



Energy and resource comparison of current applications with a focus on statistical analyses and evaluations using the example of MATLAB and R

Tim Seegert¹, Malina Bergmann¹, Josephine Brömme¹, Dennis Junger² and Volker Wohlgemuth²

Abstract: This paper compares the energy and resource efficiency between MATLAB and R, two widely used programming languages in scientific computing and data analysis. A load driver and automation software, Power Automate, were utilized as a system under test to measure and evaluate the performance of both languages. Before the experiment, specific mathematical operations and execution methods were developed in MATLAB and R scripts. The measurement and evaluation were conducted using the Oscar framework. The results indicate that R outperforms MATLAB in baseline and statistical operations, while MATLAB excels in matrix calculations. These findings provide valuable insights for selecting the most suitable programming language based on specific computational requirements, optimizing energy consumption and resource utilization.

Keywords: Blue Angel of Software, Energy Consumption, Green Coding, Green IT, Green Computing, Sustainable Software Engineering, Energy Efficiency, Resource Efficiency, MATLAB, R

1 Introduction

In an era of Big Data, complex mathematical calculations are necessary to evaluate enormous amounts of data. This requires appropriate tools that take over these tasks to simplify the work for the users. These include R and MATLAB, among others. According to the “Popularity of Programming Language Index (PYPL),” R is ranked seventh and MATLAB 14th of the most popular programming languages [PY23]. Despite the same field of application, the two languages differ. The focus of R is on statistical evaluations. MATLAB offers functions for solving various technical problems, such as data analysis or matrix calculations [Go23, Lu23]. To cope with such complex calculations, the programming languages or the programs developed with them consume energy indirectly through hardware use. These greenhouse gasses generated by all information and communication technologies are expected to have a global share of around 2.3% by 2030

¹ HTW Berlin, University of Applied Sciences, Industrial Environmental Informatics Unit, Wilhelminenhofstr.75A, 12459 Berlin, Germany {malina.bergmann|josephine.broemme|tim.seegert}@student.htw-berlin.de

² HTW Berlin, University of Applied Sciences, Industrial Environmental Informatics Unit, Wilhelminenhofstr.75A, 12459 Berlin, Germany, {dennis.junger|volker.wohlgemuth}@htw-berlin.de

[Ge20].

This work aims to determine which of the two programs has a lower energy consumption for simple mathematical operations. For this purpose, the energy consumption of different typical uses of R and MATLAB will be investigated and compared. As part of a measurement study, the hardware utilization and energy consumption of two usage scenarios were measured and evaluated to be able to make a usage recommendation if necessary. The methods used to achieve this goal are discussed in more detail in the rest of this paper. This includes the general approach to the implementation, a description of the selected standard usage scenarios and the scripts created, and an explanation of the measurement procedure. Subsequently, the results of the measurements are described, analyzed and the measuring procedure is evaluated. Finally, possible conclusions and recommendations for action are discussed.

2 Methods

The following chapter describes the methodological procedure of the efficiency comparison. The description of the technical measurement environment is followed by an explanation of the software functionalities that have been examined and the measurement procedure. Finally, the method for evaluating the results is described.

2.1 Environment and measurement setup

A suitable environment must be set up to measure the software's energy consumption and hardware utilization. This chapter describes the chosen environment setup to fulfill this requirement. The approach that has been selected follows the measurement setup of previous measurements which have been done in other projects at the HTW, Berlin University of Applied Sciences. It has been used, for example, by Junger et al. [Ju22], to measure the power consumption of material flow-oriented environmental management information systems (EMIS), especially life cycle assessment (LCA) programs. According to Kern et al., four components must be included in the measurement [Ke18]. The first component is the “System Under Test” (SUT), which executes the scripts for MATLAB and R. As described by Junger et al. [Ju22], a Windows 10-based SUT called DJ-Darwin was used for the measurement. The second component, the ‘load driver’ (LT), also runs on the SUT in the applied measurement setup. The load driver uses automation software to generate the load. For this project, the program “Power Automate” (formerly Microsoft Flow) from Microsoft was selected, described in more detail in Chapter 2.4. The automation software takes over the automated execution of the scripts. A data collector set of the Windows performance monitor was created to record the operating resource data during the tests. In addition to the processor load, the memory load, swap files, read and write operations on the hard disk, and the sent and received network data are also logged.

