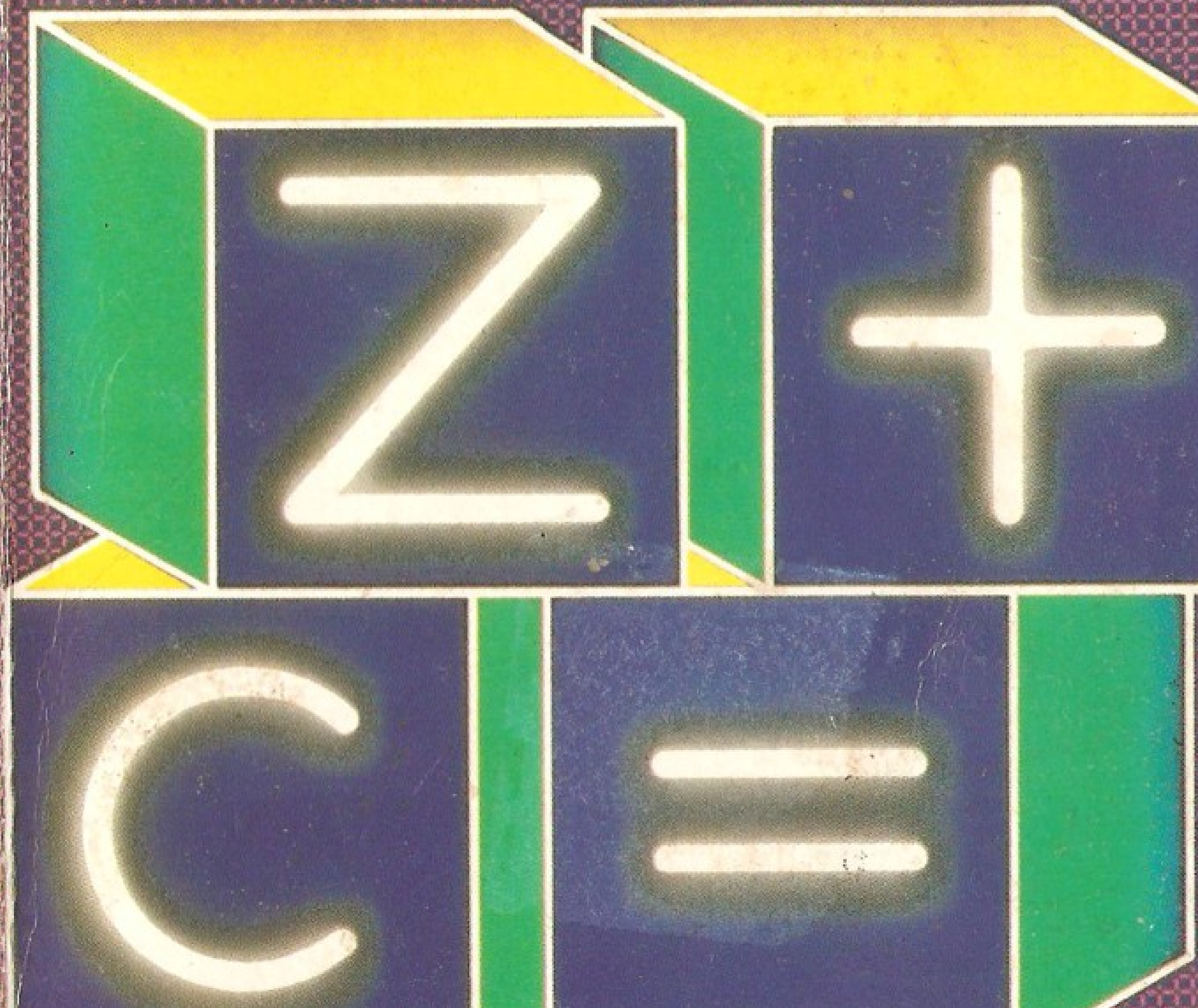
CENTURY COMMUNICATIONS

ROBERT CARTER

# MATHS TUTOR FOR THE SPECTRUM



# —MATHS TUTOR—
# FOR THE SPECTRUM

Robert Carter

CENTURY COMMUNICATIONS
LONDON

*This book is for Claire, without whom . . .*

A cassette containing the programs listed in this book is available from Century Communications Ltd, Portland House, 12–13 Greek Street, London W1V 5LE



First published in Great Britain in 1984
by Century Communications Ltd,
12–13 Greek Street,
London W1V 5LE

ISBN 0 7126 0606 8

Printed in Great Britain by
Hazell, Watson & Viney Ltd, (a Member of the BPCC Group of Companies)
Aylesbury, Bucks.

# CONTENTS

# PREFACE

## 1 Who should read this book

Maths used to be a lot of hard work for the brain, but now the home computer has arrived we can save our brains for understanding the ideas in maths, leaving the sums to the computer. This book is meant for people who want to (or have to) learn maths and are looking for an easy way to do it.

Then there are those who've been playing Wobbly Gobbly games and are beginning to wonder what else their computers can do.

Finally there's the serious programmer who is looking for a set of maths routines to incorporate into his or her programs.

## 2 How to use this book

Like everything else in this book, it's straightforward: start at the beginning and read until you get to the end, stopping to enter and run the programs you come across. (Chapter 1 is an optional chapter especially for those who know absolutely nothing about the Spectrum.)

## 3 What the book covers

I've tried to cover most of the basic applications you can put the computer to work on, without getting too complicated. You'll find most of the material in an O-level or A-level syllabus, and it will certainly help if you are doing maths at school or college. The idea was to cover the mathematical functions available on the Spectrum and to show you ways of making your computer deal with functions which do not appear on the Spectrum keyboard.

# 1

(If you can do BASIC programming you might want to skip this chapter.)

# THE JOURNEY BEGINS

(for those who've never written a program on the Spectrum – but would like to).

Plug the Spectrum in and get the TV tuned in, and you'll see a message at the bottom of the screen that reads:

© 1982 Sinclair Research Ltd

Press any key, and it will disappear leaving a flashing K in the bottom left-hand corner. This flashing letter is the *cursor*; you can read all about it on pages 13–22 of the Spectrum manual, as well as learning how to get at the keys properly.

There are two ways of getting the Spectrum to do something:

1 You can *command* it.
2 You can write a *program*.

Let's try a command. Find the B key

| | |
|---|---|
| B | * |
| Border | |

and press it. The word BORDER appears, and you are expected to press one of the number keys to choose a colour. Try any number from Ø to 7 (they have the numbers on the top row of keys, and each one has a colour written above it). You ought to choose 6. (That's YELLOW, check?) So the screen reads:

BORDER 6

Now press ENTER. See what happened? The border of the screen went yellow, the centre remained white, and there is some gubbins written at the bottom. (Forget that: it just means the computer has carried out your command.)

If that small triumph has filled you with delight, try pressing BORDER, followed by any key from 0 to 7 and then ENTER, and you can make the border any colour you like.

Tired of that? Well, make it go yellow again, because we're going to write a program, and I think yellow might just serve our purpose.

The difference between a command and a program is that a program is a series of commands, one after the other, that can be entered into the machine's memory. When you've got a bunch of them together, you can make the Spectrum go through them in order and carry out each command. It does them in order, and it knows the order because you number each command. That way you can build up a sequence of things the computer can do one after another, and the result could be quite complicated.

You want an example? This is a program that goes through your border colours one at a time. Type it in (don't bother with the spaces), starting with '10':

*Program 1*

| Line no. | Instruction | | |
|---|---|---|---|
| 10 | BORDER | 0 | ENTER |
| 20 | BORDER | 1 | ENTER |
| 30 | BORDER | 2 | ENTER |
| 40 | BORDER | 3 | ENTER |
| 50 | BORDER | 4 | ENTER |
| 60 | BORDER | 5 | ENTER |
| 70 | BORDER | 6 | ENTER |
| 80 | BORDER | 7 | ENTER |

You'll notice that when there's a line number in front of the instruction, and you ENTER it, the line gets printed up at the top of the screen. Then the next line gets tacked on below it, and so on. You will have noticed, however, that the border hasn't done anything. In other words, the Spectrum has not carried out your instructions. To make it do so you have to command it to RUN. That is, press RUN and ENTER.

Did you catch that? Maybe not. Well, believe it or not, it just turned the border eight different colours from black to white, one after the other, but it's interesting to note that it did it so fast you probably couldn't see it.

Now type the line:

15 PAUSE 50 ENTER

(If you make a mistake on any of these lines you can delete a letter or number by pressing the key called CAPS SHIFT – it's bottom left of all the keys – and whilst it's pressed down pressing the 0 key – it's top right of all the keys.)

You may have wondered why the line numbers went up in tens, 10, 20, 30, 40, etc. It would have been possible to number the lines 1, 2, 3, 4 . . . but then it wouldn't have been possible to wedge a line between, say, lines 1 and 2. As it is, we can stick line 15 between lines 10 and 20.

When you pressed ENTER, you will have seen line 15 jump into the program in the correct place. Now I want you to do the same between the other lines, i.e. type:

25 PAUSE 50 ENTER
35 PAUSE 50 ENTER
45 PAUSE 50 ENTER

and so on until each of the lines containing a BORDER instruction has a line containing a PAUSE instruction after it (no need to put one right at the end of course).

Let me tell you what the PAUSE instruction does. Big surprise: it makes the computer pause. But what about the number after PAUSE? That is the number of fiftieths of a second it will pause. So if you write PAUSE 50, it will wait just one second. If you write PAUSE 100, it will wait two seconds. If you write PAUSE 25, it will wait half a second. The only exception is PAUSE 0. This means, 'Wait for a key to be pressed before continuing.'

Anyhow, you can find this all out in the orange manual that comes with the machine and in loads of other books. Let's stick to our own business.

When you've got your sandwich ready, take a look through it and decide what it will do. Not hard, is it? You know that the computer does each line in order of its line number. So it will start at line 10 and do BORDER 0 which turns the border black. Then it will wait one second, for a PAUSE 50, giving you a chance to check that the colour is really black. Then it will move on to line 20 and do a BORDER 1 (and turn the border blue), followed by line 25 and PAUSE 50 again to let you see the blue border. And so on . . .

Run it and see if you were right.

This is all very well, but it's no big deal messing around with the border colour. It's like playing with the ash trays of your new Jaguar. So let's get into the meat of the screen display, the big white patch in the middle. (Before we start, command the machine to give a yellow border.)

Really, we want to dump the program you've just written overboard. Happily there's a command to make it do that. It's called NEW and means, 'Start again.' It also means 'erase all program material from memory,' so be careful that that is what you want to do before using the NEW command. We ought to be back at square one now with no program lines and Sinclair's copyright message back on the bottom of the screen. (The NEW command also got rid of my yellow border too – I told you you've got to be careful of that NEW command – so command the computer to give me another yellow border, please.)

Now you will see why I'm so keen on a yellow border. It makes a good clear edge round the area of the screen that is available to us to make marks on.

There are three instructions that I want you to think about.

1 PLOT (it's on the Q key)
2 DRAW (on the W key next door to Q)
3 PRINT (on the P key above ENTER)

PLOT puts a tiny little dot on the screen. Where on the screen? Well, you have to tell it where. And how do you tell it that? Simple. The screen is built up out of dots in a huge rectangle like the boxes containing the letters in a crossword puzzle. There are 45,056 of these dot locations. 256 across and 176 up. So if you say PLOT 0,0 ENTER, you will get your spot at point 0,0 or right in the bottom left-hand corner. Try it. Do you see the little black dot down there in the corner?

If you remember that the first number (the one before the comma in the command) means the *across* position, and the second number (the one after the comma) means the *up* position,



you can see that the command PLOT 255,0 will position the dot in the bottom right corner, the command PLOT 0,175 will place the dot in the top left position, and (yes, you guessed it!) PLOT 255,175 puts it in the top right corner.

But before you dash off to the keys to test it all out, try it as a program. Just enter the following lines:

*Program 2*

```
 5 REM PROGRAM 2
10 PLOT 0,0
20 PLOT 255,0
30 PLOT 0,175
40 PLOT 255,175
```

And run it with RUN followed by ENTER.

Have you found out what a colon (:) means in Spectrum BASIC? It separates instructions that you want to write all on the same line. Why would you want to do that? I'm not going to tell you, so find out yourself. The only clue is that it has something to do with saving memory and sometimes also with the IF instruction. Anyhow, you could, if you feel adventurous, enter Program 2 as:

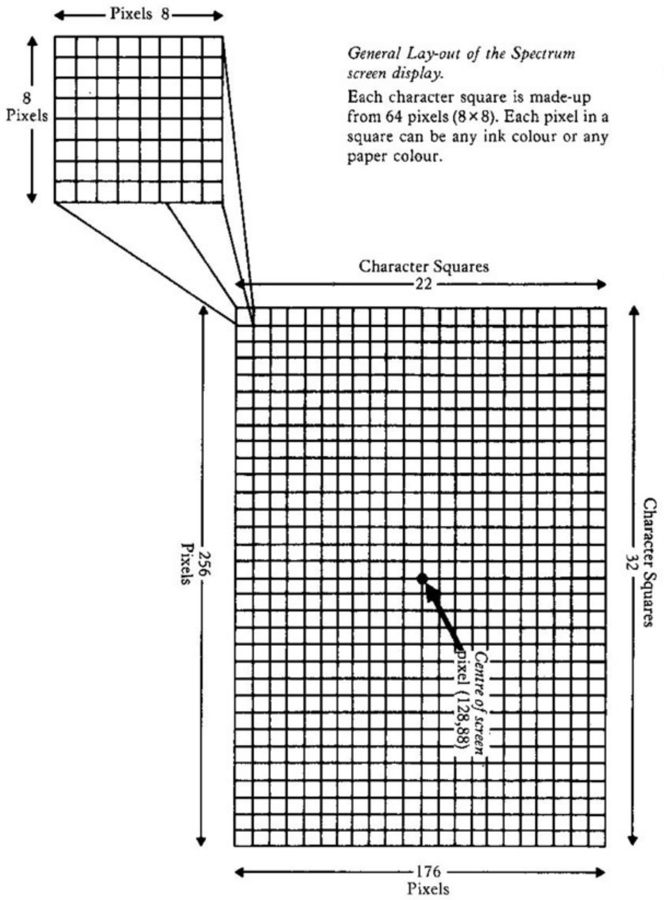
*Program 2a*

```
   5 REM PROGRAM 2a
  10 PLOT 0,0: PLOT 255,0: PLOT
0,175: PLOT 255,175
```

It's not as clear to read, but it was less work on the index fingers, and it does exactly the same thing when run. You must be careful, though, in every case when you use PLOT, to make sure you include the comma, otherwise it won't work. And if you make the numbers bigger than 255 for the across number or bigger than 175 for the up number, it will fail to work (obviously because there is no location on the screen where you're asking it to plot). The error report at the bottom of the screen will say:

B Integer out of range, 0:1

Try it with, say, PLOT 300,0 and see what you get.




*General Lay-out of the Spectrum screen display.*
Each character square is made-up from 64 pixels (8 × 8). Each pixel in a square can be any ink colour or any paper colour.



How about the DRAW command? It too must be followed by some numbers, and again these two numbers are separated by a comma, the first meaning 'across' and the second 'up'. The difference is that DRAW puts a line on the screen, not a dot, and the numbers are telling the computer how long the line is going to be across and up.

The line it draws will start at the last position plotted, or the end of the last line drawn. If there were no other PLOTs or DRAWs, it will start from 0,0 (the bottom left-hand corner, remember?).

If you're not clear about that lot, don't worry because you can see what's happening much better with an actual example. Try this lot out:

*Program 3*

```
 5 REM PROGRAM 3
10 PLOT 0,0
20 DRAW 50,0
30 DRAW 0,50
40 DRAW 28,15
50 DRAW -50,0
60 DRAW 0,-50
```

If you run it, you might see a pretty meaningless pattern appear, but you will see that it has done exactly what you told it to do. I must stress that if you want a line to go towards the right or in the up direction, the draw numbers must be positive (bigger than zero). If you want lines drawn towards the left or downwards, you must make them negative (less than zero). Going down or to the left is like taking away from the up and across numbers.

One of the reasons why any so-called complicated subject is thought to be difficult is because the people connected with it are like members of a club. There are yacht clubs and darts clubs and Reggae clubs and clubs whose members are architects. The BMA is a club of doctors, Parliament is a club of politicians, Spurs is a club of footballers. The whole world is a mass of interlocking clubs, and nothing stops you being a Spurs-supporting, yacht-sailing, Reggae-dancing architect who's also an MP!

If you were sitting in a hotel lounge and a convention of any one of these clubs came in, sat down around you and started chatting about their speciality, as an outsider you'd very quickly find yourself adrift. You'd be as left out of it as if you were a foreigner who couldn't speak the language, and the reason is because clubs speak their own languages. They call it 'jargon', and why should computer people or mathematicians be any different? They're not. And so it is useful to pick up a few words:

*Vocabulary*

The x direction: across the screen, the number before the comma in DRAW and PLOT.
The y direction: up the screen, the number after the comma in DRAW and PLOT.
Coordinates: the numbers in PLOT and DRAW, so that there is the x coordinate and y coordinate.

This system was introduced by a Frenchman, René Descartes (1596–1650), so the system is often called 'Cartesian' from the Latin version of his name. It just means up and down and across and back, like graph paper, and it's ideal for a rectangular display on a flat TV screen.

Try taking the coordinates in Program 3 and altering them. You should take care, though. If in PLOT or DRAW you use an x coordinate greater than 255, or a y coordinate greater than 175, it will be out of range. If you tried to make the Spectrum draw a line beyond the edge of the screen, you'd be given an error report saying 'Integer out of range'.

As you can see there's scope here for drawing pictures, but what a laborious business! Is that all there is to it?

Happily the answer is No!

By combining it with other computing instructions you can make interesting programs to draw some very geometrical creations. I didn't want to ask you to run before you can properly walk, but try a few of these programs without worrying too much about why or how they work. I guarantee that you will know all about it by the time you reach the end of this book.

First a note about finger technique:

1 To get at a *red* symbol *on* a key, press the red SYMBOL SHIFT key at the same time as the key you want.



2 To get at a *green* symbol *above* any key, press CAPS SHIFT and SYMBOL SHIFT until the cursor turns into an E, then press the key you want.

3 To get at a *red* symbol *under* a key, get an E cursor as in 2 above, but hold down the SYMBOL SHIFT when you press the key you want.

It sounds quite a rigmarole, doesn't it? In fact I had second thoughts about buying a Spectrum in the first place because of this stuff, but it's amazing how fast you get into it, and now I find it laborious to use an ordinary typewriter style keyboard!

Just one more thing. If you think about it, the coordinates of the centre of the screen will be x=128 and y=88, so that if you PLOT 128,88 you get a dot in the middle of the screen. Now for some automatic PLOT 'n' DRAW graphics:

*Program 4 GREEN CROSS*

```
   5 REM PROGRAM 4
     GREEN CROSS
  10 BORDER 0: PAPER 4: INK 7: C
LS : PLOT 128,88
  20 FOR N=3 TO 80 STEP 3
  30 PLOT 128+N,88+N
  40 PLOT 128+N,88-N
  50 PLOT 128-N,88+N
  60 PLOT 128-N,88-N
  70 NEXT N
```

*Program 5 SCREEN BORDER*

```
 5 REM PROGRAM 5
   SCREEN BORDER
10 PLOT 0,0
20 DRAW 255,0
30 DRAW 0,175
40 DRAW -255,0
50 DRAW 0,-175
```

This one will only show up if you have a paper and border of light colour with ink dark colour, or vice-versa. Otherwise it gets lost in the border. It can be quite a useful little routine for making your displays look more professional.

*Program 6 CHARACTER SQUARES*

```
 5 REM PROGRAM 6
   CHARACTER SQUARES
10 FOR X=8 TO 255 STEP 8
20 PLOT X,0: DRAW 0,175
30 NEXT X
40 FOR Y=8 TO 175 STEP 8
50 PLOT 0,Y: DRAW 255,0
60 NEXT Y
```

Type in Program 6, run it and you will find a grid of lines appears that cuts the screen into a number of small squares. Each of these squares is eight by eight pixels. (A pixel is a screen location like the dot made when you use a PLOT command, and so each of the squares on the screen is 64 pixels.)

If you count up the number of boxes on the screen after running Program 6, you will find there are 32 across and 22 up, so the total number is 32 times 22, or 704. Work it out yourself. And these boxes are used by the computer as printing locations. If you print anything on the screen, it must be printed in one of the little boxes generated by Program 6.

The command for printing something is PRINT, and another command will let you choose where you want to print it.

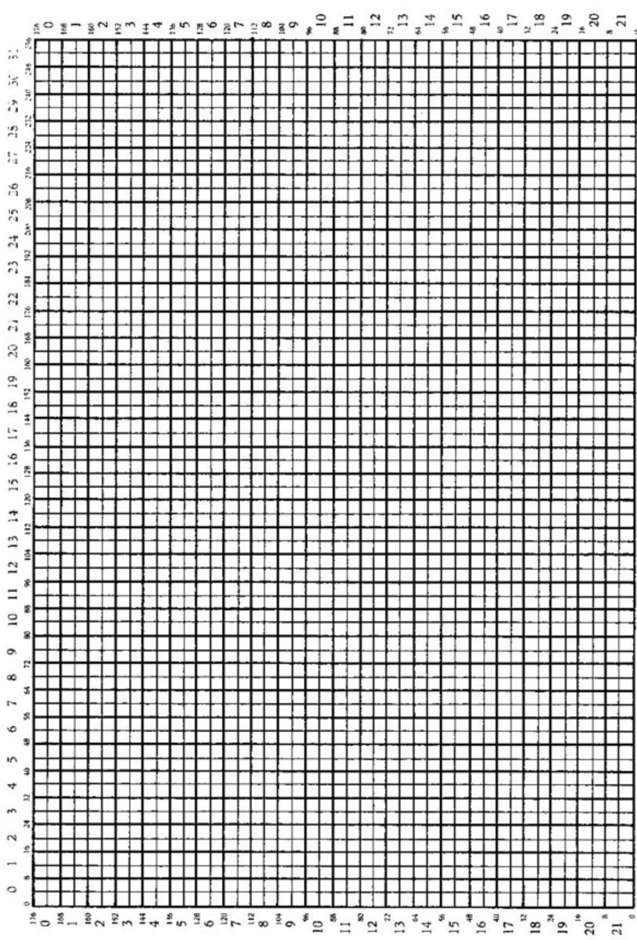
Try this procedure: command the Spectrum to PRINT 2. It will print '2' at the top left corner. Next command it to PRINT 8. It will print the number '8' on the next line down. If you want it to print a letter or a word, you must enclose that word in double quotation marks (sometimes called inverted commas), so that PRINT "Fred" will print the word 'Fred'.

The AT instruction needs two numbers. The first is a row number, from Ø to 21. The second is a column number, from Ø to 31. (Don't these numbers seem kind of familiar?)

But don't confuse them with the PLOT coordinates. They're different because the coordinates are for drawing, whereas the PRINT AT numbers are strictly line and column numbers. The main point to note is that the y coordinate of the plotting instructions starts at Ø at the *bottom* of the screen because graph



## SPECTRUM SCREEN PLOT & PRINT POSITIONS



paper does that. The row numbers start at Ø at the *top* of the screen because you read words in a book from the top line on a page and work your way down.

Anyhow, the form of the printing instruction is like this:

PRINT AT 1Ø,1Ø;"x"

Be careful to remember the row number is the first one and the column number the second. And take care to put in that semi-colon (;) or it won't work properly.

I suggest that you run Program 6, get that crosshatch on your screen, and then add to the program the following two lines (Program 6a). It will choose boxes at random and print a letter 'x' in the box. Experiment with the various bits of it and see what you can learn.

*Program 6a*

```
  5 REM PROGRAM 6a
 70 FOR N=0 TO 10: PRINT AT (RN
D*21),(RND*31);"X"
 80 NEXT N
```

That just about covers the main areas of how to get instructions into the Spectrum and how to get it to put what you want on to the screen. You can now have confidence to experiment with the Spectrum, and when you have read a bit about the display in the accompanying manual, there's no reason why you shouldn't dream up some programs of your own.

You may be interested to add a couple of extra things to your repertoire.

First, the DRAW instruction. I said you have two coordinates after the word DRAW. True, if you want to draw straight lines, but what if you want lines that bend?

Then you have to have a third number after another comma that controls the amount of bend.

Try this program:



*Program 7 ONION*

```
 5 REM PROGRAM 7
   ONION
10 BORDER 0
20 FOR N=-PI TO PI STEP 0.5
30 PLOT 80,40: DRAW 100,100,N
40 NEXT N
```

'PI' appears twice in line 2Ø, and in case you aren't sure what it is, I'll tell you. PI is on key M and is a number with a value of about 3.14. It is quite famous and has to do with circles and triangles and Greek geometry, and you can meet it again many times later in the book. For now, just remember that it is a Greek letter used as shorthand for the number 3.14.

Now when the third DRAW number is Ø, there is zero bend. When it is exactly PI, the line bends so that it has turned through 18Ø degrees. Negative numbers will bend the line to the right, positive numbers will bend it to the left. So you can maybe see how the onion pattern is built up.

The idea of FOR ... NEXT loops can be found in the Spectrum manual, Chapter 4, pages 31–33, and it helps to read about it to understand Program 7. What's happening is that, as you know, the computer takes instructions one by one in the order of the line numbers. So in Program 7 it reads line 1Ø and sets the border to colour Ø (black). Then it reads line 2Ø and sets a number called the FOR variable (in our program called n) to the number −3.14 (−PI). Then it sees line 3Ø, plots a dot at 8Ø,4Ø and then hops over the colon (:) and does a draw of 1ØØ,1ØØ with a bend of n. (Remember that n was given a value of −3.14 by line 2Ø.)

The clever part comes in line 4Ø. Because NEXT n means that the computer's attention is directed back to the line containing the FOR instruction (line 2Ø), and because in line 2Ø the STEP instruction is .5, it takes .5 and adds it to the FOR variable n, making it −3.14+Ø.5 or −2.64. So that when the Spectrum moves on to line 3Ø again, it plots 8Ø,4Ø just as before; but now, when it hops over the colon, it draws a line of 1ØØ,1ØØ with a bend of −2.64. Then it goes down to line 4Ø and is sent back to 2Ø to do the same thing again.

That's why they call it a loop, and you will keep your Spectrum going around and around until its loop variable has reached its upper limit (which is PI). Then it will go on to line 5Ø if it exists.

If you work your way through this bit of barbed wire you will find that it will go round the loop 13 times, and if you look at the display that Program 7 draws, you will find 13 curved lines.

The other goody in store is an instruction that draws circles for you. Mr Sinclair's design engineers have thoughtfully given us a CIRCLE command. It's very simple if you know what a PLOT command is. (If you don't, then go back to the beginning of Chapter 1.)

Remember that a PLOT command needs x and y coordinates. The CIRCLE does too (because it needs to know where to locate the centre of the circle). The difference is that, although a dot is just a dot, a circle is a circle of certain size. In other words you need to tell your Spectrum how big a circle it should draw. The number it needs is the radius – that's the distance from the centre to the edge of the circle. So you would write:

CIRCLE 128,88,1Ø

if you wanted to write a circle with its centre at the centre of the screen and with a radius of 1Ø pixels. Try it out!

When you've done that, maybe you could work out the biggest radius you can get on to a Spectrum screen. (The answer is 87, because 88 would try to draw a circle 2 times 88 or 176 pixels across, and that's just one pixel too big for the screen to take.)

The same applies to the CIRCLE command as applies to the DRAW command as far as running out of screen is concerned. The Spectrum will stop and give you an 'Integer out of range' error report if you do.

One last thing before we leave chapter 1. Have you noticed that a zero with a diagonal mark through it is used to represent zero on computers? It's done to make sure you don't confuse a zero with a letter O, which is an entirely different thing you must agree.

Computers also use a star symbol '*' instead of a '×' to mean 'times'. So you can't confuse 'times' with the letter 'x'.
For example:

2 * 3=6



Very last thing: a small program to show you the famous FOR . . . NEXT loop in operation using CIRCLEs.

*Program 8 CONCENTRICS*

```
 5 REM PROGRAM 8
   CONCENTRICS
1Ø FOR N=1 TO 8Ø STEP 5
2Ø CIRCLE 128,88,N
3Ø NEXT N
```

And when you have run that, try to figure it through as we did for the ONION program, and see what's happening in detail. Then make the number after STEP into x, and add line 5:

5 INPUT "Step select";x

When you add line 5 to the program, you will find that it will stop at that line and print at the bottom of the screen whatever you put inside the quotes. (If you'd put, 'My name is Julius Caesar' inside the quotes, it would have written that instead!) When it stops, it is waiting for you to choose a number for the STEP variable which you have turned into x. So the way to satisfy it is to put a number in: key any number from, say, 1 to 8Ø and press ENTER. As soon as the number goes in, the Spectrum will continue and . . . well, do it and see the effect. And don't forget to experiment.

# 2

# THE REAL LINE

## 1 The real line

Either you've come here after reading chapter 1 or you skipped chapter 1 and came here directly.

I'm going to assume that you can follow all the stuff covered in chapter 1 without too much difficulty. And since computers ultimately deal with numbers, I thought I'd begin with the Number System. It's simply overflowing with jargon. So let's wade through it.

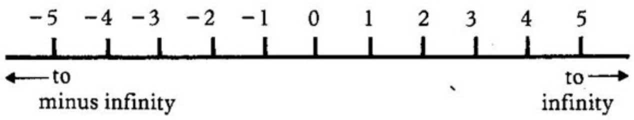
Start with zero.

This line is called the Real Line:

---

Actually, it's only a bit of it. The Real Line is infinitely long (meaning that it stretches from minus infinity to infinity), but happily we only have to bother with the middle bit of it.

At the absolute centre is zero. At regular intervals to the right are the numbers 1, 2, 3 and so on, and to the left are minus 1, minus 2, minus 3 and so on. Thus:




You can do adding and subtracting calculations on the Real Line. Step to the right for every one you want to add and one to the left for every one you want to subtract. Why not draw your own sketch of the Real Line on a sheet of paper and get a counter (you could use a tiddly-wink or a penny) and try it out.

Take the sum $2+3$. You and I both know that the answer is 5. Let the Real Line prove it. Put your counter on 2, and move



your counter three spaces to the right to add. The answer? Five!

Easy! But what about $3-8$. There are those who say it can't be done, but we know that by using the Real Line we can get at the truth: $3-8=-5$.

Now you know why the Real Line needs to be infinitely long. Because numbers can be as large positive or large negative as you like.

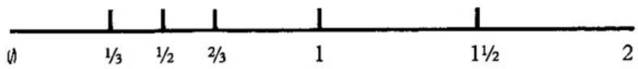
But what's all this got to do with the Number System? And what about the Spectrum? It has much more to do with the Spectrum than it seems at first, so hang on in there.

Let's look at a smaller stretch of the Real Line, the bit from zero to 2. Now both 1 and 2 are whole numbers, and in mathematician's jargon are called 'integers'. An integer is any whole number, and there are positive integers and negative integers: for example, 1, 2, −5, 2000, −37, 67,831. I'm sure you can think up numerous other examples of your own. Integers are strung out along the Real Line like a necklace made of pearls, and at first sight you'd think that all numbers were there amongst the integers. After all, one of the first things you learn as a child is to recite the list of positive integers.

But it's not the whole story. There are in fact numbers squeezed slyly in between the integers, and when you bring your spyglass to bear on the spaces between the integers, you see that they are crammed with masses of numbers rammed up against one another cheek-by-jowl.

You may think this is all very elementary and that I'm making a mountain out of a molehill, but I'm not. It's peculiar, and some very queer things go on in the crevices between the pearls.

Let's have a look at the Real Line between zero and 2:




As you can see, half way between Ø and 1 there's ½, and half way between 1 and 2 there's 1½. I've also marked the ⅓ and ⅔ positions, so that you can see where they live.

You probably have a wooden model of the positive side of the Real Line over several integers. You call it a ruler. Some rulers have all sorts of subdivisions: quarters, tenths, eighths, twelfths, sixteenths, and if you think about it you could magnify a portion and magnify that portion again and again and so on until you were looking at hundredths, or thousandths or millionths. All these fractions are important to us, but how can we handle fractions on the Spectrum?

There's no fractions key, no ½ or ¾, so when we want to talk fractions we have to convert them into decimals. Thus:

½ equals Ø.5
¼ equals Ø.25
¾ equals Ø.75

and so on. There's a calculator facility built into the Spectrum, so that you can instantly transform any fraction into a decimal with a single command. Just type:

PRINT 3/4 ENTER

and the Spectrum will give you back Ø.75 printed neatly at the top left-hand corner of the screen. Try it out and prove that:

3/8 =Ø.375
17/34=Ø.5
7/5 =1.4

Or you could get really adventurous with 355/113=3.1415929. Jargon in fractions is usually restricted to calling them 'vulgar' if the number on top is bigger than the number underneath. (Because that makes them bigger than one, like 7/5 is 1.4.) And there are names for the top and bottom numbers in a fraction. The top number is the 'numerator' and the bottom number is the 'denominator'.

But some fractions can't be turned into decimals quite exactly. Try it with our friend the third. Simple task for a computer like the Spectrum with all the computing power it has, you might suppose, dividing 1 by 3. But it can't do it. And neither can the finest computer in the land.

Why?

Because it's endless.

It's easy enough. Numbers can be exact or unending. Exact ones end after a few decimal places, like 1.23456789 or 5.75 or 0.383838. Exact numbers are exact. Once you have them, you know all about them, the whole story.

With unending ones it isn't so simple. You can never know all



about them because you can never write them down completely. And why? Because the string of decimal places never ends!

Some numbers just go on and on and on and on and on and on and on and on and on . . . forever.

You don't believe me?

I told you it was mighty peculiar, didn't I? Let me give you an example of an unending number:

1/3

There you are. No computer could ever write that out in decimal form: it has an infinite number of decimal places. It goes something like:

Ø.3333333333333333333333333333333333 . . .

and then I get bored and stop.

If you run it on the Spectrum, you'll get Ø.33333333 and then it gets bored too.

Then there's the case of ⅔. That's an unending number too. If you divide 2 by 3, you will come up with this neverending stream of numbers. And in this case the repeating decimal is 6. If you command your Spectrum to:

PRINT 2/3

it will give you back:

Ø.66666667

Can you think of a reason why the eighth place is a 7 instead of a 6? It's what we call rounding. This is quite interesting because in applications like computers where you sometimes need accuracy there's a common way of handling these unending numbers. The Spectrum will display numbers to eight decimal places. (That means it will list up to eight numbers after the decimal point. Not when the decimal doesn't go that far: it wouldn't bother to write out 3.5ØØØØØØØ for 3.5 for instance.) But when handling these unending numbers it has to stop somewhere, and it has been designed to stop at eight places. But do you see the problem?

Think of this: if I had a number as the result of some calculation that came to Ø.123456789, and the computer would only display to eight places, then I would read on its screen Ø.12345678 which is pretty close I'll admit. But it could have displayed the number 0.12345679 which would have been only 0.000000001 out from the real number instead of 0.000000009 out. Follow?

If you don't follow, don't worry. This isn't school, and if you don't mind about the intricacies of rounding, I don't mind either.

If you think you've got that rounding idea in your head, let me just add a couple of things to make it a bit clearer. If you have a number in the first non-displayed decimal place greater than five, it is clearly more accurate to bump the last displayed decimal place up by one. And that is why 2/3=0.6666666**7** on the Spectrum.

Equally, just cutting the first non-displayed number off and throwing it away if it is less than 5 (or equal to 5) is getting the best accuracy. That's why 1/3 is shown as 0.33333333. Just cutting the last figure off is called 'truncating'. (When I first came across the word 'truncate', it conjured up visions of someone cutting off an elephant's trunk, so that's how I remembered it!)

So what else is there about the Real Line? Well, I suppose you could ask why we call it the Real Line. And I could say because it's a line full of real numbers. There are also unreal numbers, but we're not going to discuss them here. I absolutely insist we leave it all until later.

## 2 Variables and constants

Try:

PRINT 3/11

and you will get 0.27272727. Here the repeat is a pair of numbers, and if you try 22/7 you'll get 3.1428571. There are only seven decimal places in that one, and it's not irrational (an irrational numbers is one that cannot be written as a fraction), but it's a good approximation to perhaps the most famous number of them all, PI. Pronounced 'pie' as in steak and kidney pie, it's the Greek letter written $\pi$. It looks to me like a little stool.

Those of you who've read Chapter 1 will have heard about PI. It's on the M key (or rather, above it, written in green), and the reason Mr Sinclair's engineers have gone to the trouble of



putting it on a key is that it comes into lots of different areas of maths, and it saves the tedious keying-in of the number every time you want to use it.

It is, in fact, irrational and if you type:

PRINT PI

you will get 3.1415927. So the approximation 22/7 is not bad, is it? Try PRINT (22/7)−PI, and you'll get back 0.0012644893 showing that there's very little difference between the two numbers.

The approximation 22/7 was discovered by the Greek mathematician Archimedes (287–212 BC).

I happen to know that a 35-figure approximation to PI is:

3.1415926535897932384626433832795028 8

because I've read it in a book. Can you see where the Spectrum rounds it off? PI is an example of an irrational number that just goes on and on without any repetitions at all.

Those of you who are good at noticing things might have seen a coincidence. Back on page 18 I said that if you were really adventurous you might like to try 355/113. Look back and see what it comes to.

That is the best approximation to PI I've ever seen just by dividing two numbers! And it didn't escape the attention of Tsu Ch'ung Chi (430–501), a Chinese engineer who found it too. If he'd had a Spectrum, he could have got to it much more quickly! He could've used Program 9.

Program 9 looks for an approximation by dividing integers by other integers; it never lets an integer go above a thousand, and it looks for PI to three decimal places. Try it and see if it will find both Archimedes' and Tsu's approximations. It takes over 15 minutes, so wait for it!)

*Program 9*

```
5 REM PROGRAM 9
PI APPROXIMATIONS
10 FOR n=4 TO 1000
20 FOR d=INT (n/4) TO n/3
30 LET p=n/d
```

```
   40 IF p>PI-0.0001 AND p<PI+0.0
001 THEN  PRINT n;"/";d,n/d
   50 NEXT d
   60 NEXT n
   70 BEEP 10,15
```

This program looks a bit more complex than those in Chapter 1, and so it is. But it does contain some quite subtle points in there, and so I thought I'd better explain them to you. You might have sussed out that 'n' stood for numerator and 'd' for denominator, and it would have been possible to run both n and d from 1 to 1000. But that would have taken ages, even on the super Spectrum. So line 20 has some means of keeping the running time short by excluding obviously wildly wide-of-the-mark examples. We know that d is going to be between a quarter and a third of n. Line 40 selects values within the required accuracy range, i.e. between PI − 0.0001 and PI + 0.0001.

Program 9 is rather daft from a mathematical point of view because it relies upon knowing PI to find approximations, but it does serve to show just how clever old Tsu must have been. We have shown in a program taking over 15 minutes to run that his approximation is the best that could be got using two integers less than a thousand!

This is all very well, but what is PI? We know its value pretty well and lots of its history, but what's all the fuss about?

It's a fundamental constant. It's so important in maths it gets into almost everything, and a couple of good examples are circles and triangles.

The reason the ancient Greeks got to know about PI is because they were into architecture, and that meant they had to be into geometry. (Just *you* try building temples without understanding circles, squares and triangles!)

Consider this. If a piece of string is laid in a circle and the circle is one yard wide, then the piece of string is PI yards long.

Yes, it's true: the diameter of any circle (the width, if you don't like the word 'diameter') is related to the distance round the circle (called the circumference) by the constant PI.

PI is called a constant because its value never changes. Other constants are all over the place in maths and the sciences – the speed of light, for example, or the constant that converts inches to centimetres, or something called 'e' that you'll be meeting later in the book. There are lots of them, but PI is probably the king of them all.



Consider also this. If you have a circular table top, and the distance from the centre to the edge is one yard, then the area of the table top will be PI square yards.

Again we have the odd result that the distance from centre to edge (radius) is related to the area by our pal PI. We can write equations for both these results (if you are uncertain about what equations are, read section 3 of this chapter and find out):

C = PI*d (where C is the circumference and d is the diameter)

A = PI*r*r (where A is the area and r is the radius)

You can see, if you draw an accurate circle and a diameter and radius on it that a diameter is twice as long as a radius, or as an equation:

d = 2*r

Let's put the Spectrum to work on circles:

*Program 10 CIRCLES AND PI*

```
    5 REM PROGRAM 10
      CIRCLES & PI
   10 INPUT "Radius in cm? (1 to
87)";R
   20 PLOT 128,88: DRAW R,0: PRIN
T "Radius ";R;" cm": BEEP 1,20
   30 PLOT 128,88: DRAW -R,0: PRI
NT "Diameter ";R*2;" cm": BEEP 1
,20
   40 CIRCLE 128,88,R: PRINT "Cir
cumference ";R*2*PI;" cm": BEEP
1,20
   50 PRINT "Area ";PI*R*R;" sq.c
m"
```

The BEEPs are in there because otherwise it zooms through the program, and you can't follow it. It could have been done with

PAUSE instructions, but BEEPs sound better!

When run the program stops to ask for a radius length in cm. It then draws a scale model of the circle (approximately 10 times smaller: measure it and see) and gives you its vital statistics.

If you want to use radius lengths greater than 87 it will not work, but if you just want to work out the vital statistics alone try program 10a:

*Program 10a   CIRCLE STATISTICS*

```
   5 REM PROGRAM 10a
     CIRCLE STATISTICS
  10 INPUT "Radius? ";R
  20 PRINT "Diameter ";R*2;" uni
ts"
  30 PRINT "Circumference ";R*2*
PI;" units"
  40 PRINT "Area ";R*R*PI;" squa
re units"
```

The beauty of that is that you can get radius lengths of any size and any units. You can choose inches, centimetres, miles, chains, rods, poles or perches . . . anything you like. I always found it strange that the area of a circle is measured in *square* inches. Well, the volume of a sphere is in *cubic* inches, isn't it?

Anyhow, all this PI stuff is getting off the beaten track. This section is supposed to be about Constants and Variables so let's talk about them. What are they?

For once we have a piece of reasonably plain English: 'constant' meaning unchanging, and 'variable' meaning the opposite. Now when you apply that to numbers it comes down to this: in an equation like C=PI*d, PI is a constant. You know its value and that value never changes no matter when you use it. You also have other quantities, d and C, which are variables. So that if you're talking about a big circle, both C and d would be bigger than the C and d in a small circle. The PI, however, is always the same.

In the Spectrum, variables are specified using a letter or a word made of letters and numbers, e.g. R, C2, Diameter, Area7. This is so because it often helps to have a name of a variable give you a clue to what it's a variable of. So if we called a variable for the radius of a circle R, we'd be able to read the program better than if we called it C or f or something.

In ordinary non-computing maths, variables are usually just a single letter, like the three sides of a triangle might be called a, b and c. And when ordinary letters run out they sometimes go on to Greek letters, especially for angles. Theta is a favourite. It looks like a circle with a bar in it: θ.

Constants are often found on old exercise books, converting one set of units into another, such as Imperial units into metric. So that you might read:

1 metre = 3.28ft
1 litre = 35.2 fluid ounces
1 mile = 1.609 kilometres

Those unhealthy looking numbers are constants. They relate Imperial measures to metric measures and vice-versa, because if you ask how many feet there are in, say, 10 metres, the equation would be:

10 metres = 3.28*10ft
= 32.8ft

The 3.28 stays in there all through until you do the multiplication.

If the Spectrum had not had a PI key, it wouldn't have stopped you using it to work out the areas of circles and so on. It would have meant that you'd have had to find an old exercise book somewhere that told you the value of PI, and then you could have defined it to the Spectrum with a LET instruction.

LET instructions are how you get the information into the computer in many cases, and sometimes they tell the computer to perform a calculation.

Remember how all along we've been using the Spectrum's built-in calculating facility? We just tell it to PRINT 5+4, and it returns 9. LET does the same thing, only invisibly!

If you were to write:

```
10 LET a=5
20 LET b=4
30 LET c=a+b
40 PRINT c
```

it would give you 9. But it only prints it because of line 4Ø. So what if you delete line 4Ø and run it again?

Nothing!

Wrong! It does do something. It sets a variable called 'a' equal to five, it sets a variable called 'b' equal to four and it adds the values of a and b and sets c equal to the sum, 9. In other words, the Spectrum has been thinking about it all without writing anything.

And almost all BASIC computing is like that: there are instruction that make the computer *think* and other instructions that make it *show* you what it has thought. In this book, we're most concerned with the thinking part, because, as Programs 1Ø and 1Øa showed, if you want pretty displays, you've got to make sure the Spectrum doesn't try to draw off its own screen or plot a letter on the other side of the room! Program 1Ø gives nice circles, but 1Øa is far more powerful as far as being a calculating aid is concerned.

But don't worry! We shall have both!

## 3 Equations

Albegra: what a horrible name! But what a powerful idea. It was algebra that built the pyramids and St Paul's Cathedral. It was algebra that put those men on the moon.

My dictionary is interesting on algebra. It says that the word comes from the Arabic (another race famous for their mathematics, the Arabs) words, *al* meaning the and *jabr* meaning to fix (more or less). So algebra is the fixer.

Most people have a hazy notion that algebra is arithmetic with letters instead of numbers. But that idea is misleading, because there doesn't seem to be any way you can do arithmetic with letters. I grant you, though, it does look like arithmetic with letters.

An equation is, if you like, one of those weighing scales you see on a zodiac sign for Libra. You know the sort: a couple of dishes hanging on the ends of a crossbar. Put something in the right-hand pan and some weights in the left-hand pan, and when the two weights are equal, you have your answer.

With equations, the same applies. You have something on the left-hand side, something on the right-hand side and between



them is an equals sign (which looks like this: =) meaning that what is on the left equals what is on the right. And that's all an equation is.

Algebra is concerned with tinkering around with equations to get a useful result and has rules that help you to alter equations to suit yourself.

Remember page 23? We had a couple of equations for the circle. Remember C=PI*d relates the distance round a circle to the distance across it. We could write:

Circumference=PI*diameter

but it's better shorthand to write:

C=PI*d

So there we are. We know the value of PI: it's 3.14159. . . etc. We only need to know the diameter and we can calculate the circumference. Do you see how it works?

By getting to know what the right-hand total is, we can say that *since it is an equation the left-hand must equal the right-hand side,* and we have our answer.

We try to arrange equations so that the thing we want to find or calculate is on its own on the left-hand side and all the other constants and variables are on the right-hand side. But what if, in the example above, we knew the circumference and wanted to calculate the diameter? That is why we have to rearrange the equations.

Now, we've decided that we want to know the diameter d and that we already know the circumference C. PI, of course, is a constant, and the Spectrum remembers its value even if we don't. So let's start off with understanding that we need to get d on its own on the left-hand side. (I'll abbreviate 'right-hand side' to RHS and 'left-hand side' to LHS.)

Rules for algebraic rearrangement

RULE 1: You can make an LHS into an RHS and vice-versa. That is, you can swap sides

(So we can make C=PI*d into PI*d=C, just swapping RHS and LHS.)

RULE 2: You can do anything to one side so long as you do it to the other side as well.

This means that you can *add* anything to both sides, *subtract* anything from both sides, *multiply* both sides by anything, *divide* both sides by anything. But take care. If you do it to one side, you must do *exactly the same* to the other side.

And it's no good just deciding, say, to add 5 to both sides and divide both sides by 14.8. You'e got to look carefully at the equation, bearing in mind what your aim is, and choose to do something that helps you achieve your aim.

In our example we have now got, PI*d=C, so that our aim of getting d alone on the left is only one step away. We can see that PI times d on the left could be made into d alone by dividing the LHS by PI. (Think about that carefully. What I'm saying is that PI*d/PI is really the same as d on its own, because it's taking d and multiplying it by PI and then dividing it by PI, and the two operations cancel one another out.)

But we must do the same to the RHS too. Giving:

PI*d/PI=C/PI

and because the LHS is really the same as d alone, we can write:

d=C/PI

And that's it! We've achieved our aim of getting d alone on the LHS.

Now let's step back a couple of paces and have a good look at exactly what it is we've achieved. What does d=C/PI mean?

It contains the same information as the equation we started out with, the same constant PI, and the same two variables d and C. We've just changed it round so that if we know the circumference we can calculate the diameter.

So let's try it out. Back to the Spectrum.

*Program 11 REARRANGEMENT*

```
5 REM PROGRAM 11
  MASSAGE
10 INPUT "Enter diameter ";d
20 LET C=PI*d
```



```
30 PRINT "Circumference is ";C
40 INPUT "Enter circumference
";C
50 LET d=C/PI
60 PRINT "Diameter is ";d
```

Line 10 asks you to choose any diameter. Line 20 uses our original equation to calculate the circumference, and line 30 prints it out for you to see. Then line 40 asks you to input a circumference, and you should read the one you have just calculated and whose value is printed at the top of the screen. If you enter that one, line 50 will do the reverse of what line 20 did, and you should get back to the diameter you started with. That way we prove that the line 50 equation (the one we manipulated) is really doing what we want it to do.

Run it a few times, but remember the diameter you used and compare it with the diameter it gives you back.

If you do it often enough you will find that sometimes the Spectrum gives you back a diameter which is not exactly equal to the one you started with. Can you see why this should be? It's not that our equation is a bit inaccurate. It's that the Spectrum can't use the actual value of PI, but only the one rounded off to eight decimal places. If you look at the error involved, though, it is really tiny, and unless you're going to use your Spectrum for astronomy or nuclear physics, it probably won't worry you too much.

Before we leave the subject, we really ought to have a look at a few more of these manipulations. Stick to Rules 1 and 2, and you won't go far wrong. Remember that adding and subtracting are opposites, and multiplying and dividing are opposites. Remember also that any number divided by itself is 1 (PI/PI for example is 1, and you can try it with other numbers,) and any number minus itself is zero. (PI − PI = 0)

Try a few of these with a pencil and paper:

If x = a + b + c, get b as the subject of the equation. (Subject just means the one on its own on the LHS.)

This is it in full:

x is the subject of the equation at present, let's write it out as, a + b + c = x

If we now subtract a from both sides we get,
a−a+b+c=x−a
and a−a is zero, so we have,
b+c=x−a
If we now do the same subtracting c from both sides,
b+c−c=x−a−c
and since c−c=0, we have,
b=x−a−c

And that's our result.

It's odd, but for ages after learning how to do algebra to the point we've now reached, being able to manipulate equations and so on, I had a kind of mental block over mathematical equations found in books. It was as if I'd been reading a book, and suddenly there was a line of Arabic, just some utterly unknown language I couldn't begin to read, which was silly because I could have read it to myself if I'd tried. I just seemed to skip over it, and so it took me a long time to get into the language of maths.

So I advise you to read equations when you get to them. Pronounce them to yourself just like you might pronounce a line of poetry to yourself. If, for example, you read:

$$x=\frac{(2\star a)+((5\star a\star b)/c)}{(c-d)}$$

it looks like a muddle of marks on the paper and it's easy for your eye to tell your brain that it has found a mess of barbed wire and you ought to miss it out. But try to read it; it sounds like this:

'x equals two times a . . .
plus . . .
five times a times b over c . . .
all over c minus d'

Notice how I've had to present the 'poem' in several lines in order to break down the parts of the right-hand side so that it means something unique. The real equation itself is broken down that way too, by using brackets. The brackets are like little parcels of numbers and variables that must be worked out as a group first.

It's a great idea because it sorts things out and makes the equation a lot easier to work out. If we represent the first bracket



in the above equation by a letter A, the second one by B and the third by C, it would simplify matters quite a lot:

If A =(2*a)
and B=((5*a*b)/c)
and C=(c−d)

then we could substitute the letters for the brackets and write:

$$x=\frac{A+B}{C}$$

which gives you the general idea.

We'll be coming back to this bracket business later, because it is really important for the Spectrum to get the brackets entered correctly. If you're not convinced of that and are inclined to yawn a bit and say it's all a bit too technical, just take a look at this example:

Enter this in your Spectrum

```
10 LET a=2: LET b=3: LET c=4
20 PRINT a*b+c
```

You ought to get back 10. Then add,

```
30 PRINT (a*b)+c
40 PRINT a*(b+c)
```

And you'll find line 30 gives you 10 as before, but line 40 gives you back 14.

Same variables, same order, same multiply and add signs, but a different result! I told you it was important. If you evaluate (that means 'work out') the two expressions (that means 'combination of variables') you will see why. The one in line 30 is (2 times 3), that is 6 . . . plus 4 . . . which is 10.

The line 40 expression is 2 times (3 plus 4), or 2 times 7, which is 14.

You might ask yourself why line 20 gave 10 as a result instead of 14. Why indeed. It's actually because the Spectrum has an inbuilt order for working out expressions. It will evaluate the multiplys and divides *before* it evaluates the adds and subtracts. Unless of course you tell it you want it to work out an expression in a certain way by using brackets, like we did in line 40. This idea of some operations being carried out before other types of operation is called priority. Before discussing it properly, though, we have to get a few other things under our belts.

Let's look at a couple of other things. On my list I have a note to talk about FOR . . . NEXT loops again, and I want to do so because they're devices that automate program operations. They make programs easier to use, and it's a good idea to get to know them.

There are four instructions in Spectrum BASIC used with FOR . . . NEXT loops: FOR, TO, STEP and NEXT.

The purpose of a FOR . . . NEXT loop is to stop the program going just from top to bottom and to make it loop back over itself several times. If that's not very clear, it's a quite simple thing to understand on the machine itself, and at the risk of boring you folks who read Chapter 1, here's a simple example:

*Program 12 SIMPLE FOR . . . NEXT LOOP*

```
 5 REM PROGRAM 12
   SIMPLE FOR LOOP
10 FOR N=1 TO 10
20 PRINT "Loop number ";n
30 NEXT n
```

Run it and see what happens. Line 10 tells the Spectrum how many times to go round the loop. You can change the values that the loop variable n starts and ends at. (Try it! Experiment with different numbers than 1 and 10.)

