HEWLETT PACKARD

HP Solve Equation Library Application Card Owner's Manual

HP 82211A
for the HP 48SX

Comments: HP Solve Equation Library Application Card Owner's Manual

We welcome your evaluation of this manual. Your comments and suggestions help us improve our publications.

HP Solve Equation Library Application Card
Owner's Manual

Please circle a response for each of the statements below. You can use the **Comments** space to provide additional opinions.

1 = Strongly Agree
2 = Agree
3 = Neutral
4 = Disagree
5 = Strongly Disagree

■ The manual is well organized.	1 2 3 4 5
■ I can find the information I want.	1 2 3 4 5
■ The information in the manual is accurate.	1 2 3 4 5
■ I can easily understand the instructions.	1 2 3 4 5
■ The manual contains enough examples.	1 2 3 4 5
■ The examples are appropriate and helpful.	1 2 3 4 5
■ The layout and format are attractive and useful.	1 2 3 4 5
■ The illustrations are clear and helpful.	1 2 3 4 5
■ The amount of information is:	too much appropriate too little

■ The chapters I refer to most frequently are:

1 2 3 4 5 6 7 8 A B C D E Index

Comments: ____________________

____

____

____

____

____

____

Name: ____________________
Address: ____________________
City/State/Zip: ____________________
Occupation: ____________________
Phone: (______) ____________________

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 38 CORVALLIS, OR

POSTAGE WILL BE PAID BY ADDRESSEE

HEWLETT-PACKARD COMPANY
DOCUMENTATION DEPARTMENT
CORVALLIS DIVISION
1000 NE CIRCLE BLVD
CORVALLIS OR 97330-9973

HP Solve Equation Library Application Card

Owner's Manual

HEWLETT PACKARD

Edition 2 July 1990
Part Number 82211-90001

Notice

For warranty information for the card, see appendix A.

Data contained in the Periodic Table application is based on the "Periodic Table of the Elements" published by Sargent-Welch Scientific Company, a VWR Company, and is used by permission.

Data contained in the Constants Library application is based on "The 1986 CODATA Recommended Values of the Fundamental Physical Constants" published in the *Journal of Physical and Chemical Reference Data* for the National Bureau of Standards, and is used by permission.

Corvallis Division
1000 N.E. Circle Blvd.
Corvallis, OR 97330, U.S.A.

Printing History

Edition 1	February 1990
Edition 2	July 1990

Contents

1

Getting Started

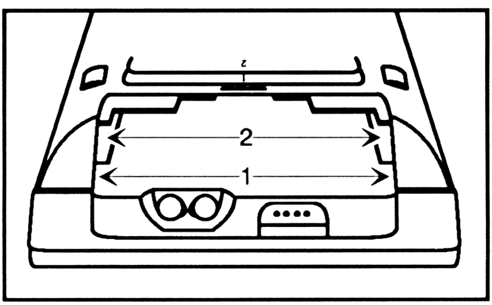
This chapter shows you how to install and start using the capabilities of the HP 82211A HP Solve Equation Library Application Card:

- Installing the card.
- Trying one of the card's applications.
- Selecting from a catalog of application items.
- Choosing unit options.
- Starting applications.

An example helps you try out one of the card's applications.

Installing and Removing the Card

The HP 48 has two *ports* for installing plug-in cards, designated port 1 and port 2. Port 1 is closest to the front of the calculator; port 2 is closest to the back. You can install your application card in either port. (See the *HP 48 Owner's Manual* for detailed information about installing and removing other types of plug-in cards.)



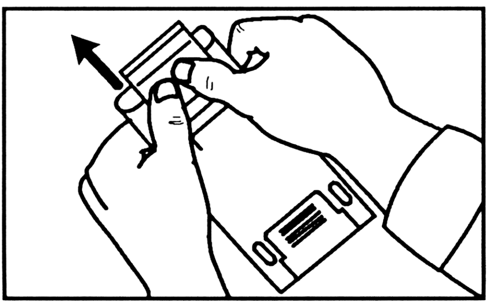
Caution

You must turn off the calculator while you're installing or removing a plug-in card. Otherwise, all of user memory could be erased.

Also, whenever you install or remove a card, the HP 48 executes a system halt, causing the contents of the stack to be lost.

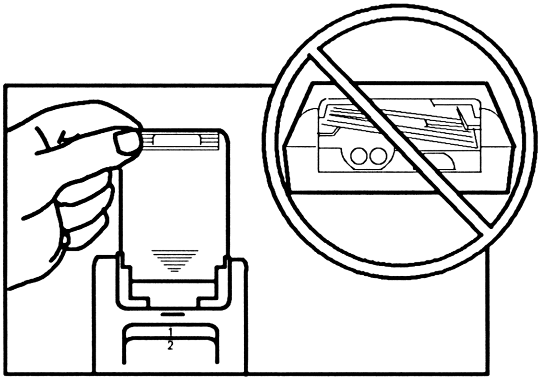
To install the application card, follow these steps:

1. Turn off the calculator. *Do not press* [ON] *until you've completed the installation procedures.*
2. Remove the port cover at the top of the calculator by pressing down against the grip area and then pushing in the direction shown below. Removing the cover exposes the two plug-in ports.



3. Select an empty port for the card—you can use either port.

4. Position the application card as shown. The triangular arrow on the card must point down, toward the calculator. Make sure the card is lined up properly with a port opening and not positioned half in one port and half in the other.



5. Slide the card firmly into the port until it stops. When you first feel resistance, the card has about 5 millimeters ($^1/_4$ inch) to go to be fully seated.
6. Replace the port cover by sliding it on until the latch engages.

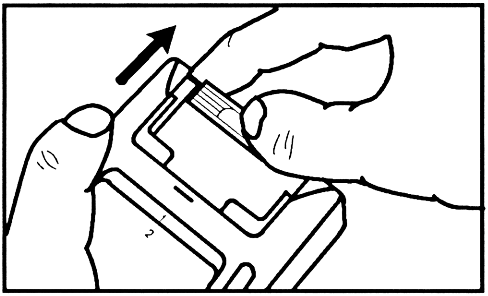
To remove the application card, follow these steps:

Caution If you have a *RAM card* that contains merged memory in your calculator and you want to remove it, you must free the merged memory before removal. Failure to do so will probably result in loss of data stored in user memory. See the *HP 48 Owner's Manual* for instructions.

---

1. Turn off the calculator. *Do not press* [ON] *until you've completed step 4.*
2. Remove the port cover.

3. Press against the card's grip as shown and slide the card out of the port.



4. Replace the port cover.

Trying an Application

The application card contains six main applications. (Technically, each application is called a *library object*.) Each application is described in a chapter in this manual.

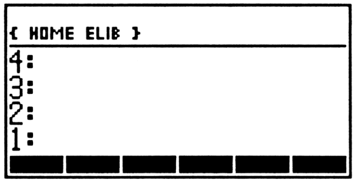
- Equation Library (chapters 2 and 3).
- Periodic Table (chapter 4).
- Constants Library (chapter 5).
- Finance (chapter 6).
- Multiple-Equation Solver (chapter 7).
- Utilities (chapter 8).

The following example illustrates how to start and use an application. (If you have trouble trying the example, press [ATTN], then start again at the second step.)

Example: Heat Conduction and Convection. Red Burns, a heat-transfer specialist, wants to design a 2-square-meter window for a heat chamber. The temperatures on the two sides of the window will be 120 °C and 35 °C. He'll use two layers of 0.5-cm glass separated by a layer of air. He knows that the glass has a heat conductivity of 1.02 W/m·K. The effective conductivity of air is 0.03 W/m·K. He assumes the convective coefficient of the air on each side of the window to be 5.5 $W/m^2 \cdot K$. What thickness of air does he need between the glass layers to limit the heat transfer rate to 150 watts?

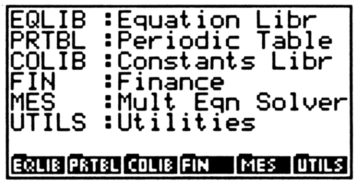
To prevent conflicts between variables for this problem and other variables already existing in your current directory, create another directory *ELIB* for this example. Then switch to the new directory.

[↱] [HOME]
['] ELIB
[↰] [MEMORY] CRDIR
[VAR] ELIB



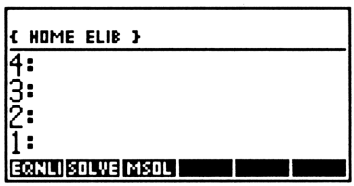
Get the LIBRARY menu. (Optionally, review the application names.)

[↰] [LIBRARY]
[↰] [REVIEW]



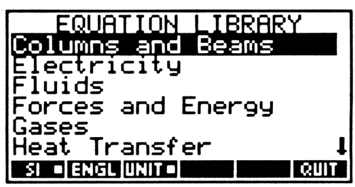
Select the Equation Library to get its main menu.

EQLIB



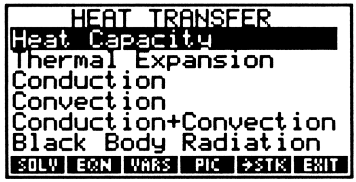
Start the application, then notice the catalog of available subjects. (If SI ■ and UNIT■ aren't flagged with small squares, press each of the corresponding menu keys once.)

EQNLI
(SI if needed)
(UNITS if needed)



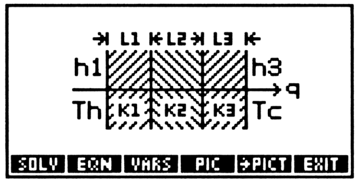
Move the highlight to the "Heat Transfer" subject and select it, then notice the catalog of available titles within this subject.

[▼] [▼] [▼] [▼] [▼]
[ENTER]



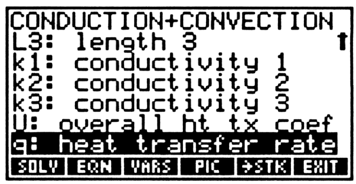
Move the highlight to the "Conduction + Convection" title, then look at the "picture" of the conduction-convection problem.

[▼] [▼] [▼] [▼]
PIC



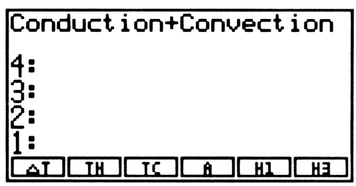
Examine the variables involved in the calculations.

VARS
[↰][▼]
[↰][▼]
[↰][▼]



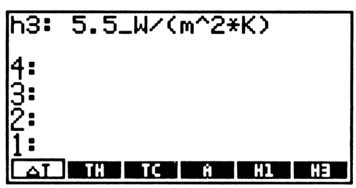
Begin solving the problem. Notice the menu labels corresponding to the variable names.

SOLV



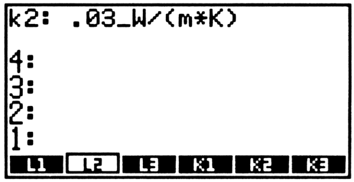
Enter values for the known variables. Notice that the menu labels turn black as you store the values.

120 TH
35 TC
2 A
5.5 [ENTER] [ENTER]
H1 H3



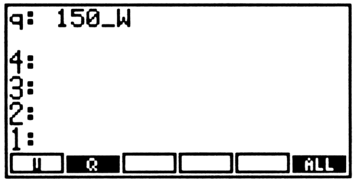
Change to the next page of variables and enter known values.

[NXT] .5 [ENTER] [ENTER]
L1 L3
1.02 [ENTER] [ENTER]
K1 K3
.03 K2



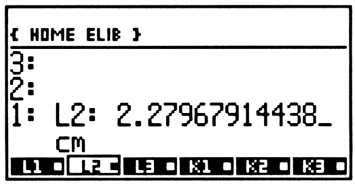
Change to the third page of variables and enter the last known value.

[NXT] 150 Q



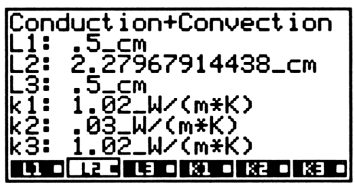
Change to the previous page of variables and solve for the unknown thickness.

[↰] [PREV]
[↰] L2



Review the values of the variables shown in the current menu.

[↰] [REVIEW]



Press [ATTN] to return to the stack display.

If you didn't obtain the same value for *L2*, one or more of the variables may have units different from the default units — SI ■ and UNIT■. (This is discussed later in this chapter under "Working with and without Units.")

Using Applications

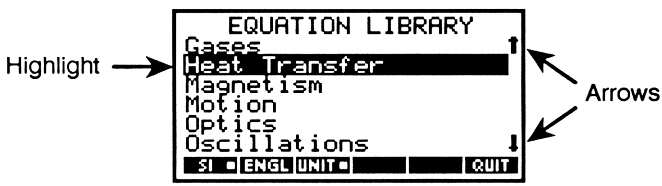
The following sections give general information about using the applications:

- Selecting from a catalog of application items.
- Choosing options for units.
- Using directories to separate problems.
- Starting applications different ways.

Viewing and Selecting in a Catalog

A *catalog* is a special environment for selecting from or viewing a collection of items.

Three of the applications (the Equation Library, the Periodic Table, and the Constants Library) may ask you to select one item from several choices. For example, in the Equation Library you can select one subject from 15 choices. The application displays the choices as a *catalog* of items — one line per item. A *highlight* marks one item. If a complete catalog doesn't fit in the display, *arrows* in the right margin indicate additional items above or below the displayed items.



You can move the highlight through the catalog to see what's included. Leave the highlight on the item you want to select. The following table lists the operations you can perform.

Operations in a Catalog

Key	Action
[▲]	Moves the highlight up through the catalog. (The highlight can wrap around to the bottom.) [↰][▲] jumps the highlight up one display. [↱][▲] jumps the highlight to the top of the catalog.
[▼]	Moves the highlight down through the catalog. (The highlight can wrap around to the top.) [↰][▼] jumps the highlight down one display. [↱][▼] jumps the highlight to the bottom of the catalog.
[α]. . .	Jumps the highlight to the next line starting with the alpha character you type, including lowercase letter or special character. (The highlight can wrap around to the top.)
Menu Keys	Perform their labeled functions, which depend upon the application.
[ENTER]	Performs a function that depends upon the application. For an item that's wider than the display, shows the complete item — press [ENTER] or [ATTN] to return to the catalog.
[ATTN]	Exits from the application.

Note

If you've set the Alpha Lock flag (flag −60) to set Alpha Lock mode, then each time you use [α] to move the highlight in a catalog, you must press [α] again to turn off alpha lock. The keystroke examples in this manual assume that flag −60 is clear (alpha lock is not automatically set).

The previous example shows how to select from a catalog.

Working with and without Units

Three applications provide options for choosing how units are used—the units *type* (SI or English) and the units *usage* (used or not used):

- The Equation Library lets you choose SI or English units, and it lets you choose to use or not use those units.
- The Periodic Table always has SI units, but it lets you choose to use or not use those units.
- The Constants Library lets you choose SI or English units, and it lets you choose to use or not use those units.

The *type* of units you use (SI or English) affects the values returned by these applications and affects the actual or implied units of your data. For example, the universal gas constant *R* is 8.31451 J/gmol·K in SI units and 10.7316 psi·ft^3/lbmol·°R in English units. (So the type of units may be important even if you don't use units.)

The *usage* of units (used or not used) affects whether units are appended to values or not—it determines whether you work with unit objects or real-number objects.

- If you use units in your calculations, conversion factors are automatically applied as needed, and your results are more easily interpreted.
- If you don't use units in your calculations, processes such as solving equations work faster, *no* conversion factors are applied, and *you* must ensure that variables and the values returned by applications use a compatible set of implied units. (See also "Using No Units with the Equation Library" in appendix B.)

You can set the type of units with the SI and ENGL menu keys for the two applications. The Units Type flag (flag 60) retains your current choice: clear for SI, set for English.

You can choose units used or not used with the UNITS menu key for the three applications. UNIT■ means units used, UNITS means units not used. The Units Usage flag (flag 61) retains your current choice: clear for units used, set for units not used.

Using Applications in Subdirectories

You can often simplify your work if you organize your computations into separate directories. This enables you to have fewer variables, equations, and programs to manage at one time. For more information about directories, see the "Directories" chapter in the *HP 48 Owner's Manual.*

Alternatives for Starting Applications

Each application includes a set of commands related to the application. These commands appear in the main menu for each application. You can start these applications — or execute any of the related commands — in several ways:

- Using the LIBRARY menu. (This method is used throughout this manual — see the previous example.)
- Typing the command name.
- Using a custom menu that includes the command name.
- In User mode, pressing a key that has the command name assigned to it.

Using the LIBRARY Menu. You can use the LIBRARY menu to access an application. The LIBRARY menu lists the six main applications in the card. To start an application or enter an application command, follow these steps:

1. Press [←][LIBRARY] to get the LIBRARY menu. The menu shows the six application labels. (Press [←][REVIEW] to see the full names.)

```
EQLIB :Equation Libr
PRTBL :Periodic Table
COLIB :Constants Libr
FIN   :Finance
MES   :Mult Eqn Solver
UTILS :Utilities

EQLIB PRTBL COLIB FIN MES UTILS
```

2. Press the menu key for the application you want, such as PRTBL. The menu changes to the main menu for the application.

3. Press the menu key for the command you want:
   - To start an application, press the left menu key, such as PERTB. (In the Utilities application, a game starts.)
   - To enter or execute an application command, press its menu key, such as MOLWT. (The command in the left menu label starts the application.)

The example earlier in this chapter shows how to start the Equation Library using the LIBRARY menu.

Typing the Command Name. You can enter an application command by typing the command in the command line. Each of the applications has a command that starts the application.

For example, you can type EQNLIB [ENTER] to start the Equation Library.

Using a Custom Menu. You can assign an application command to a menu key in the custom menu. Then you can use the command by pressing that menu key while the custom menu is displayed. (For more information see the "Customizing the Calculator" chapter in the *HP 48 Owner's Manual*.)

For example, you could put EQNLIB and PERTBL in a custom menu. Then you could start the Equation Library or Periodic Table by pressing [CST] and EQNLI or PERTB.

Using an User-Mode Key Assignment. You can assign an application command to any unshifted or shifted key. Then you can use the command by changing to User mode and pressing that key. (For more information see the "Customizing the Calculator" chapter in the *HP 48 Owner's Manual*.)

For example, you could assign MSOLVR to the [↱] [SOLVE] key. Then you could start the Multiple-Equation Solver by pressing [↰] [USR] [↱] [SOLVE].

2

Equation Library

The Equation Library is a collection of equations and commands that enable you to solve typical science and engineering problems. The library consists of more than 300 equations grouped into 15 technical subjects containing more than 100 problem titles. Each problem title contains one or more equations that help you solve that type of problem.

This chapter shows you how to do the following:

- Use the Equation Library from the keyboard.
- Get information about the equation sets.
- Use the underlying HP Solve and Multiple-Equation Solver applications.
- Use Equation Library commands.
- Access the Equation Library from a program.

A complete list of equation subjects and titles is contained in chapter 3, "Equation Reference."

When you solve problems using the Equation Library, the application uses the same numeric root finder that's used by the HP Solve application, which is built into the HP 48. This manual assumes you're familiar with the operation of the HP Solve application. If necessary, read the "HP Solve Application" chapter in the *HP 48 Owner's Manual*.

Solving a Problem with the Equation Library

Many scientific and technical problems involve determining numeric values for one or more unknown variables. Often these problems are solved using one or more equations that relate the unknown variables to known variables — but the task of solving for the unknown values is sometimes difficult.

The Equation Library provides a convenient method for solving more than 100 types of problems. Follow these steps for solving a problem using the Equation Library:

1. Get the LIBRARY menu and start the Equation Library by pressing [←][LIBRARY] EQLIB EQNLI.
2. Set the options you want by pressing the menu keys. (SI ■ indicates SI units, ENG■ indicates English units, UNIT■ indicates units used, and UNITS indicates units not used.)
3. Highlight the subject you want, then press [ENTER].
4. Highlight the title you want.
5. Optional — if you want to find out more about the equations in this set, press other keys as described under "Getting Information about Equation Sets" later in this chapter. You may want to do this to change the units of variables (see "Choosing Unit Options" later in this chapter).
6. Press SOLV to start solving the problem.
7. For each known variable, type its value and press the corresponding menu key. This stores the value. Press [NXT] to see additional variables and to enter additional values.
8. Press [←] followed by a menu key to solve for that variable. If the equation set contains more than one equation, you can press [←] ALL to solve for all remaining unknown variables — the variables not defined by you.

During this procedure, when you press SOLV to start solving a problem, the Equation Library uses one of two methods for finding solutions. The number of equations in the set determines the method for solving them:

- If the set contains only one equation, the HP Solve application built into the HP 48 finds solutions.

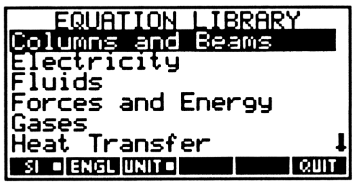
- If the set contains more than one equation, the Multiple-Equation Solver included in the application card finds solutions. (The Multiple-Equation Solver finds its solutions using the same root finder as the HP Solve application.)

If you expect to solve different types of problems, you should consider creating separate directories for each type. This minimizes conflicting uses of variable names and units among the sets of equations. (For more information, see “Avoiding Variable Conflicts” later in this chapter and “Using Applications in Subdirectories” in chapter 1.)

Example: Using the Equation Library. You estimate that Juergen Kickbacker kicked a ball at an elevation angle of 55 degrees. It landed 60 meters away. At what velocity did he kick it? What was the ball’s height at the halfway point? (Ignore the effects of drag on the ball.)

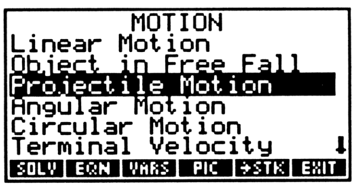
Get the LIBRARY menu and start the Equation Library. (If SI ■ and UNIT■ aren’t flagged with small squares, press each of the corresponding menu keys once.)

[→] [CLR]
[←] [LIBRARY]
EQLIB EQNLI
(SI if needed)
(UNITS if needed)



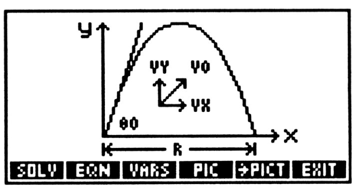
Select the “Motion” subject, then the “Projectile Motion” title.

[α] M [▼] [ENTER]
[▼] [▼]



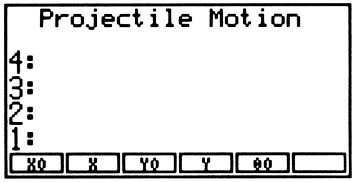
Look at the picture that describes the problem.

PIC



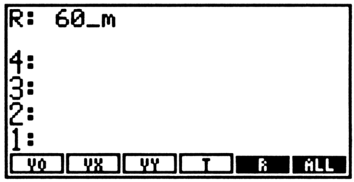
Start solving the problem.

SOLV



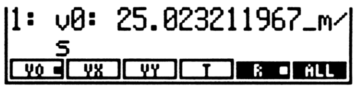
Enter known values. Define *x0* and *y0* as 0. (Notice that the menu labels turn black as you store values — this reveals that the Multiple-Equation Solver is being used to find your solutions.)

0 X0
0 Y0
55 θ0
[NXT] 60 R



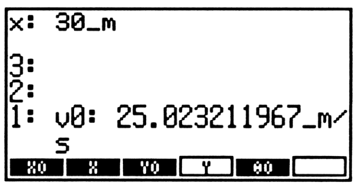
Solve for the velocity *v0*.

[←] V0



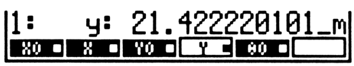
Recall the range, divide by 2 to get the halfway distance, and enter that as the *x*-coordinate.

[→] R ■
2 [÷] [NXT] [NXT] X



Solve for the height *y*. Notice that the Multiple-Equation Solver finds values for other variables as required to solve for the specified variable.

[←] Y



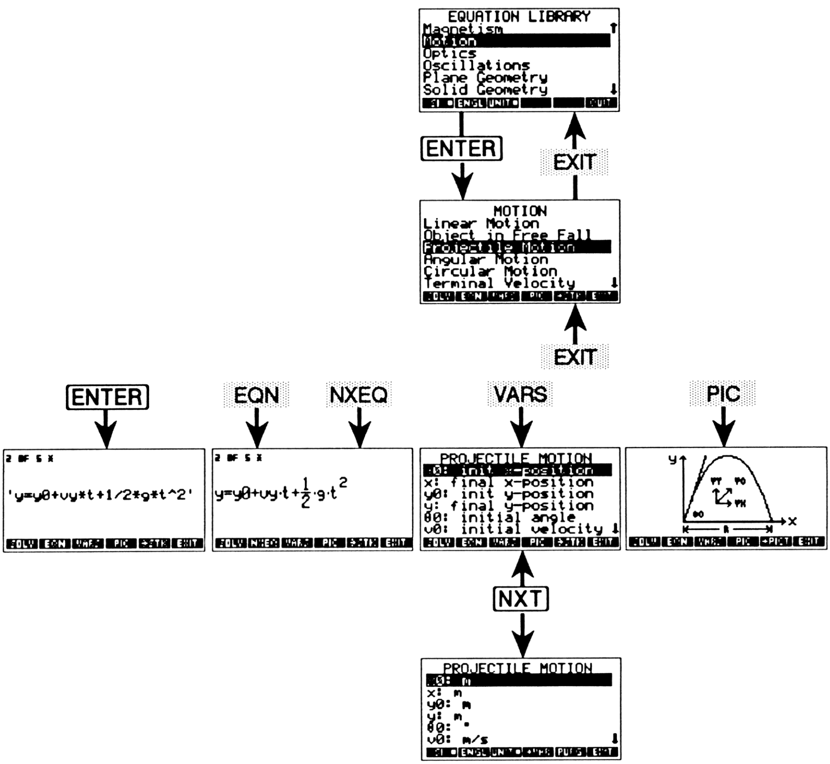
This example continues later in this chapter.

Getting Information about Equation Sets

When you select a subject and title in the Equation Library, you specify a set of one or more equations. You can get the following information about the equation set from the Equation Library catalogs:

- The equations themselves and the number of equations.
- The variables used and their units—you can also change the units.
- A picture of the physical situation (for most equation sets).

The following diagram shows how to get to the equation information. Note that you can switch among the information displays without returning to the catalog of titles. (See "Viewing and Selecting in a Catalog" in chapter 1 for details about using catalogs.)



Viewing Equations

After you select a subject and title, you can view each equation in the set. All equations have a *display form* — some equations also have a *calculation form*. The display form gives the equation in its basic form — the form you'd see in books. The calculation form includes adjustments for universal constants, unit manipulation, function substitution, variable ranges, and calculation speed. (If there is more than one equation, the number of equations appears in the upper-left corner of the equation display. If the equation has a calculation form, an * appears there.)

Operations for Viewing Equations

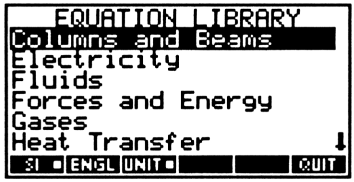
Key	Action	Example
EQN or NXEQ	Shows display form of current or next equation in EquationWriter format. (This may take from several seconds to a minute or more.)	$B=\frac{\mu 0 \cdot \mu r \cdot I}{2 \cdot \pi \cdot r}$
[ENTER]	Shows display form of current or next equation as an algebraic object. [ENTER] or [▼] shows the next equation, [▲] shows the previous.	'B=(μ0*μr*I)/(2*π*r)'
→STK	Shows calculation forms by putting a list containing the current set of equations in level 1 of the stack. (You can use OBJ→ to separate the list into individual equations.)	'B=IFTE(r<rw, CONST(μ0)*μr*I *r/(2*π*rw^2), CONST(μ0)*μr*I /(2*π*r))'

If you want to add, delete, or edit an equation, see "Changing the Equations" later in this chapter.

Example: Viewing Equations. Look at the equations for "Hooke's Law," included in the subject "Forces and Energy."

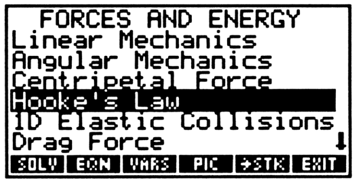
Get the library menu and start the Equation Library.

[←] [LIBRARY]
EQLIB
EQNLI



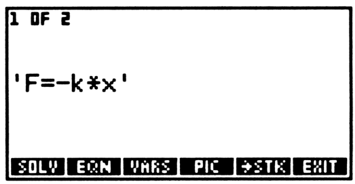
Select "Forces and Energy," then highlight "Hooke's Law."

[▼] [▼] [▼]
[ENTER]
[▼] [▼] [▼]



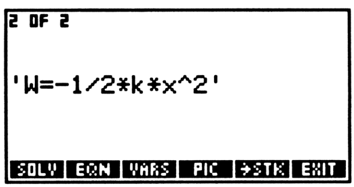
View the first equation as an algebraic object. Notice that there are two equations in the set, but no calculation form for this equation (no ✻ after the equation number).

[ENTER]



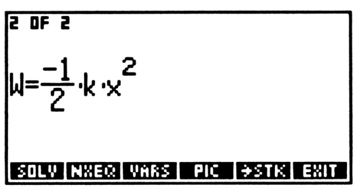
View the second equation.

[▼]



View this equation in EquationWriter format.

EQN



This example continues later in this chapter.

Viewing Variables and Selecting Units

After you select a subject and title, you can view the catalog of names, descriptions, and units for the variables in the equation set. You can also change to SI or English units and to units used or not used.

Operations in Variable Catalogs

Key	Action
VARS : Catalog of variable names and descriptions.	
[▲] [▼]	Moves the highlight up or down the catalog (also [◄] and [►]).
[α]...	Jumps the highlight to that alpha line.
[NXT]	Changes to the catalog of names and units (see below).
Menu Keys	Show other equation information.
[ENTER]	Shows the equation in algebraic form.
EXIT	Returns to the catalog of titles.
VARS [NXT]: Catalog of variable names and units.	
[▲] [▼]	Moves the highlight up or down the catalog (also [◄] and [►]).
[α]...	Jumps the highlight to that alpha line.
SI ■	Indicates SI units are active.
ENG■	Indicates English units are active.

Operations in Variable Catalogs (continued)

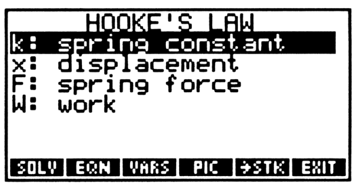
Key	Action
UNIT■	Indicates units are used.
UNITS	Indicates units are not used.
→VAR	Creates or changes all equation variables to have indicated unit type and usage — even if existing variables don't contain real or unit objects.
PURG	Purges all equation variables for this title in the current directory.
[NXT]	Changes to the catalog of names and descriptions (see above).
[ENTER]	Shows the equation in algebraic form.
EXIT	Returns to the catalog of titles.

If you want to change how units are used with the equations, press the corresponding SI , ENGL ,or UNITS menu key. For information about the effects of units, see "Choosing Unit Options" later in this chapter.

Example: Viewing Variables. This example continues from above, finding out about the equations for Hooke's Law.

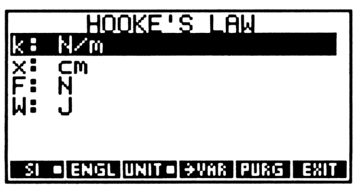
View the catalog of variable names and descriptions.

VARS



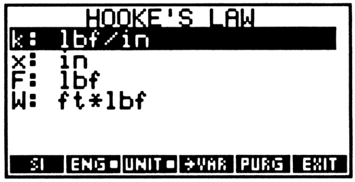
View the catalog of names and units.

[NXT]



Change the units to English—just to see the effect.

ENGL



Press SI and [NXT] to change the units back to SI and to return to the catalog of descriptions. This example continues later in this chapter.

Viewing the Picture

After you select a subject and title, you can view the picture of the problem—but only if the title has a picture.

To see the picture, press PIC . While the picture is displayed, you can do the following:

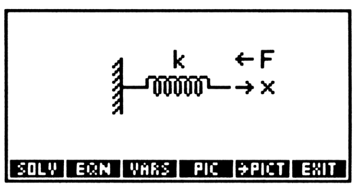
- Press →PICT to store the picture in *PICT*, the graphics memory—then you can use GRAPH ([←] [GRAPH]) to view the picture after leaving the Equation Library catalogs.
- Press the menu keys or [ENTER] to show other equation information.

For information about displaying and manipulating graphics objects, see the "More About Plotting and Graphics Objects" chapter in the *HP 48 Owner's Manual.*

Example: Viewing the Picture. This example continues from above, finding out about the equations for Hooke's Law.

View the picture for this set of equations.

PIC



Press [ATTN] to exit from the Equation Library.

Using the Solver

When you select a subject and a title in the Equation Library, you specify a set of one or more equations. Then, when you press SOLV ,you leave the Equation Library catalogs and start solving the equations you've selected.

When you press SOLV in the Equation Library, the application does the following:

- The set of equations is stored in the appropriate variable: *EQ* for one equation, *EQ* and *Mpar* for more than one equation. (*Mpar* is a reserved variable name used by the Multiple-Equation Solver.)
- Each variable is created and set to zero—but only if it doesn't already exist.
- Each variable's units are set to the conditions you specified—SI or English units, and units used or not used—*unless* the variable already exists and has units dimensionally *consistent* with what you specified.
- The appropriate solver is started: the HP Solve application for one equation, the Multiple-Equation Solver for more than one equation. (The number of equations in each set is given in its description in chapter 3.)

Using the Menu Keys

The actions of the unshifted and shifted variable menu keys for *both* solvers are identical. Notice that the Multiple-Equation Solver uses two forms of menu labels: black and white. The [NXT] key shows additional menu labels, if required. In addition, each solver has special menu keys, which are described in the following table. You can tell which solver is started by looking at the special menu labels. (Or you can check the title—the title for a library equation in the HP Solve application starts with EQ:.)

Actions for Solver Menu Keys

Operation	HP Solve Application	Multiple-Equation Solver
Store value	X	X X
Solve for value	← X	← X ← X
Recall value	→ X	→ X → X
Evaluate equation	EXPR=	
Next equation (if applicable)	NXEQ	
Undefine all		ALL
Solve for all		← ALL
Progress catalog		→ ALL
Set states		MUSE MCAL

Information about each solver is given later in this chapter.

Returning to the Solver

If you change to other menus while using the solver, you can resume the solving process where you left off:

- To resume the HP Solve application, press [→] [SOLVE].
- To resume the Multiple-Equation Solver, execute MSOLVR ([←] [LIBRARY] EQLIB MSOL or [←] [LIBRARY] MES MSOL).

To interrupt the solution process, press [ATTN].

The following three topics give general information about the three steps for solving equations.

