

HP 50g RPL Reference

Programmer's guide · 774 commands grouped by function · from the HP 50g Advanced User's Reference

A drawing of the keyboard sits under the navigation bar. Page 1 explains the User RPL language: objects, locals, tests, branching, loops, lists, flags, errors, input/output, menus, debugging — then maps every soft menu, its pages and submenus, and the keys that open them. The following pages index every command in the AUR Command & Function Reference (HP 48gII / 49g+ / 50g) by category and subgroup. Hover any command for its description and type.

Type: ● Command ● Function (usable in algebraics) ● Analytic (has derivative) ● Operation *italic* → *X* alias / same as X

Sources: HP 50g Advanced User's Reference Manual, Ed. 2 (HP, 2009) — ch. 1 RPL Programming, ch. 3 Full Command and Function Reference, app. C–E · HP 48SX Programmer's Reference Manual · J. Dossey, Advanced User-RPL Guide. Commands belonging to more than one topic are listed in each.



The keyboard

Every key does three things. Its own legend is printed on the key; the function you get after the **white left-shift**  is printed in white above-left, after the **orange right-shift**  in orange above-right, and the **yellow letter** at the key's corner is what **ALPHA** types. Press a shift once and it applies to the next key; the annunciator at the top of the display shows which is armed.

 then keywhite function – MTH, PRG, CALC, MATRICES, CONVERT...
 then keyorange function – CAT, CHARS, EQW, UNITS, BASE, TRIG...
ALPHA key · ALPHA ALPHAone letter · lock alpha (ALPHA again to leave)
ALPHA  key · ALPHA  keylowercase letter · the symbol printed on the bezel
hold  + key · hold  + keya few extras:  +MODE MODES menu,  +NXT last menu,  +VAR HOME,  +EVAL CHARS soft menu,  +7 solver soft menu,  +9 TIME soft menu,  +ENTER exact/approx toggle
F1 – F6the soft keys: whatever the six labels at the bottom of the display say – see Soft Menus
◀ ▶ ▲ ▼cursor; ▲ on the stack opens the interactive stack, ▼ edits level 1, ▶ shows PICT, ▶ swaps

Drawn from the emulator's key table (x50ng) and the AUR key-code table; the F-key labels Y= WIN GRAPH 2D/3D TBLSET TABLE are the six plot input forms.

RPL Programmer's Reference

User RPL language constructs — syntax, semantics, idioms

Program Objects

« obj₁ obj₂ ... obj_n »

program = sequence of objects

« → a b 'ABS(a-b)' »

with a local-variable structure

« ! 2 / » 'HALFFACT' STO

store under a name → run by name

OBJECT INSIDE A PROGRAM	ACTION WHEN REACHED
Command / function	Executed
Number, string, list, program, algebraic, complex, array	Put on the stack
Global name, unquoted <code>NAME</code>	Evaluated: program → run; name → evaluated; directory → made current; other → put on stack
Global name, quoted <code>'NAME'</code>	Name put on the stack
Local name, unquoted	Contents put on stack (not evaluated)
Local name, quoted	Name put on the stack
Comments <code>@ text @</code> or <code>@ to end of line</code> — stripped when the program is parsed (except inside strings). Line breaks are discarded on ENTER.	
Run: VAR-menu key · type name + ENTER · <code>'NAME' EVAL</code> · program object + EVAL. Stop: CANCEL (ON). Edit: EDIT / EDITB / VISIT / VISITB / Filer. Program-entry mode shows the PRG annunciator; command keys are entered, not executed.	

Objects & Delimiters		
SYNTAX	OBJECT	TYPE
3.14 1.E5	Real number	0
5 -12 (no point)	Exact integer (CAS)	28
(1,2) '1+2*i'	Complex	1
"text"	String	2
[1 2 3] [[1 2][3 4]]	Real vector / matrix	3
[(1,2) (3,4)]	Complex array	4
{ 1 "a" X }	List	5
'NAME' / 'x'	Global / local name	6 / 7
« ... »	Program	8
'X^2+1' 'A=B'	Algebraic (expression / equation)	9
#FFh #10d #17o #101b	Binary integer	10
GROB w h hex...	Graphics object	11
:tag:obj	Tagged object	12
1_m/s 2.5_kg	Unit object	13
DIR ... END	Directory	15
[['X' 'Y']]	Symbolic matrix	29
←name	Compiled local name	7
Other TYPE numbers: 14 XLIB name · 16 library · 17 backup · 18 built-in function · 19 built-in command · 20 system binary · 21/22 extended real/complex · 23 linked array · 24 character · 25 code · 26 library data · 27 mini font · 30 font · 31 extended object. VTYPE returns -1 if the variable does not exist.		

Local Variables

```
« → n1 n2 ... nk « program » »  k objects taken from the stack
« → n1 n2 ... nk 'algebraic' »   n1 = deepest, nk = level 1
« → r '4/3*π*r^3' →NUM »        defining procedure runs automatically
```

- Locals exist only while the defining procedure runs; never in the VAR menu; faster than globals; no STO/PURGE needed.
- Available inside **nested** structures within the same procedure — **not** inside programs called by name (pass values on the stack instead).
- Evaluating a local name returns its contents **unevaluated** — follow with `EVAL` if it holds a program or algebraic.
- Compiled locals** `→y`: visible to any program called within the structure; disappear when the structure ends. `« → →y 'IFTE(→y<0,BELOW,ABOVE)' »`
- `'c' INCR` / `'c' DECR` work on local or global counters. FOR/STEP counters are locals.
- `LOCAL` / `UNBIND` — algebraic-mode equivalents.

USER-DEFINED FUNCTIONS

```
« → x y 'SQRT(x^2+y^2)' » 'HYP' STO  then 3 4 HYP or
                                     'HYP(3,4)'
```

A program that is solely a `→` structure with an algebraic body is a UDF: usable in algebraics, takes symbolic arguments, has a derivative. With a program body it still works from stack or algebraic syntax but has no derivative and must return exactly one result. `DEFINE 'HYP(x,y)=SQRT(x^2+y^2)'` builds the same thing.

Tests & Logic			
<pre>a b < 'a<b' → 1. (true) or 0. (false); order: level 2 test level 1</pre>			
TEST	MEANING	NOTES	
<code>== ≠</code>	Equal / not equal	Names & algebraics give an expression <code>'X<5'</code> → use <code>→NUM</code> or a conditional (auto-evaluated)	
<code>< > ≤ ≥</code>	Order	Strings compare alphabetically; lists/programs equal if contents identical	
<code>SAME</code>	Identical objects	No numeric evaluation; binary integers must match wordsize; else like <code>==</code>	
<code>=</code>	Builds an equation	Not a test	
<code>AND OR XOR NOT</code>	Logic	Any nonzero = true; infix in algebraics: <code>'A<B AND C'</code> , prefix <code>'NOT Z≥4'</code>	
<code>TYPE VTYPE LININ</code>	Structure tests	<code>DUP TYPE 3 ==</code> · <code>'NAMES' VTYPE -1 ==</code> (missing) · <code>'H+Y^2' 'H' LININ</code>	
<code>FS? FC? FS?C FC?C</code>	Flag tests	?C variants clear the flag after testing	

Conditionals

```
IF test THEN true-clause END
IF test THEN true-clause ELSE false-clause END
CASE test1 THEN clause1 END
    test2 THEN clause2 END
    ... default-clause END
test « true » IFT
test « true » « false » IFTE  also 'IFTE(test,true,false)'
```

The test-clause is a command sequence (leaves 1/0 on the stack) or an algebraic (auto-evaluated — no `→NUM` needed). THEN removes the test result. CASE runs the first true clause then skips to the final END; the default clause is optional. IFT/IFTE take objects (usually programs) as clauses; IFTE is also an algebraic function.

```
« DUP IF 0 > THEN NEG END »
« → x « IF 'x>0' THEN NEG END » »
« DUP 0 > « NEG » IFT »
« → x 'IFTE(x≠0,SIN(x)/x,1)' »
« → y « CASE y TYPE 2 SAME THEN y 'STR' STO END
        y TYPE 5 SAME THEN y 'LIST' STO END
        "OTHER" END » »
```

Loops

```
start finish START clause NEXT      hidden counter, +1
start finish START clause incr STEP  incr may be negative
start finish FOR i clause NEXT       i is a local variable
start finish FOR i clause incr STEP
DO clause UNTIL test END             test after body → runs ≥ 1×
WHILE test REPEAT clause END         test first → may run 0×
```

