A computational evaluation of some free mathematical software for scientific computing

Themistoklis Glavelis, Nikolaos Ploskas, Nikolaos Samaras*

Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece

Abstract

The purpose of this paper is to present a computational comparison of some well-known free mathematical programs. Specifically, these programs are FreeMat, Mathnium, Octave, R and Scilab. All these programs are designed in order to be used either in the scientific community or the industry. Moreover, an additional aim of this paper is to propose open source alternative software which can replace commercial mathematical software. Several issues concerning available operating system platforms, data structures, languages’ structure, matrix operations, program modularization, libraries, packages, profiling tools and online help facilities are addressed in this paper. Discussions on their weaknesses and strengths characteristics are presented. Finally, a speed comparison of some frequently used algorithms, procedures and operations within software is also presented.

1. Introduction

The aim of this paper is to carry out a computational comparison of some very popular free mathematical software. There exist several hundreds of such programs which are more or less related to mathematics and statistics. These programs are interactive and offer a vast range of mathematical and statistical functions and libraries. The main advantage of these scientific computing environments is their ability to manipulate simple or sophisticated mathematical expressions and to provide concise syntax for matrix operations. Apart from that, another significant issue is the fact that these software programs can be a great choice for beginner programmers because integrate numerical analysis, matrix computation, and specialized functions in an ease-to-use graphical environment.

All reviewed programs are free software and the users do not have to concern about the cost or their permission license. There is a subtle distinction between the definition of “free software” and “open source software”. Free software is a matter of the users’ freedom to run, copy change and distribute the software. Roughly speaking, open source software is a subset of free software. The software that is compared in this paper can be categorized in two groups. The first category is the open source software where the users are able to enhance and adapt the source code according to their demands. On the other hand, there is the freely available software which permits only the free usage of them and they do not permit the access to their source code. FreeMat [16], Octave [18] and R [15] belong to the first category, while Mathnium [19] and Scilab [17] belong to the second one.

During the last decade, the number of users of open source software increases every year. Open source software is a new attitude in computer science community. More and more scientists and software engineers prefer open source software because they are able to adjust it to their needs. Moreover, the last years a number of open source mathematical programming software have introduced. Consequently, the strengths and weaknesses of these mathematical programming languages are of great value for potential users. Furthermore, users of commercial mathematical languages are able to compare their abilities with those of open source and decide if they will choose an open source alternative solution.

Principles for modern software environments to computer algebra systems can be found in Ref. [2]. There have been some reviews of mathematical systems [5,7,11] but they refer only to commercial programs, such as Matlab, Mathematica and Maple. In this paper we review the main features of the recent versions of FreeMat (version 3.6), Mathnium (version 1.051), Octave (version 3.0.3), R (version 2.8.0) and Scilab (version 5.0.2). Also, we have made computational experiments to compare the practical efficiency of these software programs.

This paper refers to amateur users but also to more experienced users. Consequently, the structure of the paper is chosen in order to introduce an amateur to mathematical programming at the same time of analyzing some more specific issues like parallel programming for experienced users. Furthermore, the sections of this paper

* Corresponding author at: 156 Egnatia Str., 54006 Thessaloniki, Greece. Tel.: +30 2310891866; fax: +30 2310891879.
E-mail addresses: glavelis@uom.gr (T. Glavelis), ploskas@uom.gr (N. Ploskas), samaras@uom.gr (N. Samaras).

1877-7503/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.jocs.2010.06.002
include all the appropriate information that a scientist or a
software engineer needs to dispose in order to decide which software
is better for him.

An outline of the rest of the paper is as follows. In Section 2, a
general description of all the reviewed software is presented and
in Section 3, there are details about the data structures of these
programs. Furthermore, in Section 4 there is useful information
referring to the programs modularization and in Section 5 there is
a general description of the programming style in these languages.
Moreover, in Section 6 there is a presentation of the libraries and
packages of each software and in Section 7, the functionality of
the programming environments (debugging and profiling tools)
is described. In Section 8, there are information about the help
facilities and the available documentation. A speed comparison of
some frequently used procedures within mathematical models is
described in Section 9. Finally, in Section 10 the final conclusions
are presented.

