Spreadsheet Investigations in Modular Arithmetic

Steve Sugden, Bond University, Australia

Abstract

Modular arithmetic has sometimes been regarded as a bit of a curiosity, at least by those unfamiliar with its importance to both abstract algebra and number theory, and with its numerous applications. However, with the ubiquity of fast digital computers, and the need for reliable digital security systems such as RSA, knowledge of this important branch of mathematics is now considered almost essential for IT professionals. Indeed, computer arithmetic is, *ipso facto*, modular. This paper describes the use of a modern graphical spreadsheet (Microsoft's Excel) to clearly illustrate the basics of modular arithmetic, and to solve certain classes of problems. Via conditional formatting and observation of patterns in Excel, students may gain structural insight, form and test conjectures, and solve problems.

1. Background

1.1. The practical utility of number theory

In our society of the 21st century, it seems that even the most sublime of human pursuits may be called upon to justify its existence in terms of dollars contributed to some "practical" cause. Given such hard constraints, it can be difficult to defend certain esoteric branches of mathematics. But for the advances of information technology of the past few decades, a defence for number theory in terms of "pragmatics" may have been something of a challenge. Despite Hardy's claim that "I have never done anything useful" [1], the "practical" utility of number theory nowadays is undeniable; its application to mathematical cryptography and cryptology alone being enough to stay the cries of the pragmatists.

1.2. The teaching of modular arithmetic

In the Australian state of Queensland, modular arithmetic is currently out of vogue in the high-school curriculum, being relegated to an "optional" topic in *Mathematics C*. State-wide, this unit was taken by about 10% of seniors in 2006 [2]. Previously described as "clock-arithmetic", at least at high-school level, modular arithmetic is often presented as something of a curiosity, with little practical application. Perhaps with the advent and dominance of digital timepieces, this terminology has faded. Created by Gauss in his *Disquisitiones Arithmeticae* [3] in 1801, modular arithmetic was indeed regarded as a curiosity at that time.

The situation regarding coverage of modular arithmetic seems to be somewhat better at tertiary level, where the material may be treated from a slightly more mature standpoint, with a view to obvious applications. However, in the author's experience, few IT graduates appreciate the fact that integer arithmetic (the only kind which is exact) on a digital computer is none other than modular arithmetic. This is so, since all integer operations are performed modulo some power of 2. This is the nature of a digital, binary CPU. To graduate with a degree in CS or IT and not know this seems incredible, at least to this writer.

1.3. "No maths please, and we don't care much for programming either!"

Increasingly, we see the watering-down of IT degrees: less programming, little or no low-level programming, and often no units on operating systems, compilers, theory of programming languages. There is nowadays considerable pressure, at least in the author's own country (Australia), to accommodate fee-paying students who make no secret of their aversion not only to mathematics, but also to computer programming. This then begs the question: "Why are such

students enrolled in an IT degree?" To enrol in IT, but then eschew not only maths but programming sounds crazy to my ancient ears (and also to many of my ageing colleagues). Perhaps we are old fashioned in our belief that CS/IT is a scientific discipline, in which appropriate, and rigorous mathematics is a necessary part of the "toolkit" for a practitioner.

1.4. Early days

I confess that I have always found number theory non-trivial. In the primary classroom, in the late 1950s and early 1960s, I learned of prime numbers, factorization of integers, highest common factor (nowadays called gcd – greatest common divisor), lowest common multiple, and other curiosities. I was educated in various schools in New South Wales, Australia. At Sydney Technical High School, in 1968, my last year of secondary school, I studied "theoretical arithmetic", which, of course, is number theory by another name. From 1974-1978 I was enrolled part-time in a coursework program *Master of Scientific Studies* at the University of Queensland. I took a unit *Analytic Number Theory* taught by Professor Clive Davis. I was at first disappointed when he told me that I was the only one to enrol, and assumed that it would be cancelled. Not so, and he ran the whole thing as if the lecture room were full of students. I found it very interesting and we covered, among other topics, the Prime Number Theorem, the transcendence of e and π, plus lots of interesting material on the Riemann zeta function (actually first defined by Euler). Material on additive number theory (partitions of integers) was also covered.