Step 1: Entering Known Values

To enter a known value for a variable, type the value and press the menu key for the variable. For example, press 12.34 `X0` to enter 12.34 units as the value of *x0*.

The Equation Library interprets your input according to the unit choices you've set — usually the same as the current units in the variables. When you press a variable menu key to store a value, the Equation Library does the following:

- If you enter a number *without units*, the variable *value* is set to the number, and the current *units* are appended, if any.
- If you enter a number *with units* (a unit object), the input *value and units* are stored in the variable. You can do this to change the units for the variable. You should be careful that the units you use are consistent with the default units for the variable — see "Working with and without Units" in chapter 1.

Note

Certain Equation Library variables use units for "cyclic" measure. Such units separate into two types: "angular" units (degrees, radians, and grads) and "rotational" units (cycles and revolutions, as in Hz and rpm). Because these units are all dimensionless, the HP 48 incorrectly converts *between* these two types, though it correctly converts *within* one type. If you enter a unit object for a variable with "cyclic" units, you must use units of the same type as the default units for that variable ("angular" or "rotational"). For example, don't try to mix r/s and rpm.

If you want to change your unit choices *after* starting the solver, see "Choosing Unit Options" later in this chapter.

If you accidentally enter a value in the wrong variable and its menu label turns black, you should change the label back to white — unless you intend to enter a value for that variable too. To make the label white, press ◻, the menu key, and `MCAL` (on the last menu page).

Don't enter values that are algebraic objects containing other equation variables — unknown variables could be used as known variables. If such relationships do exist, include them as additional equations — see "Changing the Equations" later in this chapter.

Step 2: Supplying Guesses (Optional)

You can supply deliberate guesses for a variable you want to solve for. This can speed the solution process or focus on one of several possible solutions, such as for equations involving trigonometric or polynomial expressions. *You can minimize the chance of finding undesirable values by supplying guesses for variables with more than one possible solution.*

You can supply one guess or a list of two or three guesses. The HP 48 root finder uses the guess to define where it initially searches for a solution—for more information see the "HP Solve Application" chapter in the *HP 48 Owner's Manual.*

To supply *one* guess for a variable, type the value and press the menu key for the variable. *If you're using the Multiple-Equation Solver, see the special note below.*

To supply *two* or *three* guesses (a range) for a variable, press [⬅][{ }], type the values (one of them *must* include units that are consistent with the default units for the variable, if any), press [ENTER], then press the menu key for the variable. *If you're using the Multiple-Equation Solver, see the special note below.*

Note

If you're using the Multiple-Equation Solver, the menu label turns black when you store a guess this way. You *must* show that this is a *guess*, not a *known value*, in one of these ways:

- Solve for only this variable by pressing [⬅] followed by the menu key for the variable.
- Change the menu label to white by pressing ['], the menu key for the variable, and MCAL (on the last menu page).

An example of supplying guesses is included under the next topic.

Step 3: Finding the Solution

To solve for a variable, press [↰] followed by the menu key for the variable. If you're using the Multiple-Equation Solver, you can solve for all remaining variables that you didn't define as "known" — press [↰] ALL .

For example, press [↰] X0 to solve for *x0* — or press [↰] ALL to solve for *x0* and other unknown variables.

The solver automatically treats the current value of a variable as a guess if it needs to solve for the variable. Initially, all variables created by the Equation Library have values of 0 (with appropriate units). If a variable already existed, or if a variable was found during a previous solution, the variable may have a nonzero value.

See "Using the HP Solve Application" and "Using the Multiple-Equation Solver" later in this chapter.

Example: Supplying Guesses. Consider the example at the beginning of this chapter — Juergen kicking the ball. After finding that solution, you want to find the lower vertical angle that would have produced the same 60-meter range, assuming the initial velocity is the same. (The following steps assume you tried only the examples in this chapter.)

Clear the stack, then restart the Multiple-Equation Solver and review the current variable values.

[↱] [CLR]
([↰] [LIBRARY] EQLIB if needed)
MSOL
[↰] [REVIEW]

```
  Projectile Motion
x0: 0_m
x: 30_m
y0: 0_m
y: 21.422220101_m
θ0: 55_°
X0 ▪ X ▪ Y0 ▪ Y ▪ θ0 ▪
```

Recall and save the solved velocity as a user-defined value — otherwise, this value won't be "known."

[NXT] [↱] V0 ▪
V0 ▪

```
v0: 25.023211967_m/s
4:
3:
2:
1:
V0  VX  VY  T  R  ALL
```

Store the guesses 0° and 45° in *θ0* — otherwise, the original value will be found. Then review the current variable values again.

[◀] [{ }]
[◀] [UNITS] [NXT] [NXT] ANGL
0 [▶] [_] ° [SPC] 45 [ENTER]
[▶] [LAST MENU] [NXT] [NXT]
θ0
[◀] [REVIEW]

```
 Projectile Motion
x0: 0_m
x: 30_m
y0: 0_m
y: 21.422220101_m
θ0: { 0_° 45 }
 X0 | X | Y0 | Y | θ0 |
```

Solve for the lower angle.

[◀] θ0

```
1: θ0: 35.0000000003_°
 X0 | X | Y0 | Y | θ0 |
```

This example continues later in this chapter.

Using the HP Solve Application

The Equation Library starts the HP Solve application if the equation set contains only one equation.

When the Equation Library starts the HP Solve application, it first stores the library equation and a list of variables in *EQ*. The list of variables defines the HP Solve menu labels.

The menu labels for the variable keys are white — and they remain unchanged throughout the solution process. (This differs from the menu labels for the Multiple-Equation Solver.) In addition, you can use EXPR= to verify the solution and [◀] [REVIEW] to view variable values and units.

The *HP 48 Owner's Manual* describes how to use the HP Solve application — see the "HP Solve Application" chapter in that manual to get details about finding solutions, supplying guesses, and interpreting results.

See "Solving a Sequence of Problems" and later topics in *this* chapter for additional information about using the Equation Library.

Using the Multiple-Equation Solver

The Equation Library starts the Multiple-Equation Solver if the equation set contains more than one equation.

When the Equation Library starts the Multiple-Equation Solver, it first stores a list of the equation set in *EQ* and stores the equation set plus additional information in *Mpar*.

You can get other information about the Multiple-Equation Solver in chapter 7, "Multiple-Equation Solver."

Controlling Variables

Because a solution involves many equations and many variables, the Multiple-Equation Solver must keep track of variables that are user-defined and not defined—those it can't change and those it can. In addition, it keeps track of variables that it used or found during the last solution process.

The menu labels indicate the states of the variables. They're automatically adjusted as you store values and solve for variables. You can check that variables have proper states when you supply guesses and find solutions.

Interpreting the Menu. The menu labels for the variable keys are white at first—they change during the solution process as described below. (This differs from the behavior of the HP Solve menu labels.)

Meanings of Menu Labels

Label	Meaning
XO	Value *x0* not defined by you and not used in the last solution — it can change in the next solution.
XO ■	Value *x0* not defined by you, but found in the last solution — it can change in the next solution.
XO	Value *x0* defined by you, but not used in the last solution — it can't change in the next solution (unless you solve for only this variable).
XO ■	Value *x0* defined by you and used in the last solution — it can't change in the next solution (unless you solve for only this variable).

Notice that ■ marks the variables that were used in the last solution — their values are compatible with each other. Other variables may *not* have compatible values because they weren't involved in the solution.

If you move to another directory or change *Mpar*, the solver menu changes accordingly. If a valid *Mpar* exists, the menu changes to match the current *Mpar*. If a valid *Mpar* doesn't exist, the MTH menu is displayed.

Changing Label Colors and Variable States. The menu label colors indicate the states of variables — they normally change to the proper state automatically. You can change the state of one or more variables using the MUSER and MCALC commands (MUSE and MCAL in the last page of the solver menu). You may need to do this while supplying guesses or altering the problem.

To change a variable to user-defined (black menu label), press ['], press the menu key for the variable, then press MUSE — or recall and store its value ([↱] XO XO).

To change a variable to not defined (white menu label), press ['], press the menu key for the variable, then press MCAL — "calculated" value. You can change *all* variables to not defined by pressing ALL .

To change the states of several variables, press [↰][{ }], press each variable key, press [ENTER], then press MUSE or MCAL .

If you use [STO] to enter a value for a variable, the variable state doesn't change, default units aren't appended to the value, and the relationships marked by ■ become invalid.

Interpreting Results

The Multiple-Equation Solver solves for variables by repeatedly looking through the set of equations for one that contains only one variable that's "unknown" (not user-defined and not found by the solver during this solution) — then it uses the HP 48 root finder to find that value. It continues eliminating "unknown" variables until it solves for the variable you specified — or until it can't solve for any more variables. Each time the Multiple-Equation Solver starts solving for a variable, only the variables with black menu labels are "known."

Checking Progress. During the solution process, the Multiple-Equation Solver shows the variable it's currently solving for. It also shows the type of root found by the HP 48 root finder (zero, sign reversal, or extremum) — or the problem if no root is found (bad guesses or constant). (You can watch the iterations if you press any key except [ATTN] during the root-finding process. For more information about the root finder, see the "HP Solve Application" chapter in the *HP 48 Owner's Manual.*)

The following messages indicate errors in the problem setup:

- `Bad Guess(es)`. Units may be missing or inconsistent for a variable. For a list of guesses, at least one of the list elements must have consistent units.
- `Too Many Unknowns`. The solver eventually encountered only equations having at least two unknowns. Either enter other known values, or change the set of equations — whichever is appropriate for your problem. See "Changing the Equations" later in this chapter.
- `Constant?` The initial value of a variable may be leading the root finder in the wrong direction. Supply a guess in the opposite direction from a critical value — if negative values are valid, try one.

Checking Solutions. The variables having a ▪ mark in their menu labels are related for the most-recent solution — they form a compatible set of values satisfying the equations used. The values of any variables *without* marks may not satisfy the equations because those variables weren't involved in the solution process.

If any solutions seem improper, check for the following problems:

- Wrong units. A known or found variable may have units different from those you assumed.
- No units. If you're not using units, your implied units may not be compatible among your variables or with the implied units of constants or functions. The current angle mode sets the implied units for angles.
- Multiple roots. An equation may have multiple roots, and the solver may have found an inappropriate one. Supply a guess for the variable to focus the search in the appropriate range.
- Wrong variable states. A known or unknown variable may not have the appropriate state. A known variable should have a black menu label, and an unknown variable should have a white label.
- Inconsistent conditions. If you enter values that are mathematically inconsistent for the equations, the application may give results that satisfy *some* equations, but not *all*. This includes over-specifying the problem, for which you enter values for more variables than needed to define a physically realizable problem — the extra values may create an impossible or illogical problem. (The solution satisfies the equations the solver used, but the solver doesn't try to verify that the solution satisfies *all* of the equations.)
- Not related. A variable may not be involved in the solution (no ▪ in its menu label), so it's not compatible with the variables that *were* involved.
- Wrong direction. The initial value of a variable may be leading the root finder in the wrong direction. Supply a guess in the opposite direction from a critical value — if negative values are valid, try one.

You can evaluate solutions by using the HP Solve EXPR= menu key to calculate the values of the left and right sides of each equation. You can run the HP Solve application ([→] [SOLVE]) using the list of equations in *EQ* and step through the equations one at a time — see "Using the HP Solve Application with Several Equations" later in this chapter.

You can also plot the relationship between two variables as a check of your solution — see "Plotting Equations" in chapter 7.

Checking the Process. The Multiple-Equation Solver provides a "progress catalog" that describes the last solution process. This catalog lists variables, equations, and values in the order they were used or found. To view the progress catalog, press [↱] ALL.

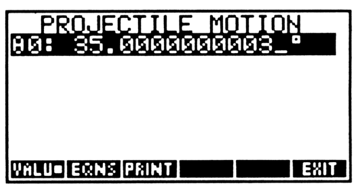
Operations in the Progress Catalog

Key	Action
[▲] [▼]	Moves the highlight up or down the catalog (also [↰] and [↱]).
[α]. . .	Jumps the highlight to that alpha line.
VALU■	Shows the values found.
EQN■	Shows the equations used to solve for the variables.
PRINT	Prints the progress catalog (values and equations).
EXIT	Returns to the Multiple-Equation Solver menu.
[ENTER]	Shows a wide (. . .) item completely — press [ENTER] or [ATTN] to return to the catalog.
[ATTN]	Returns to the Multiple-Equation Solver menu.

Example: Checking the Progress Catalog. This example continues from above — Juergen kicking the ball. Check the progress catalog for the last solution.

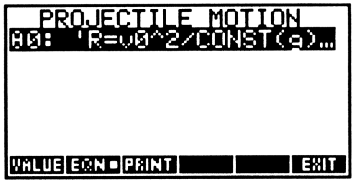
View all found values in the progress catalog.

[NXT]
[↱] ALL



View all equations used in the solution.

EQNS



Press EXIT to exit from the catalog. This example continues later in this chapter.

Altering the Problem

You may occasionally want to re-solve a problem using different known conditions. To do this, follow these steps:

1. Enter any new, known values.
2. Check that all "known" variables have black menu labels and all "unknown" variables have white menu labels—*except*, if you intend to solve for only one variable, its state doesn't matter. If necessary, change variable states—see "Controlling Variables" earlier in this chapter.
3. Solve for one or more unknown variables.

See the example under "Step 2: Supplying Guesses" earlier in this chapter.

Using the HP Solve Application with Several Equations

After starting the Multiple-Equation Solver, you can start the HP Solve application ([↱] [SOLVE]) to work with the set of equations already stored in *EQ*. This enables you to do the following:

- Solve one equation at a time. Press NXEQ to change from one equation to the next.
- Verify the solution by evaluating the left and right sides of the equation. Press EXPR= to put the left and right values on the stack.

Solving a Sequence of Problems

Often you can separate a complex problem into a sequence of simpler problems. For example, you might be able to consider a compound optical structure as a series of individual lenses and reflectors. Then you could use the Equation Library to find a solution for the first element, and apply that result to the next element.

The Equation Library usually uses compatible variable names and units for related sets of equations. The results of one solution may be preserved and be directly applicable to the next set of equations.

Choosing Unit Options

Each equation set has two types of default units: SI and English. You can choose either type. You can also choose to use or not use units. For certain sets of equations that use constants from the Constants Library, the type of units determines the numeric values of the constants — so the type of units may be important even if you don't use units. (See "Working with and without Units" in chapter 1.)

If you choose not to use units, no unit conversions are performed — *you* must ensure that all variables use a compatible set of implied units. (See also "Using No Units with the Equation Library" in appendix B.)

If you want to change your unit options *after* starting the solver — if variables already exist with different (but consistent) units — just changing the unit options is not sufficient. You must impose your new choices. You can use →VAR in the unit catalog. This creates or changes equation variables so they all have units of your new choices — SI or English units, and units used or units not used.

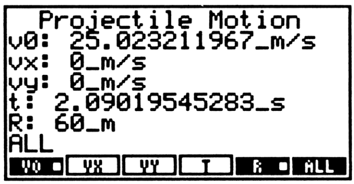
1. Restart the Equation Library and select the subject and title.
2. Press VARS [NXT] to get the units catalog.
3. Select your new unit options.
4. Press →VAR to force the variables to the new unit options.

Note If you choose *not* to use units, your results for many equation sets may seem incorrect. This can be caused by units that aren't compatible among the variables or with the implied units of constants or functions. See "Using No Units with the Equation Library" in appendix B.

Example: Changing Units. This example continues from above— Juergen kicking the ball. The earlier solutions involved units—you could have used velocities in km/hr instead of m/s. Change the setup to use no units (SI units implied). (Note that you must still specify SI or English units because the value of acceleration due to gravity depends upon this option.)

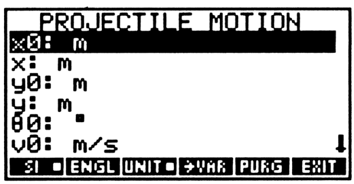
Review variable values to see that units are included.

[←] [REVIEW]



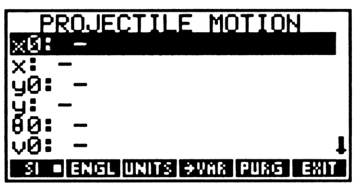
Get the unit catalog for these equations.

[←] [LIBRARY] EQLIB EQNLI
[α] M [▼] [ENTER]
[▼] [▼]
VARS [NXT]



Set the unit options to SI units (SI ▪) and units not used (UNITS). Then force the variables to have the new unit options.

UNIT▪
→VAR



Return to the solver menu and review the variable values. Note that the units are gone.

[ATTN]
[↱][LAST MENU]
[↰][REVIEW]

```
  Projectile Motion
v0: 25.023211967
vx: 0
vy: 0
t: 2.09019545283
R: 60
ALL
 V0 ■ | VX | VY | T | R ■ | ALL
```

Changing the Equations

Occasionally you may want to add, delete, or edit an equation in a set of equations from the Equation Library:

- You might add an equation if all equations in the set have at least two "unknown" variables for your problem. You can often derive an equation by combining other equations algebraically to eliminate "unknown" variables.
- You might want to impose additional constraints on variables by relating them to each other or by making them constant without entering values.

You can't change the application itself—the contents of the application card are permanent—but you *can* change the equations in *EQ*. To change *EQ* and run the Multiple-Equation Solver, follow these steps:

1. Select the subject and title from the Equation Library.
2. Edit *EQ* to create the desired list of equations:
   - If you're viewing an Equation Library catalog (you haven't started the solver yet), you can press →STK QUIT, edit the equations, then store the revised list of equations in *EQ* ([↰][SOLVE] STEQ or NEW).
   - If you've started the solver, you can press [↰][SOLVE] EDEQ, then edit the equations.
3. Get the Multiple-Equation Solver menu ([↰][LIBRARY] MES).
4. Use MINIT (MINIT) to update *Mpar* (the multiple-equation variable) according to the new *EQ*.
5. Press MSOL to start the Multiple-Equation Solver.

Avoiding Variable Conflicts

If you use several sets of equations from the Equation Library, you may occasionally find the same variable name used in more than one set. This can cause problems or confusion because each set assumes the variable contains *its* value. You can avoid this conflict by using separate directories for each problem. Note that a conflict can be caused by a variable in the current directory or in a higher directory.

If a set of equations needs a variable whose name already exists in the current directory (or in a higher directory, such as *HOME*), the units for the new and existing variables are compared for *dimensional consistency*. They're *consistent* if they describe the same type of measurement, such as velocity or temperature, or if there are no units for both variables — they need not be identical units. Otherwise, the units are inconsistent.

The Equation Library uses existing *values* and existing *units* as often as possible. This makes it possible to share data among equation sets. Variables are found and recalled "globally" (from current and higher directories) and are stored "locally" (in the current directory). Specifically, the application responds this way just before it starts the solver:

- If the variable exists in the current directory, then
  - If its units are *consistent* with the units of the new variable, the *old value* and *old units* are used by the equations.
  - If its units are *not consistent* with the units of the new variable, the Equation Library changes the variable to the *old value* and *new units*.
- If the variable exists in a higher directory, then
  - If its units are *consistent* with the units of the new variable, the *old value* and *old units* are used by the equations. But if the variable needs to change, a new variable is created in the current directory with the new value (and units, if specified).
  - If its units are *not consistent* with the units of the new variable, the Equation Library creates a new variable in the current directory with the *old value* and *new units*.
- If the variable doesn't exist, the Equation Library creates it in the current directory with zero value and specified units, if any.

Using Equation Library Commands

The Equation Library includes three commands that you can execute from the menu, in the command line, and in programs. You can view the command names by pressing [←][LIBRARY] EQLIB [←][REVIEW].

Equation Library Commands

Key	Programmable Command	Description
EQNLI	EQNLIB	Starts the Equation Library. It doesn't affect the stack.
SOLVE	SOLVEQN	Sets up and starts the appropriate solver for the specified set of equations (subject and title), bypassing the Equation Library catalogs. It sets *EQ* (and *Mpar* for more than one equation), chooses the units type according to the Units Type flag (flag 60: SI if clear, English if set), chooses to use or not use units according to the Units Usage flag (flag 61: used if clear, not used if set), and starts the appropriate solver. It takes the subject number from level 3, the title from level 2, and a "PICT" option from level 1, and it returns nothing. (Subject and title numbers are listed at the start of chapter 3. If the "PICT" option is 0, *PICT* is not affected — otherwise, the equation picture is copied into *PICT*.)

Equation Library Commands (continued)

Key	Programmable Command	Description
MSOL	MSOLVR	Gets the Multiple-Equation Solver variable menu for the set of equations defined by *Mpar*. It doesn't affect the stack. (Note that *Mpar* is not automatically updated when you revise *EQ*.)

Programming with the Equation Library

You can access the Equation Library from a program by using the Equation Library commands:

- Execute SOLVEQN to return control to the keyboard and solve the set of equations specified in levels 3 and 2 (subject and title numbers — see chapter 3) — the "PICT" option is in level 1. The current states of the Units Type and Units Usage flags (flags 60 and 61) define how units are used.
- Execute MSOLVR to return control to the keyboard and solve the set of equations specified by *Mpar*.

If you want a program to continue after you find the solution from the keyboard, you should include a HALT command after the Equation Library command. Then after you find the solution, you can press [⬑] [CONT] to resume program execution.

Example: Equation Library in a Program. Red Burns wants quicker access to the "Conduction + Convection" equations under the "Heat Transfer" subject. He wants to use SI units each time.

From the first table in chapter 3, the equations are under subject 6 and title 5. Enter the following program that starts the solver for that set of equations, then assign it to a custom menu key.

```
« 60 CF 61 CF 6 5 0 SOLVEQN »
```

Clear the stack and get the main Equation Library menu.

[↱][CLR]
[↰][LIBRARY] EQLIB

Enter the program.

[↰][« »]
60 [SPC] CF [SPC] 61 [SPC] CF [SPC]
6 [SPC] 5 [SPC] 0 SOLVE
[ENTER]

Store the program in variable *CNC*.

['] CNC [STO]

Create and display a custom menu containing this function.

[↰][{ }] [↰][{ }]
[↱][" "] C [α][+] C [▶] CNC [ENTER]
[↱][MODES] MENU

3

Equation Reference

The Equation Library consists of 15 subjects (corresponding to the sections in the table below) and more than 100 titles. Each subject and title has a number that you can use with SOLVEQN to specify the set of equations. (The page number for each title is shown in parentheses.)

Subjects and Titles

Subjects and Titles (continued)

Subjects and Titles (continued)

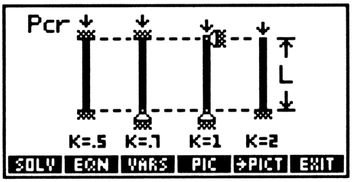
Columns and Beams

Variable Names and Descriptions

ε	Eccentricity (offset) of load
σcr	Critical stress
σmax	Maximum stress
Θ	Slope at x
A	Cross-sectional area
a	Distance to point load
c	Distance to edge fiber (Eccentric Columns), or Distance to applied moment (beams)
E	Modulus of elasticity
I	Moment of inertia
K	Effective length factor of column
L	Length of column or beam
M	Applied moment
Mx	Internal bending moment at x
P	Load (Eccentric Columns), or Point load (beams)
Pcr	Critical load
r	Radius of gyration
V	Shear force at x
w	Distributed load
x	Distance along beam
y	Deflection at x

Elastic Buckling

These four equations analyze axial forces applied to a slender column, causing it to bend and deflect laterally — failure is caused by buckling rather than compression. The boundary conditions at both ends of the column have a major impact on the buckling characteristics of the column. The effective length factor K defines the boundary conditions — whether ends are fixed, pinned, or free. The equations assume the column is slender — $K{\cdot}L/r$ is greater than about 100, depending upon the yield strength of the material.



Equations:

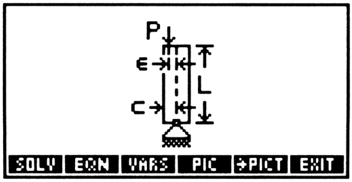
$$Pcr = \frac{\pi^2 \cdot E \cdot A}{\left(\frac{K \cdot L}{r}\right)^2} \qquad Pcr = \frac{\pi^2 \cdot E \cdot I}{(K \cdot L)^2} \qquad \sigma cr = \frac{Pcr}{A}$$

$$r = \sqrt{\frac{I}{A}}$$

Example: Given L=7.3152_m, r=4.1148_cm, E=199947961.502_kPa, A=53.0967_cm^2, K=0.7, I=8990598.7930_mm^4, solves Pcr=676.6019_kN, σcr=127428.2444_kPa.

Eccentric Columns

These two equations analyze the maximum stress induced by an eccentric, axial load applied at an offset from the axis of a slender column. See "Elastic Buckling" for the meaning of the effective length factor K. The equations assume the column is slender — $K{\cdot}L/r$ is greater than about 100, depending upon the yield strength of the material.



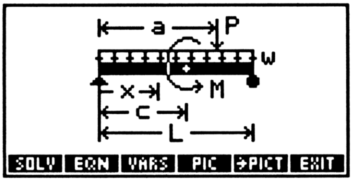
Equations:

$$\sigma max = \frac{P}{A} \cdot \left(1 + \frac{\varepsilon \cdot c}{r^2} \cdot \left(\frac{1}{\cos\left(\frac{K \cdot L}{2 \cdot r} \cdot \sqrt{\frac{P}{E \cdot A}} \right)} \right) \right) \qquad r = \sqrt{\frac{I}{A}}$$

Example: Given L = 6.6542_m, A = 187.9351_cm^2, r = 8.4836_cm, E = 206842718.795_kPa, I = 135259652.16_mm^4, K = 1, P = 1908.2571_kN, c = 15.24_cm, ε = 1.1806_cm, c = 15.24_cm, solves σmax = 140853.0970_kPa.

Simple Deflection

This equation analyzes the deflection of a simply supported beam. The equation uses superposition to combine the effect of three types of loads: a point load P, an applied moment M, and a distributed load w. The behavior at x is calculated differently depending upon the location of x relative to the point load and moment. Applied loads are positive downward, the applied moment is positive counterclockwise (right-hand rule), deflection is positive upward, slope is positive counterclockwise, internal bending moment is positive counterclockwise on the left-hand part, and shear force is positive downward on the left-hand part. The use of the equation is limited to a beam with one load of each type — you can address multiple loads by solving the problem for each load and using superposition. This equation assumes deflections are small (they don't alter the geometry of the problem), stresses are in the elastic region, and the beam has a constant cross section.



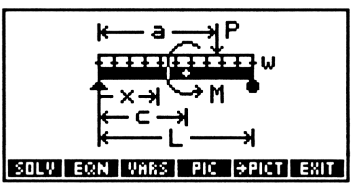
Equation:

$$y = \frac{P \cdot (L - a) \cdot x}{6 \cdot L \cdot E \cdot I} \cdot \left(x^2 + (L - a)^2 - L^2\right) - \frac{M \cdot x}{E \cdot I} \cdot \left(c - \frac{x^2}{6 \cdot L} - \frac{L}{3} - \frac{c^2}{2 \cdot L}\right) - \frac{w \cdot x}{24 \cdot E \cdot I} \cdot \left(L^3 + x^2 \cdot (x - 2 \cdot L)\right)$$

Example: Given L=20_ft, E=29000000_psi, I=40_in^4, a=10_ft, P=674.427_lbf, c=17_ft, M=3687.81_ft*lbf, w=102.783_lbf/ft, x=9_ft, solves y= −.6005_in.

Simple Slope

This equation analyzes the slope of a simply supported beam. See "Simple Deflection" for sign conventions, limitations, and assumptions.



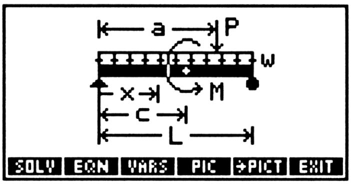
Equation:

$$\Theta = \frac{P \cdot (L - a)}{6 \cdot L \cdot E \cdot I} \cdot \left(3 \cdot x^2 + (L - a)^2 - L^2\right) - \frac{M}{E \cdot I} \cdot \left(c - \frac{x^2}{2 \cdot L} - \frac{L}{3} - \frac{c^2}{2 \cdot L}\right) - \frac{w}{24 \cdot E \cdot I} \cdot \left(L^3 + x^2 \cdot (4 \cdot x - 6 \cdot L)\right)$$

Example: Given L=20_ft, E=29000000_psi, I=40_in^4, a=10_ft, P=674.427_lbf, c=17_ft, M=3687.81_ft*lbf, w=102.783_lbf/ft, x=9_ft, solves Θ= –.0876_°.

Simple Moment

This equation analyzes the internal bending moment of a simply supported beam. See "Simple Deflection" for sign conventions, limitations, and assumptions.



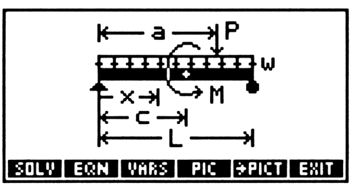
Equation:

$$Mx = \frac{P \cdot (L - a) \cdot x}{L} + \frac{M \cdot x}{L} + \frac{w \cdot x}{2} \cdot (L - x)$$

Example: Given L=20_ft, a=10_ft, P=674.427_lbf, c=17_ft, M=3687.81_ft*lbf, w=102.783_lbf/ft, x=9_ft, solves Mx=9782.1945_ft*lbf.

Simple Shear

This equation analyzes the shear of a simply supported beam. See "Simple Deflection" for sign conventions, limitations, and assumptions.



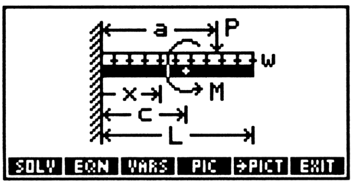
Equation:

$$V = \frac{P \cdot (L - a)}{L} + \frac{M}{L} + \frac{w}{2} \cdot (L - 2 \cdot x)$$

Example: Given L=20_ft, a=10_ft, P=674.427_lbf, M=3687.81_ft*lbf, w=102.783_lbf/ft, x=9_ft, solves V=624.387_lbf.

Cantilever Deflection

This equation analyzes the deflection of a cantilever beam. See "Simple Deflection" for sign conventions, limitations, and assumptions.



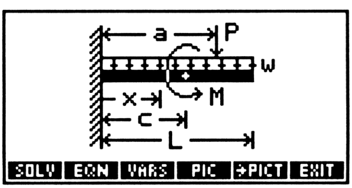
Equation:

$$y = \frac{P \cdot x^2}{6 \cdot E \cdot I} \cdot (x - 3 \cdot a) + \frac{M \cdot x^2}{2 \cdot E \cdot I} - \frac{w \cdot x^2}{24 \cdot E \cdot I} \cdot \left(6 \cdot L^2 - 4 \cdot L \cdot x + x^2\right)$$

Example: Given L=10_ft, E=29000000_psi, I=15_in^4, P=500_lbf, M=800_ft*lbf, a=3_ft, c=6_ft, w=100_lbf/ft, x=8_ft, solves y= –.3316_in.

Cantilever Slope

This equation analyzes the slope of a cantilever beam. See "Simple Deflection" for sign conventions, limitations, and assumptions.



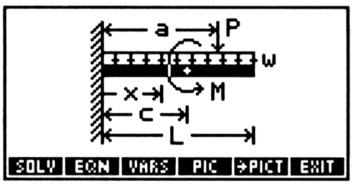
Equation:

$$\Theta = \frac{P \cdot x}{2 \cdot E \cdot I} \cdot (x - 2 \cdot a) + \frac{M \cdot x}{E \cdot I} - \frac{w \cdot x}{6 \cdot E \cdot I} \cdot \left(3 \cdot L^2 - 3 \cdot L \cdot x + x^2\right)$$

Example: Given L=10_ft, E=29000000_psi, I=15_in^4, P=500_lbf, M=800_ft*lbf, a=3_ft, c=6_ft, w=100_lbf/ft, x=8_ft, solves Θ= –.2652_°.

Cantilever Moment

This equation analyzes the internal bending moment of a cantilever beam. See "Simple Deflection" for sign conventions, limitations, and assumptions.



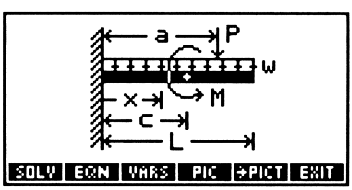
Equation:

$$Mx = P \cdot (x - a) + M - \frac{w}{2} \cdot \left(L^2 - 2 \cdot L \cdot x + x^2\right)$$

Example: Given L=10_ft, P=500_lbf, M=800_ft*lbf, a=3_ft, c=6_ft, w=100_lbf/ft, x=8_ft, solves Mx= –200_ft*lbf.

Cantilever Shear

This equation analyzes the shear of a cantilever beam. See "Simple Deflection" for sign conventions, limitations, and assumptions.



Equation:

$$V = P + w \cdot (L - x)$$

Example: Given $L=10\_ft$, $P=500\_lbf$, $a=3\_ft$, $x=8\_ft$, $w=100\_lbf/ft$, solves $V=200\_lbf$.

Electricity

Variable Names and Descriptions

εr	Relative permittivity
μr	Relative permeability
ω	Angular frequency
ω0	Resonant angular frequency
ϕ	Phase angle
ϕp,ϕs	Parallel and series phase angles
ρ	Resistivity
ΔI	Current change
Δt	Time change
ΔV	Voltage change
A	Wire cross-section area (Wire Resistance), or Solenoid cross-section area (Solenoid Inductance), or Plate area (Plate Capacitor)
C,C1,C2	Capacitance
Cp,Cs	Parallel and series capacitances
d	Plate separation
E	Energy
F	Force between charges
f	Frequency
f0	Resonant frequency

Variable Names and Descriptions (continued)

I	Current, or Total current (Current Divider)
I1	Current in R1
Imax	Maximum current
L	Inductance, or Length (Wire Resistance, Cylindrical Capacitor)
L1,L2	Inductance
Lp,Ls	Parallel and series inductances
N	Number of turns
n	Number of turns per unit length
P	Power
q	Charge
q1,q2	Point charge
Qp,Qs	Parallel and series quality factors
r	Charge distance
R,R1,R2	Resistance
ri,ro	Inside and outside radii
Rp,Rs	Parallel and series resistances
t	Time
ti,tf	Initial and final times
V	Voltage, or Total voltage (Voltage Divider)
V1	Voltage across R1
Vi,Vf	Initial and final voltages
Vmax	Maximum voltage
XC	Reactance of capacitor
XL	Reactance of inductor

Coulomb's Law

This equation represents Coulomb's law, the formal representation of the nature of electrostatic force between two charged particles. This force acts along the line connecting the two charges. It's attractive (negative) if the charges are unlike, and it's repulsive (positive) if the charges are alike. Coulomb's law applies over a wide range of separation distances — in particular, it's valid for atomic distances. You can calculate the force on a charge due to a system of charges by successively applying Coulomb's law and using the principle of superposition (in three dimensions).

