Computer Algebra and Mathematics with the HP40G

Version 1.0

Renée de Graeve

Lecturer at Grenoble I

Acknowledgments

It was not believed possible to write an efficient program for computer algebra all on one's own. But one bright person by the name of Bernard Parisse didn't know that—and did it!

This is his program for computer algebra (called ERABLE), built for the second time into an HP calculator.

The development of this calculator has led Bernard Parisse to modify his program somewhat so that the computer algebra functions could be edited and cause the appropriate results to be displayed in the Equation Editor.

Explore all the capabilities of this calculator, as set out in the following pages.

I would like to thank:

- Bernard Parisse for his invaluable counsel, his remarks on the text, his reviews, and for his ability to provide functions on demand both efficiently and graciously.
- Jean Tavenas for the concern shown towards the completion of this guide.
- Jean Yves Avenard for taking on board our requests, and for writing the PROMPT command in the very spirit of promptness—and with no advance warning. (refer to 6.4.2.).

Preface

The HP40G marks a new stage in the democratisation of the use of symbolic calculation—on the one hand by its very competitive price, and on the other hand by making it possible to execute, step-by-step, the principle algorithms taught in mathematics at secondary schools and in the first years of university.

But it was still necessary to add adequate documentation, preferably written by a teacher of mathematics. That is what you find in this guide, written by Renée De Graeve, Lecturer at the University of Grenoble I and founder of the Grenoble IREM. It contains, naturally, a complete reference of the functions for symbolic calculation, but also demonstrates, using examples taken from study for both certificate and diploma, how to take smart advantage of the calculating power of the HP40G. The guide ends with two chapters dedicated to programming: the first for learning to program, and the second to illustrate the application of algorithmic language to arithmetic programs as taught in French tertiary schools.

Bernard Parisse

Lecturer at the University of Grenoble I

Table of Contents

0 To Begin 9
- 0.1 General Information 9
  - 0.1.1 Starting the Calculator 9
  - 0.1.2 What You See 9
- 0.2 Notation 10
  - 0.2.1 Line-Based Help 10

1 Aplets 10
- 1.1 APLET Key 10
- 1.2 The Various aplets 10
- 1.3 Examples using the `Sequence` aplet 11
- 1.4 The SYMB NUM PLOT Keys 12

2 The Keyboard and CAS 13
- 2.1 What is CAS? 13
- 2.2 The current variable 13
- 2.3 How do you perform a symbolic calculation? 13
- 2.4 CAS in the Equation Editor 13
- 2.5 The keyboard in the Equation Editor 14
  - 2.5.1 MATH key 14
  - 2.5.2 SHIFT MATH (CMDS) keys 14
  - 2.5.3 VARS key 14
  - 2.5.4 SHIFT 2 (SYNTAX) keys 14
  - 2.5.5 HOME key 15
  - 2.5.6 SHIFT SYMB keys 15
  - 2.5.7 SHIFT key 15
  - 2.5.8 PLOT key 15
  - 2.5.9 NUM key 16
  - 2.5.10 VIEWS key 16
  - 2.5.11 Short-cut keys 16
- 2.6 CAS in the HOME screen 16
  - 2.6.1 PUSH 16
  - 2.6.2 POP 16
- 2.7 The keyboard in the HOME screen 17
  - 2.7.1 MATH key 17
  - 2.7.2 SHIFT F6 keys 17
  - 2.7.3 SHIFT 2 (SYNTAX) keys 17
  - 2.7.4 SHIFT 1 (PROGRAM) keys 17

3 Writing expressions in the Equation Editor 18
- 3.1 Equation Editor 18
  - 3.1.1 Access to the Equation Editor 18
  - 3.1.2 How do you select? 18
  - 3.1.3 How to modify an expression 21
  - 3.1.4 The cursor mode 21
  - 3.1.5 To see everything 21
- 3.2 Accessing the CAS functions 21
  - 3.2.1 How to type $\int$ and Σ 21
  - 3.2.2 How to call infix functions 22
  - 3.2.3 How to call prefix functions 22
- 3.3 Variables 23
  - 3.3.1 STO▷ 24
  - 3.3.2 STORE 24
  - 3.3.3 Predefined CAS variables 24

4 Computer Algebra Functions 25
- 4.1 CAS Toolbar 25
  - 4.1.1 CFG 25
  - 4.1.2 TOOL 26
  - 4.1.3 ALGB 26
  - 4.1.4 DIFF&INT 26
  - 4.1.5 REWRITE 26
  - 4.1.6 SOLVE 26
  - 4.1.7 TRIG 26
  - 4.1.8 The MATH Key 27
- 4.2 Step by Step Mode 27
- 4.3 General Use 27
  - 4.3.1 DEF 27
- 4.4 INTEGERS (and Gaussian Integers) 28
  - 4.4.1 DIVIS 28
  - 4.4.2 EULER 28
  - 4.4.3 FACTOR 28
  - 4.4.4 GCD 28
  - 4.4.5 IEGCD 29
  - 4.4.6 IQUOT 29
  - 4.4.7 IREMAINDER MOD 29
  - 4.4.8 ISPRIME? 29

4.4.9 LCM 30
4.4.10 NEXTPRIME 30
4.4.11 PREVPRIME 30
4.5 Modular Calculations 30
4.5.1 ADDTMOD 30
4.5.2 DIVMOD 30
4.5.3 EXPANDMOD 31
4.5.4 FACTORMOD 31
4.5.5 GCDMOD 31
4.5.6 INVMOD 31
4.5.7 MODSTO 31
4.5.8 MULTMOD 31
4.5.9 POWMOD 32
4.5.10 SUBTMOD 32
4.6 Rational Numbers 32
4.6.1 PROPFRAC 32
4.7 Real Numbers 32
4.7.1 FLOOR 33
4.7.2 MOD 33
4.8 Complex Numbers 33
4.8.1 ARG 33
4.8.2 CONJ 34
4.8.3 DROITE 34
4.9 Algebraic Expressions 34
4.9.1 COLLECT 34
4.9.2 EXPAND 35
4.9.3 FACTOR 35
4.9.4 | 35
4.9.5 SUBST 35
4.10 Polynomials 35
4.10.1 DEGREE 35
4.10.2 EGCD 36
4.10.3 FACTOR 36
4.10.4 GCD 36
4.10.5 HERMITE 36
4.10.6 LCM 37
4.10.7 LEGENDRE 37
4.10.8 PARTFRAC 37
4.10.9 PROPFRAC 37
4.10.10 PTAYL 38
4.10.11 QUOT 38
4.10.12 REMAINDER 38
4.10.13 TCHEBYCHEFF 38
4.11 Functions 38
4.11.1 DEF 39
4.11.2 IFTE 39
4.11.3 DERVX 39
4.11.4 DERIV 39
4.11.5 TABVAR 40
4.11.6 FOURIER 40
4.11.7 IBP 40
4.11.8 INTVX 41
4.11.9 LIMIT 42
4.11.10 LIMIT and $\int$ 42
4.11.11 PREVAL 43
4.11.12 RISCH 43
4.12 Limited and Asymptotic Expansions 43
4.12.1 DIVPC 43
4.12.2 LIMIT 43
4.12.3 SERIES 43
4.12.4 TAYLOR0 45
4.12.5 TRUNC 45
4.13 Conversion Functions 45
4.13.1 DISTRIB 45
4.13.2 EPSX0 46
4.13.3 EXP2POW 46
4.13.4 EXPLN 46
4.13.5 FDISTRIB 46
4.13.6 LIN 46
4.13.7 LNCOLLECT 47
4.13.8 POWEXPAND 47
4.13.9 SIMPLIFY 47
4.13.10 XNUM 47
4.13.11 XQ 48
4.14 Equations 48
4.14.1 ISOLATE 48
4.14.2 SOLVEVX 48
4.14.3 SOLVE 48
4.15 Linear Systems 48
4.15.1 LINSOLVE 49
4.16 Differential Equations 49
4.16.1 DESOLVE and SUBST 50
4.16.2 LDEC 50

4.17 Trigonometric Expressions 50
4.17.1 ACOS2S 50
4.17.2 ASIN2C 50
4.17.3 ASIN2T 51
4.17.4 ATAN2S 51
4.17.5 HALFTAN 51
4.17.6 SINCOS 51
4.17.7 TAN2CS2 52
4.17.8 TAN2SC 52
4.17.9 TAN2SC2 52
4.17.10 TCOLLECT 52
4.17.11 TEXPAND 52
4.17.12 TLIN 53
4.17.13 TRIG 53
4.17.14 TRIGCOS 53
4.17.15 TRIGSIN 53
4.17.16 TRIGTAN 54

5 Worked Exercises with the HP40 54

5.1 Introduction 54
5.2 Exercises on Brevet 55
5.2.1 Exercise 1 55
5.2.2 Exercise 2 55
5.2.3 Exercise 3 55
5.2.4 Exercise 4 56
5.2.5 Exercise 5 57
5.3 Exercises on Bac 57
5.3.1 Exercise 1 57
5.3.2 Exercise 2 (specialty) 60
5.3.3 Exercise 3 (non-specialty) 63
5.4 Conclusion 65

6 Programming 65

6.1 Implementation 66
6.1.1 How to edit and save a program 66
6.1.2 How to correct a program 66
6.1.3 How to run a program 66
6.1.4 How to modify a program 66
6.2 Comments 66
6.3 Variables 66
6.3.1 Their names 66
6.3.2 The concept of local variables 66
6.3.3 The concept of parameters 67
6.4 User entry 67
6.4.1 Algorithmic syntax 67
6.4.2 HP40G syntax 67
6.5 Output 67
6.5.1 Algorithmic syntax 67
6.5.2 HP40G syntax 67
6.6 The sequence of statements, or “action” 67
6.6.1 Algorithmic syntax 67
6.6.2 HP40G syntax 67
6.7 The assignment statement 67
6.7.1 Algorithmic syntax 67
6.7.2 HP40G syntax 67
6.8 Conditional statements 67
6.8.1 Algorithmic syntax 67
6.8.2 HP40G syntax 68
6.9 FOR statement 68
6.9.1 Algorithmic syntax 68
6.9.2 HP40G syntax 68
6.10 WHILE statement 68
6.10.1 Algorithmic syntax 68
6.10.2 HP40G syntax 68
6.11 Boolean Expressions 68
6.11.1 Algorithmic syntax 68
6.11.2 HP40G syntax 68
6.12 Logical Operators 68
6.12.1 Algorithmic syntax 68
6.12.2 HP40G syntax 68
6.13 Lists 69
6.13.1 Algorithmic syntax 69
6.13.2 HP40G syntax 69
6.14 Example: The Sieve of Eratosthenes 69
6.14.1 Description 69
6.14.2 Algorithmic syntax 69
6.14.3 HP40G syntax 70

7 Arithmetic Programs 71

7.1 GCD and Euclid’s Algorithm 71
7.1.1 Algorithmic syntax 71

7.1.2 HP40G syntax 71
7.2 Bézout's Identity 72
7.2.1 Iterative version without lists 73
7.2.2 Iterative version with lists 73
7.2.3 Recursive version with lists 73
7.2.4 Recursive version without lists 74
7.2.5 HP40G syntax 74
7.3 Decomposition into prime factors 75
7.3.1 Algorithmic syntax 75
7.3.2 HP40G syntax 76
7.4 Calculating $A^P \ mod \ N$ 76
7.4.1 Algorithmic syntax 76
7.4.2 HP40G syntax 77
7.5 The function "isprime" 77
7.5.1 Algorithmic syntax 77
7.5.2 HP40G syntax 78
7.6 Mr Rabin's probability method 78
7.6.1 Algorithmic syntax 79
7.6.2 HP40G syntax 79
8 GNU Free Documentation License 80

0 To Begin

0.1 General Information

0.1.1 Starting the Calculator

Press the ON button.

The HOME screen is displayed.

As you work, you can use the ON button to cancel the current operation. In other words, it has the same function as CANCEL.

To turn the calculator off, press SHIFT plus ON (OFF).

If the calculator fails to respond after several presses of the ON (CANCEL) button, you can press ON and F3 simultaneously to reinitialise the calculator.

0.1.2 What You See

From top to bottom:

1. The HOME screen
   1.a. The calculator's status
   1.b. A horizontal line
   1.c. A menu bar of commands
2. The keyboard

1. The HOME screen

1.a The calculator's status gives the HOME screen's current mode:

- RAD, DEG, or GRD when you are working with radians, degrees, or gradians.
- {FUNCTION} to indicate the name of the aplet currently selected— here, the `Function` aplet
- ▲ to indicate that you can use the up-arrow to move back through the history.

1.b Horizontal line:

Above the horizontal line is found an overview of the calculations carried out in the HOME screen.

Principle: On the screen, the requested calculation is written on the left, and the result is written on the right.

Beneath the horizontal line is the command line.

Using the up-arrow, you can move back through the history and, using COPY on the menu bar, copy a command or a preceding result into the command line.

1.c Menu Bar:

The commands on the menu bar are accessible through the six unlabelled grey keys, which we refer to here as:

F1 F2 F3 F4 F5 F6.

The menu bar can contain items consisting of groups of commands; they are marked by a folder icon.

To activate a command on the menu bar, it's enough to press the corresponding Fi key.

In the HOME screen, the menu bar contains two commands:

- STO▷, which enables you to store a value in a variable, and
- CAS, which enables you to open the Equation Editor to perform computer algebra.

2. The keyboard

You have already been introduced to:

The ON key for starting the calculator or for cancelling the current operation, and SHIFT ON for turning the calculator off.

Other keys you must know include:

- The four arrow keys (left, right, up, down), which enable you to move the cursor while you're in the Equation Editor, in a menu, and so on
- The SHIFT key, which enables you to access two functions with one key
- The ALPHA key for typing text in upper case, and the SHIFT ALPHA keys for typing text in lower case
  To remain in alphabetic mode, you must press and hold the ALPHA key.
- X, T, and θ, which enable you to type X, T, θ, and N directly, depending on the active aplet.

- The ENTER key, which is used to confirm a command.

0.2 Notation

The four arrow keys are represented by four triangles:

▷◁△▽

STO▷on the HOME menu is represented in the program by:

STO▷ or ▷

In the Equation Editor, the position of the cursor is represented by:

◂

0.2.1 Line-Based Help

With this calculator, you're able to get practical and efficient line-based help in either French or English (cf. 4.1.1).

You are shown an alphabetised list of computer algebra functions. As with the drop-down menus, you're able to access the functions by pressing the corresponding letter-keys without having to worry about pressing ALPHA.

The help consists of a succinct description of the command, as well as an example and its result. Each example can be tested with ECHO (on the menu) and can be either used as is or modified. You can view help for related commands by means of SEE1 SEE2... on the menu. For more details, refer to the description of SHIFT-2 (SYNTAX), sections 2.5.4 and 2.7.3.

1 Aplets

1.1 APLET Key

The APLET key gives you access to the list of available aplets.

This calculator in effect enables you to work with aplets.

But what is an aplet?

An aplet is a program stored in the calculator, which enables you to easily obtain three views of a mathematical object (a symbolic view, a numeric view, and a graphic view)—and all this is pre-programmed!

The various aplets enable you to work with mathematical objects such as: functions, sequences, statistical data, and so on...

Certain aplets illustrate particular classroom lessons.

1.2 The Various aplets

While you're in HOME, you can see the name of the selected aplet by looking at the status line.

Here are the possible choices for the APLET key:

- `Sequence`

  This aplet enables you to define sequences having the names:

  U1, U2 …U9, U0

  One defines U1(N):

  – To be a function of N,

  – To be a function of U1(N – 1),

  – To be a function of U1(N – 1) and U1(U – 2).

  You define (for example):

  U1(N) = N∗N+1

  and then the values of U1(1) and U1(2) are automatically calculated and displayed.

  To display the values of U1(N), choose U1, then press NUM.

You can find other examples that use the `Sequence` aplet in the following section, such as the calculation of the GCD of two numbers (cf. 1.3), and the calculation of the coefficients of Bézout's Identity (cf. 1.3).

- `Function`

  This aplet enables you to define functions having the names:

  F1(X), F2(X) …F9(X), F0(X)

  One defines F1(X):

  – To be an expression of a function of X:
  For example, the formula:
  F1(X) = X∗LN(X)
  defines the function:
  $f_1(x) = x \cdot ln(x)$

  – To use Booleans (X>0 etc.) if the function is defined in parts:
  For example, a formula of the form:
  F1(X) = X∗(X<0)+2∗X∗(X>0)
  defines the function:
  $f_1(x) = x$ If $x < 0$ and
  $f_1(x) = 2 \cdot x$ If $x > 0$

- `Parametric` for tracing curves in parametric coordinates.
- `Polar` for tracing curves in polar coordinates.
- `Solve` for solving numeric equations.
- `Statistics` for working with statistics.
- `Inference` for working with inferential statistics.
- `Quad Explorer,` for exploring quadratic functions.
- `Trig Explorer,for exploring trigonometric functions`

1.3 Examples using the `Sequence` aplet

Notation in Base *b*

Given a and b, produce the series q_n ($n \geq 1$) and r_n ($n \geq 2$) from the quotients and the remainders of the division of q_i by b, defined by:

$q_1 = a$
$q_1 = b \cdot q_2 + r_2$ $(0 \leq r_2 < b)$
$q_2 = b \cdot q_3 + r_3$ $(0 \leq r_3 < b)$
……
$q_{n-1} = b \cdot q_n + r_n$ $(0 \leq r_n < b)$

Note that if $r_{n+1} = 0$, the number $r_n r_{n+1} \ldots r_3 r_2$ is notated in base b of a, while one assumes $2 \leq b \leq 10$.

Put into B the value of the base—for example:

7 STO▷B

Put into A the number to write in base B (for example, 1789 STO▷A)

Define the following two series:

U1(1)=A
U1(2)=FLOOR(A/B)
U1(N)=FLOOR(U1(N – 1)/B)

and

U2(1)=0
U2(2)=A MOD B
U2(N)=U1(N – 1) MOD B

Therefore, q_n=U1(N) and r_n=U2(N).

Calculating the GCD

This is an application of Euclid's Algorithm on the HP40G.

Here is the description of this algorithm:

If one performs the successive Euclidean divisions:

$$A = B \times Q_1 + R_1 \quad 0 \leq R_1 < B$$
$$B = R_1 \times Q_2 + R_2 \quad 0 \leq R_2 < R_1$$
$$R_1 = R_2 \times Q_3 + R_3 \quad 0 \leq R_3 < R_2$$
…….

then after a finite number of steps (in excess of B), there exists a whole number n such that: $R_n = 0$.

We have then:

$GCD\ (A,B) = GCD\ (B, R_1) = \ldots$
$GCD\ (R_{n-1}, R_n) = GCD\ (R_{n-1}, 0) = R_{n-1}$

Using a sequence, one then writes the sequences of remainders.

With the HP40G, you use the `Sequence` aplet (APLET key, then select `Sequence`, then Start on the menu bar).

To determine the GCD(78,56), you define the sequence:

U1(1) = 78
U1(2) = 56
U1(N) = U1(N – 2) MOD U1(N – 1).

Press NUM to get the numerical list of U1(N)—that is, the list of the remainders of the successive divisions...

The final non-zero remainder is 2, so the GCD(78,56)=2.

Remark:

In HOME, you can use the variables A and B to store the two numbers, and then make U1(1)=A and U1(2)=B.

It's also important to note that A MOD 0 = A.

Calculating the Coefficients of Bézout's Identity

Euclid's Algorithm enables you to find a pair U, V such that:

$A \times U + B \times V = GCD(A, B)$

Using the idea of sequence:

Define "the sequence of remainders" R_n and two sequences U_n and V_n, such that at each step one has:

$R_n = U_n \times A + V_n \times B.$

Seeing that one has: $R_n = R_{n-2} - Q_n \times R_{n-1}$, U_n and V_n serve to satisfy the same recurrence relation (Q_n = whole-number quotient of R_{n-2} divided by R_{n-1}).

One then has (from the beginning):

$R_1 = A \;\; R_2 = B$
$U_1 = 1 \;\; U_2 = 0$ since $A = 1 \times A + 0 \times B$
$V_1 = 1 \;\; V_2 = 0$ since $B = 0 \times A + 1 \times B$

With the HP40G, using the `Sequence` aplet, you then define the sequence of remainders U1 and the sequences U2 and U3 such that for all N one has:
`U1(N)=A*U2(N)+B*U3(N).`

For this, you need the sequence of quotients, which you put into U4.

The sequences U1, U2 and U3 satisfy the same recurrence relation:

$U_n = U_{n-2} - Q_n \times U_{n-1}$, with
Q_n =`U4(N)=FLOOR(U1(N-2)/U1(N-1))`

You therefore define:

```
U1(1)=A
U1(2)=B
U1(N)=U1(N-2)-U4(N)*U1(N-1)
U2(1)=1
U2(2)=0
U2(N)=U2(N-2)-U4(N)*U2(N-1)
U3(1)=0
U3(2)=1
U3(N)=U3(N-2)-U4(N)*U3(N-1)
U4(1)=0
U4(2)=0
U4(N)=FLOOR(U1(N-2)/U1(N-1))
```

It's important to note that you use U4(N) only for N>2; you have therefore defined the two first values (which are useless!) as zero.

NUM then displays the values of these various sequences, and on the line of the final non-zero remainder you can read the GCD and the coefficients of Bézout's Identity.

1.4 The SYMB NUM PLOT Keys

In general, an aplet can be viewed in three different ways:

- A symbolic view, which corresponds to the SYMB key
- A numeric view, which corresponds to the NUM key
- A graphic view, which corresponds to the PLOT key

When these keys are SHIFTed (SETUP), this corresponds to choosing the various available parameters (choosing the parameters of the graphic window, the step size for the table etc...)

2 The Keyboard and CAS

2.1 What is CAS?

CAS enables you to perform exact or symbolic calculations:

(CAS = Computer Algebra System).

Make sure you understand the difference between:

- Exact or symbolic calculations, which are performed by means of the CAS functions. You work in `exact mode`, with infinite precision, and with the capability of performing the calculations step-by-step, AND
- Numeric calculations, which are performed by means of the MATH key's MTH menu, either in the HOME screen or in aplets or programs. You work in `approximate mode`, with a precision of 10^{-12}.

Example:

If you're working in radians in the HOME screen:

ARG (1 + *i*) returns 0.785398163397

whereas in CAS, where you're always working in radians:

ARG (1 + *i*) returns $\frac{\pi}{4}$

2.2 The current variable

When you use the symbolic calculation functions, you're working with symbolic variables (variables that don't contain a permanent value).

The name of the symbolic variable contained in VX is called the "current variable"; this is almost always X.

2.3 How do you perform a symbolic calculation?

The HP40G has been designed to use computer algebra functions in the Equation Editor.

To open the Equation Editor, press CAS on the menu bar of the HOME screen.

