos (case = periodoWC).
<Figure size 720x360 with 0 Axes>

```

c. Errores posibles

- Si el programa no encuentra el archivo titulo.txt de la simulación principal dentro de la carpeta de la corrida, la razón más probable es que GMAT no está generando los archivos de salida en el path OUTPUT_PATH, sino en otro lado.

```

File "C:\Users\Karen\Documents\Programas\miniconda3\envs\py307\lib\site-
packages\pandas\io\common.py", line 707, in get_handle
    newline="",

FileNotFoundError: [Errno 2] No such file or directory: '../Runs/Test_EpMar00-
SMA8000-ECC01-INC30-RAAN0-AOP0-TA0_NadE100-V001/Test_EpMar00-SMA8000-ECC01-INC30-
RAAN0-AOP0-TA0_NadE100-V001.txt'

```

Posibles soluciones:

- Verificar la correcta definición de los Path en gmat_startup_file.txt y load_gmat.py
- Si los path son muy largos es probable que haya un conflicto de máxima longitud con los mismos, para solucionarlo se puede configurar Windows 10 siguiendo el siguiente instructivo:
<https://learn.microsoft.com/es-es/windows/win32/fileio/maximum-file-path-limitation?tabs=registry#enable-long-paths-in-windows-10-version-1607-and-later>.

- Si GMAT está abierto, el programa enviará este error:

```
Configurando el script de GMAT.
---- Ya existe una simulacion con esta orbita y actitud. Esta segura de que quiere hacer esta
corrida? Se borrarán los resultados anteriores. S: Si, N: No
S
Traceback (most recent call last):

  File "C:\Users\Karen\Documents\Programas\miniconda3\envs\py307\lib\site-
packages\spyder_kernel\py3compat.py", line 356, in compat_exec
    exec(code, globals, locals)

  File "C:\Users\Karen\Documents\ProgramaDeCosenos\cosenos.py", line 38, in <module>
    set_script(script_name, file, values, step, duration_run)

  File "C:\Users\Karen\Documents\ProgramaDeCosenos\gmt_set_orbit.py", line 48, in set_script
    shutil.rmtree(file)

  File "C:\Users\Karen\Documents\Programas\miniconda3\envs\py307\lib\shutil.py", line 516, in
rmtree
    return _rmtree_unsafe(path, onerror)

  File "C:\Users\Karen\Documents\Programas\miniconda3\envs\py307\lib\shutil.py", line 404, in
_rmtree_unsafe
    onerror(os.rmdir, path, sys.exc_info())

  File "C:\Users\Karen\Documents\Programas\miniconda3\envs\py307\lib\shutil.py", line 402, in
_rmtree_unsafe
    os.rmdir(path)

PermissionError: [WinError 32] El proceso no tiene acceso al archivo porque está siendo
utilizado por otro proceso: 'Runs\Test_EpMar00-SMA7021-ECC839e-05-INC98.0028-RAAN27.5638-
AOP113.3397-TA59.4917_NadE100-V001'
```

La solución es cerrar el programa GMAT. En próximas versiones esto podría modificarse.

- Error en el autoreload: Esta advertencia indica que el módulo "orbital_mec" no se ha encontrado. Esto significa que la operación de recarga automática no se ha realizado correctamente. Como resultado, el programa no estará actualizado con la última versión del módulo y esto podría causar problemas en el funcionamiento del programa.