- Definite loops always run at least once. NEXT/STEP compare the counter with *finish*: continue while `≤` finish (positive step) or `≥` finish (negative step). STEP's argument may be an algebraic/name (auto-evaluated). The FOR counter is purged on exit.
- END (DO) and REPEAT (WHILE) take the test result; an algebraic there is auto-evaluated.
- Counters: `'c' INCR` / `'c' DECR` change the variable and return the new value.
- Instead of loops: `'Σ(j=1,n,j^2)'`, `SEQ`, list ops (ΣLIST, DOLIST, MAP).

```

« 1 5 FOR j j SQ NEXT »      1 4 9 16 25
« 1 9 FOR x x SQ 2 STEP »    odd squares
« 1 SWAP FOR n n n STEP »    1 2 4 8 ... ≤ n
« 1 10 START "ABC" NEXT 10 →LIST »
« WHILE DUP 2 MOD 0 == REPEAT 2 / DUP END DROP »
« DO 'c' INCR n * 's' STO+ UNTIL s 1000 > END »
« 0 1 ROT FOR j j SQ + NEXT »  Σ j² without Σ

```

List Processing

{l ₁ } ... {l _n } n « prog » DOLIST	prog gets the i-th element of each list; results → list
{list} n « prog » DOSUBS	sliding groups of n; NSUB = position, ENDSUB = # groups
{list} « prog » MAP	apply to each element (recurses into sublists)
{list} « 2-arg prog » STREAM	fold: ((e ₁ e ₂ f) e ₃ f) ...)
'f(i)' 'i' start end incr SEQ	list of results (obj index start end incr)
ΣLIST ΠLIST ΔLIST	sum · product · first differences
HEAD TAIL REVLIST SORT AUGMENT	first · rest · reverse · ascending · concatenate
ADD	element-wise + ({1 2} {3 4} + concatenates!)

Parallel processing: most one- and two-argument functions applied to a list act element-wise: {1 2 3} SQ → {1 4 9}, {1 2 3} 2 * → {2 4 6}, {1 2} {3 4} ADD → {4 6}. n may be omitted in DOLIST/DOSUBS when prog is a single command or a UDF-shaped program. NSUB/ENDSUB error outside DOSUBS.

```

{ 1 2 3 } { 4 5 6 } 2 « * » DOLIST → { 4 10 18 }
{ 1 2 3 4 } 2 « - » DOSUBS → { -1 -1 -1 }
{ 5 3 8 } « + » STREAM → 16
'k^2' 'k' 1 5 1 SEQ → { 1 4 9 16 25 }

```

Flags

n SF n CF n FS? n FC? n FS?C n	system -1...-128 · user 1...128
RCLF	→ { #sys-lo #usr-lo #sys-hi #usr-hi } 64-bit each; bit 0 = flag ±1 / ±65
#n STOF { ... } STOF	system only · system + user

FLAG	SET MEANS
-1	Principal solution only
-2	Symbolic constants (π e i MAXR MINR) evaluate to numbers
-3	Numerical results (functions of symbolics → numbers)
-17/-18	Angle mode: RAD -17 set · DEG both clear · GRAD -18 set
-19	→V2 makes a complex instead of a vector
-20 -21 -22	Underflow / overflow error · infinite result → MAXR (set)
-40 -41 -42	Clock shown · 24-h · date DD.MM
-51	Fraction mark is comma
-55	Last Arguments disabled
-56 -57	Error beep / alarm beep disabled
-60 -61 -62	Alpha lock · user-mode lock · user mode on
-63	Vectored ENTER (αENTER / βENTER)
-64	GETI/PUTI wrapped to first element
-95	Algebraic entry mode (clear = RPN)
-103 -105	Complex mode · Approximate mode
-117	Soft-key menus (clear = choose boxes)
-128	All variables real (default); clear allows complex

Default state is clear except -17, -27, -34, -90, -95, -128 and wordsize/base flags -5...-12. Save/restore around a program: RCLF → f « ... f STOF » or PUSH/POP.

Error Trapping

```

IFERR trap-clause THEN error-clause END
IFERR trap-clause THEN error-clause ELSE normal-clause END

"message" DOERR      user error (ERRN = #70000h)

#nnnh DOERR   n DOERR  raise a built-in error; 0 DOERR aborts silently

ERRN   ERRM   ERR0      last error number · message string · clear both

```

An error inside the trap-clause abandons the rest of that clause and jumps to the error-clause; execution then continues after END. Untrapped errors abort the program and show the message (arguments restored if Last Arguments is enabled). Built-in error numbers: Appendix A of the AUR.

```

« IFERR WHILE 1 REPEAT Σ+ END
  THEN "DONE" 1 DISP 1 FREEZE END »  add stack items to ΣDAT
                                     until an error

« DO "KEY IN a AND b" " " INPUT OBJ→
  UNTIL IFERR + THEN ERRM 5 DISP 2 WAIT 0
    ELSE 1 END END »                retry until two numbers
                                     were entered

« OBJ→ IF 3 == THEN "3 OBJECTS IN LIST" DOERR END »

```

Input

"prompt" PROMPT	message in status area; halts; user presses CONT
"prompt" "cmdline" INPUT OBJ→	prompt in stack area; returns the result string
"prompt" { "cmdline" pos opts } INPUT	with options (table below)
"title" { fields } { col tabs }	
{ resets } { inits } INFORM	→ { vals } 1 0
"title" { items } npos CHOOSE	→ choice 1 0
0 WAIT -1 WAIT	wait for a key → rc.p (-1 also shows the current menu)
KEY	→ 0 or rc 1 (poll); DO UNTIL KEY END
"msg" CLLCD 3 DISP 2 FREEZE HALT	free-form prompt; resume with CONT

INPUT OPTION	MEANING
"cmdline"	Initial command-line text (embedded newlines → multiple rows). Patterns: ":a: :b:" tags, "@comment@" , blanks to fill in.
n / {row col}	Cursor position; 0 = end of line; negative = replace cursor (else insert)
ALG	Algebraic/Program-entry mode
α	Alpha lock on
V	Verify result is valid object(s); re-prompt on Invalid Syntax

INFORM field = "Label" or { "Label" "help text" type₁ type₂ ... }; { } spans a column. Format col or { col tabs } (defaults 1, 3). Use NOVAL as a place-holder in resets/inits; empty fields return NOVAL. **CHOOSE** item = obj or { "shown" result }; npos 0 = view only. Both return 0 on CANCEL.

```

« "Key in a, b" { ":a: :b:" {1 0} V } INPUT OBJ→ TORSA »

« "Key in S.S. #" { "   -   " -1 } INPUT
  DUP 1 3 SUB SWAP 8 11 SUB »

« "OPTIONS" { { "ADD" 1 } { "VIEW" 2 } } 1 CHOOSE »

```

Output

obj n DISP	show obj on display line n (1 = top); strings shown without quotes
n FREEZE	keep area(s) until a key: 1 status · 2 stack/cmd line · 4 menu (add them)
CLLCD	blank the stack area
TEXT	back to the stack display
"msg" MSGBOX	modal message box
x WAIT	pause x seconds
obj tag →TAG	label a result → tag:obj (tag = string, 'name' or number)
obj →STR "Area = " SWAP + 1 DISP	label as text
freq dur BEEP	tone (Hz, seconds)
PVIEW →LCD	graphics output
obj {x y} font DISPHY	text at pixel position (font 0/1/2)

Displays made with DISP vanish when the program ends or halts unless FREEZE is used. « ... "DONE" 1 DISP 1 FREEZE » is the usual closing idiom.

Menus & Keys

{ "LBL" « action » ... } MENU	custom menu, stored in CST (persists)
{ { "LBL" « act » «  act » «  act » } } TMENU	temporary menu; plain / left-shift / right-shift actions
mm.pp MENU	built-in or library menu mm, page pp
0 MENU RCLMENU	restore previous menu · current menu number
obj rc.pf ASN	assign object to a user key
{ obj rc.pf ... } STOKEYS RCLKEYS	assign several · recall assignments
{ rc.pf ... } DELKEYS 'S' DELKEYS	delete some · delete all

Menus set by a program appear only when it halts or ends — use -1 WAIT to show the menu while paused. Menu items may be "label", name, or { "label" action }. Example: 65 MENU "Select Angle Mode" PROMPT.

Key codes rc.pf : row r (1–9), column c (1–6), plane p: 0/1 plain · 2  · 3  · 4 alpha · 5 alpha-left · 6 alpha-right; f = 1 for shift-and-hold. User assignments act in USER / 1-USER mode (flags -61/–62). WAIT returns rc.p; KEY returns rc only.

Every built-in menu — its number, pages, submenus and return keys — is drawn in the **Soft Menus** section after the guide; the keyboard itself is at the top of the page.

Execution & Debugging

obj EVAL	evaluate (see Program Objects table)
'sym' →NUM	force a numeric result
prog DEBUG	start & suspend immediately (HLT annunciator)
HALT	suspend here; CONT resumes; KILL cancels all halted programs
SST SST↓ NEXT	step over · step into · peek at next step
LASTARG	recover the last command's arguments (flag -55 clear)
PUSH ... POP	save / restore flags & current path
« prog » TEVAL	run and return elapsed time
SYSEVAL LIBEVAL FLASHEVAL	⚠ direct system access — wrong addresses crash

- A program is an object: call another program by name → **subroutine**; recursion is allowed. Locals of the caller are invisible in the callee (except ←compiled locals).
- Insert HALT to single-step from the middle; remove it afterwards. KILL clears the HLT annunciator.
- Program-entry: nested « » can be edited on multiple lines; ENTER discards line breaks.

