Experiments with data perturbations to study condition numbers and numerical stability

J. Erhel
IRISA/INRIA, Campus de Beaulieu, 35042 RENNES CEDEX, FRANCE

Abstract

Ill-conditioning as well as roundoff errors lead to inaccuracies in scientific applications. Random perturbations in the initial data allows to derive an error estimation. By varying these perturbations and under some reasonable assumption on the rounding errors, we can find an interval of perturbations where a log-linear least-squares fit gives an estimation of the regularity and the condition number of the problem.
keywords : Regularity, Condition Number, Rounding Errors.
AMS Classification : 65g05, 65f35.

1 Introduction

Numerical simulation plays an important role to solve physical problems. Though the accuracy of the results is crucial, this area has not yet received a great attention and few tools exist to control or improve numerical quality. Moreover existing tools are difficult to use.

The sensitivity to initial data depends on the problem and is measured by its regularity and its condition number, using perturbation theory. Another source of inaccuracy is the finite precision arithmetic. The sensitivity to roundoff errors depends on the algorithm and is measured by forward or backward analysis.

Deterministic methods have been designed to estimate the stability of algorithms and problems [21, 20]. They are based on a computational graph representing the program. This approach is interesting but seems to be very

expensive. Tools to use it easily have been recently designed [4], and are currently experimented.

Another idea which is not new is to use probabilities or statistics [24]. The first approach is to simulate rounding errors by perturbing randomly all the floating-point operations, to execute several times the program and then to derive from the samples of results an estimation of the accuracy of the result. The software CESTAC implements this technique [10]. Though it can be useful in some practical cases, it is quite expensive because it requires a new arithmetic, emulated by software in general. Moreover, it does not provide an estimation of the problem condition number or the numerical stability. The first release of the software PRECISE [5] is similar but it analyzes in addition the residuals associated to the problem. In general, the backward analysis of an algorithm relies on such residuals. Therefore, this technique allows to control the stability of the algorithm and then to estimate the condition number of the problem. However, it remains expensive since it requires also a new arithmetic.

These observations led to simulate only perturbations in the initial data in order to make easier their implementation and use. However small perturbations (for example when few digits are perturbed) may lead to false conclusions as quoted in [14]. Therefore, several sizes of perturbations must be used to obtain meaningful estimations, increasing of course the cost of execution. This is done in the software SCALP [13] and in the second release of PRECISE [7] along with a statistical analysis to obtain regularity and condition number estimations. In PRECISE, a statistical analysis of the residuals allows to study the backward stability of the algorithm.

This paper presents an extension of SCALP, in order to get a reliable estimation of the regularity and the condition number. It relies on the following observation. For small perturbations in the initial data, roundoff errors are predominent, whereas for sufficiently large perturbations they become negligible. The perturbations must be kept small enough to allow perturbation theory. This defines a domain of perturbations where the original method SCALP can be applied and allows to derive sharp estimations of problem conditioning and error estimation.

The experiments conducted so far deal with simple and well-known problems and algorithms, namely linear systems and quadratic equations. The results are very promising, proving at least for these examples the capabilities and usefulness of this approach. Results are given in the third section, while the second section is devoted to the description of the experimental tool.

2 Description of the methodology

Our objective is to get an estimation of the regularity and the condition number of a problem. In linear algebra, numerous condition numbers estimators have been designed [17], even for sparse matrices [3]. However, for general problems, it may be hard to design an algorithm to compute such estimations.

Our methodology uses random perturbations of the data and estimates the induced errors in the solutions. But we can only measure the perturbations in the computed solutions, including rounding errors. We vary the perturbation in the initial data in order to study the variation in the computed solution. In general, for very small data perturbations, the rounding errors are the most important and the data perturbations are negligible, whereas for sufficiently large perturbations the rounding errors become negligible. On the other hand, for sufficiently small data perturbations, the perturbations in the solutions can be approximated by using the condition number if the problem is regular.

2.1 Theoretical validation

We want to study the problem P in a neighborhood V of some initial data d_0. The first requirement is to get a unique solution, which is continuous.

Definition 2.1 *A problem is well-posed in V if it has a unique solution noted x and if this solution is continuous in V.*

The problem is solved by a direct algorithm. In other words, if the algorithm is executed with infinite precision, it gives the exact solution of the problem. To analyze the rounding errors, some assumptions on the computer arithmetic must be made [25, 11, 15]. We give here a simplified definition of numerical stability.

