DATA QUALITY IN SECURITY OUTSOURCING OF SCIENTIFIC COMPUTATIONS

JOHN R. RICE AND CHIEH-HSIEN TIAO

Contents

1. Introduction

In [1], [2], and [3], the authors have explored ways to mask data in order to outsource them to outside agents for doing scientific computations. In section 7 of [1] (p. 23), the authors tried to measure the quality of the data returned by agents by doing experiments on some cases and measure the errors of the returned data. While the measurement itself is good idea of understanding the quality of the outsourcing computations, we found some things which need to be improved.

First, in the experiment, the authors measured the errors between the returned outsourcing solutions and the local computed solutions. Without knowing the errors introduced by the local computations, we are unable to know the true quality of the outsourcing solutions. For problems which are naturally more sensitive to solve, a large error measured would not necessarily imply low outsourcing quality, while for problems which are easy, even a relatively small error measured could mean a big problem in the algorithm.

Also in the study done in [1], the test cases seem to be generated randomly. However randomly generated numbers are often uniform in their characteristics. It becomes obvious when dealing with the inverse of a matrix and solving a linear system that, from time to time the size of the entries of a result can be magnified by an order of magnitude, and some matrix computations are just too sensitive to be outsourced using the proposed algorithms. So it becomes clear that we need to take a much closer look at the effect introduced by distorting the sizes of the entries independently.

Our goal is to make the outsourcing of scientific computations not only secure but also accurate. Therefore we need algorithms which can serve both purposes equally well.

Date: December 21, 1998.

2. Matrix Multiplication

Among the three computations we performed, the matrix multiplication has the most stable algorithm (see Table 2), which means that the outsourcing results will match most closely with the local results no matter what kind of data are fed into it and how big of the size distortions that are introduced into the individual element of the matrices.

Assume C_0 is the true solution to the equation $C = A * B$, where A and B are $n \times n$ matrices with precision up to p digits. We distort the entries in A and B by up to d_A and d_B digits, which means each entry in A and B is multiplied by a number 10^s where $|s| \leq d_A$ when applied on A and $|s| \leq d_B$ when applied on B. C_1 is the result of $A * B$ computed by MATLAB directly locally, and C_2 is the result of $A*B$ by first using the outsourcing algorithm to disguise the problem, then using MATLAB on the disguised problem, and finally bringing back the solution by reversing the disguise. RE_1 and RE_2 are the relative errors of the local solutions and the relative error of the outsourcing solution respectively, i.e.,

$$RE_1 = \frac{\|C_1 - C_0\|}{\|C_0\|} \qquad and \qquad RE_2 = \frac{\|C_2 - C_0\|}{\|C_0\|},$$

where the infinity norm is used. For each combination of (n, p, d_A, d_B), 100 tests were made, and the data recorded in the following table are the averages of each 100 results.

n	p	d_A	d_B	RE_1	RE_2	$\|\|A\|\|$	$\|B\|$	$\|C\|$
10	3	0	0	1.3195e-16	1.5735e-16	4.7346e+00	4.7311e+00	1.4473e+01
10	3	8	0	1.2330e-16	1.7664e-16	1.1853e+07	4.5107e+00	3.8681e+07
10	3	8	8	1.3337e-16	1.7986e-16	1.1273e+07	1.1954e+07	6.1729e+13
10	4	0	0	1.4621e-16	1.8529e-16	4.3287e+00	4.3608e+00	1.2134e+01
10	4	12	0	1.1548e-16	1.6466e-16	9.0247e+10	4.2018e+00	2.5142e+11
10	4	12	12	9.5925e-17	1.5691e-16	1.0119e+11	9.5206e+10	4.1002e+21
50	3	0	0	2.4565e-16	2.5970e-16	1.8514e+01	1.8576e+01	2.5630e+02
50	3	8	0	2.1095e-16	2.3670e-16	4.2592e+07	1.8513e+01	5.9742e+08
50	3	8	8	1.9846e-16	2.4525e-16	4.2701e+07	4.3128e+07	8.5940e+14
50	4	0	0	2.3678e-16	2.5015e-16	1.8599e+01	1.9018e+01	2.5659e+02
50	4	12	0	2.0572e-16	2.2747e-16	3.4437e+11	1.8676e+01	4.7946e+12
50	4	12	12	1.7357e-16	2.2232e-16	3.4457e+11	3.4508e+11	4.6448e+22
100	3	0	0	3.1973e-16	3.2773e-16	3.5323e+01	3.5360e+01	9.7456e+02
100	3	8	0	2.7908e-16	3.0703e-16	7.4372e+07	3.5313e+01	2.0561e+09
100	3	8	8	2.5806e-16	3.1372e-16	7.4674e+07	7.5828e+07	2.7526e+15
100	4	0	0	3.1275e-16	3.2095e-16	3.5408e+01	3.5469e+01	9.7219e+02
100	4	12	0	2.6796e-16	2.9580e-16	5.7545e+11	3.5510e+01	1.5974e+13
100	4	12	12	2.2321e-16	2.5592e-16	5.6078e+11	5.7411e+11	1.4273e+23

