DO WE ALWAYS NEED TO SOLVE OUR PROBLEMS EXACTLY?

Miroslav Rozložník

1. SUMMARY

In this contribution we want to discuss several important and often neglected aspects, that one can frequently meet in scientific computing when solving real-world problems with mathematical models. It seems quite natural, that one should make an effort and should try to execute all the steps of this rather complex process in an optimal way. On the other hand, it is clear that almost every realistic situation is necessarily connected with some type of imperfection, error or inaccuracy. Therefore, in practice we often talk about data with noise, simplification of the model, approximations, discretization errors, algebraic error or round-off. But is then the optimality always desirable? Do we really want to spend time and resources to solve some subtask or subproblem to a high accuracy when it is not necessary? How accurate algorithms should we use? We consider a general approach developed originally in the context of numerical linear algebra and illustrate our ideas on the example of ill-posed or ill-conditioned problems in such applications as image deblurring or parameter extraction.

2. SCIENTIFIC COMPUTING AND MATHEMATICAL MODELING

Recent advances in computer technology, new software, hardware and services, and related instrumentation, as well as their latest applications, give rise to situation that scientific computing has become an indispensable part not only of investigation and research in various disciplines, but it also starts to be an integral part in the development of new products and technologies in industry and engineering. In the Wikipedia we can read:

*Computational science (or **scientific computing**) is the field of study concerned with constructing mathematical models and quantitative analysis techniques and using computers to analyze and solve scientific problems. In practical use, it is typically the application of computer simulation and other forms of computation to problems in various scientific disciplines. The field is distinct from computer science (the study of computation, computers and information processing). It is also different from theory and experiment, which are the traditional forms of science and engineering. The scientific computing approach is to gain understanding, mainly through the analysis of mathematical models implemented on computers. Scientists and engineers develop computer programs, application software, that model systems being studied and run these programs with various sets of input parameters. Typically, these models require massive amounts of calculations (usually floating-point) and are often executed on supercomputers or distributed computing platforms.*

*A **mathematical model** is a description of a system using mathematical concepts and language. The process of developing a mathematical model is termed **mathematical modeling**. Mathematical models are used not only in the natural sciences (such as physics, biology, earth science, meteorology) and engineering disciplines (e.g. computer science, artificial intelligence), but also in the social sciences (such as economics, psychology, sociology and political science); physicists, engineers, statisticians, operations research analysts and economists use mathematical models most extensively. In many cases, the quality of a scientific field depends on how well the mathematical models developed on the theoretical side agree with results of repeatable experiments. Lack of agreement between theoretical mathematical models and experimental measurements often leads to important advances as better theories are developed.*

Mathematical modeling of natural phenomena and technological processes, thus clearly plays a fundamental role in scientific computing. As it was already noted it combines various tools from engineering and applications, applied mathematics, numerical analysis, matrix computations and parallel computing. Based on the used mathematical formalism and appropriate abstract structures, recent mathematical models can have many different forms, including but not limited to integro-differential equations, differential algebraic equations or discrete problems. Typically, mathematical models result into floating point calculations that are executed on modern distributed computer architectures.

Such modeling process can be seen as a sequence of several steps or subtasks that starts with certain experimental observation or measurement, and leads to the appropriate mathematical description, often involving a system of equations with a number of unknowns representing all the important parameters from the experiment. This continuous model is further reformulated and approximated with a discretization technique resulting into discrete (usually algebraic) problem. This system is finally solved by a suitable linear algebra or optimization method, that is properly implemented on a chosen computer architecture. Can we however give a solid theoretical justification of such mathematical models?

The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct, which with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work—that is, correctly to describe phenomena from a reasonably wide area.

