A study on the calculation of the date of Easter

Taidgh Murray

February 2019

Acknowledgements

My utmost gratitude to Immo Warntjes and Jacopo Bisagni, whose advice and resources were instrumental in the formative stages of the project. They guided me and my supervisor through and unfamiliar discipline, and referred us to the precise resources necessary to investigate this topic. This project would not have been nearly as successful without their input. Kevin Jennings, who provided exemplary guidance and support, and facilitated an unconventional mathematical project, and Tadgh Wagstaff, who provided a humanities friendly front end for my algorithm, alongside providing a professional proof reading/editing of the project.

1 Introduction

Most all annual celebrations in our calendar have religious or spiritual roots. Christmas, possibly stemming from a pagan winter solstice celebration, is now widely celebrated as the birth of Jesus Christ. Saint Patrick's Day, the 17th of March, celebrates the life (and death) of our most well known saint. Casting our eyes towards those of the Islam faith will see the holidays of Eid Al-Adha and Eid Al-Fitr. Seemingly secular holidays, such as Halloween, are irrevocably tied to spiritual origins (such as All Hallow's Eve and All Saint's Day).

Most celebrations are held on a constant date each year, with universal acceptance of when they fall[1]. However, one such celebration remains mired in controversy and confusion on our calendars.

Easter is unique in that its date changes from year to year on modern calendars, but can be defined as 'The first Sunday following the Paschal Full Moon on or after the March Equinox (the 21st of March)'. Currently, there are 35 possible dates under which Easter can fall, the earliest being the 22nd of March, and the latest being the 25th of April. Easter is very closely tied to the Lunations

[1]If we are to consider a western, Gregorian Calendar, then this doesn't hold true for the holidays of Eid Al-Adha or Eid Al-Fitr. However, their dates are constant under the hijri Calendar. For more information on Islamic Holidays, please consult http://www.staff.science.uu.nl/gent0113/islam/islam_tabcal.htm

of the Moon, so much so that it supersedes the Solar Gregorian Calendar we use.

The reasoning behind the decision on the date of Christ's Resurrection intrigues me. From an outside perspective, it can seem almost arbitrary. My interest in the topic, my desire to understand how Easter relates to the rotation of the Earth, the Moon and the Sun, and the history involved in determining Easter were my impetus for this project. This project enabled me to marry my passions in the field of Mathematics, and Computer Science and Information Technology. The programming portion of my project also falls under the 'Digital Humanities' umbrella, which is a chapter of the Technology Discipline I feel doesn't get nearly enough attention from those invested in the 'harder' subjects (Computer Science, Software Development, Information Technology, etc.). My intention for the project is to gain a more comprehensive understanding of a 'general' reckoning that goes into Easter, and gain a deeper knowledge of the Easter Tables developed by the Ecclesiastical scholars.

2 History

It is of vital importance before we begin any computation, that I elucidate some terms and definitions. This project involves a lot of niche terminology, and much of the related documentation has been antiquated for quite some time. It is important that we establish a baseline understanding of the terms and topics in common use throughout the study of Computus.

2.1 The modern Calendar

Our first barrier to gaining a more comprehensive understanding of Easter is the definition of a calenders we use. Our modern calendar is the Gregorian Calendar. It was formalised by, and named for Pope Gregory XIII. A year in the Gregorian Calendar consists of 365 days, unless the year is divisible by 4, in which case an extra year is added (*i.e.* a leap year). If the year is divisible by 100, then that year is not a leap year. If that year is also divisible by 400, then it is a leap year. A simple Python implementation calculating if a given year is a leap year can be found below.

```python
def leap_year(y):
    if y%4==0:
        if y%100 == 0:
            if y%400 == 0:
                return True
            return False
        return True
```

This results in an average year being $365.25 - \frac{3}{400} = 365.2425$ days. This calendar was proposed to the church by Gregory XIII in 1582 [1]. The calendar was proposed to rectify a drift of (at the time) 10 days between the calendar in employ (the Julian Calendar), and the Earth's 'actual' date, as perceived under its solar year. The drift between the two calendars can be described mathematically as $\lfloor \frac{year}{100} \rfloor - \lfloor \frac{year}{400} \rfloor - 2$ The calendar was brought into effect officially following the 4^{th} of October. Confusingly, to rectify the drift that occurred, the day following October 4^{th} in the calendar now read as October 15^{th}.

The Julian Calendar was proposed by Julius Caesar in 46 B.C., and was first implemented on the 1^{st} of January the following year. This calendar is much simpler to understand. Each year has 365 days, except if the year is evenly divisible by 4, in which case it a leap year, and has 366 days. This means that the average Julian Calendar year has 365.25 days. Our current approximation of the Solar Year is 365.24217 days [2]. This resulted in a shift of about 3 days every 4 centuries.

The Julian Calendar was introduced to bring parity to timekeeping, as what preceded it were the so-called 'Roman Republican Calendar' and 'Legendary 10-Month Roman Calendar'. The groundwork of the Julian Calendar can be seen in the Republican Calendar, which was Greek in nature. It followed the idea that a lunar cycle was 29.5 days, and solar year consisted of 12.5 synodic months. This resulted in 12 months familiar to those of the Gregorian Calendar, and a 13^{th} intercalary month, consisting of 23 days, intermittently placed into the Year when necessary to allow the Solar and Lunar Calendars to sync up [3].