Table 2. Data on the precision of matrix multiplication with and without outsourcing.

From Table 2, we can see that both the local solutions and the outsourcing solutions constantly give back about 16 correct digits. It is independent of the size

of matrix, n, the precision of the original data, p, the order of distortions on A and B, d_A and d_B, or the size of matrices A, B, and C, $\|A\|$, $\|B\|$, and $\|C\|$.

The outsourcing solutions might introduce slightly more errors than the local solutions, but the ratio between the two errors does not seem to vary with the scaling of the matrix. Note that the scaling distortions are larger than likely to occur in applications.

3. Matrix Inversion

Among the three computations, the outsourcing algorithm for matrix inversion, after improving the algorithm for solving systems of linear equations, (see section 4), gives the biggest error in the sense that the returned outsourcing solutions lose the most digits in comparison with the local solutions. It loses about 1 to 3 more digits, especially when a large random distortion is imposed (see Table 3).

Assume B_0 is the true solution to the equation $B = A^{-1}$, where A is a $n \times n$ matrix with precision up to p digits. We distort the entries in A by a magnitude of up to d digits, which means each entry in A will be multiplied by a number 10^s where $|s| \leq d$. B_1 is the result of A^{-1} computed by MATLAB directly locally, and B_2 is the result of A^{-1} by first using the outsourcing algorithm to disguise the problem, then using MATLAB on the disguised problem, and finally bringing back the solution by reversing the disguise. RE_1 and RE_2 are the relative error of the local solution and the relative error of the outsourcing solution respectively, i.e.,

$$RE_1 = \frac{\|B_1 - B_0\|}{\|B_0\|} \qquad and \qquad RE_2 = \frac{\|B_2 - B_0\|}{\|B_0\|},$$

where the infinity norm is used except for the one data row marked. For each combination of (n, p, d), 100 tests were made, and the data recorded in the following table are the averages of each 100 results.

n	p	d	RE_1	RE_2	$\|\|A\|\|$	$\|\|B_0\|\|$	$\|\|B_1\|\|$
10	3	0	1.1753e-15	8.5756e-14	4.3288e-02	4.7346e+03	4.7346e+03
10	3	8	2.9563e-14	2.5041e-11	9.7125e-05	1.1133e+10	1.1133e+10
[1] 10	3	8	3.3989e-13	2.2222e-11	1.0464e-04	9.6405e+09	9.6405e+09
10	4	0	2.0908e-14	1.1103e-12	2.1174e-01	4.3287e+04	4.3287e+04
10	4	12	7.0634e-10	7.4208e-05	1.3995e+03	9.0247e+14	9.0247e+14
50	3	0	1.9564e-14	3.1706e-13	1.7973e-01	1.8603e+04	1.8603e+04
50	3	8	4.3834e-14	9.7076e-12	1.7747e-06	4.3118e+10	4.3118e+10
50	4	0	2.4927e-14	1.9574e-12	2.4228e-02	1.8656e+05	1.8656e+05
50	4	12	8.0252e-14	7.1275e-11	1.3798e-10	3.4271e+15	3.4271e+15
100	3	0	5.9689e-13	1.2177e-11	3.1901e+00	3.5406e+04	3.5406e+04
100	3	8	2.5586e-13	4.0428e-11	4.8142e-07	7.5540e+10	7.5540e+10
100	4	0	1.6558e-13	5.6363e-12	8.4172e-02	3.5509e+05	3.5509e+05
100	4	12	1.1221e-13	2.8073e-11	7.7871e-12	5.7545e+15	5.7545e+15

Table 3. Data on the precision of matrix inversion with and without outsourcing. The experiment marked by [1] uses the square error instead of the infinity norm.

