gu

no.42

analytical and numerical methods with the hp 48 g/g+/gx programmable calculator

gilberto e. urroz, ph. d., p.e.

ANALYTICAL AND NUMERICAL METHODS WITH THE HP 48 G/G+/GX PROGRAMMABLE CALCULATOR

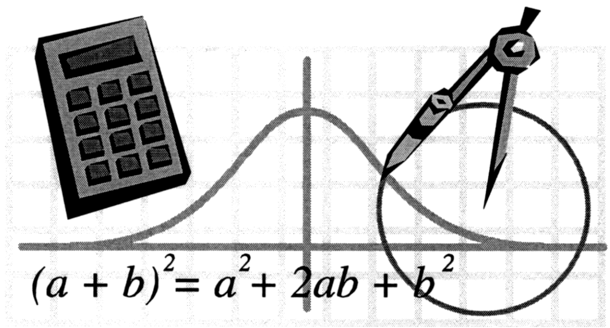
ISBN 1-58898-042-1

ANALYTICAL AND NUMERICAL METHODS WITH THE HP 48 G/G+/GX PROGRAMMABLE CALCULATOR

Gilberto E. Urroz, Ph.D., P.E.

greatunpublished.com
Title No. 42
2000

ANALYTICAL AND NUMERICAL METHODS WITH THE HP 48 G/G+/GX PROGRAMMABLE CALCULATOR



By

Gilberto E. Urroz, Ph.D., P.E.
Associate Professor
Department of Civil and Environmental Engineering
Utah State University

Working with complex numbers in the HP48G/GX calculator

Refer to the book entitled "***HP 48G Series** User's Guide*" which is provided with your HP48G or GX calculator.

Subject:	*Located in:*
General references	
Entering complex numbers	Ch. 2, p. 2-6
Setting the coordinate mode	Ch. 4, p. 4-4 to 4-5
How to access complex functions	Ch. 11, p. 11-6
Chapter on COMPLEX NUMBERS	Chapter 12, starting in page 12-11.
Cartesian and polar representation	Ch. 12, p. 12-11
Entering complex numbers	Ch. 12, p. 12-12
Complex results from real calculations	Ch. 12, p. 12-13
Complex number commands	Ch. 12, p. 12-13 and 12-14
Absolute value or module	Ch. 12, p. 12-13 (ABS)
Argument	Ch. 12, p. 12-13 (ARG)
Complex conjugate	Ch. 12, p. 12-13 (CONJ)
Real part	Ch. 12, p. 12-14 (RE)
Imaginary part	Ch. 12, p. 12-14 (IM)
Applications that may use complex numbers	
Fast Fourier Transforms	Ch. 13, p. 13-7 and 13-8
Matrices	Ch. 14, p.14-13 and 14-14
Roots of polynomials	Ch. 18, p. 18-10 and 18-11
Roots of the quadratic equation.	Ch. 20, p. 20-15 and 20-16

Other subjects covered or mentioned in class that are available in the HP48 G series calculators:

Subject:	*Located in:*
Binary Arithmetic and Number Bases	Ch. 15, p. 15-1 to 15-5.

Plotting complex functions

We define a complex variable z as $z = x + iy$, where x and y are real variables, and $i = (-1)^{1/2}$. We can also define another complex variable $w = F(z) = \Phi + i\Psi$, where, in general, $\Phi = \Phi(x,y)$, and $\Psi = \Psi(x,y)$, are two real functions of x,y. These real functions can also be given in terms of the polar coordinates r,θ if we use the polar representation for z, i.e., $z = r\cdot e^{i\theta} = r(\cos\theta + i\cdot\sin\theta)$. In such case, $\Phi = \Phi(r,\theta)$, and $\Psi = \Psi(r,\theta)$. Recall that the coordinate transformations between Cartesian and polar coordinates are:

$$r = (x^2 + y^2)^{1/2}, \qquad \tan\theta = y/x,$$
$$x = r\cos\theta, \qquad y = r\sin\theta.$$

The complex variable w is also known as a *complex function*. Another name for a complex function is "*mapping*." Thus, we say F(z) is a mapping of z. In geometric terms, this means that any figure in the x-y plane gets "mapped" onto a different figure on the Φ–Ψ plane by the complex function F(z).

As an example, take the function $w = F(z) = \ln(z) = \ln(r\cdot e^{i\theta}) = \ln(r) + i\theta$. We can identify the functions $\Phi = \Phi(r,\theta) = \ln(r)$, and $\Psi = \Psi(r,\theta) = \theta$. Therefore, curves of the form $\Phi(r,\theta) = \ln(r) = \ln(K_1)$, where K_1 is a constant, represents circles having the equation $r = K_1$ as shown in Figure 1(a). On the other hand, curves of the form, $\Psi(r,\theta) = \theta = K_2$, where K_2 is a constant, represent radial lines emanating from the origin as shown in Figure 1(b).

Figure 1 (a). Curves of r = K_1, representing constant values of Φ(r, θ).

Figure 1(b). Curves of q = K_2, representing constant values of Ψ(r, θ).

You can produce these "mappings" using the Gridmap Plots available in the PLOT environment of your HP48G/GX Programmable Calculator. The following example shows how to produce the mapping of the ln(z) function:

[↱][PLOT]	Enter the HP48G/GX PLOT environment
	Highlight the filed in front of `TYPE:`, then press
[CHOOS]	Use the up and down arrow keys ([▲] [▼]) and select `Gridmap`, then press
[OK] [▼]	Now, enter the complex function to be plotted, in this case enter:
['][↱][LN]	Define ln( ) function,
[↰][( )]	Recall that a complex number is written as an ordered pair
[α][X]	Enter X
[↰][,]	Enter the comma
[α][Y]][+][.][0][0][1]	Enter Y
[ENTER]	Enter the function
	We will keep the current definitions of independent (INDEP) and dependent (DEPND) variables, as well as the number of steps for each of them. We will, however, modify the display coordinates, by entering
[OPTS]	Use the arrow keys ([◄][►][▲][▲]) to modify the parameters to read as follows (press [OK] after entering a new value for any of the parameters):

```
X-LEFT: -1  X-RIGHT: 1
Y-NEAR: -1  Y-FAR:   1
XX-LEFT:-1  XX-RGHT: 1
YY-NEAR:-1  YY-FAR:  1
```

	When done, press
[OK]	This will send you back to the PLOT environment. To plot use:
[ERASE][DRAW]	Because you are plotting a number of curves it will take a while for your calculator to finish the plot, just be patient. To see labels enter:

[EDIT][NXT]	
[LABEL][MENU]	To recover menu, use:
[NXT]	To recover PICT menu use:
[NXT][PICT]	To see coordinates of cursor, enter:
[(x,y)]	Use the arrow keys ([◄][►][▲][▲]) to move the cursor around the screen.
	To trace a particular curve, use this:
[NXT][TRACE][(x,y)]	Use the arrow keys ([◄][►][▲][▲]) to move the cursor around the screen. The cursor moves only on curves defined by the gridmap. The bottom of the screen shows a value called INPUT with the x and y coordinates of the cursor position.
	To cancel the plot and return to the PLOT environment, use:
[NXT][CANCL]	To return to normal display use:
[ENTER]	

Exercises: Plot the mappings corresponding to the following functions:

(1) SIN((X,Y))	i.e., $F(z) = \sin(z)$
(2) SQ((X,Y)) or (X,Y)^2	i.e., $F(z) = z^2$
(3) EXP((X,Y))	i.e., $F(z) = e^z$
(4) SINH((X,Y))	i.e., $F(z) = \sinh(z)$
(5) TAN((X,Y))	i.e., $F(z) = \tan(z)$
(6) ATAN((X,Y))	i.e., $F(z) = \tan^{-1}(z)$
(7) (X,Y)^3	i.e., $F(z) = z^3$
(8) INV((X,Y)) or 1/(X,Y)	i.e., $F(z) = 1/z$
(9) √ (X,Y)	i.e., $F(z) = z^{1/2}$

All of the functions listed above represent *analytic* functions, i.e., they satisfy the Cauchy-Riemman conditions. Notice that the lines in all the figures cross each other at about 90°. This is a property of analytic functions. In other words, if Φ and Ψ are the real and imaginary parts, respectively, of an analytic complex function, lines of $\Phi(x,y)$ = constant, and $\Psi(x,y)$ = constant are orthogonal to each other.
The "mappings" achieved through analytic functions are such that the angles between lines in the function plane (Φ,Ψ) are preserved in the independent variable plane (x,y). Such mappings or transformation are said to be *conformal* (i.e., angle preserving). The application of complex variables is, therefore, referred to as *conformal mapping*.

For additional information on Gridmap Plots refer to Chapter 23, pages 23-34 and 23-35 of the ***HP 48G Series*** *User's Guide.*

Matrices and Linear Algebra

The HP48G or GX calculator has a number of operations available for application in numerical methods problems. Details on the use of the calculator for specific applications can be found in the book ***HP 48G Series*** *User's Guide* that is provided with the calculator. The following table shows a list of subjects related to linear algebra, their location in the textbook (Hoffman) and the corresponding chapters in the HP48 G Series User's Guide. I will recommend that you start browsing through those pages to get yourself acquainted with the matrix and linear algebra problems that will be covered after the Mid-term exam on October 19.

Subject in Numerical Methods Textbook	Presented in HP48G Series User's Guide in:
To enter and edit matrices or vectors	Ch. 8 - The Matrix Editor / Ch. 14, p. 14-1 to 14-2.
To create matrix/array filled with constants.	Ch. 14, p. 14-2
Diagonal matrix, p. 14	Ch. 14, p. 14-3
Identity matrix, p. 14	Ch. 14, p. 14-2
Transpose of a matrix, p. 15	Ch. 14, p. 14-10
Matrix addition, p. 16	Ch. 14, p. 14-11
Matrix product, p. 17	Ch. 14, p. 14-11
Multiplication by scalar, p. 17	Ch. 14, p. 14-11 (including change of signs, and product of matrix and vector)
Inverse matrix, p. 18	Ch. 14, p. 14-10
Solution of linear equations, p. 19 and on, and Matrix inverse method, p. 34-35	Ch. 14, p. 14-14 to 14-17/ Ch. 18, p. 18-11 to 18-13
Determinants, p. 20-22, 35-37	Ch. 14, p. 14-9
Cramer's Rule (replacing columns in matrices)	Appendices, pages G-37 and H-22
Row operations, p. 24	Ch. 14, p. 14-18 and 14-19
Augmented matrix, p. 27	Ch. 14, p. 14-18
Ill-conditioned systems, p. 37	Ch. 14, p. 14-16
LU decomposition, p. 37-41	Ch. 14, p. 14-21 (Crout method) Ch. 14, p. 14-21 to 14-25 (other methods)
Eigenvalues and eigenvectors, Ch. 2, p. 64 and on.	Ch. 14, p. 14-20 and 14-21.

The following table includes subjects not covered in the textbook.

Subjects of general interest in linear algebra not presented in textbook but available in HP48G User's Guide

Frobenius norm	Ch. 14, p. 14-8 (ABS)
Spectral norm	Ch. 14, p. 14-8 (SRNM)
Row norm	Ch. 14, p. 14-8 (RNRM)
Column norm	Ch. 14, p. 14-8 (CNRM)
Spectral radius	Ch. 14, p. 14-9 (SRAD)
Column-norm condition number	Ch. 14, p. 14-9 (COND)
Rank	Ch. 14, p. 14-9 (RANK)
Trace	Ch. 14, p. 14-9 (TRACE)
Accuracy of a matrix solution (not in textbook)	Ch. 14, p. 14-17 / Ch. 18, p. 18-12

Programming the HP48G/GX calculator - a primer

Under sub-directory NUMM\MATX you will find a variable called

[→ABSM]

(shows as →ABS). This variable contains a program that calculates the so-called Frobenius norm of a matrix **A**, ||**A**||. As an example, enter the matrix:

[[1. 5. 3.] [2. 4. -1.] [3. 0. 2.]],

press [ENTER] to keep a second copy of the matrix available, then, press [→ABS].

The result is 8.30662386292.

Swap the contents of the stack ([↰][SWAP]) and enter the keystroke sequence

[MTH][MATR][NORM][ABS].

This will give you the same result as the program →ABSM. I included the variable [→ABS] to introduce you to programming the HP48G/GX calculator.

The calculator uses a programming language referred to as *User's RPL*. Some details on the programming constructs for User's RPL is presented in chapter 29 of the ***HP 48 G Series Calculator*** *User's Guide*. The different constructs can be accessed by pressing the [PRG] button, then pressing the [BRCH] white key (BranCHing, as in branching programs). You will have the following options available:

[IF][CASE][START][FOR][DO][WHILE]

corresponding to six different programming constructs. For example, if you press [IF] you will get the following options:

[IF][THEN][ELSE][END][ ][BRCH]

which are commands in the User's RPL language. You recognize them as the same FORTRAN commands for an IF... THEN..ELSE...END IF construct.

Let's take a look at the program →ABS itself. Press [VAR][↱][→ABS]. The display will show the following screen:

```
1: << DUP 'a' STO SIZE
   OBJ→ DROP 'm' STO
   'n' STO 0 1 n FOR i
   1 m FOR j 'a(i,j)'
```

To see the entire program, press [▼] or [↰][EDIT], and press the [▼] button until reaching the bottom line of the program. Press [ENTER] to return to normal display.

A listing of the program, with the keystrokes required to enter each command is presented below:

List	Keystrokes	Action
<<	[↰][<<>>]	Starts the program prompt << ... >>
DUP	[↰][ENTER]	Duplicates contents of level 1, i.e., matrix **A**
'a' STO	['][α][↰][A][▶][STO]	Stores contents of level 1 into variable a
SIZE	[MTH][MATR][MAKE][SIZE]	Determines size of matrix A, as {n m}
OBJ→	[PRG][TYPE][OBJ→]	Decomposes the list {n m} into n, m, 2
DROP	[↰][⇦]	Drops contents of level 1, n is in 2, and m in 1
'm' STO	['][α][↰][M][▶][STO]	Stores contents of level 1 into variable m, n goes to 1
'n' STO	['][α][↰][N][▶][STO]	Stores contents of level 1 into variable m
0	[0]	We will calculate a sum. It is set at zero at this point
1 n FOR i	[1][SPC][α][↰][N][PRG][BRCH][FOR][FOR][α][↰][I]	Start a FOR loop for i = 1 to n
1 n FOR j	[1][SPC][α][↰][N][PRG][BRCH][FOR][FOR][α][↰][J]	Start a nested FOR loop for j = 1 to m
'a(i,j)' EVAL	['][α][↰][A][↰][()][α][↰][I][↰][,][α][↰][J][▶][EVAL]	Extract the element a_{ij} from matrix a
ABS SQ	[MTH][REAL][NXT][ABS][↰][x^2]	Takes absolute value of a_{ij} and squares it.
+	[+]	Adds to value in level 1 (originally a 0 was set there)
NEXT	[PRG][BRCH][FOR][NEXT]	Closes the loop with index j
NEXT	[NEXT]	Closes the loop with index I
√	[$\sqrt{x}$]	Takes square root of sum of $\lvert a_{ij}\rvert^2$
{ m n a } PURGE	[↰][{ }][α][↰][M][SPC][α][↰][N][SPC][α][↰][A] [▶][↰][PURG]	Deletes the variables m, n, and a
>>	[ENTER]	End program .

Press [ENTER] to see program in level 1. If you had been typing the program for the first time, you will then save it under the name →ABSM by using:

[VAR]['][↰][→][⇦][◀][⇦][α][α][A][B][S][M][ENTER][STO]

To see the program in action, step by step, enter a matrix in level 1, say [[1. 5. 3.] [2. 4. -1.] [3. 0. 2.]]. Then, press ['][→ABS], which will copy the program name into level 1 between quotes. Next, use the keystroke sequence:

[PRG][NXT][RUN][DBUG],

which starts the *debugger* option in RUN. Then, press [SST↓] repeatedly. At each pressing of [SST↓] you will see the current command displayed in the upper left corner of the calculator's display, and the display changing accordingly. Keep pressing the [SST↓] button until the upper left corner displays >>, which indicates the end of the program. Then, press [KILL] to end the debugging process.

There are a variety of programs in the sub-directory NUMM that you can list and debug using a procedure similar to that shown above for program →ABSM. Some of the simplest programs that you can create, without having to key in every program statement is the definition of functions. Some details are presented below.

Functions, programs and lists

Defining a function consists in creating a variable that contains the definition of the function. For example, suppose we want to define the function

There are two ways to enter the expression that defines the function:

- Using the equation editor:

  [↰][EQUATION] [α][↰][F] [↰][()] [α][↰][X] [▶] [↰][=] [α][↰][X] [y^x][3] [ENTER]

- Entering the expression directly into level 1 of the display:

  ['] [α][↰][F] [↰][()] [α][↰][X] [▶] [↰][=] [α][↰][X] [y^x][3] [ENTER]

To define the function, use the keystroke sequence: [↰][DEF].

There will be a new white key named F, i.e., [F]. To see how the function is stored into the variable f, press [↱][F]. Level 1 of the display now shows the following:

```
<< → x 'x^3' >>
```

This expression represents an RPL program that can be interpreted as follows*: take the value in level 1 and assign it to x (→ x), then, calculate the expression between quotes ('x^3 ').* The result is then placed in level 1 of the display.
Notice that the function definition procedure used above (namely, using the keystroke sequence [↰][DEF]), translates the function definition (in general, 'f(x) = expression containing x') into a program of the form

<< → x 'expression containing x'>>.

Please notice that RPL programs are always enclosed within the double quotes << >>.

Let's now evaluate our function f(x) for a given value of x, say, x = 5. First, you need to enter the value of x in level 1 of the display by pressing [5][ENTER]. Next, press the white button for F, i.e., [F]. The result is 125, or f(5) = 125.
We could evaluate the function using a list as the argument. (A list, in the HP48G/GX calculator, is a collection of objects contained between curly brackets {}. We will limit our discussion to lists of numbers.) For example, let's find the value of f({1 2 3}). First, enter the list in level 1 of the display:

[↰][{}] [1] [SPC] [2] [SPC] [3] [ENTER]

Then, press the button [F]. The result is {1 8 27}, i.e., f[{1 2 3}] = {1 8 27}. Notice that the program defining f(x) applied the function definition (namely, x^3) to each element of the argument list and produced a new list with the same number of elements.
Let's now try defining a different function,

Use the following keystroke sequence to create the variable g:

['] [α][↰][G] [↰][()] [α][↰][X] [▶] [↰][=] [MTH][HYP][SINH] [α][↰][X]

[▶] [÷][↰][()] [1] [+][α][↰][X] [y^x][2] [ENTER] [↰][DEF].

Now, evaluate g(3.5), by entering the value of the argument in level 1 ([3][.][5][ENTER]) and then pressing [G]. The result is 1.2485..., i.e., g(3.5) = 1.2485... Try also obtaining g[{1 2 3}], by entering the list in level 1 of the display ([↰][{}] [1] [SPC] [2] [SPC] [3] [ENTER]), and pressing [G]. In this case we get a division error message. The calculator display looks as follows:

```
/ Error:
Invalid Dimension
4:
3:
2: { 1.17520119364 3. …
1:              {1 1 4 9}
```

Notice that the list in level 1 has four elements. Now, drop that list by pressing [⇦]. The list now in level 1 has only three elements. Apparently, the program defined by [G] tried to perform a division between two lists that have different number of elements, and that is what causes the division error reported. (Most arithmetic operations between lists require lists with the same number of elements).

How did we end up with lists of different sizes if we started with a single list of size 3? The explanation lies in the way that the operator [+] applies to lists. The operator [+], when applied to lists, does not act as a simple addition, instead it is known as the *concatenation* operator and its function is to attach (or concatenate) the two lists. For our particular situation, the list x = {1 2 3} is squared in the denominator ($1+x^2$) of the expression that defines g(x) producing the list {1 4 9}. Then, this list is concatenated to the number 1 by way of the operator [+] in the denominator ($1+x^2$). This result in the list {1 1 4 9} = 1 + {1 4 9} shown in level 1 of the display (see above). The list in level 2 of the display represents SINH(x). In order to calculate the function g(x), the calculator tries to divide SINH(x), which is a list of three elements, by (1+x2), which is a list of four elements, which results in the division error reported.

The moral of this exercise is the following: when trying to evaluate a function whose definition includes the plus (+) sign, if the function argument is a list the result most likely will produce an algebraic error. For example, when we evaluated f[{1 2 3}], we obtained a reasonable result (namely, {1 8 27}), because the definition of the function ($f(x) = x^3$) did not include any plus signs.

To get around using the [+] sign, the HP48G/GX calculator offers the function ADD that applies particularly to lists, but that can be used to represent addition of other numbers. Instead of using the definition currently stored for the variable [G], namely, `<< → x 'SINH(x)/(1+x^2) >>`, we'll replace it with the following program: `<< 'x' STO x SINH 1 x SQ ADD / 'x' PURGE >>`. To key in the program follow these instructions:

Keystroke sequence:	Produces:	Interpreted as:
[↰] [<< >>] [ENTER]	<<	Start an RPL program
['] [α] [↰] [X] [▶] [STO]	'x' STO	Store the contents of level 1 into variable x
[α] [↰] [X]	x	Place x in level 1
[MTH][HYP][SINH]	SINH	Calculate sinh of level 1
[1] [SPC] [α] [↰] [X] [↰] [x^2] [MTH] [LIST] [ADD]	1 x SQ ADD	Calculate ($1+x^2$)
[÷]	/	
['] [α] [↰] [X] [▶] [↰] [PURGE]	'x' PURGE	Purge variable x
[ENTER]		Program displayed in level 1

To save the program use:

['] [α] [↰] [G] [STO]

Now, evaluate g(3.5) by entering the value of the argument in level 1 ([3][.][5][ENTER]) and then pressing [G]. The result is 1.2485..., i.e., g(3.5) = 1.2485...Same as before. Try also obtaining g[{1 2 3}], by entering the list in level 1 of the display ([↰][{}] [1] [SPC] [2] [SPC] [3] [ENTER]), and pressing [G]. The result now is {0.6551 ... 0.8345... 1.248...}.