Variables & Storage

obj 'NAME' STO 'NAME' RCL	store · recall (unevaluated)
'NAME' PURGE { 'A' 'B' } PURGE	delete one · several
obj 'NAME' STO+ STO- STO* STO/	storage arithmetic
'NAME' SINV SNEG SCONJ INCR DECR	in-place invert · negate · conjugate · ±1
'F(x)=x^2+1' DEFINE	creates F as a UDF ('X=5' DEFINE stores 5 in X)
'NAME' RENAME 'DIR' CRDIR HOME	rename · directories
UPDIR	
PATH VARS TVARS ORDER PGDIR CLVAR	path · names · by type · reorder · purge dir · clear

RESERVED VARIABLE	HOLDS
EQ · ΣDAT · ΣPAR	Current equation · statistic matrix · param
PPAR · VPAR · ZPAR · TPAR	Plot · 3D · zoom · param
CST · IOPAR · PRTPAR · ALRMDAT	Custom · I/O · p alarm d
Mpar · MHpar · PTPAR	Multiple equation solver · periodic
STARTUP · STARTOFF · STARTERR	Program at power · power-error
αENTER · βENTER	Vectors · ENTER (flag -6)
TOFF · EXPR · EXITED	Auto-of timeout · expres · exit sta
CASDIR: VX MODULO EPS REALASSUME PERIOD PRIMIT CASINFO	CAS va · modulu · epsilon · assum

Algebraic Precedence

#	HIGHEST → LOWEST
1	Unit attachment  (inside: parentheses,  bottom-up, * /)
2	Parenthesized expressions, innermost first
3	Functions with arguments (SIN, LOG, ...)
4	Factorial !
5	Power  (right-assoc: 2^2^3 = 2^(2^3)),  , XROOT
6	Negation, * / , MOD
7	+ -
8	== ≠ < > ≤ ≥
9	AND, NOT
10	OR, XOR
11	Left argument of  (where)
12	=

Same-precedence operators evaluate left to right. Stack syntax vs algebraic syntax: « SQ SWAP SQ + √ » ≡ « → x y 'SQRT(x^2+y^2)' ».

Idioms

RCLF → f « ... f STOF »	preserve & restore all flags
« IF DUP TYPE 0 ≠ THEN "Real expected" DOERR END ... »	argument check
« DEPTH 2 < « "Need 2 args" DOERR » IFT »	stack-depth guard
« WHILE DUP TYPE 3 == REPEAT Σ+ END »	consume all vectors on the stack
« ... "DONE" 1 DISP 1 FREEZE »	leave a message on screen
« DO UNTIL KEY END DROP »	wait for any key, discard its code
« 0 → c « WHILE ... 'c' INCR 5 ≤ AND REPEAT ... END » »	bounded loop
{ ... } « prog » MAP ΣLIST	sum of f(x) over a list
« → n « IF n 1 ≤ THEN n ELSE n 1 - FIB n 2 - FIB + END » »	recursion (stored as FIB)

System RPL — the layer underneath

Everything on this sheet is **User RPL** — the checked, documented language. Underneath it sits **System RPL**: the language the ROM itself is written in. Same interpreter, no safety net — entries are called by ROM address, arguments are never checked, counts are 20-bit *bints* instead of reals — and a typical program runs 3–10x faster. The 50g ships the toolchain: 256 ATTACH wakes the Development Library, MASD (library 257) compiles source with ASM, and the extable library from hpcalc.org gives every entry its name. A wrong word ends in “Try To Recover Memory?” — back up first.

→ [The HP 50g System RPL sheet](#) — a programmer’s guide (source format, dispatch, locals, loops, errors) and all 3,260 supported entries with addresses and stack diagrams, from *Programming in System RPL* (Kalinowski & Dominik, 2nd ed.).

Units — the complete table

UNITS menu	 6 (CONVERT) → UNITS, or n MENU
key press	multiplies by that unit
 key	converts the level-1 value to that unit
 key	divides by that unit
in a program	 key yields '1_unit' CONVERT,  yields '1_unit' /

THE 16 CATALOGUES

LENGTH (menu 43)	UNITS 2 · 43 MENU
AREA (menu 44)	UNITS 3 · 44 MENU
VOLUME (menu 45)	UNITS 4 · 45 MENU
TIME (menu 46)	UNITS 5 · 46 MENU
SPEED (menu 47)	UNITS 6 · 47 MENU
MASS (menu 48)	UNITS 7 · 48 MENU
FORCE (menu 49)	UNITS 8 · 49 MENU
ENERGY (menu 50)	UNITS 9 · 50 MENU
POWER (menu 51)	UNITS 10 · 51 MENU
PRESSURE (menu 52)	UNITS 11 · 52 MENU
TEMPERATURE (menu 53)	UNITS 12 · 53 MENU
ELECTRIC CURRENT (menu 54)	UNITS 13 · 54 MENU
ANGLE (menu 55)	UNITS 14 · 55 MENU
LIGHT (menu 56)	UNITS 15 · 56 MENU
RADIATION (menu 57)	UNITS 16 · 57 MENU
VISCOSITY (menu 58)	UNITS 17 · 58 MENU

EVERY UNIT, WITH ITS EXACT SI VALUE

All 127 units the 50g knows, from Appendix B of the AUR. Values are exact as printed by HP. Temperature units carry both a scale factor and an offset, which is why °C and °F show two figures — CONVERT handles the offset, but arithmetic on a bare °C value does not.

?	undefined · undefined unit
°	degree · $1.74532925199 \times 10^{-2}$ r
A	ampere · 1 A
a	are · 100 m ²
acre	acre · 4046.87260987 m ²
arcmin	minute of arc · $2.90888208666 \times 10^{-4}$ r
arcs	second of arc · $4.8481368111 \times 10^{-6}$ r
atm	atmosphere · 101325 kg·m·s ⁻²
au	astronomical unit · 1.495979 ×10 ¹¹ m
b	barn · 1 ×10 ⁻²⁸ m ²
bar	bar · 100000 kg·m·s ⁻²
bbl	barrel · .158987294928 m ³
Bq	becquerel · 1 1/s
Btu	international table Btu · 1055.05585262 kg·m ² /s ²
bu	bushel · .03523907 m ³
C	coulomb · 1 A·s
c	speed of light · 299792458 m/s
°C	degree Celsius · 1K or 274.15 K
cal	calorie · 4.1868 kg·m ² /s ²
cd	candela · 1 cd
chain	chain · 20.1168402337 m
Ci	curie · 3.7 ×10 ¹⁰ 1/s
ct	carat · .0002 kg
cu	US cup · 2.365882365 ×10 ⁻⁴ m ³
d	day · 86400 s
dB	decibel · 1 (dimensionless value)
dyn	dyne · .00001 kg·m·s ⁻²
erg	erg · .0000001 kg·m ² /s ²
eV	electron volt · $1.60217733 \times 10^{-19}$ kg·m ² /s ²
F	farad · 1 A ² ·s ⁴ /kg·m ²
°F	degrees Fahrenheit · 0.555555555556 K or 255.927777778 K
fath	fathom · 1.82880365761 m
fbm	board foot · .002359737216 m ³
fc	footcandle · 10.7639104167 cd·sr/m ²
Fdy	faraday · 96487 A·s
fermi	fermi · 1 ×10 ⁻¹⁵ m
flam	footlambert · 3.42625909964 cd/m ²
ft	international foot · .3048 m
ftUS	survey foot · .304800609601 m

g	gram · .001 kg
ga	standard freefall · 9.80665 m/s ²
gal	US gallon · .003785411784 m ³

galC	Canadian gallon · .00454609 m ³
galUK	UK gallon · .004546092 m ³
gf	gram-force · .00980665 kg·m/s ²
gmol	gram-mole · 1 mol
grad	gradian · $1.57079632679 \times 10^{-2}$ r
grain	grain · .00006479891 kg
Gy	gray · 1 m ² /s ²
H	henry · 1 kg·m ² /A ² ·s ²
h	Hour · 3600 s
hp	horsepower · 745.699871582 kg·m ² /s ³
Hz	hertz · 1/s
in	inch · .0254 m
inH2O	inches of water, 60 oF · 248.84 kg/m·s ²
inHg	inches of mercury, 0 oC · 3386.38815789 kg/m·s ²
J	joule · 1 kg·m ² /s ²
K	kelvins · 1 K
kg	kilogram · 1 kg
kip	kilopound-force · 4448.22161526 kg·m/s ²
knot	nautical miles per hour · .514444444444 m/s
kph	kilometers per hour · .277777777778 m/s
l	liter · .001 m ³
lam	lambert · 3183.09886184 cd/m ²
lb	avoirdupois pound · .45359237 kg
lbf	pound -force · 4.44822161526 kg·m/s ²
lbmol	pound-mole · 453.59237 mol
lbt	troy pound · .3732417216 kg
lm	lumen · 1 cd·sr
lx	lux · 1 cd·sr/m ²
lyr	light year · 9.46052840488 ×10 ¹⁵ m
m	meter · 1 m
mho	mho · 1 A ² ·s ³ /kg·m ²
mi	international mile · 1609.344 m
mil	mil · .0000254 m
min	minute · 60 s
miUS	US statute mile · 1609.34721869 m
mmHg	millimeter of mercury (torr · 133.322368421 kg/m·s ²
mol	mole · 1 mol
mph	miles per hour · .44704 m/s
N	newton · 1 kg·m/s ²
nmi	nautical mile · 1852 m