Line 20 is the part of the program that is being looped through. It happens to contain a PRINT instruction to help you see what's happening, but it could contain just about anything. Line 30 is the NEXT instruction. It serves to send the program-watching Spectrum back to line 10 where the FOR instruction is to be found and make n take its next value (in this case 2).

But what if you'd wanted to count the number down from ten to one, instead of up from one to ten?

No problem: you just use STEP. The number associated with STEP tells the Spectrum how many to add to or subtract from n each time it goes around the loop. So that if you make line 10 of our Program 12 into:

```
10 FOR n=10 TO 1 STEP -1
```



and you will find the numbers do indeed run from 10 to 1.

Experiment! Use different values of STEP variable. Get a feel for how the FOR . . . NEXT loop works. Try using small decimals like 1.5, or expressions like 2*5/3.

You will find that for a FOR . . . NEXT loop to work, the number before the TO must be smaller than that after the TO when the STEP variable is positive, and it must be the bigger number first if the STEP variable is negative. That makes sense, doesn't it.

From our point of view, it is going to be a lot easier using FOR . . . NEXT loops to take the drudgery out of some of the more complex equations we're going to get on to later.

Have you ever seen a nest of tables? It's a set of three or four coffee tables each one slightly smaller than the last so that you can stack them inside one another.

FOR . . . NEXT loops can nest as well.

You can fit FOR . . . NEXT loops one inside another like this:

*Program 13 RANK AND FILE*

```
 5 REM PROGRAM 13
   RANK AND FILE
10 FOR x=8 TO 248 STEP 8
20 FOR y=8 TO 168 STEP 8
30 PLOT x,y
40 NEXT y
50 NEXT x
```

You see how the y loop is entirely inside the x loop. If you swapped lines 40 and 50 into:

```
40 NEXT x
50 NEXT y
```

it wouldn't work.

Get DIAGONALS by changing line 30 into:

```
30 PLOT x,y: DRAW 7,7
```

or more interestingly,

```
30 PLOT x,y: DRAW 7,0: DRAW 0,7: DRAW -7,0:
DRAW 0,-7
```

to get some LADDERS.

Now go back to Program 13 in its original form, run it and count the dots made in each vertical line. That tells you how many times it goes round each y loop for each x loop. If you count the number of dots horizontally you will find the number of times it passes round the x loop. And it does it all in five seconds, which is much faster than you trying to specify each plot individually yourself.

The use of PLOT and DRAW in FOR . . . NEXT loops forms the basis of the graph plotting we'll be meeting later, and that really is quite spectacular.

# 3

# OPERATORS AND POWERS

## 1 Operators

What are operators?

We've already met them as a matter of fact. Try these in your head:

3+4=
2−1=
5*3=
6/2=

If you got 7, 1, 15 and 3, you certainly know your operators.

And that's really all there is to it. That's really all there is to maths in general. The universe comes down to chemistry, and chemistry comes down to elements, of which there are some ninety odd. Maths is based on the number system and four basic operators, and everything comes down to them in the end.

The famous four are:

Add +
Subtract −
Multiply * (or × if you are not into computers)
Divide / (or ÷ sometimes)

I doubt whether you've lived as long as you have without doing a few operations yourself sometime.

I might just have mentioned in Chapter 2 somewhere that add and subtract are opposites. (Surely this must be so, because if you take any number and add any other number to it, then subtract the same number, you get back to the number you first thought of!)

Quite true: $8+7=15$ and $15-7=8$, and it works for every pair of numbers you care to try.

The same is true of the operators * and /. There are two pairs of operators then, and the pairs are:

Add Subtract
and
Multiply Divide.

We've established that the relationships between add and subtract are that one is the inverse of the other – meaning that if you do one and then the other, you get back to where you started. You can establish for yourself that multiply and divide are inverses of one another.

But what about the relationship between, say, add and multiply? Is there a relationship?

Yes, I suppose you could say that multiplication is just a way of adding lots of times repeatedly. For example, I could say 5*4=20 (5 4s are 20, or 5 times 4 equals 20). But it would be the same as saying 4 groups of 5. And we could arrive at the answer by doing:

5+5+5+5

that is, 5 added together 4 times. And that's where the word times comes from when we say 5 times 4.

Now if we can reduce all mathematics to four operators, and we can reduce those four operators to two pairs of inverse operations, and we know that of those pairs multiplication is just a kind of repeated addition, then . . . all mathematics is just a matter of one thing: adding.

And since you already know about addition, we should have no trouble with the rest of it, should we?

Try this:

Is a+b the same as b+a?

or, in symbols,

Is a+b=b+a true?

The answer is, generally, Yes, although (believe it or not) there are some quantities that that can't be said of. But they're not ordinary numbers like we're used to, and we can leave that problem until later.



What about this, then:

Is a*b=b*a?

Again, generally speaking the answer is Yes, so it doesn't matter which way round we write the sum or product. ('Sum' means numbers added together, 'product' means numbers multiplied together.)

What about this, though?

Is a−b=b−a true?

Try it out with a few numbers for a and b. And you can soon see that it would not be possible to replace a−b with b−a in an equation, because they're not equal.

Then again, could we write:
a/b=b/a?

Nothing stops us writing it, but it isn't true!

Division is one of those peculiar inverse operations, and you can't replace a/b with b/a. Try it with a=5 and b=2, or any other pair of numbers you like. (With 5 and 2, a/b=2.5 and b/a=0.4)

In fact division can be a real pain in the neck to computer users, because it's often the source of a nasty little bug that will crash a program with the report Number too big.

The Spectrum will only go so far along the Real Line. (Naturally, it would have a job going all the way along it because it's infinitely long!) In fact the biggest number the Spectrum will handle is around 100,000,000,000,000,000,000,000,000,000,000,000,000. Not bad, you might think, but it's surprising how easy it is to exceed that limit with calculations involving probability (which we'll have a look at later on).

Consider this, though.

Take the number 1, and divide it (on your Spectrum) by 1. Command it to

```
PRINT 1/1
```

And it will return 1.

Then decrease the denominator by a factor of ten,

```
PRINT 1/0.1
```

and you get 10 returned.

Now try,

PRINT 1/0.01

and you have 100. There you can see the denominator getting smaller and smaller, and the result getting bigger and bigger. If you were to carry on reducing the denominator by a factor of ten (that just means in practice slipping in an extra 0 after the decimal point each time), you will find that on commanding your Spectrum to

PRINT 1/0.00000001

it will give you back 1E+8. And I guarantee that, if you don't know what that means now, you will after reading the rest of this chapter.

Eventually, you will find that the Spectrum has given up giving you back reasonable numbers like 10 or 100 or even 1000000 (one million) and has automatically converted to giving answers in a condensed way. 1E+8 is an example of that. It's called 'scientific notation' because scientists often use it. It's sensible, in fact, because it saves cramp when writing out a string of zeros, or eyestrain (or brainstrain more like!) when trying to read it. The number quoted above with thirty-eight zeros is a good example. Far better to write 'one followed by thirty-eight zeros', and better still to abbreviate that to '1E+38'. More about that later. For now, I must get to the point I was trying to make all along: that as you make a denominator smaller and smaller, the result gets bigger and bigger. And if you try to make the denominator zero you are in big trouble, because the result will be too big for the Spectrum to handle. (It doesn't handle infinity, and neither as mathematicians can we. The mind boggles!)

*So, we have to avoid any situation where we might divide by zero when programming.*

*Summary*

1. There are four mathematical operators to remember: +, −, *, /
2. a+b=b+a and a*b=b*a
3. But a/b does not=b/a and a−b does not=b−a
4. There is no way of handling a fraction whose denominator is zero
5. A shorthand system of writing very large numbers exists called 'scientific notation'



It might be useful to add a note about abbreviation as used by mathematicians and in mathematical texts. You will see products written without any multiplication operator between them. So that, to use an earlier example,

C=2*PI*R becomes $C=2\pi R$

Whenever the operator is absent you can be sure that you are meant to multiply the variables together.

And you would read it 'C equals two pi r'.

Remember the formula for the area of a circle?

Let me refresh your memory:

A=PI*R*R

That was how we remembered it. A mathematician might not recognise it as we've written it, at least not immediately. That is because mathematicians (famous for their shorthand) have the habit of replacing PI with their favourite little Greek letter and leaving out the multiplication operators. Moreover they collect all the similar factors together and write a number above it to remind themselves how many they've collected. So they end up with:

$A=\pi R^2$

And they read that 'A equals pi r squared'.

And that brings us to section 2.

## 2 Powers

This business of multiplying and adding is worth thinking about for a second or two. If we can write:

A=B+B+B+B as A=4*B

why can't we have a shorthand for,

A=B*B*B*B

Well, we can:

$A=B^4$

But do you see how we have to write the 4 above and to the right of the B? That causes people who write books problems, and it is pretty inconvenient for Spectrum users too, and to get around it, computer scientists came up with a flashy little symbol that says it all:

↑ (found on the H key)

It is an 'up arrow' and is easy to remember because it gives you the clue by pointing up to where the little number might have been. So now we would write:

A=B↑4

which is read as 'A equals B to the power four'.

Remember that: 'to the power'.

Here's a few for you to have a crack at. Try them on the computer. Enter,

PRINT 2↑3

You will get back 8 as your result, and this is so because 2↑3 really only means 2*2*2 (2 multiplied by itself 3 times). Try Program 14.

### *Program 14 POWERS OF TWO*

```
 5 REM PROGRAM 14
   POWERS OF TWO
10 FOR p=0 TO 10
20 PRINT 2^p
30 NEXT p
```

Try replacing line 20 with: 20 PRINT "2 to the power ";p;" equals ";2↑p

This gives a better view of what's going on. 2 to the power 0 is 1, 2 to the power 1 is 2, 2 to the power 2 is 4, and so on.

If you incorporate a FOR . . . NEXT loop to draw a graph as it does its calculating, your Spectrum will show you how the results rise dramatically. By the time p has reached 7, and the string of 7 2s gives a result of 128, we have reached the limit of our graph plotting capability (with program 14a that is!) because p=8 means that 2↑p=256 and our lines are out of range.



### *Program 14a GRAPH OF POWERS OF TWO*

```
 5 REM PROGRAM 14a
   GRAPH OF POWERS OF TWO
10 FOR p=0 TO 8
20 PRINT "2^";p;" is ";2^p
30 FOR n=0 TO 40
40 PLOT n+128,0: DRAW 0,2^p
50 NEXT n
60 NEXT p
```

Which ought to give you a 'feel' for the way powers of numbers soon take-off, growing very large very quickly.

Now go back to simple program 14, and, in line 20 instead of putting 2↑p, replace the 2 with 3. Run it. The results are growing even faster now, because you are looking at powers of three. Now try it with 10 instead, so that you are looking at powers of ten:

20 PRINT 10↑p

This is an interesting result because our old friend the scientific notation has reappeared. (Because the numbers have grown as big as 100 million, and that's where the Spectrum gets bored with writing the number out wholesale.)

Try,

20 PRINT 9↑p

And you can see that the last two values reached are in scientific notation. If you look closely at the first of them, you will be able to see that it's in two parts. It might be difficult to read it, but in fact the 3.8742049 is just an ordinary decimal number and the E+8 part means 'times 10 to the power 8'. '10 to the power 8' is just a 1 with 8 zeros after it: 100,000,000. (Unlike computers, people often write really big numbers with commas in so they can keep track better. Don't do that on your Spectrum: it won't understand at all!)

So we can write the whole thing out as:

3.8742049*100000000

or,

387420490

or, in English,

387,420,490

or even more in English,

Three hundred and eighty seven million four hundred and twenty thousand four hundred and ninety.

What a saving in effort 3.8742049E+8 turned out to be!

But actually the Spectrum wasn't giving us exactly the right answer. Because it only quoted the number to seven decimal places, it couldn't handle the whole number, which ought to have been 387,420,489. I know, because I have a calculator that works to nine decimal places!

It may worry you that you are bound to lose some precision by quoting numbers to a limited number of decimal places. But really it's better to say that the number is accurate to a millionth of one percent, and scientists rarely manage that sort of precision in their data anyway.

So where does that leave us?

So far we have considered powers which are positive integers. What about other sorts of number? It can be done. Try raising 9 to the power 1.5 by entering,

PRINT 9↑1.5

and you get 27. Which is really very surprising indeed, since we have defined powers as repeated multiplications, and how can you multiply 9 by itself 1½ times?

Let's take a new section to see how it works.

## 3 Powers, Episode Two

Remember our circle? Long ago we talked about its area and its circumference (circumference being the length round the outside), and we had:

C=2*PI*R (where R is the radius)

Could you, here and now, build up an equation for the perimeter of a square? The perimeter of a square is the line round the outside. It is probably true to say that a perimeter is to a square what a circumference is to a circle.



Have you thought what you need? First you need a variable for the perimeter, and P seems a reasonable choice. Then you have to decide what you want it in terms of. Suppose we want it in terms of the length of one side, which we can call s.

Then, we have to think what makes a square a square and not a triangle or a rectangle. Just that it has four sides, all the sides are the same length and each corner is a right angle.

From these things we can say that,

P=s+s+s+s

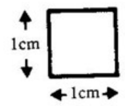
or,

P=4*s

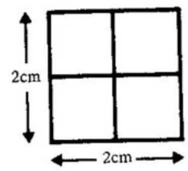
And there we have it! We have successfully developed a formula for the perimeter of a square.

What about area? You did it for a circle: what about the area of a square?

Take the square which has sides 1 centimetre long; mathematicians might use the phrases 'a 1-centimetre square' or 'a square of side 1 centimetre' to describe it. It looks like this:




If we are going to make a square by stacking these 1cm squares together, we can't do it with two or with three. We need four:




And the next biggest square is going to have nine squares in it.

Now comes the clever part. If our first square, 1cm on a side, has 1 unit of area, then a square of side 2cm has an area of 4 units. And a square of side 3cm has an area of 9 units.

Suppose we have 6 1cm squares stacked so that there are 2 rows of 3. Then the area of the rectangle (oblong) you get is 6 units, the length of the rectangle is 3cm and its height is 2cm.

Can you see what's happening? *To get the areas we're just multiplying the length by the height.*

When you look at the general formula for a rectangle, it can be written:

A=L*H (Rectangle)

where A is the area, L is length and H is height. The squares are a special sort of rectangle, though. We know they have a length and a height which are equal, and so we can write:

A=s*s (Square)

where s is the length of a side. And we can write:

A=s↑2

And that's why they call any number raised to the power two a 'square'. Program 15 will generate squares. You can enter the length of a side, and it will tell you the area. It also handles units to your taste. But you can enter any kind of unit you like, so if you enter 'eggs' for the unit of length, you will get 'square eggs' for the area! Better to stick to metres, inches, etc., I think.

*Program 15 SQUARES*

```
  5 REM PROGRAM 15
    SQUARES
 10 INPUT "Enter length of side
";s
 20 INPUT "Enter unit of length
";s$
 30 PRINT s;" ";s$;" square"
 40 LET A=s^2
 50 PRINT "has area ";A;" squar
e ";s$
 60 PAUSE 300: GO TO 10
```

Use the SHIFT CAPS and BREAK key to get out of this program.



By the same token, think of cubes. Cubes are three-dimensional squares, squares with depth, and they have length, height and depth all equal.

The volume (meaning the block of space inside the object) of a cube is going to be the product of height, length and depth, which for any old 3D oblong is:

V=L*H*D

where V is the volume, L the length, H the height and D the depth. For all cubes, we said that L=H=D, and we can call it s (for side), so that:

V=s*s*s

or,

V=s↑3

And we can write Program 16 to calculate cubes. As with the case of squares, any number raised to the power three is called a cube.

*Program 16 CUBES*

```
  5 REM PROGRAM 16
    CUBES
 10 INPUT "Enter length of side
";s
 20 INPUT "Enter unit of length
";s$
 30 PRINT s;" ";s$;" cube"
 40 LET V=s^3
 50 PRINT "has volume ";V;" cub
ic ";s$
 60 PAUSE 300: GO TO 10
```

Remember the circle? If a 3D square is a cube, a 3D circle is a sphere.

Spheres enclose space much as cubes do (like all 3D shapes do) and so a sphere has volume. What's the formula for a sphere's volume?

I have no intention of developing it for you. I'll just quote it instead:

V=(4*PI*R↑3)/3 (where R is the radius)

or, in mathematical form,

$$V=\frac{4\pi R^3}{3}$$

The volumes of all solids, as 3D shapes are called, are measured in cubic centimetres or cubic inches. They don't have to be cubes.

Interestingly, if we look at what we've been doing in general terms, we've just looked at the instruction:

PRINT X↑Y

and we've seen that X can be any kind of positive number. What if X is a negative number?

Try this on the Spectrum:

*Program 17 INVALID ARGUMENT DEMO*

```
 5 REM PROGRAM 17
   INVALID ARGUMENT DEMO
10 FOR N=5 TO -5 STEP -1
20 PRINT N^2
30 NEXT N
```

It goes all right with the positive values of n, but as soon as the n steps down to −1, the Spectrum objects.

So *don't try to raise a negative number to a power on the Spectrum.*

Although the Spectrum doesn't like it, a negative number can be squared or raised to any other power. The point to remember is that if the power you are raising a negative number to is *even* (2, 4, 6, etc.), then the result is positive, but if the power is *odd* then the result is negative. This is so, because pairs of negatives cancel one another out. But when you have an odd number of items in the product, there's one left over after all pairs have cancelled out, and it makes the result negative.

Which brings us back to peculiar kinds of Y in X↑Y. We could have numbers other than positive integers. Let's take positive fractions. What is the meaning of X↑(1/2) for example.

Try this:

*Program 18 ROOTS*

```
 5 REM PROGRAM 18
   ROOTS
10 FOR n=1 TO 10
20 LET s=n^2
30 LET r=s^(1/2)
40 PRINT n;TAB 10;s;TAB 20;r
50 NEXT n
```

This program gives some insight into the meaning of a fractional power. Line 20 gives a variable s the value of n squared. Then we raise s to the power of (1/2) in the next line. And when you run this program, you'll find that raising a number to the power 2 and then raising it to ½ brings you back to the number you started with. Does that process ring a bell? Doing some operation and then another operation and getting back to where you started?

It means that you've just done something and then done its inverse. So if raising to the power 2 and raising to the power ½ are inverses of one another, can we make a general rule?

Yes, we can. In fact, X↑Y has an inverse X↑(1/Y). Try it out yourself with Program 18, making the 2 in line 20 and the 2 in line 30 into say 3 or 5 or something.

The name of the inverse process to raising to a power is called 'taking a root'. And the power (1/2) is called a square root, and the power (1/3) is called a cube root.

For instance, with the area of a square, suppose you had a square and you knew its area was 100 sq. cm. Then you could calculate the length of its side, because we already have that

A=s↑2

and inversely,

s=A↑(1/2)

(Remember that you can do the same process to both sides of an equation.) So in our case s = 100↑(1/2)

Check it on the Spectrum with

PRINT 100↑0.5

The answer is 10, and you can check it by raising 10 to the power 2.

Suppose we take a look at our original example now: 9↑1.5

When you instruct your Spectrum to PRINT 9↑1.5, it will return 27. Can you see why?

It makes it much easier if we write 1.5 as 3/2, because the two processes at work here are made more visible. (Just satisfy yourself that 1.5 is equal to 3/2 first.)

At one and the same time we are cubing 9 and then taking its square root: that is, raising it to the power 3 and then raising that to the power (1/2).

If you are not crystal clear on that, try the following:

PRINT 9↑(3/2)

And then

```
10 PRINT 9↑3
20 PRINT 729↑(1/2)
```

and you can see the secret laid bare.

This subject is often found in maths books under the heading 'indices' (pronounced in–di–sees) because the power number is called an 'index' and, being Latin, the plural is indices. (Later we'll be meeting another Latin term ending in x, matrix, and the plural of that is matrices.)

So it is with decimal indices. Try to think of them as fractions in which the numerator raises to a power and the denominator takes a root. Program 19 runs through a few possibilities.

*Program 19 FRACTIONAL INDICES*

```
 5 REM PROGRAM 19
   FRACTIONAL INDICES
10 INPUT "Enter X ";X
20 INPUT "Enter a ";a
30 INPUT "Enter b ";b
```



```
40 LET Y=a/b
50 LET R=X^Y
60 PRINT X;" to the power (";a
;"/";b;") is ",R
```

And you can make the numbers as involved as you like.

There remains one major category of indices we've not looked at yet: negative indices.

What does 2↑(−3) mean?

We can show its value is 0.125, and because I'm used to numbers I happen to recognise that as 1/8. Which is interesting because 2↑3 is 8. Is it possible that in general,

X↑(−Y) = 1/(X↑Y)

Indeed it is. In general *X to the power minus Y equals one over X to the power Y.*

It might be possible to put together a program that demonstrates this rule:

*Program 20 NEGATIVE INDICES*

```
 5 REM PROGRAM 20
   NEGATIVE INDICES
10 INPUT "Enter X ";X
20 INPUT "Enter Y ";y
30 LET L=x^(-y)
40 LET R=1/(x^y)
50 PRINT L,R
```

If you like you can automate the program:

*Program 20a AUTONEGATIVE INDICES*

```
 5 REM PROGRAM 20a
   AUTONEGATIVE INDICES
10 LET x=RND*100
20 LET y=RND*10
30 LET L=x^(-y)
40 LET R=1/(x^y)
```

```
50 PRINT L,R
60 GO TO 10
```

And the pairs of numbers you generate are identical. Although the electronic routine in the Spectrum's ROM that deals with powers is slow (very slow!), it nevertheless seems to be accurate.

## 4 Scientific Notation

We're now in a position to take a full look at the idea of scientific notation:

*Program 21 POWERS OF TEN (COMPLETE)*

```
 5 REM PROGRAM 21
   POWERS OF TEN (COMPLETE)
10 FOR p=-10 TO 10
20 PRINT 10^p,p
30 NEXT p
```

It is just an extension of Program 14 and throws up some interesting points. It shows that scientific notation can deal with very small numbers as well as very large ones. That's what the + and − signs are all about. The results also bring to light a small bug in the Spectrum ROM. Run the program and see what I mean.

The first point to note is that 10↑0=1. (In fact any number raised to the power 0 is defined as 1.)

The second point to note is that 10↑1 is 10. (In fact any number raised to the power 1 is itself: X↑1=X.) Perhaps you could write a small program to demonstrate those two points.

The next point to note is that when the number of zeros gets beyond seven, scientific notation kicks in and we get numbers like 1E+8 (which reads '1 times 10 to the power 8'). Also, as the numbers go smaller than 0.00001 we get scientific notation coming in with numbers like 1E−6 which is read '1 times 10 to the minus 6' or zero point zero zero zero zero zero one, six zeros in all, one of them before the decimal point and five of them after it.

And finally, the bug. I think that 10↑−10 should be 1E−10. The Spectrum gives 9.9999999E−11. This is very nearly



10E−11 which is equal to 1E−10, but it isn't exact. I think something must have happened with rounding in the ROM. It's not a very serious problem, though.

By the way, the letter E stands for Exponent. Exponent is just another word for index. There is a word for the decimal part of a scientific notation number too: mantissa. So that in the number 6.397E−5, the exponent is −5 and the mantissa is 6.397.

This is the way the Spectrum does it, but if you look at a physics or astronomy book and find some large numbers, or look elsewhere for very small numbers, you'll see them written in this form:

$2.35\times10^{6}$ 2,350,000)
$1.42\times10^{-4}$ (0.000142)

which makes it very much clearer.

As an added extra, the Spectrum has a special function for giving square roots. It is called SQR and can be found on the green label above the H key. Test it out with this program:

*Program 22 SQUARE ROOTS*

```
 5 REM PROGRAM 22
   SQUARE ROOTS
10 LET X=INT (RND*100)
20 LET Y=X^.5
30 LET Z=SQR X
40 PRINT Y,Z
50 GO TO 10
```

All we are doing here is comparing the square roots generated by the two different expressions in lines 20 and 30. You would expect the two columns to be identical, and so they are. So we must have been right about all that fractional index stuff.

SQR is not happy about negative numbers, and in your use of this function you must be careful not to try to use it on one. Can you see why this restriction should be?

Try to think in terms of the square and the square root as being inverses of one another. If you start with a number that is a square of another number, say 100 which is the square of 10, then the 10 is the square root of 100. So far so good. But consider this: 10 is the number which when multiplied by itself gives 100, but there is another way of multiplying a number by itself to get 100: −10*−10=100 (because the negatives form a self-cancelling pair). So the answer to SQR 100 is really a double answer, 10 and also −10. (This is the doorway to a whole branch of mathematics called complex numbers, and we can investigate that later.)

So, since (−10)↑2 is actually 100 (though your Spectrum won't deal with it), what number when squared gives −100? Well, without starting a discussion on complex numbers, I'd have to say there isn't any such number. The very act of multiplying a number by itself if creating a self-cancelling pair, so that the two minus signs are also cancelled.

Program 23 is intended to give you a feel for the SQR function, drawing a graph of the square roots of the integers from 0 to 255.

*Program 23 ROOT GRAPH*

```
  5 REM PROGRAM 23
    ROOT GRAPH
 10 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175
 20 FOR N=0 TO 255
 30 PLOT N,0: DRAW 0,(SQR N)*10
 40 NEXT N
```

Line 10 draws a border line, the FOR . . . NEXT loop draws the graph, and the number 10 is in line 30 to enlarge the graph so it fits the screen better. Try leaving it out, or use another figure as scale factor instead.

As you can see the graph rises quickly at first and then more slowly as the numbers get bigger, just the opposite of what happened when we graphed out the powers of 2 in Program 14a. It's just what we might have expected, because powers and roots are inverses.

The next program rather obligingly draws out the graphs of a whole host of roots. Not just square roots, but cube roots, fourth roots and so on, and you can see that as the Y in X↑(1/Y) increases, the result is an increasingly flat graph.



*Program 24 ASSORTED ROOTS*

```
  5 REM PROGRAM 24
    ASSORTED ROOTS
 10 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175
 20 FOR Y=2 TO 6
 30 PRINT AT 1,1;"Graph of ";Y;
"th root   "
 40 IF Y=2 THEN  PRINT AT 1,9;"
 square root"
 50 IF Y=3 THEN  PRINT AT 1,9;"
 cube root"
 60 FOR x=0 TO 175
 70 PLOT x,(x^(1/y))*13
 80 PRINT AT 3,6;"           ":
PRINT AT 4,6;"           "
 90 PRINT AT 3,6;x: PRINT AT 4,
6;x^(1/Y)
100 NEXT X: NEXT Y
```

The values are flashed up (literally!) as the graph draws itself across the square. The scale factor (to make the graph fit the screen better) is in line 70. I've used 13, but you can try other values if you like.

It's only when you see a program like this one (slow though it is) that you realise just what a saving in time is achieved by computers. If you could take the time to do with pen and paper and a book of tables (and I doubt whether you could find sixth-root tables) what Program 24 just did for you, you would see what I mean.

Take a look on the back of a 1984 pound note and see Sir Isaac Newton.

I'll let you into a secret about Isaac. He was, you might know, perhaps the greatest scientist who ever lived. But did you know that, although he laid the foundations of optics, dynamics and mathematics during the course of his life, he failed a scholarship examination in 1663 (when he was 21 years old) due to 'woeful inadequacy in geometry'!

Imagine what Newton could have done with a Spectrum. If

Cambridge had had Sir Clive Sinclair around in 1663, they might have shared the back of a five pound note!

*The problem of the Chinese typewriter*

Before we leave this subject, I'd better mention something. Firstly, you haven't heard the last of this powers and roots business, and secondly because it was Isaac Newton and people several centuries ago who dreamed up the language of mathematics we are stuck with a small problem. Because Chinese came first and the typewriter came only in the nineteenth century, Chinese typewriters are not very practical. They don't fit the language, and with six thousand separate symbols in use in everyday language, you have to have a good memory and good eyesight to find the right key!

So it was with maths. I'm writing this book on a typewriter. It does numbers and a percent sign, but it can't deal with mathematical symbols. I have to break off every now and then and write in a symbol by hand. Unfortunately, this technique is not much good with the Spectrum. It's not much good getting your felt-tip out and scribbling on the TV screen, and so computers have their own methods of describing mathematical expressions. But what about books?

The square root sign is $\sqrt{x}$ (meaning square root of x) and is very common in maths texts. Often it is used over a whole expression like $\sqrt{ab-bc}$ (meaning root of (ab − bc)). And if you want different kinds of roots, a small number is hung over the beginning of the root sign to indicate the kind you want. So that $\sqrt[3]{y}$ means cube root of y, and $\sqrt[6]{z}$ means 6th root of z.

So there we are. If you have been thrown by all this talk of powers and roots, it's not as haphazard as it seems. In any case squares and square roots are quite common, cubes and cube roots less common by far, and the other kinds quite rarely met with. So don't despair.

Have you begun to get used to equations now? I hope so. As I say, it's a weighing scale that's in balance; one side equals the other side. But there are non-equations too, things that are not in balance, where one side is not equal to the other.

They're called inequalities (not surprisingly), and you will have to wait until the next chapter to learn about them.

# 4

# INEQUALITIES AND FUNCTIONS

## 1 Inequalities

Equalities (otherwise known as equations) are shown by a sign like this: =. So a = b means a equals b.

On the Spectrum you will find five inequalities:

| | | |
|---|---|---|
| > | meaning greater than | ($>$) |
| < | meaning less than | ($<$) |
| >= | meaning greater than or equal to | ($\geqslant$) |
| <= | meaning less than or equal to | ($\leqslant$) |
| <> | meaning not equal to | ($\neq$) |

The symbols in brackets are the symbols you would find in maths books (Chinese typewriter, remember!), and you will find that there is a subtle difference in meaning between a computer-type inequality and a maths-type inequality.

The use you put an inequality to is the important thing. An equals sign (=) is used in two ways, and this illustrates the distinction nicely. You can either use an equals sign to define a relationship, as you do on your Spectrum when you use a LET statement:

```
LET a=b+c
```

defines the variable a in terms of b and c.

Or, you can use it with an IF statement:

```
IF a=10 THEN PRINT "X"
```

meaning, 'when the variable a has taken the value 10 you must print "X"'.

With the inequalities, however, you only use them in the second sense, because you can't use an inequality in a defining way.

Try LET x<10, and you can't get it to work.

But if you were to use the inequality as part of an IF statement, it would provide a way of sorting out variables that take on different values.

Try this program:

*Program 25 SORT OUT*

```
   5 REM PROGRAM 25
     SORT OUT
  10 FOR n=0 TO 21
  20 LET x=RND-.5
  30 IF x>0 THEN  PRINT x,"POSIT
IVE"
  40 IF x<0 THEN  PRINT x,"NEGAT
IVE"
  50 IF x=0 THEN  PRINT x,"ZERO"
  60 NEXT n
```

This program will go round the loop 22 times and generate a random number. The number (stored in the variable x) will be between ½ and −½, then it will be tested by lines 30, 40 and 50 in turn. The number must be either positive or negative or zero, so each time there will be a number printed, and because the program has sorted the numbers out, the correct category will be printed alongside. This is the basis of all routines that make a distinction between numbers.

So that's the way inequalities are used on the computer. It is possible, however, to use inequalities in a defining way with ordinary maths.

For example, we can write:

$x<3$ (meaning x is less than three)

or,

$y>x$ (meaning y is greater than x)

I think you can probably see that these sorts of definitions are not precise in the same way that equations are precise. They only set the limits of the value that a variable can have: they do not tell you its exact value.



These inequalities can be used to define a range of values too. So we can say, for example,

$3<x<10$

meaning '3 is less than x, and x is less than 10' or, if you like, x is between 3 and 10. Or we could have,

$3\leqslant x\leqslant 10$

meaning x is greater than or equal to 3 and less than or equal to 10.

I'm using maths symbols here because you can't use inequalities like this on the computer to calculate anything. I think you might like to know that inequalities can be used on paper like equations. You can add the same to both sides of an equation, or divide both sides of an equation by the same number. It is possible to write an inequality,

$a>b$

add the same to both sides and it will remain true.

$a+x>b+x$

Think about this. Replace the letters with numbers. For instance, you can let a be 3 and b can be, say, 2. Then the inequality

$a>b$

is true (because 3 is greater than 2). If you choose any value for x, say 5, then the inequality:

$a+5>b+5$

is also going to be true.

You can try (in the same way) to test out the following rules that govern inequalities, which I will list in a table for you:

INEQUALITIES

| | | | | |
|---|---|---|---|---|
| If | $a>b$ | then | $a+x>b+x$ | |
| If | $a>b$ | then | $a*x>b*x$ | (when $x>0$) |
| and if | $a>b$ | then | $a*x<b*x$ | (when $x<0$) |
| If | $a>b$ | then | $1/a<1/b$ | |

| | | | | |
|---|---|---|---|---|
| If | $0<a<b$ | then | $a\uparrow x<b\uparrow x$ | (when $x>0$) |
| and if | $0<a<b$ | then | $a\uparrow x>b\uparrow x$ | (when $x<0$) |

Really, the subject of inequalities is straightforward, just common sense, in fact. The trouble is that it gets very involved after a while, so we can leave it at that, if you like. A good summary is to consider the idea that if a number is compared to another number, it must be either greater than it, less than it, or actually equal to it.

So that the opposit of $\geqslant$ is $<$, and the opposite of $\leqslant$ is $>$ (because if it's not one it must be the other!). Here's a good way of testing it out:

*Program 26 LOGICAL SORT OUT*

```
   5 REM PROGRAM 26
     LOGICAL SORT OUT
  1Ø FOR n=Ø TO 21
  2Ø LET x=INT (RND*2Ø)
  3Ø IF x>=1Ø THEN  PRINT AT n,Ø
;"*",x
  4Ø NEXT n
```

and you will get a printout of a star (asterisk) and the value of the randomly chosen variable x only if it is greater than or equal to ten.

Substitute this for line 3Ø:

30 IF x<1Ø THEN PRINT AT n,Ø; "*",x

you will get printouts only for x values less than ten.

So if we write:

3Ø IF x>=1Ø OR x<1Ø THEN PRINT AT n,Ø; "*",x

we should pick up all possible choices of x and get a printout on every line. If we'd used $>$ and $<$ as opposites, we could not guarantee picking up all values of x (because it is possible that x=1Ø) and if you substitute the line below you will see it happen

3Ø IF x>1Ø OR x<1Ø THEN PRINT AT n,Ø;"*",x

Dutifully, the program leaves a gap wherever it has generated a 1Ø.



The use of the OR command may have intrigued you. There is another one like it:

AND

You can use them as you would in English to chain together conditions like IF x is less than 1 AND y is not equal to Ø, then . . .

Another exclusive pair is, of course, = and <>, because two numbers are either equal or not equal.

This may not seem very useful at the minute, but it's all necessary ground work for some much more interesting and useful mathematical concepts we shall be looking at later.

## 2 ABS and SGN

Here's a convenient spot to introduce ABS and SGN. They are Spectrum BASIC commands: actually 'functions' is the correct term. And they are not used much at all. We'll have a look at them, though, because they are connected with numbers, and this is about maths.

ABS is a function just like INT was a function. Do you recall the effect of INT? I hope you do because we've just used it in Program 26. Put INT in front of a number (or variable), and it will hack off the decimal places and give you back a decimal-less number known as an integer (integer means 'whole number').

Try:

PRINT INT 8.4

and you get back 8: the decimals have been hacked off!

Unfortunately, if you do it with negative numbers you will get the next lowest number, so that

PRINT INT −6.4

gives you back −7 (which is not very satisfactory to my way of thinking, but there we are!)

Anyway, we can investigate the ABS function in the same way. Try this:

PRINT ABS 4.7 (ABS is in green above the G key)
PRINT ABS Ø

PRINT ABS 78639
PRINT ABS 1e+6

It isn't doing anything to those numbers. But not all numbers go through untouched. Try:

PRINT ABS −57

We seem to have found a number that is altered. And in fact it was because it was negative. In general, the ABS function finds the absolute value of a number. In other words, it looks to see if the number is negative, and if it is, it hacks off the − sign and makes it positive.

No big deal then. We understand ABS. Can we think of a use for it? Not easily. It's one of those things that makes its need apparent as you're programming: to stop values being used that will print at negative line or column numbers, for example. That kind of thing. Also, if you think back about the restrictions on the square root function, we don't want a square root of a negative number, so it is maybe safer to put SQR ABS x in some circumstances.

What about SGN. An even more singularly unused function. SGN stands for SIGNUM, which is the Latin for 'sign'. It returns a 1 if the number you apply it to is positive, a Ø if it is Ø, and a −1 if it is negative. The best way to get a feel for it is by running Program 27. SGN is above the F key.

*Program 27 SIGNUM*

```
  5 REM PROGRAM 27
    SIGNUM
 10 PRINT "Number","Signum"
 20 FOR n=1 TO 21
 30 LET x=INT ((RND-0.5)*10)
 40 PRINT AT n,0;x,SGN x
 50 NEXT n
```

And you will find your two columns of numbers corresponding as I said they would. Where the number generated by the function in line 3Ø has a positive value, SGN has returned a 1, and where the number is negative we get −1. Where it is zero, SGN x is zero too.



Simple, and maddeningly difficult to find a use for.

So far, this chapter has been a mixture of lifeless inequalities and insipid functions that don't do much, but the rest of the chapter is Most Important. Happily those of you who have plugged on through the marshes and bogs of sections 1 and 2 of this chapter are amply rewarded now, because you have already been introduced to the concepts we are about to examine without realising it.

Have a look at the opening paragraph of Chapter 9 of the Spectrum users' manual (the orange one that comes with the computer, I mean), the one that begins: 'Consider the sausage machine . . .'

That's what it says. You are now in a position to understand it. And Sinclair has a point: the function is indeed like a sausage machine.

## 3 Functions

It took me ages to get to grips with the idea of functions when I was a kid. It will take you about fifteen minutes (if that).

So far we have considered several functions. Look at the expression below:

x=ABS y

You know that if you plug in different values for the variable y, the equation will yield (give you back) different values of x according to some rule. Once you know what the rule is (and you should know the rule for ABS now, because we've just talked about it) you are home and dry and able to use the expression.

Another example of a function is:

x=SGN y

and again, once you know the rule telling you how it works, you've cracked it.

And so with x=INT y, or x=SQR y, or x=PI*y, or even x=LN y, which you may not know.

There are a whole lot of functions on the Spectrum keyboard you might not have met before, but it is, after all, the purpose of this book to give you a little background about these functions, so that you can use them with confidence and make them work for you.

I will list all the functions we might get to talk about. Then you will know what's coming:

| | | | | |
|---|---|---|---|---|
| PI | LN | SIN | ASN | DEF FN |
| SQR | EXP | COS | ACS | FN |
| ABS | | TAN | ATN | |
| SGN | | | | |

Column one isn't going to pose you any problems. I bet you can explain them all. If you can't, you might like to flip through the book up to here and refresh your memory.

Column two I'm going to deal with in the rest of this chapter because I think you'll be ready for it after Chapter 2. LN is a function called 'natural logarithm', and it is one of those functions that non-mathematicians have no idea about. But you will soon. The other column two function EXP is, interestingly enough, the inverse of LN, and we should therefore have no difficulty understanding one if we can crack the other.

Column three functions you may well have heard about. They're used in geometry, are important when dealing with circles and triangles and will be quite easy to understand. In the same way that LN stands for 'natural logarithm' and EXP is short for 'exponential', so SIN means 'sine', COS 'cosine' and TAN 'tangent'. But the abbreviations are so common that they can be used when 'speaking' an equation involving one of them to yourself, so you might say,

x=COS y+TAN y

as 'x equals cos y plus tan y'. Anyhow, before we get sidetracked once again into the world of geometry, let's discuss column four.

Column four is simply the inverse functions of those in column three. Remember, inverse means 'reverse', just like squares and square roots.

We meet the problem of the Chinese typewriter again here, because the Sinclair functions ASN, ACS and ATN are short for 'arc sine', 'arc cosine' and 'arc tangent'. But mathematicians write them as in the column on the right:

| | | |
|---|---|---|
| ASN | arc sine | $\sin^{-1}$ |
| ACS | arc cosine | $\cos^{-1}$ |
| ATN | arc tangent | $\tan^{-1}$ |



The little 'to the minus 1' indices are there really to show that the functions are the inverses of SIN, COS and TAN.

Once again, we can't go off into a discussion of these strange beasts until we've laid the foundations, no matter how tempting it might be, so let's just keep to the business in hand.

Column five: DEF FN and FN are very useful little additions to the Spectrum's mathematical repertoire. They stand for DEFine FuNction and FuNction and allow you to create combination functions of your own. (Have a look at Appendix V at the back of the book.)

But for the moment let's look at this function business in greater depth.

If we write an expression like y=SQR x, we say that y is a function of x. It's just the way it is talked about, and we could write a general expression:

$$y=f(x)$$

which reads 'y equals f of x' and means y equals a function of x, and that can be any function we care to put in. So that we could have y=ABS x, or y=SGN x or, as we had above, y=SQR x. All we're doing is plugging in any function we want to choose into the general equation $y=f(x)$.

(Yes, we could even have y=LN x, or y=EXP x).

See what I mean about most of the difficulty in maths being a question of learning the jargon? The meanings are not that hard.

Why call it a function?

Well, don't you see, if you were to 'perform a function' you would be doing a job on something. Like if your function is to control railway engines, then your job would be an engine driver, and if you were an engine driver, your function would be to control railway engines. With mathematical functions, they perform their functions on numbers. You put a number in and get a (sometimes) different number out. Just think back over the ABS and INT and SGN functions and think what they did to the numbers that came their way.

Bearing that in mind, let's take a couple of functions you've no idea about and do something with them that changes numbers.

Try this out, without worrying what's going on:

*Program 28 EXPONENTIALS*

```
  5 REM PROGRAM 28
    EXPONENTIALS
 10 FOR x=1 TO 21
 20 LET y=EXP x
 30 PRINT x,y
 40 NEXT x
```

You will get a column of x values (running from 1 to 21) and a second column of y values running from 2.7182818 to 1.3188157E+9.

Does the speed with which that function grows remind you of anything?

Maybe it reminds you of the powers of 2 function we considered earlier. Try replacing line 20 in program 28 with,

20 LET y=2↑x

and see how that goes.

That function is taking off fast too! But not quite as fast as the one for y=EXP x. Let's compare them directly:

*Program 28a EXP INVESTIGATIONS*

```
  5 REM PROGRAM 28a
    EXP INVESTIGATIONS
 10 FOR x=1 TO 21
 20 LET y=EXP x
 30 LET z=2^x
 40 PRINT x;TAB 5;y;TAB 19;z
 50 NEXT x
```

We can make a direct comparison now, and it seems that the exponential function is taking off faster than 2 to the x.

Now edit out the 2 in line 30 and insert a 3, so that you're letting z=3↑x. And what do you get?

I ran it and got a result taking off faster than EXP this time.

Now edit 2.5 into line 30 instead of 2 or 3. It's getting close, isn't it? Try a few other values, see if you can choose a number by trial and error that mimics the EXP function exactly.



It might be possible, and then again it might not be possible. It remains for you to try to zero in on the number.

In fact, I think you might be surprised to find that the number you are looking for has been staring you in the face for some time. When x=1, line 20 generates y=EXP 1 and prints it at the head of the second column. This number is 2.7182818. Try using that number in line 30, writing: 30 LET z=2.7182818↑x

Now the two columns are almost identical. Does this imply that the function in line 20 and the function in line 30 are almost the same?

Yes, it does. If we can produce an identical effect on all numbers, then it must be an identical function, and in fact EXP, known as the exponential function, is defined as this curious number 2.7182818 to the power x.

Now instead of writing this horrendous number out every time we want to use it, why not do what we did with the equally peculiar 3.14159. . .etc? (And in fact the exponential number is irrational, an infinitely long decimal, just as PI was.)

The letter we choose for the exponential number is, not unexpectedly, 'e'. I said we would get around to it later.

Because e is irrational, we can't get a value for it precisely, but e=2.7182818 is good enough for our purposes.

An actual mathematical definition of the exponential function is that it is the function whose rate of change equals itself, but let's not dwell on that for the time being!

Another interesting thing about this function is that you can get at it using a series. And since we've not covered series yet, it may not make much sense to say

$$\text{EXP } x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

And what are those numbers with exclamation marks after them? It is going to have to wait.

## 4 Introducing graphs

There is no doubt that a picture is worth a thousand words in mathematics. It is so much easier to see what's going on with some of these things when you can have a picture of it, and a graph is nothing more than a mathematical picture.

We've used graphs before, notably in Program 14a and in the quite complicated Program 24. We looked at the graph plotted by the functions for square roots, cube roots, fourth roots, etc.

Undoubtedly you've seen graphs on TV, in the newspapers, in books, on wall charts and a million other places besides, and it's such an easy thing to 'read' that you understand what it means without necessarily knowing what it's based on.

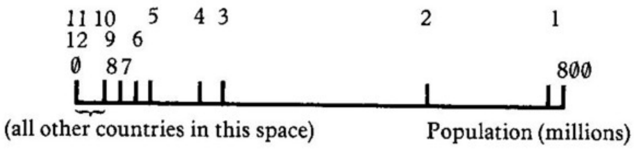
Really, it comes down to scales, like a ruler, or the Real Line: a line marked off at equal intervals with a suitable scale. You could draw out a section of the Real Line and do something with it. Suppose you had the data to represent on a graph, and you wanted to make some sort of comparison between the figures, then you could plot them along the real line.

For example, suppose we had the populations in millions of the twelve most populated countries in the world in 1973:

| | | |
|---|---|---|
| 1 | China | 782 |
| 2 | India | 584 |
| 3 | USSR | 252 |
| 4 | USA | 214 |
| 5 | Indonesia | 128 |
| 6 | Japan | 108 |
| 7 | Brazil | 100 |
| 8 | Bangladesh | 74 |
| 9 | West Germany | 62 |
| 10 | Nigeria | 61 |
| 11 | Pakistan | 59 |
| 12 | Great Britain | 56 |

We could choose a Real Line running from zero to, say 800, make a note on it to remember that the 800 marked means 800 million, mark it off in equal sections and plot our data.

And we would get:






Why did we bother to draw the Real Line to zero? Why not end it at 55 million, just below the last value? The reason is that it would distort the data in this particular case. (Populations actually start at zero: that's the theoretical limit. If you originated the graph at 55 million, it would seem as though the graph was telling you that 55 million is a minimum population.)

But that's by the by. I don't happen to think that plotting data along the Real Line is very informative anyway. It would be much better to have a two-dimensional display instead of a one-dimensional display.

Here's a Spectrum program that does both:

*Program 29 POPULATIONS 1-D*

```
  5 REM PROGRAM 29
    POPULATIONS 1-D
 1Ø PLOT Ø,88: DRAW 2ØØ,Ø
 2Ø FOR n=1 TO 12
 3Ø READ x
 4Ø PLOT x/4,89
 5Ø NEXT n
 6Ø DATA 782,584,252,214,128,1Ø
8,1ØØ,74,62,61,59,56
```

This reproduces the Real Line with points all along it. It's worth detailing the operation of Program 29, because it introduces a fresh computing concept (which I'm sure some of you will have already conquered). Line 6Ø holds the data, our twelve populations, and the computer is told that it's our data by the command DATA (which is on the D key).

So the program draws a line with line 1Ø, then goes round the FOR . . . NEXT loop, READing the data points each time and plotting them along the line (actually, one pixel higher than the line, so that you can see them). If the READ command tells the Spectrum to pick out the DATA values, and we have used the variable name x to take the data, you can see that before plotting it we divide it by four. Why?

Easy. Line 1Ø is 2ØØ pixels long, and we are plotting points in the range Ø to 2ØØ. The problem is that our data spans Ø to 8ØØ, (it really spans Ø to 8ØØ million), so we divide it by 4 to make it fit. Our scale factor is Ø.25, therefore.

Now for two dimensions:

*Program 29a POPULATIONS 2-D*

```
    5 REM PROGRAM 29a
      POPULATIONS 2-D
   10 FOR n=1 TO 12
   20 READ x
   30 PLOT n*20,x/5
   40 NEXT n
   50 DATA 782,584,252,214,128,10
  8,100,74,62,61,59,56
```

And you can make it better to see by adding 35 DRAW 0, −x/5 to it.

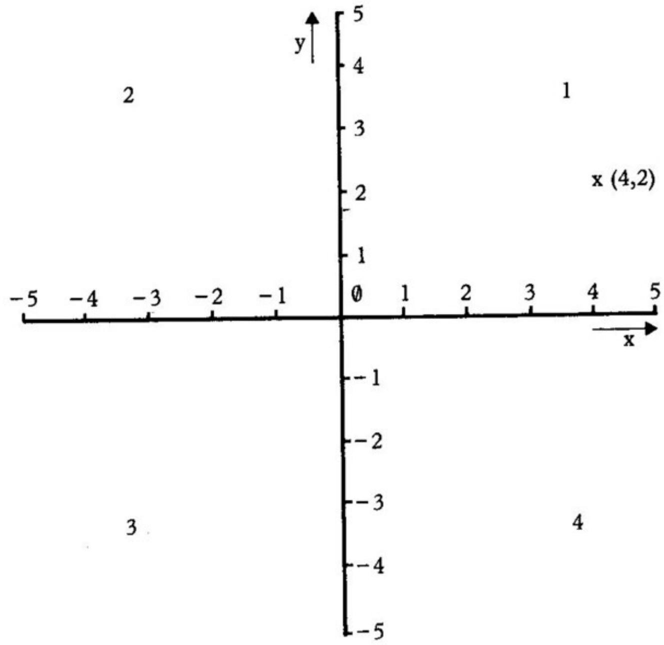
Can you see the scale factor in line 30? I had to make it 5 this time to get the vertical values to fit nicely into the picture.

Now think on this: if we have a list of values connected to another list of values, then we ought to be able to get a graph out of it. If we have a function, or any old expression involving a couple of variables and a few constants, we can get a graph out of it.

There's a convention to drawing graphs. I mentioned in Chapter 1 the French philosopher Descartes. He left us with a standard way of writing equations and graphs.

We normally use two-dimensional graphs because they fit on sheets of paper best, and we say the across-the-page direction is the 'x direction', and we call the up-the-page direction the 'y direction'. We take a couple of Real Lines and lay them at right angles (90°) so that they cross at zero on each of them:






The lines are called axes (singular axis, plural axes, pronounced ak-seez), and where they cross is called the origin. Obviously, we have four areas of paper created in this way, and we call each part a quadrant, from the Latin for a quarter.

The quadrants are numbered anticlockwise starting at top right: 1,2,3,4 as shown, and any point in the first quadrant has a positive x value and positive y value. Any point in quadrant 2 will have a negative x value, but a positive y value. Any point in quadrant 3 will have both x value and y value negative, and quadrant 4 has a positive x value, but negative y value. We can summarise this in a table:

| Quadrant | X value | Y value |
|---|---|---|
| 1 | + | + |
| 2 | − | + |

| | | |
|---|---|---|
| 3 | – | – |
| 4 | + | – |

There is a pleasant symmetry about it.

To underline exactly what I mean by 'any point in the first quadrant', I've chosen to mark the point I'd like you to consider in quadrant 1 with a cross. Next to it is:

(4,2)

And that is the name of the point. We say the little cross is at the point (4,2), and so that there's absolutely no confusion whatsoever we always specify a point by quoting the along-the-x-axis distance first, then the up-the-y-axis distance second, so that (4,2) means 'along four and up two'.

Do you recognise this lot? I should hope so: it's all in Chapter 1 and is the basis of the plotting the Spectrum has been doing for us. We have been writing things like PLOT 100,50, to mean plot the point 100 along the x axis and 50 up the y axis, so it shouldn't be very hard for us to keep in mind.

So all we have to do to get quadrant-1 type graphs, where x and y values are both positive, is to treat the screen as quadrant 1, with the origin (pint 0,0) at the bottom left-hand corner. It's tailor-made for it.

And we can move the origin to the centre of the screen (pixel 128,88) if we want a four-quadrant representation.

That's the basics. Now how about showing a function?

Try this first, so that we have the basics on the screen:

*Program 30 AXES WITH CURVES*

```
  5 REM PROGRAM 30
    AXES WITH CURVES
  10 PLOT 0,0: DRAW 255,0: PLOT
0,0: DRAW 0,175
  20 PRINT AT 0,1;"Y": PRINT AT
20,31;"X"
  30 FOR n=0 TO 13
  40 PLOT n,n^2
  50 NEXT n
```



Again you can do our little trick of drawing in the graph to make it clearer to see by adding this: 45 DRAW 0,−n↑2

Not very good is it? It's a graph of the function,

$y=x^2$

and it uses the values,

| Y | X |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 4 | 2 |
| 9 | 3 |
| 16 | 4 |
| 25 | 5 |
| 36 | 6 |
| 49 | 7 |
| 64 | 8 |
| 81 | 9 |
| 100 | 10 |
| 121 | 11 |
| 144 | 12 |
| 169 | 13 |

Now you see why I had the FOR . . . NEXT loop stop at 13, because $14^2$ is 196, and a Y value for 196 wouldn't have fitted on the screen.

What can we do about the fact that it's squashed up into a portion of the screen near the Y axis? You might be pleased to hear there's a good fiddle. You can get the program to interpolate quite easily. (Interpolate means to fill in the values between two values.) And you can make your Spectrum do it for you: just change line 30 to:

```
30 FOR n=0 TO 13 STEP 0.1
```

and change line 40 to:

```
40 PLOT n*10, n↑2
```

The Spectrum now obediently calculates the inbetween values, ten at a time, and gives you a ten times expanded graph which shows off the curve a bit better. Maybe there are a few steps in the curve, but it's almost a smooth one, and there you have a graph of a function. The curve is actually called a parabola.

It may strike you that you can get at any function using this method. If you have your function in line 4Ø, what stops you making it into any function you like? So we could plug in EXP n if we wanted. We might have to alter our scale factors, or at least replace line 45 with an appropriate substitute (or get rid of it entirely again.)

Let's try it out.