The Romans were fond of the decimal system, so it's understandable that an early calendar implemented by them would adhere to this system (A style of thinking believed to be derived from Romulus [4]) , but the consequence of this was a 51 day, unorganised 'Winter' month that bookended the calendar year. The remaining 304 days were doled out amongst the 10 months in blocks of either 30 or 31 days [4]. Moving earlier in time than the Legendary Calendar proves challenging, much of what we know has been lost to the annals of time. We do understand that the original Roman Calendar was Lunar in nature, with a new month beginning at the first sign of a new crescent moon [3].

3 Glossary

I have also opted to include a Glossary of common terms and topics that occur throughout this project. These terms and topics are essential to fully understand the discourse surrounding Computus.

Epact: The 'age' of the Moon in relation to the lunar cycle on the 22^{nd} of March. In the Gregorian Calendar, it is the age of the Moon on the 1^{st} of January. This value is between 1 and 30, these values indicating a New Moon, 14 indicating a Full Moon. It is worth noting that the Lunar 14 value is an Ecclesiastical construct, not a astronomical 'Full Moon'.

Metonic Cycle A period taken to be 19 solar years which is seen as nearly a common multiple of the solar and lunar year. This period has 235 synodic months, translated to 6939 days and 16.5 hours. 19 solar years have 6939 days and 14.5 hours. This two hour difference results in a drift of one day every 219 years.

Easter Table Easter Tables are the most common way of reckoning the date of Easter for a given year. Most commonly consists of a year, and Epact, and a date for Easter, though different tables may contain extra data, such as the age of the moon, or the beginning of lent.

Tropical Year: Also known as a Solar Year. The time it takes for the Earth to perform one revolution around the Sun. Currently approximated at 365.242189[2] days. In the Gregorian Calendar, this time is further approximated as 365.2425 days.

Intercalary: Refers to the act of inserting a leap day, week or month into a calendar to keep parity between the calendar and the Solar Calendar.

Synodic Month: The moon's orbit with respect to the line joining the Sun and the Earth. The average duration of this period is 29.530588853 days.

Saltus Lunae: Also known as the 'Leap of the Moon'. Additional day added to a calendar year to keep a lunar and solar year in alignment.

Paschal: Refers to Easter related terms. Derived from the 'Peasch', the Hebrew word for Passover.

[2]https://www.timeanddate.com/astronomy/tropicalyearlength.html

4 A Modern Understanding of Computus

4.1 Gauss' Easter Algorithm

Trying to understand and model an algorithm behind Computus is no easy feat. Many scholars of the past have developed diverse solutions to conclude when should Easter fall for a given year. That Easter falls on a different date depending on whether you practice Roman-Catholicism or Orthodox Christianity only complicates any formula. Despite this complication, one scholar was successful in providing a succinct algorithm to determine when Easter should fall for a given year.

Carl Fredrich Gauss developed a relatively simple formula for determining the date of Easter on a given year. Deciphering his algorithm did present other scholars some challenge, Gauss himself admitted that the scope of the formula was too great to describe in the original publication in 1800[3]. This formula returns the day-of-March format of the date, which treats both March and April as a conglomerated 61 day month.

a	$year \bmod 19$
b	$year \bmod 4$
c	$year \bmod 7$
k	$\left\lfloor \frac{year}{100} \right\rfloor$
p	$\left\lfloor \frac{13+8k}{25} \right\rfloor$
q	$\left\lfloor \frac{k}{4} \right\rfloor$
M	$(15 + k - q - p) \bmod 30$
N	$(4 + k - q) \bmod 7$
d	$(19a + M) \bmod 30$
e	$(2b + 4c + 6d + N) \bmod 7$

To get the date of Easter, the value will either be $22 + d + e$ if Easter falls in March, or $d + e - 9$ if Easter falls in April. If the March formula returns a value greater than 31, then the April formula is used instead. Aditionally:
If $d = 29$ and $e = 6$, April 26^{th} becomes April 19^{th}.
If $d = 28$,$e = 6$, and $a > 10$, April 25^{th} becomes April 18^{th}.
This formula can be modified to adhere to the Julian Calendar, by taking M and N to be constants 15 and 6 respectively. This also means that p, k, and q are unnecessary.

[3]The document is in German. An excerpt from the text reads: "Die Analyse, vermittels welcher obige Formel gefunden wurde [...], lässt sich daher freilich in ihrer ganzen Einfachheit hier nicht darstellen [...]", which translates to It is not possible to present here the entire analysis that led to the above formula.

Above is a formatted version of the algorithm Gauss had outlined in his publication in 1800 [5]. Gauss released an addendum in 1816, changing the p value from $\left\lfloor \frac{k}{3} \right\rfloor$ to the value seen in the above table. In Gauss' original formula, he incorrectly assumed that the Moon Equation be applied evenly every 300 years. The addendum was to fix this issue, but the old formula accurately calculated the correct date of Easter up to the year 4200.

The first 4 values in the table, are the simplest to outline:
a is the position of a given year in the Lunar Cycle, from 0-18.
b is the value of the leap years, from 0-3.
c is the cycle of the weekday, from 0-6.
k calculates the century the year is in.

