

CHAPTER 4

Powers

The square of a number X is the number multiplied by itself, that is $X \times X$. It is denoted by X^2 or $X \uparrow 2$. The square of a number is also called the second power of X .

The powers of X are products of the appropriate number of X 's. The first, second, third, fourth and fifth powers of the number 2 are the numbers

2, 4, 8, 16 and 32

We denote these powers respectively by

$2^1, 2^2, 2^3, 2^4$, and 2^5

You can calculate the powers of 2 (or any other number) with the following simple program.

Listing 4.1

LIST

```
10 REM Powers
20 MODE 1:COLOUR 3:PRINT ' TAB(17);"P
owers":@%=10
30 PRINT "This program prints out the
powers of a number entered."
40 COLOUR 2:INPUT "Number ";X:PRINT
50 REPEAT
60 INPUT "How many powers ";N
70 IF N<1 OR N<>INT(N) THEN COLOUR 3:
PRINT '"An integer please.':COLOUR 2
80 UNTIL N>=1 AND N=INT(N)
90 REM The calculation and print out
100 COLOUR 1:PRINT ' "          I
X^I "'
110 Y=1
```

Essential Maths on the BBC and Electron Computers

```
120 FOR I=1 TO N:Y=Y*X:PRINT I " " Y:N
EXT
130 COLOUR 3:PRINT CHR$(7) ' TAB(10);"
Another go? Y or N ";
140 REPEAT:G$=GET$:UNTIL G$="Y" OR G$=
"N"
150 IF G$="Y" THEN RUN
160 CLS:PRINT '"Bye for now.":END
```

RUN

Powers

This program prints out the powers of a number entered.

Number ?3

How many powers ?10

I	X^I
1	3
2	9
3	27
4	81
5	243
6	729
7	2187
8	6561
9	19683
10	59049

Another go? Y or N

If the number N is too large you'll get a 'too big at line 120' error message.

You can add a few lines to test for a possible overflow and escape.

Listing 4.2

```
10 REM Powers Plus
20 MODE 1:COLOUR 3:PRINT ' TAB(17);"P
owers":@%=10
30 PRINT "This program prints out the
powers of a number entered."'
40 COLOUR 2:INPUT "Number ";X:PRINT
50 REPEAT
60 INPUT "How many powers ";N
70 IF N<1 OR N<>INT(N) THEN COLOUR 3:
PRINT "'An integer please.":COLOUR 2
80 UNTIL N>=1 AND N=INT(N)
85 IF (N+1)*LOG(ABS(X))>126*LOG(2) TH
EN PRINT "'You'll probably overflow!'"
90 REM The calculation and print out
100 COLOUR 1:PRINT ' " I
X^I "'
110 Y=1
120 FOR I=1 TO N:Y=Y*X:PRINT I " " Y:N
EXT
130 COLOUR 3:PRINT CHR$(7) ' TAB(10);"
Another go? Y or N ";
140 REPEAT:G$=GET$:UNTIL G$="Y" OR G$=
"N"
150 IF G$="Y" THEN RUN
160 CLS:PRINT "'Bye for now.":END
```

RUN

Powers

This program prints out the powers of a number entered.

Essential Maths on the BBC and Electron Computers

Number ?6

How many powers ?10

I	X^I
1	6
2	36
3	216
4	1296
5	7776
6	46656
7	279936
8	1679616
9	10077696
10	60466176

Another go? Y or N

Your computer can also calculate the power N of a number X by using PRINT X↑N.

Listing 4.3

LIST

```
10 REM Powers again
20 MODE 1:COLOUR 3:PRINT ' TAB(14);"P
owers again":@%=10
30 PRINT "This program prints out the
powers of a number entered."
40 COLOUR 2:INPUT "Number ";X:PRINT
50 REPEAT
60 INPUT "How many powers ";N
70 IF N<1 OR N<>INT(N) THEN COLOUR 3:
PRINT '"An integer please.':COLOUR 2
80 UNTIL N>=1 AND N=INT(N)
90 IF (N+1)*LOG(ABS(X))>126*LOG(2) TH
EN PRINT '"You'll probably overflow!'"
```

```
100 REM The calculation and print out
110 COLOUR 1:PRINT ' "      I
X^I "'
120 FOR I=1 TO N:Y=Y*X:PRINT I " " X^I
:NEXT
130 COLOUR 3:PRINT CHR$(7) ' TAB(10);"
Another go? Y or N ";
140 REPEAT:G$=GET$:UNTIL G$="Y" OR G$=
"N"
150 IF G$="Y" THEN RUN
160 CLS:PRINT '"Bye for now.":END
```

RUN

Powers

This program prints out the powers of a number entered.

Number ?987654

How many powers ?12

You'll probably overflow!

