Appendix E
Universal Time and Julian Dates

James R. Wertz, *Microcosm, Inc.*

Calendar time in the usual form of date and time is used only for input and output because arithmetic is cumbersome in months, days, hours, minutes, and seconds. Nonetheless, this is used for most human interaction with space systems because it's the system with which we are most familiar. Even with date and time systems, problems can arise, because time zones are different throughout the world and spacecraft operations typically involve a worldwide network. The uniformly adopted solution to this problem is to use the local standard time corresponding to 0 deg longitude (i.e., the Greenwich meridian) as the assigned time for events anywhere in the world or in space. This is referred to as *Universal Time* (UT), *Greenwich Mean Time* (GMT), or *Zulu* (Z), all of which are equivalent for most practical spacecraft operations. The name Greenwich Mean Time is used because 0 deg longitude is defined by the site of the former Royal Greenwich Observatory in metropolitan London.

Civil time, T_{civil}, as measured by a standard wall clock or time signals, differs from Universal Time by an integral number of hours, corresponding approximately to the longitude of the observer. The approximate relation is:

$$T_{civil} \approx UT \pm (L + 7.5)/15 \tag{E-1}$$

where T_{civil} and UT are in hours, and L is the longitude in degrees with the plus sign corresponding to East longitude and the minus sign corresponding to West longitude. The conversion between civil time and Universal Time for most North American and European time zones is given in Table E-1. Substantial variations in time zones are created for political convenience. In addition, most of the United States and Canada observe Daylight Savings Time from the first Sunday in April until the last Sunday in October. Most European countries observe daylight savings time (called "summer time") from the last Sunday in March to the first Sunday in October. Many countries in the southern hemisphere also maintain daylight savings time, typically from October to March. Countries near the equator typically do not deviate from standard time.

Calendar time is remarkably inconvenient for computation, particularly over long time intervals of months or years. We need an absolute time that is a continuous count of time units from some arbitrary reference. The time interval between any two events may then be found by simply subtracting the absolute time of the second event from that of the first. The universally adopted solution for astronomical problems is the *Julian Day*, JD, a continuous count of the number of days since Greenwich noon (12:00 UT) on January 1, 4713 BC*, or, as astronomers now say, –4712. Because Julian Days start at noon UT, they will be a half day off with respect to civil dates. While this is inconvenient for transforming from civil dates to Julian dates, it was useful for astronomers because the date didn't change in the middle of the night (for European observers).

TABLE E-1. Time Zones in North America, Europe, and Japan. In most of the United States, Daylight Savings Time is used from the first Sunday in April until the last Sunday in October. In Europe, the equivalent "summer time" is used from the last Sunday in March to the first Sunday in October.

Time Zone	Standard Meridian (Deg, East Long.)	UT Minus Standard Time (Hours)	UT Minus Daylight Time (Hours)
Atlantic	300	4	3
Eastern	285	5	4
Central	270	6	5
Mountain	255	7	6
Pacific	240	8	7
Alaska	225	9	8
Hawaii	210	10	NA
Japan	135	–9	NA
Central Europe	15	–1	–2
United Kingdom	0	0	–1

As described below, there are four general approaches for converting between calendar dates and Julian dates.

Table Look-Up

Tabulations of the current Julian Date are in most astronomical ephemerides and almanacs. Table E–2 lists the Julian Dates at the beginning of each year from 1990 through 2031. To find the Julian Date for any given calendar date, simply add the *day number* within the year (and fractional day number, if appropriate) to the Julian Date for Jan 0.0 of that year from Table E–2. Day numbers for each day of the year are on many calendars or can be found by adding the date to the day number for day 0 of the month from Table E-3. Thus 18:00 UT on April 15, 2002 = day number 15.75 + 90 = 105.75 in 2002 = JD 105.75 + 2,452,274.5 = JD 2,452,380.25.

To convert from Julian Days to dates, determine the year in which the Julian Date falls from Table E–2. Subtract the Julian Date from the JD for January 0.0 of that year to determine the day number within the year. This can be converted to a date (and time, if appropriate) by using day numbers on a calendar or subtracting from the day number for the beginning of the appropriate month from Table E–3. Thus, from Table E–2, JD 2,451,608.25 is in the year 2000. The day number is 2,451,608.25 – 2,451,543.5 = 64.75. From Table E–3, this is 18:00 UT, March 4, 2000.

