A COMPUTATION EVALUATION OF SOME SOFTWARE FOR MATHEMATICAL PROGRAMMING

Themistoklis Glavelis
PhD Candidate, Department of Applied Informatics, University of Macedonia
156 Egnatia Str., 54006 Thessaloniki, Greece

Nikolaos Ploskas
PhD Candidate, Department of Applied Informatics, University of Macedonia
156 Egnatia Str., 54006 Thessaloniki, Greece

Nikolaos Samaras
Assistant Professor, Department of Applied Informatics, University of Macedonia
156 Egnatia Str., 54006 Thessaloniki, Greece

ABSTRACT

The purpose of this paper is to present some of the most well-known commercial and open source software for mathematical computer programming. Specifically, the commercial programs which will be presented are Maple, Mathematica and Matlab and the open source software are Octave and Scilab. These software packages are mainly used by scientists and software engineers. The strengths and weaknesses of these programs are detailed described through the chapters of this paper. Finally, there is a speed comparison of some frequently used algorithms, procedures and operations which are build-in functions of these mathematical programming environments.

KEYWORDS

Scientific Computing, Computational Comparison, Mathematical Software.

1. INTRODUCTION

The aim of this paper is to present a description and comparison of some of the most popular mathematical software. Maple (http://www.maplesoft.com/), Mathematica (http://www.wolfram.com/) and Matlab (http://www.mathworks.com/) are selected from other commercial software due to their wide spread among the scientific community and industry. Furthermore, they all belong to the leaders of software enterprises in mathematical programming. Moreover, Octave (http://www.gnu.org/software/octave/) and Scilab (http://www.scilab.org/) are picked because they are the most well-known open source software for mathematical programming. All of the reviewed software is very large programs and this paper does not even attempt to cover their scope. There have been many reviews of each system individually, but only a few comparisons between them that are either obsolete or don't compare all these five software (Kendrick & Amman, 1999), (Shacham et al, 1998), (Zotos, 2007), (Alsberg, 2006). The main advantage of these scientific computing environments is their ability to be used either from experienced software engineers or from amateur programmers.

Maple is a great choice for engineers, mathematicians, and scientists due to the fact that it is the result of over 25 years of cutting-edge research and development under the Waterloo Maple Inc. (also known as Maplesoft). Furthermore, there is a large number of tutorials and books referring to Maple utilities (Garvan, 2001), (Parlar, 2000). Apart from that, Maple has been selected from a vast number of universities world-wide to be used at the frameworks of lessons like mathematics. The latest compiled version of Maple is Maple 13. Mathematica is a computational mainly software program used in scientific, engineering, and mathematical fields which is developed and distributed by Wolfram Research which was founded by Stephen Wolfram in 1987. Moreover, there are many articles and books which describe analytically the advantages

and weaknesses of Mathematica (Dubin, 2003), (Tott, 2004). The latest compiled version of Mathematica is Mathematica 7. Matlab is a matrix language developed and distributed by the MathWorks Inc. As the name suggests, Matlab (MATrix LABoratory) is especially designed for matrix computations like, solving systems of linear equations or factoring matrices. Apart from that, it is general accepted that MATLAB and its numerous toolboxes can replace or enhance the usage of traditional simulation tools for advanced engineering applications. All these toolboxes and functions have thoroughly described in a numerous tutorials and guides (Palm, 2008), (Colgren, 2007). The latest compiled version of Matlab is Matlab R2010a. Octave is open source software where the users are able to enhance and adapt the source code according to their demands. Consequently, everyone is free to use it and free to redistribute it on certain conditions. Octave is a high-level language, primarily intended for numerical computations. Due to the fact that Octave is an open source software, there are many documentations either as books or available on-line through the internet (Eddelbuettel, 2000), (Malcolm, 1997). The latest compiled version of Octave is 3.2.3. Scilab is a scientific software package for numerical computations providing a powerful open computing environment for engineering and scientific applications. Scilab is open source software and since 1994 it has been distributed freely along with the source code via the internet. Moreover, Scilab includes a large number of functions and toolboxes which have been many times described in a large number of books and tutorials (Campbell, 2006), (Zhang, 2006), (Mrkaic, 2001). The latest compiled version of Scilab is 5.2.1.
An outline of the rest of the paper is as follows. In section 2, general descriptions of the systems are given. Furthermore, we introduce the numerical libraries and the useful packages and toolboxes of each software. Finally, a speed comparison of some frequently used procedures within mathematical models is presented in section 3. Finally, in section 4 we outline our conclusions.