2. General description

FreeMat is an interpreted, matrix-oriented development envi-
ronment for engineering and scientific applications, similar to
the commercial package Matlab. It provides visualization, solving
numerical problems and image manipulation, as well as plotting.

Mathnium is a program for interactive numerical computations.
In addition, Mathnium can be a great choice for beginner program-
ners. Due to its vast range of functions and libraries, Mathnium
can be an ideal choice for solving numerical problems or manip-
ulating complex arithmetic expressions without a great deal of
programming effort. It is well known that Mathnium is written
in Java. Consequently, it provides convenient language constructs
which enable users to use their own existing Java classes without
additional programming. As a result, very large number of existing
libraries of Java classes can be very easily used within the Mathnium
environment. Except for that, Mathnium offers another useful utility:
within the Mathnium environment users are able to execute
Matlab codes.

GNU Octave is a high-level language, mainly intended for
numerical computations. It uses a language that is mostly compat-
ible with Matlab. It may also be used as a batch-oriented language.
Furthermore, Octave includes a large range of tools for manip-
ulating problems referring to linear algebra, finding the roots of
linear and nonlinear equations, integrating ordinary functions and
ordinary differential equations. Another significant issue is the fact
that Octave is easily extensible and customizable according to the
demands of users.

R is an interpreted computer language and environment mainly
for statistical computing, data analysis and graphics. It is a GNU
project and is similar to the commercial S-plus language. There
are some important differences between these two software but
almost every code written for Splus [22] runs unaltered under
R language, because both they are based on S language. More-
ever, R provides a wide variety of statistical utilities like linear and
nonlinear modeling, classical statistical tests, time-series analysis,
classification and many others. In addition R provides users with
a large range of graphical techniques. Furthermore, R is highly exten-
sible. The S language is often the vehicle of choice for research in
statistical methodology, and R provides an open source route to
participate in that activity.

SciLab is a software package providing a powerful comput-
ing environment for engineering and scientific applications. SciLab
comes with a large range of functions, libraries and toolboxes. These
libraries refer to 2-D and 3-D graphics, animation, linear algebra,
sparse matrices, polynomials and rational functions, classic and
robust control, differentiable and nondifferentiable optimization,
signal processing and statistics.

3. Data structures

The matrix is a standard data type in FreeMat. The cell array is a
fairly powerful array type that is available also in FreeMat. Roughly
speaking, a cell array is a heterogeneous array type, meaning that
different elements in the array can contain variables of different
types (including other cell arrays). FreeMat supports a string array
type that operates very much as user would expect. Strings are
stored internally as 8-bit values, and are very similar to numerical
arrays in all respects. FreeMat also supports global and persistent
variables.

Numbers, numeric arrays, strings, structures, and cells are the
basic data types in Mathnium. Matnium strings are arrays of char-
acters. Consequently, users are able to use all the matrix functions
for strings as well. Mathnium also has a facility for the definition of
derived data types as classes; this feature stems from the fact that
Mathnium is implemented in Java.

The standard built-in data types in Octave are real and complex
scalars and matrices, ranges, character strings, a data structure type,
and cell arrays. Octave stores all numbers in IEEE floating point
format to double (64-bit) precision. Matrix objects can be of any
size, and can be dynamically reshaped and resized. Since a string
is just a matrix, simple manipulations can be accomplished using
standard matrix operators.

R software can support either homogeneous structures or struc-
tures with a mixture of data types. In homogeneous structures all
data must be of the same type like vectors, factors, matrices and
arrays. On the other hand, there are other structures with a mixture
of modes like data frames, time-series data and lists. In this cate-
gory, several modes are allowed in one object. For example, users
are able to store numbers, characters, expressions and strings in a
single object.