Understanding the p variable, the so called 'Moon Equation', is less trivial. We know that this serves the same purpose as $\left\lfloor \frac{k}{3} \right\rfloor$, calculating the cycle of the Lunar 14. The formula originally assumed that this cycle resulted in the Lunar 14 being moved back a day every 300 years 8 times, resulting in a period of 2400 years. However, after 7 shifts of the Lunar 14, a break of 400 years is taken before the final shift, resulting in a time space of 2500 years/8 shifts = 312.5 years per shift [6]. This cycle begins in the year 1800, up to 4300. Gauss' original algorithm, therefore, is accurate up to the year 4200.

Taking the cycle to begin in the year 1800, to determine the cycle, a simple division can be taken in the form $\frac{k-18}{25} = A$ (with B remainder if $k < 43$). This can be rewritten as $\frac{k-18-B}{25} = A$. We say that A indicates a complete cycle, and B can be seen as the number of centuries in a cycle. To determine the number of shifts that have occurred within a given 2500 cycle of a year, we can use the formula $p = 6 + 8A + (R)$. The R value indicates the the number of shifts that can occur in the cycle B.

To fully determine R, we must calculate a uniformly distributed set of numbers between 0 and n. In this case, we set our n to be 25, and use the equation $R := \frac{(xB+y)}{n}$. We note x will be 8, as there are 8 possible results we want. We take n to be 25 for the 2500 year century period. Our y value is equal to 7, as we need to repeat the values three times for each 300 year cycle, but repeat it 4 times once, to account for the 400 year break.

We can now determine R as $\frac{(8B+7)}{25}$. Plugging that back into our p formula; $p = 6+8A+\frac{(8B+7)}{25}$. We determined A to be $\frac{k-18}{25}$. Finally, our p can be written as:
$p = \frac{150}{25} + 8\frac{k-18-B}{25} + \frac{(8B+7)}{25}$, which simplifies to $\frac{13+8k}{25}$.

q is for calculating if the century is divisible by 4, fulfilling a clause outlined by the Gregorian Calendar.

M indicates the distance of a given year from the first year of the lunar cycle. When calculating the Old Style Easter date, this value was a constant of 15. This was because in the first year of the cycle, the lunar 14 always fell on the 5^{th} of April ($= 36^{th}$ of March). In the New Style, the correction of both the Lunar and Solar calendar must be taken into account, these corrections have been outlined above, the M value specifically being to correct the difference in the Lunar Year and the Calendar. This is defined as $15 + k - q - p$

Similarly, the N is to correct for the difference arising between the calendar and the Solar Year. This value was a constant of 6 in the Julian Calendar, but the correction for the difference between the Solar Year can be defined as $k - q - 2$. This is in mod 7 to account for the rules surrounding leap days.

The value d indicates the distance between March 21st, and the Lunar 14 (*i.e.* the Full Moon). The 19 value has very little to do with the 19 year Lunar Cycle. This 19 actually refers to the shift of 11 days that the Lunar 14 value must take every year to rectify the discrepancy between the Lunar and Solar year. Multiplying the value by 19, and then taking that value mod 30 is equivalent to continuously adding the value by 19, which is equivalent to taking 11 away from the value.

Finally, we get to our value e, the distance from the Lunar 14, to the following Sunday. The value is taken mod 7, as the earliest possible date of Easter is if the Full Moon were to occur on the 21^{st}, and that day being a Saturday. So the gap can be anywhere from 0 to 6 days. The expression $2b + 4c$ is for the consideration that the weekdays move forward by one day every year (except for leap days, where they move forward by two days). For example, if March 17^{th} is a Thursday one year, the following year it will be either a Friday or a Saturday. Because we are working in mod 7, we can simply add 6, which is the same as subtracting 1. A similar Ethos is applied to the $6d$ value, the Full Moon occurs later depending on the year, and must be compensated by moving it back that same number of days. The N value is used to give a starting point for the calculations for a given Century.

So, all that gives us our value e. Starting on the 22^{nd} of March, we add our d and e values $(22 + d + e)$. If this value does not exceed 31, then we have our date of Easter. If it does, then we instead use the formula $d + e - 9$ to give the date in April. Interestingly, under the Gregorian Calendar, Easter is actually periodic, and repeats every 5,700,000 years. Curiously, within this period, the most commonly occurring date of Easter is the 19^{th} of April. This is a consequence of the lunar cycle producing an Easter Sunday date of the 26^{th} of April, a date that the Catholic Church reckons as too late for Easter. The date is therefore shifted back 7 days to the 19^{th}.