To leave the Equation Editor, press ON to return to the HOME screen.

You can, however, perform computer algebra in the HOME screen, as long as you take certain precautions (cf. 2.6).

Refer to the remaining chapters of this guide for information on how to use the CAS functions.

2.4 CAS in the Equation Editor

The Equation Editor enables you to type expressions that you want to simplify, factor, differentiate, integrate, and so on, and then work them through as if on paper.

The editor is supplied with a menu bar of menus:

1. The TOOL menu contains the commands:
   `Cursor mode`
   `Edit expr.`
   `Change font`
   `Cut`
   `Copy`
   `Paste`
   - `Cursor mode` enables you to go into cursor mode (cf. 3.1.4).
   - `Edit expr.` enables you to edit (modify) the highlighted expression.
   - `Change font` enables you to choose to type using large or small characters (you can make this choice at any time).
   - `Cut` copies the selection into the buffer, then erases the selection.
   - `Copy` copies the selection into the buffer.
   - `Paste` copies the buffer to the location of the cursor (the buffer contains whatever was selected the last time Copy or Cut was chosen).
2. The `ALGB` menu contains functions that enable you to perform algebra: factoring, expansion, simplification, substitution...
3. The `DIFF&INT` menu contains functions that enable you to perform differential calculus: differentiation, integration, series expansion, limits...
4. The `REWRITE` menu contains functions that enable you to rewrite an expression in another form.
5. The `TRIG` menu contains functions that enable you to transform trigonometric expressions.
6. The `SOLVE` menu contains functions that enable you to solve equations, linear systems, and differential equations.

Chapter 3 tells you how to write an expression in the Equation Editor, how to select a sub-expression, and how to call the CAS functions.

Chapter 4 explains all the symbolic calculation functions contained in the various menus, together with examples of use.

You can consult the online help with SHIFT 2 (SYNTAX) (cf. 2.5.4) to get help for the other available functions, and you can use SHIFT MATH (CMDS) (cf. 2.5.2) to type them.

2.5 The keyboard in the Equation Editor

The keys mentioned in this section have different functions depending on whether they are pressed in the Equation Editor or in the HOME screen. For the functionality of these keys outside the Equation Editor, refer to section 2.7, or consult the User's Guide.

2.5.1 MATH key

The MATH key, if pressed in the Equation Editor, displays the functions used in symbolic calculation. These functions are contained in menus:

- The five preceding menus (cf. 2.4):
  Algebra Diff&Int Rewrite Solve Trig .
- The Complex menu... containing functions that enable you to work with complex numbers
- The Constant menu... (e *i* ∞ π)
- The Hyperb. menu... containing hyperbolic functions
- The Integer menu... containing functions that enable you to perform integer arithmetic
- The Modular menu... containing functions that enable you to perform calculations in *Z/pZ* or *Z/pZ[X]*, *p* being the value contained in the variable MODULO
- The Polynom. menu... containing functions that enable you to perform calculations with polynomials
- The Tests menu... containing:
  ASSUME UNASSUME (to make hypotheses about the parameters, and to modify the variable REALASSUME (cf. 3.3.3)
  > ≥ < ≤ == ≠ AND OR NOT
  IFTE (to write an algebraic function having the same result as an IF THEN ELSE)

Refer to section 4.1.8 for the list of the functions contained in the various menus.

2.5.2 SHIFT MATH (CMDS) keys

This key combination opens the list of all the CAS commands available in the Equation Editor.

In this way, functions that are not presented elsewhere can be called from this menu, so you don't have to type them in ALPHA mode.

2.5.3 VARS key

This key, if pressed while you're in the Equation Editor, displays the names of the variables defined in CAS.

Take special note of `namVX`, which contains the name of the current variable.

To see the contents of a variable, all you have to do is highlight its name and press F2 for VIEW on the menu bar.

To change the contents of a variable, highlight its name and press F3 for EDIT on the menu bar.

Note also on the menu bar:

PURGE, which enables you to destroy an existing variable

RENAME, which enables you to change the name of an existing variable

NEW, which enables you to define a new variable: just enter the contents, then the name

For more details, refer to section 3.3.

2.5.4 SHIFT 2 (SYNTAX) keys

While you're in the Equation Editor, the key combination SHIFT 2 (SYNTAX) opens the CAS HELP ON menu.

To get help in French, choose Français on the CFG menu, which enables you to change your configuration (cf. 4.1.1).

If there is no CAS function selected in the editor, the menu shows a list of functions available in the Equation Editor. Just highlight a function and press OK to see the help for that function.

If there is a CAS function selected in the editor, for example: FACTOR(45), the CAS HELP ON menu directly opens the help topic for FACTOR. The help consists of a short description of the command, as well as an example and its result. Each example can be copied into the Equation Editor by means of ECHO on the menu bar, where it can be used as is or modified.

Note that in the help examples, VX=X is used as the current variable. If that is not the case for you, the example will be automatically transformed, taking your value of VX into account , when you transfer it with ECHO.

You can also go directly to see the help of a command pointed to by SEE: with SEE1, SEE2... on the menu bar.

2.5.5 HOME key

Pressing the HOME key in the Equation Editor enables you to access the CAS history.

The history of the calculations performed in CAS differs from the history of the calculations performed in HOME.

As in the HOME screen history, the requested calculations are written on the left, and the results are written on the right. Using the up-arrow, you can move back through the history.

Using ENTER, or ECHO on the menu bar, you can easily copy a preceding result or a previous command.

2.5.6 SHIFT SYMB keys

While you're working in the Equation Editor, the key combination:

SHIFT SYMB (SETUP) has the same effect as choosing CFG (the first item in the ALGB menu on the menu bar; cf. 4.1.1).

This enables you to specify:

- The name of the variable contained in VX, by typing its name next to `indep var.`,
- The value of MODULO, by typing its value next to `Modulo`,
- Whether you want to work in exact mode (or in approximate mode, if you've chosen `Approx` with CHK on the menu bar)
- Whether you want to work in real mode (or in complex mode, if you've chosen `Complex` with CHK on the menu bar)
- Whether you want to work in Direct mode (or in Step by step mode, if you've chosen `Step/Step` with CHK on the menu bar)
- Whether you want polynomials to be written in decreasing order according to exponent (or increasing order, if you've chosen `Incr Pow` with CHK on the menu bar)
- Whether you want numerical factors suppressed (or enabled, if you've chosen `Num.Factor` with CHK on the menu bar)
- Whether you want to work in non-rigorous mode (or in rigorous mode, if you've chosen `Rigorous` with CHK on the menu bar so as not to neglect the absolute values!)
- Whether you want to simplify non-rational expressions (or not, if you clear the selection by pressing CHK on the menu bar).

Use OK or ENTER to confirm your choices.

2.5.7 SHIFT key

While you are working in the Equation Editor, the key combination:

SHIFT (MEMORY) plays the role of an "undo" key.

This is very useful when you've made a mistake, because it enables you to cancel the last command.

2.5.8 PLOT key

When you press PLOT in the Equation Editor, a dialog box asks you if you want to graph a function, a parametric curve, or a polar curve.

Depending on what you choose, the highlighted expression is copied into the appropriate aplet, to the location that you have specified as the destination.

Note: This supposes that the current variable is also the variable of the function or curve you want to graph: when the expression is copied, it's evaluated, and the current variable (as contained in VX) is changed to X, T, or θ, depending on the type of plot.

Note: If the function depends on a parameter, it is preferable to give the parameter a value before pressing PLOT. If, however, you want the parameterised expression to be copied with its parameter, then the name of the parameter must consist of a single letter other than X, T, or θ, so that there is no confusion.

If you choose:

- the `Function` aplet, the highlighted expression is copied into the chosen function `Fi`, and the current variable is changed to X during the copy
- the `Parametric` aplet, the real part and the imaginary part of the highlighted expression are copied into the chosen functions `Xi`, `Yi`, and the current variable is changed to T during the copy
- the `Polar` aplet, the highlighted expression is copied into the chosen function `Ri`, and the current variable is changed to θ during the copy

2.5.9 NUM key

Pressing the NUM key in the Equation Editor causes the highlighted expression to be replaced by a numeric approximation.

NUM puts the calculator into approximate mode.

SHIFT NUM does the opposite: it puts the calculator into exact mode.

2.5.10 VIEWS key

Pressing the VIEWS key in the Equation Editor enables you to move the cursor with the ▷ and ◁ arrows to see the entire highlighted expression.

2.5.11 Short-cut keys

Note: In the Equation Editor, the following short-cut keys are available on the keyboard:

SHIFT 0 for ∞
SHIFT 1 for *i*
SHIFT 3 for π
SHIFT 5 for <
SHIFT 6 for >
SHIFT 8 for ≤
SHIFT 9 for ≥

2.6 CAS in the HOME screen

You can use certain computer algebra functions directly in the HOME screen, as long as you take certain precautions:

- If you use the computer algebra functions that are found under CAS in the MATH key's menu bar (when the key is pressed in the HOME screen), use S1 as the current variable; for example:

  $\mathrm{DERVX}(\mathrm{S1}^2 - 4 \cdot \mathrm{S2}) = 2 \cdot \mathrm{S1}$

- Use the variables S1, S2, …S5, E0, E1, …E9, s1, s2, …s5, or n1, n2, …n5 as symbolic variables.
- If you want to work with symbolic matrices, you must store them in L1, L2, …L9, L0, because these matrices are interpreted in the same way as lists of lists (whereas numeric matrices are stored in M1, M2, …M9, M0). You type (for example):

  $[\mathrm{S1}+1, \mathrm{XQ}(\sqrt{2})]\mathrm{STO} \triangleright \mathrm{L1}$

Note: Certain calculation are performed in approximate mode because of the ambiguity between real and integers in HOME. Using the command XQ enables you to convert an approximate argument into an exact argument; in the example given above in section 2.1, in the HOME screen you would type (see also 2.7.1 and 2.7.3):

$$\mathrm{ARG}(\mathrm{XQ}(1+i)) = \frac{\pi}{4}$$

Using the commands PUSH and POP, you can also transfer expressions from the HOME screen history to the CAS history.

2.6.1 PUSH

In the HOME screen, you can use the PUSH command to send expressions into the CAS history.

In the HOME screen, you type:

PUSH(S1+1)

and S1+1 is written to the CAS history.

2.6.2 POP

In the HOME screen, you can use the POP command to retrieve the last expression written to the CAS history.

In the HOME screen, you type:

POP

and (for example) S1+1 is written to the HOME screen history.

2.7 The keyboard in the HOME screen

2.7.1 MATH key

This opens the menu of mathematical functions.

This key, if pressed in the HOME screen, open a window containing the mathematical (numeric) functions grouped by theme, since the MTH option on the menu bar (F1 key) is selected by default.

If you choose CAS on this window's menu bar (F3 key), you'll find the same menus as when you press the MATH key in the Equation Editor: this gives you access to all available computer algebra functions, grouped by theme, from the HOME screen (not forgetting that, in the HOME screen, the only symbolic variables are S1, S2, …S5, E0, E1, …E9, s1, s2, …s5, or n1, n2, …n5).

2.7.2 SHIFT F6 keys

The key combination SHIFT F6 (SHIFT CAS on the menu bar) opens the CAS configuration screen, which enables you change the CAS configuration from the HOME screen.

2.7.3 SHIFT 2 (SYNTAX) keys

The key combination SHIFT 2 (SYNTAX) places HELPWITH in the command line. All you need to do then is complete the line with the name of the command or CAS function for which you want help. You can enter the name of a CAS function with MATH CAS, but take care to omit the parentheses.

For example: HELPWITH DERVX opens the CAS help topic on DERVX.

If you want general CAS help in the HOME screen, press HELP, then ENTER: this gives you help on the CAS functions available in the HOME screen.

To get help in French, choose `Français` in the CFG menu, which enables you to change your configuration (cf. 4.1.1).

Each example can be copied into the HOME screen history by means of ECHO on the menu bar, where it can be used as is or modified. (Naturally, the variable X is replaced by S1.)

In addition, while in the HOME screen you will sometimes want to use the function XQ to change reals to integers.

For example:

$$\mathrm{PROPFRAC}\left(\frac{43}{12}\right) = 3.5833..$$

whereas

$$\mathrm{PROPFRAC}\left(\mathrm{XQ}\left(\frac{43}{12}\right)\right) = 3 + \frac{7}{12}$$

2.7.4 SHIFT 1 (PROGRAM) keys

This key combination, if pressed in HOME, displays the screen

PROGRAM CATALOG

It shows:

- A list of the programs that you've written, and
- a menu bar containing the commands:

EDIT NEW RUN SEND RECV.

EDIT enables you to edit the highlighted program,

NEW enables you to create a new program,

RUN enables you to run the highlighted program,

SEND and RECV are functions that enable your calculator to talk to your computer or another calculator.

For example:

If you press SEND on the menu bar, it asks:

`HP39/40 (Wire) or Disk Drive`

You highlight `HP39/40 (Wire)` to send a program to another HP40G, or you highlight `Disk Drive` to send a program to a computer.

Then press OK on the menu bar.

For example, here is how you connect a `Linux` computer to the HP40G so as to use the program C-Kermit version 7 (which you can find at the URL `www.columbia.edu/kermit`, or which you can download via anonymous `ftp` from the site `kermit.columbia.edu`):

- Connect the calculator to a data transfer cable.
- On the computer, type:

```
kermit
set line /dev/ttyS0
```
(...or `S1`, depending on the number of your serial port)
```
set speed 9600
set carrier-watch off
serv
```

- On the HP40G:

  Highlight the program called NAME, then press SEND on the menu bar and highlight `Disk drive`. Then press OK on the menu bar, and the program called NAME on the HP40G is copied into your computer.

  OR:

  Press RECV on the menu bar and highlight `Disk drive`. Then press OK on the menu bar, and the calculator displays a list of the HP40G programs on your computer. (Naturally, you have to have already created a directory on your computer where the HP40G programs are stored).

  You then highlight GCD, and the program called GCD on your computer is copied into the HP40G.

For Windows users, the connectivity program is found on the CD that comes with the HP40G.

To find out more about the use of Kermit with HP calculators, visit the URL:

`http://www.columbia.edu/kermit/hp48.html`

3 Writing expressions in the Equation Editor

3.1 Equation Editor

3.1.1 Access to the Equation Editor

The CAS key on the menu bar takes you into the Equation Editor, and the ON (CANCEL) key takes you back out.

The Equation Editor is a very efficient editor for writing, simplifying and transforming mathematical expressions.

When you type expressions in the Equation Editor, the operator that you are typing always carries over to the adjacent or selected expression.

You don’t have to preoccupy yourself with where the parentheses go... just select!

You need to view a mathematical expression as a binary tree, and the four arrow keys as enabling you move through the tree in a natural fashion:

- The right and left arrow keys enable you to move from one branch to another
- The up and down arrow keys enable you to go up and down a particular tree
- The SHIFT-up and SHIFT-down arrow keys enable you to make multiple selection (cf. 3.1.2, example 3).

3.1.2 How do you select?

There are two ways of going into selection mode:

- The up-arrow △ takes you into selection mode and selects the element adjacent to the cursor.

  Example:

  $$1+2+3+4 \ \triangle$$

  selects 4, then △ selects the entire tree 1 + 2 + 3 + 4.

- The right-arrow ▷ takes you into selection mode and selects the branch adjacent to the cursor.

  Pressing ▷ again augments the selection, adding the next branch to the right.

Example:

$$1 + 2 + 3 + 4 \triangleright$$

selects 3 + 4, then ▷selects 2 + 3 + 4, then ▷ selects 1 + 2 + 3 + 4.

Note: If you are typing a templated function with multiple arguments (such as $\sum$, $\int$, SUBST, or the like), the right arrow ▷ enables you to move through the template by changing the location of the cursor. In effect, the left and right arrow keys ▷and ◁ enable you to move from one argument to another. In this case, you always use the up arrow △ to select (cf. 3.2.1).

Examples of the way this editor works:

Press CAS on the menu bar to open the Equation Editor, then type the expressions in the examples.

- Example 1

  Typing:

  $$2 + X \times 3 - X$$

  returns:

  $$2 + X \cdot 3 - X$$

  Press ▷▷▷ to select the expression,

  then the ENTER key to produce:

  $$2 + 2 \cdot X$$

  Typing:

  $$2 + X \triangleright \times 3 - X$$

  returns:

  $$(2 + X) \cdot 3 - X$$

  Press ▷▷to select the expression,

  then the ENTER key to produce:

  $$6 + 2 \cdot X$$

  Typing:

  $$2 + X \triangleright \times 3 \ \triangle - X$$

  returns:

  $$(2 + X) \cdot (3 - X)$$

  Press ▷▷▷ to select the expression,

  then the ENTER key to produce:

  $$-(X^2 - X - 6)$$

- Example 2

  To express:

  $$X^2 - 3 \cdot X + 1$$

  type:

  $$X \ x^y \ 2 \triangleright - 3 \ X + 1$$

  To express:

  $$-X^2 - 3 \cdot X + 1$$

  type:

  $$(-) \ X \ x^y \ 2 \triangleright\triangleright - 3 \ X + 1$$

  In effect, you must select $-X^2$ before typing the rest.

- Example 3

  To express:

  $$\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}$$

  Here, the tree ends in a "+", and there are four branches; each of these branches ends in a "÷" and has two leaves.

  Press CAS on the menu bar to open the Equation Editor, then type the first branch:

  $$1 \div 2$$

  then select this branch with

  $$\triangleright$$

  Then, type

  $$+$$

  and the second branch:

  $$1 \div 3$$

then select this branch with

$$\triangleright$$

Then, type

$$+$$

and the third branch:

$$1 \div 4$$

then select this branch with

$$\triangleright$$

Then, type

$$+$$

and the fourth branch:

$$1 \div 5$$

then select this branch with

$$\triangleright$$

At this point, the desired expression

$$\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}$$

is in the Equation Editor, and $\frac{1}{5}$ is selected.

To move back through the tree and select:

$$\frac{1}{3}+\frac{1}{4}$$

first type:

$$\triangleleft\triangleleft$$

to select the 1/3, then press

SHIFT $\triangleright$

which enables you to select two contiguous branches, the one already selected and the one to the right of it, like this:

$$\frac{1}{3}+\frac{1}{4}$$

If you want, you can calculate the selected part by pressing ENTER.

This produces:

$$\frac{1}{2}+\frac{7}{12}+\frac{1}{5}$$

with the $\frac{7}{12}$ selected.

If you want to perform the partial calculation

$$\frac{1}{2}+\frac{1}{5}$$

you must first perform a permutation so that the $\frac{1}{2}$ and the $\frac{1}{5}$ are side by side. To do this, type:

SHIFT $\triangleleft$

which exchanges the selected element with its neighbour to the left. This produces:

$$\frac{7}{12}+\frac{1}{2}+\frac{1}{5}$$

with the $\frac{7}{12}$ still selected. Press

$\triangleright$ SHIFT $\triangleright$

to select

$$\frac{1}{2}+\frac{1}{5}$$

Pressing ENTER then produces the result.

Summing up: SHIFT $\triangleright$ enables you to select the selected element and its neighbour to the right

SHIFT $\triangleleft$ enables you to exchange the selected element with its neighbour to the left

The selected element remains selected, even if you move it.

3.1.3 How to modify an expression

If you're typing an expression, the DEL key enables you to erase what you've typed.

If you're selecting, you can:

- Cancel the selection without deleting the expression, by typing

  DEL

  The cursor moves to the end of the deselected portion.
- Replace the selection with an expression, just by typing the desired expression
- Transform the selected expression by applying a CAS function to it: you call the function via one of the CAS menu options
- Delete the selected expression by typing:

  ALPHA SHIFT DEL (ALPHA CLEAR)
- Delete a selected unary operator—the top of the tree—by typing:

  SHIFT DEL (CLEAR)

  For example, to replace SIN(expr) with COS(expr), select SIN(expr), then press SHIFT DEL and type COS.
- Delete a binary operator by editing the expression: you select

  Edit expr.

  from the TOOL menu on the menu bar, and then make the correction.

  Delete a binary operator and its second operand by selecting the second argument and typing:

  SHIFT DEL (CLEAR)

  For example, if you have the expression 2 + 1 with 1 selected, typing SHIFT DEL deletes +1, leaving only the 2.
- Copy an element from the history by pressing HOME. In the history, pressing ENTER or selecting ECHO on the menu bar inserts the copy where the cursor is, or in place of the selection. You can also use the commands Cut, Copy and Paste from the TOOL menu on the menu bar to delete, copy and paste expressions as you would with any text editor (cf. 2.4).

3.1.4 The cursor mode

The cursor mode enables you to select a large expression quickly. To go into cursor mode, select:

`Cursor mode` in the TOOL menu

then use the arrow keys to include your selection in a box (when you release the arrow key, the expression pointed to by the cursor is enclosed).

Then, press ENTER to select the contents of the box.

3.1.5 To see everything

By selecting `Change font` from the TOOL menu on the menu bar, you can increase or decrease the font size of the expression. This enables you to view a large expression in its entirety when you need to.

If this is insufficient to see the whole expression, then you'll need to go into cursor mode:

`Cursor mode` in the TOOL menu, then use the arrow key ▷

Then press:

the VIEWS key, then use the arrow key ▷.

3.2 Accessing the CAS functions

While you are in the Equation Editor, you can access all CAS functions, and you can access them in various ways.

General principle:

When you have written an expression in the Equation Editor, all you have to do is press ENTER to evaluate the selection (or the entire expression, if nothing is selected).

3.2.1 How to type ∫ and Σ

∑ is found on the keyboard; all you have to do is type:

SHIFT + (∑)

The symbol ∫ is also found on the keyboard; it can be produced by typing:

SHIFT d/dX (∫)

The symbols $\int$ and $\sum$ are treated as prefix functions with multiple arguments.

$\int$ and $\sum$ are automatically placed *before* the selected element, if there is one (hence "prefix functions").

You can move the cursor with the arrow keys:

▷◁

Enter the expressions according to the rules of selection explained earlier, but you must first go into selection mode by pressing △ .

Note: Do not use the index *i* to define a summation, because *i* designates the complex-number solution of $x^2 + 1 = 0$.

In numeric mode, $\sum$ performs approximate calculations.

For example:

$$\sum_{K=0}^{4} \frac{1}{K!} = 2.70833333334$$

whereas

$$1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} = \frac{65}{24}$$

The symbol "!" is obtained by typing SHIFT × .

You should note that $\sum$ can symbolically calculate summations of rational fractions and hypergeometric series that allow a discrete primitive.

Example:

If you type:

$$\sum_{K=1}^{\infty} \frac{1}{K \cdot (K+1)}$$

then select the entire expression and press ENTER, you obtain:

1

3.2.2 How to call infix functions

These functions are typed *between* their arguments—for example,

AND | MOD

You can either:

- type them in Alpha mode (using AND MOD), then type the arguments, OR
- call them by selecting a CAS menu option or by pressing a key, provided you've already written and selected the first argument.