SciLab recognizes several data types. Scalar objects are constants,
booleans, polynomials, strings and rational. Other basic objects are
lists, typed-lists and functions. The list data type can be a col-
lection of data objects not necessarily of the same type. SciLab
considers a number of data objects as matrices like scalars and
vectors. Furthermore, SciLab supports matrices of character strings
which are constructed as ordinary matrices. Furthermore, sym-

<table>
<thead>
<tr>
<th>Data type</th>
<th>Freemat</th>
<th>Mathnium</th>
<th>Octave</th>
<th>R</th>
<th>Scilab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Double</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Boolean</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Complex</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Matrices</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Strings</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Structures</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cells</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Java class</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sparse matrix</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

3. Program modularization

A script is a sequence of FreeMat commands contained in a file
with extension .m. When the script is called the effect is the same as
if the commands inside the script file were called one at a time from
the keyboard. Unlike function files (which have the same extension,
but have a function declaration), script files share the same environment as their callers. Hence, assignments made inside a script are visible to the caller which is not the case for functions. Unlike many other programming languages, FreeMat allows users to define functions that can take a variable number of input arguments. It is also possible to return a variable number of output arguments. Mathnium allows the user to define functions to organize their work with a variable number of inputs and output arguments. Moreover, inline functions can be declared in Mathnium that is convenient for defining simple functions, especially for calling functions that require a function as an argument. Furthermore, methods of Java classes in the class path can be easily imported as functions.

Complicated Octave programs can often be simplified by defining functions. Functions can be defined directly on the command line during interactive Octave sessions, or in external files (script files), and can be called just like built-in functions. A script file is a file containing (almost) any sequence of Octave commands. Octave like FreeMat can handle functions that take or return variable input or output arguments.

The R language allows the user to create objects of mode function. These are true R functions that are stored in a special internal form and may be used in further expressions and so on. In the process, the language gains enormously in power, convenience and elegance, and learning to write useful functions is one of the main ways to make your use of R comfortable and productive. R also can handle external script files. Functions are Scilab procedures ("macro", "function" and "procedure" have the same meaning). Usually, they are defined in files with Scilab's editor and loaded into Scilab.

### 5. Programming language constructs

In this section, a general review of the programming capabilities is presented. The issues that this section will focus on are: object-oriented programming, dynamic allocation of memory, sparse matrices and parallel computations.

Object-oriented programming is a style of programming that has become very popular in recent years. Much of the popularity comes from the fact that it is easier to write and maintain complicated systems with object-oriented programming. All reviewed software, except Octave, support object-oriented programming. Mathnium is written in Java and provides very simple mechanisms for using and extending existing Java classes, and implementing Java interfaces. R possesses a simple generic function mechanism which can be used for an object-oriented style of programming. FreeMat and Scilab also offer functions to write object-oriented programs without demanding much programming effort.

FreeMat also supports fully the model of dynamic programming. FreeMat is able to handle pointers with the same way as the C language does. The only thing that users should be careful is to avoid problems with memory allocation; functions are not allowed to return pointers. As a general operating mechanism, the FreeMat code must allocate the proper arrays, and then pass them by reference to the functions. Like other languages, R allows the use of pointers in order to gain some extra bytes of memory. In order to be available the utilities and functions about dynamical allocation of memory, users must load an extra library to R software, named Rindex. Similar to FreeMat, Scilab is able to create and manipulate pointers and indices in order to constrain the waste of memory. Moreover, Scilab supports pointers to any kind of type, from integers and strings to data structures.

There are many classes of mathematical problems which give rise to matrices, where a large number of the elements are zero. In this case it makes sense to have a special matrix type to handle this class of problems where only the nonzero elements of the matrix are stored. Not only does this reduce the amount of memory to store the matrix, but it also means that operations on this type of matrix can take advantage of the a-priori knowledge of the positions of the nonzero elements to accelerate their calculations. A matrix type that stores only the nonzero elements is generally called sparse. All the above languages support creation and handling of sparse matrices with a selected density.

