Preface

INSPECTION COPY

The following is a true story…

Larry Walters, a delivery driver for a Holywood film production company, had a boyhood dream to fly. As he sat in his backyard watching jets fly overhead he hatched his daring scheme. He purchased 45-50 weather balloons from an Army and Navy surplus store, tied most of them to a garden chair which he had tethered to his friend's car, and filled the balloons with helium gas. On June 2^{nd} 1982 he strapped himself into the chair with some food, drinks, water containers to act as ballast and a pellet gun. He figured he would pop a few of the many balloons when it was time to descend.

Needless to say, Larry doesn't appear to have spent much time studying maths or science at school. Helium is lighter than air and will generate a lift of just under one ounce per cubic foot at sea level (or about 1 gram per litre if you prefer metric measurements). This doesn't sound much, but total lift depends on the number of balloons and their size. The balloons were quite large – around 3½-4 foot in radius (or 7-8 foot across, based on what Larry said at the time and from experiments with weights that he and his friends are reported to have conducted). Assuming a 4 foot radius each balloon would hold around 270 cubic feet of helium* and thus generate a lift of over 14lbs (6.4kgs). So 40 or more of these would lift a man, chair and ballast without any problems! But lift is reduced by around 7.5% per 1000 feet, depending on the variations of both pressure and temperature with altitude. So, luckily for Larry, sooner or later he would stop rising…

Larry's plan was to cut the tether lines and lazily float up to a height of about 30 feet above his back yard and maybe out across the nearby desert area towards the Mountains, enjoying a few hours of flight before coming back down. But things didn't work out quite as Larry planned… When his friends cut the nylon cords anchoring the chair to the car he did not float lazily up to 30 feet. Instead, he streaked into the Los Angeles sky as if shot from a cannon, pulled aloft by the powerful lift of the helium balloons. He didn't level off at 100 feet, nor did he level off at 1000 feet. After climbing and climbing he levelled off at 16,000 feet. At that height he felt he couldn't risk shooting any of the balloons lest he unbalance the load and really find himself in trouble. So he stayed there, drifting, very cold and frightened, for more than 2½ hours.

INSPECTION COPY

Then his path crossed the primary approach corridor of Los Angeles International Airport. A TWA pilot first spotted Larry. He radioed the control tower and described passing a guy in a garden chair… with a gun… at 16,000 feet! After a period of some disbelief radar confirmed the existence of an unidentified object floating high above the airport. Eventually Larry gathered the nerve to shoot a few balloons before accidentally dropping the gun, but fortunately by then he had done enough and slowly descended. As he approached the ground the hanging tethers tangled and caught in a power line, resulting in the local electricity authority having to cut off the supply from a Long Beach neighbourhood while Larry was rescued. He eventually managed to climb down to safety where waiting members of the Los Angeles Police Department arrested him. As he was led away a reporter dispatched to cover the daring flight asked him why he had done it. Larry replied nonchalantly, "A man can't just sit around."

Report developed from contemporary newspaper reports and material published on the Darwin Awards website (www.darwinawards.com) and Mark Barry's more complete "Lawn Chair Pilot" website (http://www.markbarry.com)

Corrections to published material kindly provided by Mark Barry (personal communication). Mark has interviewed several of those present, excluding Larry Walters himself as sadly Larry later took his own life. Photos and an audio track covering the flight are included on the Lawn Chair Pilot web site. The photo here shows Larry shortly after takeoff, having just lost his glasses!

A small number of enthusiasts continue to fly 'cluster balloons' like Larry. See http://www.clusterballoon.org/ for more details

* see Appendix 1 for details of this calculation

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

Contents Page

INSPECTION COPY

INSPECTION COPY

Structure of the book

INSPECTION COPY

This is a book for people who want (or need) to analyse practical real-world problems, but are unsure how to proceed with confidence. In an effort to dispel such fears and difficulties I examine and solve a wide variety of real-world problems through a consideration of the processes of measurement and the application of very simple computational methods. This approach enables us to examine, illustrate, test and develop a broad range of ideas and procedures that work, from the comfort of our own 'garden chair', without recourse to complicated equations or programming.

Throughout this book I make use of modern desktop computer systems and the mathematical software that has been developed for them, treating these as a special kind of experimental laboratory. The computer as laboratory enables a quantum leap in our capabilities as number crunchers, professional problem solvers or simply interested amateurs. This is in part a result of the speed of computers, but increasingly as a result of their incorporation of past knowledge and experience - the accumulated results, findings and experience of generations of mathematicians. This experience is stored for our quick reference, embodied in procedures that ensure mistakes are minimised and that 'correct' processes are followed, and then presented to us through interfaces (input and output) that are understandable and visually stimulating. Such facilities do not remove or even reduce the need for understanding and insight – to an extent they demand an even greater level of understanding and care - but they do enable the process of investigation to be far speedier and often to be more effective, rewarding and fun.

In order to explore these issues I have divided this book into three main parts:

Part I deals with the why? why do we mainly use a number system based on 10 elements? why do we have numbers like zero (0) and minus 1 (–1)? why don't we (generally) have a symbol for 10, unlike the Greeks and Romans? why do we have different groups of numbers with strange names like Integers, Imaginaries and Irrationals? and why does mathematics seem to be full of strange symbols and equations? Answering these questions involves delving into some of the history and personalities involved in the development of this subject. It also leads us to look at the history of measurement (of lengths, weights, volumes, times, etc.) and the notions of dimension and connectivity. These are fascinating stories in their own right, but also highlight the role of experimental data in formulating our understanding of numbers and the central role they play in practical and theoretical analysis.

INSPECTION COPY

Part II examines the process of discovery through experimentation, focusing on the relatively new world of the computer as laboratory. Computers, and their predecessors electronic and mechanical calculators, and before these vast teams of human 'computers', have long been used to carry out tedious and repetitive numerical processing tasks. But it is a very recent phenomenon that has seen their function developing towards an experimental, interactive, enquiry-led mode of usage. This extends their application beyond that of computational and educational tool, to one in which the tool becomes a key part of the process of discovery and refinement. But our new, computationally rich world, introduces important technical issues that are unfamiliar to most people (except for professional software engineers and some mathematicians). I attempt to address many of these issues, illustrating the materials with a wide range of examples and available software tools. The software tools utilised and discussed include: Microsoft's Excel spreadsheet product; Maplesoft's Maple package, which is particularly good at *symbolic* processing (working with symbols and expressions as well as numbers); and Waterloo Software's MATLab package (MATrix Laboratory), which is a general purpose scientific suite of software with strong support for manipulation of blocks of numbers and expressions. Student versions of Maple and MATLab are available at much reduced prices. Finally, use is made of several innovative online (web-accessible) tools, which are described and discussed in the text. For all of us wishing to use computers the first step is to be able to be able to enter numbers and expressions, obtain meaningful output, and to incorporate these results into documents. The latter requirement presents a number of difficulties, so I also have included a discussion of some of these issues at the end of **Part II**.

INSPECTION COPY

Part III of this book provides a more detailed look at a selection of numbers and number sequences that are widely regarded as 'interesting'. Some argue that all numbers are interesting, and if a number appears uninteresting it is simply our failure to discover its possibilities. In fact there is a simple, but rather weak 'proof' that there are no uninteresting numbers. However, it is not practical to examine every possible number, nor even a large selection, so I focus on a personally selected subset. Many interesting numbers have even more interesting stories associated with them, and some of these are provided in the final Part of the book.

A selection of suggested books to read, web sites to explore and software to use are provided in Appendix 2 at the end of this book. In addition, on the author's web site (www.mdemsith.com) you will find copies of spreadsheets, program extracts, useful web links, additional materials and sample problems.

Intended Audience

I have written this book to be of help and interest to a wide range of readers: for those interested in learning a little more about numbers, mathematics and computer tools in general; for those who are simply seeking a refresher course with a modern perspective; for those who find mathematics and computing a rather daunting but necessary part of the modern world that they wish to understand more thoroughly; and for those interested in using computers as tools of discovery - examining, testing and visualising information and ideas in new ways. To this extent the text may serve as introductory reading for students (and hence as a resource for teachers and lecturers) including those in higher and further education (colleges, universities). It is intended to be of particular value to those studying subjects in the social sciences, physical and earth sciences, and life sciences. Such disciplines demand a thorough appreciation of mathematical and statistical ideas but incorporate very little pure mathematics or software engineering in their coursework.

Although I discuss and make use of some simple mathematics, this is not a maths text and does not assume a detailed knowledge of this subject. Readers who are more confident in some of these areas may find that they can skim through several of the Sections, in particular those dealing with the different groups of numbers that we commonly encounter. Frequent use is made of software tools, but once again little or no knowledge of programming is required. I do, however, explain the principal forms of notation and conventions used in mathematics, and some of those that apply within computing. It is useful to understand where this notation comes from and how it is used in order to be able to read and appreciate these subjects more fully. Thus, in the process of exploring numbers and data, I cover a wide range of topics including many important mathematical concepts - from number theory to topology, and from basic statistics to calculus, but introduced in an accessible and largely non-traditional manner.

This book also explores a wide range of application areas, from understanding gravity to computing the speed of light, from Internet security to digital audio and image processing, and from the exploration of number series to better ways of obtaining arithmetic results. My hope is that readers will find the material both interesting and enjoyable, and will become confident enough as a result to read more widely in this field, to conduct their own experiments with numbers and to explore the wonderful array of resources now available on the Internet. Amongst the latter is the MacTutor web site of University of St Andrews. This site provides bibliographic details of many famous mathematicians and brief discussions on key topics in mathematics. With their kind permission I have used some of their excellent material in the provision of brief profiles of several key figures from the history of this subject.

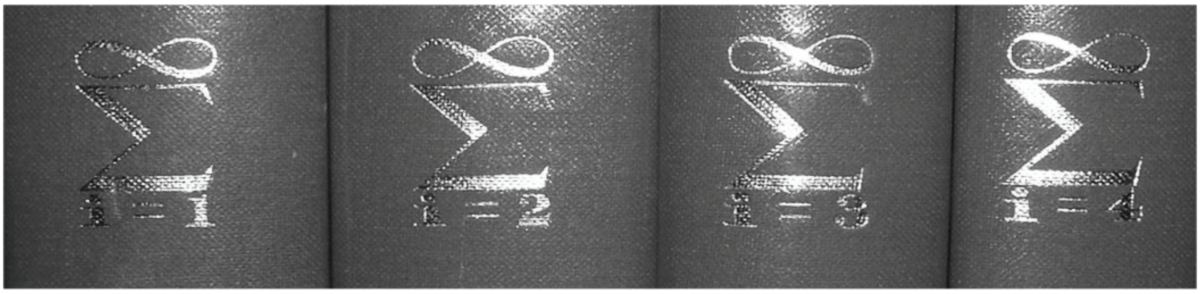
When reading this book it is useful to have a pencil and paper to hand, and if possible, have access to a desktop computer to reproduce and test out many of the examples. I make frequent use of Microsoft Excel as a form of sophisticated calculator, so limited familiarity with Excel is helpful but not a pre-requisite. An essential component of this book is the use of many worked examples, including those involving measurement of the physical world. If in doubt (or even if not) I strongly recommend that readers try reproducing the numerical examples, looking for any errors, weaknesses or unstated assumptions.