You move from one argument to the other by using the arrow keys ▷◁ .

The comma "," enables you to write a complex number:

when you type `1+2·i` or `(1,2)`, the parentheses are automatically placed when you type the comma.

If you want to type `(-1,2)`, you must be sure to select `-1` before you type the comma.

3.2.3 How to call prefix functions

These functions are typed *before* their arguments (which is the usual case). To call a prefix function:

- you can type the first argument, select it, then call the function using a menu, OR
- you can call the function using a menu or a keystroke in Alpha mode, then type the arguments.

The following example illustrates the different ways of calling a prefix function.

Example:

Say you want to factor the expression $x^2 - 4$, then find its value for $x = 4$. You know that FACTOR is the function for factoring, and that this function is found in the ALGB menu.

You also know that SUBST is the function for substituting a value for a variable in an expression, and that this function is found in the ALGB menu as well.

First possibility: Function call, then arguments

Press the F2 key to activate the ALGB on the menu bar, then highlight FACTOR and press ENTER.

FACTOR (◂) is displayed in the editor, with the cursor between the parentheses. Type your expression, using the rules of selection given earlier:

X X^y 2 ▷– 4 ▷▷▷

The following is now selected:

FACTOR $(X^2 – 4)$

Pressing ENTER then produces the result:

$(X + 2)\cdot(X – 2)$

The result is selected, and replaces the command.

You don’t see this, but after each ENTER, the display is written to the history--so in this case, FACTOR$(X^2 – 4)$ and $(X + 2)\cdot(X – 2)$ are written to the history.

At this point, you can erase the preceding result with ALPHA SHIFT DEL (CLEAR), because the result is selected.

Press the key that activates ALGB on the menu bar, then highlight SUBST and press ENTER.

SUBST(◂ , •)

is displayed in the editor, with the cursor between the parentheses at the location of the first argument.

Type your expression, using the rules of selection given earlier:

Note: Here, SUBST has two arguments, so you must go into selection mode using △ :

X x^y 2 △△– 4 ▷X = 4 ▷▷

The following is now selected:

SUBST$(X^2 – 4, X = 4)$

Pressing ENTER then produces the result:

$4^2 – 4$

The result is selected, and replaces the command. Pressing ENTER then yields the simplified result:

12

Naturally, SUBST$(X^2 – 4, X = 4)$, $4^2 – 4$ and 12 are all written to the history.

Remark:

When you call a CAS function that has arguments, you can type it in Alpha mode with its parentheses.

Second possibility: Arguments, then function call

First of all, type the expression and select it using the rules of selection given earlier.

In this case, type:

X x^y 2 ▷– 4 ▷▷

Then call FACTOR:

Press the key that activates ALGB on the menu bar, then highlight FACTOR and press ENTER.

This produces:

FACTOR$(X^2 – 4)$

Pressing ENTER then yields the result:

$(X + 2) \cdot (X – 2)$

The result is selected, and replaces the command.

Naturally, FACTOR$(X^2 – 4)$ and $(X + 2)\cdot(X – 2)$ are both written to the history.

Recall now that because your result is selected, you can apply another command to it.

At this point, then, call SUBST: press the key that activates ALGB on the menu bar, then highlight SUBST and press ENTER.

SUBST($(X + 2) \cdot (X – 2)$, ◂)

is displayed in the editor, with your expression as the first argument between the parentheses, and with the cursor at the location of the second argument.

All you have to do then is type:

X = 4 ▷▷, followed by ENTER.

This produces:

$(4 + 2) \cdot (4 – 2)$

Pressing ENTER then yields:

12

Naturally, SUBST$(X^2 – 4, X = 4)$, $(4 + 2) \cdot (4 – 2)$ and 12 are all written to the history.

Remark:

If you call a CAS function while you’re writing an expression, whatever is currently selected is copied into the function’s first or “main” argument. If nothing is selected, the cursor is placed at the appropriate location for completing the arguments.

3.3 Variables

You can store objects in variables, then access an object by using the name of its variable.

Notes:

1. Variables used in CAS cannot be used in HOME, and vice versa.
2. In HOME or in the program editor, use STO▷ (notated in here as STO▷or ▷) to store an object in a variable.
3. In CAS, use the STORE command (cf. 3.3.2) to store a value in a variable.
4. The VARS key displays a menu that contains all the available variables.

   Pressing this key while you are in HOME displays the names of the variables defined in HOME and in the aplets.

   Pressing this key while you are in the Equation Editor displays the names of the variables defined in CAS.

3.3.1 STO▷

STO▷ enables you to store an object in a HOME variable.

The names of the numeric HOME variables are the 26 letter of the alphabet, and the names of the symbolic HOME variables are S1S5, E0..E9, s1..s5, or n1..n5 Note: The variables A..Z are always available and always contain a real value.

For example, to use STO▷ on the HOME or program editor menu bar, type:

1 STO▷ A

which is displayed on the screen as:

1 ▷ A and has the effect of overwriting the preceding value of A with 1.

A is evaluated and designates the contents of A.

Remark:

The symbolic HOME variable S1 serves as the current variable while you are using certain CAS functions in HOME.

Example: Even if there exists X in VX, in HOME you type:

DERVX($S1^2 + 2 \times S1$)

to get $2 \times S1 + 2$.

3.3.2 STORE

In CAS, it's necessary to use the STORE command to store an object in a variable, or to use the VARS key in the Equation Editor (and then choose NEW or EDIT on the menu bar; cf. 2.5.3).

All you need to provide is the name of the variable.

STORE is found in the ALGB menu on the Equation Editor menu bar.

Example:

Type:

STORE($X^2 – 4$, ABC)

Or, type:

$X^2 – 4$

then select it and call STORE,

then type ABC.

ENTER confirms the definition of the variable ABC.

To destroy the variable, use the VARS key in the Equation Editor (then choose PURGE on the menu bar; cf. 2.5.3), or invoke the UNASSIGN command on the ALGB menu by typing (for example):

UNASSIGN(ABC)

3.3.3 Predefined CAS variables

VX contains the name of the current symbolic variable.

Generally this is X, so you should not use X as the name of a numeric variable, or erase the contents of X with the UNASSIGN command in the ALGB menu after having done a symbolic calculation (by typing, for example, UNASSIGN(X)).

EPS contains the value of epsilon used in the EPSX0 command (cf. 4.13.2).

MODULO contains the value of *p* for performing symbolic calculations in *Z/p.Z*. You can change the value of *p* either with the MODSTO command in the MODULAR menu (by typing, for example, MODSTO(13) to give *p* a value of 13), or with CFG in the CAS menus.

PERIOD must contain the period of a function before you can find its Fourier coefficients (cf. 4.11.6).

PRIMIT contains the primitive of the last integrated function.

REALASSUME contains a list of the names of the symbolic variables that are considered reals. If you've chosen the `Cmplx vars` option on the CFG configuration menu, these are by default:

X, Y, t, S1, and S2, as well as any integration variables that are in use.

Of course, if you've chosen the `Real vars` option on the CFG configuration menu, all symbolic variables are considered reals (cf. 4.1.1).

You can also use an assumption to define a variable such as X>1.

In a case like this, you use the ASSUME(X>1) command to make REALASSUME contain X>1.

The command UNASSUME(X) destroys all the assumptions we've previously made about X.

To see all these variables, as well as those that you've defined in CAS, press VARS in the Equation Editor (cf. 2.5.3).

4 Computer Algebra Functions

4.1 CAS Toolbar

Only the TOOL menu contains commands; the other menus are used for configuration and to contain the algebraic functions that can be written in Alpha mode.

4.1.1 CFG

All the menus except TOOL display the state of your configuration and enable you to change it.

For example, say that you see the following on the first line of a menu:

CFG: R = X S

This means that (1) you are in exact-real mode, (2) X is the current variable, and (3) you are working in Step by step mode (S).

If you highlight CFG and press OK, a menu is displayed with this at its head:

R = STEP ↑ X 13 ||

This means that (1) you are in exact-real mode, (2) Step by step mode is selected, (3) polynomials are written with their terms in ascending order by exponent, (4) X is the current variable, (5) modular calculations are carried out in Z/13Z (p = 13), and (6) you are working in `Rigorous` mode (that is, using absolute values).

You can change this configuration by selecting any of the following:

`Quit config` (when you're finished making changes)

`Complex` (or `Real`)

`Approx` (or `Exact`)

`Direct` (or `Step/Step` if you want to work in Step by step mode)

`1 + x + x²` (or `x² + x +1` ; how polynomials will appear)

`Sloppy` (or `Rigorous`, if you want to work in absolute values)

`Num. factor` (or `Symb factor`)

`Cmplx vars` (or `Real vars` if you want all symbolic variables to be treated as reals; see 3.3.3)

`English` (or `Français` if you want the line-based help to be in French)

`Default cfg` (configuration R = STEP ↓ X 13 ||).

Press OK to validate each of your choices.

Pressing CANCEL takes you out of the CFG menu (as does choosing QUIT and confirming it with OK).

The name of the current variable contained in VX, as well as the value of the variable MODULO, can be changed by means of the SHIFT SYMB (SETUP) keystroke, or by using the VARS key (see 2.5.6 and 2.5.3).

Remark: In CAS, angles are always expressed in radians. When you are the calculator HOME screen, you can use the HOME MODES menu (the SHIFT HOME keystrokes) to change this default.

4.1.2 TOOL

The functions contained in the TOOL menu are described in section 2.4.

```
Cursor mode
Edit expr.
Change Font
Cut
Copy
Paste
```

4.1.3 ALGB

```
COLLECT
DEF
EXPAND
FACTOR
PARTFRAC
QUOTE
STORE
|
SUBST
TEXPAND
UNASSIGN
```

4.1.4 DIFF&INT

```
DERIV
DERVX
DIVPC
FOURIER
IBP
INTVX
LIMIT
PREVAL
RISCH
SERIES
TABVAR
TAYLOR0
TRUNC
```

4.1.5 REWRITE

```
DISTRIB
EPSX0
EXPLN
EXP2POW
FDISTRIB
LIN
LNCOLLECT
POWEXPAND
SINCOS
SIMPLIFY
XNUM
XQ
```

4.1.6 SOLVE

```
DESOLVE
ISOLATE
LDEC
LINSOLVE
SOLVE
SOLVEX
```

4.1.7 TRIG

```
ACOS2S
ASIN2C
ASIN2T
ATAN2S
HALFTAN
SINCOS
TAN2CS2
TAN2SC
TAN2SC2
```

```
TCOLLECT
TEXPAND
TLIN
TRIG
TRIGCOS
TRIGSIN
TRIGTAN
```

4.1.8 The MATH Key

In addition to the menus already mentioned (ALGEBRA DIFF&INT REWRITE TRIG SOLVE), the following are also available:

Complex … (i ABS ARG CONJ DROITE FLOOR IM MOD – RE SIGN)

Constant … (e *i* ∞ π)

Hyperb … (ACOSH ASINH ATANH COSH SINH TANH)

Integer … (DIVIS EULER FACTOR GCD IEGCD IQUOT IREMAINDER ISPRIME? LCM NEXTPRIME PREVPRIME)

Modular … (ADDTMOD DIVMOD EXPANDMOD FACTORMOD GCDMOD INVMOD MODSTO MULTMOD POWMOD SUBTMOD)

Polynom … (EGCD FACTOR GCD HERMITE LCM LEGENDRE PARTFRAC PROPFRAC PTAYL QUOT REMAINDER TCHEBYCHEFF)

Tests … (ASSUME UNASSUME > ≥ < ≤ = = ≠ AND OR NOT IFTE)

Refer to section 2.4 and 2.5.1 for descriptions of these menus.

4.2 Step by Step Mode

You choose Step by step mode (Step/step, abbreviated S) when you want to see the details of the calculations.

The details of the calculations are displayed on the screen; you can view the next step by pressing OK.

When the screen is not big enough to display all the information, directional arrows ∏ ↓→ ° appear on the edge of the screen. You can then scroll the screen to see more information by using the arrow keys (△▽ ▷).
If you don't need to see the details of the calculations, choose Direct mode (abbreviated D).

4.3 General Use

The calculator can manage integers with unlimited precision, such as the following:

$$100!$$

(The symbol "!" is obtained by typing SHIFT-×.)

The decimal value of 100! is very large, but you can view it by using the VIEWS key.

4.3.1 DEF

As a further example:

Calculate the first six Fermat numbers $F_k = 2^{2^k} + 1$ for $k = 1..6$ and say whether they're prime.

Typing the formula

$$2^{2^2} + 1$$

gives a result of 17. You can then invoke the ISPRIME?( ) command, which is found in the MATH key's Integer menu.

The response is 1., which means TRUE. Using the history (which you access by pressing the HOME key), ECHO the expression $2^{2^2} + 1$ into the Equation Editor and change it to:

$$2^{2^3} + 1$$

Or better, define a function F(K) by selecting DEF from the ALGB menu (on the menu bar), and typing:

$$\mathrm{DEF}(\mathrm{F}(\mathrm{K}) = 2^{2^{\mathrm{K}}} + 1)$$

The response is $2^{2^K} + 1$, and F is now listed amongst the variables (which you can verify using the VARS key.)

For $K = 5$, you then type:

$$\mathrm{F}(5)$$

which gives:

$$4294967297$$

You can factor F(5) with FACTOR, which you'll find in the ALGB menu on the menu bar.

Typing:

FACTOR(F(5))

gives

641·6700417

F(6) gives:

18446744073709551617

Using FACTOR to factor it then yields:

274177·67280421310721

Note: Pay careful attention to the position of the dot in

$$2.5=\frac{5}{2}$$

versus

$$2\cdot 5=10$$

4.4 INTEGERS (and Gaussian Integers)

All the functions in this section are found in the MATH key's INTEGER menu, except MOD, which is on the MATH key's Complex menu.

For certain functions, you can use Gaussian integers (numbers of the form $a + b\cdot i$, where a and b are integers) in the place of integers.

4.4.1 DIVIS

DIVIS gives a list of the divisors of a number.

Keying in

DIVIS(12)

gives:

12 OR 6 OR 3 OR 4 OR 2 OR 1

4.4.2 EULER

EULER returns the Euler index for a whole number.

EULER(n) is equal to the number of whole numbers less than n and prime with n.

Typing:

EULER(21)

gives:

12

In other words:

E={2,4,5,7,8,10,11,13,15,16,17,19} is the set of whole numbers less than 21 and prime with 21. There are 12 members of the set, so E=12.

4.4.3 FACTOR

FACTOR decomposes an integer into its prime factors.

Typing:

FACTOR(90)

gives:

$2\cdot 3^2\cdot 5$

4.4.4 GCD

GCD returns the greatest common divisor of two whole numbers.

Typing:

GCD(18,15)

gives:

3

In Step by step mode, typing:

GCD(78,24)

gives:

78 mod 24 = 6
24 mod 6 = 0
Result 6

Pressing ENTER then causes 6 to be written to the Equation Editor.

4.4.5 IEGCD

IEGCD(A,B) returns the value of Bézout's Identity for two integers.

In other words, IEGCD(A,B) returns U AND V = D, with U, V, and D such that:

A·U+B·V=D and D=GCD(A,B).
Typing:

IEGCD(48,30)

gives

2 AND –3 = 6

In other words:

$$2\cdot 48 + (-3)\cdot 30 = 6$$

In Step by step mode, we get:

[z,u,v]: z=*u**48+*v**30
[48,1,0]
[30,0,1] * –1
[18,1, –1]* –1
[12,1, –2]* –1
[6,2, –3]* –2
Result: [6,2, –3]

Pressing the ENTER key then causes

2 AND – 3 = 6

to be written to the Equation Editor.

4.4.6 IQUOT

IQUOT returns the integer quotient of the Euclidean division of two integers.

Typing:

IQUOT(148,5)

gives:

29

In Step by step mode, the division is carried out as if in longhand:

```
148 | 5
 48 | - - -
  3 | 29
```

Press OK to execute the division step by step, then press ENTER to write the result (29) to the Equation Editor.

4.4.7 IREMAINDER MOD

IREMAINDER returns the integer remainder from the Euclidean division of two integers.

IREMAINDER is found in the MATH key's Integer menu, and MOD is found in the MATH key's Complex menu.

Typing:

IREMAINDER(148,5)

or

148 MOD 5

gives:

3

IREMAINDER works with integers or with Gaussian integers, which is what distinguishes it from MOD.

Example: typing IREMAINDER($2+3\cdot i$,$1+i$) returns i

MOD accepts real numbers (7.5 mod 2 = 1.5), but not Gaussian integers.

Try calculating:

IREMAINDER(148!,5!+2)

(The symbol "!" is obtained by typing SHIFT-×.)

In Step by step mode, the division is carried out as if in longhand, using the so-called "gallows" algorithm (see 4.4.6 for an example).

4.4.8 ISPRIME?

ISPRIME?(N) returns 1. (TRUE) if N is a pseudo-prime, and 0. (FALSE) if N is not prime.

Definition: For numbers less than 10^{14}, "pseudo-prime" and "prime" mean the same thing. But for numbers greater than 10^{14}, a "pseudo-prime" is a number with a large probability of being prime (cf. Rabin's Algorithm, section 7.6).

Typing:

ISPRIME?(13)

gives:

1.

Typing:

ISPRIME?(14)

gives:

0.

4.4.9 LCM

LCM returns the least common multiple of two integers.

Typing:

LCM(18,15)

gives:

90

4.4.10 NEXTPRIME

NEXTPRIME(N) returns the smallest pseudo-prime greater than N.

Typing:

NEXTPRIME(75)

gives:

79

4.4.11 PREVPRIME

PREVPRIME(N) returns the greatest pseudo-prime less than N.

Typing:

PREVPRIME(75)

gives:

73

4.5 Modular Calculations

All the functions in this section are found in the MATH key's MODULAR menu.

You can carry out calculations in modulo `p`—that is, in *Z/pZ* or in *Z/pZ*[*X*].

Note: For some commands, `p` must be prime.

All the examples in this section use a value for `p` of 13.

It's assumed that you have already typed:

MODSTO(13)

or that you have used the SHIFT SYMB (SETUP) keystroke to switch to MODULO 13.

The chosen representation is the symmetrical one (–1 in place of 6 modulo 7).

4.5.1 ADDTMOD

ADDTMOD performs an addition in *Z/pZ*[*X*].

Typing:

ADDTMOD(11X + 5,8X + 6)

gives:

6X – 2

4.5.2 DIVMOD

The arguments are two polynomials A[X] and B[X]. The result is a rationale fraction $\frac{A[X]}{B[X]}$ simplified as Z/pZ[X].

Typing:

DIVMOD($2X^2 + 5{,}5X^2 + 2X - 3$)

gives:

$$-\frac{4 \cdot X + 5}{3 \cdot X + 3}$$

4.5.3 EXPANDMOD

EXPANDMOD has as an argument a polynomial expression.

EXPANDMOD expands this expression in *Z/pZ*[*X*].

Typing:

EXPANDMOD(($2X^2 + 12$)·($5X - 4$))

gives:

$-(3X^3 - 5X^2 + 5X - 4)$

4.5.4 FACTORMOD

FACTORMOD has as an argument a polynomial.

FACTORMOD factors this polynomial in Z/pZ[X], providing that $p \leq 97$ and p is prime.

Typing:

FACTORMOD($-(3X^3 - 5X^2 +5X - 4$)

gives:

$-((3X - 5)(X^2 + 6))$

4.5.5 GCDMOD

GCDMOD has two polynomials as arguments.

GCDMOD calculates the GCD of the two polynomials in *Z/pZ*[*X*].

Typing:

GCDMOD($2X^2 + 5{,}5X^2 + 2X - 3$)

gives:

$-(6X-1)$

4.5.6 INVMOD

INVMOD has as an argument an integer.

INVMOD calculates the inverse of the integer in *Z/pZ*.

Typing:

INVMOD(5)

gives (since $5x - 5 = -25 = 1 \pmod{13}$):

-5

4.5.7 MODSTO

You use the MODSTO command to set the value of the MODULO variable `p`.

The examples in this section all use a value for `p` of 13—that is, they assume that you have already typed:

MODSTO(13)

4.5.8 MULTMOD

MULTMOD performs a multiplication in Z/pZ[X].

Typing:

MULTMOD($11X + 5{,}8X + 6$)

gives:

$-(3X^2 - 2X - 4)$

4.5.9 POWMOD

POWMOD(A,N) calculates A to the power of N in *Z/pZ[X]*, and POWMOD(A(X),N) calculates A(X) to the power of N in *Z/pZ[X]*.

The MODULO variable `p` must be a prime number less than 100.

Typing:

$$\mathrm{POWMOD}(11,195)$$

gives:

$$5$$

In effect: $11^{12}=1$ mod 13; $11^{195} = 11^3 = 5$ mod 13.

Typing:

$$\mathrm{POWMOD}(2X + 1,5)$$

gives:

$$6\cdot X^5 + 2\cdot X^4 + 2\cdot X^3 + X^2 - 3\cdot X + 1$$

since:

$10 = -3$ (mod 13) $40 = 1$ (mod 13) $80 = 2$ (mod 13) $32 = 6$ (mod 13)

4.5.10 SUBTMOD

SUBTMOD performs a subtraction in Z/pZ[X].

Typing:

$$\mathrm{SUBTMOD}(11X + 5, 8X + 6)$$

gives:

$$3X - 1$$

4.6 Rational Numbers

Calculate:

$$\frac{123}{12}+\frac{57}{21}$$

After you press ENTER, you get the result:

$$\frac{363}{28}$$

If you now invoke the XNUM function on the REWRITE menu, or if you press the NUM key, you get the following result:

$$12.9642857143$$

If you mix the two representations—for example:

$$\frac{1}{2}+0.5$$

the calculator asks to go into `approx` mode to do the calculation. After responding yes, you get:

$$1$$

You'll then need to go back into `exact` mode (CFG, etc…).

4.6.1 PROPFRAC

PROPFRAC is found in the MATH key's POLYNOMIAL menu.

$\mathrm{PROPFRAC}\left(\frac{A}{B}\right)$ writes the fraction $\frac{A}{B}$ in the form:

$$Q+\frac{R}{B} \quad \text{where } 0 \le R < B$$

Typing:

$$\mathrm{PROPFRAC}\left(\frac{43}{12}\right)$$

gives:

$$3+\frac{7}{12}$$

4.7 Real Numbers

Calculate:

$$\mathrm{EXP}(\pi\cdot\sqrt{20})$$

When you press the ENTER key, the response is:

$$\mathrm{EXP}(2 \cdot \sqrt{5} \cdot \pi)$$

If you then invoke the XNUM function on the REWRITE menu or press the NUM key, the result is:

$$1263794.7537$$

In the MATH key's Complex menu, you'll find the FLOOR and MOD functions, which are explained in the next two subsections.