2. NUMERICAL LIBRARIES

First of all, it is general accepted that numerical libraries is a sector of great value for mathematical programming languages. All reviewed programs have a big range of built-in mathematical functions and procedures. Apart from that, the reviewed software is not constrained only to mathematics but they are spread among many other scientific fields, like statistics and econometrics.
As it is mentioned previously, all reviewed software has many and adequate functions for most linear algebra calculations. These built-in functions have been implemented in order to simplify the process of mathematical programming for experienced and amateur users. All functions are parts of extensive numerical algorithms for a wide range of applications. Maple and Matlab are the most adequate software for linear algebra, while Mathematica, Octave and Scilab lack some functions like Smith Normal Form.
Another significant category of functions is the numerical analysis, a wide spread mathematical tool among the industry and the scientific community. Maple, Mathematica and Matlab have the most functions for numerical analysis. In contrast, Scilab lacks a function for k-Spline Interpolation and Octave misses some functions, like Inverse Fourier Transformation, Bisection and Newton method.
Statistics is a powerful tool for engineers, scientists, researchers and financial analysts in order to collect, analyze explain, and present their data. Consequently, statistics is also a significant section for mathematical programming languages. All reviewed software is an ideal option for statisticians, because they support a wide range of tasks, from basic descriptive statistics to developing and visualizing multidimensional non-linear models. They offer a rich set of statistical plot types and interactive graphics, such as polynomial plotting and response surface modelling.
Econometrics combines economic theory with statistics to analyze and test economic associations. Theoretical econometrics considers questions about the statistical properties of estimators and tests, while applied econometrics is concerned with the application of econometric methods to assess economic theories. Economical data are generally observational, rather than being derived from controlled experiments. Early work in econometrics focused on time-series data, but now econometrics also fully covers cross-sectional and panel data. For all these reasons, econometrics is a very important issue for every mathematical program. Mathematica and Matlab are more adequate in the area of econometrics, because they have more built-in functions which can be used from an econometrician than the other software. Maple and Octave lack some functions, like Durbin-Watson Test, and finally Scilab includes only the primary functions for econometrics.

3. SPEED COMPARISON

In order to gain an insight into the practical behavior of each one of the reviewed software, some computational experiments are proposed. The computational comparison has been performed on an Intel Pentium IV processor with 3.4 GHz and 1 GB RAM running under Windows 7. The reported CPU times were measured in seconds. The speed comparison tests 25 functions which are very often used within mathematical models and not only. Test set problems are categorized into five groups: miscellaneous operations, matrix operations, basic algebra, advanced algebra and statistics. The results given in Table 1 show the average times of 10 executions. All runs were made as a batch job. The 'winning' time for each test is given in bold.

Table 1 Speed Comparison.