INSPECTION COPY

PART I: FOUNDATIONS

INSPECTION COPY

Browsing along the shelves of a library I once came across a set of four books entitled "The World of Mathematics" by James R Newman. Each book had a summation symbol printed on its spine, identifying the volume number, 1, 2, 3 and 4:



I took down the first volume, opened it at random (page 370) and read the following text of a letter, dated January 16th 1913, to a Cambridge mathematician G H Hardy (1877-1947):

> *"Dear Sir,*
>
> *I beg to introduce myself as a clerk in the Accounts Department of the Port Trust Office at Madras on a salary of only £20 per annum. I am now about 23 years of age [in fact he was 25]. I have no University education but I have undergone the ordinary school course. After leaving school I have been employing the spare time at my disposal to work at Mathematics. I have not trodden through the conventional course which is followed in a University course, but I am striking out a new path for myself….*
>
> *I would request that you go through the enclosed papers. Being poor, if you are convinced that there is anything of value I would like to have my theorems published… Being inexperienced I would very highly value any advice you give me. Requesting to be excused for the trouble I give you,*
>
> *I remain Dear Sir, yours truly,*
>
> *S Ramanujan."*

Hardy found the 120 or so results enclosed with the letter extraordinary. Many of these he had no idea how to confirm, but they were so original he wrote back to Ramanujan congratulating him on his work and confirming that he would try and help him.

Replying to Hardy's letter Ramanujan wrote:

> *"I have found a friend in you who views my labours sympathetically. I am already a half starving man. To preserve my brains I want food and this is my first consideration. Any sympathetic letter from you will be helpful to me here to get a scholarship either from the university or from the government."*

In May that year Hardy arranged for him to come to Cambridge on a substantial Scholarship, including a generous allowance for his family remaining in India. Ramanujan came to England and worked alongside Hardy and his colleagues at Cambridge University on a wide range of mathematical problems before continuing ill-health seriously weakened him. Early in 1918 he was elected as a Fellow of the Royal Society and a Fellow of Trinity College Cambridge. In March 1919, ill once more, he returned to India and died there, aged 33 in 1920.

I found this extraordinary and poignant story captivating, and it encouraged me to read many more of the unusual tales that lie behind much of our modern-day science, philosophy and mathematics – we might speculate that if a largely uneducated, sick, poverty-stricken clerk from a backwater of India can achieve so much then surely we can all set our sights high, no matter what the apparent obstacles. But to make progress in almost any subject, especially those with a scientific basis, it is first helpful to understand something of the history, foundation stones and terminology used. Once these have been familiarised taking the next steps, to experimentation and analysis, is a far less daunting prospect. This first stage is the aim of the Sections that now follow.

1 Understandable fears!

INSPECTION COPY

"There are three kinds of people in the world; those who can count and those who can't." Anon

1.1 Numbers and mathematics

Many people claim to be 'number blind' or cannot 'do maths'. Actually, relatively few people (currently believed to be under 5%) exhibit genuine signs of number blindness - finding little or no meaning in numbers or simple combinations of numbers. This is a condition now known as dyscalculia, broadly similar and sometimes associated with the more commonly understood condition of dyslexia (word blindness). Fear of getting the answer wrong or of being subject to ridicule by classmates or colleagues is a much more common and pervasive phenomenon. And unlike many subjects, there is often a definite right or wrong answer so it is very obvious if we are wrong – we cannot bluff our way past this apparently universal truth.

If you were to ask 100 people which subject they liked least at school, or which they found the most frightening or difficult, the great majority would say Maths (or Math if you ask an American audience). Despite the fact that learning a foreign language or musical instrument may well be at least as difficult, if not more so, there is some kind of deeply held fear of mathematics amongst the population at large. This is not based on a fear of numbers, but of the way in which problems that involve numbers seem to need to be turned into symbols and equations (a process known as abstraction) in order to obtain the 'right' answer – or wrong answer for most of us, which is the real cause of our concern! It is clearly far too easy to get things wrong so many avoid the subject altogether.

But if you ask the same 100 people a different question, perhaps one of a more practical nature, such as "how much sales tax (or VAT in Europe) is payable on an item costing 100 pounds (or dollars or euros)?" most would be able to answer immediately, assuming they knew the relevant tax rate. So, for example, given a dress or suit costing £100.00 in the UK, before VAT has been added, nearly everyone will be able to tell you that the dress or suit will actually cost you £117.50, since this figure now includes the £17.50 of VAT which must be added to the basic price (the UK rate being 17.5%). We know this answer is correct because the sums involved are very easy – we can work it out by a simple *forwards* calculation, using multiplication, i.e. £100 *x* 17.5%=£17.50 plus the original £100 gives the answer.

It turns out that the opposite or inverse process, division, is much more difficult. I could have asked the question above the other way around: for example, "given a dress or suit costing £100 including sales tax/VAT, how much is the cost of the dress or suit *excluding* VAT". Well, the answer is clearly not £82.50, because that would mean the VAT was £17.50 which was the answer when the dress or suit cost £117.50 *including* VAT – in fact the answer to this new question must be more than £82.50, and is actually £85.11 (to the nearest whole penny), with VAT amounting to £14.89.

To obtain this solution we either need to work things out *backwards*, i.e. starting with the total and working back to the component parts using division (the *inverse* of multiplication), or we need to carry out a series of guesses, working *forwards* from pairs of guesses (e.g. £80, £90) that we are confident lie either side of the true answer. We then systematically reduce the interval until we are as close as possible to the correct value (see Box 1 for a detailed look at this process).

In the example we have been using the quick solution simply involves dividing the price including VAT, i.e. the £100, by 1.175 and rounding the answer to the nearest penny - not that easy by hand. The number 1.175 is used because this is 100% plus 17.5%, i.e. 117.5/100. To make use of this result we need to be comfortable with fractions and division, whereas the forwards approach and the guessing methods only require multiplication and need minimal understanding of fractions. Perhaps the world is divided into people (and computers as it happens) who can do fractions and those who can't! In practice fractions and percentages are not too difficult and they are so widely used in everyday life that this is one area that is really worth getting to understand: whether it is the price of a dress; deciding how much to tip in a restaurant; seeing how much interest a bank is charging you; or looking at the success rate of a new medical procedure. All share a common basic form - a systematic, tried and tested procedure for obtaining the correct answer using a well-defined series of steps. Such procedures are known as *algorithms* (see also, Box 2).

Box 1. Simple search algorithm for finding the sales tax on a £100 item

INSPECTION COPY

Let's start by using our initial guesses for the dress or suit price of £80 and £90 before sales tax is added: with £80 we get a total price for the dress or suit of £94 including sales tax at 17.5% (£80 + £80x17.5% which is £14) so this is too low. With £90 we get £105.75 - too high. So next we might try 'chopping' our initial range in half to £85 next, which would be very close to the correct answer. We would find £85 is a little too low, so we could halve the new upper interval (£85-£90) and look at £87.50, continuing this process until we have homed in the right answer (stopping when the result equals £100.00, rounded to the nearest 1p). The complete sequence in this example is shown in the following table created in Excel:

Guesses, excl VAT, £	VAT @ 17.5% (to 2 decimal places, dp)	Price, incl VAT, £
80.00	14.00	94.00
90.00	15.75	105.75
85.00	14.88	99.875
87.50	15.31	102.8125
86.25	15.09	101.3438
85.625	14.98	100.6094
85.3125	14.93	100.2422
85.15625	14.90	100.0586
85.07813	14.89	99.9668
85.11719	14.90	100.0127
85.09766	14.89	99.98975
85.10742	14.89	100.0012

INSPECTION COPY

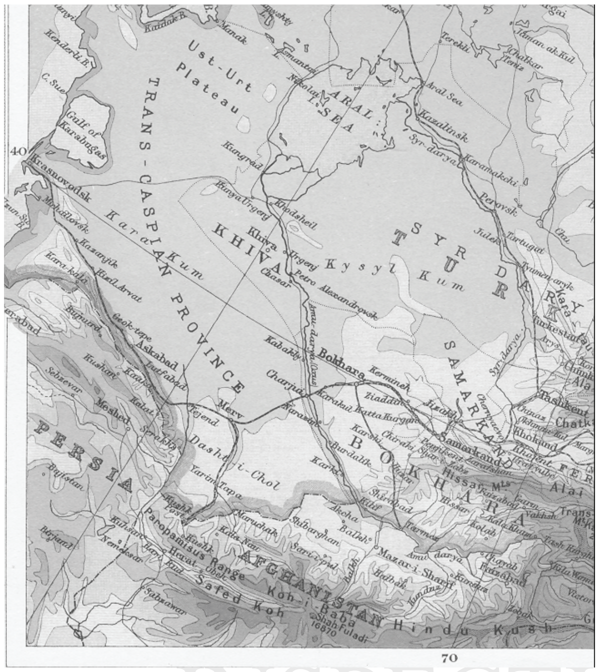
In this case the procedure is known as a *binary chop* algorithm. The reader might be surprised to know that many computers perform accurate division using an algorithm that involves a combination of guessing the answer and then seeking better and better approximations, rather than applying the kinds of rules for long division we learnt at school. I describe such procedures in more detail in **Part II**.

Our understandable fear of numbers and mathematics can be diminished (or maybe even cured) if we are aware of some important facts that we are not normally told. The first is that we can understand and solve many problems by experimentation, in particular by numerical experimentation - getting one's hands dirty, actually playing with numbers, patterns, diagrams and now, using computers, in order to see what happens and get a feel for the how? and the why? The second key fact is to appreciate that it has taken thousands of years of work and the brightest minds in history to get to where we are today, and there have been long periods (hundreds of years) when little or no progress was made at all. So we shouldn't be too despondent if we find even quite 'simple' problems difficult – we need to be patient - listening, experimenting, and learning, and then applying and testing our procedures. If the testing shows our answer is wrong, we need to go back and identify why, and then try to resolve the problem. Doing mathematics is a bit like sitting on a wall: everything on one side - the things you have already learnt or discovered - seem trivial; whilst everything on the other side, as yet undiscovered, seem unimaginably difficult!