4.7.1 FLOOR

FLOOR has as an argument a real number, and returns its whole part.

Typing:

$$\mathrm{FLOOR}(3.53)$$

gives:

$$3$$

4.7.2 MOD

MOD is an infix function that has two integers as arguments.

MOD returns the remainder of the Euclidean division of the arguments.

Typing:

$$3\ \mathrm{MOD}\ 2$$

produces the result:

$$1$$

4.8 Complex Numbers

Note: Complex numbers of the form $a + b \cdot i$, where a and b are real numbers, can be notated (a, b) or $a + b \cdot i$.

The operators +, –, *, /, ^ can be used.

Type:

$$(1+2 \cdot \mathrm{i})^2$$

Then press ENTER.

If you are not in `complex` mode, the calculator asks to change modes. After responding YES, you get:

$$-(3-4 \cdot \mathrm{i})$$

It's worth noting that this expression will not be simplified further: in exact mode, the result is always notated such that the real part of the complex number is positive.

In the MATH key's Complex menu you will find the following functions, all of which have complex numbers as parameters:

DROITE takes two complex numbers as parameters, z_1, z_2. It then returns the equation of the line through the Cartesian points, z_1 and z_2.

ARG to determine the argument of the parameter,

ABS to determine the modulus of the parameter,

CONJ to determine the conjugate of the parameter,

RE to determine the real part of the parameter,

IM to determine the imaginary part of the parameter,

- to determine the opposite of the parameter,

SIGN to determine the quotient of the parameter divided by its modulus.

4.8.1 ARG

Typing:

$$\mathrm{ARG}(3+4 \cdot \mathrm{i})$$

produces (given that in CAS, you're working in Radians):

$$\mathrm{ATAN}\left(\frac{4}{3}\right)$$

Remark:

You can do the same calculation in HOME, but you get a numeric result (0.64250, if you're working in Radians).

In HOME, typing:

ARG(XQ(3 + 4·i))

produces:

$$\mathrm{ATAN}\left(\frac{4}{3}\right)$$

4.8.2 CONJ

Typing:

CONJ(1 + 2·i)

produces the result:

1 – 2·i

Note: If you chose `Real vars` from the CFG configuration menu, then CONJ(Z)=Z; if you chose `Cmplx vars`, CONJ(Z) will be different from Z as long as Z is not in the list that contains the variable REALASSUME. It's often preferable to write the expression as a quoted expression:

QUOTE(expression), to avoid having to rewrite it. For example, if you select `Real vars` and then type:

SUBST(QUOTE(CONJ(Z)), Z = 1+ i)

you get:

CONJ(1+ i)

whereas:

SUBST(CONJ(Z), Z = 1+ i)

produces:

1 + i

Of course, if you've selected `Cmplx vars`, and Z is not in the list that contains the variable REALASSUME, then typing:

SUBST(CONJ(Z), Z = 1+ i)

gives:

CONJ(1+ i)

4.8.3 DROITE

Typing:

DROITE((1,2),(0,1))

or:

DROITE(1 + 2·i, i)

returns:

$$Y = X - 1 + 2$$

Then, ENTER produces:

$$Y = X + 1$$

4.9 Algebraic Expressions

All functions in this section can be found in the ALGB menu on the menu bar.

4.9.1 COLLECT

COLLECT has an expression as a parameter.

COLLECT combines like terms, and factors the expression over the integers.

Example:

To factor over the integers:

$$x^2 - 4$$

type:

COLLECT(X^2 – 4)

which gives:

(X + 2)· (X – 2)

To factor over the integers:

$$X^2 - 2$$

type:

COLLECT(X^2 – 2)

which gives:

$$X^2 - 2$$

4.9.2 EXPAND

EXPAND has an expression as a parameter.

EXPAND expands and simplifies this expression.

Typing:

$$\mathrm{EXPAND}((X^2+\sqrt{2}\cdot X+1)\cdot(X^2-\sqrt{2}\cdot X+1))$$

gives:

$$X^4+1$$

4.9.3 FACTOR

FACTOR has an expression as a parameter.

FACTOR factors this expression.

Example:

To factor

$$X^4+1$$

Key in:

$$\mathrm{FACTOR}(X^4+1)$$

FACTOR can be found in the ALGB menu.

In real mode, the result is:

$$(X^2+\sqrt{2}\cdot X+1)\cdot(X^2-\sqrt{2}\cdot X+1)$$

In complex mode (using CFG), the result is:

$$\frac{(2X+(1+i)\cdot\sqrt{2})\cdot(2X-(1+i)\cdot\sqrt{2})\cdot(2X+(1-i)\sqrt{2})\cdot(2X-(1-i)\cdot\sqrt{2}}{16}$$

4.9.4 |

| is an infix operator used to substitute a value for a variable in an expression (a bit like the following function SUBST).

Typing:

$$X^2-1\big|_{X=2}$$

gives:

$$2^2-1$$

4.9.5 SUBST

SUBST has two parameters: an expression dependent on a parameter, and an equality (parameter=substitute value).

SUBST substitutes the specified value for the variable in the expression.

Typing:

$$\mathrm{SUBST}(A^2+1,A=2)$$

gives:

$$2^2+1$$

4.10 Polynomials

All functions (except DEGREE) in this section can be found in the MATH key's Polynom. menu.

4.10.1 DEGREE

DEGREE has as an argument a polynomial in the current variable.

DEGREE returns the degree of this polynomial.

Note: The degree of a null polynomial is –1. The DEGREE command must be entered using the ALPHA keys.

Typing:

$$\mathrm{DEGREE}(X^2+X+1)$$

returns:

$$2$$

4.10.2 EGCD

This function returns Bézout's Identity, the EGCD (Extended Greatest Common Divisor). In other words, EGCD(A(X), B(X)) returns U(X) AND V(X) = D(X), with D, U, and V such that:

$$D(X) = U(X) \cdot A(X) + V(X) \cdot B(X)$$

Typing:

$$\text{EGCD}(X^2 + 2 \cdot X + 1, X^2 - 1)$$

gives:

$$1 \text{ AND } -1 = 2 \cdot X + 2$$

Typing:

$$\text{EGCD}(X^2 + 2 \cdot X + 1, X^3 + 1)$$

gives:

$$-(X - 2) \text{ AND } 1 = 3 \cdot X + 3$$

4.10.3 FACTOR

The FACTOR has a polynomial as an argument.

FACTOR factors this polynomial.

Typing:

$$\text{FACTOR}(X^2 - 2)$$

gives:

$$(X + \sqrt{2}) \cdot (X - \sqrt{2})$$

Typing:

$$\text{FACTOR}(X^2 + 2 \cdot X + 1)$$

gives:

$$(X + 1)^2$$

Typing:

$$\text{FACTOR}(X^4 - 2 \cdot X^2 + 1)$$

gives:

$$(X - 1)^2 \cdot (X + 1)^2$$

Typing:

$$\text{FACTOR}(X^3 - 2 \cdot X^2 + 1)$$

gives:

$$\frac{(X - 1) \cdot (2X - 1 + \sqrt{5}) \cdot (2X - (1 + \sqrt{5}))}{4}$$

4.10.4 GCD

GCD returns the GCD (Greatest Common Divisor) of two polynomials.

Typing:

$$\text{GCD}(X^2 + 2 \cdot X + 1, X^2 - 1)$$

gives:

$$X + 1$$

4.10.5 HERMITE

HERMITE has as an argument a whole number n. HERMITE returns the HERMITE polynomial of degree n. This is a polynomial of the following type:

$$H_n(x)(-1)^n \cdot e^{\frac{x^2}{2}} \cdot \frac{d^n}{dx^n}\left(e^{-\frac{x^2}{2}}\right)$$

That means that for $n \geq 0$:

$$H''_n(x) - xH'_n(x) + nH_n(x) = 0$$

and for $n \geq 1$:

$$H_{n+1}(x) - xH_n(x) + nH_{n-1}(x) = 0$$

$$H'_n(x) = nH_{n-1}(x)$$

Typing:

$$\text{HERMITE}(6)$$

produces the result:

$$64 \cdot X^6 - 480 \cdot X^4 + 720 \cdot X^2 - 120$$

4.10.6 LCM

LCM returns the LCM (Least Common Multiple) of two polynomials.

Typing:

$$\mathrm{LCM}(X^2 + 2\cdot X + 1, X^2 - 1)$$

gives:

$$(X^2 + 2\cdot X + 1)\cdot(X - 1)$$

4.10.7 LEGENDRE

LEGENDRE has as an argument a whole number n. LEGENDRE returns the polynomial L_n, a non-null solution of the differential equation:

$(x^2 - 1)\cdot y'' - 2\cdot x\cdot y' - n(n + 1)\cdot y = 0$
For $n \geq 0$, we have the Rodriguès Formula:

$$L_n(x) = \frac{1}{n!2^n}\frac{d^n}{dx^n}(x^2 - 1)^n$$

and for $n \geq 1$, we have:

$$(n + 1)L_{n+1}(x) = (2n + 1)xL_n(x) - nL_{n-1}(x)$$

Typing:

$$\mathrm{LEGENDRE}(4)$$

gives:

$$\frac{35\cdot X^4 - 30\cdot X^2 + 3}{8}$$

4.10.8 PARTFRAC

To decompose into simple elements the rational fraction:

$$\frac{x^5 - 2\cdot x^3 + 1}{x^4 - 2\cdot x^3 + 2\cdot x^2 - 2\cdot x + 1}$$

you use the PARTFRAC command.

Type:

$$\mathrm{PARTFRAC}\left(\frac{X^5 - 2\cdot X^3 + 1}{X^4 - 2\cdot X^3 + 2\cdot X^2 - 2\cdot X + 1}\right)$$

In real mode, this produces:

$$X + 2 + \frac{X - 3}{2X^2 + 2} + \frac{-1}{2X - 2}$$

In complex mode, this produces:

$$X + 2 + \frac{\frac{1 - 3\cdot i}{4}}{X + i} + \frac{\frac{-1}{2}}{X - 1} + \frac{\frac{1 + 3\cdot i}{4}}{X - i}$$

4.10.9 PROPFRAC

PROPFRAC has as an argument a rational fraction.

PROPFRAC rewrites this rational fraction so as to bring out its whole-number part.

PROPFRAC(A(X)) writes the rational fraction $\frac{A[X]}{B[X]}$ in the form:

$$Q[X] + \frac{R[X]}{B[X]}$$

where R[X] = 0, or 0 ≤ *deg*(R[X] < *deg* (B[X]).

Typing:

$$\mathrm{PROPFRAC}\left(\frac{(5\cdot X + 3)\cdot(X - 1)}{X + 2}\right)$$

gives:

$$5 \cdot X - 12 + \frac{21}{X+2}$$

4.10.10 PTAYL

PTAYL rewrites a polynomial $P[X]$ in order of its powers of $X - a$.

PTAYL takes two parameters: a polynomial P and a number a.

Typing:

$$\text{PTAYL}(X^2 + 2 \cdot X + 1, 2)$$

produces the polynomial $Q[X]$:

$$X^2 + 6 \cdot X + 9$$

Note that:

$$P(X) = Q(X - 2)$$

4.10.11 QUOT

QUOT returns the quotient of two polynomials (divided in decreasing order by exponent).

Typing:

$$\text{QUOT}(X^2 + 2 \cdot X + 1, X)$$

gives:

$$X + 2$$

4.10.12 REMAINDER

REMAINDER returns the remainder from the division of two polynomials (divided in decreasing order by exponent).

Typing:

$$\text{REMAINDER}(X^3 - 1, X^2 - 1)$$

gives:

$$X - 1$$

4.10.13 TCHEBYCHEFF

TCHEBYCHEFF has as an argument an integer n.

If $n > 0$, TCHEBYCHEFF returns the polynomial T_n such that:

$$T_n[x] = \cos(n \cdot \arccos(x))$$

For $n \geq 0$, we have:

$$T_n[x[= \sum_{k=0}^{[n/2]} C_n^{2k} (x^2 - 1)^k x^{n-2k}$$

For $n \geq 0$ we also have:

$$(1 - x^2)T''_n(x) - xT'_n(x) + n^2 T_n(x) = 0$$

For $n \geq 1$, we have:

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

If $n < 0$, TCHEBYCHEFF returns the 2nd-species Tchebycheff polynomial:

$$T_n[x] = \frac{\sin(n \cdot \arccos(x))}{\sin(\arccos(x))}$$

Typing:

$$\text{TCHEBYCHEFF}(4)$$

gives:

$$8 \cdot X^4 - 8 \cdot X^2 + 1$$

In effect:

$\cos(4 \cdot x) = Re((\cos(x) + i \cdot \sin(x))^4)$
$\cos(4 \cdot x) = \cos(x)^4 - 6 \cdot \cos(x)^2 \cdot (1 - \cos(x)^2) + ((1 - \cos(x)^2)^2$
$\cos(4 \cdot x) = T_4(\cos(x))$
Typing:

$$\text{TCHEBYCHEFF}(-4)$$

gives:

$$8 \cdot X^3 - 4 \cdot X$$

In effect:

$\sin(4 \cdot x) = \sin(x) \cdot (8 \cdot \cos(x)^3 - 4 \cdot \cos(x))$

4.11 Functions

All the functions in this section can be found in the DIFF menu on the menu bar, except (1) DEF, which is in the ALGB menu, and (2) IFTE, which is in the MATH key's Tests menu.

4.11.1 DEF

DEF has as an argument an equality between (1) the name of a function (with parentheses containing the variable), and (2) an expression defining the function.

DEF defines this function and returns the equality.

Typing:

$$\mathrm{DEF}(\mathrm{U}(\mathrm{N}) = 2^{\mathrm{N}} + 1)$$

produces the result:

$$\mathrm{U}(\mathrm{N}) = 2^{\mathrm{N}} + 1$$

Typing:

$$\mathrm{U}(3)$$

then returns:

$$9$$

4.11.2 IFTE

IFTE has three arguments: a Boolean (note the use of "= =" to test for equality), and two expressions *expr*1, *expr*2.

IFTE evaluates the condition, then returns *expr*1 if the condition is true, or *expr*2 if the condition is false.

Typing:

$$\mathrm{STORE}(2,\mathrm{N})$$

$$\mathrm{IFTE}\left(N == 0,1,\frac{N+1}{N}\right)$$

produces the result:

$$\frac{3}{2}$$

It's easy to define functions using IFTE. For example:

$$\mathrm{DEF}\left(\mathrm{F(X)} = \mathrm{IFTE}(\mathrm{X} == 0,1,\frac{\mathrm{SIN(X)}}{\mathrm{X}}\right)$$

defines the function f such that:

$f(x) = 1$ if $x = 0$, and
$f(x) = \sin(x)/x$ if $x \neq 0$

4.11.3 DERVX

DERVX calculates the derivative of a function.

For example, given:

$$f(x) = \frac{x}{x^2-1} + \ln\left(\frac{x+1}{x-1}\right)$$

calculate the derivative of f.

Type:

$$\mathrm{DERVX}\left(\frac{\mathrm{X}}{\mathrm{X}^2-1} + \mathrm{LN}\left(\frac{\mathrm{X}+1}{\mathrm{X}-1}\right)\right)$$

—OR, if you've stored the definition of $f(x)$ in F—that is, if you've typed:

$$\mathrm{STORE}\left(\frac{\mathrm{X}}{\mathrm{X}^2-1} + \mathrm{LN}\left(\frac{\mathrm{X}+1}{\mathrm{X}-1}\right),\mathrm{F}\right)$$

then type:

$$\mathrm{DERVX(F)}$$

—OR, if you've defined $f(x)$ using DEF—that is, if you've typed:

$$\mathrm{DEF}\left(\mathrm{F(X)} = \frac{\mathrm{X}}{\mathrm{X}^2-1} + \mathrm{LN}\left(\frac{\mathrm{X}+1}{\mathrm{X}-1}\right)\right)$$

then type:

$$\mathrm{DERVX(F(X))}$$

The result is a complicated expression. Pressing ENTER simplifies it, giving:

$$-\frac{3\mathrm{X}^2-1}{\mathrm{X}^4-2\cdot\mathrm{X}^2+1}$$

4.11.4 DERIV

DERIV has two arguments: an expression (or a function) and a variable.

DERIV returns the derivative of the expression (or the function) with respect to the variable given as the second parameter (used for calculating partial derivatives).

Example:

Calculate:

$$\frac{\partial(x \cdot y^2 \cdot z^3 + x \cdot y)}{\partial z}$$

Typing:

DERIV(X·Y²·Z³ + X·Y, Z)

gives:

X·Y²·3·Z²

4.11.5 TABVAR

TABVAR has as a parameter an expression with a rational derivative.

TABVAR returns the variation table for the expression in terms of the current variable.

Typing:

TABVAR(LN(X) + X)

In Step by step mode, this gives:

F =: (LN(X) + X)

$F' =: \left(\frac{1}{X} + 1\right)$

$\rightarrow: \frac{X+1}{X}$

Variation table:

$$\begin{bmatrix} -\infty & ? & 0+0 & + & +\infty & X \\ ? & ? & -\infty & \uparrow & +\infty & F \end{bmatrix}$$

4.11.6 FOURIER

FOURIER has two parameters: an expression $f(x)$ and a whole number n.

FOURIER returns the Fourier coefficient c_n of $f(x)$, considered to be a function defined over interval $[0,T]$ and with period T (T being equal to the contents of the variable PERIOD).

If $f(x)$ is a discrete series, then:

$$f(x) = \sum_{n=-\infty}^{+\infty} c_n e^{\frac{2inx\pi}{T}}$$

Example: Determine the Fourier coefficients of a periodic function with period 2·π and defined over interval [0, 2·π] by $f(x) = x^2$.

Typing:

STORE(2·π, PERIOD)

FOURIER(X²,N)

we get, after replacing EXP(2·i·N·π) with 1 and simplifying:

$$\frac{2 \cdot \mathrm{i} \cdot \mathrm{N} \cdot \pi + 2}{\mathrm{N}^2}$$

so if N ≠ 0, then:

$$c_n = \frac{2 \cdot i \cdot N \cdot \pi + 2}{N^2}$$

Typing:

FOURIER(X²,0)

gives:

$$\frac{4 \cdot \pi^2}{3}$$

so if N = 0, then:

$$c_0 = \frac{4 \cdot \pi^2}{3}$$

4.11.7 IBP

IBP has two parameters: an expression of the form $u(x) \cdot v'(x)$, and $v(x)$.

IBP returns the AND of $u(x)\cdot v(x)$ and $-v(x)\cdot u'(x)$—that is, the terms that one must calculate when one performs a partial integration.

It remains then to calculate the integral of the second term of the AND, then add it to the first term of the AND to obtain a primitive of $u(x)\cdot v'(x)$.

Typing:

IBP(LN(X),X)

gives:

X·LN(X) AND – 1

One completes the integration by calling INTVX:

INTVX(X·LN(X) AND – 1)

which produces the result:

X·LN(X) – X

Remark: If the first IBP parameter is an AND of two elements, IBP concerns itself only with the second element of the AND, adding the integrated term to the first element of the AND (so that you can perform multiple IBPs in succession).

4.11.8 INTVX

INTVX calculates a primitive of its argument.

Exercise 1

Calculate a primitive of $\sin(x) \times \cos(x)$.

Typing:

INTVX(SIN(X)·COS(X))

gives, in Step by step mode:

COS(X)·SIN(X)
Int[u'*F(u)] with u=SIN(X)

Pressing OK then sends the result to the Equation Editor:

$$\frac{\mathrm{SIN(X)}^2}{2}$$

Exercise 2

.

Given:

$$f(x) = \frac{x}{x^2-1} + \ln\left(\frac{x+1}{x-1}\right)$$

Calculate a primitive of f.

Type:

$$\mathrm{INTVX}\left(\frac{\mathrm{X}}{\mathrm{X}^2-1} + \mathrm{LN}\left(\frac{\mathrm{X}+1}{\mathrm{X}-1}\right)\right)$$

—OR, if you've stored the definition of $f(x)$ in F, type:

INTVX(F)

—OR, if you've used DEF to define F(X)—that is, if you've already typed:

$$\mathrm{DEF}\left(\mathrm{F(X)} = \frac{\mathrm{X}}{\mathrm{X}^2-1} + \mathrm{LN}\left(\frac{\mathrm{X}+1}{\mathrm{X}-1}\right)\right)$$

then type:

INTVX(F(X))

The result in all cases is equivalent to:

$$X \cdot \mathrm{LN}\left(\frac{X+1}{X-1}\right) + \frac{3}{2} \cdot \mathrm{LN}(X-1) + \frac{3}{2} \cdot \mathrm{LN}(X+1)$$

Exercise 3

Calculate:

$$\int \frac{2}{x^6+2\cdot x^4+x^2}\,dx$$

Typing:

$$\mathrm{INTVX}\left(\frac{2}{\mathrm{X}^6+2\cdot \mathrm{X}^4+\mathrm{X}^2}\right)$$

gives the result:

$$-3\cdot \mathrm{ATAN(X)} - \frac{2}{\mathrm{X}} - \frac{\mathrm{X}}{\mathrm{X}^2+1}$$

Remark:

You can also type:

$$\int_{1}^{X} \frac{2}{X^6 + 2 \cdot X^4 + X^2} \, dX$$

which gives the same result plus a constant equal to:

$$\frac{3 \cdot \pi + 10}{4}$$

Exercise 4

Calculate:

$$\int \frac{1}{\sin(x) + \sin(2 \cdot x)} dx$$

Typing:

$$\mathrm{INTVX}\left(\frac{1}{\mathrm{SIN(X)} + \mathrm{SIN}(2 \cdot \mathrm{X})}\right)$$

gives the result:

$$\frac{1}{6} \cdot \mathrm{LN}(\mathrm{COS(X)} - 1) + \frac{1}{2} \cdot \mathrm{LN}(\mathrm{COS(X)} + 1) + \frac{-2}{3} \cdot \mathrm{LN}(2 \cdot \mathrm{COS(X)} + 1)$$

Remark: If the argument to INTVX is the AND of two elements, INTVX concerns itself only with the second element of the AND.

4.11.9 LIMIT

For $n > 2$ in the following expression, find the limit as x approaches 0:

$$\frac{n \cdot \tan(x) - \tan(n \cdot x)}{\sin(n \cdot x) - n \cdot \sin(x)}$$

You can use the LIMIT command to do this.

Typing:

$$\mathrm{LIMIT}\left(\frac{\mathrm{N} \cdot \mathrm{TAN(X)} - \mathrm{TAN}(\mathrm{N} \cdot \mathrm{X})}{\mathrm{SIN}(\mathrm{N} \cdot \mathrm{X}) - \mathrm{N} \cdot \mathrm{SIN(X)}}, 0\right)$$

gives:

$$2$$

For the following expression, find the limit as x approaches $+\infty$:

$$\sqrt{x + \sqrt{x + \sqrt{x}}} - \sqrt{x}$$

Typing:

$$\mathrm{LIMIT}(\sqrt{\mathrm{X} + \sqrt{\mathrm{X} + \sqrt{\mathrm{X}}}} - \sqrt{\mathrm{X}}, +\infty)$$

produces (after a short wait):

$$\frac{1}{2}$$

Note:

The symbol "∞" is obtained by typing SHIFT-0.