*Program 3Øa* *EXPONENTIAL CURVE*

```
    5 REM PROGRAM 30a
      EXPONENTIAL CURVE
   10 PLOT 0,0: DRAW 255,0: PLOT
0,0: DRAW 0,175
   20 PRINT AT 0,1;"Y": PRINT AT
20,31;"X"
   30 FOR n=0 TO 5 STEP 0.02
   40 PLOT n*50,EXP n
   50 NEXT n
```

And there we have it, a perfectly reasonable representation of the equation

y = EXP x

or, as mathematicians write it,

$y=e^x$

How did I get figures like '5' and 'Ø.Ø2' in line 3Ø? Actually, I did it by trial and error, using a bit of common sense. I think that after playing around with it for a while you'll get the hang of it easily too.

But as usual in this book, we're running before we can properly walk, and so we must go back to pick up on a few things now.

We've done exponential curves and a parabola, but what about the humble straight line? Try this:



*Program 3Øb* *THE HUMBLE STRAIGHT LINE*

```
    5 REM PROGRAM 30b
      THE HUMBLE STRAIGHT LINE
   10 PLOT 0,0: DRAW 255,0: PLOT
0,0: DRAW 0,175
   20 PRINT AT 0,1;"Y": PRINT AT
20,31;"X"
   30 FOR x=0 TO 255
   40 LET y=88
   50 PLOT x,y
   60 NEXT x
```

You get what you might have expected. A representation of the line whose equation is,

$y=88$

A general form would have been

y = (a constant)

and we usually use k or c as letters to represent a constant, so that

$y=k$

is a general way of writing the equation $y=88$. You can adapt line 4Ø to define the whole family of horizontal lines by writing

```
4Ø LET y=k
```

and inserting

```
25 INPUT "What value for k?";k
```

and,

```
7Ø GOTO 25
```

Each time the program asks you for a new value for k, you should enter a number between Ø and 175, and you will get the members of the horizontal straight line family drawn on the graph.

When you get fed up with that, think about the relationship $y=x$. That means we can generate a few values without difficulty as before:

| X | Y |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| etc. | |
| 255 | 255 |

How do we generate these values? I'll remind you: we just choose our X values as we want and use the equation to give us our Y values. Our equation this time happens to be $y=x$, so that the X and Y columns are identical.

If I ask you to convert this into a graph, can you do it?

It's not hard. When x is 0, y is 0 so the curve goes through the point 0,0, i.e. the origin. Also, when x is 1, $y=1$ too, so the curve goes through the point 1,1. Also, when $x=2$, $y=2$, so the curve goes through the point 2,2. And so on. And you can automate it by writing, 40 LET y=x in Program 30b. Run it and see the result. It's a line that goes up from the origin at 45°, but it is a straight line. (And it stops at $x=175$).

There is a general equation for all straight lines, in fact, and perhaps we should take a look at it:

$y=mx+c$ (or in full, $y=(m*x)+c$)

where m and c are both constants.

I've written a program that lets you investigate the way that m and c affect the line, and also shows you the four-quadrant representation of a graph. It's a goody, so try it out now.

*Program 31 THE STRAIGHT LINE*

```
   5 REM PROGRAM 31
     THE STRAIGHT LINE
  10 PLOT 128,0: DRAW 0,175: PLO
T 0,88: DRAW 255,0
  20 PRINT AT 0,15;"Y": PRINT AT
 10,31;"X"
  30 INPUT "Enter Slope ";m
  40 INPUT "Enter Intercept ";c
  50 FOR x=-128 TO 127
  60 LET y=m*x+c
```



```
  70 IF y<88 AND y>-87 THEN  PLO
T x+128,y+88
  80 NEXT x
  90 GO TO 30
```

To exit from this program simply enter the STOP command when the computer asks "Enter Slope" or "Enter Intercept".

First, the program will ask you for a slope, and you should answer with a number between $-3$ and $+3$ (experiment with numbers outside this range later if you like). You will find that, if you give m a negative value, the slope is down from left to right, and if m has a positive value, the line slopes up from left to right. So now you know what mathematicians mean by positive and negative slopes. And you can also see that the constant m is a measure of the slope of the line.

The other input value is for the intercept. This simply means that, since the line must cross the y axis at some point, the value of the constant c will fix it. If you enter zero for the intercept, the line will cross the y axis at zero (i.e. through the origin), and you can try shifting the intercept up and down with values in the range $-87$ to $+88$ (because the y axis is 176 pixels long). Try a family of lines with slope 1 and different intercepts.

Slope 1 means you have a line going up at 45°. Slope 0 is horizontal. If you experiment with bigger values the line gets steeper and steeper until it exceeds the TV picture's ability to distinguish it from the y axis. For the line to be truly vertical its slope would be infinity.

And all that from the general equation of the family of straight lines:

$y=m*x+c$

This goes for all straight lines you can draw across those Cartesian coordinates and is not something specific to the Spectrum. You can try it out with graph paper or on a brick wall if you like!

At the end of the day, you might ask yourself whether there might be families of other sorts of curve. We know that there is a family of straight lines all summed up by the general equation for a straight line. Might there not be a family of parabolas, or a family of exponential curves each with its general equation?

Yes, indeed there are.

The whole subject of 2-D graphs is known to the trade as plane analytical geometry, and we shall be returning to it later.

But before we leave all this let me ask you something. Can you think of a condition for two lines to be parallel? Two straight lines. I'll give you a clue: write out their equations as

$y=m_1*x+c_1$ (for the first line)
and
$y=m_2*x+c_2$ (for the other line)

# 5

# NUMBER SYSTEMS

## 1 Number systems

I expect you're like me, four fingers and a thumb on each hand, making ten digits in all.

If so, you are equipped with a simple calculator with some quite startling implications.

It is generally believed that we use a decimal counting system because we were tipped off by our fingers and thumbs. And really it seems quite reasonable to count on your fingers, so that when you get to the end of ten you start again with a fresh handful.

The word 'decimal' comes from Latin and means a number system based on ten.

In what way is our number system based on ten?

If you compile a list of all one digit numbers you will get:

Ø, 1, 2, 3, 4, 5, 6, 7, 8, 9

So what do you do when you get to the end of your ten symbols? You go on to 1Ø – in other words you start doubling up the digits, and when you reach 99 you have to start tripling them up, and so on.

This may seem a mite elementary, but it does have some interesting knock-on effects. With our usual butterfly style, no sooner have we learned about a number system using ten symbols than we begin to wonder about systems using some other number as a base.

That word 'base': we use base to mean the number of digits employed in the system. Decimal has a base of ten.

Actually, the use of numbers to other bases has found its greatest application in the world of computers and microelectronics, and you will be able to see why when I've explained a bit more.

Look on your Spectrum keyboard. The green legend above the B key is BIN. It stands for 'binary' and means a number system that has only two types of number, zero and one.

Now, consider decimal again. If I were to write the number 4394, how would you know that it meant four thousand three hundred and thirty-four? Odd sort of question, but think carefully. The position of the number digit within the whole number is important. It would not be true to say that 4394 is the same as 3494 or 9443, because the positions of the digits matter.

In 4394, we know that the digit on the right, 4, means 4, but the 9 is really 9 times 1Ø or 9Ø, the 3 is really 3 times 1ØØ which is 3ØØ and the figure 4 on the left hand end is really 4 times 1ØØØ which is four thousand.

So we can draw up a table showing how the positions are allotted in the decimal system:

| thousands $10^3$ | hundreds $10^2$ | tens $10^1$ | units $10^0$ | decimal point | tenths $10^{-1}$ | hundredths $10^{-2}$ | thousandths $10^{-3}$ |
|---|---|---|---|---|---|---|---|
| 4 | 3 | 9 | 4 | | | | |
| | | | 0 | · | 1 | 2 | 5 |
| 1 | 2 | 3 | 4 | · | 5 | 6 | 7 |

The table shows the meanings of three decimal numbers, 4394, 0.125 and 1234.567, with the column heads showing the effect of the digit being in that particular column. Notice that the first place to the right of the decimal point is used to indicate tenths, the second place to indicate hundredths and the third to show



thousandths. And of course you can extend the columns to deal with as many digits as there are in your number.

It's easier if you look at the way I've written the column heads as powers of ten. So 4394 would be:

$$(4*10^3)+(3*10^2)+(9*10^1)+(4*10^0)$$

And you can check that it works on your Spectrum by using this command:

```
PRINT (4*1Ø↑3)+(3*1Ø↑2)+(9*1Ø↑1)+(4*1Ø↑Ø)
```

You might also check the second number whilst you're at it:

```
PRINT (1*1Ø↑-1)+(2*1Ø↑-2)+(5*1Ø↑-3)
```

This unlocks the door to us, because now we have seen exactly what decimal means, and we can extend the idea to numbers with other bases. We can try it with BIN.

Only two digits in the binary system, remember? Binary means 'connected with two', so let's see what a binary number might look like:

1ØØ1Ø11Ø1

Not very exciting is it? And what might 1ØØ1Ø11Ø1 mean?

What if we take the number on the right to start with. That particular digit is a one, and so it will mean (1*2↑Ø) – and you will recall that any number raised to the power zero is 1, so that bracket simplifies to (1*1) which is 1.

What about the next number in from the right? It's a zero, and means (Ø*2↑1) or simplifying (Ø*2). Recall that any number raised to the power 1 is itself, so that 2↑1 = 2. Remember also that any number times zero is zero, so that (Ø*2) = Ø.

We can carry on, moving in to the next number from the right, but I think it will be clearer to see if I make another table:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Ø | Ø | 1 | Ø | 1 | 1 | Ø | 1 | Digits |
| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Ø | Power of 2 |
| 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Multiplier |

I've just copied the number into the top row, numbered the column in the second row, so that the 1 in the far right column is really (1*2↑Ø) as we said above, and the middle row therefore shows powers of two. The third row just works out the powers of two for you. It just remains for you to go along the number from right to left. If there's a 1, you add the number in the bottom row on to your total, like this:

1+4+8+32+256=3Ø1

But don't panic! I can take the sting out of it for you by telling you that the Spectrum will recognise numbers as binary if you give it the BIN command. We can just enter the number with BIN in front of it, and you will get the decimal equivalent automatically worked out for you. Try

```
PRINT BIN 1ØØ1Ø11Ø1
```

and if you've got the ones and zeros in the right order, you'll get back 3Ø1.

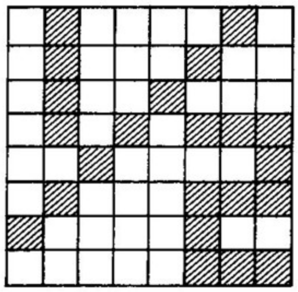
Binary systems have a great application in microcomputers because electric circuits only recognise two states: ON and OFF. If we use zero for off and 1 for on, we can get a circuit to simulate counting in binary, and then we can translate that into decimal (which we can understand). Happily, unless you're an electronics buff, you don't have to know how it does it: it is enough that it does do it.

Where it impinges on the Spectrum user most is in the process of getting up a user-defined character. Let's say we are after a letter that the Spectrum doesn't already have and want to print it on the screen, we can use a tiny little BASIC program to do it for us. We can use the BIN function to hold the data too, if we like.

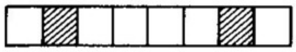
Suppose we want a symbol for a half, ½.



Design one on a grid of eight rows by eight columns like this:



Next, take the design one row at a time. Convert the row into an 8-digit binary number by making the black squares into 1 and the white squares into a Ø. So the top row:



becomes:

Ø 1 Ø Ø Ø Ø 1 Ø

And you can convert that binary number into a decimal number, either by working through the multipliers like I showed you, or simply by getting the Spectrum to do it with:

```
PRINT BIN 1ØØØØ1Ø
```

which returns 66.

Break the ½ symbol down row by row and convert the numbers and you will get:

| | | |
|---|---|---|
| Row 1 | 01000010 | 66 |
| Row 2 | 01000100 | 68 |
| Row 3 | 01001000 | 72 |
| Row 4 | 01010111 | 87 |
| Row 5 | 00100001 | 33 |
| Row 6 | 01000111 | 71 |
| Row 7 | 10000100 | 132 |
| Row 8 | 00000111 | 7 |

If you half shut your eyes you can make out our symbol-to-be from the position of the 1s in that block of binary numbers. Next we have to employ the little BASIC program that loads our eight data numbers into the correct section of the Spectrum's memory. If you look on page 94 of your Spectrum manual, you will find the program on which this is based.

*User defined character loader*

```
   5 REM USER DEFINED CHARACTER
     LOADER
  10 FOR N=0 TO 7
  20 READ row: POKE USR "a"+n,ro
w
  30 NEXT n
  40 DATA 66,68,72,87,33,71,132,
7
```

When you've run it, it will have put the ½ symbol onto the A key; when it's in graphics mode you can get it to print out just by pressing 'A'.

The ½ symbol won't be recognised as a number by the Spectrum's arithmetic handling unit. So far as the innards of the computer are concerned, the symbol is just like any other pretty graphic invention, and if you want it to be printed you must enclose it in quotes (").

Now I must tell you about 'hex'.

Hex is short for hexadecimal, meaning sixteen, and describes a number system that has sixteen different types of number. We know that binary has two types of number, 0 and 1. We know also that decimal has ten: 0,1,2,3,4,5,6,7,8 and 9. But sixteen? There aren't sixteen different kinds of number to write, so how do we cope with it?

We use the capital letters from A to F. Like this:

0
1
2
3
4
5



6
7
8
9
A (=10)
B (=11)
C (=12)
D (=13)
E (=14)
F (=15)

And that's our complete number set. When we want to write the equivalent of 16, we have to roll over and start doubling up those digits again: 10. (Note carefully that the number 10 in hex means not ten, but sixteen.)

It would not be beyond our powers to write a small program to convert decimal numbers into hexadecimal numbers, or perhaps to do the opposite. But first we should have to know the way the powers of 16 go. Try this preliminary program to provide us with the required numbers:

```
Powers of Sixteen
10 FOR n=0 TO 4
20 PRINT n,16↑n
30 NEXT n
```

You would perhaps expect the results to get bigger very quickly, and you should get:

| n | $16^n$ |
|---|---|
| 0 | 1 |
| 1 | 16 |
| 2 | 256 |
| 3 | 4096 |
| 4 | 65536 |

It happens to be a property of powers of 16 that they end in the digit 6! Anyhow, they do get bigger quickly, and if you think carefully, you should be able to convince yourself that any power of sixteen must also be a power of two (because sixteen is itself a power of two: 16=2↑4 to be precise). And this is why hex has found computer applications. It's better suited than decimal because decimal, being based on ten, generates numbers that are not powers of two.

I'm not going to go into the application of hex in this book, because this book is not going to deal with sophisticated computer matters like machine code and assembly language, etc. But perhaps it might make those topics easier when you do get around to them if you know what it's based on.

## *Program 32 HEX TO DECIMAL CONVERTER*

```
   5 REM PROGRAM 32
     HEX TO DEC CONVERTER
  10 INPUT "4-DIGIT HEX NUMBER?
";a$
  20 LET d=0
  30 IF LEN a$<>4 THEN  GO TO 10
  40 PRINT AT 5,13;a$
  50 LET x$=a$(4): LET n=1: GO S
UB 100
  60 LET x$=a$(3): LET n=16: GO
SUB 100
  70 LET x$=a$(2): LET n=256: GO
 SUB 100
  80 LET x$=a$(1): LET n=4096: G
O SUB 100
  90 PRINT AT 15,10;"IS:  ";d: S
TOP
 100 IF CODE x$>=48 AND CODE x$<
=57 THEN  LET d=d+((VAL x$)*n)
 110 IF x$="A" THEN  LET d=d+(10
*n)
 120 IF x$="B" THEN  LET d=d+(11
*n)
 130 IF x$="C" THEN  LET d=d+(12
*n)
 140 IF x$="D" THEN  LET d=d+(13
*n)
 150 IF x$="E" THEN  LET d=d+(14
*n)
```



```
 160 IF x$="F" THEN  LET d=d+(15
*n)
 170 RETURN
```

By far the longest and most involved piece of BASIC we've dealt with up to now! But it's not as forbidding as it might seem (see Appendix II, page 000, for a real conversion program). And before trying it out, understand this: you must enter the hex number in a four-digit form, and you must make sure that any letter digits that occur are entered as capital letters.

For example, if you want to know what FF is in decimal form, just run the program, and when you are asked for the number, enter: 00FF. The answer will appear quickly: 255. If you tried to enter 00ff, you would get the wrong answer, so the best thing is to put the Spectrum into CAPS LOCK as soon as you have loaded the program. You could add this to remind you:

5 PRINT PAPER 2; INK 7; FLASH 1; AT 21,10;
"USE CAPS LOCK!": PAUSE 200: CLS

And to get CAPS LOCK you press the 2 key whilst keeping your finger pressed down on the CAPS SHIFT button. You know you have got CAPS because the L type cursor is now flashing C.

As for the program itself, when it has got the number in hex form, it makes a variable called d take the value zero. (We are going to use the variable d to represent the decimal number by adding the numbers in hex notation from each place of the hex number.)

Line 30 then sees if the number has been put in with four digits and, if not, sends the program back to the beginning. Line 40 prints out the hex number you've put in, so you can see it and check if you've entered it correctly.

Lines 50 to 80 all look similar, and they are doing a similar job. They examine the digits in turn (line 50 examines the rightmost digit, for example) and sets the value of a variable x$ equal to it.

These four lines also give a value to a variable called n that holds the place value of the digit. Remember that the rightmost digit is 'units', so for this digit n = 1. For the next digit in, since we are dealing with hex, n must have the value 16. The next digit n = 16 squared which is 256, and the last one (line 80) is 16↑3 which is 4096. That's how we assign the values implied by the position of the digit in the number.

You'll notice that the lines 5Ø to 8Ø end in the instruction GO SUB 1ØØ. This makes the computer jump straight to line 1ØØ and carry on doing instructions until it finds the instruction RETURN when it will go back to where it jumped from. (This handy parcel of instructions, in our program from 1ØØ to 17Ø, known as a 'subroutine', shows you why the instruction GO SUB is called GO SUB. It means GO to the SUBroutine starting at line (number).

So what does the subroutine in our program do? Line 1ØØ checks if the digit is a regular number-type digit, and if so adds the value of that digit to our variable d whose purpose, remember, is to collect the sum total of values of the four digits together. Lines 11Ø to 16Ø check if the digit is a letter-type and give the appropriate value, so that A gives 1Ø, B gives 11 and so on.

So the full progress of the program is as follows:

1 It executes lines 1Ø to 5Ø
2 Goes to the subroutine to do the first digit.
3 Returns and does line 6Ø.
4 Goes to the subroutine to do the second digit.
5 Returns and does line 7Ø.
6 Goes to the subroutine to do the third digit.
7 Returns and does line 8Ø
8 Goes to the subroutine and does the last digit.
9 It then returns and does line 9Ø, which prints the number held by variable d, and stops.

What a complicated business! If you've followed that, you'll follow anything in this book and more besides.

If you haven't followed it, don't worry because we're leaving this subject, and so long as you've got clear the idea behind number systems you'll be OK.

Before we do leave the subject, I have a bonus for those of you who plugged on to the bitter end. I've included a set of data in the appendix that can be used in line 4Ø of the User Defined Character Loader on page 95.

It will let you define some Greek letters. (If you're feeling adventurous you might like to try a Russian alphabet!)



## 2 The elusive log

Some of you may have noticed that when I was introducing functions back in the last chapter, I quietly slipped off without an explanation of the logarithm.

It might have been bothering you, so perhaps I'd better remedy the situation before going on. Most people who dimly remember them from the days before calculators think of them as columns of nasty numbers in a booklet called 'a set of log tables'! They further remember that, before calculators, it was a morning's work to multiply two five-digit numbers, and a day and a half to divide Ø.3479 by PI times 8.96584. Whole maths lessons used to consist of rows of victimised schoolboys bashing through such meaningless calculations helped (it was supposed) by the mysterious log tables.

Fortunately those days are over, and we can cause the mystery of log tables to evaporate like vampires in the first rays of dawn.

Cast your minds back to the function,

y=EXP x

which can be found in maths books, looking like:

$y=e^x$

and which means e to the power x (where e=2.718281828 . . .).

It is called the 'exponential function' and the 'logarithmic function' is the inverse of the exponential. Considering this: if y equals e to the power x, what does x equal?

In other words, how can we manipulate the function $y=e^x$ to get x on its own on the LHS? It turns out that we can't, because we have to have the inverse of the exponential function first. We defined the idea of inverses earlier on, but let's think about it now. I said earlier that if you have a function, apply it to a number, and then apply the inverse of that function to the result, you get back to what you started from. For example, if you were to take a number and multiply it by PI, then divided the result by PI, you'd get your original number back. Thus, supposing I'd chosen 4:

| | |
|---|---|
| If | y=PI*4 |
| then | y=12.56637Ø61 . . . etc. |
| and | 12.56637Ø61/PI |

will give us back 4, which is indeed the number we first thought of!

So we need a function that is the inverse of the exponential function. Then if

y = EXP x

we can apply the inverse function to both sides. Because the effects of applying the function and then the inverse of that function will cancel one another out, you will get:

(INVERSE EXP)y = (INVERSE EXP)EXP x

which is the same as:

(INVERSE EXP)y = x

And I've already told you that the real name for the (INVERSE EXP) function is the logarithm, found on your Spectrum keyboard on a green legend above the Z key. It is abbreviated in Spectrum BASIC (and also in some maths books) to LN. (The letters stand for Logarithm and Natural, and I'll explain that later.)

Let's get it straight, then. The manipulation can be performed on the equation y = EXP x by doing the same thing to both sides. What we choose to do is something that will cancel out the EXP function and leave x alone on the RHS. The something that will do that is the inverse of the EXP function, and the inverse of the EXP function is the LN. So we get our new expression

x = LN y

which is exactly the same as,

y = EXP x

except that one expresses the relationship as x in terms of y, and the other expresses it as y in terms of x.

Get it? Got it? Good!

So whenever you find an e to a power, you can deal with it. The log to the base e, as it is called (since it deals with e raised to some power) was invented or discovered by the Scot John Napier (1550–1617). If you ever see a reference to 'Naperian logarithms', you will know that it's talking about logs to the base e, often written in maths books as,



$\log_e x$ or $\ln x$

But why choose e, that peculiar and ungainly number, for a base? Hopefully, we've already explained that it has to be a base of e in order to represent the inverse of $e^x$. It just so happens that there are many processes in nature that follow exponential functions (like the growth of populations) so it is convenient to have a log to the base e to help us deal with it. And that's why it is called 'natural'.

But that doesn't mean we can't have logarithms to any other base. We can have, in general terms, a log to base b, written,

$\log_b x$ (meaning log to base b of x)

and b can take any positive value. The most common value happens to be 1Ø, because we use a decimal system, and so we have

$\log_{10} x$ (meaning log to base 1Ø of x)

Because it is the most common form of log, it is frequently found without the little 1Ø being written after and below it as a reminder. So if you see

log x

it means log to the base 1Ø of x, and if you see

ln x

it means log to the base e of x.

That covers most of it. How about some programming?

*Program 33 NAPIER'S BONES*

```
  5 REM PROGRAM 33
    NAPIER'S BONES
 10 FOR x=1 TO 20
 20 LET y=LN x
 30 PRINT x,y
 40 NEXT x
```

That will print out the numbers from 1 to 2Ø with their corresponding logarithms. As you can see, it takes off quite slowly, as you'd expect for a function that is the inverse of an exponential.

We can construct an even more striking representation of the function y=LN x by putting it on a graph:

*Program 33a NAPERIAN GRAPH*

```
   5 REM PROGRAM 33a
     NAPERIAN GRAPH
  10 BORDER 4: PLOT 128,0: DRAW
0,175: PLOT 0,88: DRAW 255,0
  20 PLOT 128+32,0: DRAW 0,175
  30 PRINT AT 10,31;"4": PRINT A
T 10,0;"-4": PRINT AT 10,15;"0":
 PRINT AT 10,19;"1": PRINT AT 21
,15;"-1.75": PRINT AT 0,15;"1.75
"
  40 FOR x=0.2 TO 3.95 STEP .01
  50 LET y=LN x
  60 PLOT (x*32)+128,y*50+88
  70 PRINT AT 0,0;x: PRINT AT 1,
0;y
  80 NEXT x
```

And you will observe that this is more or less the same kind of thing we were doing before with the exponential function.

It's good enough to tell us quite a bit about the LN function. First, notice that it's in the positive half of the graph, which means that the values of x we chose never went negative. (In fact we chose to take values between 0.2 and 3.95, because this is the most enlightening region of the graph.)

You can see that as x gets closer to zero, the value of y gets more and more negative, and it looks as though when x is zero we can't put a definite figure on the function LN x. Mathematicians say that the function is not defined for this value. I have a sneaking suspicion that LN x equals infinity, but we can't use that result on a computer, nor does it help to use it in any mathematical calculation, so let's just leave it at that.

As x grows towards 1, the function grows closer to zero, and the curve actually crosses the x-axis at the point 1,0 meaning that LN 1=0. Enter the command,



PRINT LN 1

and you will get 0. And come to think of it, enter,

PRINT LN 0

and you will see what I mean about invalid arguments!

That was Napier's achievement. Napier had an English friend called Henry Briggs (1561–1631) from Halifax in Yorkshire, who might be said to be the father of the log table, and he popularised the log to base 10. They, as I have been at pains to point out, are called common logarithms, but more formally (and nowadays rarely) they are called Briggsian logarithms.

The trouble is that you won't find any LOG function on the Spectrum. But as we shall see, it is not going to stop us using logs to base ten, or any other base we want either for that matter. So long as we can reduce them to a combination of those functions that do appear on the computer we should be all right.

Remember my saying that multiplication is just repeated adding? There is a relationship between logs of different bases. It goes like this:

$\log_a b = \log_a c * \log_c b$

This was discovered by mathematicians way back, and I don't want to prove it or even do much explaining about it. But if you take it on trust as being true, we can get on and use its results.

First let's take the Naperian and Briggsian logs and find a relationship between them. Try to follow this argument: if

$\log_a b = \log_a c * \log_c b$

then we can let a=10, b=x and c=e, and write:

$\log_{10} x = \log_{10} e * \log_e x$

So if we have a log of the Naperian kind ($\log_e$) of a number we can find the Briggsian log ($\log_{10}$). We can simplify this equation, because $\log_{10} e$ has a value 0.1434294481. (If you can find a set of log tables or a scientific calculator, you can check this.) So our equation becomes:

$\log_{10} x = 0.1434294481 * \log_e x$

Perhaps I can say the same thing another way, then. It's true to say that

$\log_a x = \text{LN } x/\text{LN } a$

And that holds true for any base you want to use.

See if this impresses you:
If 100 = $10^2$ (check that it does!) then it follows that,

$\log_{10} 100 = 2$

And if you enter this in your Spectrum,

PRINT LN 100 / LN 10

you get the answer 2, showing that it all holds true.

Also, we said that

$\log_{10} x = 0.434294481 \star \log_e x$

so we have,

$\log_{10} 100 = 0.434294481 \star \text{LN } 100$

If you PRINT LN 100, you will get 4.6051702, so we have

$\log_{10} 100 = 0.434294481 \star 4.6051702$

and a straightforward calculation of the RHS gives the result 2, by using PRINT 0.434294481*4.6051702.

That's how to get at logs with bases other than e if you want to. What else can we do? Well, I still haven't explained in a satisfactory way how logs come to be associated with multiplying and dividing in the minds of schoolboys. Is it because of some sound mathematical principle?

Yes! And this is an aspect of powers that we never investigated in their two episodes in Chapter 3, so let's look at it now.

First, I'm going to write an equation and I want you to satisfy yourself that it is true:

$100{,}000 = 100 \star 1000$

Not difficult? A hundred times a thousand is one hundred thousand.

What if we rewrite the above equation as powers of ten?

$10^5 = 10^2 \star 10^3$

All we've done is convert 100,000 into its power-of-ten form, and convert 100 and 1000 into their power-of-ten forms. It still



holds true. But do you notice the index numbers? 5, 2 and 3 are related to one another, since 5 = 2 + 3. Might this just be a coincidence? It could be, but in fact it isn't. It's a general rule that if

$x^a = x^b \star x^c$

then,

$a = b + c$

And it appears that we might have found a method of simplifying the operation of multiplication. It's obviously easier to add two numbers together than multiply them together (unless you do it on a calculator!), and so we have only to get our problem into index form to make that simplification possible. Taking logs does just that.

In general we can say that

$\log(a \star b) = \log a + \log b$

and also

$\log(a/b) = \log a - \log b$

Adding replaces multiplication, and subtraction replaces division.

It goes further. You can make raising to a power simpler: by taking logs you will see that raising to a power is simplified to multiplication:

$\log(a \uparrow b) = (\log a) \star b$

Very interesting, but, since the invention of the calculator, only in an academic sort of way. It is important to remember, however, that our log function crops up in lots of different applications in maths and the sciences, and a superficial understanding of how it works, what it is and what it does might cast some light on other areas.

And talking of other areas, why don't we take a trip to Ancient Greece.

# 6

# ANGLES

## 1 Angles

Ancient Greece? How about Babylon instead?

Babylon was a Middle Eastern empire that flourished between 3000 and 1000 BC, and they were very keen on astrology there. So the astrologers did a brisk trade and unwittingly founded the first flowerings of science in the form of astronomy.

Now, the Babylonians used a number system that had a base of not ten, but sixty, and because they were chiefly concerned with astronomy, they used their number system in measuring angles and time.

Does this mean that the Babylonians of old had sixty fingers? Probably not. But they did have the longest surviving system of measuring time and angles at their fingertips. It has survived until today (and maybe even until tomorrow) and its base sixty aspect is intact. What about hours, minutes and seconds? And degrees, minutes and seconds?

60 seconds = 1 minute
60 minutes = 1 hour (or 1 degree)

That holds true for both angles and time.

And 6 times 60 (360) degrees is one complete revolution. The way we show degrees, minutes and seconds is similarly antiquated, but old-fashioned traditions ought not to be thrown out of the window without any thought. Degrees are shown by °, minutes by ′, seconds by ″. So if I were to write 24° 15′ 30″, you would know what it meant.

In geometry, angles appear most often in connection with the corners of triangles (about which we shall definitely be hearing much more) and, because things that rotate are connected with angles, circles tend to crop up a lot.

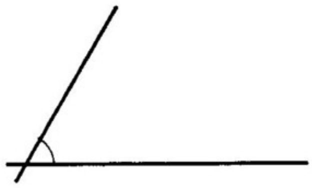


Just so you get a feel for these units, perhaps we should draw a few examples out.

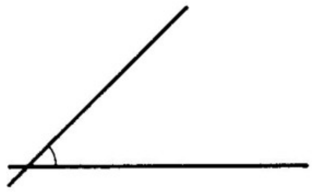
1 revolution = 360°
½ revolution = 180°
¼ revolution = 90°
1 degree = 60′ = 3600″
1 revolution = 21,600′ = 1,296,000″

Angles of these commonly found sizes look like this:

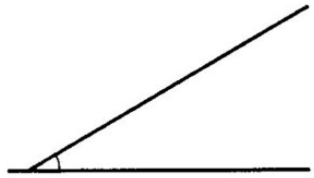
This is an angle of 90° known to the world as a 'right angle'.



Angles less than 90° are called 'acute'. 60° is a common acute angle.



45°, not surprisingly, is half a right angle, also acute because 45°<90°.



The last of our examples, again acute, again very common. Note that 30° + 60° = 90°.

But this is old-fashioned. Let's get into the modern world and get the Spectrum to demonstrate it.

*Program 34 ANGLE DRAW*

```
   5 REM PROGRAM 34
     ANGLE DRAW
  10 CLS : BORDER 4: CIRCLE 167,
88,87: DRAW  INK 2;-87,0
  20 INPUT "ANGLE IN DEGREES?";A
  30 PRINT AT 0,0;A;" DEGREES"
  40 IF A<90 THEN  PRINT AT 1,0;
"IS ACUTE"
  50 IF A=90 THEN  PRINT AT 1,0;
"IS A RIGHT ANGLE"
  60 IF A>90 AND A<180 THEN  PRI
NT AT 1,0;"IS OBTUSE"
  70 IF A=180 THEN  PRINT AT 1,0
;"IS A STRAIGHT ANGLE"
  80 IF A>180 THEN  PRINT AT 1,0
;"IS REFLEX"
  90 LET X=87*COS (A*PI/180)
 100 LET Y=87*SIN (A*PI/180)
 110 DRAW X,Y
 120 PRINT AT 21,0;"PRESS ANY KE
Y": PAUSE 0: GO TO 10
```

A few words about the program. You should have no trouble seeing what the different lines do, with the exception of lines 90 and 100 which you should not understand.

Line 10 draws our circle and a red line to measure the angle from. Line 20 lets you put in your choice of angle. (What if you put in an angle greater than 360°?) Lines 40 to 80 sort the angle into whatever type it is: acute, right angle, obtuse, straight angle and reflex. (Study the conditions for these names.)

When measuring angles with this program, remember that you are looking at the angle starting at the red line and going anticlockwise until you reach the other line.



I think this simple program gets the point over quite well.

But I'm avoiding the issue: what about lines 90 and 100? It must be said, lines 90 and 100 are the clever bit. They work out the variables X and Y that the DRAW command uses to put the angle line in the right place. They are two equations, and you might be able to recognise parts of them:

X=87*COS (A*PI/180)
Y=87*SIN (A*PI/180)

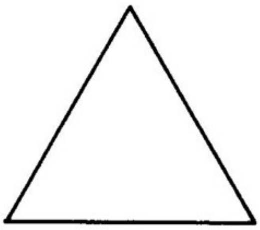
In fact you should have no difficulty with the insides of the brackets. A is the variable holding your choice of angle, PI is no stranger (and I did say before that it crops up in geometry a lot, especially in circles and angles), and 180 is a number (you might recognise the fact that 180 is half of 360, and that might ring a bell). 87, some of you might have spotted, is the radius of the circle we have drawn (check it out in line 10). And so, we have only two strange items remaining. The functions COS and SIN.

## 2 Sine, cosine and tangent

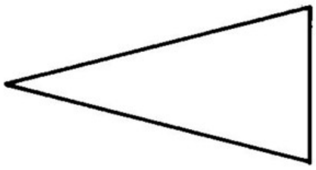
Start with a look at triangles. Now, somewhere in this book we've got a pretty good definition of a square. We said it was a shape made of four lines all the same length and all joined together at 90° to one another. How about a definition of a triangle? A triangle is three lines all joined together, and the lines can be any length, and the angles between them can be just about any angle, so long as no two lines are parallel.

That gives us a whole lot of different kinds of triangle. Some of them we can see are special kinds of triangle (with special jargon names) if we throw in a few more conditions:

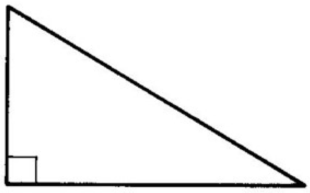
An equilateral triangle has three sides all the same length, (the name gives it away since in Latin *aequus* means equal, and *latus* means side):



An isosceles (pronounced i-*sos*-sell-eez) triangle has two sides the same length:



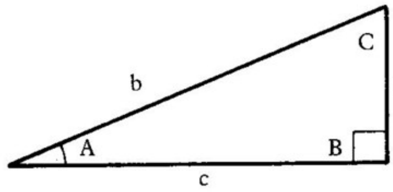
And a right-angled triangle has one angle in it of 90°, like so:



We shall concentrate our efforts on an investigation of the right-angled triangle and reach some important conclusions. It's sneaking up on Sin, Cos and Tan by the back door really.

What we need for a proper investigation is a triangle which has been labelled with names for the sides and angles so we can keep track of what we're talking about:






That's better, and you'll notice that I've made the angle A special by drawing in a little arc in the angle. And just so we can remind ourselves, it's usual to make the right angle obvious by drawing it in with a little L-shape. So we have angles A, B and C. We know that B = 90° (because B is the right angle) and we have three sides, a, b and c.

You might have noticed that I've labelled the sides opposite the angles with the same letter: side a is opposite angle A, side c is opposite angle C, and side b is opposite angle B.

There is a jargon word for the side opposite the right angle. It is called the 'hypotenuse' which comes from a Greek word meaning 'to stretch under'. I can see the point, because the hypotenuse does stretch across the right angle.

Now let's look at the angle A. So far as the angle A is concerned, we have:

side a is the opposite side
side b is the hypotenuse
side c is the adjacent side.

The word 'adjacent' means 'next to', and as you can see, the only two sides next to angle A are the hypotenuse and side c.

How about a small program to show it nicely?

*Program 35 OPPOSITE, ADJACENT & HYPOTENUSE*

```
   5 REM PROGRAM 35
     OPPOSITE, ADJACENT AND HYPO
TENUSE
  10 BORDER 4: PRINT AT 21,1;"A"
: INPUT "ADJACENT SIDE? (<255) "
;c
```

```
  20 PLOT 0,0: DRAW c,0: PRINT A
T 0,0;"ADJACENT SIDE ";c
  30 INPUT "OPPOSITE SIDE? (<175
) ";a
  40 DRAW 0,a: PRINT AT 1,0;"OPP
OSITE SIDE ";a
  50 PAUSE 100: DRAW -c,-a
  60 LET AA=ATN (a/c)
  70 PRINT AT 3,0;"ANGLE IS ";AA
*180/PI;" DEGREES"
  80 PRINT  FLASH 1;AT 4,0;"PRES
S ANY KEY": PAUSE 0: CLS : GO TO
 10
```

Line 60 is the one you'll have to ignore for the moment this time! Suffice it to say that it works out the angle A from the lengths of the sides you have entered. What I want you to get from this is that you can calculate the size of an angle in a right-angled triangle once you know the lengths of two sides.

And it doesn't have to be the opposite side and adjacent side, it can be any two sides. We're very close to finding out what these trigonometric functions are now!

If you want to find the angle in terms of the adjacent and opposite sides, you use the TANgent function.

If you want to find the angle in terms of the adjacent and hypotenuse sides, you use the COSine function.

If you want to find the angle in terms of the opposite and hypotenuse sides, you use the SINe function.

And we've exhausted our pairs of sides, which is why there are three functions connected with this. It might be a good idea to put it down in black and white with reference to our labelled triangle above:

TAN A = a/c

That's the first one. The tangent of angle A is equal to the length of the opposite side divided by the length of the adjacent side.

COS A = c/b



which means the cosine of angle A equals the adjacent side divided by the hypotenuse.

SIN A = a/b

meaning sine of A is equal to opposite divided by hypotenuse.

And that's it, a complete set of fundamental definitions of the three functions. Not so complicated after all: they are just the ratios of sides in a right-angled triangle when it comes down to it. But the range of applications of that trio of functions is absolutely stunning.

## 3 Inverse trig functions

If SIN, COS and TAN are known as the trigonometric functions, or trig functions for short, there must be a set of three inverse trig functions. And indeed there are:

The inverse of SIN is ASN ($\sin^{-1}$)
The inverse of COS is ACS ($\cos^{-1}$)
The inverse of TAN is ATN($\tan^{-1}$)

Is dawn breaking over some of the lesser mysteries of program 35? I hope so. In maths books you can often find the inverse trig functions written with the strange form shown in the brackets above. It is read 'sine to the minus one' or more simply arcsine, and so on for the other functions, and that's why the Spectrum has the abbreviations ASN, ACS and ATN beginning with A.

We could rewrite the three trig functions in terms of their inverses. It's just like the tricks we got up to with EXP and LN. Remember that if

y = EXP x

then another way of writing it would be

x = LN y

And by the same rude logic we can write

SIN A = a/b

or inversely as

A = ASN (a/b)

The poetry of which is: 'Angle A equals arcsine of a divided by b.'

Also if

COS A = c/b

we can write

A = ACS (c/b)

Figure out the poetry for yourself.

And lastly if

TAN A = a/c

then we have

A = ATN (a/c)

I hope this last bit of information casts a twinkle or two over our impenetrable line 60 in program 35. We had

60 LET AA = ATN (a/c)

and I used AA as the variable holding the angle we've been calling A. I had to use AA because I'd used the variable a to mean side a. Anyhow, line 60 reads: " LET the angle AA equal the arctan of a divided by c." and that would be mathematically equivalent to having written

60 LET TAN AA = a/c

The trouble is that this version of line 60 is not good BASIC grammer, and the Spectrum would have turned its nose up at it, leaving us no choice but to use the inverse trig function form.

One final point: the use of brackets here will not have escaped your notice, and it's quite necessary to use them in this case. Because if we'd written,

60 LET AA = ATN a/c

without brackets, the Spectrum would think we meant:

"LET angle AA = arctan a, this result then divided by c"

instead of

"LET angle AA = the arctan of the result of a divided by c"



Is that clear? It's the same as our argument about (a + b)*c not being the same as a + (b*c).

If in doubt, it's a good idea to put brackets into an expression to make sure the Spectrum doesn't get the wrong idea.

Try a few goes at Program 35. Try entering the same number for adjacent side and opposite side. Start with 100, then try 150, then anything you like, so long as it's inside the limits specified. (If you choose numbers bigger than 255 for the adjacent side or bigger than 175 for the opposite side, our program will not be able to draw the triangle and will stop with the report, "Integer out of range".

The point of putting in adjacent and opposite sides that are the same length is that you would expect the angle to be 45°, and you can check that the program works correctly by seeing if it gives you 45° every time.

You might also like to verify this business of dividing by zero, by trying to enter 0 when you are asked for the adjacent side. In effect you are setting variable c = 0 in line 60 and trying to get the Spectrum to evaluate the bracket (a/0) which it can't do. It seems that the TAN function goes towards infinity as our angle goes towards 90°. If you make the angle very small by entering, say, 255 for the adjacent side and 1 for the opposite, it will not give you a very satisfactory triangle, but might illustrate that, as the angle gets smaller (closer to zero), the TAN of the angle gets closer to zero, and in fact TAN 0 = 0.
Enter zero for the opposite side, and you can show that.

At this point, we will have to break off and follow a small digression to cast the final lumen of light over the mysteries of Program 35. It involves the expression in line 70. (It also involves the expressions in lines 90 and 100 of Program 34.) The mystery in question is PI/180 and 180/PI, which crop up without (so far) any explanation and ought to have one. The mystery turns out to be something called a 'radian'.

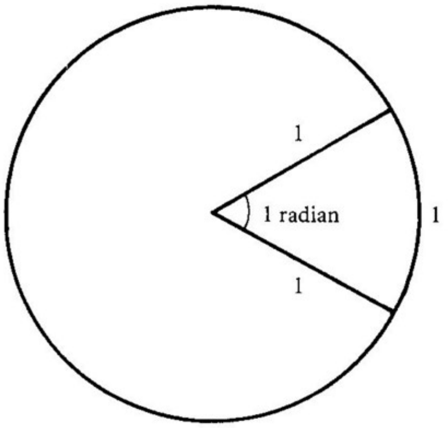
## 4 The radian

Degrees have proved a durable form of measure. The Babylonians used them, and so do we, but we have another form of angle measurement as well. It's called the radian.

There are 360 degrees in a revolution. Perhaps we would do well to find out how many radians there are in a revolution. Actually there are 2*PI radians in a revolution.

But why in heaven's name pick such a crazy number? 2*PI is about 6.2831853 (just do PRINT 2*PI). Why didn't mathematicians decimalise it, and let there be ten per revolution instead?

It's not the mathematicians' fault: it's just the way circles happen to be. If you draw a circle with a couple of radius lines drawn on it (and make them 1 unit long), and if you arrange them so that the part of the circumference between the radius lines is also one unit long (a part of the circumference is called an arc), then the angle made by the two radii (radii is plural for radius) is 1 radian. It's much clearer on a diagram:




As you can see, it looks to be about 60 degrees or so, but we can work it out exactly:

If

1 revolution = 360°

and

1 revolution = 2*PI radians

then



2*PI radians = 360°

and, more simply, dividing both sides by 2,

PI radians = 180°

So we have

$$1 \text{ radian} = \frac{180°}{\text{PI}} = 180/3.14159 \text{ etc.}$$

which brings us to

1 radian = 57.29578 degrees

or in °, ′, ″ form,

1 radian = 57° 17′ 44.8″

Now, the reason we've been in so much trouble with equations 34 and 35, is because the Spectrum is set up to expect angles quoted in radians and not in degrees. So if we want to have it print out the angle in degrees, we must incorporate an equation that converts the units, and that's what's been going on. Multiply a radian quantity by 180/PI, and you have it in degrees: and, inversely, multiply a degree quantity by PI/180, and you get it in radians.

This is a handy little program to generate a table of conversions:

*Program 36 DEGREES AND RADIANS AND TANGENTS*

```
  5 REM PROGRAM 36
 DEGREES, RADIANS AND TANGENTS
 10 FOR x=0 TO 90
 20 PRINT x;TAB 4;x*PI/180;TAB
16;TAN (x*PI/180)
 30 NEXT x
```

And immediately you will see that when the angle x is less than about 10 degrees the radian measure of x is very similar to the TAN x value. As the conversion enters its third screenful, you can see that TAN 45° is 1 (which is what we would expect since the ratio of opposite and adjacent sides is one when the sides are equal). You will also see that 1 radian is between 57 and 58 degrees (which is another result we have had) and, in the last screenful, that the value of the TAN function takes off rapidly getting too big to calculate when we reach 90°.

Mathematicians would say that 'as x tends to 90°, the value of the function tan x tends to infinity.' That's the best they can say, never coming out with it and saying that when x is 90° tan x is infinity. They can't say that because they've never been to infinity to explore it – infinity is just out of the mathematical ball park.

This 'tends to' business is quite convenient, though. We could say that as the angle tends to 180°, then the radian measure tends to PI . . . It has possibilities.

Something tells me there ought to be a program 36a. Wherever we can generate a table of numbers we can usually get a graph out of it somehow:

### *Program 36a GRAPH OF TAN & RADIAN*

```
  5 REM PROGRAM 36a
    GRAPH OF TAN &  RADIAN
 10 BORDER 4: PRINT AT 1,5;"
      ": PRINT AT 2,6;"
  "
 20 FOR x=0 TO 90 STEP .5
 25 PRINT AT 1,6;"           "
 26 PRINT AT 2,7;"            "
 30 LET y=TAN (x*PI/180)
 40 PRINT AT 0,1;"x=";INT x: PR
INT AT 1,0;"X(r)=";x*PI/180: PRI
NT AT 2,0;"TAN x=";TAN (x*PI/180
)
 50 PLOT 2*x,y*10: PLOT (2*x)-1
,(x*PI/180)*10
 55 PAUSE 10
 60 NEXT x
```

And I think you must agree that that is a bit more pleasant to watch. Pretty green border. The trouble is that the numbers printed up top have to leave their decimal places on the screen



when they get overprinted, so we could find this useful:

```
25 PRINT AT 1,6;"           "   (11 spaces)
26 PRINT AT 2,7;"            "  (12 spaces)
  and
55 PAUSE 10
```

Which creates a slower program, but more accurate figures.

Why don't you see if you can improve on this program, making it stretch across the whole screen, or using a white-on-black colour scheme to simulate a blackboard effect. Or maybe you can get rid of the super green border!

But those of you more interested in following up the further ramifications of trig functions might like to get a table together for the SIN and COS functions. If we bear in mind that we'll be constructing a graph of the function, maybe we can choose a program that shows us some useful details and clues.

It might be nice to see a range of values for the function that cover not a ¼ revolution, but a whole one, that is from 0° to 360°. And let's choose intervals of 5° so that we can get through all the data generated without trouble.

### *Program 37 SIN & COS*

```
  5 REM PROGRAM 37
    SIN & COS
 10 FOR x=0 TO 360 STEP 5
 20 LET y1=SIN (x*PI/180)
 30 LET y2=COS (x*PI/180)
 40 PRINT x;TAB 4;y1;TAB 16;y2
 50 NEXT x
```

The first page full of data shows that, when x is 0, Sin x is 0 also, but that Cos x is 1. Then as x goes towards 90°, Sin x rises from 0 to 1, so that when x=90°, Sin x=1. At the same time, Cos x is doing the opposite. It's going from 1 down to 0, so that when x reaches 90°, Cos x=0.

The next page follows the functions from 90 to 180 degrees, at which point Sin x has declined to zero again. Cos x, on the other hand has continued to decline right down to minus one.

By the time the functions have reached three quarters of a revolution at 270°, they are at minus one for Sin x and zero again for Cos x. By the time they have done one complete revolution at 360°, they have got right back to where they started from, with Sin x=0 and Cos x=1.

A graph is definitely called for here, so we can check out exactly what shape these functions give us.

We must first see that both functions are in the range from minus one to one. A mathematician might write:

$-1 \leqslant f(x) \leqslant 1$ (where f(x) is Sin x or Cos x)

Do you remember the function notation? 'f(x)' is read as 'a function of x' or 'f of x' for short.

Anyway, if we use the horizontal axis to measure off degrees, five at a time, and the vertical axis to measure from minus one to plus one, then we will get a good graph extending over 255*5 degrees, =1275 degrees, or a shade over three and a half complete revolutions. I want you to see how these functions repeat. First the SIN:

*Program 38 SINE GRAPH*

```
    5 REM PROGRAM 38
      SINE GRAPH
   10 BORDER 5: PLOT 0,88: DRAW 2
55,0
   20 PLOT 0,0: DRAW 0,175
   30 FOR x=0 TO 1275 STEP 5
   40 LET y=SIN (x*PI/180)
   50 PLOT x/5,(y*87)+88
   60 NEXT x
```

And as you run it you can con yourself into thinking that you are looking at something moving in a spiral down a tube. The Sin function seems to have a lengthways straight-line aspect and a circular aspect giving the helix or spiral.

But let's not forget that this is a 2-D graph, and what you are looking at is a set of just over three and a half revolutions of the sine wave. To get just one revolution do:



```
30 FOR x=0 TO 360
```

and

```
50 PLOT x*255/360,(y+87)*88
```

as replacements. You can identify the points along the line. Where it starts is zero degrees, and where it ends is 360 degrees (and notice that the sine curve has a value of zero at these points). It also has a value of zero at the middle, and a quick cross-check will reveal that this point must be 180°. Similarly, when the curve goes down to −1, it must be 270° along the line, and when it's up at 1, it must be at 90°. Talk about symmetry.

Now consider the data thrown up by Program 37, and think how a cosine wave will differ from a sine wave.

Failing that, use your amended program 38 with the further alteration:

```
40 LET y=COS (x*PI/180)
```

and see what you get.

It's the same form of curve, except that it's shifted along the x axis by 90° (or PI/2 radians).

For a good comparison, reconstitute our abandoned original line 40, this time making it line 35:

```
35 LET z=SIN (x*PI/180)
```

and add this, so that it plots out:

```
55 PLOT x*255/360,(z*87)+88
```

D'you see what I mean about the shift? And here's a stray mathematical thought: the curves actually cross one another at two places. One is up in the positive top half of the screen, the other time is below the line. If we just think about the positive one, we can see that it lies somewhere between 0° and 90°. For the curves to cross, at that point Sin x must equal Cos x, so we could write:

If Sin x=Cos x

then what value of x are we talking about?

And we can see that if we do:

```
PRINT SIN (PI/4), COS(PI/4)
```

we will get two numbers that are the same. PI/4 is equivalent to 45°, so if Sin x = Cos x, then x must be 45°. (What about the negative cross-over point?) True! Correction: if Sin x = Cos x, then x must either be 45° or (180 + 45)°. Only these two values of x satisfy the equation. And we'll be looking at that concept later.

For the moment I'd like to recap what we have here.

First, three trig functions that come from the triangle. Then we had three inverse trig functions. Then we had the idea of measuring angles in radians so that:

$2\pi$ radians = 360°
$\pi$ radians = 180°
$\pi/2$ radians = 90°

etc.

Now we have the sine and cosine waves, and the observation that the tan function goes off to infinity and isn't well behaved like the other two.

What was that value for Sin 45° again? The same as Cos 45°, I know, but what was its value?

Well, Sinclairs like radians, so:

PRINT SIN (PI/4)   (which means Sin 45°)

and we get 0.70710678.

Do you remember my saying that maths was like a case of everything being connected to everything else? It might not seem so yet, but you only have a quarter of the jig-saw pieces in front of you so far. Just to prove my point, and elegantly introduce the next topic, try this:

PRINT 1/SQR 2

and we get . . . 0.70710678. It's that number again!

And why should that be?

## 5 The theorem of Pythagoras

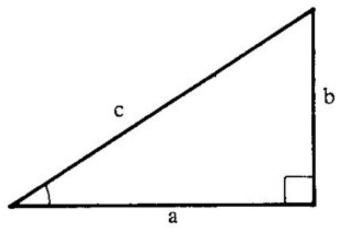
What does theorem mean? In maths a theorem is a statement that requires proof. And Pythagoras?

Pythagoras was an Ancient Greek mathematician. Pythagoras came from Samos (572–497 BC)and wandered around the Middle East before founding his school of geometry. He thought that the



Earth was the centre of the universe, which shows that even great mathematicians can have some faults. But he is remembered most for his theorem which states that in any right-angled triangle the length of the hypotenuse is equal to the square root of the sum of the squares of the other two sides. And what does that amount to?

Bring back our labelled right-angled triangle:




In symbols, Pythagoras' Theorem states:

$$c^2 = a^2 + b^2$$

or, taking the square root of both sides:

$$c = \sqrt{(a^2 + b^2)}$$

The bracket contains the 'sum of the squares of the other two sides' and the square root of it is equal to the hypotenuse.

I feel another program coming on:

*Program 39 PYTHAGORAS*

```
  5 REM PROGRAM 39
    PYTHAGORAS
 10 BORDER 4: INPUT "SIDE No.1
";a
 20 PLOT 0,0: DRAW a,0
 30 PRINT "SIDE 1 = ";a
 40 INPUT "SIDE No.2 ";b
 50 DRAW 0,b
 60 PRINT "SIDE 2 = ";b
 70 PAUSE 100: DRAW -a,-b
 80 LET c=SQR ((a^2)+(b^2))
 90 PRINT "HYPOTENUSE = ";c
```

You may recognise this as being sort of based on Program 35, except that it works out your third side instead of an angle. And you will no doubt recognise Pythagoras' Theorem doing the working out in line 8Ø.

Try entering a side of 4 and the other side of 3, and you will get 5. This is the schoolmasters' favourite, the 3, 4, 5 triangle. It will be shown to ranks of bored and lifeless schoolboys until the end of time, and all because, conveniently, $5^2=3^2+4^2$ (or if you like, $25=9+16$).