1.5. Today

These days, I am at Bond University, where unfortunately, there is very little mathematics: just a few units of statistics plus *Business Mathematics* and *Analytical Toolkit* (discrete mathematics plus some very basic statistics for IT students). Recently I wrote elsewhere about the difficulties encountered at Bond while trying to get across the basics of discrete mathematics to IT students. I also described my response to this problem: in a nutshell, it is the use of Excel to convey mathematical principles to students whose algebra is, in many cases, almost non-existent [4]. Several negative factors usually conspire to create a huge problem here, but the most significant component is always the very poor mathematical knowledge and skillset of a typical IT student at Bond. It is my impression that this problem is not confined to Bond, but essentially ubiquitous in Australia. Recent anecdotal evidence seems to indicate that the situation is perhaps even worse in the UK, USA and Canada.

1.6. Formalism is a problem

From my perspective, there seems today to be an almost total lack of ability and/or experience on the part of many students to deal with *formality* in mathematics. I refer to formal definitions, formal manipulation of symbols, i.e., *algebra*, and formal or semi-formal reasoning using accepted principles of logic. Whatever their high-school mathematics curriculum was, the students appear to have emerged with very poor problem-solving skills, little facility for algebra, and not much appreciation of the wide applicability of mathematics to solve a range of problems. They even a lack of knowledge of "standard tricks", such as quick checks for divisibility (sum of digits is a multiple of 3 or 9, or last digit is even). Is 127652761865675448 a prime? The calculator is quickly sought for "difficult" problems like 6 x 7, or even 6 + 7! How do we deal with such atrocious levels of mathematical understanding when students are enrolled at university for degrees where basic maths is required? I have written of the possibility of a partial, stopgap solution in [4].

2. Some applications of modular arithmetic to IT

2.1. Hashing

Hashing, or scatter storage, is the process where a key (usually some text string) must be mapped to an array location (index or address). The usual approach is to use a fairly simple function of the key, modulo *n*, where *n* is prime. The function needs to be simple, or more precisely, efficiently computable, as the only advantage of using hash functions for record retrieval is speed; everything else is negative.

2.2. Simultaneous linear congruences (SLCs)

$$x \equiv 1 \pmod 3$$
$$x \equiv 2 \pmod 5$$
$$x \equiv 5 \pmod 7$$

Solution of SLCs such as the set of three shown above may be viewed as computing the intersection of several arithmetic sequences, i.e., what set, or sequence is formed by the elements common to each sequence? Viewed from this perspective, the possibility of an empty sequence arises. For example, one cannot have *x* both congruent to 1 (mod 4) and to 0 (mod 2). Necessary and sufficient conditions for the existence of solutions are given by the Chinese Remainder Theorem, but I do not even mention this in class. I merely show how to find the general solution (or prove its non-existence) by algebra. This approach is complemented by one or two Excel models. The simple version of the Excel model merely lists the arithmetic sequences defined by the linear congruences. The intrinsic function COUNTIF is then used to identify solutions, which are highlighted by *conditional formatting* (CF). I require the students to be familiar with both algebraic and Excel methods. Where appropriate, this is my approach in general.

2.3. Modular inverse and modular exponentiation

The RSA public key system of encryption [5] relies on, among other things, efficient computation of modular inverse and modular exponential (see next section). Briefly, the private and public keys are mutual inverses with respect to the system modulus, which is the product of two large, distinct primes. In the simplest application of RSA, these two keys are the encryption and decryption keys respectively. Suppose we wish to compute the inverse of 4 with respect to modulus 11. Denoting this quantity by *x*, it may be defined as the solution, or a solution of the equation $4x \equiv 1 \pmod{11}$. The perennial questions of existence and uniqueness arise, plus of course, how to compute the inverse if it exists.

I have found that many students miss even the entire concept of modular inverse, answering "0.25" to the question "compute $4^{-1} \pmod{11}$". Thus, I go out of my way to avoid this misconception. I just use a table in Excel, so to "compute" mod inverse, we simply consult the table. It becomes clear via *conditional formatting* when the inverse does not exist. I then ask the students to contemplate why inverse does not always exist, but when it does, it is unique. For many more examples where CF patterns are used to illustrate principles or solve mathematical problems, see [9, 10].

