

# `orbit_calc` 2.0: manual de usuario

Fernando Mazzone

11 de octubre de 2011

## Índice

<b>1. Descripción General</b>	<b>1</b>
<b>2. Requerimientos</b>	<b>3</b>
<b>3. Diferencias con versiones anteriores</b>	<b>3</b>
<b>4. Entorno de MATLAB</b>	<b>5</b>
<b>5. Instalación y configuración</b>	<b>6</b>
<b>6. La ventana Efemérides</b>	<b>9</b>
<b>7. La ventana filtro</b>	<b>15</b>
<b>8. Ejemplos avanzados</b>	<b>17</b>
8.1. Ciclo de los nodos de La Luna . . . . .	17
8.2. Graficando una órbita . . . . .	18
8.3. Hacer una animación de Troyanos de Jupiter . . . . .	18
<b>9. ¿Quiere Colaborar con <code>orbit_calc</code> 2.0?</b>	<b>19</b>

## 1. Descripción General

El propósito fundamental del paquete de funciones para MATLAB `orbit_calc` 2.0 es que sirva como plataforma de cálculos numéricos en el

área de la mecánica celeste. El proyecto está abierto a colaboración y sugerencias. El código fuente será mantenido en la página web del Observatorio Astronómico Salvador, dentro de la sección Programas. En la sección Simulaciones Numéricas de la aludida página, se publicaran resultados de simulaciones realizadas con `orbit_calc 2.0`. Invito a colaborar, a aquellos que lo deseen, enviándome sus trabajos, ya sea resultados de simulaciones y/o nuevas funciones o scripts que expandan las capacidades de `orbit_calc 2.0`. Mi dirección de email está en la sección contacto de la página del Observatorio Astronómico Salvador. Mantendré allí un repositorio de archivos. Si Ud. quisiese mantener sus archivos en otro sitio, agradecería que se contacte para publicar el enlace a dicho lugar.

Sientasé libre de usar y/o modificar el programa a su gusto. La única condición es que Ud. site la fuente de estas modificaciones y que no incluya este código en un programa comercial.

Las funciones y componentes más importantes que constituyen el núcleo `orbit_calc 2.0` son:

1. `fuerza_nb.m`. Cálcula todas las fuerzas que actúan sobre los cuerpos de un sistema planetario dadas las posiciones de los mismos. Las posiciones puede ser un arreglo tridimensional donde las filas son los cuerpos, las columnas las coordenadas y la tercera dimensión son épocas distintas. Se asume a los planetas como puntos masa y las únicas fuerzas consideradas son las gravitatoria newtonianas, asumiendo a los cuerpos como puntos masa. La función está completamente vectorizada, no usa bucles ni ciclos, y optimizada para no hacer cálculos redundantes. Acepta una cantidad ilimitada de cuerpos, ya sea cuerpos graves o partículas test no graves. Las características del sistema planetario son recibidas por la función como un argumento. Este argumento indica cuales cuerpos son graves y que masa tienen.
2. `sistema.m`. Esta función construye los parámetros que recibe `fuerza_nb.m`. Esta diseñada para incluir los 8 planetas del sistema solar, más La Luna y Plutón. Acepta como argumento un número entero que indica la cantidad de cuerpos no graves que tendrá el sistema. Así `sistema(209)` devuelve los parámetros y condiciones iniciales para el sistema de 8 planetas, La Luna, Plutón y 209 cuerpos no graves (asteroides). `sistema.m` puede ser fácilmente modificada para construir otro tipo de sistema planetario, no necesariamente el solar.
3. **Integradores de las Ecuaciones de Movimiento.** Las ecuaciones que gobiernan la dinámica de sistemas planetarios son ecuaciones diferenciales que

hay que resolver por métodos numéricos. En el caso que nos ocupa estas ecuaciones se llaman ecuaciones de los n-cuerpos. MATLAB ofrece la ODE suite que permite integrar ecuaciones diferenciales. No obstante yo preferí programar mis propios integradores. Se pueden elegir dos: un método de colocación de paso variable (`colocacion_nb_adap.m`) y un método multipaso implícito de paso fijo (`multipaso_nb_implicito.m`). Estos integradores se pueden utilizar para integrar las ecuaciones de movimiento de los n-cuerpos o cualquier otro sistema de ecuaciones diferenciales autónomas de segundo orden.

hyphenationpre-diccionario

**Es oportuno decir que la capacidad de `orbit_calc 2.0` de leer el catálogo de estrellas UCAC3 se obtuvo de la página de Project Pluto desde donde se bajó código C++ para el propósito descripto.**

## 2. Requerimientos

1. `orbit_calc 2.0` está escrito en un lenguaje interpretado. Esto implica que para poder ejecutarlo Ud. debe tener instalado en su computadora un programa interprete de las instrucciones contenidas en `orbit_calc 2.0`. En este caso el programa interprete es Matlab versión 2010. Las predicciones de impactos de asteroides con La Tierra son representadas por mapas, para que estas funciones sean operativas se debe tener instalado la Mapping Toolbox para MATLAB.
2. El programa necesita algunas bases de datos. MPCORB.dat, COMET.dat, lista de códigos de observatorios y base de estrellas UCAC3. Las tres primeras bases son obtenidas desde internet, sólo haga click en los enlaces de arriba. Es de destacar que `orbit_calc 2.0` incluye una manera fácil de bajar las bases, descomprimirlas y convertirlas a una base apropiada para el uso de `orbit_calc 2.0`. La base de estrellas UCAC3 se obtiene solicitándola al Observatorio Naval de Estados Unidos.

## 3. Diferencias con versiones anteriores

Esta es la segunda versión de `orbit_calc`, aquí enumero las diferencias más significativas con la versión anterior.

1. **Tiempo de ejecución.** `orbit_calc` 2.0 usa integradores más eficientes que se ejecutan más rápido. Como ejemplo concreto, usando el método de colocación como integrador `orbit_calc` 2.0 demora 12 seg en calcular la posición de Ceres para el 1/1/2050 00:00:00 (UT), mientras que `orbit_calc` 1.0 demora casi 1:30 min para el mismo cálculo. El método multipaso implícito (que siempre es más rápido) de `orbit_calc` 2.0, con un paso de un día, demora para el mismo experimento 9 seg.
2. El código de `orbit_calc` 2.0 está mejor estructurado de modo que es más sencillo mantenerlo o modificarlo para que efectue otras tareas. Además el mismo código calcula la efemérides de un objeto o una cantidad grande de objetos.
3. Muestra la posición de los asteroides en cartas de estrellas.
4. La interfaz para el usuario es más simple y se incluyen más controles a fin de evitar el ingreso de datos incorrectos.
5. `orbit_calc` 1.0 presentaba en algunas máquinas problemas de memoria cuando procesaba la base de datos MPCORB.dat, `orbit_calc` 2.0 procesa por lotes dicha base y se prevee que no presente los problemas mencionados.
6. `orbit_calc` 1.0 funcionaba bajo MATLAB R14 y `orbit_calc` 2.0 está testeado para MATLAB 2010a bajo windows 32 y 64 bit. La versión R14 de MATLAB tenía problemas en máquinas de 64 bits. Como un problema colateral `orbit_calc` 2.0 ya no funciona bajo MATLAB R14.
7. `orbit_calc` 2.0 utiliza más el entorno de MATLAB, `orbit_calc` 1.0 lo evitaba usar para comodidad del usuario. Pero esta metodología acota los potenciales usos del programa.
8. En la nueva versión las bases de datos se cargan al iniciar el programa y quedan residentes en memoria. Esto acelera la ejecución de los programas respecto a la versión anterior, dado que esta última cargaba en memoria las bases con cada ejecución de una acción. El precio que se paga es usar memoria para datos que quizás no se utilicen.

## 4. Entorno de MATLAB

Sería posible compilar `orbit_calc 2.0` de modo que directamente no requiera tener instalado MATLAB en la computadora y, en ese caso, se ejecutaría como cualquier programa de windows. No obstante esto limita considerablemente los potenciales usos de `orbit_calc 2.0`. Como usuario del programa, utilzo `orbit_calc 2.0` en dos niveles diferentes.

