# On the Comparison of Numerical Methods

*Ch.Hafner, Swiss Federal Institute of Technology, CH-8092 Zurich, Switzerland*

**Abstract**

Serious comparisons of numerical methods are important for scientists who develop new codes as well as for those who use programs. Historical considerations show some errors which were made in the past and should be avoided in the future. Every numerical code is based not only on numerical but also on analytical considerations. Both of them have to be taken into account. As a result, benchmarks for complicated topics (numerical calculations of electromagnetic fields) should give more informations than just numbers like 'speed', 'memory requirement', etc.

## Historical considerations

In the early times of numerical calculations many codes were implemented and tested. They were all based on a mixture of ideas and were usually named after the most important idea. Several people believed that their method was the best and many of them tried to show this 'analytically'. But very often the mathematical knowledge of engineers dealing with numerical techniques was insufficient and sometimes the questions which arose simply could not be answered by analytical considerations. To overcome such problems the application of physical knowledge was helpful in many cases. For example, the simple Point Matching (PM) technique was used together with 'circular harmonic' analysis (Rayleigh hypothesis) in the sixties. Different people 'showed analytically' that PM could be used only for circular domains while non circular problems were solved sucessfully. Others claimed that only 'single valued' boundaries should be admitted, that 'non integer orders' should be used and so on. In fact, Russian mathematicians [1] had already given a strong analytical basis for this technique years ago. The experience with elaborated PM programs [2] made clear that the method failed in complicated cases for numerical reasons (bad convergence, under- and overflows, cancellations). The treatment of numerical details seemed to be even more important than the main ideas (This might be true for any numerical method.). At that time, the PM had already been pushed aside by the Method of Moments (MM) which seemed to have a more physical basis. The consequent improvement of both the analytical and numerical parts of the PM led to the Multiple MultiPole (MMP) method (better known in the USA as SPEX (SPherical wave EXpansion) for 3D scattering [3]) which has been sucessfully applied to various problems of 2D and 3D, static and dynamic, scattering and guided waves etc. [4].

A second approach to show the superiority of a method lies in a generalization which allows to consider 'concurrents' as special cases of the 'favorite'. For example if a Projection Technique (PT) is used (like in the MM) the PM is a special case if Dirac functions are used as testing functions. Since Dirac functions are known to be

simple but otherwise not very good testing functions, the superiority of PT over PM seems to be clear. The best choice of testing functions are the expansion functions themselves (Galerkin). The disadvantage of this choice is the occurance of integrals (scalar products of functions) which usually have to be solved numerically. On the other hand, it has been shown [4] that the generalization of PM (overdetermined systems of weighted equations) leads to the same results as PT with Galerkin's method if an adequate weighting is chosen. This means that generalized PM is superior to PT because it avoids time consuming numerical integrations without lack of quality.

A third attempt in knocking out concurrents consisted of specially tailored numerical 'comparisons': Elaborated forms of the 'favorite' were compared with weak forms of the 'concurrents'. Examples which were known to be easily solvable by the 'favorite' have been preferred and so on.

It may be surprising but the described proceeding was successful in many cases. As a result, only a few methods survived, or, more precisely, the names of only a few methods survived, because sometimes people saved their programs by simply renaming them. This is of course a legitimate consequence of the generalization of the methods. For the users it has, however, become very difficult to see which program is most suitable for their problems. Codes with the same name may differ one from another much more than codes with different names.

In the last years it has been recognized that every elaborated program has its own advantages and disadvantages too. The hope of finding 'the one and only' method seems to disappear because the implementation of huge programs show a very annoying effect: The probability of errors and the possibility of generating new errors while attempting to eliminate other errors increase with the length of the code. Though it has become possible to reanimate old ideas, to combine different methods, to look for new directions, and to compare numerical programs seriously. The question is now: How should such a comparison look like? It seems to be clear that both analytical and numerical considerations are important and that the 'dirty tricks' mentionned above should be avoided.

### Analytical considerations

If a numerical program is designed, various considerations show different ways and choices are necessary to get a certain code. The program usually gets a name which does not say anything about these choices and the details of implementation, but such informations are very important and should be known for serious comparisons. In most of the methods for calculating electromagnetic fields, the following choices are important:

1.) Field equations: Maxwell's equations in differential or integral form, wave equations, Helmholtz equations, variational integrals, other integral equations etc.