```
In [22]: runfile('C:/Users/Karen/Documents/ProgramaDeCosenos/cosenos.py', wdir='C:/Users/
Karen/Documents/ProgramaDeCosenos')
Reloaded modules: gmat_set_orbit, load_gmat, gmatpy, gmatpy.gmat_py, gmatpy._gmat_py,
gmatpy.station_py, gmatpy._station_py, gmatpy.navigation_py, gmatpy._navigation_py,
orbital_mec
[autoreload of orbital_mec failed: Traceback (most recent call last):
  File "C:\Users\Karen\Documents\Programas\miniconda3\envs\py307\lib\site-
packages\IPython\extensions\autoreload.py", line 245, in check
    superreload(m, reload, self.old_objects)
  File "C:\Users\Karen\Documents\Programas\miniconda3\envs\py307\lib\site-
packages\IPython\extensions\autoreload.py", line 394, in superreload
    module = reload(module)
  File "C:\Users\Karen\Documents\Programas\miniconda3\envs\py307\lib\imp.py", line 314, in
reload
    return importlib.reload(module)
  File "C:\Users\Karen\Documents\Programas\miniconda3\envs\py307\lib\importlib\__init__.py",
line 168, in reload
    raise ModuleNotFoundError(f"spec not found for the module {name!r}", name=name)
ModuleNotFoundError: spec not found for the module 'orbital_mec'
]
```

Si el script orbital_mec.py se encuentra en la misma ubicación que solar_ang.py, puede intentar volver a cargar el módulo. Esto se puede hacer utilizando la función de recarga de Python: "importlib.reload ()". Esta función recargará el módulo desde su ubicación actual. Si esto no resuelve el problema, puede intentar reiniciar el intérprete de Python (Spyder) y ver si esto ayuda.

→ Puede ser necesario eliminar manualmente la carpeta de la ejecución luego de la aparición de este error, para que pueda funcionar correctamente en la segunda ejecución (en el caso de que se desee ejecutar nuevamente la misma órbita y actitud)

7. Resultados obtenidos

Una vez terminada la ejecución de solar_ang.py se crea automáticamente una carpeta en Runs con el nombre del título de la simulación (Fig. 6) y en ella se guardan todos los archivos que se muestran en la Fig. 3 dentro del rectángulo.

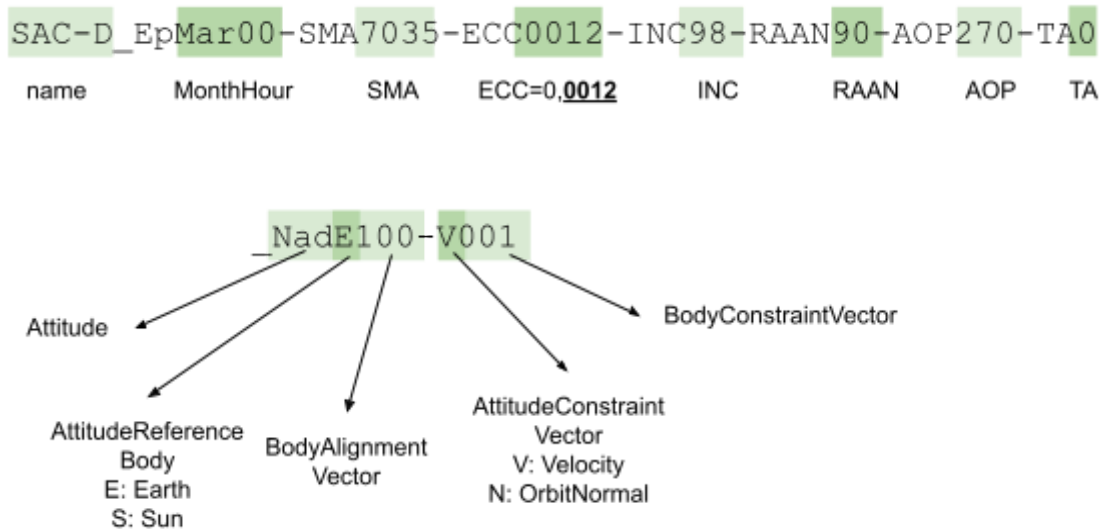
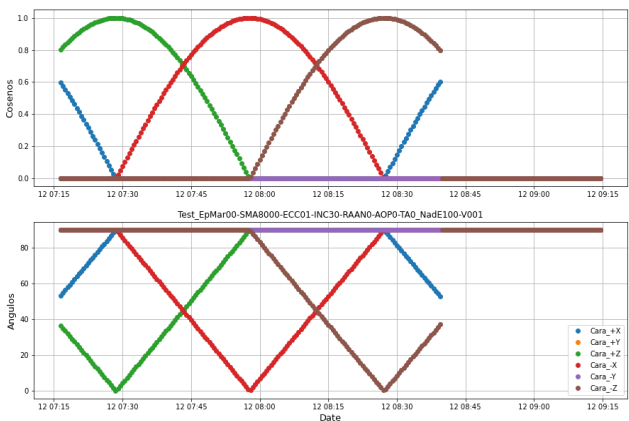


Figura 6. Título de una ejecución del programa SolarAng1. Referencias.

La descripción de los archivos de salida se detallan en la tabla 2:

Nombre	Descripción	Ejemplos
actitud.png	Gráfico del sistema de referencia del cuerpo (satélite) con la orientación dada por la actitud para los puntos de la órbita calculados (con intervalos dt) y el vector Sol de su correspondiente fecha.	<p>Mar00-SMA8000-ECC01-INC30-RAAN0-AOP0-TA0_NadE100-V</p> <p>x,y,z = ["lightblue", "blue", "darkblue"] paneles="green"</p>

actitud-WC.png	Idem anterior más: Se grafican la órbitas en gris y el eclipse en rojo para toda la segunda simulación (0.5 días + período)	<div>Mar00-SMA8000-ECC01-INC30-RAAN0-AOP0-TAO_NadE100-V</div>																																																
eclip_duration s.png	Gráfico de duración de cada eclipse en función de la fecha (durante un año)	<div>Test_EpMar00-SMA8000-ECC01-INC30-RAAN0-AOP0-TAO_NadE100-V001</div>																																																