1. Como observador aficionado de cuerpos menores, generalmente utilizo las efemérides que genera `orbit_calc 2.0` para localizar asteroides y cometas en el cielo. Además utilizo `orbit_calc 2.0` para elegir los asteroides que necesitan astrometrías. Para este uso práctico, no importa mucho que `orbit_calc 2.0` pueda calcular efemérides para el año 10219, ni que pueda predecir impactos, ni graficar órbitas o analizar evolución de parámetros orbitales.
2. Como intersetado en la evolución de las órbitas de cuerpos menores puedo querer analizar las frecuencias de variación del semieje mayor de Ceres (a modo de ejemplo). Necesitaríamos generar una efemérides perturbada por 1000 años de Ceres (`orbit_calc 2.0` demora aproximadamente 2 min en este cálculo), luego obtener los elementos keplerianos y hacer un Análisis de Fourier de los mismos. `orbit_calc 2.0` y MATLAB poseen funciones que efectuan todo el postprocesado descripto.

`orbit_calc 2.0` posee tres interfaces gráficas que controlan todas las funciones involucradas en el uso del primer nivel descripto. De modo que si Ud. quiere usar `orbit_calc 2.0` a este nivel todo lo que necesitará saber es identificar la línea de comandos en el entorno MATLAB y recordar el nombre de las funciones que abren estas interfaces (Configuracion, Efemerides y filtro) para tippearlas en la línea de comandos. Incluir interfaces gráficas que efectuen todas las operaciones para el segundo nivel de usuario sería prácticamente imposible, dado la cantidad enorme posibilidades. De modo que aquellos que quieran utilizar `orbit_calc 2.0` a este segundo nivel deberían aprender algunas sentencias de MATLAB. El lenguaje que interpreta MATLAB es de muy alto nivel y, por consiguiente, es muy fácil de aprender. Además la mayoría de los cálculos se realizan sin siquiera programar un script o función, sino que se los puede hacer tipeando instrucciones en la línea de comandos.

La figura 1 muestra el entorno MATLAB 2010a. Como se puede observar, consta de varias ventanas. La más importante de todas ellas es la ventana de línea

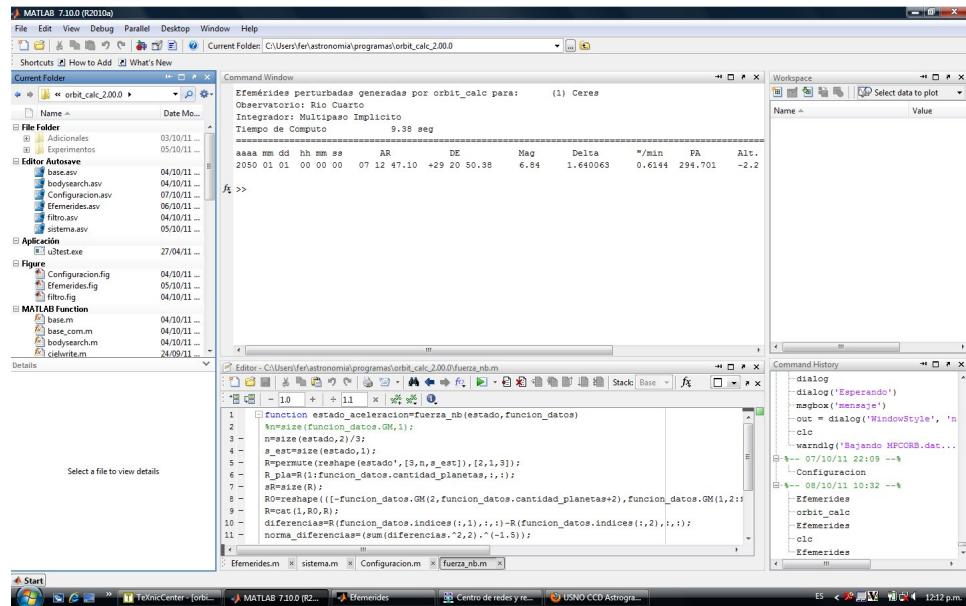


Figura 1: Entorno Matlab

de comandos “command window”. En esa ventana se ingresan comandos, en particular los comandos para abrir las interfaces gráficas de `orbit_calc 2.0`. En la ventana de comandos Ud. también podrá ver resultados de ejecuciones de `orbit_calc 2.0`. La ventana “current folder” le indica el directorio que está usando MATLAB por defecto, la puede usar de manera similar al explorar de windows. En la ventana “Workspace” se visualizan variables y en la ventana “Command History” se visualiza la historia de las sentencias ingresadas en la línea de comandos, haciendo doble click en una sentencia de la historia se vuelve a ejecutar.

## 5. Instalación y configuración

Para instalar el programa lo único que debe hacer es descargar el archivo `orbit_calc2.zip` y descomprimirlo en su disco rígido en algun directorio  
`C:\MiDirectorio`

No obstante para que el programa funcione correctamente Ud deberá configurarlo. Para ello deberá abrir Matlab y situarse en el directorio

C:\MiDirectorio\orbit\_calc\_2.00.0

Este directorio debió haber sido creado cuando Ud. descomprimió el archivo orbit\_calc2.zip. Ahora debe tipear en la línea de comandos:

>>Configuracion

y luego oprimir la tecla enter. De esta manera se le abre la ventana que se muestra en la figura 2.

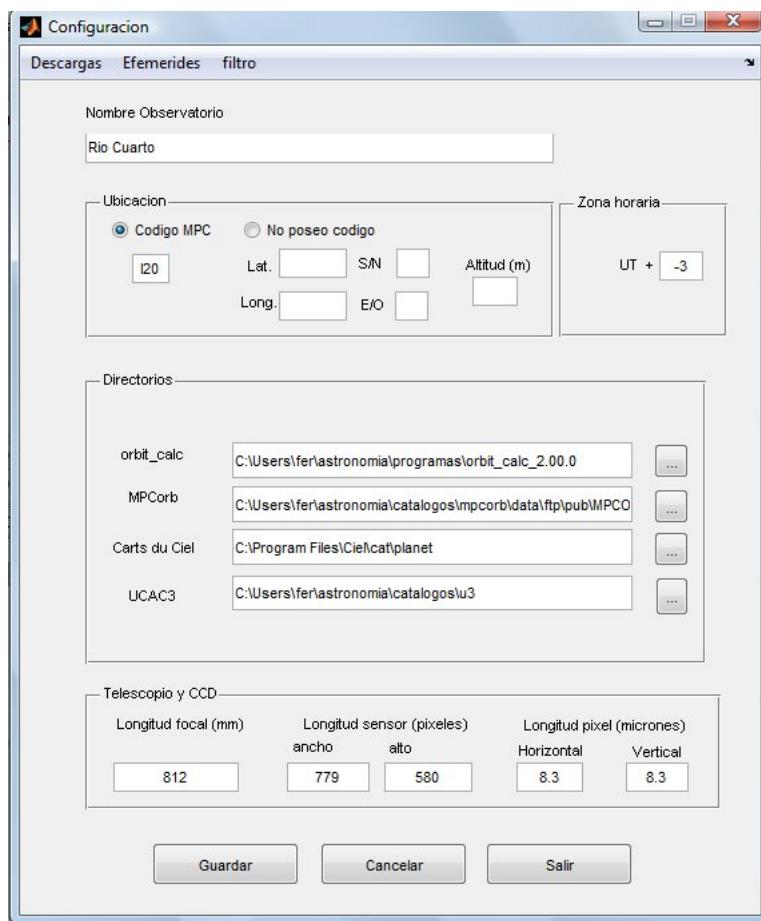


Figura 2: Ventana Configuración

En esta ventana Ud. deberá ingresar información básica de su sitio de observación y de la localización de bases de datos y otros programas con los que orbit\_calc 2.0 se puede comunicar.

