Time

Astronomers have always been concerned with time and its measurement. If you read any astronomical text on the subject you are sure to be bewildered by the seemingly endless range of times and their definitions. There's universal time and Greenwich mean time, apparent sidereal time and mean sidereal time, ephemeris time, local time, mean solar time and atomic time, to name but a few. Then there's the sidereal year, the tropical year, the Besselian year and the anomalistic year. And be quite clear about the distinction between the Julian and Gregorian calendars! (See the Glossary for the definitions of these terms.)

All these terms are necessary and have precise definitions. Happily, however, we need concern ourselves with but a few of them as the distinctions between many of them become apparent only when very high accuracy is required.

 https://doi.org/10.1017/CBO9780511861161.004

1 Calendars

A **calendar** helps us to keep track of time by dividing the year into months, weeks and days. Very roughly speaking, one **month** is the time taken by the Moon to complete one circuit of its orbit around the Earth, during which time it displays four phases, or quarters, of one week each, and a **year** is the time taken for the Earth to complete one circuit of its orbit around the Sun. In the Gregorian calendar, generally adopted in the West, we assume the convention that there are seven days in each week, between 28 and 31 in each month (see Table 1) and 12 months in each year. (Note that there are many other calendars, such as the Chinese calendar, with different definitions or rules.) By knowing the day number, and the name of the month, we are able to refer precisely to any day of the year.

January	31	July	31
February	28 (or 29 in a leap year)	August	31
March	31	September	30
April	30	October	31
May	31	November	30
June	30	December	31

Table 1. The number of days in each month.

The problem with this method of accounting for the days in the year lies in the fact that, whereas there is always a whole number of days in the civil year, the Earth actually takes about 365.2422 days to complete one circuit of its orbit around the Sun. (This is the **tropical year**; see the Glossary for its definition.) If we were to take no notice of this fact and adopt 365 days for every year, then the Earth would get progressively more out of step with the civil calendar at a rate of 0.2422 days per year. After 100 years the discrepancy would be about 24 days; after 1500 years the seasons would have been reversed so that summer in the northern hemisphere would be in December. Clearly, this system would have great disadvantages.

Julius Caesar made an attempt to put matters right by adopting the convention that three consecutive years have 365 days followed by a **leap year** of 366 days, the extra day being added to February whenever the year number is divisible by four. On average, his civil year has 365.25 days in it, a better approximation to the tropical year of 365.2422 days. Indeed, after 100 years the discrepancy is less than one day. This is the **Julian calendar** and it worked very well for many centuries until, by 1582, there was again an appreciable discrepancy between the seasons and the calendar date. Pope Gregory XIII (1502–1585) then improved on the system by abolishing the days 5 October to 14 October 1582 inclusive so as to bring the civil and tropical years back into line, and by missing out three days every four centuries. In his reformed calendar, the years ending in two zeroes (e.g. 1700, 1800, 1900 etc.) are only leap years if they are also exactly divisible by 400. Thus the year 2000 was a leap year, whereas 1700, 1800 and 1900 were not.

This system is called the **Gregorian calendar** and is the one in most general use today. According to it 400 civil years contain $(400 \times 365) + 100 - 3 = 146\,097$ days, so that the average length of the civil year is 146 097/400 = 365.2425 days, a very good approximation indeed to the length of the tropical year. In fact, this calendar will not get out of step with the tropical year for many millions of years!

2 The date of Easter

Easter Day, which always occurs on a Sunday, is the day to which such moveable feasts as Whitsun and Trinity Sunday in the Christian calendar are fixed, and is defined in *The Explanatory Supplement to the Astronomical Almanac* (1992) as follows:

> *In the Gregorian calendar, the date of Easter is defined to occur on the Sunday following the ecclesiastical full moon that falls on or next after March 21st.*

The problem is that the ecclesiastical full Moon is not the same as the astronomical full Moon. The former is based on a set of tables which do not take into account the complexity of the Moon's motion. As a fair guide, we may say that Easter Day is usually the first Sunday after the fourteenth day after the first new Moon after 21 March. Several authors have provided algorithms for calculating the date of Easter. You can, for example, use the methods and tables given in the *Book of Common Prayer* (1662) or that given in the *Explanatory Supplement*. Here we describe a method devised in 1876 which first appeared in *Butcher's Ecclesiastical Calendar*, and which is valid for all years from 1583 onwards. It makes repeated use of the result of dividing one number by another number, the integer part being treated separately from the remainder. A calculator displays the result of a division as a string of numbers either side of a decimal point. The numbers appearing before (i.e. to the left of) the decimal point constitute the integer part; the decimal point and the numbers after (i.e. to the right of) the decimal point constitute the fractional part. The remainder may be found from the latter (including the leading decimal point) by multiplying it by the divisor (i.e. the number that you divided by) and rounding the result to the nearest integer value. For example, $2000/19 = 105.263\,157\,9$. The integer part is 105, and the fractional part is $0.263\,157\,9$. Multiplying the latter by 19 gives $5.000\,000\,100$ so the remainder is 5.

We shall illustrate the method by calculating the date of Easter Day in the year 2009. This will give us practice for the sort of calculation we will be carrying out in the rest of this book.

Method		*Integer part*	*Remainder*	*Example*		
1.	Divide the year by 19.		a	$\frac{2009}{19}$	=	105.736 842 1
				a	=	14
2.	Divide the year by 100.	b	c	$\frac{2009}{100}$	=	20.090 000
				b	=	20
				c	=	9
3.	Divide b by 4.	d	e	d	=	5
				e	=	0
4.	Divide $(b+8)$ by 25.	f		f	=	1
5.	Divide $(b-f+1)$ by 3.	g		g	=	6
6.	Divide† $(19a+b-d-g+15)$ by 30.		h	$(19a+b-d-g+15) = 290$		
				h	=	20
7.	Divide c by 4.	i	k	i	=	2
				k	=	1
8.	Divide $(32+2e+2i-h-k)$ by 7.		l	l	=	1
9.	Divide $(a+11h+22l)$ by 451.	m		$(a+11h+22l) = 256$		
				m	=	0
10.	Divide $(h+l-7m+114)$ by 31.	n	p	$(h+l-7m+114) = 135$		
				n	=	4
				p	=	11
11.	The day of the month on which Easter Day falls is $p+1$.			$p+1$	=	12
	The month number is n (=3 for March, =4 for April).			n	=	4, so April
	Therefore Easter Day 2009 is					**12 April**

†$19a$ means 19 multiplied by a ($19 \times 14 = 266$ in this example).

The spreadsheet for this calculation, called DOE (the acronym for Date Of Easter), is shown in Figure 1. It makes repeated use of two spreadsheet functions, TRUNC and MOD. (These are all examples of built-in, or intrinsic, spreadsheet functions; we will make use of many of the useful ones throughout this book.) The former truncates the number at the decimal point, so gives you the integer part of the number. Thus TRUNC(23.445) is 23. In cell C8 of the spreadsheet, the formula =TRUNC(C3/100) takes the number from cell C3 (2009 in this case), divides it by 100, and returns the integer part of the result (20).

The MOD function has two arguments separated by a comma. (An argument is a number or a reference within the brackets immediately following the function name. Two or more arguments are separated by commas.‡) The first argument is divided by the second argument, and then the remainder of the result is returned. Thus MOD(13,5) is 3 since 5 goes into 13 twice ($2 \times 5 = 10$) leaving a remainder of 3 (i.e. $13-10$). In cell C7 of the spreadsheet, the formula =MOD(C3,19) takes the number from cell C3 (2009 in this case), divides it by 19, and returns the remainder (14).

We have used the spreadsheet function IF in cell H4 to replace the month number, 3 or 4, with its name equivalent, 'March' or 'April'. The IF function takes three arguments. The first is the test argument, which can be 'true' or 'false'. In this case, the test argument is C22=3, i.e. if the number in cell C22 is equal to 3 the result of the test is 'true', and if not it is 'false'. In this case, the number in cell C22 is 4 so the test returns 'false'. The IF function returns the second argument (March in this case) if the test returns 'true', or the third argument (April in this case) if the test returns 'false', as here.

‡Some spreadsheet programs use different separators; check yours.

	A	B	C	D	E	F	G	H	I	J	K
1	**Date of Easter**										
2											
3	*Input*	year	2009			*Output*	day	12	=C21		
4							month	April	=IF(C22=3,"March","April")		
5							year	2009	=C3		
6											
7	1	*a*	14	=MOD(C3,19)							
8	2	*b*	20	=TRUNC(C3/100)							
9	3	*c*	9	=MOD(C3,100)							
10	4	*d*	5	=TRUNC(C8/4)							
11	5	*e*	0	=MOD(C8,4)							
12	6	*f*	1	=TRUNC((C8+8)/25)							
13	7	*g*	6	=TRUNC((C8-C12+1)/3)							
14	8	*h*	20	=MOD((19*C7)+C8-C10-C13+15,30)							
15	9	*i*	2	=TRUNC(C9/4)							
16	10	*k*	1	=MOD(C9,4)							
17	11	*l*	1	=MOD(32+2*(C11+C15)-C14-C16,7)							
18	12	*m*	0	=TRUNC((C7+(11*C14)+(22*C17))/451)							
19	13	*n*	4	=TRUNC((C14+C17-(7*C18)+114)/31)							
20	14	*p*	11	=MOD(C14+C17-(7*C18)+114,31)							
21	15	day	12	=C20+1							
22	16	month	4	=C19							

Figure 1. Calculating the date of Easter Day 2009.

You can put any year after 1582 you like into cell C3 of the spreadsheet in place of 2009 and the date of Easter Day for that year will be calculated for you automatically. Try 2012. The answer should be 8 April.