Table 3 shows that even the inversion of a matrix is not a particularly "hard" problem to outsource, in which we mean that the local solutions can constantly give back 13 to 15 correct digits, but the outsourcing solutions introduce extra 1 to 3 missing digits compare to what the local solutions can give.

When the size of the matrices n increases, both errors from the local solutions and from the outsourcing solutions increase, but the rate of increase is the smallest in comparison to the following two factors. Both errors increase when the precision p of the original data increase, however the precision seems to affect the errors more for the outsourcing computation than for the local computation. The increment of the distortion d on the size of entries seems to have an effect similar to the change of precision.

4. System of Linear Equations

The original algorithm given in [1] for solving a system of linear equations does not produce very accurate answers even for data which are not distorted. The following is a table showing that it typically causes more than 3 additional digits lost even in the simplest situations.

n	p	d	RE_1	RE_2	$\|\|A\|\|$	$\|\|A^{-1}\|\|$	$\|b\|$
10	3	0	1.8343e-15	8.2677e-12	4.7346e+03	4.3288e-02	2.4582e+03
10	3	8	8.1337e-13	5.8857e-09	1.0293e+10	1.0676e-03	7.2889e+09
10	4	0	2.9634e-13	2.2969e-10	4.3287e+04	2.1174e-01	2.2085e+04
10	4	12	1.2059e-05	1.2732e+02	9.0983e+14	3.6512e+01	5.3072e+14

Table 4(a). Data on the precision of solving a system of linear equations using the original algorithm with and without outsourcing.

By carefully studying the original algorithm, we find that the weakness is due to the fact that, in equation (16) on page 14 of [1], the algorithm adds a randomly chosen vector V to the (unsolved) solution, and later on, in equation (18), recovers the solution by subtracting the V from the returned solution. The danger of doing so is that when the size of the true solution is much smaller than the chosen V - which is often so as we can see in the above table, the precision of the solution is compromised. In fact, from the above data, we find that when experimenting on 10×10 matrices with original data precision to be 4 digits and distortion of up to 10^{12}, one finds that no correct digits of the solutions are likely to be found.

We give another algorithm which is more accurate while not compromising security. The basic idea is to replace vector addition by matrix multiplication (see steps 3 and 4 in the algorithm 4.1).

Algorithm 4.1. *Assuming we are given a system of equations $A * x = b$, do the following for outsourcing:*

1. *Generate 3 $n \times n$ invertible matrices, P_1, P_2, and P_3, as described in* [1].
2. *Randomly generate an $n \times n - 1$ matrix R, and an integer k, $1 \leq k \leq n$.*

3. *Let* $B = [R_1, \ldots, R_{k-1}, b, R_k, \ldots, R_{n-1}]$, *i.e.* B *is a* $n \times n$ *matrix by inserting* b *into the* $k-th$ *column of* R.
4. *Compute the following locally:*

$$\begin{aligned} \tilde{A} &= P_1 * A * P_2^{-1}, \\ \tilde{B} &= P_1 * B * P_3^{-1}. \end{aligned}$$

5. *Outsource to the agent the problem:*

$$\tilde{A} * \tilde{X} = \tilde{B}.$$

6. *After getting* $\tilde{X}$ *back, compute the following locally:*

$$X = P_2^{-1} * \tilde{X} * P_3.$$

7. *The solution* x *is the* k*-th column of* X, *i.e.* $x = X_k$.

By expanding the right hand side b to an $n \times n$ matrix B in step 3, we disguise the computation using basic matrix multiplication in step 4, and thus avoid the danger of having to add a vector of incompatible size.