Software Routines Using Integer Arithmetic

A particularly clever procedure for finding the Julian Date, JD, associated with any current year, Y, month, M, and day of the month, D, is given by Fliegel and Van

---

* This strange starting point was suggested by an Italian scholar of Greek and Hebrew, Joseph Scaliger, in 1582 as the beginning of the current *Julian period* of 7,980 years. This period is the product of three numbers: the *solar cycle*, or the interval at which all dates recur on the same days of the week (28 years); the *lunar cycle*, containing an integral number of lunar months (19 years); and the *indiction* or the tax period introduced by the Emperor Constantine in 313 AD (15 years). The last time that these started together was 4713 BC and the next time will be 3267 AD. Scaliger was interested in reducing the astronomical dating problems associated with calendar reforms of his time and his proposal had the convenient selling point that it pre-dated the ecclesiastically approved date of creation, October 4, 4004 BC.

TABLE E–2. Julian Date at the Beginning of Each Year from 1990 to 2031. See text for explanation of use. The day number for the beginning of the year is called “Jan. 0.0” (actually Dec. 31st of the preceding year) so that day numbers can be found by simply using dates. Thus, Jan. 1 is day number 1 and has a JD 1 greater than that for Jan. 0. * = leap year.

Year	***JD* 2,400,000+ for Jan 0.0 *UT***	Year	***JD* 2,400,000+ for Jan 0.0 *UT***	Year	***JD* 2,400,000+ for Jan 0.0 *UT***
1990	47,891.5	2004*	53,004.5	2018	58,118.5
1991	48,256.5	2005	53,370.5	2019	58,483.5
1992*	48,621.5	2006	53,735.5	2020*	58,848.5
1993	48,987.5	2007	54,100.5	2021	59,214.5
1994	49.352.5	2008*	54,465.5	2022	59,579.5
1995	49,717.5	2009	54,831.5	2023	59,944.5
1996*	50,082.5	2010	55,196.5	2024*	60,309.5
1997	50,448.5	2011	55,561.5	2025	60,675.5
1998	50,813.5	2012*	55,926.5	2026	61,040.5
1999	51,178.5	2013	56,292.5	2027	61,405.5
2000*	51,543.5	2014	56,657.5	2028*	61,770.5
2001	51,909.5	2015	57,022.5	2029	62,136.5
2002	52,274.5	2016*	57,387.5	2030	62,501.5
2003	52,639.5	2017	57,753.5	2031	62,866.5

TABLE E–3. Day Numbers for Day 0.0 of Each Month. Leap years (in which February has 29 days) are those evenly divisible by 4. However, years evenly divisible by 100 are not leap years, except that those evenly divisible by 400 are. Leap years are indicated by * in Table E–2.

Month	Non-Leap Years	Leap Years
January	0	0
February	31	31
March	59	60
April	90	91
May	120	121
June	151	152
July	181	182
August	212	213
September	243	244
October	273	274
November	304	305
December	334	335

Flandern [1968] as a computer statement using integer arithmetic. Note that all of the variables must be defined as integers (i.e., any remainder after a division must be truncated) and that both the order of the computations and the parentheses are critical. This procedure works in FORTRAN, C, C++, and Ada for any date on the Gregorian calendar that yields JD > 0. (Add 10 days to the JD for dates on the Julian calendar prior to 1582.)

$$\begin{aligned} JD_0 = {} & D - 32{,}075 + 1461 \times (Y + 4800 + (M - 14)/12)/4 \\ & + 367 \times (M - 2 - (M - 14)/12 \times 12)/12 \\ & -3 \times ((Y + 4900 + (M - 14) / 12) / 100) /4 \end{aligned} \tag{E-2a}$$

Here JD_0 is the Julian Day beginning at noon UT on the given date and must be an integer. For a fractional day, F, in UT (i.e., day number D.F), the floating point Julian Day is given by:

$$JD = JD_0 + F - 0.5 \tag{E-2b}$$

For example, the Julian Day beginning at 12:00 UT on December 25, 2007 (Y = 2007, M = 12, D = 25) is JD 2,454,460 and 6:00 UT on that date (F = 0.25) is JD 2,454,459.75.

The inverse routine for computing the date from the Julian Day is given by:

$$L = JD_0 + 68{,}569 \tag{E-3a}$$

$$N = (4 \times L) / 146{,}097 \tag{E-3b}$$

$$L = L - (146097 \times N + 3) / 4 \tag{E-3c}$$

$$I = (4000 \times (L + 1)) / 1{,}461{,}001 \tag{E-3d}$$

$$L = L - (1461 \times I) / 4 + 31 \tag{E-3e}$$

$$J = (80 \times L) / 2{,}447 \tag{E-3f}$$

$$D = L - (2447 \times J) / 80 \tag{E-3g}$$

$$L = J / 11 \tag{E-3h}$$

$$M = J + 2 - 12 \times L \tag{E-3i}$$

$$Y = 100 \times (N - 49) + I + L \tag{E-3j}$$

where integer arithmetic is used throughout. Y, M, and D are the year, month, and day, and I, J, L, and N are intermediate variables. Finally, again using integer arithmetic, the day of the week, W, corresponding to the Julian Date beginning at 12:00 on that day is given by:

$$W = JD_0 - 7 \times ((JD + 1) / 7) + 2 \tag{E-4}$$

where W = 1 corresponds to Sunday. Thus, December 25, 2007 falls on Tuesday.