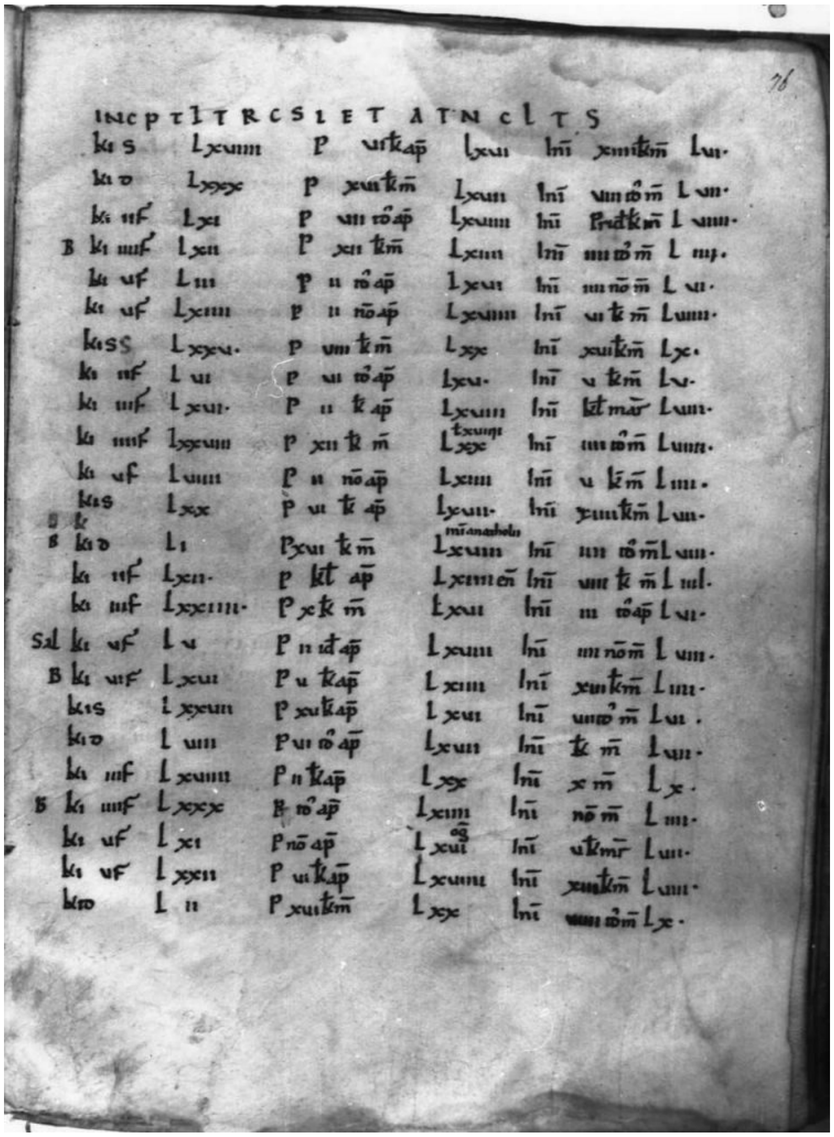
5 The Easter Tables

5.1 The Easter Table of Dinoysius Exiguus

5.1.1 Intro

In understanding the mathematical computation of Easter, we must first look to the scholars of the past. One such important figure is the Monk, Dionysius Exiguus (also named ‘Dennis the Small’, or ‘Dennis the Humble’) [7], a 6^{th} Century Monk. Dionysius was influential in developing the Christian Calendar we know today, including the the ‘Anno Domini (AD)’ date labeling system [8], which sees common use today (for clarification, the Anno Domini dating system is identical to the ‘Common Era (CE)’ system). Much of Dionysius’ work serves as a bedrock for the scholarly works of Bede in the mathematical approximation of Easter.

Below you can find a snippet excerpt from Dionysius’ Easter Table. I’ve included four dates to demonstrate how the calculations are achieved, I feel that these are all interesting dates which show how some of the quirks in calculation arises. It is worth noting that any terms relating to the term ‘Pascha’ can be interchanged with ‘Easter’. I’ve used a letter shorthand in the table. When discussing the Lunar Cycle, it should be noted that 14^{th} value indicates a full moon.



76

INCP T I T R C S L E T A T N C L T S

	ki s	L xviiii	P vi kl ap	L xvi	lnī	xiiii kl m	L vi·
	ki d	L xxx	P xvi kl m	L xvii	lnī	viii id m	L vii·
	ki iif	L xi	P vii id ap	L xviiii	lnī	Prid kl m	L viii·
B	ki iiif	L xii	P xii kl m	L xiii	lnī	iiii id m	L iiii·
	ki vf	L iii	P ii id ap	L xvi	lnī	iii nō m	L vi·
	ki vf	L xiiii	P ii nō ap	L xviiii	lnī	vi kl m	L viiii·
	ki ss	L xxv·	P viii kl m	L xx	lnī	xvi kl m	L x·
	ki iif	L vi	P vi id ap	L xv·	lnī	v kl m	L v·
	ki iiif	L xvii·	P ii kl ap	L xviiii	lnī	kl mar	L viii·
	ki iiif	L xxviii	P xii kl m	L $^{\text{txuiiii}}$ xx	lnī	iii id m	L viiii·
	ki vf	L viiii	P ii nō ap	L xiii	lnī	v kl m	L iiii·
	ki s	L xx	P vi kl ap	L xvii·	lnī	xiiii kl m	L vii·
B k							
B	ki d	L i	P xvi kl m	L $^{\text{mī anasholu}}$ xviii	lnī	iiii id m	L viii·
	ki iif	L xii·	P kl ap	L xiii mē	lnī	viii kl m	L iiii·
	ki iiif	L xxiii·	P x kl m	L xvi	lnī	iii id ap	L vi·
Sal	ki vf	L v	P ii id ap	L xviii	lnī	iiii nō m	L viii·
B	ki vif	L xvi	P v kl ap	L xiii	lnī	xvi kl m	L iiii·
	ki s	L xxvii	P xvi kl ap	L xvi	lnī	viii id m	L vi·
	ki d	L viii	P vi id ap	L xvii	lnī	kl m	L vii·
	ki iiif	L xviiii	P ii kl ap	L xx	lnī	x m	L x·
B	ki iiif	L xxx	P id ap	L xiii	lnī	nō m	L iii·
	ki vf	L xi	P nō ap	L $^{\text{ag}}$ xvi	lnī	vi kl m	L vii·
	ki vf	L xxii	P vi kl ap	L xviiii	lnī	xviii kl m	L viii·
	ki d	L ii	P xvi kl m	L xx	lnī	viiii id m	L x·

Figure 1: Dionysius' Easter Table.

