Met Office

NAME Technical Specification Document C04

Time structures and routines

David J. Thomson and Vibha Selvaratnam

Documentation issued with	:	NAME Version 8.7
Document last updated for	:	NAME Version 8.0
Last updated on	:	26/01/2018

NAME

Numerical Atmospheric-Dispersion Modelling Environment

Met Office, FitzRoy Road, Exeter, Devon, EX1 3PB, United Kingdom

At present this note mostly only discusses some calculations specific to the Gregorian calendar.

1 Introduction

The time module provides code for storing and processing times. Two derived types are provided for storing times. The first type stores the time in terms of the year, month, day, hour, minute, second, fraction of a second plus time zone information while the second type stores time in terms of fractions of a second since a reference time. Here a fraction of a second means 10^{-7} s. Both derived types use integers rather than reals to avoid issues with rounding and precision. Times of the first and second type are known as "full times" and "short times" respectively.

The types are designed to be able to store the times plus and minus infinity, to distinguish between times of events and time intervals, and to distinguish between clock times (times reflecting the time on a clock on the wall at the time the model is run) and model times (times relating to the time being modelled within the simulation).

The time module can be used with various time frames:

- Relative time frame: times defined relative to some fixed but unknown reference time; this time frame makes no use of months, years or time zones.
- Gregorian time frame: times defined using the (ISO 8601) Gregorian calendar
- 360-day year time frame: times defined using a year of twelve 30-day months; days of the week are defined by taking 1/1/2000 to be a Saturday (as it is in the Gregorian calendar).

Time frames that use years and months ("Gregorian" and "360-day year") are referred to as calendar time frames. Clock times always use the Gregorian calendar. Model times can use any calendar but it is not possible for them to use more than one calendar in a single model run.

2 Range and precision of times and overflow and underflow checks

For short times we use a reference time of 1/1/2000 00:00 UTC and we store the time relative to this time as a 64-bit integer with the same precision as full times in units of 10^{-7} seconds. This allows times up to $\sim$29,000 years either side of the reference time. These parameters could be altered to alter the range or precision of the times. For some purposes we need to calculate the Julian day number which is stored as a 32-bit integer; this imposes a less restrictive range of $\sim \pm$ 5.8 million years. In full times the years are stored in 32-bit integers which imposes an even weaker constraint on the range of times allowed. These ranges are such that there seems no need to trap for possible overflows and so we have not done that in general. However there are some exceptions. For example when calculating the time interval between releases of particles from a source as source strength divided by maximum allowed particle mass, there is a risk that, through an inappropriate choice of values, the time interval may underflow to zero or overflow leading to unexpected and hard to understand behaviour. As a result we have implemented some checks in the conversion of real number time intervals to short times.

3 Converting between (day, month, year) and Julian day number for the Gregorian calendar

The Gregorian calendar [4] was introduced in October 1582. The Gregorian calendar is based on the Julian calendar but involves two modifications. These modifications involve (i) correcting the average year length by letting years divisible by 100 be leap years only if they are also divisible by 400, and (ii) adjusting the (astronomical) time of the start of the year from that in the Julian calendar in 1582 to that in the Julian calendar at the time of the First Council of Nicaea in 325. NAME uses the Gregorian calendar before (as well as after) October 1582, when it is more properly called the "proleptic Gregorian calendar". NAME uses the ISO 8601 [5] proleptic Gregorian calendar which includes a year zero rather than the traditional proleptic Gregorian calendar which, like the Julian calendar, does not have a year zero. Years 0, -1, -2 ... correspond to 1 BC, 2 BC, 3 BC ... in the traditional proleptic Gregorian calendar. For the "first adopters" of the Gregorian calendar, the last day of the Julian calendar was 4th October 1582 and the next day, the first day of the Gregorian calendar, was 15th October 1582. The Julian calendar thus lagged the Gregorian calendar by 10 days in 1582 and loses 3 days per 400 years relative to the Gregorian calendar. Some of this detail is not very relevant to the NAME model but it has been useful in checking various formulae below against [3] and in particular against the on-line calculator given there. Hence we record it here.

