SOLVIT

An HP-41 Program to Solve Multiple-Variable Equations for Arbitrary Unknowns

by

Ted W. Beers

y=slope·x+intercept

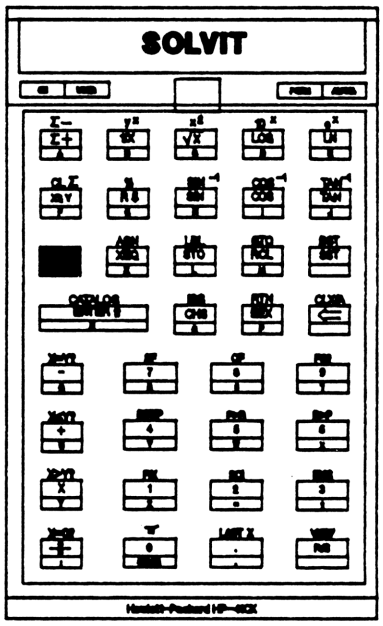
y=9.5 x=5 intercept=2

slope=?

$x=v_0 \cdot t - \frac{1}{2} g \cdot t^2 + x_0$

$x=0 \quad v_0=50 \quad x_0=10$

t=?



slope=1.5

t=10.4

INCLUDES:

- **Introduction**
- **Detailed Instructions**
- **Instruction Table**
- **Examples**
- **Technical Details**
- **Annotated Program Listings**
- **Program Barcode**

HP-41 SOLVIT Software Support

As the author of **SOLVIT, An HP-41 Program to Solve Multiple-Variable Equations for Arbitrary Unknowns**, I wish to support this software at a level appropriate to the complexity and customer cost of the package. To this end, I invite you to write to me directly if you discover problems with the software or the manual. I also welcome any comments you may have on the quality and usability of **HP-41 SOLVIT**. My address is:

Ted W. Beers
3947 NW Tamarack Dr.
Corvallis, OR 97330

Included with this Software Support sheet is a Purchase Registration form. Please fill it out and send it to me so that I can notify you if there are any revisions to the software or the manual.

I will try to respond to letters regarding **HP-41 SOLVIT**, but because of the range of applications of **SOLVIT** and the number of potential customers, I can not guarantee to answer all requests. Please enclose a self-addressed, stamped envelope with all correspondence.

Serial Number 2903A

TABLE OF CONTENTS

KEYSTROKE NOTATION

Keystroke Example	Description
[ ]	The gold shift key.
[XEQ]	A key labelled in white on the face of the key.
[ ] [GTO]	A shifted key labelled in gold letters above a key.
| Y |	A top row key located under the character(s) in the display above.
[ ] | X |	A shifted top row key located under the character(s) in the display above.
123	Numeric digit keys.
ABC	Alpha mode keys labelled in blue letters on the slanted faces of the keys.

INTRODUCTION: SOLVIT FOR THE HP-41

The HP-41 is well known for its ability to tackle complicated calculations with ease. The new HP-18C Business Consultant is appreciated for its formula solver that produces fast solutions to an endless variety of financial and business problems. **SOLVIT**, an HP-41 program to solve multiple-variable equations for arbitrary unknowns, provides the best of both these powerful calculators: the rich function set of the HP-41 combined with an equation solver styled after the HP-18C formula solver.

SOLVIT works with an equation containing any number of variables that you enter in the familiar HP-41 RPN format. Based on names that you assign to the variables in the equation, **SOLVIT** creates a custom menu that labels the keys used to input data and calculate solutions. Using the custom menu, you can assign known values to variables and have **SOLVIT** calculate the unknown value of any variable in the equation. The flexibility of this solution method encourages "what if" investigations into interesting equations that are too difficult to manipulate on paper.

These instructions start with the basics: the equipment you'll need to use **SOLVIT**. Then, a simple example is explained in detail to illustrate how **SOLVIT** assigns known data and calculates results. Subsequent sections cover all aspects of **SOLVIT**: **DETAILED INSTRUCTIONS FOR SOLVIT** explains **SOLVIT**'s features completely; **EXAMPLES** demonstrates further how **SOLVIT** simplifies a difficult problem; **TECHNICAL DETAILS** describes the more technical aspects of **SOLVIT** that will help you understand how **SOLVIT** works; The **APPENDIX** introduces supplementary programs you can use to save and recall frequently-used equations; and **REFERENCES**, **PROGRAM LISTINGS**, and **PROGRAM BARCODE** are provided in the last three sections.

Equipment Needed

To use **SOLVIT**, the following HP-41 configuration and software is required:

- HP-41C with two Memory Modules
  or
  HP-41C with one Quad Memory Module
  or
  HP-41CV
  or
  HP-41CX

- HP-41 Advantage Advanced Solutions Pac

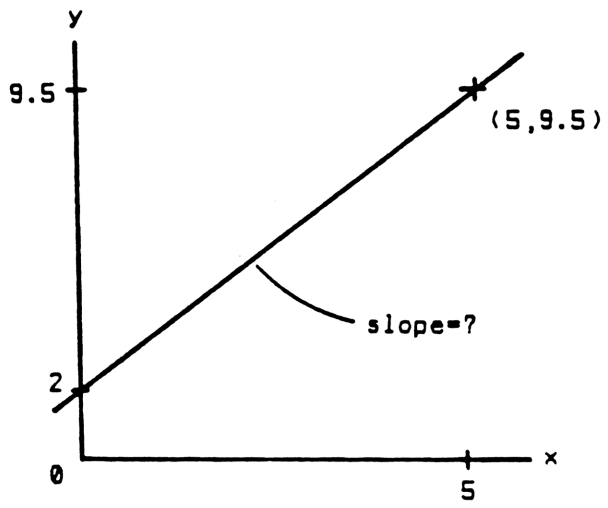
The **SOLVIT** program must be in HP-41 main memory.

For information regarding inserting HP-41 modules and loading the **SOLVIT** program, refer to your HP-41 Owner's Manual. Program barcode for **SOLVIT** is printed in the **PROGRAM BARCODE** section. Throughout the rest of these instructions, it is assumed that you have the **SOLVIT** program in main memory and that the Advantage Module is inserted.

Example 1

SOLVIT works with equations that you program in the same RPN format that the HP-41 uses for keyboard calculations. Once an equation has been entered, you specify the name of each variable in the equation, then, given values for all variables but one, **SOLVIT** is ready to solve for the unknown variable. To see how this works, try the following example. Don't worry about the details of each step; they will be explained later. For this example to work properly, the HP-41's top row keys must not have any User key assignments. If necessary, clear any User key assignments before proceeding. Refer to your HP-41 Owner's Manual for details.

