HP Forum Archive 15

[Return to Index | Top of Index]

More on ill conditioning and calculator precision

*Message #1 Posted by **Palmer O. Hanson, Jr.** on 21 Jan 2006, 11:40 p.m.*

On January 7 Rodger Rosenbaum initiated the thread "Ill conditioning and calculator precision" to carry forward discussion of an earlier thread initiated by Valentin Albillo. Near the end of the thread Rodger wrote "But, back to the original purpose I had when I started this thread. I noticed that a slightly different perturbation of the column matrix has even greater effect. Change the third element of the column matrix in Valentin's original problem from 45 to 45.1--now the exact solution is:

```
x1 = -2128901625
x2 =      268387
x3 = -1898169427
x4 = -1909014362
x5 =    -1317226
x6 =  1908985003
x7 =  3994038147
```

This is a tremendous change in the solution for an apparently insignificant change in one element."

For reference Valentin's original problem was

```
    1.3 x1 + 7.2 x2 + 5.7 x3 + 9.4 x4 + 9.0 x5 + 9.2 x6 + 3.5 x7 =  45.3
    4.0 x1 + 9.3 x2 + 9.0 x3 + 9.9 x4 + 0.1 x5 + 9.5 x6 + 6.6 x7 =  48.4
    4.8 x1 + 9.1 x2 + 7.1 x3 + 4.8 x4 + 9.3 x5 + 3.2 x6 + 6.7 x7 =  45.0
    0.7 x1 + 9.3 x2 + 2.9 x3 + 0.2 x4 + 2.4 x5 + 2.4 x6 + 0.7 x7 =  18.6
    4.1 x1 + 8.4 x2 + 4.4 x3 + 4.0 x4 + 8.2 x5 + 2.7 x6 + 4.9 x7 =  36.7
    0.3 x1 + 7.2 x2 + 0.6 x3 + 3.3 x4 + 9.7 x5 + 3.4 x6 + 0.4 x7 =  24.9
    4.3 x1 + 8.2 x2 + 6.6 x3 + 4.3 x4 + 8.3 x5 + 2.9 x6 + 6.1 x7 =  40.7
```

where the system has the exact solution:

x1 = x2 = x3 = x4 = x5 = x6 = x7 = 1.0

In the earlier thread I provided solutions to the original problem on several different machines and to a modied problem which changed the B1 value from 45.3 to 45.4 . Solutions to the modified problem with B3 changed from 45.0 to 45.1 follow.

With my HP-28S:

```
                  B/A                                           INV A * B


       X                      A*X                            X                 A*X


-30272808391.1               45.04                -30272808407.9            -87.494
     3816428.18496           48.302                    3816428.21599       -181.961
-26991815255.9               46.374               -26991815236.6           -162.843
```

-27146029351.6	18.506	-27146029342.2	-7.895
-18730860.0936	36.536	-18780860.0977	-144.096
27145611882.5	24.694	27145611823	-18.702
56794898378.3	40.67	56794898341.2	-146.409

where the signs of the elements of the solution are correct but the magnitudes are off by an order of magnitude yielding relative errors of 14.2 . The calculations of A*X do indicate that there are difficulties with the solution.

My Model 100 and TI-85 yield the following solutions

Model 100		TI-85	
X	A*X	X	A*X
-2128141186.5423	45.3	-2148392852.67	45.3014
268291.13337654	48.399	270844.2167	49.3991
-1897491405.5188	45.1	-1915548178.57	45.1
-1908332466.734	18.5999	-1926492404.72	18.5999
-1316755.4898248	36.698	-1329285.91435	36.6983
1908303118.2216	24.9	1926462776.91	24.90025
3992611486.3345	40.699	4030605692.38	40.7002

where the signs and magnitudes are correct. At least the first three digits of each element of the Model 100 answer are correct yielding relative errors of 3.57e-04. The relative errors for the TI-85 are 9.15e-03.

The MATRIX routine of the HP-41 Math Pac and ML-02 routine of the TI-59 Master Library yield the following results

HP-41		TI-59	
X	A*X	X	A*X
24666733.49	45.3	-2077198547.178	45.32
3108.726	48.4	261868.9227801	48.44
21993331.58	45.1	-1852069973.364	45.11
22118987.39	18.59	-1862651525.312	18.6
15263.2	36.7	-1285235.478686	36.7
-22118645.18	24.84	1862622879.371	24.901
-46277322.87	40.7	3897037862.598	40.71

where the magnitudes and even most of the signs of the result from the HP-41 are wrong. The magnitudes and signs of the result from the TI-59 are correct and of the correct order of magnitude yielding relative errors of 2.4e-2. Curiously, with the HP-41 the product of Matrix A and the solution vector yield the exact values of the elements of vector B. This suggests that comparing the product A*X with B to measure the acceptability of the solution may not be a good idea if the user is interested in a real world solution as opposed to a solution which is of value only in the calculator's world. The determinants for Valentin's Matrix A were