3 Converting the date to the day number

In many astronomical calculations, we need to know the number of days in the year up to a particular date. We shall choose our starting point as 0 hours on 0 January, equivalent to the midnight between 30 and 31 December of the previous year. This might seem to be a peculiar choice, but you will see that it simplifies our calculations so is a good one to make. Midday on 3 January can then be expressed as January 3.5 since precisely three and a half days have elapsed since January 0.0. This is illustrated in Figure 2. Finding the day number from the date is then a simple matter. Proceed as follows:

1. For every month up to, but not including, the month in question add the appropriate number of days according to Table 1. These totals are listed in Table 2.
2. Add the day of the month.

For example, what is the day number of 19 June (not a leap year)? The answer is day number $= 151 + 19 = 170$. If you own a programmable calculator, you may be able to use Routine R1 (at the end of this section) to write a program to make this calculation automatically. We can also use the method of the section on Julian day numbers (Section 4) as an alternative.

Later in this book we adopt the date 2010 January 0.0 as the starting point, or starting **epoch**, from which to calculate orbital positions. Days elapsed since this epoch at the beginning of each year (January 0.0) from 1990 to 2029 are tabulated in Table 3. To find the total number of days elapsed since the epoch, simply add the number of days elapsed to the beginning of the year since the epoch (Table 3) to the number of days elapsed since January 0.0 of the year in question (i.e. the result of the calculation of the previous paragraph). For example, the number of days elapsed since the epoch at 6 pm on 19 June 2009 is $-365 + 170 + 0.75 = -194.25$. The negative sign indicates that the epoch is after this date. The fraction of the day to 6 pm is $(18/24) = 0.75$ since 6 pm is 18 h on a 24-hour clock, and there are 24 hours in the day.

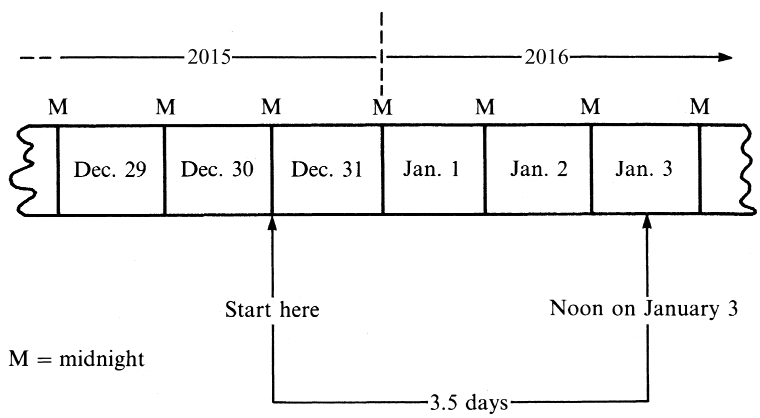


Figure 2. Defining the epoch.

	Ordinary year	Leap year
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

Table 2. The number of days to the beginning of the month.

1990	−7305	*2000	−3653	2010	0	*2020	3652
1991	−6940	2001	−3287	2011	365	2021	4018
*1992	−6575	2002	−2922	*2012	730	2022	4383
1993	−6209	2003	−2557	2013	1096	2023	4748
1994	−5844	*2004	−2192	2014	1461	*2024	5113
1995	−5479	2005	−1826	2015	1826	2025	5479
*1996	−5114	2006	−1461	*2016	2191	2026	5844
1997	−4748	2007	−1096	2017	2557	2027	6209
1998	−4383	*2008	−731	2018	2922	*2028	6574
1999	−4018	2009	−365	2019	3287	2029	6940

*Denotes a leap year.

Table 3. The number of days to the beginning of the year since the epoch 2010 January 0.0.

Routine R1: Converting the date to the day number.

1. Key in the month number (e.g. 11 for November).
2. Is it greater than 2?
   - If yes, go to step 8.
   - If no, proceed with step 3.
3. Subtract 1 from the month number.
4. Multiply by 63 in an ordinary year, or 62 in a leap year.
5. Divide by 2.
6. Take the integer part.
7. Go to step 12.
8. Add 1 to the month number.
9. Multiply by 30.6.
10. Take the integer part.
11. Subtract 63 in an ordinary year, or 62 in a leap year.
12. Add the day of the month. The result is the day number.

4 Julian dates

It is sometimes necessary to express an instant of observation as so many days and a fraction of a day after a given fundamental epoch. Astronomers have chosen this fundamental epoch as the Greenwich mean noon of 1 January 4713 BC, that is midday as measured on the Greenwich meridian on 1 January of that year. (You can look up the meaning of **meridian** and other technical terms in the Glossary at the back of the book starting on page 197.) The number of days that have elapsed since that time is referred to as the **Julian day number**, or **Julian date**†. It is important to note that each new Julian day begins at 12h 00m UT, half a day out of step with the civil day in time zone 0. (See Section 9, or the Glossary, for the precise meaning of UT.)

The term '**Before Christ**', or BC for short, usually refers to the chronological system of reckoning negative years. In this system, there is no year zero. The **Christian Era** begins with the year 1 AD (short for **Anno Domini**); the year immediately preceding this is 1 BC. Some authors have adopted different labels for the same things by referring to the Christian Era as the **Common Era** instead. They retain the same numeric values for the days, but use the label CE (Common Era) instead of AD, and BCE (**Before the Common Era**) instead of BC.

For astronomical purposes, we want to count the years logically without a gap. Thus the year immediately preceding 1 AD is designated 0; the other years BC are denoted by negative numbers, each of which has an absolute value (i.e. the number without its minus sign) which is one less than the BC value. Thus

†Sometimes the **modified Julian date**, MJD, is quoted. This is equal to the Julian date minus 2400000.5; MJD zero therefore begins at 0h on 17 November 1858.

10 BC corresponds to the astronomical year -9, and 4713 BC corresponds to -4712. We shall adopt the astronomical way of counting throughout this book. Where you see a BC (or BCE) year, subtract one from it and change its sign to negative before using it in any of the calculations. Similarly, if the result of a calculation is a negative year, remove the minus sign, add one to the year number, and append the letters BC (or BCE) after it.

The Julian date of any day in the Julian or Gregorian calendars may be found by the method given below. Here, and throughout the book, the expression TRUNC refers to the integer part of the number (i.e. the part preceding the decimal point). Thus TRUNC(22.456) is 22, and TRUNC(-3.914) is -3. You will need to look carefully in the instruction book of your calculator to see what function is offered on your machine. On ours, this is called INT (short for integer). Note that computer languages offer several truncation functions such as INT, FIX, FLOOR and TRUNC. These do similar things with positive numbers, but beware what they do with negative ones. For example, INT on some machines returns the largest (most positive) integer whose value is less than or equal to the number. In this case, INT(-3.914) is -4. Beware! You can avoid this worry by taking INT of the absolute value of the number, and then inserting a negative sign in front of the result for a negative number.

A further complication, but an important one, is to distinguish between the local date, i.e. the calendar date at your location, and the corresponding Greenwich date, i.e. the calendar date on longitude 0° with no daylight saving. These are often not the same. For example, if you live in Sydney, Australia, your time may be 10 or 11 hours ahead of the time at Greenwich depending on whether daylight saving time is in operation. If it is 03:45 in the early morning in Sydney, and the time-zone correction is $+10$ hours with daylight saving adding a further hour, the corresponding time at Greenwich is 11 hours behind, i.e. 16:45 the previous day. In this case, your local calendar date and the Greenwich date differ by 1 day. We therefore need to be precise about what we mean by the 'date'. Look to see whether it is the Greenwich date or the local date that is required in a given calculation.

As an example, we shall calculate the Julian date corresponding to the Greenwich calendar date of 2009 June 19.75 (i.e. 6 pm on 19 June).

Method		*Example*		
1.	Set y = year, m = month and d = day.	y	=	2009
		m	=	6
		d	=	19.75
2.	If $m = 1$ or 2, set $y' = y - 1$ and $m' = m + 12$;	y'	=	2009
	otherwise $y' = y$ and $m' = m$.	m'	=	6
3.	If the date is later than 1582 October 15	A	=	TRUNC(2009/100)
	(i.e. in the Gregorian calendar) calculate:			
	(a) $A =$ TRUNC(y'/100);	so A	=	20
	(b) $B = 2 - A +$ TRUNC(A/4).	B	=	$2 - 20+$TRUNC(20/4)
	Otherwise $B = 0$.	so B	=	-13
4.	If y' is negative calculate $C =$ TRUNC($(365.25 \times y') - 0.75$).	C	=	TRUNC(365.25×2009)
	Otherwise, $C =$ TRUNC($365.25 \times y'$).	so C	=	733 787
5.	Calculate $D =$ TRUNC($30.6001 \times (m' + 1)$).	D	=	TRUNC(30.6001×7)
		D	=	214
6.	Find JD$= B + C + D + d + 1\,720\,994.5$.	JD	=	**2 455 002.25**
	This is the Julian date.			

The Julian date corresponding to our adopted starting epoch of 2010 January 0.0 is 2 455 196.5. We can easily find the number of days that have elapsed since the epoch by subtracting this number from the Julian date. Thus the number of days elapsed since the epoch to 2009 June 19.75 is $2455002.25 - 2455\,196.5 = -194.25$, as found in the previous section.

The spreadsheet for the calculation of the Julian date is called CDJD (the acronym for Calendar Date to Julian Date conversion) and is shown in Figure 3. We have also provided a spreadsheet function of the same name, i.e. CDJD(GD,GM,GY), which takes three arguments GD, GM, and GY. These have exactly the same values as the input values to the spreadsheet CDJD, and represent, respectively, the calendar day, month and year at Greenwich. You could carry out exactly the same calculation as that shown in Figure 3 by deleting rows 7 to 16 entirely and replacing cell H3 with the formula =CDJD(C3,C4,C5). Why not try this for yourself (but save a copy of the full spreadsheet first)?