Definition 2.2 *We assume that the problem to solve is well-posed. Let ϵ be the machine precision and $x(\epsilon)$ be the approximate solution computed by an algorithm. This algorithm is numerically stable in V if*

$$\lim_{\epsilon->0} \|x(\epsilon) - x\| \quad = \quad 0 \ .$$

In most cases, this forward error can be obtained through a backward analysis [25].

We first prove that a stable algorithm allows to estimate the error in the exact solution by the error in the computed solution.

Proposition 2.1 *Let us assume that we solve a well-posed problem, non constant in V by an algorithm which is numerically stable. Let $V_\alpha = \{d \in V,\ \|d-d_0\| \leq \alpha\}$, $C_\alpha = \sup_{d\in V_\alpha} \|x - x_0\|$, and $C_\alpha(\epsilon) = \sup_{d\in V_\alpha} \|x(\epsilon) - x_0(\epsilon)\|$. There exists $\Delta_1(\epsilon)$ such that*

$$\lim_{\epsilon->0} \Delta_1(\epsilon) = 0, \quad (1)$$

$$\lim_{\epsilon->0} \sup_{\alpha\geq\Delta_1(\epsilon)} |\log(C_\alpha(\epsilon)/C_\alpha)| = 0. \quad (2)$$

Proof. It is sufficient to prove the following :

$$\forall\Delta_1 > 0, \quad \lim_{\epsilon->0} \sup_{\alpha\geq\Delta_1} |\log(C_\alpha(\epsilon)/C_\alpha)| = 0.$$

Writing :

$$x(\epsilon) - x_0(\epsilon) = (x - x_0) + (x(\epsilon) - x) - (x_0(\epsilon) - x_0),$$

we get, for any α small enough,

$$|C_\alpha(\epsilon) - C_\alpha| \leq 2 \sup_{d\in V} \|x(\epsilon) - x\|.$$

Now, since $\forall\alpha, \alpha \geq \Delta_1, C_\alpha \geq C_{\Delta_1} > 0$, we get :

$$\sup_{\alpha\geq\Delta_1} |C_\alpha(\epsilon)/C_\alpha - 1| \leq 2/C_{\Delta_1} \sup_{d\in V} \|x(\epsilon) - x\|,$$

which involves the result wanted thanks to numerical stability. □

It should be noted that since very small sizes of perturbations cannot be computed, the equality $C_\alpha(\epsilon) = 0$ always holds for small α.

Now we restrict the study to regular problems, according to the following definition :

Definition 2.3 *The problem is said q-regular if there exist C and q such that $C_\alpha = C\ \alpha^q + o(\alpha^q)$. The constants C and q are called respectively the condition number and the regularity of the problem.*

Example 2.1 *A problem expressed by $x = G(d)$ with G of class C^1 near d_0 is one-regular.*

Practically, our approach consists in finding the largest interval $[\Delta_1, \Delta_2]$ such that, in the interval, $C_\alpha(\epsilon) \neq 0$ and $\log(C_\alpha(\epsilon)) \simeq \log(C_\alpha) \simeq \log(C\ \alpha^q)$. This interval depends both on the numerical stability and the problem stability. If the precision is not sufficient, it may not exist. The proposition 2.1 leads readily to the following statement :

Proposition 2.2 *If there exists an interval $[\alpha_1(\epsilon), \alpha_2(\epsilon)]$ such that*

$$\forall \alpha \text{ with } \alpha_1(\epsilon) \leq \alpha \leq \alpha_2(\epsilon), \quad \log(C_\alpha(\epsilon)) \simeq \log(C_s\ \alpha^{q_s}), \tag{3}$$

if the problem is well-posed, non constant and if the algorithm is stable, then the problem is regular and estimations of the condition number C and the regularity q are given respectively by C_s and q_s. Moreover, the rounding errors are bounded by $C_{\alpha_1}(\epsilon)$.

On the other hand, if the algorithm is not stable, an interval $[\Delta_1, \Delta_2]$ where $\log(C_\alpha(\epsilon)) \simeq C_s\ \alpha_s^q$ but $\log(C_\alpha(\epsilon)) \not\simeq \log(C_\alpha)$ may exist. Such an example is provided in 2.5. In many cases, a residual can be directly related to a backward analysis of the algorithm. Hence, a small computed residual allows to conclude to small rounding errors. This is done in the method PRECISE [7]. However, this residual must be computed precisely enough. Moreover, such a residual may not be computable, for example in function evaluation of the form $x = F(y)$.

The method was described here for normwise perturbations and absolute condition numbers. However, it applies readily to componentwise perturbations or to relative condition numbers.

