(Preprint for)

J. Rojo

The Package NUMERIAL

Mathematica World - Education Today (1994)

THE PACKAGE `NUMERIAL` FOR EDUCATIONAL PURPOSES

Jesus ROJO

E.T.S. de Ingenieros Industriales

Valladolid (SPAIN)

Abstract

A Package of general utility with interesting and specific applications for educational tasks is presented. It is in fact a superb aid for teaching Algebraic Numerical Methods. In this work, the use of the Package is described. The operating manual can be found in the Package itself and in the file of Documentation that accompanies it.

1 The aim of the Package.

When starting on practice in Numerical Analysis, one of the problems is to achieve 'hands on' practice of what happens in a computer. In particular, it is difficult, in this field, to perform what the computer does at every instant: many operations, the accompaniment of rounding-off errors, the accumulation of such errors, etc. The problem is that to do all this by hand in the same way and in a reasonable time span is impossible. The reasons are twofold: a person (without a computer) handles the binary base very badly and it is impossible for him to use 32, 64 or 128 binary digits without error.

The solution is to do all this with two changes that do not affect the essence: only a few digits are handled (two or three, for instance) and everything is studied using a decimal base (which people handle well from their infance).

What we have done is to have the computer perform the sequence of operations in the same manner, with several possible motives: to create examples, to check by machine what has been done by hand, etc.

Although what we propose is of interest in all fields of Numerical Analysis, the Direct Methods for solving the linear systems are its ideal place of application. In fact, here as in no other the chain of small errors in the operations leads to results very far from the theoretical ones. In fact techniques such as pivoting to minimize these errors are studied early on, whereas in other problems they are only studied at an advanced level.

Thus, our task in the Packages that we present has been a double one: to create a routine that is capable of operating while leaving the result of each operation rounded off to `n` significant decimal digits, and to present this alternative in the Direct Methods for Linear Algebra.

2 Mathematica and the Rounding.

In the traditional programming languages, the implementation of what we wish to do was almost impossible. They are languages set up so that the computer does everything it can, as far as precision is concerned (depending, of course, on the exactitude with which it works). That is why an artificial limitation is so difficult. The rounding-off error can easily reach the order of $10 * *(-12)$, or something similar, but it is no simple matter to achieve an error to the order of $10 * *(-3)$. Besides, the important thing is not that the machine shows 2 or 3 figures on the screen, but that it should work with them internally.

However Mathematica allows us to reach our goal. This is due, fundamentally, to the existence (together with Numerical Routines) of routines for the Symbolic Calcul, which are as such exact. The only problem it has is that the rounding-off, which is finally performed through the function `Round`, does so in a slightly unusual way. Thus, `Round[1.5]` is `1` and not `2`, as it traditional.

Our first (logical) attempt was with the function `MantissaExponent`. But this function is of 'numerical' type and in the end could not serve our purpose. The function that allowed us to perform the rounding-off was the `RealDigits` which works in any base and does so with exact results.

In this way we managed to create the `Rounding` function, which rounds off, with base 10, any real number `a` to `prec` significant decimal places:

```
Rounding[a_,prec_Integer]:=
Module[{b,b1,b2,b3,i},
```

```
If[a==0,Goto[fin],];
b=Sign[a]*N[a];
b1=RealDigits[b][[1]];
b2=RealDigits[b][[2]];
If[b1[[prec+1]]>=5,b1[[prec]]++];
b3=0;
Do[
b3=b3+10^(-i)*b1[[i]]
,{i,1,prec}];
Label[fin];
If[a==0,0.,Sign[a]*N[b3*10^b2]]
]
```

Naturally, the `Module` is neither the ideal nor the most efficient (to be so it should be defined as a pure function), which is rather far from our purpose, but more than sufficient. To increase the velocity, the ideal thing would be for the creators of Mathematica to correct them, to adapt them for any base (not only 10) and to introduce them into C in the future Kernel. Because of the Packages' structures, we have called this function `CORounding`.