	A	B	C	D	E	F	G	H	I	J
1	**Greenwich calendar date to Julian date conversion**									
2										
3	*Input*	**Greenwich day**	**19.75**			*Output*	**Julian date**	**2455002.25**	=C16	
4		**Greenwich month**	**6**							
5		**Greenwich year**	**2009**							
6										
7	1	*y*	2009	=C5						
8	2	*m*	6	=C4						
9	3	*d*	19.75	=C3						
10	4	*yd*	2009	=IF(C8<3,C7-1,C7)						
11	5	*md*	6	=IF(C8<3,C8+12,C8)						
12	6	*A*	20	=TRUNC(C10/100)						
13	7	*B*	-13	=IF((C7>1582)+((C7=1582)*(C8>10))+((C7=1582)*(C8=10)*(C9>=15)),2-C12+TRUNC(C12/4),0)						
14	8	*C*	733787	=IF(C10<0,TRUNC((365.25*C10)-0.75),TRUNC(365.25*C10))						
15	9	*D*	214	=TRUNC(30.6001*(C11+1))						
16	10	JD	2455002.25	=C13+C14+C15+C9+1720994.5						

Figure 3. Finding the Julian date corresponding to the Greenwich calendar date of 6 pm on 19 June 2009.

5 Converting the Julian date to the Greenwich calendar date

It is sometimes necessary to convert a given Julian date into its counterpart in the Gregorian calendar, i.e. the calendar date at Greenwich. As mentioned in the previous section, the calendar date at Greenwich is not necessarily the same as the local calendar date where you are, but depends upon the local time, your time-zone correction, and the number of hours (if any) of daylight saving in operation. We will discuss this further in Section 9.

The method shown here works for all dates from 1 January 4713 BC†. For example, let us find the calendar date at Greenwich corresponding to the Julian date JD = 2 455 002.25.

Method	*Example*		
1. Add 0.5 to JD.	JD+0.5	=	2 455 002.75
Set I = integer part and F = fractional part.	I	=	2 455 002
	F	=	0.75
2. If I is larger than 2 299 160, calculate:			
(i) $A = \mathrm{TRUNC}\left(\frac{I - 1\,867\,216.25}{36\,524.25}\right)$;	A	=	16.0
(ii) $B = I + A - \mathrm{TRUNC}(A/4) + 1$.	B	=	2 455 015.0
Otherwise, set $B = I$.			
3. Calculate $C = B + 1524$.	C	=	2 456 539.0
4. Calculate $D = \mathrm{TRUNC}\left(\frac{C - 122.1}{365.25}\right)$.	D	=	6 725.0
5. Calculate $E = \mathrm{TRUNC}(365.25 \times D)$.	E	=	2 456 306.0
6. Calculate $G = \mathrm{TRUNC}\left(\frac{C - E}{30.6001}\right)$.	G	=	7.0
7. Calculate $d = C - E + F - \mathrm{TRUNC}(30.6001 \times G)$. This is the day of the month including the decimal part of the day.	d	=	**19.75**
8. Calculate $m = G - 1$ if G is less than 13.5, or $m = G - 13$ if G is more than 13.5. This is the month number.	m	=	**6**
9. Calculate $y = D - 4716$ if m is more than 2.5, or $y = D - 4715$ if m is less than 2.5. This is the calendar year.	y	=	**2009**

Hence the date at Greenwich in the Gregorian calendar is 2009 June 19.75, or 6 pm on 19 June of that year. Figure 4 shows the spreadsheet for this calculation. The single input value is the Julian date, entered in cell C3, and the three output values, the day (including the fraction), month and year of the corresponding calendar date at Greenwich, appear in cells H3, H4 and H5 respectively.

The spreadsheet is called JDCD, corresponding to the acronym for Julian Date to Calendar Date conversion. We have also supplied spreadsheet functions to carry out the same calculations as formulas in a spreadsheet. There are three of them since a single function can only return a single value, and we need three, i.e. the day, the month and the year. The function names are respectively JDCDay(JD), JDCMonth(JD) and JDCYear(JD), and each takes the single argument JD which must be set equal to the Julian date. You can replace the calculation part of the spreadsheet shown in Figure 4 with these three functions by deleting rows 7 to 17 and replacing cells H3, H4, and H5 by the formulas =JDCDay(C3), =JDCMonth(C3) and =JDCYear(C3) respectively. Try it for yourself, but remember to save the spreadsheet first.

†See the previous section about the meaning of the term BC.

	A	B	C	D	E	F	G	H	I
1	**Julian date to Greenwich calendar date conversion**								
2									
3	*Input*	**Julian date**	**2455002.25**			*Output*	**Greenwich day**	**19.75**	=C15
4							**Greenwich month**	**6**	=C16
5							**Greenwich year**	**2009**	=C17
6									
7	1	*I*	2455002	=TRUNC(C3+0.5)					
8	2	*F*	0.75	=C3+0.5-C7					
9	3	*A*	16	=TRUNC((C7-1867216.25)/36524.25)					
10	4	*B*	2455015	=IF(C7>2299160,C7+1+C9-TRUNC(C9/4),C7)					
11	5	*C*	2456539	=C10+1524					
12	6	*D*	6725	=TRUNC((C11-122.1)/365.25)					
13	7	*E*	2456306	=TRUNC(365.25*C12)					
14	8	*G*	7	=TRUNC((C11-C13)/30.6001)					
15	9	*d*	19.75	=C11-C13+C8-TRUNC(30.6001*C14)					
16	10	*m*	6	=IF(C14<13.5,C14-1,C14-13)					
17	11	*y*	2009	=IF(C16>2.5,C12-4716,C12-4715)					

Figure 4. Finding the calendar date at Greenwich corresponding to the Julian date of 2 455 002.25.

6 Finding the name of the day of the week

It is sometimes useful to know on what day of the week a particular date will fall. For instance, you might want to know whether your birthday will be on Sunday next year, or – perhaps working out your holiday entitlement around Christmas – which day of the week corresponds to Christmas Day. This can be found easily from the Julian date using the following calculation in which we find the name of the day of the week corresponding to 19 June 2009 at Greenwich as an example.

Method		*Example*		
1.	Find the Julian date corresponding to midnight at Greenwich (§4).			2009 June 19.0
		JD	=	2 455 001.5
2.	Calculate $A = \left(\frac{\text{JD}+1.5}{7}\right)$.	A	=	350 714.714286
3.	Take the fractional part of A, multiply by 7, and round to the nearest integer.[a] This is the weekday number n as follows:	Fractional part	=	0.714 286
		n	=	**5**
	Sunday $n = 0$ Monday $n = 1$ Tuesday $n = 2$ Wednesday $n = 3$ Thursday $n = 4$ Friday $n = 5$ Saturday $n = 6$			**Friday**

[a]This may be done by taking TRUNC(fractional part+0.5).

The spreadsheet for this calculation, FDOW (Finding the Day Of the Week; Figure 5) selects the name corresponding to the weekday number using a nested IF formula at row 7 (that is quite long and confusing to read!). The test argument of the first IF is the first argument, C6=0. If this is true, then the formula returns Sunday. If not, then a second IF statement takes the place of the third argument and the value of C6 is tested against 1 (Monday), and so on until all seven possible values of n have been tested. If the formula has still not been satisfied at that point, the text ** error is returned. This should never happen!

In row 5 we have used the INT function for finding the integer part of the argument. Since the Julian date is always positive, there is no issue here about exactly how the INT function deals with negative values. We make extensive use of INT throughout the book. Row 5 ensures that the fraction of the day after midnight is removed from the Julian date before proceeding with the calculation.

We have also supplied a spreadsheet function called FDOW(JD) which does the same calculation. It returns the text corresponding to the name of the day of the given Julian date (JD), or the text Unknown if the calculation suggests n lies outside of the range 0 to 6 inclusive. You can delete rows 5 to 7 of Figure 5 and replace cell H3 with the formula =FDOW(C3). Don't forget to save the spreadsheet first if you want to try this out.

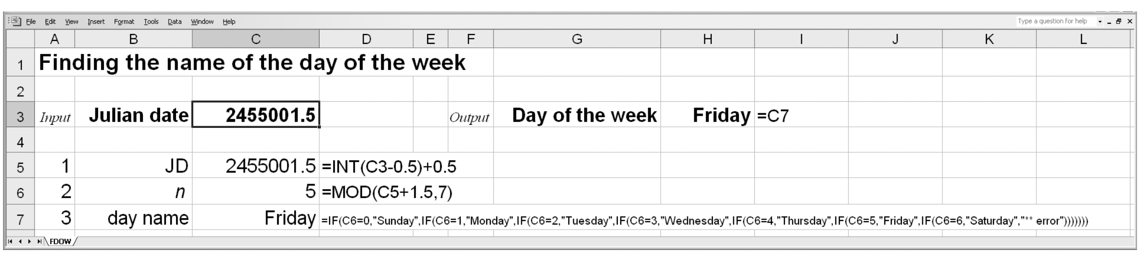


Figure 5. The Julian date 2 455 001.5 fell on a Friday at Greenwich.