2.2 Use of random perturbations

We use the approach defined by [13] in the so-called SCALP methodology. It follows a statistical approximation generally used in econometry. Here, we apply a log-linear regression to $C_\alpha(\epsilon)$ in the interval where $log(C_\alpha(\epsilon))$ approximates $log(C_\alpha)$ and where C_α is approximated by $C\ \alpha^q$. This regression is simply the best least-squares fit to the experimental values $SC_\alpha(\epsilon)$ obtained as explained in the following. Before applying the regression, we look for its domain of validity bounded by α_1 and α_2 and we validate the fit a posteriori by a coefficient of significance which must be near 1.

We use relative componentwise data perturbation as in [7], defined by

Definition 2.4 *Let n be the dimension of the input space. The data perturbation of size α is defined by*

$$d_i \;=\; d_{0,i}\,(1 + e_i\;\alpha)\;\;\; i = 1 \cdots n,$$

where $e_i \;\; i = 1 \cdots n$, are random variables taken on the unit circle.

We use the following approximation to the worst-case :

Definition 2.5 *Let N be the number of samples generated for the data perturbation of size α. The approximation to the error bound $C_\alpha(\epsilon)$ is given by*

$$SC_\alpha(\epsilon) \;=\; \max_{j=1,\ldots,N} \|x_j(\epsilon) - x_0(\epsilon)\|$$

In practice, we use a size α of perturbations defined by $\alpha = 2^{-m}$, such that $\alpha \geq \epsilon$, allowing a large range of perturbations. For each size of perturbation, the software generates N computations where the data are randomly perturbed, yielding a file of samples. It should be noted that this file must be recorded in binary format in order to avoid rounding effects. Then we first plot all these points (all the error estimations versus the size of perturbations) in log-log scale in order to detect the domain of validity. Though this step needs currently manual intervention, it could be automated.

We look for a domain $\epsilon \leq \alpha_1 \leq \alpha \leq \alpha_2$ where $SC_\alpha(\epsilon)$ does not vanish and $\log(SC_\alpha(\epsilon))$ depends almost linearly on $\log(\alpha)$, with possibly $\alpha_1 = \epsilon$. In this domain, provided that the rounding errors are negligible, we estimate the condition number C and the regularity q by the regression

$$log(SC_\alpha(\epsilon)) \;\simeq\; log(C) \;+\; q log(\alpha)$$

Furthermore, the rounding errors $\|x_0(\epsilon) - x_0\|/\|x_0\|$ are bounded by $C\;\alpha_1^q$.

Matrix	order	Conditioning	Numerical Difficulty
TRIDIAG(10)	10	Well-Cond.	None
MATPIV(7)	7	Well-Cond.	Pivoting in Gauss
DESCALED(10)	10	Artificial Ill-Cond.	Balancing in QR
HILBERT (5 or 10)	5 or 10	Ill-Cond.	None

Table 1: Characteristics of the matrices tested

Matrix	Algorithm	Spectral Condition Number	Bauer-Skeel Condition Number	Estimated Condition Number	Estimated Regularity	Exact Error	Estimated Error
HILBERT(5)	Gauss single precision	$4\ 10^5$	$8\ 10^5$	$2\ 10^5$	1.1	$9\ 10^{-3}$	$4\ 10^{-2}$
HILBERT(5)	Gauss double precision	$4\ 10^5$	$8\ 10^5$	$3\ 10^5$	1	$3\ 10^{-12}$	$3\ 10^{-10}$
HILBERT(10)	Gauss double precision	10^{13}	$2\ 10^{13}$	10^{13}	1	$3\ 10^{-4}$	10^{-2}
TRIDIAG(10)	Gauss With Pivoting	3	4	2	1	0.	$4\ 10^{-7}$
MATPIV(7)	Gauss Without Pivoting	50	100	20	1	$4\ 10^{-2}$	10^{-1}
	Gauss With Pivoting	50	100	20	1	10^{-6}	$5\ 10^{-6}$
DESCALED(10)	QR Without Balancing	10^{12}	3	1	1	$6\ 10^{-4}$	$5\ 10^{-4}$
	QR With Balancing	10^{12}	3	1	1	$6\ 10^{-15}$	$3\ 10^{-15}$

Table 2: Estimations for linear systems.

2.3 Resolution of a linear system

The problem we deal with is to solve a dense non singular linear system.