Nowadays, there are cases where the effectiveness of classical programming languages is not satisfactory with regard to the execution time. A growing number of real-world problems demand parallel computing capabilities. It is well known that parallelization can lead to important reduction in total computational time. All reviewed software, except Mathnium, support parallel programming. FreeMat implements parallel processing with MPI (Message Passing Interface). FreeMat is able to take full advantage of the benefits of MPI. Nowadays, multicore machines are everywhere. In this case, FreeMat offers an alternative, the OpenMP technique. Octave like FreeMat is able of running parallel its codes. Octave users in a cluster with several PCs can use the MPI toolbox in order to call MPI library routines.

Although R software is a programming tool which is used mainly by statisticians and econometricians, users have the possibility to execute their codes in parallel. In many cases, for example in time series, the amount of data is prohibitive for a single processor. R uses an MPI library in order to support this service. Consequently, users should load two extra packages, “Rmpi” and “taskPR” libraries. With these two packages users will be able to exploit all the advantages of parallel programming through R software. Like the other software, Scilab provides users with the opportunity to execute their programs in parallel. Similar to the previous cases, Scilab demands an MPI library. In contrast to other software, Scilab supports a generic interface to grid computing. Through this interface users can execute and monitor their codes during the execution phase. Moreover, they are able to extract very useful information about the execution time like which processor has the biggest delay and which part of the code is executed in each computer. Moreover, if something goes wrong users can be aware very fast and easy of which machine has crashed.

As it is presented in Table 2, FreeMat, R and Scilab are the most powerful tools referring to the language constructs that discussed in this section. Mathnium has a facility for the definition of derived data types as classes, but it does not support dynamic allocation of memory and parallel programming. Octave has powerful tools for parallel programming, but it does not allocate memory dynamic.

### 6. Libraries

This section covers one of the most significant issues in this paper. All reviewed programs have a big range of built-in mathematical functions. Their use is not constrained only to mathematics but it is spread into many scientific fields, like statistics and econometrics. In this section, we will review the strengths and weaknesses of these five programming languages in terms of their available functions and libraries.

FreeMat is designed to implement numerical algorithms for a wide range of applications. It supports functions of a wide variety of fields, like mathematics and statistics. It provides visualization
and image manipulation. Furthermore, it supports many options for plotting. Moreover, FreeMat has functions for import/export utilities, like reading from XML or HTML files, but does not include database connectivity. One other capability of FreeMat, that underlines its compatibility with Matlab, is the manipulation of heterogeneous array types and dynamic structure arrays. Finally, FreeMat provides codeless interface to external C/C++/FORTRAN code.

Mathnium is a program for interactive numerical computations. It has a comprehensive library of functions for a variety of problems in applied mathematics. Mathnium allows users to solve numerical problems without a great deal of programming effort. Although it is not a general purpose language, users can also extend the capabilities of Mathnium by using it as a programming language. The current version of Mathnium contains about 400 functions for various tasks, but still, comparing to other reviewed software, requires more features to be implemented in the future.

Octave’s primary scope is to be full compatible with Matlab. Octave has extensive tools for solving common numerical linear algebra problems, finding the roots of nonlinear equations, integrating ordinary functions, manipulating polynomials, and integrating ordinary differential and differential-algebraic equations. It is easily extensible and customizable via user-defined functions written in Octave’s own language, or using dynamically loaded modules written in C++, C, Fortran, or other languages.

R is a language and environment for statistical computing and graphics. R provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, etc.) and graphical techniques, and is highly extensible. Also provides database connectivity and a variety of functions for import/export utilities, like import from other commercial statistical packages (SPSS, S-PLUS, etc.). The capabilities of R are extended through user-submitted packages, which allow specialized statistical techniques as well as graphical devices.