INSPECTION COPY

Box 2. Mohammed ibn Musa al-Khowârizmî (c.780-850AD)

“Mohammed, son of Moses, from Khowârizmî”. Khowârizmî is now known as the province and town of Khiva in Uzbekistan, north of modern day Iraq



The Arab astronomer al-Khowârizmî (spelt in various ways in the Latin alphabet) wrote a book in c. 830 AD entitled ‘Hisâb Al-jabr w’al muqabâla’ which roughly translates as ‘the science of restoring [the balance in an equation] and simplification’. Our word Algebra comes from Europeanisation of the word Al-jabr in the title of this book when it was translated into Latin in the 12th century. The word Al-jabr also meant bone-setting, and even in the early 20th century barbers signs in Spain advertised ‘Algebrista y Sangrador’, or ‘bone-setting and blood-letting’. Our word Algorithm comes from the author’s name (actually where he came from) transliterated into Latin as al-Goritmi or algoritmi and thence algorithm. Initially this meant to ‘compute with positional notation’, but this developed over time into its modern day usage.

INSPECTION COPY

The last point to make is that most professional and academic scientists and mathematicians find many areas of mathematics really hard to understand, if not impossible. This is for several reasons. The first is that the term ‘mathematics’ is more like the term ‘languages’ than say any specific language such as ‘French’ or ‘Latin’. Mathematics involves use of a (very large) set of distinct and often little known languages, accepting however that most have common foundations. Just as no-one can speak or read every language in the world, so no one person can read or understand all fields of mathematics – the languages (notations, rules of grammar, etc.) are frequently so different that even the best mathematicians can only master a few. Some people are definitely better at mathematics than others, and some are even brilliant at it, but this is not so different from skill as a musician or composer, or in creating fantastic sculpture, or conversing fluently in many foreign tongues.

In addition to this multi-lingual view of the subject there is a big problem of what we mean by the word *truth*. Earlier I said that it was only natural to fear ridicule if you get the answer wrong, because in this field you are either right or wrong. But this is not actually the case – or at least, not always. It turns out that for some problems we simply cannot determine whether a particular answer or proposition is true or false; in other instances we may know that a ‘true’ or ‘best’ answer exists but is unachievable; and in very many cases there is either no known answer, multiple possible answers (possibly infinitely many), answers that make no apparent sense, or no clear agreement on what the answer should be. We should take comfort from these observations, because it shows that common sense and experience, based on results and usefulness rather than some idea of absolute truth, is often (but not always) a good way forwards.

So now we know that one of the keys to working with numbers is experimentation and testing, and that often there is not just a single, ‘true’ answer to a question, even if it appears to be relatively straightforward. We also know that for centuries, notably since the time of Descartes and Newton in the 17th century, people have been experimenting, learning from the results, and writing down rules and procedures that work – providing the building blocks from which we can continue construction without recourse to re-doing all of their work. Once this methodology was discovered and written down (and printed) in a form that was clear to a wide range of readers, progress was very rapid. This applies for almost every subject, especially those that have a scientific bias.

It is also true that occasionally some of the intermediate building blocks or even foundation stones turn out to be faulty. As a result, these may have to be discarded and everything constructed upon them questioned and possibly rejected. Alternatively we may be able to take a pragmatic view and accept that despite their failings, the results are so useful that we can ignore their weaknesses for most if not all problems. So called ‘pure mathematicians’ dislike this situation, seeking clarity and certainty in every aspect of their subject, whereas ‘applied mathematicians’ tend to be more accepting of the weaknesses, as long as the job in hand gets done and always works in practice. By ‘always works’ is meant that the answers are good enough for the problem at hand, or ‘fit for purpose’, and have never or very rarely been found to go wrong in practice. Newton’s laws of motion are a good example, which remain incredibly useful despite fundamental changes to the underlying logic being introduced by Einstein 200 years later. Rather like baking a cake, it also usually means that starting with the same set of values (ingredients) and the same procedures (preparation and cooking method), the same or very nearly the same answer (cake) will be produced – even if someone else does the cooking! Finally, as if to knock the pure mathematicians off their pedestal, it was shown in the 1930s that there are some ‘cakes’ that should be possible to bake given a set of ingredients and procedures, but turn out to be impossible to cook.

1.2 Statistics

> *“If there is a 50-50 chance that something can go wrong, then 9 times out of 10 it will”, Anon*

So we have started to address the fear of numbers and mathematics, but what about statistics? People often admire mathematicians but almost all dislike or distrust statistics and thereby, statisticians. Actually there are two quite separate aspects to this understandable response: the first relates to the use of selected numbers or “statistics”, often by politicians or the media that claim to “explain” some fact or justify a particular point of view. We are right to be deeply suspicious about almost all such information, however presented, as we shall see in some examples below.

The second meaning of the term statistics relates to the branch of mathematics that collects, analyses, manipulates and reports on data – i.e. information that has been collected in some way from the world around us. In this context ‘statistics’ is simply another language in the broad collection that is mathematics. And like many such languages it has its own rules, notation and history. What makes it different, and perhaps explains the dislike of it amongst many students who leave school and go through higher education on the road to a career in a scientific, medical or technical (SMT) discipline, is that they are obliged to study statistics as part of their degree courses. So why does statistics cause so much concern and distrust? Essentially, just as specific numbers and groups of numbers are the stuff of mathematics, so a selection of numbers from a larger set (sampling) is the basis of statistics. The process of selection, the definition of what comprises the larger set and the subsequent description of these sets are the factors that cause most of the problems. And once a selection has been made, there is endless scope for confusion and misinformation! For example, look at the employment statistics shown in Table 1-1.

Table 1-1 Regional employment data – grouping affects

	Employed (000s)	Unemployed (000s)	Total (000s) (Unemployed %)
Area A			
European	81	9	90 (10%)
Asian	9	1	10 (10%)
Total	90	10	100 (10%)
Area B			
European	40	10	50 (20%)
Asian	40	10	50 (20%)
Total	80	20	100 (20%)
Areas A and B			
European	121	19	140 (13.6%)
Asian	49	11	60 (18.3%)
Total	170	30	200 (15%)

Areas A and B both contain a total of 100,000 people who are classified as either employed or not. In area A 10% of both Europeans and Asians are unemployed (i.e. equal proportions), and likewise in Area B we have equal proportions (this time 20% unemployed). So we expect that combining areas A and B will give us 200,000 people, with an equal proportion of Europeans and Asians unemployed (we would guess this to be 15%), but it is not the case – 13.6% of Europeans and 18.3% of Asians are seen to be unemployed! The reason for this unexpected result is that in Area A there are many more Europeans than Asians, so we are working from different total populations. With statistics you have to know exactly how the information has been constructed in order to make any sense of it – so-called "headline statistics" are at best of little value and at worst, totally misleading.

The same logic applies to graphs - line diagrams that show values (such as average income per person) changing against time for example. To be meaningful a statistical graph should tell you:

(i) what the scales are

(ii) whether it starts at zero or some other value, and

(iii) how it was calculated, in particular exactly what data set and time period it is based upon.

Without all of these elements the information presented should be viewed with caution.

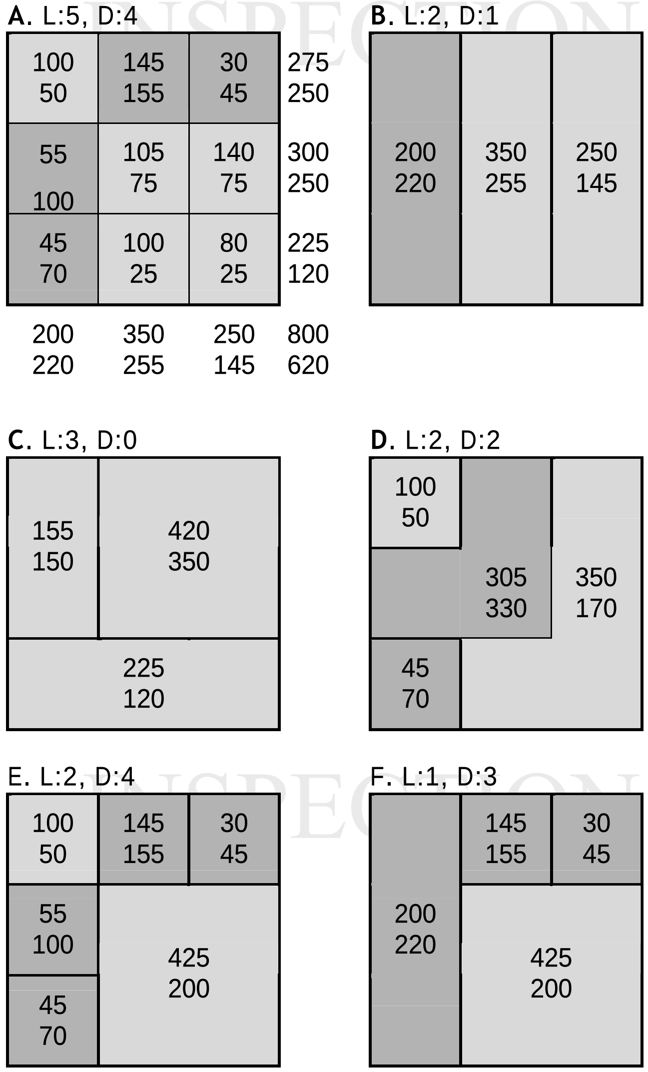
There is a further, less well known problem, which has particular importance in the process of elections and census data collection. This is due to the way in which voting and census areas are defined. Their shape, and the way in which they are aggregated, affects the results and can even change which party is elected. Figure 1.1 illustrates this issue for an idealised region consisting of 9 small voting districts. The individual zone, row, column and overall total number of voters are shown in diagram A, with a total of 1420 voters of whom roughly 56% will vote for the Light Grey party (L) and 44% for the Dark Grey party (D). With 9 voting districts we expect roughly 5 to be won by the Light Greys and 4 by the Dark Greys, as is indeed the case in this example. However, if these zones are actually not the voting districts themselves, but combinations of the zones are used to define the voting areas, then the results may be quite different. As diagrams B to F show, with a voting system of "first past the post" (majority in a voting district wins the district) then we could have a result in which every district was won by the Light Greys, to one in which 75% of the districts were won by the Dark Greys.

So it is not just the *process* of grouping that generates confusing results, but also the *pattern* of grouping. We are rarely informed of the latter problem, although it is one that is of great interest to those responsible for defining and revising electoral district boundaries.