11

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	1	3	5	7	9
3	3	6	9	1	4	7	10	2	5	8
4	4	8	1	5	9	2	6	10	3	7
5	5	10	4	9	3	8	2	7	1	6
6	6	1	7	2	8	3	9	4	10	5
7	7	3	10	6	2	9	5	1	8	4
8	8	5	2	10	7	4	1	9	6	3
9	9	7	5	3	1	10	8	6	4	2
10	10	9	8	7	6	5	4	3	2	1

Figure 1

Students can easily see that modular inverse is obtained by table lookup. They are also required to complete an assignment on RSA [6]. For this, they must write functions for modinv and modexp in VBA (Excel-hosted). Even though such functions are each 6 or 7 lines of code, they struggle. The algorithm for modinv is essentially linear search. Just like table lookup: start at 1, keep looking until you find a 1 in the row of interest, or see that it is not there (non-existence of inverse). To compute $x^n \pmod m$ is also an essential step of the RSA encryption (and decryption) process. Naive code for these is shown below; the code for modular inverse assumes its existence.

```
Function modexp(x As Long, n As Long, m As Long) As Long
  modexp = 1
  Do While n > 0
    modexp = (modexp * x) Mod m
    n = n - 1
  Loop
End Function

Function modinv(t As Long, m As Long) As Long
   modinv = 0
   Do
      modinv = modinv + 1
   Loop Until modinv * t Mod m = 1
End Function
```

2.4. Discovering properties of primes

I use modular multiplication tables with CF in Excel to clearly show certain patterns. Two sliders (also known as scrollbars) are employed: one for the modulus and one to specify a value to highlight wherever it appears in the table. For example, we may highlight 1, thus clearly illustrating the concept of modular inverse. Also in the model is automatic counting of the frequency of each residue in the table, and that of the special value in each row. It then becomes obvious when modinv fails to exist, and when it is unique. If we search for 0, then some very interesting patterns occur. Why do such special arrangements of 0 occur for the numbers 14, 22, 26, 34, 38, 46?

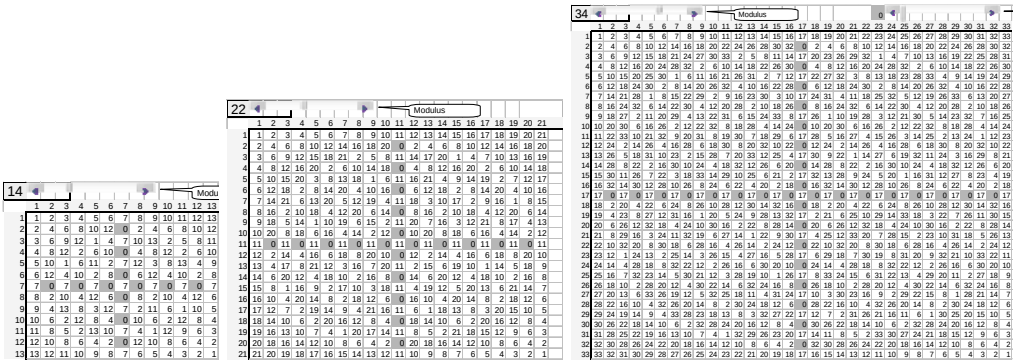

Figure 2

When and why does 0 appear in the table? Alternatively, "for which moduli do zeroes *not* appear in the table?" When does each row/column contain a 1? When does each row/column contain a full set of residues? Why is 1 sometimes absent from a row? Why does the "sparse parallel line" pattern of Figure 3 appear as it does?

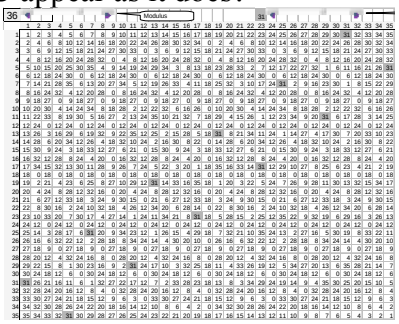

Figure 3

2.5. Terry Tao problem

Fields Medal winner Terry Tao recently published a second edition of his little book on mathematical problem-solving [7]. It is a very interesting and useful little volume, giving much insight into problem-solving. In a very lucid and entertaining manner, the author shows how he develops successful chains of reasoning. Plausible but ultimately fruitless paths are tried but discarded as false steps along the way. Like Polya, he gives some general principles for mathematical problem-solving. For number-theoretical or Diophantine problems, one of the more useful tools in the toolkit is modular arithmetic.