Eclipse_titulo _case.txt	Archivo generado por GMAT. Información de cada eclipse existente en cada simulación.	<div>Spacecraft: Sat</div> <table><thead><tr><th>Start Time (UTC)</th><th>Stop Time (UTC)</th><th>Duration (s)</th><th>Occ Body</th><th>Type</th><th>Event Number</th></tr></thead><tbody><tr><td colspan="6">Total Duration (s)</td></tr><tr><td>13 May 2011 04:58:25.882</td><td>13 May 2011 04:59:11.067</td><td>45.184909122</td><td>Earth</td><td>Umbral</td><td>1</td></tr><tr><td>45.184909122</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>13 May 2011 06:36:17.711</td><td>13 May 2011 06:37:41.108</td><td>83.397723630</td><td>Earth</td><td>Umbral</td><td>2</td></tr><tr><td>83.397723630</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>13 May 2011 08:14:17.258</td><td>13 May 2011 08:16:03.453</td><td>106.19483079</td><td>Earth</td><td>Umbral</td><td>3</td></tr><tr><td>106.19483079</td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table> <div>Number of individual events : 1192 Number of total events : 1192 Maximum duration (s) : 1114.5624955 Maximum duration at the 618th eclipse.</div>	Start Time (UTC)	Stop Time (UTC)	Duration (s)	Occ Body	Type	Event Number	Total Duration (s)						13 May 2011 04:58:25.882	13 May 2011 04:59:11.067	45.184909122	Earth	Umbral	1	45.184909122						13 May 2011 06:36:17.711	13 May 2011 06:37:41.108	83.397723630	Earth	Umbral	2	83.397723630						13 May 2011 08:14:17.258	13 May 2011 08:16:03.453	106.19483079	Earth	Umbral	3	106.19483079					
Start Time (UTC)	Stop Time (UTC)	Duration (s)	Occ Body	Type	Event Number																																													
Total Duration (s)																																																		
13 May 2011 04:58:25.882	13 May 2011 04:59:11.067	45.184909122	Earth	Umbral	1																																													
45.184909122																																																		
13 May 2011 06:36:17.711	13 May 2011 06:37:41.108	83.397723630	Earth	Umbral	2																																													
83.397723630																																																		
13 May 2011 08:14:17.258	13 May 2011 08:16:03.453	106.19483079	Earth	Umbral	3																																													
106.19483079																																																		
Sol-NormalSat. png	Gráfico de cosenos y de ángulos en función de la fecha (a lo largo de un año) Opcional agregarle líneas																																																	

Sol-NormalSat-periodoWC.png	Gráfico de cosenos y de ángulos en función de la fecha (para el Worst Case)																																																																																									
titulo.txt	Archivo generado por GMAT. Información de la posición del satélite, la órbita y el sol (con intervalos de tiempo step)	<table><tr><th>Sat.UTCGregorian</th><th>Sat.ElapsedSecs</th><th>Sat.Earth.LST</th><th>Sat.EulerAngle1</th><th>Sat.EulerAngle2</th><th>Sat.EulerAngle3</th><th>Sat.EulerAngle4</th><th>Sat.EulerAngle5</th></tr><tr><td>20 Mar 2011 00:00:00.000</td><td>0</td><td>0.14367</td><td>180</td><td>-0</td><td>60</td><td></td><td></td></tr><tr><td>1.4891e+08</td><td>7920</td><td>0</td><td>1541.9</td><td>0.12792</td><td>30</td><td>0.01</td><td></td></tr><tr><td>20 Mar 2011 