This is not just a problem confined to voting patterns. For example, if the information being gathered relates to the proportions of trace metals (for example lead and zinc) in the soil, similar issues arise. Samples based on different field boundaries would show that in some arrangements the proportion of lead exceeded that of zinc, whilst other arrangements would show the opposite results. Furthermore, suppose that fields are classified as 'contaminated' if the levels of zinc or lead are on average >100ppm (parts per million), then we would find the following results – A:4/9 (44%); B:1/3 (33%); C:1/3 (33%); D:1/4 (25%); E:2/6 (33%); and F:2/4 (50%). In this example the values in the various fields have been divided by the number of square zones (e.g. hectares) they are constructed from, in order to obtain an estimated average value per field. Thus in Case B (lead), the values in each column have been divided by 3 with only column 2 exceeding 100ppm.

The final problem relating to statistical data I wish to refer to is a very general one, but again an issue that is little understood. This is the problem that once sets of numbers have been added together and totals or averages given, you can no longer safely use this information to make specific inferences about the original numbers (and vice versa). For example, suppose we have two adjacent areas, A and B, each containing 100 people in employment. Suppose now that 100 people in area A each earn 1000 Euros/week and those in area B each earn 500 Euros/week. Then combining the information we have 200 people in a slightly larger area, and we might be tempted to say that people on average earn 750 Euros/week. But as we know, nobody in areas A or B earns 750 Euros/week. It is acceptable to say that the average earnings in the combined area are 750 Euros/week, but it is not correct to say that individuals in this area earn this amount. This is quite a subtle point and is known as the *ecological fallacy*. Recent examples of this kind of fallacy include: the observation that breast cancer rates are higher in countries that have a high fat content in their diet, and then suggesting that women who eat more fat in their diet are more likely to suffer from breast cancer; or that crime rates are higher in areas of high unemployment, and then stating that it is the unemployed who are responsible for most crimes. The inferences drawn *may* be valid, and such observations can provide very useful pointers for research, but the data only provides very tenuous support for the claims made.

Figure 1.1 Grouping data - Zone arrangement effects on voting results



INSPECTION COPY

There is a more obvious twin (or dual) of this problem, known as the *atomist fallacy*. In this case we might find that some people (individuals, 'atoms') earn 500 Euros/week and then suggest that this is true generally, i.e. that (almost) everyone earns this amount, which we also know is incorrect. Most of us are familiar with such claims from the popular press and occasionally from more reputable sources.

Issues such as these are amongst the main reasons that the techniques and models used in statistics are found by many students to be particularly difficult. Because of the many pitfalls, statistical techniques have been devised to summarise information, identify patterns and describe relationships based upon well-defined rules and assumptions. As a result these rules and models can be quite complex and if the assumptions are not met, the techniques and results are either not valid or have limited use.

So, as you knew all along, statistics is a tricky subject, full of pitfalls and requiring great care. Hopefully the comments I have made in this Section go some way to explaining why many of these problems occur, helping you to know what to look for in future.

INSPECTION COPY

2 Core components

INSPECTION COPY

"'I only took the regular course'. 'What was that?' inquired Alice. 'Reeling and Writhing, of course, to begin with', the Mock Turtle replied; 'and then the different branches of Arithmetic – Ambition, Distraction, Uglification, and Derision'." from The Mock Turtle's Story, Alice's Adventures in Wonderland, Lewis Carroll, 1865

2.1 Notation and punctuation

One of the biggest problems when working with numbers relates to the way numbers, and expressions that relate to numbers (such as equations), are written down. 'Reeling and Writhing' often seems an appropriate description of the complexities and confusions that may arise. Getting it wrong can easily result in 'Derision'!

With the widespread use of computers in the home and at work, coupled with the growing internationalism (globalisation) of communications, it is becoming essential to understand how and why numerical information is represented in particular ways. Obtaining a better understanding of these questions provides us with an appreciation of the similarities and differences in such matters around the world, and reduces our reticence to work with and explore numerical problems. Adopting a consistent, formal system of writing for numbers and operations on numbers has taken a very long time – thousands of years – and even today there are ongoing debates as to how numbers and equations should be written, stored on computers and displayed on screen and paper. In fact the answer in many cases is to accept the differences and to facilitate multiple forms of information presentation for input and output purposes, whilst ensuring a consistent and 'complete' internal representation of the information is retained within the computer.

Most countries now use a decimal or base10 number system, within its origin in our 10 fingered hands (we have 10 *digits*). This does not prevent us from working with other bases, such as 2 (binary, using the symbols 1 and 0), 5 (one hand at a time), 16 (hexadecimal, using the symbols 0…9 plus A B C D E F) or 60 (sexigesimal or Babylonian, useful in time and angle measurement). But these are not usually the written form used for numbers. Even within the decimal system the representation of numbers may not necessarily use the familiar (Arabic) digits 1,2…,9,0. For example, most Arab countries use a slightly different system of notation, known as Indic, in place of or in addition to the so-called Arabic numerals:

١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩ ٠

It is interesting to see that in this system the digit 5 is represented by a symbol that looks like 0, and the symbol for 0 is simply a raised form of dot. And there remains scope for confusion when familiar number symbols are hand-written, as is common between 2 and Z, 1 and 7 and I, 0 and O. If letters and numbers are assigned unique codes, however, such confusion *should* be avoidable.

Perhaps the first attempt to define a set of 'digital' codes to represent numbers and letters was made by the mathematician C F Gauss and his colleague W Weber, from the Physics Department at Göttingen University in Germany, in 1833. Their code table is shown in Figure 2.1, where R means Right and L means Left, indicating the direction a magnetic needle turned when an electric current was enabled on a long wire between two buildings. This coding system provides 32 values (2x2x2x2x2 or 2^5, i.e. 5 *bits*). Some symbols having dual meanings (G/J) and X is omitted. With 26 letters and 10 digits and a Space character this would have required 37 values, which would have involved an extra *bit* in the code, so some simplifications were made. Note that this pre-dates Morse Code or the first telex machines.

INSPECTION COPY

Figure 2.1 Gauss and Weber telegraph code

RRRRR	**A**	RLLLL	**1**
RRRRL	**B**	RRLLR	**2**
RRRLR	**C**	RLRLL	**3**
RRLRR	**D**	RLLRL	**4**
RLRLR	**E**	LLLRR	**5**
LRRRR	**F**	RLLRR	**6**
LRLRR	**G/J**	LLLRL	**7**
RLRRL	**H**	LLRRL	**8**
LLRLL	**I/Y**	LRRLR	**9**
LRRRL	**K**	LRLLR	**0**
RLRRR	**L**		
RRLLL	**M**		
LLLLL	**N**		
LRLLL	**O**		
LRLRL	**P**		
LLRRR	**Q**		
RRRLL	**R**		
RRLRL	**S/Z**		
LLRLR	**T**		
RLLLR	**U**		
LRRLL	**V**		
LLLLR	**W**		

In 1870 J M E Baudot patented a new coding scheme that sought to overcome the limited set of characters and functions available with 5-bit schemes. He introduced the idea using 2 of the 32 available codes to indicate that all subsequent codes were either: (a) Letters only; or (b) Numbers, symbols and functions only. The sequences he chose were equivalent to RRRRR and RRLRR in Gauss-Weber coding or 11111 and 11011 in binary, and are sometimes known as SHIFT sequences (based on comparison with early mechanical typewriters). In this way he extended the Gauss-Weber code to provide up to 62 *(*=64–2*)* usable codes. So, for example, if we call 11111 (SI) and 11011 (SO) we could code the text "The number 52 is a secret!" using something like:

(SO)THE NUMBER (SI)52(SO) IS A SECRET (SI)!

Notice that only uppercase (Capitals) OR lowercase was possible, not both. As well as supporting new characters, such as ! and ?, new features were provided by Baudot's code, including Line Feed (LF, coded as 00010) and Carriage Return (CR, coded as 01000). These enabled remote printers to layout the information being transmitted as a series of lines.

This form of coding continued in use for many years, but it too required extension, ideally by a more powerful coding scheme. In order to provide a character set, symbols, case selection and other facilities required by computers and word processors, the coding scheme known as ASCII (American Standard Code for Information Interchange) was developed. It is one of a family of digital coding systems in use today (see ASCII table, Table 2-1). ASCII dates from 1963 when it was first introduced as a 7-bit code, and then from 1968 as the 8-bit version we now use. There are around 180 national variants of ASCII registered with the International Standards Organisation (ISO).

In order to represent complex expressions or unusual mathematical symbols with a limited set of codes, quite sophisticated rules are required to define how combinations of these basic codes are to be interpreted. As we shall see in **Part II**, special 'languages' and toolsets such as TeX and MathML exist to address this problem, for example using the bracketed sequence of ASCII characters {\sum} to mean use the symbol $\sum$.

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY

Table 2-1 ASCII Codeset - 8-bit

Decimal	Binary	Character	Decimal	Binary	Character
032	00100000	SPACE	080	01010000	P
033	00100001	!	081	01010001	Q
034	00100010		082	01010010	R
035	00100011	#	083	01010011	S
036	00100100	$	084	01010100	T
037	00100101	%	085	01010101	U
038	00100110	&	086	01010110	V
039	00100111	'	087	01010111	W
040	00101000	(	088	01011000	X
041	00101001	)	089	01011001	Y
042	00101010	*	090	01011010	Z
043	00101011	+	091	01011011	[
044	00101100	,	092	01011100	\
045	00101101	-	093	01011101	]
046	00101110	.	094	01011110	^
047	00101111	/	095	01011111	_
048	00110000	0	096	01100000	`
049	00110001	1	097	01100001	a
050	00110010	2	098	01100010	b
051	00110011	3	099	01100011	c
052	00110100	4	100	01100100	d
053	00110101	5	101	01100101	e
054	00110110	6	102	01100110	f
055	00110111	7	103	01100111	g
056	00111000	8	104	01101000	h
057	00111001	9	105	01101001	i
058	00111010	:	106	01101010	j
059	00111011	;	107	01101011	k
060	00111100	<	108	01101100	l
061	00111101	=	109	01101101	m
062	00111110	>	110	01101110	n
063	00111111	?	111	01101111	o
064	01000000	@	112	01110000	p
065	01000001	A	113	01110001	q
066	01000010	B	114	01110010	r
067	01000011	C	115	01110011	s
068	01000100	D	116	01110100	t
069	01000101	E	117	01110101	u
070	01000110	F	118	01110110	v
071	01000111	G	119	01110111	w
072	01001000	H	120	01111000	x
073	01001001	I	121	01111001	y
074	01001010	J	122	01111010	z
075	01001011	K	123	01111011	{
076	01001100	L	124	01111100	|
077	01001101	M	125	01111101	}
078	01001110	N	126	01111110	~
079	01001111	O	127	01111111	DELETE

INSPECTION COPY

2.2 Number systems

> "…don't panic. Base eight is just like base ten really - if you're missing two fingers." Tom Lehrer's song 'New Math'

INSPECTION COPY

Individual numbers are typically written from left to right, even in cultures where the written form goes from right to left or up and down. The number order matters, with the leftmost digit being that with the largest value. In decimal systems, each position is 10 times the size of the position one to its right whilst in binary systems each number is twice the value of its rightmost neighbour. Thus in decimal notation the number 183 means 1*100 plus 8*10 plus 3*1 (where the symbol * means 'times' or 'multiplied by'). This is an example of a positional number system and dates from around 600AD in India and less than a century later in the Arab world, from which it was subsequently communicated to Mediterranean and northern European cultures. For special purposes collections of numbers are sometimes written in blocks, which I discuss in more detail a bit later on (Section 2.4).

