

## HP Forum Archive 15

[ [Return to Index](#) | [Top of Index](#) ]

### Even More Simultaneous Equation Solutions

Message #1 Posted by [Palmer O. Hanson, Jr.](#) on 25 Mar 2006, 3:42 a.m.

In his submission "Calculator Precision: Sure" of December 21 which started all of this Valentin Albillo proposed an investigation of the set of linear equations

$$1.3 x_1 + 7.2 x_2 + 5.7 x_3 + 9.4 x_4 + 9.0 x_5 + 9.2 x_6 + 3.5 x_7 = 45.3$$

$$4.0 x_1 + 9.3 x_2 + 9.0 x_3 + 9.9 x_4 + 0.1 x_5 + 9.5 x_6 + 6.6 x_7 = 48.4$$

$$4.8 x_1 + 9.1 x_2 + 7.1 x_3 + 4.8 x_4 + 9.3 x_5 + 3.2 x_6 + 6.7 x_7 = 45.0$$

$$0.7 x_1 + 9.3 x_2 + 2.9 x_3 + 0.2 x_4 + 2.4 x_5 + 2.4 x_6 + 0.7 x_7 = 18.6$$

$$4.1 x_1 + 8.4 x_2 + 4.4 x_3 + 4.0 x_4 + 8.2 x_5 + 2.7 x_6 + 4.9 x_7 = 36.7$$

$$0.3 x_1 + 7.2 x_2 + 0.6 x_3 + 3.3 x_4 + 9.7 x_5 + 3.4 x_6 + 0.4 x_7 = 24.9$$

$$4.3 x_1 + 8.2 x_2 + 6.6 x_3 + 4.3 x_4 + 8.3 x_5 + 2.9 x_6 + 6.1 x_7 = 40.7$$

which has the "unique" solution :

$$x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = x_7 = 1.0$$

Valentin closed with the challenge

"Now, get your preferred HP calc or computer software and try and solve it using limited accuracy, say just one decimal, then four decimals, then eight decimals. See what results you get and how do they compare with the actual, unique solution, and among themselves as the limited accuracy increases."

I solved the problem with a variety of linear equation solvers in my inventory with mantissas ranging from six to fourteen digits.

"Unique" answers as used here are the same as the "exact" answers which can be obtained with machines such as the HP-49G and TI-89. A "unique" or "exact" answer the one that a "calculated" answer from a sufficiently capable solver will approach but, in general, will not equal. Solvers sometimes get the "unique" solution for the kind of innocuous demonstration problems that typically accompany the user instructions. The "calculated" answer from a capable solver for more difficult problems such as the ones considered here will have characteristics such as (a) the signs of the elements of the "calculated" vector will agree with the signs of the corresponding elements of the "unique" vector, (b) the magnitudes of the elements of the "calculated" vector will agree with the magnitudes of the corresponding elements of the "unique" vector, and (c) at least some of the most significant digits of the elements of the "calculated" vector will agree with the corresponding most significant digits of the elements of the "unique" vector. An example of a solution which does not meet those criteria is the solution from the HP-28S for Valentin's original problem where the elements of the solution vector are a helter-skelter collection of numbers where three of the signs are not even in agreement with the signs of the elements of the "unique" solution vector. A second example which does not meet those criteria is the solution to Valentin's original problem from the ML-02 program in the TI-59 Master Library module where one of the signs is not in agreement and several of the magnitudes are in error by a factor of about 1.6 from the "unique" solution of all ones. An example that meets those criteria is the solution to Valentin's original problem from software by Maurice Swinnen operating on the Model 100 where the element of the solution which is furthest from the "unique" answer of one is 1.00339... . We can assign a single number to summarize those characteristics by calculating the RMS of the relative errors of the calculated vector where the relative error, calculated as the difference between the corresponding elements of the "unique" and "calculated" vectors divided by the corresponding element of the "unique" vector should be a small fraction.