Another component of the measurement environment is the power meter, which is responsible for recording the consumption of the SUT. The power meter used in this project is a Gude Power Control 1202-1 socket strip with integrated energy measurement for server cabinets as recommended in the “Blauer Engel” for software that aggregates and logs the energy used, averaged to one second, by a script depending on the connected computer (DJ-Einstein) via the SNMP-Protocol [Bl23]. The script was originally part of the UFOPLAN by Trier University averaging [Gu21] and can be found in the actual version in the Gitlab Repo mentioned in the footnote³. To combine the data generated during the measurement, the action logs, the resources and energy data, a computer for data collection (DJ-Dawin) is provided. This machine also runs the “Open Source Software Consumption Analysis and Reporting” (OSCAR), invented by the Umwelt-Campus Birkenfeld, to analyze the measurement data and generate reports about the measurement. [Os23a] can also be found as online software in [Os23b].

The time frame for the measurements of this project work was two days. It was part of the pilot integration of green coding in the industrial environmental informatics unit at the HTW Berlin [Ju23].

2.2 Description of the used operations and functions

Following the common standards for analyzing software regarding energy consumption and hardware utilization, idle measurements were first made for comparability. The two software products to be compared, MATLAB and RStudio, were started without using the scripts written in the respective programming languages. The running software's main noise (the general energy consumption) was measured. To be able to compare the two software products as best as possible, both MATLAB's strengths (numerical calculations [Ge20]) and R's primary function (statistical evaluations [Lu23]) were then examined in the associated development environments. The matrix calculation operations carried out to compare the numerical calculations are shown in Tab. 1, together with the methods used in each case.

³ <https://gitlab.rlp.net/green-software-engineering/mobiles-messgerat>, last accessed at [31.07.2023]

Operation	used functions / operator MATLAB	used functions / operator R
Reading matrix	<i>readmatrix()</i>	<i>matrix()</i>
Calculate determinant	<i>det(A)</i>	<i>det(A)</i>
Sum of two matrix	$A+B$	$A+B$
Multiplication of two matrices	$A*B$	$A\%*\%B$
Determine the inversion	<i>inv(A)</i>	<i>solve(A)</i>
Potentialiation of the inputs	$A.^x$	A^x
Factorization	$A*x$	$A*x$
Determination of the transposed matrix	$A.'$	<i>t(A)</i>

Tab. 1: Operations performed for the comparison of the execution of numerical calculations (A, B: matrices, x: real number)

To compare the efficiency of performing statistical evaluations, the calculation of statistical parameters was carried out in MATLAB and R in a similar manner as possible. The chosen comparison criteria and the functions used in each case are shown in Tab. 2. For R, the *ggplot()* function for graph representation is used because of the function's more uncomplicated handling and better readability. This is merely a preference of the users.

Operation	used functions / operator MATLAB	used functions / operator R
Format date	<i>datetime()</i>	<i>as.Date()</i>
Create diagram	<i>plot()</i>	<i>ggplot()</i>
Calculate mean	<i>mean()</i>	<i>mean()</i>
Calculate quantiles	<i>quantile()</i>	<i>quantile()</i>
Calculate standard deviation	<i>std()</i>	<i>sd()</i>
Calculate variance	<i>var()</i>	<i>var()</i>
Calculate median	<i>median()</i>	<i>median()</i>
Export of the diagram	<i>exportgraphics()</i>	<i>png()</i>

Tab. 2: Operations carried out for the comparison of the performance of statistical evaluations

2.3 Description of the performed execution sequences

The algorithm for each usage scenario is executed and structured similarly for MATLAB and R. There is an input file with a square 8x8 matrix used to compare the matrix calculation. The matrix saved as a text file is read directly in MATLAB with the command *readmatrix()*. In R, the CSV file must first be loaded and then converted into a matrix with the function *matrix()*. Then the above calculation operations are carried out with two manually entered 3x3 matrices and the results are saved in variables in each case. In MATLAB, a text file is created for saving the results, which is then continued for all calculation results in each case with the function *writematrix()*. Finally, the clear command is used to delete all variables stored in the workspace and to create identical starting conditions for the next measurement run. In R, the results are also saved in a text file, which is created and then filled step by step with the functions *write()* or *writetable()*. The workspace is cleaned with the command *rm(list=ls())*.