Equation:

$$F = \frac{1}{4 \cdot \pi \cdot \varepsilon 0 \cdot \varepsilon r} \cdot \left(\frac{q1 \cdot q2}{r^2} \right)$$

Example: Given *q1*=1.6E−19_C, *q2*=1.6E−19_C, *r*=4.00E−13_cm, *εr*=1.00, solves *F*=14.3801_N.

Ohm's Law and Power

These four equations represent the relationships among current, voltage, and power dissipation for a resistor. Ohm's law states that the relation between voltage and current is linearly dependent upon resistance. A conducting material obeys Ohm's law if its resistivity is independent of the direction and magnitude of the electric field. In all homogeneous materials, Ohm's law is valid over a narrow range of electric fields in the material. If the electric field is too strong, there is considerable departure from Ohm's law.

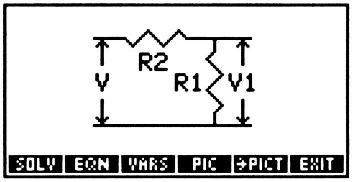
Equations:

$$V = I \cdot R \qquad P = V \cdot I \qquad P = I^2 \cdot R \qquad P = \frac{V^2}{R}$$

Example: Given *V*=24_V, *I*=16_A, solves *R*=1.5_Ω, *P*=384_W.

Voltage Divider

This equation describes the behavior of a two-resistor voltage-divider circuit. The circuit consists of two resistors in series. The equation assumes the voltage source has zero internal resistance and no current is drawn from the circuit at the voltage output.



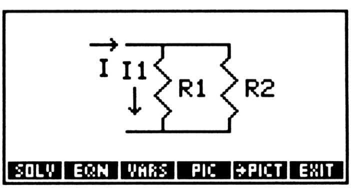
Equation:

$$V1 = V \cdot \left(\frac{R1}{R1 + R2} \right)$$

Example: Given $R1 = 40\_\Omega$, $R2 = 10\_\Omega$, $V = 100\_V$, solves $V1 = 80\_V$.

Current Divider

This equation describes the behavior of a two-resistor current-divider circuit. The circuit consists of two resistors in parallel. The equation assumes the current source has zero internal admittance.



Equation:

$$I1 = I \cdot \left(\frac{R2}{R1 + R2} \right)$$

Example: Given $R1 = 10\_\Omega$, $R2 = 6\_\Omega$, $I = 15\_A$, solves $I1 = 5.6250\_A$.

Wire Resistance

This equation relates electrical resistance, bulk resistivity, and the physical dimensions of a conductor. The equation assumes the conductor is homogeneous and isotropic and its cross-sectional area is uniform. You can use this equation for an nonhomogeneous material if you apply it to small "pill-box" elements and evaluate the resistance of the wire analytically. The equation applies only to bulk resistivity—it doesn't apply to sheet resistivity, such as skin resistance at high frequencies.

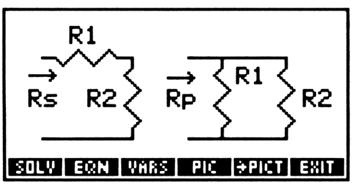
Equation:

$$R = \frac{\rho \cdot L}{A}$$

Example: Given ρ=.0035_Ω*cm, L=50_cm, A=1_cm^2, solves R=0.175_Ω.

Series and Parallel R

These two equations determine values of equivalent resistance of two ideal resistive elements connected in series and in parallel. The equations assume neither resistance is an open circuit for the series circuit and neither resistance is a short circuit for the parallel circuit.



Equations:

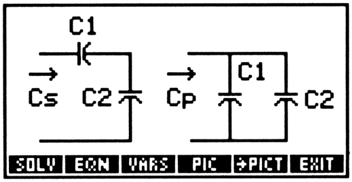
$$Rs = R1 + R2$$

$$\frac{1}{Rp} = \frac{1}{R1} + \frac{1}{R2}$$

Example: Given $R1$=2_Ω, $R2$=3_Ω, solves Rs=5_Ω, Rp=1.2000_Ω.

Series and Parallel C

These two equations determine values of equivalent capacitance of two ideal capacitive elements connected in series and in parallel. The equations assume both capacitors are ideal—no series or leakage resistances. No assumptions are made about the initial charge conditions of the capacitors. The equations assume the capacitive elements are linear, so that the laws of superposition can be applied.



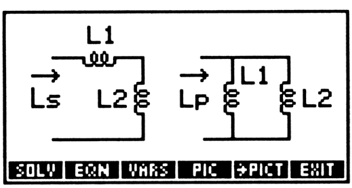
Equations:

$$\frac{1}{Cs} = \frac{1}{C1} + \frac{1}{C2} \qquad Cp = C1 + C2$$

Example: Given $C1 = 2\_\mu F$, $C2 = 3\_\mu F$, solves $Cs = 1.2000\_\mu F$, $Cp = 5\_\mu F$.

Series and Parallel L

These two equations determine values of equivalent inductance of two ideal inductive elements connected in series and in parallel. The equations assume both inductors are ideal—no series resistance. No assumptions are made about the initial conditions of the inductors. The equations assume the inductive elements are linear, so that the laws of superposition can be applied.



Equations:

$$Ls = L1 + L2 \qquad \frac{1}{Lp} = \frac{1}{L1} + \frac{1}{L2}$$

Example: Given $L1$=17_mH, $L2$=16.5_mH, solves Ls=33.5000_mH, Lp=8.3731_mH.

Capacitive Energy

This equation defines the electrostatic energy stored in a charged capacitor. This energy is the work done in charging the capacitor and is stored as electrical potential energy in the field between the plates. This energy is recovered by discharging the capacitor. The equation assumes the capacitor is a linear element.

Equation:

$$E = \frac{C \cdot V^2}{2}$$

Example: Given E=.025_J, C=20_μF, solves V=50_V.

Inductive Energy

This equation defines the magnetic energy stored in an inductor carrying current. This energy is the work done in establishing the current through the inductor and is stored as magnetic potential energy in the field around the conductor. This energy is recovered as the collapsing field generates current flow. The equation assumes the inductor is a linear element.

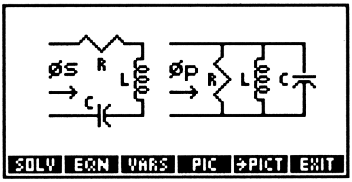
Equation:

$$E = \frac{L \cdot I^2}{2}$$

Example: Given E=4_J, L=15_mH, solves I=23.0940_A.

RLC Current Delay

These five equations describe the phase delays of ac currents in two resistor-capacitor-inductor circuits: elements in series and elements in parallel. The phase delay (angle) is positive for current lagging voltage. The equations also describe the capacitive and inductive reactances at a certain frequency. The equations assume the inputs are sinusoidal and the circuit elements are linear, time-invariant, and loss-free. (Guesses for ϕs and ϕp can help find the desired solution.)



Equations:

$$\mathrm{TAN}(\phi s) = \frac{XL - XC}{R} \qquad \mathrm{TAN}(\phi p) = \frac{\frac{1}{XC} - \frac{1}{XL}}{\frac{1}{R}} \qquad XC = \frac{1}{\omega \cdot C}$$

$$XL = \omega \cdot L$$

$$\omega = 2 \cdot \pi \cdot f$$

Example: Given $f = 107\_\mathrm{Hz}$, $C = 80\_\mu\mathrm{F}$, $L = 20\_\mathrm{mH}$, $R = 5\_\Omega$, solves $\omega = 672.3008\_\mathrm{r/s}$, $\phi s = -45.8292\_°$, $\phi p = -5.8772\_°$, $XC = 18.5929\_\Omega$, $XL = 13.4460\_\Omega$.

DC Capacitor Current

These three equations define the dc current required to change the voltage on a capacitor from an initial value to a final value in a certain time interval. For practical purposes and for meaningful values of charging current, the time interval should be very small — microseconds or nanoseconds. These equations provide a numerical approximation to analytical equations involving derivatives. However, for many applications, the charging currents of greatest interest are those occurring during a rapid rise or fall of a signal — this is an adequate approximation for such cases.

Equations:

$$I = C \cdot \left(\frac{\Delta V}{\Delta t} \right) \qquad \Delta V = Vf - Vi \qquad \Delta t = tf - ti$$

Example: Given C=15_μF, Vi=2.3_V, Vf=3.2_V, I=10_A, ti=0_s, solves ΔV=.9000_V, Δt=1.3500_μs, tf=1.3500_μs.

Capacitor Charge

This equation relates the charge and voltage on a capacitor. The equation makes no assumptions about the capacitor.

Equation:

$$q = C \cdot V$$

Example: Given C=20_μF, V=100_V, solves q=0.0020_C.

DC Inductor Voltage

These three equations represent Lenz's law, which calculates the dc voltage induced in an inductor responding to a change in current from an initial value to a final value in a certain time interval. For practical purposes and for meaningful values of induced voltage, the time interval should be very small — microseconds or nanoseconds. These equations provide a numerical approximation to analytical equations involving derivatives. However, for many applications, the induced voltages of greatest interest are those occurring during a rapid rise or fall of a signal — this is an adequate approximation for such cases.

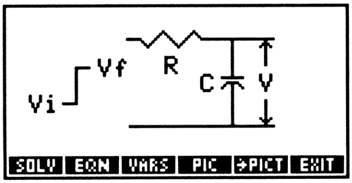
Equations:

$$V = -L \cdot \left(\frac{\Delta I}{\Delta t} \right) \qquad \Delta I = If - Ii \qquad \Delta t = tf - ti$$

Example: Given L=100_mH, V=52_V, Δt=32_μs, Ii=23_A, ti=0_s, solves ΔI= −0.0166_A, If=22.9834_A, tf=32_μs.

RC Transient

This equation defines the response of a resistor-capacitor circuit to a step change in the input voltage. The equation assumes the resistive and capacitive elements are ideal—linear, time-invariant, and loss-free. It also assumes the initial input voltage level has been stable long enough to stabilize the initial charge on the capacitor, and the input voltage jumps to its final value essentially instantaneously and holds at that level.



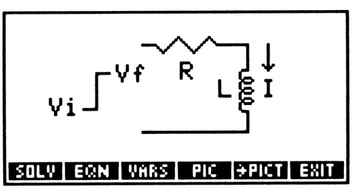
Equation:

$$V = Vf - (Vf - Vi) \cdot e^{\frac{-t}{R \cdot C}}$$

Example: Given Vi=0_V, C=50_μF, Vf=10_V, R=100_Ω, t=2_ms, solves V=3.2968_V.

RL Transient

This equation defines the response of a resistor-inductor circuit to a step change in the input voltage. The equation assumes the resistive and inductive elements are ideal—linear, time-invariant, and loss-free. It also assumes the initial input voltage level has been stable long enough to stabilize the initial current in the circuit, and the input voltage jumps to its final value essentially instantaneously and holds at that level.



Equation:

$$I = \frac{1}{R} \cdot \left(Vf - (Vf - Vi) \cdot e^{\frac{-t \cdot R}{L}} \right)$$

Example: Given Vi = 0_V, Vf = 5_V, R = 50_Ω, L = 50_mH, t = 75_μs, solves I = 0.0072_A.

Resonant Frequency

These four equations describe the resonance of series and parallel resistor-inductor-capacitor circuits and their quality factors. The resonant frequency of the circuit is independent of the connection (serial or parallel). The equations assume the circuit elements are ideal — linear, time-invariant, and loss-free. The dynamic response of the circuit is measured by the quality factor. For a series circuit, the quality factor represents the ratio of the capacitor or inductor voltage to the source voltage. For a parallel circuit, the quality factor represents the ratio of the capacitor or inductor current to the source current.

Equations:

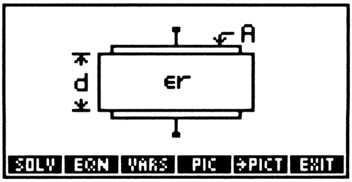
$$\omega 0 = \frac{1}{\sqrt{L \cdot C}} \qquad Qs = \frac{1}{R} \cdot \sqrt{\frac{L}{C}} \qquad Qp = R \cdot \sqrt{\frac{C}{L}}$$

$$\omega 0 = 2 \cdot \pi \cdot f0$$

Example: Given L = 500_mH, C = 8_μF, R = 10_Ω, solves $\omega 0$ = 500_r/s, Qs = 25.0000, Qp = 0.0400, $f0$ = 79.5775_Hz.

Plate Capacitor

This equation relates the capacitance of a parallel plate capacitor to its construction. The equation assumes the plates of the capacitor are so large and so close that the "fringing" effects of the electric field at the edges can be neglected — it assumes the electric field is constant everywhere in the capacitor. In practice, this is realistic if the plate separation is less than 10 percent of the width of the plates. The equation also assumes the dielectric material between the plates is made of a homogeneous material.



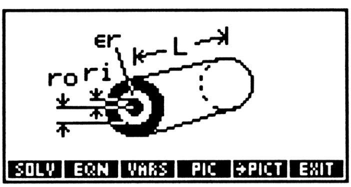
Equation:

$$C = \frac{\varepsilon 0 \cdot \varepsilon r \cdot A}{d}$$

Example: Given $C = 25\_\mu F$, $\varepsilon r = 2.26$, $A = 1\_cm^{\wedge}2$, solves $d = 8.0042E-9\_cm$.

Cylindrical Capacitor

This equation relates the capacitance of a cylindrical capacitor to its construction. A cylindrical capacitor is formed by two coaxial conductors separated by a dielectric material. The equation assumes the length of the capacitor is so large compared to the separation that the "fringing" effects of the electric field at the ends can be neglected—it assumes the electric field is uniform in the capacitor. In practice, this is realistic if the separation is less than 10 percent of the length of the conductors. The equation also assumes the dielectric material between the conductors is made of a homogeneous material.



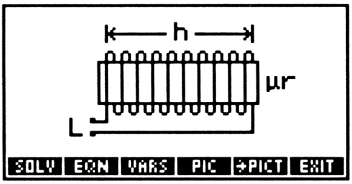
Equation:

$$C = \frac{2 \cdot \pi \cdot \varepsilon 0 \cdot \varepsilon r \cdot L}{LN\left(\frac{ro}{ri}\right)}$$

Example: Given $\varepsilon r = 1$, $ro = 1\_cm$, $ri = .999\_cm$, $L = 10\_cm$, solves $C = 0.0056\_\mu F$.

Solenoid Inductance

This equation relates the inductance of a solenoid to its geometry. The equation assumes the length of the solenoid is so large compared to the diameter of the tube that the "fringing" effects at the ends can be ignored. It also assumes the winding is uniformly spaced and the magnetic material in the solenoid is homogeneous.



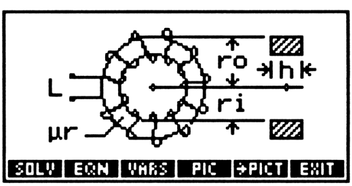
Equation:

$$L = \mu 0 \cdot \mu r \cdot n^2 \cdot A \cdot h$$

Example: Given $\mu r = 2.5$, $n = 40\_1/cm$, $A = .2\_cm\hat{}2$, $h = 3\_cm$, solves $L = 0.0302\_mH$.

Toroid Inductance

This equation relates the inductance of a toroid to its geometry. The equation assumes the toroid material is homogeneous and the cross section is rectangular.



Equation:

$$L = \frac{\mu 0 \cdot \mu r \cdot N^2 \cdot h}{2 \cdot \pi} \cdot LN\left(\frac{ro}{ri}\right)$$

Example: Given μr=1, N=5000, h=2_cm, ri=2_cm, ro=4_cm, solves L=69.3147_mH.

Sinusoidal Voltage

These two equations describe the instantaneous value of a sinusoidal voltage signal. (Guesses for ω, t, and ϕ can help find the desired solution.)

Equations:

$$V = Vmax \cdot SIN(\omega \cdot t + \phi) \qquad \omega = 2 \cdot \pi \cdot f$$

Example: Given $Vmax$=110_V, t=30_μs, f=60_Hz, ϕ=15_°, solves ω=376.9911_r/s, V=29.6699_V.

Sinusoidal Current

These two equations describe the instantaneous value of a sinusoidal current signal. (Guesses for ω, t, and ϕ can help find the desired solution.)

Equations:

$$I = Imax \cdot SIN(\omega \cdot t + \phi) \qquad \omega = 2 \cdot \pi \cdot f$$

Example: Given t=32_s, $Imax$=10_A, ω=636_r/s, ϕ=30_°, solves I=9.5983_A, f=101.2225_Hz.

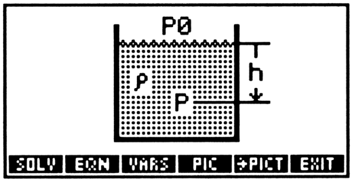
Fluids

Variable Names and Descriptions

ε	Roughness
μ	Dynamic viscosity
ρ	Density
ΔP	Pressure change
Δy	Height change
ΣK	Total fitting coefficients
A	Cross-sectional area
A1,*A2*	Initial and final cross-sectional areas
D	Diameter
D1,*D2*	Initial and final diameters
h	Depth relative to *P0* reference depth
hL	Head loss
L	Length
M	Mass flow rate
n	Kinematic viscosity
P	Pressure at *h*
P0	Reference pressure
P1,*P2*	Initial and final pressures
Q	Volume flow rate
Re	Reynolds number
v1,*v2*	Initial and final velocities
vavg	Average velocity
W	Power input
y1,*y2*	Initial and final heights

Pressure at Depth

This equation describes hydrostatic pressure as a function of depth from a reference for an incompressible fluid. Depth *h* is positive downward from the reference — altitude is represented by negative *h*.



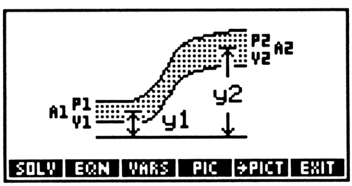
Equation:

$P = P0 + \rho \cdot g \cdot h$

Example: Given $h = 100\_m$, $\rho = 1025.1817\_kg/m\text{^}3$, $P0 = 1\_atm$, solves $P = 1106.6848\_kPa$.

Bernoulli Equation

These 10 equations represent the conservation of energy in a streamlined flow of an incompressible fluid. The equations account for differences in pressure, velocity, area, and height between inlet and outlet. The equations assume the effects of friction are negligible.



Equations:

$$\frac{\Delta P}{\rho}+\frac{v2^2-v1^2}{2}+g\cdot\Delta y=0$$

$$\frac{\Delta P}{\rho}+\frac{v2^2\cdot\left(1-\left(\frac{A2}{A1}\right)^2\right)}{2}+g\cdot\Delta y=0$$

$$\frac{\Delta P}{\rho}+\frac{v1^2\cdot\left(\left(\frac{A1}{A2}\right)^2-1\right)}{2}+g\cdot\Delta y=0$$

$$\Delta P=P2-P1 \qquad Q=A2\cdot v2 \qquad A1=\frac{\pi\cdot D1^2}{4}$$

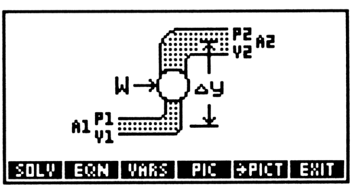
$$\Delta y=y2-y1 \qquad Q=A1\cdot v1 \qquad A2=\frac{\pi\cdot D2^2}{4}$$

$$M=\rho\cdot Q$$

Example: Given *P2*=25_psi, *P1*=75_psi, *y2*=35_ft, *y1*=0_ft, *D2*=24_in, *D1*=18_in, *ρ*=64_lb/ft^3, *v1*=100_ft/s, solves *Q*=23075.8762_ft^3/min, *M*=1476856.0769_lb/min, *v2*=122.4213_ft/s, *A2*=452.3893_in^2, *A1*=254.4690_in^2, Δ*P*=−50_psi, Δ*y*=35_ft.

Flow with Losses

These 10 equations extend Bernoulli's equation (previous title) to include power input (or output) and head loss.



Equations:

$$M \cdot \left(\frac{\Delta P}{\rho} + \frac{v2^2 - v1^2}{2} + g \cdot \Delta y + hL \right) = W$$

$$M \cdot \left(\frac{\Delta P}{\rho} + \frac{v2^2 \cdot \left(1 - \left(\frac{A2}{A1} \right)^2 \right)}{2} + g \cdot \Delta y + hL \right) = W$$

$$M \cdot \left(\frac{\Delta P}{\rho} + \frac{v1^2 \cdot \left(\left(\frac{A1}{A2} \right)^2 - 1 \right)}{2} + g \cdot \Delta y + hL \right) = W$$

$$\Delta P = P2 - P1 \qquad Q = A2 \cdot v2 \qquad A1 = \frac{\pi \cdot D1^2}{4}$$

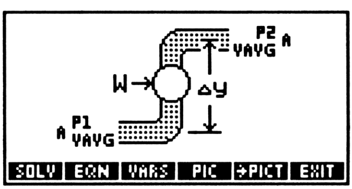
$$\Delta y = y2 - y1 \qquad Q = A1 \cdot v1 \qquad A2 = \frac{\pi \cdot D2^2}{4}$$

$$M = \rho \cdot Q$$

Example: Given $P2$=30_psi, $P1$=65_psi, $y2$=100_ft, $y1$=0_ft, ρ=64_lb/ft^3, $D1$=24_in, $D2$=18_in, hL=2.0_ft^2/s^2, W=25_hp, $v1$=100_ft/s, solves Q=36.1018_ft^3/min, M=2310.5165_lb/min, ΔP= −35_psi, Δy=100_ft, $v2$=.3405_ft/s, $A1$=452.3893_in^2, $A2$=254.4690_in^2.

Flow in Full Pipes

These eight equations adapt Bernoulli's equation (earlier title) for flow in a round, full pipe, including power input (or output) and frictional losses. The equations use the Fanning friction factor to calculate the frictional losses (see "FANNING Function" in chapter 8). The equations assume the fluid temperature and viscosity are constant.



Equations:

$$\rho\cdot\left(\frac{\pi\cdot D^2}{4}\right)\cdot vavg\cdot\left(\frac{\Delta P}{\rho}+g\cdot\Delta y+vavg^2\cdot\left(2\cdot f\cdot\left(\frac{L}{D}\right)+\frac{\Sigma K}{2}\right)\right)=W$$

$$\Delta P=P2-P1 \qquad Q=A\cdot vavg \qquad Re=\frac{D\cdot vavg\cdot\rho}{\mu}$$

$$\Delta y=y2-y1 \qquad A=\frac{\pi\cdot D^2}{4} \qquad n=\frac{\mu}{\rho}$$

$$M=\rho\cdot Q$$

Example: Given ρ=62.4_lb/ft^3, *D*=12_in, *vavg*=8_ft/s, *P2*=15_psi, *P1*=20_psi, *y2*=40_ft, *y1*=0_ft, μ=0.00002_lbf*s/ft^2, Σ*K*=2.25, ε=0.02_in, *L*=250_ft, solves Δ*P*=−5_psi, Δ*y*=40_ft, *A*=113.0973_in^2, *n*=1.0312_ft^2/s, *Q*=376.9911_ft^3/min, *M*=23524.2458_lb/min, *W*=25.8897_hp, *Re*=775780.5.

Forces and Energy

Variable Names and Descriptions

α	Angular acceleration
ω	Angular velocity
ωi,ωf	Initial and final angular velocities
ρ	Fluid density
τ	Torque
Θ	Angular displacement
a	Acceleration
A	Projected area relative to flow
ar	Centripetal acceleration at *r*
at	Tangential acceleration at *r*
Cd	Drag coefficient
E	Energy

Variable Names and Descriptions (continued)

F	Force at *r* or *x*, or Spring force (Hooke's Law), or Attractive force (Law of Gravitation), or Drag force (Drag Force)
I	Moment of inertia
k	Spring constant
Ki,Kf	Initial and final kinetic energies
m,m1,m2	Mass
N	Rotational speed
Ni,Nf	Initial and final rotational speeds
P	Instantaneous power
Pavg	Average power
r	Radius from rotation axis, or Separation distance (Law of Gravitation)
t	Time
v	Velocity
vf,v1f,v2f	Final velocity
vi,v1i	Initial velocity
W	Work
x	Displacement

Linear Mechanics

These eight equations describe the basis of linear Newtonian mechanics. They describe Newton's second law and the concepts of force, kinetic energy, work, and power.

Equations:

$$F = m \cdot a$$

$$Ki = \frac{1}{2} \cdot m \cdot vi^2$$

$$Kf = \frac{1}{2} \cdot m \cdot vf^2$$

$$W = F \cdot x$$

$$W = Kf - Ki$$

$$P = F \cdot v$$

$$Pavg = \frac{W}{t}$$

$$vf = vi + a \cdot t$$

Example: Given $t = 10\_s$, $m = 50\_lb$, $a = 12.5\_ft/s^2$, $vi = 0\_ft/s$, solves $vf = 125\_ft/s$, $x = 625\_ft$, $F = 19.4256\_lbf$, $Ki = 0\_ft*lbf$, $Kf = 12140.9961\_ft*lbf$, $W = 12140.9961\_ft*lbf$, $Pavg = 2.2075\_hp$.

Angular Mechanics

These 12 equations describe the basis of angular mechanics. They describe Newton's second law and the concepts of torque, kinetic energy, work, and power.

Equations:

$$\tau = I \cdot \alpha$$

$$Ki = \frac{1}{2} \cdot I \cdot \omega i^2$$

$$Kf = \frac{1}{2} \cdot I \cdot \omega f^2$$

$$W = \tau \cdot \Theta$$

$$W = Kf - Ki$$

$$P = \tau \cdot \omega$$

$$Pavg = \frac{W}{t}$$

$$\omega f = \omega i + \alpha \cdot t$$

$$at = \alpha \cdot r$$

$$\omega = 2 \cdot \pi \cdot N$$

$$\omega i = 2 \cdot \pi \cdot Ni$$

$$\omega f = 2 \cdot \pi \cdot Nf$$

Example: Given $I = 1750\_lb*in^2$, $\Theta = 360\_°$, $r = 3.5\_in$, $\alpha = 10.5\_r/min^2$, $\omega i = 0\_r/s$, solves $\tau = 1.1017E-3\_ft*lbf$, $Ki = 0\_ft*lbf$, $W = 6.9221E-3\_ft*lbf$, $Kf = 6.9221E-3\_ft*lbf$, $at = 8.5069E-4\_ft/s^2$, $Ni = 0\_rpm$, $\omega f = 11.4868\_r/min$, $t = 1.0940\_min$, $Nf = 1.8282\_rpm$, $Pavg = 1.9174E-7\_hp$.

Centripetal Force

These four equations describe the centripetal (radial) force on an object rotating about an axis at a constant angular velocity.

Equations:

$$F = m \cdot \omega^2 \cdot r$$

$$\omega = \frac{v}{r}$$

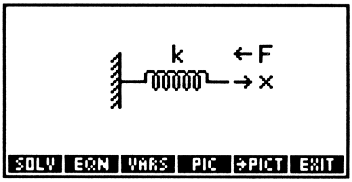
$$ar = \frac{v^2}{r}$$

$$\omega = 2 \cdot \pi \cdot N$$

Example: Given $m = 1\_kg$, $r = 5\_cm$, $N = 2000\_Hz$, solves $\omega = 12566.3706\_r/s$, $ar = 7895683.5209\_m/s$, $F = 7895683.5209\_N$, $v = 628.3185\_m/s$.

Hooke's Law

These two equations describe the force exerted by a spring and the work (potential energy) involved in stretching the spring.



Equations:

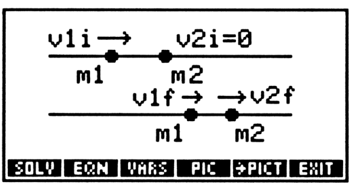
$$F = -k \cdot x$$

$$W = \frac{-1}{2} \cdot k \cdot x^2$$

Example: Given k=1725_lbf/in, x=1.25_in, solves F= −2156.25_lbf, W= −112.3047_ft*lbf.

1D Elastic Collisions

These two equations describe a linear collision between two objects. The equations are based on the laws of conservation of momentum and energy. The equations assume object 2 is initially at rest, no frictional forces act on the objects, and the collision is elastic (no energy is lost).



Equations:

$$v1f = \frac{m1 - m2}{m1 + m2} \cdot v1i$$

$$v2f = \frac{2 \cdot m1}{m1 + m2} \cdot v1i$$

Example: Given $m1$=10_kg, $m2$=25_kg, $v1i$=100_m/s, solves $v1f$= −42.8571_m/s, $v2f$=57.1429_m/s.

Drag Force

This equation describes the frictional force on a body moving through a fluid (such as air or water) that acts to retard the motion. The equation assumes the velocity is high enough to cause the fluid flow behind the object to be turbulent.

Equation:

$$F = Cd \cdot \left(\frac{\rho \cdot v^2}{2} \right) \cdot A$$

Example: Given $Cd = .05$, $\rho = 1000\_kg/m\hat{}3$, $A = 7.5E6\_cm\hat{}2$, $v = 35\_m/s$, solves $F = 22968750\_N$.

Law of Gravitation

This equation describes Newton's universal law of gravitation, which defines the gravitational attractive force that occurs between any two masses.

Equation:

$$F = G \cdot \left(\frac{m1 \cdot m2}{r^2} \right)$$

Example: Given $m1 = 2E15\_kg$, $m2 = 2E18\_kg$, $r = 1000000\_km$, solves $F = 266903.6\_N$.

Mass-Energy Relation

This equation relates the rest mass of an object to its energy. The equation was derived by Albert Einstein.

Equation:

$$E = m \cdot c^2$$

Example: Given $m = 9.1E-31\_kg$, solves $E = 8.1787E-14\_J$.

Gases

Variable Names and Descriptions

λ	Mean free path
ρ	Flow density
ρ0	Stagnation density
A	Flow area
At	Throat area
d	Molecular diameter
k	Specific heat ratio
M	Mach number
m	Mass
MW	Molecular weight
n	Number of moles, or Polytropic constant (Polytropic Processes)
P	Pressure, or Flow pressure (Isentropic Flow)
P0	Stagnation pressures
Pc	Pseudocritical pressure
Pi,Pf	Initial and final pressures
T	Temperature, or Flow temperature (Isentropic Flow)
T0	Stagnation temperature
Tc	Pseudocritical temperature
Ti,Tf	Initial and final temperatures
V	Volume
Vi,Vf	Initial and final volumes
vrms	Root-mean-square (rms) velocity
W	Work

Ideal Gas Law

These two equations represent the relation between pressure, volume, temperature, and mass of an ideal gas.

Equations:

$$P \cdot V = n \cdot R \cdot T \qquad\qquad m = n \cdot MW$$

Example: Given T=16.85_°C, P=1_atm, V=25_l, MW=36_g/gmol, solves n=1.0506_gmol, m=3.7820E–2_kg.

Ideal Gas State Change

This equation represents the change in the state of an ideal gas from initial conditions of pressure, volume, and temperature to final conditions.

Equation:

$$\frac{Pf \cdot Vf}{Tf} = \frac{Pi \cdot Vi}{Ti}$$

Example: Given Pi=1.5_kPa, Pf=1.5_kPa, Vi=2_l, Ti=100_°C, Tf=373.15_K, solves Vf=2_l.

Isothermal Expansion

These two equations describe the expansion or compression of an ideal gas at constant temperature.

Equations:

$$W = n \cdot R \cdot T \cdot LN\left(\frac{Vf}{Vi}\right) \qquad\qquad m = n \cdot MW$$

Example: Given Vi=2_l, Vf=125_l, T=300_°C, n=0.25_gmol, MW=64_g/gmol, solves W=4926.4942_J, m=.016_kg.

Polytropic Processes

These two equations describe the general case of a reversible pressure-volume change of an ideal gas with associated heat transfer such that $P \cdot V^n$ is constant. Special cases for the polytropic constant n include isothermal processes ($n = 1$), isentropic processes ($n = k$, the ratio of constant-pressure specific heat to constant-volume specific heat, both at zero pressure), and constant-pressure processes ($n = 0$).

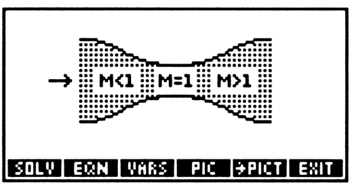
Equations:

$$\frac{Pf}{Pi} = \left(\frac{Vf}{Vi}\right)^{-n} \qquad \frac{Tf}{Ti} = \left(\frac{Pf}{Pi}\right)^{\frac{n-1}{n}}$$

Example: Given Pi=15_psi, Pf=35_psi, Vi=1_ft^3, Vf=0.50_ft^3, Ti=75_°F, solves n=1.2224, Tf=164.1117_°F.

Isentropic Flow

These four equations describe isentropic flow—the adiabatic and reversible flow of a compressible fluid. The flow of a compressible fluid behaves differently at flow velocities below and above Mach 1. The Mach number is based on the speed of sound in the fluid.



Equations:

$$\frac{T}{T0} = \frac{2}{2 + (k-1) \cdot M^2} \qquad \frac{\rho}{\rho 0} = \left(\frac{T}{T0}\right)^{\frac{1}{k-1}}$$

$$\frac{P}{P0} = \left(\frac{T}{T0}\right)^{\frac{k}{k-1}} \qquad \frac{A}{At} = \frac{1}{M} \cdot \left(\frac{2}{k+1} \cdot \left(1 + \frac{k-1}{2} \cdot M^2\right)\right)^{\frac{k+1}{2 \cdot (k-1)}}$$

Example: Given $k = 2$, $M = .9$, $T0 = 26.85\_°C$, $T = 373.15\_K$, $\rho 0 = 100\_kg/m\char`^3$, $P0 = 100\_kPa$, $A = 1\_cm\char`^2$, solves $P = 464.1152\_kPa$, $At = 0.9928\_cm\char`^2$, $\rho = 215.4333\_kg/m\char`^3$.

Real Gas Law

These two equations adapt the ideal gas law (earlier title) to emulate real-gas behavior over a wide range of conditions. The equations use the gas-compressibility Z factor to correct for nonideal behavior (see "ZFACTOR Function" in chapter 8).

Equations:

$$P \cdot V = n \cdot Z \cdot R \cdot T \qquad m = n \cdot MW$$

Example: Given $Pc = 48\_atm$, $Tc = 298\_K$, $P = 5\_kPa$, $V = 10\_l$, $MW = 64\_g/gmol$, $T = 75\_°C$, solves $n = 0.0173\_gmol$, $m = 1.1057E-3\_kg$.

Real Gas State Change

This equation adapts the ideal gas law state-change equation (earlier title) to emulate real-gas behavior. It represents the change in the state of a real gas from initial conditions of pressure, volume, and temperature to final conditions. The equations use the gas-compressibility Z factor to correct for nonideal behavior (see "ZFACTOR Function" in chapter 8).