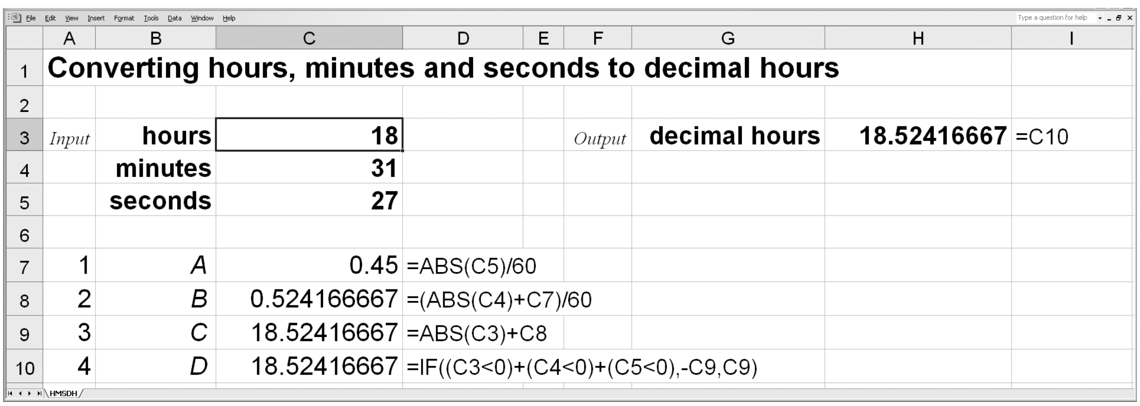
7 Converting hours, minutes and seconds to decimal hours

Most times are expressed as hours and minutes, or hours, minutes and seconds. For example, twenty to four in the afternoon may be written as 3:40 pm, or 3h 40m pm, or on a 24-hour clock as 15h 40m. In calculations, however, the time needs to be expressed in decimal hours on a 24-hour clock. The method of converting a time expressed in the format hours, minutes and seconds into decimal hours is given below. Some calculators have special keys to do this for you automatically. As an example, let's convert the time 6h 31m 27s pm into decimal hours.

Method		*Example*		
1.	Take the number of seconds and divide by 60.	27/60	=	0.450 000
2.	Add this to the number of minutes and divide by 60.	31.45/60	=	0.524 167
3.	Add the number of hours.	+6.0	=	6.524 167
4.	If the time has been given on a 12-hour clock, and it is pm, add 12.	+12.0	=	**18.524 167 hours**

The spreadsheet corresponding to this calculation is shown in Figure 6, and is called HMSDH (Hours Minutes Seconds to Decimal Hours conversion). We have defined variable names *A*, *B*, *C* and *D* in column B rows 7 to 10 for convenience. They have no counterparts in the method table above. Note that the spreadsheet converts the time already expressed on a 24-hour clock, so be careful to add 12 hours, if appropriate, to the number you enter in cell C3.

We have also supplied the spreadsheet function HMSDH(H,M,S) which will carry out this conversion for you. The three arguments correspond to the hours, minutes and seconds parts of the time to be converted to hours. You can delete rows 7 to 10 of the spreadsheet shown in Figure 6 and insert the formula =HMSDH(C3,C4,C5) in cell H3. Save a copy of your spreadsheet first. Note that in many cases, as here, you can use the function to convert partially-converted times. Thus =HMSDH(18,31,27) will give the same result as =HMSDH(18,31.524167,0), where you have expressed the same time in hours and minutes format (no seconds).



	A	B	C	D	E	F	G	H	I
1	**Converting hours, minutes and seconds to decimal hours**								
2									
3	*Input*	**hours**	**18**			*Output*	**decimal hours**	**18.52416667**	=C10
4		**minutes**	**31**						
5		**seconds**	**27**						
6									
7	1	*A*	0.45	=ABS(C5)/60					
8	2	*B*	0.524166667	=(ABS(C4)+C7)/60					
9	3	*C*	18.52416667	=ABS(C3)+C8					
10	4	*D*	18.52416667	=IF((C3<0)+(C4<0)+(C5<0),-C9,C9)					

Figure 6. Converting a time expressed in HMS format into decimal hours.

8 Converting decimal hours to hours, minutes and seconds

When the result of a calculation is a time, it is normally expressed as decimal hours, and we need to convert it to hours, minutes and seconds. (This is the reverse of the calculation in Section 7.) The method of doing so is given below. Again, some calculators have special keys to carry out this function automatically. We express the time 18.524 167 h in hours, minutes and seconds format as our example.

Method		*Example*		
1.	Take the fractional part and multiply by 60. The integer part of the result is the number of minutes.	$0.524\,167 \times 60$	=	**31**.450020
2.	Take the fractional part of the result and multiply by 60. This gives the number of seconds.	$0.450\,020 \times 60$	=	**27**.001 200 **18h 31m 27s**

The spreadsheet for this calculation is shown in Figure 7 and is called DHHMS (Decimal Hours to Hours Minutes Seconds conversion). It has more steps and slightly greater complexity than the method in the above table as it needs to deal automatically with cases in which the result of the calculation is an integer number of minutes and/or seconds exactly equal to 60, such as 10h 45m 60s. In such cases, you would increment the number of hours and/or minutes by 1, and set the number of minutes and/or seconds to zero. Thus 10h 45m 60s is better expressed as 10h 46m 0s. The spreadsheet also rounds the number of seconds to two decimal places using the spreadsheet intrinsic function ROUND in cell C9. The first argument of this function is the number you wish to round, and the second argument is the number of decimal places.

We have also supplied three spreadsheet functions to carry out this calculation. We need to have three as any function can only return one result, and so we need separate functions for the hours, minutes, and seconds. These are DHHour(H), DHMin(H), and DHSec(H) respectively, where the argument in each case is the time in decimal hours to be converted. Thus (having saved a copy first) you could delete rows 7 to 14 of the spreadsheet shown in Figure 7 and insert the formulas =DHHour(C3), =DHMin(C3) and =DHSec(C3) in cells H3, H4 and H5 respectively to get the same result.

	A	B	C	D	E	F	G	H	I
1	**Converting decimal hours to hours, minutes and seconds**								
2									
3	*Input*	**decimal hours**	**18.52416667**			*Output*	**hours**	**18**	=C14
4							**minutes**	**31**	=C12
5							**seconds**	**27**	=C10
6									
7	1	unsigned decimal	18.52416667	=ABS(C3)					
8	2	total seconds	66687	=C7*3600					
9	3	seconds (2 dp)	27	=ROUND(MOD(C8,60),2)					
10	4	corrected seconds	27	=IF(C9=60,0,C9)					
11	5	corrected remainder	66687	=IF(C9=60,C8+60,C8)					
12	6	minutes	31	=MOD(TRUNC(C11/60),60)					
13	7	unsigned hours	18	=TRUNC(C11/3600)					
14	8	signed hours	18	=IF(C3<0,-1*C13,C13)					

Figure 7. Converting a time expressed in decimal hours to HMS format.

9 Converting the local time to universal time (UT)

The basis of civilian time-keeping is the rotation of the Earth. **Universal time** (UT) is related to the motion of the Sun as observed on the **Greenwich meridian**, longitude 0°. The Earth is not a perfect time-keeper, however, and today a more uniform flow of time is available from atomic clocks. **International atomic time** (TAI) is the scale resulting from analyses by the Bureau International de l'Heure, in Paris, of atomic standards in many countries. A version of universal time, called **coordinated universal time** (UTC), is derived from TAI in such a manner as to be within 0.9 seconds of UT and a whole number of seconds different from TAI. (In June 2010, TAI−UTC = 34 s). This is achieved by including occasional leap seconds in UTC (at the end of June or December – usually the latter). UTC is the time broadcast by some national radio stations (the 'time pips') and by standard time transmission services such as DCF 77 (Mainflingen, Germany), MSF 60 (Anthorn, UK) and WWV (Colorado, USA). It is now the basis of legal time-keeping on the Earth. UTC is thus an atomic time standard (and hence as uniform as we know how to measure) but with discontinuities to keep it in line with the irregular rotation of our planet.

Another time in common use today is **GPS time**. This is an atomic time kept by the US Naval Observatory, and which is broadcast by the satellites of the global positioning system (GPS). GPS time was equal to UTC on 1980 January 6 0.0, but, unlike UTC, is not adjusted by the insertion of leap seconds. Hence GPS time is equal, in June 2010, to UTC + 15 seconds (kept to within a microsecond) and is the time you can extract from your GPS navigation device.

The amateur astronomer need not be too concerned by all this complexity. For our purposes, we can take UT = UTC = GMT without noticing the difference. (Note that in a pre-1925 definition **Greenwich Mean Time** (GMT) started at midday, so was 12 hours out with respect to UT. However, this distinction is usually overlooked and people refer to UTC and GMT as the same thing. For example, the BBC World Service gives UTC times as GMT.) Where we need greater accuracy, we will use terrestrial time (TT) for events after 1984 January 0.0, and ephemeris time (ET) before then. TT is equal to TAI + 32.184 seconds and took over from ET at the beginning of 1984 (see Section 16). (Note that TT was called **terrestrial dynamic time**, TDT, until 1991, when it was renamed by the International Astronomical Union.) UT is used as the local civil time in Britain during the winter months, but 1 hour is added during the summer to form British summer time (BST) so that the working day fits more conveniently into daylight hours. Many other countries adopt a similar arrangement; sometimes the converted time is known as **daylight saving time**.

Countries lying on meridians east or west of Greenwich do not use UT as their local civil time. It would be impractical to do so as the local noon, the time at which the Sun reaches its maximum altitude, gets earlier or later with respect to the local noon on the Greenwich meridian as one moves east or west respectively. The world is therefore divided into time zones, each zone usually corresponding to a whole number of hours before or after UT, and small countries, or parts of large countries lying within a zone, adopt the zone time as their local civil time (see Figure 8).