More operations with lists

As indicated above, lists are collections of objects such as numbers, letters, character strings, variable names, and/or operators contained between curly brackets. Some examples of lists are: { t 1 }, {"BETA" h2 4}, {1 1.5 2.0}, {a a a a}, { {1 2 3} {3 2 1} {1 2 3}}, etc. For applications in this class we will limit ourselves to numerical lists.

Simple operations with lists are accessible by using the keystroke sequence: [MTH][LIST].
The operations thus available are:

[ΔLIST][ΣLIST][ΠLIST][SORT][REVLI][ADD]

We already showed how to use [ADD]. The other operations in this set are defined as follows:

[ΔLIST]: produces a list of increments between consecutive elements in the original list. E.g., {1 2.1 3.5 4.2 3.8} [ΔLIST] produces {1.1 1.4 0.7 -0.4}.

[ΣLIST]: calculates the sum of the elements in the list. E.g., {1 2.1 3.5 4.2 3.8} [ΣLIST] produces 14.6.

[ΠLIST]: calculates the product of the elements in the list. E.g., {1 2.1 3.5 4.2 3.8} [ΠLIST] produces 117.306.

[SORT]: sorts elements in increasing order. E.g., {1 2.1 3.5 4.2 3.8} [SORT] produces {1 2.1 3.5 3.8 4.2}.

[REVLI]: reverses order of elements in list. E.g., {1 2.1 3.5 4.2 3.8} [REVLI] produces {3.8 4.2 3.5 2.1 1}

Additional operations with lists are available by pressing [PRG][LIST]. In particular, we will describe the use of the operations [OBJ→] and [→LIST].

[OBJ→] is used to decompose a list into its elements. By using this operation the elements of the list are placed in different levels of the display, with level 1 displaying the number of elements, level 2 displaying the last element on the list, level 3 displaying the second-to-last element on the list, and so on. For example, {0.5 3.2 6.5} [PRG][LIST][OBJ→] produces the following display:

```
4:          0.5
3:          3.2
2:          6.5
1:            3
```
(← number of elements)

[→LIST] is used to create a list using the values available in the different display levels. The number of elements in the list must be placed in level 1. The value in level 2 will become the last element in the list, the value in level 3 will become the second-to-last element in the list, and so on. For example, the following display:

```
4:          0.5
3:          4.2
```

```
2:                    8.5
1:                      2
```
(← number of elements)

will produce this list: {4.2 8.5} when the keystroke sequence [PRG][LIST][→LIST] is used.

Notice that [OBJ→] and [→LIST] are inverse operations.

For more information on list operations refer to section 17 in the ***HP 48G Series*** *User's Manual* that is included with your calculator. Notice that lists of number can be treated as one-dimensional arrays.

Direct Solution of Systems of linear equations using the HP48G/GX calculator

To solve, for example, the system:

$$v_A + v_C = 0,$$
$$v_A - 3v_C = -32$$

it can be re-written, using matrices and vectors, as

Or, **Ax** = **b**, where **A** is the 2x2 matrix shown above, **x** = $[v_A\ v_C]^T$, and **b** = $[0\ -32]^T$.
We can solve this system of linear equations by using the SOLVE environment in the HP48G or GX. For this example, use the following keystroke sequence:

[↱][SOLVE][▼] [▼][▼][OK]	Select *Solve lin system*
[↱][MATRIX][1][SPC][1][ENTER]	Enter matrix **A**
[▼][1][SPC][3][+/-][ENTER][ENTER]	
[▼][↰][[ ]][0][SPC][3][2][+/-][OK]	Enter vector **b**
[SOLVE]	Solve for vector **x**. The solution is **x** = [-8 8].

The solutions are displayed in level 1.

From the equation **Ax** = **b**, although not mathematically correct, we can write **x** = **b**/**A** (the formal expression would be **x** = $\mathbf{A}^{-1}$ **b**.) This operation can be performed directly in the HP48G or GX, as follows:

[↰][[ ]][0][SPC][3][2][+/-][ENTER]	Enter vector **b**
[↱][MATRIX][1][SPC][1][ENTER]	Enter matrix **A**
[▼][1][SPC][3][+/-][ENTER][ENTER]	
[÷]	Calculate **x** = **b**/**A**.

Direct Gauss-Jordan simplification using the HP48G/GX

The function [RREF], obtained by using: [MTH][MATR][FACTR][RREF], will provide the final form of the augmented matrix simplified by using the Gauss-Jordan simplification. For example, if you enter the augmented matrix:

[[2 3 5 1][3 1 -2 3][1 3 4 5]][ENTER]

into display level 1 and press [MTH][MATR][FACTR][RREF], you will get as a result the simplified augmented matrix:

[[1 0 0 -2]
[0 1 0 5]
[0 0 1 -2]]

Therefore, the solution to the system of linear equations represented by the original augmented matrix is **x** = $[-2\ 5\ -2]^T$.

Solving polynomial equations in the HP48G/GX calculator

A polynomial equation is an equation of the form:

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0.$$

The fundamental theorem of algebra indicates that there are *n* solutions to any polynomial equation of order *n*. Some of the solutions could be complex numbers, nevertheless. As an example, solve the equation:

$$3s^4 + 2s^3 - s + 1 = 0.$$

We want to place the coefficients of the equation in a vector [a_n a_{n-1}... a_1 a_0]. For this example, that vector will be [3 2 0 -1 1]. To solve for this polynomial equation using the HP48G or GX, try the following:

[↱][SOLVE][▼][▼][OK]	Select `Solve poly...`
[↰][[ ]][3][SPC][2][SPC][0]][SPC][1][+/-][SPC][1][OK]	Enter the vector of coefficients
[SOLVE]	Solve for s
[ENTER]	Returns to stack.
[PRG][TYPE][OBJ→][⇦][OBJ→] [⇦]	All solutions will be listed in the stack.

All the solutions are complex numbers: (0.432,-0.389), (0.432,0.389), (-0.766, 0.632), (-0.766, -0.632). Complex numbers in the HP48G or GX are represented as ordered pairs, with the first number in the pair being the real part, and the second number, the imaginary part. For example, the number (0.432,-0.389), a complex number, will be written normally as $0.432 - 0.389i$, where i is the imaginary unit, i.e., $i^2 = -1$.

Note: The fundamental theorem of algebra indicates that there are n solutions for any polynomial equation of order n. There is another theorem of algebra that indicates that if one of the solutions to a polynomial equation with real coefficients is a complex number, then the conjugate of that number is also a solution. In other words, complex solutions to a polynomial equation with real coefficients come in pairs. That means that polynomial equations with real coefficients of odd order will have at least one real solution.

Solving quadratic equations in the HP48G/GX calculator

Quadratic equations can result from expanding the characteristic equation for a 2×2 matrix. The HP48G/GX calculator offers an option for the direct solution of quadratic equations as described below.

For example, suppose that you need to solve the equation: $103.92\ t - 4.903\ t^2 = 518$, or,

$$-4.903t^2 + 103.92t - 518 = 0$$

To use the HP48G or GX calculator to solve this quadratic equation, use the following keystrokes:

[↱][SYMBOLIC][▲][▲][OK]	Select *Solve quad...*
[4][.][9][0][3][+/-][×][α][↰][T][y^x][2]	Type in the quadratic expression
[+][1][0][3][.][9][2][×][α][↰][T][-][5][1][8][OK]	
[α][↰][T]	Define the variable to solve for.
[OK]	Solves the quadratic equation symbolically.

The result shown is:

```
't = (-103.920+s1*25.305)/-9.806'.
```

The variable s1 should be given the values of 1 and -1 to get the two solutions. To obtain the actual solutions, use the following keystrokes:

[ENTER]	Copy the expression in the display.
[1]['][α][↰][S][1][STO]	Store the value 1 into s1.
[EVAL]	For s1 = 1, t = 8.017.
[1][+/-][α][↰][S1]	Store the value -1 into s1.
[↰][SWAP]	Swap level 1 with level 2.

[EVAL] For s1 = -1, t = 13.178.

In the SYMBOLIC environment, when solving for a quadratic equation, you can select the principal solution, which corresponds to s1 = 1, by placing a check mark (✓) in front of the word `PRINCIPAL`. For this case, the principal solution is t = 8.017.

An alternative way to solve the quadratic equation is to enter the equation into the display, then enter the solution variable, and press [↰][SYMBOLIC][QUAD].

For example, type in the expression: '-4.903*t^2+103.92*t-518=0'. Type the variable: 't'. Press [↰][SYMBOLIC][QUAD].

Calculation of eigenvalues and eigenvectors using the HP48G/GX

The equation

$$\mathbf{Ax} = \lambda\mathbf{x} \qquad [E1]$$

where **A** is a square matrix, **x** is a column vector, and λ is a scalar value, appears commonly in the analysis of certain physical systems (e.g., stresses at a point in continuous mechanics, multiple particles subjected to harmonic motion), and in the numerical solution of differential equations.

The values of λ that satisfy the equation shown above are known as the *eigenvalues* of the matrix **A**. The eigenvalues are obtained by re-writing the equation as

$$(\mathbf{A}-\lambda\mathbf{I})\mathbf{x} = 0. \qquad [E2]$$

This equation has the trivial solution $\mathbf{x} = \mathbf{0}$, regardless of the value of λ. The actual eigenvalues of **A** are obtained by realizing that the only way that equation [E2] will have a non-trivial solution is if the determinant of the accompanying matrix is zero.

The resulting equation, known as the *characteristic equation* for matrix A, is

$$\det(\mathbf{A}-\lambda\mathbf{I}) = 0 . \qquad [E3]$$

For example, for a 2×2 matrix **A**, the characteristic equation is given by $\lambda^2 - tr(\mathbf{A})\lambda+\det(\mathbf{A}) = 0$, where tr(**A**) is the trace of **A**.

The solutions to the characteristic equation are the eigenvalues of the matrix **A**. For every value of the eigenvalues λ there will be an *eigenvector* **x** obtained by solving equation [E1].

The HP48G/GX calculator provides functions for calculating the eigenvalues and eigenvectors as illustrated below. For example, to calculate the eigenvalues of

A = [[1 2 3 1][2 1 2 -2][3 2 5 -1][1 3 -2 6]],

enter the matrix in level 1 of the calculator, and press [MTH][MATR][NXT][EGVL].

The results are

[-1.06 7.71 0.90 5.45].

To calculate the eigenvectors and eigenvalues of a matrix, enter the matrix in level 1 of the display, then press [MTH][MATR][NXT][EGV]. The results of this operation includes a vector with eigenvalues in level 1 and a matrix whose columns are the eigenvectors corresponding to the eigenvalues in level 1.

For example, for the matrix [[1 2 3 1][2 1 2 -2][3 2 5 -1][1 3 -2 6]], the results are:

2:[[1.00 0.59 -0.32 0.44][-0.65 0.47 1.00 -0.08][-0.27 1.00 -0.43 0.38][0.06 0.00 -0.69 1.00]] (eigenvectors)
1: [-1.06 7.71 0.90 5.45] (eigenvalues)

Therefore, the eigenvalues and eigenvectors are:

$\lambda_1 = -1.06$, $\mathbf{x}_1 = [1.00\ -0.65\ -0.27\ 0.06]^T$,
$\lambda_2 = 7.71$, $\mathbf{x}_2 = [0.59\ 0.47\ 1.00\ 0.00\ ]^T$,
$\lambda_3 = 0.90$, $\mathbf{x}_3 = [\ -0.32\ 1.00\ -0.43\ -0.69]^T$,
$\lambda_4 = 5.45$, $\mathbf{x}_4 = [0.44\ -0.08\ 0.38\ 1.00\ ]^T$.

Check the solutions by verifying that $\mathbf{Ax} = \lambda\mathbf{x}$. For example, for the first eigenvalue and eigenvector, we find $\mathbf{Ax}_1 = [-1.06\ 0.69\ 0.29\ -0.06] = \lambda_1\mathbf{x}_1$. You can check on your own that the other eigenvalue-eigenvector combinations satisfy the eigenvalue equation $\mathbf{Ax} = \lambda\mathbf{x}$.

As an exercise, use [MTH][MATR][NXT][EGVL] to verify the eigenvalues of the matrix

A = [[2 3 5][3 1 -2][5 -2 4]]

are 8.12, 3.34, and -4.46.

Two-dimensional graphics in the HP48G/GX calculator

Plotting a curve given by an expression of the form y = f(x)

In this section we present an example of an HP48G/GX plot of a function of the form y = f(x). In order to proceed with the plot, first, purge the variable x, if it is defined in the current directory (because x will be the independent variable in the calculator's PLOT feature, you don't want to have it pre-defined). Create a sub-directory called 'TPLOT' (for test plot), or other meaningful name, to perform the following exercise.

Let's plot the function,

- Define the function using the following keystrokes:

  ['][⬑][e^x][-][α][⬑][X][y^x][2][÷][2][▶][÷][√][⬑][( )][2][×] [⬑][π][ENTER]

- Store it into a variable called y:

  ['][α][⬑][Y][α][STO]

- Enter the PLOT environment, [↱][PLOT], and select `Function` as the `TYPE`.
- Highlight the `EQ` field, and press [CHOOS]. Use the [▲] [▼] keys to select y, then press [OK].
- Type x (lowercase) as the independent variable (`INDEP`) with range -4 to 4.
- Place a check mark (✓) in the `AUTOSCALE` option.
- Plot the graph: [ERASE][DRAW].
- To see labels: [EDIT][NXT][LABEL][MENU]
- To recover the menu: [NXT][NXT][PICT]
- To trace the curve: [TRACE][(X,Y)]

Use [▶] and [◀] to move along trajectory. Check that for x = 1.05 , y = 0.231. Also, check that for x = -1.48 , y = 0.134.

- To recover the menu, and return to the PLOT environment, press [NXT][CANCL].

Some useful PLOT operations for function plots

In order to discuss these PLOT options, we'll modify the function to force it to have some real roots (Since the current curve is totally contained above the x axis, it has no real roots.) First, highlight the field in front of `EQ:` in the PLOT environment. Then, press [EDIT]. Use the following keystrokes:

[▼][-][.][1][ENTER].

The function to be plotted is now,

Before plotting, place a check mark (✓) in the `AUTOSCALE` option. To plot the graph, press

[ERASE][DRAW].

- Once the graph is plotted, press [FCN] to access the *function* menu. With this menu you can obtain additional information about the plot such as intersects with the x-axis, roots, slopes of the tangent line, area under the curve, etc.

- For example, to find the root on the left side of the curve, move the cursor near that point, and press [ROOT]. You will get the result: ROOT: -1.6635.... Press [NXT] to recover the menu.

- If you move the cursor towards the right-hand side of the curve and press [ROOT], the result now is ROOT: 1.6635... The calculator indicated, before showing the root, that the root was found through *SIGN REVERSAL*. Press [NXT] to recover the menu.

- Pressing [ISECT] will give you the intersection of the curve with the x-axis, which is essentially the root. Press [ISECT]. You will get the same message as before, namely *SIGN REVERSAL*, before getting the result I-SECT: (1.6635..., 0.0000). The [ISECT] function is intended to determine the intersection of any two curves closest to the location of the cursor. In this case, where only one curve, namely, f(x) as defined above, is involved, the intersection sought is that of f(x) with the x-axis. Press [NXT] to recover the menu.

- Place the cursor on the curve at any point and press [SLOPE]. For example, if you place the cursor at any point and press [SLOPE], you will get the value of the slope at that point. For example, at the negative root, SLOPE: 0.16670.... Press [NXT] to recover the menu.

- To determine the highest point in the curve, place the cursor near the vertex and press [EXTR]. The result is EXTRM: (0,0.2989...). Press [NXT] to recover the menu.

- Other buttons available in the first menu are [AREA] to calculate the area under the curve, and [SHADE] to shade an area under the curve. Press [NXT] to see more options. The second menu includes one button called [VIEW] that flashes for a few seconds the equation plotted. Press [VIEW]. Alternatively, you can press the button [NXEQ] to see the expression for the function f(x). Press [NXT] to recover the menu.

- The button [F(X)] gives the value of f(x) corresponding to the cursor position. Place the cursor anywhere in the curve and press [F(X)]. The value will be shown in the lower left corner of the display. Press [NXT] to recover the menu.

- Place the cursor in any given point of the trajectory and press [TANL] to obtain the equation of the tangent line to the curve at that point. The equation will be displayed on the lower left corner of the display. Press [NXT] to recover the menu.

- If you press [F'] the calculator will plot the derivative function, f'(x) = df/dx, as well as the original function, f(x). Notice that the two curves intercept at two points. Move the cursor near the left intercept point and press [FCN][ISECT], to get I-SECT: (-0.6834...,0.21585). Press [NXT] to recover the menu.

- To leave the FCN environment, press [PICT] (or [NXT][PICT]).

- Press [CANCL] to return to the PLOT environment.

Please notice that the field in front of EQ: in the PLOT environment now contains a list. If you press [EDIT] you will notice that there are two equations in the list:

```
{ '-(2*x/2*EXP(-(x^2/2))/√(2*π))' 'EXP(-(x^2/2))/√(2*π)-.1'}
```

When a function is chosen in the EQ: field the calculator creates a variable called EQ (for Equation) containing that function. Originally, EQ contained only f(x). After we pressed the button [F'] in the [FCN] environment, the calculator automatically added f'(x) to the list of equations in EQ.

- Press [ENTER] to return to the PLOT environment.

- Press [ENTER] to leave the PLOT environment.

Notice that the result of the FUNC operations in this exercise is now listed in the stack.

Saving a graph for future use

If you want to save your graph to a variable, get into the PICTURE environment by pressing

[↰][PICTURE].

Then, press

[EDIT][NXT][NXT][PICT→].

This captures the current picture into a graphics object. To return to the stack, press

[PICT][CANCL].

In level 1 of the stack you will see a graphics object described as `Graphic 131 × 64`. To store it into a variable, say FIG, type

['][α][α][F][I][G][α][STO].

Your figure is now stored in variable FIG.
To display your figure again, recall it to level 1 of the stack, by pressing [FIG]. Level 1 now reads `Graphic 131 × 64`. Enter the PICTURE environment, press

[↰][PICTURE].

Clear the current picture, [EDIT][NXT][ERASE].

Move the cursor to the upper left corner of the display, by using the [◄] and [▲] keys.

To display the figure currently in level 1 of the stack press [NXT][REPL].

To return to normal calculator function, press [PICT][CANCL].

Plotting conic curves

Some of the problems in Chapter 3 of the Numerical Methods textbook include simultaneous solution of non-linear equations known as conic equations. The name follows because they describe circles, ellipses, parabolas or hyperbolas, which are known in general as conic curves. (They result from the intersection of a plane with a cone. For example, a circle is the intersection of a cone with a plane perpendicular to the cone's main axis). The most general form of a conic curve in the x-y plane is:

$$Ax^2+By^2+Cxy+Dx+Ey+F = 0$$

We also recognize as conic equations those given in the canonical form for the following figures:

- circle: $(x-x_o)^2+(y-y_o)^2 = r^2$
- ellipse: $(x-x_o)^2/a^2 + (y-y_o)^2/b^2 = 1$
- parabola: $(y-b)^2 = K(x-a)$ or $(x-a)^2 = K(y-b)$
- hyperbola: $(x-x_o)^2/a^2 + (y-y_o)^2/b^2 = 1$ or $xy = K$,

where x_o, y_o, a, b, and K are constant.

The HP48G/GX calculator has the ability of plotting one or more conic curves by selecting Conic as the function TYPE in the PLOT environment. To illustrate this feature I have created the subdirectory TWOC (TWO Conic curves) and stored the equations corresponding to problems 3.25 and 3.27, as lists, under the variables P3.25 and P3.27.

To see the contents of those variables press the appropriate button. For example, pressing [p3.25] produces the list:

`{'(X-1)^2+(Y-2)^2=3' 'X^2/4+Y^2/3=1'},`

which we recognize as the equations of a circle centered at (1,2) with radius √3, and of an ellipse centered at (0,0) with semi-axis lengths a = 2 and b = √3.

NOTE; When writing the equation of a conic for plotting in the HP48G/GX calculator it is important to use upper-case X,Y as variables because Y is the default dependent variable for conics. The calculator does not offer an option to re-define that dependent variable, therefore, if we were to write y, or any other variable, instead of Y, you will get an error message and the plot will not be completed.

To plot these two conics, first store the list in variable P3.25 into the EQ variable by using:

[P3.25][↰][EQ]

- Then, Enter the PLOT environment, [↱][PLOT], and select `Conic` as the `TYPE`.
- Type X (uppercase) as the independent variable (`INDEP`) with range(H-VIEW) -3 to 3.
- Use a range of -1.5 to 2 for V-VIEW.
- Place a check mark (✓) in the `AUTOSCALE` option.
- Plot the graph (be patient here): [ERASE][DRAW].

NOTE: I selected the H-VIEW and V-VIEW ranges to show the intersection of the two curves. There is no general rule to select those ranges, except based on what we know about the curves. For example, for problem 3.25, we know that the circle will extend from -3+1 = -2 to 3+1 = 4 in x, and from -3+2=-1 to 3+2=5 in y. Also, the ellipse, which is centered at the origin (0,0), will extend from -2 to 2 in x, and from -√3 to √3 in y.
Also, notice that for the circle and the ellipse the region corresponding to the left and right extremes of the curves are not plotted. This is the case with all circles or ellipses plotted using Conic as the TYPE.

- To see labels: [EDIT][NXT][LABEL][MENU]
- To recover the menu: [NXT][NXT][PICT]
- To estimate the coordinates of the point of intersection, press the [(X,Y)] button and move the cursor as close as possible to those points using the arrow keys. The coordinates of the cursor are shown in the display. For example, the left point of intersection is close to (-0.705, 1.61), while the right intersection is near (1.92,0.5). These values can be used to obtain a first approximation to the solution when using a numerical method to solve the non-linear system.
- To recover the menu, and return to the PLOT environment, press [NXT][CANCL].