Example 2.2 *Given a matrix A of order n and a right-hand side b, find x such that $A\ x\ =\ b$.*

This problem has been thoroughly studied and main results can be found for example in [22, 16]. The regularity is shown to be 1., and various condition numbers have been proposed. The most current used is $\|A^{-1}\|\ \|A\|$. However, it may in practice overestimate the actual error due to artificial ill-conditioning [22]. The Bauer-Skeel condition number, defined by $\||A^{-1}|\ |A|\|$, is a componentwise condition number independent of row scaling ; it is used in practice for sparse matrices [3] and in the library LAPACK [2].

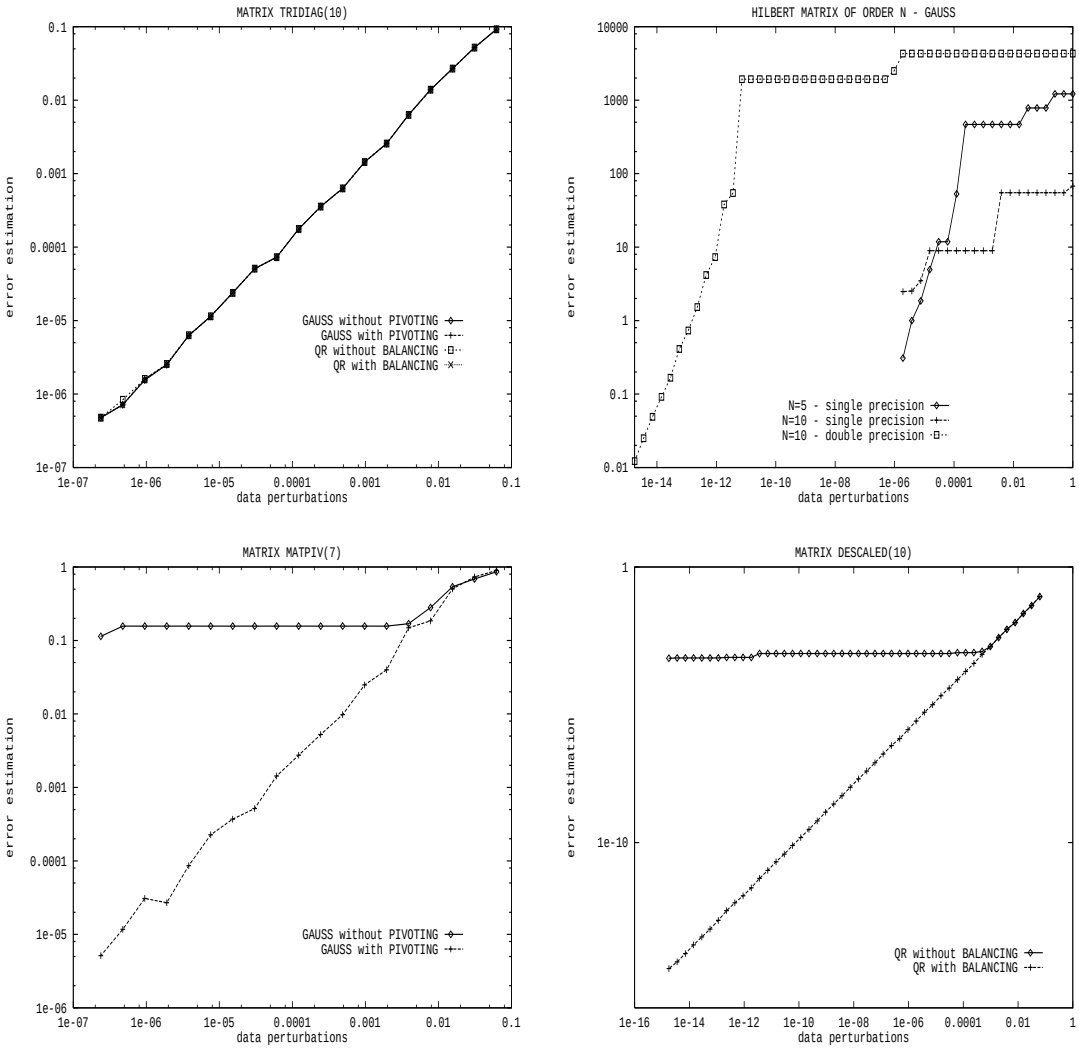

Figure 1: Results on linear systems.

We experiment with two algorithms to solve this linear system : Gaussian elimination with or without pivoting and QR factorization with or without balancing, described for example in [16].