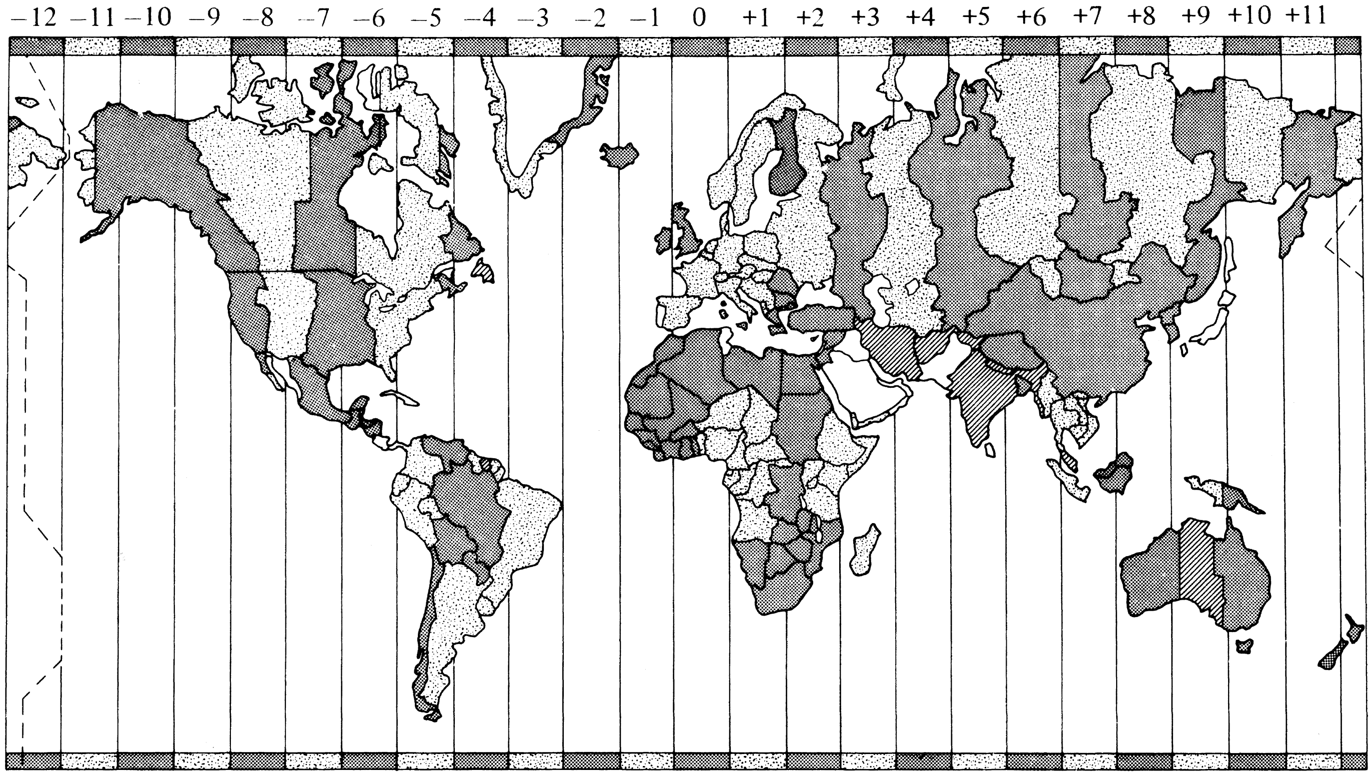


Figure 8. International time zones. This small-scale map can show only the general distribution of time zones around the world. If you are unsure of your own zone correction, you can check it by looking on the Internet, or by tuning your short-wave radio to the BBC World Service and comparing your watch with the time pips broadcast every hour from London.

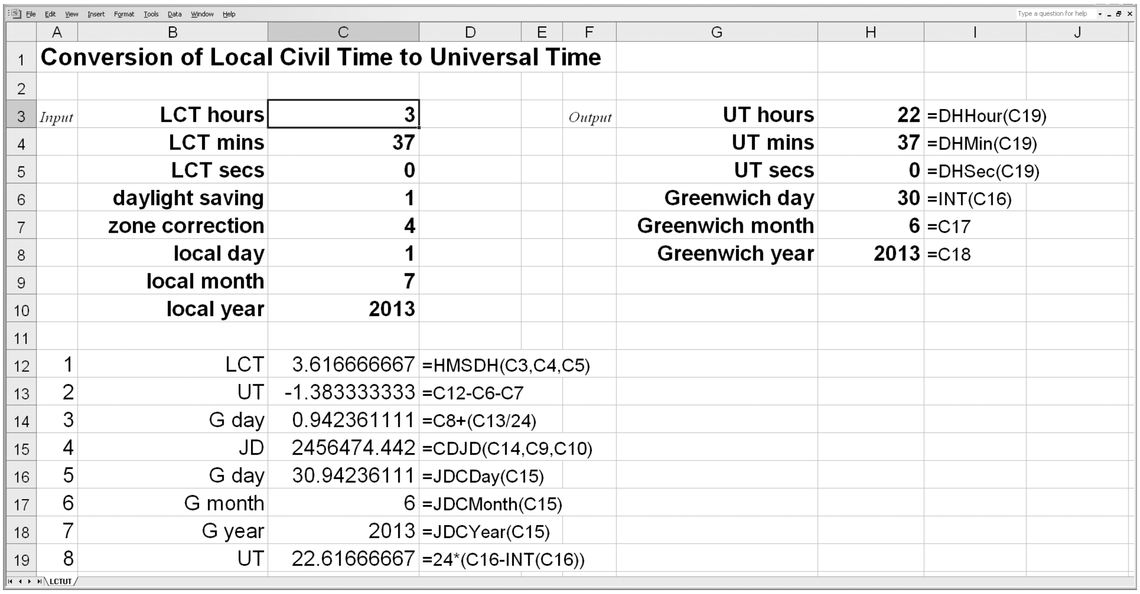
The starting point for many astronomical calculations is often the local time and date, that is the time on your watch (assumed to be correct) on the date of the calendar on your wall. We will refer to your local time as the **local civil time**, and the local date as the **local calendar date**. However, the algorithms for calculating the positions of the heavenly bodies usually begin with the time on the Greenwich meridian, **universal time** (UT), and the Greenwich calendar date. We therefore need to be able to convert times and dates from your local position to Greenwich and vice-versa. For this you need to know your time-zone correction (hours ahead of UTC) and whether or not there is daylight saving in operation.

The following method converts your local time and date into UT and Greenwich calendar date. As an example we convert daylight saving time 3h 37m in time zone +4 hours on 1 July 2013.

	Method	*Example*		
1.	Convert local civil time to zone time	3h 37m − 1h	=	2h 37m
	by removing the daylight saving correction,			
	and convert to decimal hours (§7).	Zone time	=	2.616 667 hours
2.	Subtract the time-zone offset (time zones	UT	=	$2.616\,667 - 4$
	W are negative). This is UT.		=	$-1.383\,333$ hours
3.	Divide UT by 24 and add to the local	G Day	=	$1-(1.383\,333/24.0)$ hours
	calendar day. This is Greenwich calendar day.		=	0.942 361 hours
4.	Find the Julian date corresponding to the	G cal date	=	2013 July 0.942 361
	Greenwich calendar date (§4).	JD	=	2 456 474.442
5.	Convert the Julian calendar date back	G Day	=	30.942 361
	into the Greenwich calendar date (§5).	G Month	=	6
		G Year	=	2013
6.	The day of the Greenwich calendar date is	GD	=	30
	TRUNC(G Day). Subtract this from G Day	UT	=	$0.942\,361 \times 24$
	and multiply the result by 24 to obtain		=	22.616 667
	UT in the range 0 to 24 h. Convert to hours,		=	**22h 37m 0s**
	minutes and seconds if required (§8).	G date	=	**2013 June 30**

Steps 4 and 5 of the method table above may seem a bit unnecessary. What is the point of going through the lengthy conversion from Greenwich calendar date in step 4 only to be told in step 5 to convert back again? Actually, with your human mind carrying out this calculation you may be able to go directly from step 3 to step 6 because you will be able to see that G Day = 0.942 361 is the same as G Day $= 0 + 0.942\,361$, and the day therefore corresponds to the previous day's date, i.e. 30 June, and the UT to $0.942\,361 \times 24$. Note that you have made quite a complicated calculation in doing this, and of course the year might have changed as well. Steps 4 and 5, though cumbersome, take care of all of this, and are required in any case in the spreadsheet (Figure 9).

The spreadsheet is called LCTUT, following the acronym for Local Civil Time to Universal Time conversion. The spreadsheet functions corresponding to this calculation are LCTUT, LCTGDay, LCTGMonth and LCTGYear, returning the universal time, and the day, month and year of the Greenwich calendar date respectively. Each of them takes the same eight arguments: (H,M,S,DS,ZC,LD,LM,LY), in which H, M, S are the local time (hours, minutes, seconds), DS and ZC are the daylight saving offset and zone correction (hours), and LD, LM, LY are the day, month and year of the local calendar date.



	A	B	C	D	E	F	G	H	I	J
1	**Conversion of Local Civil Time to Universal Time**									
2										
3	*Input*	**LCT hours**	**3**			*Output*	**UT hours**	**22**	=DHHour(C19)	
4		**LCT mins**	**37**				**UT mins**	**37**	=DHMin(C19)	
5		**LCT secs**	**0**				**UT secs**	**0**	=DHSec(C19)	
6		**daylight saving**	**1**				**Greenwich day**	**30**	=INT(C16)	
7		**zone correction**	**4**				**Greenwich month**	**6**	=C17	
8		**local day**	**1**				**Greenwich year**	**2013**	=C18	
9		**local month**	**7**							
10		**local year**	**2013**							
11										
12	1	LCT	3.616666667	=HMSDH(C3,C4,C5)						
13	2	UT	-1.383333333	=C12-C6-C7						
14	3	G day	0.942361111	=C8+(C13/24)						
15	4	JD	2456474.442	=CDJD(C14,C9,C10)						
16	5	G day	30.94236111	=JDCDay(C15)						
17	6	G month	6	=JDCMonth(C15)						
18	7	G year	2013	=JDCYear(C15)						
19	8	UT	22.61666667	=24*(C16-INT(C16))						

Figure 9. Converting local time and date to universal time and Greenwich date.