The "Julian day number" is a sequential numbering of days, normally defined to be zero on 1st January 4713 BC in the Julian calendar. Here we are mostly not concerned about when day zero is and we use the term with other day zeros too. It's convenient at times to think of the Julian day number as the number of days *before* the day in question, counting days from a particular day (the day with zero Julian day number).

Converting between (day, month, year) and Julian day number is trivial if one loops over the required months and years. However if done repeatedly this can have a significant cost and so more efficient approaches are desirable. This is not so trivial because of the different lengths of different months and the effect of leap years.

In the following we use integer arithmetic and define integer division to round down (rather than towards zero) before applying any explicit minus sign. In places where we need to be explicit, we denote rounding down by $\lfloor\ \rfloor$ and rounding up by $\lceil\ \rceil$. Excel is quite convenient for checking some of the results. Note that INT in Excel rounds down while TRUNC rounds towards zero, and Fortran integer division rounds towards zero.

D, M, Y and J will denote day, month, year and Julian day number.

We discuss two approaches for converting between (D,M,Y) and J.

3.1 Method 1

The first approach is based on [3, 2].

Counting days from the start of year one we have

Number of complete Februarys before day in question:

$$F = Y - 1 + (M + 9)/12$$

Number of leap days before day in question:

$$F/4 - F/100 + F/400$$

Number of non-leap days in current month before day in question:

$$D - 1$$

Number of non-leap days in current year before start of current month (checked in Excel):

$$275M/9 - 30 - 2\lfloor (M + 9)/12 \rfloor$$

Number of non-leap days before start of year:

$$365(Y - 1)$$

Hence number of days before day in question is

$$\begin{aligned}
& 365(Y-1) + 275M/9 - 30 - 2\lfloor(M+9)/12\rfloor + D - 1 + F/4 - F/100 + F/400 \\
= \; & 367(Y-1) + 275M/9 - 30 - 2\lfloor Y - 1 + (M+9)/12\rfloor + D - 1 + F/4 - F/100 + F/400 \\
= \; & 367(Y-1) + 275M/9 - 30 - 2F + F/4 - F/100 + F/400 + D - 1 \\
= \; & 367(Y-1) + 275M/9 - 30 - 2F - (3-F)/4 - F/100 + F/400 + D - 1 \\
= \; & 367(Y-1) + 275M/9 - 30 - (8F + 3 - F)/4 - F/100 + F/400 + D - 1 \\
= \; & 367(Y-1) + 275M/9 - 30 - (7(F+1) - 4)/4 - F/100 + F/400 + D - 1 \\
= \; & 367(Y-1) + 275M/9 - 29 - 7(F+1)/4 - F/100 + F/400 + D - 1.
\end{aligned}$$

Here we have used $F/4 = -(3 - F)/4 \equiv -((3 - F)/4)$ which is easily checked and arises because we round integer division downward. Note in particular that $F/4 \not\equiv -((-F)/4)$ in general because these two expressions round in opposite directions.

Counting days from the start of year zero we have

$$J_{1/1/0000} = 367Y + 275M/9 - 31 - 7(Y + \lfloor(M + 9)/12\rfloor)/4 - F/100 + F/400 + D \quad (1)$$

where we use a subscript to indicate the reference day (day zero or the day where we start counting "days before day of interest"). This enables us to easily convert from (D,M,Y) to J.

We make some checks on this formula:

$367Y + 275M/9 - 31 - 7(Y + \lfloor (M+9)/12 \rfloor)/4 + 1$ gives correct number of days before start of month relative to start of year zero for $0 \leqslant Y \leqslant 4$ (checked in Excel). Adding/subtracting 4 to/from Y changes the value by $\pm(367 \times 4 - 7) = \pm(365 \times 4 + 1)$. Hence, ignoring the issue of possible non-leap years when year is divisible by 100, the formula is correct for all years. Adding $-F/100 + F/400$ corrects for the issue of possible non-leap years when year is divisible by 100.