To exhibit both numerical and problem instabilities, we use four different matrices, the features of which are summarized in table 1. For all these matrices, we have experimented the four algorithms in single and double precision, but we report here only the main significant results. We perturb both the right-hand side and the matrix, since perturbations of the right-hand side only may lead to an effectively well-conditioned linear system where the effective condition number is much smaller than the usual one [6]. We use infinite norms for the error bounds which are estimated by taking $N = 10$ samples. Data perturbations range from 10^{-7} to 1 in single precision, and from 10^{-15} to 1 in double precision. In order to compute the exact error, we start from a known solution x and derive the right-hand side b by simply computing exactly $b = A\ x$ (using a symbolic package). We have experimented with various right-hand sides, and report only those with $x = (1, 1, \ldots, 1)$.

The table 2 contains for each matrix the spectral condition number, the Bauer-Skeel condition number (computed using MATLAB), the exact error (comparing the result with the exact solution which is given), and the estimations of the regularity, the condition number and the error. In all cases, the coefficient of significance of the regression was roughly equal to 1. (in general 0.99). For ill-conditioned matrices or unstable algorithms, we first find the domain of validity to estimate the regularity and the condition number. The figure 1 plots for each matrix the error estimations for varying data perturbations, in a log-log scale. For all cases, the domain of validity corresponds to a straight line of slope 1.

For the matrix $TRIDIAG(10)$, the four algorithms are stable and the problem is very well-conditioned. The condition number of Hilbert matrices increases with the order of the matrix, so that the domain of validity disappears for the order 10 in single precision. Double precision allows to push further this barrier on the order, but higher orders cannot be handled. The matrix $MATPIV(7)$ requires pivoting in Gaussian elimination and we find this numerical instability as shown by the horizontal line. The same phenomenon is observed for QR factorization with and without balancing for the matrix $DESCALED(10)$. Furthermore, this matrix exhibits the phenomenon of artificial ill-conditioning, since the spectral condition number is much larger than the Bauer-Skeel one. In all cases, our estimations of the condition number are in good agreement with the theoretical Bauer-Skeel

Table 3: Description of Equations Tested

Equation	a	b	Regularity	Numerical Difficulty
1	0.3	0.02	1	None
2	1.00001 10^5	1	1	Small root x_1
3	0.2	0.01	0.5	None

Table 4: Results for second-order equations in double precision

Equation	Exact Condition Number	Estimated Condition Number	Estimated Regularity	Exact Error on x_1	Estimated Error on x_1
1	5	5	1	0	$4\ 10^{-15}$
2	2	2	1	$3\ 10^{-7}$	$7\ 10^{-7}$
3	1.7	1.7	0.5	$10-8$	$5\ 10^{-8}$
Equation	Exact Condition Number	Estimated Condition Number	Estimated Regularity	Exact Error on x_2	Estimated Error on x_2
1	4	4	1	0	$4\ 10^{-15}$
2	1	1	1	0	$8\ 10^{-16}$
3	1.7	1.7	0.5	$10-8$	$5\ 10^{-8}$

condition number. They are slightly lower, may be because we do not apply general enough perturbations on the matrix.

2.4 Example of a non linear equation

Here we take a simple example of a non linear problem with a rational regularity different from 1. which consists in solving a quadratic equation. This example has been studied thoroughly in [18, 15] for example.

Example 2.3 *Given the reals a and b, find the set of complex solution of the equation $x^2 - a\,x + b = 0$.*

For sake of simplicity and without loss of generality, the coefficient of x^2 is normalized. This problem has a regularity of 1 if it has two distinct