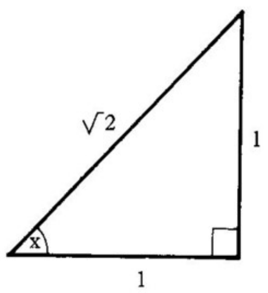
Now for the punch-line. In Program 39, try entering 1 for side 1, and 1 for side 2. Just consider it first: it won't draw a very good triangle, but what the hell. We have:

Hypotenuse $=\sqrt{(1\uparrow 2)+(1\uparrow 2)}$

1 squared is 1 times 1 (which is 1), so our hypotenuse is the square root of 1 plus 1, or in other words, the square root of two.

Hypotenuse $=\sqrt{1+1}=\sqrt{2}$

But let's look closer at that triangle:




We said that we could define trig functions in a right-angled triangle, and so if we look at the Sin of the angle marked as x, we will have

$$\text{Sin } x = \frac{\text{opposite}}{\text{hypotenuse}} = \frac{1}{\sqrt{2}} = 0.70710678$$

which is where we came in . . .



## 7 Cofunctions

Everybody knows what Sin, Cos and Tan are now, but here's a few you might meet in maths books of the smart kind where they try to shake off the beginners:

| | |
|---|---|
| SEC | Also known as Secant |
| COSEC | Alias the Cosecant |
| COT | Often called the Cotangent |

You won't find them on the Spectrum keyboard, but that need not stop us.

These cofunctions are rarely used, so you can forget them if you please. If you must know about them, then know this.

The sum of the three angles of any triangle will be 180°. If the angles are A, B and C, then we have:

$A+B+C=180°$ (=PI radians)

In particular, the right-angled triangle for which we already know one angle is 90° (let's suppose that was angle C) will obey this relationship:

$A+B+90°=180°$

or, subtracting 90° from both sides,

$A+B=90°$

Angles A and B are called complementary angles because they complement one another (meaning they add up to 90°). You will find that the two left-over angles of any old right-angled triangle are always complementary. (If one is 45 the other is 45, and if one is 30, the other must be 60, and so on.) And this is a consequence of a right angled triangle already having one angle of 90°, leaving 90° to be split between the other two.

And that's where the 'co' comes from in cofunction. The sine of 30° is equal to the *co*sine of 60°, for example. And in fact we can generalise:

SIN x = COS (90 − x)

Then we have also

TAN x = COT (90 − x)

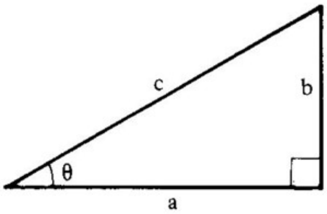
And also

SEC x = COSEC (90 − x)

So, for example, we can say Tan 20° is 0.36397023: just do PRINT TAN (20*PI/180). So Cot(90 − 20) = Cot 70° is 0.36397023 too.

This crops up again when we examine something exotic called 'trigonometric identities'. But that, as they say, is something of a different story.

To summarise the six trig functions:




(Note that the angle here has been marked $\theta$, the Greek letter 'theta'.)

*Reciprocal Relations*

| | | |
|---|---|---|
| SIN $\theta$ = b/c | COSEC $\theta$ = c/b | COSEC $\theta$ = 1/SIN $\theta$ |
| COS $\theta$ = a/c | SEC $\theta$ = c/a | SEC $\theta$ = 1/COS $\theta$ |
| TAN $\theta$ = b/a | COT $\theta$ = a/b | COT $\theta$ = 1/TAN $\theta$ |

The word reciprocal (pronounced re-*sip*-ro-cal) means 'one over' so that the reciprocal of n is 1/n, the reciprocal of Sin x is 1/Sin x or, as we have just seen, Cosec x.

It might have struck you that

| | | |
|---|---|---|
| if | Sin x = b/c | and Cos x = a/c |
| then | Sin x/Cos x | is $\frac{b/c}{a/c}$ |

And we can simplify that complicated looking fraction by multiplying the top and bottom by c, leaving b/a. Now, if we look back, we can see that b/a is equal to Tan x. So we have

$$\frac{\text{Sin x}}{\text{Cos x}} = \text{Tan x}$$

And let's take the reciprocal of both sides: remember that,



when you do 'one over' to a fraction, it just turns upside down, so that 1/(Sin x/Cos x) = Cos x/Sin x. Now we have

$$\frac{\text{Cos x}}{\text{Sin x}} = \text{Cot x}$$

(because 1/Tan x is Cot x). We really need a program to put it all up on the screen for us:

### *Program 40 TRIGONOMETRIC FUNCTIONS*

```
   5 REM PROGRAM 40
     TRIGONOMETRIC FUNCTIONS
  10 INPUT "ANGLE IN DEGREES? ";
x
  20 PRINT "ANGLE IN DEGREES    "
;x
  30 LET r=x*PI/180
  40 PRINT "ANGLE IN RADIANS    "
;r
  50 LET sin=SIN r
  60 PRINT "Sine of Angle       "
;sin
  70 LET cos=COS r
  80 PRINT "Cosine of Angle     "
;cos
  90 LET tan=TAN r
 100 PRINT "Tangent of Angle    "
;tan
 110 LET sec=1/COS r
 120 PRINT "Secant of Angle     "
;sec
 130 LET cosec=1/SIN r
 140 PRINT "Cosecant of Angle   "
;cosec
 150 LET cot=1/TAN r
 160 PRINT "Cotangent of Angle "
;cot
 170 PRINT : PRINT : GO TO 10
```

You will find that, for any of the interesting values of angle like 0°, 90°, 180°, etc., you will have no print out. That's because there is usually one function that scoots off to infinity and stops the Spectrum executing the program. That's why in making a program that will make a graph of these functions over one revolution, it's not going to be as easy as it was to draw out SIN and COS in Program 38 and its variants.

Nevertheless, difficulties can be overcome. We can use Program 40 to show us the danger points.

Let's suppose that the vertical scale is 80 pixels above and 80 pixels below the line. Let's also suppose that to get a good SIN and COS curve we have to have it go up 20 pixels above the line and down 20 below it. So, if those well behaved functions have a maximum of 1 and a minimum of −1, we can accommodate points up to 4 and down to −4. And if we exclude values of angle that give function values outside the range −4 to 4, we should be OK.

Using Program 40 we can see that the zones of danger are around the values 0°, 90°, 180°, 270° and 360°, and by looking at figures around these danger zones we can see that the functions are only well behaved in all six cases if we go beyond 15° or so on either side of the danger points. (Within 15° of 90°, for example, and that means the range 75° to 105°, the values of both Tangent and Secant are too big to plot on the scale we've chosen.)

So we must write a program that selects only the safe ranges if we want it to run OK, and that's not hard.

*Program 41 SIX TRIG GRAPHS*

```
   5 REM PROGRAM 41
     SIX TRIG GRAPHS
  10 BORDER 5: PLOT 0,88: DRAW 2
55,0: PLOT 64,0: DRAW 0,175: PLO
T 128,0: DRAW 0,175: PLOT 192,0:
 DRAW 0,175
  20 FOR x=0 TO 360
  30 IF x<15 THEN  GO TO 100
  40 IF x>75 AND x<105 THEN  GO
TO 100
  50 IF x>165 AND x<195 THEN  GO
 TO 100
  60 IF x>255 AND x<285 THEN  GO
 TO 100
  70 IF x>345 THEN  GO TO 100
  80 LET r=x*PI/180
  90 PLOT x*255/360,88+(SIN r)*2
0
  91 PLOT x*255/360,88+(COS r)*2
0
  92 PLOT x*255/360,88+(TAN r)*2
0
  93 PLOT x*255/360,88+(1/SIN r)
*20
  94 PLOT x*255/360,88+(1/COS r)
*20
  95 PLOT x*255/360,88+(1/TAN r)
*20
 100 NEXT x
```



And what a beautiful piece of Japanese calligraphy we get. Lines 30 to 70 have ensured that we have left gaps around the danger zones, and lines 90 to 95 do the donkey work.

Very beautiful. But to clarify things we need a method of selecting which function we are going to plot, and do them one at a time. So here are a few amendments I've cooked up:

```
5 POKE PRINT "Select SIN, COS, TAN, SEC, COSEC or
  COT"
6 INPUT a$: CLS
```

and alter lines 90 to 95 making them conditional by adding this:

```
90 IF a$="SIN" OR a$="sin" THEN
```

on to the front of it, making the whole thing:

```
90 IF a$="SIN" OR a$="sin" THEN
   PLOT x*255/360,88+(SIN r)*20
```

Remember that the condition must be appropriate to the line, so that line 90 deals with SIN, 91 with COS, 92 with TAN, 93 with COSEC, 94 with SEC and 95 with COT.

Now we can choose each function in turn and get a picture of it. I know it's infuriatingly slow to draw, but think how much longer it would have taken with pencil and graph paper and a book of tables!

Let's analyse what we have. The first vertical line represents 90°, the second 180° and the third 270°. The right-hand edge of the screen is 360°, and the left hand edge is 0°. Both Sin and Cos have done as we might have expected from what we already know. But what about the Tan function?

The Tan function tends to infinity as the angle tends to 90°, then it reappears as if from minus infinity, until it reaches zero at 180°. Then between 180° and 360°, it does exactly the same as it did between 0° and 180°.

The Cosec function comes down from infinity at 0°, levelling out at value 1 when the angle is 90°. It then does the reverse between 90° and 180° zooming up from 1 to infinity. Then between 180 and 360 it does something similar, only with minus infinity and minus one.

The Sec function draws a picture that looks like a sad face. It is a shifted version of the Cosec function, its limits being again plus and minus infinity and plus and minus 1.

The Cot function makes a double journey from infinity to minus infinity.

It has just occurred to me that if you are trying to read this book without looking at the programs on the screen as well, it will sound like gibberish! It's like symbiosis in biology, where a plant and animal rely on one another to survive: you've got to run the programs and read the book, or else it doesn't make much sense.

So there we are.

You have now got a grasp of the fundamentals of Spectrum maths and are ready to go on to ideas that are a bit more advanced.

# 7

# POLYNOMIALS

## 1 Polynomials

'Poly' means 'many', 'nomial' means 'number'. We can also have 'monomials' ('mono' means 'one') and 'binomials' ('bi' means 'two').

A monomial is an algebraic expression involving only one term. So that

| | |
|---|---|
| $2x$ | (2*x) |
| $-5xy$ | (-5*x*y) |
| $3q^2p$ | (3*q*q*p) |

are all monomials.

A binomial, on the other hand, is an algebraic expression involving two terms:

| | |
|---|---|
| $3x+4y$ | (3*x)+(4*y) |
| $p^2-q$ | (p*p)-q |
| $a+x$ | |

are all binomials.

You guessed, I supposed, that polynomials have many terms:

| | |
|---|---|
| $x^2+3xy+15$ | (x*x)+(3*x*y)+15 |

is an example.

Now, when we come to manipulating algebraic expressions, we have to make sure that we follow a few simple rules.

First, we can only add terms together if they are like terms: they have to be the same type. You can't add 5 oranges to 6 apples, but it is perfectly possible to add three pears to six pears. So only like terms can be added (or subtracted).

For example, if we have $8xy-2xy$, it can be simplified to $6xy$.

Or, if we want to add $4x^2-10xy+5y^2$ and $2x^2+4xy-2y^2$ together, the result would be:

$4x^2$ added to $2x^2$ giving $6x^2$

and,

$-10xy$ added to $4xy$ giving $-6xy$

and,

$5y^2$ added to $-2y^2$ giving $3y^2$

and when we collect them up, we have:

$6x^2-6xy+3y^2$

We only add like terms.

What about multiplying and dividing?

Suppose we had 3x and 5y and 6xy and wanted to multiply them together. It would be:

3*x * 5*y * 6*x*y

which would be 3*5*6*x*x*y*y or, condensing it further, $90{*}x^{2}{*}y^2$ which we could write,

$90x^2y^2$

Suppose we had a couple of binomials and wanted to multiply them together:

(x+y)*(x+y)

Then we would have to multiply each of the two terms of the first binomial by the two terms of the second binomial.
So that,

x*x would give $x^2$
x*y would give xy
y*x would give yx
y*y would give $y^2$

But notice that x*y=y*x, so that when we collect up the terms, we get

$x^2+xy+xy+y^2$

or,

$x^2+2xy+y^2$



In full, this result can be shown:

$(x+y)(x+y)=x^2+2xy+y^2$

The reverse of this process is called factorising, and the two factors of $x^2+2xy+y^2$ are said to be $(x+y)$ and $(x+y)$.

So much for that. Juggling with figures seems to be a pretty meaningless pastime. But don't forget that everything in mathematics is connected to everything else! Do you remember I asked you about something on p 76, the last page of Chapter 4? We had just got to showing that the equation of a straight line is of the form

y=m*x+c

where m turns out to be the slope of the line and c is the point where the line crosses the y-axis (called the y-axis intercept). I asked what condition must be satisfied if two lines are parallel when they are written

$y=m_1{*}x+c_1$

and

$y=m_2{*}x+c_2$

If the lines are parallel, their slopes must be the same, and that means (in our two equations) that $m_1=m_2$.

Now we said that an equation of the form $y=mx+c$ is called a linear equation (because its graph is a line), but what about equations that include terms that have $x^2$ in too?

We call them quadratic equations.

## 2 Quadratic equations

These have a general form:

$a{*}x^2+b{*}x+c=0$

and it might be useful to examine this closely.

First, we can see that there is a term in $x^2$, one in $x^1$, and one in $x^0$. The term in $x^2$ is multiplied by a constant here called a, but in fact and practice it can take any value positive or negative. The term in $x^1$ (and we leave out the index because $x^1=x$) is multiplied by another constant able to take any value and here called b. The term in $x^0$ (quite right!) is one, and it is multiplied by a third constant (here called c), and since writing c*1 would be going the long way about it, the final form in general terms is:

$$a*x^2+b*x+c=0$$

One further point about the form of the equation is that it is set equal to zero. This is because if it was set equal to any other number, you could always subtract that number from both sides and get back to the standard general form. For example, if we had,

$$3x^2+4x+5=7$$

we could subtract 7 from both sides and we would get,

$$3x^2+4x-2=0$$

which is in the general form.

But why go to all this trouble to get the equation into a standard general form?

That's a good question, and one that requires a better understanding of what the equation is actually representing. If we go back to a linear equation, we can see that if, for example,

$$3x+5=0$$

we can subtract 5 from each side and get,

$$3x=-5$$

and then divide each side by 3 and get,

$$x=-5/3$$

In other words, we have found a value for x that satisfies the equation $3x+5=0$. We can see that it satisfies the equation because, if we substitute $-5/3$ for x in our original equation, it checks out as true:

$$3*(-5/3)+5=0 \qquad \text{(which is true)}$$

Mathematicians say that they have solved the equation just as if it was a puzzle (which I suppose it is), and they say that the value $x=-5/3$ is the solution of the equation.

In this case, you can't think of any other number that will



satisfy the equation. The equation has only one solution, a unique solution.

But that's just a straightforward linear equation. What about other kinds of equation? Do other sorts of equations have solutions? And is there only one solution in every case? Maybe there are equations with more than one solution that satisfies them.

And in fact there are. Let me give you an example:

$$x^2-16=0$$

This pleasant little equation is obviously not linear because it has a term in x squared, and in fact if you look closely at it, you can see that it fits the standard quadratic form where a is 1, b is zero, and c is $-16$:

$$1*x^2+0*x+(-16)=0$$

which simplifies to our equation:

$$x^2-16=0$$

Let's try to solve it, first adding 16 to both sides:

$$x^2=16$$

and then taking the square root of both sides:

$$x=4$$

And there we have it, a solution, and we can check it out by substituting this value of x back in the equation like this:

$$4^2-16=0$$

which is,

$$16-16=0 \qquad \text{(which is obviously true)}$$

But is 4 the only solution? Because I can think of another value of x that satisfies the equation: $x=-4$

Clearly, minus 4 times minus 4 is 16 (because if you multiply two negative numbers together the negatives cancel one another out, a case where two wrongs do make a right!). When we substitute back into the original equation we have our check.

The most convincing evidence for this can be got from your

Spectrum by commanding:

PRINT −4*−4

and you get 16.

Whilst you're at it, try this:

PRINT −4*−4*−4

and you get −64, and the minus sign is back because two of the minus signs destroy one another, and one is left over to survive.

Anyway, before we get into that, let's repeat that this particular quadratic type equation has two solutions, 4 and −4, and we could write the solution as $x = \pm 4$ (read as x equals plus or minus four).

The next step is the substance of this chapter and the basis for a very important equation that all students of maths know very well indeed. Let's sneak up on it.

So far we have seen how easy it is to solve a quadratic equation which has the b constant equal to zero. It is much more difficult if that b constant is non-zero. If the term in $x^1$ has a non-zero coefficient, a mathematician might remark to another mathematician, then the puzzle is locked. And it's not obvious at all how we solve it.

(Just to keep you on the tracks, you'd better know that the constants a, b and c in our general form are called coefficients, I don't know why!)

Let's look at an example. With coefficient a equal to 2, coefficient b equal to −3 and coefficient c equal to 1, we get:

$$2x^2 - 3x + 1 = 0$$

As I say, it's not at all clear how you go about solving that, but luckily the mathematicians have a secret weapon. Someone found out long ago (I can't seem to find out who it was) that the solutions of any quadratic can be found from this equation:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

That means that the coefficients of any quadratic can be plugged into the RHS of that equation, and it will give us the two values of x that are the solution of the original quadratic. Two solutions? Yes, two solutions: note the plus-or-minus sign.



Why don't we demonstrate this magic solution finder on the quadratic we quoted above. We had a=2, b=−3 and c=1. So our solutions are given by

$$x = \frac{-(-3) \pm SQR((-3)*(-3)-(4*2*1))}{2*2}$$

Let's simplify it one step at a time:

$$x = \frac{3 \pm SQR(9-8)}{4}$$

so

$$x = \frac{3 \pm SQR\ 1}{4}$$

and SQR 1 is 1, so we have our two solutions:

$$x = 3/4 + 1/4 = 1$$

and,

$$x = 3/4 - 1/4 = 1/2$$

I'll leave it up to you to substitute first one and then the other of these solutions back in the original quadratic to check that our magic solution finder works.

Naturally, I'm going to use this result to form the heart of a useful program, but it's not just a question of going ahead and doing it. There are some hidden dangers which might not have made themselves apparent, and we ought to winkle them out before going back to the computer.

You will recall that you can't properly take the square root of a negative number. But our solution finder contains a square root of an expression, and it is possible that the coefficients will happen to combine in such a way that the expression goes negative. And then what can you do?

Specifically, our root is $\sqrt{b^2 - 4ac}$. If $b^2$ is greater than 4ac, then we're OK, and we proceed as above. If $b^2 = 4ac$, then the root is zero, and our solution is just $-b/2a$. But if $b^2$ is less than 4ac, then we have this square root of a negative number situation, and it means that we have no real solutions.

Notice that I very carefully don't say no solutions at all: I said no real solutions. This might sound like a joke, but in fact there are things called imaginary numbers, and so although we have no real solutions, we have a pair of imaginary solutions.

This intriguing and fanciful notion will be explored in a later chapter, and so, until then, you will have to imagine . . .

How about a quadratic solving program then?

*Program 42 QUADRATIC REAL ROOTS*

```
    5 REM PROGRAM 42
      QUADRATIC REAL ROOTS
   10 BORDER 5: FOR n=0 TO 7: REA
D r: POKE USR "a"+n,r: NEXT n: D
ATA 112,16,112,64,112,0,0,0
   20 PRINT AT 2,7;"ROOTS OF QUAD
RATICS"
   30 INPUT "ENTER coefficient a:
 ";a: PRINT AT 6,6;a;"*x@ + "
   40 INPUT "ENTER coefficient b:
 ";b: PRINT AT 6,14;b;"*x + "
   50 INPUT "ENTER coefficient c:
 ";c: PRINT AT 6,22;c;"  =  0"
   60 IF b*b>=4*a*c THEN  LET r1=
(-b+(SQR ((b*b)-(4*a*c))))/(2*a)
   61 IF b*b>=4*a*c THEN  LET r2=
(-b-(SQR ((b*b)-(4*a*c))))/(2*a
): GO TO 70
   65 PRINT  INK 2; FLASH 1;AT 14
,8;"NO REAL SOLUTIONS": PAUSE 20
0: CLS : GO TO 30
   70 PRINT AT 14,6;"FIRST ROOT
";R1
   80 PRINT AT 16,6;"SECOND ROOT
";R2
   90 PAUSE 0: CLS : GO TO 30
```

N.B. The symbol @ represents key A in graphics mode to give the squared symbol as defined in line 10. Chinese typewriter again!

It is quite important for you to realise that the roots of a quadratic are its solutions. It's just terminology: the roots of a quadratic are its solutions, and a quadratic's solutions are its roots.



## 3 More quadratics

It will be interesting if we can get a graph of the quadratic function, because then we can visualise the effects of the constants in the equation. It turns out that the curve drawn by a quadratic is a shape called a parabola, and it is possible to see how the coefficients affect it using Program 43. You will see that if coefficient a is positive, then the curve is U-shaped. If a is negative, then the curve is still U-shaped, but upside-down.

You can try putting in a range of different a coefficients. The range $-5$ to 5 will be enough to show that the bigger the value of a, the narrower the curve. And the bigger negative values give narrower inverted parabolas.

Then you can try varying the b coefficient, keeping the a and c coefficients the same in order to investigate what effect this has. You will find that varying b shifts the parabola left and right (along the x-axis, that is) and varying the c coefficient has an effect on the vertical shift (along the y-axis).

If you do, $a=0$, then you are reducing the equation

$$y=ax^2+bx+c$$

down to

$$y=bx+c$$

which you must by now recognise as the general equation of a line. Lines can therefore be drawn with this program by setting $a=0$. Try $a=0$, $b=3$, $c=0$, and you get a good enough straight line which goes through the origin (obviously, because $c=0$ implies the y-axis intercept is zero).

And without more ado, here it is:

*Program 43 QUADRATIC PARABOLAS*

```
    5 REM PROGRAM 43 QUADRATIC
      PARABOLAS
   10 BORDER 4: PLOT 0,88: DRAW 2
55,0: PLOT 128,0: DRAW 0,175
```

```
20 INPUT "COEFFICIENT a?";a
30 INPUT "COEFFICIENT b?";b
40 INPUT "COEFFICIENT c?";c
50 PRINT AT 0,1;"a=";a;" b=";b
;" c=";c
60  FOR x=-10 TO +10 STEP .05
70 LET y=(a*x*x)+(b*x)+c
80 IF y<87 AND y>-87 THEN  PLO
T  INK 2;(x*12)+128,88+y
90 NEXT x
100 PAUSE 0: CLS : GO TO 10
```

You will notice that as the curve (which plots in red) crosses the axes (which are in black), there is a colour change. This is a problem caused by the screen arrangements of the Spectrum. Each character square is 8 pixels by 8 pixels, and this block of $8^2$ pixels can have only one paper and one ink colour at a time. The result is that as the curve comes down towards the axes, it alters the ink colour to red.

Program 43a will be useful if you want to explore the effects of varying only one coefficient at once. (You will find that the apparently simple 'shift' caused by varying b is not really so simple. It can best be seen by letting a=0 so that it reduces to a linear form.)

*Program 43a A FAMILY OF PARABOLAS*

```
  5 REM PROGRAM 43a A FAMILY
    OF PARABOLAS
 10 BORDER 4: PLOT 0,88: DRAW 2
55,0: PLOT 128,0: DRAW 0,175
 20 LET a=0: LET c=0
 30 FOR b=-6 TO 6 STEP 2
 50 PRINT AT 0,1;"a=";a;" b=";b
;" c=";c
 60  FOR x=-10 TO +10 STEP .05
 70 LET y=(a*x*x)+(b*x)+c
 80 IF y<87 AND y>-87 THEN  PLO
T  INK 2;(x*12)+128,88+y
 90 NEXT x
100 NEXT b
```



This program is set up to vary b between −6 and 6, stepping 2, to give curves with b = −6, −4, −2, 0, 2, 4, 6 and a and c both equal to zero. You can investigate the effect of varying other coefficients by altering lines 20 and 30 to suit, not forgetting to alter line 100 to conform with line 30.

## 4 Conics

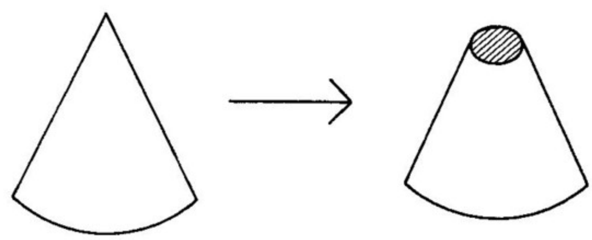
The shape of the parabola was known to the Ancient Greeks, because it belongs to a group of curves called conics which they did a lot of research into.

We should know by now that the Ancient Greeks were mad keen on simple geometric shapes, and one of the simple solid geometric shapes that caught their interest was the cone. You know that a cone is a solid with a circle for a base and, instead of rising straight up like a cylinder, it narrows to a point. Now if that point is directly above the centre of the base circle, we call the solid a right cone. (It's just that the line that runs down the middle of the cone makes a right angle with any radius of the base circle.) The top of a cone is called the vertex.

If the vertex is not directly above the centre of the base, then you have a cone, but it's not a right cone.

Incidentally, if you speak of a cone, you could also mean one with an oval base, so to be quite unambiguous one must be careful to say a right circular cone. But if you hear me talk of a cone, I'm talking about a right circular cone, unless I say otherwise.

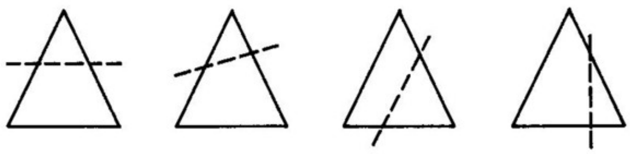
Now consider a cone that's made of a soft material like plasticine. Then take a sharp knife and swipe the top off the cone, but do it so that your swipe is parallel with the base. The shape of the cut will be circular.



If you're a surgeon you'll know that section means to cut, from the Latin secare. In mathematics, then, the cutting of cones is a subject called conic sections. Now, it's pretty obvious that you can get a circle by cutting off the top of a cone, but what I want to show you is that in fact there are four pretty interesting shapes you can get by slicing cones up:

1 Circle
2 Ellipse
3 Parabola
4 Hyperbola

If the cut is exactly horizontal, we get a circle. If the cut is slanting, we get an ellipse. If the cut is exactly parallel to one of the slanting sides, we get a parabola, and if it is more vertical than the slanting side we get a hyperbola. These four cones viewed from the side (looking suspiciously like triangles!) give you what I mean:



Just shut your eyes for a moment and meditate upon the shapes you get from this sectioning of a cone. It is not readily obvious what these shapes will look like (except probably the circle), but like many of the shapes of geometry, they crop up in the real world in the oddest places. The ellipse, for example, became the talk of the scientific world in 1609 when Johannes Kepler (1571–1630) published a book explaining that the planets moved in orbits which were ellipses. (Until then, it was thought the planets moved in a complicated combination of circles.)

But the real star of the conic sections was Apollonius, a Greek who lived between 260 and 190 BC, and who is responsible for much of the work behind this chapter.

Because we know a circle quite well already, let's look at it and its equation, and at its coordinate geometry.

I want you to consider the coordinates we've been using that let us get at the four quadrants; that is, the ones given by the instructions, PLOT 0,88: DRAW 255,0: Plot 128,0: DRAW 0,175, which draw the axes so that the origin is at the centre of the screen.



If we draw a circle with its centre at the origin, then the equation relating the x and y coordinates of all points on the circle is:

$$x^2+y^2=R^2$$

where R is the radius of the circle.

This needs to be immediately transformed into a computer program so that we can witness its effect. Suppose we choose a radius of 10. Then, since $R=10$, we have $R^2=R*R=100$, so our equation becomes

$$x^2+y^2=100$$

and we can get y alone on the LHS by rearranging it to:

$$y^2=100-x^2$$

and taking the square root of both sides:

$$y=SQR\ (100-(x*x))$$

We now have it in a form where we can run it through several values of x, and generate points we can plot out. (That way we can see if we really get a circle.)

*Program 44 CIRCLE ONE ROOT*

```
   5 REM PROGRAM 44 CIRCLE ONE
     ROOT
  10 BORDER 5: PLOT 0,88: DRAW 2
55,0: PLOT 128,0: DRAW 0,175
  20 FOR x=-10 TO 10
  30 LET y=-SQR (100-(x*x))
  40 PLOT x+128,y+88
  50 NEXT x
```

Which gives a circle with a diameter of 10. We can magnify it by making line 40 into: 40 PLOT (x*8)+128,(y*8)+88

But wait! This is not a circle: it is half a circle. Where is the other half?

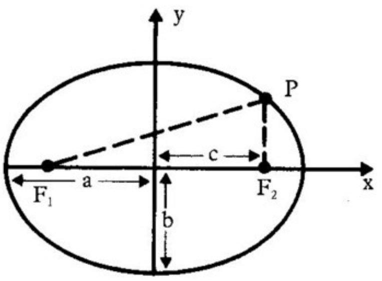
Take a look at the equation we're using. We have got a square root again, and you know that square roots have two solutions (one positive and one negative), and so we get the half circle representing the positive root, and we can arrange the other (negative root) as follows:

*Program 44a CIRCLE BOTH ROOTS*

```
5 REM PROGRAM 44a CIRCLE BOTH
  ROOTS
10 BORDER 5: PLOT 0,88: DRAW 2
55,0: PLOT 128,0: DRAW 0,175
20 FOR x=-10 TO 10 STEP .1
30 LET y1=SQR (100-(x*x))
40 LET y2=-SQR (100-(x*x))
50 PLOT (x*8)+128,(y1*8)+88
60 PLOT (x*8)+128,(y2*8)+88
70 NEXT x
```

And that makes the point nicely. You may be worrying that the two halves don't join up exactly, but this is because we're choosing steps of 1 for x. If we use STEP 0.1 in line 20 you'd get a better effect, and STEP 0.01 would be better still.

Is there a general equation for an ellipse then? Yes, there is. And to see what it means, cast your eye over the diagram below:




The jargon of ellipses is that the long axis is called the major axis, the short axis is called the minor axis (if the two are equal then your ellipse has just become a circle!). An ellipse has two



points called foci (which is just the plural of focus), and if you choose any point on the ellipse, then the sum of the distances to the two foci is a constant. This means we have a handy definition of an ellipse: that curve given by a point which moves so that the sum of the distance to two fixed points is a constant.

In symbols, we have this, which you can relate to the diagram:

$$F_1P + F_2P = \text{a constant}$$

It so happens that when the point P is at the end of the major axis we can see easily that the constant is actually the length of the major axis, so that, since we've written a for the *semi*major axis, we can say

$$F_1P + F_2P = 2a$$

It can also be figured out by using geometry and triangles and a couple of hours of valuable time that $a^2 = b^2 + c^2$.

The naming of the various bits of the ellipse has been standardised by now, and the distances a (called semimajor axis) for half of the major axis, and b (called semiminor axis) for half the minor axis are well known and traditional. And because things generally get to be traditional because they are good in practice, you can be sure that they are good choices. You can see that the distances a and b are the sort of maximum and minimum radius lengths when you think of the ellipse as a squashed circle. The use of the semimajor and semiminor axes gives us this general equation for the ellipse when the centre of the ellipse is at the centre of our coordinate (graph) system:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

Which you can compare with the general equation of the circle by setting both a and b equal to R. It amounts to what we said above about an ellipse being a circle that has been squashed.

Anyhow, to get the subject of our equation as y (like we had to do for the circle above) we must manipulate the standard equation. First we multiply both sides by $a^2$ and $b^2$,

$$\frac{x^2a^2b^2}{a^2} + \frac{y^2a^2b^2}{b^2} = a^2b^2$$

Then we can cancel things that appear on the top and bottom of fractions:

$$x^2b^2 + y^2a^2 = a^2b^2$$

Then we can subtract $x^2b^2$ from both sides:

$$y^2a^2 = a^2b^2 - x^2b^2$$

Divide both sides by $a^2$:

$$y^2 = b^2 - \frac{x^2b^2}{a^2}$$

And then make the RHS into a bracket by taking out a factor $b^2$:

$$y^2 = b^2(1 - \frac{x^2}{a^2})$$

and finally take the square root of both sides

y = ± SQR ((b*b)*(1 − (x*x)/(a*a)))

Then we're all set to do our ellipse program.

*Program 45 ELLIPSE*

```
   5 REM PROGRAM 45 ELLIPSE
  10 BORDER 5: PLOT 0,88: DRAW 2
55,0: PLOT 128,0: DRAW 0,175
  20 LET a=10: LET b=5
  30 FOR x=-10 TO 10 STEP .5
  40 LET y1=SQR ((b*b)*(1-(x*x)/
(a*a)))
  50 LET y2=-SQR ((b*b)*(1-(x*x)
/(a*a)))
  60 PLOT (x*8)+128,(y1*8)+88
  70 PLOT (x*8)+128,(y2*8)+88
  80 NEXT x
```

If you compare this with program 44a, you will see there are a great many similarities. The principal difference is that we have altered the circle-giving formulas into the ellipse-giving formulas of lines 40 and 50. Also, this time we've got a means of setting values for a and b in line 20. There is the opportunity for you to edit line 20 and put in values of a and b of your own choice, to see the effect is has on the ellipse. You should note that as we said before, when a = b (that is, when you give them both the same value) you get an equation that is in essence the equation of a circle.



The flattening of the ellipse is specified by the choice of a and b, and mathematicians and astronomers call this flattening the eccentricity. The circle has an eccentricity of zero, and as the eccentricity increases so the ellipse gets more cigar-shaped.

In actual fact, there is (as you might have supposed) a definition of eccentricity and a symbol for it: e (not to be confused by the uninitiated with the exponential constant, e). Think about the diagram of the ellipse we had earlier, and see that the distance we have marked as c is the distance from the centre of the ellipse to its focus (or one of its foci: it doesn't matter which one because the ellipse is symmetrical). So we can write

the focal distance is c
the semimajor axis is a

and we define the eccentricity as:

$$e = c/a$$

and since we had earlier that $a^2 = b^2 + c^2$, then it follows that $c^2 = a^2 - b^2$, and therefore

$$e = \frac{SQR(a^2 - b^2)}{a}$$

and a quick manipulation will give you:

$$e = SQR(1 - (b^2/a^2))$$

So you can see that the eccentricity depends upon a and b in this curious manner. It would, of course, be possible to add line 90 to your program 45, giving a print-out of the eccentricity.

```
90 PRINT "ECCENTRICITY=";SQR(1-(b*b)/(a*a))
```

This particular program can't cope with situations where a<b, but running through the sample of values we get:

| a | b | e | |
|---|---|---|---|
| 10 | 10 | 0 | (when a=b it's a circle) |
| 10 | 5 | 0.8660254 | (=Sin 60°) |
| 10 | 1 | 0.99498744 | |
| 10 | 0.1 | 0.99995 | |

You can see that as the ellipse gets more flattened, the eccentricity gets closer to 1.

The question arises, what happens when the eccentricity is 1? The answer is that you get a parabola.

Just to give you a practical example of the ellipse before we go after the parabola, let's look at those elliptical wanderers, the planets.

| Planet | Eccentricity | Mean distance from Sun (millions of Km) |
|---|---|---|
| Mercury | 0.2056289 | 57.91 |
| Venus | 0.0067864 | 108.21 |
| Earth | 0.0167209 | 149.60 |
| Mars | 0.0933791 | 227.94 |
| Jupiter | 0.0484550 | 778.34 |
| Saturn | 0.0556402 | 1427.01 |
| Uranus | 0.0472421 | 2869.6 |
| Neptune | 0.0085840 | 4496.7 |
| Pluto | 0.2502 | 5898.9 |

As you can see, the eccentricities are quite small, so the orbits are very close to being circles. I might add that the mean distance of a planet from the Sun is going to be important to us if we want to calculate the maximum and minimum distances between planet and Sun. The mean distance is equal to the semimajor axis.

Planets travel in elliptical orbits with the Sun at one focus. This is a statement of Kepler's First Law (there are two others), and it allows us to get at our distances. The mean distance is going to be (if we refer back to the diagram) equal to the semimajor axis a. The maximum distance is $a+c$, and the minimum distance is $a-c$. So in terms of the eccentricity e,



Maximum distance $= a + ae = a(1+e)$
Minimum distance $= a - ae = a(1-e)$

And you could write a program that calculated these figures for all the planets.

But enough of that: what about the parabola?

As we said, the parabola has an eccentricity of 1, and we also know that parabolas are the result of quadratics. So it will come as no great surprise to learn that the general equation of a parabola is:

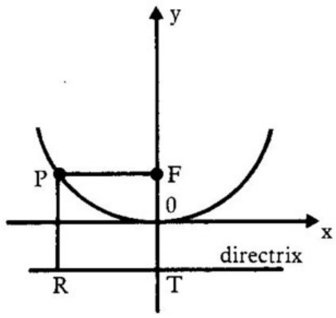
$x^2=4py$

where p is a constant called the focal distance.

It has a quadratic sort of form, but has been panel-beaten into a more useful shape for our purposes. We have to rearrange it to get y as the subject:

y=(1/4p)*x²

and the diagram is:




This special form of parabola is the standard one. It is defined as the curve given when a point P moves so that its distance to a fixed point F and to a line RT (called the directrix) are equal. In symbols, wherever you are on the parabola, and wherever you put point P, this holds true:

RP=PF

where RP is the line through P at right angles to the directrix.

Our equation above is for the parabola when you draw it on Cartesian coordinates with the point O (not point T) as the origin.

The focal distance p is equal to OF, so it must also be equal to OT because of the definition of the parabola.

And once again, we have everything we need for a program.

*Program 46 PARABOLA*

```
   5 REM PROGRAM 46 PARABOLA
  10 BORDER 5: PLOT 0,88: DRAW 2
55,0: PLOT 128,0: DRAW 0,175
  20 LET p=3
  30 FOR x=-10 TO 10 STEP 0.1
  40 LET y=x*x/(4*p)
  50 PLOT (x*8)+128,(y*8)+88
  60 NEXT x
```

Experiment with various values of p to get parabolas with different amounts of bend.

The parabola has proved very important to astronomers too, but in a different way from the ellipse. Astronomers like to use big telescopes, and the way to make big telescopes is by using a curved mirror instead of a lens to focus the light rays. This type was first developed by Sir Isaac Newton (the great English genius), and the curve of the mirror is based on a parabola. (A parabola in three dimensions is called a paraboloid.) The parabola has the property of reflecting all the rays parallel to its axis to the focus.

Which leaves us with one conic to go, the hyperbola.

It might be interesting to look over a complete list of eccentricities for our conics:

| CONIC SECTION | ECCENTRICITY RANGE |
|---|---|
| Circle | Ø |
| Ellipse | Ø<e<1 |
| Parabola | 1 |
| Hyperbola | e>1 |

It might also be interesting to look at the general equation of the hyperbola compared with that of the ellipse.



Let's compare the hyperbola and the ellipse in general equation form:

*Ellipse*

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$$

*Hyperbola*

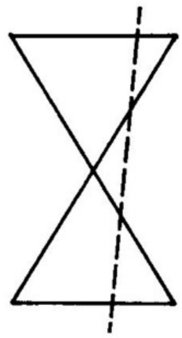
$$\frac{x^2}{a^2}-\frac{y^2}{b^2}=1$$

Very similar you will agree: just that minus sign. And there are some interesting differences. First, we have to consider the cone that is at the bottom of all of this, because it was never as simple as you thought it was!

Do you recall the four cones we drew out to demonstrate the four conic sections (page 130)? In the first three, that cone is sufficient, but once you cut at an angle greater than the angle of the cone itself and begin to get hyperbolas you have to include the other cone, the cone that is upside down and standing on top of the original cone.

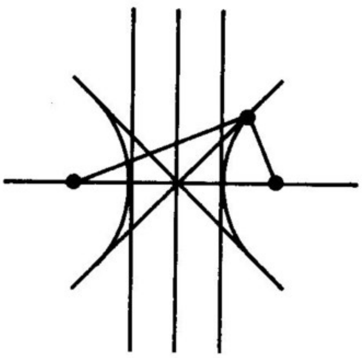
To see this other cone, take a pencil or something else long and thin. Hold it in the middle between thumb and index finger. Now hold the end of the pencil and rotate it in a circle. You will find that the far end of the pencil sweeps out a cone surface in the air, but so does the nearer half of the pencil (because its end moves in a circle too!)

The upshot of all this is that the hyperbola has two curves, because we are looking at the section of two cones.



And in the case of the other conic sections, the angle of cut means that the other cone escapes without being cut, and there is only one curve.

And when you do cut the hyperbola, the curve is as below:



Each of the two branches of the hyperbola has a focus. The general equation only holds true if the coordinate system used in drawing a graph of the curve has its origin at O. As you can see, each branch comes in from infinity and goes out to infinity, getting ever closer to one of the diagonal lines (called asymptotes), but never actually reaching it.

The definition of a hyperbola is a curve which has a constant difference of distances from two fixed points, so that

$$ABS(PF_2 - PF_1) = ABS(PF_1 - PF_2)$$

It must be possible to rearrange our general equation to get y as the subject, and so to exploit it for program purposes, but there arises a potential problem. Watch carefully:

Starting with the general equation, we have

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$$

Rearranging,

$$\frac{y^2}{b^2} = \frac{x^2}{a^2} - 1$$



And multiplying both sides by $b^2$,

$$y^2 = \frac{x^2b^2}{a^2} - b^2$$

And taking out a factor of $b^2$ on the RHS, to form a bracket,

$$y^2 = b^2((x^2/a^2) - 1)$$

And finally, taking square roots,

y = ±b*SQR((x*x/a*a) − 1)

And we're all set. It must be remembered, however, that we must avoid situations where the bracket that we're going to take a square root of goes negative. In other words, $x^2/a^2$ must always remain greater than 1. (Check line 35 in the program below.)

*Program 47 HYPERBOLA*

```
   5 REM PROGRAM 47 HYPERBOLA
  10 BORDER 5: PLOT 0,88: DRAW 2
55,0: PLOT 128,0: DRAW 0,175
  20 LET a=5: LET b=5
  30 FOR x=-10 TO 10 STEP .1
  35 IF ((x*x)/(a*a))<1 THEN  GO
 TO 80
  40 LET y1=b*SQR (((x*x)/(a*a))
-1)
  50 LET y2=-b*SQR (((x*x)/(a*a)
)-1)
  60 PLOT (x*8)+128,(y1*8)+88
  70 PLOT (x*8)+128,(y2*8)+88
  80 NEXT x
```

And the safety measure that hops the program over dangerous values:

35 IF ((x*x)/a*a))<1 THEN GO TO 80

You will find that this program is more sensitive about the values of a and b you can use. But again, your Spectrum won't burst into flames if you put in an out-of-range value, so don't be afraid to explore the limits of the program, and play about with the variables so that the program will do what you want it to.

And that just about wraps it up for the conic sections. They came from solid geometry (solid just means 3-D) and found many applications in science and engineering. You could certainly use the parabola in games designing, because (as we shall see later) the parabola is the curve followed by a projectile. That means that a cannonball, or any other kind of object thrown through the air, moves in a curve that is a parabola.

I must stress that the equations we've used to describe the four conic sections are only true if you consider their graphs to be drawn on a Cartesian coordinate system (regular graph paper with the x-direction at 90° to the y-direction). The origin of that coordinate system must be where we have put it, which means at the centres of circles and ellipses, at the vertex of a parabola and between the two parts of the hyperbola.

But you know there are other coordinate systems. One major coordinate system that we are going to have a look at is the one used in radar screens and star maps and maps of Antarctica.

The idea is that, instead of the directions x and y, you have an origin with lines radiating out from it in all directions and concentric circles going round it. (Concentric just means circles all drawn with their centres in the same place.)

It might not seem too promising an idea but in fact it makes our equations simpler and is a great help in Spectrum programming.

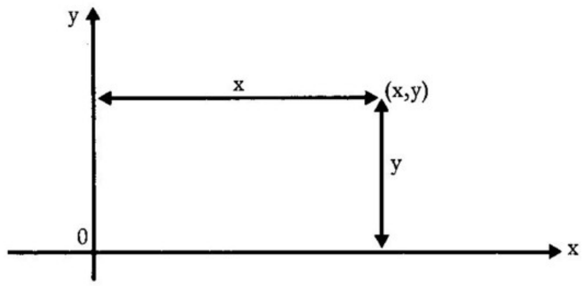
# 8

# POLAR COORDINATES

## 1 Polar coordinates

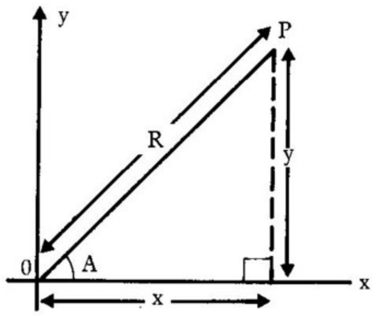
As I sneakily explained after you'd read about conic sections and how to do them on Cartesian coordinates, there is an easier way! Consider this notion first, and I'm sure you'll come to agree with me:

If you have a point on a plane and an ordinary set of Cartesian coordinates, then you can call that point (x,y).




Now try drawing a line from the origin to the point and, instead of thinking of the point as so-far-along and so-far-up, think instead of the length of the line from origin to point and the angle that line makes with the x-axis. Take a little time off and maybe take a pencil and paper, so that you can prove to yourself that any point you like can be specified not only by x and y values, but also by length and angle values. It is usual to call these variables R (for length) and $\theta$ (for angle), but we'll have to alter the theta, $\theta$, into an ordinary letter that the Spectrum will recognise as a variable. Let's call it A (for angle).

You can see on the diagram below that I've drawn both coordinate systems, because it would be a good idea to be able to convert from one to the other (much as you can convert from metres to feet and back).




A conversion can be made by considering the geometry of the situation. We have a right-angled triangle, and so we can use both Pythagoras' Theorem and the trigonometric ratios. It follows that

$x^2 + y^2 = R^2$

or, if you like,

$R^2 = x^2 + y^2$

It also follows that

$\text{Tan } A = y/x$

So that we have a way of getting both R and A if we know x and y. But what if we know R and A and want to work out x and y? In that case, we need these equations:

$x = R \text{ Cos } A$ (from $\text{Cos } A = x/R$)

and

$y = R \text{ Sin } A$ (from $\text{Sin } A = y/R$)

Now we can convert our cumbersome and inelegant equation for a circle (say) into a form far more suited to the Spectrum.



This length-and-angle system is called the polar coordinate system, and if you have a scientific calculator, you might be lucky enough to have a couple of buttons marked P and R. The P stands for Polar and the R stands for Rectangular (meaning the x,y system), and it shows that you can easily convert between them. In fact, what such calculators are doing is to use the very same equations as those we've just looked at.

For example, if you take the equation of the circle in rectangular form, you have

$x^2 + y^2 = R^2$ (where R is the radius)

You can see that it's a perfect thing to convert into Polar form, because you have automatically from the conversion equations,

$x = R \text{ Cos } A$
$y = R \text{ Sin } A$

Furthermore, if we're going to do a program, all we have to do is run the angle A round one complete revolution, and since it's radians we have to use with angles on the Spectrum, the FOR loop we use will have the form,

FOR A=0 TO 2*PI

The problem is that 2*PI is a shade over six, so we don't get many points plotted unless we step .1 or something small. I favour stepping at intervals that are multiples of PI, because that way you get the points in a symmetrical pattern. Run program 48 to see how superior this polar system is to the rectangular one.

*Program 48 POLAR CIRCLE*

```
 5 REM PROGRAM 48 POLAR CIRCLE
10 LET R=50
20 FOR A=0 TO 2*PI
30 LET x=R*COS A
40 LET y=R*SIN A
50 PLOT x+128,y+88
60 NEXT A
```

Now do you see the problem with the step? Seven points, and not particularly symmetrical either! Try STEP PI/10 in line 20 and you get 20 plots (10 for each PI of the 2*PI) all nicely spaced. Increase the denominator to choose how many plots you want at will, and you can even test the circle with,

70 CIRCLE 128,88,50

Of course, you can alter the radius of the circle by setting R equal to a different value, and you can draw semicircles or other fractions of circles by changing the limits of the FOR . . . NEXT loop. Try, for example,

20 FOR A=PI TO 5*PI/2 STEP PI/50

See what I mean?

And you can move the centre of the circle just as easily by changing 128 and 88 in line 50.

But it gets better!

You can do ellipses quite easily, because for ellipses

x=R*COS A          becomes x=aa* COS A

and,

y=R*SIN A          becomes y=bb* SIN A

(Because I was using the variable A for the angle I couldn't use it for the semimajor axis which I would normally call a, but if I put aa and bb I think you will realise what I mean.)

So we have:

*Program 49   POLAR ELLIPSE*

```
 5 REM PROGRAM 49 POLAR
   ELLIPSE
10 LET aa=127: LET bb=87
20 FOR A=0 TO 2*PI STEP PI/20
30 LET x=aa*COS A
40 LET y=bb*SIN A
50 PLOT x+128,y+88
60 NEXT A
```

This time there's no problem with any dimension that will fit on the screen: make aa any value you like between zero and 127 and make bb any value between zero and 87.



Furthermore, you can try a few polar equations out for other shapes. There's a famous shape called the cardioid (named for the fact that it's heart-shaped) which crops up in acoustics. Its polar equation is

$R=2a(1-\text{Cos }\theta)$

or, in Spectrum form,

R=2*aa*(1-COS A)

It's as easy as PI to convert it into a program!

*Program 50   CARDIOID*

```
 5 REM PROGRAM 50 CARDIOID
10 LET aa=30
20 FOR A=0 TO 2*PI STEP PI/50
30 LET R=2*aa*(1-COS A)
40 LET x=R*COS A
50 LET y=R*SIN A
60 PLOT x+128,y+88
70 NEXT A
```

I always think it looks more like a kidney than a heart – perhaps they should have called it a nephroid!

This polar coordinate thing is great, much better for some applications than your common or garden rectangular system.

Try this:

*Program 51   ARCHIMEDEAN SPIRAL*

```
 5 REM PROGRAM 51 ARCHIMEDEAN
   SPIRAL
10 LET aa=15
20 FOR A=0 TO 2*PI STEP PI/50
30 LET R=aa*A
40 LET x=R*COS A
50 LET y=R*SIN A
60 PLOT x+128,y+88
70 NEXT A
```

The spiral, which is named after its inventor, the Greek scientist Archimedes (287–212 BC), has the property that the radius arm (the line from a point on the spiral to the centre) is proportional in length to the angle through which it has rotated. The bigger the angle A, the longer the line R. It can be made easier to see if we make the computer draw the lines in for us. (This can be done with any of our polar graphs quite simply.)

Just add DRAW −x,−y to the PLOT instruction:

```
60 PLOT x+128,y+88: DRAW -x,-y
```

The result is a beautiful pattern (obviously used by nature to construct fossil shells!) made even more beautiful by the fact that you now understand it mathematically.

Then there is the logarithmic spiral given by the equation

$$r = ae^{k\theta}$$

or, in Spectrum language,

r = aa* EXP(k*A)

And if you're into definitions, its a spiral curve where the logarithm of the ratio of any two radius arms is proportional to the angle between them.

Now for the program:

*Program 52 LOGARITHMIC SPIRAL*

```
 5 REM PROGRAM 52 LOGARITHMIC
   SPIRAL
10 LET aa=20: LET k=.2
20 FOR A=0 TO 2*PI STEP PI/50
30 LET R=aa*EXP (k*A)
40 LET x=R*COS A
50 LET y=R*SIN A
60 PLOT x+128,y+88
70 NEXT A
```

When you experiment with different values of aa and k, you will find that the Spectrum comes up with an 'Out of range' error report if you go off the screen to the right or top. But if you go off at the bottom or to the left, it will just bounce the curve back off the edge of the screen in a strange manner. Beware this



effect: it has nothing to do with the curves we're trying to draw, and the moral is to keep the figure away from the edges.

It is interesting to run spirals through more than just one revolution (mathematicians use the word 'convolution' to mean a spiral revolution). So try writing

```
20 FOR A=0 TO 8*PI STEP PI/50
```

so that you get four convolutions.

So far we've been taking the equations of curves I already knew about (because I have been taught them!). But there's nothing to stop us experimenting with polar equations plucked out of the air, trying them in the sort of equation form we've already used, and in the sort of program we've been using. They say that if an infinite number of monkeys sit down at an infinite number of typewriters, sooner or later one will write a Shakespeare play! Maybe it follows that we might throw up a few interesting shapes. Let's have a go!

For instance, this one:

R = aa*COS(2*A)

(LET aa = 80) gives you a butterfly shape. And

R = 80*SIN A

draws you a circle of radius 80, up from the centre of the screen, and goes round it twice. (You can see that it goes round twice if you put in a STEP value that's not a multiple of PI, say STEP 0.1).

Then there's

R = 100*(COS A)*(COS A)

which means R = 100 $\text{Cos}^2$A, the mathematical way of writing Cos A squared. As you can see it's an infinity sign or a pair of spectacles. Or try

R = 30*(1 + COS A)

or even

R = 30*(1 + 3*COS A)

or perhaps

R = 20*(1 + 2*SIN A)

Then there's

R=8Ø*SIN (3*A)

And I'm sure that you can experiment your way to many more spirographic patterns.

## 2 Curve sketching

If you can plot the graph of a function, it can tell you a lot about the way the function goes: remember the way the exponential curve took off quickly?

You already know that some curves are 'periodic', meaning that they repeat themselves after a certain period (examples are the Sin and Cos curves). Is it possible to make satisfactory graphs of polynomials on the Spectrum?

