SOLVESYS 4.0

Simultaneous Nonlinear Equations and Least Squares Solver for the HP48G series

HTML documentation v. 1.07 (august, 1999)

DISCLAIMERS AND LEGAL ISSUES

VERSION 4.0 RELEASE NOTES

The present library is a completely rewritten version of SOLVESYS. Since most of the interface has changed, I suggest you read the entire manual even if you're familiar with previous versions of the program.

SOLVESYS 4.0 includes support for nonlinear least squares problems (overdetermined systems). The major application of this is datafitting, but SOLVESYS is not limited to this case. Underdetermined systems are now treated as an error to avoid "false" solutions (e.g. due to an unassigned constant)

Two main features known from v3.1 and earlier releases have been removed; analytic derivatives (just wasn't worth it compared to the speed of numeric derivatives) and ranges

for variables (too much overhead for little gain). Also there's no "complex number" switch in v4.0, but complex valued results are allowed if complex numbers are also given as start guesses.

Two new error tolerances are available, one for the least squares case (cosine test) and a tolerance for the convergence of the iterate values. The latter applies to both zero finding and least squares.

SOLVESYS 4.0 does not include a stack version of the solver. If you need this feature, you must stick to version 3.1.

SOLVESYS 4.01 (released January 11, 1998) fixes a bug in the v4.0 library: Using the solver after "Too many unknowns" error will in certain situations generate random errors or even cause a memory clear! More precisely, the bug occurs if trying to solve a system with *at least* one constant defined but still more unknowns than equations. This correctly produces a "Too many unknowns" error but when trying to solve the system again with altered variable definitions, SOLVESYS returns some error or what is worse. Thanks to Carlos Hartwich for reporting the bug.

Furthermore a change to the convergence tests has been implemented. This should fix some potential convergence issues for linear least-squares. This change causes the LSQ value in the INFO box not to be updated/computed until *XTOL* has been satisfied.

SOLVESYS 4.02 (released August 12, 1998) corrects a few problems related to linear equations solving. The solver now terminates with a "Zero" message if an equation set evaluates *exactly* to zero regardless of the *XTOL* tolerance. This prevents the solver from stalling if the equations are solved (with no error) in the first iteration without the user having to change the *XTOL* setting manually.

SOLVESYS 4.03 (released August 30, 1999)

- Bug in the linesearch algorithm fixed. The bug causes duplicate stack objects and incorrect damping factor computations if/when cubic interpolation is used (** displayed in the status line). The bug is not visible to the user because the extra stack objects (one real number per cubic interpolation) are removed when SOLVESYS is closed. The damping factor is restricted to be within certain limits, so the incorrect computation still gives a useful (but not optimal) result.
- Bug in equation editor fixed. This bug is quite severe and is caused by [CLEAR] not updating the highlight index correctly. Fortunately, the bug is rather unlikely to occur in practice. The following sequence will reproduce the bug: 1) In the equation editor, create (at least) two equations, 2) highlight the last equation, 3) [CLEAR] all equations, 4) [ADD] a new equation, 5) Now, note that this equation is not highlighted. Subsequent operations such as [COPY] or [DEL] will cause an "insufficient memory" error or worse.
- To save memory, selection of error tolerances is now done using choose boxes. This offers a range from 1E-1 to 1E-7 for the three tolerances XTOL, EQTOL and LSQTOL. A few other minor cosmetic changes to the interface have been made.

An HP49G release is planned if/when HP releases updated entry points.

INSTALLATION

To install SOLVESYS on your HP48G/GX:

1. Transfer SOLVESYS.LIB to the HP48G/GX.
2. Recall the library to the stack and press `n [STO]`, where `n` is the desired RAM port (0 if you have a HP48G). After storing the library, you can purge the original copy to save memory.
3. Turn the HP48 off, then on. This adds SOLVESYS to your library menu.

The checksum for SOLVESYS.LIB (version 4.03 in the ABOUT box) is # 8F42h, size 3807 bytes.