I	X^I
1	987654
2	9.754604237E11
3	9.634173894E17
4	9.515230383E23
5	9.397755348E29
6	9.28173066E35
7	

Too big at line 120

As you may know or can soon discover, N does not have to be a whole

number.

But what does $2^{\uparrow 1.7}$ mean? It is reasonable that it should be a number between $2^{\uparrow 1}$ and $2^{\uparrow 2}$, that is, between 2 and 4. Indeed, the value of $2^{\uparrow 1.7}$ is about 3.24900959. An approximate way of finding the value of $2^{\uparrow 1.7}$ (without using your computer) is to use some graph paper. Plot the powers of 2 from the first power to the fifth power — see **Figure 15**. Next draw a smooth curve through these points — see **Figure 16**. The approximate value of $2^{\uparrow 1.7}$ can be read from this graph — see **Figure 17**.

Similar graphs could be drawn for powers of other numbers. Of course, you don't need to do this as your computer will give the answer immediately.

Negative powers of numbers also make sense: these are simply defined by the following rule:

$$X^{-N} = 1/X^N.$$

Thus $2^{-2} = 1/2^2$ which is $1/4$ or 0.25 . By convention, the zero-th power of a number is 1.

Powers of a number behave in a nice way according to the following rule:

$$X^M * X^N = X^{M+N}.$$

So that, for instance,

$$3^4 * 3^2 = 3^6$$

and

$$10^{-2} * 10^1 = 10^{-1}.$$

Square roots

For a number X the number $X^{1/2}$ has a special name — it is called the *square root* of X . The square root of a number is that number whose square is the original number. Thus the square root of 9 is 3 and the square root of 2 is approximately 1.41421356 as you can readily check by multiplying this number with itself.

The notation $X^{1/2}$ for the square root of X fits into the way that multiplication of powers works:

$$X^{1/2} * X^{1/2} = X^1$$

which is X of course.

The square root of a number X can also be found with your computer by using `PRINT SQR(X)`.