— John von Neumann, [1]

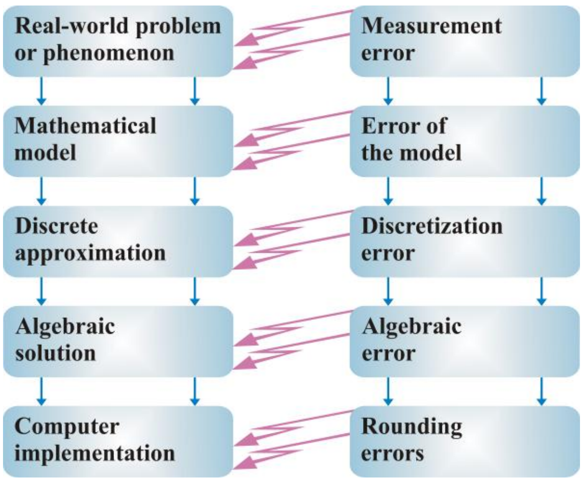

Figure 1.
General scheme of mathematical model and associated errors

3. MATHEMATICAL MODEL AND ASSOCIATED ERRORS

It seems quite natural that one should make an effort and try to execute all steps of this rather complex process in an optimal way. On the other hand, it is clear that for the most realistic situations the optimality is not achievable and each step is necessarily accompanied with some associated errors. This can be illustrated on each level starting from the experiment up to the implementation in the particular computer environment (Figure 1). The general approach we will present here has been studied and advocated by J. Liesen and Z. Strakoš. It is based on the concept of the backward error analysis developed and popularized in the context of numerical linear algebra by J.H. Wilkinson. For details we refer to his books [2,3], the papers [4,5,6] and the recent book [7].

No measuring instrument is perfect. The notion of the accurate measurement is just an idealization, all measurements lead to the approximations of the investigated quantities. Therefore the values we obtain from the experiment can be considered only as the estimates of their actual values. To examine whether these estimates are realistic we can examine the size of the **measurement error.** It is defined as the difference between the actual value and the observed value of the given quantity. The measurement error is determined by the accuracy limits of the instrument, but also by the level of the noise affecting the measured quantity. By the noise we consider random data, signal without meaning or unwanted information produced as a by-product of activities, which are irrelevant to the studied quantity or measured signal. The next step is the construction of the mathematical model. The mathematical description is again coupled only with some approximation of the observed phenomenon or technological process. The notions such as perfect model or exact representation correspond again only to the ideal situation. In practice, the lack of information we have leads to mathematical description, which is a representation of the phenomenon that is not complete, but still close enough to be useful for understanding the real-world problem. Given the description, the mathematical problem we have is often still too complex to be solved analytically and it must be simplified or approximated by a problem which is easier to solve by the available tools. Since we have only the approximation to the more complex problem, we can talk about the **error of the model.** Many mathematical models lead to the solution of various types of equations, often continuous and with a large number of unknowns and parameters. Since the fully analytic approach is possible only for a limited number of problems, the continuous model is usually further reformulated and approximated by some discretization technique resulting into a discrete problem. This approximation necessarily gives rise to the error on the next level, called the **discretization error**. Discretization techniques lead in many cases to large discrete problems. Some of them may be formulated incorrectly without having a unique solution, and even if their exact solutions exist and are unique, they cannot be found easily even by very efficient solution techniques. Then just an approximation to the exact solution can be a reasonably good option and its choice may reduce significantly the complexity of the solution process. Many iterative and approximation algorithms delivering a sequence of reasonably good approximate solutions in a reasonably short time have been developed in last decades. The quality of such approximate solution can be measured in terms of its distance to the exact solution of the discrete (algebraic) problem, called the **algebraic error** or the **approximation error.** Finally, the solution algorithms are implemented and executed on a modern computer architecture that is often distributed and enables parallel processing. However, arithmetic operations are performed in finite precision arithmetic and this leads to **rounding errors.** Rounding errors and their accumulation during the computations can have a significant effect on the computed quantities and lead to further approximation to the already approximated solution of the algebraic problem. Thus on each level we always work only with approximations. When we can expect that the whole process is still meaningful?