Yes, but we have to arrange it so that it fits the screen, and so that the Spectrum can understand it properly. This means that the figures used to plot the functions must be carefully set up, and the equations put in as strings of x's multiplied together instead of as one x raised to a power:

x↑2 must be written x*x

and

x↑3 must be written x*x*x

Let's ease ourselves into this business with a simple prototype:

*Program 53 CURVE SKETCHER*

```
  5 REM PROGRAM 53 CURVE
    SKETCHER
 10 BORDER 5: LET a=1: LET b=0:
LET c=1: LET d=1
 20 PLOT 0,88: DRAW 255,0: PLOT
128,0: DRAW 0,175
 30 FOR x=-16 TO 16 STEP .125
 40 LET y=x
 50 IF y>-88 AND y<87 THEN  PLO
T (x*8)+128,y+88
 60 NEXT x
```



Try it out and see that you get the straight line you would expect from the equation y=x. (The graph is stretched 8 times.)

Then replace line 4Ø with,

4Ø LET y=(a*x)+b

so that you can plug in any value of a or b you want to test in line 1Ø. This allows you to examine the linear equations.

Then try

4Ø LET y=(a*x*x)+(b*x)+c

letting you examine the various parabolas of the quadratic.

Then you can look at what are called cubic equations where there is a term in x cubed:

4Ø LET y=(a*x*x*x)+(b*x*x)+(c*x)+d

and equations where the highest term is of degree four (called quartic); you'll need to define another coefficient, e, in line 1Ø,

4Ø LET y=(a*x*x*x*x)+(b*x*x*x)+(c*x*x)+(d*x)+e

And so on.

Now that we know what we can do, how do we get useful information out of it?

First it's important to appreciate the fact that a cubic equation gives a new shape. You can see it if we let a=1, b=Ø, c=Ø and d=Ø and have as our equation the general cubic:

4Ø LET y=(a*x*x*x)+(b*x*x)+(c*x)+d

Run it and see the curve come from the bottom of the screen, zig-zag through the origin and go off the top of the screen. It's as if the negative half of a parabola had been swung down. To see a more general view of the cubic curve, plug in the values a=1, b=6, c=4 and d=1Ø.

Now try this. Let line 4Ø become,

4Ø LET y=a*(x−b)*(x−c)*(x−d)

and plug in these values: a=−1Ø, b=3, c=2, d=1

First you find that the curve crosses the x-axis at three points, and then, if you're observant, you notice that it crosses it at the points where x=b, x=c and x=d. Furthermore, you can witness the effect of making a positive (say 1Ø instead of −1Ø) − the curve still cuts the x-axis in three places and those places are the same (because you've done nothing to the coefficients b, c or d) but the curve goes the other way.

The point is that when the curve crosses the x-axis, the value of y is zero, and if the value of y is zero, either a is zero or one of the three brackets is zero. We can see that a is not zero, so therefore with no hesitation we can say that either

$x-b=0$ (i.e. $x=b$)

or

$x-c=0$ (i.e. $x=c$)

or

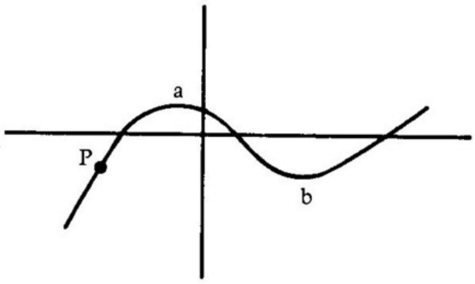
$x-d=0$ (i.e. $x=d$)

So you see that it is possible to extract information from the curve, and if our equation had arisen from a real application, then examining the curve would give us information about that application.

It is a powerful tool in science and engineering and leads us neatly on to a notion that we will meet a bit later, that of maxima and minima.

Isaac Newton (again!) dreamed up a whole branch of mathematics on his own, and nowadays we call it calculus. It has to do with the rates at which things change: like speed being the rate at which distance changes, and acceleration being the rate at which speed changes. Anyway, the graphs of functions are important in seeing how these rates of change work.

For the moment, let's just content ourselves with a jargon lesson:






The curve above has two 'turning points'. The turning point at a is a maximum because it's going from uphill to downhill, and the turning point at b is a minimum because the curve is going from downhill to uphill. (This is not a heavy-duty mathematical definition, but I'm sure you get what I mean.) We say that a curve going uphill has a positive slope (and you can knit that in with what you already know about linear equations), conversely a curve on its way downhill has got a negative slope.

The slope of the curve at these turning points is zero. If that's puzzling, consider this. Suppose a point P travels along the curve from left to right, and suppose also that you've put a ruler up against the curve so that it only touches it at P. (Try drawing it out on a large piece of paper.) Now the slope of the curve at the point P is the same as a line just touching the curve at that point (the slope of your ruler). We call that line just touching the curve a tangent, and it has absolutely nothing to do with the trig function Tan.

You can see that as the point moves along the curve towards point a, the tangent gets more and more horizontal, until when you actually reach point a the tangent is exactly horizontal. If P continues to move towards the right along the curve, the tangent begins to dip downwards and then flattens out again, becoming horizontal at b. So at a and b, the slope (being horizontal) is zero.

Now, between a and b, there is a point where the slope is at a maximum value, and this point is called a point of inflexion.

And for the moment that's all we need to know about curves.

## 3 The solution of triangles

Breaking away from the heady world of functions, etc., let's get down to some hard-nosed calculations. Away from the mind-numbing aspects of the square root of a minus number, far from things that tend to infinity, is the humble triangle, and there are a couple of equations that help us work them out.

Let's suppose, for example, that you have a triangle made by taking three pencils which are all different lengths. Now, you can measure the lengths of the pencils (so you know the lengths of the sides of the triangle), but can you work out the angles? The answer is Yes.

Or supposing you have two pencils laid so that they form an angle of, say, 30° to each other, could you work out how long the third pencil would have to be to make a triangle with the other two? Again the answer is Yes.

And this is how it's done.

*If you know three sides:*

$$\text{Cos A} = \frac{b^2 + c^2 - a^2}{2bc}$$

This formula is known as the cosine rule, and you just plug in the values of the three sides a, b and c.

### *Program 54 COSINE RULE*

```
   5 REM PROGRAM 54 COSINE RULE
  10 INPUT "Side a?";aa
  20 INPUT "Side b?";b
  30 INPUT "Side c?";c
  40 LET R=ACS (((b^2)+(c^2)-(aa
^2))/(2*b*c))
  50 LET A=R*180/PI
  60 PRINT "Angle opposite side
a is          ";A;" degrees"
   5 REM PROGRAM 55 COSINE RULE
     AGAIN
  10 INPUT "Side a?";a
  20 INPUT "Side c?";c
  30 INPUT "ANGLE B (degrees)?";
B
  40 LET R=B*PI/180
  50 LET bb=SQR ((a^2)+(c^2)-(2*
a*c*COS R))
  60 PRINT "Last side is ";bb
```

Test it works by deliberately feeding it an equilateral triangle, which has all three sides the same length and all three angles the same length too (so they are 60° each). You could even test it with a 3, 4, 5 triangle (you know that one is a right-angled triangle), and put in the 3, 4 and 5 in different orders until you come up with 90°.



*If you know two sides and the angle between them:*

$$b^2 = c^2 + a^+ - 2ac \text{ Cos B}$$

This is just a rearranged version of the cosine rule into which you plug your sides and angle.

### *Program 55 COSINE RULE AGAIN*

```
   5 REM PROGRAM 55 COSINE RULE
     AGAIN
  10 INPUT "Side a?";a
  20 INPUT "Side c?";c
  30 INPUT "ANGLE B (degrees)?";
B
  40 LET R=B*PI/180
  50 LET bb=SQR ((a^2)+(c^2)-(2*
a*c*COS R))
  60 PRINT "Last side is ";bb
```

Again, it can be tested with an equilateral or a 3, 4, 5 triangle. Notice that the program takes the trouble to convert the angle to radians, so that the Spectrum can handle it properly, and that I have had to use the variable called bb to hold side b (so that the computer doesn't confuse it with the angle B).

*If you know one side and two angles:*

$$\frac{a}{\text{Sin A}} = \frac{b}{\text{Sin B}} = \frac{c}{\text{Sin C}}$$

This equation (or double equation) is called the sine rule. The way I've written it is just the shorthand version of:

$$\frac{a}{\text{Sin A}} = \frac{b}{\text{Sin B}}$$

and

$$\frac{b}{\text{Sin B}} = \frac{c}{\text{Sin C}}$$

and

$$\frac{c}{\text{Sin C}} = \frac{a}{\text{Sin A}}$$

If you know two of the three angles and remember that the three angles add up to 180°, then you can work out the third angle. And once you know the length of one side you know what size the triangle is, and so the lengths of the other two sides can be got from the sine rule.

In solving a quadratic we are looking for values of x that will satisfy the equation. In solving triangles we are trying to find the values of the three angles and three sides. It is possible to solve all triangles if we know three quantities, and the sine and cosine rules help us.

### *Program 56 SINE RULE*

```
  5 REM PROGRAM 56 SINE RULE
 10 INPUT "Angle A? ";A
 20 INPUT "Angle B? ";B
 30 INPUT "Side c? ";cc
 40 LET C=180-(A+B)
 50 LET aa=cc*SIN (A*180/PI)/SI
N (C*180/PI)
 60 LET bb=aa*SIN (B*180/PI)/SI
N (A*180/PI)
 70 PRINT "Side a is ";aa
 80 PRINT "Side b is ";bb
 90 PRINT "Angle C is ";C
```

To summarise:

*The Cosine Rule*

$a^2=b^2+c^2-(2\star b\star c \text{ Cos A})$

$b^2=c^2+a^2-(2\star a\star c \text{ Cos B})$

$c^2=a^2+b^2-(2\star a\star b\star \text{ Cos C})$

REMEMBER TO USE RADIANS ON THE SPECTRUM.

USE THIS RULE FOR TWO SIDES AND INCLUDED ANGLE.

Notice that the three forms are cyclic, meaning that if you memorise the first version, you can get the second and third versions simply by moving the letters on by one (i.e. when you have an a make it a b, and where you have a letter b make it a c, and when you have a c make it into a: see what I mean about cyclic?).



And with a swift bit of rearranging you can get these:

$$\text{Cos A}=\frac{b^2+c^2-a^2}{2bc}$$

$$\text{Cos B}=\frac{c^2+a^2-b^2}{2ac}$$

$$\text{Cos C}=\frac{a^2+b^2-c^2}{2ab}$$

REMEMBER TO USE RADIANS ON THE SPECTRUM.

USE THIS FORM FOR THREE SIDES.

Note again the cyclic aspect of this trio of equations.

I don't suppose it's necessary to waste space summarising the sine rule; it's all set out nicely for you above, so let me throw in an interesting little titbit for you instead.

The area of a triangle can be found with a lot of messing about with geometry and so on, but let's just use the result:

Let $s=\frac{1}{2}(a+b+c)$ (where a, b and c are the three sides)

This quantity s is half the sum of the three sides and is known as the semi-perimeter.

Then the area of the triangle is given by:

$$\text{Area}=\sqrt{s(s-a)(s-b)(s-c)}$$

So we can solve our triangle to the point where we know all three sides and all three angles. We can then apply our area equation and we've got the lot.

### *Program 57 AREA OF TRIANGLE*

```
  5 REM PROGRAM 57 AREA OF
    TRIANGLE
 10 INPUT "Side a?";a
 20 INPUT "Side b?";b
 30 INPUT "Side c?";c
 40 LET s=(a+b+c)/2
 50 LET area=SQR (s*(s-a)*(s-b)
*(s-c))
 60 PRINT "Area is ";area;" squ
are units."
```

I have produced several of our formulas apparently out of thin air. It is important for you to realise that these formulas have been produced by hard-working mathematicians over the centuries and should not be treated with disrespect.

Since these formulas have been developed from first principles (that means from basics) by complicated arguments and buckets of logic, it would be perfectly possible for me to reproduce them here in glorious black and white. The drawbacks are that you are probably not as interested in the proofs and theorems as you are in the end result, and it would make the book twice as thick and probably twice as expensive. And a waste of money it would be too, because once you get a taste for maths (and a feeling that you can understand it all right), you can get a book out of the library, or buy one, even, and look up the proofs and theorems to your heart's content.

## 4 Trigonometrical equations

Those mathematicians also produced a number of other results which you will find useful whenever you meet equations containing trig terms. For example, one of the relations is:

$$\mathrm{Cos}^2 x + \mathrm{Sin}^2 x = 1$$

or, in Spectrum-ese,

((COS x)*(COS x))+((SIN x)*(SIN x))=1

Now you can check that this relationship is true by using program 58. It runs through the values of x from Ø to 2*PI one degree at a time, and evaluates the expression 'cos squared x plus sin squared x' (which ought to be 1):

*Program 58 TRIG IDENTITY*

```
   5 REM PROGRAM 58
     TRIG IDENTITY
  10 FOR x=0 TO 2*PI STEP PI/180
  20 PRINT AT 0,0;((COS x*COS x)
+(SIN x*SIN x)),x*180/PI
  30 NEXT x
```



which should convince you that the relationship holds true. This is a handy way of testing any relationship over the range in which you want to use it. With trig quantities it follows that, if they hold true in the range Ø to 2*PI, they hold true for any value, because they repeat themselves.

Given the relationship that cos squared x plus sin squared x equals 1, we can solve the following trig equation:

$$\mathrm{Cot}\ x + \mathrm{Tan}\ x = 2$$

Now this is not an equation that holds true for all values of x. It's like the situation where we had a quadratic and were trying to solve it; there's a value of x that will satisfy cot x plus tan x equals 2, and we are trying to find that value of x.

If we start by noting that Cot x=1/Tan x, and that Tan x = $\frac{\mathrm{Sin}\ x}{\mathrm{Cos}\ x}$, then we can rewrite the equation as:

$$\frac{\mathrm{Cos}\ x}{\mathrm{Sin}\ x} + \frac{\mathrm{Sin}\ x}{\mathrm{Cos}\ x} = 2$$

and if we multiply each term by Sin x and Cos x, we get

$$(\mathrm{Cos}\ x)^*(\mathrm{Cos}\ x) + (\mathrm{Sin}\ x)^*(\mathrm{Sin}\ x) = 2^*(\mathrm{Sin}\ x)^*(\mathrm{Cos}\ x)$$

or, if you like,

$$\mathrm{Cos}^2 x + \mathrm{Sin}^2 x = 2\ \mathrm{Sin}\ x\ \mathrm{Cos}\ x$$

Now the magic bit. We know from our trig identity that the LHS of the above equation is equal to 1, so we can substitute:

$$1 = 2\ \mathrm{Sin}\ x\ \mathrm{Cos}\ x$$

Trig identities are so called because they hold true for all values of the variable (so they're identical). Let me introduce you to another one:

$$\mathrm{Sin}(2^*x) = 2^*\mathrm{Sin}\ x^*\mathrm{Cos}\ x$$

which you are at liberty to check out if you like in the same way as Program 58.

It may have struck you that we can employ this new identity to push our solution one stage further on.

$1=2\ \text{Sin x Cos x}$

becomes

$1=\text{Sin } 2x$

or, if you like,

$2x=\arcsin 1$

and the angle whose Sin is 1 is 90°, so that

$2x=90°$

therefore,

$x=45°$

which looks like a solution to me. You can check it by substituting it back in the original equation.

The fact is that you need gallons of practice at playing with trig equations before you get used to them: it's like a Rubik cube. I can offer you a nice list of trig identities that all hold true and which you can use for substitution into any trig equations you come across.

*1 The three forms of $Sin^2x+Cos^2=1$*

| | |
|---|---|
| $\text{Sin}^2x+\text{Cos}^2x=1$ | You can get any one of these |
| $\text{Sec}^2x=1+\text{Tan}^2x$ | from the first one by careful |
| $\text{Cosec}^2x=1+\text{Cot}^2x$ | rearranging. |

*2 Addition formulas*

Sin (A+B) = Sin A Cos B + Cos A Sin B
Sin (A−B) = Sin A Cos B − Cos A Sin B
Cos (A+B) = Cos A Cos B − Sin A Sin B
Cos (A−B) = Cos A Cos B + Sin A Sin B

from which it follows

$$\text{Tan }(A+B)=\frac{\text{Tan A}+\text{Tan B}}{1+\text{Tan A Tan B}}$$

$$\text{Tan }(A-B)=\frac{\text{Tan A}-\text{Tan B}}{1+\text{Tan A Tan B}}$$



*3 Double angle formulas*

Let A=B in the above identities and we get double angle formulas: (A=B=x)

$\text{Sin } 2x=2\ \text{Sin x Cos x}$
$\text{Cos } 2x=\text{Cos}^2x-\text{Sin}^2x$
$\quad=2\ \text{Cos}^2x-1$

*4 Half Angle Formulas*

Let $\theta=x/2$,

$\text{Sin }\theta=2\ \text{Sin }(\theta/2)\ \text{Cos }(\theta/2)$
$\text{Cos }\theta=\text{Cos}^2(\theta/2)-\text{Sin}^2(\theta/2)$
$\quad=1-2\ \text{Sin}^2(\theta/2)$

$$\text{Tan }\theta=\frac{2\ \text{Tan}(\theta/2)}{1-\text{Tan}^2(\theta/2)}$$

*5 Sum and difference formulas (factor formulas)*

Sin A + Sin B = 2 Sin ((A+B)/2) Cos ((A−B)/2)
Sin A − Sin B = 2 Cos ((A+B)/2) Sin ((A−B)/2)
Cos A + Cos B = 2 Cos ((A+B)/2) Cos ((A−B)/2)
Cos A − Cos B = 2 Sin ((A+B)/2) Sin ((A−B)/2)

*6 Product formulas*

Sin A Cos B = ½(Sin(A+B) + Sin(A−B))
Cos A Sin B = ½(Sin(A+B) − Sin(A−B))
Cos A Cos A = ½(Cos(A+B) + Cos(A−B))
Sin A Sin B = ½(Cos(A−B) − Cos(A+B))

Quite a bit to remember, but remembering them is not necessary: you can look them up any time! The main point is that you're aware of their existence. It will help you to see how some trig equations are manipulated if you know these formulas.

So where does the Spectrum come in on all this? Simply this: look back at how in Program 58 (which is only three lines long and probably a lot easier to remember than the trig identities themselves) we used the program to verify an identity, and so whenever you come up with an alternative form of any of these identities you can test whether you've done it properly by making your computer print out an evaluation of each side of the equation for the whole range from 0 to 2*PI. Program 59 is a general program for verifications of this kind. You just plug one side of your identity into line 2Ø before the comma, and the other side of the identity after the comma.

*Program 59 IDENTITY VERIFIER*

```
    5 REM PROGRAM 59 IDENTITY
      VERIFIER
   10 FOR x=0 TO 2*PI STEP PI/180
   20 PRINT COS (2*x),(2*COS x*CO
S x)-1
   30 NEXT x
```

This let's you check one of the double angle formulas in section 3 above.

One of the limits of computation is that you cannot get complete accuracy without an infinite string of decimal places, so you will find that columns generated by Program 59 are not absolutely identical. But don't worry: it's sufficiently similar for our purposes.

Another advantage of Program 59 is that you can check that you have correctly rendered the mathematical forms of the identities that I have given you into Spectrum BASIC forms and got your brackets in the right places.

This approach to mathematics would make a pure mathematician bite off his own head with disapproval. It's the sort of thing that engineers and scientists do, actually using numbers. Seriously though, it's the logical steps from first principles to a final result that make the mathematical process and would constitute what mathematicians would recognise as a proof. They have a word for this kind of computation approach (beloved of engineers and other practical people) and that word is empirical (from the Greek, meaning 'the way of experience'). The equivalent English expression is 'Suck it and see.'

Well, we have sucked the identities, and we have seen them to be more or less identical.

# 9

# COMPLEX NUMBERS

## 1 Complex numbers

We now open a chapter that is going to be easy to understand. I always thought this was the easiest subject with the most difficult sounding name. Let me put you right about it.

1. A real number is a number found on the Real Line. It can be a fraction, integer, decimal, negative or irrational. Any usual number you know about is a real number.
2. An imaginary number, not found on the Real Line, is any multiple of the square root of minus 1, and I am going to explain it fully in a moment.
3. A complex number is a two-part number consisting of a real number and an imaginary number.

But what do they look like? What do they mean? How do you use them?

Consider this quadratic:

$$x^2+x-6=0$$

we get, using our formula and the fact that $a=1$, $b=1$ and $c=-6$,

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

so that

$$x=\frac{-1\pm\sqrt{(1*1)-(4*1*-6)}}{2*1}$$

which is

$$x=\frac{-1\pm\sqrt{1-(-24)}}{2}=\frac{-1\pm\sqrt{25}}{2}$$

and therefore

$$x=\frac{-1+5}{2} \quad \text{and} \quad x=\frac{-1-5}{2}$$

so that

$$x=2 \quad \text{and} \quad x=-3$$

And we see that we have a couple of roots, one of which is positive and one of which is negative. *Both of them are real.* But try the next quadratic (and in this one the discriminant $(b^2-4ac)$ is less than zero):

$$2x^2-x+1=0$$

Here, $a=2$, $b=-1$ and $c=1$, so that $(b^2-4ac)=(1-8)=-7$
And when we substitute it into our quadratic solver, we get

$$x=\frac{-(-1)\pm\sqrt{(-7)}}{4}$$

So that,

$$x=\frac{1+\sqrt{-7}}{4} \quad \text{and} \quad x=\frac{1-\sqrt{-7}}{4}$$

And we're stuck, because we can't take the square root of a negative number. And stuck it stayed for centuries until someone decided they were fed up with letting the fox get away and said why don't we just factorise the root whenever it's of a negative number, so that we get

$$\sqrt{-7}=\sqrt{7}*\sqrt{-1}$$

Then we could evaluate $\sqrt{7}$ in the normal way and just write a symbol next to it to indicate the square root of minus one. And the symbol they chose was j (only some of them thought i was a better choice). So that, going back to our quadratic, we'd write:

$$x=\frac{1+\sqrt{7}*j}{4} \quad \text{and} \quad x=\frac{1-\sqrt{7}*j}{4}$$

and since root seven is about 2.6, we have:

$$x=\tfrac{1}{4}+j(2.6/4) \quad \text{and} \quad x=\tfrac{1}{4}-j(2.6/4)$$

Looking at the first root, we recognise ¼ as a real number, and



we note that $j(2.6/4)=j(0.65)$, where j represents $\sqrt{-1}$, is an imaginary number, and the whole thing, $\tfrac{1}{4}+j(0.65)$, is a number with both real and imaginary parts, a complex number.

As I say, you will often find the letter i used to represent the square root of minus 1. But many people find i a bad choice, because when they discovered electricity they started calling the current in a wire by the letter i, and when you use complex number analysis of electric circuits (which is a lot of fun!) you certainly don't need to be lumbered with one i meaning current and another i meaning root minus 1. So in this book root minus 1 is j.

The first program we can get on with is one that handles all roots of quadratics, be they real or complex. And how can we do that?

Well, we must make the Spectrum print up a letter j whenever it would normally blow its mind by trying to extract the root of a negative number. Take a look back at Program 42 on page 126 and consider Program 60 as a development of it:

*Program 60 QUADRATIC ALL ROOTS*

```
   5 REM PROGRAM 60
     QUADRATIC ALL ROOTS
  10 BORDER 5: FOR n=0 TO 7: REA
D r: POKE USR "a"+n,r: NEXT n: D
ATA 112,16,112,64,112,0,0,0
  20 PRINT AT 2,7;"ROOTS OF QUAD
RATICS"
  30 INPUT "ENTER coefficient a:
 ";a: PRINT AT 6,6;a;"*x   + "
  40 INPUT "ENTER coefficient b:
 ";b: PRINT AT 6,14;b;"*x + "
  50 INPUT "ENTER coefficient c:
 ";c: PRINT AT 6,22;c;"  =  0"
  60 IF b*b>=4*a*c THEN  LET r1=
(-b+(SQR ((b*b)-(4*a*c))))/(2*a)
  61 IF b*b>=4*a*c THEN  LET r2=
(-b-(SQR ((b*b)-(4*a*c))))/(2*a
): GO TO 70
  65 LET R=-b/(2*a)
```

```
66 LET I=SQR ((4*a*c)-(b*b))/(
2*a)
67 PRINT AT 15,3;R;" + j(";I;"
) "
68 PRINT AT 17,3;R;" - j(";I;"
) "
69 PAUSE 0: CLS : GO TO 20
70 PRINT AT 14,6;"FIRST ROOT
";R1
80 PRINT AT 16,6;"SECOND ROOT
";R2
90 PAUSE 0: CLS : GO TO 30
```

Line 65 makes the Spectrum work out the real part, and line 66 makes it work out the imaginary part's coefficient (the bit that appears in brackets), by first multiplying the discriminant by minus 1, so that $(b^2-4ac)$ becomes $(4ac-b^2)$. Lines 67 and 68 then print out the answers, remembering to put in a j symbol to compensate for that trick with the discriminant.

Try it with a = 1, b = −6 and c = 34 and see what you get.

So, we can write a general form for our complex number:

$$x + jy$$

where x is any real number and y is any real number. A few examples would be:

$3 + j4$
$2 - j10$
$-5 + j0.5$
$\pi - j\frac{1}{8}$

and so on.

So now you know what real and imaginary numbers are, and that complex numbers have a real and an imaginary part. In a while we'll be seeing how it's possible to do arithmetic with complex numbers (that is, add them, subtract them, multiply and divide them), but, for now, let's look at our symbol j.

It's clear from what we've said already that

$$j = \sqrt{-1}$$



That means that whenever we find a number that is the square root of a minus number, we can do our factorising ruse, so that

$$\sqrt{-64} = j\sqrt{64} = j8$$

or,

$$\sqrt{-0.32} = j\sqrt{0.32} = j0.5657$$

But we can do algebra with j itself, since if

$$j = \sqrt{-1}$$

then, squaring both sides,

$$j^2 = -1$$

and, multiplying both sides by j,

$$j^3 = -1 * j = -j$$

and again,

$$j^4 = -1 * j^2 = -1 * -1 = 1$$

And to summarise that:

$j = j$
$j^2 = -1$
$j^3 = -j$
$j^4 = 1$

And higher powers repeat this pattern. Program 61 will print out any power of j for you:

*Program 61 POWERS OF j*

```
 5 REM PROGRAM 61 POWERS OF j
10 PRINT AT 0,11;"POWERS OF j"
20 INPUT "ENTER power (pos int
)   ";n
25 IF n-INT n<>0 THEN  GO TO 2
0
30 IF (n/4)-INT (n/4)>=0 THEN
 LET a$="1"
31 IF (n/4)-INT (n/4)>=0.25 TH
EN  LET a$="j"
```

```
  32 IF (n/4)-INT (n/4)>=0.5 THE
N  LET a$="-1"
  33 IF (n/4)-INT (n/4)>=0.75 TH
EN  LET a$="-j"
  40 PRINT AT 10,12;"j^";n;" =";
a$
  50 PAUSE 0: CLS : GO TO 20
```

Lines 40 to 43 sort out the power to see if it's a multiple of 1, 2, 3 or 4, by taking a look at what's left over when you subtract an integer from a decimal. (Divide any integer by 4, and your number will end in a decimal part which is either .25, .5, .75 or zero.)

Line 30 is there to stop you putting in a power that is not an integer.

These powers of j might not seem like a big deal in themselves, but it does get interesting once you realise that, if you get a power of j appearing in any equation, you can replace it with what that power is equal to. For example, if you come across any $j^2$ terms, you can replace them with $-1$.

You should be able to spot by now when I'm working round to something. The fact is that I'm laying the foundations for a spot of arithmetic of the complex number kind.

First we have addition. Let's take two complex numbers:

$a+jb$ and $c+jd$

and then let's add them together. Easy. All you have to remember is that you add the real parts and imaginary parts separately, so that,

$$(a+jb)+(c+jd)=a+c+j(b+d)$$

And you can probably already see that subtraction is:

$$(a+jb)-(c+jd)=a-c+j(b-d)$$

with the reasoning that, again, the real parts and imaginary parts are treated separately.

Now take a look at this. Suppose you have two binomials (expressions involving two terms), and you want to multiply them together. Two linear expressions, for example:

$$(ax+b)*(cx+d)$$



Then you would multiply each of the terms of the first binomial by each of the terms of the other binomial, drawing in curved lines to illustrate it,

$$(ax + b) \quad * \quad (cx + d) \quad = (acx^2 + adx + bcx + bd)$$

It doesn't take a genius to see what happens when you apply this concept to complex numbers (which are of a binomial form in that they have two numbers):

$$(a+jb)*(c+jd)=(ac+jad+jbc+j^2bd)$$

which we can simplify by substituting $-1$ for $j^2$

$$(ac+jad+jbc-bd)$$

and collecting together the real terms on the left and imaginary terms on the right,

$$(ac-bd)+j(ad+bc)$$

which is the product we were after. And it should not be difficult to write small programs that allow us to perform these three complex operations, complex in the sense of operations with so-called complex numbers.

*Program 62 COMPLEX ADDITION ETC.*

```
   5 REM PROGRAM 62
     COMPLEX ADDITION
  10 PRINT "FIRST NUMBER?"
  20 INPUT "Real Part? ";R1
  30 PRINT AT 10,2;R1
  40 INPUT "Imaginary Part? ";I1
  50 PRINT AT 10,2;R1;" + j(";I1
;")"
  60 PRINT AT 0,0;"SECOND NUMBER
?"
  70 INPUT "Real Part? ";R2
  80 PRINT AT 10,2;R1;" + j(";I1
;")  PLUS   ";R2
```

```
  90 INPUT "Imaginary Part? ";I2
: CLS
 100 PRINT AT 10,2;R1;" + j(";I1
;") PLUS  ";R2;" + j(";I2;")"
 110 PRINT AT 14,13;"EQUALS"
 120 PRINT AT 18,12;R1+R2;" + j(
";I1+I2;")"
```

All you have to do is change line 120 into:

```
120 PRINT AT 18,12;R1−R1;" + j(";I1−I2;")"
```

and you have got yourself a subtraction variant. (Of course, it makes more sense visually if you replace the word PLUS in lines 80 and 100 with the word MINUS.)

Furthermore, following closely the reasoning above regarding the prescribed method of multiplication, we can do:

```
120 PRINT AT 18,12;;(R1*R2)−(I1*I2);"+j(";
    (R1*I2)+(R2*I1);")"
```

and with the proviso that you alter the PLUSes into TIMESes you have a program that multiplies complex numbers. Neat eh?

There remains a little matter of division. First check this out. If you play about with Program 62 in multiplication mode you may find that an interesting piece of empirical information dawns upon you.

An occasional answer comes up which has its j coefficient zero, and when a complex number has a zero imaginary part, all that's left is a real number.

The condition for the j coefficient to be zero is:

$$R1*I2+R2*I1=0$$

obviously, since the expression on the LHS is the very one we've been using to generate the j coefficient in our program. And if we rearrange the above equation we get:

$$R1*I2=-R2*I1$$

and finally

$$R1/R2=-(I1/I2)$$

So therefore, if we can find numbers where the above expression



holds true, then we've found a pair of complex numbers which when multiplied together give a simple real number. I have such an example right here:

$$(3+j5) \quad \text{and} \quad (3-j5)$$

easier to spot in practice than in theory. One is the same as the other except its connecting sign is − instead of +, and any pair like this when multiplied together give a real number. The pair is important to us, and we say that one complex number is the complex conjugate of another complex number if the above relationship holds true.

Now, I don't have any idea how to divide a complex number by another complex number, but using this complex conjugate idea, I can multiply top and bottom of a complex division by the conjugate of the bottom and reduce it to a real quantity.

In other words,

$$\frac{(2+j8)}{(3+j4)} \text{ is equal to } \frac{(2+j8)*(3-j4)}{(3+j4)*(3-j4)}$$

and, as we've just explained, $(3+j4)*(3-j4)$ is a real number. Using Program 62 I find that $(3+j4)*(3-j4)=25$.
and $(2+j8)*(3-j4)=(38+j16)$, so we have as our answer:

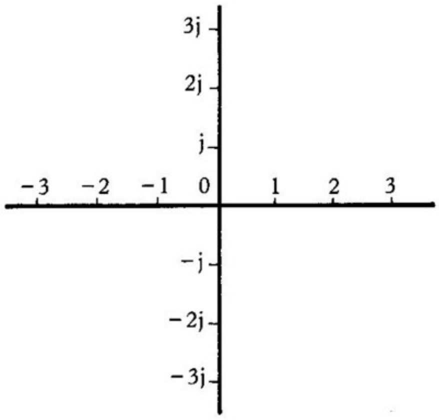
$$\left(\frac{38}{25}+j\frac{16}{25}\right)$$

Which appears to have helped us achieve our end. How about a division program? No problem! Based entirely on these principles which we have discussed I have pleasure in announcing yet another alternative line 120. (I suggest you take a big breath and alter lines 80 and 100 from PLUS to OVER.)

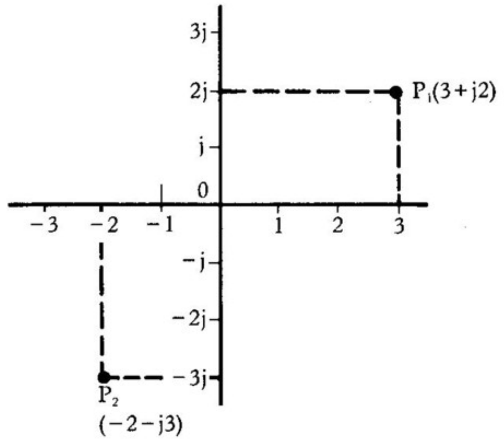
```
120 PRINT AT 18,12;((R1*R2)+(I1*I2))/((R2*R2)+(I2*I2));
    "+j(";((R2*I1)−(R1*I2))/((R2*R2)+(I2*I2));")"
```

So now we know how to do all the arithmetical things to complex numbers, I can show you the Argand diagram. This is a method of showing complex numbers on a graph. Dead simple: all you have to do is take our old friend the Real Line and make it into the x-axis of a graph. Then take the imaginary line and turn it into a y-axis. The resulting graph is an Argand Diagram. (Imaginary lines are just like real lines except that they're marked off in multiples of j instead of multiples of 1.)

Let me show you:




And the four quadrants you get here make up what is called the complex plane. It's called that because you can represent any complex number on it just by measuring along as far as the real coefficient of your number, then measuring up (or down) as far as the imaginary coefficient. Like this:



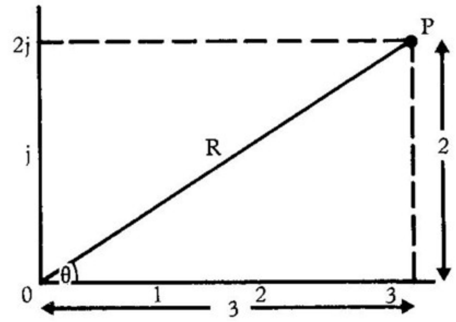



I've drawn in two points on the complex plane, $P_1$ and $P_2$, which represent two different complex numbers: $P_1$ represents $3+j2$, and $P_2$ represents $-2-j3$.

And, fairly obviously, you can plot any complex number at all on the complex plane.

Does some of this seem curiously familiar to you? It might well do, because it's just what we were doing with graphs of x versus y way back in Chapter 4. Do you remember what we went on to do with the Cartesian coordinate system? We brought in the idea of polar coordinates. Again, you don't need to be much of a genius to see that we can represent complex numbers on a polar diagram, so let's investigate that.

Below is a representation of just the top right quadrant of the complex plane, and the number $3+j2$.




From which you ought to be able to see that,

$$R^2 = 3^2 + 2^2$$

or

$$R = \sqrt{9+4} = \sqrt{13} = 3.606$$

And,

$$\operatorname{Tan} \theta = 2/3 \quad \text{or} \quad \theta = \operatorname{Tan}^{-1}(2/3) = 33.7^\circ$$

And in fact it is an alternative form for $3+j2$ to be represented $3.606\ \underline{|33.7^\circ}$

If we have a complex number z, then we can write:

$z = a + jb$ (standard form)
$z = r(\text{Cos}\ \theta + j\ \text{Sin}\ \theta)$ (polar form)
where $r = \sqrt{a^2 + b^2}$
and $\theta = \text{Tan}^{-1} b/a$
Also $a = r\ \text{Cos}\ \theta$ and $b = r\ \text{Sin}\ \theta$

Which allows us to interconvert standard to polar and vice versa. And it would not be beyond our capabilities to construct a program to do it.

However, I thought it would be best to wait until you have heard of a third way of expressing a complex number. It's called the exponential form and is derived from the polar form. Being the exponential form, you'd expect it to be e to-the-power-something, and being a representation of a complex number, you'd expect it to involve j. You'd be right:

$r\ e^{j\theta}$

where r is the same as it is in polar form, and theta is the angle appearing in the polar form except that it must be in radians.

Conversions of polar to exponential and vice versa are easy:

$z = re^{j\theta} = r(\text{Cos}\ \theta + j\text{Sin}\ \theta)$

If for example we have,

$z = 3(\text{Cos}\ 45° + j\text{Sin}\ 45°)$

can be written as,

$z = 3e^{j\frac{\pi}{4}}$

The reason for having all these different forms is that, when people had to calculate things by hand (and brain!) it was found to be easier to add and subtract in standard form, easier to multiply and divide using the polar form, and easy to take the logarithm of a complex number if it is expressed exponentially. Quite a rigmarole, isn't it? Computer power lets us choose how we will, though.

For the record, when multiplying complex numbers in polar form, you add the angles and multiply the r's together. For division, we subtract the angles and divide the r's.



And taking the log in exponential form,
if

$z = re^{j\theta}$

then

$\log_e z = \log_e r + j\theta$

One of Sir Isaac Newton's friends, and a refugee from religious persecution in France (after the Edict of Nantes was revoked in 1685), was Abraham De Moivre (1667–1754). He found a relationship known as De Moivre's theorem which can be written:
if

$z = r(\text{Cos}\ \theta + j\text{Sin}\ \theta)$

then

$z^n = r^n(\text{Cos}(n\theta) + j\text{Sin}(n\theta))$

which helps us to raise complex numbers to a power. (De Moivre's theorem works for any value of n, positive, negative or even fractional.)

A useful programming tip which allows you to choose the number of decimal places in a displayed number is connected to the INT function.

Suppose you have a number like 45.346578 and you want to quote it to one decimal place. It would be possible to quote it to zero decimal places using INT:

INT 45.346578

returns 45 (disposing of the unsightly decimal part).

But to get one decimal place, you must first multiply the number by ten (which causes one place of decimals to hop over the decimal point). Then apply INT to the number, which in our example would give 453, and then divide by 10 again to recover our original number. And it can all be done in one line so that if x = 45.346578,

PRINT INT(x*10)/10 gives 45.3

Notice how the use of brackets makes sure that first the bracket is computed, then the INT is taken, and finally the result is divided by 10. Try this:

```
10 LET x=45.346578
20 PRINT x, INT(x*10)/10
```

and you will see what I mean.

And of course, if you are interested to get a number to more than one decimal place, you have to multiply by 100 for two decimal places, 1000 for three decimal places, and so on, always remembering to divide by the same number that you multiplied by.

You can see it working by typing

```
20 PRINT x, INT(x*100)/100
```

giving 45.34, or

```
20 PRINT x, INT(x*1000)/1000
```

giving 45.346.

Notice that this doesn't round the number: it just truncates it. You must take care using negative numbers, since INT −4.3 will return −5, and so you'd have to overcome that problem by adding 1 at a suitable point:

```
10 LET x=-45.346578
20 PRINT x, INT((x*10)+1)/10
```

Then you get the correct truncation.

I just thought I'd better tell you about that before introducing you to the next program, because it explains how lines 70 and 80 work.

*Program 63 COMPLEX NUMBER CONVERSION*

```
   5 REM PROGRAM 63
     COMPLEX NUMBER CONVERSION
  10 INPUT "Real Coefficient? ";
a
  20 INPUT "Imaginary Coefficien
t? ";b
  30 LET r=SQR ((a*a)+(b*b))
  40 LET Thr=ATN (b/a)
  50 LET Thd=(Thr*180/PI)
  60 PRINT a;" + j(";b;")"
  70 PRINT AT 5,0;INT (r*100)/10
0;" (Cos ";INT (Thd*100)/100;" +
 j Sin ";INT (Thd*100)/100;")"
  80 PRINT AT 10,0;INT (r*100)/1
00;" exp(j ";INT (Thr*100)/100;"
)"
```



The variables Thd and Thr stand for (and hold) Theta in degrees, and Theta in radians. And the three forms are printed out on the screen for you to examine.

## 2 Hyperbolic functions: Introduction

These are not really as frightening as they sound. But this one is a real con trick, because you can learn all about hyperbolic functions without ever understanding why they call them hyperbolic functions. In fact, hyperbolic functions have precious little to do with hyperbolas! Strange, isn't it.

Remember Sin, Cos and Tan? Now try this:

| Hyperbolic Function | Symbol | Pronunciation |
|---|---|---|
| Hyperbolic Sine | sinh | 'shine' |
| Hyperbolic Cosine | cosh | 'cosh' – as in mugging. |
| Hyperbolic Tangent | tanh | 'than' – as in thin. |

Obvious again that the h is there after the symbol to remind us that we're dealing with the hyperbolic function and not the normal (or circular, as they're called) functions.

You will certainly find Sin, Cos and Tan on the Spectrum keyboard, but Sinclair didn't bother with the hyperbolic functions. However, that is not going to stop us.

First a few definitions:

$\text{Sinh } x = \frac{1}{2}(e^{x} - e^{-x})$
$\text{Cosh } x = \frac{1}{2}(e^{x} + e^{-x})$

and just the same as circular functions

$$\text{Tanh } x = \frac{\text{Sinh } x}{\text{Cosh } x}$$

$$= \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$

So we can define our own functions in terms of the EXP function, and in Spectrumese we can write:

LET sinhx=((EXP x)−(EXP−x))/2

and

LET coshx=((EXP x)+(EXP−x))/2

and,

LET tanhx=((EXP x)−(EXP−x))/((EXP x)+(EXP−x))

Using these definitions over a suitable range we can draw a small graph of Sinh, Cosh and Tanh using Programs 64, 65 and 66 respectively:

*Program 64 GRAPH OF SINH X*

```
   5 REM PROGRAM 64
     GRAPH OF SINH X
  10 PLOT 0,88: DRAW 255,0: PLOT
 128,0: DRAW 0,175
  20 FOR x=-3 TO 3 STEP .1
  30 LET sinhx=((EXP x)-(EXP -x)
)/2
  40 PLOT (x*30)+128,(sinhx*5)+8
8
  50 NEXT x
```

And

*Program 65 GRAPH OF COSH X*

```
   5 REM PROGRAM 65
     GRAPH OF COSH X
  10 PLOT 0,88: DRAW 255,0: PLOT
 128,0: DRAW 0,175
  20 FOR x=-3 TO 3 STEP .1
```



```
  30 LET coshx=((EXP x)+(EXP -x)
)/2
  40 PLOT (x*30)+128,(coshx*5)+8
8
  50 NEXT x
```

And

*Program 66 GRAPH OF TANH X*

```
   5 REM PROGRAM 66
     GRAPH OF TANH X
  10 PLOT 0,88: DRAW 255,0: PLOT
 128,0: DRAW 0,175
  20 FOR x=-3 TO 3 STEP .1
  30 LET tanhx=((EXP x)-(EXP -x)
)/((EXP x)+(EXP -x))
  40 PLOT (x*30)+128,(tanhx*50)+
88
  50 NEXT x
```

With the last graph you will find that the curve begins quite flat and ends quite flat, and the following addition might prove illuminating:

```
60 PLOT 0,88+50: DRAW 255,0
70 PLOT 0,88-50: DRAW 255,0
```

And you can see the limits between which the tanh function goes, the centre-line being at 88 pixels up from the bottom (middle of the screen) and the limits being plus or minus 50 above and below the line. Remember that line 40 plots the function with a 50 times exaggeration, so that the function's limits are plus 1 and minus 1.

It is rare to find hyperbolic function tables these days, and so you might have a use for this program, which asks you for an x and supplies Sinh x, Cosh x and Tanh x:

*Program 67 HYPERBOLIC FUNCTIONS*

```
   5 REM PROGRAM 67
     HYPERBOLIC FUNCTIONS
```

```
10 INPUT "ENTER x ";x
20 PRINT "If x equals ";x
30 PRINT "Sinh x is    ";((EXP
x)-(EXP -x))/2
40 PRINT "Cosh x is    ";((EXP
x)+(EXP -x))/2
50 PRINT "Tanh x is    ";((EXP
x)-(EXP -x))/((EXP x)+(EXP -x))
60 PRINT : PAUSE 0: CLS : GO T
O 10
```

You can assign any number to x between $-88$ and 88 in this program. Exceed it, and you will find the results generated by the Sinh and Cosh expressions are too big for the computer.

Unlike the circular functions which recur every two pi radians, hyperbolic functions do not come round again.

If we continue our parallels with circular functions, we'll eventually ask ourselves what the inverse of hyperbolic functions are. Remember that the symbol $\mathrm{Sin}^{-1}x$ does not mean 1/Sin x (because $(\mathrm{Sin}\ x)^{-1}$ would mean that). $\mathrm{Sin}^{-1}x$ means 'the angle whose Sine is x'. So, for example, if

$$\mathrm{Sin}\ x = 0.5$$

then

$$x = \mathrm{Sin}^{-1}0.5$$

And do you remember that because of the confusion that can arise with this power of minus 1 notation, and in an attempt to sell the product to a wider audience, it was decided to adopt the 'arc' notation, whereby we write

$$x = \arcsin 0.5$$

instead of

$$x = \mathrm{Sin}^{-1}0.5$$

(It also helps those of us who have not got Chinese typewriters!) So it will come as no surprise to learn that the hyperbolic functions have their arc functions too:

| | | |
|---|---|---|
| arcsinh x | or | $\sinh^{-1} x$ |
| arccosh x | or | $\cosh^{-1} x$ |
| arctanh x | or | $\tanh^{-1} x$ |



In practice, we would have to use these inverse functions to, for example, find the value whose sinh is 1. Let's follow that particular problem through to see how we might solve it.

We want to find $\sinh^{-1} 1$, in other words,

$$\mathrm{Sinh}\ x = 1$$

Therefore

$$\frac{e^x - e^{-x}}{2} = 1$$

and so

$$e^x - e^{-x} = 2$$

Multiplying by $e^x$ throughout

$$(e^x)^2 - 1 = 2{\star}e^x$$

and rearranging, we get a quadratic in $e^x$

$$(e^x)^2 - 2(e^x) - 1 = 0$$

to which we can apply our formula for solving quadratics and get that $e^x$ equals

$$\frac{2 \pm \sqrt{4+4}}{2} \qquad (a = 1, b = -2, c = -1)$$

so that

$$e^x = 1 + \sqrt{8/2}$$

or

$$e^x = 1 - \sqrt{8/2}$$

so that

$$e^x = 2.414$$

or

$$e^x = -0.414$$

And since $e^x$ has to be a positive number, then the first root is the one we're interested in. So taking logs we get 0.881. Therefore

$x = 0.881$ (approx.)

which is our answer. And we can check it by substituting it back into the original equation or, even simpler, running Program 67 with x = 0.881373587 (to quote it at greater length). Then we get the report back from the Spectrum that sinh 0.881373587 is 1. In other words, we have found the number whose hyperbolic sine is 1. And what a very long business it was to do it manually.

It would be perfectly possible for me to drag your fragile interest in hyperbolic functions through the similar rigmaroles that get us at the inverse cosh and tanh functions. But I'm not going to do that, because we have a secret weapon built into Program 68. And, like all effective secret weapons, it's based on firm no-nonsense principles.

If in the previous argument we had let $y = \sinh^{-1}x$, implying that $x = \sinh y$, then we could have followed the calculation through in general terms, arriving finally at $e^y = x + \sqrt{x^2+1}$. Taking natural logs we'd have

$$y = \log_e (x + \sqrt{x^2+1})$$

and since $y = \sinh^{-1} x$, we have

$$\sinh^1 x = \log_e (x + \sqrt{x^2+1})$$

And you can do the same with cosh and tanh, getting

$$\cosh^{-1}x = \log_e (x + \sqrt{x^2-1})$$

and

$$\tanh^{-1}x = \tfrac{1}{2} \log_e ((1+x)/(1-x))$$

which we can render into Spectrumese and use in a program.

*Program 68 INVERSE HYPERBOLIC*

```
  5 REM PROGRAM 68
    INVERSE HYPERBOLIC
 10 INPUT "ENTER x ";x
 20 PRINT "The number x is ";x
 30 PRINT "arcsinh x = ";LN (x+
SQR ((x*x)+1))
 40 STOP
110 INPUT "ENTER x ";x
120 PRINT "The number x is ";x
130 PRINT "arccosh x = ";LN (x+
SQR ((x*x)-1))
140 STOP
210 INPUT "ENTER x ";x
220 PRINT "The number x is ";x
230 PRINT "arctanh x = ";0.5*LN
((1+x)/(1-x))
240 STOP
```



The reason this program is divided into three by STOP statements is that, if you enter any old x, you will find that the program can work out one of the functions, but not another, because the ranges of the functions are very different. As we've already seen, the tanh function is bounded between +1 and −1. So if you ask for $\tanh^{-1}$ of a number outside this range, obviously it cannot be supplied. It will let you get away with 0.999999999 (nine nines), but thinks that 0.9999999999 (ten nines) is one and won't have it.

Using Program 68 is a cinch: just RUN for Arcsinh, RUN 100 for Arccosh, and RUN 200 for Arctanh. To do all three, each time the program stops hit CONT (on the C key) and it will hop over the STOP lines and do the next section.

Now what about hyperbolic cofunctions? This really is going where no man has gone before. You'd have to look a long time before finding one of these on your calculator. But they do crop up occasionally in calculus, and so let's have a brief look at them.

You may have heard the term 'reciprocal' used from time to time: just in case it has escaped your notice, it means '1 over' something. So the reciprocal of 4 is 1 over 4, or a quarter. Similarly, the reciprocal of Tan x is 1/Tan x, or Cot x, and so we have our hyperbolic cofunctions (otherwise known as reciprocal hyperbolic functions):

| | |
|---|---|
| coth x = 1/tanh x | (pronounced 'coth') |
| sech x = 1/cosh x | (pronounced 'shek') |
| cosech x = 1/sinh x | (pronounced 'co-shek') |

I won't waste your time any further with these. It's enough that you know what they are and how they relate to the hyperbolic functions. This will allow you to create Spectrumese definitions like

```
LET cothx=((EXP x)+(EXP -x))/((EXP x)-(EXP -x))
LET sechx=2/((EXP x)-(EXP -x))
```

Of course, if you feel like a bit of exploring sometime . . . But that's up to you: I've got two more revelations to cover before we can bring down the curtain on hyperbolic functions.

First there's the relationship between hyperbolic and ordinary circular trig functions, and then we'll go on to have a look at the hyperbolic equivalents of the identities we listed for circular trig functions.

Do you remember, way back in the section on complex numbers, we had two alternative ways of writing the complex number? After the standard form (a+jb) you could have the exponential form $re^{j\theta}$, and the polar form r(Cos $\theta$+jSin $\theta$). So we can say, if

$$z=r{\star}e^{j\theta}$$

and

$$z=r{\star}(\text{Cos }\theta+j\text{Sin }\theta)$$

then it must be true to say

$$r{\star}e^{j\theta}=r{\star}(\text{Cos }\theta+j\text{Sin }\theta)$$

and, dividing both sides by r, we get

$$e^{j\theta}=\text{Cos }\theta+j\text{Sin }\theta$$

It also works out that

$$e^{-j\theta}=\text{Cos }\theta-j\text{Sin }\theta$$

(Note the way the minus sign appears on both sides: you might like to spend a few minutes seeing how we get that result from basics.)

I'm now going to take the next step (which is going to bring us closer to the definition of hyperbolic functions) of adding these expressions involving the exponential, and at the same time adding their polar equivalents. We get:



$$e^{j\theta}+e^{-j\theta}=\text{Cos }\theta+j\text{Sin }\theta+\text{Cos }\theta-j\text{Sin }\theta$$

The terms in Sin $\theta$ cancel out, leaving

$$e^{j\theta}+e^{-j\theta}=\text{Cos }\theta+\text{Cos }\theta$$
$$=2{\star}\text{Cos }\theta$$

Remembering now our definition of Cosh x:

$$\text{Cosh x}=\frac{e^x+e^{-x}}{2}$$

and substitute j$\theta$ for x,

$$\text{Cosh j}\theta=\frac{e^{j\theta}+e^{-j\theta}}{2}$$

But, from our argument above, we have already decided that

$$e^{j\theta}+e^{-j\theta}=2{\star}\text{Cos }\theta$$

so that, dividing both sides by 2, we get

$$\frac{e^{j\theta}+e^{-j\theta}}{2}=\text{Cos }\theta$$

which in turn is equal to Cosh j$\theta$. We can therefore write this:

$$\text{Cos }\theta=\text{Cosh j}\theta$$

which is the first of our relationships. We can produce the others in the same way – try it yourself – but I'm just going to quote the results:

$$j\text{Sin }\theta=\text{Sinh j}\theta$$

and

$$j\text{ Tan }\theta=\text{Tanh j}\theta$$

Here is a tabulation and alongside the reverse forms have been tabulated too, allowing you to see Tanh in terms of Tan, etc., and the reverse, Tan in terms of Tanh, etc.:

| | |
|---|---|
| Sinh j$\theta$ = j*Sin $\theta$ | Sin j$\theta$ = j*Sinh j$\theta$ |
| Cosh j$\theta$ = Cos $\theta$ | Cos j$\theta$ = Cosh $\theta$ |
| Tanh j$\theta$ = j*Tan $\theta$ | Tan j$\theta$ = j*Tanh $\theta$ |

All this means that we can now set about listing the hyperbolic trig identities, or at least a few of them:

HYPERBOLIC TRIG IDENTITIES

$\text{Cosh}^2\theta - \text{Sinh}^2\theta = 1$
$\text{Sinh } 2\theta = 2*\text{Sinh } \theta*\text{Cosh } \theta$
$\text{Cosh } 2\theta = \text{Cosh}^2\theta + \text{Sinh}^2\theta$
$\text{Cosh}(A+B) = \text{CoshA}*\text{CoshB} + \text{SinhA}*\text{SinhB}$

And you will see that, if you compare these with the ordinary circular identities, the differences are only of sign. These arise when you have a $\text{Sinh}^2$ involved (either directly or indirectly) with the calculation, because you get an extra factor of $j^2$ in the working which is, of course, minus 2 and will alter the sign.