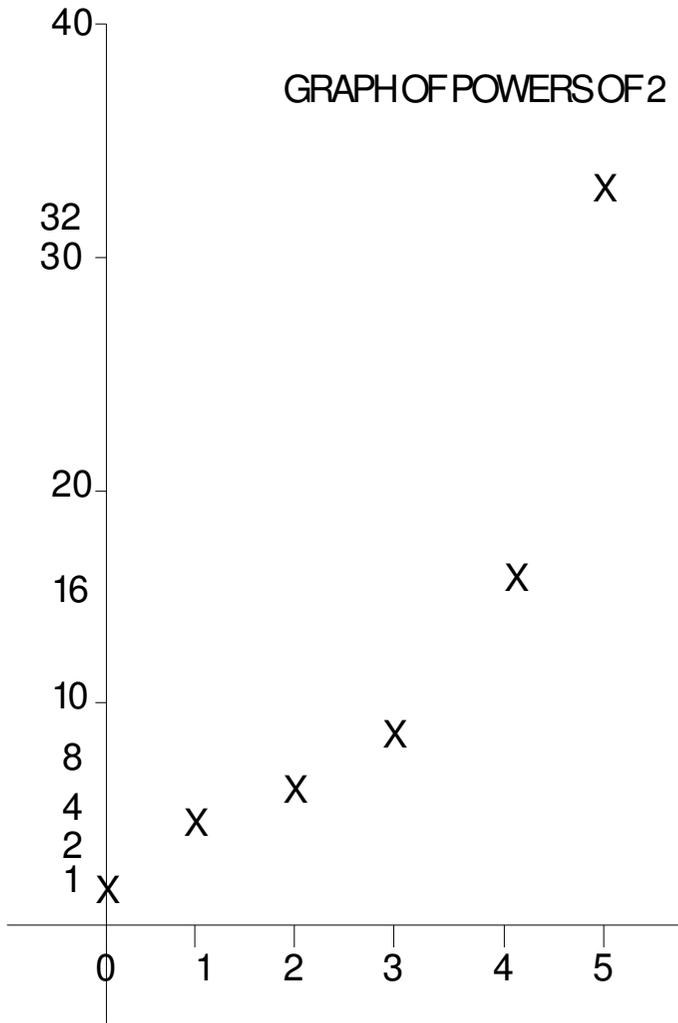


Figure 15

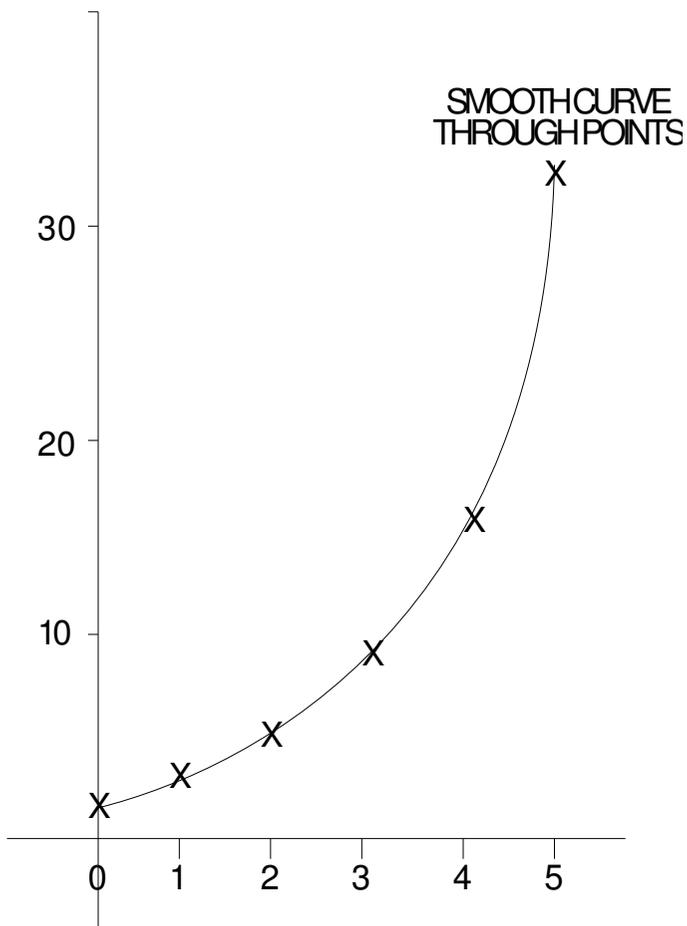


Figure 16

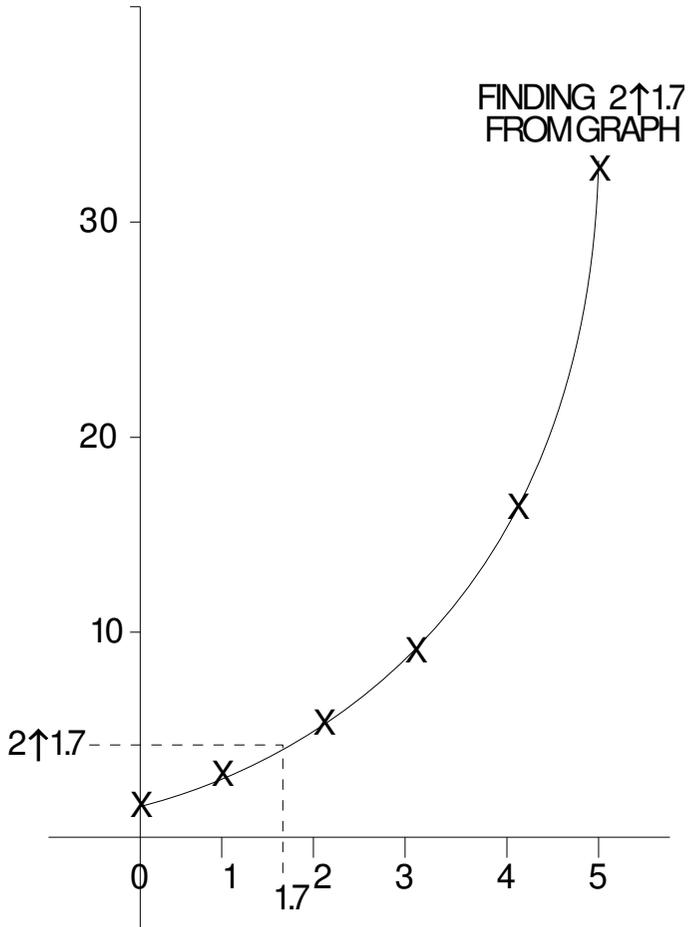


Figure 17

Imaginary numbers?

The square of a number is always positive so you should not expect to be able to find the square root of a negative number. Indeed if you ask your computer to PRINT SQR(-1) it will respond with a ' -veroot' error message.

You may have heard of imaginary numbers and complex numbers which are associated with square roots of negative numbers. Mathematicians are never deterred by seemingly impossible things such as the square root of -1. One simply creates a new symbol to stand for this number. Thus we let I stand for the square root of -1. There is nothing strange about this: I exists in the same way as negative numbers exist.

We can add the number I to itself, to other numbers and multiply it by other numbers. We can therefore form numbers such as

$$2 + 3*I, 1.41412*I, 9 - I, 10 - 8*I$$

Numbers involving I are called complex number, usually to distinguish them from ordinary or real numbers. Any complex number may be written in the form

$$X + Y*I$$

for some real numbers X and Y. We call X the real part and Y the imaginary part of the complex number.