oz ounce · .028349523125 kg
ozfl US fluid ounce · 2.95735295625 ×10⁻⁵ m³

ozt troy ounce · .311034768 kg
ozUK UK fluid ounce · 2.8413075 ×10⁻⁵ m³
P poise · .1 kg/m·s
Pa pascal · 1 kg/m·s²
pc parsec · 3.08567818585 ×10¹⁶ m
pdl poundal · .138254954376 kg·m/s²
ph phot · 10000 cd·sr/m²
pk peck · .0088097675 m³
psi pounds per square inch · 6894.75729317 kg·m/s²
pt pint · .000473176473 m³
qt quart · .000946352946 m³
R roentgen · .000258 A·s/kg
r radian · 1 r
°R degrees Rankine · 0.555555555556 K
rad rad · .01 m²/s²
rd rod · 5.02921005842 m
rem rem · .01 m²/s²
rpm revolutions per minute · 60 1/s
S siemens · 1 A²·s³/kg·m²
s second · 1 s
sb stilb · 10000 cd/m²
slug slug · 14.5939029372 kg
sr steradian · 1 sr
St stokes · .0001 m²/s
st stere · 1 m³
Sv sievert · 1 m²/s²
T tesla · 1 kg/A·s²
t metric ton · 1000 kg
tbsp tablespoon · 1.47867647813 ×10⁻⁵ m³
therm EEC therm · 105506000 kg·m²/s²
ton short ton · 907.18474 kg
tonUk long ton (UK · 1016.0469088 kg
torr torr (mmHg · 133.322368421 kg/m·s²
tsp teaspoon · 4.92892159375 ×10⁻⁶ m³
u unified atomic mass · 1.6605402 ×10⁻²⁷ kg
V volt · 1 kg·m²/A·s³
W watt · 1 kg·m/s³
Wb weber · 1 kg·m²/A·s²
yd international yard · .9144 m
yr year · 31556925.9747 s

μ micron · 1 ×10⁻⁶ m
Å Angstrom · 1 ×10⁻¹⁰ m
Ω ohm · 1 kg·m²/A²·s³

Soft Menus every keyboard menu, its pages and submenus, and the keys that open them — read off ROM 2.15 with flag -117 set

How the soft menu work

A **soft menu** is the row of six labels across the bottom of the display; **F1–F6** press them. A menu with more than six entries has **pages**: **NXT** steps forward (and wraps from the last page to the first), **PREV** steps back. Labels ending in a submenu open a deeper menu; the **last page of every MTH and PRG submenu ends with a key named for its parent** (**MTH** , **PRG** , **LIST** , **BRCH** ...) that takes you back up — the dashed amber lines in the trees. A dotted rule inside a card is a page break: press **NXT** . Two more ways up: press the parent's key again (**SYMB** for MTH), or **hold** **MODE** and press **NXT** for LAST MENU (menu 0). **MODE** (CUSTOM) and **VAR** are the two menus you fill yourself.

Flag -117 must be set (MODE → FLAGS → 117 “Soft MENU”, or -117 SF). Out of the box it is clear and the very same keys open a **choose box** instead: a numbered list where MTH 3 = LIST, PRG 6 = LIST, and so on — the numbers in the trees are those item numbers (F1 = 1 ... F6 = 6, next page continues 7...). Either way the menu numbers work from a program: **11 MENU** shows MTH LIST, **34 TMENU** shows PRG LIST temporarily, **0 MENU** restores the last menu.

WORKING WITH LISTS – TWO MENUS

SYMB	F3	(MTH LIST · 11 MENU)	ALIST ΣLIST ΠLIST SORT REVLIST ADD – the arithmetic on a whole list
EVAL	F6	(PRG LIST · 34 MENU)	ELEM PROC OBJ→ -LIST SUB REPL – build, split, slice
EVAL	F6 F1	(ELEM · 35)	GET GETI PUT PUTI SIZE POS · NXT · HEAD TAIL – one element at a time
EVAL	F6 F2	(PROC · 36)	DOLIST DOSUBS NSUB ENDSUB STREAM REVLIST · NXT · SORT SEQ – apply a program across a list
EVAL	F5	(PRG TYPE · 33)	OBJ→ -LIST also live here, with the other converters
MAP – in no menu at all			{ list } « prog » MAP applies a program to every element, but HP gave this CAS-library command no soft-menu slot: find it in the catalog (SYMB CAT) or type it. The menu-reachable cousins are DOLIST / DOSUBS in PRG LIST PROC – and DB48X fixes the omission by putting Map straight in its ListMenu
back up			NXT then F6 (LIST) from ELEM or PROC; NXT then F6 (PRG) from LIST; SYMB or EVAL from anywhere

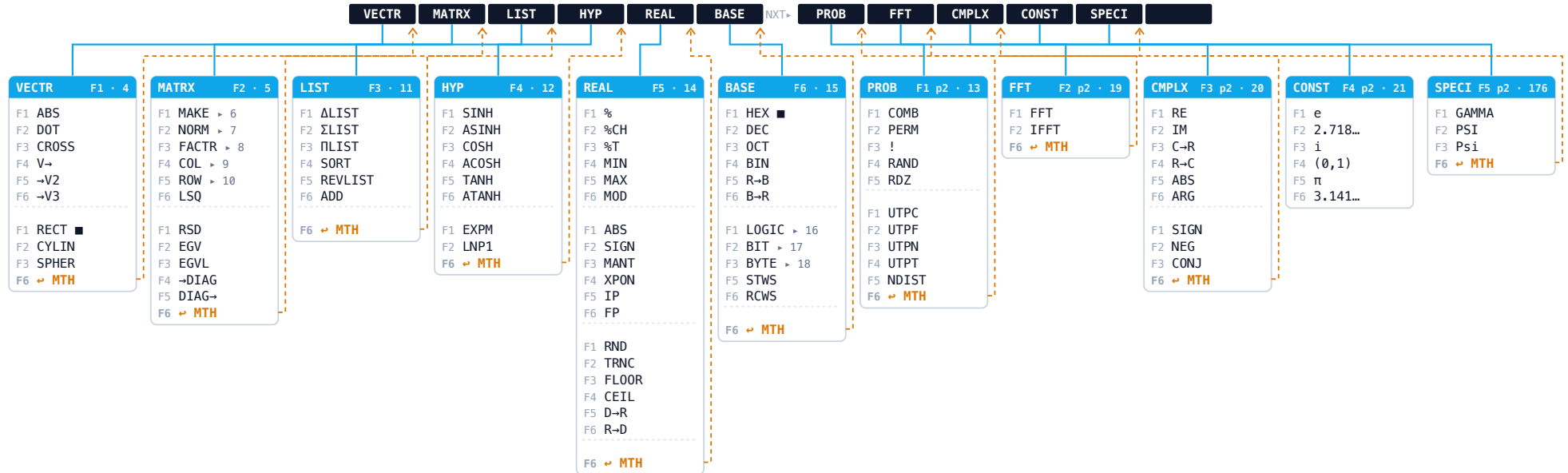
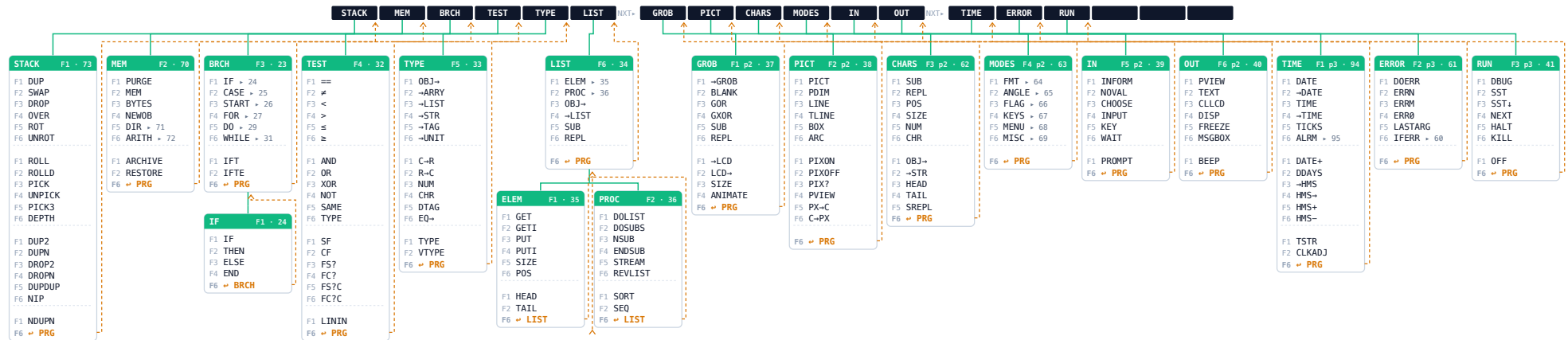
Menu labels on the calculator are cut to five characters (REVL, DOLIS, ENDSU...); the trees spell the full command so you can find it in the index below.