The Problem: Find the slope of the straight line in the figure below. The line intercepts the y-axis at y = 2, and one point on the line is located at x = 5, y = 9.5.



The Solution: The straight line equation is

y = slope * x + intercept

There are four variables in this equation, which we'll call **Y**, **SLOPE**, **X**, and **INTER**.

This equation is entered into HP-41 memory as a program called **LINE** with the following procedure; the first two steps assure that you have enough memory for the example.

Keystrokes	Display
[XEQ] [ALPHA] SIZE [ALPHA] 010	X register +
[ ] [GTO] [.] [.] [PRGM]	00 REG nnn +
[ ] [LBL] [ALPHA] LINE [ALPHA]	01 LBL LINE
[RCL] 02	02 RCL 02
[RCL] 03	03 RCL 03
[*]	04 *
[RCL] 04	05 RCL 04
[+]	06 +
[RCL] 01	07 RCL 01
[-]	08 -
[PRGM] [ ] [GTO] [.] [.]	X register

+ If the HP-41 displays **PACKING**, then **TRY AGAIN**, clear one or more programs and repeat this step.

+ If nnn < 016, clear one or more programs and repeat this step.

SOLVIT is initialized to solve the straight line equation in the following manner:

Keystrokes	Display
[ ] [FIX] 2	X register
[XEQ] [ALPHA] SOLVIT [ALPHA]	EQUATION?
LINE [R/S]	NO. OF VARS=?
4 [R/S]	NAME, VAR 1?
Y [R/S]	NAME, VAR 2?
SLOPE [R/S]	NAME, VAR 3?
X [R/S]	NAME, VAR 4?
INTER [R/S]	LINE Y S X I

The last display is the menu for the straight line equation, which labels the HP-41's top row keys. When labelled in this manner, the top row keys are called menu keys. Note that the menu contains only the first character of each variable name. In general, a menu can contain the first one or two characters of a variable name.

The example can now be solved by first providing the known values for **Y**, **X**, and **INTER**, and then solving for **SLOPE**.

Keystrokes	Display
9.5 | Y | +	Y S X I
5 | X |	Y S X I
2 | I |	Y S X I
| S |	*WORKING*
	SLOPE=1.50
	Y S X I

+ | m | is menu key m, the top row key under the character(s) m. If necessary, press [R/S] before | m | to see which key to press.

The solution to this example is a slope of 1.5.

Of course, this simple example could probably be solved more quickly by just rearranging the equation to the form

slope = (y - intercept)/x

However, while some useful equations can be manipulated this easily, many cannot. But no matter how complicated the equation, **SOLVIT** can always provide the friendly user interface illustrated above.

DETAILED INSTRUCTIONS FOR SOLVIT

Rewriting an Equation in the SOLVIT Format

Equations are usually written in algebraic form (also known as infix form, because operators are placed between their operands). **SOLVIT** requires equations to be programmed in an RPN form (also known as postfix form, because operators come after their operands). The conversion from an equation in algebraic form to **SOLVIT**'s RPN form is very straightforward (examples below are excerpted from **Example 1**):

1. Write the equation in algebraic form.
   (Example: y = slope * x + intercept)

2. Rearrange the equation
   from the form

   left side = right side

   to the form

   right side - left side = 0
   or
   left side - right side = 0

   (Example: slope * x + intercept - y = 0)

3. Disregard the "= 0" portion of the equation and rewrite the left side of the equation in RPN form.
   (Example: slope x * intercept + y -)

4. Replace each variable with an HP-41 **RCL nn** statement, where nn = 01, 02, 03, ..., up to the number of variables in the equation. Variables that appear more than once in an equation are assigned the same **RCL nn** statement.
   (Example: **RCL 02 RCL 03 * RCL 04 + RCL 01 -**)

The nn in each **RCL nn** statement must be remembered to initialize **SOLVIT** later. Each nn determines the position in the display where the corresponding variable name will appear as a label. In this example, **Y** will be first, followed by **SLOPE**, **X**, and **INTER**. Although the order of register assignment can be arbitrary, it seems reasonable (as done here) to assign registers to variables as they appear from left to right in the algebraic form of the equation.

Programming an Equation

Once written in the proper format as described above, an equation is entered into program memory just like any other program. Each **RCL nn** statement, operator, or special instruction becomes one

program step. More specifically, a good approach to programming an equation is as follows:

1. Press [ ] [GTO] [.] [.]. The HP-41 will display **PACKING**, then the X register. This step packs main memory to make room for the equation and positions the program pointer to an area of program memory where the equation can be entered.

2. Press [PRGM] to enter Program mode. The HP-41 will display the **PRGM** annunciator.

3. Enter a name for the equation in the form of a global label statement. Press [ ] [LBL] [ALPHA] name [ALPHA], where name is the name to be associated with the equation (up to six characters). The name should be meaningful for the equation.

4. Enter each program step exactly as written in the **SOLVIT** format described above. If there are not enough program registers, the HP-41 will display **PACKING**, then **TRY AGAIN**, as it attempts to free more registers. If you still can't enter program steps, reduce the SIZE or clear key assignments or alarms, then try again to continue. If this fails, clear one or more programs from main memory, then go back to the last line entered and continue.

5. Press [PRGM] to exit Program mode.

6. Press [ ] [GTO] [.] [.] to pack program memory and to add an **END** statement to the equation program. Your equation program will remain in main memory until explicitly cleared with the **CLP** command or a similar operation.

Following this approach for the straight line example will produce the **LINE** program described in **Example 1**.

All of the HP-41's functions, including tests and branching, can be used to customize an equation. In addition, **SOLVIT** provides a special optional feature, described in **Equation Program Techniques**, that may be useful in some equation programs.

Initializing SOLVIT

Before **SOLVIT** can work with an equation, it must know the equation's name and the name of each variable. The process of providing these items is called initialization, accomplished as follows (examples below are excerpted from **Example 1**):

1. Start the **SOLVIT** program. Press [XEQ] [ALPHA] **SOLVIT** [ALPHA] or press a key in User mode to which **SOLVIT** has been assigned.