The symbols we use for writing or displaying numbers are generally called "glyphs", and vary across the world, much as written languages vary in the alphabets or composite graphics used (such as hieroglyphs or Chinese characters). In Ancient Egypt, although they developed base10 number systems, these were non-positional systems with different hieroglyphs for separate powers of 10 and symbols repeated in blocks rather than on a line for multiples less than 10.

Zero was introduced as a placeholder initially. If a number is written in positional form and there is no placeholder, when a column is missing it causes confusion. For example, the number 305 could be written as 3 5 if we had no placeholder for the 10s, but is this really 305, or maybe 35 or 3005 written without sufficient care? The number system of the Babylonians suffered from this problem. The solution, dating from around the 2nd or 3rd century BC in India and from the 7th century AD in the Arab world, was to add a symbol, at first a dot and then a small circle, ∘, to identify the empty column. Subsequently it was found useful to have a symbol for zero which could be used in algebraic operations, such as $x+0=x$, $y–z=0$, q times 0=0. The Italian mathematician Leonardo of Pisa, also known as Fibonacci (see Box 3), is regarded as being responsible for introducing the use of zero in Europe, in 1202 AD. The words zero and cipher both derive from the Arabic word *as-sifr* (meaning *empty*), which Fibonacci transliterated as *zephirum*. The equals symbol, =, was introduced much later, in 1557, by Welshman Robert Recorde.

In modern Western decimal (Indo-Arabic) notation the number 111 means 1*100 plus 1*10 plus 1*1 (or $1*10^2+1*10^1+1*10^0$), whereas in binary (base2) this number means 1*4 plus 1*2 plus 1*1 (or $1*2^2+1*2^1+1*2^0$) i.e. 7 in decimal notation. The same rules apply for any other base, such as base8 (octal), base16 (hex), or base60. Numbers less than 1 are treated in the same way, but with a separator (generally displayed as a full stop or 'period') to indicate the break in values. For example, 0.111 in base10 arithmetic means

$$1*\frac{1}{10}+1*\frac{1}{100}+1*\frac{1}{1000}$$

whereas in binary it equates to the decimal fraction 7/8, because it means

$$1*\frac{1}{2}+1*\frac{1}{4}+1*\frac{1}{8}$$

Whilst computers generally work in binary, for simplicity and speed, input and output is handled in the form we are most familiar with or explicitly choose, and the computer simply converts this to and fro without us having to worry about the processes involved.

But what about 1/3? Written in decimal form this is 0.3333… the three dots indicating that there are an unlimited number of similar digits to the right, and the same is true for binary, where it is 0.010101010…, again a repeating number. In base3, however, 1/3 is simply 0.10000… because the first digit after the point represents precisely one third. So the choice of base affects the way in which numbers are stored, and how many digits are needed to correctly represent the information. However, for two main reasons this observation does not help us. The first reason is that for most practical purposes we have to choose one format or base for storing our data, and this tends to be binary. It is impractical to store information in many different formats and switch between them at will or in some kind of super-intelligent manner. The second reason is that some numbers cannot be represented by a finite number of digits, no matter what base is chosen, i.e. they shown an ever changing and everlasting pattern of digits (or lack of pattern) no matter how we describe or store them. Numbers of this type are discussed further in Section 5.3. Readers will realise that a problem with computers is that once a number such as 1/3 is stored as a very long binary sequence it is almost impossible to reverse the process, i.e. recognising that it is really 1/3. It is also clear that there is a limit to the number of digits that computers can store, so they will almost always store some numbers inaccurately.

Box 3. Fibonacci, or Leonardo of Pisa (c.1170 or 1175 to 1240 or 1250)

Fibonacci (*filius* i.e. *son* of Bonacci) was born in Italy but was educated in North Africa where his father was posted. He travelled widely with his father and recognised the enormous advantages of the mathematical systems used in the countries they visited. The head of the statue of Fibonacci shown here was completed in 1863 and is sited near the Leaning Tower in Pisa, Northern Italy. The image is not based on any portraits or statues of Fibonacci, since none are known to exist.

Fibonacci ended his travels around the year 1200 and at that time he returned to Pisa. There he wrote a number of texts that played an important role in reviving ancient mathematical skills and he made significant contributions of his own. Fibonacci lived in the days before printing, so his books were hand written and the only way to have a copy of one of his books was to have another hand-written copy made. Of his books we still have copies of Liber abaci (1202), Practica geometriae (1220), Flos (1225), and Liber quadratorum. Given that relatively few hand-made copies would ever have been produced, we are fortunate to have access to his writing in these works. However, we know that he wrote some other texts that unfortunately are lost. The book, Liber abaci, introduced the Hindu-Arabic place-valued decimal system and the use of Arabic numerals into Europe. Indeed, although mainly a book about the use of Arab numerals (which became known as *algorism*, see Box 2), simultaneous linear equations are also studied in this work. Certainly many of the problems that Fibonacci considers in Liber abaci were similar to those appearing in Arab sources.

A problem in the third section of Liber abaci led to the introduction of the Fibonacci numbers and the Fibonacci sequence for which Fibonacci is best remembered today: "A certain man put a pair of rabbits in a place surrounded on all sides by a wall. How many pairs of rabbits can be produced from that pair in a year if it is supposed that every month each pair begets a new pair which from the second month on becomes productive?" The total number of pairs of rabbits at the end of each month is given by the sequence 1, 1, 2, 3, 5, 8, 13, 21, 34, 55,… (Fibonacci omitted the first term in Liber abaci). This sequence, in which each number is the sum of the two preceding numbers, has proved extremely fruitful and appears in many different areas of mathematics and science (see further, Section 15.3).

The widespread convention is that positional number systems operate horizontally, from left to right. This can present problems for languages that are written from right to left. For example, if you are typing on a computer in Hebrew and start to enter a number, the software will normally switch the direction or mode of screen display, typically 'pushing' the numbers from right to left whilst leaving the active position or flashing *cursor* at the right hand end of the number. When you start typing text again the cursor will leap back over the number typed to the left hand end of the text. The left to right convention is not used in a consistent manner for numeric dates, such as 12/09/2004 (meaning the 12^{th} of September 2004 in Europe, but the 9^{th} of December 2004 in North America) whereas it is applied for times, 3:30 PM or 15:30 hrs. It would make more sense if dates were always written in YYYY/MM/DD order, e.g. 2004/09/12, and this would certainly help numeric sorting of such date strings, but this arrangement is less common (see further, Section 4 which deals with Times and Dates).

Similar issues arise with negative numbers and expressions. Negative numbers are usually displayed with either a preceding minus sign (–) or may be shown in brackets or a different colour (e.g. red) for financial applications. In some cultures the order of operations, including those involving subtractions, retains the language order. Thus in Arabic, A–B typically means subtract A from B rather than B from A, although this would normally only apply when the relation is written or typed using Arabic characters. Likewise the expression *w=s–n* would appear in Arabic in the reverse order (i.e. right to left) as:

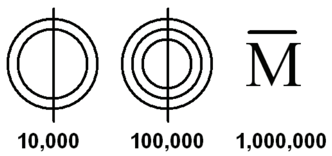
و = س - ن

Within computers the need to store a minus symbol for most numbers (so-called *signed* numbers as opposed to *un-signed* numbers) often results in an extra position being required in memory or on disk, the "sign bit". This requirement reduces the maximum size of number that a computer can store directly - since each bit (BInary digiT) provides a multiple of 2, the effect of having signed numbers is to halve the size of the maximum stored number. Such issues are discussed in more detail in Section 2.8, which covers Precision and Accuracy.

Modern number systems are positional and as we have seen, both the order and position of the numbers has significance. This was not always the case. Before the Arabic-based positional system was widely adopted the notations used by Roman, Greek and the majority of earlier civilisations used ordered but non-positional systems. Complex numerical operations using this arrangement would be extremely time consuming and prone to error. In practice, these cultures did not use their written number systems for most of their direct calculations or mathematical analyses. Instead they used words and drawings (geometric diagrams for example) for problem solving, and several versions of the abacus for arithmetic operations – for example using counting pebbles, or *calculi*, from whence we get the terms *calculate* and *calculus* (and, incidentally, calcium and calcify). Interestingly, if you are ever unfortunate enough to suffer from kidney stones or gall stones, which are both very painful conditions, the doctors still refer to these as *calculi* because they are often calciferous.

However, getting back to early number systems, addition is actually quite simple using Roman numerals and similar systems. For example the addition sum 585+59 in Roman numerals is DLXXXV + LIX and all we need to do is add up similar letters and write the result down, so we have DLLXXXXIV or the equivalent DCXLIV because L+L=C (where D=500, C=100, L=50, X=10, V=5, and I=1) and XL can be used in place of XXXX. The largest number that was written in Roman numerals using a single letter was M=1000, but still larger numbers were needed. A variety of special graphical symbols were adopted for 10,000 and 100,000, and in some cases lines drawn above a number would indicate multiplication by 1,000. Examples of these conventions are illustrated in Figure 2.2:

Figure 2.2 Representation of large numbers in the Roman system



Roman characters (the main Western alphabet) are used in expressions and equations in many ways, but the widespread convention is that characters from the end of the alphabet, such as *x*, *y* and *z* are used for variables - objects that may represent a variety of specific values or numbers – whilst characters from the start of the alphabet, *a*, *b*, *c* etc. are used for known or unknown constants, such as 3 or 2.4. This convention is due to the French mathematician and philosopher, René Descartes (see further, Box 4) who wrote down these guidelines and many others that we continue to use today, in the early 17th century.

Greek characters are also widely used in equations, however principally in the West. Their function is to provide an additional set of symbols that are distinct from the Roman alphabet. Typically lower case Greek characters used for variables and for selected *constants* (objects whose values are fixed and pre-defined). Upper case Greek characters are used widely for *operators* (objects that describe how a collection of variables are to be handled).