Having saved a copy of the spreadsheet of Figure 9, you could delete rows 12 to 19 and insert these spreadsheet functions in cells H3 to H8 as follows:

```
=DHHour(LCTUT(C3,C4,C5,C6,C7,C8,C9,C10))
=DHMin(LCTUT(C3,C4,C5,C6,C7,C8,C9,C10))
=DHSec(LCTUT(C3,C4,C5,C6,C7,C8,C9,C10))
=LCTGDay(C3,C4,C5,C6,C7,C8,C9,C10)
=LCTGMonth(C3,C4,C5,C6,C7,C8,C9,C10)
=LCTGYear(C3,C4,C5,C6,C7,C8,C9,C10).
```

Note that the first three of these use nested functions, e.g. the function DHHour takes as its argument the result of running the function LCTUT. You can nest functions in this way almost indefinitely, although the resulting formula rapidly becomes unreadable as the nesting gets deeper.

10 Converting UT and Greenwich calendar date to local time and date

The result of an astronomical calculation can sometimes be a time and a date, usually the UT and calendar date at Greenwich. The following method will convert to the corresponding local civil time and calendar date appropriate to a point on the Earth in a given time zone, with or without daylight saving in operation. As in the previous section, the local date and the Greenwich date may not be the same, and we need to take account of differences in dates spanning month and/or year boundaries. Continuing with the previous example, what is the local civil time and local calendar date corresponding to 22h 37m UT when the Greenwich calendar date is 30 June 2013, in time zone +4 h and with daylight saving in operation?

Method		*Example*		
1.	Convert UT to decimal hours (§7).	22h 37m	=	22.616 667 hours
2.	Add the time zone offset (time zones W are negative)	LCT	=	$22.616\,667 + 4 + 1$
	and the daylight saving offset. This is the local civil time.		=	27.616 667
3.	Find the Julian date corresponding to the Greenwich	LJD	=	2 465 473.5
	calendar date (§4) and add (LCT/24).		+	27.616 667/24
			=	2 456 474.651
4.	Convert this local Julian date back into the local calendar	L Day	=	1.150 694
	date (§5). Take the integer part to get the local day number.	L date	=	**2013 July 1**
5.	Subtract the integer day from L Day and multiply the result	LCT	=	$0.150\,694 \times 24$
	by 24 to obtain the local civil time in the range 0 to 24 h.		=	3.616 667
	Convert to hours, minutes and seconds if required (§8).		=	**3h 37m 0s**

A word here about rounding errors. In the method examples of both this and the previous section, you may have become aware of small differences in the last one or two decimal places between your calculated values and those shown in the method tables. For example, if we put 0.150 694 into a calculator (step 5) and multiply by 24, we get 3.616 656 instead of 3.616 667 as shown. This is because of rounding errors, and there are two causes. First, the calculator maintains calculations accurate to about 11 or 12 significant figures, but in steps 3 and 4 we 'use up' seven of those in specifying the integer part of the Julian date, leaving only 4 or 5 for the fractional part. The calculator does its best, but the error on the last place creeps in and shows itself as a discrepancy. The spreadsheet calculation usually has much higher precision so does not suffer from this particular problem. We have shown full-precision results in the tables, rounded to six decimal places. Second, we have displayed the results of each calculation only to six decimal places. The truncation can make a small difference as here. With nine places of decimals, the value of LCT in step 5 is 0.150 694 444. Multiply this by 24 and round to six decimal places and you get 3.616 667 as shown.

As in the method of the previous section, you may be able to leave out steps 3 and 4 which are included to make sure that the month and year boundaries are properly dealt with. You can see that the value of LCT in the second step, 27.616 667 h, is equivalent to 1 day (24 h) plus 3.616 667 h. The local civil time is therefore 3.616 667 h = 3h 37m, and the local date is the Greenwich date plus one day, so 30 June 2013 becomes 1 July 2013.

The spreadsheet for this section is shown in Figure 10 and is called UTLCT (Universal Time to Local Civil Time conversion). Not having the advantage of the intelligence of the human brain, the program has to carry out the conversions to and from the Julian date (rows 15 and 16) for every calculation in order to deal properly with the month and year boundaries. In this case, without these steps, the spreadsheet would report the local date as 31 June 2013 – logically correct but not a recognised date for June which has only 30 days.

The corresponding spreadsheet functions are UTLCT, UTLCDay, UTLCMonth and UTLCYear, which return respectively the local civil time in hours, the day, the month, and the year of the local calendar date. Each takes the same eight arguments (H,M,S,DS,ZC,GD,GM,GY) in which H, M and S are the universal time (hours, minutes, seconds), DS and ZC are the daylight saving adjustment and zone correction (both in hours), and GD, GM and GY are the day, month and year of the Greenwich calendar date.

	A	B	C	D	E	F	G	H	I	J
1	**Conversion of Universal Time to Local Civil Time**									
2										
3	*Input*	**UT hours**	**22**			*Output*	**LCT hour**	**3**	=DHHour(C20)	
4		**UT mins**	**37**				**LCT Min**	**37**	=DHMin(C20)	
5		**UT secs**	**0**				**LCT sec**	**0**	=DHSec(C20)	
6		**daylight saving**	**1**				**day**	**1**	=C17	
7		**zone correction**	**4**				**month**	**7**	=C18	
8		**Greenwich day**	**30**				**year**	**2013**	=C19	
9		**Greenwich month**	**6**							
10		**Greenwich year**	**2013**							
11										
12	1	UT	22.61666667	=HMSDH(C3,C4,C5)						
13	2	zone time	26.61666667	=C12+C7						
14	3	local time	27.61666667	=C13+C6						
15	4	local JD+local time	2456474.651	=CDJD(C8,C9,C10)+(C14/24)						
16	5	local day	1.150694444	=JDCDay(C15)						
17	6	integer day	1	=TRUNC(C16)						
18	7	local month	7	=JDCMonth(C15)						
19	8	local year	2013	=JDCYear(C15)						
20	9	LCT	3.616666663	=24*(C16-C17)						

Figure 10. Converting universal time and Greenwich date to local civil time and local date.

You can therefore delete rows 12 to 20 (save a copy first) and insert the following formulas in cells H3 to H8 respectively:

```
=DHHour(UTLCT(C3,C4,C5,C6,C7,C8,C9,C10))
=DHMin(UTLCT(C3,C4,C5,C6,C7,C8,C9,C10))
=DHSec(UTLCT(C3,C4,C5,C6,C7,C8,C9,C10))
=UTLCDay(C3,C4,C5,C6,C7,C8,C9,C10)
=UTLCMonth(C3,C4,C5,C6,C7,C8,C9,C10)
=UTLCYear(C3,C4,C5,C6,C7,C8,C9,C10).
```

Note that the first three of these use nested functions, e.g. the function DHHour takes as its argument the result of running the function UTLCT. You can nest functions in this way almost indefinitely, although the resulting formula rapidly becomes unreadable as the level of nesting increases.

11 Sidereal time (ST)

Universal time (UT), and therefore the local civil time in any part of the world, is related to the apparent motion of the Sun around the Earth. Roughly speaking, we may take 1 **solar day** as the time between two successive passages of the Sun across the meridian as observed at a particular place. Astronomers are interested, however, in the motion of the stars; in particular they need to use a clock whose rate is such that any star is observed to return to the same position in the sky after exactly 24 hours have elapsed according to the clock. Such a clock is called a sidereal clock and its time, being regulated by the stars, is called **sidereal time** (ST). Solar time, of which UT is an example, is not the same as sidereal time because during the course of 1 solar day the Earth moves nearly 1 degree along its orbit round the Sun. Hence, the Sun appears progressively displaced against the background of stars when viewed from the Earth; turning that around, the stars appear to move with respect to the Sun. Any clock, therefore, which keeps time by the Sun does not do so by the stars.

There are about 365.25 solar days in the year†, the time taken by the Sun to return to the same position with respect to the background of stars. During this period, the Earth makes about 366.25 revolutions around its own axis; there are therefore this many sidereal days in the year. Each sidereal day is thus slightly shorter than the solar day, 24 hours of sidereal time corresponding to 23h 56m 04s of solar time. Universal time and Greenwich sidereal time agree at one instant every year at the autumnal **equinox** (around 22 September). Thereafter, the difference between them grows in the sense that sidereal time runs faster than universal time, until exactly half a year later the difference is 12 hours. After 1 year, the times again agree.

The formal definition of sidereal time is that it is the hour angle of the vernal equinox (see Section 18).