I recently ran Valentin's original problem on the HP-49G (actually on an emulator) with the following results:

INV A \* B

B/A

REF

-1.076	-0.6472	1.00148007239
0.999786	1.00110019	0.999999813411
0.407	-0.7415	1.00131966086
0.144	0.6554	1.00132720057
0.999124	0.9996398	1.00000091517
0.553	1.1592	0.998672819845
0.61	3.5385	0.997223232158

where for the RREF solution the input is a 7x8 matrix with A augmented by B. One sign is incorrect for the INV A \* B solution. Two of the signs are incorrect for the B/A solution. The RREF solution was much closer to the "unique" solution where I note that for the first time in all the calculations that I have reported in the series of threads since Valentin's original challenge a solver has performed better than my old Model 100 using the simultaneous equation program by Swinnen and Thomas..

In subsequent threads Rodger Rosenbaum proposed three modifications to Valentin's problem: (1) with B(1) changed from 45.3 to 45.4; (2) with B(3) changed from 45.0 to 45.1; and (3) with A(3,7) changed from 6.7 to 6.8, A(7,7) changed from 6.1 to 6.0 and B(3) changed from 45.0 to 45.1. "Unique" solutions for Rodger's modified problems were found using the exact mode of the HP-49G.

The RMS relative errors for Valentin's original problem and for Rodger's three modifications for seventeen solvers which are available in my inventory are

Solver	Valentin	Rodger 1	Rodger 2	Rodger 3
Sharp PC-1261	18.386	9.823E-01	9.673E-01	1.250E-07
Model 100 Swinnen SP	11.506	1.014	9.996E-01	2.832E-04
HP-28S B/A	3.927	13.417	13.220	1.253E-10
HP-41 "seq"	2.371	1.019	1.004	2.445E-07
HP-49G B/A	1.327	1.065E-03	1.080E-03	5.477E-13
HP-49G INV A * B	9.060E-0	1.094E-03	1.080E-03	1.573E-10
Sharp PC-1261 Swinnen	7.664E-01	1.117	1.101	2.647E-09
TI-59 ML-02	7.367E-01	2.464E-02	2.428E-02	9.433E-11
TI-95 Math Module	3.147E-01	2.955E-03	2.906E-03	3.297E-11
HP-41 Math Pac	3.018E-01	1.027	1.008	2.005E-08
CC-40 Math Module	2.260E-01	1.025E-01	1.010E-01	3.202E-10
Model 100 HCN DP	1.214E-01	2.215E-03	2.182E-03	2.321E-11
TI-85 INV B * A	3.582E-02	9.270E-03	9.155E-03	8.683E-12
CC-40 Swinnen	5.125E-03	8.466E-03	8.342E-03	2.701E-12
TI-85 Simul Equations	4.274E-03	9.292E-03	9.155E-03	4.399E-12
Model 100 Swinnen DP	1.799E-03	3.625E-04	3.572E-04	9.217E-13
HP-49G RREF	1.472E-03	8.281E-05	8.352E-05	0

where the solvers are listed in the order of decreasing RMS relative errors for Valentin's original problem. Obviously missing in a forum dedicated to HP products are solutions from the HP-15C and the Advantage module of the HP-41. I don't have those capabilities in my inventory. Perhaps someone else will provide the missing material.

The RMS relative errors for all solvers are substantially smaller for Rodger's third modification. That modification seems to have dumbed down Valentin's original problem so that almost any solver can succeed. For example, solvers such as the the B/A solution with the HP-28S and the ML-02 routine with the TI-59 which couldn't even get all the signs correct with Valentin's original problem get rather good results with Rodger's third modification. Even the Model 100 solver operating in the single precision mode with a six digit mantissa gets four digits correct for Rodger's third modification.

On machines such as the TI-85 and HP-49G the results indicate that row reduction type software will outperform matrix inversion software. That observation agrees with my experience with linear equation solutions to other problems. I do not claim to know whether the apparently inferior performance is inherent in the matrix inversion methodology or whether matrix inversion software just tends to be less carefully written.

#### NOTES ON THE HARDWARE AND SOFTWARE OF THE SOLVERS

1. B/A software uses built-in matrix routines to divide the vector by the matrix. That idea is discussed on page 66 of the *HP-28S Advanced Scientific Calculator Reference Manual* and is expected to yield somewhat better results than multiplying the inverse of the matrix by the vector. I do not know if the idea applies for other machines.

