

**Correcting the software of the 40cm astrograph control system at
Centro Astronómico de Yebes**

T. Alonso Albi

Informe Técnico IT-OAN 2008-6

Change Record

Version	Date	Author	Remarks
1.0	4-Jun-2008	T. Alonso Albi	First version
1.1	16-Jun-2008	T. Alonso Albi	Forgot to mention another change

Contents

1	Introduction	3
2	The control system software	3
2.1	The Aster program	3
2.2	The ephemeris flaw	3
3	Solution to the problem	4
3.1	Efficient computation of ephemeris	4
3.2	Swiss ephemeris library	5
4	Modifications to the control system	5
4.1	File swedll32.dll	5
4.2	File syssolar.bas	5
4.3	File Aster.frm	9
4.4	Other modifications and comments	9

1 Introduction

In this document some modifications to the software of the 40cm astrograph control system are described. This software had a known flaw that avoided the astronomer to properly use the telescope after March, 2007.

2 The control system software

In this section we will briefly describe the software of the control system and how to modify the code.

2.1 The Aster program

The control system software (hereafter **Aster** program) is written in Visual Basic 6.0 and runs on Windows 2000 in the leftmost computer at the control room of the astrograph. The software organization is quite disperse: many unnecessary files that reside on the same location as some critical ones. The easiest and safest solution was to modify the program in that same computer. We tried to install **Microsoft Visual Studio Enterprise 6.0** in a laptop running Windows XP, but after some necessary updates from Microsoft Update, the registration of some dynamic link (dll) libraries and a full software reinstallation (following certain Internet forums) it was not possible to properly install neither **Microsoft Visual Basic** nor the **Aster** project. A future development to fully redo the control system in some system-independent programming language is desirable.

The usage of Visual Basic in the computer next to the astrograph was more fluent. The **Aster** project can be opened from the recent projects tab, either in its new version or in the unmodified old one. The program is organized in two parts: a set of forms, composed of several visual windows with some background code, and a set of modules that contain mainly mathematical subroutines called by the code in the forms.

2.2 The ephemeris flaw

The ephemerides are computed by means of a non-standard/undocumented modification of the original JPL DE/LE200 [1] numerical integration files. The transformation program from the standard JPL files to this format, if it existed, seems to be lost, and the file present in **Aster** was prepared to perform ephemerides calculations from 1987 to 2006. At startup, the program tries to perform ephemerides calculations but the necessary Chebyshev coefficients are not available for current dates, and the program crashes.

The process of ephemerides computations starts by reading the coefficients in the ephemeris file. This reading process is done at startup and whenever the date is changed. Since it is required at startup, the only way to use the control system is to change the date of the computer (to year 2005 for example) prior to the execution of the software for these coefficients to be available. Obviously a change of about three years in the date produces severe limitations in the operation of the control system like the impossibility to automatically center planets in the

field of view, since the current sky at a given time differs from the same instant three years before. Even in fixed objects like stars the difference can easily exceed four minutes of time or one degree, which is by far greater than the pointing accuracy of the astrograph of about one arcminute.

After some transformations, the coefficients are stored in memory to be later used when necessary, for example when the astronomer selects the option Solar System sources and asks the program for a list of available sources of this kind.

It is important to note that this method is not efficient. The coefficients are stored in memory even when not needed, and are not read again until the date changes. Also there's a flaw in this process, since according to the code a change of month or year will not produce a new reading process call. Only a change in the date (day of month) triggers it.

3 Solution to the problem

In this section two possible solutions to the problem are first discussed. In section 3.2 we describe the implemented one.

3.1 Efficient computation of ephemeris

To compute ephemeris in a more efficient way it is desirable to implement an algorithm accurate enough and at the same time independent of any of the JPL numerical integration theories. The second condition is adequate since it avoids any input operations in the program, typically slower compared to maths, and reduces the amount of hard disk memory necessary for these calculations. A possible way to accomplish these requirements is to use orbital elements. It is the fastest and most efficient way, but requires these orbital elements to be included in a file for every one or several months and every year, which means that eventually the problem could arise again (although with an easy and standard update of the files). This method can easily be implemented to achieve accuracies better than 0.05 arcseconds.