Scilab supports a large amount of functions in the fields of standard mathematics, linear algebra and numerical analysis. Moreover, Scilab supports a large number of functions referring to statistics, stochastic statistics and distributions, although that this is not its strong feature. Scilab has built-in support for linear algebra (customized LAPACK), linear programming, constrained quadratic and nonlinear programming, data exchange with Matlab, optimal control and signal processing. In addition to the built-in functions there are a few packages, written by Scilab users, that are available free of charge from the Internet. Furthermore, Scilab supports many functions for import/export utilities. Graphs are also very impressive. A number of readily available tools allow the users to take full advantage of their work, including copying of graphs to clipboard or printing to postscript.

### 7. Functionality of the programming environment (debugging and profilers)

This section covers some special features of the programming environment of these languages, like debugging and profiling. Moreover, this section attempts to review the available facilities that these software offer to simplify the programming effort.

FreeMat provides a debugging functionality that allows users to examine where the error occurred. FreeMat has a quite simple profiler where only functions and scripts are profiled. Relatively with the programming environment, FreeMat offers a clean and functional environment. It also supports a simple editor with highlighting capabilities and multiple tabs.

Mathnium uses only a command line interface. The fact that Mathnium does not support a graphical interface is one of its ‘real weaknesses. Mathnium provides some basic options for debugging errors, but it does not have a profiling utility.

The latest version of Octave provides a quite simple graphical environment with a built-in editor. There are some other interesting external front-end for Octave, like kOctave [20] and QtOctave [21]. Octave may also be used as a batch-oriented language. In addition, JHandles package provides some basic features for GUI programming, but it is far from a complete version. Furthermore, Octave supports easy debugging but has not implemented yet a profiler. Finally, it is also possible to execute Octave code directly in a C++ program.

R provides a functional graphical user interface. However, its incorporating editor is only a text pad and does not offer any utility, like syntax highlighting. Due to its S heritage, R has stronger object-oriented programming facilities than most other statistical computing languages. Finally, R provides a variety of functions for debugging errors and profiling functions and scripts.

Scilab has a simple and practical graphical user interface with a powerful and rich in options built-in editor that supports syntax highlighting, command completion and insertion of breakpoints. Among other features Scilab’s graphical user interface provides a tool to translate Matlab scripts to Scilab compatible files. In addition, it supports many debugging and profiling utilities. Last but not least, Scilab is the only of the reviewed languages that allows the user to create a complete graphical user interface.

As it is shown in Table 3, FreeMat provides a debugger, but its profiler is very simple. Mathnium weaknesses are the lack of a graphical user interface and profiler. Octave provides easy debugging, but does not include a profiler. R provides a variety of functions for debugging errors and profiling functions and scripts, but his editor does not offer common utilities, like syntax highlighting. Finally, Scilab is the only software that allows users to create a graphical user interface and its editor supports syntax highlighting, command completion and insertion of breakpoints.

### 8. Help facilities and documentation

FreeMat comes with a solid documentation. The manual is well written and easy to follow. FreeMat’s help environment is very useful and provides searching index. FreeMat provides many examples that help users to find out the syntax and scope of various functions. The website of FreeMat contains documentation links and frequently asked questions. Mathnium provides neither a separate help environment nor a reference manual. Users can search for help in its website where a full list of all Mathnium functions is presented. There are also some examples explaining the key features of the programming language. Octave is shipped with a well-written manual. The help environment in Octave is not user-friendly and does not provide useful operations like searching. John Eaton, Octave’s creator, maintains a website, which provides useful information and many documentation links. Moreover, there exists a mailing list. There are only a few texts referring to Octave [4,8,10].

R comes with a reference manual as well as several brochures describing special features like data import/export functions. R help environment provides searching for functions, but the majority of
documentation files are cited in the website of the project. The number of texts which describes R is large. The most recent of them are [3,6,12–14]. R maintains a commented list of books on its website. The help environment in Scilab is very usable and easy-to-handle. Scilab provides plenty of examples in question. Searching through help files is straightforward with the apropos command. Many examples of command usage can also be obtained by running commands with empty argument lists. Scilab maintains a website where someone can find useful information and links. Moreover, there are a few texts that describe Scilab [1,9].