12:00:00.000</td><td>43200</td><td>24.636</td><td>-155.51</td><td>14.405</td><td>63.407</td><td></td><td></td></tr><tr><td>1.4895e+08</td><td>6987.6</td><td>3182.3</td><td>1550.5</td><td>-0.89816</td><td></td><td></td><td></td></tr><tr><td>21 Mar 2011 00:00:00.000</td><td>86400</td><td>52.178</td><td>-127.99</td><td>25.533</td><td>73.737</td><td></td><td></td></tr><tr><td>1.4898e+08</td><td>7996.7</td><td>50.532</td><td>356.1</td><td>29.976</td><td>0.0094422</td><td></td><td></td></tr><tr><td>21 Mar 2011 12:00:00.000</td><td>129600</td><td>83.405</td><td>-96.776</td><td>29.965</td><td>89.554</td><td></td><td></td></tr><tr><td>1.4898e+08</td><td>7996.7</td><td>50.532</td><td>356.1</td><td>29.976</td><td>0.0094422</td><td></td><td></td></tr><tr><td>22 Mar 2011 00:00:00.000</td><td>172800</td><td>114.52</td><td>-65.653</td><td>26.008</td><td>105.44</td><td></td><td></td></tr><tr><td>1.4901e+08</td><td>7996.6</td><td>102.95</td><td>352.13</td><td>29.971</td><td>0.0091615</td><td></td><td></td></tr></table>	Sat.UTCGregorian	Sat.ElapsedSecs	Sat.Earth.LST	Sat.EulerAngle1	Sat.EulerAngle2	Sat.EulerAngle3	Sat.EulerAngle4	Sat.EulerAngle5	20 Mar 2011 00:00:00.000	0	0.14367	180	-0	60			1.4891e+08	7920	0	1541.9	0.12792	30	0.01		20 Mar 2011 12:00:00.000	43200	24.636	-155.51	14.405	63.407			1.4895e+08	6987.6	3182.3	1550.5	-0.89816				21 Mar 2011 00:00:00.000	86400	52.178	-127.99	25.533	73.737			1.4898e+08	7996.7	50.532	356.1	29.976	0.0094422			21 Mar 2011 12:00:00.000	129600	83.405	-96.776	29.965	89.554			1.4898e+08	7996.7	50.532	356.1	29.976	0.0094422			22 Mar 2011 00:00:00.000	172800	114.52	-65.653	26.008	105.44			1.4901e+08	7996.6	102.95	352.13	29.971	0.0091615		
Sat.UTCGregorian	Sat.ElapsedSecs	Sat.Earth.LST	Sat.EulerAngle1	Sat.EulerAngle2	Sat.EulerAngle3	Sat.EulerAngle4	Sat.EulerAngle5																																																																																			
20 Mar 2011 00:00:00.000	0	0.14367	180	-0	60																																																																																					
1.4891e+08	7920	0	1541.9	0.12792	30	0.01																																																																																				
20 Mar 2011 12:00:00.000	43200	24.636	-155.51	14.405	63.407																																																																																					
1.4895e+08	6987.6	3182.3	1550.5	-0.89816																																																																																						
21 Mar 2011 00:00:00.000	86400	52.178	-127.99	25.533	73.737																																																																																					
1.4898e+08	7996.7	50.532	356.1	29.976	0.0094422																																																																																					
21 Mar 2011 12:00:00.000	129600	83.405	-96.776	29.965	89.554																																																																																					
1.4898e+08	7996.7	50.532	356.1	29.976	0.0094422																																																																																					
22 Mar 2011 00:00:00.000	172800	114.52	-65.653	26.008	105.44																																																																																					
1.4901e+08	7996.6	102.95	352.13	29.971	0.0091615																																																																																					