TEST	Maple (13)	Mathematica (7)	Matlab (R2008b)	Octave (3.2.3)	Scilab (5.2.1)
MISCELLANEOUS OPERATIONS					
Loop test 100000 x 100000	11930,598	31819,800	**9774,985**	43425,000	72147,885
3000 x 3000 random matrix ^ 1000	4,960	**0,671**	4,103	3,463	4,353
Sorting of 10000000 random values	0,950	0,702	**0,267**	0,296	0,328
FFT over 2^20 random values	0,857	0,515	0,614	**0,484**	2,013
Calculation of 2000000 Fibonacci numbers	198,464	19,718	**2,066**	4,571	4,680
Plot 2d on 200000 points	0,953	**0,134**	1,560	1,243	1,076
Plot 3d on 200000 points	2,745	**0,456**	4,876	2,564	10,085
Average performance for the tests of this group	30,24%	**67,59%**	59,01%	43,66%	27,92%
MATRIX OPERATIONS					
Matrix multiplication among two 3000x3000 random arrays	14,760	10,840	**9,504**	13,276	17,847
Transpose of a 3000x3000 random matrix	1,160	0,172	0,149	**0,125**	0,359
Hessenberg form of a 3000x3000 random array	167,510	49,077	**36,915**	174,130	41,683
Average performance for the tests of this group	32,40%	78,54%	**94,63%**	64,26%	58,86%
BASIC ALGEBRA					
Determinant of a 3000x3000 random array	5,700	6,848	4,537	5,616	**4,336**
Inverse of a 3000x4000 random array	37,050	19,984	**12,707**	16,224	14,274
Eigenvalues of a 3000x3000 random array	390,530	**62,853**	407,302	487,500	171,850
Eigenvectors over a 3000x3000 random array	945,490	293,663	**133,780**	1039,000	1406,530
3000x3000 dot product matrix	12,870	12,652	**5,885**	6,895	10,202
Linear system solve of 3000 equations	5,880	5,678	**4,950**	5,679	163,410
Average performance for the tests of this group	45,67%	68,18%	**85,91%**	59,56%	50,08%
ADVANCED ALGEBRA					
Cholesky decomposition of a 2000x2000 random array	5,670	**2,010**	2,485	2,683	2,262
Lu decomposition of a 1500x1500 random array	7,960	5,897	**4,701**	5,834	20,286
Qr decomposition of a 1200x1200 random array	121,870	19,640	**17,469**	21,263	17,909
Singular value decomposition of a 2000x2000 random array	1015,640	93,054	**20,349**	576,950	52,915
Schur decomposition of a 1500x1500 random array	1626,790	27,050	**4,519**	733,790	95,566
Average performance for the tests of this group	22,23%	61,45%	**96,18%**	48,36%	50,55%
STATISTICS					
Principal component factorization over a 3000x300 random array	95,645	8,740	8,375	**6,540**	7,830
Gamma function on a 3000x3000 random matrix	6,380	7,085	**0,855**	30,092	3,885
Gaussian error function on a 3000x3000 random array	1,930	9,623	1,300	**0,187**	0,453
Linear regression over a 3000x3000 random array	60,150	4,852	4,798	5,710	**2,730**

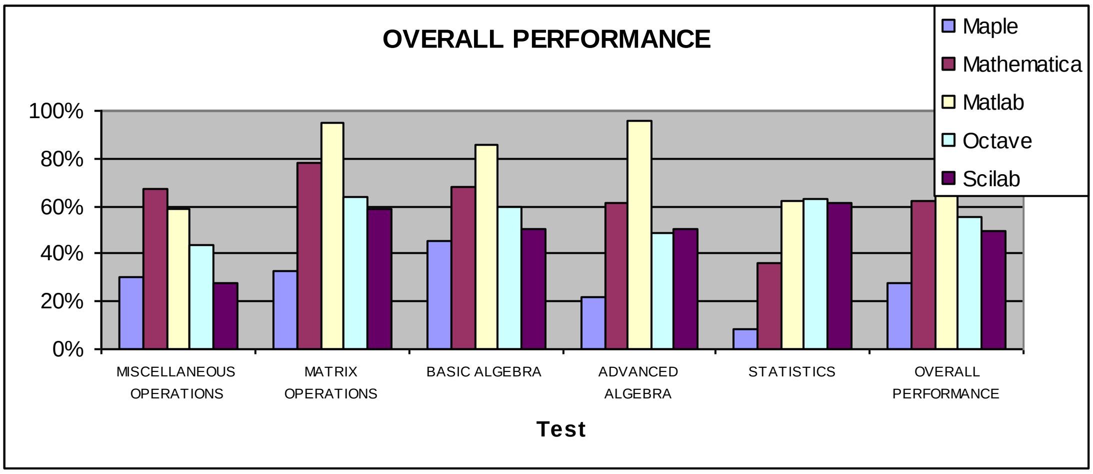
Average performance for the tests of this group	8,62%	36,28%	62,35%	**62,66%**	61,71%
OVERALL PERFORMANCE	27,83%	62,41%	**79,61%**	55,70%	49,82%

The software's performance for each test has been calculated using the relation

$$\frac{BT < software >_k}{T < software >_i} x(100), i = 1, 2, ..., 5, i \neq k$$

where $BT < software >_k$ is the best timing result for a specific test, run by all five programs and $T < software >_i$ is the timing result of the software i, $i \neq k$ for the same test. The best timing is then set equal to 100%. To calculate the overall performance for each one software, we add the percentage values for every test and divide it by the total number of tests. Finally, the largest percentage corresponds to the best overall performance.