…all models are approximations. Essentially, all models are wrong, but some are useful. However, the approximate nature of the model must always be borne in mind…

— George Edward Pelham Box, [8]

4. IS THE OPTIMALITY ALWAYS DESIRABLE?

As already noted, one must always take into account that each consecutive subproblem is a partial result from the approximation on the previous level. Although this observation seems quite trivial, there are several examples, when it has been neglected and the solution on the next (and especially on the final) level has been performed without the use of information from preceding steps. We can illustrate our point on the applications, where the mathematical description and its further simplification lead to the ill-posed or ill-conditioned system of linear algebraic equations. By the **ill-posed problem** we mean a problem, that either it does not have a classical solution in a standard class of possible solutions, its solution is not unique or it is very sensitive with respect to the change of input data (arbitrarily small errors in the data can lead to infinitely large error in the solution), while the **ill-conditioned problem** still yields a unique solution, which depends continuously on the input data, but the measure of sensitivity of its solution (the condition number) is fairly large. Discrete ill-posed problems arise in connection with the numerical solution of inverse problems such as image restoration and tomography. A typical application is image deblurring problem [9,10], where one wants to recover the original image from the image blurred by the noise. The level of this noise has a significant effect on the size of the details in the reconstructed image and therefore it is important to develop reliable algorithms to recover as much as possible information from the recorded data. A relatively very simple linear model, that connects the recorded blurred image with the desired sharp image can be derived under some simplification of the original mathematical description. For details we refer to Chapter 1 of the book [10]. For the first sight, one might think, that the reasonably accurate solution of this ill-conditioned system of linear equations would lead to the desired result. But even if we perform this computation with the reliable algorithm (such as Gaussian elimination), this so called naïve approach completely fails and the computed image is dominated by the rounding errors and by the errors (noise) in the blurred image. It would seem then natural to replace the Gaussian elimination and try to solve the linear system without rounding errors (we say, in exact arithmetic), but such approach will not significantly help and the result will remain qualitatively the same: it is still dominated by the contribution of the noise to the solution of the linear system. Apparently, it makes no sense to compute the optimal (exact) solution to the problem which is ill-posed (or ill-conditioned) and to insist on its original formulation, where we know that a small change in the input data (system matrix and right-hand side vector) can lead to large change in the output data (solution of the system). Do we really want to waste time and resources to solve the (sub)problem to a high accuracy when it is no necessary? One must rather try to introduce an additional information to the problem (called penalty, restriction or filter in various contexts) that defines the subclass of potential solutions and to use some from very popular regularization techniques leading to “more regular” problems; see e.g. the material in books [9,10].

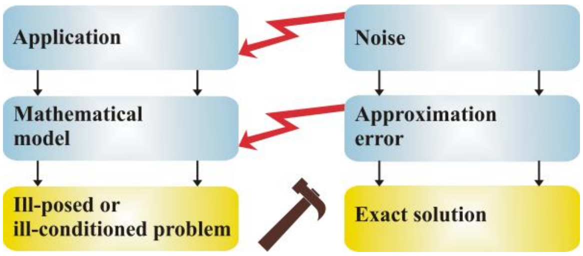

Figure 2.
Example of application leading to the ill-posed or to the ill-conditioned linear problem

Most problems involve inexact input data and obtaining a highly accurate solution to an imprecise problem may not be justified..

— J.J. Dongarra, J.R. Bunch, C.B. Moler and G.W. Stewart, [11]

5. BACKWARD ERROR ANALYSIS AND STABLE ALGORITHMS

Although the effects of rounding errors have been investigated in many areas, the concept of the backward error analysis has been thoroughly developed only in several fields; one of them is certainly numerical linear algebra with the pioneering work of Wilkinson and others (see the references [2,3,4]). If we are interested in the error of the approximate solution computed in finite precision arithmetic, then it is useful to interpret the actual rounding errors that arise during the computation using the (small) perturbations of the input data. Such backward error analysis allows to reduce the question on the error of the approximate solution to the question on the sensitivity of the output data with respect to the (small) changes of the initial input data. It follows then, that we can separate the description of a numerical behavior of the algorithm in finite precision arithmetic (its numerical stability) from the perturbation theory, that studies the sensitivity of the given problem.Therefore the backward error analysis enables to identify the cases, when the large error of the approximate solution is caused by the improper choice of the algorithm (its unstability) and when it is just an inevitable consequence of bad properties of the problem itself. In the case of the image deblurring, it is evident that the large error is due to the ill-conditioning of the system.