For 1/3/1900 to 28/3/2100, the formula (1) equals $367Y + 275M/9 - 31 - 7(Y + \lfloor (M+9)/12 \rfloor)/4 - 15 + D$ and agrees with the formula given at [3] apart from an offset.

The offset is not straightforward to check because [3] counts days from the start of 4713 BC defined in the Julian calendar (year $= -4712$ if we, incorrectly, include a year zero in the Julian calendar) and because the formula given in [3] is not valid before 1/3/1800 and is in effect two formulae for 1/3/1800 to 28/2/1900 and 1/3/1900 to 28/3/2100. However, instead of restricting (1) to 1/3/1900 to 28/3/2100, we can add $-F/100 + F/400 + 15$ to the formula in [3] in its post 1/3/1900 form to give a universal formula to use for the check. The formulae from [3] for midday (hour = 12) is then

$$367Y + 275M/9 - 7(Y + \lfloor (M+9)/12 \rfloor)/4 - F/100 + F/400 + D + 1721029$$

(note Julian Day in [3] is defined as zero at 1200 1st Jan 4713 BC and so the midday value is appropriate). The offset between the formulae is thus 1721060. In 1582 the Julian calendar lagged by 10 days which reduced to one day at the First Council of Nicaea and puts the Julian calendar 2 days ahead in year zero. The Julian years $-$4712 to $-$0001 contain $4712 \times 365.25 = 1721058$ days so the offset should indeed be $1721058 + 2 = 1721060$.

It does not seem straightforward to use this type of approach in going from J to (D, M, Y).

3.2 Method 2

The second approach follows [1] and is also described in [3].

We consider a shifted year starting in March and, consistent with this, we define

$$M' = \begin{cases} M - 2 & \text{if } 3 \leqslant M \leqslant 12 \\ M + 10 & \text{if } 1 \leqslant M \leqslant 2 \end{cases} = M - 2 + 12JF$$

and

$$Y' = \begin{cases} Y & \text{if } 3 \leqslant M \leqslant 12 \\ Y - 1 & \text{if } 1 \leqslant M \leqslant 2 \end{cases} = Y - JF$$

where

$$JF = \begin{cases} 0 & \text{if } 3 \leqslant M \leqslant 12 \\ 1 & \text{if } 1 \leqslant M \leqslant 2 \end{cases} = (14 - M)/12.$$

Note that the definition of Y' is chosen so that Y and Y' agree for 10 months per year. However the value of Y' is related to whether the previous (and not the current) shifted year has a leap day in the same way that Y is related to whether a normal year has a leap day. This difference is actually convenient because in considering a given shifted year Y', we are mostly interested in whether the previous year was a leap year. This is because the leap day (if any) occurs at the end of the shifted year.

We define dy as "day of year" using the convention that the year starts in March and starts at day zero, that is, counting days from the start of the shifted year, dy is the number of days prior to the day in question. Similarly we define dm as "day of month" starting at zero and note that $dy - dm$ is the number of days of the year before the current month. dy, M' and $dy - dm$ are related by

$$M' = 80(dy + 31)/2447, \quad dy - dm = 2447M'/80 - 30 = 367M'/12 - 30$$

(checked in Excel). Note $M' = 12(dy + 31)/367$ doesn't work because, for $dy = 336$ and $M' = 11$ (31st January), this relation is exact but we need it to round strictly downwards. Other linear expressions (combined with rounding down to integer values) are of course possible here. Such methods rely on the possibility of fitting a straight line on a M' versus dy plot that goes through the last day of every month (except February) as illustrated in figure 1. Here we use the $2447/80$ factor when going in either direction.