titulo-WC.txt	Archivo generado por GMAT. Información de la posición del satélite, la órbita y el sol para el Peor Caso (con un step menor)	<table><tr><th>Sat.UTCGregorian</th><th>Sat.ElapsedSecs</th><th>Sat.Earth.LST</th><th>Sat.EulerAngle1</th><th>Sat.EulerAngle2</th><th>Sat.EulerAngle3</th><th>Sat.EulerAngle4</th><th>Sat.EulerAngle5</th></tr><tr><td>12 Dec 2011 00:00:00.000</td><td>0</td><td>247.41</td><td>67.234</td><td>-18.066</td><td>114.4</td><td></td><td></td></tr><tr><td>-2.7029e+07</td><td>-2914.4</td><td>-6944.6</td><td>-2456.7</td><td>1545.8</td><td>-0.53319</td><td></td><td></td></tr><tr><td>12 Dec 2011 00:00:30.000</td><td>30</td><td>248.9</td><td>68.721</td><td>-18.699</td><td>113.93</td><td></td><td></td></tr><tr><td>-2.7028e+07</td><td>-2723.2</td><td>-6992.2</td><td>-2539.8</td><td>1546</td><td>-0.53326</td><td></td><td></td></tr><tr><td>12 Dec 2011 00:01:00.000</td><td>60</td><td>250.39</td><td>70.219</td><td>-19.321</td><td>113.44</td><td></td><td></td></tr><tr><td>-2.7027e+07</td><td>-2530.1</td><td>-7034.7</td><td>-2621</td><td>1546.2</td><td>-0.53331</td><td></td><td></td></tr><tr><td></td><td>7998.2</td><td>2.9604</td><td>7118.6</td><td>30.028</td><td>0.0095327</td><td></td><td></td></tr></table>	Sat.UTCGregorian	Sat.ElapsedSecs	Sat.Earth.LST	Sat.EulerAngle1	Sat.EulerAngle2	Sat.EulerAngle3	Sat.EulerAngle4	Sat.EulerAngle5	12 Dec 2011 00:00:00.000	0	247.41	67.234	-18.066	114.4			-2.7029e+07	-2914.4	-6944.6	-2456.7	1545.8	-0.53319			12 Dec 2011 00:00:30.000	30	248.9	68.721	-18.699	113.93			-2.7028e+07	-2723.2	-6992.2	-2539.8	1546	-0.53326			12 Dec 2011 00:01:00.000	60	250.39	70.219	-19.321	113.44			-2.7027e+07	-2530.1	-7034.7	-2621	1546.2	-0.53331				7998.2	2.9604	7118.6	30.028	0.0095327																										
Sat.UTCGregorian	Sat.ElapsedSecs	Sat.Earth.LST	Sat.EulerAngle1	Sat.EulerAngle2	Sat.EulerAngle3	Sat.EulerAngle4	Sat.EulerAngle5																																																																																			
12 Dec 2011 00:00:00.000	0	247.41	67.234	-18.066	114.4																																																																																					
-2.7029e+07	-2914.4	-6944.6	-2456.7	1545.8	-0.53319																																																																																					
12 Dec 2011 00:00:30.000	30	248.9	68.721	-18.699	113.93																																																																																					
-2.7028e+07	-2723.2	-6992.2	-2539.8	1546	-0.53326																																																																																					
12 Dec 2011 00:01:00.000	60	250.39	70.219	-19.321	113.44																																																																																					
-2.7027e+07	-2530.1	-7034.7	-2621	1546.2	-0.53331																																																																																					
	7998.2	2.9604	7118.6	30.028	0.0095327																																																																																					