2. At the prompt **EQUATION?**, key in the name (that is, the global label) of the equation program in main memory to be solved and press **[R/S]**. Do not press **[ALPHA]** before or after keying in the equation name. Note: A previously-initialized equation can be restarted by just pressing **[R/S]** at the **EQUATION?** prompt. Refer to **Restarting SOLVIT**.
(Example: **LINE [R/S]**)

3. At the prompt **NO. OF VARS=?**, key in the number of variables in the equation and press **[R/S]**.
(Example: **4 [R/S]**)

SOLVIT requires a minimum SIZE of **2 * (NO. OF VARS) + 2**. If the current SIZE is too low, the HP-41 will prompt **SIZE>=**nnn, where nnn is the minimum SIZE needed. To increase the SIZE, press **[XEQ] [ALPHA] SIZE [ALPHA]** nnn, then press **[R/S]** to continue. If the HP-41 displays **PACKING**, then **TRY AGAIN**, clear key assignments or alarms, then try to increase the SIZE again. If this doesn't work, clear one or more programs from main memory, then try again to initialize **SOLVIT**, starting at step 1.

4. For each variable in the equation, the HP-41 will prompt **NAME, VAR** nn**?**, where nn = 1, 2, 3, ..., up to **NO. OF VARS**. For each prompt, key in the full name (up to six characters) of the variable corresponding to the **RCL** nn statement in the equation program and press **[R/S]**. Do not press **[ALPHA]** before or after keying in each variable's name.
(Example: **Y [R/S] SLOPE [R/S] X [R/S] INTER [R/S]**)

5. After all variable names have been keyed in, the HP-41 will display the equation name as it checks main memory for the 13 free registers required by the **SOLVE** function. If there are not enough free registers, the HP-41 will display **NO ROOM**. If possible, reduce the SIZE (but not below the minimum specified in step 3), clear key assignments or alarms, then press **[R/S]** to continue. If this fails, clear one or more programs from main memory, then restart **SOLVIT** as described in **Restarting SOLVIT**.

6. When **SOLVIT** has been completely initialized, the first menu (described below) will be displayed. The initialization process need not be repeated each time **SOLVIT** is used; the data will remain intact until the data registers used to store parameters for **SOLVIT** are altered.

SOLVIT is now ready to solve your equation using the techniques described next.

Basic Use of SOLVIT

Each menu used by **SOLVIT** consists of the first one or two characters of a variable's name positioned over one of the HP-41's five top row keys. When labelled this way, the top row

keys are called menu keys, because their use changes with the menu above. Throughout these instructions, | m | means menu key m, the top row key under the character(s) m. If the equation has more than five variables, the menu key to the right of the last variable in the menu is labelled with | ↑↑ |, indicating that pressing it will cause the next menu, containing more variable names, to be displayed. When a menu shows the last set of variables, pressing | ↑↑ | causes the menu with the first set of variables to be displayed.

For **SOLVIT** to operate properly, the top row keys (and all other keys you press to use **SOLVIT**) must not have any User key assignments. If necessary, use **ASN** to clear assignments from the necessary keys. Refer to your HP-41 Owner's Manual for details.

While using **SOLVIT**, you can exit User mode temporarily to access the top row key functions. You must enter User mode again before attempting to use a **SOLVIT** menu.

Solving an equation for any unknown variable with **SOLVIT** is basically a two-step process (examples below are excerpted from **Example 1**):

1. Assign all known values to their corresponding variables by keying in the value and pressing the variable's menu key. If necessary, press | ↑↑ | one or more times first to display the appropriate menu. To restore the menu after keying in a number (if you need to see which menu key to press), press **[R/S]**; the appropriate menu key can then be pressed to assign the value.
   (Example: **9.5 | Y | 5 | X | 2 | I |**)

2. Press the menu key of the variable for which a solution is desired (the "unknown", or solve variable). The HP-41 will display ***WORKING*** as it solves for the variable. If found, the solution will be displayed for approximately four seconds, then the menu will be restored. If no solution is found, the HP-41 will display **NO SOLUTION**.
   (Example: **| S |**)

Instruction Table for SOLVIT

SOLVIT has many features that expand the basic capability described in **Basic Use of SOLVIT**. Most of the features are summarized in the **Instruction Table** below, and all are described in detail afterwards. Remember that while **SOLVIT** is busy (that is, while the **PRGM** annunciator is on), you should not press any keys unless you intend to interrupt the program.

Instruction Table for SOLVIT

Instructions	Keystrokes	Display
Set display format	[ ] [FIX] n or [ ] [SCI] n or [ ] [ENG] n	X register
Set all variables to value	value [XEQ] [J]	menu
Clear (set to zero) all variables	[XEQ] [J]	menu
Restore current menu	[R/S] or	menu
(ALPHA or program changed)	[J]	menu
Display next menu	↓ ↑↑ ↓	next menu
Assign value to variable m	value ↓ m ↓	menu
View full name and value of variable m	[ ] ↓ m ↓	varnam=value menu
Recall value of variable m	[XEQ] ↓ m ↓ or	value
(five or fewer variables)	[RCL] ↓ m ↓	value
View/Recall value of last solve variable	[<-]	value
Solve for variable m	↓ m ↓	*WORKING* varnam=value menu or
(no solution found)		*WORKING* NO SOLUTION varnam=NO SOL menu

Menu Format

SOLVIT works with variable names of up to six characters. To create a menu of up to five variables, each variable name is abbreviated to its first one or two characters, depending on the variable's position in the display. **SOLVIT** builds and displays a menu in the format

aa b cc d ee

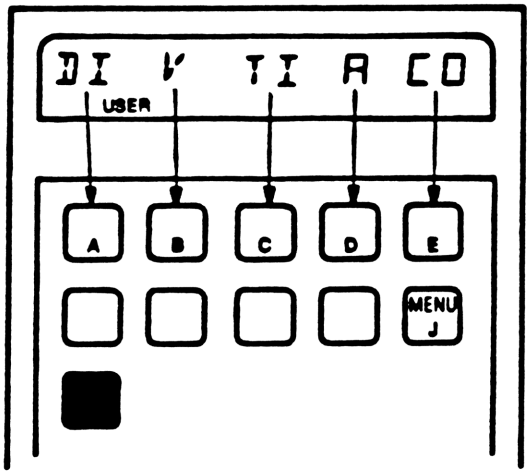
where

aa, cc, and ee = the first two characters of the first, third, and fifth variables in the menu, and

b and d = the first characters of the second and fourth variables in the menu.

Each variable is represented in the menu by the first one or two characters in its name. This display restriction is necessary because of the HP-41's 12-character display.

As an example, the figure below shows the HP-41 display and top row keys corresponding to the menu for an equation with the variables **DISTAN**, **VELOC**, **TIME**, **ACCEL**, and **COLOR**.