The chosen algorithm is sufficiently stable (or in the other words sufficiently reliable) for solving the given problem, if the error of the computed approximate solution can be projected into perturbations of the input data on the same or on the smaller level than the level of their inaccuracy or uncertainty. Again, it makes no sense to spend more computational time and resources and to solve the problem to a higher accuracy than it is necessary. The only criterion for the choice of the algorithm is then the (parallel) efficiency of its implementation on the given computer architecture.

There does seem to be some misunderstanding about the purpose of an a priori backward error analysis. The main purpose of such an analysis is either to establish the essential numerical stability of an algorithm or to show why it is unstable and in doing so to expose what sort of change is necessary to make it stable. The precise error bound is not of great importance.

— James Hardy Wilkinson, [12]

6. ACKNOWLEDGEMENT

This work is supported by Grant Agency of the Academy of Sciences of the Czech Republic under the project IAA100300802 and by the Institutional Research Plan AV0Z10300504 “Computer Science for the Information Society: Models, Algorithms, Applications“.

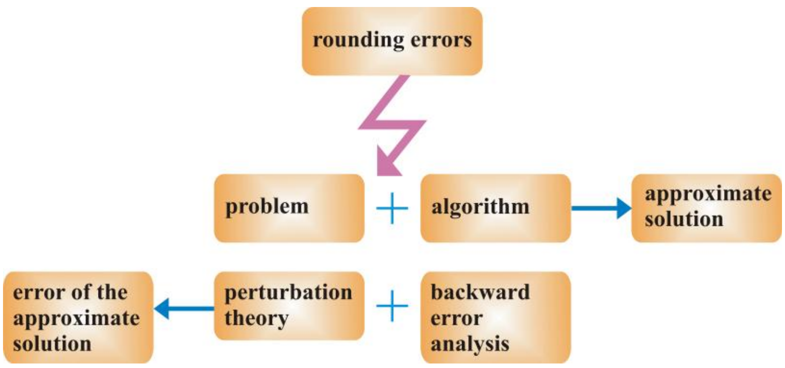

Figure 3.
Schematic description for the estimation of the error in numerical linear algebra

7. LIST OF REFERENCES

[1] J. von Neumann, '*Method in the Physical Sciences'*, in *The Unity of Knowledge*, editted by L. Leary (1955), 158. Reprinted in John von Neumann, F. Bródy (ed.) and Tibor Vámos (ed.), *The Neumann Compendium* (2000), 628.

[2] J. H. Wilkinson, *Rounding errors in algebraic processes*, Prentice-Hall Inc., 1963.

[3] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK, 1965.

[4] J. von Neumann and H. H. Goldstine. *Numerical inverting of matrices of high order.* Bull. Amer. Math. Soc. Volume 53, Number 11 (1947), 1021-1099.

[5] Z. Strakoš and J. Liesen. *On Numerical Stability in Large Scale Linear Algebraic Computations*, Zeitschrift fuer Angewandte Mathematik und Mechanik, 85, 5 (2005), 307-325.

[6] J. Liesen and Z. Strakoš. *On the interplay between the PDE discretization and numerical solution of the resulting algebraic problems*, submitted to Foundations of Computational Mathematics, 2011.

[7] J. Liesen and Z. Strakoš. *Krylov Subspace methods. Principles and Analysis..* Oxford University Press, to appear.

[8] G.E.P. Box and N.R. Draper. *Empirical Model-Building and Response Surfaces*, Wiley, 1987.

[9] P.C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion,* SIAM Monographs on Mathematical Modeling and Computation 4, SIAM, Philadelphia, 1998.

[10] P.C. Hansen, J.G. Nagy and D.P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering,* Fundamentals of Algorithms, SIAM, Philadelphia, 2006.

[11] J.J. Dongarra, J.R. Bunch, C.B. Moler and G.W. Stewart. *LINPACK Users' Guide,* SIAM, Philadelphia, 1977.

[12] J.H. Wilkinson. Numerical Linear Algebra on Digital Computers. IMA Bulletin 10 (1974), 354-356.