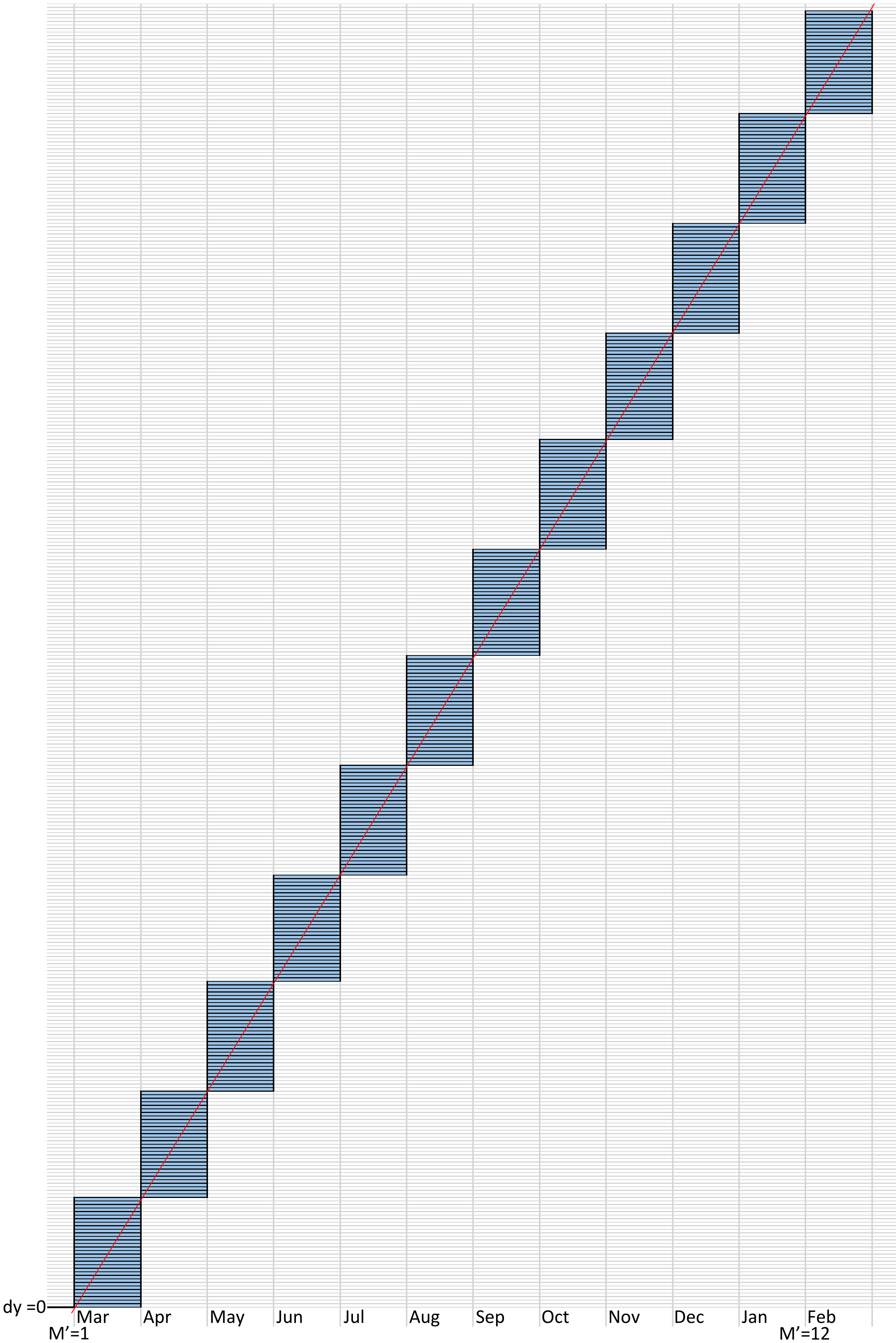

Figure 1: Illustration of linear relationship between M' and dy

Counting days from the start of year zero we have

$$\begin{aligned} J_{1/1/0000} &= 60 + 365Y' + Y'/4 - Y'/100 + Y'/400 + dy - dm + dm \\ &= 60 + 365Y' + Y'/4 - Y'/100 + Y'/400 + 2447M'/80 - 30 + D - 1 \\ &= 29 + 365Y' + Y'/4 - Y'/100 + Y'/400 + 2447M'/80 + D. \end{aligned} \quad (2)$$

