Open source Kālacakra calendar software

Modern astronomical calculations

Nālanda: The image above is of a mango tree in the ruins of Nālanda University, Bihar, India. This was the birth place of Śāriputra, one of the Buddha's two main students. When the Buddha visited this place, long before the founding of the university, the area was reputedly covered in mango trees. Kālacakra was certainly taught here, and the university housed Khagola, a famous astronomical observatory. One of the greatest of all the old Indian astronomers, Āryabhaṭa, born 476 CE, is known to have studied here and worked in the observatory. Whether in any sense the Kālacakra calendar was developed here we cannot say, but part of the calculations can be traced back to those in use by Varāha Mihira, a contemporary of Āryabhaṭa. It is thought that the modern term for astronomy in use in India, Khagola-shastra, derives from the name of Nālanda's observatory.

On this page is made available open source software for the Kālacakra calendar, using modern astronomical methods: Kcal.

Software is available on this site for two different implementations of the Kālacakra calendar. The one on this other page uses the traditional calculation methods as given in the original Kālacakra Tantra. That software implements data for the original epoch given in the tantra, plus an updated epoch, based on modern astronomy. However, the calculations all use the mixed-radix system as given in the tantra.

The software available on this page, however, uses modern calculation systems throughout. This is the same software that is used to calculate the calendars available on this page. The calendar depends upon the calculation of the longitudes of the Sun and Moon, and this software implements modern trigonometric formulae for these, in the C programming language.

The calculations used for the longitude of the Sun are the VSOP87 solution, devised by P. Bretagnon and G. Francou. The data used for these calculations are to be found here. These calculations are an algebraic solution to the modern numerical integration techniques developed by the Jet Propulsion Laboratory, as are the calculations used for the longitude of the Moon. The earlier version (2.0) of this software used for the longitude of the Moon the algebraic methods (analytical solution ELP2000-85) given in "Lunar Tables and Programs from 4000 B.C. to A.D. 8000", by Michelle Chapront-Touzé and Jean Chapront. These calculations were a little less accurate than those for the Sun. The maximum error given for these lunar calculations is 0.021°, 1,000 years ago, and 0° in the year 2000. This was borne out in testing.

The calculations based on ELP2000-85 have now (since v.2.1) been replaced by the most recent solution, ELP/MPP02. The data for this is available here. That link includes a pdf file describing the solution, together with necessary data files and an example Fortran file illustrating how to use the data. (Those two links, and one other in this paragraph, are currently [July 2013] broken; they will be replaced if alternatives appear.) The new routine used in the Kālacakra software here is essentially a translation into C of the relevant parts of that Fortran file. There is also a similar translation into C of the Fortran in the General Astrodynamics Library, in the file gal_gmopv02.c. The new ELP/MPP02 solution is much more complex than the earlier ELP2000-85; as a consequence, the latest version of the Kcal program runs much more slowly than earier versions. There are two slightly different sets of data given with ELP/MPP02, depending on the data to which the solution has been fit, either the DE405 ephemeris calculated by the Jet Propulsion Laboratory or the Laser Lunar Ranging (LLR) data. It is the latter that has been used here. For a discussion of the recent developments in lunar theory, see a recent paper by Jean Chapront and Gerard Francou.

Results obtained using the new lunar routine in Kcal have been tested against the output of both the original Fortran file and the routine in the General Astrodynamics Library. Unlike those two routines, the one used here does not yet calculate the rectangular coordinates of the Moon, but the longitude, referred to the mean ecliptic of date. It also calculates the latitude and lunar distance, but these are not used. Use may be made of these, and possibly the rectangular coordinates, in the future. The most likely reason for using them would be for greater detail in eclipse predictions.

As before, results from the software for full and new Moon times have been compared with data given by NASA, here. These data are published correct to the nearest minute, and results of both versions 2.0 and 2.1 of the software for the present year mostly give new and full Moons at the same time, sometimes plus or, rarely, minus one minute. In tests run for 1,000 years ago, for the earlier version 2.0, the new and full Moon times are two or three minutes earlier than those given by NASA. Given the speed of travel of the Moon, this is what one would expect for the stated error of about 0.02°. The new software is much more accurate, and the results for 1,000 years ago are similar to those for the present day – either the same time, or plus or minus one minute. Some sample test results are available here. The first column gives the NASA data, the second results from the earlier version of Kcal, and the right-hand column results for the latest version. So that others have the option of repeating these tests, the code used in testing the new and full Moon times has been left in the source file, k1.c.

The results of calculations for both the Sun and the Moon have also been extensively checked against examples given by NASA and others. In checking the error during the last 3,000 years it was found that the error in delta T – the difference between Universal Time and Dynamical Time (which replaced the original Ephemeris Time, and is nowadays bizarrely called Terrestial Time) – was greater than the errors in the calculations themselves. A more accurate method for calculating delta T has been published by NASA, and this has been implemented in this latest version of the software.

When this software was first created, several other components were planned to be included, but have not yet been built into the program. One example of information not yet included is the calculation for the position of the Moon's node, known as Rāhu in Sanskrit. This will be added in one of the next versions, in the near future. One of the main reasons for detailing the position of Rāhu would be for the calculation of eclipses – something that is very inaccurate in the traditional systems. The calendar contains eclipse information based on the methods given by Jean Meeus, and these are really quite good enough for most purposes. Anybody interested in looking for further details on any eclipse is best referred to the data put together by Fred Espenak on the NASA web site.