12 Conversion of UT to Greenwich sidereal time (GST)

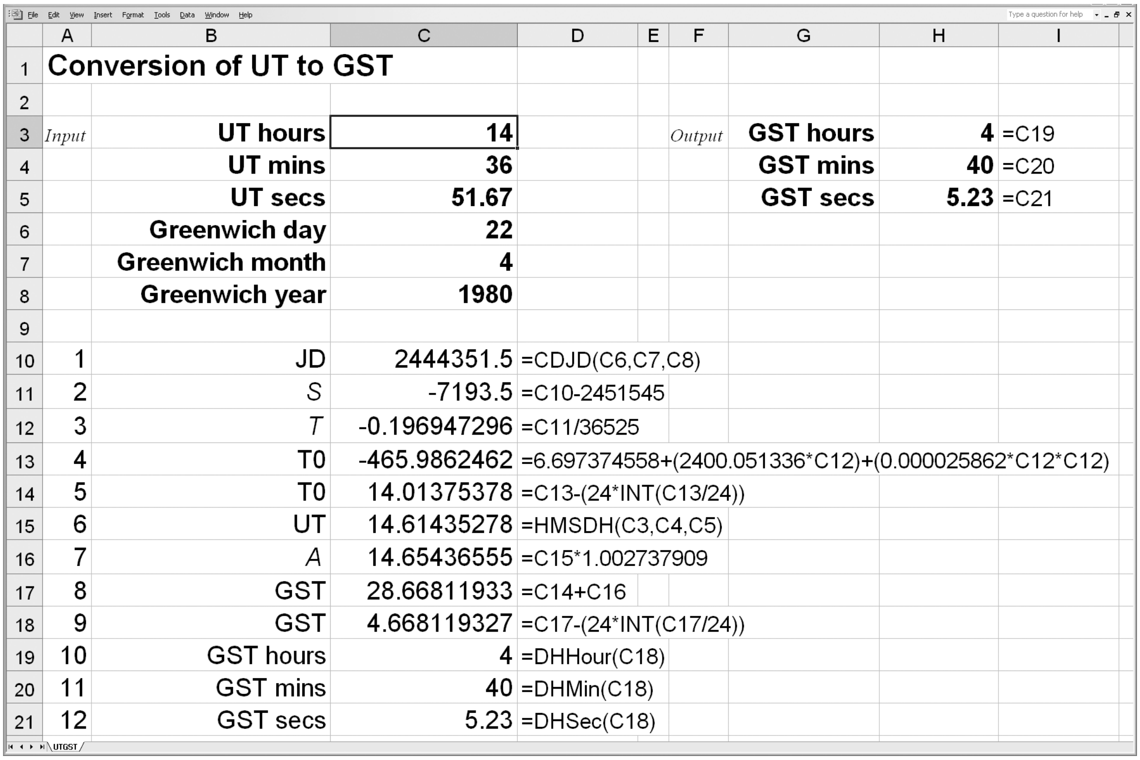
This section describes a simple procedure by which the UT may be converted into GST. It is accurate to better than one tenth of a second. For example, what was the GST at 14h 36m 51.67s UT on Greenwich date 22 April 1980?

	Method	*Example*		
1.	Find the Julian date corresponding to 0h on this Greenwich calendar date (§4).	JD	=	2 444 351.5
2.	Calculate $S = \mathrm{JD} - 2\,451\,545.0$.	S	=	−7 193.5
3.	Calculate $T = S/36\,525.0$.	T	=	−0.196 947
4.	Find $T0 = 6.697\,374\,558 + (2\,400.051\,336 \times T)$	$T0$	=	−465.986 246
	$+ (0.000025862 \times T^2)$. Reduce the result to the		+	24×20
	range 0 to 24 by adding or subtracting multiples of 24.	$T0$	=	14.013 754
5.	Convert UT to decimal hours (§7).	UT	=	14.614 353
6.	Multiply UT by 1.002 737 909.	A	=	14.654 366
7.	Add this to $T0$ and reduce to the range 0 to 24 if necessary		+	14.013 754
	by subtracting or adding 24. This is the GST.	GST	=	4.668 120
8.	Convert the result to hours, minutes and seconds (§8).	GST	=	**4h 40m 5.23s**

The spreadsheet for this calculation is shown in Figure 11 and is called UTGST (an acronym for UT to GST conversion). The step of reducing to the range 0 to 24 is achieved, for example in row 14, by subtracting (24×INT(C13/24)) from C13. The INT function returns the whole number of times that 24 goes into the value of C13, and this is multiplied by 24 before being subtracted from the value in C13, just as is done in step 4 of the method table. This trick is used in many spreadsheets throughout the book.

We have also supplied the spreadsheet function UTGST(H,M,S,GD,GM,GY) which takes six arguments H, M, S (UT in hours, minutes and seconds) and GD, GM, GY (Greenwich calendar date as days, months, and years). It returns the GST in hours corresponding to the values of the arguments.

†See the definition of the year given in the Glossary.



	A	B	C	D	E	F	G	H	I
1	Conversion of UT to GST								
2									
3	*Input*	**UT hours**	**14**			*Output*	**GST hours**	**4**	=C19
4		**UT mins**	**36**				**GST mins**	**40**	=C20
5		**UT secs**	**51.67**				**GST secs**	**5.23**	=C21
6		**Greenwich day**	**22**						
7		**Greenwich month**	**4**						
8		**Greenwich year**	**1980**						
9									
10	1	JD	2444351.5	=CDJD(C6,C7,C8)					
11	2	*S*	-7193.5	=C10-2451545					
12	3	*T*	-0.196947296	=C11/36525					
13	4	T0	-465.9862462	=6.697374558+(2400.051336*C12)+(0.000025862*C12*C12)					
14	5	T0	14.01375378	=C13-(24*INT(C13/24))					
15	6	UT	14.61435278	=HMSDH(C3,C4,C5)					
16	7	*A*	14.65436555	=C15*1.002737909					
17	8	GST	28.66811933	=C14+C16					
18	9	GST	4.668119327	=C17-(24*INT(C17/24))					
19	10	GST hours	4	=DHHour(C18)					
20	11	GST mins	40	=DHMin(C18)					
21	12	GST secs	5.23	=DHSec(C18)					

Figure 11. Converting universal time and Greenwich date to Greenwich sidereal time.

You can try this for yourself by deleting rows 10 to 21 (after saving a copy) and inserting the following formulas in cells H3, H4 and H5 respectively:

```
=DHHour(UTGST(C3,C4,C5,C6,C7,C8))
=DHMin(UTGST(C3,C4,C5,C6,C7,C8))
=DHSec(UTGST(C3,C4,C5,C6,C7,C8)).
```

13 Conversion of GST to UT

Here we deal with the reverse problem of the previous section, namely that of converting a given GST into the corresponding UT. The problem is complicated, however, by the fact that the sidereal day is slightly shorter than the solar day so that on any given calendar date a small range of sidereal times occurs twice. This range is about 3m 56s long, the sidereal times corresponding to UT 0h to 0h 3m 56s occurring again from UT 23h 56m 04s to midnight (see Figure 12). The method given here correctly converts sidereal times in the former interval, but not in the latter.

The accuracy of this method is the same as that of Section 12, namely better than one tenth of a second. Continuing our previous example, at GST = 4h 40m 5.23s on Greenwich date 22 April 1980, what was the UT?

Method		*Example*		
1.	Find the Julian date corresponding to 0h on this Greenwich calendar date (§4).	JD	=	2 444 351.5
2.	Calculate S =JD−2 451 545.0.	S	=	−7 193.5
3.	Calculate $T = S/36\,525.0$.	T	=	−0.196 947
4.	Find $T0 = 6.697\,374\,558 + (2\,400.051\,336 \times T)$ + $(0.000\,025\,862 \times T^2)$. Reduce the result to the range 0 to 24 by adding or subtracting multiples of 24.	$T0$	=	−465.986 246
			+	24 × 20
		$T0$	=	14.013 754
5.	Convert GST to decimal hours (§7).	GST	=	4.668 119
6.	Subtract $T0$ and reduce to the range 0 to 24 if necessary by subtracting or adding 24.	A	=	−9.345 635
		B	=	14.654 365
7.	Multiply B by 0.997 269 566 3. The result is the UT.	UT	=	14.614 353
8.	Convert the result to hours, minutes and seconds (§8).	UT	=	**14h 36m 51.67s**

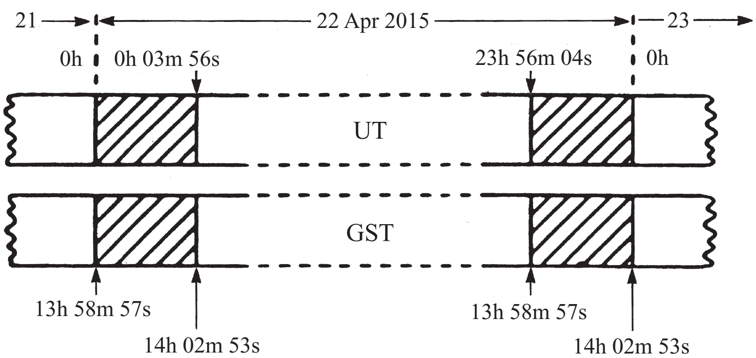


Figure 12. UT and GST for 22 April 2015. The hatched intervals of GST occur twice on the same day.

Figure 13 shows the corresponding spreadsheet, labelled GSTUT (an acronym for GST to UT conversion). It follows the method given in the table quite closely, but incorporates an extra step in row 22. This step tests to see whether the UT lies in the range 0h 0m 0s to 0h 3m 56s. If it does, it may not be the desired conversion since an equally-valid range of UT for this date corresponding to the given GST is 23h 56m 04s to 0h 0m 0s. There is insufficient information for us to be able to determine, on the GST and Greenwich calendar date alone, which is the desired result. The IF function in row 13 therefore issues a status flag, actually a word of text. This is OK if there is no ambiguity in the conversion, and Warning if there is.