Equation:

$$\frac{Pf \cdot Vf}{Zf \cdot Tf} = \frac{Pi \cdot Vi}{Zi \cdot Ti}$$

Example: Given $Pc = 48\_atm$, $Pi = 100\_kPa$, $Pf = 50\_kPa$, $Ti = 75\_°C$, $Tc = 298\_K$, $Vi = 10\_l$, $Tf = 250\_°C$, solves $Vf = 30.1703\_l$.

Kinetic Theory

These four equations describe properties of an ideal gas. The equations assume Maxwellian distribution statistics and perfectly elastic collisions for gas molecules.

Equations:

$$P = \frac{n \cdot MW \cdot vrms^2}{3 \cdot V}$$

$$vrms = \sqrt{\frac{3 \cdot R \cdot T}{MW}}$$

$$\lambda = \frac{1}{\sqrt{2} \cdot \pi \cdot \left(\frac{n \cdot NA}{V}\right) \cdot d^2}$$

$$m = n \cdot MW$$

Example: Given $P = 100$_kPa, $V = 2$_l, $T = 26.85$_°C, $MW = 18$_g/gmol, $d = 2.5$_nm, solves $vrms = 644.7678$_m/s, $m = 1.4433E-3$_kg, $n = .0802$_gmol, $\lambda = 1.4916$_nm.

Heat Transfer

Variable Names and Descriptions

α	Expansion coefficient
δ	Elongation
$\lambda 1, \lambda 2$	Lower and upper wavelength limits
λmax	Wavelength of maximum emissive power
ΔT	Temperature difference
A	Area
c	Specific heat
eb12	Emissive power in the range $\lambda 1$ to $\lambda 2$
eb	Total emissive power
f	Fraction of emissive power in the range $\lambda 1$ to $\lambda 2$
h,h1,h3	Convective heat-transfer coefficient
k,k1,k2,k3	Thermal conductivity
L,L1,L2,L3	Length
m	Mass
Q	Heat capacity
q	Heat transfer rate
T	Temperature

Variable Names and Descriptions (continued)

Tc	Cold surface temperature (Conduction), or Cold fluid temperature
Th	Hot surface temperature, or Hot fluid temperature (Conduction + Convection)
Ti,Tf	Initial and final temperatures
U	Overall heat transfer coefficient

Heat Capacity

These two equations calculate heat capacity of a material in terms of specific heat, mass, and temperatures.

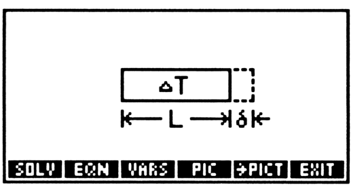
Equations:

$$Q = m \cdot c \cdot \Delta T \qquad Q = m \cdot c \cdot (Tf - Ti)$$

Example: Given ΔT=15_°C, *Ti*=0_°C, *m*=10_kg, *Q*=25_kJ, solves *Tf*=15_°C, *c*=.1667_kJ/(kg*K).

Thermal Expansion

These two equations describe the linear expansion of a material due to a temperature change. These equations are valid over a wide temperature range.



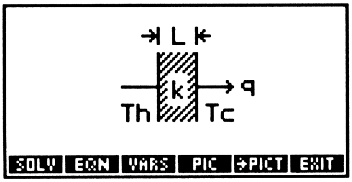
Equations:

$$\delta = \alpha \cdot L \cdot \Delta T \qquad \delta = \alpha \cdot L \cdot (Tf - Ti)$$

Example: Given ΔT=15_°C, *L*=10_m, *Tf*=25_°C, δ=1_cm, solves *Ti*=10_°C, α=6.6667E−5_1/°C.

Conduction

These two equations describe the rate of heat transfer through a homogeneous slab of material. The two temperatures are surface temperatures. The equations assume a one-dimensional model of heat transfer due to conduction — lateral heat transfer is ignored.



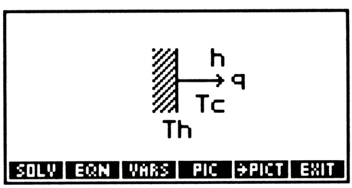
Equations:

$$q = \frac{k \cdot A}{L} \cdot \Delta T \qquad q = \frac{k \cdot A}{L} \cdot (Th - Tc)$$

Example: Given *Tc* = 25_°C, *Th* = 75_°C, *A* = 12.5_m^2, *L* = 1.5_cm, *k* = .12_W/(m*K), solves *q* = 5000_W, ΔT = 50_°C.

Convection

These two equations describe the rate of heat transfer from a surface due to convection. One temperature is the surface temperature, and one temperature is the fluid temperature. The equations assume the convective coefficient includes effects such as fluid characteristics, turbulence, and convection flows.



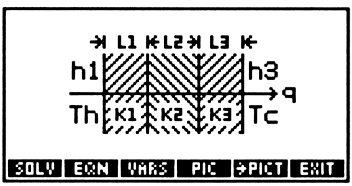
Equations:

$$q = h \cdot A \cdot \Delta T \qquad q = h \cdot A \cdot (Th - Tc)$$

Example: Given Tc=300_K, A=200_m^2, h=.005_W/(m^2*K), q=10_W, solves ΔT=10_°C, Th=36.8500_°C.

Conduction + Convection

These four equations describe the rate of heat transfer in a composite medium of up to three layers. Heat is transferred by convection to and from the fluid surrounding the composite medium and conductively within the medium. If you have fewer than three layers, give the extra layers a zero thickness and any nonzero conductivity. The two temperatures are fluid temperatures. If you know the *surface* temperature (instead of the *fluid* temperature) for a surface, set the corresponding convective coefficient to 10^{499} (or other large value). The equations assume a one-dimensional model of heat transfer—lateral heat transfer is ignored.



Equations:

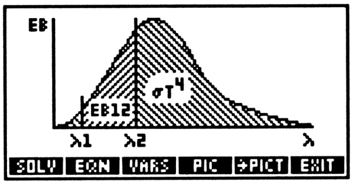
$$q = \frac{A \cdot \Delta T}{\frac{1}{h1} + \frac{L1}{k1} + \frac{L2}{k2} + \frac{L3}{k3} + \frac{1}{h3}} \qquad q = \frac{A \cdot (Th - Tc)}{\frac{1}{h1} + \frac{L1}{k1} + \frac{L2}{k2} + \frac{L3}{k3} + \frac{1}{h3}}$$

$$U = \frac{q}{A \cdot \Delta T} \qquad U = \frac{q}{A \cdot (Th - Tc)}$$

Example: Given ΔT=35_°C, Th=55_°C, A=10_m^2, $h1$=.05_W/(m^2*K), $h3$=.05_W/(m^2*K), $L1$=3_cm, $L2$=5_cm, $L3$=3_cm, $k1$=.1_W/(m*K), $k2$=.5_W/(m*K), $k3$=.1_W/(m*K), solves Tc=20_°C, U=0.0246_W/(m^2*K), q=8.5995_W.

Black Body Radiation

These five equations describe black-body radiation. A black body is defined as a body that emits the maximum amount of energy at every wavelength for a specified temperature. The equations also include Wien's displacement law, which relates temperature and the wavelength of maximum emissivity. The equations use the black-body emissive-power function (see "F0λ Function" in chapter 8).



Equations:

$eb = \sigma \cdot T^4$

$eb12 = f \cdot eb$

$q = eb \cdot A$

$f = F0\lambda(\lambda 2, T) - F0\lambda(\lambda 1, T)$

$\lambda max \cdot T = c3$

Example: Given T=1000_°C, $\lambda 1$=1000_nm, $\lambda 2$=600_nm, A=1_cm^2, solves λmax=2276.0523_nm, eb=148984.2703_W/m^2, f=.0036, $eb12$=537.7264_W/m^2, q=14.8984_W.

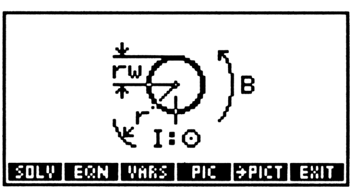
Magnetism

Variable Names and Descriptions

μr	Relative permeability
B	Magnetic field
d	Separation distance
Fba	Force
I,Ia,Ib	Current
L	Length
N	Total number of turns
n	Number of turns per unit length
r	Distance from center of wire
ri,ro	Inside and outside radii of toroid
rw	Radius of wire

Straight Wire

This equation relates the current carried by a wire and the resulting magnetic field. The magnetic field is calculated differently depending upon whether the point is inside or outside the wire. The equation assumes the wire is infinitely long and the material surrounding the wire is magnetically homogeneous. For most practical problems, the wire radius and the distance to the wire are much smaller than the physical length of the wire, so the assumption is appropriate.



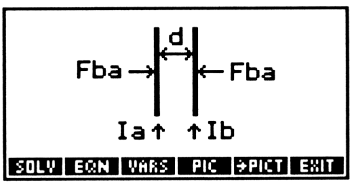
Equation:

$$B = \frac{\mu 0 \cdot \mu r \cdot I}{2 \cdot \pi \cdot r}$$

Example: Given $\mu r = 1$, $rw = .25\_cm$, $r = .2\_cm$, $I = 25\_A$, solves $B = .0016\_T$.

Force between Wires

This equation describes the force between two long, parallel wires carrying currents. The force on each wire lies in the plane of the wires, is perpendicular to the wires, and is positive for an attractive force (for currents having the same sign). The equation assumes the force is caused solely by the magnetic field produced by the current carried in the wires. It also assumes the separation is small compared to the length of the parallel wires.



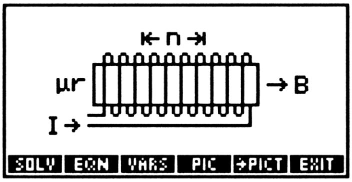
Equation:

$$Fba = \frac{\mu 0 \cdot \mu r \cdot L \cdot Ib \cdot Ia}{2 \cdot \pi \cdot d}$$

Example: Given $Ia = 10\_A$, $Ib = 20\_A$, $\mu r = 1$, $L = 50\_cm$, $d = 1\_cm$, solves $Fba = 2.0000E-3\_N$.

B Field in Solenoid

This equation describes the magnetic field density in a long, ideal solenoid. The equation is based on Ampere's law and the law of Biot and Savart. The magnetic field is uniform at the core of the solenoid cylinder. The equation neglects fringing and stray fields. The equation assumes the current in the wire is either dc or low-frequency ac — secondary effects such as inductive, eddy, and skin effects are ignored.



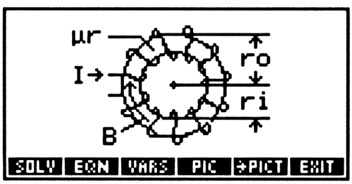
Equation:

$$B = \mu 0 \cdot \mu r \cdot I \cdot n$$

Example: Given $\mu r = 10$, $n = 50$, $I = 1.25\_A$, solves $B = 0.0785\_T$.

B Field in Toroid

This equation describes the magnetic field density in an ideal toroid. The equation is based on Ampere's law and the law of Biot and Savart. The magnetic field is uniform at the core of the toroid. The equation neglects fringing and stray fields. The equation assumes the current in the wire is either dc or low-frequency ac — secondary effects such as inductive, eddy, and skin effects are ignored. The equation also assumes the toroid can be described by averaging its inner and outer radii, but makes no other assumptions about its geometry.



Equation:

$$B = \frac{\mu 0 \cdot \mu r \cdot I \cdot N}{2 \cdot \pi} \cdot \left(\frac{2}{ro + ri} \right)$$

Example: Given $\mu r = 10$, $N = 50$, $ri = 5\_cm$, $ro = 7\_cm$, $I = 10\_A$, solves $B = 1.6667E-2\_T$.

Motion

Variable Names and Descriptions

α	Angular acceleration
ω	Angular velocity (Circular Motion), or Angular velocity at *t* (Angular Motion)
$\omega 0$	Initial angular velocity
ρ	Fluid density
Θ	Angular position at *t*
$\Theta 0$	Initial angular position (Angular Motion), or Initial vertical angle (Projectile Motion)
a	Acceleration
A	Projected horizontal area
ar	Centripetal acceleration at *r*
Cd	Drag coefficient
m	Mass
M	Planet mass
N	Rotational speed
R	Horizontal range (Projectile Motion), or Planet radius (Escape Velocity)
r	Radius
t	Time
v	Velocity at *t* (Linear Motion), or Velocity at *r* (Circular Motion), or Terminal velocity (Terminal Velocity), or Escape velocity (Escape Velocity)
v0	Initial velocity
vx	Horizontal component of velocity at *t*
vy	Vertical component of velocity at *t*
x	Horizontal position at *t*
x0	Initial horizontal position
y	Vertical position at *t*
y0	Initial vertical position

Linear Motion

These four equations describe the one-dimensional motion of an object with a constant linear acceleration. The equations assume the object experiences no frictional or velocity effects from the surrounding air.

Equations:

$$x = x0 + v0 \cdot t + \frac{1}{2} \cdot a \cdot t^2 \qquad x = x0 + \frac{1}{2} \cdot (v0 + v) \cdot t$$

$$x = x0 + v \cdot t - \frac{1}{2} \cdot a \cdot t^2 \qquad v = v0 + a \cdot t$$

Example: Given *x0*=0_m, *x*=100_m, *t*=10_s, *v0*=1_m/s, solves *v*=19_m/s, *a*=1.8_m/s^2.

Object in Free Fall

These four equations describe the one-dimensional motion of an object subject to acceleration due to gravity near the Earth's surface. Velocity is positive in the upward direction. The equations assume the object experiences no frictional, buoyant, or velocity effects from the surrounding air.

Equations:

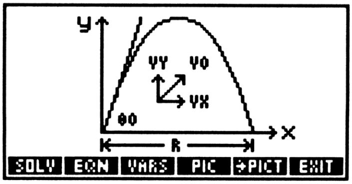
$$y = y0 + v0 \cdot t - \frac{1}{2} \cdot g \cdot t^2 \qquad v^2 = v0^2 - 2 \cdot g \cdot (y - y0)$$

$$y = y0 + v \cdot t + \frac{1}{2} \cdot g \cdot t^2 \qquad v = v0 - g \cdot t$$

Example: Given *y0*=1000_ft, *y*=0_ft, *v0*=0_ft/s, solves *t*=7.8843_s, *v*= −253.6991_ft/s.

Projectile Motion

These five equations describe the two-dimensional motion of a projectile near the Earth's surface. The range is the horizontal distance traveled before returning to the initial elevation. The equations assume the object experiences no frictional, buoyant, or velocity effects from the surrounding air and only gravity acts on the object. These equations apply to motion at low velocities or in a vacuum. (A guess for *Θ0* can help find the desired solution.)



Equations:

$$x = x0 + v0 \cdot COS(\Theta 0) \cdot t$$

$$y = y0 + v0 \cdot SIN(\Theta 0) \cdot t - \frac{1}{2} \cdot g \cdot t^2$$

$$vx = v0 \cdot COS(\Theta 0)$$

$$vy = v0 \cdot SIN(\Theta 0) - g \cdot t$$

$$R = \frac{v0^2}{g} \cdot SIN(2 \cdot \Theta 0)$$

Example: Given *x0*=0_ft, *y0*=0_ft, *Θ0*=45_°, *v0*=200_ft/s, *t*=10_s, solves *R*=1243.2399_ft, *vx*=141.4214_ft/s, *vy*=−180.3186_ft/s, *x*=1414.2136_ft, *y*=−194.4864_ft.

Angular Motion

These four equations describe the motion of an object with a constant angular acceleration.

Equations:

$$\Theta = \Theta 0 + \omega 0 \cdot t + \frac{1}{2} \cdot \alpha \cdot t^2$$

$$\Theta = \Theta 0 + \omega \cdot t - \frac{1}{2} \cdot \alpha \cdot t^2$$

$$\Theta = \Theta 0 + \frac{1}{2} \cdot (\omega 0 + \omega) \cdot t$$

$$\omega = \omega 0 + \alpha \cdot t$$

Example: Given *Θ0*=0_°, *ω0*=0_r/min, *α*=1.5_r/min^2, *t*=30_s, solves Θ=10.7430_°, *ω*=.7500_r/min.

Circular Motion

These three equations describe the tangential velocity and centripetal (radial) acceleration of an object rotating at a constant angular velocity (uniform circular motion).

Equations:

$$\omega = \frac{v}{r} \qquad ar = \frac{v^2}{r} \qquad \omega = 2 \cdot \pi \cdot N$$

Example: Given r=25_in, v=2500_ft/s, solves ω=72000_r/min, ar=3000000_ft/s^2, N=11459.1559_rpm.

Terminal Velocity

This equation relates the terminal velocity of an object falling near the Earth's surface to the properties of the object and the surrounding fluid (such as air or water). The terminal velocity is the velocity at which the drag force and the gravitational force are in equilibrium.

Equation:

$$v = \sqrt{\frac{2 \cdot m \cdot g}{Cd \cdot \rho \cdot A}}$$

Example: Given Cd=.15, ρ=.025lb/ft^3, A=100000_in^2, m=1250_lb, solves v=1757.4709_ft/s.

Escape Velocity

This equation relates the escape velocity of a projectile to the properties of the planet. The escape velocity is the initial velocity required of the projectile to escape the gravitational pull of the planet. The escape velocity is independent of the direction in which the projectile is fired. The equation assumes the rotation of the planet has no effect.

Equation:

$$v = \sqrt{\frac{2 \cdot G \cdot M}{R}}$$

Example: Given M=1.5E23_lb, R=5000_mi, solves v=3485.1106_ft/s.

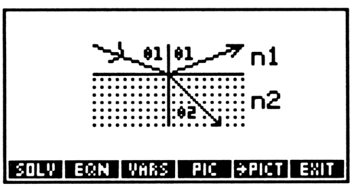
Optics

Variable Names and Descriptions

Θ1	Angle of incidence
Θ2	Angle of refraction
ΘB	Brewster angle
Θc	Critical angle
f	Focal length
m	Magnification
n,n1,n2	Index of refraction
r,r1,r2	Radius of curvature
u	Distance to object
v	Distance to image

Law of Refraction

This equation relates the angle of incidence and the angle of refraction to the indexes of refraction of the incident and refracting media. The angles must be less than 90 degrees. (Guesses for *Θ1* and *Θ2* can help find the desired solution.)



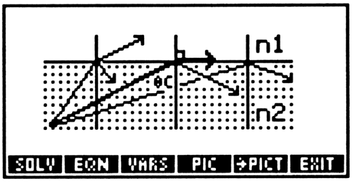
Equation:

n1 ·SIN(Θ1) = n2 ·SIN(Θ2)

Example: Given *n1* = 1, *n2* = 1.333, *Θ1* = 45_°, solves *Θ2* = 32.0367_°.

Critical Angle

This equation defines the critical angle for the interface between two media. The critical angle is the angle of incidence above which a light ray is totally reflected back into the original medium — no refraction occurs. The incident medium must have the greater index of refraction (*n2*>*n1*), otherwise the critical angle doesn't exist. (A guess for θc can help find the desired solution.)



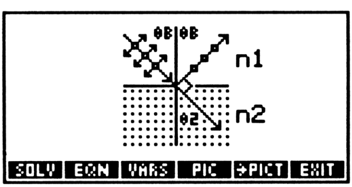
Equation:

$$\mathrm{SIN}(\theta c) = \frac{n1}{n2}$$

Example: Given $n1=1$, $n2=1.5$, solves $\theta c = 41.8103\_°$.

Brewster's Law

These two equations define the Brewster angle for the interface between two media. The Brewster angle is the angle of incidence at which the reflected wave is completely polarized with its plane of vibration at right angles to the plane of incidence. (Light waves and other electromagnetic waves have polarization properties associated with them, and the electric field vector for each wavetrain in the beam can be resolved into a component perpendicular to the plane of incidence and a component lying in the plane of incidence. In unpolarized light these two components are of equal magnitude.) (A guess for θ*B* can help find the desired solution.)



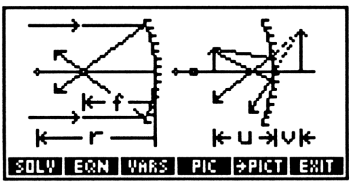
Equations:

$$\mathrm{TAN}(\Theta B) = \frac{n2}{n1} \qquad \Theta B + \Theta 2 = 90$$

Example: Given $n1 = 1$, $n2 = 1.5$, solves $\Theta B = 56.3099\_°$, $\Theta 2 = 33.6901\_°$.

Spherical Reflection

These three equations describe reflection from a spherical mirror, either concave or convex. The focal length and curvature radius are positive in the direction of the reflected light (concave mirror, as shown in the picture). The object distance is positive in front of the surface. The image distance is positive in the direction of the reflected light (in front of the surface). The magnification is positive for an upright image, negative for an inverted image. If parallel light rays shine on the mirror, the reflected rays appear to converge at the focus.



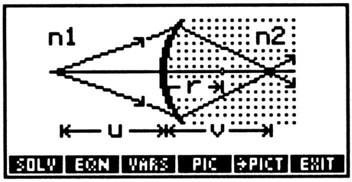
Equations:

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \qquad f = \frac{1}{2} \cdot r \qquad m = \frac{-v}{u}$$

Example: Given $u = 10\_\mathrm{cm}$, $v = 300\_\mathrm{cm}$, $r = 19.35\_\mathrm{cm}$, solves $m = -30$, $f = 9.6774\_\mathrm{cm}$.

Spherical Refraction

This equation describes refraction at a spherical interface between two media. The curvature radius is positive in the direction of the refracted light (convex interface, as shown in the picture). The object distance is positive in front of the interface. The image distance is positive in the direction of the refracted light (behind the interface).



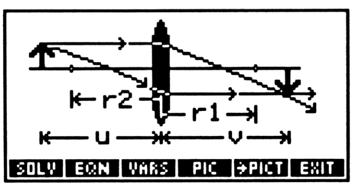
Equation:

$$\frac{n1}{u} + \frac{n2}{v} = \frac{n2 - n1}{r}$$

Example: Given u=8_cm, v=12_cm, r=2_cm, $n1$=1, solves $n2$=1.5000.

Thin Lens

These three equations describe magnification for a thin lens, a lens whose thickness is small compared to the object distance. The curvature radii are positive in the direction of the refracted light (convex to the incident light)—*r1* is for the front surface and *r2* is for the back surface. The object distance is positive in front of the lens. The image distance is positive in the direction of the refracted light (behind the lens). The magnification is positive for an upright image, negative for an inverted image.



Equations:

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \qquad \frac{1}{f} = (n - 1)\cdot\left(\frac{1}{r1} - \frac{1}{r2}\right) \qquad m = \frac{-v}{u}$$

Example: Given $r1$=5_cm, $r2$=20_cm, n=1.5, u=50_cm, solves f=13.3333_cm, v=18.1818_cm, m= −.3636.

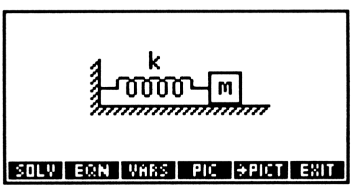
Oscillations

Variable Names and Descriptions

ω	Angular frequency
ϕ	Phase angle
Θ	Cone angle
a	Acceleration at *t*
f	Frequency
G	Shear modulus of elasticity
h	Cone height
I	Moment of inertia
J	Polar moment of inertia
k	Spring constant
L	Length of pendulum
m	Mass
t	Time
T	Period
v	Velocity at *t*
x	Displacement at *t*
xm	Displacement amplitude

Mass-Spring System

These three equations describe the linear oscillatory motion of an object attached to a spring. The motion is harmonic if the displacement of the object from its rest position is small. The equations assume frictional forces are negligible.



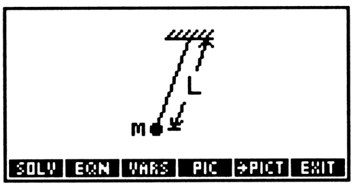
Equations:

$$\omega = \sqrt{\frac{k}{m}} \qquad T = \frac{2 \cdot \pi}{\omega} \qquad \omega = 2 \cdot \pi \cdot f$$

Example: Given k=20_N/m, m=5_kg, solves ω=2_r/s, T=3.1416_s, f=.3183_Hz.

Simple Pendulum

These three equations describe the motion of a simple pendulum, in which a mass hangs from a string and swings beneath the pivot. The equations assume the angular displacement from the rest position is small (the sine of the displacement angle equals the angle in radians) and the mass of the string is negligible.



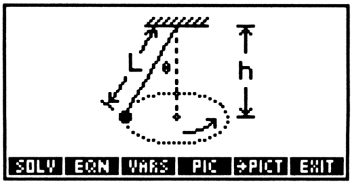
Equations:

$$\omega = \sqrt{\frac{g}{L}} \qquad T = \frac{2 \cdot \pi}{\omega} \qquad \omega = 2 \cdot \pi \cdot f$$

Example: Given L=15_cm, solves ω=8.0856_r/s, T=.7771_s, f=1.2869_Hz.

Conical Pendulum

These four equations describe the motion of a conical pendulum, in which a mass hangs from a string and moves in a horizontal circle. The motion of the string describes the surface of a right, circular cone. The equations assume the mass of the string is negligible. (A guess for Θ can help find the desired solution.)



Equations:

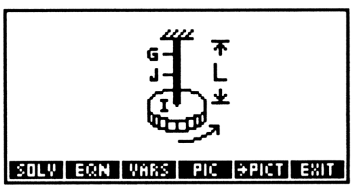
$$\omega = \sqrt{\frac{g}{h}} \qquad h = L \cdot COS(\Theta) \qquad T = \frac{2 \cdot \pi}{\omega}$$

$$\omega = 2 \cdot \pi \cdot f$$

Example: Given L=25_cm, h=20_cm, solves Θ=36.899_°, T=.8973_s, ω=7.0024_r/s, f=1.1145_Hz.

Torsional Pendulum

These three equations describe the motion of a torsional pendulum, in which the restorative force is caused by the twisting of a suspension wire. The equations assume the restoring torque is directly proportional to the angular displacement and the mass of the suspension wire is negligible.



Equations:

$$\omega = \sqrt{\frac{G \cdot J}{L \cdot I}} \qquad T = \frac{2 \cdot \pi}{\omega} \qquad \omega = 2 \cdot \pi \cdot f$$

Example: Given G=1000_kPa, J=17_mm^4, L=26_cm, I=50_kg*m^2, solves ω=1.1435E−3_r/s, f=1.8200E−4_Hz, T=5494.4862_s.

Simple Harmonic

These four equations describe the instantaneous displacement, velocity, and acceleration of an object with one-dimensional, simple harmonic motion. (Guesses for ω, t, and ϕ can help find the desired solution.)

Equations:

$$x = xm \cdot COS(\omega \cdot t + \phi) \qquad a = -\omega^2 \cdot xm \cdot COS(\omega \cdot t + \phi)$$

$$v = -\omega \cdot xm \cdot SIN(\omega \cdot t + \phi) \qquad \omega = 2 \cdot \pi \cdot f$$

Example: Given *xm*=10_cm, ω=15_r/s, ϕ=25_°, *t*=25_μs, solves *x*=9.0615_cm, *v*=−0.6344_m/s, *a*=−20.3884_m/s^2, *f*=2.3873_Hz.

Plane Geometry

Variable Names and Descriptions

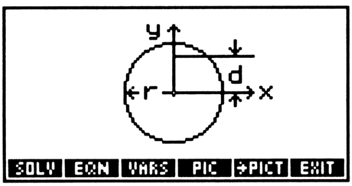
β	Central angle of polygon
Θ	Vertex angle of polygon
A	Area
b	Base length (Rectangle, Triangle), or Length of semiaxis in *x* direction (Ellipse)
C	Circumference
d	Distance to rotation axis in *y* direction
h	Height (Rectangle, Triangle), or Length of semiaxis in *y* direction (Ellipse)
I,Ix	Moment of inertia about *x* axis
Id	Moment of inertia in *x* direction at *d*
Iy	Moment of inertia about *y* axis
J	Polar moment of inertia at centroid
L	Side length of polygon
n	Number of sides

Variable Names and Descriptions (continued)

P	Perimeter
r	Radius
ri,ro	Inside and outside radii
rs	Distance to side of polygon
rv	Distance to vertex of polygon
v	Horizontal distance to vertex

Circle

These five equations describe certain properties of a circle, including moments of inertia.



Equations:

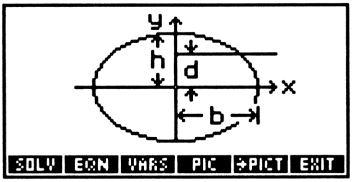
$$A = \pi \cdot r^2 \qquad C = 2 \cdot \pi \cdot r$$

$$I = \frac{\pi \cdot r^4}{4} \qquad J = \frac{\pi \cdot r^4}{2} \qquad Id = I + A \cdot d^2$$

Example: Given *r*=5_cm, *d*=1.5_cm, solves *C*=31.4159_cm, *A*=78.5398_cm^2, *I*=4908738.5_mm^4, *J*=9817477.0_mm^4, *Id*=6675884.4_mm^4.

Ellipse

These five equations describe certain properties of an ellipse, including moments of inertia. The equation for the circumference *C* is an approximation. (You can find the moments of inertia about both axes by swapping the major and minor semiaxis lengths.)



Equations:

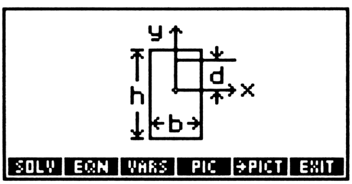
$$A = \pi \cdot b \cdot h \qquad I = \frac{\pi \cdot b \cdot h^3}{4} \qquad J = \frac{\pi \cdot b \cdot h}{4} \cdot \left(b^2 + h^2\right)$$

$$C = 2 \cdot \pi \cdot \sqrt{\frac{b^2 + h^2}{2}} \qquad Id = I + A \cdot d^2$$

Example: Given b = 17.85_μm, h = 78.9725_μin, d = .00000012_ft, solves A = 1.1249E−6_cm^2, C = 7.9805E−3_cm, I = 1.1315E−10_mm^4, J = 9.0733E−9_mm^4, Id = 1.1330E−10_mm^4.

Rectangle

These five equations describe certain properties of a rectangle, including moments of inertia. (You can find the moments of inertia about both axes by swapping the base and height lengths.)



Equations:

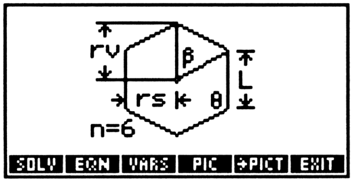
$$A = b \cdot h \qquad I = \frac{b \cdot h^3}{12} \qquad J = \frac{b \cdot h}{12} \cdot \left(b^2 + h^2\right)$$

$$P = 2 \cdot b + 2 \cdot h \qquad Id = I + A \cdot d^2$$

Example: Given b = 4_chain, h = 7_rd, d = 39.26_in, set guesses for I, J, and Id in km^4, solves A = 28328108.2691_cm^2, P = 23134.3662_cm, I = 2.9257E−7_km^4, J = 1.8211E−6_km^4, Id = 2.9539E−7_km^4.

Regular Polygon

These six equations describe certain dimensions of a regular polygon. (A guess for *n* can help find the desired solution.)



Equations:

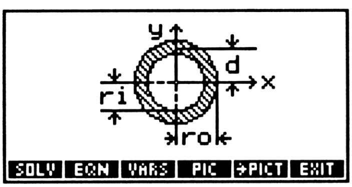
$$A = \frac{\frac{1}{4} \cdot n \cdot L^2}{\mathrm{TAN}\left(\frac{180}{n}\right)} \qquad rs = \frac{\frac{L}{2}}{\mathrm{TAN}\left(\frac{180}{n}\right)} \qquad rv = \frac{\frac{L}{2}}{\mathrm{SIN}\left(\frac{180}{n}\right)} \qquad \beta = \frac{360}{n}$$

$$P = n \cdot L \qquad \theta = \frac{n-2}{n} \cdot 180$$

Example: Given $n=8$, $L=.5\_yd$, solves $A=10092.9501\_cm\char`^2$, $P=365.7600\_cm$, $rs=55.1889\_cm$, $rv=59.7361\_cm$, $\theta=135\_°$, $\beta=45\_°$.

Circular Ring

These four equations describe certain properties of a circular ring, including moments of inertia.



Equations:

$$A = \pi \cdot \left(ro^2 - ri^2 \right)$$

$$I = \pi \cdot \left(\left(\frac{ro + ri}{2} \right)^3 \right) \cdot (ro - ri)$$

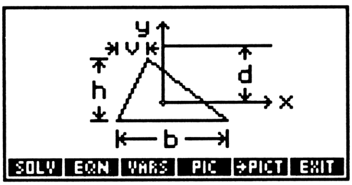
$$J = 2 \cdot \pi \cdot \left(\left(\frac{ro + ri}{2} \right)^3 \right) \cdot (ro - ri)$$

$$Id = I + A \cdot d^2$$

Example: Given *ro*=4_μ, *ri*=25.0_kÅ, *d*=.1_mil, solves *A*=3.0631E−7_cm^2, *I*=1.6177E−10_mm^4, *J*=3.2353E−10_mm^4, *Id*=3.5938E−10_mm^4.

Triangle

These six equations describe certain properties of a triangle, including base, height, and moments of inertia.



Equations:

$$A = \frac{b \cdot h}{2}$$

$$P = b + \sqrt{v^2 + h^2} + \sqrt{(b - v)^2 + h^2}$$

$$Ix = \frac{b \cdot h^3}{36}$$

$$Iy = \frac{b \cdot h}{36} \cdot \left(b^2 - b \cdot v + v^2 \right)$$

$$J = \frac{b \cdot h}{36} \cdot \left(h^2 + b^2 - b \cdot v + v^2 \right)$$

$$Id = Ix + A \cdot d^2$$

Example: Given *h*=4.33012781892_in, *v*=2.5_in, *P*=15_in, *d*=2_in, solves *b*=5.0000_in, *Ix*=11.2764_in^4, *Iy*=11.2764_in^4, *J*=22.5527_in^4, *A*=10.8253_in^2, *Id*=54.5776_in^4.

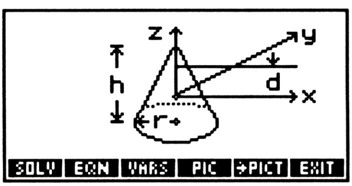
Solid Geometry

Variable Names and Descriptions

A	Total surface area
b	Base length
d	Distance to rotation axis in *z* direction
h	Height in *z* direction (Cone, Cylinder), or Height in *y* direction (Parallelepiped)
I,Ixx	Moment of inertia about *x* axis
Id	Moment of inertia in *x* direction at *d*
Izz	Moment of inertia about *z* axis
m	Mass
r	Radius
t	Thickness in *z* direction
V	Volume

Cone

These five equations describe certain properties of a cone, including moments of inertia.