For the inverse we have the following algorithm (where "days" means days before day of interest counting days starting at some reference day; "years", "4-year periods", "100-year periods", "400-year periods" refer to periods occurring periodically starting on 1/3/0000; and '$=$' is assignment)

J	$=$	$J_{1/1/0000} - 60$	days from 1/3/0000
QC	$=$	$J/146097$	(complete) 400-year periods from 1/3/0000
J	$=$	$J - 146097QC$	days from start of current 400-year period
C	$=$	$\min(J/36524, 3)$	100-year periods from start of current 400-year period
J	$=$	$J - 36524C$	days from start of current 100-year period
Q	$=$	$J/1461$	4-year periods from start of current 100-year period
J	$=$	$J - 1461Q$	days from start of current 4-year period
Y	$=$	$\min(J/365, 3)$	years from start of current 4-year period
J	$=$	$J - 365Y$	days from start of current year ($= dy$)
Y	$=$	$400QC + 100C + 4Q + Y$	($= Y'$)
J	$=$	$J + 31$	($= dy + 31$)
M	$=$	$80J/2447$	($= M'$)
JF	$=$	$M/11$	($= 1$ in Jan/Feb and 0 otherwise)
D	$=$	$J - 2447M/80$	($= D$)
M	$=$	$M + 2 - 12JF$	($= M$)
Y	$=$	$Y + JF$	($= Y$).

We note the year calculations can also be done a bit differently. We adjust J to start counting at 1/3/0300 using $J_{1/3/0300} = J_{1/1/0000} - 60 - 3 \times 36524$. We will regard 4-, 100- and 400-year periods as referring to periods occurring periodically starting on 1/3/0300. We note that, with this choice, the "long" 100-year periods occur at the start of the 400-year periods, the "short" 4-year periods occur at the end of the 100-year periods, and the leap years occur at the end of the 4-year periods. We calculate the number C of complete 100-year periods as follows. Let $N = 36524$ be the number of days in a short 100-year period. For given C we have $CN + (C+3)/4 \leqslant J \leqslant (C+1)N + (C+4)/4 - 1$ and it's easily checked that $4J/(4N+1)$ gives the correct value of C (we check that $CN + (C+3)/4 \leqslant J \leqslant (C+1)N + (C+4)/4 - 1$ is true for this value of C). The number of days in these 100-year periods is $((4N+1)C+3)/4$. We put $J = J - ((4N+1)C+3)/4$ to give the number of days left.

To calculate the number Y of complete years left is more complicated. If there were leap years at the start of each 4-year period we could proceed in a similar way to the calculation of C above. Let $N = 365$ now be the number of days in a non-leap year. For given Y we would have $YN + (Y+3)/4 \leqslant J \leqslant (Y+1)N + (Y+4)/4 - 1$ and $4J/(4N+1)$ would give the correct value of Y. However the leap years are not at the start of each 4-year period, and we can't shift the time origin to make it so because there are some 4-year periods with no leap years. If instead we add 1 to J and use $Y = 4(J+1)/(4N+1)$ then this corrects for the leap years being at the end instead of the start of the 4 year periods, with the exception of the last day of each 4-year period (other than the last 4-year period of a short 100-year period) which will now appear to be in the next 4-year period. However these are precisely the cases for which there is exact division in $4(J+1)/(4N+1)$, so making a very small reduction will correct these cases while not altering the other cases. A possible solution is to calculate Y as $4000(J+1)/(1000(4N+1)+1)$ (checked in Excel). Alternatively one could round up and subtract 1 which can be written as