Exercise 2.3 on p24 of [7] is to show that $x^4 + 131 = y^4$ cannot have integer solutions. Ignoring Terry's tacit advice to use modular arithmetic, at first I tried to prove this by a clever(?) rearrangement and factorization: $\left(x^2-4\right)\left(x^2+4\right)=3\left(y^2-7\right)\left(y^2-7\right)$. The LHS is congruent to 0 or 4 (mod 5), whereas the RHS is always 0 or 3 (mod 5) . I was pretty pleased with myself, but soon realized that an even simpler proof is possible! This was inspired by simply tabulating the original problem in Excel, with a range of small moduli. Details can later be filled-in rigorously, by modular algebra. Here we have yet another use of Excel to produce a clearly-recognized pattern, forming the basis of a rigorous proof.

3. Conclusion

I have very briefly outlined how the modern graphical spreadsheet may be used to create interesting and visually appealing lessons in the basics of modular arithmetic. There is much more that could be done, but in any case, the literature is clear: students learn mathematics in different ways. The traditional way of algebra has been successful for only a minority of students [8]. Spreadsheets offer the chance for students to leverage their powerful visual perceptive abilities in support of moving to the more formal mathematical language of algebra, and basic notions of proof [8]. Unfortunately, such opportunities are often routinely bypassed by math educators [4].

The topic of modular arithmetic is interesting mathematically, and has many applications. The modern spreadsheet environment offers a goldmine of possibilities for the mathematics teacher when modular arithmetic is investigated. For students whose algebraic background is modest, the spreadsheet still allows many concepts to be conveyed. Once these are seen graphically, the algebraic approach can then be used to establish theorems and solve problems. For a very brief summary of some of my other uses of Excel in math instruction, see [4]. For recent summaries of applications by others, see [11, 12].

Last year a mathematical colleague told me of a paper he recently had published in a prestigious international journal. The basic patterns that made up the "visual proof" were discovered by constructing an Excel model. This was omitted from the paper as the author had serious doubts concerning the paper's acceptance if it were revealed that the theorems proved therein were discovered or motivated by using a spreadsheet. I believe it is time that mathematics educators stopped regarding the ubiquitous Microsoft Excel as merely an accounting tool and seriously examined its possibilities for the illustration of the many beautiful patterns of mathematics to their students.

References

1. Hardy, GH (1940). *A Mathematician's Apology*, p150.
2. Stevens, W (2007). *Private communication*, Queensland Studies Authority.
3. Gauss, CF (1801). *Disquisitiones Arithmeticae.*
4. Sugden SJ (2007). Spreadsheets: an overlooked technology for mathematics education. *The Australian Mathematical Society Gazette*, **34**(2): 68-74, invited paper.
5. Gilbert, GT and Hatcher, RL (1999). *Mathematics Beyond the Numbers*, Wiley, 526-531.
6. Sugden SJ (2003). Practical Number Theory in a Discrete Mathematics Class: The RSA Cryptosystem in Microsoft Excel and on the Web. *Remarkable Delta 03 Communications*, 252-258. Queenstown, NZ, November 2003.
7. Tao, T (2006). *Solving mathematical problems*, Oxford University Press.
8. Sutherland, R, and Rojano, T (1993). A Spreadsheet Approach to Solving Algebra Problems. *Journal of Mathematical Behaviour* **12**(4): 351-383.
9. Sugden SJ (2005). Colour by Numbers: Solving Algebraic Equations without Algebra. *Spreadsheets in Education*, **2**(1): 101-114.
10. Abramovich S, Sugden SJ (2003). Spreadsheet Conditional Formatting: An Untapped Resource for Mathematics Education. *Spreadsheets in Education* **1**(2): 85-105.
11. Baker JE, Sugden SJ (2003). Spreadsheets in Education: The First 25 Years. *Spreadsheets in Education*, **1**(1): 18-43.
12. Sugden SJ (2006). Spreadsheets in Education: A Peer-Reviewed Medium For Active Learning. *ICTM3 Proceedings*, Paper #112 (CD-ROM). Istanbul, Turkey, 30 June - 5 July 2006.