SOLUTION OF NON-LINEAR EQUATIONS

Non-linear equations solved using the PLOT environment

Example 1. Graphic solution of an equation of the form f(x) = 0.
Store the expression `'x^3-2*x^2-2*x+1'` into the variable [EQ]. To plot the function use the procedure outlined above for plotting expressions of the type y = f(x). For this particular case, use the following keystroke sequence to see the plot: [↱][PLOT][ERASE][DRAW] Modify the plot range if needed. Press [EDIT][NXT][LABEL] to see the labels. Press [NXT][NXT][PICT][FCN] and use the [ROOT] key to find the three roots of the curve. You should obtain as solution the values: -1, 0.3819, and 2.6180. To get out of the plot environment use: [NXT][NXT][PICT][CANCL][ENTER].

Example 2. Graphic solution using an equation of the form x = g(x).
(Note: the numerical solution of an equation of the form x = g(x), by simply iterating the equation is known as *fixed-point iteration.* It is not commonly used because it may, at times, produce unstable solutions, or simply converge to a preferred value. It is included here only as an illustration.)

We will use a graphic approach to solve the equation,

$$f(x) = Exp(x) - (3x + 2) = 0,$$

by writing it as:

$$x = Exp(x) - (2x + 2),$$

$$x = (Exp(x) - 2)/3,$$

and,

$$x = ln(3x + 2)$$

Create the variables EQ1, EQ2, and EQ3 containing the following lists of functions of x:

`EQ1 = {x 'EXP(x)-(3*x+2)'},`

`EQ2 = { x '(EXP(x)-2)/3'},`

and

`EQ3 = {x 'LN(3*x+2)'}.`

They are used to show the intersection of the curves y = x and y = g(x), which is the solution to each of the corresponding problems.

First store the contents of EQ1 into EQ. To plot the corresponding figure use: [↱][PLOT][ERASE][DRAW]. The resulting figure shows two points of intersection. To find what those points are, use [FCN], then move the cursor close to either one of them, say the left one, and press [ISECT].

The result is I-SECT: (-0.4552, -0.4552). The second point of intersection is located at (2.1253,2.1253). In other words, the solutions for the expression $x = exp(x) -(3x+2)$ are x=-0.4552 and x = 2.1253. Press [NXT][NXT][PICT][CANCL][ENTER] to return to the normal display. The coordinates of the intersection points will be shown in the stack.

To solve the cases corresponding to EQ2 and EQ3 using the PLOT environment, first copy the contents of EQ2 or EQ3 into EQ, then follow a procedure similar to that shown above. You may need to change the plot ranges (horizontal and vertical), but the general procedure is the same as above.

Using the SOLVE environment to solve for non-linear equations

The HP48G/GX calculator's SOLVE environment can be used to solve for different type of non-linear equations. For example, when solving for eigenvalues of a matrix, we demonstrated how to solve for quadratic and polynomial equations. For a general, non-linear equation of the form $f(x_1,x_2, \ldots,x_n) = 0$, you can use the SOLVE environment, selecting the option *Solve equation....*
For example, let's solve the equation: $e^x\text{-}sin(\pi x/3) = 0$. Simply enter the expression as an HP48G/GX algebraic, i.e.,

['][⬑][e^x] [α][⬑][X] [▶] [-] [SIN] [⬑][π] [×] [α][⬑][X] [÷] [3] [▶] [⬑][=][0][ENTER]]

and store it into variable EQ, i.e.,

['] [α][α][E][Q][α][STO]

Then, enter the SOLVE environment and select *Solve equation...*, by using:

[↱][SOLVE][OK]

The equation is already loaded, all you need to do is highlight the field in front of X: by using [▼], and press [SOLVE]. The solution shown is X: 4.5006...
To obtain a negative solution, try entering a negative number in the X: field, for example, [3][+/-][OK] [▼] [SOLVE]. The solution is now X: -3.045.
For additional examples see your calculator's manual.

Solution of systems of non-linear equations

Systems of non-linear equations will be solved using the library SOLVESYS, developed by Sune Bredahl, a computer scientist from Denmark, (e-mail: c947086@student.dtu.dk), and available in the Internet at the following URL: www.gbar.dtu.dk/~c947086/hp48.html

Installing the SOLVESYS library

1. You should have a variable called SOL in your HOME directory. This variable contains the library SOLVESYS. To install SOLVESYS use these keystrokes:

Keystrokes	Display shows
[↱][SOL]	Library 1550: SOL.... (*)
[0][STO]	(blank display)

(*) 1550 is an ID number for your library selected by the library's author.

2. To check if installation was successful, enter: [↰][LIBRARY][PORTS][:0:]. There should be a variable labeled [1550] in port 0.
3. The next step is to turn the calculator OFF and ON. The normal display will take a few seconds to show. The calculator at that point is installing and attaching the library to the HOME directory.

Using SOLVESYS

Let's use SOLVESYS to solve the following system of non-linear equations given as a list:

```
{'(X-1)^2+(Y-2)^2=3' 'X^2/4+Y^2/3=1'}.
```

Now, store this list into variable [EQ] by using:

[↰][EQ]

Next, start the SOLVESYS library by using:

[↱][LIBRARY][SOLVE][SOLVE]

The system is already loaded, as shown in the display.

- Press [INIT] to initialize values. The next screen (START VALUES) is used to enter initial values of X and Y, with default values of 1.0. Let's use those values to solve the system.

- Press [SOLVE]. The results are shown in an unnamed screen (let's call it the SOLVE screen) that shows how the values of X and Y are changing. It also shows a value called ε, which represents the error in the iteration. When the solution converges, you will get a message box indicating the convergence criteria (! Zero, in this case). This solution converged to X = 1.9063, Y = 0.5239, with an error ε = 1.2785E-6.

- Press [OK] to leave the SOLVE screen, and return to the START VALUES screen. The values of X and Y shown are the solution.

- Press [ON] to quit the SOLVESYS library. The most current solution will be shown in the stack.

To get a different solution you may want to start by entering different values in the START VALUES screen. For example, try using X = -1, Y = -1.

- Start the SOLVESYS library by using: [↱][LIBRARY][SOLVE][SOLVE]

- Press [INIT] to initialize values. Enter [1][+/-][OK] [1][+/-][OK].

- Press [SOLVE]. This solution converged to X = -0.6910, Y = 1.6253, with an error ε = 5.5148E-8.

- Press [OK] to leave the SOLVE screen.

- Press [ON] to quit the SOLVESYS library.

Press [VAR] to recover your variable menu.

Another example: Manning's equation for circular cross-section.

Manning's equation is used to solve for uniform flow (constant-depth flow) in an open channel. The equation is written as,

$$Q = (C/n)\,(A^{5/3}/P^{2/3})\,S^{1/2},$$

where Q = flow rate or volumetric discharge [m^3/s, ft^3/s = cfs], A = cross-sectional area [m^2, ft^2], P = wetted perimeter [m, ft], S = channel bed slope [m/m, ft/ft, i.e., dimensionless], C = units coefficient [dimensionless, C = 1.0 for the S.I. system, C = 1.489 for the English system], n = Manning's resistance coefficient [dimensionless, a function of the channel surface roughness].

For a circular cross-section, if Y = flow depth [m, ft], D = diameter of the circular cross-section [m, ft], and ß = central half-angle [radians], then

$$Y = (D/2)(1-\cos \beta),$$

$$A = (D^2/4)(\beta - \cos\beta \sin\beta),$$

and,

$$P = \beta D.$$

Replacing the expressions for A and P into Manning's equation and raising both sides to the third power results in:

$$((D^2/4)(\beta - \cos\beta\sin\beta))^5/(\beta D)^2 = (Q\cdot n/C\cdot\sqrt{S})^3.$$

To solve for values of Y and ß, given Q = 1 cfs, C = 1.489, n = 0.012, D = 3 ft, and S = 0.00001, enter the following equations:

```
'Y=(D/2)*(1-cos(ß))'
```

and

```
'(D^2/4*(ß-cos(ß)*sin(ß))^5/(ß*D)^2=(Q*n/C*√S)^3'
```

in levels 1 and 2 of the stack. Then enter:

[2][PRG][TYPE][→LIST]

to create the list:

```
{ 'Y=(D/2)*(1-cos(ß))'  '(D^2/4*(ß-cos(ß)*sin(ß))^5/(ß*D)^2=(Q*n/C*√S)^3'}.
```

Store the list in variable EQ:

['][α][E][α][Q] [STO].

To use SOLVESYS, use the following keystrokes: [↱][LIBRARY][SOLVE][SOLVE][INIT].

Enter the values of the constants in the problem within square brackets:

[↰][[ ]][1][.][4][8][9][OK]   [↰][[ ]][3][OK]   [↰][[ ]][1][OK]   [↰][[ ]][0][.][0][0][0][0][1][OK]

Leave the value of 1 for Y and skip to the N: field to enter

[↰][[ ]][0][.][0][1][2][OK].

Leaving the value of 1 for ß, also, press [SOLVE]. The first screen shows that there are two equations, with two unknowns and 5 constants. Next, you get a screen showing the values of the variables as they change, as well as the current value of the error, ε. The iterations stop when $\varepsilon = 1.9200\times10^{-9}$. Press [OK] to see the solution as:

$$Y = 1.3869,\ \beta = 1.4953.$$

Press [QUIT] to end the library operation. The solution will be listed in the stack. Also, the library would have created variables corresponding to the constant values used in the solution, namely, N, S, D, Q, and C. These variables are always created in the HOME directory, regardless of which directory was used to launch the library. Press [↰] [HOME] [VAR] to see these variables. To delete the variables, use:

[↰][{}][N][S][D][Q][C][ENTER].

The following list will be shown in stack level 1:

{ n S D Q C}

To purge the variables enter: [↰][PURGE].

Polynomial operations using the HP48G/GX's own features

Using the *Solve poly...* option in the HP48G/GX *SOLVE* environment you can: (1) find the solutions to a polynomial equation; (2) obtain the coefficients of the polynomial having a number of given roots; and, (3) obtain an algebraic expression for the polynomial as a function of X.

Finding the solutions to a polynomial equation

(This subject was covered earlier in section VII, however we repeat it here for the sake of completeness). A polynomial equation is an equation of the form:

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0.$$

The fundamental theorem of algebra indicates that there are *n* solutions to any polynomial equation of order *n*. Some of the solutions could be complex numbers, nevertheless. As an example, solve the equation:

$$3s^4 + 2s^3 - s + 1 = 0.$$

We want to place the coefficients of the equation in a vector [a_n a_{n-1}... a_1 a_0]. For this example, let's use the vector [3 2 0 -1 1]. To solve for this polynomial equation using the HP48G or GX, try the following:

[↱][SOLVE][▼][▼][OK]	Select `Solve poly...`
[↰][[ ]][3][SPC][2][SPC][0]][SPC][1][+/-][SPC][1][OK]	Enter the vector of coefficients
[SOLVE]	Solve for s
[ENTER]	Returns to stack.
[PRG][TYPE][OBJ→][⇦][OBJ→] [⇦]	All solutions will be listed in the stack.

All the solutions are complex numbers: (0.432,-0.389), (0.432,0.389), (-0.766, 0.632), (-0.766, -0.632). Complex numbers in the HP48G or GX are represented as ordered pairs, with the first number in the pair being the real part, and the second number, the imaginary part. For example, the number (0.432,-0.389), a complex number, will be written normally as 0.432 - 0.389*i*, where *i* is the imaginary unit, i.e., $i^2 = -1$.

Note: The fundamental theorem of algebra indicates that there are n solutions for any polynomial equation of order n. There is another theorem of algebra that indicates that if one of the solutions to a polynomial equation with real coefficients is a complex number, then the conjugate of that number is also a solution. In other words, complex solutions to a polynomial equation with real coefficients come in pairs. That means that polynomial equations with real coefficients of odd order will have at least one real solution.

Generating polynomial coefficients given the polynomial's roots

Suppose you want to generate the polynomial whose roots are the numbers [1 5 -2 4]. To use the HP48G/GX calculator for this purpose follow the following steps:

[↱][SOLVE][▼][▼][OK]	Select `Solve poly...`
[▼][↰][[ ]][1][SPC][5][SPC][2][+/-][SPC][4][OK]	Enter the vector of roots
[SOLVE]	Solve for the coefficients
[ENTER]	Returns to stack. Coefficients are listed in stack.

Note: according to the fundamental theorem of algebra, if you want to get a polynomial with real coefficients, but having complex roots, you must include the complex roots in pairs of conjugate numbers. To illustrate the point, generate a polynomial having the roots [1 (1,2) (1,-2)]. Verify that the resulting polynomial has only real coefficients. Also, try generating a polynomial with roots [1 (1,2) (-1,2)], and verify that the resulting polynomial will have complex coefficients.

Generating an algebraic expression for the polynomial

You can use the HP48G/GX calculator to generate an algebraic expression for a polynomial given the coefficients or the roots of the polynomial. The resulting expression will be given in terms of (upper case) X, however, you can replace that variable by any other variable you like, as shown below.
To generate the algebraic expression using the coefficients, try the following example. Assume that the polynomial coefficients are [1 5 -2 4]. Use the following keystrokes:

[↱][SOLVE][▼][▼][OK]	Select `Solve poly...`
[↰][[ ]][1][SPC][5][SPC][2][+/-][SPC][4][OK]	Enter the vector of roots
[▲][SYMB]	Highlight coefficients; generate symbolic expression
[ENTER]	Returns to stack. Expression shown in stack.

The resulting expression is given as `'X^3+5*X^2-2*X+4'`.

To generate the algebraic expression using the roots, try the following example. Assume that the polynomial roots are [1 3 -2 1]. Use the following keystrokes:

[↱][SOLVE][▼][▼][OK]	Select `Solve poly...`
[▼][↰][[ ]][1][SPC][3][SPC][2][+/-][SPC][1][OK]	Enter the vector of roots
[▼] [SYMB]	Highlight roots, generate algebraic expression.
[ENTER]	Returns to stack. Expression listed in stack.

The resulting expression is listed as `'(X-1)*(X-3)*(X+2)*(X-1)'`. To expand the products, you can use:

[↰][SYMBOLIC]	Access symbolic manipulation menu
[EXPA] (press it 9 - 10 times, until all products are expanded)	
[COLCT]	Collects powers of X

The resulting expression is: `'-6+X^4_3*X^3-3*X^2+11*X'`.

To cut down in the amount of time required to generate the final version of the polynomial starting with the roots, it is better to generate the coefficients first, then generate the algebraic expression while highlighting the coefficients. For example, for this case try:

[↱][SOLVE][▼][▼][OK] Select `Solve poly...`
[▼][↰][[ ]][1][SPC][3][SPC][2][+/-][SPC][1][OK] Enter the vector of roots
[SOLVE] Solve for the coefficients
[SYMB] Highlight roots, generate algebraic expression.
[ENTER] Returns to stack. Expression listed in stack.

The resulting expression is: `'X^4_3*X^3-3*X^2+11*X-6'`. The coefficients are listed in stack level 2.

Evaluating a polynomial (or any) expression

Suppose that you want to evaluate the expression obtained above for X = 5. Keeping the expression in stack level 1, enter the following:

[↰][{ }][α][X][SPC][5][ENTER] Enter the list {X 5}
[↰][SYMBOLIC] Access symbolic manipulation menu
[NXT][|] Use the "evaluate at" expression

The result is 224. This procedure can be used to evaluate any HP48G/GX algebraic in level 2 using the values in the list in level 1.

Replacing the variable in the polynomial's algebraic expression

Press [↱][UNDO] to recover the last operands. If instead of using {X 5} we use {X x}, the variable X will be replaced by x in the polynomial. Try the following:

[⇦][↰][{ }] [α][X] [SPC] [⇦][α][↰] [X] [ENTER] Enter the list {X x}
[↰][SYMBOLIC] [NXT][|] Use the "evaluate at" expression

The result is the polynomial: : `x^4_3*x^3-3*x^2+11*x-6'`.

The Method of Least Squares for Straight-line Approximation

The problem can be stated as follows: suppose that we have *n* paired observations (x_i, Y_i); we predict Y by means of

$$y = b + m\,x,$$

where b and m are constant.
Define the prediction error as,

$$e_i = Y_i - y_i = Y_i - (b + m\,x_i).$$

The method of least squares requires us to choose a, b so as to minimize the sum of squared errors (SSE)

$$S(b,m) = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} [Y_i - (b + mx_i)]^2$$

From the conditions

$$\frac{\partial S}{\partial b} = 0 \qquad \frac{\partial S}{\partial m} = 0$$

We get the, so-called, *normal equations*:

$$b \cdot N + m \cdot \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} Y_i$$

$$b \cdot \sum_{i=1}^{n} x_i + m \cdot \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i \cdot Y_i$$

This is a system of linear equations with b and m as the unknowns.

<u>HP48G application</u>:

Use the sub-directory CFIT (Curve FITting) within directory PF&I. Store the following data into a matrix. Store the matrix into a variable called EXP1. (Column 1 represents x, and column 2, y.)

x	20	60	100	140	180	220	260	300	340	380
Y	0.18	0.37	0.35	0.78	0.56	0.75	1.18	1.36	1.17	1.65

Then copy it into ΣDAT:

[EXP1][↰][STAT][DATA][↰][ΣDATA]

First, plot the data to see if it follows a linear trend:

[↰][STAT][PLOT][SCATR] (shows good linear trend) [CANCL].

Get summary statistics:

[↰][STAT][SUMS][ΣX][ΣY][ΣX^2][ΣY^2][ΣX*Y][NΣ].

Results in: $\Sigma x = 2000$; $\Sigma y = 8.35$; $\Sigma x^2 = 532000$; $\Sigma\, y^2 = 9.1097$; $\Sigma xy = 2175.4$, $n = 10$.

The normal equations are now:

$$10\,b + 2000\,m = 8.35$$
$$2000\,b + 532000\,m = 2175.40$$

or, in matrix notation,

Use the following commands to solve for the vector $[b\ m]^T$:

[↱][SOLVE][▲][▲][OK]
[↱][MATRIX]
[1][0][SPC][2][0][0][0][ENTER][▼]
[2][0][0][0][SPC][5][3][2][0][0][0][ENTER][ENTER]
[▼][↰][[ ]][8][.][3][5][SPC][2][1][7][5][.][4][OK]
[SOLVE]

Result is X: [6.92424242424E-2...

[EDIT]

Bottom display shows 6.92424E-2.

[▶]

Bottom display shows 3.8287E-3.

[ENTER][ON]

The results obtained are then b = 0.0924 and m = 0.00383.

Using the "Fit data" feature in the HP48G/GX

The HP48G calculator has its own feature for determining the least-square linear fitting to a set of data points (x,y). It uses the ΣDAT matrix where x and y are stored in columns. To access this feature, use the following keystrokes:

[↱][STAT][▼][▼][OK]

The display shows the current ΣDAT, already loaded. Change your set up screen to the following parameters, if needed:

```
X-COL: 1   Y-COL: 2
 MODEL: Linear Fit
```

Then, press [OK], to get the following results:

```
1: '6.924242…E-2 + 3.828787…E-3*X'
2: Correlation: 0.9514813
3: Covariance: 56.1555
```

We can write the following equation: $y = 0.06924 + 0.00383\,x$.

Covariance and Correlation

For a sample of data points (x,Y), we define covariance as

The *sample correlation coefficient* for x,Y is defined as

Where s_x, s_Y are the sample standard deviations of x and y, respectively, i.e.

The values s_{xY} and r_{xY} are the "*Covariance*" and "*Correlation*," respectively, obtained by using the "*Fit data*" feature of the HP48G calculator.

Linearized relationships

Many curvilinear relationships "straighten out" to a linear form. For example, the different models for data fitting provided by the HP48G calculator can be linearized as described below:

Type of Fitting	Actual Model	Linearized Model	Independent variable ξ	Dependent Variable η	Covariance $s_{\xi\eta}$
Linear	$y = a + bx$	$y = a + bx$ [same]	x	y	s_{xy}
Logarithmic	$y = a + b \ln(x)$	$y = a + b \ln(x)$ [same]	$\ln(x)$	y	$s_{\ln(x),y}$
Exponential	$y = a\, e^{bx}$	$\ln(y) = \ln(a) + bx$	x	$\ln(y)$	$s_{x,\ln(y)}$
Power	$y = a\, x^b$	$\ln(y) = \ln(a) + b \ln(x)$	$\ln(x)$	$\ln(y)$	$s_{\ln(x),\ln(y)}$

The sample *covariance* of ξ,η is given by
Also, defining the sample *variances* of ξ and η, respectively, as

The sample *correlation coefficient* $r_{\xi\eta}$ is

The general form of the *regression equation* is $\eta = b + m\xi$.

Another equation that can be re-cast in the linear form ($\eta = b + m\xi$) is the *reciprocal (linear) function*

$$y = 1/(a + ax),$$

which can be re-written as

$$1/y = a + ax.$$

In general, equations of the form

$$y = a + a\, f(x), \text{ or } g(y) = a + ax,$$

can be analyzed using a linear relationship, by taking either

$$\xi = f(x), \text{ or } \eta = g(y),$$

in the linearized form

$$\eta = b + m\xi.$$

For example, for centrifugal pumps, the relationship between the discharge, Q, and the energy head H, is usually a quadratic equation of the form

$$H = b + mQ^2,$$

where b and m are constant. Therefore, if we take

$$\xi = Q^2 \text{ and } \eta = H,$$

the relationship is re-cast as

$$\eta = b + m\xi.$$

To illustrate this particular case we solve the following exercise. The data shown below comes from tests performed in a centrifugal pump (NOTE: First enter the program CRMAT, described in the following page):

Q(gpm)	H(ft)
0	100
110	90
180	80
250	60
300	40
340	20

In order to determine the relationship H-Q you need to perform a linear fitting between $\xi = Q^2$ and $\eta = H$. Here is a suggested procedure using the HP48G calculator:

1) Enter the values of Q as a list and place two copies of the list in the display:

[↰][{}][0][SPC][1][1][0][SPC][1][8][0][SPC][2][5][0][SPC][3][0][0][SPC][3][4][0][ENTER][ENTER]

2) Squared the list in level 1 (i.e., generate $\xi = Q^2$): [↰][x^2]
3) Enter the values of H as a list (i.e., enter $\eta = H$):

[{}][1][0][0][SPC][9][0][SPC][8][0][SPC][6][0][SPC][4][0][SPC][2][0][ENTER]

4) Create a matrix of three columns by pressing: [3][CRMAT]
5) Save the matrix into a variable, say PUMP: ['][α][α][P][U][M][P][ENTER][STO]
6) Activate the data fit feature of the HP48G calculator: [↱][STAT][▼][▼][OK].
7) Load PUMP into ΣDAT, by selecting the field in front of `ΣDAT:` in the set-up screen, and using [CHOOS].
8) Also, change the following parameters:
`X-COL: 2    Y-COL: 3    MODEL: Linear Fit`

9) Then, press [OK], to get the following results:

```
3: '100.357358269+-6.78644112646E-4*X'
2: Correlation: -.998337705738
1: Covariance: -1395200
```

Line 3 in the display gives the fitted equation, namely, $H = 100.36 - 6.79\text{x}10^{-6}\ Q^2$, with H(ft), Q(gpm). The value of the correlation coefficient is very close to -1, indicating a good fitting of the data.