Once we decide to use the symbol I for SQR(-1) we can find square roots of other negative numbers.

$$\begin{aligned} \text{SQR}(X) &= \text{SQR}(\text{ABS}(X)*\text{SGN}(X)) \\ &= \text{SQR}(\text{ABS}(X)) * \text{SQR}(\text{SGN}(X)) \end{aligned}$$

Thus, for instance,

$$\begin{aligned} \text{SQR}(-9) &= \text{SQR}(9) * \text{SQR}(-1) \\ &= 3*I \end{aligned}$$

But, of course, the BBC and Electron micros will not operate this way because they don't know about complex numbers.

The next program shows how you could incorporate complex numbers into your computer. Simple arithmetic operations may then be performed. Note that multiplication of complex numbers follows the following rule:

$$\begin{aligned} (A + B*I) * (C + D*I) &= A*C + A*D*I + B*I*C + B*I*D*I \\ &= A*C + A*D*I + B*C*I + B*D*I*I \\ &= A*C + A*D*I + B*C*I + B*D*-1 \\ &= A*C - B*D + (A*D + B*C)*I \end{aligned}$$

Listing 4.4**LIST**

```

10 REM Complex numbers
20 MODE 1:COLOUR 3:PRINT ' TAB(12);"C
omplex numbers":@%=10
30 PRINT "This program performs multi
plication"
40 PRINT "and division of complex num
bers." '
50 PRINT "Enter complex numbers as re
quested." '
60 COLOUR 2:PRINT "First number" '
70 INPUT "      Real part ";A
80 INPUT "Imaginary part ";B
90 PRINT '"Second number" '
100 INPUT "      Real part ";C
110 INPUT "Imaginary part ";D
120 COLOUR 1:PRINT '"What do you want
do do?" '
130 PRINT "1 Multiply the two numbers.
"
140 PRINT "2 Divide the first number b
y the second "
150 REPEAT
160   INPUT "Enter 1 or 2 ";N:PRINT
170   UNTIL N=1 OR N=2
180   ON N GOSUB 300,350
190   COLOUR 3:PRINT ' CHR$(7) TAB(10);"
Another go? Y or N ";
200   REPEAT:G$=GET$:UNTIL G$="Y" OR G$=
"N"
210   IF G$="Y" THEN RUN
220   CLS:PRINT '"Bye for now.":END
300 REM Multiplication
310 COLOUR 2:PRINT '"The product of th
e numbers:"
320 PRINT "      Real part ";A*C-B*D
330 PRINT "Imaginary part ";A*D+B*C

```

Essential Maths on the BBC and Electron Computers

```
340 RETURN
350 REM Division
360 COLOUR 2:X=C*C+D*D:IF X=0 THEN PRI
NT "Division not possible.":RETURN
370 PRINT "The division of the first b
y second:"
380 PRINT "      Real part ";(A*C+B*D)/
X
390 PRINT "Imaginary part ";(-A*D+B*C)
/X
400 RETURN
```

RUN

Complex numbers

This program performs multiplication
and division of complex numbers.

Enter complex numbers as requested.

First number

```
      Real part ?3
Imaginary part ?2
```

Second number

```
      Real part ?2
Imaginary part ?-3
```

What do you want do do?

- 1 Multiply the two numbers.
- 2 Divide the first number by the second

Enter 1 or 2 ?1

The product of the numbers:

Real part 12

Imaginary part -5

Another go? Y or N

Quadratic equations

Quadratic equations often arise when solving problems of one sort or another. The general form of a quadratic equation is

$$A*X^2 + B*X + C = 0$$

where A, B and C are known numbers with A non-zero. The problem is to find those numbers X that satisfy the equation. These are called the *roots* of the quadratic equation. For example the values $X = 1$ and $X = 2$ satisfy the following quadratic equation

$$X^2 - 3*X + 2 = 0$$

as you can readily verify.

Usually there are two roots to a quadratic equation; often the roots are complex numbers.

There is a very straightforward formula which provides the roots of a quadratic equation:

$$\frac{-B + \text{SQR}(B^2 - 4*A*C)}{2*A}$$

and

$$\frac{-B - \text{SQR}(B^2 - 4*A*C)}{2*A}$$

The key to the nature of the roots is found in that part of the formula involving the square root function,

$$\text{SQR}(B^2 - 4*A*C)$$

which is called the *discriminant*.

If $B^2 - 4*A*C > 0$ then there are two real roots.

Essential Maths on the BBC and Electron Computers

If $B^2 - 4*A*C = 0$ then the two roots are real and the same.

If $B^2 - 4*A*C < 0$ then there are two complex roots.

The next problem calculates the roots of a quadratic equation.