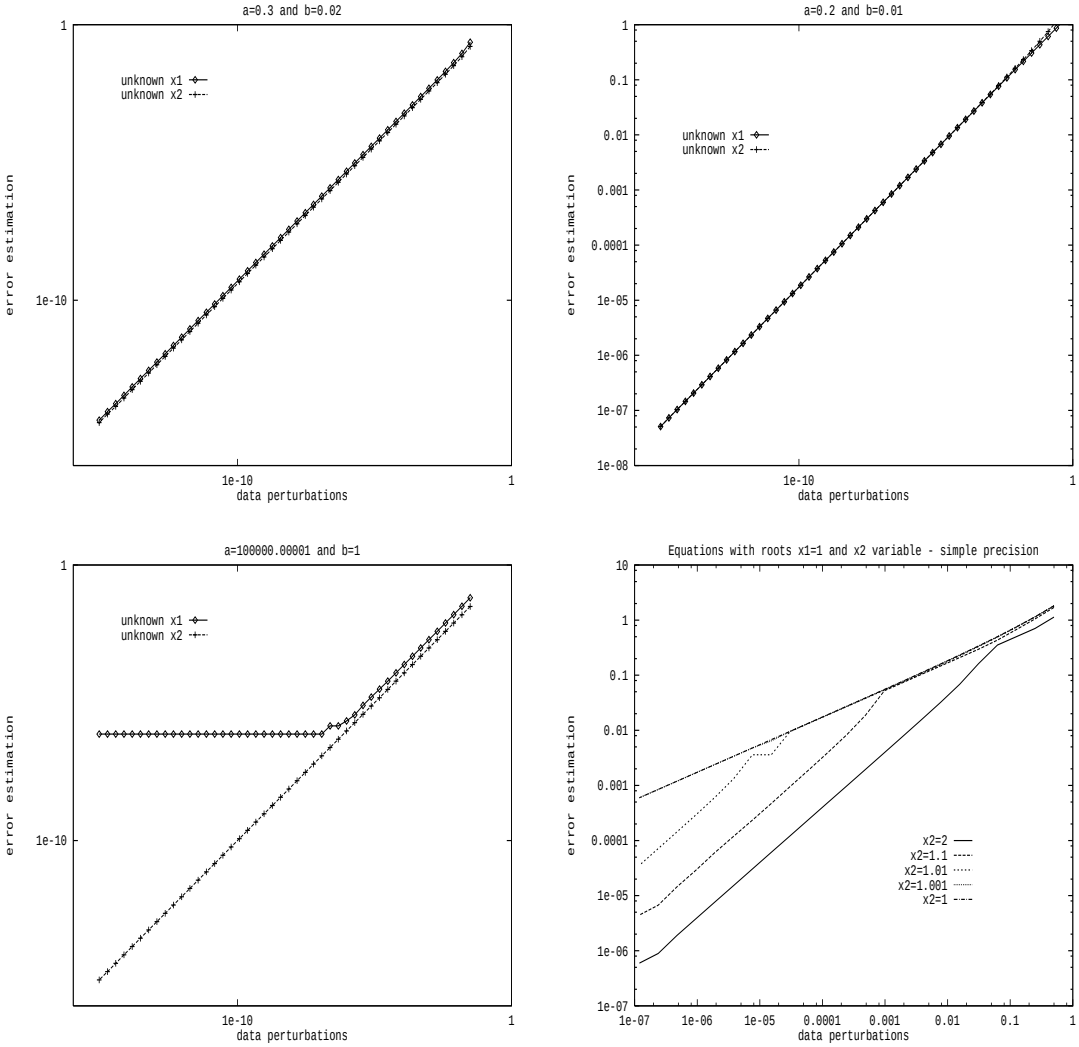

Figure 2: Results on quadratic equations.

solutions. On the other hand, it has a regularity of 0.5 if it has a double solution :

Proposition 2.3 *If x_1 and x_2 are two distinct solutions of the equation $x^2 - a\ x + b = 0$, then the problem has a regularity 1 with the relative condition numbers C_1 and C_2 corresponding to each solution given by $C_i = \frac{|a|\ |x_i| + |b|}{|x_i|\ |2x_i - a|}$ $i = 1, 2$. If x is a double solution of the same equation, then the problem has a regularity 0.5 with a relative condition number given by $C = \sqrt{3}$.*

Proof. If the equation has two solutions, then $2x_i - a \neq 0$ and we can write at the first order,

$$(2x - a)\Delta x - x\Delta a + \Delta b = 0$$

Applying it to perturbations of the form $\Delta a = \alpha\ a$ and $\Delta b = \beta\ b$ we obtain with $|\alpha| \leq \eta$ and $|\beta| \leq \eta$:

$$|\Delta x_i|/|x_i| \leq C_i\ \eta$$

which is reached for some α and β.

If the equation has one solution, then we must develop the perturbations :
$(\Delta x)^2 - \Delta a\ \Delta x - x\ \Delta a + \Delta b = 0$,
$(\Delta x - \Delta a/2)^2 = x\ \Delta a - \Delta b + (\Delta a)^2/4$.
Applying it to perturbations of the form $\Delta a = \alpha\ a$ and $\Delta b = \alpha\ b$ we obtain with $|\alpha| \leq \eta$ and $|\beta| \leq \eta$:
$\frac{\Delta x}{x} = \alpha \pm \sqrt{\alpha^2 + 2\alpha - \beta} \approx \sqrt{2\alpha - \beta}$ for small η,
$|\Delta x|/|x| \leq \sqrt{3}\ \sqrt{\eta}$
which is reached for some α and β.

□

The simplest algorithm to solve this quadratic equation is to compute the discriminant Δ and the solutions in the complex field by

$$\Delta = a^2 - 4b, \quad x_1 = (a - \sqrt{\Delta})/2, \quad x_2 = (a + \sqrt{\Delta})/2,$$

here, $\sqrt{\Delta}$ means $\sqrt{\Delta}$ if $\Delta \geq 0$ and $i\ \sqrt{|\Delta|}$ if $\Delta < 0$.