To check that SOLVESYS has been correctly installed, press [right-shift]+[2] to enter the library menu, choose the `[SOLVESYS]` menukey and you should see the `[SOLVESYS]` `[FITEQ][ABOUT]` labels.

To remove SOLVESYS.LIB:

1. Put the library ID (*n*`: 1550`) in the stack
2. Execute the sequence: `[DUP][DETACH][PURGE]`

If the HP48 gives an "Object in use" error at this point, press `[ON]+[C]` and repeat the above steps.

To transfer SOLVESYS.LIB to another HP48G/GX:

1. Put the library ID (`n: 1550`) in the stack and press `[RCL]`.
2. Store the returned library in any variable
3. Transfer this variable to the other HP48 - use binary transfer.

WHAT IS SOLVESYS?

SOLVESYS is graphical environment for solving systems of linear and nonlinear equations and (nonlinear) least squares problems. The library works exclusively on the HP48G series and is less than 4 kb in size.

SOLVESYS is designed to either zero or least-squares minimize sets of *m* linear or nonlinear equations in *n* unknowns.When *m=n*, the problem is obviously to find a solution satisfying all equations. If *m>n*, the equations are generally inconsistent so instead SOLVESYS searches for a least squares solution, i.e. the minimizer of the sum of squared equation deviations. Least squares minimization is commonly used in datafitting applications. The case *n>m* is subject to error.

Nonlinear equations must be solved iteratively, i.e. the user provides an initial guess which the solver (hopefully) can use to find the true solution. It can be necessary to try out different starting guesses before the solver succeeds in this quest, but if qualified starting values are provided, equations can often be solved with only a few iterations.

SOLVESYS does not allow units in equations or variables but references to the CONST() function are possible. Furthermore SOLVESYS cannot be used to solve systems with discrete functions or variables.

The solver uses a (Gauss-) Newton method with a mixed quadratic and cubic linesearch procedure. For more details on this algorithm, see Dennis and Schnabel [1] or Press et al. [2].

[1] Dennis, J.E., and Schnabel, R.B. 1983, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Englewood Cliffs, NJ: Prentice-Hall).
[2] Press, W.H. et al. 1992, *Numerical Recipes in C: The Art of Scientific Computing*. 2nd ed. (Cambridge: University Press).

EXAMPLES

EXAMPLE: SOLVING A SYSTEM OF NONLINEAR EQUATIONS

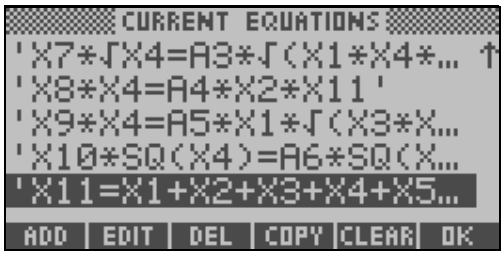
The following system (propane combustion?) is a mixture of 11 linear and nonlinear equations in 11 unknowns:

$x_1 + x_4 = 3$
$2x_1 + x_2 + x_4 + x_7 + x_8 + x_9 + 2x_{10} = 10 + r$
$x_2 + 2x_5 + x_6 + x_7 = 8$
$2x_3 + x_5 = 4r$
$x_1\, x_5 = a_1\, x_2\, x_4$
$x_6\, x_2^{½} = a_2\, (x_2\, x_4\, x_{11})^{½}$
$x_7\, x_4^{½} = a_3\, (x_1\, x_4\, x_{11})^{½}$
$x_8\, x_4 = a_4\, x_2\, x_{11}$
$x_9\, x_4 = a_5\, x_1\, (x_3\, x_{11})^{½}$
$x_{10}\, x_4^2 = a_6\, x_4^2\, x_{11}$
$x_{11} = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}$

$a_1 = 0.193$
$a_2 = 0.002597$
$a_3 = 0.003448$
$a_4 = 0.00001799$
$a_5 = 0.0002155$
$a_6 = 0.00003846$
$r = 4.056734$

There are two ways to input equations into SOLVESYS. Either store the equation list in the 'EQ' variable prior to starting SOLVESYS or enter the equations manually in the SOLVESYS equation editor (see below).