Listing 4.5

LIST

```
10 REM Quadratic equations
20 MODE 1:COLOUR 3:PRINT ' TAB(10);"Q
uadratic equations":@%=10
30 PRINT "This program solves quadrat
ic equations"
40 PRINT "like A*X*X + B*X + C = 0.'"
:COLOUR 2
50 REPEAT
60 INPUT "Value of A ";A
70 IF A=0 THEN COLOUR 3:PRINT "A can
not be zero! Try again.":COLOUR 2
80 UNTIL A<>0
90 INPUT "Value of B ";B
100 INPUT "Value of C ";C
110 REM Start of calculation
120 D=B*B-4*A*C
130 COLOUR 1:ON SGN(D)+2 GOSUB 200,300
,400
140 COLOUR 3:PRINT ' CHR$(7) TAB(10);"
Another go? Y or N ";
150 REPEAT:G$=GET$:UNTIL G$="Y" OR G$=
"N"
160 IF G$="Y" THEN RUN
170 CLS:PRINT '"Bye for now.":END
200 REM Two complex roots
210 PRINT '"There are two complex root
s.'"
220 X=-B/A/2:Y=ABS(SQR(-D)/A/2)
230 PRINT X;" + ";Y;"*I"
240 PRINT X;" - ";Y;"*I"
```

```
250 RETURN
300 REM Two equal roots
310 PRINT "The two roots are equal."
320 PRINT -B/A/2
330 RETURN
400 REM Two real roots
410 PRINT "There are two real roots."
420 D=SQR(D):X=(-B-D)/2:IF B<0 THEN X=
(-B+D)/2
430 PRINT X/A:PRINT C/X
440 RETURN
```

RUN

Quadratic equations

This program solves quadratic equations
like $A*X*X + B*X + C = 0$.

Value of A ?1

Value of B ?2

Value of C ?3

There are two complex roots.

-1 + 1.414213562*I

-1 - 1.414213562*I

Another go? Y or N

If there are two real roots, then the program uses the quadratic formula to find one of the roots — the one with the largest absolute value. The other root is found by using the fact that the product of the two roots is C/A .

Solving other equations

Solving quadratic equations is relatively straightforward. This is not so for other equations such as

$$9X^5 - 3X^4 + X^3 - X^2 + 5X - 4 = 0$$

If the equation only involves non-negative powers of X, as in the example above, we call the equation a *polynomial* equation. The largest non-zero power of X that occurs is called the degree of the polynomial. In the above equation the degree is 5. The quadratic equation is a polynomial equation of degree 2. Quadratic equations usually have 2 roots; a polynomial equation of degree N usually has N roots.

Apart from some special cases there are no general formulae for solving polynomial equations. Indeed this lack has led to some very interesting mathematics, but that's another story and will not be covered here.

We can, however, use our computer to work out roots of polynomial equations by a repeated guess-and-try process. Essentially the computer tries many different numbers to find out which satisfies the equation.

The next program provides a method of finding a real root of a polynomial equation. It is mathematically crude and occasionally will not find a root, even if there is one. The program does however illustrate the general method involved.

Listing 4.6

LIST

```
10 REM Poly Roots
20 MODE 1:COLOUR 3:PRINT ' TAB(10);"R
oots of polynomials":@%=10
30 PRINT "This program attempts to fi
nd roots of"
40 PRINT "polynomials such as " '
50 PRINT "A*X^N + B*X^(N-1) + ... + C
*X + D = 0.    ":COLOUR 2
60 REPEAT
70 INPUT "Degree of polynomial ";N
80 IF N<2 OR N<>INT(N) THEN COLOUR 3
:PRINT "An integer >= 2 please.":COLOUR
2
90 UNTIL N>1 AND N=INT(N)
```

```

100 DIM A(N):PRINT "Enter coefficient
s term by term."
110 FOR I=0 TO N
120 REPEAT
130 IF I<N THEN PRINT " Coefficient
of X^";N-I; ELSE PRINT "Constant coeffi
cient";
140 INPUT " ";A(I)
150 IF A(0)=0 THEN COLOUR 3:PRINT "N
ot zero please.":COLOUR 2
160 UNTIL A(0)<>0
170 NEXT
180 PRINT "Enter range over which sea
rch of roots is to be attempted."
190 REPEAT
200 INPUT "Lower value ";A
210 INPUT "Upper value ";B:PRINT
220 IF A>=B THEN CLOUR 3:PRINT "First
value should be lower.":COLOUR 2
230 UNTIL A<B
240 REM The search
250 S=B-A:T=0:TEST=-1:D=1E-9
260 REPEAT
270 COLOUR 3:PRINT "Test run ";T+1;"
** ";10^T;" divisions **":PROCStep
280 IF TEST THEN S=S/10:T=T+1
290 UNTIL T=4 OR NOT TEST
300 IF T=4 THEN PRINT "Sorry - cannot
locate roots."
310 COLOUR 3:PRINT CHR$(7) TAB(10);"
Another go? Y or N ";
320 REPEAT:G$=GET$:UNTIL G$="Y" OR G$=
"N"
330 IF G$="Y" THEN RUN
340 CLS:PRINT "Bye for now.":END
400 REM Step by step search
410 DEF PROCStep
420 X1=A:Y1=FNPPoly(A)