NOTE: The program [*CRMT*] allows you to put together a $p\times n$ matrix (i.e., p rows, n columns) out of n lists of p elements each. This program can be very useful in producing matrices out of lists of numbers, as is the case in data fitting problems. To use this program, enter the n lists in the order that you want them as columns of the matrix, enter the value of n, and press [*CRMT*]. To create the program enter the following keystrokes:

Keystroke sequence:	Produces:
[↰] [<< >>] [ENTER]	<<
[↰] [ENTER]	DUP

['] [α] [↰] [N] [▶]	'n'
[STO] [1]	STO 1
[↰] [SWAP]	SWAP
[PRG] [*BRCH*] [*FOR*] [*FOR*]	FOR
[α] [↰] [J]	j
[PRG] [*TYPE*] [*OBJ->*]	OBJ→
[*->ARR*]	→ARRY
[PRG] [*BRCH*] [*IF*] [*IF*]	IF
[α] [↰] [J] [SPC]	j
[α] [↰] [N]	n
[PRG] [*TEST*] [<]	<
[PRG] [*BRCH*] [*IF*] [*THEN*]	THEN
[α] [↰] [J] [SPC] [1] [+]	j 1 +
[↰] [STACK] [*ROLL*]	ROLL
[PRG] [*BRCH*] [*IF*] [*END*]	END
[PRG] [*BRCH*] [*IF*] [*NEXT*]	NEXT
[PRG] [*BRCH*] [*IF*] [*IF*]	IF
[α] [↰] [N] [SPC] [1]	n 1
[PRG] [*TEST*] [>]	>
[PRG] [*BRCH*] [*IF*] [*THEN*]	THEN
[1] [SPC]	1
[α] [↰] [N] [SPC] [1] [-]	n 1 -
[PRG] [*BRCH*] [*FOR*] [*FOR*]	FOR
[α] [↰] [J] [SPC]	j
[α] [↰] [J] [SPC] [1] [+]	j 1 +
[↰] [STACK] [*ROLL*]	ROLL
[PRG] [*BRCH*] [*IF*] [*NEXT*]	NEXT
[PRG] [*BRCH*] [*IF*] [*END*]	END
[α] [↰] [N] [SPC]	n
['] [α] [↰] [N] [▶]	'n'
[↰] [PURGE]	PURGE
[MTH] [*MATR*] [*COL*] [*COL→*]	COL→
[ENTER]	Program is displayed in level 1

To save the program:
['] [α] [α] [C] [R] [M] [T] [α] [STO]

NOTE: for additional statistical data analysis features of the HP 48 G/GX programmable calculator see Appendix A.

Symbolic and Numerical Derivatives using the HP48G/GX calculator

---

The following examples show how to use the HP48G/GX calculator's own features for obtaining and evaluating simple derivatives. The examples are taken from my Dynamics class (ENG202) of Spring Quarter 1998. To try these exercises I suggest you create a sub-directory named DIFFE, and work the examples within that sub-directory.

Differentiation

First, we will show some problems that involve differentiation. For example, given the position of a particle, $x(t) = 2.5e^{-t} + \sin(0.5t)$, determine expressions for the velocity, $v = dx/dt$, and for the acceleration, $a = dv/dt = d^2x/dt^2$. Type in the expression for x(t), and store it in a variable xP:

['][2][.][5][×][↰][e^x][α][↰][-][t][▶][+][SIN][.][5][×][α][↰][T][ENTER]
['][α][↰][x][α][P][STO]

Enter the symbolic operation environment and select differentiation, as follows:

[↱][SYMBOLIC][▼][OK]

Select expression in xP to be differentiated, [CHOOS](highlight xP)[OK]
Enter t as the differentiation variable: [▼][α][↰][T][OK]
Select symbolic result: [CHOOS](highlight `Symbolic`)[OK]
Differentiate: [OK]

The result is:

`'2.5*-EXP(-t)+COS(0.5*t)*0.5'`

To simplify the result, try the following sequence: [↰][SYMBOLIC][COLCT]

The result is now `'-(2.5*EXP(-t)) + 0.5*COS(0.5*t)'`, or, $v = dx/dt = -2.5\ e^{-t} + 0.5\ cos(0.5\ t)$.

Store it into a variable called vP, ['][α][↰][v][α][P][STO].

To determine the next derivative, i.e., a = dv/dt, we first store the current result in the variable EXPR, by using: [VAR][VP][↰][EXPR]. Then, we use a procedure similar to the previous derivative, namely,

[↱][SYMBOLIC][▼][OK] [▼][α][↰][T][OK] [CHOOS](highlight `Symbolic`)[OK] [OK]

The result is: `'-(2.5*EXP(-t))+0.5*-(SIN(0.5*t)*0.5)'`

Press [↰][SYMBOLIC][COLCT] to get: `'2.5*EXP(-t)-0.25*SIN(0.5*t)'`, i.e., $a = dv/dt = 2.5\ e^{-t} + 0.25\ sin(0.5\ t)$.
You may want to store it into a variable called aP, ['][α][↰][a][α][P][STO].

Note: An alternative way to obtain a symbolic derivative is to just type the expression in the stack, then type the differentiation variable, and use the sequence [↱][∂]. For example, to get the expression for x(t) in the stack, press [VAR][XP]. Then, press ['][α][↰][T][ENTER]. Finally, press [↱][∂]. The result is the same as before, namely, `'2.5*-EXP(-t)+COS(0.5*t)*0.5'`.

Evaluating derivatives

To determine the derivative of x(t) at a given value of t, say, dx/dt| $_{t=0.5}$, try the following:

[VAR][XP][↰][EXPR][↱][SYMBOLIC][▼][OK][▼][α][↰][T][OK]
[CHOOS](highlight `Numeric`)[OK] [▼][.][5][OK][OK]

The result is -1.032, i.e., dx/dt| $_{t=0.5}$ = -1.032. This represents a *numeric (or numerical) derivative*. To check, replace t = 0.5, into $v = dx/dt = -2.5\ e^{-t} + 0.5\ cos(0.5\ t)$. The result is the same.

Note: The following is an alternative way to evaluate a numeric derivative:

[.][5]['][α][↰][T][STO][VAR][XP]['] [T][ENTER] [↱][∂].

The result, as above, is -1.032.
You can check that the answer is right by pressing

[VAR][VP][EVAL].

You should get -1.032. (Recall that vP contains the symbolic expression for dx/dt found earlier).
If you want to work with the symbolic result again, you will need to purge the variable t:

['][T][↰][PURGE].

Note: A second alternative way to evaluate the derivative at a specific value of t is to use the *evaluate* function as follows:
First, obtain the derivative using:

[VAR][XP]['] [α][↰][T][ENTER] [↱][∂].

Next, create a list with the name of the independent variable and the value to be replaced (e.g., t = 0.5):

[↰][{}][α][↰][T][SPC][.][5][ENTER].

Finally, use the evaluate at function:

[↰][SYMBOLIC][NXT][|].

The result, as before, is -1.032.

You can use the HP48G or GX calculator to take *derivative of equations* in which there are functions defined in both sides of the equation. For example, we can have the calculator obtain the derivative of the equation: x(t) = 2 r cos θ(t), as follows:

['] [α][↰][X] [↰][()] [α][↰][T] [▶] [↰][=] [2][×] [α][↰][R] [×][COS] [α][↱][F] [↰][()] [α][↰][T]
[ENTER]
['] [α][↰][T] [ENTER]
[↱][∂]
[↰][SYMBOLIC][COLCT]

The result is

`'derx(t,1) = -(2*derθ(t,1)*SIN(θ(t))*r)',` or

$$x'(t) = -2\ \theta'(t) \sin \theta(t).$$

Take a second derivative, as follows:

['] [α][↰][T] [ENTER] [↱][∂] [↰][SYMBOLIC][COLCT]

The result is now:

```
'derderx(t,1,1,0)=-((2*derθ(t,1)^2*COS(θ (t))
    +2*derderθ (t,1,1,0)*SIN(q(t)))*r', or
```

$$x''(t) = -(2\ [\theta'(t)]2 \cos \theta(t) + 2\ \theta''(t) \sin \theta(t))\ r.$$

Implicit differentiation is also possible. Try taking the derivative with respect to t of the equation: $r^2 = 2\ \theta^3$. Use the following keystrokes:

['][↰][()] [α][↰][R] [↰][()] [α][↰][T] [▶] [▶] [y^x] [2]
[↰][=] [2][×] [↰][()] [α][↱][F] [↰][()] [α][↰][T] [▶] [▶] [y^x] [3] [ENTER]
['] [α][↰][T] [ENTER]
[↱][∂]
[↰][SYMBOLIC][COLCT]

The result is:

```
'2*derr(t,1)*r(t) = 6*derθ(t,1)* θ(t)^2',
```

or:

$$2\ r'(t)\ r(t) = 6\ \theta'(t)\ [\theta(t)]^2.$$

Integration with the HP48G/GX calculator

The following examples show how to use the HP48G/GX calculator's own features for obtaining and evaluating simple derivatives and integrals. The examples are taken from the Dynamics class (ENG202) of Spring Quarter 1998. To try this exercises I suggest you create a sub-directory named INTGR and work the examples within that sub-directory.

Integration

Given $a(t) = 1/(1 + t^2)$, with initial conditions v = 5, x = 2, at t = 1, determine expressions for v(t) and x(t). From, a = dv/dt, we can write $dv = dt/(1 + t^2)$, integrating using the boundary conditions we have, From which it follows that:

We will use the HP48G or GX calculator to evaluate the integral in the right-hand side of the equation above, as follows:

- First, enter the expression for the integrand, and store it in a variable called aP (remember to purge t):

['][1][÷][↰][()][1][+][α][↰][T][y^x][2][ENTER] [↰][AP]
['][T][↰][PURGE]

- Next, enter the SYMBOLIC environment for integration, selecting aP as the expression to integrate, and selecting the limits of integration:
[↱][SYMBOLIC][OK][CHOOS](select aP)[OK][▼][α][↰][T][OK][1][OK][α][↰][T]
- Next, select symbolic and perform the symbolic integration:
[CHOOS](select Symbolic)[OK][OK]

The result is a long expression involving derivatives, e.g., ∂t(t), as shown below.

To simplify the expression, press [EVAL]. The simplified result is `'ATAN(t) - 0.785'`. Type [5][+][↰][SYMBOLIC][COLCT], to get the expression for v(t), namely, `'4.215+ATAN(t)'`. We can write, then, *v(t) = 4.215 + atan(t)*. Store the expression in variable vP: [VAR][↰][VP].

To integrate v(t), we start from *dx/dt = v(t),* to write *dx = (4.215 + atan(t))dt*, and, using the initial conditions, we write

To evaluate the right-hand side of the above equation, use this keystroke sequence:

[VP][↰][EXPR] [↱][SYMBOLIC][OK][▼][α][↰][T][OK][1][OK][α][↰][T][OK][EVAL]

To obtain x(t) enter [2][+][↰][SYMBOLIC][COLCT], which results in

'-2.653-0.5*LN(1+t^2)+ATAN(t)*t+4.214',

or,

$$x(t) = t\ \mathrm{atan}(t) - 1/2\ \ln(1 + t^2) + 4.215t - 2.653.$$

You may want to store this last result into variable xP: [↰][XP].

Numerical Integration

If you are only interested in obtaining the value of the integral, say, the right-hand side of

you can use the SYMBOLIC environment and select Numeric for the RESULT field in the INTEGRATE environment. Let's calculate the integral above using the following keystrokes (recall that the integrand is stored in the variable aP):

[VAR][AP][↰][EXPR] [↱][SYMBOLIC][OK][▼][α][↰][T][OK][1][OK][5][OK]
[CHOOS][▼][OK][OK]

The result is 0.588, and from the equation above it follows that v = 5+0.588 = 5.588, when t = 5. To verify the result, enter a value of 5 for t and evaluate the value of the variable vP (which is the expression for v(t)):

[5]['][α][↰][T][STO][VP][EVAL].

We recover the same result for v, i.e., 5.588. Purge the variable t: ['][T][↰][PURGE].
When you perform a numerical integration there is a new variable created named IERR, which contains an estimate of the uncertainty in the numerical result. *(See the **HP48G Series** User's Guide, pages 20-6 and 20-7 for more details*). For the present example, the value of IERR is 0.001. In other words, the numerical result is within ±0.001 of the calculated value of 0.588.

Shortcut for symbolic and numerical integration

We can use the [∫] key to calculate integrals without using the INTEGRATE environment. In order to accomplish this, we need to type in an expression of the form:

' ∫ (*lower limit of integration*, *upper limit of integration*, *integrand*, *integration variable*)'

and, then, press [EVAL]. If needed, press [EVAL] again, and use [↰][SYMBOLIC][COLCT] to simplify the resulting expression.
For example, to integrate the right-hand side of

type the following:

['][↱][∫][1] [↰][,] [α][↰][T] [↰][,] [1][÷][↰][()][1][+][α][↰][T][y^x][2][▶] [↰][,][α][↰][T][ENTER]

To get ' ∫ (*1*, *t*, *1/(1+t^2)*, *t*)'. Press [EVAL] [EVAL]. The result is, as before, 'ATAN(t)-0.785'.

Another possibility is to enter the lower limit in level 4, the upper limit in level 3, the integrand in level 2, and the integration variable in level 1, and then press [↱][∫]. For the present case, since we already have the integrand stored in variable aP, this procedure would be more convenient. Try the following keystroke sequence:

[1][ENTER] [α][↰][T] [ENTER] [AP] [α][↰][T] [ENTER].

Your display will look like this:

```
4:              1.000
3:                't'
2:       '1/(1+t^2)'
1:                't'
```

Now, press [↱][∫] to calculate the integral. Press [EVAL] to simplify the result.

Numerical integration can also be computed by direct use of the [↱][∫] keys by having both limits of integration be numbers. For example, to calculate the right hand side of
use the following: [1][ENTER] [5] [ENTER] [AP] [α][↰][T] [ENTER] [↱][∫]. The result is given symbolically. Press [EVAL], or [↰][→NUM], to obtain the value of the integral. You should get, as before, 0.588.

Try the following exercise:

['][↱][∫][0] [↰][,] [5] [↰][,] [1][÷][α][↰][T] [↰][,][α][↰][T][ENTER] [EVAL] [EVAL]

You will get an error, because the integrand, 1/t, is not defined at t = 0. Therefore, the integral is indefinite.

Let's change the lower limit of integration to 1, by typing:

['][↱][∫][1] [↰][,] [5] [↰][,] [1][÷][α][↰][T] [↰][,][α][↰][T][ENTER] [EVAL] [EVAL]

The result is 1.609. Checking with the exact result we find that

It is possible, using the HP48G or GX, to integrate an equation. We will use the following example to illustrate not only <u>*integration of equations*</u>, but also the use of the EQUATION editor.

Problem: A particle is moving with an acceleration $a = -1.5\ v^{1/2}$, with $v = 4$, when $t = 0$. Determine and expression for the velocity, $v(t)$, and evaluate the velocity at $t = 2$.
From the definition of acceleration, $a = dv/dt$, and using the initial conditions indicated above, we can write the following integral equation:

To use the HP48G or GX calculator to evaluate the integrals in the equation above, use the following keystroke sequence:

[↰][EQUATION][↱][∫][4][▶][α][↰][V][▶][1][÷][√x][α][↰][V] [▶][▶][▶][α][↰][V][↰][=]
[-][1][.][5][↱][∫][0][▶][α][↰][T][▶][1][▶][α][↰][T]

The screen should look as:

Press [ENTER].
The display should show: ' ∫ (4, *v*, *1/√v*, *v*) = -1.5*∫ (0, *t*, *1*, *t*) ' .
To evaluate the integral, press [EVAL][EVAL], which gives the result *'2* v - 4 = -(1.5*t)'*. To isolate the variable v, try the following:

['][α][↰][V][ENTER]	Indicate the variable to isolate
[↰][SYMBOLIC][ISOL]	Isolate the variable v

The result is: *'v = SQ((-(1.5*t)+4)/2)'*, i.e.,

To evaluate the value of v at t = 2, press [ENTER] to copy the expression to level 1 (keep a copy in level 2 for possible future use). Then, define the value of t using a list {t 2}, i.e.,

[↰][{ }] [α][↰][T][SPC][2][ENTER].

Finally, use the evaluate function: [↰][SYMBOLIC][NXT][|], to get *'v = 0.250'*.

Integration with no closed-form solution

If there is no closed-form solution for an integral, numerical integration is the only possibility. For example, if v(t) = exp($-t^2$), with x = 0 at t = 0, determine x for t = 5. We will try to find first a closed-form solution, by evaluating

['][↰][e^x][-][α][↰][T][y^x][2][ENTER][VAR][↰][EXPR]
[↱][SYMBOLIC][OK][▼][α][↰][T][OK][0][OK][α][↰][T][OK]

The result given by the calculator is '∫ (0, t, EXP(-t^2),t) ', which indicates that there is no closed-form solution for the integral. A numerical integration can be obtained as follows:

[↱][SYMBOLIC][OK][▼][α][↰][T][OK][0][OK][5][OK][CHOOSE][▼][OK][OK]

The result is 0.886, i.e., x(t = 5) = 0.886.
Note: numerical integration can be performed even if the integral has a closed-form solution. The procedure is similar to that shown above.

Try also typing:

['][↱][∫][0] [↰][,] [α][↰][T] [↰][,] [↰][e^x][-][α][↰][T] [y^x][2] [↰][,][α][↰][T] [▶] [↰][,]
[α][↰][T][ENTER][EVAL].

After trying to evaluate the integral you will recover the same original expression, namely,

```
'∫(0, t, EXP(-t^2),t) ',
```

therefore, no closed-form expression is available. Change it to a numerical integration by typing:

['][↱][∫][0] [↰][,] [5] [↰][,] [↰][e^x][-][α][↰][T] [y^x][2] [↰][,][α][↰][T] [▶] [↰][,]
[α][↰][T][ENTER][EVAL].

Notice that, by using [EVAL], you still do not get a numerical result. In this case, the appropriate key sequence to use is [↰][→NUM]. The result is 0.866, as before.

Multiple integrals

Multiple integrals may result from calculations of mass and moments of inertia of bodies.
For example, calculate the moment of inertia with respect to the origin of a disk of unit thickness described in polar coordinates by $\{0 < r < a,\ 0 < \theta < 2\pi\}$ whose density is given by $\rho(r,\theta) = \alpha r^3$.
In polar coordinates a differential of area is given by

$$dA = r\ dr\ d\theta,$$

for a unit thickness; a differential of volume would be,

$$dV = (1)dA = r\ dr\ d\theta;$$

the corresponding differential of mass would be

$$dm = \rho\ dV = \rho(r,\theta)\ r\ dr\ d\theta\ = = \alpha r^4\ dr\ d\theta;$$

and, the differential of moment of inertia would be

$$dI_o = r^2\ dm = \alpha r^6\ dr\ d\theta.$$

Therefore, the moment of inertia will be given by:

We will use this problem to illustrate the use of *the EQUATION editor* in the HP48G series calculator. Use the following keystrokes to build the double integral:

[↰][EQUATION] [↱][∫][▶][0][▶][2][×][↰][π][▶][↱][∫][0] [▶] [α][↰][A]
[▶] [α][↱][A] [×] [α][↰] [R] [y^x][6] [▶][▶] [α][↰] [R] [▶][▶][α][↱][F][ENTER]

The integral is shown in level 1 of the display as:

`' ∫(0, 2*π, ∫(0,a,α*r^6,r),θ)'`

To evaluate press: [EVAL] [EVAL] [EVAL] [ ][SYMBOLIC][COLCT]

The result is `'0.286*a^7*π'`.

Try the following example: determine the mass of a body of unit thickness described in Cartesian coordinates by $\{0 < x < 5,\ 0 < y < 1\}$, and whose density is given by $\rho(x,y) = x^2 + y^2$.

In Cartesian coordinates a differential of area is given by

$$dA = dx\ dy,$$

for a unit thickness; a differential of volume would be,

$$dV = (1)dA = dx\ dy;$$

the corresponding differential of mass would be

$$dm = \rho\ dV = \rho(x,y)\ dx\ dy\ = = (x^2 + y^2)\ dx\ dy.$$

Therefore, the mass of the body will be given by:

We will use this problem to illustrate the use of the EQUATION editor in the HP48G series calculator. Use the following keystrokes to build the double integral:

[↰][EQUATION] [↱][∫][0][▶][5][▶][↱][∫][0] [▶] [1][▶]
[↰] [()][α][↰][X] [y^x][2] [▶] [+][α][↰][Y] [y^x][2]
[▶][▶][▶] [α][↰] [Y] [▶][▶][α][↰] [X] [ENTER]

The integral is shown in level 1 of the display as:

```
' ∫(0, 5, ∫(0,1,(x^2+y^2),y),x) '
```

Press [ENTER] to keep an additional copy of the integral expression available for future use. Press [EVAL] three or four times. You will notice that the calculator tries to evaluate the inner integral while showing a complicated expression involving derivatives (∂) and evaluations (|). You would also notice that if you continue pressing the [EVAL] key, the expressions keep alternating, meaning that the calculator has reached an impasse in its evaluation effort. To help out try the keystroke combination:

[↰][SYMBOLIC][COLCT]

The integral simplifies now to '∫ (0,5,0.33333333333+x^2,x)'. Press [EVAL] twice to get the value of the integral as 43.3333333334.