Equations:

$$V = \frac{\pi}{3} \cdot r^2 \cdot h$$

$$A = \pi \cdot r^2 + \pi \cdot r \cdot \sqrt{r^2 + h^2}$$

$$Ixx = \frac{3}{20} \cdot m \cdot r^2 + \frac{3}{80} \cdot m \cdot h^2$$

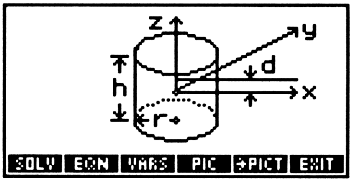
$$Izz = \frac{3}{10} \cdot m \cdot r^2$$

$$Id = Ixx + m \cdot d^2$$

Example: Given r=7_cm, h=12.5_cm, m=12.25_kg, d=3.5_cm, solves V=641.4085_cm^3, A=468.9953_cm^2, Ixx=0.0162_kg*m^2, Izz=0.0180_kg*m^2, Id=0.0312_kg*m^2.

Cylinder

These five equations describe certain properties of a cylinder, including moments of inertia.



Equations:

$$V = \pi \cdot r^2 \cdot h$$

$$A = 2 \cdot \pi \cdot r^2 + 2 \cdot \pi \cdot r \cdot h$$

$$Ixx = \frac{1}{4} \cdot m \cdot r^2 + \frac{1}{12} \cdot m \cdot h^2$$

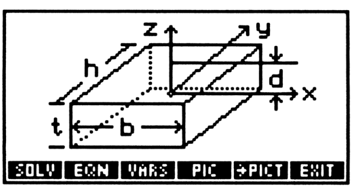
$$Izz = \frac{1}{2} \cdot m \cdot r^2$$

$$Id = Ixx + m \cdot d^2$$

Example: Given r=8.5_in, h=65_in, m=12000_lbs, d=2.5_in, solves V=14753.7045_in^3, A=3925.4200_in^2, Ixx=4441750_lb*in^2, Izz=433500_lb*in^2, Id=4516750_lb*in^2.

Parallelepiped

These four equations describe certain properties of a parallelepiped, including moments of inertia.



Equations:

$$V = b \cdot h \cdot t$$

$$A = 2 \cdot (b \cdot h + b \cdot t + h \cdot t)$$

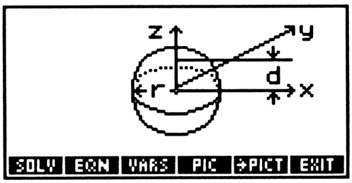
$$I = \frac{1}{12} \cdot m \cdot \left(h^2 + t^2\right)$$

$$Id = I + m \cdot d^2$$

Example: Given b=36_in, h=12_in, t=72_in, m=83_lb, d=7_in, solves V=31104_in^3, A=7776_in^2, I=36852_lb*in^2, Id=40919_lb*in^2.

Sphere

These four equations describe certain properties of a sphere, including moments of inertia.



Equations:

$$V = \frac{4}{3} \cdot \pi \cdot r^3$$

$$A = 4 \cdot \pi \cdot r^2$$

$$I = \frac{2}{5} \cdot m \cdot r^2$$

$$Id = I + m \cdot d^2$$

Example: Given d=14_cm, m=3.75_kg, Id=486.5_lb*in^2, solves r=21.4273_cm, V=41208.7268_cm^3, A=5769.5719_cm^2, I=0.0689_kg*m^2.

Solid State Devices

Variable Names and Descriptions

αF	Forward common-base current gain
αR	Reverse common-base current gain
γ	Body factor
λ	Modulation parameter
μn	Electron mobility
ϕp	Fermi potential
ΔL	Length adjustment (PN Step Junctions), or Channel encroachment (NMOS Transistors)
ΔW	Width adjustment (PN Step Junctions), or Width contraction (NMOS Transistors)
a	Channel thickness
Aj	Effective junction area
BV	Breakdown voltage
Cj	Junction capacitance per unit area
Cox	Silicon dioxide capacitance per unit area
E1	Breakdown-voltage field factor
Emax	Maximum electric field
G0	Channel conductance
gds	Output conductance
gm	Transconductance
I	Diode current
IB	Total base current
IC	Total collector current
ICEO	Collector current (collector-to-base open)
ICO	Collector current (emitter-to-base open)
ICS	Collector-to-base saturation current

Variable Names and Descriptions (continued)

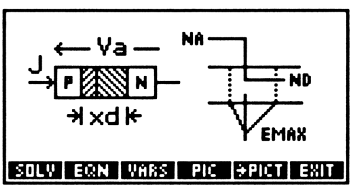
ID,IDS	Drain current
IE	Total emitter current
IES	Emitter-to-base saturation current
IS	Transistor saturation current
J	Current density
Js	Saturation current density
L	Drawn mask length (PN Step Junctions), or Drawn gate length (NMOS Transistors), or Channel length (JFETs)
Le	Effective gate length
NA	P-side doping (PN Step Junctions), or Substrate doping (NMOS Transistors)
ND	N-side doping (PN Step Junctions), or N-channel doping (JFETs)
T	Temperature
tox	Gate silicon dioxide thickness
Va	Applied voltage
VBC	Base-to-collector voltage
VBE	Base-to-emitter voltlage
Vbi	Built-in voltage
VBS	Substrate voltage
VCEsat	Collector-to-emitter saturation voltage
VDS	Applied drain voltage
VDsat	Saturation voltage
VGS	Applied gate voltage
Vt	Threshold voltage
Vt0	Threshold voltage (at zero substrate voltage)
W	Drawn mask width (PN Step Junctions), or Drawn width (NMOS Transistors), or Channel width (JFETs)

Variable Names and Descriptions (continued)

We	Effective width
xd	Depletion-region width
xdmax	Depletion-layer width
xj	Junction depth

PN Step Junctions

These eight equations describe the behavior of a silicon PN-junction diode. The equations use the Shockley equation relating saturation current density, applied voltage, and temperature. They use the "two-sided step-junction" model, in which the doping density changes abruptly at the junction from the N-side density to the P-side density. The equations assume the device is made of silicon, the current density is determined by minority carriers injected across the depletion region, and the PN junction is rectangular in its layout. The equations use the intrinsic density of silicon (see "SIDENS Function" in chapter 8). When applying this set of equations, you should keep in mind practical ranges of parameters. For example, the doping densities are in the range 10^{13} to 10^{19} cm^{-3}, resulting in a built-in voltage of 0.5 to 0.9 volts. Maximum electric fields are in the range 10 to 110 kV/cm. Breakdown voltage is in the range of a few volts to several thousand volts. The junction model is valid for a range of temperatures from 77 to 500 K.



Equations:

$$Vbi = \frac{k \cdot T}{q} \cdot LN\left(\frac{NA \cdot ND}{ni^2}\right)$$

$$xd = \sqrt{\frac{2 \cdot \varepsilon si \cdot \varepsilon 0}{q} \cdot (Vbi - Va) \cdot \left(\frac{1}{NA} + \frac{1}{ND}\right)}$$

$$Cj = \frac{\varepsilon si \cdot \varepsilon 0}{xd}$$

$$Emax = \frac{2 \cdot (Vbi - Va)}{xd}$$

$$BV = \frac{\varepsilon si \cdot \varepsilon 0 \cdot E1^2}{2 \cdot q} \cdot \left(\frac{1}{NA} + \frac{1}{ND}\right)$$

$$J = Js \cdot \left(e^{\frac{q \cdot Va}{k \cdot T}} - 1\right)$$

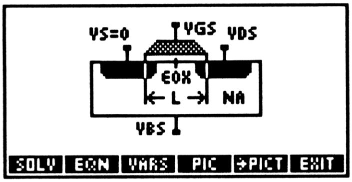
$$Aj = (W + 2 \cdot \Delta W) \cdot (L + 2 \cdot \Delta L) + \pi \cdot (W + L + 2 \cdot \Delta W + 2 \cdot \Delta L) \cdot xj + 2 \cdot \pi \cdot xj^2$$

$$I = J \cdot Aj$$

Example: Given *ND*=1E22_cm^−3, *NA*=1E15_cm^−3, *T*=26.85_°C, *Js*=1E−6_μA/cm^2, *Va*=−20_V, *E1*=3.3E5_V/cm, *W*=10_μ, Δ*W*=1_μ, *L*=10_μ, Δ*L*=1_μ, *xj*=2_μ, solves *Vbi*=.9962_V, *xd*=5.2551_μ, *Cj*=2005.0141_pF/cm^2, *Emax*=79908.5240_V/cm, *BV*=358.0825_V, *J*=−1.0E−12_A/cm^2, *Aj*=3.1993E−6_cm^2, *I*=−3.1993E−15_mA.

NMOS Transistors

These 10 equations describe the behavior of a silicon NMOS transistor. The equations use a two-port network model, include linear and non-linear regions in the device characteristics, and are based on a gradual-channel approximation (the electric fields in the direction of current flow are small compared to the electric fields in the direction perpendicular to current flow). The drain current and transconductance are calculated differently depending on whether the transistor is in the linear, saturated, or cutoff region. The equations use the intrinsic density of silicon (see "SIDENS Function" in chapter 8). The equations assume the physical geometry of the device is a rectangle, second-order length-parameter effects are negligible, short-channel, hot-carrier, and velocity-saturation effects are negligible, and subthreshold currents are negligible.



Equations:

$$We = W - 2 \cdot \Delta W$$

$$Le = L - 2 \cdot \Delta L$$

$$Cox = \frac{\varepsilon ox \cdot \varepsilon 0}{tox}$$

$$IDS = Cox \cdot \mu n \cdot \left(\frac{We}{Le} \right) \cdot \left((VGS - Vt) \cdot VDS - \frac{VDS^2}{2} \right) \cdot (1 + \lambda \cdot VDS)$$

$$\gamma = \frac{\sqrt{2 \cdot \varepsilon si \cdot \varepsilon 0 \cdot q \cdot NA}}{Cox}$$

$$Vt = Vt0 + \gamma \cdot \left(\sqrt{2 \cdot ABS(\phi p) + ABS(VBS)} - \sqrt{2 \cdot ABS(\phi p)} \right)$$

$$\phi p = \frac{-k \cdot T}{q} \cdot LN\left(\frac{NA}{ni} \right)$$

$$gds = IDS \cdot \lambda$$

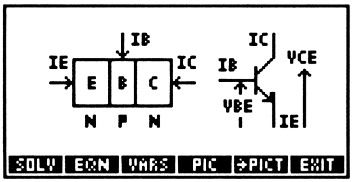
$$gm = \sqrt{Cox \cdot \mu n \cdot \left(\frac{We}{Le} \right) \cdot (1 + \lambda \cdot VDS) \cdot 2 \cdot IDS}$$

$$VDsat = VGS - Vt$$

Example: Given *tox* = 700_Å, *NA* = 1E15_cm^–3, *μn* = 600_cm^2/(V∗s), *T* = 26.85_°C, *Vt0* = .75_V, *VGS* = 5_V, *VBS* = 0_V, *VDS* = 5_V, *W* = 25_μ, *ΔW* = 1_μ, *L* = 4_μ, *ΔL* = .75_μ, *λ* = .05_1/V, solves *We* = 23_μ, *Le* = 2.5_μ, *Cox* = 49330.4750_pF/cm^2, *γ* = .3725_V^.5, *ϕp* = –.2898_V, *Vt* = .75_V, *VDsat* = 4.25_V, *IDS* = 3.0741_mA, *gds* = 1.5370E–4_S, *gm* = 1.4466_mA/V.

Bipolar Transistors

These eight equations describe the behavior of an NPN silicon bipolar transistor. The equations are based on large-signal models developed by J.J. Ebers and J.L. Moll. The equations include the reciprocity theorem, which defines the strong relationship between forward and reverse common-base current gains and saturation currents. The offset voltage is calculated differently depending on whether the transistor is saturated or not. The equations also include the special conditions when the emitter-base or collector-base junction is open, which are convenient for measuring transistor parameters.



Equations:

$$IE = -IES \cdot \left(e^{\frac{q \cdot VBE}{k \cdot T}} - 1\right) + \alpha R \cdot ICS \cdot \left(e^{\frac{q \cdot VBC}{k \cdot T}} - 1\right)$$

$$IC = -ICS \cdot \left(e^{\frac{q \cdot VBC}{k \cdot T}} - 1\right) + \alpha F \cdot IES \cdot \left(e^{\frac{q \cdot VBE}{k \cdot T}} - 1\right)$$

$$IS = \alpha F \cdot IES$$

$$IS = \alpha R \cdot ICS$$

$$IB + IE + IC = 0$$

$$ICO = ICS \cdot (1 - \alpha F \cdot \alpha R)$$

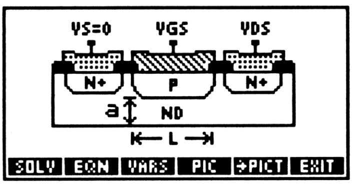
$$ICEO = \frac{ICO}{1 - \alpha F}$$

$$VCEsat = \frac{k \cdot T}{q} \cdot LN\left(\frac{1 + \frac{IC}{IB} \cdot (1 - \alpha R)}{\alpha R \cdot \left(1 - \frac{IC}{IB} \cdot \left(\frac{1 - \alpha F}{\alpha F}\right)\right)}\right)$$

Example: Given $IES = 1E-5\_nA$, $ICS = 2E-5\_nA$, $T = 26.85\_°C$, $\alpha F = .98$, $\alpha R = .49$, $IC = 1\_mA$, $VBC = -10\_V$, solves $VBE = .6553\_V$, $IS = 0.0000098\_nA$, $ICO = .000010396\_nA$, $ICEO = .0005198\_nA$, $IE = -1.0204\_mA$, $IB = .0204\_mA$, $VCEsat = 0\_V$.

JFETs

These seven equations describe the behavior of a silicon N-channel junction field-effect transistor (JFET). The equations are based on the single-sided step-junction approximation, which assumes the gates are heavily doped compared to the channel doping. The drain current is calculated differently depending on whether the gate-junction depletion-layer thickness is less than or greater than the channel thickness. The equations use the intrinsic density of silicon (see "SIDENS Function" in chapter 8). The equations assume the channel is uniformly doped and end effects (such as contact, drain, and source resistances) are negligible.



Equations:

$$Vbi = \frac{k \cdot T}{q} \cdot LN\left(\frac{ND}{ni}\right)$$

$$xdmax = \sqrt{\frac{2 \cdot \varepsilon si \cdot \varepsilon 0}{q \cdot ND} \cdot (Vbi - VGS + VDS)}$$

$$G0 = q \cdot ND \cdot \mu n \cdot \left(\frac{a \cdot W}{L}\right)$$

$$ID = G0 \cdot \left(VDS - \frac{2}{3} \cdot \sqrt{\frac{2 \cdot \varepsilon si \cdot \varepsilon 0}{q \cdot ND \cdot a^2}} \cdot \left((Vbi - VGS + VDS)^{\frac{3}{2}} - (Vbi - VGS)^{\frac{3}{2}} \right) \right)$$

$$VDsat = \frac{q \cdot ND \cdot a^2}{2 \cdot \varepsilon si \cdot \varepsilon 0} - (Vbi - VGS)$$

$$Vt = Vbi - \frac{q \cdot ND \cdot a^2}{2 \cdot \varepsilon si \cdot \varepsilon 0}$$

$$gm = G0 \cdot \left(1 - \sqrt{\frac{2 \cdot \varepsilon si \cdot \varepsilon 0}{q \cdot ND \cdot a^2} \cdot (Vbi - VGS)} \right)$$

Example: Given *ND* = 1E16_1/cm^3, *W* = 6_μ, *a* = 1_μ, *L* = 2_μ, *μn* = 1248_cm^2/(V*s), *VGS* = −4_V, *VDS* = 4_V, *T* = 26.85_°C, solves *Vbi* = .3493_V, *xdmax* = 1.0479_μ, *G0* = 5.9986E−4_S, *ID* = .2268_mA, *VDsat* = 3.2537_V, *Vt* = −7.2537_V, *gm* = .1462_mA/V.

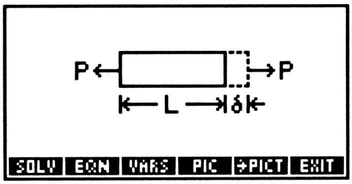
Stress Analysis

Variable Names and Descriptions

δ	Elongation
ε	Normal strain
γ	Shear strain
ϕ	Angle of twist
σ	Normal stress
$\sigma 1$	Maximum principal normal stress
$\sigma 2$	Minimum principal normal stress
σavg	Normal stress on plane of maximum shear stress
σx	Normal stress in *x* direction
$\sigma x1$	Normal stress in rotated-*x* direction
σy	Normal stress in *y* direction
$\sigma y1$	Normal stress in rotated-*y* direction
τ	Shear stress
τmax	Maximum shear stress
$\tau x1y1$	Rotated shear stress
τxy	Shear stress
Θ	Rotation angle
$\Theta p1$	Angle to plane of maximum principal normal stress
$\Theta p2$	Angle to plane of minimum principal normal stress
Θs	Angle to plane of maximum shear stress
A	Area
E	Modulus of elasticity
G	Shear modulus of elasticity
J	Polar moment of inertia
L	Length
P	Load
r	Radius
T	Torque

Normal Stress

These three equations relate the fundamental concept of stress and strain: A load (stress) exerted on an element produces a change in length (strain). The equations assume a linear relationship between normal stress and normal strain, which is valid only for small values of stress and strain and only until the material yields to the stress.



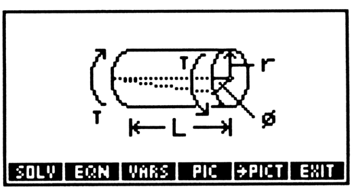
Equations:

$$\sigma = E \cdot \varepsilon \qquad \varepsilon = \frac{\delta}{L} \qquad \sigma = \frac{P}{A}$$

Example: Given P=40000_lbf, L=8_in, A=3.14159265359_in^2, E=10E6_psi, solves δ=0.0102_in, ε=0.0013, σ=12732.3954_psi.

Shear Stress

These three equations relate the concepts of shear stress and shear strain. A torque (shear stress) applied to a cylindrical element produces a twist (shear strain). The equations assume a strictly torsional load and a linear relationship between shear stress and shear strain, which is valid only for small values of shear strain and only until the material yields to the stress.



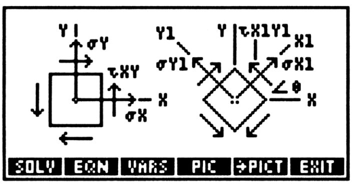
Equations:

$$\tau = G \cdot \gamma \qquad \gamma = \frac{r \cdot \phi}{L} \qquad \tau = \frac{T \cdot r}{J}$$

Example: Given L=6_ft, r=2_in, J=10.4003897419_in^4, G=12000000_psi, τ=12000_psi, solves T=5200.1949_ft*lbf, ϕ=12.96_°, γ=0.3600_°.

Stress on an Element

These three equations relate stress and strain along reference axes to stress and strain along rotated axes. The rotated axes are orthogonal, and they're in the same plane as the reference axes. Stresses and strains are positive in the directions shown. (A guess for Θ can help find the desired solution.)



Equations:

$$\sigma x1 = \frac{\sigma x + \sigma y}{2} + \frac{\sigma x - \sigma y}{2} \cdot COS(2 \cdot \Theta) + \tau xy \cdot SIN(2 \cdot \Theta)$$

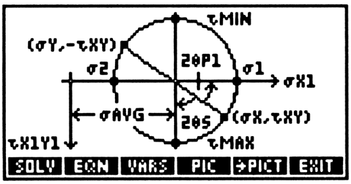
$$\sigma x1 + \sigma y1 = \sigma x + \sigma y$$

$$\tau x1y1 = -\left(\frac{\sigma x - \sigma y}{2}\right) \cdot SIN(2 \cdot \Theta) + \tau xy \cdot COS(2 \cdot \Theta)$$

Example: Given σx=15000_kPa, σy=4755_kPa, τxy=7500_kPa, Θ=30_°, solves $\sigma x1$=18933.9405_kPa, $\sigma y1$=821.0595_kPa, $\tau x1y1$= −686.2151_kPa.

Mohr's Circle

These seven equations represent the graphical form of stress-and-strain transformations called Mohr's circle. Mohr's circle is useful for defining the relationships between normal and shear stresses acting on various inclined planes at any point in a stressed object. The applied normal and shear stresses define the circle, from which stresses are defined at any rotation from the original *xy*-axis. The principal *x1y1*-axis is the one along which the stresses are algebraically highest and lowest and the shear stress is zero. The equations assume the problem is two-dimensional — the results don't extend to three dimensions. (A guess for *Θp1* can help find the desired solution.)



Equations:

$$\sigma 1 = \frac{\sigma x + \sigma y}{2} + \sqrt{\left(\frac{\sigma x - \sigma y}{2}\right)^2 + \tau xy^2}$$

$$\sigma 1 + \sigma 2 = \sigma x + \sigma y$$

$$\mathrm{SIN}(2 \cdot \Theta p1) = \frac{\tau xy}{\sqrt{\left(\frac{\sigma x - \sigma y}{2}\right)^2 + \tau xy^2}}$$

$$\Theta p2 = \Theta p1 + 90 \qquad \Theta s = \Theta p1 - 45$$

$$\tau max = \frac{\sigma 1 - \sigma 2}{2} \qquad \sigma avg = \frac{\sigma x + \sigma y}{2}$$

Example: Given *σx* = −5600_psi, *σy* = −18400_psi, *τxy* = 4800_psi, solves *σ1* = −4000_psi, *σ2* = −20000_psi, *Θp1* = 18.4349_°, *Θp2* = 108.4349_°, *τmax* = 8000_psi, *Θs* = −26.5651_°, *σavg* = −12000_psi.

Waves

Variable Names and Descriptions

β	Sound level
λ	Wavelength
ω	Angular frequency
ρ	Density of medium
B	Bulk modulus of elasticity
f	Frequency
I	Sound intensity
k	Angular wave number
s	Longitudinal displacement at x and t
sm	Longitudinal amplitude
t	Time
v	Speed of sound in medium (Sound Waves), or Wave speed (Transverse Waves, Longitudinal Waves)
x	Position
y	Transverse displacement at x and t
ym	Transverse amplitude

Transverse Waves

These four equations describe the instantaneous behavior of transverse waves—the waves oscillate at right angles to the direction of propagation. Waves generated in a string are transverse waves. The equations give the instantaneous displacement of the vibrating material in terms of position and time. The equations assume the following effects are negligible: damping effects, gravitational effects, and any effects of the transmitting medium on the dispersion characteristics of the waves. (Guesses for k, x, ω, and t can help find the desired solution.)

Equations:

$$y = ym \cdot SIN(k \cdot x - \omega \cdot t) \qquad k = \frac{2 \cdot \pi}{\lambda} \qquad \omega = 2 \cdot \pi \cdot f$$

$$v = \lambda \cdot f$$

Example: Given *ym* = 6.37_cm, *k* = 32.11_r/cm, *x* = .03_cm, *ω* = 7000_r/s, *t* = 1_s, solves *f* = 1114.0846_Hz, *λ* = .1957_cm, *y* = 2.6655_cm, *v* = 218.0006_cm/s.

Longitudinal Waves

These four equations describe the instantaneous behavior of longitudinal waves — the waves oscillate direction of propagation. Sound waves generated in a long tube are longitudinal waves. The equations give the instantaneous displacement of the vibrating material in terms of position and time. The equations assume the following effects are negligible: damping effects, gravitational effects, and any effects of the transmitting medium on the dispersion characteristics of the waves. (Guesses for *k*, *x*, *ω*, and *t* can help find the desired solution.)

Equations:

$$s = sm \cdot COS(k \cdot x - \omega \cdot t) \qquad k = \frac{2 \cdot \pi}{\lambda} \qquad \omega = 2 \cdot \pi \cdot f$$

$$v = \lambda \cdot f$$

Example: Given *sm* = 6.37_cm, *k* = 32.11_r/cm, *x* = .03_cm, *ω* = 7000_r/s, *t* = 1_s, solves *s* = 5.7855_cm, *v* = 2.1800_m/s, *λ* = .1957_cm, *f* = 1114.0846_Hz.

Sound Waves

These four equations describe the properties of sound waves in a medium. The medium is characterized by its density and bulk modulus of elasticity. The sound waves are characterized by their intensity, velocity, frequency, and amplitude. The sound intensity is related to a standard intensity reflecting the lower limit of human auditory senses. These equations assume the displacement is small (first-order approximations are valid) and the wave is sinusoidal.

Equations:

$$v = \sqrt{\frac{B}{\rho}}$$

$$\beta = 10 \cdot \mathrm{LOG}\left(\frac{I}{I0}\right)$$

$$I = \frac{1}{2} \cdot \rho \cdot v \cdot \omega^2 sm^2$$

$$\omega = 2 \cdot \pi \cdot f$$

Example: Given sm = 10_cm, ω = 6000_r/s, B = 12500_kPa, ρ = 65_kg/m^3, solves v = 438.5290_m/s, I = 5130789412.97_W/m^2, β = 217.1018_dB, f = 954.9297_Hz.

4

Periodic Table

The Periodic Table application gives you access to a large collection of information about 106 chemical elements.

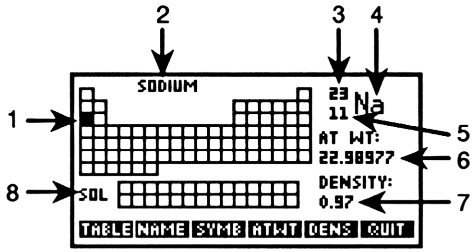
This chapter shows you how to do the following:

- Scan the Periodic Table display and find the names, symbols, and properties of elements.
- Calculate molecular weights.
- Include Periodic Table commands in programs.

Using the Periodic Table

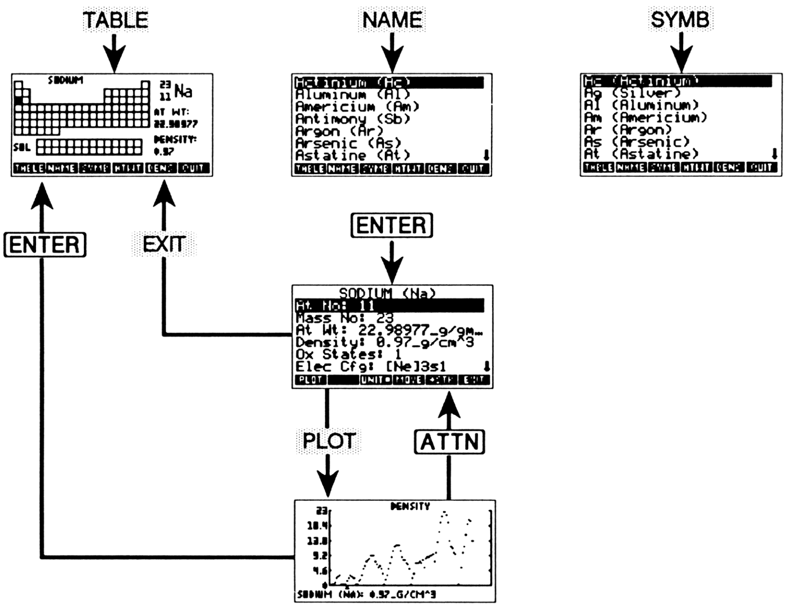
To start the Periodic Table, follow these steps:

1. Get the LIBRARY menu ([←] [LIBRARY]).
2. Get the Periodic Table main menu by pressing PRTBL.
3. View the Periodic Table display by pressing PERTB.



The display shows a picture of the periodic table. Each square represents one element. The black pointer (1) marks the location of the current element. For the current element, the display also shows the element name (2), mass number (3), symbol (4), atomic number (5), atomic weight (6), density (7), and physical state (8) — solid, liquid, gas, or synthetic — at standard temperature for gases and at room temperature for others. Each of these properties is listed fully with units in the catalog of properties — see "Finding Element Properties" later in this chapter.

The Periodic Table is a special environment in which you can find information about elements and molecules. The following diagram shows how to switch among the displays and catalogs.



In the main Periodic Table display, you can perform the operations listed in the following table.

Operations in the Periodic Table Display

Key	Action
[▲] [▼] [◀] [▶]	Moves the pointer in the table, wrapping around rows or columns. With [↱], moves the pointer to the extreme of the current row or column. [◀] or [▶] wraps to the adjacent row. [▲] or [▼] moves to or from the rare-earth area.
[α]...[ENTER]	Jumps the pointer to the element having the symbol entered, or calculates molecular weights. Multiple-character entry allowed.
ATWT	Puts the atomic weight of the current element on the stack.
DENS	Puts the density of the current element on the stack.
QUIT	Exits from the application.
[ATTN]	Exits from the application.

Finding Element Names and Symbols

The Periodic Table provides two catalogs containing names and symbols of elements in alphabetical order. You can use these catalogs to find names or symbols or to make a certain element the current element:

- To find the symbol for a certain element name, press NAME. Use [▲] and [▼] to move to the element, or use [α] to enter the first letter of the name.
- To find the name for a certain element symbol, press SYMB. Use [▲] and [▼] to move to the element, or use [α] to enter the first letter of the symbol.
- To make a certain element the current element, do either of the previous steps, then press TABLE.

You can also find element properties from the name and symbol catalogs — see the next topic.

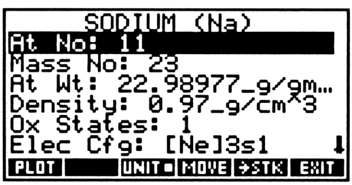
Finding Element Properties

The Periodic Table contains a collection of physical properties of elements, such as melting-point temperature and heat of vaporization. Values are based on the “Periodic Table of the Elements” published by the Sargent-Welch Scientific Company — they may differ from values from other sources.

To get the catalog of properties for a certain element, follow these steps:

1. Select the element in one of these ways:
   - Move the pointer to the element in the Periodic Table display.
   - Press [α], type the symbol for the element, then press [ENTER].
   - Get the name or symbol catalog, then move the highlight to the element.
2. Press [ENTER].

The catalog shows each property name and value, including units (if units are used).



The property catalog provides the operations listed in the following table. (See “Viewing and Selecting in a Catalog” in chapter 1 for details.) The property you point to last is the one that’s highlighted in the *next* property catalog.

Operations in the Property Catalog

Key	Action
[▲] [▼]	Moves the highlight up or down the catalog (also [↰] and [↱]).
[α]...	Jumps the highlight to that alpha line.
[ENTER]	Shows a wide (...) item completely — press [ENTER] or [ATTN] to return to the catalog.
PLOT	Plots the highlighted property as a function of atomic number. (See "Plotting Properties" later in this chapter.)
UNIT■	Indicates units used.
UNITS	Indicates units not used.
MOVE	Moves the highlighted property to the top of the catalog for this and future catalogs.
→STK	Puts the highlighted property in level 1 of the stack.
EXIT	Returns to the Periodic Table display at this element.
[ATTN]	Exits the Periodic Table.

The following table lists the properties in the order they're included in the Periodic Table, plus the type of object associated with each property. Certain properties are not numeric — they're represented as string objects.

Properties of Elements

Property	Type of Object*	Property Number
Atomic Number	Real	1
Mass Number †	Real	2
Atomic Weight	Unit	3
Density †	Unit	4
Oxidation States †	String	5
Electronic Configuration	String	6
State	String	7
Melting Point	Unit	8
Boiling Point	Unit	9
Heat of Vaporization	Unit	10
Heat of Fusion	Unit	11
Specific Heat	Unit	12
Group (U.S. customary)	String	13
Family	String	14
Crystal Structure	String	15
Atomic Volume †	Unit	16
Atomic Radius	Unit	17
Covalent Radius	Unit	18
Thermal Conductivity †	Unit	19
Electrical Conductivity †	Unit	20
First Ionization Potential	Unit	21
Electronegativity (Pauling's number)	Unit	22
Oxide Behavior	String	23
Element Name ‡	String	24
Element Symbol ‡	Name	25

* Properties returning unit objects return real objects if units aren't used. Unknown values are returned as the string "–".
† See the notes that follow this table.
‡ Not included in catalog of properties, but listed in the title of the catalog.

Notes about properties: Mass number for a stable element is based on the isotope with the highest percent abundance; for a radioactive element, it's based on the longest half-life. Density for a gas is at 273 K with units of g/l; for others, it's at 300 K with units g/cm^3. Oxidation states are in order of most stable to least stable. Atomic volume for a gas is for its liquid state at the boiling point; for others, it's derived from the density at 300 K. Thermal conductivity is measured at 300 K. Electrical conductivity is measured at 293 K.

Some properties for certain elements are annotated with special information, such as indicating estimated values or special conditions. For example, the note for the atomic weight of francium indicates that the value is for francium's most stable form. Such properties appear in the catalog with * as the first character—you can search for this annotation by pressing [α] [×]. To see the property and annotation, highlight it and press [ENTER].

If you have one or more properties that you use frequently, you can change the catalog to make these properties always appear at the top of the catalog. To move a property to the top of the catalog (for this and future catalogs), highlight the property you want at the top, then press MOVE. (Note that this does not affect the property numbers listed in the previous table.) To restore all properties to their original order, purge variable *PTpar* ([VAR] ['] PTPAR [↰][PURGE]).

The application always displays the property catalog with full precision, regardless of the current display mode.

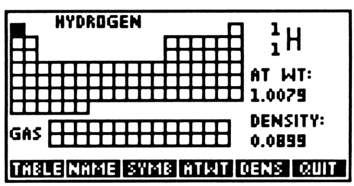
Example: Calculating with Properties. Ella Mental knows that bromine and mercury are normally liquids. She wants to find the temperature intervals for which they're liquid.

Move to the *HOME* directory for this example. Then get the LIBRARY menu.

[↱][HOME]
[↰][LIBRARY]

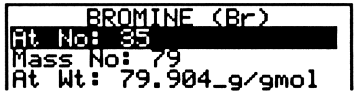
Get the Periodic Table menu, then start the application.

PRTBL
PERTB



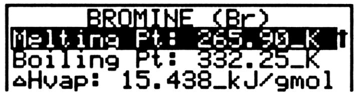
Make bromine the current element — be sure to use proper uppercase and lowercase ([↰]) letters — note that the alpha keyboard locks on. Then get its property catalog. (If UNIT■ isn't flagged with a small square, press that menu key once.)

[α] Br [ENTER]
[ENTER]
(UNITS if needed)



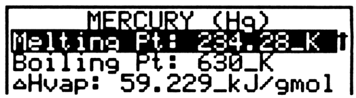
Move to the "Boiling Pt" property and copy it to the stack. Move to the "Melting Pt" property and copy it to the stack.

[α] B →STK
[▲] →STK



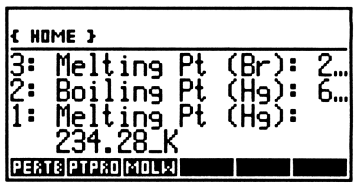
Return to the Periodic Table display. Then make mercury the current element and get its property catalog.