```

Essential Maths on the BBC and Electron Computers

```
430 FOR X=A+S TO B STEP S
440 X2=X:Y2=FNPoly(X)
450 IF Y1*Y2<=D THEN PROCfine
460 Y1=Y2:X1=X
470 NEXT
480 ENDPROC
490 DEF FNPoly(X)
500 LOCAL I,Y
510 Y=A(0):FOR I=1 TO N:Y=Y*X+A(I):NEX
T
520 =Y
600 DEF PROCfine
610 PRINT "Fine tuning solution.":B$="
"
620 IF ABS(Y1)<D THEN X=X1:TEST=0
630 IF TEST AND ABS(Y2)<D THEN X=X2:TE
ST=0
640 REPEAT
650 IF TEST THEN Z=X:X=(X1+X2)/2:Y=FN
Poly(X):IF ABS(Y)<D THEN TEST=0
660 IF TEST AND ABS(Z-X)<D THEN B$="
probably ":TEST=0
670 IF Y*Y2>0 THEN X2=X ELSE X1=X
680 UNTIL NOT TEST
690 COLOUR 1:PRINT '"There is";B$;"a r
oot at ";X:X=B
700 IF B$<>" " THEN PRINT '"Value of p
olynomial is ";FNPoly(Z)
710 ENDPROC
```

RUN

Roots of polynomials

This program attempts to find roots of polynomials such as

$A * X^N + B * X^{(N-1)} + \dots + C * X + D = 0.$

Degree of polynomial ?3

Enter coefficients term by term.

Coefficient of X^3 ?1

Coefficient of X^2 ?0

Coefficient of X^1 ?0

Constant coefficient ?-8

Enter range over which search of roots
is to be attempted.

Lower value ?-10

Upper value ?10

Test run 1 ** 1 divisions **

Fine tuning solution.

There is a root at 2

Another go? Y or N

Occasionally the program will say that there is ' probably a root at some number. On such an occasion the value of the polynomial at that number is printed. The closer this value is to zero, the closer you are to the root.

Newton's method

The method of finding roots of equations given in the previous section can be improved by using the so-called Newton' s method.

Suppose we want to find a root of the polynomial equation

$$9 * X^5 - 3 * X^4 + X^3 - X^2 + 5 * X - 4 = 0$$

Essential Maths on the BBC and Electron Computers

Denote the polynomial function by $P(X)$ and let $P'(X)$ be the following polynomial.

$$5*9*X^4 - 4*3*X^3 + 3*X^2 - 2*X + 5.$$

In fact $P'(X)$ is the *derivative* of $P(X)$, but this need not concern us. Now, if Y is approximately a root of the equation $P(X) = 0$ then the following

$$Y - P(Y)/P'(Y)$$

is usually a better approximation provided that $P'(Y)$ is non-zero

The next program uses this technique to find roots of polynomial equations.

Listing 4.7

LIST

```
10 REM Poly roots via Newton
20 MODE 1:COLOUR 3:PRINT '"Roots of p
polynomials by Newton:s method. "':@%=10
30 PRINT "This program attempts to fi
nd roots of"
40 PRINT "polynomials such as"
50 PRINT "A*X^N + B*X^(N-1) + ... + C
*X + D = 0. "':COLOUR 2
60 REPEAT
70 INPUT "Degree of polynomial ";N
80 IF N<2 OR N<>INT(N) THEN COLOUR 3
:PRINT "An integer >= 2 please."':COLOUR
2
90 UNTIL N>1 AND N=INT(N)
100 DIM A(N):PRINT '"Enter coefficient
s term by term."'
110 FOR I=0 TO N
120 REPEAT
130 IF I<N THEN PRINT " Coefficient
of X^";N-I; ELSE PRINT "Constant coeffi
cient";
140 INPUT " ";A(I)
150 IF A(0)=0 THEN COLOUR 3:PRINT "N
```

```

ot zero please.":COLOUR 2
  160 UNTIL A(0)<>0
  170 NEXT
  180 PRINT "Enter a guess value for a
root."
  190 INPUT "Guess value ";X
  200 REM Next bit calculates P'(X)
  210 DIM B(N):FOR I=0 TO N:B(I)=(N-I)*A
(I):NEXT
  220 REM Start finding root
  230 J=1 : REM counter
  240 D=1E-9 : REM Accuracy
  250 TEST=-1 : REM TEST=-1 until a root
is found or nearly found
  260 B$=" ":Y=FNPoly(X):Y1=FNPolyPrime(
X)
  270 IF ABS(Y)<D THEN TEST=0
  280 IF TEST:REPEAT:PROCFind:UNTIL NOT
TEST
  290 IF TEST=0 THEN COLOUR 1:PRINT '"Th
ere is ";B$;"a root at ";X
  300 IF TEST=0 AND B$<>" " THEN PRINT '
"Value of polynomial is ";FNPoly(Z)
  310 COLOUR 3:PRINT CHR$(7) ' TAB(10);"
Another go? Y or N ";
  320 REPEAT:G$=GET$:UNTIL G$="Y" OR G$=
"N"
  330 IF G$="Y" THEN RUN
  340 CLS:PRINT '"Bye for now.":END
  400 DEF FNPoly(X)
  410 LOCAL I,Y
  420 Y=A(0):FOR I=1 TO N:Y=Y*X+A(I):NEX
T
  430 =Y
  440 DEF FNPolyPrime(X)
  450 LOCAL I,Y
  460 Y=B(0):FOR I=1 TO N-1:Y=Y*X+B(I):N
EXT