Examples of the use of Greek lower case characters are:

> α (alpha), β (beta), δ and ε (delta and epsilon, often used to indicate very small amounts, i.e. numbers that are very close to 0), γ (gamma, frequently used as a specific Constant), μ (mu, the symbol associated with the population average or mean in statistical parlance), π

INSPECTION COPY

(pi, perhaps the most famous Constant), and θ and ϕ (theta and phi, the symbols often used to indicate angular variables).

Eastern societies, such as Japan, China and Korea typically use local symbols or character sets in such cases, rather than Greek, depending often on whether their audience is domestic or international.

Examples of operators that use Greek upper case characters are:

Δ (Delta, widely used to indicate a Difference operation, i.e. to calculate the difference between two values); Γ (Gamma, generally used as an abbreviation for a special function whose values are related to numbers like 5*4*3*2*1, also denoted 5! or 5 *factorial* - Γ is used when calculating *n*! where *n* is a fraction); Π (Capital Pi, used to indicate the Product operation, i.e. where a series of numbers or expressions should be *multiplied* together); and Σ (Sigma, the summation operator we came across earlier, used when a series of numbers or expressions should be *added* together).

Box 4. René Descartes (1596-1650)

Descartes was educated at the Jesuit college of La Flèche in Anjou. He entered the college at the age of eight years, just a few months after the opening of the college in January 1604. In 1618 he started studying mathematics and mechanics and began to seek a unified science of nature. After two years in Holland he travelled through Europe. He spent time in 1623 in Paris where he made contact with Mersenne (see Box 7), who kept him in touch with the scientific world for many years. By 1628 he settled in Holland and began work on his first major treatise on physics. This work was near completion when news reached him that Galileo had been condemned to house arrest. Descartes decided not to risk publication at that time, but he was pressed by his friends to publish some of his ideas and he duly wrote a treatise on science which was published in 1637 – an Appendix, *La Géométrie* ("Geometry") is by far the most important part of this work

2.2.1 Cardinal and Ordinal numbers

"Chapters in books are usually given the Cardinal numbers 1,2,3,4,5,6 and so on. But I have decided to give my chapters prime numbers 2,3,5,7,11,13 and so on because I like prime numbers." Christopher Boone, an Asperger's Syndrome boy, aged 15, in Chapter 19 (i.e. 8) in "The curious incident of the dog in the night-time", by Mark Haddon, 2003. Asperger's Syndrome is a form of autism.

The name Cardinal number is usually reserved for numbers that are used for counting. Whilst Arabic numerals are the most widely used in the Western world for dealing with *quantities* - counting and the display of numerical procedures - other forms are used for ordered sequences or series of numbers. Frequently Roman numerals I, II, III, IV, V…, X, etc. or the lower case versions i, ii, iii, iv… are used to indicate an ordered sequence and in this context are called Ordinal Numbers. Western clocks frequently use Roman numerals for displaying the time, but interestingly they normally show IV as IIII – this is believed to be for design reasons (visually balancing the VIII) rather than having any technical or historical basis. Other letters were used by the Romans for larger numbers, such as 50, 100 or 1000 as noted earlier, but there is no Roman symbol for zero or for values less than zero. Likewise, the Hebrew tradition uses letters from their alphabet as numbers in an Ordinal context - for example, for chapter numbers and page numbers in books and for writing the Hebrew calendar. With purely Ordinal numbers the notions of "greater than", >, and less than, <, and equality, =, may make sense (e.g. Chapter V is great than/after Chapter IV), but numerical operations like addition or division in general are not meaningful.

INSPECTION COPY

2.3 Symbols

Symbols such as +, −, ×, ÷, = are fairly universal and have consistent meaning, but there is an enormous range of other symbols and combinations of symbols that are used with varying degrees of consistency. To be certain of their meaning one often has to check the context in which they are being used. Even these symbols have had a varied history. For example the symbol now used for division, ÷, was formerly used for subtraction, and even today division is written in many different forms - the last of the forms shown below, with the line separating the top (*numerator*) and bottom (*denominator*), was introduced by Fibonacci:

$$a \div b,\ a/b,\ a{:}b,\ \frac{a}{b}$$

The symbol /, known as a *solidus*, is used in a variety of contexts: in text it often means 'or' as in "sales tax/VAT" used earlier; or 'per' as in "metres/sec" and, as we shall see later (Section 3.5) in the definition of gravitational acceleration (metres/sec^2). These last two examples are both forms of division, in each case being in terms of units of measure rather than specific numbers. And of course, to complicate matters, / is used as a field separator in web addresses and computer file directory specifications.

The multiplication symbol is sometimes written as a raised dot, $a \cdot b$ or an asterisk, $a*b$ (often in computing). The equals sign, =, which derives from using two lines of the same length, is the source of considerable problems, since $a=b$ can mean that this is an equation, which we may wish to retain or manipulate, or that *a* should be replaced by *b* (i.e. an assignment, *a* is *b*). This distinction is important in determining what happens next, whether we are working on paper or using a computer. Often separate symbols are used for equality and assignment, to avoid this confusion. Note that whilst for many purposes the order or precedence of operations is left to right, assignment is often (but not always) a right to left operator.

INSPECTION COPY

Examples of other symbolic notation used widely and with broadly consistent meaning include those shown in Figure 2.3. There are a range of national and international standards providing guidance on symbol usage, including the International Standards Organisation (ISO) standard ISO6862:1996 and the USA ANSI standard ANSI Y10.20.

Figure 2.3 Selected symbols

Symbol type	Examples	Interpretation
Relational symbols	$\leq \geq \cong \equiv$	less than or equal to; greater than or equal to; approximately equal to (also ≈); equivalent to/congruent to
Set theory symbols	$\cup \cap \subseteq \notin$	union; intersection; partial or total subset; not a member of
Logic symbols	$\therefore \forall \because \exists$	therefore; for all members (upside down A); because; there exists (backwards facing E)
Fractions and roots	$\frac{a}{b}$ $^{a}\!/\!_{b}$ $\sqrt[b]{a}$	*a* divided by *b*; *a* divided by *b*; the *b*th root of *a* (e.g. the square root if *b*=2, or if *b* is omitted)

The last example shown above, that involving the *b*th root, illustrates a limited *vertical* positioning notation system that is also widely used, again due to Descartes. This notation involves the use of characters that are raised and smaller, called superscripts (above the written line), and their equivalent lowered symbols, called subscripts, as shown below:

$$A_c^b \text{ or } x_j^2$$

INSPECTION COPY

The convention most commonly used is that superscripts and subscripts are either lower case letters or numbers, and are placed immediately to the right of the object they refer to. Superscripts, often mean "raised to the power of…" or "raised to the exponent", and subscripts often mean "a member of…". In this context subscripts may be repeated or multiple, as in x_{ij}, but this usage is not consistent or universal. In right to left languages (and at the discretion of the writer) placement may be to the left of the object. The use of two or more subscripts is also frequently similar to the use of two or more variables in geometry to identify the location of a point along a line, on a plane, or in 3 dimensions. In the former case, however, it is used normally to indicate in which row, or row and column in a square or rectangular table or block of numbers the value is located. Thus x_{ij} means the value to be found in row *i*, column *j*. If there is only one row or column of numbers only one subscript is needed and the set $\{x_i\}$ is called a *vector*. Here the use of curly brackets { } is taken to mean that the included entries are a well-defined collection of items (a *set*), in this case x_1, x_2, x_3 etc. With two subscripts the set $\{x_{ij}\}$ is called a *matrix*, and with more subscripts the term matrix is still used. The subscripts i,j etc. in such cases only take positive whole number (Integer) values: 1,2,3… etc.

Another widely used convention for subscripts is that if a dot or blob is shown where you would have expected a letter, then you assume that all possible values that can be ascribed to the letter are included – i.e. this is a form of shorthand to avoid the excessive use of additional symbols such as summation signs. Thus $\{x_{i\bullet}\}$ typically means that the values have been added across the columns of a matrix, so this represents the vector of totals for each row; likewise, $\{x_{\bullet j}\}$ is the set of column totals and $\{x_{\bullet\bullet}\}$ is the grand total. Dots are also sometimes used directly above a symbol, with a variety of meanings (for example to indicate that number is a repeated decimal, as in:

$$0.3\dot{3}$$

In some instances, where a repeated decimal exhibits a repeating pattern of digits, as in 0.10891089… this is written with either two dots above (*overdots*), or a line above (a *vinculum*):

$$0.\dot{1}08\dot{9} \quad \text{or} \quad 0.\overline{1089}$$

If the object is an *operator* (i.e. a symbol that identifies a particular procedure or operation that is to be carried out) such as a Sum, Product or Integral symbol, then the sub- and super-script values indicate the lower and upper ranges to be used in the calculation. So, for example, the expression

$$\sum_{i=1}^{10} x_i$$

means find the sum of the x_i values for i=1,2,3,…,10, or $x_1+x_2+…+x_{10}$, and the expression

$$y_i = \sum_{j=1}^{4} w_{ij} x_{ij} \text{ for } i = 1,2,3$$

means find the 3 sums $y_1=w_{11}x_{11}+w_{12}x_{12}+w_{13}x_{13}+w_{14}x_{14}$, $y_2=w_{21}x_{21}+…$, $y_3=w_{31}x_{31}+…$, where in this case the *w*'s may be weights or constants and the *y*'s and *x*'s unknown values. So, for example, we might have

$$y_1=1x_{11}+5x_{12}+0x_{13}+3x_{14}$$
$$y_2=2x_{21}+0x_{22}+7x_{23}+1x_{24}$$
$$y_3=6x_{31}+1x_{32}+3x_{33}+5x_{34}$$

In this case we can also regard the sums as a simple equation relating the *y*'s to the *x*'s (*linear* equations, since all the *x*'s are to the power 1, there are no powers of 2, 3 etc.).

2.4 Blocks of numbers (matrices)

The notation in the last example is rather cumbersome and potentially confusing. One solution, which has many additional benefits and applications, is to focus on the weights or constants, and write these down as a block of numbers or symbols. In the example just given we have 3 rows of 4 values, which we can write in the form:

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \end{bmatrix} \text{ or } \begin{bmatrix} 1 & 5 & 0 & 3 \\ 2 & 0 & 7 & 1 \\ 6 & 1 & 3 & 5 \end{bmatrix}$$

This arrangement is called a *matrix*, in this case with 3 rows and 4 columns. The idea of working with blocks of numbers or expressions in this form dates from around 1855, so it is a relatively

INSPECTION COPY

recent innovation in mathematics, although the fundamental ideas date back more than 2000 years to early Chinese and Babylonian mathematics. If there is only 1 row or 1 column (or *dimension*) it tends to be called a *vector*. We generally only work with 1- or 2-dimensional matrices. If a square matrix (one with the same number of rows and columns) contains all zeros it is called the zero matrix (usually denoted **O**), and if it contains 1's along its diagonal (top left to bottom right) and zeros elsewhere, it is called a unit or Identity matrix (usually denoted **I** for identity). These two arrangements perform the functions of the numbers 0 and 1, but for matrix arithmetic. Capital letters, often in bold type, are used as an abbreviation for a matrix.