If the equation has more than five variables, the menu key to the right of the last variable in the menu is defined as | ↑↑ |, indicating that pressing it will cause the next menu, containing more variables, to be displayed.

Any menu keys not currently assigned to variables are left blank in the menu display.

Setting the Display Format

SOLVIT displays all results using the current display format (for example, **FIX 2**). You can change the display format any time

a menu is displayed. To restore the menu after changing the display format, press **[R/S]**.

Setting or Clearing all Variables

All variables of an equation can be set simultaneously to any one value by keying in the desired value and pressing **[XEQ] [J]**. Alternatively, all variables can be cleared (set to zero) by just pressing **[XEQ] [J]**.

SOLVIT distinguishes between the two options by noting whether or not you key in a number before pressing **[XEQ] [J]**.

Restoring the Menu

The current menu can be restored after keying in a number (or any time the display has been cleared) by pressing **[R/S]**. This does not affect subsequent operation of **SOLVIT**.

Occasionally (for example, when the ALPHA register has been changed or the program pointer has been repositioned), pressing **[R/S]** will not restore the menu. When this happens, press **[J]** to recreate the menu and then display it. If this fails, refer to **Restarting SOLVIT**.

Displaying the Next Menu

When an equation contains six or more variables, **SOLVIT** can't display a menu containing them all. Instead, **SOLVIT** creates multiple menus, each displaying up to four menu key labels and the label | ↑↑ |. The order in which labels are displayed in the menu structure corresponds to the order in which variable names are keyed in when **SOLVIT** is initialized. Pressing menu key | ↑↑ | causes the next menu, containing the next set of menu key labels, to be be displayed. At the last menu, pressing | ↑↑ | causes the first set of labels to be displayed. All unlabelled (blank) menu keys operate like | ↑↑ |.

Assigning Values to Variables

To assign a value to a variable, key in the desired value and press the menu key corresponding to that variable. Alternatively, you can calculate the value for a variable (for example, **45 [SIN]**) and then press its menu key. However, **SOLVIT** won't recognize that you want to assign the value unless a number has been keyed in. If necessary, perform some operation like **[1] [*]** before pressing the menu key.

Recalling Values of Variables

To display the full name of a variable and its value, press **[ ]**

and then the variable's menu key. The HP-41 will display the variable's full name (varnam) and current value (value), according to the display format, in the format

varnam=value

After approximately four seconds, the menu will be restored.

To recall to the X register the value of a variable, press **[XEQ]** and then the variable's menu key. If there are five or fewer variables in the equation, a quicker alternative is to press **[RCL]** and then the variable's menu key.

To view the current value of the variable last solved for (which is always in the X register when a menu is first displayed), press **[<-]**. If no variable has been solved for yet, the X register will contain the equation control value (see **Data Register Usage**).

Solving for a Variable

To solve the equation for a specific variable (the solve variable), press the variable's menu key without first keying in a number. The HP-41 will display ***WORKING*** for up to several minutes as it solves the equation. If a solution is found, the solve variable's full name (varnam) and the solution value (value) will be displayed, according to the display format, in the format

varnam=value

After approximately four seconds, the menu will be restored. The solution value is automatically assigned to the solve variable.

Optionally, an initial estimate for the solution can be supplied by assigning the desired initial estimate to the solve variable before solving (see **Assigning Values to Variables**). **SOLVIT** always uses the current value of the solve variable as the initial estimate of its solution, with one exception: if the current value is ALPHA data, **SOLVIT** uses zero as the initial estimate.

The solution **SOLVIT** calculates is one of perhaps several. If you suspect there may be other solutions, you can assign a new initial estimate and solve for the desired variable again.

When No Solution Is Found

For some complicated equations or bad initial estimates, **SOLVIT** may not be able to find a solution to the equation. When this happens, the HP-41 will display **NO SOLUTION.** After approximately one second, the HP-41 will indicate that no solution was found for variable varnam by displaying **varnam=NO SOL.** After

approximately four seconds, the menu will be restored. The ALPHA data value "NO SOL" will be assigned to solve variable varnam.

It may be possible to find a solution to the equation by choosing another initial estimate. Refer to **Solving for a Variable.**

Interrupting SOLVIT

During the time that **SOLVIT** is attempting to solve for a variable (while ***WORKING*** is displayed), **SOLVIT** can be stopped by pressing **[R/S]**. Because **SOLVIT** repeatedly executes the equation program, it is possible that the program pointer many not be in **SOLVIT** when **[R/S]** is pressed. Therefore, restoring the menu after interrupting **SOLVIT** can be a one- or two-step operation. First, press **[J]** to attempt to restore the menu. If the menu is not restored, refer to **Restarting SOLVIT**, described next.

Restarting SOLVIT

If the SOLVIT program has been exited temporarily to do other calculations, you may resume work with the same equation without reinitializing **SOLVIT**, but only if the data registers used to store parameters for **SOLVIT** have not been altered. To restart **SOLVIT**, start **SOLVIT** as instructed in **Initializing SOLVIT**, but instead of keying in the equation name at the **EQUATION?** prompt, just press **[R/S]**. **SOLVIT** will continue at step 5 of the initialization process.

EXAMPLES

Example 1

Refer to the **INTRODUCTION** and **DETAILED INSTRUCTIONS FOR SOLVIT**, where this example is examined in detail.

Example 2

The Problem: If a ball is thrown from a balcony 10 meters above the ground with an upward velocity of 50 meters per second, how long will it take the ball to hit the ground?

The Solution: The equation describing the vertical motion of the ball is

$$x = v0 * t - 0.5 * g * t^2 + x0$$

where

x = the height of the ball,
v0 = the initial velocity of the ball,
t = the time the ball spends traveling in each direction,
g = the acceleration due to gravity (9.8 m/s^2),
and x0 = the initial height of the ball.

Since g is a constant, this equation really has only four variables.

♦ Programming:

Enter the equation program **MOTION**, below. Note that the expression **0.5 * g * t^2** has been simplified to **4.9 * t^2**.

```
01♦LBL "MOTION"
02 RCL 01
03 RCL 02
04 *
05 4.9
06 RCL 02
07 X↑2
08 *
09 -
10 RCL 03
11 +
12 RCL 04
13 -
14 END
```

In the equation program, register numbers for **v0** and **x0** have been chosen to insure both characters in their names will appear in the menu. This is a justifiable transgression from the order guideline suggested in **Rewriting an Equation in the SOLVIT Format**.