EXIT
NAME
[α] M [▼] [▼] [▼]
[ENTER]



Move to the "Boiling Pt" property and copy it to the stack. Move to the "Melting Pt" property and copy it to the stack. Then return to the stack.

[▼] →STK
[▲] →STK
[ATTN]



Calculate the temperature interval for mercury.

[−]

Calculate the temperature interval for bromine.

[◆] [−]

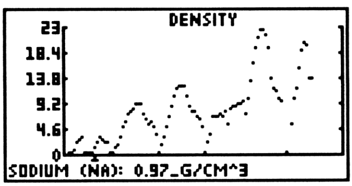
Plotting Properties

You can plot any numeric-valued property as a function of atomic number for all elements. (You *can't* plot non-numeric properties.) This is useful for finding elements with certain characteristics and for observing the periodic nature of many properties.

To plot a property, follow these steps:

1. Get a property catalog for any element.
2. Highlight the property you want to plot.
3. Press PLOT.

The plot includes a pointer just below the plot marking the current element. The bottom line gives the name, symbol, and property value for the current element.



You can scan the plot to find certain elements or values. You can return to the property catalog for the original element or to the Periodic Table display for another element.

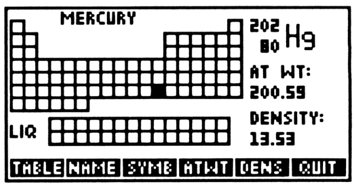
Operations in a Property Plot

Key	Action
[◀] [▶]	Moves the plot pointer to the left or right. With [↰], jumps the pointer 10 elements left or right. With [↱], jumps the pointer to the first or last elements.
[ENTER]	Returns to the Periodic Table display at the element marked by the plot pointer.
[ATTN]	Returns to the previous property catalog.

Example: Plotting Properties. Red Burns wants to find the three elements with the greatest heat conductivities.

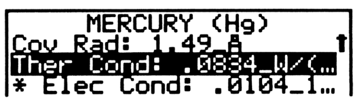
Start the Periodic Table. The current element doesn't matter.

[↰] [LIBRARY]
PRTBL
PERTB



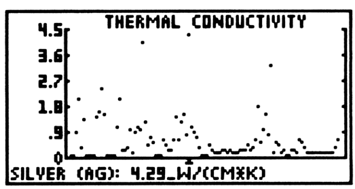
Get the property catalog for the current element, then move to the "Ther Cond" property.

[ENTER]
[α] T



Plot the property. Then move the pointer to each of the three highest points, noting that they're for copper, silver, and gold.

PLOT
[◀] or [▶]. . .



Press [ATTN] [ATTN] to return to the stack.

Putting Information on the Stack

You can put information on the stack from the Periodic Table display, from a catalog of properties, or from a command:

- From the Periodic Table display, press ATWT to put the atomic weight of the current element on the stack, or press DENS to put its density on the stack.
- In a property catalog, press →STK to put any highlighted property on the stack.
- From outside the Periodic Table, use the PTPROP command to put any property on the stack. (See "Using Periodic Table Commands" later in this chapter.)

The stack entry contains a tag showing the property name and element symbol. The value may be a number (with or without units) or a string, depending on the property. You control the use of units — see the next topic.

Choosing Unit Options

The Periodic Table properties and commands have two choices for units: SI units or no units (implied SI units). The numeric values don't change for these choices, but the type of objects put on the stack does depend on your choice.

In any catalog of properties, you can choose SI units or no units using the UNITS menu key. UNIT■ means SI units used, UNITS means units not used. See "Working with and without Units" in chapter 1 for details.

Calculating Molecular Weights

You can calculate the molecular weight of any compound using the Periodic Table:

- From the Periodic Table display, press [α], enter the formula for the compound, then press [ENTER]. Use [←] to correct typing mistakes. The operation is described below.

- From outside the Periodic Table, use the MOLWT command. (See "Using Periodic Table Commands" later in this chapter.)

When you press [α] in the Periodic Table display, the application prompts for the molecular formula and automatically locks the alpha keyboard on. The application accepts any formula containing combinations of items listed below. (If you enter only an element symbol, the pointer moves to that element instead.)

- Element symbols (letters). Only the first letter of a two- or three-letter symbol is uppercase (use [↰] to enter lowercase letters) — NiN and NIN are different compounds. Letters that don't form valid symbols aren't accepted.
- Subscripts (numbers). Indicate multiples of the preceding element or group. Decimal numbers are allowed.
- Groups (parentheses). Join elements, subscripts, and groups. Use [↰][( )] for (. Use [↱][( )] for), or unlock alpha and press [▶].

These are examples of valid molecular formulas:

Input:	**Meaning:**
NaF	NaF
H2SO4	H_2SO_4
Mg(OH)2	$Mg(OH)_2$
Zn3(Fe(CN)6)2	$Zn_3[Fe(CN)_6]_2$

Press [ENTER] to calculate the molecular weight. Then

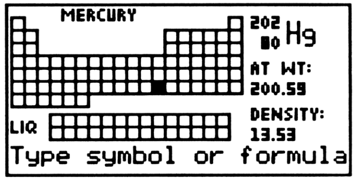
- To put the molecular weight and formula on the stack (and return to the Periodic Table display), press [ENTER].
- To just return to the Periodic Table display, press [ATTN].

The molecular weight contains SI units if you're using units — otherwise, no units are included. You can change this option in the property catalog for any element — see "Choosing Unit Options" earlier in this chapter.

Example: Calculating Molecular Weights. While studying eating disorders, Eaton Greenes wants to find the molecular weights of vitamin B_1 (thiamine, $C_{12}H_{17}ClN_4OS$) and TNT (trinitrotoluene, $CH_3C_6H_2(NO_2)_3$).

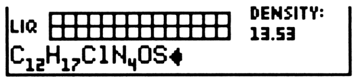
Clear the stack and start the Periodic Table. Then prepare to enter the molecular formula.

[→][CLR]
[←][LIBRARY]
PRTBL PERTB
[α]



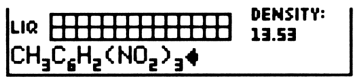
Enter the formula for vitamin B_1. Note that chlorine (Cl) near the middle has a lowercase "el."

C12H17ClN4OS



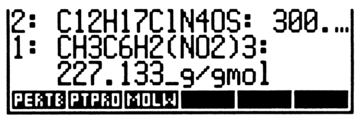
Calculate the molecular weight and put the formula and value on the stack. Next enter the formula for TNT.

[ENTER] [ENTER]
[α] CH3C6H2 [←][( )] NO2 [→][( )] 3



Calculate the molecular weight and put the formula and value on the stack. Return to the stack to see the two weights.

[ENTER] [ENTER]
QUIT



Separate the last result to get the formula and weight.

[PRG] OBJ OBJ→

Press [→][CLR] to clear the stack.

Using Periodic Table Commands

The Periodic Table includes three commands that you can execute from the menu, in the command line, and in programs. You can view the command names by pressing [←][LIBRARY] PRTBL [←][REVIEW].

Periodic Table Commands

Key	Programmable Command	Description
PERTB	PERTBL	Starts the Periodic Table. It doesn't affect the stack.
PTPRO	PTPROP	Returns the specified property for the specified element. From level 2 it takes the element's atomic number or symbol as a name (with certain restrictions). From level 1 it takes the property number. It returns to level 1 the property, usually a value or a string. It chooses to use or not use units according to the Units Usage flag (flag 61: SI units if clear, no units if set). See "Getting Properties with a Command" below.
MOLW	MOLWT	Returns the molecular weight for the specified molecular formula. From level 1 it takes the formula as a string (such as "H2O") or name (with certain restrictions, such as 'H2O'). It returns to level 1 the molecular weight. It chooses to use or not use units according to the Units Usage flag (flag 61: SI units if clear, no units if set). See "Calculating Molecular Weights with a Command" below.

You can use the PTPROP and MOLWT commands to get results without using the Periodic Table display. See the following topics.

If you use PTPROP or MOLWT with the HP Solve application or the Multiple-Equation Solver, you must consider the element name or formula name to be a placeholder variable (a dummy variable)—you can't solve for it. (If it appears in a Multiple-Equation Solver menu, you should make it a user-defined variable.)

Getting Properties with a Command

You can get element properties using the PTPROP command instead of using the Periodic Table display. Define the element using either its symbol (a name, such as 'Si') or its atomic number (such as 14). Define the property using the property number (such as 9 for the boiling point temperature). Property numbers are listed in the "Properties of Elements" table under "Finding Element Properties" earlier in this chapter.

For example, both of these program segments return the boiling-point temperature for silicon:

```
« ... 'Si' 9 PTPROP ... »
« ... 14 9 PTPROP ... »
```

If you use PTPROP as an algebraic function, you *must* use the symbol to define the element—you can't use its atomic number. For example, this program segment is valid:

```
« ... 'PTPROP(Si,9)' EVAL ... »
```

Calculating Molecular Weights with a Command

You can calculate molecular weights using the MOLWT command instead of using the Periodic Table display. With certain restrictions, you can define the molecular formula in either of two forms:

- A string, such as "H2O". Valid for any formula.
- A name, such as 'H2O'. Valid only for formulas without parentheses.

You can store a molecular formula in a variable, then use the variable name with MOLWT. You should do this when you want to use MOLWT in an expression and the formula contains parentheses or matches an HP 48 command name.

You must use care when naming a variable that contains a formula string or name. Make sure the variable name itself isn't a valid formula—for example, start the variable name with a lowercase letter. (If the variable name is a valid formula, using MOLWT with the variable name returns the molecular weight for the variable name, not for the formula it contains.)

For example, if you store the formula for ethyl alcohol (`"C2H6O"`) in variables *SUDS* and *HOPS*, `'MOLWT(SUDS)'` [EVAL] gives the molecular weight for ethyl alcohol, but `'MOLWT(HOPS)'` [EVAL] gives the weight for hydrogen-oxygen-phosphorus-sulfur. ("SUDS" isn't a valid molecular formula, but "HOPS" is.)

If your formula is the same as an HP 48 reserved word, such as SIN, you must supply it as a string.

Programming with the Periodic Table

You can access the Periodic Table from a program by using the Periodic Table commands (described under "Using Periodic Table Commands" earlier in this chapter):

- Execute PERTBL to start the Periodic Table. Then after you finish using the application, the program resumes execution.
- Execute PTPROP to get properties of elements. The current state of the Units Usage flag (flag 61) determines if SI units are used or not used.
- Execute MOLWT to calculate molecular weights. The current state of the Units Usage flag (flag 61) determines if SI units are used or not used.

Example: Scanning the Periodic Table. Experimental chemist Cynthia Sizer often needs to find elements that have a numeric property within a certain range. She can use the following program to get a tagged list of elements that have the specified property within the specified limits. To use the program, enter the property number (level 3) and two limits (levels 2 and 1) on the stack. The program returns a list of values tagged with element symbols to level 1. For example, 8, 90_°F, and −90_°F says to find elements with melting-point temperatures (property number 8) between 90 °F and −90 °F. If she omits units on the limit in level 1, SI units are implied. (The program takes about 2 minutes to run.)

Program:	**Comments:**
`« MAX LASTARG MIN DUP`	Finds upper and lower limits and creates unit sample.

`IF DUP TYPE`
`THEN 61 CF`
`ELSE 61 SF`
	Tests if unit sample has units. If units are entered, clears the Units Usage flag; otherwise, sets the flag.

`END → pnum hi lo unit`	Creates local variables for property number, limits, and unit sample.
`« 0 1 106` `FOR elem`	Initializes number of solutions and FOR element counter.

`elem pnum PTPROP`
`DUP TYPE → prop type`
`«`
	Uses PTPROP to get property value. Creates local variables for element property and object type.

`IF 'type==0 OR`
`type==13'`
`THEN`
	Tests if property is numeric (real or unit object). Processes only if numeric.

`IF 'prop≥lo AND`
`prop≤hi'`
`THEN prop`
	Tests if property is within limits, then includes value.

`IF 61 FC?` `THEN unit CONVERT`	If units are used, gets unit sample and converts property units.
`END elem 25 PTPROP`	Uses PTPROP to get element symbol (property 25).

```
      →TAG SWAP 1 +
     END
    END
   »
  NEXT
 » →LIST 61 CF
 1000 .1 BEEP
»
```

Uses element symbol as tag for value, then increments number of solutions. Leaves value and number of solutions on stack.

Converts solutions on stack into list of solutions, clears the Units Usage flag.

Checksum: # 2532d
Bytes: 400.0

(For information about the checksum and byte count, [←] [MEMORY] BYTES, see the "Calculator Memory" chapter in the *HP 48 Owner's Manual*.)

5

Constants Library

The Constants Library contains a collection of common physical constants and quantities. You can use them in equations and programs. (Several of these constants are used by the Equation Library.) The following table lists them in the order they appear in the Constants Library.

Constants Library

Name	Description
NA	Avogadro's number
k	Boltzmann constant
Vm	Molar volume
R	Universal gas constant
StdT	Standard temperature
StdP	Standard pressure
σ	Stefan-Boltzmann constant
c	Speed of light in vacuum
ε0	Permittivity of vacuum
μ0	Permeability of vacuum
g	Acceleration due to gravity
G	Gravitational constant
h	Planck's constant
hbar	Dirac's constant
q	Electronic charge
me	Electron rest mass

Constants Library (continued)

Name	Description
qme	q/me ratio (electron charge-to-mass)
mp	Proton rest mass
mpme	mp/me ratio (proton mass / electron mass)
α	Fine structure constant
ϕ	Magnetic flux quantum
F	Faraday constant
R∞	Rydberg constant
a0	Bohr radius
μB	Bohr magneton
μN	Nuclear magneton
λ0	Photon wavelength
f0	Photon frequency
λc	Compton wavelength
rad	1 radian
twoπ	2π radians
angl	180° angle *
c3	Constant from Wien's displacement law
kq	k/q (Boltzmann / electronic charge)
ε0q	ε0/q (permittivity / electronic charge)
qε0	q · ε0 (electronic charge × permittivity)
εsi	Dielectric constant of silicon
εox	Dielectric constant of silicon dioxide
I0	Reference intensity

* For no units: 180, 2π, or 200, depending on the angle mode.

Note that one name uses an accented character: ϕ. To type this character, press [α] O [α] [↱] [9]. You can't search for this character in the catalog, but it's right after the "α" entry ([α] [↱] A).

Getting Constants from the Catalog

To view the Constants Library catalog and retrieve constants, follow these steps:

1. Get the LIBRARY menu ([←] [LIBRARY]).
2. Get the Constants Library main menu by pressing COLIB.
3. Get the catalog of constants by pressing CONLI.

The constants catalog is a special environment in which you can view the descriptions and values of the constants, set the use of units, and retrieve values. You can perform the operations listed in the following table. (See "Viewing and Selecting in a Catalog" in chapter 1 for details.)

Operations in the Constants Catalog

Key	Action
[▲] [▼]	Moves the highlight up or down the catalog (also [←] and [→]).
[α]...	Jumps the highlight to that alpha line.
[ENTER]	Shows a wide (...) item completely — press [ENTER] or [ATTN] to return to the catalog.
SI ■	Indicates SI units are active.
ENG■	Indicates English units are active.
UNIT■	Indicates units are used.
UNITS	Indicates units are not used.
VALUE	Indicates descriptions are shown.
VALU■	Indicates values are shown using the current display format.

Operations in the Constants Catalog (continued)

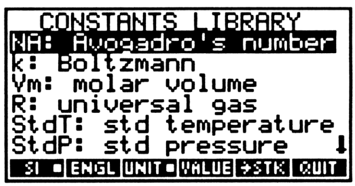
Key	Action
→STK	Puts the highlighted constant in level 1 of the stack using the current units selections.
QUIT	Exits from the catalog.
[ATTN]	Exits from the catalog.

The Constants Library provides two types of units: SI and English. In addition, you can choose to use or not use the units — this determines whether →STK returns a unit object or a real number. If you want to change how units are used with the constants, press the corresponding SI , ENGL ,or UNITS menu key. See "Working with and without Units" in chapter 1 for details.

Example: Selecting Dirac's Constant. Physicist Shirley Bright needs Dirac's constant for her energy calculation. Use the constants catalog to get the value.

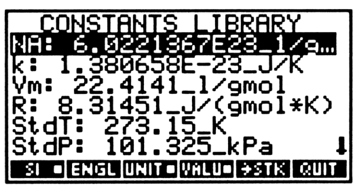
Clear the stack, then get the constants catalog.

[↱] [CLR]
[↰] [LIBRARY]
COLIB CONLI



Set SI units (SI ■) and units used (UNIT■), then view the values in the catalog.

(SI if needed)
(UNITS if needed)
VALUE



Find Dirac's constant, named "hbar" — remember to use [⬅] for lowercase. View its full value.

[α] h
[▼]
[ENTER]

```
 CONSTANTS LIBRARY

hbar: 1.05457266E-34_J
*s
```

Place the value on the stack, then exit the catalog.

[ENTER]
→STK QUIT

```
1: hbar:
   1.05457266E-34_J*s
CONLI CONS
```

Getting Constants with Commands

The Constants Library includes two commands. You can view the command names by pressing [⬅][LIBRARY] COLIB [⬅][REVIEW].

Constants Library Commands

Key	Programmable Command	Description
CONLI	CONLIB	Gets the Constants Library catalog. It doesn't affect the stack.
CONS	CONST	Returns the value of the specified constant. It takes the constant name (such as 'g') from level 1, and it returns the constant to level 1. It chooses the units type according to the Units Type flag (flag 60: SI if clear, English if set), and it chooses to use or not use units according to the Units Usage flag (flag 61: SI units if clear, no units if set).

Using CONST in Equations and Programs

You can use the CONST command in equations and programs to retrieve constants. For example, an equation for free-fall velocity using CONST might look like this:

```
'V=V0-CONST(g)*T'
```

`CONST(g)` returns the acceleration due to gravity using units as specified by the Units Type and Units Usage flags (flags 60 and 61).

In a program that performs the same operation, CONST takes the constant's name from the stack:

```
« V0 'g' CONST T * - V STO »
```

You can have variables with the same names as constants, but you should include ' marks around the constant names in the program. This ensures that CONST finds the constant name on the stack, rather than a variable value.

If you use CONST with the HP Solve application or the Multiple-Equation Solver, you must consider the constant name to be a placeholder variable (a dummy variable)—you can't solve for it. (If it appears in a Multiple-Equation Solver menu, you should make it a user-defined variable.)

Changing Constants for CONST

You may want to temporarily substitute a different value for a constant without having to alter your equation or program.

For the CONST command, you can override the library value of any constant by storing *your* value in a certain variable in the current directory (or higher directory). The variable name is "const" plus the constant name. (This new value does *not* change the value in the constants catalog.)

Purge the variable when you want to revert to the library value.

For example, if you want to use 0.00831451 kJ/gmol·K as the universal gas constant, store `.00831451_kJ/gmol*K` in variable *constR*. The CONST command returns this value for *R* regardless of the type of units you're using (whether the Units Type flag, flag 60, is clear or set) — units are dropped if the Units Usage flag (flag 61) is set.

6

Finance

The Finance application provides time-value-of-money (TVM) and amortization capabilities. You can use it for compound-interest and amortization calculations.

Compound interest occurs when earned interest is added to the principal at specified compounding periods, and then the combined amount earns interest. Many financial calculations are compound interest calculations—for example, savings accounts, mortgages, pension funds, leases, and annuities.

Amortization calculations determine the amounts applied toward principal and interest in a series of payments.

This chapter shows you how to use the Finance application:

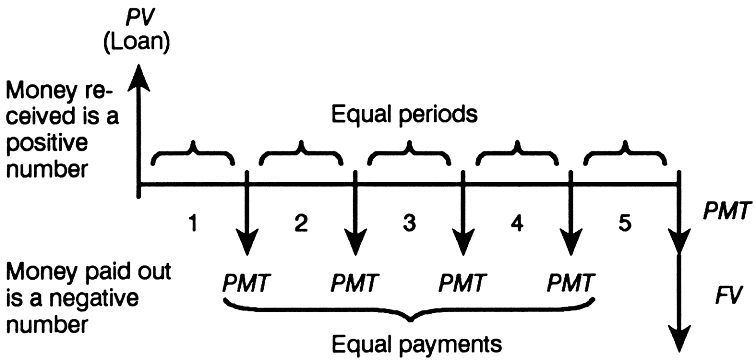
- Creating cash flow diagrams.
- Making TVM calculations.
- Making amortization calculations.
- Programming using Finance commands.

Making Cash Flow Diagrams

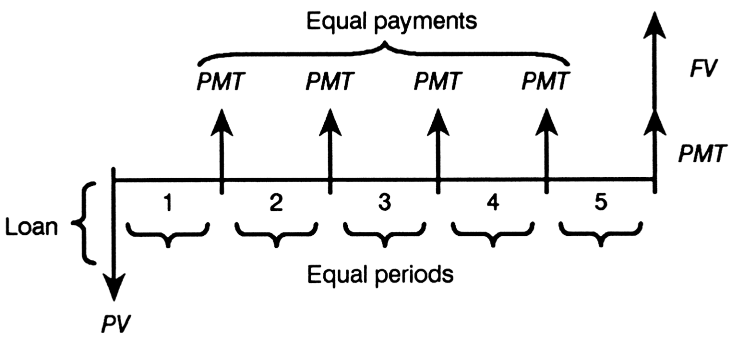
You can represent and understand many types of financial transactions using *cash flow diagrams*. A cash flow diagram is a time line divided into equal segments representing the compounding periods. Arrows represent the cash flows. Money received is a positive value, and money paid out is a negative value.

The cash flow diagram for a transaction depends on the point of view you take in your problem statement. For example, a loan is an initial positive cash flow for the borrower, but it's an initial negative cash flow for the lender.

The following cash flow diagram shows a loan from a *borrower's* point of view.

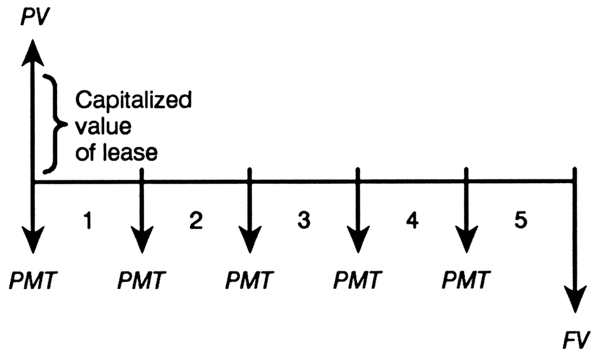


The following cash flow diagram shows a loan from a *lender's* point of view.

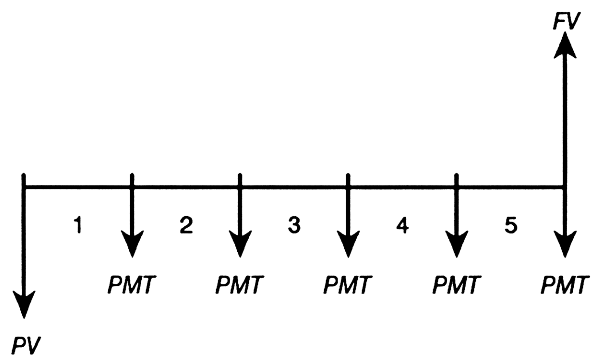


In addition, cash flow diagrams specify *when* payments occur relative to the compounding periods: at the *beginning* of each period or at the *end*. The Finance application provides both of these payment modes: Begin mode and End mode.

The following cash flow diagram shows lease payments at the *beginning* of each period.



The following cash flow diagram shows deposits into an account at the *end* of each period.



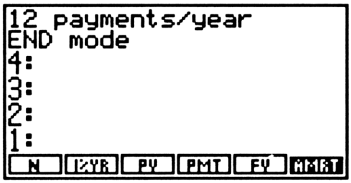
Making TVM Calculations

To make TVM calculations, follow this general procedure — details are given below:

1. Get the Finance menu ([⬅][LIBRARY] FIN).
2. Start the TVM application by pressing TVM .

3. Check and set these TVM conditions:
   - Number of payments per year.
   - Payments at beginning or end of periods.
4. Store values for the four known TVM variables.
5. Find the unknown value.

When you first start the TVM application, it displays a message showing the current number of payments per year and the payment mode (Begin or End). The application also displays the TVM menu.



The TVM menu operates similarly to the HP Solve menu. It provides the operations listed in the following table for the five TVM variables: *N*, *I%YR*, *PV*, *PMT*, and *FV*. (The variables are described below.)

TVM Menu Keys

Operation	Key	Action
Store value	[N]	Creates variable if necessary and stores value in variable.
Solve for value	[←] [N]	Creates variable if necessary and solves for value of variable.
Recall value	[→] [N]	Recalls value of variable to stack.

The TVM menu contains the five TVM variables plus several other functions. The [NXT] key shows additional menu labels, which control the TVM conditions.

Operations in the TVM Menu

Key	Action
N	Stores, solves, or recalls the total number of payments (compounding periods).
I%YR	Stores, solves, or recalls the percent nominal annual interest rate.
PV	Stores, solves, or recalls the present value — the initial cash flow or the value of a series of future cash flows. (To a lender or borrower, *PV* is the amount of the loan — to an investor, *PV* is the initial investment.) *PV* always occurs at the beginning of the first period.
PMT	Stores, solves, or recalls the amount of each periodic payment. The payments are the same amount, and no payments are skipped. Payments can occur at the beginning or end of each compounding period.
FV	Stores, solves, or recalls the future value — the amount of the final cash flow, or the compounded value of the series of previous cash flows. *FV* always occurs at the end of the last period.
AMRT	Calculates amortization (remaining balance, interest, and principal) based on four TVM variables. (See "Calculating Amortization" later in this chapter.)
PYR	Stores the number of payments (compounding periods) per year. The value must be an integer. ⮫ PYR recalls the value.

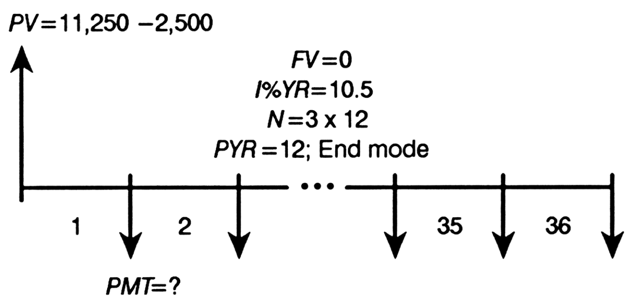
Operations in the TVM Menu (continued)

Key	Action
BEG	Sets Begin mode, used when payments occur at the beginning of each compounding period. (Sets the Payment Mode flag, flag 62.)
END	Sets End mode, used when payments occur at the end of each compounding period. (Clears the Payment Mode flag, flag 62.)
⬅ [REVIEW]	Displays the current contents of the TVM variables in the menu.

If you want to change the number of payments per year or the payment mode (Begin or End mode), use the menu keys on the second page of the menu.

Because TVM calculations use *all* TVM variables, you must be sure you have four correct values stored—even if the problem statement doesn't refer to a variable. For example, to calculate a payment for a fully amortized loan, you must make sure *FV* is 0. If a transaction has no payments, make sure *PMT* is 0.

Example: A Car Loan. Otto Tailfin is financing the purchase of a car with a 3-year loan at 10.5% annual interest, compounded monthly. The purchase price of the car is $11,250, and his down payment is $2500. What are his monthly payments? (Assume that payments start at the end of the first period.)



Change to the *HOME* directory and clear the stack. Get the TVM menu, then set the number of payments per year and the payment mode.

[→] [HOME] [→] [CLR]
[←] [MODES] 2 FIX
[←] [LIBRARY] FIN TVM
[NXT] 12 PYR END

```
12 payments/year
END mode
4:
3:
2:
1:
PYR  BEG  END■
```

Store the known TVM variables. Make sure you set the final value to 0, because the loan is fully paid after 3 years (3 × 12 payments). Review the known values — the unknown *PMT* variable may have any value.

[NXT]
3 [ENTER] 12 [×] N
10.5 I%YR
11250 [ENTER] 2500 [−] PV
0 FV
[←] [REVIEW]

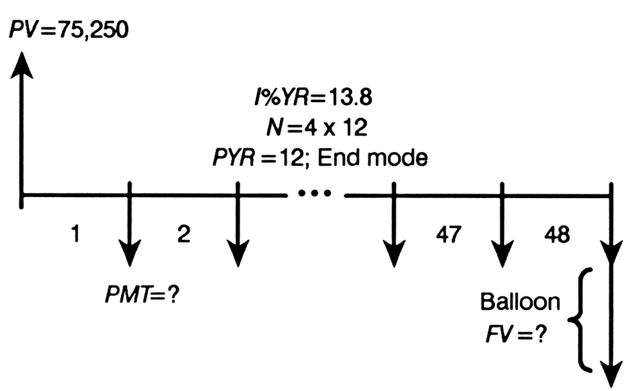
```
12 payments/year
END mode
N: 36.00
I%YR: 10.50
PV: 8,750.00
PMT: undefined
FV: 0.00
N  I%YR  PV  PMT  FV  AMRT
```

Solve for the payment.

[←] PMT

```
1:            PMT: -284.40
N  I%YR  PV  PMT  FV  AMRT
```

Example: A Mortgage with a Balloon Payment. Russ T. Pipes has taken out a 25-year, $75,250 house mortgage at 13.8% annual interest. He expects to sell the house in 4 years, repaying the loan in a balloon payment. Find the size of the balloon payment — the value of the mortgage after 4 years of payments.



Store the number of payments per year and set the payment mode.

[→] [CLR]
[←] [MODES] 2 FIX
[←] [LIBRARY] FIN TVM
[NXT] 12 PYR END

```
12 payments/year
END mode
4:
3:
2:
1:
PYR BEG END■
```

Store the known TVM variables and calculate the monthly payment for the 25-year mortgage.

[NXT] 25 [ENTER] 12 [×] N
13.8 I%YR 75250 PV
0 FV
[←] PMT

```
1:          PMT: -894.33
N I%YR PV PMT FV AMRT
```

To begin finding the balloon payment, store the actual dollars-and-cents payment, which is the computed payment rounded to two decimal places. (Otherwise, *PMT* would still have fractional cents.)

894.33 [+/–] PMT

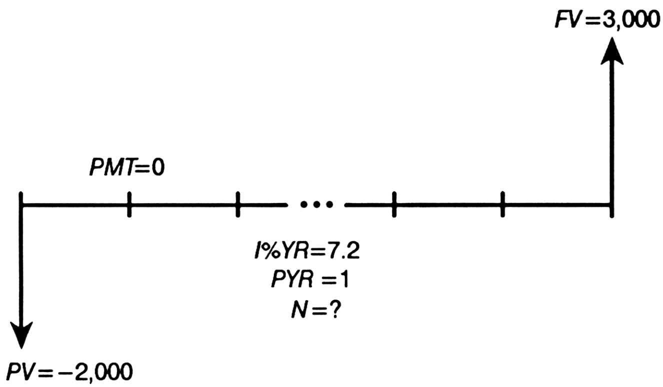
```
PMT: -894.33
4:
```

Store the number of payments made in 4 years, then calculate the final value — the balloon payment.

4 [ENTER] 12 [×] N
[←] FV

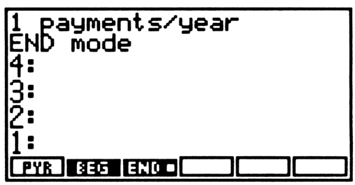
```
2:          PMT: -894.33
1:         FV: -73408.81
N I%YR PV PMT FV AMRT
```

Example: A Savings Account. Penny Horder deposits $2000 into a savings account that pays 7.2% annual interest, compounded annually. If she makes no other deposits into the account, how long does it take for the account to contain $3000?



Set the number of compounding periods per year and the payment mode.

[↱] [CLR]
[↰] [MODES] 2 [FIX]
[↰] [LIBRARY] [FIN] [TVM]
[NXT] 1 [PYR] [END]



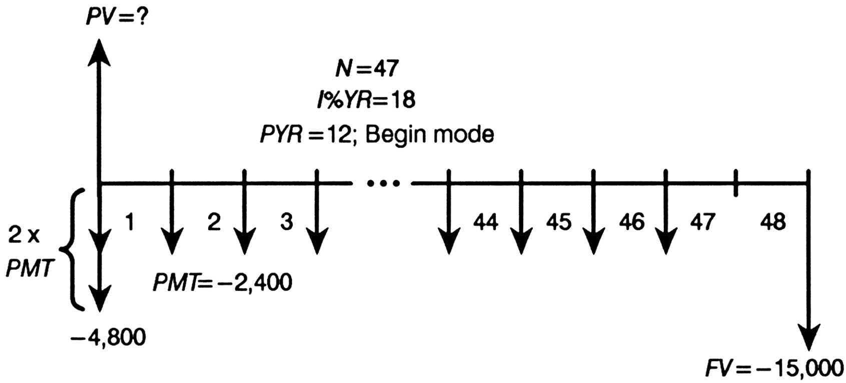
Store the known TVM variables and calculate the number of payments.

[NXT] 7.2 [I%YR]
2000 [+/-] [PV]
0 [PMT] 3000 [FV]
[↰] [N]

It takes 6 years (more than five compounding periods) to achieve a balance of at least $3000. Find the actual balance at the end of 6 years.

6 [N]
[↰] [FV]

Example: A Lease. Sandy Lome is leasing farm equipment for 4 years. The monthly payment is $2400. An additional $2400 payment at the beginning of the leasing period replaces the final payment. The leasing agreement includes an option to buy the equipment for $15,000 at the end of the leasing period. Calculate the capitalized value of the lease, assuming that the interest rate Lome pays to borrow funds is 18%, compounded monthly.



Make the calculation in four steps:

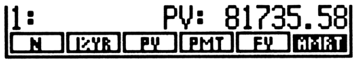
1. Calculate the present value for the 47 monthly payments — the initial investment required to make the monthly payments.
2. Include the advance payment.
3. Calculate the present value of the buy option — the initial investment required to generate the option price after 48 months.
4. Add the values calculated in steps 1, 2, and 3.

Store the number of payments per year and set Begin mode.

[→][CLR]
[←][MODES] 2 FIX
[←][LIBRARY] FIN TVM
[NXT] 12 PYR BEG

Step 1: Find the present value for the monthly payments.

[NXT] 47 [N] 18 [I%YR]
2400 [+/-] [PMT] 0 [FV]
[⬅] [PV]



Step 2: Add the additional advance payment to the present value.

2400 [+]

Step 3: Calculate the present value of the buy option.

48 [N] 0 [PMT]
15000 [+/-] [FV]
[⬅] [PV]

Step 4: Add the value of the buy option to the earlier amount.

[+]

Calculating Amortization

Amortization calculations determine the amounts applied toward principal and interest in a payment or series of payments. The calculations use the TVM menu—see "Making TVM Calculations" earlier in this chapter to find out how to use the TVM menu.