To obtain –∞, type:

(–) ∞

To obtain +∞, type:

(–) (–) ∞

You can also find the symbol "∞" in the MATH key's `Constant` menu.

4.11.10 LIMIT and ∫

For the following expression, determine the limit when a approaches $+\infty$:

$$\int_{2}^{a} \left(\frac{x}{x^2 - 1} + \mathrm{LN}\left(\frac{x+1}{x-1}\right)\right) dx$$

In the Equation Editor, type:

$$\int_{2}^{+\infty} \left(\frac{\mathrm{X}}{\mathrm{X}^2 - 1} + \mathrm{LN}\left(\frac{\mathrm{X}+1}{\mathrm{X}-1}\right)\right) d\mathrm{X}$$

Note: To obtain the symbol "+∞", type:

(–) (–) ∞ (SHIFT 0)

This produces:

$$+\infty - \frac{7 \cdot \mathrm{LN}(3)}{2}$$

or, after simplification:

$$+\infty$$

4.11.11 PREVAL

PREVAL has three parameters: an expression F(VX) dependent on the variable contained in VX, and two expressions A and B.

PREVAL (F(X),A,B) returns F(B)–F(A).

PREVAL is used for calculating an integral defined from a primitive: it evaluates this primitive between the two limits of the integral.

Typing:

$$\mathrm{PREVAL}\ (X^2 + X, 2,3)$$

gives:

$$6$$

4.11.12 RISCH

RISCH has two parameters: an expression and the name of a variable.

RISCH returns a primitive of the first parameter with respect to the variable specified in the second parameter.

Typing:

$$\mathrm{RISCH}((2\cdot X^2 +1)\cdot \mathrm{EXP}(X^2+1),X)$$

gives:

$$X\cdot \mathrm{EXP}(X^2 + 1)$$

Remark: If the RISCH parameter is the AND of two elements, RISCH concerns itself only with the second element of the AND.

4.12 Limited and Asymptotic Expansions

All functions in this section can be found in the DIFF menu on the menu bar.

It's customary to write the expansions in ascending order by exponent of the variable; you set this up by choosing $1 + x + x^2$... in CFG.

4.12.1 DIVPC

DIVPC has three arguments: two polynomials A(X), B(X) (where $B(0) \neq 0$), and a whole number *n*.

DIVPC returns the quotient Q(X) of the division of A(X) by B(X), in increasing order by exponent, and with $\deg(Q) \leq n$ or $Q = 0$.

$Q[X]$ is then the limited n^{th}-order expansion of $\frac{A[X]}{B[X]}$ in the vicinity of X = 0.

Typing:

$$\mathrm{DIVPC}(1 + X^2 + X^3, 1 + X^2,5)$$

gives:

$$1+X^3 - X^5$$

Note: When the calculator asks to go into "increasing powers" mode, respond `yes`.

4.12.2 LIMIT

LIMIT has two arguments: an expression dependent on a variable, and an equality (a variable = the value to which you want to calculate the limit). It is often preferable to use a quoted expression:

QUOTE(expression), to avoid rewriting the expression in normal form (i.e., not to have a rational simplification of the arguments) during the execution of the LIMIT command.

Typing, for example:

$$\mathrm{LIMIT}\left(\mathrm{QUOTE}\left((2X-1)\cdot \mathrm{EXP}\left(\frac{1}{X-1}\right)\right), X=+\infty\right)$$

gives:

$$+\infty$$

4.12.3 SERIES

Expansion in the vicinity of x = a

Example:

Give a limited 4^{th}-order expansion of $\cos(2\times x)^2$ in the vicinity of $x = \pi/6$.

For this you use the SERIES command.

Typing:

$$\mathrm{SERIES}\left(\mathrm{COS}(2\cdot \mathrm{X})^2, \mathrm{X}=\frac{\pi}{6}, 4\right)$$

gives:

$$\left.\frac{1}{4}-\sqrt{3}\cdot \mathrm{h}+2\cdot \mathrm{h}^2+\frac{8\cdot\sqrt{3}}{3}\cdot \mathrm{h}^3+\frac{-8}{3}\cdot \mathrm{h}^4+\mathrm{O}\left(\frac{\mathrm{h}^5}{4}\right)\right|_{\mathrm{h}=\mathrm{X}-\frac{\pi}{6}}$$

Expansion in the vicinity of x = +∞ or x = –∞

Example 1:

Give a 5^{th}-order expansion of arctan(x) in the vicinity of $x = +\infty$, taking as infinitely small $h=\frac{1}{x}$.

Typing:

$$\mathrm{SERIES}(\mathrm{ATAN}(\mathrm{X}),\mathrm{X}=+\infty, 5)$$

gives:

$$\left.\frac{\pi}{2}-\mathrm{h}+\frac{1}{3}\cdot \mathrm{h}^3+\frac{-1}{5}\cdot \mathrm{h}^5+\mathrm{O}\left(\frac{\pi \mathrm{h}^6}{2}\right)\right|_{\mathrm{h}=\frac{1}{\mathrm{X}}}$$

Example 2:

Give a 2^{nd}-order expansion of $(2x-1)e^{\frac{1}{x-1}}$ in the vicinity of $x = +\infty$, taking as infinitely small $h=\frac{1}{x}$.

Typing:

$$\mathrm{SERIES}\left((2\mathrm{X}-1)\cdot \mathrm{EXP}\left(\frac{1}{\mathrm{X}-1}\right), \mathrm{X}=+\infty, 3\right)$$

gives:

$$\left.\frac{12+6\cdot \mathrm{h}+12\cdot \mathrm{h}^2+17\cdot \mathrm{h}^3}{6\mathrm{h}}+\mathrm{O}\left(2\mathrm{h}^3\right)\right|_{\mathrm{h}=\frac{1}{\mathrm{X}}}$$

Example 3:

Give a 2^{nd}-order expansion of $(2x-1)e^{\frac{1}{x-1}}$ in the vicinity of $x = -\infty$, taking as infinitely small $h=\frac{-1}{x}$.

Typing:

$$\mathrm{SERIES}\left((2\mathrm{X}-1)\cdot \mathrm{EXP}\left(\frac{1}{\mathrm{X}-1}\right), \mathrm{X}=-\infty, 3\right)$$

gives:

$$\left.-\frac{12-6\cdot \mathrm{h}+12\cdot \mathrm{h}^2-17\cdot \mathrm{h}^3}{6\cdot \mathrm{h}}+\mathrm{O}\left(-\left(2\cdot \mathrm{h}^3\right)\right)\right|_{\mathrm{h}=\frac{-1}{\mathrm{X}}}$$

Unidirectional expansion

To perform an expansion in the vicinity of $x = a$ where $x > a$, use a positive real (such as 4.) for the order; to perform an expansion in the vicinity of $x = a$ where $x < a$, use a negative real (such as –4.) for the order

Example 1:

Give a 2^{nd}-order expansion of $\frac{(1+X)^{\frac{1}{X}}}{X^3}$ in the vicinity of $X = 0^+$.

Typing:

$$\mathrm{SERIES}\left(\frac{(1+\mathrm{X})^{\frac{1}{\mathrm{X}}}}{\mathrm{X}^3}, \mathrm{X}, 2.\right)$$

gives:

$$\left.\frac{2\cdot \mathrm{e}-\mathrm{e}\cdot \mathrm{h}}{2\cdot \mathrm{h}^3}+\mathrm{O}\left(\frac{\mathrm{e}}{\mathrm{h}}\right)\right|_{\mathrm{h}=\mathrm{X}}$$

Example 2:

Give a 2nd-order expansion of $\frac{(1+X)^{\frac{1}{X}}}{X^3}$ in the vicinity of X = 0^-.

Typing:

$$\text{SERIES}\left(\frac{(1+X)^{\frac{1}{X}}}{X^3},X,-2.\right)$$

gives:

$$\left.\frac{2\cdot e-e\cdot h}{2\cdot h^3}+O\left(\frac{e}{h}\right)\right|_{h=X}$$

Example 3:

Give a 2nd-order expansion of $\frac{(1+X)^{\frac{1}{X}}}{X^3}$ in the vicinity of X = 0.

Typing:

$$\text{SERIES}\left(\frac{(1+X)^{\frac{1}{X}}}{X^3},X,2\right)$$

gives:

$$\left.\frac{2\cdot e-e\cdot h}{2\cdot h^3}+O\left(\frac{e}{h}\right)\right|_{h=X}$$

4.12.4 TAYLOR0

TAYLOR0 has a single argument: the function of *x* to expand. It returns the function's limited 4th-relative-order expansion in the vicinity of *x*=0 (if *x* is the current variable).

Typing:

$$\text{TAYLORO}\left(\frac{\text{TAN}(P\cdot X)-\text{SIN}(P\cdot X)}{\text{TAN}(Q\cdot X)-\text{SIN}(Q\cdot X)}\right)$$

gives:

$$\frac{P^3}{Q^3}-\frac{Q^2P^3-P^5}{4\cdot Q^3}\cdot X^2$$

Note: "4th-order" means that the numerator and the denominator are expanded to the 4th relative order (here, the 5th absolute order for the numerator, and for the denominator, which is given in the end, the 2nd order (5 – 3), seeing that the exponent of the denominator is 3).

4.12.5 TRUNC

TRUNC enables you to truncate a polynomial at a given order (used to perform limited expansions).

TRUNC has two arguments: a polynomial and X^n.

TRUNC returns the polynomial truncated at order n – 1; that is, the returned polynomial has no terms with exponents ≥ n.

Typing:

$$\text{TRUNC}\left(\left(1+X+\frac{1}{2}\cdot X^2\right)^3,X^4\right)$$

gives:

$$1+3\cdot X+\frac{9}{2}\cdot X^2+4\cdot X^3$$

4.13 Conversion Functions

All functions in this section can be found in the REWRITE menu on the menu bar.

4.13.1 DISTRIB

DISTRIB enables you to apply the distributivity of multiplication in respect to addition in a single instance.

DISTRIB enables you, when you apply it several times, to carry out the distributivity step by step.

Typing:

$$\text{DISTRIB}\ ((X+1)\cdot(X+2)\cdot(X+3))$$

gives:

$$X\cdot(X+2)\cdot(X+3)+1\cdot(X+2)\cdot(X+3)$$

4.13.2 EPSX0

EPSX0 has as a parameter an expression in X, and returns the same expression with the values less than EPS replaced by zeroes.

Typing:

EPSX0(0.001 + X)

gives, if EPS=0.01:

$$0+X$$

or, if EPS=0.0001:

$$.001+X$$

4.13.3 EXP2POW

EXP2POW transforms an expression of the form exp(n*ln(x)), rewriting it as a power of x.

Typing: EXP2POW(EXP(N*LN(X)))

gives: X^N

Take careful note of the difference between this function and LNCOLLECT, as shown in the following examples:

LNCOLLECT(EXP(N*LN(X))) = EXP(N*LN(X))

LNCOLLECT(EXP(LN(X)/3)) = EXP(LN(X)/3)

EXP2POW(EXP(LN(X)/3)) = $\sqrt[3]{X}$

4.13.4 EXPLN

EXPLN takes as an argument a trigonometric expression.

EXPLN transforms the trigonometric function into exponentials and logarithms WITHOUT linearising it.

EXPLN puts the calculator into `complex mode`.

Typing:

EXPLN(SIN(X))

gives:

$$\frac{\mathrm{EXP}(i\cdot X)-\frac{1}{\mathrm{EXP}(i\cdot X)}}{2\cdot i}$$

4.13.5 FDISTRIB

FDISTRIB enables you to apply the distributivity of multiplication with respect to addition all at once.

Typing:

FDISTRIB((X + 1)·(X + 2)·(X + 3))

gives:

$$X^3+6\,X^2+11\cdot X+6$$

after simplification.

4.13.6 LIN

LIN has as an argument an expression containing exponentials and trigonometric functions.

LIN linearises the expression (in terms of exp($n\cdot x$)).
LIN puts the calculator into `complex mode` when dealing with trigonometric functions.

Example 1

Typing:

LIN(SIN(X))

gives:

$$-\frac{i}{2}\cdot \mathrm{EXP}(i\cdot X)+\frac{i}{2}\cdot \mathrm{EXP}(-(i\cdot X))$$

Example 2

Typing:

$$\mathrm{LIN}(\mathrm{COS}(X)^2)$$

gives:

$$\frac{1}{4}\cdot \mathrm{EXP}(-(2\cdot i\cdot X))+\frac{1}{2}+\frac{1}{4}\cdot \mathrm{EXP}(2\cdot i\cdot X)$$

Example 3

Typing:

$$\mathrm{LIN}((\mathrm{EXP}(X) + 1)^3)$$

gives:

$$3\cdot \mathrm{EXP}(X) + 1 + 3\cdot \mathrm{EXP}(2\cdot X) + \mathrm{EXP}(3\cdot X)$$

4.13.7 LNCOLLECT

LNCOLLECT has as an argument an expression containing logarithms.

LNCOLLECT regroups the terms in the logarithms. It's therefore preferable to use an expression that has already been factored (using FACTOR).

Typing:

$$\mathrm{LNCOLLECT}(\mathrm{LN}(X + 1) + \mathrm{LN}(X - 1))$$

gives:

$$\mathrm{LN}((X - 1)\cdot(X + 1))$$

4.13.8 POWEXPAND

POWEXPAND writes a power in the form of a product.

Typing:

$$\mathrm{POWEXPAND}((X + 1)^3)$$

gives:

$$(X + 1)\cdot(X + 1)\cdot(X + 1)$$

4.13.9 SIMPLIFY

SIMPLIFY simplifies an expression automatically.

As with all automated simplification routines, however, one should not expect miracles.

Typing:

$$\mathrm{SIMPLIFY}\left(\frac{\mathrm{SIN}(3\cdot X)+\mathrm{SIN}(7\cdot X)}{\mathrm{SIN}(5\cdot X)}\right)$$

gives, after simplification:

$$\left(4\cdot \mathrm{COS}(X)^2-2\right)$$

4.13.10 XNUM

XNUM has an expression as a parameter.

XNUM puts the calculator into `approximate mode` and returns the numeric value of the expression.

Typing:

$$\mathrm{XNUM}(\sqrt{2})$$

gives:

$$1.41421356237$$

4.13.11 XQ

XQ has a real numeric expression as a parameter.

XQ puts the calculator into `exact mode` and gives a rational or real approximation of the expression.

Typing:

XQ(1.41421)

gives:

$$\frac{66441}{46981}$$

Typing:

XQ(1.414213562)

gives:

$$\sqrt{2}$$

4.14 Equations

All the functions in this section are found in the SOLV menu on the menu bar.

4.14.1 ISOLATE

ISOLATE isolates a variable in an expression or an equation.

ISOLATE has two parameters: an expression or equation, and the name of the variable to isolate.

Note: The result from ISOLATE is not editable in the Equation Editor, but is echoed to the CAS history.

Typing:

ISOLATE(X^4 – 1 = 3,X)

gives:

$$\{X = \sqrt{2} *i, X=\sqrt{2} *-1, X=-\sqrt{2} *i, X=\sqrt{2} \}$$

in the CAS history, which can be seen by pressing the HOME key, and then VIEW on the menu.

4.14.2 SOLVEVX

SOLVEVX has as a parameter either (1) an equality between two expressions in the variable contained in VX, or (2) a single such expression (in which case “= 0” is implied).

SOLVEVX solves the equation.

Example 1

Typing: SOLVEVX(X^4 – 1 = 3)

gives, in real mode:

$$(X = \sqrt{2}\ ) \text{ OR } (X = -\sqrt{2}\ )$$

or, in complex mode:

$$(X = \sqrt{2}\ ) \text{ OR } (X = -(\sqrt{2} \cdot i\ )) \text{ OR } (X = \sqrt{2} \cdot -1) \text{ OR } (X = \sqrt{2} \cdot i)$$

Example 2

Typing: SOLVEVX((X – 2)·SIN(X))

gives, in real mode:

$$(X = 2 \cdot \pi \cdot n1) \text{ OR } (X = -(2 \cdot \pi \cdot n1 - \pi) \text{ OR } (X = 2)$$

4.14.3 SOLVE

SOLVE has as two parameters: (1) either an equality between two expressions, or a single expression (in which case “= 0” is implied), and (2) the name of a variable.

SOLVE solves the equation.

Typing:

SOLVE(X^4 – 1 = 3,X)

gives, in real mode:

$$(X = \sqrt{2}\ ) \text{ OR } (X = -\sqrt{2}\ )$$

or, in complex mode:

$$(X = \sqrt{2}\ ) \text{ OR } (X = -(\sqrt{2} \cdot i)) \text{ OR } (X = \sqrt{2} \cdot -1) \text{ OR } (X = \sqrt{2} \cdot i)$$

4.15 Linear Systems

All the functions in this section are found in the SOLV menu on the menu bar.

4.15.1 LINSOLVE

LINSOLVE enables you to solve a system of linear equations.

It's assumed that the various equations are of the form *expression = 0.*

LINSOLVE has two arguments: the first members of the various equations separated by AND, and the names of the various variables separated by AND.

Example 1

Typing:

LINSOLVE(X + Y + 3 AND X – Y + 1, X AND Y)

gives:

(X = –2)AND(Y = –1)

or, in Step by step mode (CFG etc...):

L2 = L2 – L1

$$\begin{bmatrix} 1 & 1 & 3 \\ 1 & -1 & 1 \end{bmatrix}$$

ENTER

L1 = 2·L1 – – L2

$$\begin{bmatrix} 1 & 1 & 3 \\ 0 & -2 & -2 \end{bmatrix}$$

ENTER

Reduction Result

$$\begin{bmatrix} 2 & 0 & 4 \\ 0 & -2 & -2 \end{bmatrix}$$

ENTER

The following is then written to the Equation Editor:

(X = –2) AND (Y = –1)

Example 2

Type:

(2·X + Y + Z = 1) AND (X + Y +2·Z = 1) AND (X + 2·Y + Z = 4)

Then, invoke LINSOLVE and type the unknowns:

X AND Y AND Z

and press the ENTER key.

The following result is produced if you're in Step by step mode (CFG etc...):

L2 = 2·L2 – L1

$$\begin{bmatrix} 2 & 1 & 1 & -1 \\ 1 & 1 & 2 & -1 \\ 1 & 2 & 1 & -4 \end{bmatrix}$$

then press OK:

L3 = 2·L3 – L1

$$\begin{bmatrix} 2 & 1 & 1 & -1 \\ 0 & 1 & 3 & -1 \\ 1 & 2 & 1 & -4 \end{bmatrix}$$

...and so on until, finally:

Reduction Result

$$\begin{bmatrix} 8 & 0 & 0 & 4 \\ 0 & 8 & 0 & -20 \\ 0 & 0 & -8 & -4 \end{bmatrix}$$

then press ENTER, and:

$$\left(X = -\frac{1}{2}\right) \text{AND} \left(Y = \frac{5}{2}\right) \text{AND} \left(Z = -\frac{1}{2}\right)$$

is written to the Equation Editor.

4.16 Differential Equations

All the functions in this section are found in the SOLV menu on the menu bar.

4.16.1 DESOLVE and SUBST

DESOLVE enables you to solve differential equations.

The parameters are: the differential equation (where y' is written `d1Y(X)`), and the unknown Y(X).

Example

Solve:

$$y'' + y = \cos(x) \quad y(0) = c_0 \quad y'(0)=c_1$$

Typing:

$$\text{DESOLVE}(d_1d_1Y(X) + Y(X) = \text{COS}(X), Y(X))$$

gives:

$$Y(X) = \frac{2\cdot cC0\cdot COS(X) + (X + 2\cdot cC1)\cdot SIN(X)}{2}$$

in real mode. cC0 and cC1 are integration constants ($y(0)$ = cC0 $y'(0)$ = cC1).

You can then assign values to the constants using the SUBST command. To produce the solutions for $y(0) = 1$, type:

$$\text{SUBST}\left(Y(X) = cC0\cdot COS(X) + \frac{X + 2\cdot cC1}{2}\cdot SIN(X), cC0 = 1\right)$$

which gives:

$$Y(X) = \frac{2\cdot 1\cdot COS(X) + (X + 2\cdot cC1)\cdot SIN(X)}{2}$$

It is possible to solve for the constants from the outset. If $y'(0)=2$, the typing

DESOLVE((d_1d_1Y(X)+Y(X)=COS(X)) AND (Y(0)=1) AND (d1Y(0)=2),Y(X))

Gives

$$Y(X) = \frac{2\cdot COS(X) + (X + 4)\cdot SIN(X)}{2}$$

when simplified.

4.16.2 LDEC

LDEC enables you to directly solve linear equations having constant coefficients.

The parameters are the second member and the characteristic equation.

Solve:

$$y'' - 6\cdot y' + 9\cdot y = x\cdot e^{3\cdot x}$$

Typing:

LDEC(X·EXP(3·X),X² – 6·X + 9)
gives:

$$\frac{\left(X^3 + 6\cdot X\cdot cC1 + (6 - 18\cdot X)\cdot cC0\right)\cdot EXP(3\cdot X)}{6}$$

cC0 and cC1 are integration constants (y(0) = cC0 y′(0) = cC1).

4.17 Trigonometric Expressions

All the functions in this section are found in the TRIG menu on the CAS Toolbar.

4.17.1 ACOS2S

ACOS2S has as an argument a trigonometric expression.

ACOS2S transforms the expression by replacing arccos(x) with $\frac{\pi}{2}$ – arcsin(x).

Typing:

ACOS2S(ACOS(X) + ASIN(X))

gives: $\frac{\pi}{2}$

when simplified.

4.17.2 ASIN2C

ASIN2C has as an argument a trigonometric expression.

ASIN2C transforms the expression by replacing arcsin(x) with $\frac{\pi}{2}$ – arcos(x).

Typing:

$$\mathrm{ASIN2C(ACOS(X) + ASIN(X))}$$

gives:

$$\frac{\pi}{2}$$

when simplified.

4.17.3 ASIN2T

ASIN2T has as an argument a trigonometric expression.

ASIN2T transforms the expression by replacing arcsin(x) with $\arctan\left(\frac{x}{\sqrt{1-x^2}}\right)$.

Typing:

$$\mathrm{ASIN2T(ASIN(X))}$$

gives:

$$\mathrm{ATAN}\left(\frac{\mathrm{X}}{\sqrt{1-\mathrm{X}^2}}\right)$$

4.17.4 ATAN2S

ATAN2S has as an argument a trigonometric expression.