Keys that open a menu

is the white left-shift key, the orange right-shift key — drawn as on the keyboard. Underlined blue opens a soft menu, violet a choose box or form. Rows 5–6 and 10 carry no menus and are left out.

<u>FILES</u>	<u>BEGIN</u>	<u>CUSTOM</u>	<u>END</u>	i	
APPS	MODE	TOOL			
UPDIR	<u>COPY</u>	RCL	<u>CUT</u>	PREV	<u>PASTE</u>
VAR	STO➤	NXT			
CMD	<u>UNDO</u>	<u>PRG</u>	<u>CHARS</u>	MTRW	<u>EQW</u>
HIST	EVAL	'	SYMB	DEL	CLEAR
USER	<u>ENTRY</u>	<u>S.SLV</u>	<u>NUM.SLV</u>	<u>EXP&LN</u>	<u>TRIG</u>
ALPHA	7	8	9		
	<u>CALC</u>	<u>ALG</u>	<u>MATRICES</u>	<u>STAT</u>	<u>CONVERT</u>
	4	5	6		
	<u>ARITH</u>	<u>CMPLX</u>	DEF	<u>LIB</u>	<u>#</u>
	1	2	3		

hold + MODE	MODES soft menu (63)
hold + NXT	LAST MENU
hold + VAR	HOME
hold + EVAL	CHARS soft menu (62)
hold + 7	solver soft menu (74)
hold + 9	TIME soft menu (167)
A ... F	the six PLOT input forms

MTH press ☐ SYMB · 3 MENU · NXT wraps to page 1, ☐ NXT = PREVPRG press ☐ EVAL · 22 MENU · NXT wraps to page 1, ☐ NXT = PREV

CONVERT  6 · 131	UNITS BASE TRIG REWRITE MATRIX – each opens the  -key menu of the same name (42 168 140 172 173)
UNITS  6 · 42	TOOLS LENG AREA VOL TIME SPEED ⇄ MASS FORCE ENRG POWR PRESS TEMP ⇄ ELEC ANGL LIGHT RAD VISC · every unit menu (43–59) ends with ⇄ UNITS
BASE  3 · 168	HEX DEC OCT BIN R→B B→R ⇄ LOGIC BIT BYTE · STWS RCWS · LOGIC/BIT/BYTE (169–171) end with ⇄ BASE
ARITH  1 · 125	INTEGER POLY MODULO PERMUTATION DIVIS FACTORS ⇄ LGCD PROPFRAC SIMP2 · submenus 127 126 128 174
CMPLX  1 · 130	ARG ABS CONJ i IM NEG ⇄ RE SIGN
CALC  4 · 123	DERIV LIMIT DIFF GRAPH DERVX INTVX · submenus 162 163 164 177
ALG  4 · 124	COLLECT EXPAND FACTOR LNCOLLECT LIN PARTFRAC ⇄ SOLVE SUBST TEXPAND
MATRICES  5 · 129	CREATE OPER FACTOR QUADF LIN→S LINAP ⇄ EIGEN VECTOR · CREATE (146) has 3 pages and ends with ⇄ MATRX
S.SLV  7 · 120	DESOLVE ISOL LDEC LINSOLVE SOLVE SOLVEVX ⇄ ZEROS
EXP&LN  8 · 121	EXPLN EXPM LIN LNCOLLECT LNP1 TEXPAND ⇄ TSIMP
TRIG  8 · 122	HYP ACOS2S ASIN2C ASIN2T ATAN2S HALFTAN ⇄ SINCOS TAN2SC TAN2SC2 TCOLLECT TEXPAND TLIN ⇄ TRIG TRIGCOS TRIGSIN TRIGTAN TSIMP · HYP (161) ends with ⇄ TRIG
SYMB · 143	ALG ARITH CALC GRAPH SOLVE TRIG ⇄ EXPLN · each (151 134 152 153 154 139 155) ends with ⇄ SYMB
TOOL	EDIT VIEW RCL STO► PURGE CLEAR ⇄ CASCMD HELP
VAR · 2	your variables in the current directory;  VAR = UPDIR
CST  MODE · 1	the custom menu in variable CST (empty until you build one – see Menus & Keys)
LIB  2 · 111	attached libraries; empty on a clean calculator
MODES hold  +MODE · 63	FMT ANGLE FLAG KEYS MENU MISC (64–69) – same as PRG NXT MODES
CHARS hold  +EVAL · 62	the PRG CHARS soft menu (plain  EVAL opens the character browser instead)
solver hold  +7 · 74	ROOT DIFFEQ POLY SYS TVM (75–79) – the 48-style soft-menu solver; plain  7 is a choose box
TIME tools hold  +9 · 167	the time soft menu; plain  9 is a choose box
choose boxes only	APPS ·  7 NUM.SLV ·  5 STAT ·  9 TIME ·  9 FINANCE (input form) ·  SYMB CAT ·  EVAL CHARS – no soft-menu form, NXT does nothing

Number after the name = argument for MENU / TMENU. ⬅ = next page (NXT). Sources: HP 50g AUR appendices G (keyboard shortcuts), H (menu-number table), I (command menu paths); menu contents checked on ROM 2.15.

Command Reference 774 commands · 25 categories · hover for details

Stack24

DUPLICATE & DROP

- DUP Duplicate Object
- DUP2 Duplicate 2 Objects
- DUPN Duplicate n Objects
- DUPDUP Duplicate Twice
- NDUPN Duplicate n Times
- DROP Drop Object
- DROP2 Drop 2 Objects
- DROPN Drop n Objects
- ; Drop Safe
- CLEAR Clear

REORDER

- SWAP Swap Objects
- OVER Over
- ROT Rotate Objects
- UNROT Reverse Rotate
- ROLL Roll Objects
- ROLLD Roll Down
- PICK Pick Object
- PICK3 Copy Level 3
- UNPICK Replace Level n
- NIP Drop Level 2

INSPECT & HISTORY

- DEPTH Depth
- LASTARG Last Arguments
- LAST Last Arguments → LASTARG
- ANS Recall History Answer

Arithmetic74

BASIC

- + Add
- − Subtract
- * Multiply
- / Divide
- ^ Power
- √ Square Root
- SQ Square
- INV Inverse (1/x)
- NEG Negate
- XRROOT xth Root of y

SIGN & MAGNITUDE

- ABS Absolute Value
- SIGN Sign
- MAX Maximum
- MIN Minimum
- MOD Modulo

PARTS & ROUNDING

IP

Integer Part

FP

Fractional Part

FLOOR

Floor

CEIL

Ceiling

RND

Round

TRNC

Truncate

MANT

Mantissa

XPON

Exponent

PERCENT

%

Percent

%CH

Percent Change

%T

Percent of Total

EXPONENTIAL & LOG

EXP

Exponential

LN

Natural Logarithm

LOG

Common Logarithm

ALOG

Common Antilogarithm

EXPM

Exponential Minus 1

LNP1

Natural Log of x Plus 1

TRIGONOMETRIC

SIN

Sine

COS

Cosine

TAN

Tangent

ASIN

Arc Sine

ACOS

Arc Cosine

ATAN

Arc Tangent

HYPERBOLIC

SINH

Hyperbolic Sine

COSH

Hyperbolic Cosine

TANH

Hyperbolic Tangent

ASINH

Arc Hyperbolic Sine

ACOSH

Inverse Hyperbolic Cosine

ATANH

Arc Hyperbolic Tangent

ANGLE & HMS

D→R

Degrees to Radians

R→D

Radians to Degrees

→HMS

Decimal to HMS

HMS→

HMS to Decimal

HMS+

HMS Plus

HMS−

HMS Minus

SPECIAL FUNCTIONS

!

Factorial (Gamma)

FACT

Factorial (Gamma) → !

GAMMA

Gamma Function

PSI

Polygamma

Psi

Digamma

COMB

Combinations

PERM

Permutations

RANDOM

RAND

Random Number

RDZ

Randomize

RANM

Random Matrix

CONSTANTS

π

Pi

e

e

i

i

MAXR

Maximum Real

MINR

Minimum Real

∞

Infinity

?