5.1.2 The Algorithm

Table 1: 4 chosen years on Dionysius' Easter Table.

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
574	7	14	7	Saturday	2	March	22	5	Thursday	25	March	17
...												
579	12	9	6	Friday	7	March	27	2	Monday	2	April	20
...												
585	3	15	7	Saturday	13	March	21	4	Wednesday	25	March	18
...												
596	14	17	7	Saturday	5	April	18	4	Wednesday	22	April	18
...												

Here are what each letter corresponds to:
A: The Year
B: Indiction Number
C: Epact
D: Concurrent
E: Concurrent Day
F: Lunar Cycle
G: Easter Full Moon
H: Lunar 14
I: Paschal Moon
J: Paschal Moon Day
K: Easter Date
L: Easter Month
M: Lunar Age

Indiction: The Indiction refers to a 15-year cycle introduced by the Romans, largely for taxation and other fiscal related purposes. In the past, years in the cycle would be referred to as "year y of the Indiction", but as time went on, years were simply referred to as the y^{th} Indiction. It is commonly believed that the Indiction cycle began on September 1^{st}, 312 C.E. The calculation for the Indiction of a given year, y, is Indiction $\equiv (y+3) \pmod{15}$.

Epact: The Epact, loosely translated as 'Added Days', can be described as the age of the moon (in days) on March 22^{nd}(The March Equinox). Relating this value to the Solar and Lunar Calendar poses a challenge, as there is a drift between the two Calendars of about 11 days. This value is periodic, resetting itself every 19 years. In the Gregorian Calendar, this value corresponds to the moon's age on January 1^{st}. The Epact on a given year is Epact $= y' * 11 \pmod{30}$, where $y' = y \pmod{19}$

Concurrent: A number, between 1 and 7, that corresponds to the weekday of March 24th, counted from Sunday. Typically, this number is incremented by

1 each year, except in the case of a leap year, in which the value is incremented by 2. The Concurrent day is just the corresponding day of the week for the Concurrent.

Lunar Cycle: This number represents the index of the given year in relation to the 19-year periodic lunar cycle. This value simply increments by one every year, from 1-19.

Easter Full Moon: This is the month that the full moon which will determine Easter falls under. This information is derived from the Epact. If the Epact is quite young, or quite old, then it stands to reason that the Easter Full Moon will occur sometime in March. Otherwise, the lunar cycle will 'spill' over to April, and the Easter Full Moon will occur sometime then instead. It's worth noting that calculating the Easter Full Moon will NOT give us the Month in which Easter will occur. This only really applies to March, as there are cases where a full moon can occur in March, but Easter ends up being in April. So, to summarise:

```
if epact<5 or epact>15:
    Easter Full Moon = April
else:
    Easter Full Moon = March
```

Lunar 14: The Lunar 14 is the next date after the Epact in which the moon is full. This can occur either in March or in April. In layman's terms, to get the Lunar 14 value, you keep adding days to March 22nd, and increment the Epact by 1, until the Epact value is 14. The corresponding date is the Lunar 14 value. The Lunar 14 is not allowed to share its date with the Epact, *i.e.* the Epact can never also be the Sunday before Easter. Below is a pseudo-code explanation of the Lunar 14 calculation.

```
if Easter Full Moon in April:
    Lunar14 = (35-Epact)mod(30)
else:
    Lunar14 = 35-Epact
```

The 35 values here are likely referring to the 35 possible dates of Easter. Interestingly, if the Epact is 15 (*i.e.*, the Full Moon occurs on March 21st), then instead of running through another lunar cycle to get to the next full moon occurring on April 20th, the lunar 14 is taken to be the previous day. This can be seen in the table above. If this March 21st is a Saturday, then the following day is the earliest possible date for Easter.

Paschal Moon: This is the corresponding weekday number (Counting from Sunday) on which the Lunar 14 occurs. In other words, this is the full moon before Easter. This value uses the Easter Full Moon, the Concurrent and the Lunar 14 value to calculate the Paschal Moon weekday number.

```
if Easter Full Moon in March:
    PaschalMoon = (Lunar14 + Concurrent + 4)mod(7)
else:
    PaschalMoon = (Lunar14 + Concurrent)mod(7)
```

Paschal Moon Day: The corresponding day of the week, taken from the Paschal Moon value.

Easter Date: The date on which Easter falls for a given year. This is calculated by incrementing the Lunar 14 date until the next Sunday occurs. Easter is not allowed to share its date with the Lunar 14, thus the Lunar 14 must be incremented by the appropriate value between 1 and 7.

Easter Month: The Month that the Easter date falls under.

Lunar Age: The age of the Moon on Easter.

5.2 The Latercus Easter Table

5.2.1 Intro

The Easter Table of the Latercus asked for a different mindset to fully understand its intricacies. Documentation on the Latercus Easter Table was scarce until the 1980s, only secondary sources discussing it existing. It wasn't until 1984, that two scholars, Daniel McCarthy and Dáibhí ó Cróinín provided an deeper understanding of the so called 'Lost Irish 84-Year Easter Table' [9]. Their scholarly work served as a vital basis for reverse-engineered algorithm.