Exact	1.e-07
HP-41	-86.31e-7
HP-28S	0.0703238892937e-07
TI-59	1.024890772918e-07
TI-85	0.990927530944e-07

Rodger also wrote "Try something I described in an earlier post; add .1 to the {3,7} element of the A matrix, and subtract .1 from the {7,7} element. This tiny change reduces the condition number of the A matrix from about 3E12 to about 400, and using this slightly perturbed A matrix with the perturbed column matrix mentioned just above gives a HP48G solution:

```
x1 = 1.20999247798
x2 = 1.01325966115
x3 = 1.26616078717
x4 = 1.29157926088
x5 = .997263593179
x6 = .690061410742
x7 = .499897531935
```

This isn't as close to [1 1 1 1 1 1 1] as we would like, but given what the solution was with just the third element in the column matrix perturbed and using the original A matrix, this inprovement is an astounding result considering that only two tiny perturbations were made to the A matrix.

My purpose here is to show that with proper techniques we can find reasonable solutions to very ill-conditioned systems. Systems as badly conditioned as Valentin's example are unlikely to occur in practice; they must be artificially constructed. But, if we can handle them, then we can certainly handle the ones we encounter in real life problems.

Palmer, it would be good if you tried solving the system on your various calculators, consisting of the column matrix perturbed in the third element, and the {3,7} and {7,7} elements of the A matrix perturbed as I described. I think you will find that you don't get wildly varying results with these changes. For example, on my HP48S, which uses the same matrix routines as the HP28, I get about 9 or 10 digit agreement with the solution posted above, whether I use the divide key or invert the A matrix and multiply times the column matrix."

My HP-28S yielded

B/A		INV A *B	
X	A*X	X	A*X
1.20999247808	45.3	1.20999247808	45.2999999988
1.01325466114	48.4	1.01325966123	48.3998999983
1.26616078724	45.1	1.26616078741	45.0999999989
1.29157926094	18.6	1.29157926101	18.6000000005
0.99726359318	36.7	0.997263593181	36.6999999992
0.690061410687	24.9	0.690061410466	24.9000000001
0.499897531784	40.7	0.499897531481	40.6999999991

My HP-28S solutions are consistent with Rodger's results from his HP-48S. Note that when using the B/A method the product of matrix A and the solution vector yields exactly the B vector.

The simultaneous equation solution with my TI-85 is

```
1.2099924779883
1.0132596611477
1.2661607871723
1.2915792608817
0.99726359317885
0.69006141074113
0.49989753192964
```

where the product of the A matrix and the solution matrix yields exactly the B matrix.

The solution from my Model 100 is

X	B - A*X
1.2099924779846	-1e-12
1.0132596611482	2e-12
1.2661607871704	1e-12

```
1.291579260881            1e-12
0.99726359317867          0
0.69006141074154          0
0.49989753193417          1e-12
```

where in the second column I listed B - A*X rather than A*X to avoid typing in all those zeroes. I wasn't pleased with the TI-85 and Model 100 results. We were hoping to get a solution closer to all ones using the modification to Valentin's problem, but the TI-85 and Model 100 solutions were further away, not closer than with Valentin's original problem. It occurred to me that perhaps the exact solution to the perturbed problem wasn't all ones. I went to my HP-49G emulator and solved the problem in the exact mode with the following results

```
x1 = 9667493448/7989713675  =   1.2099924779845
x2 = 8095654571/7989713675  =   1.013259661148
x3 = 10116362156/7989713675 =   1.2661607871699
x4 = 10319348483/7989713675 =   1.2915792608801
x5 = 7967850568/7989713675  =   0.99726359317876
x6 = 1102678618/1597942735  =   0.6900614107426
x7 = 3994038147/7989713675  =   0.49989753193502
```

which confirmed my suspicions.

The results obtained with the MATRIX routine in the HP-41 Math Pac module and the ML-02 routine in the TI-59 Master Library module:

```
   HP-41                 TI-59


1.209992482          1.209992478108
1.013259663          1.013259661129
1.266160800          1.266160787202
1.291579278          1.29157926087
0.997263593          0.9972635931871
0.690061393          0.6900614107704
0.499897511          0.4998975318243
```

where the differences from the exact answer are only seen in the lower order digits of each machine. The determinants for the modified matrix A are

```
Exact           798.9713675
HP-41           798.9713752
HP-28S          798.971367542
TI-59           798.9713674893
TI-85           798.9713675
```

Some playing with the numbers on my HP-28S:

If I multiply Valentin's original A matrix times the X vector from from the third modification with A(3,7), A(7,7) and B3 changed I get

```
45.3
48.4
45.0500102468
18.6
36.7
24.9
40.7499897532
```

where the RMS relative error with respect to the B vector of the third modification is 2.672e-02. Furthermore, If I multiply the original A matrix by a vector obtained by rounding the X vector to the nearest 0.25, i.e., a vector made up of the elements 1.25, 1, 1.25, 1.25, 1, 0.75, 0.5 I get

```
45.35
48.45
```

```
45.025
18.6
36.7
24.9
40.725
```

which has n RMS relative error with respect to the B vector of the third modification of 2.988e-02

As a novice to the idea of perturbing problems to relieve ill-conditioning I am not sure what all of this means.