Undefined

NUMERIC ↔ EXACT

→NUM

Evaluate to Number

XNUM

To Numeric

XQ

To Rational

→Q

To Quotient

→Qπ

To Quotient Times π

R→I

Real to Integer

I→R

Integer to Real

Integers & Modular29

NUMBER THEORY

GCD

Greatest Common Divisor

LCM

Least Common Multiple

IEGCD

Extended GCD (int)

IABCUV

Integer Bézout Solve

ICHINREM

Chinese Remainder (int)

CHINREM

Chinese Remainder (poly)

IDIV2

Integer Div & Remainder

IQUOT

Integer Quotient

IREMAINDER

Integer Remainder

EULER

Euler Totient φ(n)

PA2B2

Prime → a²+b²

IBERNOULLI

Bernoulli Number

PRIMES & FACTORS

ISPRIME?

Prime Test

NEXTPRIME

Next Prime

PREVPRIME

Previous Prime

FACTORS

Prime Factors

DIVIS

List of Divisors

MODULAR ARITHMETIC

MODSTO

Set Modulus

ADDTMOD

Modular Add

SUBTMOD

Modular Subtract

MULTMOD

Modular Multiply

DIVMOD

Modular Divide

POWMOD

Modular Power

INVMOD

Modular Inverse

EXPANDMOD

Modular Expand

FACTORMOD

Modular Factorize

GCDMOD

Modular GCD

DIV2MOD

Modular Euclidean Division

RREFMOD

Modular RREF

Complex Numbers13

PARTS

RE

Real Part

IM

Imaginary Part

ARG

Argument

CONJ

Conjugate

ABS

Absolute Value

SIGN

Sign

CONVERT

R→C

Real to Complex

C→R

Complex to Real

→V2

Stack → Vector/Complex

V→

Vector/Complex → Stack

COORDINATE MODE

RECT

Rectangular Mode

CYLIN

Cylindrical Mode

SPHERE

Spherical Mode

Vectors & Matrices82

CREATE & CONVERT

→ARRAY

Stack to Array

ARRAY→

Array to Stack

→V2

Stack → Vector/Complex

→V3

Stack to 3-Element Vector

V→

Vector/Complex → Stack

CON

Constant Array

IDN

Identity Matrix

RDM

Redimension Array

RANM

Random Matrix

HILBERT

Hilbert Matrix

VANDERMONDE

Vandermonde Matrix

LCXM

Build Matrix from Program

AXL

Array ↔ List

AXM

Numeric → Symbolic Matrix

ELEMENTS

GET

Get Element

GETI

Get and Increment Index

PUT

Put Element

PUTI

Put and Increment Index

SIZE

Size

ROWS & COLUMNS

→ROW

Matrix to Rows

ROW→

Rows to Matrix

ROW+

Insert Row

ROW−

Delete Row

RSWP

Row Swap

RCI

Multiply Row by Constant

RCIJ

Add Multiplied Row

→COL

Matrix to Columns

COL→

Columns to Matrix

COL+

Insert Column

COL−

Delete Column

CSWP

Column Swap

→DIAG

Matrix Diagonal to Array

DIAG→

Vector to Matrix Diagonal

PRODUCTS & NORMS

DOT

Dot Product

CROSS

Cross Product

HADAMARD

Hadamard Product

ABS

Absolute Value

CNRM

Column Norm

RNRM

Row Norm

SNRM

Spectral Norm

SRAD

Spectral Radius

COND

Condition Number

TRACE

Matrix Trace

DET

Determinant

RANK

Matrix Rank

TRN

Transpose Matrix

TRAN

Transpose (no conjugate) → ...

DECOMPOSITIONS

LU

LU Decomposition

LQ

LQ Factorization

QR

QR Factorization

qr

qr Factorization (square)

SVD

Singular Value Decomposition

SVL

Singular Values

SCHUR

Schur Decomposition

CHOLESKY

Cholesky Factorization

EGV

Eigenvalues and Eigenvectors

EGVL

Eigenvalues

JORDAN

Jordan Decomposition

PCAR

Characteristic Polynomial

PMINI

Minimal Polynomial

DIAGMAP

Apply Function to Diagonaliz...

MAD

Matrix Adjoint & Details

LINEAR SYSTEMS

LSQ

Least Squares Solution

RSD

Residual

REF

Row Echelon Form

RREF

Reduced Row Echelon

rref

RREF with Pivots

LINSOLVE

Solve Linear System

SYST2MAT

System → Matrix

VECTOR SPACES

BASIS

Basis of Subspace

IBASIS

Intersection Basis

KER

Kernel Basis

IMAGE

Image Basis

GRAMSCHMIDT

Gram-Schmidt Orthonor...

QUADRATIC FORMS & ISOMETRIES

AXQMatrix → Quadratic Form

QXAQuadratic Form → Matrix

GAUSSDiagonalize Quadratic Form

SYLVESTER Sylvester Decomposition

ISOMIsometry Characteristics

MKISOMMake Isometry Matrix

FOURIER

FFTDiscrete Fourier Transform

IFFTInverse Discrete Fourier Transform

Lists25

BUILD & SPLIT

~LISTStack to List

LIST~List to Stack

OBJ~Object to Stack

AUGMENTConcatenate

REVLISTReverse List

SORTAscending Order Sort

ACCESS

HEADFirst Listed Element

TAILLast Listed Elements

SUBSubset

POSPosition

REPLReplace

SIZESize

GETGet Element

PUTPut Element

ARITHMETIC

ADDAdd List

ZLISTList Sum

NLISTList Product

ALISTList Differences

ITERATION

DOLISTDo to List

DOSUBSDo to Sublist

MAPMap Program over List

SEQSequential Calculation

STREAMStream Execution

NSUBNumber of Sublist

ENDSUBEnding Sublist

Strings & Tags14

STRINGS

~STRObject to String

STR~Evaluate String

CHRCharacter

NUMCharacter Number

SREPLString Find & Replace

C\$Counted String Delimiter

COMMON (LIST / STRING)

SUBSubset

POSPosition

REPLReplace

SIZESize

HEADFirst Listed Element

TAILLast Listed Elements

TAGGED OBJECTS

~TAGStack to Tag

DTAGDelete Tag

Binary Integers21

BASE

BINBinary Mode

OCTOctal Mode

DECDecimal Mode

HEXHexadecimal Mode

WORDSIZE & CONVERT

STWSSet Wordsize

RCWSRecall Wordsize

R~BReal to Binary

B~RBinary to Real

SHIFT

SLShift Left

SRShift Right

SLBShift Left Byte

SRBShift Right Byte

ASRArithmetic Shift Right

ROTATE

RLRotate Left

RRRotate Right

RLBRotate Left Byte

RRBRotate Right Byte

BITWISE LOGIC

ANDAnd

OROR

XORExclusive OR

NOTNOT

Tests & Logic16

COMPARISON

<Less Than

≤Less Than or Equal

>Greater Than

≥Greater Than or Equal

==Logical Equality

≠Not Equal

SAMESame Object

LOGIC

ANDAnd

OROR

XORExclusive OR

NOTNOT

CONDITIONAL FUNCTIONS

IFTIF-THEN

IFTEIF-THEN-ELSE

TYPE TESTS

TYPEType

VTYPEVariable Type

LININLinear Test

Symbolic Algebra59

EXPAND & COLLECT

EXPANDExpand & Simplify

EXPANExpand Products

COLLECTCollect / Factor

COLCTCollect Terms → COLLECT

FACTORFactorize

DISTRIBDistribute (one step)

FDISTRIBFull Distribution

POWEXPANDPower → Product

SIMPLIFY

SIMPLIFYSimplify

SEVALSimplify Sub-expressions

TSIMPSimplify Exp/Ln

TEXPANDExpand Transcendentals

LINLinearize Exponentials

LNCOLLECTCollect Logarithms

EXP2POWExp(Ln) → Power

EXPLNTrig → Exp/Ln

SOLVE

SOLVESolve

SOLVEVXSolve wrt VX

ZEROSZeros of Function

ISOLIsolate Variable

QUADSolve Quadratic Equation

MSLVMultiple Solve (numeric)

DESOLVESolve ODE

LDECLinear ODE (const coeff)

SUBSTITUTE & MATCH

SUBSTSubstitute

|Where

↑MATCHBottom-Up Match and Replace

↓MATCHMatch Pattern Down

SHOWShow Variable

APPLYApply to Arguments

QUOTEQuote Argument

EQUATIONS

=Equals

EQ~Equation to Stack

EXLRExtract Left & Right Sides

RATIOPrefix Divide

DEFINEDefine Variable or Function

DEFDefine (algebraic)