Below you can find a excerpt of the Latercus Easter Table, taken directly from the Oxford Companion to the Year. This recreation was taken from the discovery of the Latercus table in a manuscript in Padua in the 1980s, with some corrections outlined fully by McCarthy in his 'Easter Principles' paper. The original table only outlined a single 84-year cycle from 438 to 521. In this recreation, 4 instances of the AD year has been added for each index in the cycle, during which this table was still relevant [8]. A short explanation of each column can be found below the table. Daniel McCarthy and Dáibhí ó Cróinín's work on Latercus was instrumental in outlining any issues or corrections that would not have made themselves apparent at first glance in the table. I will go clarify this as these issues arise in the description of the algorithm below.

Table 2: A sample of 4 rows from the Latercus Easter Table [8, p. 873].

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**
1	354, 438, 522, 606, 690	Sat. (7)	19	**27 Mar.**	16	16 Feb.	6
...							
7	360, 444, 528, 612, 696	Sat. (7)	25	**23 Apr.**	20	15 Mar.	10
...							
28	381, 465, 549, 633, 717	Fri. (6)	17	**28 Mar.**	15	17 Feb.	5
...							
75	428, 512, 596, 680, 764	Sun. (1)	28	**15 Apr.**	14	7 Mar.	5

Here is a short explanation of each column:

A - Cycle: This is the value of the year in the 84 year cycle.

B - Year: The Anno Domini formatted year(s).

C - 1 Jan.: This is the Feria of the 1^{st} of January for the given year. In brackets, the numerical value is given.

D - Epact: This is the Epact on January 1^{st}. Conveniently, this can also be taken as the Epact on March 1^{st}, which we will use in the algorithm.

E - Easter: This is the Date of Easter, given in Day and Month. The earliest date this value can occur on is March 25^{th}, the latest date being 23^{rd} of April.

F - Lune: This is the age of the moon on the date of Easter. This value also occurs later in the table **(H)**, and dictates the age of the moon on the Initium.

G - Initium: The Initium can be considered the beginning of Lent. In the Insular Church, this was the Wednesday 40 days before Easter, counting inclusively. The Initium date would be one week after our modern Ash Wednesday.

5.2.2 The Algorithm

In order to calculate Easter under the Latercus Easter Table, some mathematical legwork must be done first. We don't have insight into how the monks of the day reckoned Easter, so we recreate their algorithm in a plausible way. The first principle of this algorithm is that it should be simple to use; thus, the only piece of information asked from the user is the Year. Each other column should be generated from year, once the starting values of the Epact and the Feria are initialised. The first values I opted to calculate are the Cycle Number, the Feria on January 1^{st}, and the Epact.

Cycle Number: The Insular Church operates on an 84-year cycle. We can use modular division to find where a given year lies within this cycle. Let's take year 438, the first year in the cycle as tabulated, as an example. Calculating 438 mod 84 gives us 18. Since we want the cycle number of 438 to be one, we take 17 away from the year. Generally, we can subtract 17 from the year, and take the remainder mod 84 to give its cycle number. In Python format, the cycle number is calculated like so, refereed to as 'index' in the Python code:

```
def cycle(year):
    index = (year-17)%84
    if(index == 0):
        index = 84
    return index
```

Epact and Feria (January 1^{st}): These two values are calculated independently of the Year. It is unreasonable to calculate these values dynamically, based on the year alone, as they depend on the both the year, and their previous values. An initial value for both the Feria and the Epact must therefore be specified, and all other values for both variables can then be generated.

Due to the 11 day disparity between the Lunar and Solar calendars, the Epact must be incremented by 11 days, annually to maintain accuracy. This rule is not constant. As an aside, I must clarify that the Insular Church did indeed operate on an 84 year cycle, but, it must be specified that within this cycle exists six 14-year cycles. Following the end of a given 14 year cycle, a Saltus Lunae occurs, and the Epact must be incremented by 12, instead of 11. The first Epact value in the cycle is 19, as tabulated. However, it has been concluded, by McCarthy and ó Cróinín [9], the value of the Epact used in calculations was one greater than the tabulated Epact value. My algorithm has taken this into account, and the initial Epact has been incremented to 20 rather than 19.

The rules for the Feria are similar to the rules for the concurrent outlined in the study of Dionysius. It is incremented by one every year, except in the case of leap years, where it is incremented by two. Both of these values are stored in arrays to be accessed later in the algorithm.

```
def feria_and_epact():
    for i in range(0, 84):
        if i == 0:
            epact.append(20)
            feria.append(7)
        else:
            incrementer = 11
            if i%14 == 0:
                incrementer = 12
            # the -1 value just returns the last value in the list
            next_epact = (epact[-1] + incrementer)%30
            if next_epact == 0:
```

```
            next_epact = 30
        epact.append(next_epact)
        x = (feria[-1] + 1)%7
        if (i+1)%4 == 0:
            x = (feria[-1] + 2)%7
        if x == 0:
            x = 7
        feria.append(x)
```

This gives us the Epact and Feria on the 1st of January, but we must also get these values for the 1st of March. Conveniently, the Epacts are the same values on both dates [8, p. 870].

```
march_fer = (fer + 31 + feb)%7
if march_fer == 0:
    march_fer = 7
return march_fer
```