2. Swinnen software uses a solution adapted from the Simultaneous Equation routine on pages 113 ff of the *Mathematics Library: Application Software for the Sharp EL-5500 and PC-1403 Scientific Computers* by Swinnen and Thomas.
3. The Model 100 Swinnen DP solver uses the Swinnen routine and the default mode of the Radio Shack Model 100 which provides a fourteen digit mantissa.
4. The Model 100 Swinnen SP uses the Swinnen program and single precision mode of the Radio Shack Model 100 which provides a six digit mantissa.
5. The CC-40 Swinnen solution uses the Swinnen program instead of the CC-40 Math Module solver.
6. The Model 100 HCN DP solution uses an older program derived from a least squares solver which was on the old Honeywell Computer Network (HCN) of the 1960's. It does not allow zeroes in the diagonal elements.
- 7.. The Sharp PC-1261 solution uses the program on page 264 of the *Sharp Pocket Computer PC-1260/PC-1261 Instruction Manual*.. The program will not work with  $A(1,1) = 0$  and with certain other combinations of the matrix elements which yield a zero divisor..
8. The Sharp PC-1261 Swinnen solution uses the Swinnen program.
9. The HP-41 "seq" solution uses Valentin Albillo's program from pages 63-64 of the V7N5 (June 1980) issue of the *PPC Calculator Journal*. The program shares the same deficiency as the Sharp PC-1261 solution when  $A(1,1) = 0$  and when other combinations of matrix elements yield a zero divisor. Valentin's program detects the zero divisor and allows a rearrangement of the problem without starting completely over.

#### A MESSY HP-49G PROGRAM

In his submission "More on ill conditioning and calculator precision" on January 23 Rodger proposed "... let's create a 7x7 perturbation matrix consisting of 49 uniformly distributed random numbers varying from about -.5 ULP to about +.5 ULP, and then add that perturbation matrix to the A (call the perturbed matrix A') matrix of Valentin's original system, and solve the perturbed system...". He explained

"The procedure can be done on the HP49 emulator (or a real HP48G or HP49) easily. Assume that the original A matrix and the original column matrix B are stored in variables of those names. Then this program will generate a perturbation matrix, add it to the A matrix, and solve the system.

```
<< B A RANM 247.487 / A + / -3 RND >>
```

This can be executed repeatedly to see results such as I gave above.

The constant 247.487 is to cause the perturbation matrix to have a Frobenius norm of about .1414, which is the same as the norm of perturbation matrix used to create the slightly perturbed matrix I suggested and you used in your previous post; the maximum perturbation of an individual element of the A matrix will be about +- .0404, just slightly less than .5 ULP (.5 ULP would be .05)."

When I found that the RREF solver on the HP-49G consistently yielded "calculated" solutions which were closer to the "unique" solutions than the B/A solver I decided to see if I could modify Rodger's program to replace the B/A solution with the RREF solution. My plan was to follow his program to generate the perturbed matrix, augment the perturbed matrix with the B vector and solve with the RREF routine. I had difficulty doing that because I couldn't get the AUGMENT command to attach the B vector at the right side of the perturbed matrix yielding the 7x8 matrix needed by the RREF routine. No matter what I tried the AUGMENT command would attach the B vector at the top or bottom of the perturbed matrix rather than at the right of the perturbed matrix. Eventually, I transposed the perturbed matrix, used the AUGMENT command to attach the B vector at the bottom yielding an 8x7 matrix, and transposed the 8x7 matrix back to yield the 7x8 matrix I wanted. My complete routine was.

```
<< B A RANM 247.487 / A + TRAN B AUGMENT TRN RREF ->COL DROP -3 RND >>
```

There must be an easier way to do what I wanted to do but I couldn't find it. Maybe if I had a manual .... Maybe someone more familiar with the HP-49G can help.

### Re: Even More Simultaneous Equation Solutions

Message #2 Posted by [John Smitherman](#) on 25 Mar 2006, 1:09 p.m.,  
in response to message #1 by Palmer O. Hanson, Jr.