FXNDNumerator & Denominator

DROITELine Through Two Points

TRIG REWRITING

TRIGExp/Ln → Trig

TRIGCOSSimplify to Cosines

TRIGSINSimplify to Sines

TRIGTANSimplify to Tangents

TLINLinearize Trig

TCOLLECTCollect Trig Products

HALFTANHalf-Angle Tangent Rewrite

SINCOSExp/Ln → Sin/Cos

TAN2SCtan → sin/cos

TAN2SC2tan → sin2x/(1+cos2x)

TAN2CS2tan → (1-cos2x)/sin2x

ACOS2Sacos → asin Rewrite

ASIN2Casin → acos Rewrite

ASIN2Tasin → atan Rewrite

ATAN2Satan → asin Rewrite

EXP2HYPExp → Hyperbolic

VARIABLES

LNAMEVariable Names in Expr

LVARList Variables

STOVXSet CAS Variable

RCLVXRecall CAS Variable

Calculus33

DERIVATIVES

∂Derivative

DERIVPartial Derivatives

DERVXDerivative wrt VX

dnDifferential Operator

INTEGRALS

∫Integral

INTAntiderivative at Point

INTVXAntiderivative wrt VX

RISCHRisch Integration

IBPIntegration by Parts

PREVALDefinite Antiderivative

SUMS & LIMITS

ΣSummation

SIGMADiscrete Antiderivative

SIGMAVXDiscrete Antiderivative wrt VX

limLimit → LIMIT

LIMITLimit

SERIES

TAYLRTaylor Polynomial

TAYLOR0Taylor at 0 (4th order)

SERIESSeries Expansion

TRUNCTruncate Series

DIVPCSeries Quotient

VECTOR CALCULUS

CURLCurl

DIVDivergence

LAPLLaplacian

HESSHessian & Gradient

POTENTIALScalar Potential

VPOTENTIALVector Potential

TRANSFORMS

LAPLaplace Transform

ILAPInverse Laplace

FOURIERFourier Coefficient

FUNCTION ANALYSIS

DOMAINDomain of Function

TABVARVariation Table

TABVALTable of Values

SIGNTABSign Table

Polynomials32

ROOTS & COEFFICIENTS

- PROOTPolynomial Roots
- PCOEFMonic Polynomial Coefficients
- PEVALPolynomial Evaluation
- PTAYLTaylor Polynomial at a
- HORNERHorner Scheme
- DEGREEPolynomial Degree
- FCOEFRoots/Poles → Rational
- FRROOTSRoots & Poles

DIVISION & GCD

- QUOTPolynomial Quotient
- REMAINDERPolynomial Remainder
- DIV2Euclidean Division
- EGCDExtended GCD (poly)
- ABCUVPolynomial Bézout Solve
- LGCDGCD of List
- RESULTANTResultant
- SIMP2Simplify Pair
- EPSX0Zero Small Coefficients

FRACTIONS

- PARTFRACPartial Fractions
- PROPFRACTProper Fraction

GRÖBNER & INTERPOLATION

- GBASISGröbner Basis
- GREDUCEGröbner Reduce
- REORDERReorder Variables
- LAGRANGELagrange Interpolation

SPECIAL POLYNOMIALS

- CYCLOTOMICCyclotomic Polynomial
- HERMITEHermite Polynomial
- LEGENDRELegendre Polynomial
- TCHEBYCHEFFTchebycheff Polynomial
- STURMS Sturm Sequences
- STURMABSturm Root Count

PERMUTATIONS

- C2PCycles → Permutation
- P2CPermutation → Cycles
- CIRCCompose Permutations

Solvers & Finance23

NUMERIC SOLVER

- ROOTRoot-Finder
- SOLVEQNStart Solver for Equation Set
- EQNLIBEQuation Library
- SOLVERSolver Menu
- STEQStore in EQ
- RCEQRecall from EQ

MULTIPLE-EQUATION SOLVER

- MSOLVRMultiple-Equation Solver
- MROOTMES Root
- MINITMES Initialize
- MITMMES Menu Item Order
- MCALCMES Calculated Var
- MUSERMES User Var

DIFFERENTIAL EQUATIONS (NUMERIC)

- RKFRKF Initial-Value Solve
- RKFERRKF Error Estimate
- RKFSTEP RKF Next Step
- RRKRRK Initial-Value Solve
- RRKSTEP RRK Next Step & Method
- RSBERRRosenbrock Error Estimate

FINANCE (TVM)

- TVMTVM Menu
- TVMROOTTVM Root
- TVMBEGPayment at Start of Period
- TVMENDPayment at End of Period
- AMORTAmortize

Units & Constants20

UNIT OBJECTS

- Unit Attachment
- UNITStack to Unit Object
- UVALUnit Value
- CONVERTConvert Units
- UBASEConvert to SI Base Units
- UFACTFactor Unit

CONSTANTS LIBRARY

- CONSTConstant Value
- CONLIBOpen Constants Library
- CONSTANTSConstants Menu

PHYSICS & ENGINEERING

- TDELTA Temperature Delta
- TINC Temperature Increment
- ZFACTOR Gas Compressibility Z Factor
- DARCYDarcy Friction Factor
- FANNINGFanning Friction Factor
- F0ABlack Body Emissive Power
- SIDENS Silicon Intrinsic Density

PERIODIC TABLE

- PERTBLPeriodic Table
- PTPROPElement Property
- MOLWTMolecular Weight
- PERINFOPeriodic Table Info

Program Control38

STRUCTURE

- <>Program Delimiters
- RPL>Run RPL in Algebraic

CONDITIONALS

- IFIF Conditional Structure
- THENTHEN
- ELSEELSE
- ENDEND
- CASECASE Conditional Structure

DEFINITE LOOPS

- STARTSTART Definite Loop Structure
- FORFOR Definite Loop Structure
- NEXTNEXT
- STEPSTEP

INDEFINITE LOOPS

- DO DO Indefinite Loop Structure
- UNTILUNTIL
- WHILEWHILE Indefinite Loop Structure
- REPEATREPEAT
- ENDEND

ERROR TRAPPING

- IFERRIf Error Conditional Structure
- DOERRDo Error
- ERRNError Number
- ERRMError Message
- ERR0Clear Last Error Number

LOCAL VARIABLES

- Create Local Variables
- LOCALCreate Locals (algebraic)
- UNBINDRemove LOCAL Variables

EXECUTION

- EVALEvaluate Object
- SYSEVALEvaluate System Object
- LIBEVALEvaluate Library Function
- FLASHEVALEvaluate Flash Function
- HALTHalt Program
- CONTContinue Program Execution
- KILLCancel Halted Programs
- WAITWait
- PUSHSave Flags & Dir
- POPRestore Flags & Dir

DEBUGGING

- DBUGDebug
- SSTSingle Step
- SST+Step Into Subroutine
- NEXTNEXT

Flags8

SET & CLEAR

- SFSet Flag
- CFClear Flag

TEST

- FS?Flag Set?
- FC?Flag Clear?
- FS?CFlag Set? Clear
- FC?CFlag Clear? Clear

SAVE & RESTORE

- RCLFRecall Flags
- STOFStore Flags

Input & Output26

INPUT

- INPUTInput
- PROMPTPrompt
- PROMPTSTOPrompt
- INFORMInput Form (Dialog Box)
- CHOOSEChoose Box
- NOVALINFORM Place Holder

KEYBOARD

- KEYKey
- KEYEVALEvaluate Key Press
- WAITWait
- KEYTIMESet Keytime
- KEYTIME→Recall Keytime

OUTPUT

- DISPDisplay
- DISPYDisplay
- MSGBOXMessage Box
- CLLCDClear LCD
- FREEZEFreeze Display
- TEXTShow Stack Display
- SCROLLView Object
- BEEPBeep

MENUS

- MENUDisplay Menu
- TMENUTemporary Menu
- RCLMENUREcall Menu Number

USER KEYS

- ASNAssign
- STOKEYSStore Key Assignments
- RCLKEYSRecall Key Assignments
- DELKEYSDelete Key Assignments

Variables & Memory47

STORE & RECALL

- STOStore
- Store
- RCLRecall
- PURGE Purge
- STOREStore (algebraic)
- UNASSIGNPurge & Return Values
- RENAMERename Object

STORAGE ARITHMETIC

- STO+Store Plus
- STO-Store Minus
- STO*Store Times
- STO/Store Divide
- SINVStore Inverse
- SNEGStore Negate
- SCONJStore Conjugate
- INCRIncrement
- DECRDecrement

DIRECTORIES	
HOME	HOME Directory
UPDIR	Up Directory
CRDIR	Create Directory
PGDIR	Purge Directory
PATH	Current Path
VARS	Variables
TVAR\$	Typed Variables
ORDER	Order Variables
CLVAR	Clear Variables
CLUSR	Clear Variables → CLVAR
DIR	Directory Structure
OBJECT INFO	
TYPE	Type
VTYPE	Variable Type
BYTES	Byte Size
NEWOB	New Object
MEM	Memory Available
PORTS & LIBRARIES	
ARCHIVE	Archive HOME
RESTORE	Restore HOME
FREE	Free RAM Card (obsolete)
MERGE	Merge RAM Card (obsolete)
PVARS	Port-Variables
PINIT	Port Initialize
LIBS	Libraries
ATTACH	Attach Library
DETACH	Detach Library
EDITORS	
EDIT	Edit
EDITB	Edit Best
VISIT	Edit Variable
VISITB	Edit Variable (best editor)
FILER	File Manager
EQW	Equation Writer