ATAN2S transforms the expression by replacing arctan(x) with $\arcsin\left(\frac{x}{\sqrt{1+x^2}}\right)$.

Typing:

$$\mathrm{ATAN2S(ATAN(X))}$$

gives:

$$\mathrm{ASIN}\left(\frac{\mathrm{X}}{\sqrt{1+\mathrm{X}^2}}\right)$$

4.17.5 HALFTAN

HALFTAN has as an argument a trigonometric expression.

HALFTAN transforms sin(x), cos(x) and tan(x) in the expression, rewriting them in terms of tan(x/2).

Typing:

$$\mathrm{HALFTAN}\left(\frac{\mathrm{SIN}(2\cdot \mathrm{X})}{1+\mathrm{COS}(2\cdot \mathrm{X})}\right)$$

gives, after simplification:

$$\mathrm{TAN(X)}$$

Typing:

$$\mathrm{HALFTAN(SIN(X)^2 + COS(X)^2)}$$

gives:

$$\left(\frac{2\cdot \mathrm{TAN}\left(\frac{\mathrm{X}}{2}\right)}{\mathrm{TAN}\left(\frac{\mathrm{X}}{2}\right)^2+1}\right)^2+\left(\frac{1-\mathrm{TAN}\left(\frac{\mathrm{X}}{2}\right)^2}{\mathrm{TAN}\left(\frac{\mathrm{X}}{2}\right)^2+1}\right)^2$$

or, after simplification:

$$1$$

4.17.6 SINCOS

SINCOS takes as an argument an expression containing complex exponents.

SINCOS then rewrites this expression in terms of sin(x) and cos(x).

Typing:

$$\mathrm{SINCOS(EXP(i\cdot X))}$$

gives:

$$\mathrm{COS(X) + i\cdot SIN(X)}$$

after turning on complex mode, if necessary.

4.17.7 TAN2CS2

TAN2CS2 takes as an argument a trigonometric expression.

TAN2CS2 transforms this expression by replacing tan(x) with $\frac{1-\cos(2\cdot x)}{\sin(2\cdot x)}$.

Typing:

TAN2CS2(TAN(X))

gives:

$$\frac{1-\mathrm{COS}(2\cdot \mathrm{X})}{\mathrm{SIN}(2\cdot \mathrm{X})}$$

4.17.8 TAN2SC

TAN2SC has as an argument a trigonometric expression.

TAN2SC transforms this expression by replacing tan(x) with $\frac{\sin(x)}{\cos(x)}$.

Typing:

TAN2SC(TAN(X))

gives:

$$\frac{\mathrm{SIN(X)}}{\mathrm{COS(X)}}$$

4.17.9 TAN2SC2

TAN2SC2 has as an argument a trigonometric expression.

TAN2SC2 transforms this expression by replacing tan(x) with $\frac{\sin(2\cdot x)}{1+\cos(2\cdot x)}$.

Typing:

TAN2SC2(TAN(X))

gives

$$\frac{\mathrm{SIN}(2\cdot \mathrm{X})}{1+\mathrm{COS}(2\cdot \mathrm{X})}$$

4.17.10 TCOLLECT

TCOLLECT has as an argument a trigonometric expression.

TCOLLECT linearises this expression in terms of sin($n\cdot x$) and cos($n\cdot x$), then (in `Real mode`) reconstructs the sine and cosine of the same angle.

Typing:

TCOLLECT(SIN(X) + COS(X))

gives:

$$\sqrt{2}\cdot \mathrm{COS}\left(\mathrm{X}-\frac{\pi}{4}\right)$$

4.17.11 TEXPAND

TEXPAND has as an argument a trigonometric expression.

TEXPAND develops this expression in terms of sin(x) and cos(x).

Example 1

Typing:

TEXPAND(COS(X + Y))

gives:

COS(Y)·COS(X) – SIN(Y)·SIN(X)

Example 2

Typing:

TEXPAND(COS(3·X))

gives:

$4\cdot\mathrm{COS(X)}^3 - 3\cdot\mathrm{COS(X)}$

Example 3

Typing:

$$\mathrm{TEXPAND}\left(\frac{\mathrm{SIN}(3\cdot X)+\mathrm{SIN}(7\cdot X)}{\mathrm{SIN}(5\cdot X)}\right)$$

gives, after pressing ENTER to simplify:

$$4\cdot \mathrm{COS}(X)^2-2$$

4.17.12 TLIN

TLIN has as an argument a trigonometric expression.

TLIN linearises this expression in terms of sin(n·x) and cos(n·x).

Example 1

Typing:

$$\mathrm{TLIN}(\mathrm{COS}(X)\cdot \mathrm{COS}(Y))$$

gives:

$$\frac{1}{2}\cdot \mathrm{COS}(X-Y)+\frac{1}{2}\cdot \mathrm{COS}(X+Y)$$

Example 2

Typing:

$$\mathrm{TLIN}(\mathrm{COS}(X)^3)$$

gives:

$$\frac{1}{4}\cdot \mathrm{COS}(3\cdot X)+\frac{3}{4}\cdot \mathrm{COS}(X)$$

Example 3

Typing:

$$\mathrm{TLIN}(4\cdot \mathrm{COS}(X)^2-2)$$

gives:

$$2\cdot \mathrm{COS}(2\cdot X)$$

4.17.13 TRIG

TRIG has as an argument a trigonometric expression.

TRIG simplifies this expression using the identity $\sin(x)^2+\cos(x)^2=1$.

Typing:

$$\mathrm{TRIG}(\mathrm{SIN}(X)^2+\mathrm{COS}(X)^2+1)$$

gives:

$$2$$

4.17.14 TRIGCOS

TRIGCOS has as an argument a trigonometric expression.

TRIGCOS simplifies this expression, using the identity $\sin(x)^2+\cos(x)^2=1$ to rewrite it in terms of cosines.

Typing:

$$\mathrm{TRIGCOS}(\mathrm{SIN}(X)^4+\mathrm{COS}(X)^2+1)$$

gives:

$$\mathrm{COS}(X)^4-\mathrm{COS}(X)^2+2$$

4.17.15 TRIGSIN

TRIGSIN has as an argument a trigonometric expression.

TRIGSIN simplifies this expression, using the identity $\sin(x)^2+\cos(x)^2=1$ to rewrite it in terms of sines.

Typing:

$$\mathrm{TRIGSIN}(\mathrm{SIN}(X)^4+\mathrm{COS}(X)^2+1)$$

gives:

$$\mathrm{SIN}(X)^4-\mathrm{SIN}(X)^2+2$$

4.17.16 TRIGTAN

TRIGTAN has as an argument a trigonometric expression.

TRIGTAN simplifies this expression, using the identity $\sin(x)^2 + \cos(x)^2 = 1$ to rewrite it in terms of tangents.

Typing:

$$\mathrm{TRIGTAN}(\mathrm{SIN}(X)^4 + \mathrm{COS}(X)^2 + 1)$$

gives:

$$\frac{2\cdot \mathrm{TAN}(X)^4 + 3\cdot \mathrm{TAN}(X)^2 + 2}{\mathrm{TAN}(X)^4 + 2\cdot \mathrm{TAN}(X)^2 + 1}$$

5 Worked Exercises with the HP40

5.1 Introduction

Begin by selecting CAS:

to do this, press F6 for CAS on the menu bar.

The various commands used in this chapter are found:

- in the Equation Editor menus:
  - ALGB (CFG DEF FACTOR SUBST TEXPAND)
  - DIFF (DERIVX DERIV INTVX INT LIMIT TABVAR)
  - REWRITE (DISTRIB LIN POWEXPAND XNUM)
  - SOLV (LINSOLV)
- and in the MATH key's menu:
  - Complex (DROITE RE IM)
  - Integer (IEGCD ISPRIME? PROPFRAC).

Next, put the calculator into exact real algebraic mode:

to do this, press ALGB on the menu bar and highlight CFG, then press OK on the menu bar.

It suffices then to choose Default cfg, then OK on the menu bar, but you can also choose Direct mode or Step by step mode (Step/step), then quit the configuration menu with CANCEL on the menu bar.

Don't forget that you must press ENTER after each command!

In the remainder of this chapter, you will find portions of the 1999 mathematical proof of Brevet d'Amiens, and the 1999 mathematical proof (series S) of Bac.

The examples illustrate as many features of the HP40G as possible...

It's worth noting, though, that it's still up to the student to take care to check the calculations and to know what course to follow when carrying out calculations.

5.2 Exercises on Brevet

5.2.1 Exercise 1

Given *A*:

$$\frac{\frac{3}{2}-1}{\frac{1}{2}+1}$$

calculate the result of *A* in the form of an irreducible fraction, showing each step of the calculation.

In the Equation Editor, enter the value of A by typing:

3 ÷ 2 ▷–1 ▷▷÷ 1 ÷ 2 ▷+ 1

▷selects the denominator.

ENTER simplifies the denominator, giving:

$$\frac{\frac{3}{2}-1}{\frac{3}{2}}$$

Then, select the numerator using ◁

ENTER simplifies the numerator, giving:

$$\frac{\frac{1}{2}}{\frac{3}{2}}$$

△ selects the entire fraction, and ENTER simplifies the fraction, giving:

$$\frac{1}{3}$$

5.2.2 Exercise 2

Given the number *C*:

$$C = 2\sqrt{45} + 3\sqrt{12} - \sqrt{20} - 6\sqrt{3}$$

write *C* in the form $d\sqrt{5}$, where *d* is a whole number.

In the Equation Editor, we enter the value of *C* by typing:

$2\sqrt{45}$ ▷▷$+3\sqrt{12}$ ▷▷ $-\sqrt{20}$ ▷▷$-6\sqrt{3}$

▷ ▷ ▷ selects $-6\sqrt{3}$ and

◁ selects $-\sqrt{20}$

▽▽selects 20

Invoke the FACTOR command, which is found in the ALGB menu.

Pressing ENTER then causes 20 to be factored into $2^2 \cdot 5$.

△ selects $\sqrt{2^2 \cdot 5}$ and ENTER returns $2\sqrt{5}$

▷ selects $-2\sqrt{5}$

SHIFT ◁ exchanges $3\sqrt{12}$ with $-2\sqrt{5}$

◁ selects $2\sqrt{45}$

▽ ▷▽ selects 45

Invoke the FACTOR command, which is found in the ALGB menu.

Pressing ENTER then causes 45 to be factored into $3^2 \cdot 5$.

△selects $\sqrt{3^2 \cdot 5}$ and ENTER replaces $\sqrt{3^2 \cdot 5}$ with $3\sqrt{5}$

△selects $2 \cdot 3\sqrt{5}$

SHIFT▷selects $2 \cdot 3\sqrt{5}$ and $-2\sqrt{5}$ and then ENTER completes the operation and gives:

$4\sqrt{5}$

It remains to transform $3\sqrt{12}$ and to see that this term is combined with $-6\sqrt{3}$.

The result is:

$C = 4\sqrt{5}$

5.2.3 Exercise 3

Given the expression $D = (3x-1)^2 - 81$

1. Expand and reduce *D*
2. Factor *D*
3. Solve the equation: $(3x-10)(3x+8)=0$
4. Evaluate *D* for $x=-5$

1. First, write *D* into the Equation Editor by typing:
   3X – 1 ▷▷X^y2 ▷– 81
   Select $(3X-1)^2$ (▷◁), then press ENTER to expand the expression. This gives:
   $9X^2-6X+1-81$
   To do the expansion step by step, press MEMORY (SHIFT ,) to recall the previous expression, then invoke POWEXPAND for $(3\cdot X-1)^2$, then execute DISTRIB a couple of times on the result to obtain:
   $9X^2-6X+1$
   △ selects the entire expression, then pressing ENTER reduces it to:
   $9X^2-6X-80$
2. Invoke FACTOR to obtain:
   $(3X+8)\cdot(3X-10)$
   You can also retrieve the original expression, select 81 to factor it into 3^4, and work out the difference between the two squares...
3. Invoke the SOLVEX command, then press ENTER to obtain:
   $$\left(X=-\frac{8}{3}\right)\text{OR}\left(X=\frac{10}{3}\right)$$
4. Search for *D* in the history (HOME key), then highlight *D* and press ENTER to confirm your choice.
   Invoke SUBST, then complete the second argument:
   $X=-5$
   Then press ENTER to obtain:
   $(3\cdot -5-1)^2-81$
   Pressing ENTER once more yields the result:
   175
   ...therefore, D = 175 when X= –5.

5.2.4 Exercise 4

A baker produces two different assortments of biscuits and macaroons.

A packet of the first assortment contains 17 biscuits and 20 macaroons.

A packet of the second assortment contains 10 biscuits and 25 macaroons.

Both packets cost 90¢.

Calculate the price of one biscuit, and the price of one macaroon.

Let *x* be the price in cents of one biscuit, and *y* the price of one macaroon. The problem is then to solve:

$$\begin{cases} 17x+20y & =90 \\ 10x+25y & =90 \end{cases}$$

In the Equation Editor, type:

$\text{LINSOLVE}(17\cdot X+20\cdot Y-90 \text{ AND } 10\cdot X+25\cdot Y-90, X \text{ AND } Y)$

If you're working in Step by step mode, this produces:

$L_2=17\cdot L_2-10\cdot L_1$

$$\begin{bmatrix} 17 & 20 & -90 \\ 10 & 25 & -90 \end{bmatrix}$$

$L_1=45\cdot L_1-4\cdot L_2$

$$\begin{bmatrix} 17 & 20 & -90 \\ 0 & 225 & -630 \end{bmatrix}$$

ReductionResult :

$$\begin{bmatrix} 765 & 0 & -1530 \\ 0 & 225 & -630 \end{bmatrix}$$

Pressing ENTER then produces the result:

$$(X=2)\text{AND}\left(Y=\frac{14}{5}\right)$$

If you select $\frac{14}{5}$ and press the NUM key (or invoke XNUM), you get:

$$(X = 2)\,\mathrm{AND}\,(Y = 2.8)$$

—in other words, the price of one biscuit is 2¢, and the price of one macaroon is 2.8¢.

Note: If the calculator has gone into `Approx` mode, put it back into `Exact` mode using CFG.

5.2.5 Exercise 5

Say that A and B are points having the coordinates:

A(–1 ; 3) and B(–3 ; –1), where the unit of measure is the centimetre.

1. Find exact length of AB in centimetres.
2. Determine the equation of the line AB.

First Method

Type:

STORE((–1, 3), A), accept the change to Complex mode, if necessary.

STORE((–3, –1), B)

The vector *AB* has coordinates B – A.

1. Typing:

$$\mathrm{ABS}(B - A)$$

gives:

$$2\sqrt{5}$$

2. Apply the DROITE command, on the MATH key's Complex menu:

$$\mathrm{DROITE}\ (A,B)$$

gives:

$$y = 2 \cdot (x - -1) + 3$$

Pressing ENTER then produces the result:

$$Y = 2 \cdot X + 5$$

Second Method

1. Typing:

$$(-3, -1) - (-1, 3)$$

gives:

$$-(2+4\cdot i)$$

Apply the ABS command to get:

$$|-(2+4\cdot i)|$$

gives:

$$2 \cdot \sqrt{5}$$

2. Typing:

$$\mathrm{DROITE}((-1, 3), (-3, -1))$$

gives:

$$Y = 2 \cdot (X - -1) + 3$$

Pressing ENTER then produces the result:

$$Y = 2 \cdot X + 5$$

5.3 Exercises on Bac

5.3.1 Exercise 1

The object of this exercise is to trace the curve Γ described by *M* and given by $\frac{1}{2} \cdot z^2 - z$, given that it's slope, *m*, given by *z* describes a circle *C* with center O and having a radius of 1. Let *t* be a real number in [–π, π], and let *m* be the point of *C* given by $z = e^{i \cdot t}$.

1. Calculating the coordinates of *M*:
   First, make `t` the VX variable (SHIFT SYMB (SETUP) keys)

   Then, enter the expression $\frac{1}{2} \cdot z^2 - z$ into the Equation Editor.

   In the Equation Editor, type:

   ALPHA Z X^y 2 ▷÷ 2 ▷– ALPHA Z ▷▷

The expression $\frac{Z^2}{2} - Z$ is selected.

Since $z = e^{i \cdot t}$, invoke SUBST and complete the second argument by typing:

$$\mathrm{SUBST}\left(\frac{Z^2}{2} - Z, Z = \mathrm{EXP}(i \cdot t)\right)$$

which gives:

$$\frac{\mathrm{EXP}(i \cdot t)^2}{2} - \mathrm{EXP}(i \cdot t)$$

Then, linearises the expression with the command:

LIN

The result (after accepting the switch to Complex mode) is:

$$-1 \cdot \mathrm{EXP}(i \cdot t) + \frac{1}{2} \cdot \mathrm{EXP}(2 \cdot i \cdot t)$$

Now store the result with the STORE command by typing:

$$\mathrm{STORE}\left(-1 \cdot \mathrm{EXP}(i \cdot t) + \frac{1}{2} \cdot \mathrm{EXP}(2 \cdot i \cdot t), M\right)$$

then pressing ENTER.

- To calculate the real part of this expression, enter the command:

  RE

  which returns:

  $$\frac{\mathrm{COS}(t \cdot 2) - 2 \cdot \mathrm{COS}(t)}{2}$$

  Then, define the function $x(t)$ by invoking DEF:

  Note: You'll need to type = X(t), then exchange X(t) with the expression $\frac{\mathrm{COS}(t \cdot 2) - 2 \cdot \mathrm{COS}(t)}{2}$. To do this, highlight X(t) with ▷ and type SHIFT ◁ .

  Highlight the entire expression and select the DEF command:

  $$\mathrm{DEF}\left(X(t) = \frac{\mathrm{COS}(t \cdot 2) - 2 \cdot \mathrm{COS}(t)}{2}\right)$$

  Then press ENTER.

- To calculate the imaginary part of this expression, type the command:

  IM(M)

  which returns:

  $$\frac{\mathrm{SIN}(t \cdot 2) - 2 \cdot \mathrm{SIN}(t)}{2}$$

  Then, define the function $y(t)$ (in the same way as you defined $x(t)$):

  $$\mathrm{DEF}\left(Y(t) = \frac{\mathrm{SIN}(t \cdot 2) - 2 \cdot \mathrm{SIN}(t)}{2}\right)$$

  Then press ENTER.

To find an axis of symmetry for Γ, calculate $x(-t)$ and $y(-t)$ by typing:

X(–t)ENTER

which produces:

$$\frac{\mathrm{COS}(t \cdot 2) - 2 \cdot \mathrm{COS}(t)}{2}$$

In other words: $x(-t) = x(t)$

Then type:

Y(–t)ENTER

which produces:

$$\frac{-\mathrm{SIN}(t \cdot 2) + 2 \cdot \mathrm{SIN}(t)}{2}$$

In other words: $y(-t) = -y(t)$

If $M_1(x(t), y(t))$ is part of Γ, then $M_2(x(-t), y(-t))$ is also part of Γ.

Since M_1 and M_2 are symmetrical with respect to the x-axis, we can deduce that the x-axis is an axis of symmetry for Γ.

Calculate $x'(t)$:

Typing:

DERIV(X(t),t)

returns:

$$\frac{2\cdot(-2\cdot \mathrm{SIN}(t\cdot 2)-2\cdot -\mathrm{SIN}(t))}{4}$$

or, after simplification (ENTER):

$$-((2\cdot \mathrm{COS}(t)-1)\cdot \mathrm{SIN}(t))$$

You can now define the function $x'(t)$ by invoking DEF:

Note: You'll need to type = X1(t), then exchange X1(t) with the expression $-(\mathrm{SIN}(t)\cdot(2\cdot \mathrm{COS}(t)-1)$. To do this, highlight X1(t) and type SHIFT ◁

$$\mathrm{DEF}(\mathrm{X1}(t)=-((2\cdot \mathrm{COS}(t)-1)\cdot \mathrm{SIN}(t))$$

Then press ENTER.

Calculate $y'(t)$:

Typing:

$$\mathrm{DERIV}(\mathrm{Y}(t),t)$$

returns:

$$\frac{2\cdot(2\cdot \mathrm{COS}(t\cdot 2)-2\cdot \mathrm{COS}(t))}{4}$$

or, after simplification (ENTER):

$$2\cdot \mathrm{COS}(t)^2-\mathrm{COS}(t)-1$$

Invoke FACTOR to factor the expression:

$$\mathrm{FACTOR}(2\cdot \mathrm{COS}(t)^2-\mathrm{COS}(t)-1)$$

then press ENTER.

The response is:

$$(\mathrm{COS}(t)-1)\cdot(2\cdot \mathrm{COS}(t)+1)$$

You can now define the function $y'(t)$ (in the same way as you defined $x'(t)$):

$$\mathrm{DEF}(\mathrm{Y1}(t)=(\mathrm{COS}(t)-1)\cdot(2\cdot \mathrm{COS}(t)+1))$$

Variations of $x(t)$ and $y(t)$

For this, you trace $x(t)$ and $y(t)$ on the same graph.

Type X(t) in the Equation Editor and press ENTER.

Then press the PLOT key.

Select `Function` using the dialog box, and select F1 as the destination.

Then, do the same thing with Y(t), making F2 the destination.

To graph the functions: quit CAS (using the ON (CANCEL) button), choose the `Function` aplet, and check F1 and F2. You'll have to set the window's parameters (SHIFT PLOT), then press PLOT to see the graphs.