Hi Palmer. How did you define/calculate the RMS?

Thanks,

John

### Re: Even More Simultaneous Equation Solutions

Message #3 Posted by [Palmer O. Hanson, Jr.](#) on 25 Mar 2006, 8:36 p.m.,  
in response to message #2 by John Smitherman

I calculate the relative error for each of the seven elements of the solution vector and then find the RMS of the seven relative errors.

### Re: Even More Simultaneous Equation Solutions

Message #4 Posted by [Rodger Rosenbaum](#) on 25 Mar 2006, 10:02 p.m.,  
in response to message #1 by Palmer O. Hanson, Jr.

Palmer says:

I recently ran Valentin's original problem on the HP-49G (actually on an emulator) with the following results:

INV A * B	B/A	RREF
-1.076	-0.6472	1.00148007239
0.999786	1.00110019	0.999999813411
0.407	-0.7415	1.00131966086
0.144	0.6554	1.00132720057
0.999124	0.9996398	1.00000091517
0.553	1.1592	0.998672819845
0.61	3.5385	0.997223232158

I just discovered something that astounds me in the process of doing your examples. I don't get what you get when I do these tests on my HP49. It hasn't been used for some years, and perhaps the version of the firmware in it has known bugs. My HP49 has version 1.05 firmware--what do you get when you type **VERSION** in your emulator, Palmer?

And, if anybody else has an HP49 with up to date firmware, could you try this and report your results?

Anyway, what I get is:

INV A * B	B/A	RREF
1.00018	1.00018	1.00124
1.9043	1.9043	-33231081.9872
1.3779	1.3779	4704.81157655
1.00065	1.00065	4678.11919981
0.7689	0.7689	6770866.25288
0.553	0.553	-46565.78990985
-0.785	-0.785	-2226.81759925

I have done a little checking, with some anomalous results. The HP49 gives  $-1.00000533622E-7$  for the determinant of the A matrix (Valentin's original), but the HP48G and HP49G+ give  $9.98918882E-8$ .

I've triple checked to make sure that I typed in the matrix correctly. Flag 54 is set on all the machines. I created Hilbert matrices of order 7, 8, 9, and 10, and computed the inverses and the determinants of each. I got the same results on the HP49 as on the HP48G and HP49G+. Something peculiar is going on here; in almost every other test I have tried, my HP49 gets the same result as the HP48G and HP49G+.

The RMS relative errors for Valentin's original problem and for Rodger's three modifications for seventeen solvers which are available in my inventory are

Solver	Valentin	Rodger 1	Rodger 2	Rodger 3
Sharp PC-1261	18.386	9.823E-01	9.673E-01	1.250E-07
Model 100 Swinnen SP	11.506	1.014	9.996E-01	2.832E-04
HP-28S B/A	3.927	13.417	13.220	1.253E-10
HP-41 "seq"	2.371	1.019	1.004	2.445E-07
HP-49G B/A	1.327	1.065E-03	1.080E-03	5.477E-13
HP-49G INV A * B	9.060E-0	1.094E-03	1.080E-03	1.573E-10
Sharp PC-1261 Swinnen	7.664E-01	1.117	1.101	2.647E-09
TI-59 ML-02	7.367E-01	2.464E-02	2.428E-02	9.433E-11
TI-95 Math Module	3.147E-01	2.955E-03	2.906E-03	3.297E-11
HP-41 Math Pac	3.018E-01	1.027	1.008	2.005E-08
CC-40 Math Module	2.260E-01	1.025E-01	1.010E-01	3.202E-10
Model 100 HCN DP	1.214E-01	2.215E-03	2.182E-03	2.321E-11
TI-85 INV B * A	3.582E-02	9.270E-03	9.155E-03	8.683E-12
CC-40 Swinnen	5.125E-03	8.466E-03	8.342E-03	2.701E-12
TI-85 Simul Equations	4.274E-03	9.292E-03	9.155E-03	4.399E-12
Model 100 Swinnen DP	1.799E-03	3.625E-04	3.572E-04	9.217E-13
HP-49G RREF	1.472E-03	8.281E-05	8.352E-05	0

By the way, as I must have mentioned in an earlier posting, the HP48G and HP49G+ solve Valentin's original system exactly using the B/A method. That is, they get  $[1 \ 1 \ 1 \ 1 \ 1 \ 1]$  for the solution vector.