The other main feature originally intended was the positions of the five main planets: Mercury, Venus, Mars, Jupiter and Saturn. Software was prepared for this purpose, implementing the VSOP87 solution, but not built into the calendar. That software will be made available here in the near future, together with test code. When ready, this will be described in more detail below.

Download software

The links below will download both the executable file for the software and the source code. As with any open source project, comments, suggested changes and additions will all be welcome. Please address these to Edward Henning. In particular, if you find any errors, please report these.

To download the software and its source code:

Click below for:	
Kcal source version 2.0	C source code plus executable file
Kcal version 2.0	For non-programmers: 32-bit executable file plus example batch file
Kcal source version 2.1	C source code plus executable file, including the new lunar solution, ELP/MPP02
Kcal version 2.1	For non-programmers: 32-bit executable file plus example batch file
Kcal source version 3.0	C source code plus executable file, all Chinese and Tibetan information removed; a pure Kālacakra calendar
Kcal version 3.0	For non-programmers: 32-bit executable file plus example batch file
Kcal source version 3.1	C source code plus executable file; fixes an error in the routine for delta T
Kcal version 3.1	For non-programmers: 32-bit executable file plus example batch file

Description of the software

Those not interested in the source code but in just using the software, please scroll down to the section below on using the software.

The software as supplied here is a 32-bit Windows NT command line executable, to ensure compatibility with Windows Vista and Windows 7. Earlier versions were 16-bit DOS programs, and these could not be run on anything later than Windows XP. This version of the software has been developed using the Open Watcom C/C++ compiler version 1.9.

k1.c – Main control routines, plus calculation and printing of the calendar. One previously unused routine in this module was conv2hms. The calendar is printed with times given in terms of the units used in the Kālacakra Tantra, of nāḍī, pala and breaths, measured from mean daybreak. Some may prefer hours, minutes and seconds of Local Mean Solar Time, and this routine performs this conversion. It has been implemented since v.3.0.

eclipse.c – Eclipse calculations; method from Astronomical Algorithms, Jean Meeus, p. 379, etc.

sunmon.c – Routines for calculating solar and lunar longitudes. These are the basis of the calendar.

jd.c – Julian day and other routines associated with time, including delta T (ΔT) calculation. That calculation is an implementation of the method given here, on the NASA web site.

bcd.c – Binary coded decimal routines for large integer arithmetic. These are used by the module jd.c. As with the BCD routines available with other programs on this site, this is not a comprehensive library of BCD routines, but includes just those necessary for the main calculations.

Planetary calculations – details regarding this will be added when the software itself is made available.

Using the software

In order to create your own calendars for any location, accurate through a period of at least a couple of thousand years either side of the present day, download the calendar software (Kcal, see above), and follow the instructions below:

The archive file (kexe1307.zip is the latest) needs to be unzipped, or decompressed; it contains two files:

kcal.exe – this is the executable file for creating a calendar. This is a command line program and needs to be run in a text, or command line, window on a Windows PC. This is not Windows software, but it is compatible with the text mode, or command mode, of Windows systems (including Windows Vista and Windows 7.)

go2010.bat – this is a sample batch file for running kcal.exe. The software is not interactive like other programs available from this site. It accepts commands such as given in the batch file, and then prints a calendar for a particular year to disk in a text file. The batch file given as an example contains just one line:

```
kcal uk2010.txt London,_UK 2010 -0.17
```

First is "kcal", the name of the software for creating the calendar. The next four items are parameters to be passed to the software. All four of these must be present, and there must be one space between each. If these parameters are not given correctly, the software will not operate properly. (It will default to the Greenwich meridian and prompt for a year.)

The first parameter, "uk2010.txt" is the name of the plain text file that will be produced containing the calendar. The second parameter, "London,_UK" contains information that will be placed in the first line of the calendar, for the place for which the calendar is calculated. In this example, the first line of the calendar becomes:

```
Calendar for London, UK, and other places near longitude -0.17
```

Notice that the underscore in "London,_UK" is replaced by a space in the final calendar.

The next parameter, "2010", is the western year for which the calendar is to be calculated. This is used in the calculation, and also in the first line of the calendar.

The final parameter, "-0.17", is the longitude in degrees of the place for which the calendar is to be calculated. A negative number must be used for places west of Greenwich, and a positive number (the +sign is not needed) for places east of Greenwich. This value is used in the calculation and is also placed in the first line of the calendar.

Naturally, a batch file with many lines can be used to create several calendars in one run. For example, the first three lines of the batch file used to generate the calendars on this web site are:

```
kcal sfous00.txt San_Francisco,_US 2000 -122.4
kcal chius00.txt Chicago,_US 2000 -87.65
kcal bostus00.txt Boston,_US 2000 -71.06
```

The most recent version of this software is dated 26 July 2013. Further versions will be placed here in due course, either to correct bugs in the software or to add new features.

E. Henning.
Last updated 25 November 2015.
Return to calendar introduction.