Assume x_0 is the true solution to the equation $A * x = b$, where A is a $n \times n$ matrix with precision up to p digits. We distort the entries in A by a magnitude of up to d digits, which means each entry in A will be multiplied by a number 10^s where $|s| \leq d$. x_1 is the solution to $A * x = b$ computed by MATLAB directly locally, and x_2 is the solution to $A * x = b$ by first using the outsourcing algorithm to disguise the problem, then using MATLAB on the disguised problem, and finally bringing back the solution by reversing the disguise. RE_1 and RE_2 are the relative error of the local solution and the relative error of the outsourcing solution respectively, i.e.,

$$RE_1 = \frac{|x_1 - x_0|}{|x_0|} \qquad and \qquad RE_2 = \frac{|x_2 - x_0|}{|x_0|},$$

where the infinity norm is used except for the one data row marked, For each combination of (n, p, d), 100 tests were made, and the data recorded in the following table are the averages of each 100 results.

	n	p	d	RE_1	RE_2	$\|\|A\|\|$	$\|\|A^{-1}\|\|$	$\|b\|$
	10	3	0	5.6941e-15	9.5077e-15	4.1533e+03	7.5258e-02	2.3261e+03
	10	3	8	4.9117e-11	1.7773e-11	9.2371e+09	2.3480e-02	5.5849e+09
[1]	10	3	8	3.0290e-12	5.8685e-12	9.6318e+09	7.5415e-04	1.0247e+10
	10	4	0	2.6071e-14	4.2836e-13	4.2795e+04	7.3152e-01	2.2879e+04
	10	4	8	2.8866e-09	4.1092e-09	1.1879e+11	1.2819e-01	6.8397e+10
	10	4	12	6.8512e-05	1.0344e-04	9.4660e+14	3.2218e+01	5.3541e+14
	50	3	0	7.0219e-14	7.2583e-14	1.8549e+04	2.1238e-01	9.9571e+03
	50	3	8	1.9056e-13	2.2464e-13	4.3535e+10	1.4479e-06	2.5295e+10
	50	4	0	2.1020e-13	1.8087e-13	1.8722e+05	5.4172e-02	9.7469e+04
	50	4	12	1.7457e-12	2.5374e-12	3.4247e+15	1.2083e-10	1.9443e+15
	100	3	0	7.4311e-13	9.2974e-13	3.5306e+04	2.1027e+00	1.8321e+04
	100	3	8	3.6745e-13	4.7536e-13	7.5046e+10	4.6451e-07	4.1878e+10
	100	4	0	4.4013e-13	5.1464e-13	3.5424e+05	1.0810e-01	1.8269e+05
	100	4	12	5.1983e-13	6.4081e-13	5.6712e+15	1.2723e-11	3.2306e+15

Table 4(b). Data on the precision of solving a system of linear equations using the improved algorithm with and without outsourcing. The experiment marked by [1] uses the square error instead of the infinity norm.

From the data in Table 4(b), we can see that neither solution has as many digits correct as the ones returned by the multiplication algorithm, but the outsourced algorithm solutions are as correct as those obtained by local computations. We know that there are "hard" linear equation problems where we can not expect to have highly accurate solutions even for good algorithms, the best we can hope for is that the outsourcing solutions match closely the accuracy of local solutions - which we have achieved in this algorithm. In general, the returned solutions have very slightly less accuracy than the local solutions, but the ratio between them does not seem to vary over any changes of factors.

5. Conclusion

There are two areas where further studies can be done.

First, there are still rooms for improvement in the matrix inversion algorithm. Ideally, in a reliable algorithm, the outsourcing solutions should match closely with the local solutions. We believe that 1 to 2 more digits correctness should be able to be obtained if we study the original algorithm closely enough to find out where the weakness arises.

Second, the data seems to indicate that the matrices of smaller size are more vulnerable to the effect of high precisions and high distortions. A 10×10 matrix usually performs a lot better than a 50×50 or 100×100 matrix on matrix inversion and systems of linear equations problems with small precisions and no distortions. But once given higher precisions and higher distortions, the performance suddenly gets very bad for the smaller size matrix, while the bigger size matrix maintains certain quality of performances. The phenomena deserves more study.

References

[1] **Atallah, M.**, **Pantazopoulos, K.**, **Spafford, E.**: *Secure Outsourcing of Some Computations.* CSD-TR 96-074, 1-26 (1996)
[2] **Atallah, M.**, **Rice, J.**: *Secure Outsourcing of Scientific Computations.* preprint, 1-46 (1998)
[3] **Atallah, M.**, **Rice, J.**: *Masking Scientific Computations.* preprint, 1-6 (1998)

Department of Computer Sciences, Purdue University, IN 47907, USA