```

Essential Maths on the BBC and Electron Computers

```
470 =Y
500 DEF PROCFind
510 IF Y1=0 THEN COLOUR 3:PRINT '"Division by zero - sorry - make another guess."' :TEST=1
520 IF TEST:Z=X:X=X-Y/Y1:J=J+1:Y=FNPoly(X):Y1=FNPolyPrime(X)
530 IF TEST AND ABS(Y)<D THEN TEST=0
540 IF TEST AND ABS(Z-X)<D THEN B$=" probably " :TEST=0
550 IF J>1000 THEN COLOUR 3:PRINT '"Sorry, can't find a root."' :TEST=1
560 ENDPROC
```

RUN

Roots of polynomials by Newton's method.

This program attempts to find roots of polynomials such as

$$A \cdot X^N + B \cdot X^{(N-1)} + \dots + C \cdot X + D = 0.$$

Degree of polynomial ?4

Enter coefficients term by term.

Coefficient of X³ ?-4

Coefficient of X² ?6

Coefficient of X¹ ?-4

Constant coefficient ?1

Enter a guess value for a root.

Guess value ?.5

There is a root at 0.994920963

Another go? Y or N

Exponential function

Functions like 2^X , 10^X are called *power functions*, because the variable X occurs as a power. Power functions are accessed on your computer by using PRINT2LX, etc. There is one important power function that the computer singles out; the exponential function EXP(X).

The exponential function is based on powers of the number E which has the value 2.71828183 approximately. Do not confuse this E with the E that appears when numbers are printed using scientific notation. The number E itself is defined by:

$$E = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \frac{1}{6!} + \dots$$

where \dots means that the sum carries on forever, and the symbol $!$ stands for factorial which is defined by

$$N! = N * (N - 1) * (N - 2) * \dots * 2 * 1$$

that is, the product of the integers from 1 to N . For instance, $4!$ is $4*3*2*1$ which is 24.

The exponential function is defined by

$$\begin{aligned} \text{EXP}(X) &= E^X \\ &= E \uparrow X \end{aligned}$$

So that, in particular, $E = \text{EXP}(1)$. Try typing the following on your computer.

```
E = EXP(1)
PRINT E↑,EXP(5)
```

Because of the properties of power functions we have the following properties of the exponential function.

$$\begin{aligned} \text{EXP}(X) * \text{EXP}(Y) &= \text{EXP}(X + Y) \\ \text{EXP}(X) / \text{EXP}(Y) &= \text{EXP}(X - Y) \end{aligned}$$

There is a very straightforward formula which may be used to calculate EXP(X) for any number X. This is:

$$\text{EXP}(X) = 1 + \frac{X}{1!} + \frac{X^2}{2!} + \frac{X^3}{3!} + \frac{X^4}{4!} + \frac{X^5}{5!} + \frac{X^6}{6!} + \dots$$

Logarithmic function

What number X satisfies EXP(X) = 3? Since EXP(1) = 2.71828183 we see that X is just over 1. In fact the answer is 1.09861229 approximately.

The number X that satisfies EXP(X) = N is called the (natural) logarithm of N. It is denoted by LN(N).

The logarithmic function has the following properties.

$$\text{LN}(X*Y) = \text{LN}(X) + \text{LN}(Y)$$

$$\text{LN}(X/Y) = \text{LN}(X) - \text{LN}(Y)$$

$$\text{LN}(X^N) = N*\text{LN}(X)$$

It is because of these properties that logarithms are important when performing multiplication, division, etc., without computers.

