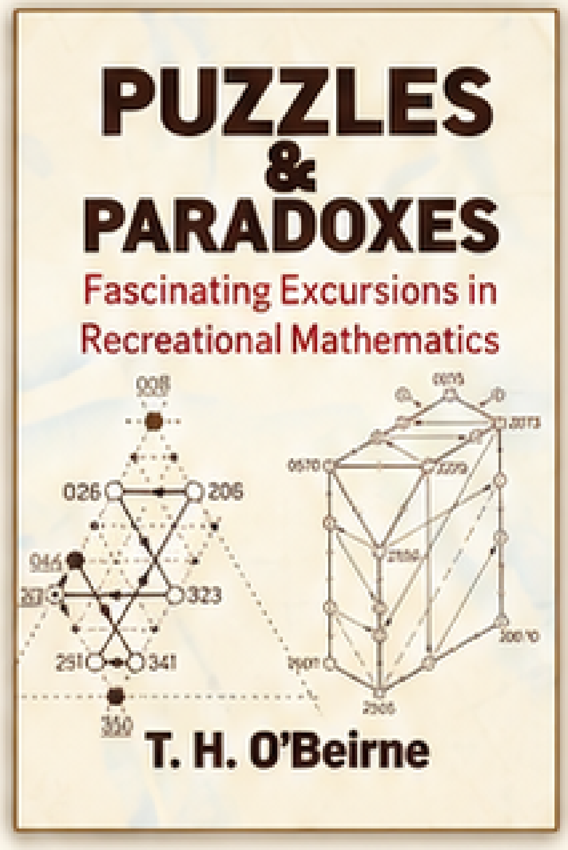
Review of T. H. O'Beirne's 'How ten divisions lead to Easter'



New Scientist, No. 228, 30 March 1961, pp. 828–829

This short article is an elegant example of recreational mathematics with practical consequences.

O'Beirne briefly explains why Easter varies from year to year, then reduces a historically intricate problem to a compact chain of integer operations.

The article's enduring merit lies in its clarity: it is historically aware, mathematically tidy, and exceptionally friendly to implementation in code.

It is important to see that the ten-division procedure is not a new ecclesiastical rule, but a cleaner computational formulation of the Gregorian computus—the traditional method by which the Christian Church determines the date of Easter.

Many presentations of Gauss's Easter algorithm require a patchwork of awkward exceptional corrections to handle rare late-April anomalies. O'Beirne's presentation is widely admired because it makes those adjustments part of the arithmetic itself. The result is a ten-division sequence of integers that yields the correct answer for every year in the Gregorian calendar, without ad hoc fixes.

Useful links

Original Google Books preview: https://books.google.com/books?id=zfzhCoOHurwC&pg=PA828

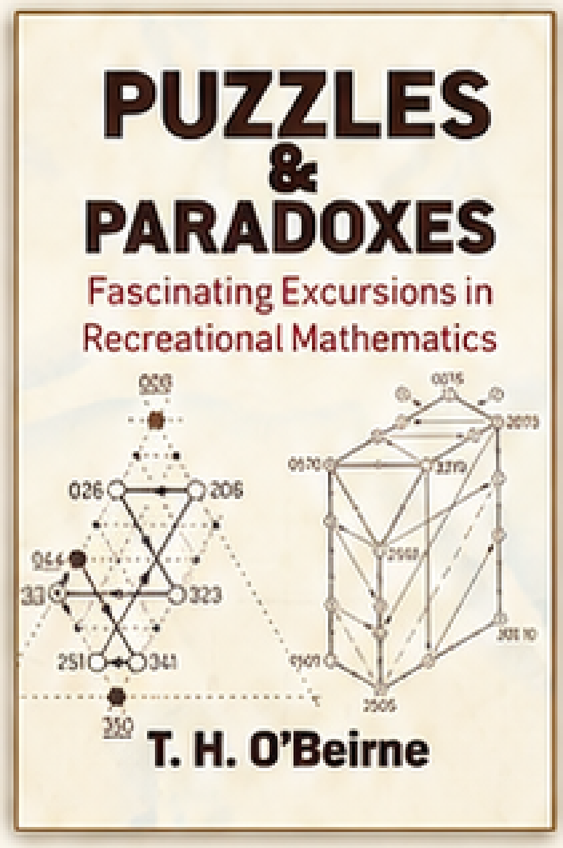
Wayback Machine capture list: https://web.archive.org/web/*/https://books.google.com/books?id=zfzhCoOHurwC&pg=PA828

Book / cover reference: https://www.amazon.com/Puzzles-Paradoxes-Fascinating-Recreational-Mathematics/dp/0486246132

Note: This page is a review and commentary, not a reproduction of the original article.