I also outline a method for getting the amount of days from a given Feria to the next Sunday. This method is used when working with the other tables, for example, Dionysius, to determine Easter Sunday, and I thought it best to outline it here. To get to the next Sunday from a given Feria, you take 8 less the value of the Feria. If the Feria happens to be 1, however, this value becomes 0 (as a 1 indicates you are already on a Sunday, and movement to the next Sunday is unnecessary).

```
get_to_sunday = 8 - feria
if feria == 1:
    get_to_sunday = 0
```

With all these values calculated, we may now look at computing the date of Easter. For the sake of computational simplicity, in my algorithm, I proposed that the days of March and April be combined, to create a single 61-day conglomerated month, the date of which I have called 'Day of March' in my code. Then, when all calculations are concluded, the months are split back up, and the results are converted back into day-month format. To convert back from the conglomerate month, the following method will suffice:

```
month = "March"
if(day_of_march)>31:
    month = "April"
    day_of_march -= 31
return month, day_of_march
```

When we get to the beginning of March, we split the Epact into three cases: the Epact can be less than 14, equal to 14 or greater than 14. The calculations to be performed vary depending on which case we are in. I've included a snippet of code in Python to showcase how the algorithm plays out, along with an explanation of my reasoning in English.

Case 1: Epact < 14 In the case that our Epact is lower than 14, this implies that Easter cannot possibly fall in March. Recall that the earliest allowable date of Easter is the 25th. If the Epact is less that 14, then the latest possible value of the Full Moon (Lunar 14) is 14th of March, which corresponds to the Epact being 1. Thus, the latest that Easter can possibly occur is the 21st of March, a week from the date of the Full Moon. This value is too early for Easter.

Abstractly, my proposal was to skip ahead 30 days to the next Lunar Cycle, bringing all relevant values forward as necessary. From there, perform the required calculations to get the date of Easter, and then revert the dates back to their non-conglomerate dates.

We start on the 31st Day of March, and determine the Feria on this date. Because we added 30 (one lunar cycle), our Epact value remains the same. We take 14 less the Epact to get the number of days until the next Full Moon (Lunar 14), and add this number of days to our 'Day of March' value, and to our Feria value, interpreted modulo 7.

With the Lunar 14 date determined, we must then get the distance to the next Sunday, as outlined earlier. Add this value to the 'Day of March' to get our Date of Easter in the conglomerated month. Run the conversion method, and we get our Easter Date, month included.

```python
# move to new cycle
day_of_march += 30
# get march 31st feria
march_fer = (march_fer+30)%7
if march_fer ==0:
    march_fer =7
# get to next lunar 14, move march date & feria to that day
epact_filler = 14-m_e
day_of_march += epact_filler
march_fer = (march_fer + epact_filler)%7
if march_fer == 0:
    march_fer = 7
# get to next sunday after lunar 14
get_to_sunday = 8-march_fer
if march_fer == 1:
    get_to_sunday = 0

day_of_march += get_to_sunday

if day_of_march>31:
    mon = "April"
    day_of_march -= 31

print("Easter falls on : {} {}".format(mon, day_of_march) )
```

Case 2: Epact = 14 Our first step if the Epact is 14 is the same as above. We know that Easter cannot possibly fall in March (or this Lunar Cycle), so we jump ahead 30 days to the next cycle. Following this, we only need find the next occurring Sunday to find our Easter date.

```
# move to new cycle
day_of_march += 30
# get march 31st feria
march_fer = (march_fer+30)%7
if march_fer == 0:
    march_fer = 7
# get next sunday
get_to_sunday = 8-march_fer
if(march_fer) == 1:
    get_to_sunday = 0

day_of_march += get_to_sunday

if(day_of_march)>31:
    mon = "April"
    day_of_march -= 31

print("Easter falls on : {} {}".format(mon, day_of_march) )
```

Case 3: Epact > 14 When the Epact is greater than 14, the determination of Easter becomes less clear cut, the Epact can be greater than 14, and allow for a date in Easter, though the time frame for this to occur is thin. Our first task is to reset the Lunar Cycle. Our Full Moon has already occurred, so we must move into the next Cycle, and find the Lunar 14 there. This distance is calculated by taking 30 less the current Epact, and adding 14 to that value. Find the Feria on this date in a similar manner, ensuring that the resultant value is taken mod 7. From here, we get the distance to the next Sunday, and add it to our 'Day of March'. If the 'Day of March' value is greater than or equal to 26, our work is done, and we have found our date of Easter.

If we find that the 'Day of March' value is smaller than 26, then we need to move into the next Lunar Cycle. Find the Lunar 14 value in the next cycle by taking the current Epact, and adding 14 plus the distance from the Epact to the beginning of the Lunar Cycle. We already know our current Feria is 7 (Sunday). Add the distance between now and the next Full Moon to the 'Day of March', and to the Feria mod 7. From this date, move forward to the next Sunday, which will be Easter.

```
# gets distance between beginning of lunar cycle
reset_epact = 30-m_e
# gets to the next full moon (eg if epact is 20, value is 10+14)
get_to_14 = reset_epact + 14
```

```
# set march date as first full moon
day_of_march += get_to_14

# get feria on the day
march_fer = (march_fer + get_to_14)%7
if march_fer == 0:
    march_fer = 7
# get feria on full moon march
# get distance from feria to easter sunday
get_to_sunday = 8-march_fer
if march_fer == 1:
    get_to_sunday = 0
# get easter sunday
day_of_march += get_to_sunday

# if the easter date falls on or before the 26th, its too early
if(day_of_march)<26:
    # define the date and epact that are too early
    too_early_date = day_of_march
    too_early_moon_age = get_to_sunday + 14
    too_early_feria = 7

    # reset to next lunar age, then go back to lunar 14 once more
    too_early_moon_age_reset = 30-too_early_moon_age
    get_to_14_2 = too_early_moon_age_reset + 14

    # get date & feria on next full moon
    day_of_march += get_to_14_2
    march_fer = (too_early_feria + get_to_14_2)%7

    if march_fer == 0:
        march_fer = 7
    get_to_sunday = 8-march_fer
    if march_fer == 1:
        get_to_sunday = 0

    # get easter sunday (Taking one away solves a discrepancy with the table, if the code g
    day_of_march += get_to_sunday - 1

    if(day_of_march)>31:
        mon = "April"
        day_of_march -= 31

    print("Easter falls on : {} {}".format(mon, day_of_march) )
```