For example, if we wish to perform the operation `a+b` and to store the result `c` to be used and rounded-off to three significant decimal digits, what we should do is

```
c = CORounding[a+b,3] .
```

This function can be used in the rounding-off of any operation in Numerical Analysis. We have done it in the sine of the Direct Routines for Linear Algebra.

3 Direct Methods for Linear Algebra.

The Package `NUMERIAL` is made up of the methods LU and Choleski, implemented in practically all their varieties. The methods are performed with a matrix `A` and a independent term `b` both arbitrary (except in the usual imposition of these methods). The method LU is performed with or without partial pivoting, that is exchange of rows. The use of partial double precision is possible but not obligatory, which in our case means that the number of significant decimal digits is double that of the established `prec` number. In fact, it is an option similar to that which computers perform. As is well-known, this partial double precision allow us to work in double precision at the key moments without amplifying the memory reserve of the matrices.

Another possible option is the iterative refinement or iterative improvement of the solution, an option both usual and cheap of the direct methods when we want to enhance the solution. In the refinement, the operations that have to be performed with most precision can also be done in double precision (with the same meaning as before). In this case, as is well-known, there is no need for an increase in the memory reserve either. The refinement can be carried out in two modalities, a necessary number of refinements or iterative refinements until the difference between the solutions obtained is (in norm) less than a quantity (tolerance). The most usual norms are used.

The idea, in all the methods, is the same: in each elementary operation (addition, subtraction, multiplication, division or square root), the result is rounded-off to `prec` significant decimal digits before continuing. The rounding-off is carried out using the above mentioned `Rounding` function. It is natural that the repeted use of this function slows down the execution of the program; this is a small price to pay for using pedagogical routines.

The functions which has this Package are as follows:

```
ALFactor[A,b,n,prec,lu,pi,pp],
ALCholeski[A,b,n,prec,pp],
ALFactorwithRefinement[A,b,n,prec,lu,pi,no,tol,pp,pr],
ALCholeskiwithRefinement[A,b,n,prec,no,tol,pp,pr],
ALFactorwithnRefinements[A,b,n,prec,lu,pi,nref,pp,pr],
ALCholeskiwithnRefinements[A,b,n,prec,nref,pp,pr],
```

The majority of the variables (`lu, pi, pp, no, pr`) are strings which dictate the explained variants that can be used. The initials ‘AL’ are there to enable an easier grouping of the routines of this Package.

As is well-known, Mathematica is an interpreted and interactive language. This allows us to compile and interrupt the routines by performing steps or intermediate comparisons. All the routines leave a series of variables at the `Global` context level to the end. On finishing a routine one can work with these variables. Doing it in this way means we conserve more elements at our disposal without the necessity to increase the number of arguments transmitted to the functions. Naturally, and so as not to interfere in the general progress of a program, the rest of the variables have been introduced as local variables of the `Module`.

4 Other Packages.

Although, for the sake of brevity, we have included no other Packages than `NUMERIAL` and the `NUMERICO` Package it needs, we have other similar Packages, more or less developed, organized according to the same system. For example, `NUMERIIN` performs numerical integrations, or `NUMERIRT` calculates the roots of scalar functions. The Package we present, however, is the one that has the greatest pedagogical content.

The existence of other Packages is why we have not included `NUMERICO` in `NUMERIAL`, since `NUMERICO` has routines that all the other Packages need.

5 Bibliography.

The basic subject is so general that references include almost all the books on Numerical Analysis. Here we will mention only the following:

BURDEN, R.L. and FAIRES, J.D.: Numerical Analysis (third edition). Prindle, Weber & Schmidt (1985)

ELDEN, L. and WITTMEYER-KOCH, L.: Numerical Analysis. An Introduction. Academic Press (1990)

SCHWARZ, H.R.: Numerical Analysis. A Comprehensive Introduction. Wiley (1989)

FORSYTHE, G.E. and MOLER, C.B.: Computer Solution of Linear Algebraic Systems. Prentice-Hall (1967)

as they are the ones we use habitually in our engineering courses. The theme of rounding-off can also be found in the first chapters of these books.

Jesus ROJO e-mail: jesroj@wmatem.eis.uva.es