Figure 1. Group and Overall Performance



In the above figure, the average time of each group of tests are presented for all reviewed software. The first group refers to miscellaneous operations like 2d and 3d plotting, sorting numbers. In this group Mathematica is the winner and Matlab follows with no significant difference. On the other hand, Scilab has the worst time which mainly depends on the loop test and 3d plotting. In the next group test which refers to matrix operations, Matlab has the best performance. In contrast to Matlab, Maple demands more time to accomplish the matrix operations and it came to the last position of these tests. In basic algebra, Matlab is again the winner and Mathematica comes next. The other three programming languages have no significant differences. In the next group of tests, advanced algebra, Matlab has the most significant difference comparing to execution times of the other software. Moreover, Octave and Scilab have almost the same performance and Maple comes last. In the last group of tests, which refers to statistics, Matlab, Octave and Scilab are very close with Octave to lead the average of execution times. Finally, in overall performance Matlab includes the best results due to the fact that it comes first in three of the five groups of tests.

4. CONCLUSIONS

The choice of data analysis software is not an easy decision and it depends on the needs and expectations of users. Maple, Mathematica, Matlab, Octave and Scilab are easy-to-use languages which allow a fast implementation and prototyping of mathematical and statistical algorithms. Despite their strong similarities, there are substantial differences between these matrix programming languages. They differ from each other in terms of usability, richness and finally in terms of performance.

Maple is the best solution for doing symbolic computations. Moreover, Maple has a large library of mathematical and econometrical functions. The main disadvantage of Maple is its' speed.
Mathematica is generally regarded as having the best graphical capability. Furthermore, Mathematica has a wide variety of functions for almost everything. A real weakness of Mathematica is its' complex syntax.
Matlab's strengths are the large number of available toolboxes, the possibility to develop graphical user-interfaces easily and the functions and competence for nearly all important topics. In contrast, the symbolic calculations with Matlab are weak.
Octave is a language compatible with Matlab, but it is not a top performer on Windows. It has a wide variety of mathematical and statistical functions that makes it a powerful tool for numerical computations.
Scilab is a free alternative of Matlab. Scilab offers a large number of functions for mathematical programming. Its performance at speed comparison is quite satisfactory and one disadvantage of Scilab was the restrictions referring to the dimensions of data that can create and handle.

5. REFERENCES

Alsberg B. and Hagen O. J., 2006. How can octave replace Matlab in chemometrics. *Chemometrics and Intelligent Laboratory Systems*. Vol. 84, No. 1-2, pp 195-200.

Campbell S. et al., 2006. *Modeling and simulation in Scilab/Scicos*, Springer, New York. USA.

Colgren R. D., 2007, *Basic Matlab*, Simulink and Statefow. AIAA Education Series, Virginia, USA.

Dubin D., 2003. *Numerical and Analytical Methods for Scientists and Engineers*, John Wiley & Sons Inc,

Eddelbuettel D., 2000. Econometrics with Octave. *Journal Of Applied Econometrics*, Vol. 15, pp 531-542.

Garvan F., 2002. *The Maple Book*. CRC Press, Florida, USA.

Kendrick, D. A. and Amman, H. M., 1999. Programming Languages in Economics. *Computational Economics*, Vol. 14, No. 1-2, pp 151-181.

Malcolm M., 1997. Octave: A Free High-Level Language for Mathematics. *Linux Journal*, Vol. 1997 (July), No.8.

Mrkaic M., 2001. Scilab as an Econometric Programming System. *Journal Of Applied Econometrics*, Vol. 16, No. 4, pp 553–559.

Palm W. J., 2008. *A Concise Introduction to Matlab*. McGraw-Hill, Rhode Island, USA.

Parlar M., 2000. *Interactive Operations Research with Maple*. Birkhauser, New York, USA.

Shacham M. et al, 1998. Comparing for interactive solution of nonlinear algebraic equations. *Computers & Chemical Engineering*, Vol. 22, No. 1-2, pp 323-331.

Tott M., 2004. *The Mathematica GuideBook for Programming*. Springer/Verlag, New York, USA.

Zhang Z., et al., 2006. *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg, Germany.

Zotos K., 2007. Performance comparison of Maple and Mathematica. *Applied Mathematics and Computation*, Vol. 188, No. 2, pp 1426-1429.

(2010) The Maplesoft website. [Online]. Available at: http://www.maplesoft.com/

(2010) The Wolfram Research website. [Online]. Available at: http://www.wolfram.com/

(2010) The MathWorks website. [Online]. Available at: http://www.mathworks.com/

(2010) The John W. Eaton. website. [Online]. Available at: http://www.gnu.org/software/octave/

(2010) The INRIA website. [Online]. Available at : http://www.scilab.org/