The following relations hold between the exponential and logarithmic function.

$$\text{EXP}(\text{LN}(N)) = N$$

$$\text{LN}(\text{EXP}(N)) = N$$

One simple use of the logarithmic function is testing of large numbers. For instance, the number X satisfies the relation

$$X < 10^{\uparrow N}$$

if and only if the following relation is satisfied

$$\text{LN}(X) < N*\text{LN}(10)$$

Such a reformation is useful because the number $10^{\uparrow N}$ itself may cause an 'Exp range' error.

A related function is LOG(X), the common logarithm of X. The function EXP(X) is the logarithm of X to base E, while LOG(X) is the logarithm of X to base 10. The relationship between LOG(X) and powers of 10 are shown below.

$$\text{LOG}(10^{\uparrow N}) = N$$

$$10^{\uparrow N}(\text{LOG}(N)) = N$$

Roots of other functions

Given a function such as $X \cdot \text{EXP}(X) + 1$, a root of the function is a number which, when substituted into the function, gives 0.

Two programs for finding the roots of polynomials were given earlier on. The first of these can be adapted for finding roots of other functions. When the program is RUN you will be asked to type in the formula involving X.

Listing 4.8

```
LIST
  10 REM Roots of functions
  20 MODE 1:COLOUR 3:PRINT ' TAB(11);"R
oots of functions":@%=10
  30 PRINT "This program attempts to fi
nd roots of functions."
  40 PRINT "Enter your function using X
as the vari-able, for example"
  50 PRINT "          X*EXP(X)*LN(ABS(X)+1)
":COLOUR 2
  60 INPUT "Function of X: ";X$
  70 DEF FNFunction(X)=EVAL(X$)
  80 PRINT '"Enter range over which sea
rch of roots is to be attempted."'
  90 REPEAT
 100 INPUT "Lower value ";A
 110 INPUT "Upper value ";B:PRINT
 120 IF A>=B THEN COLOUR 3:PRINT "Firs
t value should be lower.":COLOUR 2
 130 UNTIL A<B
 140 REM The search
 150 S=B-A:T=0:TEST=-1:D=1E-9
 160 REPEAT
 170 COLOUR 3:PRINT "Test run ";T+1;"
** ";10^T;" divisions **":PROCStep
 180 IF TEST THEN S=S/10:T=T+1
 190 UNTIL T=4 OR NOT TEST
 200 IF T=4 THEN PRINT '"Sorry - cannot
locate roots."
 210 COLOUR 3:PRINT CHR$(7) ' TAB(10);"
```

Essential Maths on the BBC and Electron Computers

```
Another go? Y or N ";
  220 REPEAT:G$=GET$:UNTIL G$="Y" OR G$=
"N"
  230 IF G$="Y" THEN RUN
  240 CLS:PRINT "Bye for now.":END
  300 REM Step by step search
  310 DEF PROCStep
  320 X=A:Y=FNFunction(X):X1=A:Y1=Y
  330 FOR X=A+S TO B STEP S
  340 Y=FNFunction(X):X2=X:Y2=Y
  350 IF Y1*Y2<=D THEN PROCFine
  360 Y1=Y2:X1=X
  370 NEXT
  380 ENDPROC
  400 DEF FNPoly(X)
  410 LOCAL I,Y
  420 Y=A(0):FOR I=1 TO N:Y=Y*X+A(I):NEX
T
  430 =Y
  440 DEF PROCFine
  450 PRINT "Fine tuning solution.":B$="
"
  460 IF ABS(Y1)<D THEN X=X1:TEST=0
  470 IF TEST AND ABS(Y2)<D THEN X=X2:TE
ST=0
  480 REPEAT
  490 IF TEST THEN Z=X:X=(X1+X2)/2:Y=FN
Function(X):IF ABS(Y)<D THEN TEST=0
  500 IF TEST AND ABS(Z-X)<D THEN B$="
probably ":TEST=0
  510 IF Y*Y2>0 THEN X2=X ELSE X1=X
  520 UNTIL NOT TEST
  530 COLOUR 1:PRINT "There is";B$;"a r
oot at ";X:X=B
  540 IF B$<>" " THEN PRINT "Value of p
olynomial is ";FNPoly(Z)
  550 ENDPROC
```

RUN

Roots of functions

This program attempts to find roots of functions.

Enter your function using X as the variable, for example

$X * \exp(X) * \ln(|X| + 1)$

Function of X: $X * \exp(X) - 1$

Enter range over which search of roots is to be attempted.

Lower value ?-1

Upper value ?1

Test run 1 ** 1 divisions **

Fine tuning solution.

There is a root at 0.5671432903

Another go? Y or N