Over the last 150 years it has been found that working with matrices and vectors provides a very convenient and fast way of solving a large range of problems, from optimally cutting fabrics and carpets in a factory to modelling the workings of national economies, and from statistical data analysis to the solution of many problems in physics and astronomy. Part of the reason for their widespread usage is that they are very well-suited to a computational context, i.e. in the frameworks provided by modern computer systems and software.

I am not going to try and cover the ins and outs of matrices and matrix algebra here, but it is helpful to see some examples of their use, as these are so varied. The first example involves the analysis of chemical samples, whilst the second looks at finding the shortest route through a simplified road network (similar problems apply to telecommunications networks and electronic circuits). In **Part II** I take a look at their application in digital photography.

Our first example involves finding the solution to a question about grain harvests posed and solved by matrix-like methods in China around 300BC. I am going to re-express this as a modern-day pharmaceutical problem:

> *A laboratory is provided with 3 samples in powder form that are claimed to be a new headache remedies. The samples weigh 39, 34 and 26 grams respectively. Chemical analysis shows that the samples each contain a different mix of the same 3 ingredients: the first mix is in the proportions 3:2:1, the second is 2:3:1 and the last is 1:2:3. The lab is asked to determine the actual amounts of each ingredient that have been used to create these 3 samples.*

We can write this problem down using equations, calling the three unknown ingredient amounts x, y and z, as follows:

$$39 = 3x + 2y + z$$
$$34 = 2x + 3y + z$$
$$26 = x + 2y + 3z$$

In matrix notation this would be written as shown below, where the arrows indicate how the matrix multiplication process operates:

$$\begin{bmatrix} 39 \\ 34 \\ 26 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

In this layout we can see that we have an arrangement of the form ***b***=***Ax*** where ***b*** is a 3x1 column vector of numbers, ***A*** is a 3x3 matrix of numbers and ***x*** is a 3x1 column vector of unknowns [*x y z*] that we wish to determine. The rules of multiplication are obviously a bit different from those we are used to. The basic idea is that each *row* of ***A*** multiplies the *column* of ***x*** and the results are added together, element by element. Thus row 1 of ***A*** times column 1 of ***x*** (there is only 1 column in this example) gives 3x+2y+z. To do such multiplications the dimensions of the two matrices must be compatible, so that the number of columns in the first matrix must equal the number of rows in the second. In general ***A·B*** is not the same as ***B·A***, so the order of multiplication matters as well as the sizes of the matrices. In the first case (***A·B***) we are *pre*-multiplying by ***A***, in the second (***B·A***) *post*-multiplying by ***A***.

Now, the arrangement above is of the form ***b***=***Ax***, so it would be great if we could re-arrange this by 'dividing' both sides by ***A*** to give ***b/A***=***x*** thus solving our problem for ***x***, i.e. the 3 unknown elements *x*, *y* and *z*. In matrix algebra the equivalent to division is called *inversion* and the inverse of **A** is written A^{-1}. We can then pre-multiply both sides of ***b***=***Ax*** by A^{-1} to give:

INSPECTION COPY

$A^{-1}b = A^{-1}Ax$ or $A^{-1}b = x$ (since $A^{-1}A = I$ and $Ix = x$)

INSPECTION COPY

We do need to check that the calculation of A^{-1} is possible and does not give us the equivalent of dividing by zero, but here we will assume that the operation is safe! You can compute A^{-1} using Maple, MATLab or Excel (using its built-in function Minverse()). To save time here I will simply write the answer down, as given by Maple (i.e. as fractions rather than decimals):

$$A^{-1} = \begin{bmatrix} +\frac{7}{12} & -\frac{1}{3} & -\frac{1}{12} \\ -\frac{5}{12} & +\frac{2}{3} & -\frac{1}{12} \\ +\frac{1}{12} & -\frac{1}{3} & +\frac{5}{12} \end{bmatrix}$$

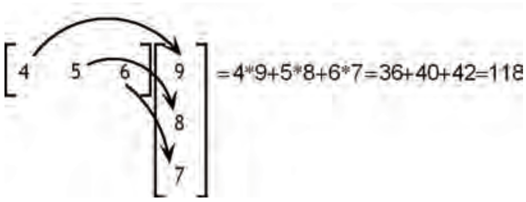
Now we can compute $A^{-1}b$ and we find x=37/4, y=17/4 and z=11/4. If you put these values back into the original equations you will see that these are the correct amounts of each ingredient in the samples supplied to the lab.

In general, if we have two matrices $A_{p,q}$ and $B_{q,r}$ where p, q and r are the sizes of the matrices (in rows, columns order) then it is valid to calculate $C=A*B$ and the result will be $C_{p,r}$. So the dimensions rule is that the inner dimensions must match: (p,q)x(q,r) gives a result of dimension (p,r). A simple example serves to illustrate the process, in this case a 2x2 matrix times a 2x1 matrix, giving a 2x1 matrix:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} x \begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} a*e+b*f \\ c*e+d*f \end{bmatrix}$$

If p=r=1 we have two matrices which consist of 1 row for A and 1 column for B (thus both are *vectors*). For example, in the multiplication below with q=3, we have the dimensions (1,3)x(3,1)=(1,1):

INSPECTION COPY



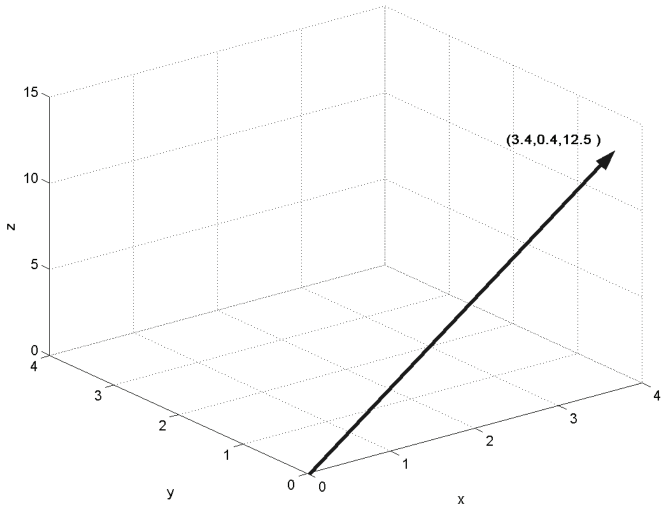
So when two *vectors* are multiplied in this way the result is a single value (a 1x1 matrix or *scalar*). This result will be used in **Part II** in our discussion of smarter ways to divide and multiply numbers. A row vector, such as v=[4,5,6] or v=[3.4,0.4,12.5] can also be seen as a point in 3-dimensional space (see Figure 2.4). This form of representation of space using a box-like arrangement of straight lines at right angles to each other (*axes*) is another convention introduced by Descartes, and for this reason it is known as *Cartesian*.

If we represent our 3 element vector, v, as an arrow extending from the point [0,0,0] to [3.4,0.4,12.5] then we often need to know the *magnitude* or size (length) of this vector. We can obtain this number very simply by evaluating the product $v*v^T$ where v^T is the vector v arranged as a column (this arrangement is called the *transform* of v, for which the superscript T is used). This gives us a single value that is the sum of the squares of the elements of v. In our 3-space example we find: $v*v^T = 3.4^2+0.4^2+12.5^2 = 167.97$ thus $|v| = \sqrt{167.97} = 12.96$, where the vertical lines in $|v|$ indicate that we are showing the size (length) of the vector. So we can think of a vector in an additional way, as an object like an arrow, with a direction and a magnitude. This is often useful for problems involving variables of this kind, as in modelling fluid flows and in the study of electromagnetic and gravitational phenomena.

INSPECTION COPY

Figure 2.4 3D representation of a vector

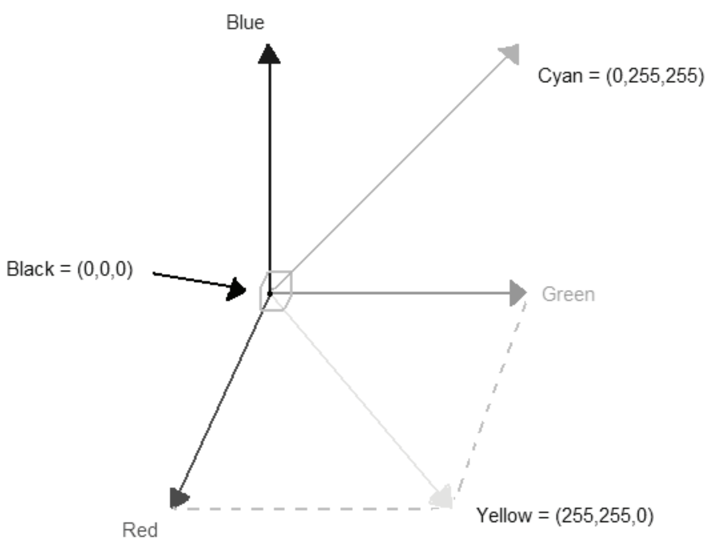
COPY



A simple example of the use of vectors is in the representation of the colours used in digital images. Typically, coloured digital images are stored as a set of values of picture elements, or *pixels*, in row and column (matrix) format. So an image that is 500x500 pixels would contain a number or set of numbers associated with every pixel identifying the colour of that pixel. The colour information is usually stored as three elements: Red, Green and Blue, each of which may have 256 values (2^8 or 8 bits, i.e. 1 byte) from 0 to 255. (0,0,0) is then Black, (255,0,0) is Red, (0,255,0) is Green, (0,0,255) is Blue, and (255,255,255) is White. All other colours are a mix of the three RGB components, giving a total of $(2^8)^3=2^{24}$ colours, or 16.8 million, all stored in 3 bytes per pixel. These arrangements can be conveniently represented as vectors in 3D space (or *colour space* in this case), as shown in Figure 2.5. In **Part II** I show how you can use these forms of image representation to manipulate digital images. For our 500x500 picture we need 750,000 (500x500x3) bytes for storage, assuming no compression. But with most images, compression using coding schemes such as JPEG or GIF are the norm, so far less space is required on digital photocards and computer disks.