Software Routines Without Integer Arithmetic

While most computer languages provide integer arithmetic, spreadsheets such as Excel or MatLab typically do not. (See below for use of Excel and MatLab DATE functions.) Similar capabilities are available using integer (INT) or truncation (TRUNC in Excel, FIX in MatLab) functions. INT and TRUNC are identical for positive numbers, but differ for negative numbers: INT(–3.1) = –4, whereas

TRUNC(–3.1) = –3. It is the TRUNC or FIX function which is equivalent to integer arithmetic. Thus, using the same variables as above, we can rewrite Eqs. (E-2) for computation of JD from the date as:

$$C = \mathrm{TRUNC}((M - 14)/12) \tag{E-5a}$$

$$\begin{aligned} JD_0 = {} & D - 32{,}075 + \mathrm{TRUNC}(1{,}461 \times (Y + 4{,}800 + C)/4) \\ & + \mathrm{TRUNC}(367 \times (M - 2 - C \times 12)/12) \\ & - \mathrm{TRUNC}(3 \times (\mathrm{TRUNC}(Y + 4{,}900 + C) / 100) /4) \end{aligned} \tag{E-5b}$$

$$JD = JD_0 + F - 0.5 \tag{E-5c}$$

where again JD_0, Y, M, D, and C are integers and F and JD are real numbers. Applying the same rules to Eq. (E-3) gives the inverse formula for the date in terms of JD as:

$$L = JD + 68{,}569 \tag{E-6a}$$

$$N = \mathrm{TRUNC}((4 \times L) / 146{,}097) \tag{E-6b}$$

$$L = L - \mathrm{TRUNC}((146097 \times N + 3) / 4) \tag{E-6c}$$

$$I = \mathrm{TRUNC}((4000 \times (L + 1)) / 1{,}461{,}001) \tag{E-6d}$$

$$L = L - \mathrm{TRUNC}((1{,}461 \times I) / 4) + 31 \tag{E-6e}$$

$$J = \mathrm{TRUNC}((80 \times L) / 2{,}447) \tag{E-6f}$$

$$D = L - \mathrm{TRUNC}((2{,}447 \times J) / 80) \tag{E-6g}$$

$$L = \mathrm{TRUNC}(J / 11) \tag{E-6h}$$

$$M = J + 2 - 12 \times L \tag{E-6i}$$

$$Y = 100 \times (N - 49) + I + L \tag{E-6j}$$

where the variables are the same as Eq. (E-3), except that D is now a real number corresponding to the date and fraction of a day. Finally, Eq. (E-4) for the day of the week becomes:

$$\begin{aligned} W &= JD - 7 \times \mathrm{TRUNC}\,((JD + 1.5) / 7) + 2.5 \\ &= JD - 7 \times \mathrm{INT}\,((JD + 1.5) / 7) + 2.5 \end{aligned} \tag{E-7}$$

where $1 \le W < 2$ corresponds to Sunday. The examples given above can also serve as test cases for Eqs. (E-5), (E-6), and (E-7).

Modified Julian Date

The Julian Date presents minor problems for space applications. Because it was introduced principally for astronomical use, Julian Dates begin at 12:00 UT rather than 0 hours UT, as the civil calendar does (thus the 0.5 day differences in Table E–2). In addition, the 7 digits required for the Julian Date did not permit the use of single precision arithmetic in older computer programs. This is no longer a problem with modern computer storage and number formats. Nonetheless, various forms of truncated Julian dates have gained at least some use.

The most common of the truncated Julian dates for astronomical and astronautics use is the *Modified Julian Date*, MJD, given by:

$$MJD = JD - 2{,}400{,}000.5 \tag{E-8}$$

MJD begins at midnight, to correspond with the civil calendar. Thus, in using

Table E–2, the MJD is given by adding the day of the year (plus fractions of a day, if appropriate) to the number in the table, with the ".5" at the end of the table-listing dropped. For example, the MJD for 18:00 UT on Jan. 3, 2002 = MJD 52,277.75. The definition of the MJD given here is that adopted by the International Astronomical Union in 1997. Note, however, that other definitions of the MJD have been used. Thus, the most unambiguous approach remains the use of the full Julian Date.