El Centro de Planetas Menores (MPC) asigna un código a los observatorios de cuerpos menores. Si su lugar de observación tiene uno de estos códigos ingréselo en el campo “Código MPC”, orbit\_calc 2.0 leera en la base de datos de códigos de observatorios e identificará el nombre de su observatorio y posición sobre el globo terrestre. La base de datos de códigos de observatorio es provista con la distribución de orbit\_calc 2.0, no obstante se otorgan nuevos códigos regularmente, de modo que es buena política actualizar esta base periódicamente. Si Ud. no tiene un código MPC, tilde la opción “No poseo código” e ingrese su latitud, Norte o Sur, y su longitud, E o O y la altura a nivel del mar. En ambas situaciones, con o sin código, Ud. deberá ingresar su huso horario (campo Zona Horaria) ya que esto depende de la política del país donde reside y es imprevisible para orbit\_calc 2.0. Para que el programa funcione correctamente debe tener configuradas correctamente la fecha y hora de su computadora.

Seguidamente Ud. tendrá que indicar a orbit\_calc 2.0 donde se encuentran algunos directorios. En primer lugar donde se alojó orbit\_calc 2.0. En segundo lugar donde se alojan las bases de asteroides y cometas MPCorb.dat y Comet.dat respectivamente. Sin estas bases orbit\_calc 2.0 no funciona. Si Ud. ya posee MPCorb.dat y Comet.dat en algún directorio indique ese directorio en el campo MPCorb. Si no los posee elija un directorio donde quiera que orbit\_calc 2.0 los descargue.

orbit\_calc 2.0 puede comunicarse con el programa tipo planetario Carts du Ciel versión 2.7, de modo que de manera muy rápida Ud. puede transferir los resultados de una búsqueda de asteroides a Carts du Ciel. Para ello es necesario que Ud. indique a orbit\_calc 2.0, en el campo Carts du Ciel de Configuracion, donde está alojado el archivo asteroides.dat que usa Carts du Ciel. Si este campo no es llenado o es llenado incorrectamente, orbit\_calc 2.0 seguirá funcionando. En el campo UCAC3 Ud. deberá decir a orbit\_calc 2.0 donde aloja el catálogo UCAC3. Sin esta información, orbit\_calc 2.0 no podrá hacer cartas estelares.

Los restantes campos son parámetros de telescopios y CCD que los observadores conocen bien. Si Ud. no tiene telescopio, sólo deje los parámetros que la Configuración le muestra por defecto, *no los borre* pues el programa no funcionará correctamente.

Antes de salir oprima el botón guardar para que los cambios que haga tomen efecto.

Si Ud. no tiene las bases de datos de asteroides y cometas las deberá descargar. Para ello es suficiente que su computadora este conectada a internet e ir a Descargas en la barra de menús de la ventana Configuración y cliquear en MPCorb.dat

y luego en Comet.dat. **Atención:** MATLAB parece no prever barras de progreso de descargas, de modo que, cuando haga click en MPCorb.dat, orbit\_calc 2.0 solo le mostrará un mensaje de advertencia. Ud. no podrá saber cuanto de la descarga ha sido completado. El programa descarga el archivo MPCorb.dat.gz que es un archivo que supera los 32Mb de modo que la descarga puede demorar un tiempo prolongado, debe ser paciente. Luego de bajar las bases, el programa las descomprime automáticamente y la convierte en bases apropiadas para orbit\_calc 2.0. Luego de ejecutado este último paso el programa imprime en la línea de comandos información estadística, por ejemplo cuantos asteroides, cometas, NEOs y TNOs procesó, entre otras. De manera alternativa, Ud. puede bajar las bases por su cuenta, si procede así deberá indicarle a orbit\_calc 2.0 que convierta estas bases a la base apropiada para su funcionamiento, esto se consigue tipeando

```
>>base  
y  
>>base_com
```

en la línea de comandos. Estas funciones (base y base\_com) construyen la base que necesita orbit\_calc 2.0.

## 6. La ventana Efemérides

La ventana Efemerides permite generar efemérides de asteroides o cometas. Ud puede solicitar efemérides de un cuerpo, en cuyo caso orbit\_calc 2.0 lo buscará en las bases de datos, o indicar a orbit\_calc 2.0 alguna base compatible, ingresando el nombre de un archivo. En esta última situación orbit\_calc 2.0 generará efemérides para todos los cuerpos en la base asumiendo que ellos son particulass test no graves. Se accede a la ventana en cuestión tipeando

```
>>Efemerides
```

desde la línea de comandos (note que no hay que usar acentos). Se accede también a esta ventana desde la barra de menús de las otras ventanas: Configuracion y filtro.

La generación de una efemérides es muy sencilla. Note que el cuadro “Epoca inicio” aparece lleno con la fecha y hora (UT) que orbit\_calc 2.0 obtiene de su computadora. Si Ud. desea que la primer época de su efemérides sea otra, llene los campos de fecha y hora con los valores deseados. Recuerde que siempre orbit\_calc 2.0 entenderá que estos valores corresponden a la escala de tiempo universal (UT), esto es su hora local más el valor de zona horaria que Ud.

ingresó en la configuración. Las restantes épocas de las efemérides, se especifican en el cuadro “Epochas Efemerides”. Allí Ud. indica cuantas épocas distintas tendrá su efemérides y cuanto estarán separadas estas épocas. Para esto último, debe especificar la separación y elegir la unidad de tiempo deseada.

En el cuadro “Método de Integración” se elige el método que `orbit_calc` 2.0 usará para integrar las ecuaciones de movimiento. Esta elección requiere un cierto conocimiento técnico. Los valores que están establecidos por defecto funcionan correctamente en la mayoría de los casos. No obstante si Ud. quiere experimentar con los resultados que se obtienen modificando estos valores, tome en cuenta que:

1. La forma de integrar las ecuaciones de movimiento para modelos que incluyen perturbaciones (efemérides perturbadas) es a través de métodos aproximados. En esencia un método parte de una condición inicial conocida del sistema (este es el fin de las bases de datos) y de a pequeños pasos de tiempo intenta llegar a la posición que Ud. le solicitó.
2. Cuanto más pequeño es el paso más exacto y lento es el método.
3. El método multipaso es más rápido que el método de colocación
4. En contrapartida, el método de colocación es adaptativo, esto implica que es capaz de modificar la longitud de los pasos. Este aspecto es muy importante.
5. Cuando más grande es el orden más preciso será el método.
6. Estos son lineamientos generales, no son exactamente ciertos en toda ocasión.

Antes de describir el cuadro “Opciones de Salida” describamos las opciones que tenemos para el campo donde se ingresa el objeto. Allí Ud. puede ingresar designaciones de asteroides y cometas. La designación que puede ingresar es la numeración de un asteroide, nombre, designación compacta o designación provisional. Ejemplos:

1. Ingresando “2P” obtendrá efemérides para el cometa 2P/Encke.
2. Ingresando “1588” obtiene efemérides para (1588) Descamisada.
3. Al mismo resultado se llega ingresando “Descamisada”

4. Ingresando K10T07K se obtienen efemérides para el asteroide no numerado 2010 TK7. Alternativamente puede ingresar su designación provisional 2010 TK7.
5. Ingresando “Sistema Solar” orbit\_calc 2.0 efectua una integración del sistema solar sin ningún asteroide o cometa.

También puede elegir un archivo usando el botón archivo “Archivo” al costado del campo de objetos. Este archivo debe tener un formato válido. Los formatos válidos son:

1. Archivo de texto \*.txt o \*.dat con el formato descripto por el MPC. Este formato es estandard y otros programas (ejemplo find\_orb) producen archivos con este formato.
2. Archivos MAT de matlab \*.mat. En este caso el archivo debe contener una estructura llamada “cuerpo”. Esta estructura tiene los campos cuerpos.a, cuerpos.e, etc. El programa filtro de orbit\_calc 2.0 produce este tipo de archivos. Con filtro Ud. podrá generar un archivo \*.mat con asteroides seleccionados por clase, elementos keplerianos, etc. Luego de ello podrá estudiar la evolución de todos ellos generando una efemérides.