As an alternative way, the most efficient known method developed so far is the Steve L. Moshier's semi-analytical theories [2]. His C++ original software is freely available since early 90's, and provides an excellent fit to JPL DE/LE404 numerical integration theory from 1000 B.C. to 3000 A.D, including the Sun, the Moon, Pluto and all the planets. The inaccuracy compared to the numerical integration is about 0.05 arcseconds at most, which means that is rather more accurate than the old JPL DE/LE200, even at current time (by some 0.1 arcsecond in the outer planets and specially in Pluto, and about one arcsecond in the Moon, increasing in an accelerated way at about 1 arcsecond/century²). The necessary coefficients are included in the original source code as static objects, but the amount of memory consumed is stable and not very high (500 kBytes). The speed of the calculations is not as high as in JPL theories, but this is not a problem nowadays. It should be noted that the old JPL DE/LE200 numerical integration is not suitable for high precision ephemerides required in telescope control systems or astronomical almanacs. The current state-of-the-art numerical integration is the JPL DE/LE414 (DE418 and DE421 partially available), whereas the one currently recognized by the IAU is JPL DE/LE405, which is the one used in the control system of the 40m radiotelescope.

Two solutions were considered for the astrograph control system, both implementing the Moshier solution to planetary motion. The first one is an already developed Java code (or the original C++ one), maybe using a temporary file to communicate the input/output data with the control system. This is not very efficient. The other solution came up when searching Internet for a more efficient and already available library for the same purpose for Windows systems. In this search the Swiss library appeared to be the right answer.

3.2 Swiss ephemeris library

The Swiss ephemeris library [3] is a freely/commercial available library that implements both the Moshier and the JPL DE/LE405 numerical integration theories. The free version can be used provided that the software is used privately (not available to others either in a free or commercial way) or with a public free license (publishing the source code). This library implements the Moshier theory as a backup method if the Swiss ephemerides files (corresponding to a non-standard JPL DE/LE405 numerical integration) are not available. The library comes with a simple example program written in Visual C++, Visual Basic, and some other programming languages.

4 Modifications to the control system

To apply the correction a testing program was created from scratch to see which steps are required to properly use the dynamic linked library for topocentric ephemerides calculations. Since it is 'dynamic', it is enough to include the .dll file in the same directory of the executable program. For topocentric calculations an additional call to certain subroutine is necessary in Swiss, so it was implemented and successfully tested. The modifications of the control system were applied after creating a backup copy of the old **Aster** project. These changes are described next.

4.1 File swedll32.dll

This file is the dynamic link library of the Swiss ephemeris. It is only necessary to copy the file to the development/execution directory (both are the same, located in C:/Aster and C:/AsterNuevo for old and new versions) to allow the program to find and use it at runtime.

4.2 File syssolar.bas

A new subroutine *SOLARSYS* was written to perform ephemerides calculations through the Swiss library. The new *SOLARSYS* subroutine is listed here.

```
Sub SOLARSYS(OBJETO As String, Altu As Double, Alfa As Double, Delta As Double, Alfap As Double,
Deltp As Double, Radio As Double, Distancia As Double)
,
```

```
' Calculo de la posicion de los planetas, Sol y Luna
```

```

,
' Tomas Alonso, CAY-IGN, mayo 2008
,
' Subrutina que emula la antigua subrutina SOLARSYS utilizando la libreria
' swiss, que hace uso de la dll swedl32.dll. Calculos basados en la teoria
' de efemerides de Steve L. Moshier, un ajuste a JPL DE/LE 404 con un error
' maximo de pocas centesimas de segundo de arco.
' Esta subrutina corrige el problema de las efemerides en fechas posteriores
' a marzo (?) del año 2007. No requiere ningun fichero adicional de efemerides
' (salvo por la dll), tiene una validez desde el año 1000 a. C. al 3000 d. C.,
' y sus calculos superan en precision a la teoria JPL DE/LE 200.
' Muchas funciones del programa ya no son necesarias, pero se ha preferido
' modificar el codigo original lo menos posible
,

