

Background.

In order to do date calculations we must translate any given date specified by the three components year, month, and day, into an integer day number. If we choose a base date for day 0 wisely, calculations will be simplified. For this method the base date will be March 1, year 0. Note that this year 0 is not the Gregorian year 0, it is a reference date only. However, calculations involving dates after the establishment of the Gregorian calendar will be correct.

Year calculations.

To four decimal places, the length of a year has been measured as 365.2424 days. Various calendars have tried to accommodate the extra fraction, which is just less than a quarter of a day. In October of 1582, the Gregorian calendar was instituted. It takes the length of a year to be 365.2425 days, which is one part in ten thousand (0.0001) too big, but is accurate enough to last through the next few millenia. This approximation 365.2425 can be written as:

$$365.0000 + 0.2500 - 0.0100 + 0.0025$$

or

$$365 + 1/4 - 1/100 + 1/400$$

Now the number of days in y years is:

$$d = 365y + y/4 - y/100 + y/400 \quad (1)$$

If we now make this a discrete function by considering integer d's and disregarding remainders, we have:

$$d = 365y + \text{int}(y/4) - \text{int}(y/100) + \text{int}(y/400) \quad (2)$$

Every four years the second term $y/4$ will add an extra day, but that will be neutralized every century by the third term $y/100$. However, every four centuries, the leap day will be added after all, as the last term $y/400$ kicks in. This discrete function is the source of the Gregorian leap year rules. Function (2) approximates the real function (1) very closely, and the two are equal for years divisible by 400. In the intervening years of this 400 year cycle, the error maxima are 1.4775 days in year 303 of the cycle, and -0.72 days in year 96 of the cycle. Because of this error, finding the year given a number of days (the inverse function) is not exact, but we can find a very close approximation. Given d days, the year number can be approximated as

$$y = d / 365.2425 \quad (3)$$

This will be correct for nearly all dates, except in some cases where d is within the error maxima from a year boundary. When this happens, we can test for it and adjust the year accordingly.

Month calculations.

Because the lengths of the months are not the same, and especially because the length of February is not fixed, it would seem at first to be impossible to calculate day offsets in a year without looking up tables and checking for leap years. However, we can get around the problem by defining the start of our calendar year with March. Numbering the months in this way has the special advantage that leap days are always added at the end of the year, and do not change the day offsets for the beginnings of the months. Using basic linear regression on the month indexes and the day offsets, we can find functions to map each to the other. Given the month index, we can calculate the day number for the first of the month from the integer part of the function

$$d = f(m) = (306*m + 5)/10:$$

$$d = f(m) = (306*m + 5)/10$$

	m	f(m)	int(f)	
Mar	0	0.5	0	
Apr	1	31.1	31	
May	2	61.7	61	
Jun	3	92.3	92	
Jul.	4	122.9	122	
Aug	5	153.5	153	
Sep	6	184.1	184	
Oct	7	214.7	214	
Nov	8	245.3	245	
Dec	9	275.9	275	
Jan	10	306.5	306	
Feb	11	337.1	337	

Now, the inverse function, given the day index into the year, we can calculate the month index by taking the integer part of the function $m = f(d) = (100*d + 52)/3060$.

The following table shows the values of the function for the first and last days of each month:

$$m = f(d) = (100*d + 52)/3060$$

month	length	index	s=first day	e=last day	f(s)	f(e)
Mar	31	0	0	30	0.017	0.997
Apr	30	1	31	60	1.030	1.978
May	31	2	61	91	2.010	2.991
Jun	30	3	92	121	3.024	3.971
Jul	31	4	122	152	4.004	4.984
Aug	31	5	153	183	5.017	5.997
Sep	30	6	184	213	6.030	6.978
Oct	31	7	214	244	7.010	7.991
Nov	30	8	245	274	8.024	8.971
Dec	31	9	275	305	9.004	9.984
Jan	30	10	306	336	10.017	10.997
Feb	28/29	11	337	365/366	11.030	11.945