As it is shown in Table 4, Freemat, R and Scilab provide all the appropriate help facilities that a user needs. On the other hand, Mathnium has only an online documentation that makes it difficult to users to find easily the information that they want. Finally, Octave lacks of a searching index in its environment and its website does not offer an online documentation.

### Table 4 Supported help facilities.

<table>
<thead>
<tr>
<th>Help facilities and documentation</th>
<th>Freemat</th>
<th>Mathnium</th>
<th>Octave</th>
<th>R</th>
<th>Scilab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help environment</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Manual</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Searching index</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Online documentation</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Mailing list; user forum</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

### 9. Computational evaluation

In order to gain an insight into the practical behavior of each one of the reviewed software, some computational experiments are performed. The computational comparison has been performed on an Intel Pentium IV processor with 3.4 GHz and 1 GB RAM run-
ning under Windows XP Professional SP3. The reported CPU times were measured in seconds. The speed comparison tests 40 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 5 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.

The computation evaluation includes a large number of tests which refer to a vast range of research areas. The functions which have been used in speed comparison are the built-in functions of each software. Also it is necessary to mention that neither the functions have been optimized nor a specific compilation has been made. The paper does not target on the algorithms which have been used in each function. In contrast, it focuses on the performance because users mainly care about that.

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

$$BT < \text{software} > k \times \frac{T < \text{software} > i}{i = 1, 2, ..., 5, \ i \neq k}$$

where $BT < \text{software} > k$ is the best timing result for a specific test, run by all five programs and $T < \text{software} > i$ 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 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. Functions which are not supported by a program (denoted by the mark ‘-‘) have not been judged in these calculations and have therefore no influence on the overall timing result.
Below in this section, there are some figures extracted from the speed comparison. There are presented six figures, one for each group of tests as they are shown in the Table 5 and one for the average and overall performance. The time in the y-axis in the figures below is in logarithmic scale, so the time is multiplied by an appropriate number in each case because the logarithm of a number less than 1 is negative.

In Fig. 1, some miscellaneous operations is presented like loop test, sorting a vector, factorial of a big integer, plotting, etc. All the times in this group have been multiplied by 1,000. In this section, Octave has the best performance at execution time, as it is shown in Fig. 6. On the other hand, Mathnium is the slowest software. Moreover, R performance is closest to the Octave execution time and at last the results of the rest two programs are not quite satisfactory.

In Fig. 2, there is the performance of all software at the section of matrix operations. All the times in this group have been multiplied by 1,000. In this section, Scilab is the “winner” of this comparison, as it is shown in Fig. 6. On the other hand, Mathnium has the worst execution time and it differs a lot from the execution time of the rest software. It is remarkable the fact that Mathnium has the worst performance at the half of the tests in this group. Another significant issue is the performance of FreeMat which has the second worst execution time. Moreover, Octave and R software are close to each other and both are quite fast in this section.

In Fig. 3, some basic algebra operations are presented like the inverse of a matrix, eigenvalues and eigenvectors, etc. All the times in this group have been multiplied by 10. In this section, R software has the best execution time. Furthermore, Mathnium has the worst execution time. Moreover, Mathnium has the largest delay in 4 out of 7 tests. Moreover, FreeMat and Scilab are very close to the R performance and Octave is following as it is shown in Fig. 6.

In Fig. 4, there is the performance of all software at the section of advanced algebra operations. All the times in this group have been multiplied by 1,000. In this section, Scilab is the “winner” of this comparison, as it is shown in Fig. 6. The remarkable in this section is that Scilab is the “winner” but the execution time of Scilab and R do not differ in many aspects. On the other hand, Mathnium has again the worst execution time but in this case its time is closer to the other software and more specifically Mathnium is closest to Octave’s performance. FreeMat is a bit faster than these two software.

In Fig. 5, there is the performance of all reviewed programs at the section of statistics. All the times in this group have been multiplied again by 1,000. According to Fig. 6, Mathnium has the worst performance and FreeMat computes faster these statistics tests. For the rest software, Scilab has the shortest time after FreeMat and R follows. Octave has the second worst execution time.