• Initialization:

Keystrokes	Display
[ ] [FIX] 2	X register
[XEQ] [ALPHA] SOLVIT [ALPHA]	EQUATION?
MOTION [R/S]	NO. OF VARS=?
4 [R/S]	NAME, VAR 1?
V0 [R/S]	NAME, VAR 2?
T [R/S]	NAME, VAR 3?
X0 [R/S]	NAME, VAR 4?
X [R/S]	MOTION
	V0 T X0 X

• Solving:

A solution for T is desired when the ball reaches the ground (X=0). The initial velocity and height are given in the problem statement.

To illustrate a point, the variables in the equation will be cleared by [XEQ] [J] before assigning values and solving for T. After clearing all variables, it isn't necessary to assign a value to X, since zero is the desired value.

Keystrokes	Display
[XEQ] [J]	V0 T X0 X
50 | V0 |	V0 T X0 X
10 | X0 |	V0 T X0 X
| T |	*WORKING*
	T=-0.20
	V0 T X0 X

This obviously isn't the value for T expected. However, the "error" is explained easily. Pressing [XEQ] [J] cleared all variables. Therefore, when T was solved for, its initial estimate was zero, which caused SOLVIT to find a solution, near zero, that corresponds to the time it takes for ball to go from the ground to the balcony with an initial velocity greater than 50 m/s. The positive solution for T can be found by choosing a better initial estimate:

Keystrokes	Display
10 | T |	V0 T X0 X
| T |	*WORKING*
	T=10.40
	V0 T X0 X

The ball remains in the air for 10.4 seconds.

TECHNICAL DETAILS

How SOLVIT Solves It

The computation power of **SOLVIT** is derived from the HP-41 Advantage Module function **SOLVE**. Like **SOLVIT**, **SOLVE** repeatedly executes a program until it finds a value that causes the program to return zero, thereby solving for a variable in the equation. But **SOLVE** is limited because it can only solve an equation containing one variable; every other number within the equation program must be a constant.

SOLVIT extends the power of **SOLVE** in two major ways:

- **SOLVIT** solves an equation program containing any number of variables.

- **SOLVIT** provides a very simple method for inputting data and getting results.

SOLVIT leverages the power of **SOLVE** by having **SOLVE** always solve the same special program, called **SVRTN**. **SVRTN** takes the current estimate passed by each iteration of **SOLVE** and stores it in the data register corresponding to the solve variable. Then **SVRTN** passes execution to the equation program, which really treats all other variables as constants, since their data registers aren't changed. However, since **SVRTN** updates the solve variable's data register with each iteration, **SOLVE** eventually finds a value in that data register that solves the equation -- and therefore is the value of the solve variable.

For a comprehensive description of the algorithm used by **SOLVE** to calculate a solution, refer to the HP-15C Advanced Functions Handbook (Hewlett-Packard, 1982).

Equation Program Techniques

- **SOLVIT** sets user flag 05 before executing the equation program the first time. Since the equation program is usually executed many times (while ***WORKING*** is displayed) before a solution is found, this feature makes possible special procedures within an equation program to handle unique circumstances that may arise only occasionally. To use this option in an equation program, follow these guidelines:

  - Include a flag 05 test to check if the special condition is in effect.

  - If the special condition is in effect, perform any necessary steps based on the condition.

- When the special condition has been resolved, clear flag 05 to speed subsequent equation program execution.

For example, suppose you would like the **LINE** equation program described in **Example 1** to insure that **SLOPE** input is never zero; instead, a default slope of 1 is desired. The **LINE** program could be changed to check the slope every time it's executed, replacing zero with 1 as necessary, but this method has two problems:

- In most cases, you will assign a non-zero slope. Therefore, the **LINE** program will usually be wasting time checking if the slope is zero.

- If you are solving for **SLOPE**, the equation program will continually thwart **SOLVIT**'s efforts to attempt a solution of zero slope.

By adapting the **LINE** program to check for a zero slope only once, when the equation program is first executed, both these problems are eliminated. By this method, the beginning of the **LINE** program can be changed to the following:

```
01♦LBL "LINE"
02 FC?C 05
03 GTO 00
04 RCL 02
05 X=0?
06 1
07 STO 02
08♦LBL 00
```

- If you would like to observe how **SOLVIT** converges on a solution for an equation, insert a **VIEW X** instruction at the beginning of the equation program. This will cause the HP-41 to display the current estimate of the solve variable with each execution of the equation program.

- Almost any HP-41, module, or peripheral function can be included in an equation program. The exceptions are the functions **AK**, **PASN**, **PSIZE**, all card reader functions, and any other functions that alter the HP-41's SIZE, key assignments, or alarms.

Flag Usage

SOLVIT uses two user flags and two system flags during its operation, as described in the following table.

SOLVIT Flag Usage

Flag	Use
05	Used during menu formatting. Set before solving for a variable. May be left set by **SOLVIT**.
06	Set if **NO. OF VARS** is greater than five; cleared otherwise.
21	Cleared to display (but not print) ***WORKING***, then matched to flag 55 (printer present).
27	Set to enter User mode.

Data Register Usage

For an equation with *nn* variables, **SOLVIT** uses data registers **00** through **(2 * *nn*) + 1**. Thus, **SOLVIT** requires a SIZE of at least **(2 * *nn*) + 2**. **SOLVIT** uses data registers as described in the table below.

SOLVIT Data Register Usage

Data Register	Use
00	equation control value *iii.nnnsss* +
01	value of variable 1
02	value of variable 2
...	...
nn-1	value of variable *nn*-1
nn	value of variable *nn*
nn+1	name of variable 1
nn+2	name of variable 2
...	...
2**nn*-1	name of variable *nn*-1
2**nn*	name of variable *nn*
2**nn*+1	name of equation program

\+ *iii* = index to current menu number (1, 2, ...),
nnn = number of variables in equation, and
sss = register number of last solve variable, or zero.

Stack Register Usage

Many of **SOLVIT**'s features manipulate values in the stack. The following table describes the state of the stack after each action indicated.