However, this algorithm is unstable to compute x_1 if this root is very small or in other words if $|b|$ is small compared to $|a|$. In any case, a stable algorithm to compute the two roots is given by

$$x_1 = 2b/(a + sign(a)\ \sqrt{\Delta}), \qquad x_2 = (a + sign(a)\ \sqrt{\Delta})/2.$$

We have experimented our methodology with three equations, defined in table 3. We study separately the two roots x_1 and x_2 in order to detect the numerical instability in the root x_1. The results of our methodology are plotted in figure 2 and summarized in table 4. The first case demonstrates numerical stability and well-conditioned problem. In the second case, the numerical instability in the root x_1 is illustrated by an horizontal line, showing that the perturbations could capture the rounding errors. In the third case, we clearly obtain a regularity of 0.5 with a condition number roughly equal to $\sqrt{3}$. In all cases, our results are in good agreement with the theory. If the roots are close, the curve shows a broken line with first a regularity equal to 1. then a regularity equal to 0.5. As the roots become closer, the line with a slope equal to 1. disappears and for very close roots, the computed regularity is equal to 0.5 as if they were equal (the two curves are identical). The same phenomenon is observed in [8] with the method PRECISE.

2.5 Example of an unstable algorithm

Here we illustrate the case of an unstable algorithm where the rounding errors are not captured by the perturbations by an example which was provided to us by W. Kahan [19] and which we slightly modified. The problem consists in evaluating a very simple function :

Example 2.4 *Given $y > 0$, find $x = y + 1$.*

However, the algorithm to compute x is rather sophisticated, as shown in figure 3. Due to rounding errors, the computed expression $Q(y)^2/80$ is not null but very small such that the computed $e(Q(y)^2/80)$ is null for most values of y. Consequently, the computed solution x is roughly equal to $y+y^2$ for most cases instead of $y + 1$, so that the computed condition number is false.

In that case, the absolute rounding errors are not negligible but are not captured by the estimated error. The results for three different values of

$$x \;=\; e(y, Q(y)^2/80) \;+\; y$$
$$Q(v) \;=\; |v - \sqrt{1+v^2}| \;-\; 1/(v + \sqrt{1+v^2})$$
$$e(t,u) \;: if\ (u = 0)\ \ e(t,u) = 1 \text{ else } e(t,u) = t^2$$

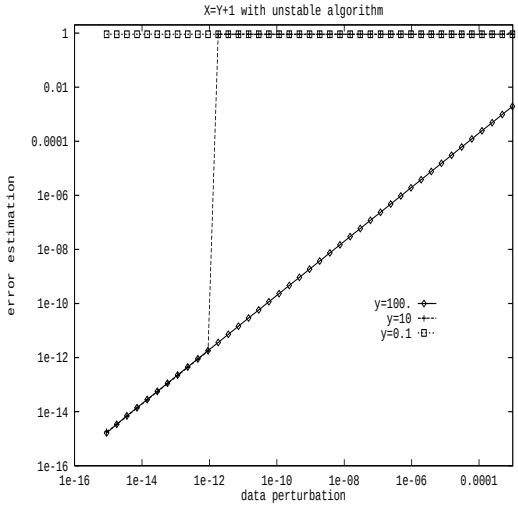

Figure 3: An example of rounding errors not captured by perturbations.

y are plotted in figure 3. For $y = 0.1$ some computed results are roughly equal to $y + y^2$ whereas others are roughly equal to y so that the rounding errors are captured and dominate in the estimated error. For $y = 100$ all the computed results are equal to $y + y^2$ so that the rounding errors seem small and the condition number estimation is false. We estimate in fact the condition number of the problem $x = y + y^2$ which is about 2. instead of 1. For $y = 10$ the numerical instability appears only for perturbations larger than 10^{-12}.

3 Conclusion

Problem conditioning and numerical stability combine together to deliver results with some inaccuracy. Tools based on random perturbations of the data are designed to study experimentally numerical stability and problem stability. Our approach is based on a log-linear regression of the error estimation versus the size of the perturbation and provides estimations of the condition number and the regularity of the problem. Experiments conducted so far show the effectiveness of the methodology for stable algorithms and for some unstable ones. Moreover, this tool is quite cheap since it does not require any special arithmetic, general and easy to use. It will be integrated in the next future into a toolbox, called Aquarels, which is currently under development [12]. This toolbox integrates into a consistent and user-friendly software structure tools contributing to improve or control the numerical quality of scientific software. Aquarels integrates in particular interval analysis, multiple precision and perturbation techniques and contains extensions to Fortran in order to apply easily these techniques. Various perturbations approaches can actually be easily included in this toolbox.