To make amortization calculations, follow this general procedure:

1. Get the Finance menu ([⬅] [LIBRARY] [FIN]).
2. Start the TVM application by pressing [TVM].
3. Check and set these TVM conditions:
   - Number of payments per year.
   - Payments at beginning or end of periods.
4. Store values for four TVM variables: *I%YR*, *PV*, *PMT*, and *FV*. These variables define the payment schedule. (You can calculate these using the TVM menu.)
5. Change the display mode to the accuracy you want, such as 2 Fix mode.
6. Enter the number of payments to amortize in level 1 on the stack.

7. Find the amortization amounts by pressing AMRT :
   - Level 3: Amount of the payments applied toward principal.
   - Level 2: Amount of the payments applied toward interest.
   - Level 1: Balance of the loan after the payments are made.

To continue amortizing the loan, store the new balance in *PV*, enter the new number of payments to amortize, and press AMRT . Repeat for each successive amortization period.

To amortize a series of future payments starting at payment *p*, perform the following steps:

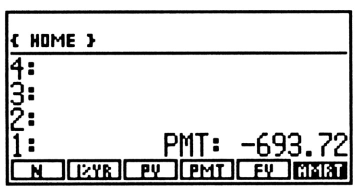
1. Calculate the balance of the loan at payment $p-1$. (To do this, put $p-1$ on the stack and amortize using the initial *PV*.)
2. Store the new balance in *PV*.
3. Amortize the series of payments starting at the new *PV*.

The amortization operation reads the values from the TVM variables, rounds the numbers it gets from *PV* and *PMT* to the current display mode, then calculates the amortization rounded to the same setting. The original variables aren't changed.

Example: Amortization Calculations. Rufus Leekin has taken out a 30-year, $65,000 mortgage at 12.5% annual interest. Calculate the amount of the first year's and second year's payments that are applied toward principal and interest.

The payment is not given, so you must calculate it.

[→] [CLR]
[←] [MODES] 2 FIX
[←] [LIBRARY] FIN TVM
[NXT] 12 PYR END
[NXT] 30 [ENTER] 12 [×] N
12.5 I%YR 65000 PV
0 FV
[←] PMT



Find the amortization for the first year (12 payments).

12 AMRT

Store the balance after 12 payments into *PV*, then compute the amortization for the second year.

PV
12 AMRT

```
3:   Principal: -239.49
2:   Interest: -8085.15
1:    Balance: 64549.03
 N  I%YR  PV  PMT  FV  AMRT
```

Press [←][MODES] STD to return to Standard display mode.

Programming with Finance Commands

The Finance application includes several commands for making financial calculations. You can use these commands to solve financial problems without using the TVM menu. You can review the command names by pressing [←][LIBRARY] FIN [←][REVIEW].

Finance Commands

Key	Programmable Command	Description
TVM	TVM	Displays the TVM menu. It doesn't affect the stack.
TVMB	TVMBEG	Sets Begin payment mode. (Sets the Payment Mode flag, flag 62.) It doesn't affect the stack.
TVME	TVMEND	Sets End payment mode. (Clears the Payment Mode flag, flag 62.) It doesn't affect the stack.

Finance Commands (continued)

Key	Programmable Command	Description
TVMR	TVMROOT	Solves for the specified TVM variable using values from the remaining TVM variables. It uses the payment mode according to the Payment Mode flag (flag 62: End if clear, Begin if set). It takes the variable name from level 1, and it returns its value in level 1.
AMOR	AMORT	Performs amortization calculations using values from the TVM variables (except *N*). It uses the payment mode according to the Payment Mode flag (flag 62: End if clear, Begin if set). It takes the number of payments from level 1, and it returns the principal amount in level 3, the interest amount in level 2, and the remaining balance in level 1.

The procedure for using the Finance commands in a program is the same as for making calculations from the keyboard:

1. Use TVMBEG or TVMEND to set the payment mode.
2. Establish the values of the known TVM variables, including *PYR*.
3. Use TVMROOT to solve for the unknown variable, or use AMORT to find the amortization amounts.

If you use TVMROOT with the HP Solve application or the Multiple-Equation Solver, you must consider the TVM variable name to be a placeholder variable (a dummy variable)—you can't solve for it. (If it appears in a Multiple-Equation Solver menu, you should make it a user-defined variable.)

Example: Programming with TVMROOT and AMORT. Buck Lender wants a quick way to calculate auto-loan information. He can use the following program — it calculates the monthly payments and the total interest paid for a loan given the number of years of the loan, the interest rate, and the amount to borrow. To use the program, enter the number of years (level 3), the annual interest rate (level 2), and the amount of the loan (level 1) on the stack. The program returns the monthly payment (level 2) and the total loan interest (level 1).

Program:	**Comments:**
«	
'PV' STO 'I%YR' STO 12 * 'N' STO 0 'FV' STO	Stores the amount to borrow, the interest rate, and the number of periods. Sets the future value to 0.
TVMEND 2 FIX	Sets End mode and 2 Fix mode.
'PMT' TVMROOT DUP 'PMT' STO	Solves for the payment, stores the value, and leaves its rounded value on the stack.
N AMORT	Amortizes the full loan.
ROT DROP2	Drops the balance and principal.
»	

Checksum: # 32686d
Bytes: 137.5

(For information about the checksum and byte count, [⬅][MEMORY] BYTES, see the "Calculator Memory" chapter in the *HP 48 Owner's Manual*.)

Press [⬅][MODES] STD to return to Standard display mode.

7

Multiple-Equation Solver

The Multiple-Equation Solver is an application that enables you to solve a set of two or more equations for unknown variable values. It does this by selecting equations one at a time and finding its roots. For example, the Equation Library uses the Multiple-Equation Solver to find solutions, as described in chapter 2, "Equation Library."

This chapter shows you how to do the following:

- Define a set of equations.
- Solve equations from the keyboard.
- Solve equations using commands, such as in a program.

When you solve problems using the Multiple-Equation Solver, the application uses the same numeric root finder that's used by the HP Solve application, which is built into the HP 48. This manual assumes you're familiar with the operation of the HP Solve application. If necessary, read the "HP Solve Application" chapter in the *HP 48 Owner's Manual.*

Solving a Set of Equations

Many computational problems involve determining numeric values for one or more unknown variables. Often these problems are solved using one or more equations that relate the unknown variables to known variables — but the task of solving for the unknown values is sometimes difficult.

The Multiple-Equation Solver provides a convenient tool for solving this type of problem. Follow this general procedure to solving a problem using the Multiple-Equation Solver:

1. Create a set of equations that defines the relationships among the variables in your problem, and initialize the Multiple-Equation Solver to that set.
2. Use the Multiple-Equation Solver to enter the values of known variables and to solve for unknown variables.

The following sections describe the general steps.

Defining a Set of Equations

The Multiple-Equation Solver uses the list of equations stored in *EQ*. Actually, "equations" in this context includes programs, expressions, and variable names that evaluate to a single value. The Multiple-Equation Solver requires that *EQ* contain more than one equation — that is, the HP Solve application would include the NXEQ menu label for *EQ*. The solver uses *EQ* to create a reserved variable *Mpar* that's used during the solution process. *Mpar* contains the equation set plus additional information. (Because *Mpar* is a variable, you can have a different multiple-equation set for each directory in memory.)

Designing the Equations

When you design a set of equations, you should do it with an understanding of how the Multiple-Equation Solver uses the equations to solve problems.

The Multiple-Equation Solver uses the same process you'd use to solve for an unknown variable — assuming you weren't allowed to create additional equations. You'd look through the set of equations for one that has only one variable that you don't know. You'd use the HP 48 root finder to find its value. Then you'd do this again until you've found the variable you want.

So you should choose your equations to allow likely unknown variables to occur individually in equations. You must avoid having two or more unknown variables in all equations. You can also specify the equations in an order that's best for your problems.

For example, the following three equations define initial velocity and acceleration based on two observed distances and times. The first two equations alone are mathematically sufficient for solving the problem, but each equation contains two unknown variables. Adding the third equation allows a successful solution because it contains only one of the unknown variables.

$$x_1 = v_0 + a \cdot t_1$$
$$x_2 = v_0 + a \cdot t_2$$
$$(x_2 - x_1) = a \cdot (t_2 - t_1)$$

To create more robust equations, you can include functions that ensure proper and faster calculations — for example, CONST and TDELTA from the application card and UBASE, EXP, and IFTE from the HP 48. See appendix B, "Equation Library Notes," for examples.

If your equations use the IFTE function, the Multiple-Equation Solver considers the states of *all* variables when deciding how many unknown variables it contains — even though it may need to use only one of the conditional expressions. For example, the expression `'a=IFTE(b<10,c,d)'` contains four variables, though only three may be needed for a solution — either *c* or *d* may not be needed.

If your equations use any of the following functions, their variables won't necessarily be detected by the Multiple-Equation Solver: Σ, ∫, ∂, |, QUOTE, APPLY, PTPROP, TVMROOT, MOLWT, and CONST.

Creating the Equations

You can use one of these techniques to create a set of equations:

- Create or recall equations on the stack, use →LIST ([PRG] `OBJ` `→LIST`) to combine them into a list, then store the list in *EQ*. Execute MINIT to create *Mpar*.
- Create equations and save them in variables (you can use `NEW` in the HP Solve menu), then use `EQ+` in the HP Solve equation catalog to combine the equations. Execute MINIT to create *Mpar*. (See the detailed steps below.)

To combine several equations using the HP Solve menu, follow these steps:

1. Press [←] [SOLVE] to get the HP Solve menu.
2. If the equations don't already exist, enter an equation, then press `NEW` to store it in a variable. Repeat this for each equation.
3. Press `CAT` to get the catalog of equations.
4. Move the pointer to an equation you want to include in *EQ*, then press `EQ+`. This creates or updates a list of selected equations and displays the list in the status area. (Use [←] `EQ+` to remove the last entry from the list.) Repeat this step for each equation you want to link.
5. Press `SOLVR` to store the list in *EQ*. (This also starts the HP Solve application.)
6. Execute MINIT ([←] [LIBRARY] `MES` `MINIT`) to create *Mpar* from *EQ*.

The list of equations in *EQ* may contain menu definitions, but those definitions are ignored by MINIT when it creates *Mpar*. However, you can reorder the menu labels using MITM, described under "Changing the Title and Menu" later in this chapter.

Example: Creating a Set of Equations. During his investigation of conical buildings, architect T. P. Bilder wants to find certain geometric parameters. He needs a set of equations that relate the radius, height, slant height, and volume of a conical structure. Create the two equations `'L=√(R^2+H^2)'` and `'V=π*R^2*H/3'`. Use the HP Solve application to name them *LCONE* and *VCONE* and to store them in the equation catalog. Then start the Multiple-Equation Solver.

Clear the stack, then key in and store the first equation.

[→] [HOME] [→] [CLR]
['] L [←] [=] [√x] [←] [( )]
R [y^x] 2 [+] H [y^x] 2
[ENTER]
[←] [SOLVE] `NEW` LCONE [ENTER]

```
Current equation:
LCONE: 'L=√(R^2+H^2)'
4:
3:
2:
1:
SOLVR ROOT NEW EDEQ STEQ CAT
```

Key in and store the second equation.

['] V [⬅][=] [⬅][π] [×]
R [y^x] 2 [×] H [÷] 3
[ENTER]
NEW VCONE [ENTER]

```
Current equation:
VCONE: 'V=π*R^2*H/3'
4:
```

In the Equation catalog, move the cursor to equation *VCONE* and put the equation in a list.

CAT
([▲] or [▼] as needed)
EQ+

```
{ VCONE }
▶VCONE: 'V=π*R^2*H/3'
 EQ: 'VCONE'
 LCONE: 'L=√(R^2+H^2)'
 ELIB: dir
PLOTR SOLVR EQ+ EDIT →STK VIEW
```

Move the cursor to *LCONE* and add it to the list.

([▼] as needed)
EQ+

```
{ VCONE LCONE }
 VCONE: 'V=π*R^2*H/3'
 EQ: 'VCONE'
▶LCONE: 'L=√(R^2+H^2)'
```

Execute SOLVR to store the list of equations in *EQ*.

SOLVR

```
VCONE: 'V=π*R^2*H/3' '
4:
3:
2:
1:
V R H EXPR= NXEQ
```

Execute MINIT to establish *Mpar*.

[⬅][LIBRARY] MES MINIT

```
1:
MSOL MINIT MITM MUSE MCAL MROO
```

This example continues later in this chapter.

Solving the Equations

Much of the information about using the Multiple-Equation Solver is covered in chapter 2, "Equation Library." The Multiple-Equation Solver works with your set of equations in the same way it works with a set from the Equation Library.

Press MSOL in the Multiple-Equation Solver main menu ([←][LIBRARY] MES MSOL) to start the Multiple-Equation Solver — to get the variable menu labels. When you press MSOL , the application does the following:

- Menu labels, variable values and status, equation title, and progress information are read from *Mpar* in the current (or higher) directory. (See "Defining a Set of Equations" earlier in this chapter.)
- The title and solver menu are displayed.

MSOL (MSOLVR) is analogous to [←][SOLVE] SOLVR — or [→][SOLVE] — in the HP Solve application.

Using the Multiple-Equation Solver Menu

The following table summarizes the actions for the solver menu keys. The [NXT] key shows additional menu labels.

Solver Menu Keys

Operation	Key	Action
Store value	X X	Creates variable if necessary, stores value in variable, and makes variable user-defined. If the value has no units, the units of the previous value are appended, if any.
Solve for value	[←] X [←] X	Creates variable if necessary, solves for value of variable, and makes variable not user-defined.
Recall value	[→] X [→] X	Recalls value of variable to stack.
Undefine all	ALL	Makes all variables not user-defined.
Solve for all	[←] ALL	Creates variables if necessary and solves for all variables not user-defined (or as many as possible).
Progress catalog	[→] ALL	Shows information about last solution.
User-defined	MUSE	Sets states to user-defined for variable or list of variables on the stack.
Calculated	MCAL	Sets state to *not* user-defined (calculated result) for variable or list of variables on the stack.

The menu labels for the variable keys are white at first — they change during the solution process as described below.

Because a solution involves many equations and many variables, the Multiple-Equation Solver must keep track of variables that are user-defined and not defined — those it can't change and those it can. In addition, it keeps track of variables that it used or found during the last solution process.

The menu labels indicate the states of the variables. They're automatically adjusted as you store values and solve for variables. You can check that variables have proper states when you supply guesses and find solutions.

Notice that ■ marks the variables that were used in the last solution — their values are compatible with each other. Other variables may *not* have compatible values because they weren't involved in the solution.

Meanings of Menu Labels

Label	Meaning
X0	Value *x0* not defined by you and not used in the last solution — it can change in the next solution.
X0 ■	Value *x0* not defined by you, but found in the last solution — it can change in the next solution.
X0	Value *x0* defined by you, but not used in the last solution — it can't change in the next solution (unless you solve for only this variable).
X0 ■	Value *x0* defined by you and used in the last solution — it can't change in the next solution (unless you solve for only this variable).

See "Controlling Variables" in chapter 2 for more information about using the solver menu.

Returning to the Solver

If you change to other menus while using the solver, you can resume the solving process where you left off—execute MSOLVR ([⬅] [LIBRARY] MES MSOL).

To interrupt the solution process, press [ATTN].

The following three topics give general information about the three steps for solving equations.

Step 1: Entering Known Values

To enter a known value for a variable, type the value and press the menu key for the variable. For example, press 12.34 X0 to enter 12.34 units as the value of *x0*.

The Multiple-Equation Solver interprets your input according to the units (or no units) currently existing in the variables. See "Step 1: Entering Known Values" in chapter 2 for details.

Step 2: Supplying Guesses (Optional)

You can supply deliberate guesses for a variable you want to solve for. This can speed the solution process or focus on one of several possible solutions, such as for equations involving trigonometric or polynomial expressions. You can minimize the chance of finding undesirable values by supplying guesses for variables with more than one possible solution. See "Step 2: Supplying Guesses" in chapter 2 for details. An example is included under "Step 3: Finding the Solution" in chapter 2.

Step 3: Finding the Solution

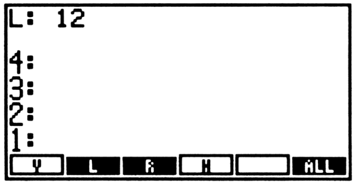
To solve for a variable, press [⬅] followed by the menu key for the variable. Or you can solve for all remaining variables that you didn't define as "known"—press [⬅] ALL .

For example, press [⬅] X0 to solve for *x0*—or press [⬅] ALL to solve for *x0* and other unknown variables.

Example: Solving Equations. This example continues from above, checking conical buildings for T. P. Bilder. Find the volume and height of a right circular cone whose radius is 5 meters and whose slant height is 12 meters.

Start the Multiple-Equation Solver. In case the variables contain unit objects from other problems, purge the variables. Then enter the known variables. (Notice that the menu labels change to black.)

MSOL
[←] [{ }] V L
R H [ENTER]
[←] [PURGE]
5 R 12 L

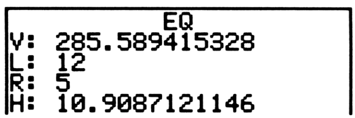


Now solve for the volume and height.

[←] ALL

The status messages that appeared and the marked menu labels show that both *H* and *V* have been found, so all variables now represent their components of the same cone. Look at their values.

[←] [REVIEW]



This example continues later in this chapter.

Interpreting Results

See "Interpreting Results" in chapter 2 for information about the following:

- Interpreting error messages that occur during the solution.
- Understanding seemingly improper solutions.
- Checking the progress catalog, which describes the last solution process. (An example is included.)

You can also plot the relationship between two variables as a check of your solution — see "Plotting Equations" later in this chapter.

Altering the Problem

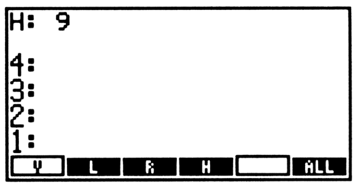
You may occasionally want to re-solve a problem using different known conditions. To do this, you must ensure that all "known" variables have black menu labels and all "unknown" variables have white menu labels — *except*, if you intend to solve for only one variable, its state doesn't matter.

Use MUSE and MCAL to change the states as required — see "Controlling Variables" in chapter 2 for details.

Example: Altering the Problem. This example continues from above, checking conical buildings for T. P. Bilder. Find the radius and volume of a cone with a height of 9 meters and the same slant height as in the previous problem.

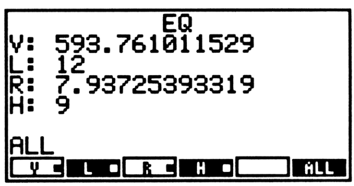
Change the height to 9 meters. (Notice when you change a value, the marks disappear from the labels.)

9 [H ■]



Find the radius and volume. To do this, first change variable *R* so its value is *not* user-defined (it's "calculated") — so the solver will *find* its value, not *know* its value.

['] R
[NXT] MCAL
[NXT] [⬑] ALL
[⬑] [REVIEW]



This example continues later in this chapter.

Solving a Sequence of Problems

Often you can separate a complex problem into a sequence of simpler problems, each with a set of equations. If you use the same variable names in each set for shared parameters, the solution for one problem becomes available as input for the next.

Setting Up Default Units

You can set up default units for the variables in your equations. This can simplify your solutions in these ways:

- Your variables can use *any* set of consistent units — conversion factors are automatically applied as needed in finding the solution.
- You can enter variable values without including units — units are automatically appended. (See "Step 1: Entering Known Values" in chapter 2 for details.)

One way to set up default units for a set of equations is to create a program that uses STO to store a unit object in each variable. The values aren't critical, but zero is probably the safest value because it's used as a guess when you solve for a variable.

For more information about using units, see the next topic, "Working with and without Units."

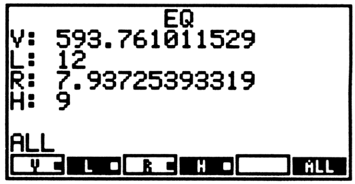
Example: Setting Up Default Units. This example continues from above, checking conical buildings for T. P. Bilder. Create a program that sets up default units of ft for linear dimensions, and yd^3 for volume.

Enter the following program on the stack:

```
« 0_ft DUP DUP 'R' STO 'H' STO 'L' STO
        0_yd^3 'V' STO »
```

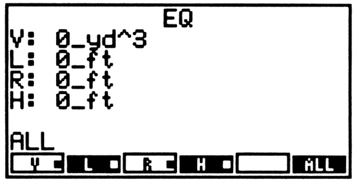
Store the setup program in *SETUP*. Review the current variable values.

['] SETUP [STO]
[←][LIBRARY] MES MSOL
[←][REVIEW]



Run the program, then review the variable values. These new units will be appended to your numeric entries.

[VAR] SETUP
[→][LAST MENU]
[←][REVIEW]



This example continues later in this chapter.

Working with and without Units

You can store variable values with or without units when you're solving equations. If variables have dimensionally consistent units, conversion factors are automatically applied in finding the solution. *But this happens only if units are used.* If you choose not to use units, no such conversions are performed—*you* must ensure that all variables use a compatible set of implied units. If you use units, the results are more easily interpreted. If you don't use units, the solver works faster.

Your equations can use information from the Equation Library, Periodic Table, or Constants Library. But you must set the unit options for those applications to match the usage in your equations. See "Working with and without Units" in chapter 1 for details.

For example, suppose your equations use no units, but SI units are implied, and the equations use the speed of light from the Constants Library, CONST(c). Then you must set up the proper conditions for the Constants Library. For SI units, the Units Type flag (flag 60) must be clear. For units not used, the Units Usage flag (flag 61) must be set.

If you want to delete units from existing variables, use PURGE to purge the variables. For example, you can press [⬑][{ }], several variable menu keys, [ENTER], then [⬑][PURGE].

If a variable with units exists in a higher directory, use [STO] to store any value — it creates a new variable without units in the current directory.

Example: Deleting Units from Variables. This example continues from above, checking conical buildings for T. P. Bilder. In the previous example, units were added to all equation variables. Now revert to the original condition — no units used with the variables.

Create a list containing the variables, then purge them. Note that the menu labels don't change.

```
[⬑][{ }]
[ V ▪ ] [ L ▪ ] [ R ▪ ] [ H ▪ ]
[ENTER] [⬑][PURGE]
[⬑][REVIEW]
```

```
                 EQ
V: Undefined
L: Undefined
R: Undefined
H: Undefined

ALL
 V ▪ | L ▪ | R ▪ | H ▪ |     | ALL
```

Make all variables not defined. Then store a value to check that units aren't used.

```
[ALL]
6 [ H ]
[⮫] [ H ]
```

```
1:                              6
 V | L | R | H |     | ALL
```

Press [⮫][CLR] to clear the stack. This example continues later in this chapter.

Changing the Title and Menu

The default title for a set of equations is "EQ." Menu labels for variables are automatically assigned in the order the variables occur in the equations.

You can change the title and variable menu order using the MITM command. However, you must do this *after* you've created the *Mpar* variable using MINIT.

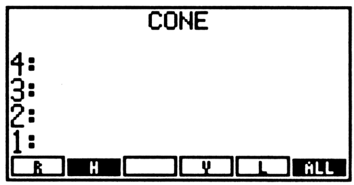
To change the title and menu, follow these steps:

1. Use MINIT to create *Mpar*. (See "Creating the Equations" earlier in this chapter.)
2. Put on the stack a string containing the new title (level 2) and a list containing the variable names in the order you want (level 1). Use " " to indicate a blank label. You must include *all* variables in the original menu and no others, and you must match uppercase and lowercase names properly.
3. Execute MITM ([←][LIBRARY] MES MITM) to change the title and menu in *Mpar*.

Example: Changing the Title and Menu. This example continues from above, checking conical buildings for T. P. Bilder. Change the title and menu for this set of equations. (You already used MINIT to create *Mpar*.)

Change the title to "CONE" and the order of variables to "R, H, blank, V, L."

[→][" "] CONE [ENTER]
[←][{ }] R H
[→][" "] [▶] V L [ENTER]
[←][LIBRARY] MES MITM
MSOL



This example continues later in this chapter.

Plotting Equations

After you use the solver to enter values for known variables, you can plot the relationship between two variables in your set of equations. The HP 48 Plot application uses the equation in *EQ* to plot the graph:

- If your set contains more than one equation, you can replace *EQ* with a short program that relates the two variables—see below. (If your set contains only one equation, *EQ* is ready to start plotting.)
- If your set contains more than one equation, but one of the equations expresses the relationship between the two variables, you could extract that equation from the list in *EQ*, then store it as *EQ*—or you could use the method described next.

For a set of several equations, you should replace *EQ*—this doesn't affect the Multiple-Equation Solver because it uses *Mpar*. To set up *EQ* for a set of several equations, follow these steps:

1. Make the independent variable user-defined. To do this you could store any value in the variable using the Multiple-Equation Solver—or you could use MUSER.
2. Create a program that solves for the dependent variable using MROOT:

   « '*dependent-variable*' MROOT »

3. Store the program in *EQ*.

When *EQ* is ready, use the commands in the PLOTR menu ([→] [PLOT]) to draw the plot. It's best *not* to use autoscaling. *Note that the plotting can take a long time* because the Multiple-Equation Solver is invoked for each point—you can use a larger plot resolution to reduce the number of points. For information about plotting, see the "Basic Plotting and Function Analysis" and "More About Plotting and Graphics Objects" chapters in the *HP 48 Owner's Manual*. (In particular, the operations in the GRAPHICS FCN menu don't work with program plots.)

Example: Plotting Equation Variables. This example continues from above, checking conical buildings for T. P. Bilder. For a slant height of 5 meters, plot the volume as a function of radius. (The example assumes the Curve Filling flag, flag −31, is clear.)

Make all variables not user-defined. Enter the given value of *L*. Make the independent variable *R* be user-defined.

ALL
5 [L]
0 [R]

```
R: 0
4:
3:
2:
1:
 R  H     V  L  ALL
```

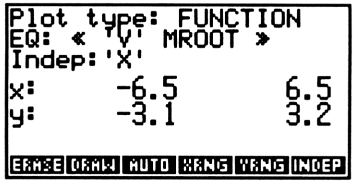
Create a program to solve for *V*.

[←] [« »] ['] [V] [►]
[←] [LIBRARY] MES MROO [ENTER]

```
1:          « 'V' MROOT »
MSOL MINIT MITM MUSE MCAL MROO
```

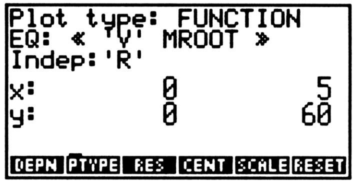
Start the Plot application, store the program in *EQ*, set the plot type to FUNCTION, and get the PLOTR menu.

[←][PLOT] STEQ
PTYPE FUNC
PLOTR



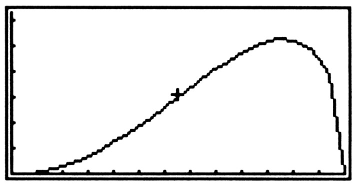
Set the independent variable to *R*, the x-range to 0 – 5, the y-range to 0 – 60, and the resolution to 0.25.

[VAR] ['] R
[→][LAST MENU] INDEP
0 [ENTER] 5 XRNG
0 [ENTER] 60 YRNG
[NXT] .25 RES



Plot the relationship. (This process takes several minutes.) Then clear the menu labels to see the plot of volume vs. radius.

[NXT] [NXT]
ERASE DRAW (wait...)
[–]



Press [ATTN] to return to the stack display. This example continues later in this chapter.

Solving Equations with Commands

The Multiple-Equation Solver includes six commands that you can execute from the menu, in the command line, and in programs. You can use these commands to solve problems without using the variable menu keys. You can view the command names by pressing [←][LIBRARY] MES [←][REVIEW].

Multiple-Equation Solver Commands

Key	Programmable Command	Description
MSOL	MSOLVR	Begins the solver using the existing *Mpar*. It doesn't affect the stack.
MINIT	MINIT	Creates a new *Mpar* from *EQ*. It doesn't affect the stack.
MITM	MITM	Changes the title and variable menu in *Mpar*. It takes a title string from level 2 and a list of variables from level 1, and it returns nothing.
MUSE	MUSER	Makes a variable be user-defined. It takes either a variable name, a list of variables, or "ALL" from level 1, and it returns nothing.
MCAL	MCALC	Makes a variable be not user-defined—a calculated value. It takes either a variable name, a list of variables, or "ALL" from level 1, and it returns nothing.
MROO	MROOT	Solves for one or more variables starting with only user-defined values, which can be set by MUSER and MCALC. It leaves found values in the variables and displays no status messages. It can take a variable name from level 1, and it returns the found value to level 1. Or it can take "ALL" from level 1, and it returns nothing to the stack.

Example: Solving with Commands. This example continues from above, checking conical buildings for T. P. Bilder. Use *commands* to set all variables so they're not user-defined, to set user-defined values for an 8-meter height and 3-meter radius, and to solve for the volume.

Use the VAR menu to store values (note that [◄] is required to store values) and the Multiple-Equation Solver main menu to execute commands.

[VAR] 8 [◄] H 3 [◄] R
[◄] [LIBRARY] MES
[►] [" "] ALL MCAL
[◄] [{ }] H [SPC] R [ENTER] MUSE
['] V MROO

```
1:           75.3982236858
MSOL MINIT MITM MUSE MCAL MROO
```

8

Utilities

The Utilities application consists of the Minehunt game, four user-defined units, and eight additional commands. All of the items are described in this chapter. You can review the command names by pressing [←][LIBRARY] UTILS [←][REVIEW].

Utilities Commands and Units

Key	Programmable Command	Description
MINE	MINEHUNT	Starts the Minehunt game.
ZFACT	ZFACTOR	Calculates the gas compressibility factor *Z*.
FANNI	FANNING	Calculates the Fanning friction factor *f*.
DARCY	DARCY	Calculates the Darcy friction factor *d*.
F0λ	F0λ	Calculates the fraction of total black-body emissive power.
SIDEN	SIDENS	Calculates the intrinsic density of silicon as a function of temperature.
TDELT	TDELTA	Calculates a temperature increment.
TINC	TINC	Adds a temperature increment.
GMOL	gmol	User-defined unit (gram-mole).

Utilities Commands and Units (continued)

Key	Programmable Command	Description
LBMO	lbmol	User-defined unit (pound-mole).
RPM	rpm	User-defined unit (rpm).
DB	dB	User-defined unit (dB).
ELVER	ELVERSION	Displays the title and version of the HP Solve Equation Library.

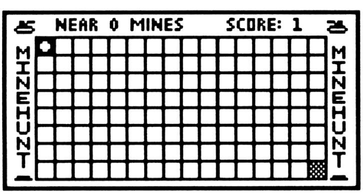
Minehunt Game

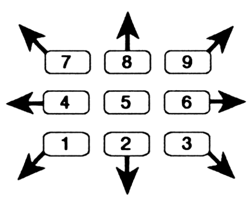
The Minehunt game is a battlefield adventure. Your mission is to travel safely across a battlefield strewn with buried mines. Your life depends on your ability to navigate safely!

Playing the Game

Use the command MINEHUNT or press [←] [LIBRARY] UTILS MINE to begin the Minehunt game.

In the game, you are standing in the upper-left corner of an 8 × 16 battlefield grid. Your mission is to travel safely to the lower-right corner, avoiding buried mines along the way.





To traverse the battlefield, press the number keys or arrow keys. The number keys allow you to move diagonally.

Your only ally in this dangerous trip is a slightly defective mine detector. The detector can't tell you where the mines are, but it is sensitive enough to say how many are nearby. If a mine is under any of the eight squares adjacent to your position, the detector will tell you. For example, if there is a mine under the square to your right and a mine under the square to your left, the mine detector says `NEAR 2 MINES`. Up to seven mines may be next to you!

As you travel, the squares you occupy turn black, and the number of black squares shows as the score at the top of the battlefield. You can try to complete your mission using the least number of squares — or you can try to visit the largest number of squares.

The game ends when you reach the lower-right corner of the battlefield or (heaven forbid!) step on a mine. Then, your suddenly-enhanced vision shows all the mines. At this time you can press the left menu key to start another game.

Interrupting and Restarting a Game

You can exit the game at any time by pressing [ATTN].

To interrupt and save a game, press [STO]. This creates a variable *MHpar* in the current directory and ends the game. This variable contains the status of the interrupted game. If *MHpar* exists when you next start Minehunt, the interrupted game resumes and *MHpar* is purged.

Changing the Number of Mines

You can change the number of mines in the battlefield by creating a variable named *Nmines* containing the desired number. When Minehunt starts, it uses this variable to change the default number of mines. If *Nmines* contains a real number, the absolute value of the integer portion defines the number of mines in the battlefield. There must be at least 1 mine and no more than 64 mines. If *Nmines* is negative, the mines are visible during the game.

User-Defined Units

The Utilities application provides four user-defined units: "gmol" (gram-moles, mol), "lbmol" (pound-moles, approximately 454 mol), "rpm" (revolutions per minute, 1/min), and "dB" (decibels, dimensionless). You can use their Utilities menu keys as typing aids. To fully use these units, add them to the custom menu.

Example: Putting Units in a Custom Menu. Graham Molwaite and Desi Bell want a custom menu with all four user-defined units.

Clear the stack and get the Utilities menu. Create a list and insert a unit object for each unit.

[→] [CLR]
[←] [LIBRARY] UTILS [NXT]
[←] [{ }]
1 [→] [_] GMOL [SPC]
1 [→] [_] LBMO [SPC]
1 [→] [_] RPM [SPC]
1 [→] [_] DB [ENTER]

```
1: { 1_gmol 1_lbmol 1_
   rpm 1_dB }
TDELT TINC GMOL LBMO RPM DB
```

Add this to the custom menu. Then enter a value of 6 gram-moles.

[→] [MODES] MENU
6 GMOL

```
1:                6_gmol
GMOL LBMO RPM DB
```

You get the custom menu at any time by pressing [CST]. (The custom menu information is stored in variable *CST*.) The following examples show how to use the custom menu:

- Press GMOL to add units to the number you're entering or to append units to the unit numerator of the object in level 1.
- Press [→] GMOL to append units to the unit denominator of the object in level 1.
- Press [←] GMOL to convert the unit object in level 1 to "gmol."

For more information, see the "Unit Management" chapter in the *HP 48 Owner's Manual.*

Commands

The Utilities application provides eight miscellaneous commands. Several calculate physical parameters used by the Equation Library. You can execute these commands in the command line, in equations, and in programs.