En el cuadro “Opciones de Salida” se le indica al programa que variables y graficos retornará luego de su ejecución. Las opciones de salida son:

Tildando Datos Observacionales se mostrarán en pantalla posiciones en coordenadas esféricas J2000, además de otros parámetros útiles para el observador, como se muestra en la figura 3.

```
Command Window
Efemérides perturbadas generadas por orbit_calc para: 2010 TK7
Observatorio: Rio Cuarto
Integrador: Colocacion Adaptativo
Tiempo de Computo 0.79 seg
=====
aaaa mm dd hh mm ss AR DE Mag Delta " /min PA Alt. Az. E.Luna E. Sol Fase
2011 10 09 20 08 00 06 37 54.28 -47 10 05.56 21.33 0.388220 1.5723 108.774 -6.9 16.0 103.8 89.0 69.68
2011 10 09 20 09 00 06 37 54.42 -47 10 06.06 21.33 0.388218 1.5725 108.781 -7.0 15.8 103.8 89.0 69.68
2011 10 09 20 10 00 06 37 54.57 -47 10 06.57 21.33 0.388216 1.5727 108.787 -7.1 15.7 103.8 89.0 69.68
2011 10 09 20 11 00 06 37 54.72 -47 10 07.08 21.33 0.388214 1.5730 108.794 -7.1 15.5 103.8 89.0 69.68
2011 10 09 20 12 00 06 37 54.86 -47 10 07.58 21.33 0.388212 1.5732 108.800 -7.1 15.3 103.8 89.0 69.68
2011 10 09 20 13 00 06 37 55.01 -47 10 08.09 21.33 0.388210 1.5734 108.806 -7.2 15.2 103.8 89.0 69.68
2011 10 09 20 14 00 06 37 55.15 -47 10 08.60 21.33 0.388208 1.5736 108.813 -7.3 15.0 103.8 89.0 69.68
2011 10 09 20 15 00 06 37 55.30 -47 10 09.11 21.33 0.388206 1.5738 108.820 -7.3 14.8 103.8 89.0 69.68
2011 10 09 20 16 00 06 37 55.45 -47 10 09.61 21.33 0.388204 1.5740 108.826 -7.4 14.7 103.8 89.0 69.68
2011 10 09 20 17 00 06 37 55.59 -47 10 10.12 21.33 0.388202 1.5742 108.833 -7.4 14.5 103.8 89.0 69.68
f2 >>
```

Figura 3: Efemérides observacionales

Tildando Mapa de estrellas `orbit_calc` 2.0 generará un mapa de estrellas correspondiente a un rectángulo celeste de las dimensiones que se especifiquen en el campo “Tamaño”. La flecha puesta sobre el asteroide indica la dirección de movimiento del mismo. El mapa está orientado de la manera usual: izquierda-este, arriba-norte. La imagen intenta simular un placa obtenida a través de un cámara CCD, de allí que se observé con ruido. La escala de los ejes son minutos de arco, medidos desde la posición del asteroide. El tiempo al que corresponde la imagen es el de la primera época de las efemérides que Ud. solicite. Ver figura 4.

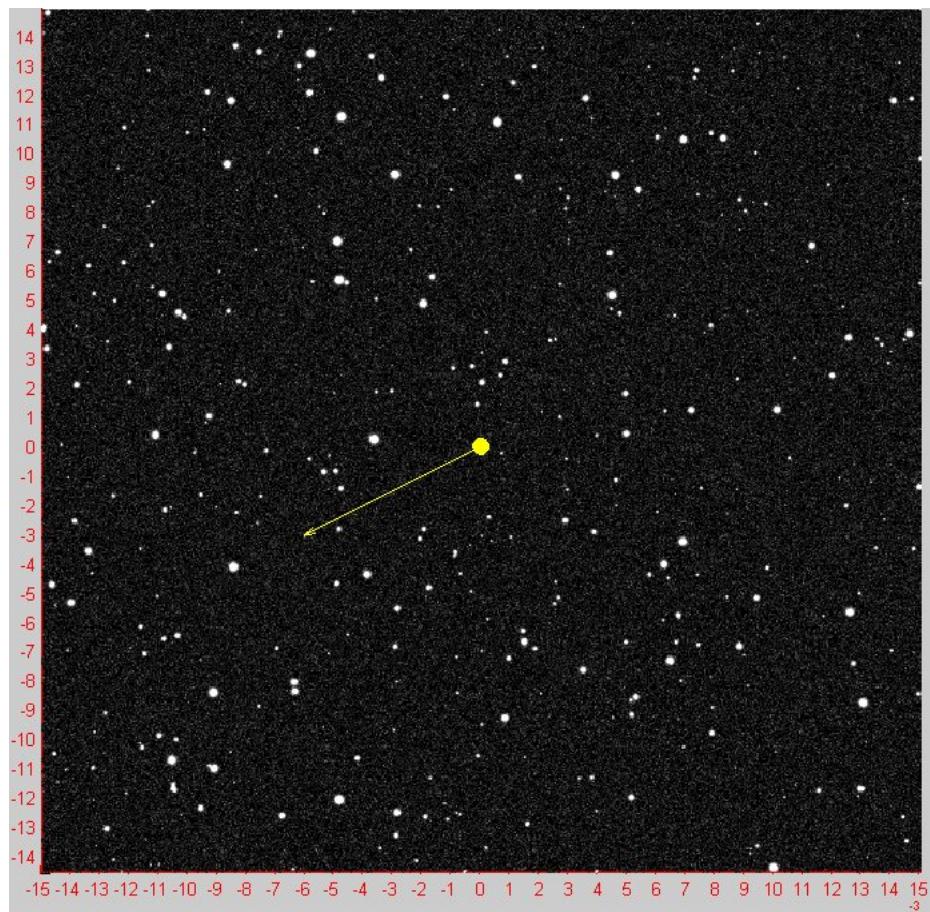


Figura 4: Mapas de Estrellas

Si Ud. tilda cualquiera de las opciones anteriores y está generando efemérides para un conjunto grande de asteroides ingresados a través de un archivo, `orbit_calc`

2 . 0 entenderá que Ud. está queriendo estudiar la incertidumbre de la posición en el cielo para algún asteroide y que los asteroides son clones de órbitas de ese asteroide. Una manera de generar esto es usando la función montecarlo de `find_orb`. Esta es la única situación que se me ocurre que alguien necesitaría conocer la posición en el cielo de muchos asteroides. Ante esta circunstancia, `orbit_calc` 2 . 0 analizará la distribución en el cielo de las posiciones de los asteroides. Si detecta que se distribuyen en una región de un diámetro mayor que  $20^\circ$ , terminará el análisis y le informará que la región es mayor a  $20^\circ$ . El valor de  $20^\circ$  es fácilmente configurable, si alguien quiere considerar regiones más grandes comuníquese con quien escribe para saber como hacerlo. Si el diámetro de la región es menor a  $20^\circ$ , buscará la forma de cubrir de manera eficiente esa región con cuadros centrados en alguna de las posiciones de los asteroides. El tamaño de esos cuadros es el que Ud. informó en la Configuración de `orbit_calc` 2 . 0. El programa por último muestra una placa con dibujos de rectángulos que corresponden a las placas que Ud debería tomar para cubrir la región. En la línea de comandos `orbit_calc` 2 . 0 imprimirá las posiciones y demás parámetros y marcará con un asterisco \*, a los asteroides que son centro de los rectángulos.

En la figura 5, se muestra el análisis anterior para la cápsula de la JAXA Hayabusa. Con la instalación de `orbit_calc` 2 . 0 se incluye el archivo `Hayabusa.dat` en la carpeta “Experimentos”, que se crea en el directorio donde Ud. instaló `orbit_calc` 2 . 0. Con este archivo Ud. puede hacer el análisis anterior. La fecha de este análisis es 10/06/2010. Si Ud. usa otra fecha es probable que la región de incertidumbre exceda los  $20^\circ$ .