```

```
Dim X(7) As Double
```

```
Dim tjd_ut As Double, tjd_et As Double
```

```
Dim CAL As Integer, retval As Integer, planet As Long, rflag As Long
```

```
Dim ret_flag As Integer
```

```
Dim serr As String
```

```
Dim longit
```

```
Dim latit
```

```
Dim altit
```

```
Dim Alfa_p As Double, Delta_p As Double
```

```
Dim H As Double, HP As Double, PAR As Double, Rr As Double
```

```
Dim C As Double, S As Double, N As Integer, I As Integer, l As Integer, Ier As Integer
```

```
Dim Cosd As Double, Sind As Double, Cosh As Double, Sinh As Double, Coshp As Double, ddd As Double
```

```
Dim Rsin As Double, Rcos As Double, Vsik As Double
```

```
Dim Lati As Double
```

```
Dim Tuniv As Double, LSTap As Double
```

```
H = U_T / 3600#
```

```
CAL = 103 ' g for gregorian calendar
```

```
tjd_ut = swe_julday(ano, mes, dia, H, 1)
```

```
retval = swe_date_conversion(ano, mes, dia, H, CAL, tjd_ut)
```

```
tjd_et = tjd_ut + swe_deltat(tjd_ut)
```

```
' If retval <> 0 Then
```

```
' Call MsgBox("Illegal Date", vbOKOnly)
```

```
' Exit Sub
```

```
' End If
```

```
Static RADIOS(10) As Double
```

```
RADIOS(1) = 2439#  
RADIOS(2) = 6052#  
RADIOS(3) = 696000#  
RADIOS(4) = 3397.2  
RADIOS(5) = 71398#  
RADIOS(6) = 60000#  
RADIOS(7) = 25400#  
RADIOS(8) = 24300#  
RADIOS(9) = 2500#  
RADIOS(10) = 1738#
```

```
If OBJETO = "MERCURIO" Then  
planet = SE_MERCURY  
l = 1  
ElseIf OBJETO = "VENUS" Then  
planet = SE_VENUS  
l = 2  
ElseIf OBJETO = "SOL" Then  
planet = SE_SUN  
l = 3  
ElseIf OBJETO = "MARTE" Then  
planet = SE_MARS  
l = 4  
ElseIf OBJETO = "JUPITER" Then  
planet = SE_JUPITER  
l = 5  
ElseIf OBJETO = "SATURNO" Then  
planet = SE_SATURN  
l = 6  
ElseIf OBJETO = "URANO" Then  
planet = SE_URANUS  
l = 7  
ElseIf OBJETO = "NEPTUNO" Then  
planet = SE_NEPTUNE  
l = 8  
ElseIf OBJETO = "PLUTON" Then  
planet = SE_PLUTO  
l = 9  
ElseIf OBJETO = "LUNA" Then  
planet = SE_MOON  
l = 10  
Else  
ler = 1  
Aux_Str = "Error en el Nombre de un cuerpo del Sistema Solar"
```



```

Activa_Mensajes ("ROJO")
Exit Sub
End If

serr$ = String(255, 0)
rflag = SEFLG_MOSEPH + SEFLG_EQUATORIAL ' + SEFLG_TOPOCTR
longit = LONGITUD# * 15# / 3600#
latit = LATITUD#
altit = ALTURA#

ret_flag = swe_calc(tjd_et, planet, rflag, X(1), serr$)

Alfa = X(1) * 3600# / 15#
Delta = X(2) * 3600#
Alfap = Alfa
Deltap = Delta
Distancia = X(3)
Radio = Atn(RADIOS(1) / (Distancia * 149597870.7)) * (180# / Pi) * 3600#

LSTap = LST / 3600#
Tuniv = U_T
Lati = LATITUD * GtR
C = Sqr(Cos(Lati) ^ 2 + ((1# - 1# / 298.25) * Sin(Lati)) ^ 2)
S = C * (1# - 1# / 298.25) ^ 2
Rsin = (S + 0.0000001568 * Altu) * Sin(Lati)
Rcos = (C + 0.0000001568 * Altu) * Cos(Lati)
Alfa_p = X(1) * GtR
Delta_p = X(2) * GtR

H = LSTap * 15 * GtR - Alfa_p
If H > Pi Then H = H - Pi * 2
If H < -Pi Then H = H + Pi * 2
Cosd = Cos(Delta_p)
Sind = Sin(Delta_p)
Cosh = Cos(H)
Sinh = Sin(H)
,
' CALCULO DE LA POSICION TOPOCENTRICA DEL OBJETO
,
' CALCULA PARALAJE DE LOS PLANETAS
' Y DE LA LUNA A PARTIR DE LA EXPRESION
' Sin(PP) = 6378.137 / R
' PAR = 6378.137 / Distancia / 149600000#
Rr = 1# / PAR