The algorithm as outlined above will provide the Latercus Reckoning of

Easter in all cases, in the sense that the values predicted here agree with those tabulated in the Oxford Companion to the Year [8]. The Lune for Easter can be calculated by adding the distance from the Lunar 14 before Easter and the distance to the Sunday. As a consequence of our algorithm, the Lune for Easter must lie between the values of 14 and 20. I used this Lune to calculate the Lune for the Initium.

The Initium and its related Lune is calculated using the ‘Day of March’ value we used to determine Easter. The Initium is 40 days prior to Easter, counting inclusively. The Lune for the Initium is calculated by taking 40 from the Lune of Easter, and interpreting the result modulo 30. Since the Lune for Easter lies between 14 and 20, it follows that the Lune for the Initium lies between 4 and 10 (Since subtracting 40 is equivalent to subtracting 10 modulo 30).

Since the 40 is inclusive, we take 39 from the ‘Day of March’ value. If the value is negative, our Initium occurs in February. We can adjust the resulting value appropriately to give us our February Initium, similar to how we adjust between March and April.

```
def initium(day_of_march, year, epact):
    # Get initium Epact
    epact = (epact - 40)%30
    # Adjust february depending on if leap year
    feb = 28
    if year%4==0:
        feb = 29
    # Get initium date and month
    init = day - 39
    month = "March"
    if init <= 0:
        month = "February"
        init = feb - -init

    return init, month, epact
```

6 Conclusion

Much can be said towards the topic of Computus. Its history and scope is so great that one could not reasonably provide a comprehensive understanding of the topic in perpetuity, especially in the scribings I have provided. The methods I have outlined above are only a baseline understanding of their respective Easter reckonings, and it is likely that much of the nuance has been lost, from a historical standpoint. That is to say nothing of the plethora of documentation and manuscripts being digitised online, as well as those still being discovered in catholic churches to this day.

From a computational standpoint, I was successful in producing three online calculators for the methods I have outlined in this paper, per my impetus of the project. These calculators can be found on my websites here (taidghmurray.ie/coding-work) and here (http://www.computus.app). The source code can be found on my github here(github.com/tmurray19/CS4101). With more time on my hands, I intend to revisit this project and expand it, including an online calculator for the Victorius Easter Table, another method for reckoning Easter lost to the march of progress.

From a personal standpoint, I was elated to have such an intriguing project to work on. Whether through a stroke of luck or a twist of faith, I was given the opportunity to work on an ideal project, one that meshed well with my unique skill set and interests. The topic afforded me insight into a subject I had little understanding of. It illuminated just how vast and complex reckoning the date of Easter can be to me. I am extremely thankful to have been given the opportunity to work under the two scholars, Jacopo Bisagni and Immo Warntjes. It was a delight to get to interface with bright minds who were passionate about the subject.

Other terms

Kalendas, Nones and Ides: The 1st, 5th (7th for short months), and 15th (13th for short months) days of a given month, based on the phases of the moon. The Romans used these three 'anchors' in a month to reference the rest of the days for that month[4].

[4]http://www.polysyllabic.com/?q=calhistory/earlier/roman/kalends

References

[1] Gregory XIII. Inter gravissimas, 1582.

[2] Sean E. Urban and P. Kenneth. Seidelmann. *Explanatory supplement to the Astronomical almanac.* University Science Books, 2013. pages 587 - 595.

[3] Theodor Mommsen. *The History of Rome*, volume Vol. 1 of *The Period Anterior to the Abolition of the Monarchy.* Richard Bently, 1864. page 216 - 218.

[4] Ambrosius Aurelius Theodosius. Macrobius and Robert A. Kaster. *Saturnalia.* Harvard University Press, 2011. pages 137, 155.

[5] Carl Friedrich Gauss. Berechnung des osterfestes, 1800.

[6] S. Vince, N.L. de La Caille, J. Bradley, T. Mayer, and F.X.F. von Zach. *A Complete System of Astronomy.* Number v. 1 in A Complete System of Astronomy. J. Mawman, 1823. page 566.

[7] The Editors of Encyclopaedia Britannica. Dionysius exiguus. *Encyclopædia Britannica*, Apr 2013.

[8] B.J. Blackburn and L. Holford-Strevens. *The Oxford Companion to the Year.* Holford-Strevens, Leofranc. Oxford University Press, 1999.

[9] Daniel Mccarthy and Dáibhí Ó Cróinín. The 'lost' irish 84-year easter table rediscovered. *Peritia*, 6-7:227–242, 1987.