La opción de impacto se puede usar para predecir el impacto, o el sobrevuelo, de un asteroide con La Tierra. Los eventos de impacto son muy poco frecuentes, de hecho sólo hay un asteroide catalogado que ha impactado con La Tierra: 2008 TC3. El campo que está al lado y a la derecha de la checkbox “impacto” indica la altura sobre el nivel del mar que Ud. quiere que `orbit_calc` 2 . 0 registre como impacto. Si Ud especifica, por ejemplo, 100km `orbit_calc` 2 . 0 va a marcar en un mapa todos los puntos correspondientes a momentos en que la altura del asteroide estuvo entre 0km (impacto) y 100km. El modelo de La Tierra la presupone un elipsiode. Si Ud. especifica un número grande, digamos 10000 km, puede ocurrir en realidad que el asteroide no choque con La Tierra. Pero esto es útil para ver la parte de La Tierra que sobrevuela el asteroide. **`orbit_calc` 2 . 0 predice impactos sólo para las épocas de las efemérides que Ud. ha especificado.** Así si Ud. generó una efemérides y en ninguno de las épocas de estas efemérides el asteroide se encontró a una altura menor a la especificada en el campo de altura, `orbit_calc` 2 . 0 no marcará nada a pesar que el asteroide pueda in-

cluso chocar con La Tierra en algún momento intermedio a las épocas que Ud. especificó. Este inconveniente se puede sortear facilmente generando épocas muy juntas, digamos con saltos de 1 seg, en momentos que Ud ha identificado como aquellos donde el asteroide estuvo cerca de La Tierra. Por ejemplo: Si Ud genera una efemérides de un año de duración empezando el 1/1/2008 con saltos de 1 día para el asteroide 2008 TC3, Ud descubrirá mirando el valor de Delta (Delta=distancia observador-objeto) que 2008 TC3 estuvo muy cerca suyo el 07/10/2008. Luego puede generar una efemérides empezando el 6/7/2008 con saltos de 1 min abarcando 3 días (3 días=4320 min). Así descubrirá que 2008 TC3 estuvo extremadamente cerca suyo el 7/10/2010 entre las 2:40 y 2:55 (UT). En este punto es conveniente aumentar la precisión del método de integración (un impacto requiere muy buena precisión). Utilice el método de integración de colocación con una tolerancia de  $10^{-22}$ . Ahora genere una efemérides con salto de 1seg, empezando a las 2:40 del 7/10 por 15 min=900 seg y tilde la opción impacto, puede destildar la caja Datos Observacionales. Sea un poquito paciente, `orbit_calc 2.0` calcula la efemérides rápido, pero hacer el gráfico a la mapping toolbox de Matlab le costará un tiempo (varios segundos). Ud conseguirá los mapas de las figuras 6 y 7 (con zoom).

El impacto es marcado en el norte de Sudan y al norte del río Nilo. Puede comparar la predicción de `orbit_calc 2.0` con las predicciones de la NASA y con las imágenes en infrarrojo de la explosión tomadas por el satélite Meteosat 8 / EUMETSAT (ver figura8). La mayor diferencia es sobre la hora exacta del impacto. `orbit_calc 2.0` no está utilizando la mejor forma de convertir el tiempo de las efemérides a hora civil.

Podrá observar Ud. que en los mapas hay marcados impactos sobre el Océano Índico. Estos corresponden en realidad a cuando el asteroide saldría del globo terrestre. Es facil distinguirlos de los primeros por la hora que está impresa en los mapas.

Para predecir un sobrevuelos, considerar el caso del asteroide 2011 CQ1. Este asteroide nos sobrevoló el 04/02/2011. Genere una efemérides para él abarcando todo el día 04/02/2011 con saltos de un minuto y ponga la altura mínima a 13km. Obtendrá el mapa de la figura 9.

Si en lugar de un asteroide Ud ha ingresado un archivo con muchos de ellos, `orbit_calc 2.0` interpretará que Ud. esta queriendo hacer una análisis de incertidumbre de un impacto, y que los asteroides en el archivo son variantes de órbita de un mismo asteroide. En este caso `orbit_calc 2.0` solo buscará el primer momento en que los asteroides entraron en contacto con La Tierra (altitud=0) y marcará todos aquellos que lo han hecho. En la carpeta Experimentos, que se

instala dentro de la carpeta principal de `orbit_calc 2.0`, Ud. encontrará el archivo `Hayabusa.dat`. Allí hay 222 clones de órbita de la cápsula de la JAXA Hayabusa. Elija este archivo, marque la opción impacto y genere una efemérides con los siguientes parámetros: comienzo 2010 06 13 14:00:00 (UT), separación 10 seg, cantidad 720, método colocación adaptativo y tolerancia  $10^{-20}$ . Ahora sea más paciente con la construcción del gráfico y conseguirá la imagen de la figura 10.

En este caso, el impacto predicho esta algo lejos del que verdaderamente ocurrió. No obstante, las órbitas fueron obtenidas de muy pocas observaciones y el arco de ellas es sólo 45min. Para generar los clones de órbitas use `find_orb`. Si es oportuno decir que la predicción que se hace con `find_orb + orbit_calc` es similar a las que hicieron, desde las misma observaciones, Bill Gray y Aldo Vitagliano con `find_orb+Guide` y SOLEX respectivamente.

Las restantes opciones de salida en la ventana Efemerides, corresponden a posiciones en distintos marcos de referencia de todos los cuerpos intervinientes en la integración. Esto incluye al cuerpo/s que Ud indicó en el campo objeto, más todos los planetas, la luna y plutón. Estos datos no son mostrados en pantalla, `orbit_calc 2.0` creará una variable que Ud. podrá ver en la ventana del workspace. Más adelante en la sección de técnicas avanzadas vamos a discutir más este tema.

## 7. La ventana filtro

La ventana filtro (ver figura 11) no ha sufrido prácticamente ninguna modificación respecto a la versión anterior. De modo que en este punto voy a ser breve. Estas son las mayores diferencias apreciables por el usuario:

1. La ventana es ligeramente más simple.
2. filtro funciona más rápido dado que la base de datos de asteroides queda residente en memoria.
3. filtro muestra los resultados en la ventana de comandos de Matlab.

La función de filtrado de `orbit_calc 2.0` le permite extraer asteroides de la base de datos MPCorb.dat por diferentes criterios. A saber:

1. Posición en el cielo, distancia ángular a La Luna y El Sol y brillo del asteroide.

2. Clase orbital del asteroide, número de oposiciones, fecha de la última observación, arco que cubren las observaciones, parámetro de incertidumbre  $U$ .
3. Semieje mayor, excentricidad e inclinación de la órbita.
4. Lugar que ocupa los asteroides en MPCOrb.dat.

Creo que el funcionamiento es bastante transparente, de modo que no voy a ser extenso aquí. Sólo quería resaltar los siguientes puntos.

1. En el campo de oposiciones debe especificar números enteros. Estos representan cantidad de oposiciones.
2. En última observación Ud debe especificar (en años) dos números reales que representan, una cantidad de años mayor y una menor que han transcurrido a la última observación.
3. El campo  $U$  debe llenarse con un entero entre 0 y 9 inclusive.
4. Del mismo modo, arco se llena con enteros que representan días.
5. El semieje mayor, excentricidad e inclinación se llenan con números reales positivos.
6. Rango de designación procesa los asteroides en un cierto rango, digamos, por ejemplo aquellos que están entre la fila 400000 y 420000 de MPCOrb.dat.
7. La opción Carts du Ciel pone inmediatamente los resultados de la búsqueda en el programa Carts du Ciel 2.7. Ud. debería tener bien configurado la dirección de Carts du Ciel en la ventana Configuración.
8. La opción html genera una página web con enlaces que Ud puede usar rápidamente para consultar más información de los asteroides.
9. El botón guardar graba los elementos keplerianos de los asteroides encontrados en un archivo MAT. **Esta opción no toma en cuenta la región del cielo especificada.**

## 8. Ejemplos avanzados

En esta sección desarrollo algunos ejemplos de como usar `orbit_calc 2.0` para predecir el comportamiento dinámico y/o para realizar distintos gráficos. Es conveniente para entender estos ejemplos que Ud. lea como Matlab manipula arreglos multidimensionales.