Whichever way is used, the equation editor should eventually look something like this (last 5 equations shown):



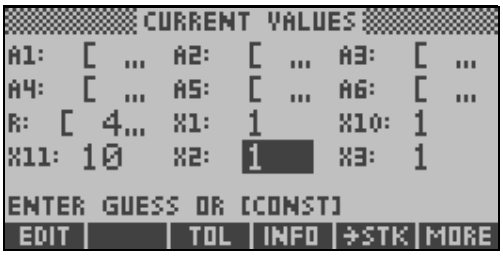
When [OK]is pressed, SOLVESYS analyzes the equations for variables and displays the "Current Values" dialog box shown below. This should be filled out according to the following rules;

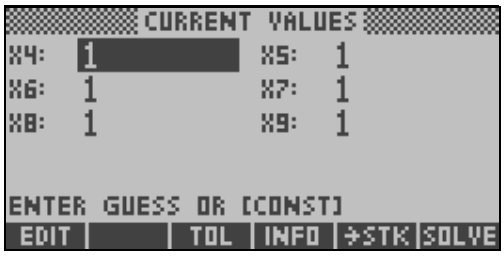
- For each unknown, enter a real or complex valued starting guess. This must be your best guess of the solution you're looking for - don't use the default values if you know

any better.
- Constants (if any) are assigned by placing the value(s) in square brackets, like `[ 0.28 ]`. The value will then be substituted into the equations before the solver begins (this does not affect the 'EQ' variable).

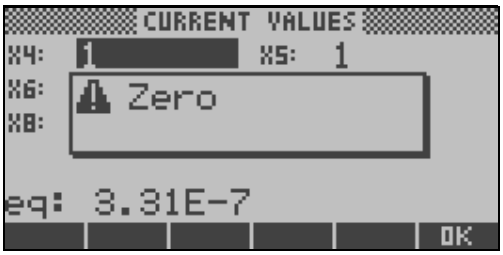
Following these rules, all the constansts (a_1,...,a_6 and r) are entered in square brackets. For the unknowns the default guesses are used except x_{11} we set to 10 (since x_{11} is the sum of the other variables).

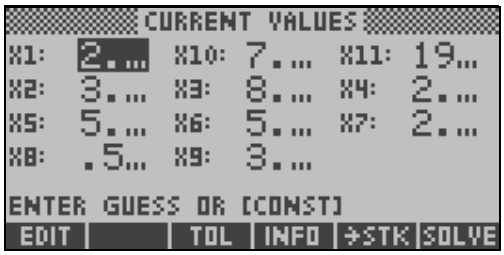
The display should then look like this. Press `[MORE]` to show the remaining unknowns:



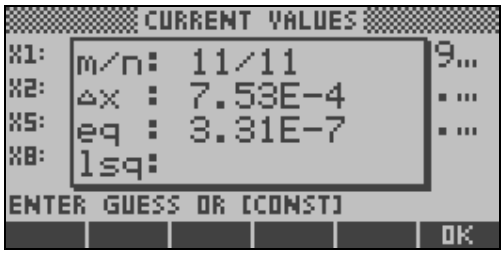


On the second screen, press `[SOLVE]` and after some iterations a solution (Zero) is found;





`[->STK]` can be used to copy the result to the stack. `[INFO]` gives some extra information;



"m/n" is ratio between equations and variables, "dX" is the relative difference of the last two computed results (relates to the accuracy of the current solution) and "eq" is the root-mean-square error of the equations at the current solution (this is the same value as shown during iteration). See the Error Tolerances section for more details.

To quit SOLVESYS, first press `[ON]` to return to the equation editor and press `[ON]` again to exit.

EXAMPLE: NONLINEAR LEAST SQUARES MINIMIZATION

If there are more equations than unknowns (inconsistent equations), SOLVESYS will search for a "best fit" solution.