Now, drop that value from level 1 of the stack to get the original integral in level 1. Press [↰][→NUM] to obtain a numerical evaluation of the integral. It takes about 1 minute to obtain the result 43.3333333334.

<u>*Note*</u>: Using [EVAL] and [↰][SYMBOLIC][COLCT] seems to produce results faster than using [↰][→NUM]. Therefore, whenever possible, try to simplify the integration by using [EVAL] and [↰][SYMBOLIC][COLCT], before trying to use [↰][→NUM]. Also, be aware that, if numerical integration is the only way possible, a double integral will take a relatively large amount of time to be evaluated. Therefore, be patient. However, if an integral is taking an inordinate amount of time, say more than 5 minutes, to be evaluated, it is better to quit and try simplifying the integral by hand before attempting a numerical integration. If no simplification is possible, I would recommend using a numerical integration in the computer.

Multiple integrals can also have variable limits, for example, if you were to calculate the integral

You can write the integral directly into the stack by using:

['][↱][∫][0][↰][,][5] [↰][,][↱][∫][0]][↰][,] [1][+] [α][↰][X] [↰][,]
[↰][()][α][↰][X] [y^x][2] [+][α][↰][Y] [y^x][2] [▶][↰][,] [α][↰] [Y]
[▶][↰][,][α][↰] [X] [ENTER]

The integral is shown in level 1 of the display as:

```
' ∫(0, 5, ∫(0,1+x,(x^2+y^2),y),x) '
```

Press [ENTER] to keep an additional copy of the integral. Press [EVAL] three or four times to simplify the expression as much as possible, then, use [↰][SYMBOLIC][COLCT].

The integral simplifies now to '∫ (0,5,0.33333333333*(1+x)^3+x^2*(1+x),x)'. Pressing [EVAL] reduces the integral to ' 149.583333334+∫(0,5,x^2*x,x)'. If you try pressing [EVAL] again the calculator will take no action, indicating that it cannot simplify the integral on its own anymore. However, you can help a little by using [↰][SYMBOLIC][COLCT]. The integral will now read ' 149.583333334+∫(0,5,x^3,x)'. Pressing [EVAL] twice more will get the result of 305.833333334.

Now, drop that value from level 1 of the stack to get the original integral in level 1. Press [↰][→NUM] to obtain a numerical evaluation of the integral. It takes about 1 to 2 minutes to obtain the result 305.833333334.

As mentioned above, some integrals simply take too much time to be evaluated and it would be worth using the computer for their evaluation to save time.

One of them is:

I stopped my calculator after about 5 minutes without any solution.

Note: Although the HP48G/GX programmable calculator packs a lot of power in terms of its ability of solving symbolic and numerical problems, it has limitations due to its size and speed. Integrating complicated expression in the HP48G/GX calculator may result in the use of an inordinate amount of time. To ensure obtaining a result in a reasonable amount of time, limit your integrals to simple expressions. For more information on the HP48G/GX Symbolic Integration Patterns see pages 20-30 and 20-31 of the "***HP 48G Series*** *User's Guide*" that is provided with your calculator. If you can simplify your integrand to one of the patterns listed in those pages then you should have no problem obtaining a symbolic integration. If numerical integration is required, also keep your integrands simple.

Solution of Differential Equations with the HP48G/GX

The subject of differential equations will not be examined in this class, it belongs in the subsequent course (Numerical Methods II). However, to give you a preview of the material to be covered in that class I have prepared some examples of using the HP48G/GX calculator for the solution of simple differential equations. The examples are taken from the Dynamics class (ENG202) from Spring Quarter 1998.

Solving a first-order differential equation

Suppose we want to solve the differential equation,

$$dv/dt = -1.5\ v^{1/2},$$

with v = 4 at t = 0. We are asked to find v for t = 2. You can actually solve for v(t) by simple integration, however, we will show you how to use other methods that can be applied to any first-order differential equation.

First, create a subdirectory called ODE1, and move into that subdirectory. Next, create the expression defining the derivative and store it into variable EQ:

['][1][.][5][+/-][×][√x][α][↰][V][ENTER]
['][α][α][E][Q][α][STO]

Then, enter the SOLVE environment:

[↱][SOLVE][▼][OK][▼][α][↰][T][OK][0][OK][2][OK][α][↰][V][OK][4][OK]

To solve, press: [SOLVE](wait)[EDIT]

The result is 0.2499 ≈ 0.25.

Solution presented as a table of values

Suppose we wanted to produce a table of values of v, for t = 0.00, 0.25, …, 2.00, we will proceed as follows:
First, prepare a table to write down your results:

t	v
0.00	0.00
0.25	
…	…
2.00	

Next, within the SOLVE environment, change the final value of the independent variable to 0.25, use :

[▲][.][2][5][OK] [▶][▶] [SOLVE](wait)[EDIT]	Solves for v at t = 0.25, v = 3.285. Write down values of x and x' in the table.
[OK][INIT+][▲][.][5][OK][▶][▶][SOLVE](wait)[EDIT]	Change initial value of t to 0.25, and final value of t to 0.5, solve again for v(0.5) = 2.640.
[OK][INIT+][▲][.][7][5][OK][▶][▶][SOLVE](wait)[EDIT]	Change initial value of t to 0.5, and final value of t to 0.75, solve again for v(0.75) = 2.066.
[OK][INIT+][▲][1][OK][▶][▶] [SOLVE](wait)[EDIT]	Change initial value of t to 0.75, and final value of t to 1, solve again for v(1) = 1.562

Repeat for t = 1.25, 1.50, 1.75, 2.00. To finish, press [OK], [ON]. The different solutions will be shown in the stack, with the latest result in level 1.

The final results look as follows:

t	v
0.00	4.000
0.25	3.285
0.50	2.640
0.75	2.066
1.00	1.562
1.25	1.129
1.50	0.766
1.75	0.473
2.00	0.249

Plotting the solution to a differential equation of the form dy/dt = f(t,y)

When we can not obtain a closed-form solution for the integral, we can always plot the integral by selecting `Diff Eq` in the `TYPE` field of the PLOT environment as follows: suppose that we want to plot x(t) for v(t) = $\exp(-t^2)$, with x = 0 at t = 0. We know there is no closed-form expression for the integral, however, we know that the definition of v(t) is basically a differential equation, namely, dx/dt = $\exp(-t^2)$. The HP48G series calculator allows for the plotting of the solution of differential equations of the form

$$Y'(T) = F(T,Y).$$

For our case, we let Y→x and T→t, therefore, F(T,Y)→f(t,x) = exp($-t^2$).

Let's plot the solution, x(t), for t = 0 to 5, by using the following keystroke sequence:

[↱][PLOT] To enter PLOT environment

(highlight the field in front of `TYPE`, use the [▲] [▼] keys)
[CHOOS] (highlight `Diff Eq`, use the [▲] [▼] keys) [OK]

[▼][↰][e^x][-][α][↰][T] [y^x][2] [OK]. To define f(t,x).
[α][↰][T][OK] To define t (lowercase) as the independent variable (`INDEP`)
[0][OK][5][OK] To set the range of values of t [0, 5].
[α][↰][X][OK] To define x (lowercase) as the dependent variable (`SOLN`)
[0][OK] To define the initial condition (or initial value) for x
[OPTS] To define plot options. Within this environment, move to the different fields using the [▲] [▼][◄] [►] keys, making sure that the following limits are defined:

```
H-VAR: 0    H-VIEW: -1  5
V-VAR: 1    V-VIEW: -1  1.5
```

Press [OK] when done.
[ERASE][DRAW] To plot the graph.

When you observe the graph being plotted, you'll notice that the graph is not very smooth. That is because the plotter is using a time step that is too large. To refine the graph and make it smoother, use a step of 0.1. Try the following keystrokes:

[CANCL][OPTS] [►][.][1][OK][OK][ERASE][DRAW]

The plot will take longer to be completed, but the shape is definitely smoother than before.

With step = Dflt *With step = 0.1*

Try the following:

[EDIT][NXT][LABEL][MENU] To see axes labels and range.

Notice that the labels for the axes are shown as 0.000 (horizontal) and 1.000 (vertical). These are the definitions for the axes as given in the OPTS screen (see above), i.e., H-VAR (t): 0, and V-VAR(x): 1.

[NXT][NXT][PICT] To recover menu and return to PICT environment.
[(X,Y)] To determine coordinates of any point on the graph.

Use [►] and [◄] to move the cursor in the plot area. At the bottom of the screen you will see the coordinates of the cursor as (X,Y). The HP48G is using X and Y as the default names for the horizontal and vertical axes, respectively.

[NXT][CANCL] To recover the menu and return to the PLOT environment
[ON] To return to stack.

Integrating second-order ordinary differential equations (ODEs)

Problems involving the interaction of a harmonic force (e.g., a mass-spring system) and a damping force result in the equation of motion being a second-order ODE. Integration of such ODEs can be accomplished by defining the solution as a vector. As an example, suppose that a spring-mass system is subject to a damping force proportional to its speed. The resulting differential equation is:

or,

$$x'' = -18.75\,x - 1.962\,x',$$

subject to the initial conditions, v = x' = 6, x = 0, at t = 0. We want to find x, x' at t = 2.
Re-write the ODE as:
or,

$$w' = Aw,$$

where $w = [\,x\ \ x'\,]^T$, and A is the 2 x 2 matrix shown above.
The initial conditions are now written as $w = [0\ \ 6]^T$, for t = 0. (Note: The symbol $[\ ]^T$ means the transpose of the vector or matrix).
To solve this problem using the HP48G or GX, create a subdirectory called ODE2 and move into that subdirectory. First, we'll create the matrix A, as follows:

[↱][MATRIX][0][SPC][1][ENTER][▼][1][8][.][7][5][+/-][SPC][1][.][9][6][2][+/-][ENTER][ENTER]
['][α][A][STO]

Then, use the following keystroke sequence to solve for the differential equation for t = 2 s:

[↱][SOLVE][▼][OK]	To enter SOLVE environment, solving ODEs
['][α][α][A][×][W][α][OK]	Define F(T,Y), as Aw
[α][↰][T] [0][OK][2][OK] [α][W][OK]	Define t as independent variable and define range.
[↰][[ ]] [0][SPC][6] [OK]	Enter initial conditions for w
[SOLVE]	Solve for w(t=2). Wait until hourglass disappears.
[EDIT]	To see the solution vector.

The solution reads [.16716... -.6271...], i.e., x(2) = 0.16716, and x'(2) = v(2) = -0.6271.
Press [CANCL] to return to SOLVE environment.

Solution presented as a table of values

In the previous example we were interested only in finding the values of the position and velocity at a given time t. If we wanted to produce a table of values of x and x', for t = 0.00, 0.25, …, 2.00, we will proceed as follows: First, prepare a table to write down your results:

t	x	x'
0.00	0.00	6.00
0.25		
…	…	…
2.00		

Next, within the SOLVE environment, change the final value of the independent variable to 0.25, use:

[▲][.][2][5][OK] [▶][▶] [SOLVE](wait)[EDIT]	Solves for w at t = 0.25, w = [0.968 1.368]. Write down values of x and x' in the table.
[OK][INIT+][▲][.][5][OK][▶][▶][SOLVE](wait)[EDIT]	Change initial value of t to 0.25, and final

value of t to 0.5, solve again for w(0.5) = [0.748 -2.616]

[OK][INIT+][▲][.][7][5][OK][▶][▶][SOLVE](wait)[EDIT] Change initial value of t to 0.5, and final value of t to 0.75, solve again for w(0.75) = [0.0147 -2.859]

[OK][INIT+][▲][1][OK][▶][▶] [SOLVE](wait)[EDIT] Change initial value of t to 0.75, and final value of t to 1, solve again for w(1) = [-0.469 -0.607]

Repeat for t = 1.25, 1.50, 1.75, 2.00. To finish, press [OK], [ON]. The different solutions will be shown in the stack, with the latest result in level 1.

The final results look as follows:

t	x	x'		t	x	x'
0.00	0.000	6.000		1.25	-0.354	1.281
0.25	0.968	1.368		1.50	0.141	1.362
0.50	0.748	-2.616		1.75	0.227	0.268
0.75	-0.015	-2.859		2.00	0.167	-0.627
1.00	-0.469	-0.607				

Plotting x, x' vs. t for a second-order ODE

To plot x, x' vs. t, use the following:
[↱][PLOT], highlight the `TYPE` field and choose `Diff Eq` (Use [CHOOS]...[OK]). Change the initial and final values of t to 0 and 2, respectively, and the initial value of w to [0 6]. The screen should look like this:

```
TYPE:    Diff Eq          ∠: Rad
F:       'A*W'
INDEP: t   INIT: 0     FINAL: 2
SOLN:  W   INIT: [ -... _STIFF
```

Before plotting, make the following changes in the OPTS screen:

[OPTS] [▼][▶][1][+/-][OK][2][.][5][OK][▶] [5][+/-][OK] [5][OK] [.][5][OK] [5][OK]

Also, make sure that there is *not* a check mark, ✓, in front of `PIXELS` in the lower right corner of this screen. Now, press [◀][◀][▲][1][OK] (This indicates that we want to plot x, the first variable in the vector w). Press [ERASE][DRAW].
To plot the x' vs. t curve, press

[CANCL][OPTS] [▼][▼][2][OK][DRAW] *warning: do not press [ERASE]*

To see labels, press [EDIT][NXT][LABEL][MENU]. The x-axis is identified as 0.000, while the y-axis is identified as 2.000, since variable 2 (x') was the last plotted.
Press [NXT][NXT][PICT], to return to the PICT environment. Press [CANCL] to return to the PLOT environment, and press [ON] to return to the stack.
While in normal calculator mode, you can always see the last graph you produced, if it hasn't been erased, by pressing [↰][PICTURE].

Appendix A - Statistical functions in the HP48G/GX using [↰][STAT]

The keystroke combination [↰][STAT] provides direct access to several of the statistical functions in the calculator, namely:

[DATA][ΣPAR][1VAR][PLOT][FIT][SUMS]

Pressing the key corresponding to any of these menus provides access to different functions as described below.

[DATA]: Commands under this menu are used to manipulate the statistics matrix ΣDATA.

[Σ+]: add row in level 1 to bottom of ΣDATA matrix.

[Σ-]: removes last row in ΣDATA matrix and places it in level of 1 of the stack. The modified ΣDATA matrix remains in memory.

[CLΣ]: erases current ΣDATA matrix.

[ΣDAT]: places contents of current ΣDATA matrix in level 1 of the stack.

[↰][ΣDAT]: stores matrix in level 1 of stack into ΣDATA matrix.

[STAT]: returns to STAT menu.

[ΣPAR]: Commands under this menu are used to modify statistical parameters. The parameters shown in the display are:

Xcol: indicates column of ΣDATA representing x (Default: 1)
Ycol: indicates column of ΣDATA representing y (Default: 2)
Intercept: shows intercept of most recent data fitting ((Default: 0)
Slope: shows slope of most recent data fitting (Default: 0)
Model: shows current data fit model (Default: LINFIT)

n [XCOL]: changes Xcol to n.

n [YCOL]: changes Xcol to n.

[MODL]: lets you change model to LINFIT, LOGFIT, EXPFIT, PWRFIT or BESTFIT by pressing the appropriate button, or press [ΣPAR] to return to the ΣPAR menu.

[ΣPAR]: shows statistical parameters.

[RESET]: reset parameters to default values

[INFO]: shows statistical parameters

[NXT][STAT]: returns to [STAT] menu.

[1VAR] : Commands under this menu are used to calculate statistics of columns in ΣDATA matrix.

[TOT]: show sum of each column in ΣDATA matrix.

[MEAN]: shows average of each column in ΣDATA matrix.

[SDEV]: shows standard deviation of each column in ΣDATA matrix.

[MAXΣ]: shows maximum value of each column in ΣDATA matrix.

[MINΣ]: shows average of each column in ΣDATA matrix.

x_s, Δx, n [BINS]: provides frequency distribution for data in *Xcol* column in ΣDATA matrix with the frequency bins defined as $[x_s,x_s+\Delta x]$, $[x_s,x_s+2\Delta x]$,..., $[x_s,x_s+n\Delta x]$.

[NXT]: to access the second menu. Within this menu you will find the following commands:

[VAR]: shows variance of each column in ΣDATA matrix.

[PSDEV]: shows population standard deviation (based on *n* rather than on *(n-1)*) of each column in ΣDATA matrix.

[PVAR]: shows population variance of each column in ΣDATA matrix.

[MINΣ]: shows average of each column in ΣDATA matrix.

[STAT]: returns to [STAT] menu.

[PLOT]: Commands under this menu are used to produce plots with the data in the ΣDATA matrix.

[BARPL]: produces a bar plot with data in *Xcol* column of the ΣDATA matrix.

[HISTP]: produces histogram of the data in *Xcol* column in the ΣDATA matrix, using the default width corresponding to 13 bins unless the bin size is modified using
[↰][STAT][1BAR][BINS]. Press [CANCL] to return to normal display.

[SCATR]: produces a scatterplot of the data in Ycol column of the ΣDATA matrix vs. the data in Xcol column of the ΣDATA matrix. Press [CANCL] to return to normal display. Equation fitted will be stored in the variable EQ.

[STAT]: returns to [STAT] menu.

[FIT]: Commands under this menu are used to fit equations to the data in columns *Xcol* and *Ycol* of the ΣDATA matrix.

[ΣLINE]: provides the equation corresponding to the most recent fitting.
[LR]: provides intercept and slope of most recent fitting.
y [PREDX]: given y find x for the fitting y = f(x).
x [PREDY]: given x find y for the fitting y = f(x).
[CORR]: provides the correlation coefficient for the most recent fitting.
[COV]: provides sample co-variance for the most recent fitting

[NXT]: to access the second menu. Within this menu you will find the following commands:
[PCOV]: shows population co-variance for the most recent fitting.
[STAT]: returns to [STAT] menu.

[SUMS]: Commands under this menu are used to obtain summary statistics of the data in columns *Xcol* and *Ycol* of the ΣDATA matrix.

[ΣX]: provides the sum of values in *Xcol* column.
[ΣY]: provides the sum of values in *Ycol* column.
[ΣX^2]: provides the sum of squares of values in *Xcol* column.
[ΣY^2]: provides the sum of squares of values in *Ycol* column.
[ΣX*Y]: provides the sum of $x \cdot y$, i.e., the products of data in columns *Xcol* and *Ycol*.
[NΣ]: provides the number of columns in the ΣDATA matrix.

Example: let ΣDATA be the matrix:

- Type the matrix in level 1 of the stack by using the MATRIX editor: [↱][MATRIX]. When done entering values, press [ENTER].

- To store the matrix into ΣDATA, use: [↰][STAT] [DATA] [↰][ΣDAT]

- Calculate statistics of each column: [STAT][1VAR]

[TOT]	produces [38.5 87.5 82799.8]
[MEAN]	produces [5.5. 12.5 11828.54…]
[SDEV]	produces [3.39… 6.78… 21097.01…]
[MAXΣ]	produces [10 21.5 55066]
[MINΣ]	produces [1.1 3.7 7.8]
[NXT][VAR]	produces [11.52 46.08 445084146.33]
[PSDEV]	produces [3.142… 6.284… 19532.04…]
[PVAR]	produces [9.87… 39.49… 381500696.85…]

- Generate a scatterplot of the data in columns 1 and 2 and fit a straight line to it:

[STAT][ΣPAR][RESET]	resets statistical parameters
[NXT][STAT][PLOT][SCATR]	produces scatterplot
[STATL]	draws data fit as a straight line
[CANCL]	returns to main display

- Determine the fitting equation and some of its statistics:

[STAT][FIT][ΣLINE]	produces `'1.5+2*X'`
[LR]	produces `Intercept: 1.5, Slope: 2`
3 [PREDX]	produces 0.75
1 [PREDY]	produces 3. 50
[CORR]	produces 1.0
[COV]	produces 23.04
[NXT][PCOV]	produces 19.74

- Obtain summary statistics for data in columns 1 and 2: [STAT][SUMS]

[ΣX]	produces 38.5
[ΣY]	produces 87.5
[ΣX^2]	produces 280.87
[ΣY^2]	produces 1370.23
[ΣX*Y]	produces 619.49
[NΣ]	produces 7

- Fit data using columns 1 (x) and 3 (y) using a logarithmic fitting:

[NXT][STAT][ΣPAR][3][YCOL]	select Ycol = 3, and
[MODL][LOGFI]	select Model = Logfit
[NXT][STAT][PLOT][SCATR]	produce scattergram of y vs. x
[STATL]	show line for log fitting

Obviously, the log-fit is not a good choice.

[CANCL]	returns to normal display.

- Select the best fitting by using:

[STAT][ΣPAR][MODL][BESTF]	shows EXPFIT as best fit for these data
[NXT][STAT][FIT][ΣLINE]	produces '2.6545*EXP(0.9927*X)'
[CORR]	produces 0.99995… (good correlation)

2300[PREDX]	produces 6.8139
5.2 [PREDY]	produces 463.37

- To return to STAT menu use: [NXT][STATS]

- To get your variable menu back use: [VAR].

Part II – Programs for numerical methods using the HP 48 G/G+/GX

This part of the guidebook contains information describing a number of sub-directories where numerical solutions have been implemented for matrix and linear algebra operations, eigenvalue estimation, solution of non-linear equations, polynomial approximation and interpolation, and numerical integration.

<u>**Note**</u>: You will find many references to "the textbook" in this guide book. The textbook used to develop the present guidebook was Hoffman, J.D. , "Numerical Methods for Engineers and Scientists," Mc-Graw Hill, Inc. , New York. 1992. However, the subjects referred to should be available in any numerical methods textbook.

Numerical methods with matrices

Within the sub-directory HOME\NUMM\MATX you will find the following variables:

[→ABS][MSIM][LUFAC][TRIDG][ITRM][EIGEN]

The operation of [→ABS] and of [MSIM] has been discussed previously. In this guidebook we discuss the operation of the other subdirectories.