### 8.1. Ciclo de los nodos de La Luna

El nodo ascendente de La Luna tiene tiene un movimiento de precesión el cual completa una vuelta entera cada 18.6 años aproximadamente. Esto Ud. lo puede averiguar rápidamente con `orbit_calc 2.0`. Tilde la opción “Coordenadas Baricéntricas” y genere una efemérides del sistema solar por 36525 días=100años (puede ser menos). Ud. podrá ver en el workspace tres nuevas variables: t tiempos en su efemérides, PosBari= posiciones de los cuerpos del sistema solar, VelBari=velocidades. PosBari es un arreglo tridimensional, la primera coordenada es el cuerpo. Los cuerpos del sistema solar se enumeran en `orbit_calc 2.0` con el siguiente orden: Sol, Mercurio, Venus, Tierra, Luna, Marte, Jupiter, Saturno, Urano, Neptuno, Plutón. Ocasionalmente puede faltar el Sol en alguna variable, por ejemplo en coordenadas heliocéntricas es irrelevante pues las coordenadas del Sol son (0, 0, 0). La Tierra es el cuarto cuerpo en PosBari, así, a las posiciones de La Tierra se acceden con

```
>>PosBari(4,:,:)
```

Si tiepea lo anterior en la línea de comandos tendrá que esperar que Matlab imprima las 36525 posiciones en La Pantalla. Esto no le servirá de nada, en su lugar tipee:

```
>>Tierra=squeeze(PosBari(4,:,:));
```

La función squeeze de Matlab elimina dimensiones de longitud 1. `PosBari(4,:,:)` representa un arreglo dimensional con un solo cuerpo (La Tierra) de modo que ya no es necesario mantenerlo en un arreglo tridimensional, la función squeeze elimina la primera dimensión que es redundante. El punto y coma final impide que Matlab muestre en pantalla los resultados de la operación. Haga lo mismo con la velocidad de La Tierra y repita todo con La Luna.

```
>> Tierra=squeeze(PosBari(4,:,:));
>> VTierra=squeeze(VelBari(4,:,:));
>> Luna=squeeze(PosBari(5,:,:));
>> VLuna=squeeze(VelBari(5,:,:));
```

Calcule la posición y velocidad de La Luna Relativa a La Tierra

```
>> PosRel=Luna-Tierra;
>> VelRel=VLuna-VTierra;
```

Vamos a necesitar las masas de La Tierra, esto lo obtenemos fácilmente con la función de `orbit_calc 2.0` sistema:

```
>> GM= sistema(0);
```

GM1 es una variable que contiene el producto de las masas de los planetas con la constante de gravedad universal de Newton. Nosotros necesitamos esto para La Tierra que es el cuarto cuerpo, es decir necesitamos GM(4).

Note en el Workspace de Matlab que PosRel y VelRel son variables de dimensión 3x36525. La próxima función que vamos a usar es una función específica de `orbit_calc 2.0` que convierte posiciones y velocidades en elementos keplerianos. Esta función asume que las coordenadas deben estar sobre la segunda dimensión, esto es necesitamos convertir los arreglos PosRel y VelRel a arreglos de 36525x3. La función que deseamos invocar no forma parte del núcleo básico de `orbit_calc 2.0`, por consiguiente vamos a tener que ir al directorio donde ella se aloja, que es el directorio Adicionales que está donde Ud. descomprimió `orbit_calc 2.0`. Una vez allí tipee.

```
>> [a, e, incli, node, peri, M]=vector2kepler([PosRel,VelRel],GM(4));
```

Ahora grafique los resultados con:

```
>> plot((t-t(1))/365.25,node);
```

obtendrá el gráfico de la figura 12. La cuenta  $(t-t(1))/365.25$  la hice para convertir días a años. Use la herramienta “Data Cursor” de la barra de menús de la figura en Matlab para medir el período.

## 8.2. Graficando una órbita

Si Ud quiere graficar una órbita puede proceder de la siguiente forma. Supongamos que el cuerpo es Apophis. Genere una órbita en el rango deseado, digamos desde el presente hasta el 2040, para la opción de salida elija coordenadas heliocéntricas. Luego tipee

```
>> Apophis=squeeze(PosHelio(12,:,:,:));
>> plot3(Apophis(1,:),Apophis(2,:),Apophis(3,:))
```

Se le va a crear la gráfica de la figura 13

## 8.3. Hacer una animación de Troyanos de Jupiter

Los asteroides Troyanos son interesantes de graficar en un sistema rotante respecto al cuerpo al que están asociados. Configure filtro para extraer todos los troy-

anos de Jupiter entre el rango 1 a 100000 y oprima el botón guardar y almacene esta pequeña base de asteroides (hay aproximadamente 900 asteroides) con algún nombre. En la carpeta experimentos yo puse una de muestra: JupTroy.mat. Ahora en la ventana Efemerides genere una efemérides, para este archivo, tildando la opción Marco Rotante y eligiendo a Jupiter para el marco. Una elección apropiada para las épocas son unos 20 años y con saltos de 20 días (a modo de ejemplo). Ud conseguirá en el workspace las variables t, PosRot y VelRot. En estas variables PosRot y VelRot hay almacenadas posiciones y velocidades de algunos cuerpos que no nos interesan. En particular los primeros 10 cuerpos en PosRot corresponden a planetas. En PosRot no aparecen las coordenadas del Sol ya que en este marco el origen del sistema está en el Sol. A nosotros el único planeta que nos interesa es Jupiter que ocuparía el lugar 6 y además nos interesan los restantes asteroides que van del lugar 11 al 947 (937 son los cuerpos del ejemplo que incluyo). La siguiente sentencia en la línea de comandos elegirá los indices para cuerpos de nuestro interés:

```
>> I=[6, 11:947];'
```

En I está el entero 6 y del 11 al 947. Ahora nos quedamos en PosRot con lo que nos interesa

```
>> PosRot=PosRot(I,:,:,:);
```

En el directorio Adicionales encontrará una función (llamada precisamente animacion) que realiza una animación. Es una función muy básica que incluyo como ejemplo. Ud puede mejorarla a su gusto. Situese en el directorio Adicionales y tipee

```
>> animacion(PosRot)
```

generara una animación como se muestra en la figura 14 que será guardada en un archivo \*.avi .

Los anteriores son unos pocos ejemplos. En la página del Observatorio Salvador Ud. puede apreciar más. Estaría gustoso de conocer otros experimentos que, si Ud. quiere, puedo publicar o crear el enlace dentro de la página del observatorio.

## 9. ¿Quiere Colaborar con `orbit_calc` 2.0?

Aquí van algunas cuestiones que me gustaría ver desarrolladas para `orbit_calc` 2.0. Primero enumero dos items que se podrían desarrollar con conocimientos de programación y al final uno más fuertemente matemático.

1. El software Aladin es, creo, la mejor herramienta para investigar el cielo. Este software acepta invocarlo mediante scripts. Sería interesante, en lugar

de que `orbit_calc 2.0` construya mapas de estrellas que directamente envie los datos a Aladin.

2. De la misma manera, Google Earth es una herramienta excelente para visualizar mapas. Sería interesante que en lugar de la Mapping toolbox Matlab construya archivos \*.kml y visualizar con Google Earth los impactos.
3. Sería interesante incorporar nuevos integradores a `orbit_calc 2.0`, en particular métodos simplécticos. Es de destacar que una variante del método de colocación que usa actualmente `orbit_calc 2.0` es simpléctico. La variante consiste en usar épocas de colocación dadas por los ceros de los polinomios de Legendre.