$$\begin{aligned} \lceil 4(J+1)/(4N+1) \rceil - 1 &= \lceil (4(J+1) - (4N+1))/(4N+1) \rceil \\ &= -\lfloor ((4N+1) - 4(J+1))/(4N+1) \rfloor \\ &= 99 - \lfloor 99 + ((4N+1) - 4(J+1))/(4N+1) \rfloor \\ &= 99 - \lfloor (100(4N+1) - 4(J+1))/(4N+1) \rfloor \\ &= 99 - (100(4N+1) - 4(J+1))/(4N+1) \\ &= 99 - (146096 - 4J)/1461 \end{aligned}$$

where we added and subtracted 99 so that the numerator in the fraction is non-negative and the expression can also be interpreted with rounding towards zero as in Fortran integer division. The number of days in these years is $(4N+1)Y/4$. We put $J = J - (4N+1)Y/4$ to give the number of days left.

This leads to the following algorithm:

$$\begin{array}{lcll} J & = & J_{1/1/0000} - 109632 & \text{days from 1/3/0300} \\ C & = & 4J/146097 & \text{100-year periods from 1/3/0300} \\ J & = & J - (146097C + 3)/4 & \text{days from start of current 100-year period} \\ Y & = & 4000(J + 1)/1461001 & \text{years from start of current 100-year period} \\ J & = & J - 1461Y/4 & \text{days from start of current year } (= dy) \\ Y & = & 100C + Y + 300 & (= Y') \\ J & = & J + 31 & (= dy + 31) \\ M & = & 80J/2447 & (= M') \\ JF & = & M/11 & (= 1 \text{ in Jan/Feb and 0 otherwise}) \\ D & = & J - 2447M/80 & (= D) \\ M & = & M + 2 - 12JF & (= M) \\ Y & = & Y + JF & (= Y). \end{array} \tag{3}$$

In NAME we use (2) and (3) with some pre and post processing to ensure that any quantities to be rounded after integer division are non-negative and so standard Fortran rounding of integer division (which is towards zero rather than downwards) can be used. For converting (D, M, Y) to J we use

$$\begin{array}{ll} \text{Preprocess:} & QC = Y/400 - 1; Y = Y - 400QC \\ \text{Postprocess:} & J = J + 146097QC \end{array}$$

and for converting J to (D, M, Y) we use

$$\begin{array}{ll} \text{After } J = J_{1/1/0000} - 109632\text{:} & QC = J/146097 - 1; J = J - 146097QC \\ \text{Postprocess:} & Y = Y + 400QC \end{array}$$

with integer division here being rounding towards zero.

We note that 4000×36525, the largest value possible for $4000(J + 1)$ in (3), is within the range which is storable as a 32-bit integer and so there is no risk of overflow in converting from J to (D,M,Y) using 32-bit integers provided the values of $J_{1/1/0000}$ and $J_{1/1/0000} - 109632$ are in range. Similarly, there is no risk of overflow in converting from (D,M,Y) to J provided that the values of $J_{1/1/0000}$ and $146097QC$ (which are similar in size for extreme times) are in range. These restrictions constitute much weaker constraints on the range of times that can be treated than that imposed by the storage model for short times.

References

[1] Fliegel, H. F. and van Flandern, T. C., 1968, Communications of the ACM, **11**, 657. 3

[2] U.S. Naval Observatory, 1990, Almanac for Computers. 2

[3] U.S. Naval Observatory, 2015, Converting Between Julian Dates and Gregorian Calendar Dates, http://aa.usno.navy.mil/faq/docs/JD_Formula.php. 2, 3

[4] Wikipedia, 2017, Gregorian calendar, https://en.wikipedia.org/wiki/Gregorian_calendar. 1

[5] Wikipedia, 2017, ISO 8601, https://en.wikipedia.org/wiki/ISO_8601. 1