But I suppose all this is getting a mite tedious and involved. It doesn't matter: you're not going to be examined on this stuff, but it does serve to clue you in on maths and what it's really all about. And since this is not a maths text book, we can do what we like. Normally they start simple and get progressively more complicated, but not in this book. There are still a few simple ideas left to dig out, and once we've got them sorted out, we'll be in a much better position.

It hardly needs saying, but since everything in maths is connected to everything else, you might find this interesting.

# 10

# THE FACTORIAL

## 1 The factorial

Occasionally you will find that you open a maths book and, amongst the strange hooked symbols you find there, you sometimes see a number with an exclamation mark after it. This is a factorial. Let me explain. You can calculate the factorial of a number if that number is a positive integer, and the way you do it is by multiplying together all the other positive integers that are less than it. And that's it! For example:

factorial three = 3! = 3*2*1 = 6

or

factorial six = 6! = 6*5*4*3*2*1 = 720

and so on. Easy once you've cracked the code! Let's have a program:

*Program 69 FACTORIAL*

```
 5 REM PROGRAM 69
   FACTORIAL
10 LET F=1
20 FOR x=1 TO 33
30 LET F=F*x
40 PRINT x,F
50 NEXT x
```

And this little routine will calculate the factorial numbers up to factorial 33. It can't cope with greater than 33 because 34 give you a 'Number too big' error report. It's easy to see why when we look at the speed with which this function takes off. Run

Program 69 and see that by the time x has reached 12, x! has been able to outstrip the Spectrum's ability to render numbers in ordinary form, and it has had to resort to scientific notation. By the time it reaches 34 it has outstripped the Spectrum's ability to represent numbers at all!

Even my flash calculator can only manage 69! (which is approximately equal to 1.711224525*10↑98), because it can only show exponents as far as two digits, and 7Ø! is greater than 1Ø↑1ØØ.

This is worth commenting upon if only for the fact that such monstrously huge numbers are so big that they outstrip the number of atoms that exist in the universe. You can't use them for counting, so what can you use them for? The answer is that they crop up in various places in maths, especially in the idea of probability. Think what the odds are, for instance, against your rolling a thousand successive double sixes whilst playing Monopoly! Quite a big number (so big I can't get it out of my calculator either!).

Now that you've got factorials straight, we can move on to a subject that I found a bit difficult to swallow when I first came across it. Not that it's remotely difficult to understand in any academic sense. It's just amazing that there's yet another (and quite simple) way of getting all those other functions we've spent so long discussing. All the Sines and Cosines, the exponentials and logs (and many others) can all be got from a string of numbers called a series.

## 2 Series

The number e that we have been using in the exponential calculations throughout this book is a curious number. We've already noted that its value is 2.718281828 . . . etc., but we can calculate it from a series of numbers. I'll write out the series for you:

$$\frac{1}{0!}+\frac{1}{1!}+\frac{1}{2!}+\frac{1}{3!}+\frac{1}{4!}+\frac{1}{5!}+\frac{1}{6!} \quad \ldots \text{ etc.}$$

In other words, keep adding a list of the reciprocals of the factorials, and you get closer and closer to e.



The first term is 1/0!, and 0! is taken to be 1 (not zero as you might have thought), so that

| | |
|---|---|
| after one term we have as our sum | 1 |
| after two terms we have | 1 + 1 = 2 |
| after three terms we have | 2 + ½ = 2.5 |
| after four terms we have | 2.5 + 0.16666 etc. |
| | = 2.666666666 etc. |

After four terms, we've added 1/4! or 1/(4*3*2*1) which is 1/24 or 0.0416666 etc. bringing our total to: 2.708333333 etc.

The fifth term amounts to Ø.00833333 etc., and if we add that to our total, we get 2.716666666 etc., and so on. Each term adds on a progressively smaller number, and eventually we'd get to our value of 2.718281828 . . . to whatever accuracy we like.

Program 7Ø does it for you automatically, and so all you have to do is this: when the program stops with

n?

enter 1 the first time, enter 2 the second time, 3 the third time, and so on, until you get to 1Ø. The program will do the spadework and give you back approximations of e to 1 + n terms.

When n = 1Ø, the Spectrum's calculation of e is put up (using the expression EXP 1) in line 9.

The subroutine 1Ø to 5Ø you might recognise as a modified form of Program 69, which supplies the appropriate factorials.

*Program 7Ø APPROXIMATION TO e*

```
   2 REM PROGRAM 7Ø
     APPROXIMATION TO e
   3 LET tot=1
   4 FOR a=1 TO 1Ø
   5 GO SUB 1Ø
   6 LET tot=tot+(1/F)
   7 PRINT a,tot
   8 NEXT a
   9 PRINT : PRINT "e",EXP 1: ST
OP
  1Ø LET F=1: INPUT "n?";n
```

```
20 FOR x=1 TO n
30 LET F=F*x
40 NEXT x
50 RETURN
```

And if our Spectrum would let us do it, we'd be able to approximate e to as many places as we wished with this method. As a matter of fact, that original series is a series for $e^x$ (e raised to the power x) when x has been set equal to one, and for a general x, we can write

$$e^x = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \frac{x^n}{n!}$$

The symbol at the end of the series, $x^n/n!$ is called the general term and shows us how to calculate any term: you just set n to increasingly great values to get the terms of the series.

It is strange that an expression like $e^x$ can be approximated by an infinite string of fractions involving factorials, don't you think? I think it's very hard to believe but it is true.

Furthermore, as I said earlier, there are other series approximations:

$$\log_e(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$$

which works when x has a value between $-1$ and 1, and we also have

$$\log_e(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \dots$$

by substituting $-x$ for x in the first equation.

And there are series for our three trig functions and three hyperbolic functions too. At this stage, don't worry about how these series were discovered. They come from something known to the mathematical world as Maclaurin's series, but the explanation of that will have to wait. Instead, let me pluck them out of thin air:

$$\text{Sin } x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \dots$$



$$\text{Cos } x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} \dots$$

$$\text{Tan } x = x + \frac{x^3}{3} + \frac{2x^5}{15} + \dots$$

where x is in radians. Also

$$\text{Sinh } x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} \dots$$

$$\text{Cosh } x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} \dots$$

and there are others. The important thing is that you are aware that these series exist, that some functions can be expanded into a series and evaluated for any given value. This is in fact very similar to what the Spectrum does inside its own processor when you tell it to come up with a SIN or an EXP, etc. (Actually, it uses something called a Chebychev polynomial, but that's a different story!)

The business of how mathematicians came up with these series expansions is also of interest, but you have to know a bit of calculus before looking at it usefully.

You can replace any occurrence of one of these functions with its series expansion in any program you are using.

## 3 Progressions

We've had a look at the idea of a factorial, all integers smaller than a particular integer multiplied together (see section 1 of this chapter) and we've seen that functions of x can be built out of a series of powers of x. Now we are in a position to play with some sequences of numbers. There are two which are of main importance: arithmetic progression and geometric progression.

The names don't mean much. The word 'progression' simply means a sequence of numbers written one after another (usually written separated by commas) so that

1, 2, 3, 4, 5, 6, . . .

is a progression of sorts, and so is

2, 4, 6, 8, 10, . . .

and even

−1, −3, −5, −7, . . .

But what makes an *arithmetic* progression is that you can always get the next number by adding on a constant number, so that

2, 4, 6, 8, 10, . . .

is an arithmetic progression because you add 2 to the first number to get the second, and you add 2 to the second number to get the third, and so on. If I were to ask you what the next number in that arithmetic progression is (after 10) you could tell me, because you would know that it must be 10 plus 2, which is 12. Arithmetic progressions are known to mathematicians as APs. Geometric progressions are GPs.

So what is a geometric progression? You get an AP from *adding* on a constant number each time to make the next figure in the sequence. With a GP you *multiply* by a constant number each time to make the next figure. An example is:

2, 4, 8, 16, 32, . . .

Here you are multiplying by 2 each time.

The constant number you add each time in an AP is called the common difference, and the constant number you multiply by each time in a GP is called the common factor.
Here's a GP with common factor 3:

3, 9, 27, 81, . . .

And it doesn't just work with integers. Your common difference or common factor can be fractional, giving something like:

1, 1½, 2, 2½, . . . (common difference is ½)

which is an AP, or

1, 0.5, 0.25, 0.125, . . . (common ratio is 0.5)

which is a GP.



You can write a program that makes your Spectrum generate progressions. Program 71 asks you for a 'G' for geometric progression or 'A' for arithmetic progression. Depending on your choice, you are asked to specify the common difference or common ratio. Then you have to tell the Spectrum where you want it to start the progression from: in other words you have to enter the first term.

Then your machine will list the first few terms of the progression and give you the value of the term you specify as well as the sum of all the terms as far as that term. Clear? If not, play with Program 71.

*Program 71 PROGRESSIONS*

```
   5 REM PROGRAM 71 PROGRESSIONS
  10 BORDER 5: INPUT "Do you wan
t Arithmetic or          Geometric?
 ";a$
  20 IF a$(1)="g" OR a$(1)="G" T
HEN  GO TO 150
  30 PRINT "ARITHMETIC PROGRESSI
ON"
  40 INPUT "ENTER first term ";a
  50 PRINT "First Term is ";a
  60 INPUT "ENTER common differe
nce ";d
  70 PRINT "Common Difference is
 ";d
  80 PRINT : PRINT "Progression
is:"
  90 PRINT : PRINT "Term No.";"T
erm"
 100 FOR x=1 TO 10: PRINT x,a+((
x-1)*d): NEXT x
 110 INPUT "ENTER Any Term No. "
;n
 120 LET nth=a+(n-1)*d: PRINT :
PRINT "Term No. ";n;" is ";nth
```

```
 130 LET Sn=(n/2)*((2*a)+((n-1)*
d)): PRINT : PRINT "Sum to ";n;"
 terms is ";Sn
 140 STOP
 150 PRINT "GEOMETRIC PROGRESSIO
N"
 160 INPUT "ENTER first term ";a
 170 PRINT "First Term is ";a
 180 INPUT "ENTER common ratio "
;r
 190 PRINT "Common Ratio is ";r
 200 PRINT : PRINT "Progression
is:"
 210 PRINT : PRINT "Term No.";"T
erm"
 220 FOR x=1 TO 10: PRINT x,a*(r
^(x-1)): NEXT x
 230 INPUT "ENTER Any Term No. "
;n
 240 LET nth=a*(r^(n-1)): PRINT
: PRINT "Term No. ";n;" is ";nth
 250 LET Sn=a*(1-(r^n))/(1-r): P
RINT : PRINT "Sum to ";n;" terms
 is ";Sn
```

Which is, you must admit, a lengthy program so far as we're concerned in this book, but it is in fact two programs in one. Lines 10 and 20 are concerned with giving you either the first half of the program or the second half. If you choose an AP you get line 100 running round its FOR . . . NEXT loop to give you the first ten terms of the AP you have specified. Then line 120 will work out the value of any term you like (try asking for the hundredth term!), and line 130 will tell you the sum of all the terms up to the one you've chosen. How these two facilities work is explained shortly.

Had you gone for a GP, you would get the terms written out by the FOR . . . NEXT loop in line 220 and the other facilities provided by lines 240 and 250 respectively.

There are, of course, formulas that control these things, and which appear in their Spectrum BASIC form in the program.



For an AP, the nth term is given by this equation:

$$\text{nth term} = a+(n-1)*d$$

(If you've never come across this nth business before, it's just a way of saying first, second, third, fourth, etc. in general terms, so that, if n equals 6, then nth is sixth.)

Also, for an AP, we have that the sum to n terms (usually denoted by $S_n$) is given by:

$$S_n = \frac{n}{2}(2a+(n-1)d)$$

By the same token, for a GP the nth term is given by:

$$\text{nth term} = a*r^{(n-1)}$$

and the sum to n terms by:

$$S_n = \frac{a*(1-r^n)}{(1-r)}$$

So no big secret. If you want to find out how these equations were derived in the first place, it's not a mind-bendingly difficult process to follow: next time you're down at the public library see if you can find it in a book.

I'd like to take a look at an AP where the first term is one and the common difference is also one.

Why?

Because it's interesting. If you try it on Program 71 you'll get

1, 2, 3, 4, 5, . . .

They are the positive integers. And you can rewrite the sum to n terms for this particular case by putting $a=1$ and $d=1$ in our general AP sum to n terms formula:

$$S_n = \frac{n}{2}(2a+(n-1)d)$$

becomes

$$S_n = \frac{n}{2}(2*1+(n-1)*1)$$

which simplifies to,

$$S_n = \frac{n}{2}(n+1)$$

which could be useful in a program sometime. It's a bit like the factorial except that the numbers are added together instead of multiplied together.

What we've been calling $S_n$ or in words 'the sum to n terms' is one of those things that the mathematicians have got hold of and sneaked a bit of Greek into. The mathematical notation is that, instead of using the letter S to stand for sum, you use the Greek letter S to stand for sum. And to save you flipping through the appendix in search of the Greek letter S, let me tell you it's called 'sigma', and the capital sigma looks like this:

$$\Sigma$$

And if you think it looks like being Chinese typewriter time again, then you're dead right. Luckily, the appendix tells you the user-defined graphic data for our letter sigma (upper case) is:

126, 34, 16, 8, 16, 23, 126, Ø

And if you deck the sigma about with a garland of numbers you get the mathematicians' form of sum to n terms. The number above it shows the maximum number you are summing to, and the number below the sigma shows where you're starting from. For example, if we were to write 'the sum of integers from 1 to n', we'd have,

$$\sum_1^n r^2 = \frac{n}{2}(n+1)$$

which is a result we established above, remember?

If we wanted to write the 'sum of r squareds as r takes the values from 1 to n', we would write:

$$\sum_1^n r = \frac{n(n+1)(2n+1)}{6}$$

which is a result that you could derive with the help of a maths book. All I want you to see is the way the notation is used, though, and perhaps remember the useful result which you might have occasion to put in a program if you get into more advanced programming.



Whilst we're at it, and to add a drop more practice, you would write

$$\sum_1^n r^3 = \left[\frac{n(n+1)}{2}\right]^2$$

which is the sum to n terms of the integers cubed:

$1^3, 2^3, 3^3, 4^3, 5^3, \ldots$

which is

1, 8, 27, 64, 125, . . .

And to test it out, let $n=5$ in our formula so that

$$\sum_1^5 r^3 = \left[\frac{5(6)}{2}\right]^2 = 15^2 = 225$$

which checks out OK, and of course it works for any value of n.

This notation of 'the sum from (bottom number) to (top number)' will reappear in a slightly different guise when we get on to something the mathematicians call integration. I know it's a bit difficult to see at the moment, but it's worth remembering. It happens to be that way right through maths. It's generally a case of learning things which don't seem to have any kind of application whatsoever, but you find eventually they're of huge importance.

## 4 Sum to infinity

The progressions we have looked at are sequences of numbers, and they can be as long as we choose to make them. We could get the Spectrum to keep printing out terms until it burst a chip or the terms got too big to handle, because there are an infinite number of possible terms. We saw that the positive integers were nothing more than an AP with $a=1$ and $d=1$, and we already know from our work with the Real Line that there are an infinite number of positive integers. And whatever AP we choose we could never find the sum of all of them, because the sum gets bigger by a bigger amount as we go on. But this is not true of a GP.

Consider the following:

1, 1/2, 1/4, 1/8, 1/16, . . .

You can recognise that as a GP with $a=1$ and $r=½$. You can also see that the terms are getting smaller as we go on, so that as the number of the term tends to infinity, the term itself tends to zero. (Do you remember this 'tends to' terminology?)

Mathematicians write a little arrow to mean 'tends to' and they show infinity by this symbol:

$\infty$

which looks to me like a figure 8 taking a rest.

We already know that our sum to n terms of a GP is given by,

$$S_n = \frac{a(1-r^n)}{(1-r)}$$

so for our particular series, with $a=1$ and $r=½$,

$$S_n = \frac{1(1-½^n)}{(1-½)} = 2(1-1/2^n)$$

and, as $n \to \infty$, $(1/2^n) \to 0$. (As n tends to infinity, $(1/2^n)$ tends to zero.)

We can write the sum to infinity as $S_\infty$, and know that it means the sum of all the terms in the series.

And by substituting infinity for n and zero for $(1/2^n)$, we have

$$S_\infty = 2(1-0) = 2$$

Which is quite remarkable really, because it means that without being able to use infinity in maths, we've been able to see what the sum of an infinite GP is! And you can check it out on the Spectrum, adding your terms on each time until you reach the limit of its calculating power. Use Program 71 to do this. You will see that 2 is indeed the limit that the sum of the series approaches:

$S_{10} = 1.9980469$

and

$S_{25} = 1.9999999$

and



$S_{26} = 2$

and any number bigger than 26 will yield 2 also.

There's an old riddle based on this, about a frog in a well. The frog is able to jump one metre up on its first jump and, presumably because it gets tired, can manage to jump only half as far on its second jump, half as far again on its third jump and so on. The well is two metres deep. How long will it take the frog to jump out?

Well, since it has to jump an infinite number of times to reach two, we find that the poor old frog is stuck for all eternity!

Does this mean that we can always find the sum to infinity of a GP?

No, it doesn't. There are clearly some GPs which get bigger and bigger, depending on the value of r. If it is bigger than 1, then you're out of luck, but if it's less than 1, you're OK.

Now note that (if you haven't already found it out) Program 71 gets itself in a twist if you try to enter an r with a negative value. This is a real shame, because you can easily think of a GP with a negative r. It goes into a spin because you are trying to raise a negative number to a power (which the Spectrum doesn't want to do).

In fact the condition that must be observed if our sum to infinity is going to be a finite number (and not infinity) is that r must lie between 1 and minus 1. In other words the range of r is:

$-1<r<1$

So far as Program 71 is concerned we must have

$0<r<1$

If you think your programming is up to it, why not figure out a program which does deal with negative r (I'm sure it's possible.)

## 5 Convergence

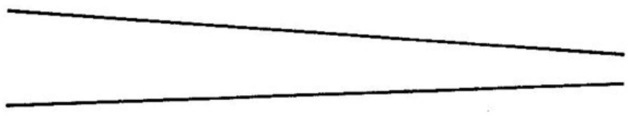
We can think of series that are neither arithmetic nor geometric progressions without too much trouble. For example, we could have a series which is got from the general term 1/n, by letting n equal successive positive integers:

1, 2, 3, 4, 5, . . .

so that our series becomes:

1, 1/2, 1/3, 1/4, 1/5, . . .

And if this was a GP (which it is not) you would think to yourself that since each term is getting smaller there would be no trouble finding a sum to infinity:



This is what we mean by convergence. If we look at a real GP, we find that it converges:

1, 1/2, 1/4, 1/8, 1/16, . . .

And clearly the GP

1, 2, 4, 8, 16, . . .

must be divergent.

But what about our example above generated by 1/n? There can be no doubt that each term gets smaller, but that does not mean it is going to be convergent.

If I were to say, 'All dogs have four legs', I could not go on to say, 'This animal has four legs, and therefore it must be a dog!' It would only be true to say something like, 'This animal is a dog, and therefore it must have four legs.'

Similarly, simply because a series has terms that tend to zero as the number of the term tends to infinity does not mean that the series necessarily converges. (It certainly won't converge if the numbers don't get smaller, but there are some series which, although the terms get smaller, they don't get smaller fast enough.) Our series generated by 1/n is an example, and this series is actually divergent.

Now that we have disposed of that common misapprehension, let's see if we can't find a rule that will tell us if a series is convergent or not.

A test which shows a series that will not converge is called the comparison test. You just get a series that you know is convergent and compare it with the one you want to test. If the



terms of your test series are each less than the corresponding terms of your known series, then your test series will be convergent too.

And to put it the other way round, the test series will be divergent if the corresponding terms are greater than the terms of the known series.

But which known series can we use for comparison? Well, our mathematician friends often use this one (which is known to converge):

$$\frac{1}{1^x}+\frac{1}{2^x}+\frac{1}{3^x}+\frac{1}{4^x}+\ \ldots$$

This series converges if x is greater than 1. It diverges if x is less than or equal to 1.

So that's the comparison test. There is also d'Alembert's test, named after Jean le Rond d'Alembert (1717?–1783), the French mathematician. (Some people prefer to call it the generalised ratio test.)

It is usual to write the terms of a series in general as the letter u with a subscript number, so that $u_1$ is the first term, $u_2$ is the second term, $u_3$ is the third term, etc. And these tests only apply to positive terms. It is perfectly possible to have a series with negative terms, or even alternate terms which are positive and negative, but we're confining ourselves to series with only positive terms.

D'Alembert's ratio test is useful because we don't have to bring in any other series apart from the one we have under test. There are no comparisons to be made, except for comparing the terms in the series with their neighbouring terms.

So that, if our series is

$u_1, u_2, u_3, u_4, u_5, \ldots$

we can call our general term $u_n$. And the next one will be $u_{n+1}$. D'Alembert asks us to make the ratio

$$\frac{u_{n+1}}{u_n}$$

and then to find the limiting value as n tends to infinity.

If the limiting value as n tends to infinity is less than 1, then the series converges.

If the limiting value as n tends to infinity is greater than 1, then the series diverges.

If the limiting value as n tends to infinity is exactly 1, then the series might converge and it might diverge.

What do I mean by 'the limiting value'? A small example will make everything clear. Suppose you have this series, and you want to know if it converges:

$$\frac{1}{2}+\frac{2}{3}+\frac{2^2}{4}+\frac{2^3}{5}+\frac{2^4}{6}+\ \ldots$$

I've been at pains to point out that, if $u_1$ is the first term, and $u_2$ is the second term, then we can call the nth term $u_n$. The nth term is a general term and can be represented by a formula so that when you substitute a value for n, you get any term you like. If you think about the series we have above, you can see that the nth term is given by the equation

$$u_n=\frac{2^{n-1}}{1+n}$$

(Try putting n equal to 5, say, to give the fifth term.) And it stands to reason that the term after the nth term is the n+1th term, and to get that you just substitute n+1 in the equation for the nth term, like this:

$$u_{n+1}=\frac{2^n}{2+n}$$

And we can therefore get the ratio d'Alembert is remembered for:

$$\frac{u_{n+1}}{u_n}=\frac{2^n}{2+n}\div\frac{2^{n-1}}{1+n}$$

which we can write:

$$\frac{u_{n+1}}{u_n}=\frac{2^n}{2+n}\star\frac{1+n}{2^{n-1}}$$

$$=2\frac{(1+n)}{(2+n)}$$

Now we can divide top and bottom by n to get



$$\frac{u_{n+1}}{u_n}=\frac{2(\frac{1}{n}+1)}{(\frac{2}{n}+1)}$$

Now comes the idea of the limit, because we know that as n tends to infinity 1/n tends to zero, and 2/n also tends to zero. The shorthand version of the phrase 'the limit of d'Alembert's ratio as n tends to infinity' is written:

$$\lim_{n\to\infty}\frac{u_{n+1}}{u_n}=\frac{2(\emptyset+1)}{(\emptyset+1)}=2$$

So this limit has been shown to be greater than one. Therefore the series diverges.

I know you've had to bear with me for the last few pages, and you must have asked yourself where it's all leading, but it is important to understand why we're doing what we're doing. Using a simple FOR . . . NEXT loop we can get the computer to simulate the above process. In Program 72, we call the numerator (top part of the fraction) of d'Alembert's ratio uN, and we call the denominator (bottom part of the fraction) of d'Alembert's ratio uD. Then we get the computer to print out successive values of the ratio uN/uD as we make n bigger. That way it indicates the limit to which it is tending. It's important to remember that the limit we're looking at is d'Alembert's ratio, not the limit of the series itself.

### *Program 72 D'ALEMBERT'S RATIO*

```
 5 REM PROGRAM 72
   D'ALEMBERT'S RATIO
10 FOR n=0 TO 10 STEP .5
20 LET uD=(2^(n-1))/(1+n)
30 LET uN=(2^n)/(2+n)
40 PRINT n,uN/uD
50 NEXT n
```

Program 72 uses the example we've discussed, and by plugging in your general term $u_n$ as uD in line 2∅, and your next term $u_{n+1}$ as uN in line 3∅, it is possible to get your d'Alembert's ratio for any series you meet. Try running Program 72 with line 10 taking the limit closer by substituting:

```
10 FOR n=10 TO 100 STEP 10
```

and your results will show you clearly that the limit is indeed 2. Extend the range again and you'll find that eventually the computer can't cope with 2↑n, and you get a 'Number too big' error.

## 6 The binomial expansion

There's a famous (and important) expansion we've yet to mention. It pops up again later and forms the basis of a dozen and one different higher applications. As usual, I won't bore you with how we get there: I'll just quote it and show you how you can use it. If you're that bothered about it, you can find a public library and look up the derivation in a text book. The binomial expansion is its name, and it is just the expansion of a binomial expression – that's one with just two numbers – which has been raised to a power.

The simplest version is $(1+x)$ raised to the power n, and it turns out that $(1+x)^n$ has an expansion that goes like this:

$$1+nx+\frac{n(n-1)}{2!}x^2+\frac{n(n-1)(n-2)}{3!}x^3+\ldots$$

It's worth thinking about that a second, just to get it straight. The first term is 1, the second term is n times x, the third term is n times 1-less-than n times x to the power 2 over 2 factorial . . . So can we see the pattern? Each successive term is composed of three elements: the string of n's getting smaller each time all multiplied together, the x raised to a power that gets bigger by 1 each time, and the denominator which is that power made factorial.

Now come a few observations. When n is positive, there are $n+1$ terms in all. When n is fractional or negative, there are an infinite number of terms. The expansion only works when x is greater than $-1$ and less than plus 1 (i.e. the range of x is $-1<x<1$).

That in itself would allow us to evaluate something like $(1+½)^2$, for example (try it and see!). But it has much more



weighty implications. If you don't believe me, just try a binomial expansion with x set equal to 1/n, so that you have $(1+\frac{1}{n})^n$: if you play about with that long enough, you'll find out how we get the expansion of $e^x$ that was quoted earlier in this chapter!

A more general binomial expansion is $(a+x)^n$ which you can see is equal to $(a(1+\frac{x}{a}))^n$. And the expansion of that is:

$$a^n 1+n\frac{x}{a}+\frac{n(n-1)}{2!}\frac{(x)^2}{(a)}+\frac{n(n-1)(n-2)}{3!}\frac{(x)^3}{(a)}+\ldots$$

As before, when n is positive, there are $n+1$ terms. When n is fractional or negative, there are an infinite number of terms. And in this case, the expansion is only valid if $(x/a)$ is in the range from $-1$ to 1.

Let's look now at a particular expansion of $(1+x)^n$, setting n equal to zero and increasing it by one each time:

$(1+x)^0=1$
$(1+x)^1=1+x$
$(1+x)^2=1+2x+x^2$
$(1+x)^3=1+3x+3x^2+x^3$
$(1+x)^4=1+4x+6x^2+4x^3+x^4$

and so on.

Do you see what's happening? I'm just multiplying out the brackets on the LHS to get the sequence of powers of x on the RHS. Alternatively, you can think of it in terms of the binomial expansion giving the RHS in each case. Now look carefully at the coefficients of the terms $x^0, x^1, x^2, \ldots$ and you can write them out in the form of a pyramid:

```
        1
       1 1
      1 2 1
     1 3 3 1
    1 4 6 4 1
```

and so on.

Can you see how it's building up? And can you tell what the next layer of the pyramid would be? Clearly it would be the expansion of $(1+x)^5$, and you could work it out from first principles that way. But it's easier to look at the pyramid directly and generate the next row by adding together the two elements directly above to left and right, so that we would get this:

```
 1  4  6  4  1
  V  V  V  V
1  5  10  10  5  1
```

Once you get this pattern fixed in your mind, it's possible to generate any expansion of $(1+x)^n$. This fact was first remarked on by the Frenchman Blaise Pascal (1623–1662). In fact the pyramid structure is called Pascal's triangle.

Program 73 will generate these values for you, labelling them in powers of x down to $(1+x)^{10}$, and there's no reason why you shouldn't modify the program so that you get it to calculate the coefficients of any expansion you want (screen size limitations notwithstanding!).

*Program 73 PASCAL'S TRIANGLE*

```
   2 REM PROGRAM 73
     PASCAL'S TRIANGLE
   5 PRINT "x0 x1 x2 x3 x4 x5 x6
 x7 x8 x9 "
  10 FOR n=0 TO 10
  20 FOR r=0 TO n
  30 GO SUB 100: GO SUB 200: GO
SUB 300
  40 PRINT AT n+1,r*3;Fn/(Fnr*Fr
)
  50 NEXT r
  60 NEXT n
  70 FOR x=0 TO 255 STEP 24: PLO
T x,175: DRAW 0,-100: NEXT x: ST
OP
 100 LET Fn=1: FOR x=1 TO n: LET
 Fn=Fn*x: NEXT x: RETURN
 200 LET Fnr=1: FOR x=1 TO n-r:
LET Fnr=Fnr*x: NEXT x: RETURN
 300 LET Fr=1: FOR x=1 TO r: LET
 Fr=Fr*x: NEXT x: RETURN
```



It's interesting to note that the structure of Program 73 is that of a main program (lines 10 to 80) with subroutines at 100, 200 and 300 allowing the necessary calculation. Those subroutines each calculate a factorial, and you should compare their structure with Program 69. It rests on the fact that the general term for the coefficients in our expansion is given by the expression

$$n!\,/(n-r)!\;r!$$

## 7 Fibonacci Series

Because you've been so patient and stuck with me throughout all this stodgy stuff about expansions, I thought I'd get back to insanity and give us all a rest.

Leonardo da Pisa, who was around in Italy at the end of the 12th century (1170–1230), picked up his mathematics from the Arabs and introduced the idea of our present day numbers to Europe (previously Roman numerals and an assortment of Greek letters were used). But he's best remembered for a little device that links together such diverse things as the arrangement of seeds in the centre of a sunflower, the curve of a nautilus shell, Greek architecture and art, Mahler's symphonies, foolscap paper and Alice in Wonderland.

Leonardo da Pisa was known by his pseudonym 'Fibonacci', and his sequence of numbers goes like this:

1, 1, 2, 3, 5, 8, 13, 21, . . .

If you think you've cracked it try to figure out the next term. If you got 34, then you have correctly seen that each term is the sum of the two previous terms, so that 5 is 3 plus 2, and so on.

Program 74 does it for you:

*Program 74 FIBONACCI SERIES*

```
   5 REM PROGRAM 74
     FIBONACCI SERIES
  10 PRINT "Term No.";"Term": PR
INT 1,1: PRINT 2,1
  20 LET a=1: LET b=1
  30 FOR n=3 TO 21
  40 LET f=a+b
```

```
50 PRINT n,f
60 LET a=b: LET b=f
70 NEXT n
```

Alter line 30 to generate as many terms as you like, so that

30 FOR n=3 TO 100

will give you 100 terms.

If we call the terms of the Fibonacci series $u_1, u_2, u_3, \ldots u_n$, then it happens that

$$u_{n+1}^2 = u_n{*}u_{n+2} + (-1)^n$$

which you can check for $n=6$:

$$u_n = 8\ ,\ u_{n+1} = 13\ ,\ u_{n+2} = 21 \text{ and } (-1)^6 = 1$$

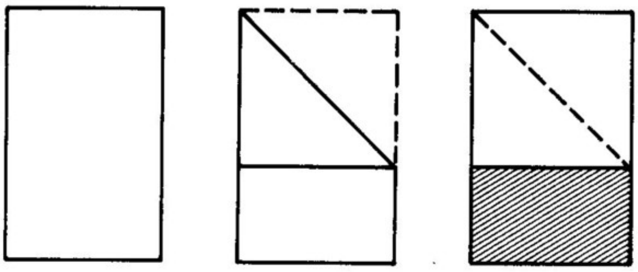
So

$$13^2 = 8{*}21 + 1$$
$$169 = 168 + 1$$

which is obviously true!

This was first discovered by a mathematician called Charles Dodgson (1832–1898) who is rather better known for his books (*Alice in Wonderland*, etc.) published under the name Lewis Carroll.

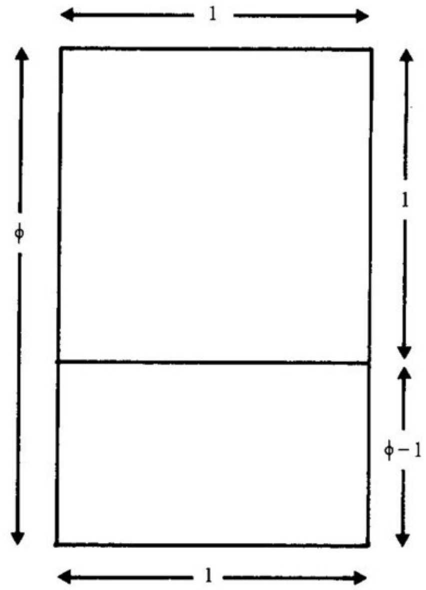
Before the days of metric sizes, paper in England used to be measured in Imperial sizes, and one size was called foolscap. An interesting thing about foolscap is that if you bent down one corner of the sheet as if you were making a paper aeroplane:





and then bend it back, you would get the crease across the paper making a square, and there would be an oblong left at the bottom shown on the third diagram as shaded. The property of the foolscap size is that the shaded bit left over is the same shape as the original foolscap (only smaller!).

We can look at this mathematically. If we call the width of the foolscap sheet 1 unit of length, and the length of it $\phi$ (Greek letter phi) units of length, then we said that the ratios of the bit left over are the same as the whole thing. In other words, 1 is to $\phi$ as $(\phi-1)$ is to 1.




In mathematical terms, we can represent these ratios as fractions,

$$\frac{1}{\phi} = \frac{(\phi-1)}{1}$$

And we can rearrange that and solve for $\phi$, using the quadratic technique:

$$1=\phi(\phi-1)$$
$$0=\phi^2-\phi-1$$

so that

$$\phi=\frac{1\pm\sqrt{5}}{2}$$

Root 5 has the value 2.236067977, and so $\phi$ equals $-0.618033988$ or $\phi$ equals 1.618033989. Since $\phi$ is a length, it is meaningless if it's negative, and so we choose the positive value.

This ratio, 1 to 1.618033989, was known to the Greeks and was called the Golden Section. It was a proportion much used by them in art and architecture.

But it's not obvious what all this has to do with the Fibonacci series. Well, it does have a connection, and it's this: the ratios of consecutive terms of the Fibonacci series tend to $\phi$ as the number of the term tends to infinity! So

$$\lim_{n\to\infty} \frac{u_{n+1}}{u_n}=\phi \qquad \text{(Remember d'Alembert?)}$$

The third term is 2, the fourth 3, and our ratio equals $3/2=1.5$. The thirteenth term is 233, the fourteenth 377, and our ratio is now $377/233=1.618025751$, which is much closer. The thirty-first term over the thirtieth is $1346269/832040=1.618033989$, which is correct to 9 decimal places.

Mahler's symphonies and the spirals of the nautilus shell and sunflower seeds are up to you to discover about yourself: I'll just say that, if you look up the word phyllotaxis in a good encyclopaedia, you'll get to it.

Now, didn't I warn you that everything in mathematics is connected to everything else?

# 11

# VECTORS

## 1 Vectors

This is good fun and easy to get at on a computer. (I told you this book was going to break with tradition and get easier by degrees.)

Almost anything you can measure in numbers is either a scalar or a vector. A scalar quantity is one that has only size, whereas a vector quantity has both size and direction.

An example of a vector quantity is velocity. You can have 30 miles per hour due north, or 130 km/h north-west, or 1 metre per second along the x-axis, etc. Size and direction.

An example of a scalar is mass: if you have a brick of one kilogram, the mass of that brick is not associated with any direction. One kilogram is a magnitude, a size.

All this means that you can represent scalars by numbers, but for vectors you have to have a way of conveying both the magnitude and the direction.

To represent a vector quantity, we draw a line. The length of the line represents the magnitude part of the vector and the direction of the line represents the direction of the vector. Two vectors are only equal if both their magnitudes and directions are equal. Below is a collection of different vectors:

You'll notice that they're not just lines, but arrows, and this is necessary if you think about it, because a line can point in two opposite directions at once, whereas the arrow is definitely pointing in only one direction. Program 75 draws some random vectors for you, putting a blob on the end of each vector to represent the arrow head.

*Program 75 RANDOM VECTORS*

```
   5 REM PROGRAM 75
     RANDOM VECTORS
  10 BORDER 6: PRINT "Various Ra
ndom vectors"
  20 FOR n=0 TO 20: LET A=INT (R
ND*2)
  30 LET x1=30+RND*195: LET y1=3
0+RND*115:
  40 LET x2=RND*30: LET y2=RND*3
0
  50 IF A=0 THEN  PLOT x1,y1: DR
AW x2,y2: CIRCLE x1+x2,y1+y2,1
  60 IF A=1 THEN  CIRCLE x1-x2,y
1-y2,1: PLOT x1,y1: DRAW -x2,-y2
  70 NEXT n
```

Fairly trivial I suppose, but it gets the idea across. Now suppose you've got a ball on the end of a string, and you hold it above your head and make the ball travel round in a circle. Then you can think of the string running from your fist to the ball as being a vector. Assuming you keep a tight hold on the string you have a constant magnitude (so the length of the string remains constant) but the direction moves continuously.

Program 76 shows 'snapshots' of the string as it makes its way around the full 360°. The snapshots are at intervals of 9°, as you can see from line 20 (everything is in radians, so if 2*PI is 360° then PI/20 is 360/40 = 9°). You can use line 10 to choose your string length between 0 and 87.

*Program 76 RADIUS VECTORS*

```
5 REM PROGRAM 76
  RADIUS VECTORS
10 LET MAG=50
20 FOR A=0 TO 2*PI STEP PI/20
30 LET x=MAG*COS A
40 LET y=MAG*SIN A
50 PLOT x+128,y+88
60 DRAW -x,-y
70 PAUSE 20: CLS
80 NEXT A
```



You have to remember that a line is not a vector until you specify its direction, so we can think of the set of radius vectors swept out in Program 76 as all being directed from the centre outwards, or if you like, from the circle inwards to the centre. If you delete line 70 and run Program 76 again, the vectors will accumulate on the screen.

You can imagine what would happen if, instead of string, you used elastic. Then you could get the magnitude of the radius vector to vary as well as its direction and, depending upon the way you did it, you could get a set of shapes other than circles. The sun is a fist swinging the ball of the earth around on a string made of gravity, and such is the property of gravity that the shape the earth moves in is an ellipse.

*Program 76a RADIUS VECTORS OF AN ELLIPSE*

```
5 REM PROGRAM 76a
  RADIUS VECTOR OF AN ELLIPSE
10 LET M1=30: LET M2=60
20 FOR A=0 TO 2*PI STEP PI/20
30 LET x=M1*COS A
40 LET y=M2*SIN A
50 PLOT x+128,y+88
60 DRAW -x,-y
70 PAUSE 20: CLS
80 NEXT A
```

Of course, returning to Program 76 for a moment, if you alter line 40 to LET y = 0, you will see a vector which starts at its full length, decreases to zero, continues to decrease beyond zero until it is of a similar magnitude to the one it started with except pointing the other way, increases to zero again and then back to its original size. This is a vector going through the range of its magnitude changes *with no direction changes.* It is an example of what is known to scientists as SHM (which stands for Simple Harmonic Motion), and serves to show that if we have a vector A, then the vector −A has the same length and direction, but points in the opposite direction:


So can you do arithmetic with vectors? You certainly can. Let's begin with addition. If you have a vector A, and want to add to it a vector B, you end up with a vector C. If you represent the vectors A and B as lines with the right lengths and angles, the sum vector C will be represented by a line that starts at the beginning of the first vector and ends at the end of the second vector. The triangle that is formed by this process is half of what's called the parallelogram of vectors. Program 77 will make everything very clear by adding two randomly selected vectors A and B ad nauseam. If you sit through five minutes of it, you'll never forget how vectors add.

*Program 77 ADDITION OF VECTORS*

```
  5 REM PROGRAM 77
    ADDITION OF VECTORS 2
 10 PRINT "VECTOR A"
 20 LET X1=RND*127: LET Y1=RND*
87
 30 LET X2=RND*127: LET Y2=RND*
87
 40 PLOT 1,1: DRAW X1,Y1
 41 PLOT 1,1: DRAW X2,Y2
 50 PAUSE 100
 60 PRINT AT 0,9;"PLUS VECTOR B
"
```



```
 70 PLOT X1,Y1: DRAW X2,Y2
 71 PLOT X2,Y2: DRAW X1,Y1
 80 PAUSE 100
 90 PRINT AT 1,0;"EQUALS VECTOR
C"
100 DRAW 1-(X1+X2),1-(Y1+Y2)
110 PAUSE 200: CLS : GO TO 10
```

Relentless, isn't it?

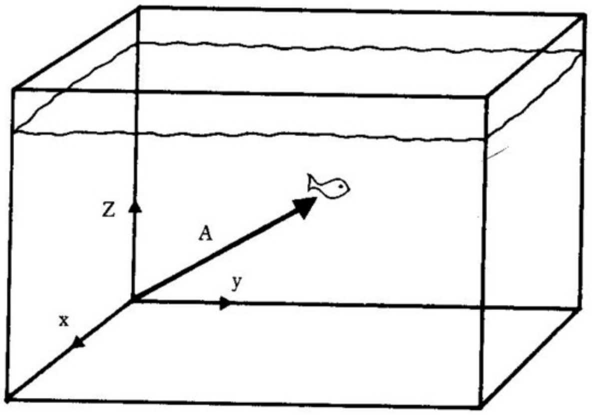
Other aspects of vector arithmetic are:

*Subtraction* We can think of A−B as being A+(−B), and we already know that −B represents B with the same magnitude, but exactly opposite direction.

*The product of a vector and a scalar* If you multiply a vector A by, say, 10, the effect is to make it 10 times as long and to leave the direction unchanged. Multiply by a negative scalar, and you get the direction turned opposite. (This stands to reason since we know about −A, and that's just A*(−1), after all.)

*The null vector* This is just the vector version of zero. If A=B, then the null vector is defined as A−B. It has zero magnitude and no definite direction.

One useful application of vector maths is three-dimensional mapping. If you think of a fishtank with water in it, and one fish, and you decide to call the back left-hand corner of the tank the origin, then you can construct a 3-D coordinate system like the one below:




If we make the width of the tank the x direction, the length of it the y direction and the height the z direction, as shown, then wherever 'Jaws' swims, we can imagine a vector (maybe a thin beam of light!), which we can call A, from the origin to the fish. As he swims about, A changes. It gets longer and shorter, so the magnitude alters. Also the direction of the vector alters.

We can break the position of the fish, and hence the vector, into three components along the x, y and z directions, and this process is known as resolving the vector into components. This program demonstrates the idea.

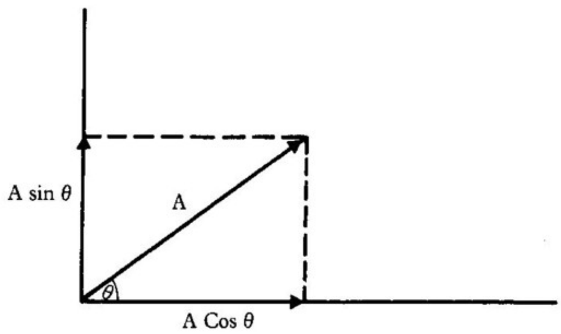
*Program 78 RESOLVING A VECTOR*

```
   5 REM PROGRAM 78
     RESOLVING A VECTOR
  10 INK 0: BORDER 6: PLOT 0,0:
DRAW 71,71: DRAW 100,0: PLOT 71,
71: DRAW 0,100
  20 PRINT AT 18,2;"x": PRINT AT
 12,20;"y": PRINT AT 3,7;"z": PR
INT AT 0,10;"3D COORDINATES": PA
USE 100
```



```
  30 INK 2: PLOT 71,71: DRAW 50,
30
  40 PRINT AT 10,11;"A": PRINT A
T 1,10;"VECTOR A": PAUSE 50
  50 INK 4: PLOT 71,71: DRAW -30
,-30: PRINT AT 15,4;"Ax": PRINT
AT 2,10;"X COMPONENT"
  60 PAUSE 50: DRAW 80,0: PRINT
AT 17,9;"Ay": PRINT AT 3,10;"Y C
OMPONENT"
  70 PAUSE 50: DRAW 0,60: PRINT
AT 13,16;"Az": PRINT AT 4,10;"Z
COMPONENT"
```

The important notion here is that, if you have a vector in three-dimensional space, you can represent it as three numbers standing for 'so much along', 'so much up', 'so much in'. It works in situations where you are considering only two dimensions too: in this case you have two directions in which to resolve the vector. If we set up a 2-D coordinate system and draw in a general vector A, we can see that the components of A in the x direction and y direction are $A*\cos\theta$ and $A*\sin\theta$ respectively.

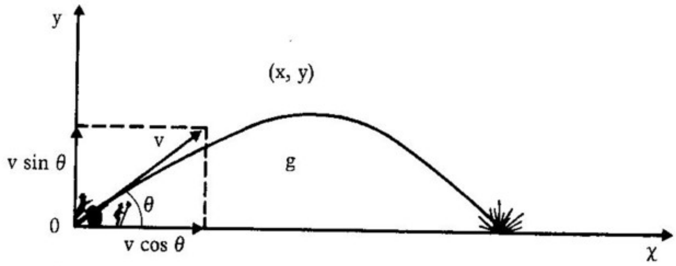



This idea crops up in considering the behaviour of projectiles, which means anything thrown and moving under gravity. If we examine the basis for projectiles, we can easily make a computer simulation of, for example, the way a cannonball would move.

Suppose we have an artillery piece firing a shell into the air so that the barrel of the gun is making an angle of θ degrees with the horizontal. Then the only factor we can vary is the amount of powder charge we use to fire the ball: the more gunpowder, the farther the ball will be projected. Because of the way gravity operates, the shape of the curve the ball traces out is a parabola (if we ignore the minor factor of air resistance).

We need to specify what's known as muzzle velocity (which means the speed at which the ball leaves the cannon) and realise that, once the ball is actually flying through the air and free of the explosive charge that sent it on its way, there is only one force, gravity, acting on the ball.

Since gravity only acts downwards, then when we resolve our vectors into directions parallel to and perpendicular to the horizontal, only the vertical component is affected by gravity. We therefore have a horizontal component that is a constant velocity, and a vertical component that is a constant acceleration in the downward direction.




Suppose your muzzle velocity is v m/s (that's metres per second), and suppose t represents the time elapsed after firing the gun. Then, since in the horizontal direction the velocity is constant, the distance of the ball is given by the equation,

$$x = v{*}\text{Cos}\theta{*}t$$

The vertical motion is more involved and is governed (or at least described) by an equation discovered by Newton. If we consider the downward acceleration to be called g, then the vertical position y is given by an equation similar to the one that yields x, but with an extra bit to represent the effect of gravity. Newton figured it out to be:



$$y = v{*}\text{Sin}\theta{*}t - gt^2$$

These equations form the basis of Program 79, and the equation for y gives us the time taken for the ball to arrive. When it does arrive $y=0$, and we can solve the quadratic in t to get

$$\text{time of flight} = \frac{2{*}v{*}\text{Sin}\theta}{g}$$

By substituting this back into the equation for x, we can get the horizontal distance covered, known as the range.

$$\text{Range} = \frac{v^2{*}\text{Sin } 2\theta}{g}$$

To prove that all this works I've included Program 79 which shows it happening graphically, and Program 80 gives you the numbers.

*Program 79 PROJECTILES*

```
   5 REM PROGRAM 79 PROJECTILES
  10 BORDER 5: INPUT "ENTER Muzz
le Velocity ";v
  20 INPUT "ENTER Angle of Eleva
tion ";a
  30 LET g=10: LET a=a*PI/180
  40 FOR t=0 TO v*SIN a/g STEP .
1
  50 LET x=v*t*COS a
  60 LET y=(v*t*SIN a)-g*t^2
  70 PLOT x,y
  80 NEXT t
```

Try this program, and, when it asks for the muzzle velocity, give it a number between 0 and 100. Try a variety of angles of elevation, and prove to yourself that a given charge gets the greatest range if the angle is 45°.

You could certainly incorporate this routine in a game, perhaps one that simulated an artillery duel between two armies.

But for those who prefer exact numbers:

*Program 80 PROJECTILE PARAMETERS*

```
    5 REM PROGRAM 80
      PROJECTILE PARAMETERS
   10 INPUT "Muzzle Velocity? (m/
s) ";v
   20 INPUT "Angle of Elevation?
(Degs) ";a
   30 LET g=9.81: LET a=a*PI/180
   40 LET TF=2*v*SIN a/g
   50 LET R=2*v*v*(SIN a)*(COS a)
/g
   60 LET H=v*v*(SIN a)*(SIN a)/(
2*g)
   70 PRINT "Muzzle Velocity",v;"
 m/s"
   80 PRINT "Elevation Angle",a*1
80/PI;" degs"
   90 PRINT "Time of Flight",TF;"
 secs"
  100 PRINT "Range",R;" metres"
  110 PRINT "Max Height",H;" metr
es"
```

You will notice (if I point it out) that Program 79 has the value 10 for g, whereas Program 80 uses the considerably more accurate value of 9.81 $m/s^2$. This is known as the acceleration due to gravity and may be regarded as a constant for anywhere on the surface of the Earth. (Go to Mars, and the acceleration due to Martian gravity is different: there it's 3.6 $m/s^2$.) It's a function of the mass of the planet you happen to be standing on and how far you are from the planet's centre, and, like all accelerations, is measured in the metric system in metres per second squared ($m/s^2$), also written ($ms^{-2}$).

I just thought it would be a good idea for you to see that all this theory has a big box of applications attached to it, and so if you want to follow this subject up more closely, get a book on the A level applied maths syllabus.

But in vector arithmetic our next conquest will be multiplication. There are two types of multiplication that can



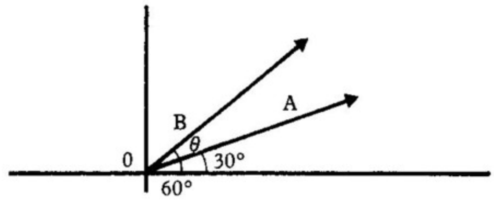
happen to vectors. If A and B are our two vectors, then we can have the scalar product or the vector product.

Let's deal with the scalar product first. This is often called the 'dot' product, because in maths books they show the product with a dot. The definition is,

$A.B = A*B*\cos\theta$

where $\theta$ is the angle between the vectors. It's called the scalar product because the result is a scalar quantity.

An example is this:




If A has a magnitude of 10, B has magnitude 5, A is 30° up from the horizontal, and B is 60° up. The angle between them is therefore 30°, so from our definition, the dot product is,

$$\begin{aligned} A.B &= 10*5*\cos 30° \\ &= 50*0.866 \\ &= 43.3 \end{aligned}$$

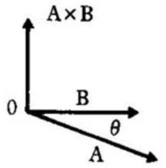
Notice that Cos 90° is zero, so that the dot product of two vectors which are at right angles is zero.

What of the vector product?

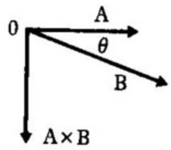
This is defined as a vector and is often called the cross product because it is written with a cross instead of a dot. We define

$A \times B = A*B*\sin\theta$

This cross product is not at all as simple as it might seem, because although the two vectors form a plane, the cross product is at right angles to that plane. Neither does $A \times B = B \times A$, which is strange enough. In fact, $A \times B = -(B \times A)$, as we can see from the diagram




If you imagine vectors A and B drawn on a sheet of paper, and to get from A to B you have to rotate anticlockwise, then the vector A×B points upwards out of the paper. If you'd had to rotate clockwise to get from A to B, as below,




then your product vector would point downwards into the paper.

The rule is one that crops up in elementary electricity, when you talk about the magnetic fields that occur when current is put through a solenoid. It's known as the corkscrew rule and helps you to remember the sense of the cross product. If you take your right hand and curl your fingers round and stick your thumb out, then if your fingers are rotating A to B the cross product points in the direction of your thumb. Compare it with the diagrams above and play about until it sinks in. (Don't do it on a bus or in any public place, or else people will think you're a loony.)

What about a program that take the sting out of it? All you have to remember is that a minus sign in the answer to a cross product means a downward pointing vector instead of an upward pointing one.

*Program 81 VECTOR PRODUCTS*

```
 5 REM PROGRAM 81
   VECTOR PRODUCTS
10 INPUT "VECTOR A? ";a
20 INPUT "VECTOR B? ";b
30 INPUT "ANGLE BETWEEN? ";Th
```



```
  40 PRINT "Magnitude of A = ";a
  50 PRINT "Magnitude of B = ";b
  60 PRINT "Angle Between A & B
= ";Th;" Degs"
  70 PRINT "DOT PRODUCT A.B = ";
A*B*COS (Th*PI/180)
  80 PRINT "CROSS PRODUCT AxB =
";A*B*SIN (Th*PI/180)
  90 PRINT "CROSS PRODUCT BxA =
";-A*B*SIN (Th*PI/180)
   5 REM PROGRAM 82
  10 DIM A(3,3)
  20 FOR r=1 TO 3
  30 FOR c=1 TO 3
  40 READ A(r,c)
  50 PRINT AT r*4,c*4;A(r,c)
  60 NEXT c
  70 NEXT r
  80 DATA 4,5,1,2,0,9,8,4,3
```

It's important to remember that the dot product gives a scalar, so that when in our program it says:

DOT PRODUCT A.B=70.710678

the number is a magnitude only. In the cross products, the numbers show the magnitudes of the product vectors. But the direction is at right angles to the plane of the vectors A and B, up if it's a positive number and down if it's negative.

## 2 Matrix algebra

This might seem strange at first, but it all ties in with what we've already learned. It's really just the mathematics of the computer's own array system.

You use the command DIM to dimension an array, and what you're doing is just the same as what you do in a LET statement. Suppose you've got a variable called x, and you want x to hold the value 10, then you would command,

LET x=10

Inside the computer a location called x is created and the value 10 is stored in it.