LUFACT: LU factorization using Crout method

Press [LUFAC] to get into this sub-directory. You will find the following variables:

[A][→LUF][P][Ap][U][L]

A is used to store a square matrix, e.g.,

In the calculator, use the following keystrokes:

[[2 3 5][3 1 -2][1 3 4]] [ENTER]	To enter matrix in level 1
[↰][A]	To store matrix in variable A
[A]	To check that storage took place

Press [→LUF] to perform the LU factorization through Crout method with pivoting. You will get a message box with the message "Ready", indicating that the factorization was performed. Press [OK]. The result is the matrices **P**, **Ap**, **U** and **L**, such that

$$\mathbf{P} \cdot \mathbf{A} = \mathbf{Ap} = \mathbf{L} \cdot \mathbf{U}$$

P is a permutation matrix indicating any row exchange (pivoting) used to improve the LU decomposition. **A** is the original matrix. **Ap** is the permuted matrix, which can be calculated by multiplying **P** with **A**. **L** and **U** are the lower and upper triangular matrices resulting form the Crout method application on **Ap**.

For example, for the matrix A used above, press [P] to get:
This can be interpreted as indicating that, after pivoting, the original row 1 ended in row 3, row 2 into row 1, and row 3 into row 2. To verify this, press [Ap], to get

Compare the location of each row in Ap with those of A to see how the matrix P can be used to interpret the pivoting.

The corresponding triangular matrices L and U are obtained by pressing [L] and [U], respectively. For the present case they are:

When using matrices **L** and **U** for solving for the linear system, **A**·**x**=**b**, we first re-write the linear system as **P**·**A**·**x** = **P**·**b**. If we call **P**·**b** = **b***, then, the linear system becomes **Ap**·**x** = **b***. Since, **Ap** = **L**·**U**, we re-write the linear system as **L**·**U**·**x** = **b***, from which we can write **U**·**x** = **b'**, **Lb'** = **b***. Therefore, we would first solve for **b'** from **L**·**b'** = **b***, and then solve for **x** by using **U**·**x** = **b'**.

For example, for the present case, if **b** = $[1\ \ 3\ \ 5]^T$, we will calculate **b*** = **P**·**b** = $[3\ \ 5\ \ 1]^T$. Then, solve for **b'** from **L**·**b'** = **b***, i.e., if **b'** = $[b_1'\ b_2'\ \ b_3']^T$, we need to solve the set of equations:

$$
\begin{aligned}
3\cdot b_1' &= 3\\
b_1'+2.67\cdot b_2' &= 5\\
b_1'+2.33\cdot b_2'+2.25\cdot b_3' &= 1
\end{aligned}
$$

The solution is b_1'=1.00, b_2' = 1.50, and b_3'=-2.00, i.e., **b'** = $[1\ \ 1.5\ \ -2]^T$.
Next, we solve for **x** from **U**·**x** = **b'**, i.e., if **x** = $[x_1\ x_2\ \ x_3]^T$, we need to solve the set of equations:

$$
\begin{aligned}
x_1+0.33\cdot x_2-0.67\cdot x_3 &= 1\\
x_2+1.75\cdot x_3 &= 5\\
x_3 &= 3
\end{aligned}
$$

The solution is x_1= -2.00, x_2 = 5.00 , and x_3= -2.00, i.e., **x** = $[-2\ \ 5\ \ -2]^T$. To check, press [A], enter the vector x, [-2 5 -2][ENTER], and multiply, [×]. You should recover the vector **b** = $[1\ \ 3\ \ 5]^T$.

TRIDG: Thomas algorithm for the solution of linear equations with a tri-diagonal matrix

The sub-directory TRIDG is located at HOME\NUMM\MATX\TRIDG. Within that subdirectory you find the following variables:

[A][B][→SOL][X][→AB][→X]

To operate using this sub-directory you need to enter the tri-diagonal matrix (n×n) as a matrix **A** of three rows and n columns. The columns correspond to the diagonal below the main diagonal, the main diagonal itself, and the diagonal above the main diagonal, respectively. This means that the elements a11 and an3 will be zero.

For example, try entering this matrix: [[0 4 -1][-1 4 -1][-1 4 -1][-1 4 0]][ENTER]. To store it in A, use [↰][A].

Also, enter a vector **b** = [150 200 150 100][ENTER]. Store it in variable b using: [↰][B].

To solve for vector x in the equation **Ax** = **b** press the button [→SOL]. The result for this particular case is **x** = [57.89 81.57 68.42 42.11].

The variables [→AB] and [→X] are used to implement the Thomas algorithm: [→AB] simplifies **A** and **b**, and [→X] performs the back substitution to solve for **x**. These two programs are called by [→SOL]. To see details of these programs use: [↱][→AB], [↱][→X], and [↱][→SOL].

ITRM: Iterative methods for solution of systems of linear equations

This subdirectory, located in HOME\NUMM\MATX\ITRM contains programs that allow you to show the average residual and the current iteration of the value of x using either the Jacobi method ([→JAC]) or successive overrelaxation ([→SOR]). [Note: when used with ω = 1.00, the SOR method becomes the Gauss-Seidel iterative method.]

To solve the system **Ax** = **b**, you need to enter the square matrix **A** and the vector **b** in the corresponding variables A and b. For example:

[[4 -1 0 0] [-1 4 -1 0] [0 -1 4 -1] [0 0 -1 4]][ENTER] [↰][A].

[150 200 150 100][ENTER] [↰][B].

We need to enter an initial value for the vector x, say [0 0 0 0][ENTER] [↰][X].

To use the *Jacobi method*, just press [→JAC]. The result is the average absolute value of the residual, R = 150, and the improved value of x = [37.5 50 37.5 25]. The average residual is still pretty large, therefore, we need to keep iterating. Press [→JAC] and watch the results until the value of R is small enough. For the present case, the values of R for the second through the twelfth iteration are: 59.375, 24.22, 9.76, 3.96, 1.60, 0.65, 0.26, 0.11, 0.043, 0.017, 0.007. The latest value is close enough to zero. The corresponding solution is [57.89 81.58 68.42 42.10]. Compare with those obtained through use of the tri-diagonal method.

To use the *overrelaxation method*, enter a value for the parameter ω, say, [1][.][5] [↰][ω]. Re-initialize the solution by using: [0 0 0 0][ENTER] [↰][X]. Then, press the [→SOR] button until the residual is as close to zero as you want it. For the first twelve iterations you should get the following values of R: 211.16, 57.40, 27.57, 12.91, 7.23, 5.17, 2.89, 1.51, 0.88, 0.54, 0.23, 0.081. At this point the solution is x = [57.88 81.58 68.43 42.12]. A better rate of convergence is achieved by using ω = 1.1, with the average residual being 0.014 after 6 iterations only.

Recall that by using ω = 1.0, the SOR method becomes the Gauss-Seidel iteration method.

EIGEN: Calculation of eigenvalues

This sub-directory, located at HOME\NUMM\MATX\EIGEN contains three sub-directories, namely [CEQ2], [CEQ3], and [FADLE]. The sub-directories [CEQ2] and [CEQ3] are used to produce the coefficients of the polynomial that represents the characteristic equation det(**A**-λ**I**) = 0, for the matrix A having a size of 2×2 and 3×3, respectively.

For a 2×2 matrix, the characteristic equation is given by:

$$\lambda^2 - \mathrm{tr}(\mathbf{A})\lambda + \det(\mathbf{A}) = 0,$$

where tr(**A**) is the trace of **A**.

For a 3×3 matrix, the characteristic equation is given by:

$$\lambda^3 - \mathrm{tr}(\mathbf{A})\lambda^2 + (\Sigma\det(\mathbf{M}_{ij}))\lambda - \det(\mathbf{A}) = 0,$$

where tr(**A**) is the trace of **A**, and $\mathbf{M}_{ij}$ are the reduced 2×2 matrices, $\mathbf{M}_{ij}$ = [[a_{ii} a_{ij}][a_{ji} a_{jj}]], and the summation includes three reduced matrices corresponding to (i = 1, j=2), (i = 1, j=3), and (i = 2, j=3).

To operate [CEQ2] or [CEQ3] enter the corresponding 2×2 and 3×3 matrix into A, then press [→EVAL]. The result, in any of those sub-directories, is a vector containing the coefficients of the polynomial characteristic equation corresponding to descending order of powers of λ, and a second vector with the roots of the corresponding polynomial, i.e., with the eigenvalues.

For example, in [CEQ2], enter [[2 3][3 1]] [↰][A], then press [→EVL]. The results are:
Coeff: [1 -3 -7], i.e., $\lambda^2 -3\lambda-7 = 0$, and, λ = [-1.54 4.54].

For a second example, in [CEQ3], enter [[2 3 5][3 1 -2][1 3 4]] [↰][A], then press [→EVL]. The results are: Coeff: [1 -7 6 -18], i.e., $\lambda^3 -7\lambda^2 +6\lambda-18 = 0$, and, λ = [(0.25,0.65),(0.25,-0.65),(6.50,0)]. In this case two of the eigenvalues are complex numbers, namely, λ_1= 0.25+0.65 *i*, and λ_2 = 0.25-0.65*i*.

The *sub-directory FADLE* (HOME\NUMM\MATX\EIGEN\FADLE) contains an implementation of the Fadeev-Levelier method to generate the polynomial characteristic equation for a square matrix of any size. To operate, enter the matrix A, for example: [[1 2 3 1][2 1 2 -2][3 2 5 -1][1 3 -2 6]] [↰][A], then press [→EVL]. The result is this case is Coeff: [1 -13 39 19 -40], i.e., $\lambda^4 -13\lambda^3 +39\lambda^2 +19\lambda -40 = 0$, and, λ = [0.90 -1.06 5.45 7.71].

Note: once you have determined the eigenvalues of a matrix, you can find their associated eigenvectors by using the equation $(\mathbf{A}-\lambda\mathbf{I})\mathbf{x} = 0$, and solving for **x** as outlined in page 69 of the textbook.

POWER(1): calculating the largest (in absolute value) eigenvalue by the power method

This subdirectory, located in HOME\NUMM\MATX\EIGEN\POWER, contains the program [→NEW] that lets you iterate the procedure for the power method for eigenvalues. This method calculates improved values of the largest (in absolute value) eigenvalue of a matrix and its associated eigenvector.
Within sub-directory POWER you will find the following variables:

[A][AINV][X][IU][→NEW][→NEI]

To obtain the largest (in absolute value) eigenvector, enter the matrix **A** (n×n) in the corresponding variable A. Enter a first guess for the vector **x**, typically a vector will all n components equal to 1.0, in variable x. The method requires that one of the components in the vector remain equal to 1.0 at each iteration. Therefore, store the index corresponding to that component in variable IU, i.e., IU can take a value of 1 to n. Press [→NEW] to obtain the next value of the eigenvalue λ and its associated eigenvector x. Continue pressing [→NEW] until the eigenvalue converges to a steady value.

As an example, use: [[2 3 5][3 1 -2][1 3 4]] [↰][A]
[1 1 1][ENTER] [↰][X]
1 [↰][IU]
Press [→NEW] until λ converges.

For this example you should get values of λ = 10.0, 6.60, 6.36, 6.49, 6.51, 6.50, 6.50, 6.50. Thus, the largest (in absolute value) eigenvalue converges to 6.50. The corresponding eigenvector is [1.00 0.28 0.73].
Repeat the problem with IU = 2, i.e., 1 [↰][IU]. The values of λ are now 2.00, 8.00, 7.38, 6.49, 6.45, 6.50, 6.51, 6.50, 6.50. As expected, λ converges to 6.50, but the eigenvector is now [3.59 1.00 2.63]. You can, however, show that [3.59 1.00 2.63] = 3.59·[1.00 0.28 0.73], i.e., the two eigenvectors are parallel, and, therefore, equivalent. [Note: vectors **u** and **v** are parallel if **u** = c·**v**, where c is a real constant].

POWER(2): calculating the smallest (in absolute value) eigenvalue by the inverse power method

Sub-directory POWER also contains the program [→NEI] that lets you iterate the procedure for the inverse power method for eigenvalues. This method calculates improved values of the smallest (in absolute value) eigenvalue of a matrix and its associated eigenvector.
First, enter the matrix **A** (n×n) in level 1 of the display, and press [ENTER] to obtain a second copy. Store the matrix in the variable A for future reference. Press the [1/x] button to obtain the inverse, $\mathbf{A}^{-1}$. Enter a first guess for the vector **x**, typically a vector will all n components equal to 1.0, in variable x. The inverse power method also requires that one of the components in the vector remain equal to 1.0 at each iteration. Therefore, store the index corresponding to that component in variable IU. Press [→NEI] to obtain the next value of the eigenvalue λ and its associated eigenvector x. Continue pressing [→NEI] until the eigenvalue converges to a steady value.

As an example, use:

```
[[ 2 3 5][3 1 -2][1 3 4]] [ENTER][ENTER]
[↰][ A ]
[1/x]
[↰][AINV]
[ 1 1 1 ][ENTER] [↰][ X ]
1 [↰][ IU ]
Press [→NEI] until λ converges.
```

For this example you should get values of λ = 9.00, 0.55, 2.54, -1.73, 1.31, -3.30, 0.73, -12.17, 0.22, 9.94, -0.29, 3.50, -0.92, 1.95, -1.90, at this point you have already performed 15 iterations and there is no convergence. (I tried up to 40 iterations without convergence). The reason is that this particular matrix has only one real eigenvalue, the largest one that we found earlier, i.e., λ = 6.50. The other two eigenvalues are complex, therefore, the inverse power method will not be able to provide a solution for the smallest (in absolute value) eigenvalue, and we would have to rely on the Fadeev-Leverlier method presented earlier to obtain the other characteristic equation and the eigenvalues.
There is no reliable way to determine whether a matrix of real coefficients has complex eigenvalues. One known fact is, however, that symmetric real matrices have only real eigenvalues. Let's try using the inverse power method with a symmetric matrix:

```
[[ 2 3 5][3 1 -2][5 -2 4]] [ENTER] [ENTER]
[↰][ A ]
[1/x]
[↰][AINV]
[ 1 1 1 ][ENTER] [↰][ X ]
1 [↰][ IU ]
Press [→NEI] until λ converges.
```

The first ten values of λ for this case are: 3.67, 9.31, 2.20, 6.39, 2.21, 4.77, 2.58, 4.07, 2.88, 3.07. Although the eigenvalue seems to be oscillating from a small number to a larger one and so forth, it is actually converging to a number ...(the last two values). Let's keep using the [→NEI] program, with values of λ = 3.55, 3.19, 3.46, 3.25, 3.41, 3.29, 3.38, 3.31, 3.36, 3.33, 3.35, 3.33, 3.34, 3.34, 3.34. Convergence is achieved after about 25 iterations. It seems that convergence is slower for the inverse power method. The smallest (in absolute value) eigenvalue is, therefore, λ = 3.34, and its associated eigenvector is [1.00 2.16 -1.03].
You can check, by using [1 1 1] as the initial value of x, and repeated use of the program [→NEW], that the largest (in absolute value) eigenvalue for this symmetric real matrix is λ = 8.12, and its associated eigenvector is [1.00 0.09 1.17].

Shifting eigenvalues procedure

I have not prepared a separate program for applying the method of shifting eigenvalues (page 77-79 in textbook) since it basically uses the power and inverse power methods already programmed in sub-directory POWER.

As an example, we will follow the procedure suggested in page 77 of the textbook for the matrix **A** used above, i.e., **A** = [[2 3 5][3 1 -2][5 -2 4]]. Start by storing **A** in variable A. The steps to follow next, as indicated in page 77, are:

1. *Solve for the largest (in magnitude) eigenvalue* λ_1. You can verify that $\lambda_1 = 8.12$ using the power method as shown above.

2. *Shift the eigenvalues of A by* $s = \lambda_1$ *(= 8.12).* This is accomplished by calculating a matrix $\mathbf{A}_s = \mathbf{A} - \lambda_1\mathbf{I}$. In the calculator you can use the following keystrokes:

   [A][8][.][1][2][ENTER][3][MTH][MAKE][IDN][×][-]

   Store this new matrix in A.

3. *Solve for the eigenvalue* λ_s *of the shifted matrix* A_s *by the power method.* Using the power method procedure outlined above, we find $\lambda_s = -12.58$.

4. *Calculate the largest eigenvalue of opposite sign by* $\lambda = \lambda_s + s$. For this case, $\lambda = \lambda_s + s = -12.58 + 8.12 = -4.46$.

Solution of non-linear equations

Under subdirectory NUMM you will have a subdirectory called NLEQ (Non-Linear EQuations) containing the following subdirectories:

- TPLOT: used to illustrate x-y plots in the calculator
- INTH: interval-halving solution for non-linear equations
- P3.7: refers to problem 3.7 in the textbook, in which the fixed-point iteration method is used
- NWT: Newton-Raphson method for solving non-linear equations
- SECN: secant method for solving non-linear equations
- TWOC: used to illustrate conic plots in the calculator
- NLSY: examples of systems of non-linear equations when the equations are not those of conics.

NOTE: The examples contained in TWOC and NLSY will be solved using the library SOLVESYS, developed by Sune Bredahl (e-mail: c947086@student.dtu.dk), and available in the Internet at the following URL: www.gbar.dtu.dk/~c947086/hp48.html
(Mr. Bredahl is a computer scientist from Denmark who develops binary applications for the HP48G/GX calculator as a hobby).

Interval Halving Method

Use subdirectory INTH. Within that subdirectory you will find the following variables:

[A][B][C][F][→NXC][X]

If you press [NXT] you will get the next menu:

[EQ][PPAR][ZPAR][CHSL][CHKS][ ]

For solving non-linear equations using the interval-halving method we need to be concerned only with variables A, B, C, F, and →NXC. To plot the function f(x) we need to use the variable EQ. X, PPAR and ZPAR are used in the plotting of the function. CHSL and CHKS are programs used in the implementation of the interval-halving method.
Press [NXT] or [VAR] to get to the main menu. Press [↱][F] to see the function that is currently defined in variable F. You will get the following program:

```
<< → x 'x^3-2*x^2-2*x+1' >>
```

This is the function from problem 1(d).

Graphics solution

If you press [NXT][EQ], you will find that the current value of the EQ variable is the function `'x^3-2*x^2-2*x+1'`, i.e., the same as the algebraic expression included in F. You use the variable EQ to plot the function and give you an idea of where the roots are located. To plot the function use the procedure outlined above for plotting expressions of the type y=f(x). For this particular case use the following keystroke sequence to see the plot:[↱][PLOT][ERASE][DRAW] (The plot ranges are already defined). Press [EDIT][NXT][LABEL] to see the labels. Press [NXT][NXT][PICT][FCN] and use the [ROOT] key to find the three roots of the curve. You should obtain as solutions: -1, 0.3819, and 2.6180. Now, that would be cheating since we have not use interval-halving at all. To use the method, try the following. Get out of the plot environment:

[NXT][NXT][PICT][CANCL][ENTER].

Numerical solution

Press [VAR] to see the main menu. Now, we know that the first solution is -1 (because we cheated, as shown above), but suppose that we didn't. We did notice from the graph that the curve goes from negative to positive when x goes from -1 to 0, so let's use a = -1 and b=0, by using:

[2][+/-][↰][A]
[0][↰][B]

Then, to obtain c, press [→NXC] (NeXt C). It shows c:-1. Pressing [→NXC] shows the message: `A solution exists at a.` Press [OK] and then [A] to see what the solution is. Surprise! It is equal to -1. Let's try something more challenging.
Select a = 0, b = 1.5. Enter:

[0][↰][A]
[1][.][5][↰][B]

The press [→NXC]. You'll get c: 0.75. Press [→NXC] again. Keep pressing it until the value of c converges to a steady value. You should get values of c = 0.375, 0.5625, 0.4687, 0.4218, 0.3984, 0.3867, 0.3808, 0.3837, 0.3823, 0.3815, 0.3819, 0.3821, 0.3820, 0.3820, 0.3819, 0.3819, 0.3819. After 18 iterations the interval-halving method is providing a value of c close to 0.3819. To see how well our value of 0.3819 satisfies the condition f(x) = 0, enter 0.3819 in level 1 of the display and press the key [F]. We find that f(0.3819) = 0.000204. In other words, the error incurred in evaluating f(x) with x = 0.3819 is of the order of 2E-4. That is close enough to f(x) = 0, that we will take the second root of the equation to be x = 0.3819. Compare with the value found above using the ROOT command in the HP48G.

To find the third root, try using a = 2, b = 3. Within 18 iterations your solution should converge to x = 2.6180, which is the same found through the use of the ROOT command.

If you want to use this subdirectory to solve for the roots of a different function you need to replace the values of the variables F and EQ. As an illustration, let's use interval halving to solve for problem 1(a). First enter the algebraic expressing defining the function, i.e., 'x-COS(x)' in level 1 of the display:

['][α][↰][X][-][COS][α][↰][X][ENTER][ENTER]

Then, store this expression into EQ:

[NXT][↰][EQ]

Now, modify the expression in level 1 to read `'f(x) = x-COS(x)'` by using:

[↰][EDIT][▶][α][↰][F][↰][()] [α][↰][X][▶][↰][=][ENTER]

Now, we will define the function as a program by using:

[↰][DEF]

To check the current contents of variable F, press [↱][F]. It should show: `<< → x 'x - COS(x)' >>`

To plot the function use a procedure similar to that shown above. We will go directly to solving the equation by using the interval limits suggested by the problem, i.e., a = 0.5 and b = 1.0. Enter those values in the corresponding variables:

[0][.][5][↰][A]
[1][↰][B]

Using the program [→NXC], after about 20 iterations, the solution converges to c: 0.7390. Let's check the value of the function f(x) for x = 0.7390. Enter that value in level 1 of the display and press [F]. The result is -0.0001424. The error is then of the order of 1E-4.

Fixed point iteration

An example of this method is contained in subdirectory P3.7 (Problem 3.7). Within that subdirectory you will find the following menu:

[XA][XB][XC][EQ][EQ1][EQ2]

Pressing [NXT] shows the second menu:

[EQ3][X][PPAR]

The variables XA, XB, and XC were loaded by using [↰][DEF] for the following expressions:

```
'xa(x)=EXP(x)-(3*x+2)'
'xb(x)=(EXP(x)-2)/3'
'xc(x)=LN(3*x+2)'
```

The resulting programs can be seen by pressing the following:

[↱][XA]	To get: `<< → x, 'EXP(x)-(3*x+2)' >>`
[↱][XB]	To get: `<< → x, '(EXP(x)-2)/3'>>`
[↱][XC]	To get: `<< → x, 'LN(3*x+2)' >>`

The variables EQ1, EQ2, and EQ3 contain lists of functions of x, namely,

`EQ1 = {x 'EXP(x)-(3*x+2)'}`, `EQ2 = { x '(EXP(x)-2)/3'}`, and `EQ3 = {x 'LN(3*x+2)'}`.