In practice the method is applied only to a specific part of a scientific code. It may be applied successively to more and more restricted parts, in order to isolate the difficulties. For such experiments, the choice of the input data to perturb and of the output data to analyze is relevant.

Acknowledgments

We thank M. Crouzeix and B. Philippe for their helpful suggestions for improving the paper. We are indebted to W. Kahan for providing a meaningful example of unstable algorithm and for valuable discussions. Judicious remarks from H. Stetter helped to clarify several points in the paper.

References

[1] American National Standard. *IEEE Standard for Binary Floating-Point Arithmetic.* ANSI/IEEE Std 754-1985, publié par IEEE, New-York.

[2] E. Anderson and al. *LAPACK Users' Guide.* SIAM, 1992.

[3] M. Arioli, J.W. Demmel, I.S. Duff. *Solving sparse linear systems with sparse backward error.* SIAM J. Matrix Anal. and Applics.

[4] B. Bliss, M.C. Brunet and E. Gallopoulos. *Automatic Program Instrumentation with Applications in Performance and Error Analysis*

[5] M.C. Brunet. *Contribution à la fiabilité de logiciels numériques et à l'analyse de leur comportement: une approche statistique.* Thèse de Doctorat de l'Université Paris-IX, Janvier 1989.

[6] T. Chan, D. Foulser. *Effectively Well-Conditioned Linear Systems.* SIAM J. Sci. Stat. Comput., vol. 9, no 6, November 1988, pp. 963-969.

[7] F. Chatelin, V. Frayssé. *Arithmetic Reliability of Algorithms* Symposium on High Performance Computers, Montpellier (France), October 1991.

[8] F. Chatelin, V. Frayssé. *Stability of nonlinear problems presenting algebraic singularities* CERFACS Research Report, TR/PA/91.

[9] F. Chatelin. *Résolution approchée d'équations sur ordinateurs.* Notes de cours de DEA de statistique, Université Paris VI, 1989.

[10] J.M. Chesneaux. *Modélisation et conditions de validité de la méthode CESTAC.* C.R. Acad. Sci. Paris, t.307, Série I, p. 417-422, 1988.

[11] J. Erhel. *Erreurs de calcul des ordinateurs* IRISA Report no 552, September 1990.

[12] J. Erhel, B. Philippe *AQUARELS : a Problem-Solving Environment for Numerical Quality* 13^{th} IMACS World Congress on Computation and Applied Mathematics, Dublin (Ireland), June 1991.

[13] P. François. *Contribution à l'étude de méthodes de contrôle automatique de l'erreur d'arrondi : la méthodologie SCALP.* Thèse de l'INPG, Décembre 89.

[14] P. François, J.M. Muller. *Faut-il faire confiance aux ordinateurs ?* Research Report LIP, 1990.

[15] D. Goldberg. *What every Computer Scientist Should Know about Floating-point Arithmetic.* ACM Computing Survey, 3, no 1, March 1991.

[16] G. H. Golub, C. F. Van Loan. *Matrix Computations.* The John Hopkins University press, 1983.

[17] N.J. Higham. *A Survey of Condition Number Estimation for Triangular matrices.* SIAM Review, 29, pp. 575-596, 1987.

[18] W. Kahan. *A Survey of Error Analysis.* Proceedings of the IFIP Congress'71, Information Processing, North-Holland, 2, pp. 1214-1239, 1972.

[19] W. Kahan. Private communication, July 1992.

[20] J.L. Larson, A.H. Sameh. *Algorithms for roundoff error analysis- A relative approach.* Computing, 24, pp. 275-297, 1980.

[21] W. Miller, D. Spooner. *Software for roundoff analysis, II.* ACM TOMS, 4, pp. 369-387, 1978.

[22] G. W. Stewart. *Introduction to Matrix Computations* Academic Press, 1973.

[23] G. W. Stewart, Ji-guang Sun. *Matrix Perturbation Theory* Academic Press, 1990.

[24] J. Vignes. *Review on stochastic approach to round-off error analysis and its applications.* Mathematics and Computers in Simulation, 30, pp.481-492, Special issue on stochastic methods and round-off error analysis, 1988.

[25] J.H. Wilkinson. *Rounding errors in algebraic processes* Englewoods Cliffs, N.J.: Prentice-Hall, 1963.