A common application of this is datafitting where one is attempting to fit *m* observations to an expression of *n* unknown parameters for which the "best-fit" values are required. If each observation is inserted in the expression, the result is a system of *m* "observational" equations in *n* unknowns. The library utility `FITEQ` can be used to create this equation set.

An example follows.

The relationship between the measured pressure and temperature in saturated steam can be written as

$$Y=a\text{ALOG}[b\text{T}/(c+\text{T})]+e_Y$$

where Y is the measured pressure of the steam at various values of a controllable temperature T
a, *b* and *c* are unknown parameters (to be estimated)
e_Y is a normally distributed measurement error (of Y) with mean zero and variance v.

The following data were collected.

T:	0	10	20	30	40	50	60	70	80	85	90	95	100	105
Y:	4.14	8.52	16.31	32.18	64.62	98.76	151.13	224.74	341.35	423.36	522.78	674.32	782.04	920.01

Half the job is to obtain some guesses of the parameters to solve for. In this example, the T=0 value in the first measurement causes *a*=4.14 in the equation so we can use this as a starting value. Furthermore by inserting *a*=4.14 in the next two "observational" equations we have;

$$8.52=4.14*\text{ALOG}[10b/(c+10)]$$
$$16.31=4.14*\text{ALOG}[20b/(c+20)]$$

Taking LOG on both sides of both equations and multiplying with (*b*+10) and (*c*+20) respectively gives;

$$(c+10)*\text{LOG}(8.52)=\text{LOG}(4.14)*(c+10)+10b$$
$$(c+20)*\text{LOG}(16.31)=\text{LOG}(4.14)*(c+20)+20b$$

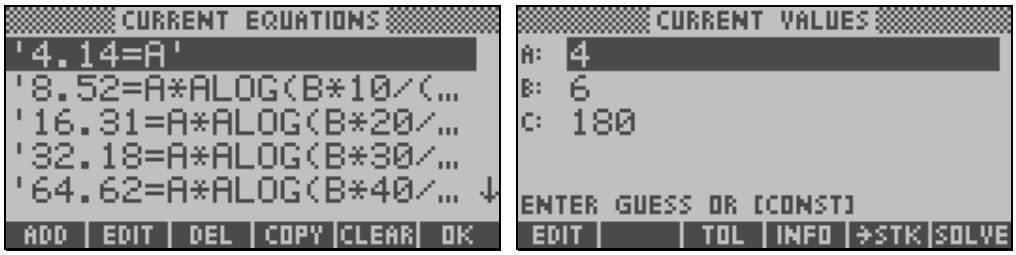
which are just two *linear* equations with the solution (*b*,*c*)=(5.93,179). Thus an initial guess such as (*a*,*b*,*c*)=(4,6,180) seems reasonable.

The utility `FITEQ` is used to create the 14 equations needed. First store the data as vectors (or lists) in global variables such as `'T'` and `'Y'` in the current directory. Then apply `FITEQ` with the arguments;

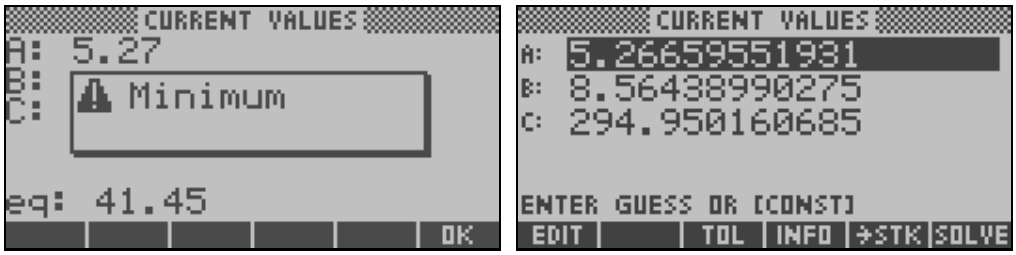
```
3: 'Y(k)=A*ALOG(B*T(k)/(C+T(k)))'
2: 'k'
1: 14
```

where the level 2 variable is the data index used in the level 3 expression and level 1 is the number of observations. Make sure the variables `{ A B C }` do not exist in the current or parent directories.