Back in the Equation Editor (press the HOME key, then CAS on the menu bar), we can get exact outputs from the curve Γ:

- Values of $x(t)$ and $y(t)$

  Find the values of $x(t)$ and $y(t)$ for $t=0,\frac{\pi}{3},\frac{2\cdot\pi}{3},\pi$ by typing in succession (ENTER is pressed twice in most cases for further simplification):

  $$\mathrm{X}(0)\ \text{ENTER}$$

  Response: $\frac{-1}{2}$

  $$\mathrm{X}\left(\frac{\pi}{3}\right)\ \text{ENTER ENTER}$$

  Response: $\frac{-3}{4}$

  $$\mathrm{X}\left(2\cdot\frac{\pi}{3}\right)\ \text{ENTER ENTER}$$

  Response: $\frac{1}{4}$

  $$\mathrm{X}(\pi)\text{ENTER ENTER}$$

  Response: $\frac{3}{2}$

  $$\mathrm{Y}(0)\ \text{ENTER}$$

  Response: 0

$$\mathrm{Y}\left(\frac{\pi}{3}\right) \text{ ENTER ENTER}$$

Response: $\dfrac{-\sqrt{3}}{4}$

$$\mathrm{Y}\left(2\cdot\frac{\pi}{3}\right) \text{ ENTER ENTER}$$

Response: $\dfrac{-3\cdot\sqrt{3}}{4}$

$$\mathrm{Y}(\pi)\text{ENTER ENTER}$$

Response: 0

- Slope of the tangents $(m = \dfrac{y'(t)}{x'(t)})$

Find the values of $\dfrac{y'(t)}{x'(t)}$ for $t = 0, \dfrac{\pi}{3}, \dfrac{2\cdot\pi}{3}, \pi$ by typing in succession:

$$\mathrm{LIMIT}\left(\frac{\mathrm{Y1(t)}}{\mathrm{X1(t)}}, \mathrm{t}=0\right) \text{ENTER}$$

Response: 0

$$\mathrm{LIMIT}\left(\frac{\mathrm{Y1(t)}}{\mathrm{X1(t)}}, \mathrm{t}=\frac{\pi}{3}\right) \text{ENTER}$$

Response (answer YES when asked UNSIGNED INF. SOLVE?): ∞

$$\mathrm{LIMIT}\left(\frac{\mathrm{Y1(t)}}{\mathrm{X1(t)}}, \mathrm{t}=\frac{2\cdot\pi}{3}\right) \text{ENTER}$$

Response: 0

$$\mathrm{LIMIT}\left(\frac{\mathrm{Y1(t)}}{\mathrm{X1(t)}}, \mathrm{t}=\pi\right) \text{ENTER}$$

Response (answer YES when asked UNSIGNED INF. SOLVE?): ∞

Here, then, are the variations of $x(t)$ and $y(t)$:

t	0		$\frac{\pi}{3}$		$\frac{2\pi}{3}$		π
$x'(t)$	0	-	0	+	?	+	0
$x(t)$	$\frac{-1}{2}$	↓	$\frac{-3}{4}$	↑	$\frac{1}{4}$	↑	$\frac{3}{2}$
$y(t)$	0	↓	$-\frac{\sqrt{3}}{4}$	↓	$\frac{-3\sqrt{3}}{4}$	↑	0
$y'(t)$	0	-	?	-	0	+	?
m	0		∞		0		∞

- The curve Γ:

  Now graph the parametric curve.

In the Equation Editor, type X(t) + i ·Y(t), then press ENTER.
Then press:

PLOT, and select `Parametric` using the dialog box, specifying X1,Y1 as the destination.

To make the graph of the curve Γ: quit CAS (using the ON (CANCEL) button), then choose the `Parametric` aplet. Check X1(T), Y1(T), and choose default values in PLOT SETUP.

5.3.2 Exercise 2 (specialty)

Define the following for a natural whole number:

$$a_{\mathrm{n}} = 4\times10^{\mathrm{n}} - 1,\ b_{\mathrm{n}} = 2\times10^{\mathrm{n}} - 1,\ c_{\mathrm{n}} = 2\times10^{\mathrm{n}} + 1$$

Begin by typing:

$$\mathrm{DEF(A(N)} = 4\cdot10^{\mathrm{N}} - 1)$$
$$\mathrm{DEF(B(N)} = 2\cdot10^{\mathrm{N}} - 1)$$
$$\mathrm{DEF(C(N)} = 2\cdot10^{\mathrm{N}} + 1)$$

1. Now do the following:

   a. Calculate $a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3$:

      Simply type (in succession):

`A(1)`

Result: 39

`B(1)`

Result: 19

`C(1)`

Result: 21

`A(2)`

Result: 399

`B(2)`

Result: 199

`C(2)`

Result: 201

`A(3)`

Result: 3999

`B(3)`

Result: 1999

`C(3)`

Result: 2001

b. Number of digits, and divisibility
In this, the calculator is used only to try out different values of *n* …

Show that the whole numbers *n* such that:

$$10^{n} \le n < 10^{n+1}$$

have ($n + 1$) digits in decimal notation.

We have:

$$3 \cdot 10^{n} < a_n < 4 \cdot 10^{n}$$
$$10^{n} < b_n < 2 \cdot 10^{n}$$
$$2 \cdot 10^{n} < c_n < 3 \cdot 10^{n}$$

so a_n, b_n, c_n, have ($n + 1$) digits in decimal notation.

Moreover, $d_n = 10^n - 1$ is divisible by 9, since its decimal notation can only end in 9.

We also have:

$$a_n = 3 \cdot 10^{n} + d_n$$

and

$$c_n = 3 \cdot 10^{n} - d_n$$

so a_n and c_n are both divisible by 3.

c. b_3 is a prime number
Typing:

ISPRIME?(B(3))

gives:

1.

which means true.

To prove that b_3 is a prime number, it's necessary to show that 1999 is not divisible by any of the prime numbers less than or equal to $\sqrt{1999}$.

As 1999 < 2025 = 45², that means testing the divisibility of 1999 by n = 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41.

1999 is not divisible by any of these numbers, so we can conclude that 1999 is prime.

d. $a_{2n} = b_n \times c_n$
Typing:

B(N)·C(N)

produces:

$$4 \cdot (10^{N})^2 - 1$$

after applying the EXP2POW command. This is the value a_{2n}.

Decomposition of a_6 into its prime factors:

Typing:

FACTOR(A(6))

yields:

$$3 \cdot 23 \cdot 29 \cdot 1999$$

e. b_n and c_n are relatively prime.

In this part, the calculator is useful only for trying out different values of n…

To show that c_n and b_n are relatively prime, it's enough to remark that

$$c_n = b_n + 2$$

That means that the common divisors of c_n and b_n are the common divisors of b_n and 2, as well as the common divisors of c_n and 2. b_n and 2 are relatively prime because b_n is a prime number different from 2. So:

$$\text{PGCD}(c_n, b_n) = PGCD(c_n, 2) = PGCD(b_n, 2) = 1$$

Given the equation:

$$b_3 \cdot x + c_3 \cdot y = 1$$

a. It has at least one solution, as it's actually a form of Bézout's Identity.

In effect, Bézout's Theorem says:

If a and b are relatively prime, there exist x and y such that:

$$a \cdot x + b \cdot y = 1$$

Therefore, the equation:

$$b_3 \cdot x + c_3 \cdot y = 1$$

has at least one solution.

b. Typing:

IEGCD(B(3),C(3))

returns:

1000 AND –9999 = 1

In other words:

$$1 = b_3 \times 1000 + c_3 \times (-999)$$

so we have a particular solution:

$x = 1000$, $y = -999$.

The rest can be done on paper:

$c_3 = b_3 + 2$ and $b_3 = 999 \times 2 + 1$

so $b_3 = 999 \times (c_3 - b_3) + 1$, or:

$$b_3 \times 1000 + c_3 \times (-999) = 1$$

c. In this part, the calculator is not used for finding the general solution.

We have:

$$b_3 \cdot x + c_3 \cdot y = 1$$

and

$$b_3 \times 1000 + c_3 \times (-999) = 1$$

so, by subtraction:

$$b_3 \cdot (x - 1000) + c_3 \cdot (y + 999) = 0$$

or:

$$b_3 \cdot (x - 1000) = -c_3 \cdot (y + 999)$$

According to Gauss's Theorem, c_3 is prime with b_3, so c_3 is a divisor of (x – 1000).

So, there exists $k \in Z$ such that:

$$(x - 1000) = k \times c_3$$

and

$$-(y + 999) = k \times b_3$$

Solving for x and y, we get:

$$x = 1000 + k \times c_3$$

and

$$y = -999 - k \times b_3 \text{ for } k \in Z$$

This gives us:

$$b_3 \cdot x + c_3 \cdot y = b_3 \times 1000 + c_3 \times (-999) = 1$$

The general solution for all $k \in Z$ is therefore:

$$x = 1000 + k \times c_3$$

$$y = -999 - k \times b_3$$

5.3.3 Exercise 3 (non-specialty)

Before you begin, check that the calculator is in exact real mode with `X` as the current variable; if not, select `Default cfg` in CFG.

Given:

$$u_n = \int_0^2 \frac{2x+3}{x+2} e^{\frac{x}{n}} dx$$

1. Do the following:

   a. Variation of $g(x) = \dfrac{2x+3}{x+2}$ for $x \in [0,2]$:

   Typing:

$$\text{DEF}\left(\text{G(X)} = \frac{2\text{X}+3}{\text{X}+2}\right)$$

   then:

$$\text{TABVAR(G(X))}$$

   yields:

$$\begin{array}{ccccccc} -\infty & + & -2 & + & +\infty & \text{X} \\ 2 & \uparrow & \infty & \uparrow & 2 & \text{F} \end{array}$$

   The first line gives the sign of $g'(x)$ according to x, and the second line the variations of $g(x)$. Note that for TABVAR the function is always called F.

   We can deduce, then, that $g(x)$ increases over [0, 2].

   If you were in Step by step mode (for this, choose Step/step and then OK on the CFG menu bar), you would obtain (although the function is labelled F):

$$\text{F} =: \frac{2 \cdot \text{X}+3}{\text{X}+2}$$

   Press ENTER:

$$\text{F}' =: \frac{2 \cdot (\text{X}+2) - (2 \cdot \text{X}+3)}{(\text{X}+2)^2}$$

   Using the down-arrow (▽), scroll down the screen:

$$\rightarrow \frac{1}{(\text{X}+2)^2}$$

   Then press ENTER to obtain the table of variations.

   If you're not in Step by step mode, you can also request the calculation of the derivative by typing:

$$\text{DERVX(G(X))}$$

   which produces the preceding calculation.

   To calculate $g(0)$ and $g(2)$, type:

$$\text{G(0)}$$

   Response: $\dfrac{3}{2}$

$$\text{G(2)}$$

   Response: $\dfrac{7}{4}$

   whence the inequality:

$$\frac{3}{2} \le g(x) \le \frac{7}{4} \quad for\ x \in [0,2].$$

   b. The calculator is not needed here... simply stating that

$$e^{\frac{x}{n}} \ge 0 \quad for\ x \in [0,2]$$

is sufficient to show that, for $x \in [0,2]$, we have:

$$\frac{3}{2}e^{\frac{x}{n}} \le g(x)e^{\frac{x}{n}} \le \frac{7}{4}e^{\frac{x}{n}}$$

c. To integrate the preceding inequality, type:

$$\int_0^2 e^{\frac{X}{N}} dX$$

which produces:

$$N \cdot e^{\frac{2}{N}} - N$$

We can then deduce that:

$$\frac{3}{2}(ne^{\frac{2}{n}} - n) \le u_n \le \frac{7}{4}(ne^{\frac{2}{n}} - n)$$

To justify the preceding calculation, it's necessary to assume that $n \cdot e^{\frac{x}{n}}$ is a primitive of $e^{\frac{x}{n}}$.

If you're not sure, you can type:

$$\text{INTVX}\left(\text{EXP}\left(\frac{X}{N}\right)\right)$$

The simplified result is: $N \cdot \text{EXP}\left(\frac{X}{N}\right)$

d. To find the limit of $(ne^{\frac{2}{n}} - n)$ when $n \to +\infty$:

$$\text{LIMIT}\left(N \cdot \text{EXP}\left(\frac{2}{N}\right) - N, N = +\infty\right)$$

The result is:

$$2$$

Note:

The variable VX is set equal to N; use the SHIFT SYMB (SETUP) keys to reset VX to X.

To check the result, we can say that:

$$\lim_{x \to 0} \frac{e^x - 1}{x} = 1$$

and that therefore:

$$\lim_{x \to +\infty} \frac{e^{\frac{2}{n}} - 1}{\frac{2}{n}} = 1$$

or, simplifying:

$$\lim_{x \to +\infty} (e^{\frac{2}{n}} - 1) \cdot n = 2$$

If the limit L of u_n exists as n tends to $+\infty$ in the inequalities in 1b), we get:

$$\frac{3}{2} \cdot 2 \le L \le \frac{7}{4} \cdot 2$$

2. Do the following:

a. $g(x) = 2 - \frac{1}{x+2}$ and the calculation of $I = \int_0^2 g(x)dx$

Typing:

PROPFRAC(G(X))

returns:

$$2 - \frac{1}{X+2}$$

To calculate the integral *I,* type:

$$\int_0^2 2 - \frac{1}{X+2} dX$$

This produces:

–(LN(2)–4)

Working by hand: $2x+3=2(x+2)-1$, so

$$g(x)=2-\frac{1}{x+2}$$

Then, integrating term by term between 0 and 2 produces:

$$\int_0^2 g(x)dx=\left[2x-\ln(x+2)\right]_{x=0}^{x=2}$$

—that is, since `ln4=2ln2`:

$$\int_0^2 g(x)dx=4-\ln 2$$

b. The calculator is not needed here... simply stating that $e^{\frac{x}{n}}$ increases for $x\in[0,2]$ is sufficient to yield the inequality:

$$1\le e^{\frac{x}{n}}\le e^{\frac{2}{n}}$$

Then, through multiplication, $g(x)$ being positive over [0, 2], we get:

$$g(x)\le g(x)e^{\frac{x}{n}}\le g(x)e^{\frac{2}{n}}$$

and then, integrating:

$$I\le u_n\le e^{\frac{2}{n}}I$$

c. Convergence of u_n

Find the limit of $e^{\frac{2}{n}}$ when $n\to+\infty$:

$$\text{LIMIT}\left(\text{EXP}\left(\frac{2}{\text{N}}\right),\text{N}=+\infty\right)$$

This returns:

$$1$$

In effect, $\frac{2}{n}$ tends to 0 as n tends to $+\infty$, so $e^{\frac{2}{n}}$ tends to $e^0=1$ as n tends to $+\infty$.

As n tends to $+\infty$, the remainder u_n is the portion between I and a quantity that tends to I (cf. inequalities 2b)).

So then, u_n converges, and its limit is I.

We have therefore shown that:

$$L=I=4-\ln 2$$

5.4 Conclusion

You can see that, by that making good use of the HP40G calculator , you can use it for a large proportion of the calculations...

It's necessary to note, however, that arithmetic does require more reasoning. Here, the calculator enables you to check your work...

6 Programming

6.1 Implementation

6.1.1 How to edit and save a program

From the HOME screen, to get access to the catalogue of programs, press the SHIFT 1 (PROGRAM) keys.

A screen then appears containing the list of available programs and a menu bar (EDIT NEW SEND RECV RUN).

To create a new program, press F2 (NEW).

You are asked for the name of the program. Note: If you are not in Alpha mode, press F4 (A..Z) to go into it!

Type its name, then press F6 (OK).

You enter your program, and your work is automatically saved when you leave the editor by pressing HOME or SHIFT 1 (PROGRAM).

6.1.2 How to correct a program

If the syntax is invalid, the calculator displays:

`Invalid Syntax Edit program?` Respond by pressing F6 (YES).

The calculator automatically positions the cursor where the compiler has detected the error. All you need do then is correct it!

6.1.3 How to run a program

To run a program, open the catalogue of programs by pressing SHIFT 1 (PROGRAM).

A screen is then displayed containing the list of available programs and the menu bar EDIT NEW SEND RECV RUN.

Highlight the name of the program and press F6 (RUN).

6.1.4 How to modify a program

To modify a program (and overwrite the old one), open the catalogue of programs by pressing SHIFT 1 (PROGRAM). A screen is then displayed containing the list of available programs and the menu bar EDIT NEW SEND RECV RUN.

Highlight the name of the program and press F1 (EDIT).

If you want to save and edit the old program under a new name, then:

- Open the catalogue of programs (SHIFT 1 (PROGRAM))
- Press F2 (NEW) and type the new name, then press F6 (OK). The Editor opens.
- Press VARS, then the letter P to highlight Program.
- Using the arrows, highlight the name of the old program, then press F4 (VALUE) (to select VALUE on the menu bar), then F6 (OK).

This copies the text of the old program into the editor.

6.2 Comments

It's important to comment your programs.

In writing algorithms, a comment commonly starts with / / and ends at the end of the line.

In the HP40G, a comment starts with @ and ends at the end of the line or at the next @, whichever comes first.

Note:

Don't forget to put a space after the @.

The character @ is obtained by typing SHIFT VAR (CHARS), then highlighting the character and choosing ECHO on the menu bar.

6.3 Variables

6.3.1 Their names

These are the containers in which you can store values, numbers, expressions, and other objects.

With the HP40G, only the 26 letters of the alphabet, and the Greek letter θ, are available for storing real numbers.

6.3.2 The concept of local variables

This concept does not exist in the HP40G.

You can use only global variables.

6.3.3 The concept of parameters

It's not possible for programs written on the HP40G to pass parameters.

You therefore can't use the HP40G programming language to write functions having parameters.

6.4 User entry

6.4.1 Algorithmic syntax

So that the user is able to enter a value for a variable during the course of the program's execution, one writes in algorithms:

```
accept A
```

And for entering values into A and B, one writes:

```
accept A,B
```

6.4.2 HP40G syntax

INPUT A; "NAME";"A="; ;0:

If it annoys you to have to type all the punctuation in the INPUT command, you might prefer to use the PROMPT command instead (thanks to Jean Yves!).

PROMPT A: opens a window that asks you to enter the value of A.

In the remainder of this guide, programs that were written before the existence of PROMPT use the subroutine IN, which enables you to enter two values into A and B.

6.5 Output

6.5.1 Algorithmic syntax

In algorithms one writes:

display "A =", A

6.5.2 HP40G syntax

DISP 3; "A ="A: 3 represents the number of the line where A will be displayed

Or:

MSGBOX "A="A:

6.6 The sequence of statements, or "action"

An action is a sequence of one or more statements.

6.6.1 Algorithmic syntax

In algorithmic language, you use the space or the linebreak to terminate the statement.

6.6.2 HP40G syntax

: indicates the end of the statement.

6.7 The assignment statement

A value or an expression is stored in a variable by means of an assignment.

6.7.1 Algorithmic syntax

In algorithms, one writes (for example):

2*A–> B to store 2*A in B

6.7.2 HP40G syntax

The arrow is obtained by means of the STO▷key on the menu bar.

One types (for example):

2*A STO▷B

6.8 Conditional statements

6.8.1 Algorithmic syntax

If *condition* then
action
endif

If *condition* then
action1 else
action2
endif

Example:

If A = 10 or A< B then
B–A–> B else
A–B– >A
endif

6.8.2 HP40G syntax

IF *condition* THEN
action:
END:

IF *condition* THEN
action1: ELSE
action2:
END:

Note the use of “= =”to denote the condition of equality.

Example:

IF A = = 10 OR A< B THEN
B – A STO▷B: ELSE
A – B STO▷A:
END:

6.9 FOR statement

6.9.1 Algorithmic syntax

For I = A to B do *action* endfor

For I = A to B (step P) do *action* endfor

6.9.2 HP40G syntax

FOR I = A TO B STEP 1 ; *action*: END:

FOR I = A TO B STEP P ; *action*: END:

6.10 WHILE statement

6.10.1 Algorithmic syntax

While *condition* do *action* endwhile

6.10.2 HP40G syntax

WHILE *condition* REPEAT *action* : END:

6.11 Boolean Expressions

A condition is a function that has a Boolean as a value—that is, it is either TRUE or FALSE.

6.11.1 Algorithmic syntax

To express a condition, simply use the operators:

= > > ≤ ≥ ≠

6.11.2 HP40G syntax

Note: for the HP40G calculator, equality is denoted by:

==

Otherwise, the operators are the same.

6.12 Logical Operators

6.12.1 Algorithmic syntax

To express complex conditions, use the logical operators:

`or and not`

6.12.2 HP40G syntax

Or, and, and *not* are represented on the HP40G as OR, AND, and NOT.

6.13 Lists

6.13.1 Algorithmic syntax

In algorithms, you use { } to define a list.

For example, {} designates a void list, and {1, 2, 3} is a list with three elements.

The + is used to concatenate two lists, or a list and an element, or an element and a list:

{1, 2, 3} -> TAB

TAB + 4 -> TAB (TAB now designates {1, 2, 3, 4}

TAB[2] indicates the second element of TAB, here 2.

6.13.2 HP40G syntax

The list variables have the following names: L0, L1, L2, … L9.

You use { } to delimit a list.

For example, {1, 2, 3} is a list with three elements.

But {} does not designate a void list; for that, you must use the command:

SUB L1;L1;2;1 to initialise the list L1 as void.

Following are some useful commands:

MAKELIST(I*I, I, 1, 10, 2) designates a list of the squares of the first five odd whole numbers (2 indicates the step of I).

L1(I) designates the I^{th} element of the list.

CONCAT (L1, {5}) designates a list having the element 5 in addition to the elements of the list L1.

You can also use:

AUGMENT(L1,5), which designates a list having the element 5 in addition to the elements of the list L1.

SUB L2; L1; 2; 4 is a command that copies into L2 the elements of L1 having indices from 2 to 4.

Note the difference between functions and commands:

- Functions return a value, and they have parentheses enclosing their arguments (which are separated by commas), whereas
- Commands do not return values, and their arguments are written directly after the name of the command (and are separated by semi-colons).

6.14 Example: The Sieve of Eratosthenes

6.14.1 Description

To find the prime numbers less than or equal to *N:*

1. Write the numbers from 1 to *N* in a list.
2. Cross out 1, and let 2 be the first value of *P*.
   If $P*P \le N$, then taking the numbers from *P* to *N,*
3. Cross out all the multiples of *P* from *P*P* onward.
4. Augment *P* by 1.
   If *P*P* is less than or equal to *N,* then taking the non-excluded numbers from *P* to *N,*
5. Change *P* to the smallest non-excluded number between *P* and *N*.
6. Repeat steps 3, 4 and 5 until *P*P* exceeds *N.*

6.14.2 Algorithmic syntax

```
function sieve(N)

local TAB PRIMES I P
//TAB and PRIMES are lists
{} ->TAB
{} ->PRIMES

//do steps 1 and 2
for I = 2 to N do
   TAB+I -> TAB
endfor
0 + TAB -> TAB
2-> P
//TAB is the list 0 2 3 4 …N
//1 has been excluded by replacing it with 0

while P*P ≤ N do
   //exclude all multiples of P from P*P onward
   for I = P to FLOOR(N/P) do
      //FLOOR(N/P) designates the integer part of N/P
      0 -> TAB [I*P]
```

```
   endfor
   P+1-> P
   //Find the smallest non-excluded number
   //between P and N
   while (P*P ≤ N) and (TAB[P] = 0) do
      P + 1 -> P
   endwhile
endwhile
```

```
//write the result to the list PRIMES
for I = 2 to N do
   If  TAB[I] ≠ 0  then
      PRIMES + I -> PRIMES
   endif
endfor
Result: PRIMES
```

6.14.3 HP40G syntax

Following is the program SIEVE:

The user must input a value for N.

At the end, the list L2 contains the prime numbers less than or equal to N.

```
INPUT N; "SIEVE";"N="; ;10:
ERASE:
MAKELIST(I,I,1,N,1) -> L1:
0 -> L1(1):
2 -> P:

WHILE P*P ≤ N REPEAT
   FOR I = P TO INT(N/P) STEP 1;
      0->L1(I*P):
   END:
   DISP 3;""L1:
   P+1->P:
   WHILE P*P ≤ N AND L1(P) == 0 REPEAT
      P+1->P:
   END:
END:

{2}->L2:
@we know that 2 is prime
FOR I=3 TO N STEP 1;
   IF L1(I) ≠ 0 THEN
      CONCAT(L2,{I}) ->L2:
   END:
END:
DISP 3 ;"PRIMES" L2:
FREEZE:
```

7 Arithmetic Programs

7.1 GCD and Euclid's Algorithm

Given the two whole positive numbers A and B, find their GCD (Greatest Common Divisor).