SOLVIT Stack Register Usage

Keystrokes	Stack Registers After + X	Y	Z	T	L
value **[XEQ] [J]** to set all variables to *value*	X	-	Y	-	-
[XEQ] [J] to clear variables	-	-	Y	-	-
[R/S] to restore menu	X	Y	Z	T	L
[J] to restore menu	-	-	-	-	-
↓ ↑↑ ↓ to display next menu	-	-	-	-	-
↓ m ↓ to assign value to variable	X	-	-	-	-
[ ] ↓ m ↓ to view name and value of variable	-	-	-	-	-
[XEQ] ↓ m ↓ to recall value (*v*) of variable	*v*	X	-	-	-
[RCL] ↓ m ↓ to recall value (*v*) of variable	*v*	X	Y	Z	L
[<-] to view/recall value of last solve variable	X	Y	Z	T	L
↓ m ↓ to solve for variable	-	-	-	-	-

+ The stack register name entries in the table indicate where the contents of the each register are located *after* the operation. "-" indicates the contents are not meaningful.

APPENDIX: STORING AND RECALLING EQUATIONS

When using **SOLVIT** with a number of different equations, the advantage of saving the equation program and its menu structure as one unit becomes apparent. Programs and functions already exist to conveniently store programs and data registers in extended memory or on mass storage (Beers and Beers, 1985), but not to link the equation program with its menu structure (a new concept motivated by the **SOLVIT** menu interface).

The programs explained below provide the cabability to store and recall, within the **SOLVIT** environment, equation programs and their menu structure. Each program works by accessing two similarly-named files in extended memory or mass storage. To use these programs, all you need in addition to the basic **SOLVIT** essentials is the appropriate module (either Extended Memory or HP-IL) to access the storage medium. (**RCLMS** requires both modules.) Program barcode for these programs is provided in the **PROGRAM BARCODE** section.

STOXM and STOMS (STOre equation, eXtended Memory/Mass Storage)

STOXM and **STOMS** can be executed any time a **SOLVIT** menu is displayed (or outside of **SOLVIT**, if no data registers have been altered). **STOXM** and **STOMS** are non-prompting programs. When either is executed, the HP-41 will display ***WORKING*** as **STOXM** or **STOMS** takes data directly from the data registers and creates two files in extended memory or mass storage:

- A data file, containing the equation menu structure, with the name **?name**, where name is the name of the equation program.

- A program file, containing the equation program, with the same name as the equation program.

If either file already exists, it is replaced.

When **STOXM** or **STOMS** is completed, **SOLVIT** is restarted automatically at step 5 of the initialization process.

RCLXM and RCLMS (ReCaLl equation, eXtended Memory/Mass Storage)

RCLXM and **RCLMS** can be executed when a **SOLVIT** menu is displayed or outside of **SOLVIT** in order to initialize and start **SOLVIT**. At the prompt, **EQUATION?**, key in the name of the equation you wish to recall and press **[R/S]**. If you can't remember the name of the desired equation, just press **[R/S]**, and the HP-41 will display a directory of extended memory or mass storage, then prompt again for the equation name. If the directory listing is interrupted by **[R/S]**, the HP-41 will prompt **R/S**, indicating that you should

press **[R/S]** to prompt again for the equation name. The HP-41 will display ***WORKING*** as **RCLXM** or **RCLMS** recalls information from extended memory or mass storage in the same two files stored by **STOXM** or **STOMS**. If a program by the same name as the equation program already exists in main memory, it is not replaced.

When **RCLXM** or **RCLMS** is completed, **SOLVIT** is started or restarted automatically at step 5 of the initialization process, ready to solve the selected equation.

REFERENCES

Beers, Thomas W., and Ted W. Beers. 1985. OS-41, an Operating System for the HP-41 Handheld Computer and Peripherals. T.W. Beers, West Lafayette, IN. 472 p.

Hewlett-Packard. 1982. HP-15C Owner's Handbook. Hewlett-Packard, Corvallis, OR. 293 p.

Hewlett-Packard. 1982. HP-15C Advanced Functions Handbook. Hewlett-Packard, Corvallis, OR. 221 p.

Hewlett-Packard. 1985. HP-41 Advantage; Advanced Solutions Pac. Hewlett-Packard, Corvallis, OR. 115 p.

HP-41 Program Listing: SOLVIT, SVRTN

```
 01♦LBL "SOLVIT"          Start of SOLVIT.
CF 23 "EQUATION?" AON     Skip EQUATION? prompt if executed from
FC? 50 PROMPT AOFF          another program.
FC?C 23 GTO 06 ASTO L     Prompt for equation program name.
"NO. OF VARS=?" PROMPT    Skip NO. OF VARS=? prompt if EQUATION?
LASTX RCL Y 2 * 2 +         prompt not answered.
XROM "SZ?" FC?C 25        Prompt for number of variables in
PROMPT R↑ X<>Y              equation.
STO IND Y 1.4 R↑ +        Check for minimum SIZE.
LASTX 1 E3 / ISG X
STO 00 AON                Calculate and store equation control
                            value.
 34♦LBL 00
"NAME, VAR " AIP "├?"     Prompt for full name of each variable in
PROMPT ASTO IND Y           equation.
ISG Y ISG X GTO 00
AOFF

 44♦LBL 06
CF 06 CF 22 SF 27 5       Initialize user and system flags.
RCL 00 FRC 1 E3 *
INT X>Y? SF 06 2 *        Check for 13 free registers required by
1 + VIEW IND X              SOLVE.  SIRTN is a no-op program in
"SIRTN" CLST SOLVE          the HP-41 Advantage Module.

 64♦LBL J                 [J]: Initialize values to build menu.
3 FC? 06 4 RCL 00
INT LASTX FRC 1 E3 *
INT + + LASTX RCL 00
FRC 1 E3 * INT 2 *
RCL Z X<Y? X<>Y RDN
1 E3 / + CLA ASTO Y
ASTO Z CF 05

 96♦LBL 07                Build menu.
"****" FS? 05 "├*"        Extract first one or two characters of
ARCL IND X "├ " ASTO L      variable name.
"*" FC? 05 "├*"
ARCL L ASHF ASTO L
CLA ARCL Y ARCL Z
ARCL L "├ " ASTO Y
ASHF ASTO Z FC?C 05
SF 05 ISG X GTO 07
CLA ARCL Y ARCL Z
RCL 00 1 E3 * FRC
1 E3 * RCL IND X          Add ¦ ↑↑ ¦ if there are more than five
FS? 06 "├↑↑"                variables.

133♦LBL 15                Display menu.
PROMPT GTO 15
```

```
136♦LBL 10
RCL 00  1 E3  *  INT
1 E3  /  X<>Y  FC?C 22
CLX
```

[XEQ] [J]: Initialize values to set or clear all variables.

```
146♦LBL 08
STO IND Y  ISG Y  GTO 08
GTO 15
```