The RMS relative errors for all solvers are substantially smaller for Rodger's third modification. That modification seems to have dumbed down Valentin's original problem so that almost any solver can succeed. For example, solvers such as the the B/A solution with the HP-28S and the ML-02 routine with the TI-59 which couldn't even get all the signs correct with Valentin's original problem get rather good results with Rodger's third modification. Even the Model 100 solver operating in the single precision mode with a six digit mantissa gets four digits correct for Rodger's third modification.

Of course, that was the point of that modification; to show that there is a way to solve a badly ill-conditioned system even on a calculator with only a few digits. I got that perturbation by means of the Singular Value Decomposition, and I'll post a note to show how to do it.

When I found that the RREF solver on the HP-49G consistently yielded "calculated" solutions which were closer to the "unique" solutions than the B/A solver I decided to see if I could modify Rodger's program to replace the B/A solution with the RREF solution. My plan was to follow his program to generate the perturbed matrix, augment the perturbed matrix with the B vector and solve with the RREF routine. I had difficulty doing that because I couldn't get the AUGMENT command to attach the B vector at the right side of the perturbed matrix yielding the 7x8 matrix needed by the RREF routine. No matter what I tried the AUGMENT command would attach the B vector at the top or bottom of the perturbed matrix rather than at the right of the perturbed matrix. Eventually, I transposed the perturbed matrix, used the AUGMENT command to attach the B vector at the bottom yielding an 8x7 matrix, and transposed the 8x7 matrix back to yield the 7x8 matrix I wanted. My complete routine was.

```
<< B A RANM 247.487 / A + TRAN B AUGMENT TRN RREF ->COL DROP -3 RND >>
```

There must be an easier way to do what I wanted to do but I couldn't find it. Maybe if I had a manual .... Maybe someone more familiar with the HP-49G can help.

Here's a program to set up and solve using RREF. Assuming B and A are on the stack (B in level 2, A in level 1):

```
<< SWAP { 7 } RDM 8 COL+ RREF >>
```

If you want the result to be the solution vector in the form that B/A returns it, then use:

```
<< SWAP { 7 } RDM 8 COL+ RREF 8 COL- SWAP DROP { 7 1 } RDM >>
```

To get the result of a random perturbation of A, do this (this program assumes that the A and B matrices are stored in variables of those names):

```
<<B A RANM 247 .487 / A + SWAP { 7 } RDM 8 COL+ RREF 8 COL- SWAP DROP { 7 1 } RDM >>
```

## Re: Even More Simultaneous Equation Solutions

Message #5 Posted by [Rodger Rosenbaum](#) on 27 Mar 2006, 5:04 a.m.,  
in response to message #1 by [Palmer O. Hanson, Jr.](#)

Palmer said:

I recently ran Valentin's original problem on the HP-49G (actually on an emulator) with the following results:

INV A * B	B/A	RREF
-1.076	-0.6472	1.00148007239
0.999786	1.00110019	0.999999813411
0.407	-0.7415	1.00131966086
0.144	0.6554	1.00132720057
0.999124	0.9996398	1.00000091517
0.553	1.1592	0.998672819845
0.61	3.5385	0.997223232158

I think there is a typo in the 5th entry under RREF; instead of 1.00000091517, it should be 1.00000091577. If you make this change and then premultiply this solution vector (the RREF vector) by the A matrix (using a 12 digit HP calculator such as the HP48G or HP49G+), you will get exactly the B vector. In other words, the RREF solution vector is an "exact" solution, and there is not a unique solution. This can only happen with floating point arithmetic, of course. In infinite precision arithmetic there is only one exact solution, namely {1 1 1 1 1 1}. I wonder how many "exact" solutions there are in 12 digit BCD floating point arithmetic using "round to even". The rounding mode will be one factor determining the "exact" floating point solutions.

[ [Return to Index](#) | [Top of Index](#) ]



Go back to the main exhibit hall