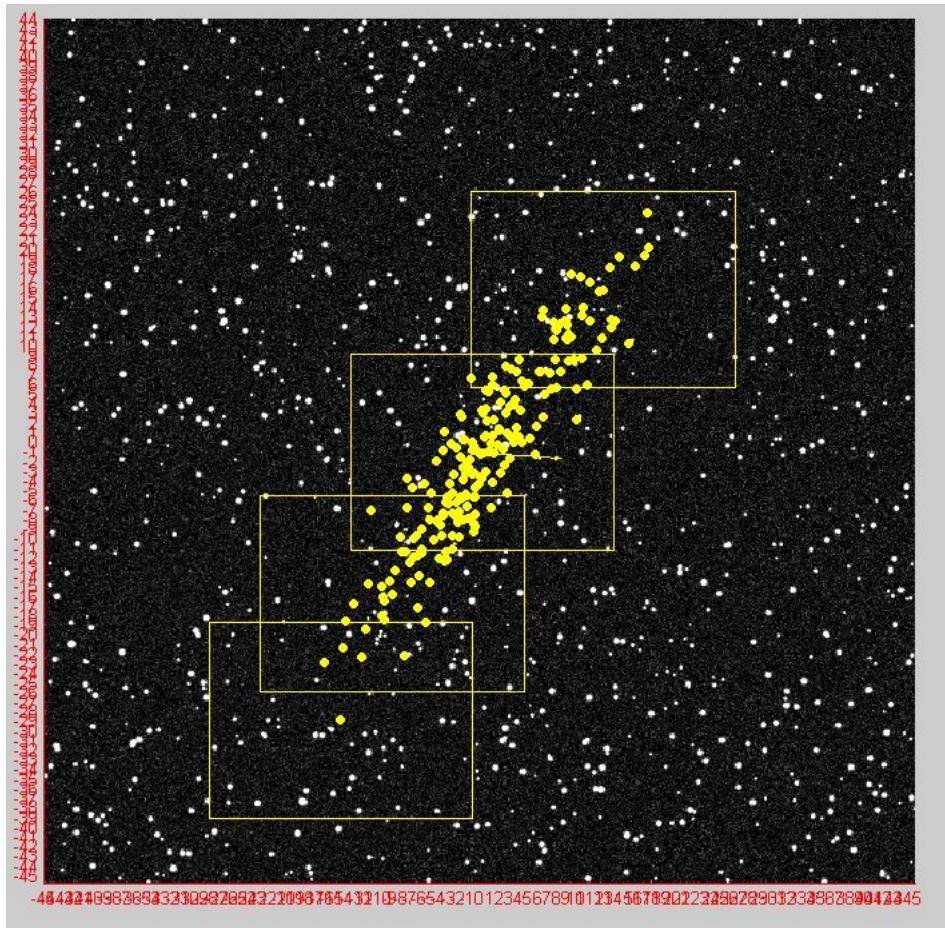


Figura 5: Mapa Incertidumbre

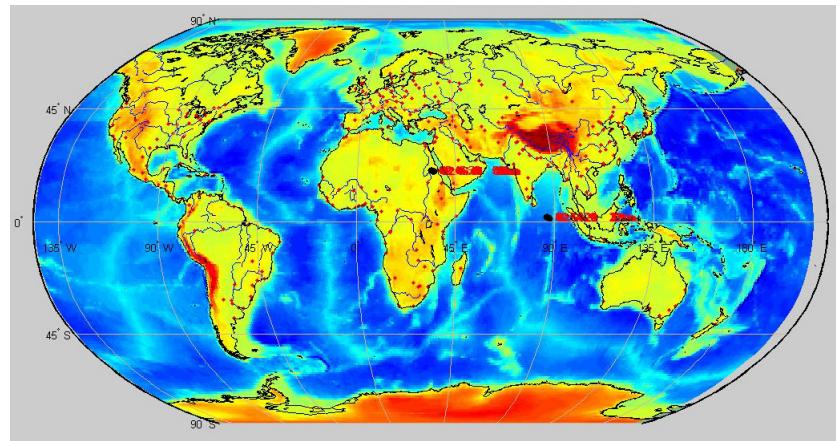


Figura 6: Impacto de 2008 TC3

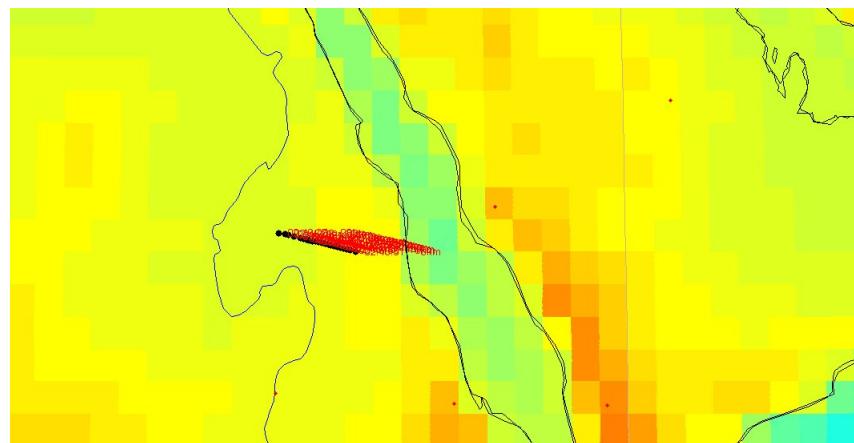


Figura 7: Impacto de 2008 TC3

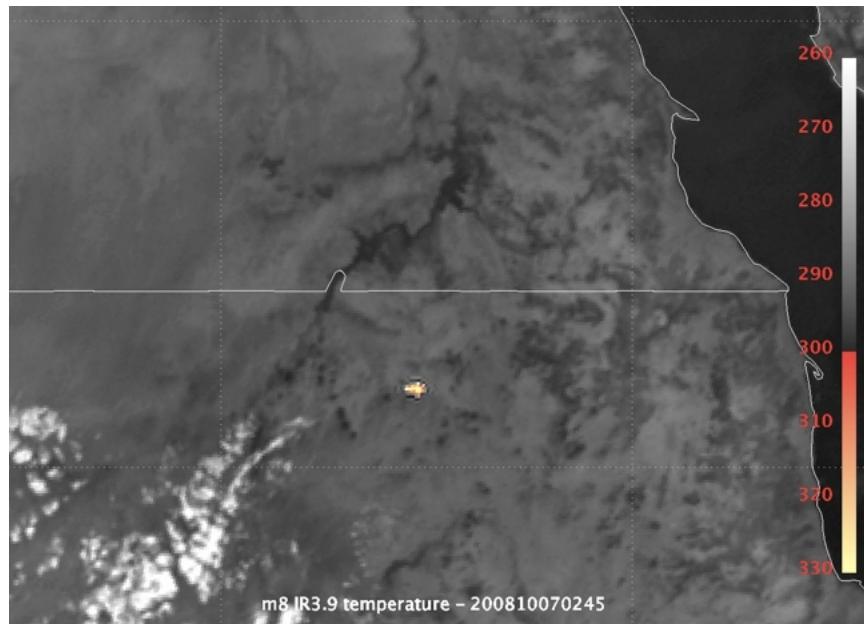


Figura 8: Impacto de 2008 TC3, Meteosat 8 / EUMETSAT

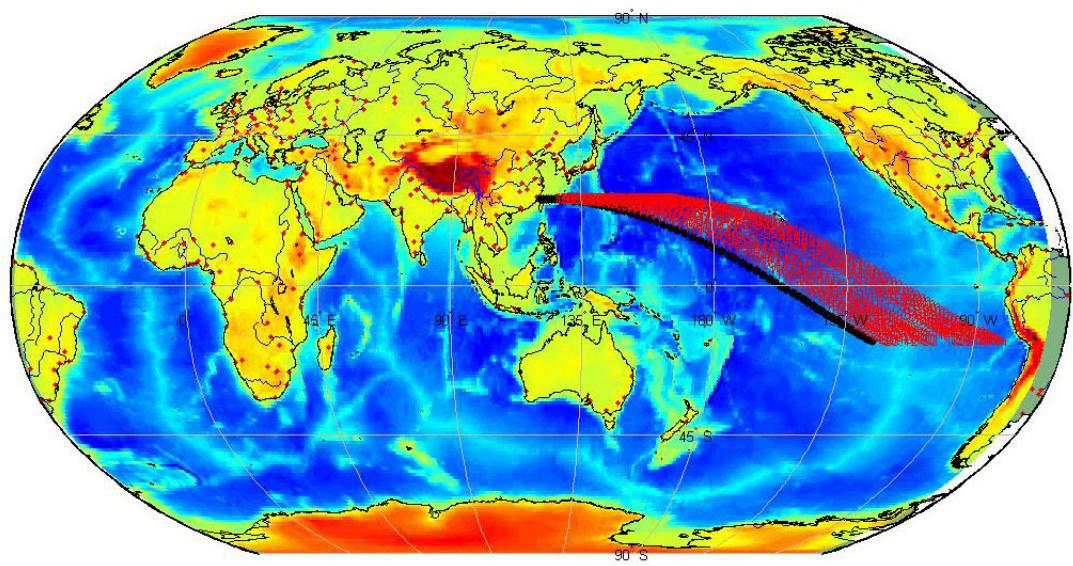