For example, you can include ZFACTOR in an equation:

```
'Z=ZFACTOR(Tr,Pr)'
```

You can include ZFACTOR in a program:

```
« ... Tr Pr ZFACTOR 'Z' STO ... »
```

ZFACTOR Function

The ZFACTOR(*Tr*, *Pr*) function calculates the gas compressibility factor *Z*, a correction factor for the nonideal behavior of a real gas. (The derived *Z* factor is accurate for hydrocarbon gases only.) *Tr* is the reduced temperature—the ratio of the actual temperature (*T*) to the pseudocritical temperature (*Tc*). (You must calculate the ratio using absolute temperatures.) *Pr* is the reduced pressure—the ratio of the actual pressure (*P*) to the pseudocritical pressure (*Pc*). *Tr* and *Pr* must be real numbers or unit objects that reduce to dimensionless numbers. *Tr* must be between 1.05 and 3.0. *Pr* must be between 0 and 30.

ZFACTOR takes *Tr* from level 2 and *Pr* from level 1, and it returns the *Z* factor in level 1.

FANNING Function

The FANNING(ε/D, Re) function calculates the Fanning friction factor f, a correction factor for the frictional effects of certain fluid flows (constant temperature, cross-section, velocity, and viscosity—typical pipe flow). ε/D is the relative roughness—the ratio of the conduit roughness to its diameter. Re is the Reynolds number (conduit diameter × average velocity × fluid density / fluid viscosity). The function uses different computation routines for laminar flow ($Re \leq 2100$) and turbulent flow ($Re > 2100$). e/D and Re must be real numbers or unit objects that reduce to dimensionless numbers. ε/D and Re must be greater than 0.

FANNING takes ε/D from level 2 and Re from level 1, and it returns the Fanning friction factor in level 1.

DARCY Function

The DARCY(ε/D, Re) function calculates the Darcy friction factor d as the Fanning friction factor times 4. See the previous topic.

DARCY takes ε/D from level 2 and Re from level 1, and it returns the Darcy friction factor in level 1.

F0λ Function

The F0λ(λ, T) function calculates the fraction of total black-body emissive power at temperature T between wavelengths 0 and λ. If you use no units, λ must have implied units of meters and T must have implied units of K.

F0λ takes λ from level 2 and T from level 1, and it returns the dimensionless fraction in level 1.

SIDENS Function

The SIDENS(T) function calculates the intrinsic density of silicon, *ni*, as a function of temperature, T. If T is a unit object, it must reduce to a pure temperature, and the density is returned as a unit object with units of $1/cm^3$. If T is a real number, its units are assumed to be K, and the density is returned as a real number with implied units of $1/cm^3$. T must be between 0 and 1685 K.

SIDENS takes T from level 1, and it returns the density in level 1.

TDELTA Function

The TDELTA(T_2, T_1) function subtracts two points on a temperature scale, yielding a temperature *increment* (not an actual temperature). T_2 is the final temperature, and T_1 is the initial temperature. The returned increment is the temperature change. If T_2 and T_1 are unit objects, the increment is returned as a unit object with the same units as T_2 (the level 2 argument, unlike most unit management operations). If they're real numbers, the increment is returned as a real number.

For unit objects, the increment is calculated differently from the way [−] calculates it. T_1 is converted to the units of T_2, the numeric values are subtracted, and the units are appended.

TDELTA takes T_2 from level 2 and T_1 from level 1, and it returns the temperature increment in level 1.

TINC Function

The TINC(T_1, ΔT) function adds a temperature *increment* (not an actual temperature) to a point on a temperature scale. Use a negative increment to subtract the increment from the temperature. T_1 is the initial temperature, and ΔT is the temperature increment. The returned temperature is the resulting final temperature. If T_1 and ΔT are unit objects, the final temperature is returned as a unit object with the same units as T_1 (the level 2 argument, unlike most unit management operations). If they're real numbers, the final temperature is returned as a real number.

For unit objects, the final temperature is calculated differently from the way [+] calculates it. ΔT is converted to an *increment* on the temperature scale used by T_1, the numeric values are added, and the units are appended. (A 1° increment on the Celsius or Kelvin scale equals a 1.8° increment on the Fahrenheit or Rankine scale.)

TINC takes T_1 from level 2 and ΔT from level 1, and it returns the final temperature in level 1.

ELVERSION Command

The ELVERSION command displays the HP Solve Equation Library Application Card title, product number, and software version number.

ELVERSION doesn't affect the stack.

A

Support and Service

Calculator Support

You can obtain answers to questions about using your application card from our Calculator Support department. Our experience has shown that many customers have similar questions about our products, so we have provided the following section, "Answers to Common Questions." Also, the *HP 48 Owner's Manual* includes similar information about the calculator itself. If you don't find the answer to your question below or in appendix A of the *HP 48 Owner's Manual*, contact us at the address or phone number on the inside back cover.

Answers to Common Questions

Q: *I'm not sure whether the application card is malfunctioning or if I'm doing something incorrectly. How can I verify that it and the calculator are operating properly?*

A: You can check the application card by turning on the calculator and pressing [⬑] [LIBRARY]. The calculator checks application cards whenever it turns on—if `Invalid Card Data` or `Port Not Available` is displayed at turn-on, the card requires service. If the LIBRARY menu doesn't include the application names shown in chapter 1, the card may require service—but first check that you installed it properly. To check the calculator, see appendix A of the *HP 48 Owner's Manual*.

Q: *What do three dots (...) mean at the end of a catalog display line?*

A: The three dots (called an *ellipsis*) indicate that the displayed object is too long to show on one line. To view the complete object, highlight the object and press [ENTER]—press [ENTER] or [ATTN] to return to the catalog.

Q: *While using the Equation Library or Multiple-Equation Solver, I accidentally stored a value in the wrong variable and the menu label turned black. What should I do?*

A: It depends on your problem. If you *do* know a value for the variable with the black label, simply store the new value — the label *should* be black. If instead you want to *solve* for that variable, you must either solve for only that variable (press [↰] and that menu key) or else turn the label white again — see "Controlling Variables" in chapter 2.

Q: *The results from the Equation Library or Multiple-Equation Solver are unusual or illogical. What's wrong?*

A: Several conditions can cause this problem, including improper unit usage. See "Interpreting Results" in chapter 2.

Q: *The calculator displays* `Bad Guess(es)` *while running the Equation Library or Multiple-Equation Solver. What's wrong?*

A: The HP 48 root finder encountered variable values or units that prevented a solution. See "Interpreting Results" in chapter 2.

Q: *The calculator displays* `Too Many Unknowns` *while running the Equation Library or Multiple-Equation Solver. What's wrong?*

A: The Multiple-Equation Solver encountered a combination of unknown variables that prevented a solution. See "Interpreting Results" in chapter 2.

Q: *The calculator beeps and displays a message different from the ones listed above. How do I find out what's wrong?*

A: See appendix E in this manual for application card messages. See appendix B in the *HP 48 Owner's Manual* for calculator messages, which can occur while using the application card.

Q: *While using the Equation Library, I removed units from all variables and get wrong answers for the same values. What's wrong?*

A: Your implied units may not be compatible, especially with those of constants or functions. See "Choosing Unit Options" in chapter 2.

Q: *I can't find some variables I used earlier. Where did they go?*

A: You may have been using the variables in a different directory. If you can't remember which directory you were using, you'll need to check all the directories in your calculator.

Q: *Sometimes my HP 48 seems to pause for a few seconds during a calculation or while displaying an equation in the Equation Library. Is anything wrong?*

A: Nothing is wrong. The calculator does some system cleanup from time to time to eliminate temporary objects created from normal operation. This cleanup process frees memory for current operations. Some large equations in the Equation Library require extra time to be displayed in EquationWriter format. The ⧗ annunciator shows that a time-consuming operation is in process.

Q: *Why does the Multiple-Equation Solver seem to take more time to run than the built-in HP Solve application?*

A: The Multiple-Equation Solver may need to solve several equations in order to find your solution. (Also see the next question.)

Q: *Why do equations from the Equation Library seem to take longer to solve than my own equations?*

A: If you're using units in your calculations, the underlying math routines carry an additional task of sorting out the units. But this is usually faster than using a rapid solve procedure and then sorting out the units by hand. You can use the no-units option in the Equation Library to speed up calculations, but remember that units are implied by the SI or English setting.

Q: *While using the Periodic Table, I use* [α] *to search for a property, but the calculator just beeps. Why?*

A: The property may be annotated with special information, so the first character may be *. Press [α] [×] to search for a property marked this way.

Environmental Limits

To maintain the reliability of HP 48 plug-in application cards observe the following temperature and humidity limits:

- Operating temperature: 0 to 45 °C (32 to 113 °F).
- Storage temperature: −20 to 60 °C (−4 to 140 °F).
- Operating and storage humidity: 90% relative humidity at 40 °C (104 °F) maximum.

Limited One-Year Warranty

What Is Covered. *The card is warranted by Hewlett-Packard against defects in materials and workmanship for one year from the date of original purchase.* If you sell your unit or give it as a gift, the warranty is automatically transferred to the new owner and remains in effect for the original one-year period. During the warranty period, we will repair or, at our option, replace at no charge a product that proves to be defective, provided you return the product, shipping prepaid, to a Hewlett-Packard service center. (Replacement may be made with a newer model of equal or better functionality.)

This warranty gives you specific legal rights, and you may also have other rights that vary from state to state, province to province, or country to country.

What Is Not Covered. This warranty does not apply if the product has been damaged by accident or misuse or as the result of service or modification by other than an authorized Hewlett-Packard service center.

No other express warranty is given. The repair or replacement of a product is your exclusive remedy. **ANY OTHER IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS IS LIMITED TO THE ONE-YEAR DURATION OF THIS WRITTEN WARRANTY.** Some states, provinces, or countries do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you. **IN NO EVENT SHALL HEWLETT-PACKARD COMPANY BE LIABLE FOR CONSEQUENTIAL DAMAGES.** Some states, provinces, or countries do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Products are sold on the basis of specifications applicable at the time of manufacture. Hewlett-Packard shall have no obligation to modify or update products, once sold.

Consumer Transactions in the United Kingdom. This warranty shall not apply to consumer transactions and shall not affect the statutory rights of a consumer. In relation to such transactions, the rights and obligations of Seller and Buyer shall be determined by statute.

If the Card Requires Service

Hewlett-Packard maintains service centers in many countries. These centers will repair a card, or replace it with the same model or one of equal or better functionality, whether it is under warranty or not. There is a service charge for service after the warranty period. Calculators and accessories normally are serviced and reshipped within 5 working days.

- **In the United States:** Send the card to the Corvallis Service Center listed on the inside of the back cover.
- **In Europe:** Contact your Hewlett-Packard sales office or dealer, or Hewlett-Packard's European headquarters (address below) for the location of the nearest service center. *Do not ship the card for service without first contacting a Hewlett-Packard office.*

Hewlett-Packard S.A.
150, Route du Nant-d'Avril
P.O. Box CH 1217 Meyrin 2
Geneva, Switzerland
Telephone: 022 780.81.11

- **In other countries:** Contact your Hewlett-Packard sales office or dealer or write to the Corvallis Service Center (listed on the inside of the back cover) for the location of other service centers. If local service is unavailable, you can ship the card to the Corvallis Service Center for repair.

  All shipping, reimportation arrangements, and customs costs are your responsibility.

Service Charge. Contact the Corvallis Service Center (inside back cover) for the standard out-of-warranty repair charges. This charge is subject to the customer's local sales or value-added tax wherever applicable.

Calculator products damaged by accident or misuse are not covered by the fixed charges. These charges are individually determined based on time and material.

Shipping Instructions. If your card requires service, ship it to the nearest authorized service center or collection point.

- Include your return address and a description of the problem.
- Include proof of purchase date if the warranty has not expired.
- Include a purchase order, check, or credit card number plus expiration date (VISA or MasterCard) to cover the standard repair charge.
- Ship your card postage *prepaid* in adequate protective packaging to prevent damage. Shipping damage is not covered by the warranty, so we recommend that you insure the shipment.

Warranty on Service. Service is warranted against defects in materials and workmanship for 90 days from the date of service.

B

Equation Library Notes

About half of the equations in the Equation Library have both a *display form* and a *calculation form* (indicated by * in the equation display). The display form is the easily recognizable form, shown when you press [ENTER] or EQN in the Equation Library. The calculation forms have a number of enhancements that cause the equation to be more complete, accurate, or robust. (See "Viewing Equations" in chapter 2.)

Differences between Display and Calculation Forms

Calculation forms provide enhanced performance by using one or more of the following techniques:

- Embedding universal constants using CONST.
- Eliminating fractional units using UBASE.
- Dropping extraneous radians using CONST(rad).
- Making implied trigonometric units explicit using CONST(rad).
- Allowing nonabsolute temperature units using UBASE.
- Allowing alternate interpretation of temperature units using TDELTA.
- Providing special processing for EXP and LN using UBASE.
- Replacing variables with calculation functions.
- Expanding the equation's range of validity using IFTE.

You can use these same techniques for enhancing your equations. The following sections describe the techniques in detail.

Embedding Constants

Example: Gases — Ideal Gas Law

Display form: P·V=n·R·T

Calculation form: P·V=n·CONST(R)·UBASE(T)

Using CONST. Many equations use universal constants, such as the universal gas constant (*R*) for gases. Rather than require that you look up a value for these, they're embedded in the equation. This simplifies the use of the equation and allows an unnecessary variable to be removed from the solver menu.

The embedded constants must have units if the equation variables have units, and must not have units if the equation variables are being used without units. To allow for this without altering the equation for each unit option, every constant is embedded as a function that extracts the constant from the Constants Library. In the above example, CONST(R) returns the gas constant for use in the equation.

Unit Operations. CONST respects the SI/English unit selection that the Equation Library uses (Units Type flag: SI if clear, English if set), so it returns either an SI or English constant with appropriate units. When units are being used, the SI or English units associated with the constant are of no consequence. Their purpose is to ensure dimensional consistency for the solver. The solver will convert the solved result, in whatever units it is calculated in, to the desired units for that variable, as indicated by the variable's current value.

CONST also respects the no-units selection that the Equation Library uses (Units Usage flag: units if clear, no units if set), so it returns either an SI or English dimensionless value if no units are being used. When units are not being used, the SI or English value of the constant is of great importance. The value implies a set of units that the (dimensionless) variable values must use in order to get correct results. For example, if the SI and no-units options are being used, CONST(R) returns 8.31451, a value that implies that the (dimensionless) pressures must be in Pa, the volumes must be in m^3, and the temperatures must be in K.

Overriding CONST. You may want to use an equation with no units to get faster solutions. If the equation has embedded constants, the implied set of units may not match what you want to use. To replace the library constant with one consistent with your needs, you can place an override constant in a variable named "const" followed by the constant name (for example, *constR* to override the universal gas constant). When CONST tries to find R in the Constants Library, it first looks in *constR* for an override value.

Operating without Units. To allow use with both the units and no-units options, the override value should have units. As before, the units are necessary only to ensure dimensional consistency for the solver. CONST strips off the units if you're using the no-units option, in which case the units implied by the dimensionless value are very important. If .00831451_kJ/mol·K is placed in *constR* as an override value, it is used in both SI and English modes when units are used. When the no-units option is being used, the implied units for pressure will be kPa, a more common SI measure of pressure.

A Special Case. CONST(angl) is a special case. CONST(angl) is `180_°`, and it's used whenever angular constants are needed, such as for the angles of a regular polygon. When the no-unit option is being used, operations using angles are dependent on the current angle mode. CONST(angl) returns 180 degrees, π radians, or 200 grads accordingly.

Eliminating Fractional Units

Example: Electricity—Resonant Frequency

Display form: $Qs=\frac{1}{R}\cdot\sqrt{\frac{L}{C}}$

Calculation form: $Qs=\frac{1}{R}\cdot\sqrt{UBASE\left(\frac{L}{C}\right)}$

The HP 48 unit management system allows units with fractional powers (such as $acre^{.5}$). Unit conversions are done with unit exponents truncated to signed, eight-bit integers, rather than with full floating point values. This means that combinations of units with fractional powers that should cancel out may not cancel out properly. The main operations where this occurs are square roots and noninteger powers. Equations can accommodate this by using UBASE with the quantity prior to either one of these operations.

There are two instances in "Solid State Devices" where unit exponents are either .5 or 1.5. The solution for these cases uses the following principle:

$$A \cdot B = \sqrt{\text{UBASE}(A^2 \cdot B^2)}$$

or, in the case of fractional powers,

$$A^{.5} \cdot B^{.5} = \sqrt{\text{UBASE}(A \cdot B)}$$

Squaring the variables eliminates the fractional powers. UBASE combines and cancels like units, and the square root can then produce a valid unit with an integer power.

Dropping Extraneous Radians

Example: Electricity—RLC Current Delay

Display form: $XL = \omega \cdot L$

Calculation form: $XL = \dfrac{\omega \cdot L}{\text{CONST(rad)}}$

In practice, there are many situations where radians can be completely ignored. In the above example, *XL* must be in ohms, but $\omega \cdot L$ is in $r \cdot mH/s$, with an extraneous r (radian) unit. Radians in the HP 48 unit management system are dimensionless, but still have a conversion factor (allowing π radians to be converted to 180 degrees). Because radians are dimensionless, they do not affect dimensional consistency checks, but their conversion factor does alter the converted value. Consequently, in contexts where common practice is to drop the radian units, the calculator will yield results off by a factor of 2π.

Equation Library equations multiply or divide by 1_r to eliminate the extra radian unit. Because embedded unit objects cannot reside in equations and still work with the no-units option, the correction factor is CONST(rad).

Making Trigonometric Units Explicit

Example: Columns and Beams — Eccentric Columns

Display form: $\ldots \text{COS}\left(\frac{L}{2 \cdot r} \cdot \sqrt{\frac{P}{E \cdot A}}\right) \ldots$

Calculation form: $\ldots \text{COS}\left(\frac{\text{CONST(rad)} \cdot L}{2 \cdot r} \cdot \sqrt{\text{UBASE}\left(\frac{P}{E \cdot A}\right)}\right) \ldots$

Calculated angles that are dimensionless but implied to be in radians are multiplied by CONST(rad). This prevents unexpected factors of 2π from being included in conversions. If those angles are arguments of trigonometric functions, the presence of the radian unit (or any other angular unit) overrides the current angle mode, which does not affect calculations with units.

Allowing Nonabsolute Temperature Units

Example: Gases — Ideal Gas Law

Display form: $P \cdot V = n \cdot R \cdot T$

Calculation form: $P \cdot V = n \cdot \text{CONST(R)} \cdot \text{UBASE(T)}$

Many real-world calculations require absolute temperatures (K), but the available temperature values may be in nonabsolute units (°F, °C). UBASE is used to convert any supplied temperature units to absolute units (K). This is also done whenever a temperature ratio is calculated.

Allowing Alternate Interpretation of Temperature Units

Example: Heat Transfer — Convection

Display form: $Q = h \cdot A \cdot (Th - Tc)$

Calculation form: $Q = h \cdot A \cdot TDELTA(Th, Tc)$

All temperature additions and subtractions in the HP 48 unit management system are performed based on absolute temperatures. This preserves mathematical commutativity of addition, but does not always match engineering needs. Engineers often distinguish between a point on a temperature scale (degree Fahrenheit) and a distance (or increment) along that scale (Fahrenheit degree). TDELTA is used to subtract two temperature points and return a temperature increment, which is a different type of subtraction operation.

Special Processing for EXP and LN

Example: Electricity — RC Transient

Display form: $V = Vf - (Vf - Vi) \cdot e^{\frac{-t}{R \cdot C}}$

Calculation form: $V = Vf - (Vf - Vi) \cdot EXP\left(UBASE\left(\frac{-t}{R \cdot C}\right)\right)$

EXP and LN do not allow dimensioned arguments. UBASE eliminates dimensionally consistent units by first converting all units to their base SI units, then combining and canceling like units. This approach is also required for any function that does not allow dimensioned arguments, but EXP and LN are the only such functions used in the Equation Library.

Equation Library equations are displayed using the symbolic constant *e* raised to a power, but calculated using the EXP function to improve speed and accuracy.

Replacing Variables with Calculation Functions

Example: Gases — Real Gas Law

Display form: $P \cdot V = n \cdot Z \cdot R \cdot T$

Calculation form:

$$P \cdot V = n \cdot ZFACTOR\left(\frac{UBASE(T)}{UBASE(Tc)}, \frac{P}{Pc}\right) \cdot CONST(R) \cdot UBASE(T)$$

Certain engineering quantities are replaced by calculation functions. These functions improve the speed and accuracy of the equations using them and provide useful utilities for engineering uses other than these equations. The Equation Library uses the ZFACTOR, FANNING, F0λ, and SIDENS functions — see chapter 8 for details.

Expanding the Range of Validity

Example: Columns and Beams — Cantilever Moment

Display form: $Mx = P \cdot (x - a) + M + \frac{w}{2} \cdot (L^2 - 2 \cdot L \cdot x + x^2)$

Calculation form:

$$Mx = IFTE(x \le a, P \cdot (x - a), 0) + IFTE(x \le c, M, 0) + \frac{w}{2} \cdot (L^2 - 2 \cdot L \cdot x + x^2)$$

IFTE, the algebraic form of the IF...THEN...ELSE...END structure, evaluates different expressions based on certain conditions. In this beam equation, for example, a single equation is applied to beam conditions on either side of a point load or applied moment.

Using No Units with the Equation Library

The Equation Library makes an extra effort to allow units to be used. Using the unit management system provides these benefits:

- Allows the flexibility of arbitrary units for inputs.
- Annotates calculated values with their physical meaning.
- Eliminates the drudgery associated with verifying unit consistency.

For many practical engineering problems, resolving the units is a far more complex problem than the calculation itself. This process causes the solver to execute more slowly. However, when the units are the majority of the problem, it is faster to use a slower solver and let it sort out the units than to use a faster solver and resolve the unit problems manually.

For those equations where the units are not a significant part of the problem, you can select the no-units option in the Equation Library. In this case you're responsible for managing the implied units, as described in this section.

Dimensional Coexistence

When units aren't used, all values supplied to an equation must coexist dimensionally. This means that the implied units of each value must combine and cancel properly with the implied units of the other values. For example, in the Motion — Linear Motion equation v=v0+a·t, velocity in mph would not coexist dimensionally with a time in minutes and an acceleration in ft/s^2. Using the no-units option requires that you resolve each coexistence conflict manually. This is important even if you use the default SI or English units — the units supplied by the Equation Library are based on common scientific and engineering practice, not based on best dimensional coexistence. *If you enter values with units for an equation, and then choose the no-units option, those same values without their units will probably give results that seem wrong.* This is not too surprising for English units, because they rarely coexist dimensionally. Pure SI units generally do coexist, but the Equation Library default SI units include commonly-used prefixes such as mm^4, cm, kPa, and kg. In general, this prevents values used with SI units from coexisting dimensionally when used without units.

For example, you can solve for energy using the "Electricity — Inductive Energy" equation. If you enter 5 for *I* and 300 for *L*, the result for *E* is 3.75_J if you use units and 3750 if you use no units.

Embedded Constants

Embedded constants imply a certain set of units on the variables. This set makes dimensional coexistence more difficult, because the Constants Library has selected the implied units even if units are not being used. You can use the constant override variable to change the implied units without altering the equation itself—see "Overriding CONST" under "Embedding Constants" earlier in this appendix.

For example, you can solve for pressure using the "Fluids—Pressure at Depth" equation. If you enter 14.7 for *P0*, 62.4 for ρ, and 10 for *h*, the result for *P* is 19.0_psi if you use English units, but the result is 20091.28 if you use no units (English implied) and don't set up *constg* to account for the implied units.

Absolute Temperatures

In most instances, absolute-temperature units must be used. In a few equations, you can use nonabsolute temperatures as long as they are on the same temperature scale.

Angle Mode

All trigonometric calculations are performed in the current angle mode (degrees, radians, or grads). Angular variable values, either supplied or calculated, must be consistent with that mode.

C

Operations, Variables, Flags, and Identifiers

The HP Solve Equation Library Application Card provides the operations summarized in the following table. See the appropriate chapters for detailed information.

Commands and Units

Name	Description	Chapter
AMORT	Calculates amortization.	6
CONLIB	Starts the Constants Library.	5
CONST	Returns a constant.	5
DARCY	Calculates the Darcy friction factor.	8
dB	User-defined unit (dB).	8
ELVERSION	Displays the title and version of card.	8
EQNLIB	Starts the Equation Library.	2
F0λ	Calculates black-body power fraction.	8
FANNING	Calculates the Fanning friction factor.	8
gmol	User-defined unit (gram-mole).	8
lbmol	User-defined unit (pound-mole).	8
MCALC	Makes variables not user-defined.	7
MINEHUNT	Starts the Minehunt game.	8
MINIT	Creates *Mpar* from *EQ*.	7
MITM	Changes the solver title and menu.	7
MOLWT	Returns a molecular weight.	4
MROOT	Solves for variables.	7

Commands and Units (continued)

Name	Description	Chapter
MSOLVR	Gets the solver menu.	2, 7
MUSER	Makes variables user-defined.	7
PERTBL	Starts the Periodic Table.	4
PTPROP	Returns a property for an element.	4
rpm	User-defined unit (rpm).	8
SIDENS	Calculates intrinsic density of silicon.	8
SOLVEQN	Starts the solver for a subject and title.	2
TDELTA	Calculates a temperature increment.	8
TINC	Adds a temperature increment.	8
TVM	Displays the TVM menu.	6
TVMBEG	Sets Begin payment mode.	6
TVMEND	Sets End payment mode.	6
TVMROOT	Solves for a TVM variable.	6
ZFACTOR	Calculates the *Z* factor.	8

The application card uses four reserved variables to store information. *MHpar*, *Mpar*, and *PTpar* are type 26 objects (Library Data), which you can't edit.

Reserved Variables

Variable	Purpose
MHpar	Stores the Minehunt game status.
Mpar	Stores the equation set for the Multiple-Equation Solver.
Nmines	Stores the number of mines to use for Minehunt. If negative, the mines are visible during the game.
PTpar	Stores pointer position information for the Periodic Table.

The application card uses three user flags. The default state for the user flags is "clear."

User Flag Definitions

Flag	Name	Clear	Set
60	Units Type	SI units	English units
61	Units Usage	Use units	Use no units
62	Payment Mode	End mode	Begin mode

The application card contains eight library objects. A unique library identifier (port and number) is associated with each library object — but the identifier depends upon the port in which you installed the card (port 1 or port 2). To see the library numbers, press [←] [LIBRARY] [NXT] and PORT1 or PORT2. Seven of the library objects are automatically attached to the *HOME* directory when you plug in the application card. This means that you can immediately access all of the features from any of your subdirectories. However, you *can* restrict access by attaching an individual library object (specified by its library identifier) to a lower directory and unattaching it from the *HOME* directory. See the "Using Plug-In Cards and Libraries" chapter in the *HP 48 Owner's Manual* for more information.

Library Names and Identifiers

Library Name	Library Identifier
EQLIB	:1:273 or :2:273
PRTBL	:1:272 or :2:272
COLIB	:1:271 or :2:271
FIN	:1:270 or :2:270
MES	:1:269 or :2:269
UTILS	:1:268 or :2:268
(Equation Reference)	:1:267 or :2:267
(Catalog Utility)	:1:266 or :2:266

D

References

Anderson, Herbert L., ed. *Physics Vade Mecum*. American Institute of Physics, 1981.

Avallone, Eugene A., and Theodore Baumeister III, ed. *Marks' Standard Handbook for Mechanical Engineers*, 9th ed. McGraw-Hill, 1987.

Cohen, E. Richard, and Barry N. Taylor. "The 1986 CODATA Recommended Values of the Fundamental Physical Constants," In *Journal of Physical and Chemical Reference Data*, Vol. 17, no. 4. National Bureau of Standards, 1988.

Davis, Raymond E., Kenneth D. Gailey, and Kenneth W. Whitten. *Principles of Chemistry*. Saunders College Publishing, 1984.

Dean, John A., ed. *Lange's Handbook of Chemistry*, 12th ed. McGraw-Hill, 1979.

Dranchuk, P.M., R.A. Purvis, and D.B. Robinson. "Computer Calculations of Natural Gas Compressibility Factors Using the Standing and Katz Correlation," In *Institute of Petroleum Technical Series*, no. IP 74-008. 1974.

Gere, James M., and Stephen P. Timoshenko. *Mechanics of Materials*, 2d ed. PWS Engineering, Boston, 1984.

Halliday, David, and Robert Resnick. *Fundamentals of Physics*, 3d ed. John Wiley & Sons, 1988.

Hewlett-Packard Company. *HP-41C Thermal Transport and Science Pac*. 1983.

Mahan, Bruce H. *University Chemistry*, 3d ed. Edison Wesley, 1975.

McKelvey, John P., and Howard Grotch. *Physics for Science and Engineering*. Harper & Row, 1978.

Meriam, J.L., and L.G. Kraige. *Engineering Mechanics*, 2d ed. John Wiley & Sons, 1986.

Muller, Richard S., and Theodore I. Kamins. *Device Electronics for Integrated Cicuits*, 2d ed. John Wiley & Sons, 1986.

Perry, Robert H., and Cecil H. Chilton, ed. *Chemical Engineers' Handbook*, 5th ed. McGraw-Hill, 1973.

Plonsey, Robert, and Robert E. Collin. *Principles and Applications of Electromagnetic Fields*. McGraw-Hill, 1961.

Popov, Egor P. *Introduction to Mechanics of Solids*. Prentice-Hall, 1968.

Reynolds, William C., and Henry C. Perkins. *Engineering Thermodynamics*, 2d ed. McGraw-Hill, 1977.

Roark, Raymond J. *Formulas for Stress and Strain*, 4th ed. McGraw-Hill, 1965.

Sargent-Welch Scientific Company. *Periodic Table of the Elements*. 1980.

Serghides, T.K. "Estimate Friction Factor Accurately," In *Chemical Engineering*, Mar. 5, 1984.

Siegel, Robert, and John Howell. *Thermal Radiation Heat Transfer*, Vol. 1. National Aeronautics and Space Administration, 1968.

Smith, Ralph J. *Electronics: Circuits and Devices*. John Wiley and Sons, 1980.

Sze, S. *Physics of Semiconductors*, 2d ed. John Wiley & Sons, 1981.

Takacs, G. "Comparisons Made for Commputer Z-Factor Calculations," In *Oil and Gas Journal*, Dec. 20, 1976.

Weast, Robert C., ed. *CRC Handbook of Chemistry and Physics*, 65th ed. Chemical Rubber Publishing Company, 1985.

Welty, Wicks, and Wilson. *Fundamentals of Momentum, Heat and Mass Transfer*. John Wiley & Sons, 1969.

Zigrang, D.J., and N.D. Sylvester. "A Review of Explicit Friction Factor Equations," In ASME *Journal of Resources Technology*, Vol. 107. June 1985.

E

Messages

This appendix lists all error messages given by the HP Solve Equation Library Application Card. In the first table, messages are listed alphabetically. In the second table, they're listed numerically by message number.

Messages Listed Alphabetically

Message	Meaning	# (hex)
All Variables Known	There are no unknown variables to solve for.	10D05
Bad Molecular Formula	Formula is invalid or incomplete. Look for mismatched parentheses or invalid element names.	11001
EQ Invalid for MINIT	*EQ* must contain at least two equations (or programs) and two variables.	10D03
Illegal During MROOT	Multiple-Equation Solver command attempted during MROOT execution.	10D06
I%YR/PYR ≤ -100	Interest per period must be greater than −100%.	10E03
Invalid N	Attempted to calculate *I%YR* with $N < 1$ or $N \geq 10^{10}$.	10E04
Invalid Mpar	*Mpar* variable was not created by MINIT.	10D01

Messages Listed Alphabetically (continued)

Message	Meaning	# (hex)
Invalid PYR	*PYR* must be a positive real number.	10E05
Invalid #Periods	AMRT requires a positive integer number of periods.	10E06
Many or No Solutions	A value for *I%YR* cannot be calculated. Check the values stored in *PV*, *PMT*, and *FV*. Check for correct signs.	10E02
No Solution	A value for *I%YR* cannot be calculated. Check the values stored in *PV*, *PMT*, and *FV*. Check for correct signs.	10E01
Single Equation	Only one equation has been supplied to the Multiple-Equation Solver. Use the HP Solve application.	10D02
Too Many Unknowns	The Multiple-Equation Solver can't calculate a value given the current knowns. Supply another value or add an equation.	10D04
Undefined Constant	The name supplied to CONST isn't in the Constants Library.	10F01
Undefined Element	The element supplied to PTPROP doesn't exist.	11002
Undefined Property	The property number supplied to PTPROP isn't in the Periodic Table.	11003
Undefined TVM Variable	The variable name supplied to TVMROOT isn't *N*, *I%YR*, *PV*, *PMT*, or *FV*.	10E07

Messages Listed Numerically

# (hex)	Message
Multiple-Equation Solver Messages	
10D01	Invalid Mpar
10D02	Single Equation
10D03	EQ Invalid for MINIT
10D04	Too Many Unknowns
10D05	All Variables Known
10D06	Illegal During MROOT
Finance Messages	
10E01	No Solution
10E02	Many or No Solutions
10E03	I%YR/PYR ≤ -100
10E04	Invalid N
10E05	Invalid PYR
10E06	Invalid #Periods
10E07	Undefined TVM Variable
Constants Library Message	
10F01	Undefined Constant
Periodic Table Messages	
11001	Bad Molecular Formula
11002	Undefined Element
11003	Undefined Property

Index

C

F

G

H

I

N

O

P

Q

R

S

Contacting Hewlett-Packard

For Information About Using the Plug-In Card. If you have questions about how to use the calculator or plug-in card, first check the table of contents, the index, and "Answers to Common Questions" in appendix A. (Also check the *HP 48 Owner's Manual*, appendix A.) If you can't find an answer in those manuals, you can contact the Calculator Support department:

Hewlett-Packard
Calculator Support
1000 N.E. Circle Blvd.
Corvallis, OR 97330, U.S.A.

(503) 757-2004
8:00 a.m. to 3:00 p.m. Pacific time
Monday through Friday

For Service. If your calculator or plug-in card doesn't seem to work properly, refer to appendix A for diagnostic instructions and information on obtaining service. If you are in the United States and your calculator or card requires service, mail it to the Corvallis Service Center:

Hewlett-Packard
Corvallis Service Center
1030 N.E. Circle Blvd.
Corvallis, OR 97330, U.S.A.
(503) 757-2002

If you are outside the United States, refer to appendix A for information on locating the nearest service center.

HP Calculator Bulletin Board System. The Bulletin Board provides for the exchange of software and information between HP calculator users, developers, and distributors. It operates at 300/1200/2400 baud, full duplex, no parity, 8 bits, 1 stop bit. The telephone number is (503) 750-4448. The Bulletin Board is a free service—you pay for only the long-distance telephone charge.

Contents

HEWLETT PACKARD

Part Number 82211-90001

Edition 2

English

Printed in Canada 7/90