```

```

ddd = Atn(Rcos * Sinh / (Rr * Cosd - Rcos * Cosh))
Alfap = Alfa_p - ddd
HP = H + ddd
Coshp = Cos(HP)
Deltp = (Rr * Sind - Rsin)
Deltp = Deltp / (Rr * Cosd * Cosh - Rcos) * Coshp
Deltp = Atn(Deltp)
,
' TRANSFORMA ANGULOS A HORAS Y GRADOS
,
Alfa_p = Alfa_p * 180# / Pi / 15# * 3600
Alfap = Alfap * 180# / Pi / 15# * 3600
Delta_p = Delta_p * 180# / Pi * 3600
Deltp = Deltp * 180# / Pi * 3600
If Alfap < 0 Then Alfap = Alfap + 86400#
H = H * 180# / Pi / 15# * 3600#
HP = HP * 180# / Pi / 15# * 3600#
If H > 43200# Then H = H - 86400#
If H < -43200# Then H = H + 86400#
If HP > 43200# Then HP = HP - 86400#
If HP < -43200# Then HP = HP + 86400#
End Sub

```

4.3 File Aster.frm

This file is the main form of the program, with the design of a visual window and some background code to control its behaviour. No modifications are necessary in the window itself, only on its background. The call to subroutines *BARICENTRICAS* and *LECTURA_JPL2000* have been commented in the *Private Sub Tim_Timer* subroutine.

4.4 Other modifications and comments

The file **sysSolar.bas** now contains also a header of about 1000 lines of code that is needed to call the functions of the Swiss library. This header declares the functions available in the .dll file and provides some useful constants to properly use them.

Some other modifications were performed in **sysSolar.bas** and **astronom.bas**. The changes apply to the names of some subroutines (*POSICI*, *POSPLAN*, *VELOC*, *SOLARSYS* in **sysSolar.bas** and *BARICENTRICAS*, *LECTURA_JPL2000* in **astronom.bas**) related to ephemerides calculations, in which a suffix '_OLD' were added in their names. This was made to test that no call to an ephemeris algorithm is present in any of the other modules of the software, that would yield a compilation error. Another aim is to ensure that further modifications will not use those (now unnecessary and not recommended) subroutines to avoid new problems with ephemerides computations.

All those listed subroutines (and probably much more like precession or nutation corrections for example, up to 30%-40% of the original source code) are now unnecessary, but it seems preferably to keep them in the code to solve any issue that could arise when testing or reversing the changes. The chosen solution was constrained by the condition that code should be changed as little as possible.

The new *SOLARSYS* subroutine performs topocentric corrections using the same method as the old one, instead of obtaining directly topocentric coordinates from the Swiss library. This is due to some kind of error inside Swiss or between the library and Visual Basic, and the exact cause is unknown. This solution yields an error of 5 arcseconds in the position of the Moon, and a negligible one for the other Solar System bodies. The accuracy of the new implementation against the old one is similar, with a pointing improvement of 1-2 arcseconds in the Moon and of the 0.1 arcsecond range for the planets, as it was expected.

References

- [1] JPL numerical integration files available at <ftp://ssd.jpl.nasa.gov/pub/eph/planets/ascii>.
- [2] Programs by S. Moshier can be downloaded from <http://www.moshier.net>.
- [3] The Swiss ephemeris library is located at <http://www.astro.ch/swisseph>.