The input for the statistical evaluations is a text file downloaded from the website of the German Weather Service with the climate data of the weather station in Marzahn. The file is 1.16 MB and contains data from 1993 to 2021 [De23]. The text file is read into MATLAB and R and the column TMK (temperature) is extracted for further analyses. The date values from the column “MESS_DATUM” must first be converted into the datetime format and saved. In MATLAB, these date values are saved with the associated temperature data as a timetable; in R, a data frame of the related data pairs is created. The data from 2018 and 2021 are each extracted as examples and the calculations listed above are carried out with these data and saved in variables. In addition, plots created from the temperature curve and the result variables are found in the text file as in the matrix calculation. Finally, the variables saved in the workspace are removed and the windows opened for the plots are closed. The command ‘close all’ is used in MATLAB and *graphics.off()* in R.

2.4 Procedure and execution of the measurement

An essential element that enables the measurement and comparison of resource and energy usage of the software is the automation software (see Chapter 2.1). The automation software is used to synchronize the system time and, based on this, for logging on to the SUT. It also automates user inputs in MATLAB and RStudio that cannot be measured manually. The following settings offer a success-based approach. A resolution of 0.1 seconds was chosen to address, execute and log the actions. A time frame of 120 seconds per test run was set for all measurements performed, baseline (before the installation of MATLAB and RStudio) and usage scenarios - idle, matrix calculations, statistical calculations (once for each of the installed software components to be measured). This ensured that all operations were run through every scenario, i.e., each measurement. Every single of these test runs was repeated 30 times to calculate the statistical mean. Above all, this part of the desktop automation serves the comparability of the measurements and their evaluation.

The measurement procedure is divided into seven essential steps:

1. Energy measurement is started on the energy data aggregator.
2. Collecting the performance data using the data collector set of the Windows monitoring software 'perfmom.exe' on the SUT is started, then wait.
3. The software to be tested is opened, in this analysis, MATLAB or RStudio.
4. Power Automate is opened and the respective (MATLAB or RStudio) automation is started. Power Automate is closed so as not to interfere with the measurement.
5. The end of the measurement, after 30 repetitions, is indicated by a pop-up window.
6. The energy measurement on the energy data aggregator is stopped.
7. The SUT saves the PC data and the action logging, and the aggregator saves the energy measurement on an external storage medium.

If technical or other malfunctions occur during the entire process, the measurement process must be restarted. Following a successful measurement, which provides all the necessary data for an evaluation as output, the review described in Chapter 2.5 is carried out with the OSCAR tool developed for this purpose.

2.5 Evaluation of the measurement results with OSCAR

The tool OSCAR, designed to evaluate software's resource and energy consumption, was developed within the UFOPLAN project "Sustainable Software Design" [Os23a]. For the evaluation of the measurement data generated for MATLAB and R, these are uploaded to OSCAR via a GUI, for which the three essential data sets of the measurement, described in Chapter 2.4, are used:

1. Actions - are the log files of the SUT, which were used for the synchronization of the measurements, since in these start and end times of test runs, actions and operations performed by the automation software were recorded with the corresponding time stamp.
2. Electrical power - describes the data recorded by the power meter in terms of average, minimum and maximum power consumption in watts, synchronized with the logged actions on the SUT, with a corresponding time stamp.
3. Hardware utilization - is also timestamped data that describes the utilization of various hardware components. In addition to CPU usage data, data on main memory (RAM), hard disk usage and network activity were also listed for the measurements.

The data sets from all three areas described above are read into OSCAR as CSV files with separated values. If necessary, final adjustments can be made in OSCAR regarding the

character set or timestamp per read-in file before the final report can be generated and downloaded [Os23b].

The evaluation report per measurement (this can be a baseline measurement as well as a usage scenario) consists of eleven pages for evaluating the resource consumption and the energy consumption, including the calculations for the electrical work and one graph each for the electrical power, processor utilization, working memory utilization, received and sent network data, i.e., also bytes read and written to the fixed memory, each over the time course of the usage scenario. Due to the large number of measurements and thus reports generated in OSCAR, the main results are presented and interpreted in the following chapter.