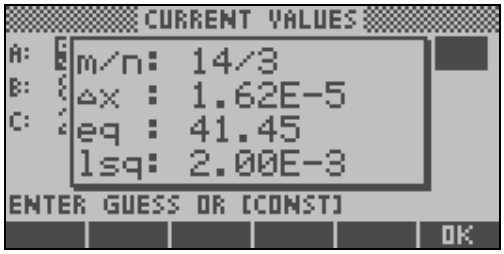
After applying `FITEQ`, start SOLVESYS and the equation set editor appears. The equations are already created so press `[OK]`and fill in the initial guesses (a,b,c)=(4,6,180).



Press [SOLVE] and after a few iterations, the solver has found a minimizer;



Using 3 significant digits, the best fit function is Y(T)=5.27*ALOG[8.56*T/(295+T)]. Press [INFO] to get some more details about this minimum:



Since the "eq" value is the rms error (see Error Tolerances below), the error sum of squares (*SSE*) is 41.45^2=1718. The estimated variance *v* of the measurement error e_Y is *SSE*/(*m*-*n*)=1718/11=156.2.

Be aware that there are no simple ways to distinguish between a local or global minimum. This means that the choice of starting values is very crucial. With poor values, the solver could just as well converge towards any of the local minima listed below (and there are even more!):

(*a*,*b*,*c*)=(4.14,-168,1), eq=1603
(*a*,*b*,*c*)=(4.14,infinity,-infinity), eq=1603
(*a*,*b*,*c*)=(305,infinity,0), eq=1127
(*a*,*b*,*c*)=(0,infinity,infinity), eq=1083

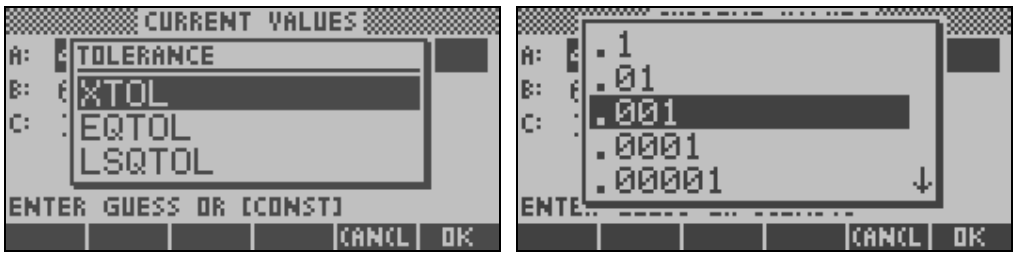
A good initial guess is not only a real time-saver, it will also pick out the right (global) minimum for you!

Note: Weighted least-squares

Often it is desired to use different weights for observations used in datafitting, eg. when the observations have unequal variance. To compute a weighted fit, divide the equations (eg. the EQ list produced by FITEQ) with a list of standard-deviations (square root of variance) of the corresponding observations and use SOLVESYS on the new list.

ERROR TOLERANCES

It is possible to modify the error tolerances used for convergence testing although it is not recommended. Press `[TOL]` to access the error tolerance choose boxes.



There are three tolerances available and values from 1E-1 to 1E-7 are possible for all three (default value always highlighted). Again it is NOT recommended to change these values unless there is specific reason to do so.

(The actual results of the tests below are displayed when pressing the `[INFO]` menukey).

XTOL

Tolerance for overall convergence. This is the *relative difference* between the last two computed results. A value of 10^{-p} usually corresponds to p significant digits of the computed result, but values below 10^{-6} should not be trusted to comply with this. The default value is 10^{-3} corresponding to 3 significant digits.

XTOL is the "primary" test in the sence that *EQTOL* or *LSQTOL* are not tested if *XTOL* fails. However, for several reasons, *XTOL* is ignored if the equation values are *exactly* zero (this forces an "Zero" message).