There are numbers which are not just one number at a time, but many at a time. You know this already from our previous work, but we're now going to consider it in detail.

A vector can be thought of as a list of numbers, so that, if I say a four vector, I mean a vector composed of four numbers. If I say a twelve vector, I mean one with twelve separate elements. Happily the Spectrum allows us to work with such vectors.

Suppose I had a vector to hold numbers of four different types of fruit: apples, oranges, pears, bananas. Then if I wrote four vector like this:

A=(10, 8, 14, 20)

you would know that it meant 10 apples, 8 oranges, 14 pears and 20 bananas.

The way to get the Spectrum to accept this four vector is to have the line

```
10 DIM A(4)
```

at the beginning of your program, then the Spectrum will know to set aside a region of its memory to hold the data for a four vector called A.

Then we can use a FOR . . . NEXT loop to read the data into A like this:

```
20 FOR n=1 TO 4
30 READ A(n)
40 NEXT n
50 DATA 10,8,14,20
```

If we had another four vector B, so that,

B=(12,16,9,5)

you would know what that meant, and you could do the sum

A+B

because you'd know that

A+B=(10+12,8+16,14+9,20+5)
=(22,24,23,25)



All the elements of A combine with the corresponding elements of B, and you get a sum which is a four vector.

Now let's suppose a small-time greengrocer wanted to computerise his business, and that one of his shop's stock inventories was four vector A, the other shop's inventory was vector B (I did say small time!). Both of these are quantity vectors, and we've chosen to write them out as rows.

Now suppose he has a price list like this:

apples 10p each
oranges 12p ,,
pears 14p ,,
bananas 8p ,,

then he could make a column vector (call it P for price), so that

$$P=\begin{pmatrix}10\\12\\14\\8\end{pmatrix}$$

Here comes the important bit: if our greengrocer wanted to work out the value of his stock in shop A he would have to multiply the corresponding row vector elements from A with the column vector elements from P, then add them all together, so that

$$A*P=(10,\ 8,\ 14,\ 20)*\begin{pmatrix}10\\12\\14\\8\end{pmatrix}$$
$$=(10*10)+(8*12)+(14*14)+(20*8)$$
$$=100+96+196+160$$
$$=552 \text{ pence}$$

Definitely small-time – and I think he's undercharging for that fruit too! Notice how the multiplying together of two four vectors has given a single number 552?

We could also model a scalar multiplication of a vector, by supposing that the greengrocer wanted to carry three times his present stock levels in store B. We would have

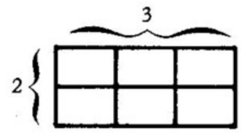
3*B=3*(12,16,9,5)
=(36,48,27,15)

That was a quick run down on one-dimensional number groups, but you know it's possible on your Spectrum to have arrays of more than just one dimension.

If you said

10 DIM A(2,3)

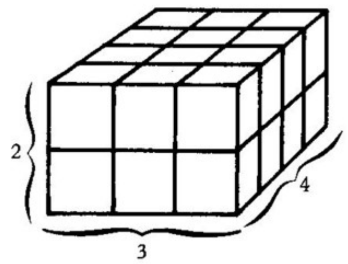
you would have reserved yourself a block of memory in two dimensions like this:




and if you'd said

10 DIM A(2,3,4)

you would have made yourself a three-dimensional array:




It would be possible to have a four-dimensional array (though I would not be able to draw you one!) or even a ten-dimensional one if you liked, but let's confine ourselves to the two-dimensional variety.

If a vector is a list of numbers, then a matrix is a bunch of vectors such that you could have, a three-by-three matrix, A:

$$A=\begin{pmatrix}4&5&1\\2&0&9\\8&4&3\end{pmatrix}$$



And the way to get that inside your computer is like this:

*Program 82 THE MATRIX*

```
 5 REM PROGRAM 82
10 DIM A(3,3)
20 FOR r=1 TO 3
30 FOR c=1 TO 3
40 READ A(r,c)
50 PRINT AT r*4,c*4;A(r,c)
60 NEXT c
70 NEXT r
80 DATA 4,5,1,2,0,9,8,4,3
```

The program prints you out a copy of your chosen matrix just so you can see it has done it properly. You might have already guessed that I have used r to represent row and c to represent column and that, because we're dealing with a two-dimensional array, we have to nest two FOR . . . NEXT loops to fill it up. And when it is up on the screen, convince yourself that it can be pictured as three column vectors:

| | | |
|---|---|---|
| 4 | 5 | 1 |
| 2 | 0 | 9 |
| 8 | 4 | 3 |

or as three row vectors:

4,5,1     2,0,9     8,4,3

or as nine independent elements.

If we wanted to add two three-by-three matrices (that's the plural of matrix), we would just add corresponding elements, so that,

$$\begin{pmatrix}2&5&1\\1&6&8\\9&3&5\end{pmatrix}+\begin{pmatrix}4&0&8\\5&5&1\\0&4&6\end{pmatrix}=\begin{pmatrix}6&5&9\\6&11&9\\9&3&5\end{pmatrix}$$

and subtraction is the same – you just subtract corresponding elements.

To achieve a scalar multiplication you would multiply each element of the matrix by a number, so that

$$3*\begin{pmatrix} 4 & 5 & 1 \\ 2 & 0 & 9 \\ 8 & 4 & 3 \end{pmatrix} = \begin{pmatrix} 12 & 15 & 3 \\ 6 & 0 & 27 \\ 24 & 12 & 9 \end{pmatrix}$$

which is not hard either. So we could write a small routine to stick on the end of Program 82 that would allow us to multiply our chosen matrix by any scalar:

```
  5 REM PROGRAM 82a
    SCALAR MULTIPLIER
 10 DIM A(3,3)
 20 FOR r=1 TO 3
 30 FOR c=1 TO 3
 40 READ A(r,c)
 50 PRINT AT r*4,c*4;A(r,c)
 60 NEXT c
 70 NEXT r
 80 DATA 4,5,1,2,0,9,8,4,3
 90 STOP
100 INPUT "Scalar Multiplier? "
;X
110 FOR r=1 TO 3: FOR c=1 TO 3:
 LET A(r,c)=A(r,c)*X: PRINT AT r
*4,c*4;A(r,c): NEXT c: NEXT r
```

Just run the whole program, and when it stops you enter CONTINUE and it will methodically go through the matrix and multiply each element out for you faultlessly by whatever scalar you like.

It shouldn't be too difficult for you to take these ideas and adapt Program 82 to allow you to handle matrices bigger than 3 by 3. If you don't insist on displaying the matrix you can handle very big arrays on the Spectrum. You would have to get it to print out the elements in sequence rather than all at once, though.

How about multiplying a couple of matrices together? It can be done. But it's a bit involved. Firstly we have to make sure that our two matrices are conformable. This just means that if you



want to multiply matrix A by matrix B you must make sure that A has the same number of columns as B has rows. If that's not the case, then you can't multiply them!

Then you take the first row and multiply each elements of it by each element of the first column of the other. (I told you it was involved!) Then you take the next row of the first matrix and multiply each element by each element of the next column of the other. You add these multiplications up to get the elements of the product. And you keep it up until you've got to the end.

It's a damnable rigmarole and just the sort of thing mathematicians congratulate themselves on being able to do. Let me give you an example:

$$\begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix}\begin{pmatrix} a & d \\ b & e \\ c & f \end{pmatrix} = \begin{pmatrix} (a*a+b*b+c*c) & (a*d+b*e+c*f) \\ (d*a+e*b+f*c) & (d*d+e*e+f*f) \end{pmatrix}$$

Let me give you a numerical example:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}\begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix} = \begin{pmatrix} (1*1+2*2+3*3) & (1*4+2*5+3*6) \\ (4*1+5*2+6*3) & (4*4+5*5+6*6) \end{pmatrix}$$

$$= \begin{pmatrix} (\ 1+\ 4+\ 9) & (\ 4+10+18) \\ (\ 4+10+18) & (16+25+36) \end{pmatrix}$$

$$= \begin{pmatrix} 14 & 32 \\ 32 & 77 \end{pmatrix}$$

The result is a 2×2 matrix.

Note that the product element formed by the first row and first column goes in the position 1,1 of the final matrix, and so on.

This is just the sort of gobbledegook that computers glory in, so here goes for a program that multiplies two 2×3 matrices like those above:

### *Program 83 PRODUCT OF 2×3 MATRICES*

```
  5 REM PROGRAM 83
    PRODUCT OF 2X3 MATRIX
 10 DIM A(2,3): FOR r=1 TO 2: F
OR c=1 TO 3: INPUT "?";A(r,c):
```

```
PRINT AT r*4,c*4;A(r,c): NEXT c:
NEXT r:
  20 INK 2: DIM B(3,2): FOR r=1
TO 3: FOR c=1 TO 2: INPUT "?";B(
r,c): PRINT AT r*4,16+c*4;B(r,c)
: NEXT c: NEXT r:
  30 LET C1=(A(1,1)*B(1,1))+(A(1
,2)*B(2,1))+(A(1,3)*B(3,1))
  40 LET C2=(A(1,1)*B(1,2))+(A(1
,2)*B(2,2))+(A(1,3)*B(3,2))
  50 LET C3=(A(2,1)*B(1,1))+(A(2
,2)*B(2,1))+(A(2,3)*B(3,1))
  60 LET C4=(A(2,1)*B(1,2))+(A(2
,2)*B(2,2))+(A(2,3)*B(3,2))
  70 PRINT AT 15,2;C1;AT 15,7;C2
  80 PRINT AT 17,2;C3;AT 17,7;C4
  90 INK 0
```

It stands to reason that if you have two square matrices, then they are conformable, and you can therefore multiply them together. The result is a matrix of the same order. By this I mean that if you multiply two $3\times3$ matrices you get a $3\times3$ matrix as the result.

Does there exist a matrix that you can multiply another matrix by so that you get a result identical with the one you started with?

Let me put it another way. If you consider ordinary numbers, any number multiplied by 1 is itself. Is there a matrix version of 1 such that any matrix multiplied by it is itself?

In fact there is. It's known as the identity matrix and, for a $3\times3$ matrix, looks like this:

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Notice how the elements are all zero except the ones located on the top-left-to-bottom-right diagonal which are all 1's.

For any $n\times n$ matrix, there exists an identity matrix such that $AI=IA=A$, where A is an $n\times n$ matrix and I is the identity matrix.



Whilst we're at it, we might as well look at inverse matrices. With ordinary numbers you know that

$$a^{-1}*a=1$$

So with matrices:

$$A^{-1}*A=I$$

Which is all very well, but how do we find it?

You think you've seen some complicated stuff? Well, let me tell you, the process of finding an inverse of a matrix is mind-blowingly confusing.

This is the point where a textbook would say, 'Considerations of this nature are beyond the scope of the present work.' So let's move on to the final section of this chapter.

## 3 Determinants

First we have to refresh ourselves concerning simultaneous equations. These are equations that you solve not one at a time but several at a time (hence the term simultaneous). You use simultaneous equations only when you can't avoid it. If you've an equation in a variable x, let's say, for example:

$$3x-9=0$$

we can solve it quite simply (the solution is $x=3$).

But if you have an equation in two unknowns, say x and y, then you can't solve it unless you have two equations:
Suppose we have,

$$3x-5y=6$$

and

$$4x+3y=37$$

The first thing to do is try to eliminate the y term from the pair, and we do this by making the coefficients of y the same. This can be achieved in this particular case by multiplying the top equation by 3 on both sides and the bottom equation by 5 on both sides, giving:

$$9x-15y=18$$

and

$20x+15y=185$

If we then add the two equations together, we have got rid of the y terms, and our result reduces to an equation in x alone (which we know we can solve):

$29x=203$

therefore

$x=203/29=7$

Armed with this snippet of information we can go back to one of our original equations (choose the simplest one unless you enjoy hard sums) and substitute the value of x we have just found:

$3*7-5y=6$

therefore

$5y=21-6=15$

so

$y=3$

So using two equations in two unknowns and a bit of natty technique, we've achieved a solution. You can check that it figures out OK by substituting y=3 and x=7 back into the two equations we started with.

Just beware that the two equations are independent. That means that they are distinct equations, and that you haven't made one out of the other by some method. To show you exactly what I mean, consider this:

$4x+6y+2=0$
$2x+3y+1=0$

These two equations are not independent, because the bottom one has been concocted by just halving all the coefficients. (Or maybe it was the top one that was made by doubling the coefficients of the bottom equation!) Whatever it was, they're not independent and so are no use to us.

In general, we can solve n independent simultaneous equations in n unknowns, so that if we had equations in x, y and z we'd need three independent equations, and so on.



We can do two at a time by computer without much sweat at all:

*Program 84 SIMULTANEOUS EQUATIONS*

```
  5 REM PROGRAM 84
    SIMULTANEOUS EQUATIONS
 10 PRINT "Solving Simultaneous
 Equations"
 20 PRINT  INK 2;"FIRST EQUATIO
N"
 30 PRINT  INK 2;"    y = ax + b
"
 40 INPUT "ENTER a"; a
 50 INPUT "ENTER b"; b
 60 PRINT  INK 4;AT 1,0;"SECOND
 EQUATION"
 70 PRINT  INK 4;"    y = cx + d
"
 80 INPUT "ENTER c"; c
 90 INPUT "ENTER d"; d
100 LET x=(d-b)/(a-c)
110 LET y=(a*(d-b)/(a-c))+b
120 PRINT AT 15,10;"x = ";x
130 PRINT AT 17,10;"y = ";y
```

It only requires you to get the equations into the correct form to enter the coefficients into Program 84. So the example above becomes

$3x-5y=6$ therefore $y=\frac{3x}{5}-\frac{6}{5}$

and

$4x+3y=37$ therefore $y=-\frac{4x}{3}+\frac{37}{3}$

you would consequently enter the following:

when asked for coefficient a, enter 3/5
when asked for coefficient b, enter −6/5
when asked for coefficient c, enter −4/3
and when asked for coefficient d, enter 37/3

You can enter these fractions, and the computer will work out the answers for you. For example, in the above case, you don't have to convert coefficient a into Ø.6 before entering it. Also, it doesn't matter which equation you choose to be first or second: the program handles either equation first or second.

Of course, it would, as I have said, be possible to extend this to as many equations as you want, but it would also be incredibly tedious to work through say five or six simultaneous equations.

Mathematicians are known for being lazy, and it wasn't long before they figured out a way to automate the process. It involves what is known as a determinant. It looks like a matrix but is much better fun. Whereas a matrix is usually written enclosed in ordinary curved brackets, you find determinants enclosed in straight lines, like this:

$$\begin{vmatrix} 3 & 5 \\ 4 & 2 \end{vmatrix}$$

But for the moment, it has been troubling me that Program 84 requires so much algebra before you can use it. Can't we design a version that will deal with the coefficient as they are presented without the need to convert them?

Let's see: the usual form of simultaneous equations is:

$$a_1x + b_1y + d_1 = 0$$
$$a_2x + b_2y + d_2 = 0$$

So

$$x = \frac{b_1d_2 - b_2d_1}{a_1b_2 - a_2b_1}$$

and

$$y = -\frac{a_1d_2 - a_2d_1}{a_1b_2 - a_2b_2}$$

Notice that the denominator in these two equations is now the same, being $a_1b_2 - a_2b_1$, and so long as this does not equal zero we're in business. Look out for our 'warhead' in equation form in lines 13Ø and 14Ø, and see how line 12Ø intercepts a zero denominator.



*Program 85 SIMULTANEOUS EQUATIONS AGAIN*

```
  5 REM PROGRAM 85
    SIMULTANEOUS EQUATIONS AGAIN
 10 PRINT "Solving Simultaneous
 Equations"
 20 PRINT  INK 2;"FIRST EQUATIO
N"
 30 PRINT  INK 2;"    a1x   +  b1
y  +  d1  =  0"
 40 INPUT "ENTER a1 "; a1
 50 INPUT "ENTER b1 "; b1
 60 INPUT "ENTER d1 "; d1
 70 PRINT  INK 4;AT 1,0;"SECOND
 EQUATION"
 80 PRINT  INK 4;"    a2x   +  b2
y  +  d2  =  0"
 90 INPUT "ENTER a2 "; a2
100 INPUT "ELTER b2 "; b2
110 INPUT "ENTER d2 "; d2
120 LET D=(a1*b2-a2*b1): IF D=0
 THEN  PRINT  INK 2; FLASH 1;AT
15,8;"DENOMINATOR ZERO": STOP
130 LET x=(b1*d2-b2*d1)/D
140 LET y=-(a1*d2-a2*d1)/D
150 PRINT AT 15,10;"x = ";x
160 PRINT AT 17,10;"y = ";y
```

Try it with our original pair of equations, so that,

| | | | |
|---|---|---|---|
| a1 = 3 | and | a2 = | 4 |
| b1 = −5 | | b2 = | 3 |
| d1 = −6 | | d2 = | −37 |

Now that you're familiar with the form of the expressions we have to deal with in this subject, I'll tell you how we define a determinant. As you can see, we have had to deal with 'something times something minus something times something', and a simple determinant is shorthand for it. So

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

is equivalent to writing,

a*d−c*b

So it's possible to give the solution to our simultaneous equations in determinant form like this:

$$x = \frac{\begin{vmatrix} b1 & d1 \\ b2 & d2 \end{vmatrix}}{\begin{vmatrix} a1 & b1 \\ a2 & b2 \end{vmatrix}}$$

and

$$y = -\frac{\begin{vmatrix} a1 & d1 \\ a2 & d2 \end{vmatrix}}{\begin{vmatrix} a1 & b1 \\ a2 & b2 \end{vmatrix}}$$

And because the word determinant begins with the letter D, and the Greek letter D is delta, and we use it to show a determinant, it looks like this:

$\Delta$

You will find details of how to get it into your Spectrum user-defined graphics in Appendix I (page 287).

So we can write

$$x = \frac{\Delta 1}{\Delta 0} \qquad \text{where } \Delta 1 = \begin{vmatrix} b1 & d1 \\ b2 & d2 \end{vmatrix}$$

$$\Delta 2 = \begin{vmatrix} a1 & d1 \\ a2 & d2 \end{vmatrix}$$

$$y = \frac{\Delta 2}{\Delta 0} \qquad \Delta 0 = \begin{vmatrix} a1 & b1 \\ a2 & b2 \end{vmatrix}$$

So to solve this:

$$6x+3y-33=0 \qquad a1=6 \quad b1=\ 3 \quad d1=-33$$

$$13x-4y-19=0 \qquad a2=13 \quad b2=-4 \quad d2=-19$$

we can use determinants. Program 85 or even regular school methods will show that $x=3$ and $y=5$, and using determinants we can show the same thing:



$$\frac{x}{\Delta 1} = \frac{-y}{\Delta 2} = \frac{1}{\Delta 0}$$

and since

$$\Delta 1 = \begin{vmatrix} b1 & d1 \\ b2 & d2 \end{vmatrix} = \begin{vmatrix} 3 & -33 \\ -4 & -19 \end{vmatrix} = -189$$

$$\Delta 2 = \begin{vmatrix} a1 & d1 \\ a2 & d2 \end{vmatrix} = \begin{vmatrix} 6 & -33 \\ 13 & -19 \end{vmatrix} = 315$$

$$\Delta 0 = \begin{vmatrix} a1 & b1 \\ a2 & b2 \end{vmatrix} = \begin{vmatrix} 6 & 3 \\ 13 & -4 \end{vmatrix} = -63$$

therefore

$$x = \frac{189}{63} \qquad \text{and} \quad y = \frac{315}{63}$$

which, if you care to check them, is dead right.

But isn't this using a sledgehammer to crack a nut? Isn't it making more work for ourselves instead of less? The answer must be Yes, when we consider only two simultaneous equations. But what if we want to deal with large numbers of equations at once? Then determinants come into their own. They're important items in maths, and you ought to be aware of them. Apart from being a powerful tool in the solution of linear equations, you can employ determinants in the manipulation of matrices, and take the determinant of any square matrix. It has the same elements and is the same size, and the value of it tells us things about the matrix. For example, if the determinant of a matrix turns out to be zero, then we say that the matrix is singular.

Anyhow, what about a program that evaluates determinants of order 2?

### *Program 86 DETERMINANT EVALUATION*

```
 5 REM PROGRAM 86
   DETERMINATION EVALUATION
10 DIM A(2,2)
```

```
  20 FOR r=1 TO 2
  30 FOR c=1 TO 2
  40 INPUT "ENTER next element "
;A(r,c)
  50 PRINT AT r*3,11+c*3;A(r,c)
  60 NEXT c
  70 NEXT r
  80 LET D=(A(1,1)*A(2,2))-(A(2,
1)*A(1,2))
  90 PRINT AT 15,4;"VALUE OF DET
ERMINANT IS"
 100 PRINT AT 17,14;D
```

You just plug in your values row by row, element by element, and it gives you back the numerical equivalent.

Of course, if we are dealing with three simultaneous equations, like these:

$$a_1x+b_1y+c_1z+d_1=0$$
$$a_2x+b_2y+c_2z+d_2=0$$
$$a_3x+b_3y+c_3z+d_3=0$$

we have to use determinants of order three. And, just as before, our solution boils down to this:

$$\frac{x}{\Delta_1}=\frac{-y}{\Delta_2}=\frac{z}{\Delta_3}=\frac{-1}{\Delta_0}$$

where,

$$\Delta_1=\begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_2 \\ b_3 & c_3 & d_3 \end{vmatrix} \Delta_2=\begin{vmatrix} a_1 & c_1 & d_1 \\ a_2 & c_2 & d_2 \\ a_3 & c_3 & d_3 \end{vmatrix} \Delta_3=\begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix} \Delta_0=\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}$$

It's easier to remember how to arrive at these results than it seems, because for example $\Delta_1$, which is associated with the variable x, has elements of all the coefficients except those of the x terms in our original equations. In other words, $\Delta_1$ consists of b's c's and d's, but not a's.

To solve a 3×3 matrix you have to use a special method which goes like this:



If our determinant is

$$\begin{vmatrix} A & B & C \\ D & E & F \\ G & H & I \end{vmatrix}$$

then the value of it is:

A times the determinant left after removing the terms in the row and column occupied by A
minus
B times the determinant left after removing the terms in the row and column occupied by B
plus
C times the determinant left after removing the terms in the row and column occupied by C

which is:

$$A*\begin{vmatrix} E & F \\ H & I \end{vmatrix} - B*\begin{vmatrix} D & F \\ G & I \end{vmatrix} + C*\begin{vmatrix} D & E \\ G & H \end{vmatrix}$$

This is solvable since we can evaluate these order two type determinants.

Now have a go at Program 87 which does it all for you. Notice that it incorporates a subroutine that does the work of Program 86.

### *Program 87 ORDER 3 DETERMINANT*

```
   5 REM PROGRAM 87
     ORDER 3 DETERMINANT
  10 DIM A(3,3): FOR r=1 TO 3: F
OR c=1 TO 3: INPUT "ENTER next e
lement ";A(r,c): PRINT AT r*3,10
+c*3;A(r,c): NEXT c: NEXT r
  20 LET D1=(A(2,2)*A(3,3))-(A(3
,2)*A(2,3))
  30 LET D2=(A(2,1)*A(3,3))-(A(3
,1)*A(2,3))
  40 LET D3=(A(2,1)*A(3,2))-(A(3
,1)*A(2,2))
```

```
   50 LET D=(A(1,1)*D1)-(A(1,2)*D
2)+(A(1,3)*D3)
   60 PRINT AT 15,4;"VALUE OF DET
ERMINANT IS"
   70 PRINT AT 17,14;D
```

If you want to test it out on some real order three determinants to test that your program works, try these:

$$\begin{vmatrix} 2 & 2 & 2 \\ 3 & 1 & 1 \\ 4 & 2 & 1 \end{vmatrix} = 4; \quad \begin{vmatrix} 4 & 1 & 5 \\ 8 & 2 & 6 \\ 12 & 3 & 7 \end{vmatrix} = 0; \quad \begin{vmatrix} 4 & 3 & 5 \\ 1 & 6 & 1 \\ 7 & -2 & 8 \end{vmatrix} = -23$$

And if you're really adventurous, you might like to devise a program that evaluates a determinant of order four! (You just have to extend with rigid logic the ideas we have already discussed.)

If not, don't worry about it. We're already way beyond most Advanced Level syllabuses and into the sort of things a person reading sciences or engineering at university might encounter in the maths lectures (and exams!).

# 12

# RATES OF CHANGE

## 1 Rates of change

Calculus is a major branch of mathematics, and I believe the name derives from the fact that before calculators were invented people did their sums by counting out small stones ('calculus' means pebble in Latin), a bit like the abacus, but no frame, I guess. It was invented by the inevitable Sir Isaac Newton, and though the Germans reckon their man, G. W. Leibnitz (1646–1716), invented it in 1673, it was in fact Newton who got there first. Not that it matters much.

Calculus is generally regarded as being of two kinds: integral calculus (which deals with a process called integration) and differential calculus (which deals with the process known as differentiation).

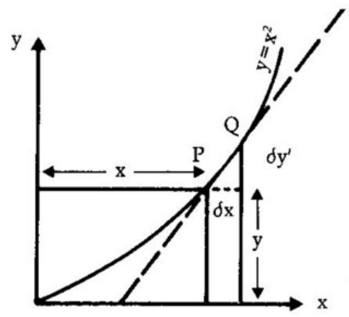
In this chapter we'll have a look at what differentiation is all about, and in the next chapter we'll investigate integration.

This section is entitled 'Rates of Change', and that's really what differentiation is about. And there's no better way of looking at it than by considering the familiar ideas of distance, velocity and acceleration.

For the moment I would ask you to consider most carefully this notion which is at the very root of calculus. Do you remember when we were dealing with series we used the idea of quantities that got smaller and smaller, tending to zero, so that some function of that quantity tended to a given value?

The idea is called finding the limit.

Now suppose that we have one of our simplest functions, with its attendant graph, $y=x^2$




I want you to think of two points on the curve, P and Q, and connect them together by a line. Suppose that the horizontal separation of P and Q is called δx (read as 'delta x': we use delta in mathematics to mean a little bit, so delta x means a little bit of x, though mathematicians will call it a small increment in x.

Suppose also that the vertical separation is called δy. Then it follows that the slope of the line from P to Q is δy/δx. If we want to find the slope of the curve at any one point (which is the same as the slope of the tangent to the curve at that point), we can consider P and Q sliding along the curve until they get so close together that they merge. Then the slope of the curve at the point occupied by both P and Q will be what we're after.

We can do this mathematically by sliding Q towards P so that δy and δx both get smaller, until in the limit P and Q coincide. Instead of δy/δx, we write this limiting slope as dy/dx (with English instead of Greek letters). So the slope at the point P is called dy/dx and we say this 'dee-why-by-dee-eks' as though it was one word. In fact, if you think about it, it is no longer a ratio like δy/δx, but the limit of a ratio, which is a single thing with the meaning of 'the slope of the curve at that point'.

In our particular example we can work out what that slope is. It stands to reason that if y is a function of x, then dy/dx will be a function of x too:

At P, $y=x^2$
At Q, $y+\delta y=(x+\delta x)^2=x^2+2x\delta x+(\delta x)^2$



If we take y off the LHS and y off the RHS (remembering that $y=x^2$), then we get

$$\delta y=2x\delta x+(\delta x)^2$$

and to get our slope ratio, we divide both sides by δx:

$$\delta y/\delta x=2x+\delta x$$

Now, if we start sliding P and Q so that they coincide, we have to take the limit as δx tends to zero, and as δx tends to zero, $2x+\delta x$ tends to 2x, we can therefore write:

$$dy/dx=2x$$

And there we have it, and of huge importance it is too. We can find the slope of $y=x^2$ at any point by just substituting into the equation

$$\text{slope}=dy/dy=2x$$

There is a way of extending this to other functions too. For example, if we have any polynomial in x, or any power of x, we can get the slope, dy/dx, by the following process.

Take each term of the polynomial separately. Multiply the term by the power of x, and reduce the power by one. You can see $y=x^2$ obeys this.

We call dy/dx the differential or the derivative of y, and we call the process of finding the derivative differentiation. The branch of maths dealing with differentials is called differential calculus, and equations containing terms that are derivatives are called differential equations.

Let's look at that formula for finding the differential of any function:

If

$$y=a*x^n$$

then

$$dy/dx=n*a*x^{(n-1)}$$

If we try it with $y=x^2$, $n=2$ and $a=1$. Therefore

$$dy/dx=2*x^1=2x$$

And if we try it with $y=5x^3$, $n=3$ and $a=5$. Therefore

$dy/dx = 3*5*x^{(3-1)} = 15x^2$

Let's have a program that does it for you:

*Program 88 DIFFERENTIATION OF POLYNOMIAL TERMS*

```
   5 REM PROGRAM 88
     POLYNOMIAL DIFFERENTITATION
  10 BORDER 6: PRINT " DIFFERENT
IATION OF POLYNOMIALS"
  20 PRINT : PRINT "             n
                  n-1"
  30 PRINT "If Y= a*x  then dy/d
x = n*a*x    "
  40 PRINT AT 7,3;"For Example:"
  50 PRINT : PRINT "             Y
="
  60 INPUT "Choose constant a: "
;a
  70 PRINT AT 9,14;a;" x"
  80 INPUT "Choose power n: ";n
  90 IF n<>1 THEN  PRINT AT 8,17
;n
 100 PRINT : PRINT "   So,"
 110 IF n<>2 THEN  PRINT AT 13,2
2;n-1
 120 PRINT AT 14,10;"dy/dx = ";
n*a;" x"
 130 IF n=1 THEN  PRINT AT 16,16
;"= ";n*a
 140 IF n<0 THEN  PRINT AT 17,0;
"   Or,    dy/dx = ";a*n;" / x":
 PRINT AT 16,25;ABS (n-1)
 150 PRINT AT 20,0;" (dy/dx show
s slope of curve.)"
```

You've got to get the right number of spaces in the PRINT statements so that it reads well on the screen. But once you've got that sorted out it'll do your polynomial terms for you.



Suppose you have

$y = 3x^3 + 8x^2 + 4x + 3$

Just take each term separately, do Program 88 on it and you get

$dy/dx = 9x^2 + 16x + 4 + 0$

Remember that $x^1$ is x, $x^0$ is 1 and zero times anything is zero, which is why the final term, +3, when multiplied by Ø, its power of x, is itself Ø.

When it comes down to physical meaning, remember that dy/dx is the slope, but it is also *the rate of change of y with respect to x.*

This is important when we consider sports cars and motor bikes, because the rate of change of distance is velocity, and the rate of change of velocity is acceleration. So if the letter s represents distance, then ds/dt represents velocity, which we can call v. And it follows that the differential of v is acceleration, so that $a = dv/dt$:

$v = ds/dt$
$a = dv/dt$

Note that we have equations of s in terms of t, and that we are differentiating with respect to time t. Note also that what we have achieved is to get acceleration from distance by differentiating twice. It follows that,

$$a = dv/dt = \frac{d}{dt}(ds/dt) = \frac{d^2s}{dt^2}$$

This symbol is read 'dee-too-ess-by-dee-tee-squared' and, like dy/dx or ds/dt, is usually regarded as a single symbol representing the second differential of a quantity. If you differentiate it again, you would get the rate of change of acceleration (which doesn't seem to have a name in our language), and we would write it:

$d^3s/dt^3$

That would be the third differential.

Similarly, looking at our original example,

$y = 3x^3 + 8x^2 + 4x + 3$

and

$$dy/dx = 9x^2 + 16x + 4$$

and

$$\frac{d^2y}{dx^2} = 18x + 16$$

Terrific, so long as our function is a polynomial in x. But what if it has other functions in it like Sin x, or $e^x$ or $\log_e x$?

Just as we worked out the $y = x^2$ case, it is possible to calculate from first principles the differentials of $y = \text{Sin } x$, $y = e^x$, or $y = \log_e x$.

If you're inquisitive and intelligent, you might like to look these up in a textbook, just to satisfy yourself that it can be done. But as usual, I'll save myself for the results, and if you run Program 89, it will give you a page full of derivatives:

*Program 89 STANDARD DERIVATIVES*

```
   5 REM PROGRAM 89
     STANDARD DERIVATIVES
  10 BORDER 6: GO SUB 1000: PRIN
T " SOME STANDARD DIFFERENTIALS"
  20 PRINT : PRINT "y","dy/dx"
  30 PRINT : PRINT "Sin x","Cos
x","Cos x","-Sin x","Tan x","Sec
 x","Cot x","-Cosec  x","Sec x"
,"Sin x/Cos  x","Cosec x","-Cos
x/Sin  x"
  40 PRINT : PRINT "e^x","e^x","
e^a*x","a*e^a*x","a^x","(a^x)*LN
 a","LN x","1/x"
  50 PRINT : PRINT "Arcsin x","1
/  SQR(1-x )","Arccos x","-1/  S
QR(1-x )","Arctan x","1/(1+x )"
 999 STOP
1000 FOR n=0 TO 7: READ r: POKE
USR "a"+n,r: NEXT n: DATA 112,16
,112,64,112,0,0,0
```



```
1010 FOR n=0 TO 7: READ r: POKE
USR "b"+n,r: NEXT n: DATA 0,16,1
6,124,16,16,124,0: RETURN
```

Lines 1000 and 1010 define a little '2' for showing 'squared' and '±' respectively, so that key a in graphics mode gives the '2' index and key b in graphics mode gives '±'. When typing in the program, before you run it for the first time, you will find that when you want to put in the little 2, you should obtain graphics mode (G cursor) by pressing the CAPS SHIFT and '9' keys together, then put an a where you want the '2', or a b if you want a '±' sign.

After running it once, the Spectrum remembers the symbols and will list your program correctly.

You can now look up that if

$$y = e^{2x}$$

then

$$dy/dx = 2e^{2x}$$

etc.

The next area of differentiation to conquer is differentiating a product. So if you have

$$y = x^2 * \text{Sin } x$$

you can find dy/dx. There is a rule, called the product rule, which says that if,

$$y = u * v$$

and you have a product of two functions of x multiplied together, then

$$dy/dx = u * \frac{dv}{dx} + v * \frac{du}{dx}$$

In words, the derivative of a product is equal to the first function times the differential of the second function plus the second function times the differential of the first function.

So in our example, if we let $u = x^2$ and $v = \text{Sin } x$,

$$dy/dx = x^2 * \text{Cos } x + 2x * \text{Sin } x$$

Obviously, it doesn't matter which function you choose to be the u and which you choose to be the v. Using Programs 88 and 89 you can find the elements you want and just plug them into the product rule. Simple.

Also, you can do the same with quotients (by that I mean a couple of functions where one is divided by the other), using the quotient rule:

If

$$y = u/v$$

then

$$dy/dx = \frac{v\frac{du}{dx} - u\frac{dv}{dx}}{v^2}$$

Look at it carefully, and see that this time the expression for the derivative is no longer symmetrical, so it does matter which function you choose for u and which for v in practice. This is true because if you choose the right way it makes simplifying the function much simpler, whereas choosing the wrong way can make the answer hard to simplify.

As an example of this in operation, consider how we get the derivative of y = Tan x.

We know that

$$y = \text{Tan } x = \frac{\text{Sin } x}{\text{Cos } x}$$

Let u = Sin x and v = Cos x, so that by using the quotient rule we can get

$$dy/dx = \frac{\text{Cos } x . \frac{d}{dx}(\text{Sin } x) - \text{Sin } x . \frac{d}{dx}(\text{Cos } x)}{\text{Cos}^2 x}$$

$$= \frac{\text{Cos } x(\text{Cos } x) - \text{Sin } x(-\text{Sin } x)}{\text{Cos}^2 x}$$

$$= \frac{\text{Cos}^2 x - \text{Sin}^2 x}{\text{Cos}^2 x}$$

We know from our trig identities that $\text{Cos}^2 x - \text{Sin}^2 x$ equals 1, so



$$dy/dx = 1/\text{Cos}^2 x = \text{Sec}^2 x$$

which we know is right.

Of course, we had to know the derivatives of Sin x and Cos x to work it through, so where do we find these from first principles? I'll give you a clue.

Remember this?

$$\text{Sin } x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \ldots$$

and this?

$$\text{Cos } x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \ldots$$

You can take this to as many terms as you like and do a Program 88 on it, giving you a fairly clear indication of what's going on.

## 2 Motion

I shouldn't like you to get the idea that mathematics was found under a rock one day by Sir Isaac Newton, but I have to admit that much of it comes from his unique intellect, and one of his great triumphs was the laws of motion.

Try this: Let s represent the distance travelled by a large rock dropped over a vertical cliff. And let a be the acceleration due to gravity. Then this equation relates the variables to time elapsed, t:

s=u*t+½*a*t²

Here, u (which represents the initial speed of the rock) is zero.

We can use this equation in a variety of different ways: for example, we can plug in our value for the acceleration due to gravity on the Earth, choose a number of seconds after dropping the rock, and it will give us the distance the rock has fallen.

We'll use a = 10 m/s², choose 10 seconds, and then do

```
PRINT 0*10+10*10↑2/2
```

The answer that comes back is 500 which is in metres since we specified the acceleration due to gravity in metres/second². If you want it in feet, you have to use the value for a of,

$a = 32$ ft/s$^2$

By differentiating the original equation we get

$$s = ut + \frac{at^2}{2}$$

becoming

$$\frac{ds}{dt} = v = u + at$$

so that after 10 seconds, not only has our rock fallen 500 metres, but also its velocity has increased to:

PRINT 0+10*10

which is 100 metres/second.

If we differentiate again,

$$\frac{d^2s}{dt^2} = \frac{dv}{dt} = a = a$$

which just lets us know that the acceleration is the acceleration, which is known in the trade as an identity (or truism) and, if nothing else, encourages us that we're doing it right!

This all holds true under any conditions so long as a constant acceleration is applied to the object, which is reasonably true of a rock dropped over a cliff. It's also true of the projectiles we talked about earlier.

We can make a small program to use this result:

*Program 90 FALLING OBJECTS*

```
  5 REM PROGRAM 90
    FALLING OBJECTS
 10 PRINT "Falling Objects"
 20 INPUT "ENTER Initial fallin
g velocity";u
 30 INPUT "ENTER Time elapsed";
t
 40 LET s=u*t+9.81*t*t/2
 50 PRINT AT 8,0;"Distance fall
en is ";s;" metres"
 60 LET v=u+9.81*t
```



```
 70 PRINT AT 12,0;"Velocity aft
er ";t;" seconds is ";s;" m/s"
```

And it's probably not beyond you to construct a program that can be used in other situations where the acceleration is constant, but not necessarily the acceleration due to gravity (like with a drag-racer, for example, or a rocketship).

With falling objects, this equation is strictly true for all velocities not close to the speed of light, if the body is falling in a vacuum. If it is not falling in a vacuum, but through air, then obviously there comes a point where the air-brake counteracts the pull of gravity, and you reach a 'terminal velocity'. This is true if you leap out of a plane. Before you open your parachute, you will accelerate until the rush of air against your body takes effect, and your terminal velocity will be about 120 miles per hour. The big surface area of your parachute brings down your terminal velocity to a reasonable figure, unless, of course, you go parachuting on the moon, which has no air. Although the acceleration due to gravity is only 1/6 that of the Earth, you would quickly reach a velocity that would mash you on impact.

Alter Program 90 for lunar conditions, by changing 9.81 to 1.6 and run it. You will see that after only 10 seconds you are travelling at 80 m/s which is about 180 miles per hour – and on the moon your parachute doesn't work!

## 3 Taylor's & Maclaurin's theorems

Now that we've got to grips with the idea of differentiating, it's possible to go back to series and see the basis for them.

Maclaurin's theorem is a particular case of Taylor's theorem, so it might be best to start by looking at that. But first I'll just mention the alternative notation which you will find used for derivatives, and which is very useful in our particular application.

If

$y = f(x)$ (read 'eff of eks')

then

$\frac{dy}{dx} = f'(x)$ (read 'eff dash of eks')

and

$$\frac{d^2y}{dx^2} = f''(x) \qquad \text{(read 'eff double dash of eks')}$$

and so on.

Maclaurin says that if we assume we can represent the function by a series of powers of f, so that

$$f(x) = A_0 + A_1x + A_2x^2 + A_3x^3 + A_4x^4 + \ldots$$

then we can find the values of the A coefficients by successive differentiations of the polynomial:

$$f'(x) = A_1 + 2A_2x + 3A_3x^2 + 4A_4x^3 + 5A_5x^4 + \ldots$$

and

$$f''(x) = 2A_2 + 2*3A_3x + 3*4A_4x^2 + \ldots$$

and

$$f'''(x) = 2*3A_3 + 2*3*4A_4x + \ldots$$

etc.
which must all hold true when $x = 0$. So when we substitute $x = 0$ into the successive derivatives, we get

$$A_0 = f(0)$$
$$A_1 = f'(0)$$
$$A_2 = \frac{1}{2!}f''(0)$$
$$A_3 = \frac{1}{3!}f'''(0)$$

etc.
And then we just substitute the A coefficients back into the expansion we assumed, to get

$$f(x) = f(0) + xf'(0) + \frac{x^2}{2!}f''(0) + \frac{x^3}{3!}f'''(0) + \ldots$$

This is known as Maclaurin's theorem, and we can use it to get any of the jolly good expansions we quoted in the chapter on series.

Let's have a go with $e^x$

Assume that $e^x = A_0 + A_1x + A_2x^2 + A_3x^3 + \ldots$



Happily for us if you differentiate $e^x$ with respect to x, then you get $e^x$, so that $f(x) = f'(x) = f''(x) =$ etc. for this happy case, and we have

$$e^x = A_1 + 2A_2x + 3A_3x^2 + 4A_4x^3 + \ldots$$

and

$$e^x = 2A_2 + 2*3A_3x + 3*4A_4x^2 + \ldots$$

and

$$e^x = 2*3A_3 + s*3*4A_4x + \ldots$$

etc.
Putting $x = 0$ in each of the above expansions, we get

$$A_0 = 1, \quad A_1 = 1, \quad A_x = 1/2!, \quad A_3 = 1/3! \text{ etc.}$$

so that

$$e^x = 1 + x + x^2/2! + x^3/3! + \ldots$$

And there we are!

Taylor's theorem is concerned with the identical process except that it starts with $f(a + x)$ instead of $f(x)$, so you can see that Maclaurin's theorem is a particular case of Taylor's theorem where $a = 0$.

The result for Taylor's theorem is:

$$f(a + x) = f(a) + xf'(a) + \frac{x^2}{2!}f''(a) + \frac{x^3}{3!}f'''(a) + \ldots$$

One application which might already have struck you is that expanding $(a + x)^n$ results in the binomial theorem.

## 4 Differentials and differential equations

Remember that the symbol dy/dx has a meaning (the limit of $\delta y/\delta x$ as $\delta x$ tends to zero) as a whole, but that does not imply that you can split it in two and have a meaning for dy and dx separately any more than you can cut a number 8 in half and have a meaning for the two separate hoops!

However, you will often find what amounts to $\frac{dy}{dx} = 2x$, written as

$dy = 2x . dx$

How can this be? Well, just that we define things called differentials so that the ratio of them equals dy/dx. And we write them dy and dx, making sure that we have

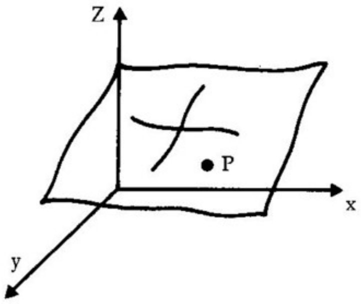
$$\frac{dy}{dx} = dy/dx$$

I only mention this because you'll come across it in the next chapter which concerns integration, and it might be of some use to you on that score. And whilst we're discussing notation, you might get occasion to open a maths book some time in the future and run across a funny-looking dy/dx symbol in which the d's are not quite d's and neither are they Greek deltas, but something peculiar and halfway between.

This is called a partial derivative, and looks like this:

$\frac{\partial y}{\partial x}$ (read as 'partial-dee-why-by-dee eks')

It turns up when you have a quantity that varies with two quantities. For example, if you have a surface defined in a three-dimensional coordinate system, like this:




then you can see that z is a function of both x and y, because as any point P roams about on the surface, z takes on a variety of different values as x and y change. The two lines drawn on the surface are parallel to the x and y axes, though. If the point moves along one of them x will change, but y will remain



constant. But if the point moves along the other line, then y changes as x remains constant. So we can consider the partial differential with respect to x and the partial differential with respect to y.

In practice, suppose we have a surface with an equation

$$z = 3x^2 + 4y^2$$

then to find $\partial z/\partial x$ we differentiate with respect to x, treating y as a constant:

$$\frac{\partial z}{\partial x} = 6x$$

and to find $\partial z/\partial y$ we differentiate with respect to y, treating x as a constant:

$$\frac{\partial z}{\partial y} = 8y$$

It's as simple as that.

Finally, a word concerning differential equations is in order. In the Horizons starter software pack that comes free with the Spectrum is a nice example of the use of differential equations. After building it up on the cover of the cassette as 'complex', the program shows a population model concerning foxes and rabbits, and it's good fun (and instructive) to watch. It revolves around the classic example always used to illustrate how differential equations can be applied, and at its heart are two differential equations.

But what are differential equations? Nothing too complicated in fact, just an ordinary equation with ordinary terms except that one of the terms (at least) is a derivative of some kind. As with determinants, you can say a differential equation has an order, so that if its highest derivative is a first derivative it will be of order one, if there's a second derivative, but none higher, it will be of order two, and so on.

An example of a differential equation of the second order is:

$$\frac{d^2y}{dx^2} - 3\frac{dy}{dx} + 2y = 0$$

In general, you can't solve a differential equation without using integration, which we haven't yet looked at. So we can't look at that, but we can have a little look at the rabbits and foxes thing.

The idea is that you have an island, and to keep it simple it's inhabited only by rabbits and foxes. The foxes eat the rabbits, and the rabbits eat grass. But since it's a limited island, if you suppose there's plenty of rabbits around, then the number of foxes will increase (because of plentiful food supply). Soon there will be so many foxes that the rabbits begin to have a rough time of it, and their numbers begin to decrease. Soon, the rabbits decline so much that they are difficult to find, and scarce rabbit means hungry fox, so the island can't support so many foxes and they decline too. Pretty soon, the fox population is so low that hardly any rabbit eating is being done, so the rabbit population begins to pick up again, and so we are back to the beginning again.

It's worth working through this because it really is a classic example of a model being made of reality by a scientist (in this case a very simple ecological model). In mathematical terms, you have this:

Let the rabbit population be r. Let the fox population be f. That means that the rate of change of the rabbit population is the derivative of r with respect to time, which we write as

$$\frac{dr}{dt}$$

And similarly the rate of change of the fox population is

$$\frac{df}{dt}$$

Now for the clever bit. We suppose that the rate of change of the rabbit population depends on two factors:

1 The number of rabbits (since the more rabbits there are, the more baby bunnies will be born!).
2 The number of rabbits times the number of foxes.

Also, we suppose the fox population depends upon two factors:

1 The number of foxes (making more baby foxes).
2 The number of foxes per rabbit, f/r.

Now if something is proportional to something else, then we can write that it equals a constant times that factor. So we can write two differential equations, like this:

dr/dt=a*r−b*r*f

and

df/dt=c*f−g*f/r

where the letters a, b, c and g are constants. So if you look at the situation after an increment of time (small period) called dt, the rabbit population will be incremented by dr and the fox population by df. And by using a FOR . . . NEXT loop to increment the time by a small amount and by resorting to a number of different graphical display techniques, the whole operation can be followed through.

Clearly, the solution of a pair of differential equations like this depends on the initial conditions: in our case, on how many rabbits and foxes you start with. Take a look at the program and see what I mean.

As with ordinary equations, differential equations come in a variety of different forms, and there are standard methods you can use to solve them. But this really is getting into the realms of higher maths, so we'll have to leave it for another book.

# 13

# ANTI-DIFFERENTIATION

## 1 Anti-differentiation

You know that the raising of e to a power can be reversed by taking logs (to base e). And you've just read about the process of differentiation. So it's no great leap into the unknown to start considering the reverse process of differentiation.

It does exist. It is important in maths and is called integration.

The first thing to remember is that the notation is a little off-putting: it looks difficult, and so many people assume it is. But in reality it is only the reverse of differentiation, and that can't be too hard.

If you have

$y=x^2$

then

$dy/dx=2x$

So it seems reasonable to suppose that, if you have $y=2x$ and then integrate it, you will get something equal to $x^2$. That something is called the *integral of y with respect to x*, and we represent the word integral with a long kind of drawn out letter S that looks like this:

$\int$

We represent the 'with respect to x' by the symbol:

dx

So that we write that if

$y=2x$

then



$\int y\ dx=x^2$

Pretty straightforward, isn't it. It looks like the issue of the Chinese typewriter is raising its ugly head again, but we'll get to that in a moment. We've managed to overlook one small point: if you differentiate a constant you get zero.

This means that, when you integrate, there is no way of knowing what it was in the first place. For example, if

$y=x^2+C$ (where C is a constant)

then

$dy/dx=2x$

So really, to take account of this, we should have added that possible constant in when we integrated it back. It could be zero, or any numerical value, and without further information we have no way of telling, but we have to add it in like this: if

$y=2x$

then

$\int y\ dx=x^2+C$

Of course, if you are supplied with more information, you can find the value of C. So if, for example, we know that the LHS of the equation is 10 when x is 3, then

$10=3^2+C$

so

$C=10-9=1$

In the same way that there is a general method of going from y to dy/dx, there is also a reverse method to go from dy/dx to y and from y to $\int y\ dx$. Compare the two methods:

$$\begin{array}{lcl} y & \xrightarrow{\text{differentiation}} & dy/dx \\ x^n & & n*x^{n-1} \end{array}$$

$$\begin{array}{lcl} dy/dx & \xrightarrow{\text{integration}} & y \\ x^n & & \dfrac{x^{n+1}}{n+1}+C \end{array}$$

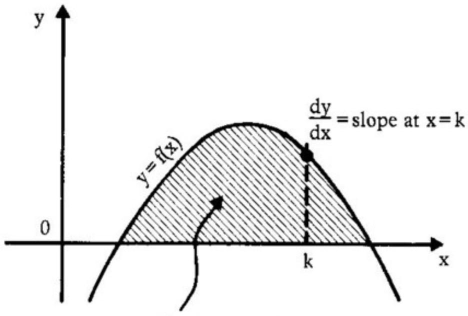
Notice that to differentiate we multiplied by the index and then subtracted one from the index. Now when integrating we add one to the index and divide by the new index. And we do not forget to add in C (which is called the constant of integration).

Before looking at a program, I just want to discuss the meaning of an integral. You remember that if y equals some function of x,

$$y = f(x)$$

then you can draw a graph of it, and the derivative (differentiated form), dy/dx, gives you the slope of the curve. By substituting in a value of x, you can find the slope of the curve at any given point.

Now, the integral, $\int y\ dx$, also has a meaning, and it in fact represents the area under the curve.



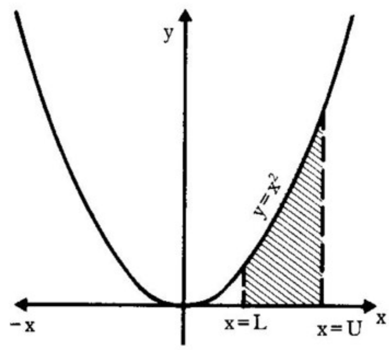

$\int y\ dx$ = Area between curve and x-axis

If you know the world of calculus well, you'll be aware that the integration sign often gets written with a couple of small numbers against it, one at the top and one at the bottom, like this:

$$\int_0^5 y\ dx$$

They're called limits and are values of x between which you are considering the integral. Not all functions obligingly curve over below the x axis again like in our diagram, so that the area under the curve would be infinite. Take the $x^2$ function for example:






The shaded area is bounded by the curve, the x-axis and the lines x = L and x = U. Here L is the lower limit and U the upper limit, so that we could write it out as an equation:

Shaded Area = $\int y\ dx$

$$= \int_L^U x^2\ dx = \left[\frac{x^3}{3}\right]_L^U$$

If you like, you can regard these limits as meaning the area from x = 0 to U minus the area from x = 0 to L. So if you substitute the limit values into the equation you can subtract them to get a value for the area. Suppose we'd said that U = 5 and L = 3, then we'd have

$$\left[\frac{x^3}{3}\right]_3^5 = \frac{5^3}{3} - \frac{3^3}{3} = 41⅔ - 9 = 32⅔$$

This integrating between definite limits is called, not unreasonably, definite integration. You will have noticed that the constant of integration has not made its appearance here, and that's because the limits make up for it.

It's an interesting business because it allows us to find areas of quite sophisticated shapes, as we've already seen. The curve $y = x^2$ is by no means a simple shape (it's a parabola).

Now for some computing. First we need to define an integration sign, and because we're keen to avoid the character spilling out of one character square, we'll take the liberty of shortening it somewhat. I think you'll agree that it's just about recognisable!

```
10 FOR r=0 TO 7: READ r: POKE USR "a"+n,r:
   NEXT r: DATA 8,20,16,16,16,16,80,32
```

If you run that on its own it will set up the ∫ sign on the A key (graphics mode), or you can tuck it into a program if you prefer.

This program sorts out your polynomials for you and is in effect just the reverse of Program 88.

*Program 91 POLYNOMIAL INTEGRATION*

```
   5 REM PROGRAM 91
     POLYNOMIAL INTEGRATION
  10 BORDER 6: PRINT " INTEGRATI
ON OF POLYNOMIALS"
  20 PRINT : PRINT "              n
              n+1"
  30 PRINT "If y= a*x    then   y
dx= a*x  +C": PRINT  OVER 1;AT 3
,24;"___": PRINT AT 4,24;"n+1"
  40 PRINT AT 7,3;"For Example:"
  50 PRINT : PRINT "              y
="
  60 INPUT "Choose constant a: "
;a
  70 PRINT AT 9,14;a;" x"
  80 INPUT "Choose power n: ";n
  90 PRINT AT 8,18;n
 100 PRINT : PRINT "   So,"
 110 IF n=-1 THEN  PRINT AT 14,1
0;" ydx = a LN x"
 120 IF n<>-1 THEN  PRINT AT 14,
10;" ydx = ";a;" x";AT 13,21;n+1
;AT 16,18;n+1;AT 14,24;"+ C": PL
OT 136,52: DRAW 24,0
 130 PRINT AT 20,0;" ( ydx shows
 area under curve.)": STOP
```



```
1000 FOR n=0 TO 7: READ r: POKE
USR "a"+n,r: NEXT n: DATA 8,20,1
6,16,16,16,80,32
```

Remember to do

RUN 1000

to get the UDG fixed up first, and where you see ∫ in the listing, you must type "a" in Graphics mode.