Set or clear all variables.

```
151♦LBL A
0  GTO 09
```

```
154♦LBL B
1  GTO 09
```

```
157♦LBL C
2  GTO 09
```

| m |: Initialize values to set or solve for variable m, or display next menu.

```
160♦LBL D
3  GTO 09
```

```
163♦LBL E
FS? 06  GTO 10  4
```

```
167♦LBL 09
RCL 00  INT  +  RCL 00
FRC  1 E3  *  INT  X<Y?
GTO 10  FC?C 22  GTO 11
RCL Z  STO IND Z  GTO 15
```

Determine whether to display next menu (GTO 10), solve (GTO 11), or set a variable.

```
183♦LBL 10
FC? 06  GTO 15  4
ST+ 00  RCL 00  INT
LASTX  FRC  1 E3  *  INT
X<>Y  X<=Y?  GTO J
RCL 00  FRC  1  +
STO 00  GTO J
```

Initialize values to display next menu.

```
204♦LBL 11
CF 21  " *WORKING*"
AVIEW  FS? 55  SF 21
SF 05  RCL IND Y  RCL Z
1 E3  /  RCL 00  1 E3  *
INT  +  1 E3  /  STO 00
RDN  SIGN  X≠0?  LASTX
ENTER↑  "SVRTN"  SOLVE
GTO 12  "NO SOLUTION"
AVIEW  ASTO X
```

Solve for a variable.

Update equation control value to contain solve variable register number.

Solve SVRTN, which executes equation program.

```
234♦LBL 12
RCL 00  FRC  1 E3  *
INT  LASTX  FRC  1 E3  *
+  CLA  ARCL IND X  "⊢="
ARCL Y  AVIEW  LASTX
RCL Z  STO IND Y  GTO J
```

Display solution.

```
254♦LBL a
0  GTO 13

257♦LBL b
1  GTO 13

260♦LBL c                    [] | m |: Initialize values to display
2  GTO 13                      full name and value of variable m.

263♦LBL d
3  GTO 13

266♦LBL e
FS? 06  GTO 15  4

270♦LBL 13                   Determine whether to ignore illegal
RCL 00  INT  LASTX  FRC        command (GTO 15).
1 E3  *  INT  X<> Z  +
STO Z  X<>Y  +  LASTX  2
*  X<Y?  GTO 15  CLA
ARCL IND Y  "┣="             Display full name and value.
ARCL IND Z  AVIEW  GTO J

294♦LBL 01
0  GTO 14

297♦LBL 02
1  GTO 14

300♦LBL 03                   [XEQ] | m |:  Initialize values to
2  GTO 14                      recall variable m.

303♦LBL 04
3  GTO 14

306♦LBL 05
FS? 06  GTO 15  4

310♦LBL 14                   Determine whether to ignore illegal
RCL 00  INT  +  RCL 00         command (GTO 15).
FRC  1 E3  *  INT  X<Y?
GTO 15  RDN  RDN             Recall value.
RCL IND T  GTO 15

325♦LBL "SVRTN"              Program SVRTN, which executes equation
RCL 00  FRC  1 E3  *           program.
INT  LASTX  FRC  1 E3  *
X<>Y  2  *  1  +             Store current estimate in solve variable
RCL IND X  X<> Z  R↑           register.
STO IND Y  GTO IND T         Execute equation program.
END
```

HP-41 Program Listing: STOXM, STOMS, RCLXM, RCLMS

```
 01◆LBL "STOXM"          Start of STOXM.
SF 06  GTO 00

 04◆LBL "STOMS"          Start of STOMS.
CF 06

 06◆LBL 00
XEQ 05  2  RCL 00  FRC
1 E3  *  INT  *  ISG X   Build data file name ?name.
AOFF  "?"  ARCL IND X
SF 25  FS? 06  PURFL     Purge existing data file.
FC? 06  PURGE  FC?C 25
XEQ 05  "?"  ARCL IND X
LASTX  2  +  FS? 06
CRFLD  FC? 06  CREATE  0 Create data file.
FS? 06  SAVERX  FC? 06   Store equation control value.
SEEKR  FC? 06  WRTRX
RDN  DSE X  X<>Y  1 E3   Store  variable  and  equation  program
/  +  FS? 06  SAVERX       names.
FC? 06  WRTRX  ATOX
FS? 06  SAVEP  FC? 06    Store equation program.
WRTP  FC?C 06  VERIFY
GTO 15                   Restart SOLVIT.

 60◆LBL "RCLXM"          Start of RCLXM.
SF 06  GTO 01

 63◆LBL "RCLMS"          Start of RCLMS.
CF 06

 65◆LBL 01               Prompt for equation program name.
CF 23  "EQUATION?"  AON
PROMPT  AOFF  FS?C 23    Display extended memory or mass  storage
GTO 02  "R/S"  ASTO X      directory  if  EQUATION?  prompt  not
FS? 06  EMDIR  FC? 06      answered.
DIR  GTO 01
```

ANNOTATED PROGRAM LISTINGS

```
 80◆LBL 02
ASTO X  XEQ 05  "?"
ARCL X  CLX  FS? 06
SEEKPTA  FC? 06  SEEKR
FS? 06  GETRX  FC? 06
READRX  ATOX  2  RCL 00
FRC  1 E3  *  INT  *
LASTX  1  +  X<>Y  2  +
ASTO Z  XROM "SZ?"
FC?C 25  PROMPT  FC? 50
XEQ 05  R↑  1 E3  /  +
FS? 06  GETRX  FC? 06
READRX  CLA  ARCL Y  11
SF 25  PASN  FS?C 25
GTO 03  XEQ 05  CLA
ARCL Z  FS? 06  GETSUB
FC? 06  READSUB

136◆LBL 03
CF 06  CLA  PASN

140◆LBL 15
GTO "SOLVIT"

142◆LBL 05
" *WORKING*"  CF 21
AVIEW  FS? 55  SF 21
END
```

Build data file name ?name.

Recall equation control value.

Check for minimum SIZE.

Recall variable and equation program names.

Check for existing equation program.

Recall equation program, if necessary.

Clear key assignment made by check for existing equation program.

Restart SOLVIT.

Display, but not print, *WORKING*.