EQTOL

Tolerance to check if the equations have been zeroed (should be named *ZEROTOL*?). The test computes the rms (root mean square) error of the equation values. For example, for the system { xy=9, x+y=6} the rms error at (x,y)=(2.9, 3.1) is $[(8,99-9)^2+(6-6)^2]^{½}= 0.01$. *EQTOL* defaults to 10^{-5} which should be sufficient for most purposes.

In case the rms error is *exactly* zero, the solver assumes that a zero has been found and terminates with a "Zero" message (possibly ignoring the *XTOL* tolerance above).

LSQTOL

This tolerance is used only if $m>n$ ie. for nonlinear least-squares problems. It is a cosine test from Dennis & Schnabel (sort of scaled gradient test). The test returns values between 0 and 1 where a value close to 0 indicates a least squares minimizer. The test seems to be rather strict and will usually be close to 1 even near a minimum so I suggest using 0.1>*LSQTOL*>0.001 and not lower. The default value is 0.01.

LSQTOL does not apply to zero-residual least squares, however the *EQTOL* test above will return a "Zero" message in this case.

TERMINATION AND ERROR MESSAGES

Some (more or less) common error and termination messages returned by SOLVESYS are listed below.

RETURN MESSAGES FROM THE SOLVER:

Zero:
: A result satisfying *XTOL* and *EQTOL* was found (or only *EQTOL* if an exact zero was encounted).

Minimum:
: A result satisfying *XTOL* and *LSQTOL* was found. It is not guaranteed to be a global minimum.

Bad Guess(es):
: The same result was computed twice but did not pass the tolerance tests (see Error Tolerances) . Typically this indicates that the solver has landed on a local minimizer of the sum of squared equation deviations. Also, if the solver is started at a point where the Jacobian matrix is singular, this error is likely to occur.

Non-Real Result:
: A complex valued result occured although no complex valued start guesses were provided (indicating a complex result to be valid). This is caused by a variable moving into a region where some equation or its derivative do not evaluate to a real number. Either, better initial values must be provided, or in simpler cases the equations may be manipulated to avoid complex domain functions and/or derivatives. Alternatively, try using complex valued guesses with zero imaginary parts. Be aware that the HP48 power function (`^`) returns complex results for all negative arguments raised to a non-integer power. For example, the expression `'(-1)^(1/3)'` evaluates to a complex result instead of -1. THis can be fixed by replacing `'X^(a/b)'` with `'XROOT(b,X^a)'`.

Undefined Result:
: (Math exception.) Calculation such as 0/0 occurred or the `XROOT` function was used with arguments that cannot give a real-valued result. Check the current solution or try with other starting guesses.

Infinite Result:
: (Math exception.) Calculation such as 1/0 occurred. Check the current solution or try with other starting guesses.

INITIALIZATION ERRORS:

All Variables Known:
: All variables were defined as constants (i.e. placed in square brackets).

Too Many Unknowns:
: The number of unknowns is greater than the number of equations. Although the solver in principle can handle this situation too, a solution to such a system is unlikely to be of any interest.

Constant?:
: One or more equations do not contain any unknowns. SOLVESYS does not allow redundant equations.

Inconsistent Units:
: Units are not supported.

Undefined Name:
: Some expression(s) include(s) one or more variables SOLVESYS is unable to identify.

For example, variables that only appear inside the HP48 summation function (sigma) are not detected. If this rare error occurs, adding an dummy term like X-X to any equation should fix the problem.

FITEQ ERRORS:

Invalid User Function:
A referenced data variable (such as `'T'` in the example above) does not contain a list or vector of data values.

Bad Argument Value:
The number of observations argument passed to `FITEQ` is greater than the number of observations present in at least one of the data vectors or lists.

HOW TO REACH THE AUTHOR

Feel free to contact me if you have any comments, questions, problems or suggestions. Also if you find any bugs (or what you think is a bug), please notify me. My email is sune_bredahl@hotmail.com, ICQ # 43726969.

The latest version of SOLVESYS and other related information are available from: http://ww.student.dtu.dk/~c947086/hp48.html.