Figura 9: Sobrevuelo de 2011 CQ1

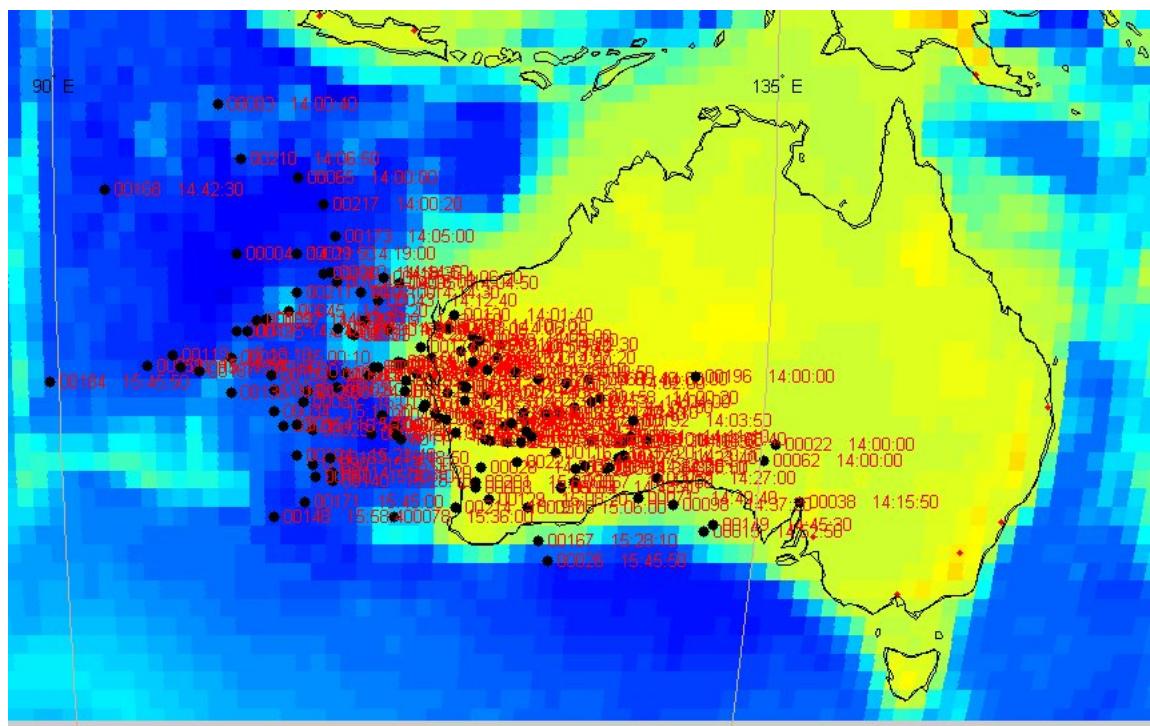


Figura 10: Impacto de Hayabusa

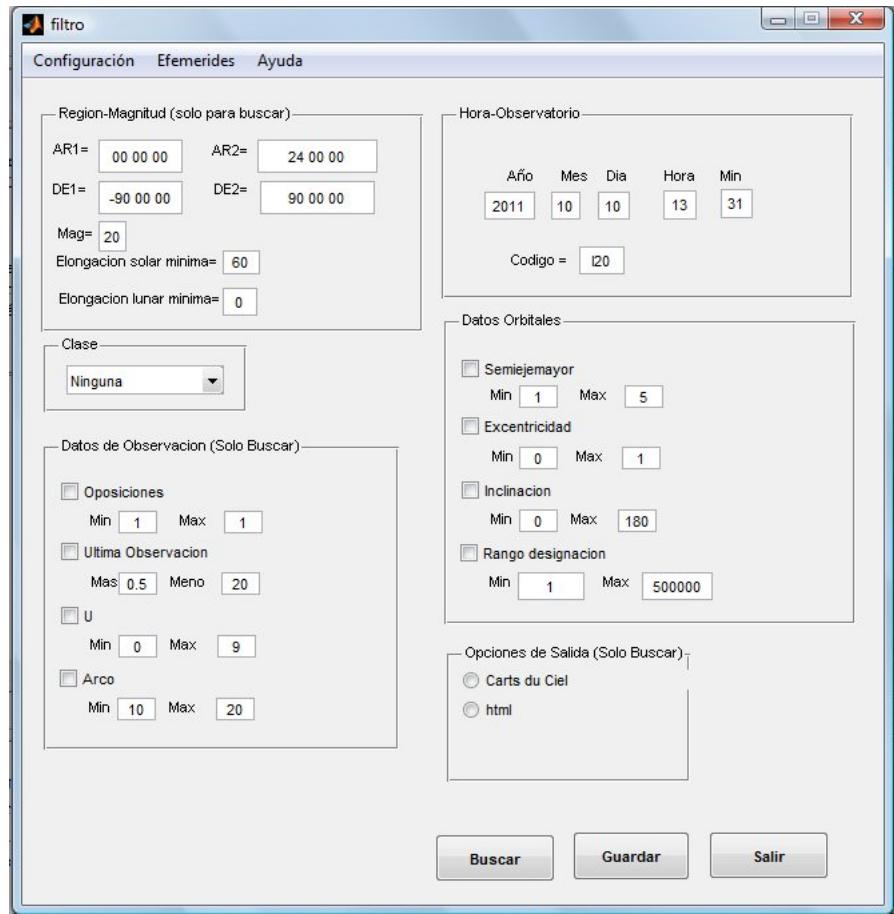


Figura 11: Ventana Filtro

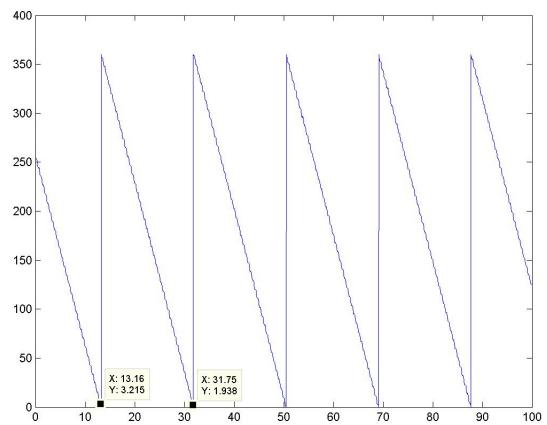


Figura 12: Precesión de los nodos de La Luna

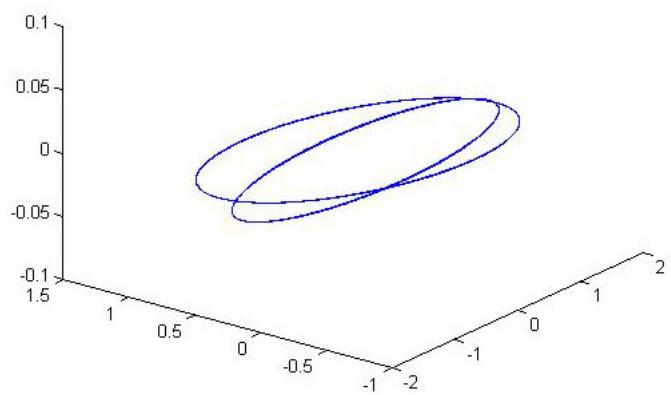


Figura 13: Órbita Apophis

Figura 14: Troyanos de Jupiter