SOLVIT
PROGRAM

ROW 1: LINES 1-3

ROW 2: LINES 3-6

ROW 3: LINES 7-11

ROW 4: LINES 11-15

ROW 5: LINES 16-25

ROW 6: LINES 25-33

ROW 7: LINES 34-36

ROW 8: LINES 36-42

ROW 9: LINES 43-51

ROW 10: LINES 51-61

ROW 11: LINES 61-66

ROW 12: LINES 67-77

SOLVIT
PROGRAM

ROW 13: LINES 78-87

ROW 14: LINES 88-95

ROW 15: LINES 96-100

ROW 16: LINES 101-106

ROW 17: LINES 106-113

ROW 18: LINES 113-120

ROW 19: LINES 120-128

ROW 20: LINES 128-133

ROW 21: LINES 134-141

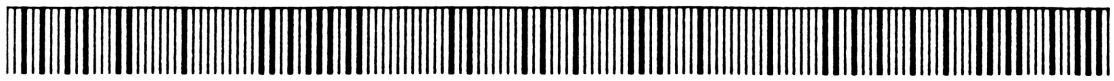
ROW 22: LINES 141-149

ROW 23: LINES 150-156

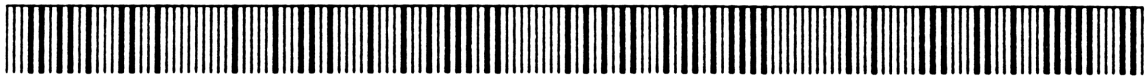
ROW 24: LINES 157-164

SOLVIT
PROGRAM

ROW 25: LINES 164-173



ROW 26: LINES 174-181



ROW 27: LINES 182-188

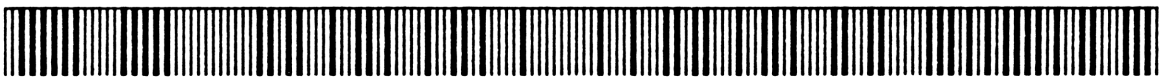
ROW 28: LINES 189-197

ROW 29: LINES 198-206

ROW 30: LINES 206-209

ROW 31: LINES 209-216

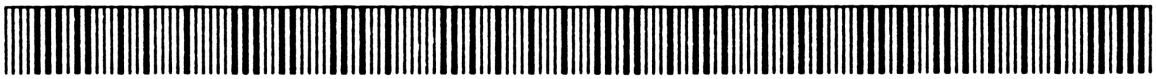
ROW 32: LINES 216-225



ROW 33: LINES 226-231

ROW 34: LINES 231-233

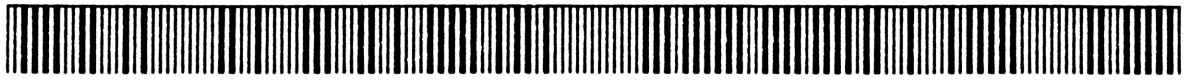
ROW 35: LINES 233-242



ROW 36: LINES 242-250

SOLVIT
PROGRAM

ROW 37: LINES 251-257

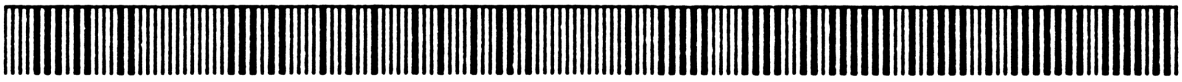


ROW 38: LINES 257-265

ROW 39: LINES 265-273

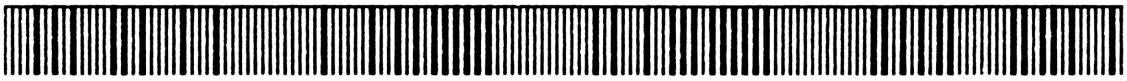
ROW 40: LINES 274-282

ROW 41: LINES 283-290



ROW 42: LINES 291-299

ROW 43: LINES 299-308



ROW 44: LINES 308-317

ROW 45: LINES 318-325

ROW 46: LINES 325-328

ROW 47: LINES 329-339

ROW 48: LINES 340-345

STOXM
PROGRAM

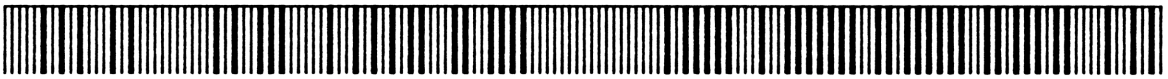
ROW 1: LINES 1-3

ROW 2: LINES 4-7

ROW 3: LINES 7-15

ROW 4: LINES 16-22

ROW 5: LINES 23-29



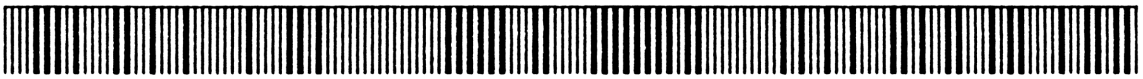
ROW 6: LINES 30-37

ROW 7: LINES 37-44

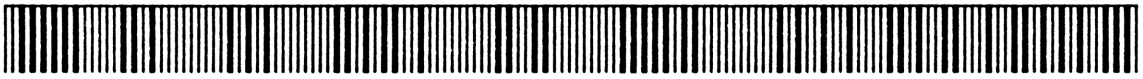
ROW 8: LINES 45-51

ROW 9: LINES 52-58

ROW 10: LINES 58-60



ROW 11: LINES 61-63

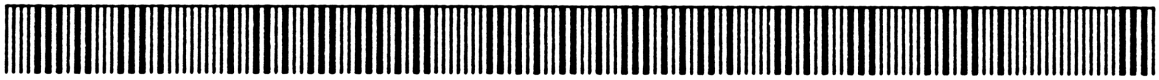


ROW 12: LINES 64-67

STOXM
PROGRAM

ROW 13: LINES 67-73

ROW 14: LINES 74-80



ROW 15: LINES 81-87

ROW 16: LINES 87-93

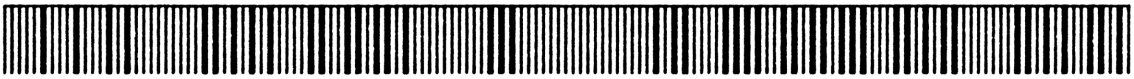
ROW 17: LINES 94-103

ROW 18: LINES 104-112

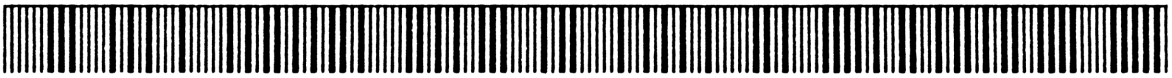
ROW 19: LINES 113-119

ROW 20: LINES 120-126

ROW 21: LINES 127-133



ROW 22: LINES 133-140



ROW 23: LINES 141-143

ROW 24: LINES 143-147

STOXM
PROGRAM

ROW 25: LINES 147-148