Modes & System	
NUMBER FORMAT	
STD	Standard Mode
FIX	Fix Mode
SCI	Scientific Mode
ENG	Engineering Mode
ANGLE MODE	
DEG	Degrees
RAD	Radians Mode
GRAD	Grads Mode

COORDINATE MODE	
RECT	Rectangular Mode
CYLIN	Cylindrical Mode
SPHERE	Spherical Mode
FONTS	
FONT6	System Font 6
FONT7	System Font 7
FONT8	System Font 8
→FONT	Set font
FONT→	Recall System Font
→MINIFONT	Set Minifont
MINIFONT→	Recall Minifont
UFL1→MINIF	UFL1 → Minifont
HEADER & LANGUAGE	
→HEADER	Set Header Size
HEADER→	Recall Header Size
→NDISP	Set Program Lines Shown
→LANGUAGE	Set Language
LANGUAGE→	Recall Language
SYSTEM	
OFF	Off
VERSION	Software Version
VER	CAS Version
WSLOG	Warmstart Log
ROMUPLOAD	ROM Upload (obsolete)
EASTER EGGS	
RULES	Credits
DEDICACE	CAS Dedication
MINEHUNT	Minehunt Game

Plotting	
2D PLOT TYPES	
FUNCTION	Function Plot Type
CONIC	Conic Plot Type
POLAR	Polar Plot Type
PARAMETRIC	Parametric Plot Type
TRUTH	Truth Plot Type
BAR	Bar Plot Type
HISTOGRAM	Histogram Plot Type
SCATTER	Scatter Plot Type
DIFFEQ	DiffEq Plot Type
SLOPEFIELD	Slope Field Plot Type

3D PLOT TYPES	
FAST3D	Fast 3D Plot Type
WIREFRAME	Wireframe Plot Type
PARSURFACE	Parametric Surface Plot
PCONTOUR	Contour Plot Type
YSLICE	Y-Slice Plot Type
GRIDMAP	Grid Map Plot Type
RANGES & AXES	
XRNG	x-Axis Display Range
YRNG	y-Axis Display Range
INDEP	Independent Variable
DEPND	Dependent Variable
RES	Resolution
AXES	Axes
ATICK	Axes Tick-Mark
3D VIEW VOLUME	
XVOL	X Volume Coordinates
YVOL	Y Volume Coordinates
ZVOL	Z Volume Coordinates
XXRNG	Input Plane X Range
YYRNG	Input Plane Y Range
EYEPT	Eye Point
NUMX	Number of X-Steps
NUMY	Number of Y-Steps
SCALING	
AUTO	Autoscale
CENTR	Center
SCALE	Scale Plot
SCALEH	Multiply Height
SCALEW	Multiply Width
*H	Multiply Height → SCALEH
*W	Multiply Width → SCALEW
PMIN	PICT Minimum
PMAX	PICT Maximum
PDIM	PICT Dimension
DRAW	
DRAW	Draw Plot
DRAX	Draw Axes
LABEL	Label Axes
ERASE	Erase PICT
PLOT	Plot Setup
PLOTADD	Add to Plot
GRAPH	Picture Environment (ali...
PICTURE	Picture Environment
PVIEW	PICT View
PICT	PICT
DRAW3DMATRIX	Draw 3D Matrix Plot

STATISTICAL PLOTS	
BARPLOT	Draw Bar Plot
HISTPLOT	Draw Histogram Plot
SCATRPLOT	Draw Scatter Plot
SCLE	Scale Sigma
Graphics (PICT / GROB)	
PRIMITIVES	
LINE	Draw Line
TLINE	Toggle Line
BOX	Box
ARC	Draw Arc
PIXELS	
PIXON	Pixel On
PIXOFF	Pixel Off
PIX?	Pixel On?
GRAPHICS OBJECTS	
→GROB	Object to GROB
GROB	GROB Delimiter
BLANK	Blank Graphics Object
GOR	Graphics OR
GXOR	Graphics Exclusive OR
GROBADD	Stack GROBs Vertically
ANIMATE	Animate
SUB	Subset
REPL	Replace
SIZE	Size
SCREEN	
→LCD	GROB to LCD
LCD→	LCD to GROB
COORDINATES	
C→PX	Complex to Pixel
PX→C	Pixel to Complex

Statistics	
DATA (ΣDAT)	
Σ+	Sigma Plus
Σ−	Sigma Minus
CLΣ	Clear Sigma
STOΣ	Store Sigma
RCLΣ	Recall Sigma
NΣ	Number of Rows

SUMMARY	
TOT	Total
MEAN	Mean
SDEV	Standard Deviation
VAR	Variance
PSDEV	Population Standard Deviation
PVAR	Population Variance
MAXΣ	Maximum Sigma
MINΣ	Minimum Sigma
BINS	Sort into Frequency Bins
SUMS	
ΣX	Sum of x-Values
ΣY	Sum of y-Values
ΣX^2	Sum of Squares of x-Values
ΣY^2	Sum of Squares of y-Values
ΣX*Y	Sum of x*y
COLUMNS	
XCOL	Independent Column
YCOL	Dependent Column
COLΣ	Column Sigma
REGRESSION	
LR	Linear Regression
CORR	Correlation
COV	Covariance
PCOV	Population Covariance
ΣLINE	Regression Formula
PREDX	Predicted x-Value
PREDY	Predicted y-Value
PREDV	Predicted y-Value → PREDY
CURVE FIT MODEL	
LINFIT	Linear Curve Fit
LOGFIT	Logarithmic Curve Fit
EXPFIT	Exponential Curve Fit
PWRFIT	Power Curve Fit
BESTFIT	Best-Fitting Model
DISTRIBUTIONS	
UTPC	Upper Chi-Square Distribution
UTPF	Upper F Distribution
UTPN	Upper Normal Distribution
UTPT	Upper Student's t Distribution
NDIST	Normal Distribution

Time & Alarms15

CLOCK	
DATE	Date
-DATE	Set Date
TIME	Time
-TIME	Set System Time
TICKS	Ticks
CLKADJ	Adjust System Clock
TSTR	Date and Time String
DATE ARITHMETIC	
DATE+	Date Addition
DDAYS	Delta Days
ALARMS	
STOALARM	Store Alarm
RCLALARM	Recall Alarm
DELALARM	Delete Alarm
FINDALARM	Find Alarm
ACK	Acknowledge Alarm
ACKALL	Acknowledge All Alarms

I/O & Communications33

KERMIT	
SEND	Send Object
RECV	Receive Object
REC�	Receive Renamed Object
KGET	Kermit Get
SERVER	Server Mode
FINISH	Finish Server Mode
PKT	Packet
KERRM	Kermit Error Message
XMODEM	
XSEND	XModem Send
XRECV	XModem Receive
XGET	XModem Get
XPUT	XModem Send
XSERV	XModem Server
SERIAL	
OPENIO	Open I/O Port
CLOSEIO	Close I/O Port
XMIT	Serial Transmit
SRECV	Serial Receive
BUFLEN	Buffer Length
SBRK	Serial Break
STIME	Serial Time-Out

PARAMETERS (IOPAR)	
BAUD	Baud Rate
PARITY	Parity
CKSM	Checksum
TRANSIO	I/O Translation
PRINTING	
PR1	Print Level 1
PRST	Print Stack
PRSTC	Print Stack (Compact)
PRVAR	Print Variable
PRLCD	Print LCD
CR	Carriage Right
DELAY	Delay
OLDPRT	Old Printer
APPLICATIONS	
STRM	StreamSmart

CAS Menus & Settings24

CAS MENUS	
MAIN	CAS Main Menu
ALGB	Algebra Menu
ARIT	Arithmetic Menu
CMLX	Complex Menu
CONSTANTS	Constants Menu
DIFF	Calculus Menu
EXP&LN	Exp & Ln Menu
INTEGER	Integer Menu
MATHS	Maths Menu
MATR	Matrix Menu
MODULAR	Modular Menu
POLYNOMIAL	Polynomial Menu
REWRITE	Rewrite Menu
SOLVER	Solver Menu
TESTS	Tests Menu
TRIGO	Trigonometry Menu
MENUXY	CAS Menu Range
HELP	
CASCMD	CAS Command List
HELP	CAS Help
SETTINGS & ASSUMPTIONS	
CASCFG	Reset CAS Settings
ASSUME	Assume Variable Domain
UNASSUME	Remove Assumptions
ADDTOREAL	Assume Real Names
TEVAL	Timed Evaluate