Gauss's original rule and the later improvement

Why O'Beirne's formulation matters



Gauss's Easter rule was a major arithmetical achievement, but many practical presentations of it were somewhat untidy because they needed rare exceptional corrections.

One defect was later noticed, and Gauss himself supplied a manuscript correction. O'Beirne explicitly mentions that manuscript note.

O'Beirne emphasizes a purely arithmetical procedure: the algorithm should not depend on ad hoc verbal exceptions added at the end.

The issue concerns rare late-April anomalies: without the proper correction, certain formulations can drift into impossible or incorrect late-April outcomes.

O'Beirne notes a correction connected with the case that would otherwise give 26 April in 1981, and shows how his own version folds the needed adjustment into the arithmetic.

> *O'Beirne's achievement is not to change Easter, but to make the calculation internally self-correcting.*

This is the real improvement—not a different rule of the Church, but a cleaner way of encoding the same Gregorian computus.

The correction term m that O'Beirne introduces is usually 0 and only occasionally becomes 1, thereby automatically handling the rare exceptional years.

In this way, O'Beirne preserves the historical substance of Gauss's method while giving it a form that is rigorous, compact, and well suited to modern computation.

O'Beirne's article thus stands at the meeting point of history, number theory, and practical programming—an elegant example of recreational mathematics with real intellectual depth.

Note: This page is a review and commentary, not a reproduction of the original article.

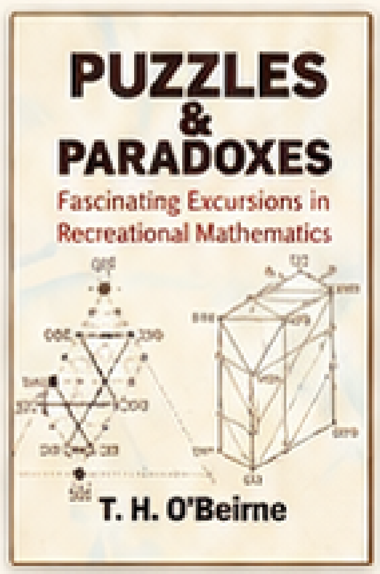
Excerpt from my PHP code implementation

The Easter calculation in code

```
$easter_year = $vacation_year;
$a = $easter_year%19;
$b = intval($easter_year/100);
$c = $easter_year%100;
$d = intval($b/4);
$e = $b%4;
$g = intval((8*$b + 13)/25);
$h = (19*$a + $b - $d - $g + 15)%30;
$i = intval($c/4);
$k = $c%4;
$l = (2*$e + 2*$i - $h - $k + 32)%7;
$m = intval(($a + 11*$h + 19*$l)/433);
$n = intval(($h + $l - 7*$m + 90)/25);
$p = ($h + $l - 7*$m + 33*$n + 19)%32;
$easter = sprintf("%04d-%02d-%02d", $easter_year, $n, $p);
```

Where the improvement is

- The variables a through l follow O'Beirne's ten-division arithmetic almost step by step.
- h encodes the moon-related part of the calculation, and l aligns the result with the weekday cycle.
- The crucial improvement is the line "m = intval(($a + 11∗$h + 19∗$l)/433);"; this auxiliary correction term is usually 0 but occasionally 1, and it automatically handles the rare late-April exceptional cases.
- The next two lines convert the corrected result into the month n (March or April) and the day p of that month.
- In the full program, the Easter date is then reused to derive Corpus Christi by adding 60 days.



The code is therefore a faithful implementation of O'Beirne's cleaned-up version of the Gregorian Easter algorithm.

Note: This page is a review and commentary, not a reproduction of the original article.