INSPECTION COPY

Figure 2.5 Vector representation of colours in 3D space



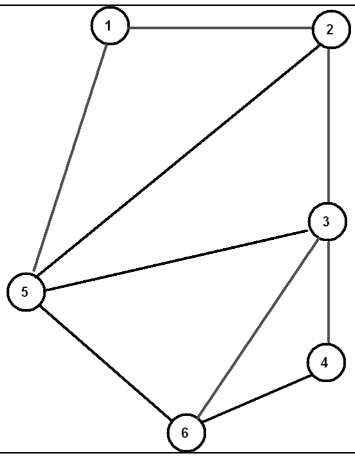
INSPECTION COPY

A rather different example involves using matrices to represent the structure of a road network, shown in Figure 2.6. In this case each entry in the matrix provides information on the road links – if two places are linked they are given a value of 1, or a value of 0 to indicate that they are not directly connected (places are not regarded as linked to themselves, so the diagonal entries are all set to 0). In this case the rows indicate *From* and the columns indicate *To*. Because there are no one-way streets shown the connectivity matrix is symmetric – its upper and lower sections either side of the diagonal from top left (shown shaded) are mirror images of each other. We are going to use this representation to identify the shortest paths through a road network.

INSPECTION COPY

Figure 2.6 Graph and matrix representation of a road network

Simple road network



Binary connectivity matrix

0	1	0	0	1	0
1	0	1	0	1	0
0	1	0	1	1	1
0	0	1	0	0	1
1	1	1	0	0	1
0	0	1	1	1	0

Let us denote the binary connectivity matrix by the letter **C** and calculate $\mathbf{C}^2=\mathbf{C}*\mathbf{C}$ (take my word for it, this is useful). First we multiply row 1 by column 1 to get cell position c_{11}=0*0+1*1+0*0+0*0+1*1+0*0=2. This gives us the number of ways in which we can leave location 1 and get back to it in 2 steps – which can be confirmed from the diagram (from 1→2→1 and 1→5→1). Now we calculate the next cell entry in the same way, row 1 column 2, or c_{12}=0*1+1*0+0*1+0*0+1*1+0*0=1, so there is just 1 way of getting from location 1 to location 2 in two steps – this is the route from 1 to 5 to 2. More generally, if we compute the whole of $\mathbf{C}^2$ we get a matrix showing all the possible two-step routes from every location to every other one. Likewise, from $\mathbf{C}^3$ we get all 3-step routes and we can continue this process until there are no cells in $\mathbf{C}^n$ that contain 0's. This occurs when every point is reached from every other point, and is called the *diameter* of the network (or sometimes the solution time).

Instead of doing all this by hand, you can perform these multiplications very simply in Maple or MATLab, and it is also fairly easy to get Excel to generate these results (using its Mmult() function). The results are shown below:

$\mathbf{C}^1$

0	1	0	0	1	0
1	0	1	0	1	0
0	1	0	1	1	1
0	0	1	0	0	1
1	1	1	0	0	1
0	0	1	1	1	0

$\mathbf{C}^2$

2	1	2	0	1	1
1	3	1	1	2	2
2	1	4	1	2	2
0	1	1	2	2	1
1	2	2	2	4	1
1	2	2	1	1	3

$\mathbf{C}^3$

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

By the time 3-steps links have been made all the cells contain non-zero values, so the diameter of the network is 3 and we stop there. In these matrices I have shaded some of the cells – they are shaded if they contain a value>0 having had a 0 entry in the previous matrix. These identify the length (in terms of steps) of the shortest paths from any location to any other. For example, consider the set of routes from location 1 in our original diagram. There are 3 different ways in which the route from location 1 to location 4 can be achieved (1-2-3-4, 1-5-3-4 and 1-5-6-4) and this is indicated by the number 3 in the cell of C^3 that has been shaded. But that's more than enough about matrices for the time being. I will return to their use in **Part II**.

2.5 Punctuation

In the case of computers, the important thing is to store numbers in as precise, consistent and meaningful format as is possible with the available technology. We then need to convert this information into alternative forms for processing, display or output according to a set of rules that apply for the country and/or data type in question. Most of us are familiar with this problem in the context of the formats that other countries use for dates, times, telephone numbers and currencies. Less familiar are country-specific number-handling rules.

A simple example serves to illustrate some of the problems – the mysterious comma. In the quote shown below, from the delightful book by Lynn Truss on punctuation in English, the meaning of the phrase

"Eats shoots and leaves"

is entirely changed if a comma is included after the word "Eats". With a comma after "Eats" we might be describing a lone gunfighter, entering a bar for a meal, shooting a customer and exiting rapidly. Without a comma added we could be describing a Panda.

In the context of numbers, commas have an equally vital role, and are often used to break long numbers into three digit subsets (thousands). For example:

12456789.00 can be written as 12,456,789.00

for greater clarity. Occasionally, but not in North America, 12456789.00 is written as 12456789·00, i.e. using a raised or 'middle' dot. However in many countries, including much of Europe, the use of the period (raised or otherwise) is not acceptable since commas are used in preference to decimal point notation and the 'correct' format is:

12456789,00 which may also be written as 12 456 789,00

The risk, in this second example, is that digits can be inserted into the gaps, which invites forgery, so for many financial institutions (banks etc.) such an arrangement is unsatisfactory. It is however officially permitted in Belgium, Luxemburg and French-speaking Canada.

By contrast, in India only the first block of a thousand is separated by a comma. After this commas are used to separate hundreds, as in:

12456789.00 which may be written as 1,24,56,789.00

Very large numbers, such as those with perhaps 20 or more digits, which are important in applications such as data security, are often written with blocks of 5 digits separated by commas or spaces.

Such differences matter a great deal, since not only can they lead to confusion when you read them, but they may result in incorrect processing of numbers. For example, a widely used data exchange format is the readable text arrangement known as Comma Separated Value (CSV) files. But if commas form part of the format of an individual number, this data format may produce files that are completely incorrect. To solve this difficulty a CSV file for use in France, say, typically uses a semi-colon (;) to separate the numbers rather than a comma. In countries that use a right to left mode of writing, commas are reversed, hence in Arabic:

, becomes ،

Other punctuation symbols are used in a wide range of contexts, but rarely with consistent or universal meaning. The exclamation mark, written in its standard form as !, is used to denote a factorial operation (as noted earlier), i.e.

$n!=n(n-1)(n-2)\ldots1$

e.g. 4!=4*3*2*1

Quotation marks " " and ' '; brackets ( ), { }, [ ]; and spaces are widely used to separate and group numbers and expressions, but there is much scope for confusion and error. For example, when $<a,b>$ is used does it refer to a form of bracketing, or arithmetic operations of greater than and less than, or perhaps some other operation such as assignment? Likewise, the use of square brackets, such as [0,5] is often used to mean "numbers in the range 0 to 5, *including* the values 0 and 5" (generally called the *closed* interval), whilst (0,5) may refer to the same range but *excluding* 0 and 5 (generally called the *open* interval), or may mean simply the pair of values 0 and 5.

As far as possible, most computer software packages, particularly those that are designed for symbolic analysis, will attempt to identify and report on such issues from the context in which the

INSPECTION COPY

user is working, but there is no guarantee that this interpretation will be correct or unique. In the case of the use of brackets, it is essential to check that every opened bracket has a corresponding closing bracket, and to recognise that nested brackets, i.e. bracketed expressions within other brackets, are normally to be calculated from the inside outwards and from left to right. So, $(a*(b*(c–2)))$ is calculated as c–2 (=R1 say), then b*R1 (=R2 say), and finally as a*R2. It is also important to observe that bracketing is only a valid operation for finite series – with infinite series it may give incorrect results.

Bracketing focuses attention on rules of precedence, i.e. rules relating to the order in which operations will be carried out. Mathematical expressions entered into computer programs are processed according to well-defined rules of precedence. These rules will vary according to the particular software involved, but typically operations are evaluated from left to right, with bracketing over-riding other rules, as described above. The precedence rules for basic mathematical operations in many software packages are similar, and are as shown in Figure 2.7 (these are Excel's rules, highest precedence first):

Figure 2.7 Excel precedence rules

–	Negation (as in –1)
%	Percent
^	Exponentiation
* and /	Multiplication and division
+ and –	Addition and subtraction
= < > <= >= <>	Comparison

Because there is a very limited subset of easily typed characters (letters, numbers and sundry symbols) on most computer systems, these characters are used and re-used in many different ways. As we noted earlier, the widespread use of the ASCII 8-bit coding scheme means that at most 255 distinct letters, numbers and symbols are available. To extend this set an alternative coding system using more than 8 bits is required or symbols must be used in combination. For example the combination := may be used as a single operator, meaning *assignment*, such that the left hand side of the expression is assigned the value or expression on the right hand side. This is differentiated from = meaning *equality*, such that the two sides of the expression have the same value. Entering more complex expressions and symbol arrangements into computer systems requires additional facilities in the software one uses. This subject is covered in more detail in **Part II**.

2.6 Rounding

> *"When completing your personal tax return please do not use pence. Round* **down** *your income and gains. Round* **up** *your tax credits and tax deductions. Round to the nearest pound." UK Inland Revenue Income Tax form guidance, 2004*

Rounding numbers up or down is a source of constant difficulties. In most countries the norm in basic arithmetic operations is to round decimal numbers that end in 0…4 down, and numbers that end in 5…9 up. For example, 12.045 would be rounded up to 12.05 whilst 12.044 would be rounded down to 12.04. This rule favours rounding up, since a number ending in 0 does not strictly speaking require rounding. An alternative, used as the default in some software packages (such as Maple), is to round to the nearest whole number, with ties being rounded to the nearest *even* value. To illustrate this, we can set the maximum number of digits to be handled to just 2 and then look at an expression like 1.25x2. The software will change the value 1.25 to 1.2, because it rounds to the nearest even value in the case of a tie (it could go up or down in this case), and instead of producing the correct answer 2.5, it calculates the result as 2.4; had we chosen the expression 1.35x2 then answer given would have been 2.8. In practice you may not observe this effect unless you are working with large numbers, because the default number of digits used by the package in question is 10, so it will look absolutely correct in most instances. But the principle remains the same – rounding numbers alters results.

A good example is exam marking. In Table 2-2 we see the marks awarded to two students for 4 exam questions. Each question has been marked out of 50, and the average of the examiners' marks used to score each question, with halves being rounded up. Student 1 is awarded 71% for her exam paper whilst Student 2 receives 70%, despite the fact that Student 2 actually scored more on the paper than Student 1, as can be seen from the totals.