Euclid's Algorithm is based on the recursive definition of GCD:

GCD(A,0) = A

GCD(A,B) = GCD(B, A mod B) if $B \neq 0$

where A mod B designate the remainder of the Euclidean division of A by B.

Here is the description of this algorithm:

Perform the successive Euclidean divisions:

$$A = B \times Q1 + R1 \quad 0 \leq R1 < B$$

$$B = R1 \times Q2 + R2 \quad 0 \leq R2 < R1$$

$$R1 = R2 \times Q3 + R3 \quad 0 \leq R3 < R2$$

......

After a finite number of steps, there exists a whole number n such that: $R_n = 0$.

One then has:

$GCD(A,B) = GCD(B, R_1) = \ldots$

$GCD(R_{n-1}, R_n) = GCD(R_{n-1},0) = R_{n-1}$

7.1.1 Algorithmic syntax

- Iterative version

  If $B \neq 0$, calculate R=A mod B. Then, make A equal to B, and B equal to R, and repeat the calculation until B=0. The GCD is then A.

```
function GCD(A,B)
Local R
While B ≠ 0 do
   A mod B ->R
   B ->A
   R ->B
endwhile
Result: A
endfunction
```

- Recursive version

  One simply writes the recursive definition given earlier.

```
function GCD(A,B)
If B ≠ 0 then
   Result: GCD(B,A mod B)
Else
   Result: A  (Result)
endif
endfunction
```

7.1.2 HP40G syntax

- Iterative version for two whole numbers

  First, type the subroutine IN, which enables the user to enter two numbers A and B:

```
INPUT A;"A"; ; ; 1:
INPUT B;"B"; ; ; 1:
ERASE:
```

Then type the GCD program:

```
RUN IN:
DISP 3;"GCD "{A,B}:
WHILE B ≠ 0 REPEAT
   A MOD B->R:
   B ->A:
   R ->B:
END:
DISP 4;"GCD " A:
FREEZE:
```

- Recursive version for two whole numbers A and B

  You cannot create recursive functions on the HP40G... but you can create the program GCDR:

```
DISP 3;"GCD "{A,B}:
FREEZE:
IF B ≠ 0 THEN
```

```
   A MOD B ->R:
   B ->A:
   R ->B:
   GCDR:
ELSE
   DISP 3;"GCD "A:
   FREEZE:
END:
```

First, the values of A and B are stored.

The program GCDR displays the GCD that it is in the process of calculating.

The recursive call `GCDR` returns you to the GCDR program, which you must execute by pressing RUN on the menu bar.

The GCDR program then displays the intermediate GCD calculations.

You can also replace `GCDR` in the preceding program with `RUN GCDR` to avoid having to press RUN on the menu bar, and to suppress the display of the intermediate values, so that you can use this program in a larger program that caters for input and output:

From the recursive program GCDR, we can derive the recursive program PR:

```
IF B ≠ 0 THEN
   A MOD B -> R:
   B -> A:
   R -> B:
   RUN PR:
END:
```

The program PR can be inserted into a larger program catering for input and output:

```
PROMPT A:
PROMPT B:
RUN PR:
ERASE:
MSGBOX A:
```

- Iterative version for two complex numbers.

  If you use the symbolic calculation function IREMAINDER in place of MOD in the preceding programs, GCD (or PR) can then have Gaussian whole numbers as parameters on condition that you replace the names of the variables A, B and R with Z1, Z2 and Z3, and that you change the test for completion. When creating the new version start by recalling the contents of the earlier version, as described in 6.1.4. Here is the iterative version:

```
PROMPT Z1:
PROMPT Z2:
ERASE:
DISP 3;"GCD "{Z1,Z2}:
WHILE ABS(Z2) ≠ 0 REPEAT
   XNUM(IREMAINDER(XQ(Z1),XQ(Z2))) ->Z3:
   Z2 ->Z1:
   Z3 ->Z2:
END:
DISP 4;"GCD "Z1:
FREEZE:
```

- Iterative version for two polynomials.

  The variables E1, E2,… enable you to store expressions, which is what you must do if you deal with polynomials! If you use the symbolic calculation function REMAINDER in place of MOD in the preceding programs, GCD (or PR) can then have polynomials as parameters on condition that you replace the names of the variables A, B and R with E1, E2 and E3, and that you change the test for completion. The CAS must be in Exact and Direct modes for this program.

```
PROMPT E1:
PROMPT E2:
ERASE:
DISP 1;"GCD of":
DISP 2;E1" and":
DISP 3;E2" is":
WHILE DEGREE(E2) ≠ -1 REPEAT
   REMAINDER(E1,E2) ->E3:
   E2 ->E1:
   E3 ->E2:
END:
DISP 4;E1:
FREEZE:
```

  You enter (for example):

  $E1 = S1^2 – 1$ and $E2 = S1^2 – 2 * S1 + 1$ to find the GCD equal to 2*S1 – 2.

7.2 Bézout’s Identity

Bézout’s function (A,B) returns the list:

$\{U,V,GCD(A,B)\}$ where U and V are such that:

$A \times U + B \times V = GCD(A,B)$.

7.2.1 Iterative version without lists

Euclid's Algorithm enables us to find a pair U and V such that:

$A \times U + B \times V = GCD(A,B)$.

In effect, if we call A_0 and B_0 the values the A and B at the start, then we have:

$$A = A_0 \times U + B_0 \times V \quad \text{with} \quad U = 1 \quad \text{and} \quad V = 0$$

$$B = A_0 \times W + B_0 \times X \quad \text{with} \quad W = 0 \quad \text{and} \quad X = 1$$

You then derive A, B, U, V, W and X in such a way that the two relations above are always satisfied.

If:

$A = B \times Q + R \quad 0 \le R < B \quad (R = A \bmod B \text{ and } Q = FLOOR(A/B))$

then:

$R = A - B \times Q = A_0 \times (U - W \times Q) + B_0 \times (V - X \times Q) =$
$A_0 \times S + B_0 \times T$ with $S = U - W \times Q$ and $T = V - X \times Q$

It remains then to repeat the process with:

B in the role of A (`B->A W->U X->V`), and
R in the role of B (`R->B S->W T->X`)

whence the algorithm:

```
function Bezout (A,B)
local U,V,W, X, S, T, Q, R
1->U  0 ->V  0 ->W  1 ->X
While B ≠ 0 do
   A mod B ->R
   FLOOR(A/B) ->Q
   //R=A-B*Q
   U-W*Q ->S
   V-X*Q ->T
   B->A W->U X->V
   R->B S->W T->X
endwhile
Result: {U, V, A}
endfunction
```

7.2.2 Iterative version with lists

You can simplify the algorithmic syntax below by using fewer variables: for this, use the lists LA LB LR to store the triplets {U, V, A}, {W, X, B} and {S, T, R}. This is quite easy, as the calculator knows how to add lists of the same length (by adding elements with the same index), as well as how to multiply a list by a number (by multiplying each of the list's elements by the number).

```
function Bezout (A,B)
local LA LB LR
{1, 0, A}->LA
{0, 1, B}->LB
While LB [3] ≠ 0 do
   LA-LB*FLOOR(LA [3] /LB[3])->LR
   LB->LA
   LR->LB
endwhile
Result: LA
endfunction
```

7.2.3 Recursive version with lists

Bézout's Function can be recursively defined by:

$Bezout(A, 0) = \{1, 0, A\}$

If $B \neq 0$, is it necessary to define $Bezout(A, B)$ as a function of $Bezout(B, R)$ when

$R = A - B \times Q$ and $Q = FLOOR(A/B)$.

We have:

$$Bezout(B, R) = LT = \{W, X, gcd(B, R)\}$$

$$\text{with } W \times B + X \times R = gcd(B, R)$$

Then:

$$W \times B + X \times (A - B \times Q) = gcd(B, R), \text{ or}$$
$$X \times A + (W - X \times Q) \times B = gcd(A, B).$$

whence, if $B \neq 0$ and $Bezout(B, R) = LT$, we have:

$$Bezout(A, B) = \{LT[2], LT[1] - LT[2] \times Q, LT[3]\}.$$

```
function Bezout (A,B)
local LT Q R
```

```
If B ≠ 0 then
   FLOOR(A/B) ->Q
   A- B*Q->R
   Bezout(B,R)->LT
   Result: {LT[2], LT[1]-LT[2]*Q, LT[3]}
else Result: {1, 0, A}
endif
endfunction
```

7.2.4 Recursive version without lists

If you use global variables for A B D U V T, you can view the function Bezout as using A B to calculate the values for U V D (AU+BV=D) by means of a local variable Q.

One can then write:

```
program Bezour
local Q
If B ≠ 0 then
   FLOOR(A/B) ->Q
   A-B*Q->T
   B->A
   T->B
   Bezour
   U-V*Q->T
   V->U
   T->V
else
   1->U
   0->V
   A->D
endif
```

7.2.5 HP40G syntax

- Iterative version with lists

  Here, you also use the program IN, which enables the user to enter two numbers A and B:

```
INPUT A;"A"; ; ; 1:
INPUT B;"B"; ; ; 1:
ERASE:
```

  Now type the BEZOUT program:

```
RUN IN:
DISP 3;"BEZOUT"{A,B}:
{1, 0, A}->L1:
{0, 1, B}->L2:
WHILE L2(3) ≠ 0 REPEAT
   L1-L2*FLOOR(L1(3)/L2(3)) ->L3:
   L2->L1:
   L3->L2:
END:
DISP 4;"U V GCD "L1:
FREEZE:
```

- Recursive version without lists

  Type the program BEZOUR, using the commands:

  PUSH (the command PUSH(A) stores the contents of A on a stack), and

  POP (the POP command retrieves values stored on the stack)

```
PROGRAM BEZOUR
IF B ≠ 0 THEN
   PUSH (FLOOR(A/B)):
   A MOD B->T:
   B->A:
   T->B:
   RUN BEZOUR:
   U-V*POP->T:
   V->U:
   T->V:
ELSE
   1->U:
   0->V:
   A->D:
END:
```

  PUSH (FLOOR(A/B)) has the effect of putting the different values of FLOOR(A/B) onto a stack, and POP recovers them.

  T is an auxiliary variable.

  BEZOUR takes, as user input, the values of the global variables A and B, and assigns values to the global variables U and V such that:

  $A \cdot U + B \cdot V = GCD(A, B)$.

We can then write the final program BEZOURT, which caters for the input of A and B and the output of {U, V, D}:

```
PROGRAM BEZOURT
PROMPT A:
PROMPT B:
RUN BEZOUR:
ERASE:
MSGBOX{U,V,D}:
```

Remarks:

If you use the symbolic calculation functions IREMAINDER and IQUOT(XQ(Z1),XQ(Z2)) in place of MOD and FLOOR(A/B) in the preceding programs, then BEZOUT and BEZOUR can take Gaussian integers as parameters, on condition that you replace the names of the variables A, B, R... with Z1, Z2, Z3...

If you use the symbolic calculation function REMAINDER in place of MOD in the preceding programs, then BEZOUT and BEZOUR can take polynomials as parameters, on condition that you replace the names of the variables A, B, R... with E1, E2, E3... and that you change the test for completion.

7.3 Decomposition into prime factors

7.3.1 Algorithmic syntax

- Initial algorithm

  Let *N* be a whole number.

  For all numbers D from 2 to N, test the divisibility of N by D.

  If D is a divisor of N, then find the divisors of N/D, and so on... with N/D taking the role of N. The process stops when N = 1.

  As divisors are found, they are stored in the list FACT.

```
function facprimes(N)
local D FACT
2->D
{}-> FACT
While N ≠ 1  do
   If N mod D = 0 then
      FACT + D -> FACT
      N/D -> N
   Else
      D+1 -> D
   endif
endwhile
Result: FACT
endfunction
```

- First improvement

  One only tests the divisors D that are between 2 and FLOOR(√N).

  In effect, if N = D1* D2, then we can say:

  Let D1 ≤ FLOOR(√N), and let D2 ≤ FLOOR(√N); otherwise, we would have:

  D1 * D2 ≥ (FLOOR(√N) + 1)² > N.

```
function facprimes(N)
local D FACT
2-> D
{} -> FACT
While  D*D ≤ N do
   If  N mod D = 0 then
      FACT + D -> FACT
      N/D -> N
   Else
      D+1-> D
   endif
endwhile
FACT + N -> FACT
Result: FACT   (Result)
endfunction
```

- Second improvement

  One looks to see whether 2 is a divisor of N, then one tests only the odd divisors D that are between 3 and FLOOR(√N).

  In the list FACT, each divisor is written followed by its exponent:

  decomp(12)={2,2,3,1}.

```
function facprimes(N)
local K D FACT
{}->FACT
0 -> K
While N mod 2 = 0 do
   K+1 -> K
   N/2 -> N
```

```
   endwhile
   If K ≠ 0  then
      FACT + {2 K} -> FACT
   endif
   3-> D
   While D*D ≤ N do
      0 -> K
      While N mod D = 0   do
         K+1 -> K
         N/D -> N
      endwhile
      If K ≠ 0 then
         FACT + {D K} -> FACT
      endif
      D+2 -> D
   endwhile
   If N ≠ 1 then
      FACT + {N 1} -> FACT
   endif
   Result: FACT  (Result)
   endfunction
```

7.3.2 HP40G syntax

This is a translation of the last algorithm.

The HP40G calculator does not understand {}, so to initialise L1 as an empty list you must type: SUB L1;L1;2;1.

Here is the program FACTPRIMES:

```
INPUT N;"N";;;1:
ERASE:
0 -> K:
SUB L1;L1;2;1:
WHILE N MOD 2 == 0 REPEAT
   1+K -> K:
   N/2 ->N:
END:
IF K ≠ 0 THEN
   {2, K} -> L1:
END:
3->D:
WHILE D*D ≤ N REPEAT
   0 -> K:
   WHILE N MOD D == 0 REPEAT
      K+1 -> K:
      N/D -> N:
   END:
   IF K ≠ 0 THEN
      CONCAT (L1, {D, K}) -> L1:
   END:
   2+D -> D:
END:
IF N ≠ 1 THEN
   CONCAT (L1, {N,1}) -> L1:
END:
DISP 3; "FACT " L1:
FREEZE:
```

7.4 Calculating $A^P \, mod \, N$

7.4.1 Algorithmic syntax

- First algorithm

  This algorithm uses two local variables, POWER and I.

  Write an iterative program such that at each stage, POWER represents A^I mod N.

```
function powermod (A, P, N)
local POWER, I
1 -> POWER
for I  = 1 to P  do
   A*POWER mod N -> POWER
endfor
Result: POWER
endfunction
```

- Second algorithm

  This algorithm uses a single local variable POW, but it varies P such that at each stage of the iteration we have:

  $result = POW * A^P \pmod{N}$

```
function powermod (A, P, N)
local POW
```

```
1 -> POW
While P>0 do
   A*POW mod N -> POW
   P-1 ->P
endwhile
Result: POW
endfunction
```

- Third algorithm

  One can easily modify this program by taking into account that:

  $A^{2*P} = (A*A)^P$.

  Therefore when P is even, we have the relation:

  $POW * A^P = POW * (A * A)^{P/2}$ (mod N)

  and when P is odd, we have the relation:

  $POW * A^P = POW * A * A^{P-1}$ (mod N)

  We are left with a very fast algorithm for A^P mod N:

```
function powermod (A, P, N)
local POW
1->POW
While P>0 do
   If P mod 2=0 then
      P/2->P
      A*A mod N->A
   Else
      A*POW mod N ->POW
      P-1->P
   endif
endwhile
Result: POW
endfunction
```

It goes without saying that if P is odd, then P–1 is even.

One can then write:

```
function powermod (A, P, N)
local POW
1->POW
While P>0 do
   If P mod 2=1 then
      A*POW mod N->POW
      P-1->P
   endif
   P/2->P
   A*A mod N->A
endwhile
Result: POW
endfunction
```

7.4.2 HP40G syntax

The calculation of A^p mod N is treated in the program on the probability method of Mr Rabin. Please refer to the HP40G version of that program (7.6).

7.5 The function "isprime"

7.5.1 Algorithmic syntax

- Initial algorithm

  Write a Boolean function of N that is equal to TRUE when N is prime and FALSE when it's non-prime.

  For this, find whether N has a divisor ≠ 1 and ≤ FLOOR(√N) (the whole part of the square root of N).

  The case where N=1 is treated separately!

  Here, the Boolean variable PRIME is used, which is TRUE by default, and which is set to FALSE if a divisor of N is found.

```
Function isprime(N)
local PRIME, I, J
FLOOR(√N)->J
If N = 1 then
   FALSE->PRIME
else
   TRUE->PRIME
endif
2->I
While PRIME and I ≤ J do
```

```
      If N mod I=0 then
         FALSE->PRIME
      else
         I+1->I
      endif
   endwhile
   Result: PRIME
   endfunction
```

- First improvement

  Of course, we can test to see if N is even, then look only for odd divisors of N.

```
function isprime(N)
Local PRIME, I, J
FLOOR(√N)->J
If (N = 1) or (N mod 2 = 0) and (N≠2) then
   FALSE->PRIME
else
   TRUE->PRIME
endif
3->I
While PRIME and I ≤ J do
   If N mod I = 0 then
      False -> PRIME
   else
      I+2->I
   endif
endwhile
Result: PRIME
endfunction
```

- Second improvement

  We can test to see if N is divisible by 2 or 3, then look only for divisors of N that are of the form $6 \times k - 1$ or $6 \times k + 1$.

```
function isprime(N)
local PRIME, I, J
FLOOR(√N)->J
If (N = 1) or (N mod 2 = 0) or (N mod 3 = 0) then
   FALSE->PRIME
else
   TRUE->PRIME
endif
If N=2 or N=3 then
   TRUE->PRIME
endif
5->I
While PRIME and I ≤ J do
   If (N mod I = 0) or (N mod (I + 2) = 0) then
      FALSE->PRIME
   else
      I + 6 -> I
   endif
endwhile
Result: PRIME
endfunction
```

7.5.2 HP40G syntax

```
INPUT N;"N";;;1:
IF N MOD 2==0 OR N MOD 3==0 OR N==1 THEN
   0 ->P:
ELSE
   1->P:
END:
IF N==2 OR N==3 THEN
   1->P:
END:
5->I:
FLOOR(√N)->J:
WHILE I ≤ J AND P REPEAT
   IF N MOD I==0 OR N MOD (I+2)==0 THEN
      0->P:
   ELSE
      I+6->I:
   END:
END:
ERASE:
DISP 5;P:
FREEZE:
```

7.6 Mr Rabin's probability method

If N is prime, then all numbers K less than N are prime with N, so according to Fermat's Little Theorem we have:

$K^{N-1} = 1 \pmod N$

If N is not prime, the integers K such that:

$K^{N-1} = 1 \pmod N$

are very few indeed.

More precisely, one can show that if N > 4, the probability of finding such a number K is less than 0.25.

A number N such that $K^{N-1} = 1 \pmod N$ for 20 random tries of K is called a pseudo-prime number. The probability method of Rabin consists of choosing a random number K (1< K <N) and calculating:

$K^{N-1} \pmod N$

If $K^{N-1} = 1 \pmod N$, another random number is tried, whereas if $K^{N-1} \neq 1 \pmod N$, it is certain that N is not prime.

If $K^{N-1} = 1 \pmod N$ is obtained for 20 tries of K, one can conclude that N is prime with a very small probability of error—less than 0.25^{20}, or on the order of 10^{-12}.

Naturally, this method is used to test whether large numbers are pseudo-primes.

7.6.1 Algorithmic syntax

Let us suppose that:

Random(N) returns a random whole number between 0 and N – 1.

Calculation of:

$K^{N-1} \bmod N$

is carried out using the very fast algorithm developed earlier (7.4.1, 3rd algorithm).

Note the statement:

`powermod(K, P, N)`, the function that calculates $K^P \bmod N$.

```
Function isprime(N)
local K, I, P
1->I
1->P
While P = 1 and I < 20 do
   Random(N-2)+2->K
   powermod(K, N-1, N)->P
   I+1->I
endwhile
If P =1  then
   Result: TRUE
else
   Result: FALSE
endif
endfunction
```

7.6.2 HP40G syntax

```
INPUT N;"N";;;1:
RANDSEED TIME:
1->I:
1->P:
WHILE I < 20 AND P==1 REPEAT
   FLOOR (RANDOM * (N-2))+2->K:
   N-1->M:
   @Calculate K ^ M mod N and store it in P.
   1->P:
   WHILE 0 < M REPEAT
      IF M MOD 2 == 0 THEN
         M / 2 -> M:
         (K * K) MOD N ->K:
      ELSE
         K*P MOD N ->P:
         M - 1 ->M:
      END:
   END:
   @P contains K ^ M mod N and M=N-1.
   I+1 ->I:
END:
ERASE:
IF P==1 THEN
   DISP 3; "PRIME " N:
ELSE
   DISP 3; "NOT PRIME " N:
END:
FREEZE:
```

Remark:

You can also use the computer algebra function POWMOD, substituting:

```
MODSTO(N):
POWMOD(K,N-1) -> P:
```

for the statements between the comments (@). This results in:

```
PROMPT N:
RANDSEED TIME:
1->I:
1->P:
WHILE I < 20 AND P==1 REPEAT
   FLOOR(RANDOM * (N-2))+2->K:
   MODSTO(N):
   POWMOD(K,N-1) -> P:
   I+1 ->I:
END:
ERASE:
IF P==1 THEN
   DISP 3;"PRIME " N:
ELSE
   DISP 3;"NOT PRIME " N:
END:
FREEZE:
```

8 GNU Free Documentation License

Version 1.1, March 2000

Preamble
The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially.

Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense.

It complements the GNU General Public License, which is a copyleft license designed for free software. We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is command or reference.

Applicability and Definitions
This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

1. Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3. You may also lend copies, under the same conditions stated above, and you may publicly display copies.

2. Copying in Quantity

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects. If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages. If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public. It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

3. Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

a) Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

b) List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

c) State on the Title page the name of the publisher of the Modified Version, as the publisher.

d) Preserve all the copyright notices of the Document.

e) Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

f) Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

g) Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

h) Include an unaltered copy of this License.

i) Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

j) Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

k) In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

l) Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

m) Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

n) Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section. If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles. You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties, for example statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard. You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

4. Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice. The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History” in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

5. Collections of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects. You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

6. Aggregation with Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document. If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

7. Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

8. Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. Future Revisions of this License

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified

version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.