The associated spreadsheet functions are:

GSTUT(H,M,S,GD,GM,GY) and
eGSTUT(H,M,S,GD,GM,GY),

where the arguments H, M, S represent the hours, minutes and seconds of the GST, and GD, GM, GY represent the day, month, year of the Greenwich calendar date. GSTUT returns the UT in hours corresponding with the argument values. You can use the other function, eGSTUT, to determine whether or not the conversion is ambiguous, and it returns status text OK or Warning as appropriate. Thus rows 10 to 22 can be

	A	B	C	D	E	F	G	H	I
1	**Conversion of GST to UT**								
2									
3	*Input*	**GST hours**	**4**			*Output*	**UT hours**	**14**	=C19
4		**GST mins**	**40**				**UT mins**	**36**	=C20
5		**GST secs**	**5.23**				**UT secs**	**51.67**	=C21
6		**Greenwich day**	**22**				**Status**	**OK**	=C22
7		**Greenwich month**	**4**						
8		**Greenwich year**	**1980**						
9									
10	1	JD	2444351.5	=CDJD(C6,C7,C8)					
11	2	*S*	-7193.5	=C10-2451545					
12	3	*T*	-0.196947296	=C11/36525					
13	4	T0	-465.9862462	=6.697374558+(2400.051336*C12)+(0.000025862*C12*C12)					
14	5	T0	14.01375378	=C13-(24*INT(C13/24))					
15	6	GST (hours)	4.668119444	=HMSDH(C3,C4,C5)					
16	7	*A*	-9.345634337	=C15-C14					
17	8	*B*	14.65436566	=C16-(24*INT(C16/24))					
18	9	UT	14.61435289	=C17*0.9972695663					
19	10	UT hours	14	=DHHour(C18)					
20	11	UT mins	36	=DHMin(C18)					
21	12	UT secs	51.67	=DHSec(C18)					
22	13	warning flag	OK	=IF(C18<(0.065574),"Warning","OK")					

GSTUT

Figure 13. Converting Greenwich date and Greenwich sidereal time to universal time.

deleted in the spreadsheet of Figure 13 (but save a copy first) and the following formulas inserted into cells H3 to H6 respectively:

```
=DHHour(GSTUT(C3,C4,C5,C6,C7,C8))
=DHMin(GSTUT(C3,C4,C5,C6,C7,C8))
=DHSec(GSTUT(C3,C4,C5,C6,C7,C8))
=eGSTUT(C3,C4,C5,C6,C7,C8).
```

Why not try this for yourself?

14 Local sidereal time (LST)

The Greenwich sidereal time discussed in the previous sections is the sidereal time correct for observations made on the Greenwich meridian, longitude 0°. It is in fact the local sidereal time (LST) for the Greenwich meridian. As you move west or east from longitude 0°, however, the local sidereal time gets earlier or later respectively because the hour angle of the vernal equinox, which defines the local sidereal time, changes. You can calculate your local sidereal time, given the Greenwich sidereal time, very easily as the difference between the two times in hours is simply the geographical longitude in degrees divided by 15. Longitudes west give local sidereal times earlier than GST, and longitudes east later. You should express longitudes E as positive numbers, and longitudes W as negative numbers. Take the example: what is the local sidereal time on the longitude 64° W when the GST is 4h 40m 5.23s?

Method		*Example*		
1.	Convert GST to decimal hours (§7).	GST	=	4.668 119 h
2.	Convert the geographical longitude in degrees to its equivalent in hours by dividing by 15. Note that longitudes W are negative.	−64°	=	−4.266 667 h
3.	Add this to the GST. Bring the result into the range 0 to 24 by adding or subtracting 24 if necessary. This is the local sidereal time (LST).	LST	=	0.401 453 h
4.	Convert LST to hours, minutes and seconds (§8).	LST	=	**0h 24m 5.23s**

The spreadsheet labelled GSTLST (an acronym for GST to LST conversion) is shown in Figure 14. The corresponding spreadsheet function is GSTLST. This takes four arguments with the Greenwich sidereal time in the first three arguments expressed as hours, minutes and seconds, and the geographical longitude of the observer in decimal degrees in the fourth argument. It returns the local sidereal time in decimal hours. Hence rows 8 to 14 of the spreadsheet shown in Figure 14 could be deleted (after saving a copy) and the following spreadsheet formulas inserted consecutively into cells H3, H4 and H5:

```
=DHHour(GSTLST(C3,C4,C5,C6))
=DHMin(GSTLST(C3,C4,C5,C6))
=DHSec(GSTLST(C3,C4,C5,C6)).
```

 https://doi.org/10.1017/CBO9780511861161.004

	A	B	C	D	E	F	G	H	I
1	**Conversion of GST to LST**								
2									
3	*Input*	**GST hours**	**4**			*Output*	**LST hour**	**0**	=C12
4		**GST mins**	**40**				**LST min**	**24**	=C13
5		**GST secs**	**5.23**				**LST sec**	**5.23**	=C14
6		**geog long**	**-64**						
7									
8	1	GST	4.668119444	=HMSDH(C3,C4,C5)					
9	2	offset	-4.266666667	=C6/15					
10	3	LST (hours)	0.401452778	=C8+C9					
11	4	LST (hours)	0.401452778	=C10-(24*INT(C10/24))					
12	5	LST hour	0	=DHHour(C11)					
13	6	LST min	24	=DHMin(C11)					
14	7	LST sec	5.23	=DHSec(C11)					

Figure 14. Converting Greenwich sidereal time to local sidereal time.

15 Converting LST to GST

This problem is the reverse of that treated in Section 14, namely, given the local sidereal time at a particular place, what is the corresponding Greenwich sidereal time? As an example, we shall calculate the GST when the LST on longitude 64° W is 0h 24m 5.23s.

Method	*Example*		
1. Convert the LST to decimal hours (§7).	LST	=	0.401 453 h
2. Convert the geographical longitude in degrees to its equivalent in hours by dividing by 15. Note that longitudes W are negative.	−64°	=	−4.266 667 h
3. Subtract this from the LST. Bring the result into the range 0–24 by adding or subtracting 24 if necessary. This is the GST.	GST	=	4.668 119 h
4. Convert GST to hours, minutes and seconds (§8).	GST	=	**4h 40m 5.23s**

The spreadsheet labelled LSTGST (an acronym for LST to GST conversion) is shown in Figure 15. The corresponding spreadsheet function is also called LSTGST. This takes four arguments with the local sidereal time in the first three arguments expressed as hours, minutes and seconds, and the geographical longitude of the observer in decimal degrees in the fourth argument. It returns the Greenwich sidereal time in decimal hours. Hence rows 8 to 14 of the spreadsheet shown in Figure 15 could be deleted (after saving a copy) and the following spreadsheet formulas inserted consecutively into cells H3, H4 and H5:

```
=DHHour(LSTGST(C3,C4,C5,C6))
=DHMin(LSTGST(C3,C4,C5,C6))
=DHSec(LSTGST(C3,C4,C5,C6)).
```

	A	B	C	D	E	F	G	H	I
1	**Conversion of LST to GST**								
2									
3	*Input*	**LST hours**	**0**			*Output*	**GST hours**	**4**	=C12
4		**LST mins**	**24**				**GST mins**	**40**	=C13
5		**LST secs**	**5.23**				**GST secs**	**5.23**	=C14
6		**geog long**	**-64**						
7									
8	1	GST	0.401452778	=HMSDH(C3,C4,C5)					
9	2	long (hours)	-4.266666667	=C6/15					
10	3	GST	4.668119444	=C8-C9					
11	4	GST	4.668119444	=C10-(24*INT(C10/24))					
12	5	GST hours	4	=DHHour(C11)					
13	6	GST mins	40	=DHMin(C11)					
14	7	GST secs	5.23	=DHSec(C11)					

Figure 15. Converting local sidereal time to Greenwich sidereal time.

16 Ephemeris time (ET) and terrestrial time (TT)

Universal and sidereal times are both tied directly to the period of the rotation of the Earth about its polar axis. The Earth is being used in effect as the balance wheel of a cosmic clock whose tick defines the length of the day. With the advent of extremely accurate atomic clocks, however, it has become apparent that the Earth's rotation is not strictly uniform but shows small erratic fluctuations which are not well understood. UT and ST, being reckoned by this irregular cosmic clock, are therefore not strictly uniform either. Astronomers need a system of time that *is* uniform since the theories of celestial mechanics assume that such a quantity exists. For example, two solid bodies in orbit about one another far away from any external influences should have an unchanging orbital period when measured on a regular clock. Before 1984, astronomers adopted **ephemeris time** (ET) for this purpose. It was calculated from the motion of the Moon and assumed to be uniform. Nowadays, atomic clocks provide the most uniform measure of time, and since 1984 **terrestrial time** (TT) has been used instead of ET. (In fact TT was called TDT, for **terrestrial dynamic time**, until renamed and slightly redefined by the International Astronomical Union in 1991.) TT is tied to the atomic time scale, TAI (see Section 9), by the equation:

$$\mathrm{TT} = \mathrm{TAI} + 32.184\,\mathrm{s}.$$

The constant offset of 32.184 seconds was chosen to make TT equal to ET at the beginning of 1984. ET itself was chosen to agree as nearly as possible with the measure of universal time during the nineteenth century, and it is unlikely that TT will differ by more than a few minutes in the twenty-first.

The primary unit of ET was the length of the tropical year at 1900 January 0.5 ET which contained 31 556 925.974 7 ephemeris seconds. The primary unit of TAI, and hence TT, is the SI second, defined to be the duration of 9 192 631 770 periods of the radiation corresponding to the transition between two hyperfine levels of the ground state of the caesium 133 atom. We need not be too concerned by all this since very high accuracy is not the aim of the book. In almost every case we can take ET = TT = UT without noticing the difference. Only when calculating the motion of the Moon, and predicting eclipses (Section 71), will it pay us to take account of the difference between ET/TT and UT. In January 2010 this was 66.07 seconds, UT being behind TT; that is

$$\mathrm{TT} - \mathrm{UT} = \Delta T = 66.07 \text{ seconds}.$$

Figure 16 shows how ΔT has varied since 1620; we can predict that its value in the year 2020 might be around 70 seconds, but only direct observations at that time will confirm this.

Pulsars with very stable rates of spin have been discovered which appear to be even more precise than our best atomic clocks. TAI may well lose its place as the fundamental measure of time during this century, and be replaced by another scale based on the pulsars – GBT (galactic barycentric time) perhaps.

Figure 16. The variation of ΔT since 1620.