They are used to show the intersection of the curves y = x and y = f(x), which is the solution to each of the corresponding problems.

Graphics solution

For example, the current content of EQ is the same as EQ1. To plot the figure use: [↱][PLOT][ERASE][DRAW]. The resulting figure shows two points of intersection. To find what they are, use [FCN], then move the cursor close to either one of them, say the left one, and press [ISECT]. The result is I-SECT: (-0.4552, -0.4552). The second point of intersection is located at (2.1253,2.1253). In other words, the solutions for the expression $x = exp(x) -(3x+2)$ are x=-0.4552 and x = 2.1253. Press [NXT][NXT][PICT][CANCL][ENTER] to return to the normal display. The coordinates of the intersection points will be shown in the stack.
To solve case b and c using the PLOT environment, first you need to copy the contents of EQ2 or EQ3 into EQ, then follow a procedure similar to that shown above. You may need to change the plot ranges (horizontal and vertical), but the general procedure is the same as above.

Numerical solution

The solution using fixed-point iteration consists in entering an initial value for whichever case you want to solve, the pressing the corresponding function [XA], [XB], or [XC], until the function converges.

(a) For example, enter 0.0 and press [XA] repeatedly. You should get values oscillating between positive and negative without convergence. Let's try a different starting value, say -0.5. Enter that value and press [XA] repeatedly, the results again diverge. Let's use a positive value, say, 2.0, to start the solution. Again, the solution diverges. It seems that the fixed-point equation defined in XA is very unstable in the sense that it quickly diverges regardless of the initial value chosen for the solution.

(b) Let's try instead XB, with x = 0. Press [XB] repeatedly to get: -0.3333, -0.4278, -0.4493, -0.01388, -0.3379, -0.4289, -0.4495, -0.4503, -0.4549, -0.4551, -0.4552, -0.4552. One solution is then, x = -0.4552. Starting with values of 1.0 or 2.0 will get you back to the negative solution. Starting with a value of 2.5 or 3.0 will give you a divergent process. It seems that using XB will only allow you to obtain one solution.

(c)Let's try instead XC with x = 0 (or any other positive number). Pressing [XC] repeatedly will give you a solution at x = 2.1253. Starting with a negative value, say -1.0, will give you a complex solution. The expression for XC, therefore converges only to the positive solution.

Moral of the exercise: fixed-point iteration depends not only on the initial value selected for the iteration, but also in the equation chosen to solve the problem. One of the expressions used here always gave us divergence, and the other two would only show convergence towards one, but not the other, solution.

Solving other problems: simply define your function xf = f(x) by entering the expression in level 1, as 'xf = f(x) ' and using [↰][DEF]. Then, select an initial value, and press [XF] repeatedly until convergence is achieved or until it is clear that the solution oscillates or diverges. In the latter case, try a different expression for xf = f(x).

Newton-Raphson method

Use subdirectory NWT. In that subdirectory you will find the menu:

[X][→NEW][F][FP][EQ][PPAR]

The next menu is obtained by pressing [NXT]:

[ZPAR] [ ] [ ] [ ] [ ] [ ]

EQ, PPAR and ZPAR are used for plotting. The function [F] is a program, generated by using [↰][DEF], while [EQ] contains the same algebraic expression as used in [F]. The function [FP] contains the derivative of [F]. This was also defined using [↰][DEF].

The current examples has f = `<< → x 'x^3-2*x+1'>>`, fp = `<< → x '3*x^2-4*x'>>` , EQ = `'x^3-2*x+1'`.

Graphics solution

The procedure is the same as that shown for the Interval-halving method. Therefore, the graphics solution will not be presented here. The solutions, obtained using the command ROOT in the PLOT environment are: -0.6180, 1.00, 1.6180.

Numerical solution

Enter an initial value for x, then press [→NEW] until the solution converges. For example, set x = -10, i.e.,

[1][0][+/-][↰][X]

Pressing [→NEW] repeatedly produces values of x = -10.0, -6.4735, -4.1380, -2.6053, -1.6223, -1.0291, -0.7263, -0.6286, -0.6181, -0.6180, -0.6180, -0.6180. The solution converges to x = -0.6180. Enter that value in level 1 of the stack and press [F]. The result is f(-0.6180) = 0.0001229, i.e., the error is of the order of 1E-4.
Now, enter a value of x = 10.0, and press [→NEW] repeatedly to prove that the solution converges to x = 1.6180, with and error of the order of 4E-5.
To find the third solution you have to enter a value between 0.5 and 1.25, otherwise, the solution converges to either of the two solutions found before. For example, try x = 0.8. The solution converges to 1.0 in 4 iterations. The error is zero, i.e., this is an exact solution.

A second example

You can use the procedure programmed in [→NEW] (for NEWton-Raphson) to obtain numerical solutions to other equations as long as you enter the original function (using '*f(x) = expression containing x*' and [↰][DEF]) and its derivative (using '*fp(x) = expression containing x*' and [↰][DEF]). You can even use the calculator to obtain the derivative for you. The following example shows you how to proceed.

Suppose that $f(x) = e^x - \sin(\pi x/3)$, the procedure to load EQ, F and FP is as follows:

- Enter the algebraic expression defining f(x) in level 1 of the stack, and make two copies of it:

  ['][↰][e^x] [α][↰][X] [▶] [-] [SIN] [↰][π] [×] [α][↰][X] [÷] [3] [ENTER][ENTER][ENTER]

  There will be three copies of the expression `'EXP(x)-SIN(π*x/3)'` in levels 1 through 3 of the stack.

- For graphical analysis, store one copy of the expression in EQ by using: [↰][EQ]
- Modify the second copy to make it look like this: `'f(x)=EXP(x)-SIN(π*x/3)'`. Use:

  [↰][EDIT][▶][α][↰][F][↰][()] [α][↰][X][▶][↰][=][ENTER]

- Load f(x) by using: [↰][DEF]

Next, we need to take the derivative of x using the HP48G/GX command, but first, we need to purge the variable [X]. Otherwise, the derivative will be evaluated at the current value of x instead of producing a general expression fp(x).

- Purge [X] by using: ['][X][↰][PURG]
- Evaluate the derivative by entering: ['] [α][↰][X] [ENTER] [↱][∂]

  The resulting expression is: `'EXP(x)-COS(π*x/3)*(π/3)'`
- Modify the expression to read: `'fp(x) = EXP(x)-COS(π*x/3)*(π/3)'` by using:
- [↰][EDIT][▶][α][α] [↰][F] [↰][P] [↰][()] [↰][X] [α] [▶][↰][=][ENTER]

- Load fp(x) by using: [↰][DEF]
- Enter a value of x, say x = 1.0, by using: [1] ['] [α][↰][X] [STO].
- Press [VAR], if needed, to see the subdirectory variable menu.

At this point all the required variables are loaded and you can proceed as in the first example described above. However, there is a small surprise in store for you. If you press [→NEW], instead of getting a numerical value, you get an algebraic expression. Try it. That is so because, every time that you have p in an expression, the calculator tends to preserve the expression as an algebraic rather than producing a numeric result. To force the calculator to produce numeric results when evaluating [F] and [FP], we need to modify the programs stored in those variables as follows:

First, recall and modify the contents of [F] by using:

[↱][F][↰][EDIT][▼][▼] (i.e., position cursor to the left of the >> symbol) [↰][→NUM][ENTER]

Store the modified program into [F]:

[↰][F]

Next, recall and modify the contents of [FP] by using:

[↱][FP][↰][EDIT][▼][▼] (i.e., position cursor to the left of the >> symbol) [↰][→NUM][ENTER]

Store the modified program into [FP]:

[↰][FP]

Reload the value of 1 into x, by using:

[1] [↰][X].

Now, press [→NEW]. You should get x = 0.1560. Pressing [→NEW] repeatedly will produce a solution that converges to x = -3.0454. (See problem 1(b) in the Numerical Methods textbook).

NOTE: In some instances, when obtaining the symbolic derivative using the HP48G/GX ∂ command , you will need to simplify the resulting expression by using the command [COLCT] available in the menu [↰][SYMBOLIC].

The Secant Method

Use subdirectory SECN. In that subdirectory you will find the menu:

[X1][X2][→NEW][→FP][F][FP]

The next menu is obtained by pressing [NXT]:

[X][EQ][PPAR] [ZPAR] [ ] [ ]

X, EQ, PPAR and ZPAR are used for plotting. The function [F] is a program, generated by using [↰][DEF], while [EQ] contains the same algebraic expression as used in [F]. The program [→FP] calculates the derivative of [F] using the secant method. Current values of the derivative are stored in variable [FP]. The method requires you to provide two initial values [X1] and [X2], where x_2 is very close to x_1. The program used to achieve convergence is [→NEW], which calls program [→FP].

The current examples has the same function as in the interval-halving example, i.e.,

f = `<< → x 'x^3-2*x^2-2*x+1'>>`, EQ = `'x^3-2*x^2-2*x+1'`.

Graphics solution

The procedure is exactly the same as that shown for the Interval-halving method. Therefore, the graphics solution will not be presented here. The solutions, obtained before, are: -1, 0.3819, and 2.6180.

Numerical solution

Enter initial values for x_1 and x_2, then press [→NEW] until the solution converges. For example, set $x_1 = 0$, $x_2 = 0.1$, i.e.,

[0][↰][X1] [.][1][↰][X2]

Pressing [→NEW] repeatedly produces values of x = 0.4566, 0.3741, 0.3818, 0.3819, 0.3819, 0.3819. The solution converges to x = 0.3189. Enter that value in level 1 of the stack and press [F]. The result is f(-0.6180) = 0.00020, i.e., the error is of the order of 2E-4.
Now, enter a value of x_1 = -2.0, x_2 = -1.9, and press [→NEW] repeatedly to prove that the solution converges to x = -1, an exact solution, i.e., error = 0.0, in about 10 iterations.
To find the third solution use, for example, x_1 = 5.0, x_2 = 5.1. Convergence to x = 2.6180 is achieved in about 8 iterations.

Solving other problems

For the secant method you only need to define f(x), by entering *'f(x) = expression containing x'* in level 1 of the stack and using [↰][DEF]. The procedure for the solution is as outlined above.

Polynomial Approximation and Interpolation

Use sub-directory PA&I under sub-directory NUMM. The following sub-directories are available:

POLN:	for polynomial evaluation and deflation.
DFPL:	direct-fit polynomials
LGNP:	Lagrange polynomials
TABLS:	for generating difference tables
NF&B:	Newton forward- and backward-difference polynomials
CFIT:	for least-squares approximation
LSPL:	for least-squares polynomial fitting

Polynomial evaluation and deflation

Use sub-directory POLN. You will find the following variables:

[INF0][A][→PX][↑DFP][N][B]

Pressing [NXT] you will get the following menu:

[QX][NSTM][REVV][V→L][L→V][ ]

Press [NXT] to get to main menu. Press [INF0] to get an explanation of the sub-directory's operation. The instructions indicate that you should store the vector **a** = [a_n a_{n-1} a_{n-2} ... a_3 a_2 a_1 a_0], containing the coefficients of the polynomial, $P_n(x) = a_nx^n + a_{n-1}x^{n-1} + ... + a_2x^2 + a_1x + a_0$, into variable [A], by using [↰][A]. Next, enter a value of x in level 1 of the display, and, finally, press the [→PX] key to evaluate the polynomial P_n at x, as well as the deflated polynomial, Q_{n-1} at x. The program also provides the coefficients of the deflated polynomial, $P'_n(x)= Q_{n-1}(x)$, as a vector.
As an example, store the vector [1 -15 85 -225 274 -120] into [A]. These values correspond to the coefficients of the polynomial, $P_5(x) = x^5-15x^4+85x^3-225x^2+274x-120$. Enter the value 2.5 in the display, and press [→PX]. The results are:

```
4:                x: 2.5
3:        P5(x): -1.40625
2:           Q4(x): .5625
1: Q4-Coeff:[ 1 -12.5 53.75 -90.625 47.4375]
```

These results are interpreted as follows:

$P_5(2.5) = -1.40625$; $P'_5(2.5) = Q_4(2.5) = 0.5625$, and $Q_4(x) = x^4 -12.5x^3 +53.75x^2 -90.625x + 47.4375$.

Deflating is accomplished when we know one of the roots of the polynomial. (Deflating is basically synthetic division). For example, with the same vector of coefficients as before, enter a value of x = 2 in the display, and press [→PX]. The results are:

```
4:                  x: 2
3:              P5(x): 0
2:            Q4(x): -6
1: Q4-Coeff:[ 1 -13 59 -107 60]
```

These results are interpreted as follows:

$P_5(2) = 0$ (i.e., x=2 is a root of $P_5(x)$); $P'_5(2) = Q_4(2) = -6$, and $Q_4(x) = x^4 -13x^3 + 59x^2 -107x + 60$.

Pressing [↑DFT] will take you to sub-directory DFPL.

Direct-fit Polynomial

Use sub-directory DFPL. You will find the following variables:

[INF0][X][F][N][→GET][↑POL]

Pressing [NXT] you will get the following menu:

[CHK][GCFF][CRMT][ ][ ][ ]

Press [NXT] to get to main menu. Press [INF0] to get an explanation of the sub-directory's operation. For direct-fit polynomials you are required to store lists of values of x and f(x), these are the lists {x} and {f}, as well as the order of the polynomial you want to fit to the data, i.e., n. If m is the size of the list, n should be chosen so that $1 \leq n < m$. Press [→GET] to get the coefficients of the fitted polynomial.
For example, store {3.4 3.5 3.6 } in [X], and {0.294118 0.285714 0.277778} in [F]. Next, store a value of 5 into [N], and press [→GET]. As a result, you will get a message indicating that you should "Use $1 \leq n < 3$". Store a value of 2 into [N], and press [→GET]. The result is now the vector:

[2.34E-2 -.2455 .858314],

or the polynomial: $P_2(x) = 0.0234x2-0.2455x+0.85831$.

To evaluate the fitted polynomial for a given value of x, you can press the key [↑POL], which will take you into sub-directory POLN (see above). Next, you can store the vector currently in the display into [A], enter the value of x, and press [→PX] to get the evaluated polynomial. For this case, for example, you can check that $P_2(3.55) = 0.28168$. Also, check that $P_2(3.4) =0.29411799998$, $P_2(3.5) = 0.285713999998$, and $P_2(3.6) = 0.0.2777779999998$.

Lagrange polynomials

Use sub-directory LGNP. You will find the following variables:

[INF0][XL][FL][N][→PX][GETC]

Pressing [NXT] you will get the following menu:

[GETF][PRDT][CHK][ ][ ][ ]

Press [NXT] to get to main menu. Press [INF0] to get an explanation of the sub-directory's operation. The instructions require that you store lists with values of {x} and {f} into variables [X] and [F], respectively. You also need to enter the degree of the polynomial, n, into variable [N]. . If m is the size of the lists {x} and {f}, n should be chosen so that $1 \leq n < m$. To evaluate the polynomial, enter the value of x in level 1 of the stack and press [→PX].

If the lists used for the polynomial evaluation are $xL = \{x_1\ x_2\ ...\ x_m\}$, and $fL = \{f_1\ f_2\ ...\ f_m\}$, the polynomial evaluated can be written as:

$$P_n(x) = C_1(x)f_1 + C_2(x)f_2 + ... + C_{n+1}f_{n+1},$$

where

$$C_j(x) = N_j(x)/D_j(x), \quad (j = 1,2, ..., n+1),$$

and

For example, store the lists xL = { 3.4 3.5 3.55 3.65} and fL = { 0.294118 0.285714 0.28169 0.273973}. Store n = 5, enter a value of 3.55 in level 1, and press [→PX]. You will get, as a result, a message requesting you to "`Use 1 ≤ n <  4`". Enter a value of n = 3, enter 2.5 in level 1, and press [→PX] again. The results are now: `x:` 2.5, P3(x): 0.397414, interpreted as: $P_3(2.5) = 0.387414$.

Tables of differences

Use sub-directory TABLS. You will find the following variables:

[INFO][FL][N][→TBL][→ΔF][→∇F]

Pressing [NXT] you will get the following menu:

[ΔF][∇F][F][M][→ZLIS][CRMT]

Press [NXT] to get to main menu. Press [INFO] to get an explanation of the sub-directory's operation. The instructions indicate that you should store the list {fL}, and the level of differences in the table, n, into variables [fL] and [N], respectively. The next step is to generate a table of differences by using [→TBL]. Once this table has been generated (it is stored in variable [F] and shown in level 1 of the stack), you can use either [→ΔF] to generate the vector $\Delta F = [f_o\ \Delta f_o\ \Delta^2 f_o ...\ \Delta^n f_o]$ (for forward-difference polynomials), or the vector $\nabla F = [f_o\ \nabla f_o\ \nabla^2 f_o ...\ \nabla^n f_o]$ (for backward-difference polynomials).

For example, enter the list fL = {0.333333 0.322581 0.,312500 0.303030 0.294118 0.285714 0.277778 0.270270 0.263158 0.256410 0.250000} into the variable [FL]. Enter n = 5 into [N]. Then, press [→TBL], to get the following table:

	1	2	3	4	5	6
1	*0.333333*	*-0.010752*	*0.000671*	*-0.000060*	*0.000007*	*-0.000004*
2	0.322581	-0.010081	0.000611	-0.000053	0.000003	0.000007
3	0.312500	-0.009470	0.000558	-0.000050	0.000010	-0.000010
4	0.303030	-0.008912	0.000508	-0.000040	0.000000	0.000008
5	0.294118	-0.008404	0.000468	-0.000040	0.000008	-0.000008
6	0.285714	-0.007936	0.000428	-0.000032	0.000000	*0.000006*
7	0.277778	-0.007508	0.000396	-0.000032	*0.000006*	**0**
8	0.270270	-0.007112	0.000364	*-0.000026*	**0**	**0**
9	0.263158	-0.006748	*0.000338*	**0**	**0**	**0**
10	0.256410	*-0.006410*	**0**	**0**	**0**	**0**
11	*0.250000*	**0**	**0**	**0**	**0**	**0**

This table contains the same information as that in Table 4.5, page 131, of the textbook for f(x) and the differences, except for the alignment of the table entries (Table 4.5 is reproduced below). The top row of the table above is recognized as the values $\Delta F = [f_o\ \Delta f_o\ \Delta^2 f_o \ldots \Delta^n f_o]$, while the values at the bottom of each column, and above the bold-faced zeros, represent $\nabla F = [f_o\ \nabla f_o\ \nabla^2 f_o \ldots \nabla^n f_o]$. To get those vectors, try pressing [→ΔF] and [→∇F]. The results are:

ΔF = [0.333333 -0.010752 0.000671 -0.000060 0.000007 -0.000004]
∇F = [0.250000 -0.006410 0.000338 -0.000026 0.000006 0.000006]

Newton Forward- and Backward-Difference Polynomials

Use sub-directory NF&BP. You will find the following variables:

[INF0][S][FL][N][→GFP][→GBP]

Pressing [NXT] you will get the following menu:

[CSI][ ][ ][ ][ ][ ]

Press [NXT] to get to main menu. Press [INF0] to get an explanation of the sub-directory's operation. The instructions indicate that you should store a value of s into [S], a list of values {fL} into [FL], and the order of the polynomial to be calculated, n, into [N]. Pressing [→GFP] will calculate the Newton forward-difference polynomial according to equation (4.67) in page 134 of the textbook. Pressing [→GPB] will calculate the Newton backward-difference polynomial according to equation (4.80) in page 137 of the textbook.
The formulas are based on tables of values of x and f(x) such that the values of x are equally spaced by an amount h, i.e., $xL = \{x_a\ \ x_a+h\ x_a+2h\ \ldots\ x_a+(m-1)h\}$. If x is the value at which we want to evaluate the polynomial, then,

$$s = (x-x_o)/h$$

where x_o represents the first value in the list xL if using the forward-difference polynomial, or the last element of the list xL if using the backward-difference polynomial. (We do not actually use the xL list in the calculations, however).
For example, using the data of table 4.5, page 131 (reproduced below), of the textbook, we could select $x_o = 3.40$, and use a Newton forward-difference polynomial of order n = 3 evaluated at x = 3.44. From the table, it follows that h = 0.1, therefore, $s = (x-x_o)/h = (3.44-3.40)/0.10=0.4$. For a polynomial of order n we need the to have at least 4 elements. For $x_o = 3.40$, and a Newton forward-difference polynomial, therefore, we can select, fL = { 0.294118 0.285714 0.277778 0.270270 }. For this example, use the following keystrokes:

0.4 [↰][S]	Enter value of s
{ 0.294118 0.285714 0.277778 0.270270 } [↰][FL]	Enter list xL
3 [↰][N]	Enter polynomial order n = 3
[→GFP]	Calculate forward-difference polynomial

The result is 0.29069768, i.e., $P_3(x=3.44) = P(s=0.4) = 0.29069768$.

Difference table for *f(x)= 1/x*

x	f(x)	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$	$\Delta^5 f(x)$
3.0	0.333333					
		-0.010752				
3.1	0.322581		0.000671			
		-0.010081		-0.000060		
3.2	0.312500		0.000611		0.000007	
		-0.009470		-0.000053		-0.000004
3.3	0.303030		0.000558		0.000003	
		-0.008912		-0.000050		0.000007
3.4	0.294118		0.000508		0.000010	
		-0.008404		-0.000040		-0.000010
3.5	0.285714		0.000468		0.000000	
		-0.007936		-0.000040		0.000008
3.6	0.277778		0.000428		0.000008	
		-0.007508		-0.000032		-0.000008
3.7	0.270270		0.000396		0.000000	
		-0.007112		-0.000032		0.000006
3.8	0.263158		0.000364		0.000006	
		-0.006748		-0.000026		
3.9	0.256410		0.000338			
		-0.006410				
4.0	0.250000					

As a second example, still using the data shown in the Table above, we could select x_o = 3.50, and use a Newton backward-difference polynomial of order n = 3 evaluated at x = 3.44. For h = 0.1, s = $(x-x_o)/h$ = (3.44-3.50)/0.10 = -0.6. For x_o = 3.50, and a Newton backward-difference polynomial, we can select fL = { 0.312500 0.303030 0.294118 0.285714 }. For this example, use the following keystrokes:

-0.6 [↰][S]	Enter value of s
{ 0.312500 0.303030 0.294118 0.285714} [↰][FL]	Enter list xL
3 [↰][N]	Enter polynomial order n = 3
[→GBP]	Calculate forward-difference polynomial

The result is 0.29069824, i.e., $P_3(x=3.44) = P_3\ (s=-0.6) = 0.290698248$.