2.) Continuity/boundary equations: Continuity of certain components of the electric or magnetic field, potentials, derivatives of potentials, current density, energy

flow etc.

3.) Primary functions (functions which are approximated directly by a series of expansion functions): Certain field components, potentials, charges, currents, energy densities etc.

4.) Type of expansion functions: Dirac functions, piecewise constant functions, linear functions, continuous functions, differentiable functions, solutions of the field equations, Green functions, etc.

5.) Method to get a system of equations for the unknown parameters in the expansions: Error Method (EM), PT, PM etc. (The EM defines a certain error which has to be minimized.) Since different forms of these methods exist some more information is necessary here: Definition of the error (EM), the scalar product and the testing functions (PT), the weighting (generalized PM) etc.

6.) Algorithm to solve the system of equations: Gauss, Cholesky, least squares, iterative methods etc.

In practice, different approaches sometimes lead to identical solutions. Good codes should usually be derivable with different approaches. For example, the MMP programs were designed with this intention. They use analytical solutions of the field equations (This means that automatically different forms of field equations may be used with identical results.). These solutions are continuous, differentiable Green functions. Usually but not necessarily the continuity of all field components is used and certain field components are chosen as primary functions. The system of equations in the MMP programs may be derived with EM, PT or generalized PM and is solved by fast updating routines using Given's plane rotations.

### Numerical considerations

To get information about the speed and application range of numerical programs, numerical comparison standards seem to be useful. The problem of benchmarks for computers is already well known: Tests which give one single number which represents the 'speed' of a machine are misleading in many cases. For example benchmarks show that an 80286 based AT personal computer is much faster than an 8088 based XT. The MMP programs have been implemented and tested on such machines. It has been shown [5] that these programs run faster on an XT (with 8MHz clock) than on an AT (with 8MHz clock) for not too small problems. Comparisons of the MMP programs on very different machines showed astonishing effects which usually depended on the size and type of the considered problem. On the other hand it may be expected that the comparison of different programs will depend not only on the problem to be calculated but also on the machine and on the compiler which is used.

In fact 'speed' is not the only important feature of a program. Users usually want to get a program on the machine they already have. New and faster machines very often require time consuming adaptions of the codes. For these reasons it is

essential that a program is written in a way that it may be easily implemented on other machines (which even might not exist when the program is created). From this follows that common languages should be preferred and special features of a compiler on a certain machine should be ignored.

At present, very often two different types of examples are proposed as benchmarks for numerical field calculations:

1.) Examples which have a well known 'analytic' solution. They allow a calculation of the errors made by the considered program. Usually such examples can be treated by new and not yet highly sophisticated techniques. Because they are not really difficult to solve (otherwise an analytic solution would not be known) they give not much information on the application range.

2.) Examples without an 'analytic' solution which are well known from measurements and from calculations with mature programs. It is clear that such examples are much more important from a practical point of view. But they may be crucial for newly born codes. For this reason, they are preferred by some established scientists who try to defend their methods from being threatened by new promising ones.

To avoid benchmarks which result in meaningless numbers, a serious comparison of numerical calculations must be based on very different examples and should give much more information ('choices' of the method, application range, used machines, compilers, memory requirements, error checks, input/output support etc.) then just some numbers. Testing examples should include 'analytically' solved problems as well as 'practically well known' problems. They should include various configurations, both 2D and 3D geometries, open and closed structures, harmonic and impulsive time dependencies, scattering and eigenvalue problems, perfect and imperfect conductors, lossless and lossy dielectrics, etc. Of course, no program will be able to solve all the testing examples effectively. But this will just give the desired information for the users.

## References


[1] I.N.Vekua: *New Methods for solving Elliptic equations*, North-Holland Publ. Comp., Amsterdam, 1967.

[2] Ch.Hafner: SFIT thesis 6683, Zurich, 1980.

[3] A.C.Ludwig: IEEE Trans. AP 34, NO 34, 1986.

[4] Ch.Hafner: *Numerische Berechnung elektromagnetischer Felder*, Springer, Berlin, 1987.

[5] Ch.Hafner: URSI Radio Science Meeting, Syracuse, 1988.