Spreadsheets such as Excel or MatLab

Spreadsheets, such as Excel or MatLab, typically store dates internally as some form of day count and allow arithmetic operations, such as subtraction. Thus, we can either subtract two dates directly to determine a time interval or convert them to Julian Dates by simply finding the additive constant, K, given by:

$$K = JD - I \tag{E-9}$$

where I is the internal number representing a known date, JD. Once this is determined, then the JD for any date is:

$$JD = K + I \tag{E-10}$$

Many versions of Excel use Jan. 1, 1904, as "day 0," such that K_{Excel} = 2,416,480.5. However, this should be checked for individual programs because other starting points are sometimes used and the starting point is a variable parameter in some versions of Excel. While this can be a very convenient function, Excel date routines run only from 1904 to 2078.

MatLab typically uses Jan. 1, 0000, 0:0:0 as "day 0." Thus, in the formula above, K_{MatLab} = 1,721,058.5.

Any of the day counting approaches will work successfully over its allowed range. However, systems intended for general mathematics or business use may not account correctly for leap years and calendar changes when historical times or times far in the future are being evaluated. Thus, the use of the full Julian Date remains the most unambiguous solution, particularly if a program or result is to be used by more than one person or program. For a more extended discussion of time systems, see for example, Seidelmann [1992] or Wertz [1999].

References

Fliegel, Henry F. and Thomas C. Van Flandern. 1968. "A Machine Algorithm for Processing Calendar Dates." *Communications of the ACM*, vol. II, p. 657.

Seidelmann, P. Kenneth. 1992. *Explanatory Supplement to the Astronomical Almanac*. Mill Valley, CA: University Science Books.

Wertz, James R. 1999. *Spacecraft Orbit and Attitude Systems*, vol. I. Torrance, CA and Dordrecht, The Netherlands: Microcosm, Inc. and Kluwer Academic.

Space Mission Analysis and Design, 3rd Edition

Errata as of November 9, 1999*

The following errata are provided to keep this volume as useful as possible. We would appreciate any other corrections or suggestions being reported to: Donna Klungle, Microcosm, Inc., 2377 Crenshaw Blvd., Suite 350, Torrance, CA 90501, Phone: (310) 320-0555, Fax: (310) 320-0252 or E-MAIL: bookproject@smad.com.

By the way, if your book doesn't have a red cover and say **Third Edition**, then you have either the first or second edition of SMAD, which are very different. A separate errata sheet is available for the second edition.

** Most recent additions to the errata have page numbers highlighted in red.*

ERRATA

PAGE	THIRD EDITION, FIRST PRINTING
136	In Eq. (6-9), "ν^2" should be "V^2"
143	In Eq. (6-20), "1.032.37" should be "1.032 37"
144	In the caption of Table 6-4, "precursor" should be "precession" In Table 6-4, the row for $\dot{\omega}$ for the Shuttle, "0.002 42" should be "0.000 31" and "0.001 10" shoud be "0.000 14"
177	In Fig. 7-8, the F10.7 and Ap labels are reversed from top to bottom, i.e., F10.7 = 225, Ap = 20 is the top line, F10.7=175, Ap = 16 is the second line, etc. Also, in the caption, "Ballistic Coefficient of 65 kg/m^2" should be "Ballistic Coefficient of 100 kg/m^2"
192	On Fig. 7-10, "S/2" is half the distance between the centers of the two coverage circles.
192	On Fig. 7-11, the spacing between the two orbits is D_{maxS}.
198	In Table 7-12, Step 6, "Table 7-14" should be "Table 7-13" in both columns.
218	On Fig. 8-11, "1–8" should be "1–8 Å"
255	In Fig. 9-4, in the Visible, Ultraviolet, and Infrared regions of the spectrum, the wavelength units should be "μm," not "m".
265	In Table 9-9, in the Visible column of the top row, "0.366" should be "0.366 m".
428	In Fig. 11-14, the three minus signs in the labels should all be plus-or-minus signs (e.g., 1,367 ± 5 W/m^2, Albedo 30 ± 5 %, 237 ± 21 W/m^2).
566	In Table 13-12, the asterisk corresponding to the footnote should be in the rightmost column (Downlink Power) of the X-band row. The value in that cell should be "–142/4 kHz*"
908-913	Appendix E has been both corrected and revised to improve clarity. The revised appendix is contained in the following pages.
843	In the upper NOTE on Fig. 21-3, "GEO" should be "LEO."

For periodic updates on this and other books in the Space Technology Library series, visit the Space Technology Library (STL) in the Book Store at:

http://www.smad.com