name-resumen.txt	Datos de la corrida del programa de cosenos	<div>Parámetros de la simulación:</div> <div>- Epoch = '15 Jan 2024 12:00:00.000'</div> <div>- SMA = 6991.1</div> <div>- ECC = 0.0018616</div> <div>- INC = 98.017</div> <div>- RAAN = 145.4</div> <div>- AOP = 39.257</div> <div>- TA = 170.03</div> <div>- AttitudeReferenceBody = Earth</div> <div>- BodyAlignmentVectorX = 1</div> <div>- BodyAlignmentVectorY = 0</div> <div>- BodyAlignmentVectorZ = 0</div> <div>- AttitudeConstraintType = Velocity</div> <div>- BodyConstraintVectorX = 0</div> <div>- BodyConstraintVectorY = 0.1391</div> <div>- BodyConstraintVectorZ = 0.99</div> <div>- Duración = 0.5672604482036532 días</div> <div>- Step = 0.0003472222222222222 días</div> <div>Resultados WC:</div> <div>- Cantidad de órbitas simuladas = 9</div> <div>Parámetros de la simulación WC:</div> <div>- BodyConstraintVectorX = 0</div> <div>- BodyConstraintVectorY = 0</div> <div>- BodyConstraintVectorZ = 0</div> <div>- AttitudeConstraintType = Velocity</div> <div>- Duración = 367 días</div> <div>- Step = 0.5 días</div> <div>Resultados:</div> <div>- Periodo = 5811.30272479564 segundos</div> <div>- Cantidad de órbitas simuladas = 5450</div> <div>- Paneles solares:</div> <div>- Panel 1: [0, 1, 0]</div>																																																																																								
cosenos.txt	Archivos de texto plano que contienen una lista de datos de cada step. Se enlistan la fecha, distancia, y cosenos/ángulos de	Fecha Distancia cosCara_+X cosCara_+Y																																																																																								
cosenos-WC.txt		cosCara_-X cosCara_-Y																																																																																								
cosenos-periodoWC.txt																																																																																										

	cada cara y panel desplegable, si lo hubiera. ⁶	2023-01-16 19:25:16 147159766.47711518 -0.8048063268341425 0.485065085043930260.342050638881233770.8048063268341425 -0.48506508504393026 -0.34205063888123377
angulos.txt		Fecha Distancia angCara_+X angCara_+Y angCara_+Z angCara_-X angCara_-Y angCara_-Z
angulos-WC.txt		2023-01-16 19:24:16 147159410.78632966 141.3638283875238 60.982617966124174 66.851190538373 38.63617161247622 119.01738203387583 113.14880946162701
angulos-period WC.txt		2023-01-16 19:24:46 147159764.35403195 142.502713145064 60.984036868716444 68.42073441738249 37.497286854936 119.01596313128357 111.57926558261751 2023-01-16 19:25:16 147159766.47711518 143.59155539035336 60.98326330871279 69.99814058671022 36.40844460964661 119.01673669128722 110.0018594132898 2023-01-16 19:25:46 147160131.68015635 144.627067344974 60.98280526941679 71.58170258735998 35.372932655026005 119.0171947305832 108.41829741264

Tabla 2. Resultados del programa

8. Apéndice

a. Apéndice A: Definiciones de mecánica orbital

Las normales a los paneles solares (en rojo en la Fig. 7) se ingresan en base del sistema body (en negro en la Fig.7). Las mismas se definen como

$$\bar{n}_f = a_{f1} \cdot \hat{i}_B + a_{f2} \cdot \hat{j}_B + a_{f3} \cdot \hat{k}_B$$

siendo $\hat{i}_B = [1, 0, 0]$, $\hat{j}_B = [0, 1, 0]$, $\hat{k}_B = [0, 0, 1]$ 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.

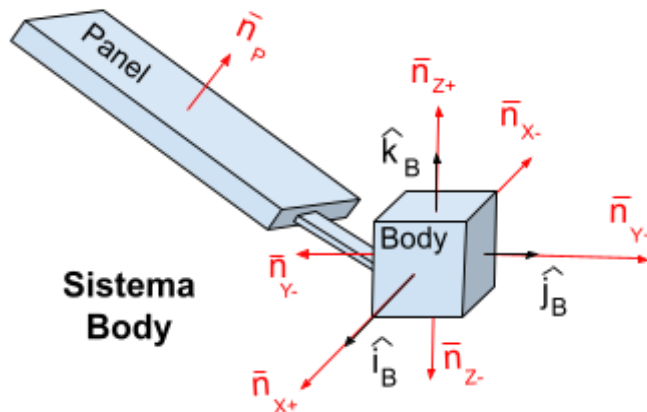


Figura 7. Sistema de referencia Body (negro) y vectores normales a los paneles solares.

Finalmente, el ingreso de los paneles queda: `paneles = [[ap11, ap12, ap13], [ap21, ap22, ap23], ...]`.

⁶ Las columnas están separadas por tabs ("t" en Python).