For those of you who are keen to get on with more integration, there are UDGs in the appendix that squeeze 'dx' into one character space, and a rather more professional integration sign that stretches from the character square above the line and into the one below it.

But down to business: if you run Program 91, when a=1 and n= −1, then you are integrating this:

$y = 1/x$

and, although it's hard to see why,, it's true that

$\int y dx = \log_e x + C$

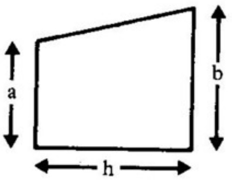
so line 110 is there to take care of it.

It is tempting to reproduce a version of Program 89 which shows how you get standard derivatives of the common functions, but this time showing standard integrals. In fact, it's a bit of a waste of time, because once you've got a list of derivative conversions, you can easily see what 'going in the other direction' will give you (always remembering the constant C, of course!).

Most maths texts have a list of standard integrals for lots of different functions, and the more comprehensive the book the longer the list. (I've included some later on, but not many.) Definite integrals are something we can get at a bit easier on the computer, because here we are dealing with numbers rather than manipulation of formulas. The principles of this aspect of integration need an explanatory word, so let's take a look at Simpson's rule and the trapezoidal rule.

Taking the trapezoidal rule first, what we're doing is just the same as if you drew an irregular shape on some graph paper and tried to find the area by counting up all the tiny squares. The difference is that the trapezoidal rule uses strips.

A trapezium, incidentally, is a shape like this:

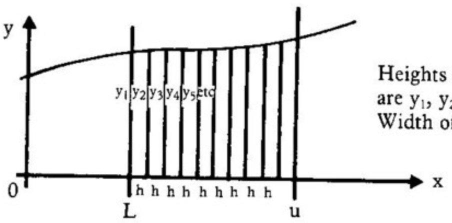



Trapezium
Area = ½(a + b)*h

with one pair of opposite sides parallel. And if you take a function and the area under it, you can make strips of the area, so that each of the strips is a trapezium. (This, of course, means that you're considering the curve to be a series of straight lines, but that unfortunately is where the approximation comes in.)

Clearly, if you make the strips exceedingly small and narrow, you will get more accurate results, but the price you pay is having to deal with more strips. Happily, we can take advantage of the accuracy, and the computer will foot the bill!

Suppose we illustrate it with ten strips:




Heights of LHS of strips are $y_1, y_2, y_3, \ldots, y_{11}$
Width of each strip is h

Then our area will be the sum of the trapeziums:

$$\text{Area} = \frac{y_1+y_2}{2}h + \frac{y_2+y_3}{2}h + \ldots + \frac{y_{10}+y_{11}}{2}h$$

so

$$\text{Area} = h\left(\frac{y_1+y_{11}}{2} + y_2 + y_3 + y_4 + \ldots + y_{10}\right)$$

which in words amounts to

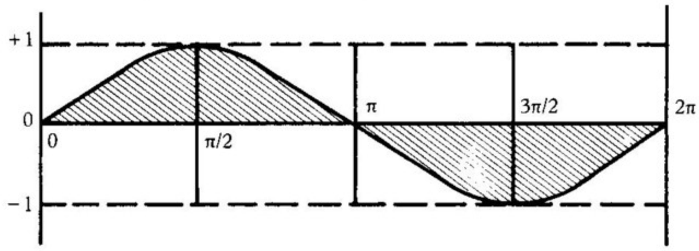


Area = width of strip*((first edge + last edge)/2 + sum of others)

*Program 92 TRAPEZOIDAL RULE*

```
   5 REM PROGRAM 92
     TRAPEZOIDAL RULE
  10 INPUT "Upper Limit? ";U
  20 INPUT "Lower Limit? ";L
  30 DIM A(101): LET RANGE=U-L:
LET h=RANGE/100: LET n=1: LET s=
0
  40 FOR x=L TO U STEP h
  50 LET y=SIN x
  60 LET A(n)=y
  70 LET n=n+1: NEXT x
  80 FOR n=2 TO 100
  90 LET S=S+A(n)
 100 NEXT n
 110 LET AREA=h*(((A(1)+A(101))/
2)+S)
 120 PRINT AREA
```

In this particular incarnation, we're looking at y = Sin x. The function can be put in at line 50 to your own taste. The integration of the Sin x function is interesting. Remember we're in radians on the computer, so you could set upper limit to PI and lower limit to zero. The answer is approximately 2 units. The area over a full revolution, however, (from 2*PI to zero), is approximately zero, because the area below the x axis between PI and 2*PI is counted as negative. So you should beware of that when using this rule with the repetitive functions.



Simpson's rule is a refinement that tries to lay pieces of parabola against the curve instead of straight lines. The derivation is not particularly hard to follow if you care to look it up some time. The rule amounts to this:

$$\text{Area} = \frac{h}{3}(y_1 + 2(y_{odd}) + 4(y_{even}) + y_{11})$$

or in words:

Area = h/3*((first edge + last edge) + 2(odd edges) + 4(even edges))

It is a refinement that is generally more accurate, and you can run the two programs on a given integration problem to see how they both compare with the real answer!

*Program 93 SIMPSON'S RULE*

```
  5 REM PROGRAM 93
    SIMPSON'S RULE
 10 INPUT "Upper Limit? ";U
 20 INPUT "Lower Limit? ";L
 30 DIM A(101): LET RANGE=U-L:
LET h=RANGE/100
 40 LET n=1: FOR x=L TO U STEP
h
 50 LET y=SIN x
 60 LET A(n)=y
 70 LET n=n+1: NEXT x
 80 LET EVEN=0: LET ODD=0
 90 FOR n=2 TO 100 STEP 2
100 LET EVEN=EVEN+A(n)
110 LET ODD=ODD+A(n+1)
120 NEXT n
130 LET AREA=(h/3)*(A(1)+A(101)
+(4*EVEN)+(2*ODD))
140 PRINT AREA
```

Notice that in both Program 92 and 93 we are using 100 strips which takes a little time to calculate (so please excuse the blank screen!)

In the same way that differentiating successively will give you



$dy/dx$, $d^2y/dx^2$, $d^3y/dx^3$, etc.

you can get successive integrals, too, like this:

$$y = \int\int\int f(x)dx\ dx\ dx$$

And as you might imagine, you just integrate one at a time, like stripping off coats of paint. It's really like a ladder, where you differentiate to go down a rung and integrate to go up a rung. In fact, now you know about it, it really makes you wonder why you thought there was anything very complicated about it. Like I've been saying all along, it's simple stuff dressed up in jargon to make it appear difficult.

But to return to the mainstream of our discussion, not every time you see integration signs ganging up together will they be straightforward successive integrations. Sometimes they're multiple integrals, so that the first integration is with respect to x, the second might be with respect to y and the third with respect to z:

$$\int\int\int f(x,y,z)dx\ dy\ dz$$

For example,

$$\int\int\int (x^2 + y^2 + z^2)dx\ dy\ dz$$

And it might interest you to know that the result is the same whichever order you integrate in.

Standard integrals worth listing here are few, because they are in every book on integration you care to look in, but to save you the effort:

| y | $\int y\ dx$ |
|---|---|
| $1/x$ | $\log_e x + C$ |
| $a^x$ | $a^x * \log_a e + C$ |
| $e^x$ | $e^x + C$ |
| Sin x | $-$Cos x + C |
| Cos x | Sin x + C |
| Tan x | $-$log Cos x + C |
| Sinh x | Cosh x + C |
| Cosh x | Sinh x + C |
| Tanh x | log Cosh x + C |

And now you know that the integral of Sin x is $-\text{Cos } x$, you can check that example used with the trapezoidal and Simpson's rules: if

$$y = \text{Sin } x$$

then

$$\int_0^{\pi} \xi \, \delta\chi = \Big[-\text{Cos } x\Big]_0^{\pi}$$
$$= (-\text{Cos } \pi) - (-\text{Cos } 0)$$

And we can show that Cos $\pi$ is $-1$ and Cos 0 is 1, so our area is:

$$-(-1) - (-1) = 1 + 1 = 2$$

How does Simpson do?

Now, I'm not suggesting that you're going to be able to find a standard integral identical to any given expression you want to integrate. So how do you proceed when faced with a complicated expression? I'd love to be able to say 'Simple' and to give you a short program to deal with it, but I can't.

It takes quite a bit of ingenuity to figure out how to integrate some expressions, and all I can say is that you have to practise hard if you want to get to be good at it. There are a few standard methods for products and so on, but it is mostly a question of experience. Generally speaking, the expression must be prepared so that it fits one of the standard forms, and that can be done in a variety of different ways such as 'trigonometric substitution' or using 'partial fractions'.

So far we've looked at integration as 'anti-differentiation' and seen that the process is essentially the finding of areas under curves, but perhaps we should look closer at the numbers aspect of it.

Suppose you have $y = f(x)$ (y is a function of x), then you can draw up a table of corresponding values just as you would if you were going to draw a graph. You choose a suitable interval for x, which amounts to choosing to cut the area into small strips, and then you sum the strips together to get the whole area. The important point to note is that in integration the strips are made vanishingly small. The width of those strips goes from being a small width, $\delta x$, to being vanishingly small, dx.

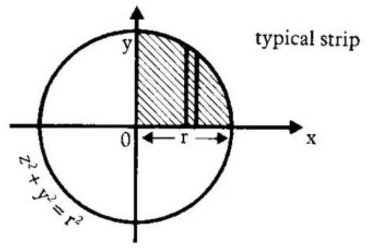


It is possible to imagine that, if you had a plane area made up of strips, you could also make up a solid figure from slices, and there's a good chance of being able to use a method involving integration to find volumes. And so you can.

It's also possible to find the lengths of curves by using an equation that uses both differentiation and integration.

Before leaving the subject, I would like to demonstrate the method for areas, using as our example a figure we already know about, the circle.

First we have to decide how to carve up the circle into strips, and the most convenient way to do it is like this:




We know that the equation of that circle is $x^2 + y^2 = r^2$ where r is the radius of the circle. The area we're going to look at is the quadrant that I've shaded, and you can see that our representative slice goes from limits of 0 to r. So

$$\text{Area of quadrant} = \int_0^r y \, dx$$

and we know that

$$x^2 + y^2 = r^2$$

and so

$$y = \sqrt{r^2 - x^2}$$

Therefore

$$\text{Area} = \int_0^r \sqrt{r^2 - x^2} \, dx$$

Without worrying about how to integrate that expression, suppose you'd looked it up as a standard integral and found it to be

$$\frac{r^2}{2}\star \text{Sin}^{-1}(x/r)+\tfrac{1}{2}x\star\sqrt{r^2-x^2}$$

So our area is that expression with limits of 0 and r, and we substitute values of r and 0 wherever we find x.

When $x=r$, $\text{Sin}^{-1}(x/r)=\text{Sin}^{-1}(1)$ (the angle whose sine is 1) which, in radians is PI/2.

And by similar reasoning, $\text{Sin}^{-1}(0/r)=\text{Sin}^{-1}0=0$

So we have

$$\text{Area}=(\tfrac{1}{2}r\sqrt{r^2-r^2}+\tfrac{1}{2}r^2\star \text{PI}/2)-0$$

which reduces to,

$$\text{Area}=\frac{\text{PI}\star r^2}{4}$$

But remember that this is the area of our quadrant which is only a quarter of the full circle, and so the area of the full circle is four times this:

$$\text{Area of circle}=\text{PI}\star r^2$$

which we know is correct.

It might seem a sledgehammer cracking a nut, but in fact the method can be used for other shapes, any shape you know the equation of, and so it is a powerful idea. Our circle example is interesting, though, and it might be a test of our powers to do it in polar coordinates. Actually, that makes the problem much simpler to handle, but that, as they say, is another story . . .

# 14

# THE RANDOM FACTOR

## 1 The random factor

The final chapter, sorry to say, but it deals with one of the most interesting aspects of the Spectrum microcomputer. In fact the contents of this chapter really serve as an introduction to one of the most fascinating branches of mathematics, the mathematics of chance.

It's a funny old world, nothing is certain, nothing is absolute. Right from tiny electrons to vast galactic groupings, our universe is one where matter and energy move and interact with apparent randomness or unpredictability. Yet there are rules, for all is not chaos, and the forms which mathematicians and scientists observe in the universe arise out of those rules.

In Tom Stoppard's play *Rosenkrantz and Guildenstern are Dead*, the two principal characters open by tossing a coin which, no matter how many times they toss it, always comes up heads. This leads them to deduce that something has happened to their world (they are in fact dead!) and serves to remind us that out of unpredictability comes some clue to the world we live in and the rules which govern it. Why don't we take a look at it?

Let's start with the function RND which is found on your Spectrum in green above the T key. RND stands for random, and the idea is that if you say

```
PRINT RND
```

the computer will give you back a number between Ø and 1, and that number will seem completely random.

Try it a few times or, better still, get a page full of these random numbers by running this:

```
1Ø FOR n=1 TO 21: PRINT RND : NEXT n
```

You will see that the numbers are indeed between Ø and 1, and don't look obviously like the numbers of any well defined sequence or series, such as those we've been used to seeing. It would be easy to convince yourself that the numbers were really independent of one another and completely random, but unfortunately it isn't true.

The RND function is derived from a function. If you think about it, it's going to be very difficult to get a thing so intensely ordered and predictable as a microchip to be really random: they're just not built that way. So we have to simulate it. And the function the Spectrum uses is in effect a jumbled sequence of numbers that depends upon the length of time after the computer was switched on.

You can look into the memory location that stores this number, just to see that it exists, by running this:

```
10 PRINT AT 0,0; PEEK 23672; AT 0,0;"  "
20 GO TO 10
```

Not very satisfactory because it flashes due to the blanking out you must do between each printing, but you can just about make out that this number is running from Ø to 255. It gets bigger by one every fiftieth of a second (which means it runs through the range of numbers in 256/5Ø secs, or a little over 5 seconds), and then it starts again.

The RND function picks out this number and uses it to calculate the random number you get back, and although it's not truly random, it's good enough for most games and for some statistical modelling.

So we have a random number between Ø and 1. What if we want a random number between Ø and 1Ø?

Easy: try this:

```
10 FOR n=1 TO 21: PRINT (RND*10):NEXT n
```

And if you want integers between Ø and 1Ø, slip in an INT function:

```
10 FOR n=1 TO 21: PRINT INT(RND*10): NEXT n
```

And if you want a random integer between 5 and −5, you have to subtract five:

```
10 FOR n=1 TO 21: PRINT INT(RND*10)-5: NEXT n
```



I'm sure you get the idea. The point to note is that you can choose the range of your random number. But beware: the number you multiply RND by is an absolute upper limit of that range. So if you have (RND*1Ø), you can never get 1Ø, only 9.999999

Now many games programs incorporate the random factor so that your 'opposition' behaves rather unpredictably. But when you're designing a game, it's a real nuisance to have your variables jumping about when you're trying to test a program. Far better if you could select the RND sequence from a given point each time instead of it depending on the contents of memory location 23672. The instruction RAND (white legent on the T key) allows you to do this. You can start anywhere in the sequence of 65535 random numbers by typing

```
RANDOMIZE n
```

where n is in the range $1 \leqslant n \leqslant 65535$.

To see what I mean, try the following:

```
10 RANDOMIZE 1: FOR n=1 TO 21: PRINT RND :
   NEXT n
```

Run this several times, and each time you get the same sequence of twenty-one numbers. But take out the randomize function, and you will get a different sequence each time. And if you alter the number associated with RAND you get a different sequence from the one you started with, but the new sequence will be the same each time you run it. Try it.

What we need to do is examine probabilities, look at something called permutations and combinations, and then perhaps show how we can model some physical situations on the computer using what we have learned in theory.

## 2 Permutations and combinations

A permutation is just the name we give to a way of ordering things. I don't mean as in ordering eggs and bacon for breakfast, but ordering as in putting into a given order. For example, if we have three letters, A, B, and C, how many permutations are there?

There are six:

A B C
A C B
B A C
B C A
C A B
C B A

And that's it: there are no other ways to do it.

If you have two objects, then you have only two permutations:

A B
B A

And if you have four? Well, let's see:

| | | | |
|---|---|---|---|
| A B C D | B A C D | C A B D | D A B C |
| A B D C | B A D C | C A D B | D A C B |
| A C B D | B C A D | C B A D | D B A C |
| A C D B | B C D A | C B D A | D B C A |
| A D B C | B D A C | C D A B | D C A B |
| A D C B | B D C A | C D B A | D C B A |

And there we are: 24 permutations. But this is tedious, and maybe we can think of a way to automate it. Let's look at the facts so far:

| | | | | | |
|---|---|---|---|---|---|
| No. of objects: | 1 | 2 | 3 | 4 | . . . |
| No. of permutations: | 1 | 2 | 6 | 24 | . . . |

Do you recognise this?

Try Program 69, and then come back.

So that's it! The factorial function. That was the physical meaning of it.

Suppose now we have a bag containing letters. Suppose we have the letters A, B, C and D. How many ways are there of taking just two of those letters out?

We could have:

1 A and B
2 A and C
3 A and D
4 B and A



5 B and C
6 B and D
7 C and A
8 C and B
9 C and D
10 D and A
11 D and B
12 D and C

Twelve options. There's a way of representing this. We use a letter P (for permutation), and if we're choosing 2 from 4 as above, we'd write

$$^{4}P_{2}$$

The top number is the total you're picking from, and the bottom number is the number you're selecting. In our case

$$^{4}P_{2} = 12$$

and there is a general formula for choosing so many items from a total number:

$$^{n}P_{r} = n!/(n-r)!$$

So in our case we'd have

$$^{4}P_{2} = 4!/(4-2)! = 24/2 = 12$$

which all checks out nicely.

But look again at our example. So long as you take notice of the order that you drew the letters out of the bag, there would indeed be twelve permutations, but if the order doesn't matter to you, and drawing a pair, say, A and B, is the same as drawing the pair B and A (options 1 and 4 in our list), then you have fewer outcomes.

A careful examination of the list will show you that there are now only six outcomes. We call these outcomes combinations, and we represent this mathematically as

$$^{n}C_{r} = 6$$

It so happens that

$$^{n}C_{r} = {}^{n}P_{r}/r! = n!/(n-r)!*r!$$

This is so because each permutation of $^{n}P_{r}$ consists of r objects which can be arranged amongst themselves in r! ways.

And now that's out of the way, how about a program or two? If you can't recall how we generated factorials, you might like to refresh yourself by taking a look back over Program 69. Then try Program 94.

*Program 94 PERMUTATIONS*

```
   5 REM PROGRAM 94
     PERMUTATIONS
  10 PRINT "  PERMUTATION IS THE
 NUMBER OF      WAYS OF CHOOSING R
 OBJECTS        FROM A TOTAL OF N.
"
  20 INPUT "TOTAL NUMBER OF OBJE
CTS? ";T
  30 INPUT " NUMBER OF CHOSEN OB
JECTS? ";R
  40 LET N=T: GO SUB 100
  50 LET FT=f: LET N=T-R: GO SUB
 100
  60 LET P=FT/f: PRINT AT 14,1;"
PERMUTATIONS ";P: STOP
 100 LET f=N: FOR x=N TO 2 STEP
-1: LET f=f*(x-1): NEXT x: RETUR
N
```

The subroutine in line 100 is the bit of code that works out the factorial. (This has the same restriction on it as the original factorial program so far as maximum number is concerned. That is, it will work with numbers up to 33, but go bigger and you will get back a 'Number too big' error report.)

Now how about a combinations routine?



*Program 95 COMBINATIONS*

```
   5 REM PROGRAM 95
     COMBINATIONS
  10 PRINT "  COMBINATION IS THE
 NUMBER OF      WAYS OF CHOOSING R
 OBJECTS        FROM A TOTAL OF N,
WHEN THE        ORDER OF SELECTION
 DOES NOT       COUNT."
  20 INPUT "TOTAL NUMBER OF OBJE
CTS? ";T
  30 INPUT " NUMBER OF CHOSEN OB
JECTS? ";R
  40 LET N=T: GO SUB 100
  50 LET FT=f: LET N=T-R: GO SUB
 100
  60 LET RT=f: LET N=R: GO SUB 1
00
  70 PRINT AT 14,6;"COMBINATIONS
 ";FT/(RT*f): STOP
 100 LET f=N: FOR x=N TO 2 STEP
-1: LET f=f*(x-1): NEXT x: RETUR
N
```

Before moving on, just take a look at this for a bit of mathematical connectedness. You can use the combinations program to get a line of Pascal's triangle! Say you want the sixth line, you start by writing down 1. Then the next term is $^{6}C_{1}=6$, the next is $^{6}C_{2}=15$, then $^{6}C_{3}=20$, then $^{6}C_{4}=15$, then $^{6}C_{5}=6$, and lastly a 1. You can write out the line in full:

1 6 15 20 15 6 1

It works for any line you like. And notice that it's symmetrical, so that it follows that

$$^{n}C_{r} = {}^{n}C_{n-r}$$

It also follows that in the binomial expansion,

$$(a+x)^{n}$$

is a series of terms like

$^{n}C_{r} * a^{n-r} * r^{r}$

which leads on to something called recurrence or recursion formulas, but we're not going to look at those here.

What we really ought to do is have a look at the wonderful world of randomness again, this time with permutations and combinations under our belt and looking through the window on probability.

## 3 Probability

This is important if you are going to take up gambling or engage in any prediction-making, and the way we approach it is via those classic tools of chance, the coin and the die. ('Die' is just the singular of 'dice', in case you didn't know: you have one die or several dice.)

Take a coin and ignore the remote possibility that it can come down on its edge (or not come down at all). Then you can say that it has two possible states: it can be 'heads' or it can be 'tails', and there is obviously an equal chance of it being either.

Then we say that the probability of heads is Ø.5, and the probability of tails is Ø.5, which we write like this:

p(heads) = Ø.5
p(tails) = Ø.5

Note that p(tails) OR p(heads) = Ø.5 + Ø.5 = 1, and that a probability of Ø means 'impossible' and a probability of 1 means 'certain'.

It is important to realise that this does not predict that, if you toss a coin ten times, then you will get five heads and five tails. What it does mean is that, if you toss the coin a vast number of times, you get half heads and half tails (more or less), and that, if you could toss it an infinite number of times, you would get half heads and half tails. It is related to the number of times you try it, and as the number of trials tends to infinity, the results tend to their predicted probability. Let's simulate this with a program:



*Program 96 PROBABILITY TRIALS*

```
   5 REM PROGRAM 96
     PROBABILITY TRIALS
  10 INPUT "Number of trials? ";
N: LET H=0: LET T=0
  20 FOR x=1 TO N
  30 IF RND>=.5 THEN  LET H=H+1:
 PRINT AT 10,24;H: NEXT x
  40 LET T=T+1: PRINT AT 10,8;T
  50 PRINT AT 4,15;x
  60 IF T<>0 THEN  PRINT AT 18,1
2;H/T
  70 NEXT x
```

Whilst we must bear in mind that we're not dealing with a random phenomenon here, just with the computer's pseudo-random generator, it is sufficiently good to demonstrate the notion. The program prints the number of the trial at the top of the screen, the number of tails on the left, the number of heads on the right and finally the ratio of heads to tails at the bottom. If you run the program again and again with larger and larger numbers of trials, you will find that the bottom number will generally get closer to one.

Now for a dice program. I could take ages explaining about this program and how it works, but it would be better if you just loaded it on to your computer and played it a while. The only thing to remember is that it simulates the throwing of a pair of dice 500 times and builds up what is known as a frequency histogram of the results. The results can be any integer from 2 to 12 (got by adding the scores of the two dice, like when you play Monopoly), and if waiting four seconds between throws begins to bore you, just hold a key down to speed things up. The program freezes after 500 goes, and you have to break it in the usual way by pressing CAPS SHIFT and the BREAK keys simultaneously.

*Program 97 DICE*

```
   2 REM PROGRAM 97   DICE
   5 FOR x=2 TO 12: PRINT  PAPER
4;AT 0,(x*3)-6;x: NEXT x
```

```
  10 DIM x(12): PRINT AT 4,14;"T
HROW";AT 7,6;"DIE 1";AT 7,22;"DI
E 2"
  20 FOR n=1 TO 500
  30 LET D1=INT (RND*6)+1
  40 LET D2=INT (RND*6)+1
  50 LET S=D1+D2
  60 LET x(s)=x(s)+1: PRINT AT 1
,(s*3)-6;x(s): PLOT (s*24)-48,x(
s): DRAW  INK (s/2)-1;15,0
  70 PRINT AT 5,16;n;AT 8,8;D1;A
T 8,24;D2
  80 PAUSE 200
  90 NEXT n
 100 GO TO 100
```

Sooner or later you will find that, although the throwing of dice is random, the histogram you get at the bottom of the screen builds up in a characteristic way. There are generally taller strips in the middle and shorter strips at the edges. Why should this hump shape occur?

Fortunately throwing a couple of dice is a nice simple system to look into, and that's just what I propose we do.

First let's examine how we arrive at any given score. We have die number 1 with possible scores of 1, 2, 3, 4, 5 and 6, and the same for die number 2. So for a total score of 2, we must have a 1 on die 1, and a 1 on die 2, and this is the only way you can score 2.

To get a score of 3, you can have either a 2 on die 1 and a 1 on die 2, or vice versa, i.e. there are two separate ways.

To get a score of 4, you can have a 2 on both dice, a 3 on die 1 and a 1 on die 2, or vice versa, which is three ways. And so it goes on. To summarise:

| Score (n) | Number of ways to get it | p(n) |
|---|---|---|
| 2 | 1 | 1/36 |
| 3 | 2 | 2/36 |
| 4 | 3 | 3/36 |
| 5 | 4 | 4/36 |
| 6 | 5 | 5/36 |
| 7 | 6 | 6/36 |
| 8 | 5 | 5/36 |
| 9 | 4 | 4/36 |
| 10 | 3 | 3/36 |
| 11 | 2 | 2/26 |
| 12 | 1 | 1/36 |



This shows that there are six ways to get a score of 7 and makes 7 the most popular result. Obviously, if the dics are not loaded, each of the 36 different score combinations are equally likely, and so we would expect the probability of a 7 score to be six times as likely as a 2 score or a 12 score. We also know the absolute probabilities will be, for example, 6/36 for a score of 7, which is 1/6, or 0.1666667, and for a score of, say, 3 probability is 3/36 = 1/12, which is 0.0833333. You will find that, if you convert all the p(n)s into decimals and add them together, you will get 1. This must be so, since it is certain that if you throw a couple of dice, the score must be between 2 and 12 inclusive!

Try running the dice program a few rounds, and compare the frequency with which the combinations turn up with the predicted probabilities. Just divide the occurrence numbers for each score (at the top of the screen) by 500, and compare them with the p(n) predictions.

## 4 Probability distributions

Around the seventeenth and eighteenth centuries there flourished a family called Bernoulli that turned out a whole bunch of excellent mathematicians and scientists. One of them, probably Jacques Bernoulli (1654–1705), came up with a thing called the Bernoulli distribution, otherwise known as the binomial distribution. It's quite interesting, and it goes like this. If the probability of something happening is p, and the probability of it not happening is q (where q = 1 − p, since p + q must equal 1), then what is the probability of it happening x times out of n trials?

The formula that tells you is:

$$p(x) = {}^{n}C_{x} \star p^{x} \star q^{(n-x)}$$

So, for example, the probability of getting 2 heads in 6 tosses of a coin is

$$p(2) = {}^{6}C_{2} \star (0.5)^{2} \star (0.5)^{6-2} = \frac{6!}{4!2!} \star (0.5)^{6} = 0.234375$$

Program 98 extends this idea, asking you how many trials you want, how many successes you want, and the probability of one success. Try it with the above example, putting in 6, 2 and 0.5 respectively. Or you could see the chance of getting 2 'sixes' with 3 rolls of a die, where the number of trials will be 3, the number of successes 'getting a six' is 2 and the chance of throwing 1 six is 1/6. You ought to get 0.0694.

### *Program 98 BINOMIAL DISTRIBUTION*

```
   5 REM PROGRAM 98
      BINOMIAL DISTRIBUTION
  10 INPUT "NUMBER OF TRIALS? ";
N
  20 INPUT "NUMBER OF SUCCESSES?
 ";X
  30 INPUT "PROBABILITY OF ONE S
UCCESS? ";P
  40 LET M=N: GO SUB 100
  50 LET FT=f: LET M=N-X: GO SUB
 100
  60 LET RT=f: LET M=X: GO SUB 1
00
  70 PRINT AT 14,1;"PROBABILITY
OF ";X;" SUCCESSES"," IN ";N;" T
RIALS IS ";(FT/(RT*f))*(P^X)*((1
-P)^(N-X)): STOP
 100 LET f=M: FOR Y=M TO 2 STEP
-1: LET f=f*(Y-1): NEXT Y: RETUR
N
```

The next idea concerns the business of collecting data. If you think of a collection of people, for example, and you take out a tape measure and start writing down a list of how tall they are, you will get what is known in the trade as 'raw data' on heights. Then supposing you want to draw some conclusions about that data, you need to process it and condense it into two numbers. The first is called the mean, and the second is called the standard deviation.

The first, the mean, is just the average, so you get it by adding all the values together and dividing by the number of values. So if you've got values 2, 3, 4, 5 and 6, you add them together to get 20 and divide by the number of values (five) giving a mean of 4.

The mean, then, is the value about which the values cluster. It is quite useful in drawing conclusions about the data as you must already be well aware!

The standard deviation (SD) is what is known as 'a measure of central tendency' and really just shows you how closely the values are clustering about the mean. If the SD is large, then the clustering is loose, and your values are all over the place. If the SD is small, then the cluster is tight, and all the values are close to the mean.

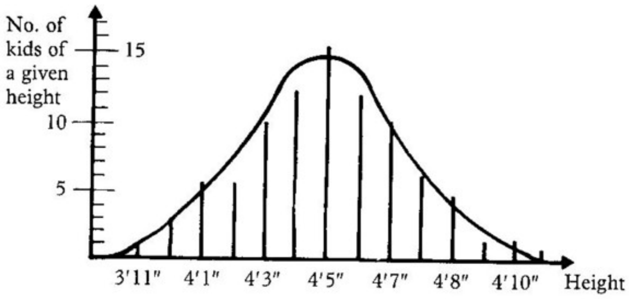
Get hold of some raw data (how you do that is up to you: you might like to count the number of seconds between successive cars on a road or something else liable to be reasonably random). Then plug the values into Program 99. It will give you back the mean and the standard deviation. The way it works out the standard deviation, incidentally, is by taking each of your values away from the mean (to see how far away from the central value each value is), squaring it, adding all those squared differences together and dividing the result by the number of values, and finally (phew!) taking the square root.

### *Program 99 MEAN & STANDARD DEVIATION*

```
   5 REM PROGRAM 99
      MEAN & STANDARD DEVIATION
  10 INPUT "NUMBER OF VALUES? ";
N: DIM A(N): LET S=0
  20 FOR X=1 TO N: INPUT "NEXT V
ALUE? ";A(X): PRINT A(X): NEXT X
  30 FOR X=1 TO N: LET S=S+A(X):
 NEXT X: PRINT AT 0,10;"SUM OF V
ALUES IS ";S
  40 PRINT AT 1,10;"MEAN IS ";S/
N
  50 LET SD=0: FOR X=1 TO N: LET
 D=((S/N)-A(X))*((S/N)-A(X)): LE
T SD=SD+D: NEXT X: PRINT AT 2,10
;"ST.DEV. IS ";SQR (SD/N)
```

Notice that if all of your values are the same, there is no deviation from the central value so the $SD=0$.

The thing about heights of people, or weights of people, or lengths of manufactured components or anything else of a randomly decided nature is that there are usually few of your sample that are much bigger than the mean, and similarly few that are much smaller than the mean. Most of the sample will have characteristics close to the mean value. So if you were to take the heights of a class full of children, measuring to the nearest inch, you might have a sample mean of 4ft 5in. Then if you plotted out the number of people of a given height against the height on a graph, then you'd get a graph that was a hump, with the centre of the hump coinciding with the mean, like this:




The standard deviation is the measure of the sharpness of the hump in an inverse sort of way, so that if your standard deviation is small you have a sharp curve, and if the standard deviation is large then you have a flatter curve. Once you know the mean and the standard deviation of a sample, you can prepare an idealised curve that you can expect the real results more or less to follow (given a big enough sample). This curve has an equation first developed by the German genius Karl Friedrich Gauss (1777–1855), and the distribution is called either the normal or Gaussian distribution.

The equation, in case you're interested, is:

$$EXP(-(X-MEAN)^2/2*SD^2)/SD*\sqrt{2}*PI$$

One interesting little utility is to be able to draw a graph of data points you have collected. Suppose you learn that the profits of a small business over 18 months goes like this:



| | | | | | |
|---|---|---|---|---|---|
| JAN | £320 | JUL | £530 | JAN | £1106 |
| FEB | £310 | AUG | £300 | FEB | £ 995 |
| MAR | £405 | SEP | £495 | MAR | £1047 |
| APR | £312 | OCT | £615 | APR | £1272 |
| MAY | £378 | NOV | £700 | MAY | £1320 |
| JUN | £440 | DEC | £950 | JUN | £1471 |

Now those figures would be much easier to visualise if they were put on a graph, and Program 100 lets you do just that.

*Program 100 GRAPH*

```
   5 REM PROGRAM 100      GRAPH
  10 BORDER 0: PAPER 1: INK 7: C
LS : PLOT 0,175: DRAW 0,-175: DR
AW 255,0
  20 INPUT "HOW MANY DATA POINTS
?";N
  30 LET s=INT (255/N): DIM x(N)
  40 FOR i=1 TO N: PRINT AT 0,0;
"Value no ";i;"? ": INPUT x(i):
NEXT i
  50 PRINT AT 0,0;"
                   ": LET imax=x
(1): FOR i=1 TO N: IF x(i)>imax
THEN  LET imax=x(i)
  60 NEXT i: LET sc=imax/170
  70 DIM y(N): FOR i=1 TO N: LET
 y(i)=x(i)/sc: NEXT i
  80 FOR i=1 TO N: CIRCLE (i*s)-
(s-3),y(i)+3,1: PLOT (i*s)-(s-3)
,y(i)+3: IF i<N THEN  DRAW s,y(i
+1)-y(i)
  90 NEXT i: PRINT AT 0,0;"MAX V
ALUE = ";imax
 100 LET t=0: FOR i=1 TO N: LET
t=t+x(i): NEXT i: LET M=t/N: PRI
NT AT 21,16;"MEAN = ";INT (M*100
)/100
```

There are times, however, when you want to order your data not according to time (as with the business profits), but in order of magnitude (like with the heights). If you have a lot of data, it really is a pain to put it into ascending order (so that the values get bigger). So let me present Program 101 which does it for you. Just tell the computer how many values you have, than enter then one at a time, and the computer will print out its attempts to sort them into ascending order. It will eventually get there and stop.

*Program 101 ORDERED DATA*

```
   5 REM PROGRAM 101
     ORDERED DATA
  10 INPUT "Number of values? ";
n: DIM D(n): FOR x=1 TO n: INPUT
 "Next Value? ";D(x): PRINT D(x)
': NEXT x
  20 PRINT : FOR x=1 TO n-1: LET
 f=0: FOR y=1 TO n-1: IF D(y)<=D
(y+1) THEN  GO TO 40
  30 LET q=D(y): LET D(y)=D(y+1)
: LET D(y+1)=q: LET f=1
  40 NEXT y: IF f=0 THEN  GO TO
60
  50 FOR z=1 TO n: PRINT D(z);"
";: NEXT z: PRINT : PRINT : NEXT
 x
  60 STOP
```

And that just about wraps it up. I hope you've enjoyed our rather abandoned romp through the field of maths. I hope, too, that some of it may be of help to you sometime.

And in case you're put off by the thought of tapping out 100 or so programs, I've put them on to a cassette and made them available, price £5 post and package included, from:

Century Communications Ltd
Portland House
12–13 Greek Street
London W1.

# APPENDIX 1

## User-defined graphics

There follow some lists of data that can be used to define sets of useful characters that do not already exist on the Spectrum. The symbols can be obtained by running a short program that loads the data into a special section of memory, and once loaded they may be got on to the screen by pressing any of the keys A to T when the cursor is flashing G. To get the cursor into graphics mode, press CAPS SHIFT and 9.

The character loading program is below:

*Appendix Program 1 CHARACTER LOADER*

```
   5 REM PROGRAM APPENDIX 1
     LOWER CASE GREEK
  10 FOR n=0 TO 7: READ r: POKE
USR "a"+n,r: NEXT n
  20 PRINT "  ": GO TO 10
  30 DATA 0,0,18,44,68,74,50,0
  40 DATA 56,36,56,36,36,60,32,0
  50 DATA 0,49,74,12,20,20,8,0
  60 DATA 28,32,16,24,36,36,24,0
  70 DATA 0,0,28,32,56,32,28,0
  80 DATA 28,8,8,16,56,4,24,0
  90 DATA 0,88,36,36,36,4,4,0
 100 DATA 56,68,68,124,68,68,56,
0
 110 DATA 8,0,16,16,16,20,8,0
 120 DATA 0,0,72,80,96,80,72,0
 130 DATA 0,32,16,8,8,20,34,0
 140 DATA 0,36,100,36,60,34,32,0
```

```
150 DATA 0,4,2,54,20,8,8,0
160 DATA 16,28,16,28,32,24,4,56
170 DATA 0,0,56,68,68,68,56,0
180 DATA 0,0,62,84,20,36,34,0
190 DATA 0,12,18,18,44,32,64,0
200 DATA 0,0,0,62,72,72,48,0
220 DATA 0,0,62,72,8,16,16,0
230 DATA 0,0,100,36,36,24,0,0
240 DATA 8,8,28,42,28,8,8,0
250 DATA 0,34,84,8,20,37,66,0
260 DATA 8,8,106,42,28,8,8,0
270 DATA 0,0,32,66,82,44,0,0
```

The key your character will appear on is specified in line 20 inside the quotes, and the eight numbers separated by commas in the DATA command of line 30 define the character. This example loads the Greek letter alpha. Other useful symbols are specified in the tables that follow:

LOWER CASE GREEK LETTERS

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| alpha | 0, | 0, | 18, | 44, | 68, | 74, | 50, | 0 |
| beta | 56, | 36, | 56, | 36, | 36, | 60, | 32, | 0 |
| gamma | 0, | 49, | 74, | 12, | 20, | 20, | 8, | 0 |
| delta | 28, | 32, | 16, | 24, | 36, | 36, | 24, | 0 |
| epsilon | 0, | 0, | 28, | 32, | 56, | 32, | 28, | 0 |
| zeta | 22, | 8, | 8, | 16, | 56, | 4, | 24, | 0 |
| eta | 0, | 88, | 36, | 36, | 36, | 4, | 4, | 0 |
| theta | 56, | 68, | 68, | 124, | 68, | 68, | 56, | 0 |
| iota | 8, | 0, | 16, | 16, | 16, | 20, | 8, | 0 |
| kappa | 0, | 0, | 72, | 80, | 96, | 80, | 72, | 0 |
| lambda | 0, | 32, | 16, | 8, | 8, | 20, | 34, | 0 |
| mu | 0, | 36, | 100, | 36, | 60, | 34, | 32, | 0 |
| nu | 0, | 4, | 2, | 54, | 20, | 8, | 8, | 0 |
| xi | 16, | 28, | 16, | 28, | 32, | 24, | 4, | 56 |
| omicron | 0, | 0, | 56, | 68, | 68, | 68, | 56, | 0 |
| pi | 0, | 0, | 62, | 84, | 20, | 36, | 34, | 0 |
| rho | 0, | 12, | 18, | 18, | 44, | 32, | 64, | 0 |
| sigma | 0, | 0, | 0, | 62, | 72, | 72, | 48, | 0 |
| tau | 0, | 0, | 62, | 72, | 8, | 16, | 16, | 0 |
| upsilon | 0, | 0, | 100, | 36, | 36, | 24, | 0, | 0 |



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| phi | 8, | 8, | 28, | 42, | 28, | 8, | 8, | 0 |
| chi | 0, | 34, | 84, | 8, | 20, | 37, | 66, | 0 |
| psi | 8, | 8, | 106, | 42, | 28, | 8, | 8, | 0 |
| omega | 0, | 0, | 32, | 66, | 82, | 44, | 0, | 0 |

```
   5 REM PROGRAM APPENDIX 2
     UPPER CASE GREEK
  10 FOR n=0 TO 7: READ r: POKE
USR "a"+n,r: NEXT n
  20 PRINT " ": PAUSE 0: CLS : G
O TO 10
  30 DATA 0,24,36,36,60,36,110,0
  40 DATA 120,36,36,56,36,36,124
,0
  50 DATA 126,34,32,32,32,32,112
,0
  60 DATA 8,8,20,20,34,34,127,0
  70 DATA 126,34,40,56,40,34,126
,0
  80 DATA 126,70,12,24,48,98,126
,0
  90 DATA 119,34,34,62,34,34,119
10
 100 DATA 62,65,85,93,85,65,62,0
 110 DATA 56,16,16,16,16,16,56,0
 120 DATA 110,36,40,48,40,36,118
,0
 130 DATA 8,8,20,20,34,34,119,0
 140 DATA 99,54,62,42,42,34,119,
0
 150 DATA 103,34,50,42,38,34,115
,0
 160 DATA 66,126,0,126,0,126,66,
0
 170 DATA 56,68,68,68,68,68,56,0
 180 DATA 119,34,34,34,34,34,119
,0
 190 DATA 120,36,36,56,32,32,112
,0
 200 DATA 126,34,16,8,16,34,126,
```

```
0
 220 DATA 127,73,8,8,8,8,28,0
 230 DATA 99,34,20,8,8,8,28,0
 240 DATA 28,8,v2,42,62,8,28,0
 250 DATA 119,34,20,6,20,34,119,
0
 260 DATA 28,73,107,42,62,8,28,0
 270 DATA 28,34,65,65,54,85,119,
0
```

UPPER CASE GREEK LETTERS

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ALPHA | 0, | 24, | 36, | 36, | 60, | 36, | 110, | 0 |
| BETA | 120, | 36, | 36, | 56, | 36, | 36, | 124, | 0 |
| GAMMA | 126, | 34, | 32, | 32, | 32, | 32, | 112 | 0 |
| DELTA | 8, | 8, | 20, | 20, | 34, | 34, | 127, | 0 |
| EPSILON | 126, | 34, | 40, | 56, | 40, | 34, | 126, | 0 |
| ZETA | 126, | 70, | 12, | 24, | 48, | 96, | 126, | 0 |
| ETA | 119, | 34, | 34, | 62, | 34, | 34, | 119, | 0 |
| THETA | 62, | 65, | 85, | 93, | 85, | 65, | 62, | 0 |
| IOTA | 56, | 16, | 16, | 16, | 16, | 16, | 56, | 0 |
| KAPPA | 110, | 36, | 40, | 48, | 40, | 36, | 118, | 0 |
| LAMBDA | 8, | 8, | 20, | 20, | 34, | 34, | 119, | 0 |
| MU | 99, | 54, | 62, | 42, | 42, | 34, | 119, | 0 |
| NU | 103, | 34, | 50, | 42, | 38, | 34, | 115, | 0 |
| XI | 66, | 126, | 0, | 126, | 0, | 126, | 66, | 0 |
| OMICRON | 56, | 68, | 68, | 68, | 68, | 68, | 56, | 0 |
| PI | 119, | 34, | 34, | 34, | 34, | 34, | 119, | 0 |
| RHO | 120, | 36, | 36, | 56, | 32, | 32, | 112, | 0 |
| SIGMA | 126, | 34, | 16, | 8, | 16, | 34, | 126, | 0 |
| TAU | 127, | 73, | 8, | 8, | 8, | 8, | 28, | 0 |
| UPSILON | 99, | 34, | 20, | 8, | 8, | 8, | 28, | 0 |
| PHI | 28, | 8, | 62, | 42, | 62, | 8, | 28, | 0 |
| CHI | 119, | 34, | 20, | 8, | 20, | 34, | 119, | 0 |
| PSI | 28, | 73, | 107, | 42, | 62, | 8, | 28, | 0 |
| OMEGA | 28, | 34, | 65, | 65, | 54, | 85, | 119, | 0 |



MISCELLANEOUS MATHEMATICAL SYMBOLS

Powers and Superscripts

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| squared | 112, | 16, | 112, | 64, | 112, | 0, | 0, | 0 |
| cubed | 112, | 16, | 112, | 16, | 112, | 0, | 0, | 0 |
| 4 | 64, | 80, | 120, | 16, | 16, | 0, | 0, | 0 |
| 5 | 112, | 64, | 112, | 16, | 112, | 0, | 0, | 0 |
| −1 | 24, | 8, | 104, | 8, | 8, | 0, | 0, | 0 |
| −2 | 14, | 2, | 110, | 8, | 14, | 0, | 0, | 0 |
| −3 | 14, | 2, | 110, | 2, | 14, | 0, | 0, | 0 |

One character square will take two of these small numbers across, and small subscripts can be defined by moving the symbol to the bottom of the square instead of the top.

Other symbols useful in maths programs

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ½ | 66, | 68, | 72, | 87, | 33, | 71, | 132, | 7 |
| dx | 0, | 0, | 16, | 16, | 117, | 82, | 117, | 0 |
| $\partial$ | 96, | 16, | 8, | 60, | 68, | 68, | 56, | 0 |
| $\nabla$ | 0, | 127, | 34, | 34, | 20, | 20, | 8, | 8 |
| $\int$ | 0, | 0, | 0, | 0, | 0, | 0, | 12, | 18 |
| | 16, | 16, | 16, | 16, | 8, | 8, | 8, | 8 |
| | 72, | 48, | 0, | 0, | 0, | 0, | 0, | 0 |

# APPENDIX 2

## Hexadecimal converter

Earlier I was explaining about numbers to different bases, and how, if you want to understand machine code, numbers to the base 16 are important. I'm including a small program here that takes all the sting out of those conversions. You just select the direction you want, HEX to DEC, or else DEC to HEX, and go ahead.

*Appendix Program 3  HEXDEC*

```
   5 REM PROGRAM APPENDIX 3
     HEXDEC
  10 POKE 23658,8: CLS : BORDER
5: INPUT "ENTER 1 for DEC to HEX
                2 for HEX to DEC
: ";a: IF a=2 THEN  GO TO 70
  20 LET A$="0000": LET X=4: INP
UT "ENTER DECIMAL NO.(<=65535) "
;DEC
  30 LET N=INT (DEC/16): LET Q=D
EC-16*N: LET A$(X)=CHR$ (Q+48+7*
(Q>9))
  40 LET DEC=N: LET X=X-1: IF DE
C>0 THEN  GO TO 30
  50 PRINT AT 10,5;"HEX NUMBER I
S ";A$
  60 PRINT AT 21,3;"PRESS ANY KE
Y TO RUN AGAIN": PAUSE 0: GO TO
10
```



```
  70 INPUT "ENTER HEX NO. (4 DIG
IT) ";A$
  80 LET DEC=0: FOR X=1 TO 4
  90 LET DEC=DEC*16+CODE A$(X)-4
8-7*(A$(X)>"9"): NEXT X
 100 PRINT AT 10,5;"DECIMAL NUMB
ER IS ";DEC: GO TO 60
```

The POKE in line 10 is a harmless device that sets a flag inside the machine so that it goes automatically into capital letter mode. Otherwise the program will screw up.

# APPENDIX 3

## A series for PI

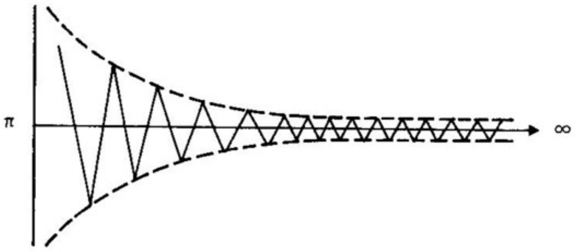
There exists an infinite series whose sum is PI, and which, once we know about it, will allow us to calculate PI to as many places as we please. It goes like this:

PI = 4(1/1 − 1/3 + 1/5 − 1/7 + 1/9 − 1/11 + 1/13 − . . .)

Notice that it's basically the series of odd fractions, alternating positive and negative, and the whole show multiplied by 4.
Try this:

*Appendix Program 4  PI SERIES*

```
    5 REM PROGRAM APPENDIX 4
      PI SERIES
   10 PRINT AT 1,0;PI: LET s=0: F
OR n=0 TO 10000: LET k=1
   20 IF n/2<>INT (n/2) THEN  LET
 k=-1
   30 LET q=k/((2*n)+1)
   40 LET s=s+q: PRINT AT 0,0;s*4
   50 NEXT n
```





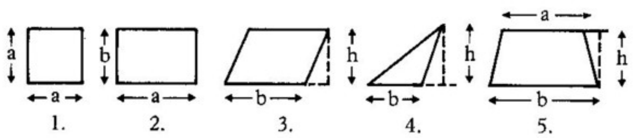

Don't be alarmed if the number oscillates for several minutes. It is trying to converge to PI, but even with ten thousand terms this series is so slow to converge that it takes an age for the oscillations to die out in the fourth decimal place! There are much faster converging series in existence, but I thought you might like to see this one in action.

# APPENDIX 4

## Geometric shapes

*Plane figures*

1 Square: Area $= a^2$ (where a is the length of a side).
2 Rectangle: Area $= a*b$ (where a and b are sides).
3 Parallelogram: Area $= b*h$ (where b is the side and h is the vertical height).
4 Triangle: Area $= \frac{1}{2}*b*h$ (where b = base and h = height).
5 Trapezium: Area $= \frac{1}{2}*(a+b)*h$ (where a and b are the lengths of the parallel sides and h is the height).




6 Circle: Area $= PI*r^2 = PI*d^2/4$
Circumference $= 2*PI*r = PI*d$ (where r is the radius and d is the diameter).
7 Annulus (ring): Area $= (PI/4)*(D*D - d*d)$ (where D is the diameter of the larger circle and d that of the smaller).
8 Ellipse: Area $= PI*D*d/4$ (where D is the major axis length and d is the minor axis).

*Solids*

1 Sphere: Volume $= (PI*d\uparrow 3)/6 = \frac{4}{3}*PI*r\uparrow 3$.
Surface area $= PI*d\uparrow 2 = 4*PI*r\uparrow 2$.
2 Right cylinder: Volume = area of base*vertical height.
Surface area = perimeter of base*height + 2*Area of base.



3 Oblique cylinder: Volume = area of base* perpendicular height.
Surface area = perimeter of perpendicular slice*length of side.
4 Right cone: Volume = area of base*⅓(perpendicular height).
Surface area = perimeter of base*½ sloping length + area of base.
5 Oblique cone: Volume = area of base*⅓(perpendicular height).

# APPENDIX 5

## Defined functions

There's a useful facility on the Spectrum that I don't suppose you've used yet, and I've hardly mentioned it either. The reason is that it wasn't necessary to any of our programs, and I was trying to keep things as straightforward as possible. Nevertheless, the facility is there, and you ought to be aware of how to use it. It concerns keys 1 and 2; the red legends underneath them read DEF FN and FN respectively.

Maybe we have a good long equation like

```
y=LN((1+x)/(1+(3*x)))+SIN (x/2)
```

We could write that generally as

```
y=f(x)
```

which means 'y equals a function of x', and we can call that function up on our computer so long as it knows it. We tell it what f(x) is by using DEF FN. And we do it like this:

```
10 DEF FN A(x)=LN((1+x)/(1+(3*x)))+SIN (x/2)
```

Notice that it is just like a LET statement except that you can define the function in terms of its variable (or variables) with a letter and another letter enclosed in brackets. In the above example, we have defined the function A(x), read as 'A of x'.

The next step is to use FN to call up the function you have defined, whenever you need it in a program, like this:

```
100 FN A(2)
```

It then takes the so-called 'dummy variable' x, and pops whichever value you've specified in your FN function into the expression where the x appears. Easy!



But it doesn't stop there, because if it did, you might as well use a regular LET statement. The good thing about the DEF FN statement is that you can define a function of not just one, but many independent variables. To keep it simple, let's look at the three variables x, y and z and a function of them called A(x,y,z). Then you have

```
10 DEF FN A(x,y,z)=x+y+z
```

and

```
10 LET Y=FN A(1,2,3)
```

which will return the function you have defined with the variables x getting 1, y getting 2, and z getting 3. In other words, you get 1+2+3=6.

So just make it do that for you by making line 100 into

```
100 PRINT FN A(1,2,3)
```

or

```
110 PRINT Y
```

and you'll get 6.

Or why not try:

```
10 DEF FN A(x,y,z)=SQR(x↑2+y↑2+z↑2)
20 PRINT FN A(3,4,5)
```

and you will get 7.0710678.

Impressive? Yes. But that's not all. You can actually use it with string variables, too. String variables handle strings of characters such as words, and you can define functions which act like the commands LEFT$ and RIGHT$ and MID$ which are standard in the BASICs used by other machines like the BBC B, etc. If you'd like to know more about this, look at your orange Users' Manual – the one that came with your Spectrum.

This book provides an introduction to mathematics for the non-mathematical person who has access to the Sinclair Spectrum. With the help of over 100 short programs, the book shows the ability of the micro to perform fast calculations of a laborious kind and to draw illustrative and animated graphs and diagrams. The style of the book is very informal and refreshingly different to the dry textbook/ revision booklet approach.

The book will appeal, in addition to people of limited mathematical ability (or people whose maths are rusty), to 'O' and 'A' Level pupils and to teachers wishing to illustrate mathematical principles and techniques.

ISBN 0-7126-0606-8

£7.95