Finally, Rodger suggested that someone should translate a Fortran program for singular value decomposition (SVD) for use with BASIC. I took a quick look. Someone else will have to do it. I haven't done any Fortran programming since about 1970.

Re: More on ill conditioning and calculator precision

*Message #2 Posted by **Rodger Rosenbaum** on 23 Jan 2006, 9:58 a.m.,*
in response to message #1 by Palmer O. Hanson, Jr.

Let me make some further observations regarding Valentin's original system. It was intended to be an example of a set of measurements one might make in the lab, for example. The entries in the A matrix were recorded as two digit quantities, presumably because they were real measurements which could not be taken to be more accurate than two digits. Consider the first entry in the A matrix, namely 1.3. This might have been measured with a digital meter giving more than two digits, but rounded and recorded with only two digits. The actual measurement could have been any number x, 1.25 < x < 1.35; it could have been 1.273, or it could have been 1.329, for example. Either of those readings would have been rounded and recorded as 1.3. So each entry in the A matrix could have actually been measured as the recorded value +- .5 ULP.

So, 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. If we were to round A' to 2 digits, we would get back the original A matrix. This A' matrix could indeed be the matrix of measurements made with the hypothetical digital meter before they were rounded to two digits; we have no way of knowing since we only have the values after they were rounded. In fact, if we create multiple 7x7 perturbation matrices filled with the previously described random numbers and add them to the original A matrix, all of these A' matrices could have been the actual measurements. There's no reason so far to prefer one of them over the others, including Valentin's original. In fact, the original *unperturbed* A matrix would seem to be a somewhat unlikely candidate because that would mean that all 49 measurements had exactly 2 digit precision. And if the measurements were made with some instrument that only gave 2 digit readings, the real value of the measured quantity was still undoubtedly contaminated with noise, and a value somewhere in the range of recorded value +- .5 ULP was the true value.

You can get insight about the behavior of the original system by repeatedly creating a perturbation matrix, A', and adding it to the original A matrix and solving the system. Each of those solutions is just as likely as any other, including the [1 1 1 1 1 1 1] solution. I have done this for 150 repititions and here are some of the results (rounded to 3 digits):

```
[-8.71 1.33 -7.52 -6.28 .804 8.33 18.5]
[.785 .981 .841 .773 1.01 1.22 1.39]
[1.55 .948 1.38 1.23 1.03 .788 .172]
[1.2 .864 .763 .371 1.06 1.69 1.33]
[4.08 1.06 4.08 4.34 .98 -2.4 -5.27]
[5.44 1.1 5.05 5.21 .963 -3.25 -7.51]
[9.18 1.08 7.65 8.17 .911 -6.07 -13.6]
[-20.7 .628 -19.6 -20.6 1.06 23.2 43.5]
[-1 .808 -1.36 -1.72 1.1 3.83 5.65]
```

This procedure can't indicate that any one of these solutions is any more likely than another, or than the [1 1 1 1 1 1 1] solution, to be the "true" solution.

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).

Now, in your previous post, you said:

"I wasn't pleased with the TI-85 and Model 100 results. We were hoping to get a solution closer to all ones using the modification to Valentin's problem, but the TI-85 and Model 100 solutions were further away, not closer than with Valentin's original problem. It occurred to me that perhaps the exact solution to the perturbed problem wasn't all ones. I went to my HP-49G emulator and solved the problem in the exact mode..."

My discussion should have indicated that the perturbation I suggested wasn't intended to give a solution of [1 1 1 1 1 1 1], because we don't yet know if that is really the "best", or "true", solution. What I was trying to show is that if you carry out certain procedures (a simple perturbation of the A matrix here), you can reduce the condition number enough that the new system can be solved by almost any calculator to reasonable accuracy. The solution we got, and with respect to which all the calculators you tried gave good results, is undoubtedly much closer to the "true" solution than the exact solution:

```
x1 = -2128901625
x2 =      268387
x3 = -1898169427
x4 = -1909014362
x5 =    -1317226
x6 =  1908985003
x7 =  3994038147
```

In another post I'll talk about using the singular value decomposition to get the "best", or "true", solution.

[Return to Index | Top of Index]

GTO Go back to the main exhibit hall