Least-Square Polynomial Fitting

Suppose that n data points (x_i,y_i) can be fitted to a polynomial relationship of the form

$$y = a_o + a_1x + a_2 x^2 + \ldots + a_n x^n,$$

Where *n* is the order of the polynomial. The parameters $a_{o,}$ $a_{1.}$ $a_2 \ldots a_n$, are the solution to a set of linear equations given, in matrix form, by $\mathbf{M}\cdot\mathbf{b} = \mathbf{c}$, where the vectors **b** and **c**, and the matrix **M**, are defined as follows:

$$\mathbf{b} = [a_o\ a_1\ a_2 \ldots a_n]^T ; \qquad \mathbf{c} = [\Sigma Y\ \ \Sigma xY\ \ \Sigma x^2Y \ldots \Sigma x^nY]^T$$

Subdirectory LSPL in your HP48G calculator contains a program [→SOL] that calculates the parameters in the vector **b** when given the values of x and Y.

To perform a polynomial regression using the programs in this sub-directory use this procedure:

1. Store the values of x and y lists, and the value of n. [→DAT] to set up the data matrix necessary to solve for the vector of coefficients $\mathbf{b} = [a_o, a_1, a_2,\ldots, a_n]$.
2. Press [→SOL] to solve for **b**. The display will show the values of the vector of coefficients **b** $[a_o, a_1, a_2,\ldots, a_n]$., the value of n, and the correlation coefficient r. (Note: the time required for the calculator to perform the regression increases with the value of n, i.e., a regression with n = 5 will take much longer than a regression with n = 2).
3. Press [→PLT] to plot residual errors, e vs. y.
4. Enter a value of x and press [→Y] to get $y = a_o + a_1x + a_2x^2 + \ldots +a_nx^n$.
5. Press [NXT] for the next menu.
6. Press [N] to see the number of data points, N.
7. Press [B] to see the vector **b**.
8. Press [R] to see the correlation coefficient for the multiple linear regression, r.
9. Press [YBAR] to see the mean value of Y, $\overline{Y}$.
10. Press [YH] to see the vector **y**. This vector contains the regression values corresponding to the original values of the independent variables, i.e., $y_i = a_o + a_1x_i + a_2x_i^2 + \ldots +a_nx_i^n$, for i = 1, 2, ..., N.
11. Press [SSE] to see the error sum of squares, $SSE = \Sigma(Y_i - y_i)$.
12. Press [SST] to see the total sum of squares, $SST = \Sigma(Y_i - \overline{y})$.
13. Press [SE] to see the standard error of estimate, s_e. This value is an estimator of the standard deviation of the distribution of the independent variable x.

<u>*Example*</u>
Fit a polynomial to the following data:

x	y
0	12.0
1	10.5

2	10.0
3	8.0
4	7.0
5	8.0
6	7.5
7	8.5
8	9.0

Use values of n = 2, 3, 4 and 5. And determine the values of r for each one of them. Then, select the value of n with the best value of r, and determine the coefficients of the polynomial regression equation. Finally, determine the value of y expected if x = 3.5.

Follow this procedure:
1) Store the values of x and y as lists into the corresponding variables. For example, for x, use the following keystrokes: [↰][{}][0][SPC][1] [SPC][2][SPC][3][SPC][4][SPC][5]...[8][ENTER][↰][x]
2) Store the value of n, n = 2. Use: [2][↰][n]
3) Press [→SOL]. The correlation coefficient is, r = 0.96056.
4) Store a new value of n, n = 3. Use: [3][↰][N]. The result is r = 0.96107.
5) Repeat step 4) for n = 4 and 5. Verify, that r(n=4) = 0.97051, and r(n=5) = 0.97051.

The results indicate that r reaches a maximum value of 0.9751 for n = 4. Using n = 5 does not improve in the value of r. To verify what is happening, we show below the values of the **b** vectors and the correlation coefficients, r, for each value of n:

n = 2,	**b** = [12.18 -1.85 0.18]	r = 0.96056
n = 3,	**b** = [12.12 -1.71 0.14 0.004]	r = 0.96107
n = 4,	**b** = [11.93 -0.71 -0.50 0.13 -0.01]	r = 0.97051
n = 5,	**b** = [11.93 -0.71 -0.50 0.13 -0.01 -1.32×10-12≈0]	r = 0.97051

The three first coefficients of the case n = 3 are very similar to the coefficients of the case n = 2, with the last coefficient of the case n = 3 being very close to zero. This explains the fact that the correlation coefficients for those two cases are very similar. The correlation coefficient improves a little when we use n = 4 or 5 (r is the same for this two cases), over either n = 2 or 3. Notice that the first four coefficients of the case n = 5 are the same as those of the case n = 4, with the last coefficient of the case n = 5 being almost zero. In other words, there is no improvement in the regression by using n = 5 over n = 4. There was a slight improvement on the regression going from n = 2 to n = 4. Let's select the case n =4 as the case with the best correlation. Repeat step 4 for n = 4.

6) Store the value of n, n = 4. Use: [4][↰][N]
7) Press [→SOL].
8) Press [3][.][5][ENTER][→Y], the result is y: 7.7695.

Additional information:

Press [→PLT] to see residual errors vs. y. Press [STATL] to see the zero axis for errors. Press [CANCL] to return to normal display.

Press [NXT][NN] to see the number of points, N = 9.

Press [B] to get the vector **b** = [11.93 -0.71 -0.50 0.13 -0.01]. This means that our regression equation can be written as:

$$y= 11.93 -0.71 x -0.50 x^2 + 0.013 x^3 -0.01x^4.$$

Press [YBAR] to see $\overline{Y}$ = 8.944.

Press [YH] to see the list y = {11.93 10.84 9.43 8.21 7.48 7.36 7.78 8.48 8.99 }.

Press [SSE] to see the error sum of squares, SSE = $\Sigma(Y_i - y_i)$ = 1.20.

Press [NXT][SST] to see the total sum of squares, SST = $\Sigma(y_i - \bar{y})$ = 20.72

Press [SE] to see the standard error of estimate, s_e = 0.41.

Press [M] to see the matrix **M**.

Press [C] to see the vector **c**.

Multivariate Polynomial Approximation

Two examples are shown following: the linear bivariate polynomial: $z = a + bx + cy$, and the quadratic bivariate polynomial, $z = a + bx + cy + dx^2 + ey^2 + fxy$. In both cases the solution for the coefficients (a, b, c, etc.) requires solving a matrix equation of the form $\mathbf{A\xi} = \mathbf{b}$. Methods for the solution of the matrix equation were discussed elsewhere. The main problem will be to build the matrix **A** and the vector **b**. This could be easily approached, however, by using lists to store the values of x and y, and operating with those lists.

For example, using :

x	1150	1150	1150	1200	1200	1200	1250	1250	1250
y	800	1000	1200	800	1000	1200	800	1000	1200
z	1380.4	1500.2	1614.5	1377.7	1499	1613.6	1375.2	1497.1	1612.6

Where x represents P, y represents T, and z represents C_p.

Create a sub-directory to store this problem, and within that directory store the variables:

x = { 1150 1150 1150 1200 1200 1200 1250 1250 1250 }
y = { 800 1000 1200 800 1000 1200 800 1000 1200 }
z = { 1380.4 1500.2 1614.5 1377.7 1499 1613.6 1375.2 1497.1 1612.6 }

Then, use operations with lists to calculate the coefficients of the system of linear equations that result from the least-square method for this case:

For example, to calculate N, press:
[X][PRG][LIST][ELEM][SIZE]. The result is N = 9. Press [VAR] to recover variables.

To calculate, for example, Σx_i, use:
[X][MTH][LIST][ΣLIST]. The result is Σx_i = 10800.

To calculate, $\Sigma x_i y_i$, use:
[VAR][X][Y][×][MTH][LIST][ΣLIST]. The result is $\Sigma x_i y_i$ = 10800000.

To calculate, for example, Σx_i^4, use:

[VAR][X][4][y^x][MTH][LIST][ΣLIST].]. The result is Σx_i^4 = 1.87920375E13.

As a final example, to calculate , $\Sigma x_i^3 y_i$, use:

[VAR][X] [3][y^x][Y][×][MTH][LIST][ΣLIST]. The result is $\Sigma x_i^3 y_i = 1.5606E13$.

Many of the 36 coefficients required in equations (4.132) are repeated, therefore, the number of coefficients to be calculated is 21. The resulting matrix, as shown in page 153, is symmetric.

Data for exercises
Data for application of the methods discussed in this guidebook is available in sub-directory DATA.

Numerical Integration in the HP48G/GX calculator

Use sub-directory NMIN under sub-directory NUMM. The following sub-directories are available:
NFDP: for numerical integration using Newton forward-difference polynomials.
NECO: for numerical integration using Newton-Cotes formulas
GAUS: for numerical integration using Gaussian Quadrature

Numerical integration using Newton forward-difference polynomials

Use sub-directory NFDP. You will find the following variables:

[INF0][H][F][→TRP][→S13][→S38]

Pressing [NXT] you will get the following menu:

[GETF][CS13][CH13][CS38][CH38][ ]

Press [NXT] to get to main menu. Press [INF0] to get an explanation of the sub-directory's operation. The instructions indicate that you should store the constant x-increment, h, and a list of values of f(x), {f} = {f_1 f_2 f_3 ... f_n} corresponding to {x_o, $x_1 = x_0+h$, $x_2 = x_0+2h$, ..., $x_n = x_0+nh$}. The numerical integrals can be calculated using [→TRP] for the trapezoidal rule, [→S13] for the Simpson's 1/3 rule, or [→S38] for the Simpson's 3/8 rule. If you want to calculate the values of f given a function f(x) you can use [GETF] to get the list {f}. The following examples illustrate the use of the different programs.

Trapezoidal rule

Calculate the integral of the function f(x) = 1/x, 1 < x < 2, using the trapezoidal rule. Within sub-directory NFDP press [VAR][NXT][GETF] to get into directory GETF with the purpose of generating the list of values of f(x) to be used in the integration. The variables to be entered are x0 = 1, xN = 2, h = 0.1 (arbitrarily selected). Use the following keystrokes:

[1][↰][X0] [2][↰][XN] [.][1][↰][H]

Next, we need to define the function f(x) = 1/x by using the following keystrokes:

['] [α][↰][F] [↰][()] [α][↰][X] [▶] [↰][=] [1][÷] [α][↰][X] [ENTER] [↰][DEF]

Pressing [↱][F] will show the program that defines the function as: `<< → x '1/x' >>`. Press [DEL] to clear the stack.
Press [→GTF] to get the list of values of f. The results, for this particular case are:

```
3: x: { 1 1.1 1.2 1.3...
2: f: { 1 .909090909...
1:                h: .1
```

These results show the value of h in level 1, the f list in level 2, and the corresponding x list in level 3. We need to store the values of h and f in the upper directory NFDP to calculate the integrals. To get to that directory press the key labeled [↑UP]. Next, store the values of h and f as follows:

[↰][H][↰][F]

The values of x are not needed. They are only shown to verify the integration limits. You can erase the x list by pressing [DEL]. Next, press [→TRP] to calculate the numerical integral using the trapezoidal rule. The result is 0.69377140317. Check this value against the exact value ln(2) = 0.69314718056. The absolute error is error is

Absolute Error = |Integral - Exact Value| = |0.69377140317-0.69314718056 | = 0.00062422261.

The relative error is,

Relative Error = (Absolute Error/Exact Value)·100% = (0.00062422261/0.69314718056)·100% = 0.09%

Simpson's 1/3 Rule

The Simpson's 1/3 rule requires that the number of data points in the list f be odd (or the number of intervals, n, between x0 and xN be even). The size of the current f list can be determined by using the following keystrokes:

[F][PRG][LIST][ELEM][SIZE]

The result is 11, an odd number, therefore, we can use the Simpson's 1/3 rule to calculate the integral with our current data by pressing [VAR][→S13]. The result is 0.69315023069. With Absolute Error = 0.00000305013, and Relative Error = 0.00044%.

To determine the value of h corresponding to a value of n when splitting the interval between x_0 and x_n, use

$$h = (x_n - x_0)/n.$$

Recall that n is the number of intervals for values of x between x_0 and x_n. For the Simpson's 1/3 rule, n must be even. For example, if we want to get 20 intervals (i.e., n = 20) between x0 = 1 and xN = 2 for the function f(x) = 1/x, we calculate h to be

$$h = (x_n - x_0)/n = (2 - 1)/20 = 0.05.$$

To obtain the f list, use the following:

[NXT][GETF] [1][↰][X0] [2][↰][XN] [.][0][5][↰][H]

The function f(x) has been defined before, therefore, we go directly to calculating {f} by pressing [→GTF]. The results are now:

```
3: x: { 1 1.1 1.2 1.3…
2: f: { 1 .9523809523…
1:              h: .1
```

Press [↑UP] to get back to NFDP, and store the values of h and f:

[↰][H][↰][F]

Press [DEL] to clear up the display. Then press [→S13] to calculate the numerical integral using Simpson's 1/3 rule. The result is 0.693147374667.

Note: when using [→S13] if the number of intervals is not even a warning will be shown and the calculation stopped.

Simpson's 3/8 rule

The Simpson's 1/3 rule requires that the number of intervals, n, between x0 and xN be a multiple of three. For example, we can use n = 9 in the integral calculated above by choosing h as

$h = (x_n - x_0)/n = (2 - 1)/9 = 0.111111111111$

Within sub-directory NFDP press [VAR][NXT][GETF] and store h = 0.111111111111 into [H]. Then, press [→GTF]. The results are now:

```
3: x: { 1 1.1111111111...
2: f: { 1 .90000000000...
1:      h: .11111111111
```

Press [↑UP] to get back to NFDP, and store the values of h and f:

[↰][H] [↰][F]

Press [DEL] to clear up the display. Then press [→S38] to calculate the numerical integral using Simpson's 3/8 rule. The result is 0.693157302259.

Note: when using [→S38] if the number of intervals is not a multiple of three a warning will be shown and the calculation stopped.

Try pressing [→S13] with the current data. You will get a message box with the warning "Simpson 1/3 Rule: Number of intervals must be even." Press [OK] to clear the message.

Press [↰][UP] to get to the upper directory.

Newton-Cotes Formulas

Use sub-directory NECO. You will find the following variables:

[INF0][N][H][F][→INT][GETF]

Pressing [NXT] you will get the following menu:

[α][β][ ][ ][ ][ ]

Press [NXT] to get to main menu. Press [INF0] to get an explanation of the sub-directory's operation. The instructions indicate that you should store the order of the polynomial (also the number of intervals), n, with $1 \le n \le 7$, the constant x-increment, h, and a list of values of f(x), $\{f\} = \{f_1\ f_2\ f_3\ \ldots\ f_n\}$ corresponding to $\{x_o,\ x_1 = x_0+h,\ x_2 = x_0+2h,\ \ldots,\ x_n = x_0+nh\}$. The numerical integrals are calculated using [→INT]. If you want to calculate the values of f given a function f(x) you can use [GETF] to get the list {f}. The following examples illustrate the use of this sub-directory.
Notice that n and h are related by the formula

$$h = (x_n - x_0)/n$$

For example, if we want to integrate, say f(x) = exp(x), between $x_0 = 0$ and $x_n = 1$, using n = 5, then

$$h = (x_n - x_0)/n = (1 - 0)/5 = 0.2$$

In sub-directory NECO, press [GETF] to get into subdirectory GETF, which will be used to generate the list {f } corresponding to this problem. Then use the following keystrokes:

[0][↰][X0] [1][↰][XN] [5][↰][N]

Next, we need to define the function f(x) = exp(x) by using the following keystrokes:

['] [α][↰][F] [↰][()] [α][↰][X] [▶] [↰][=][↰][e^x] [α][↰][X] [ENTER] [↰][DEF]

Pressing [↱][F] will show the program that defines the function as: `<< → x 'EXP(x)' >>`. Press [DEL] to clear the stack.
Press [→GTF] to get the list of values of f. The results, for this particular case are:

```
4: x: { 1 .2 .4 .6 .8…
3: f: { 1 1.221402758…
2:                h: .1
1:                 n: 5
```

These results show the value of n in level 1, h in level 2, the f list in level 3, and the corresponding x list in level 4. We need to store the values of n, h and f in the upper directory NFDP to calculate the integral. To get to that directory press the key labeled [↑UP]. Next, store the values of n, h and f as follows:

[↰][N] [↰][H] [↰][F]

The values of x are not needed. They are only shown to verify the integration limits. You can erase the x list by pressing [DEL]. Next, press [→INT] to calculate the numerical integral using the Newton-Cotes formula (for n = 5, in this case). The result is 1.71828231299. Compared to the exact result, *e-1* = 1.71828182846, the absolute error is 0.00000048453, and the relative error is 0.0000281%.

Press [↰][UP] to get to the upper directory.

Romberg Integration

This algorithm basically uses the trapezoidal rule to calculate the integral for a given value of h, I(h), and for intervals corresponding to h/2, I(h/2), then extrapolates a value of the integral as

$$I(extrapolated) = I(h/2) + (I(h/2)-I(h))/3.$$

Therefore, to calculate Romberg integration use the [→TRP] in sub-directory NFDP to calculate I(h) and I(h/3). Then, use the formula shown above to calculate I(extrapolated).

For example, above we used the trapezoidal rule to calculate the integral of f(x) = 1/x, between $x_0 = 1$ and $x_n = 2$, with h = 0.1. The result was I(h) = 0.69377140317. We can then calculate I(h/2) by using h = 0.05, in which case I(h/2) = 0.69330338179. The Romberg integration result will, therefore, be

$$I(extrapolated) = I(h/2) + (I(h/2)-I(h))/3 = 0.69330338179 + (0.69330338179-0.69377140317)/3,$$

or

I(extrapolated) = 0.693147374663.

The absolute error is 0.000000194103, and the relative error is 0.000028%.

Gaussian Quadrature

Use sub-directory GAUS. You will find the following variables:

[INF0][A][B][F][→GS2][→GS3]

Pressing [NXT] you will get the following menu:

[→GS4][→TC][C][M][→MC][F]

Press [NXT] to get to main menu. Press [INF0] to get an explanation of the sub-directory's operation. The instructions indicate that you should store the limits of integration $a = x_o$, and $b = x_n$, and define the function f(x) using the [DEF] key. The numerical integrals are calculated using [→GS2], [→GS3], and [NXT][→GS4], for the Gaussian quadrature formulas corresponding to n = 2, 3, and 4, respectively. If you want to calculate the values of f given a function f(x) you can use [GETF] to get the list {f}. The following example illustrates the use of this sub-directory.
To calculate the integral of f(x) = 1/x, between a = 1 and b = 2, store the values of a and b using:
[1][↰][A] [2][↰][B].

Then, define f(x) = 1/x, by using:

['] [α][↰][F] [↰][()] [α][↰][X] [▶] [↰][=] [1][÷] [α][↰][X] [ENTER] [↰][DEF]

Pressing [↱][F] will show the program that defines the function as: `<< → x '1/x' >>`. Press [DEL] to clear the stack.
Press [→GS2] to get the 2nd-order Gaussian quadrature integral, the result is 0.692307692305.
Press [→GS3] to get the 3rd-order Gaussian quadrature integral, the result is 0.69312169312.
Press [NXT][→GS4] to get the 4th-order Gaussian quadrature integral, the result is 0.693146417445.
The exact result is ln(2) = 0.69314718056.

About The Author

Gilberto E. Urroz is an Associate Professor of Civil and Environmental Engineering and a researcher at the Utah Water Research Laboratory, both at Utah State University, in Logan, Utah. He has been a teacher of engineering disciplines for more than 15 years both in his native Nicaragua and in the United States. His teaching experience includes courses on introductory physics, engineering mechanics, probability and statistics for engineers, computer programming, fluid mechanics, hydraulics, and numerical methods. His research interests include mathematical and numerical modeling of fluid systems, hydraulic structures, and erosion control applications.

Dr. Urroz is an expert on the HP 48 G and HP 49 G series calculator and has written several books on applications of these computing devices to disciplines such as engineering mechanics, hydraulics, and science and engineering mathematics. His personal interests include reading, music, opera, theater, and taijiquan.

ABOUT GREATUNPUBLISHED.COM

Greatunpublished.com is a website that exists to serve writers and readers, and remove some of the commercial barriers between them. When you purchase a greatunpublished.com title, whether you receive it in electronic form or in a paperback volume, you can be assured that the author is receiving a majority of the post-production revenue. Writers who join greatunpublished.com support the site and its marketing efforts with a per-title fee, and a portion of the site's share of profits are channeled into literacy programs.

So by purchasing this title from greatunpublished.com you are helping to revolutionize the publishing industry for the benefit of writers and readers. And for this we thank you

This book includes applications of the HP 48 G/G+/GX calculator to analytical and numerical methods commonly found in science and engineering mathematics: matrices, linear and non-linear equations, differentiation, integration, data fitting, and ordinary differential equations.

Gilberto E. Urroz is an Associate Professor of Civil and Environmental Engineering and a researcher at the Utah Water Research Laboratory, both at Utah State University, in Logan, Utah. He has been a teacher of engineering disciplines for more than 15 years both in his native Nicaragua and in the United States. His teaching experience includes courses on introductory physics, engineering mechanics, probability and statistics for engineers, computer programming, fluid mechanics, hydraulics, and numerical methods. His research interests include mathematical and numerical modeling of fluid systems, hydraulic structures, and erosion control applications.

ISBN 1-58898-042-1

90000>

9 781588 980427

www.greatunpublished.com