In the last figure, the average performance of all the group tests for all the reviewed software is presented. It is obvious; the fact that Mathnium has the worst performance in all sections and as a result it is the last in the final order. On the other hand, Scilab and R software are very close to each other and the final “winner” is Scilab but with a very short difference. This short difference between these two programs can lead us to the conclusion that Scilab and R are equivalent. With another set of data the results may be different, with the R at the first place and Scilab following. In addition, at the third place comes the FreeMat software and Octave has the second worst performance. It is significant the fact that FreeMat is close to the performance of R and Scilab. Apart from the fact that Octave has the second worst performance, Octave has the most adequate libraries as well as Scilab. This conclusion stems from the fact that these two programs participate in all tests of the present paper.
10. Conclusions

The choice of mathematical software is a difficult task. FreeMat, Mathnium, Octave, R 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 term of performance.

FreeMat is a free environment for rapid engineering and scientific prototyping and data processing. It is similar to commercial systems such as Matlab from Mathworks, and IDL from Research Systems, but is open source. FreeMat offers a clean and functional environment with a user-friendly editor. Although FreeMat misses some functions and especially functions in statistics, its execution time in this area are very satisfactory.

Mathnium is a program for interactive numerical computations and it is inspired by Matlab. It is written in Java, and it provides convenient language constructs which do not require from user to actually write formal Java programs in order to use existing Java classes. Mathnium uses only a command line interface. The lack of a graphical user interface and documentation resources are its’ real weaknesses. Moreover, Mathnium has the worst performance in all groups of tests and in statistics lacks some basic functions.

Octave is a language compatible with Matlab, but it is not a top performer on Windows, because it runs under cygwin. It has a wide variety of mathematical and statistical functions that makes it a powerful tool for numerical computations. It includes all the tests of the computational evaluation and its best performance is in miscellaneous operations like plots and creating random numbers. A real weakness is its poor user interface, but as mentioned before that problem can be solved using a compatible external user interface.

R is a free software environment for statistical computing and graphics. R is one of the fastest open source data analysis packages. It provides many additional packages for all kind of statistics. One of its disadvantages is its incorporating editor that is only a text pad and does not offer any utility, like syntax highlighting.

Scilab is a free alternative of Matlab. Scilab proposes more functions than Octave, but it is not 100% compatible with the Matlab language. Scilab offers functions for creating graphical user interfaces. It is the fastest software in the benchmark that was performed. However, the only disadvantage of Scilab was the restrictions referring to the dimensions of data that can create and handle. Nevertheless, this drawback can be overcome with the use of parallel programming which is a strong Scilab’s feature.

Finally, the main advantage of Octave is his ability of being totally compatible with Matlab but his computational performance is as satisfactory as Scilab’s. Scilab has the best performance in computational evaluation and it may be the best solution for a scientist in the area of mathematical programming. Another significant of Scilab is its friendly user interface which is a real weakness of Octave. In contrast, R may be an ideal alternative for statisticians or econometricians who use commercial software such as S-Plus. Furthermore, Mathnium is the only software which has a facility for the definition of derived data types as classes but his performance in computational evaluation is not satisfactory. Freemate involves quite satisfactory execution times in speed comparison but it misses a number of functions in some research area like statistics and this is its main disadvantage towards the other software.

References

**Ploskas Nikolaos** is a Ph.D. student at the Department of Applied Informatics, University of Macedonia, Greece. He received his Bachelor of Science degree and Master degree in Computer Systems from the Department of Applied Informatics of the University of Macedonia, Greece. His primary research interests are in mathematical programming, analysis and design of algorithms, parallel and distributed programming, grid technologies and operations research.

**Nikolaos Samaras** is an Assistant Professor at the Department of Applied Informatics, University of Macedonia, Greece. He received his Bachelor of Science degree and Ph.D. degree in Applied Informatics from the University of Macedonia, Greece. His primary research interests are in mathematical programming, experimental evaluation of algorithm performance, grid computing, computer applications and computational operations research.