3 Description and interpretation of the results

The results, like the measurements already described, consist of three main components: the baseline and idle measurements (each as a comparison of the measured values entirely without additionally installed software) and the software to be analyzed (MATLAB and RStudio). The description and subsequent interpretation of the individual results in this section interprets this sequence of measurements. The results described in the following chapter are averages that OSCAR has calculated across the 30 repetitions for each scenario. Fig. 1 shows differences in the measured values regarding all the parameters examined; for example, in the baseline measurement, the average electrical work is 0.31 Watt-hour (Wh), but in the idle scenario of RStudio, it is already 0.34 Wh. In comparison, MATLAB consumes 0.36 Wh without active use. If we look at the permanent memory usage, i.e., in this context reading and writing data to the hard disk, 271.55 MB are consumed in the baseline scenario, 302.39 MB in the idle scenario of RStudio and just 252.74 MB in MATLAB. These different results in the measurements alone clearly show that for significant comparisons, considerably more measures are necessary and, in addition, the other activities on the PC would have to be shut down or restricted even further than was already the case for the measurements in this study. Comparing the usage scenarios of RStudio and MATLAB, it becomes apparent that the results for the calculations are close. RStudio performs better than MATLAB in terms of memory utilization, both MB read and written to and from the hard disk, as well as RAM utilization; there are slight deviations in the data transferred via the network only. The situation is similar for both programs with regard to CPU utilization during statistical calculations. The calculations of the matrices show apparent differences. While RStudio uses 1.94% of the CPU, MATLAB uses only 1.39% of the CPU in the tested scenarios. Since the processor is one of the components in the PC that consumes the most power [Co23], it makes sense in the analyzed scenarios to select the respective software according to the purpose for which it was designed and thus in correspondence to its strengths.

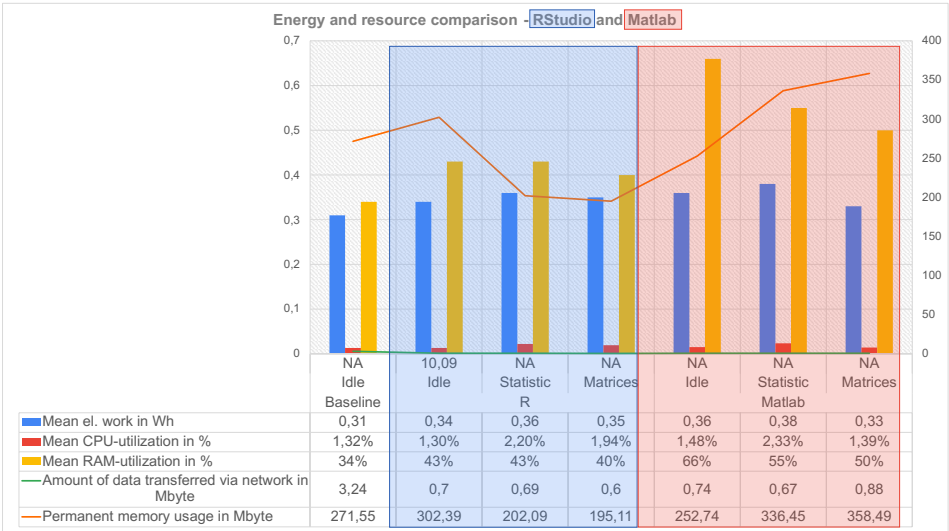


Fig. 1: Results comparison resource and energy use of MATLAB and R

Looking more closely at the power consumption (see Fig. 2) of R and MATLAB in the further analysis of the results, it becomes clear that they perform better, depending on their strengths and original purposes. While MATLAB requires 0.33 Wh to calculate matrices, R needs 0.35 Wh. Considering and thus deducting the idle consumption, MATLAB requires 0.03 Wh, whereas R only needs 0.01 Wh for the calculation of the matrices. For the statistical calculations, R consumes 0.36 Wh while MATLAB already consumes 0.38 Wh. However, subtracting the idle power usage for both of those, the statistical calculations equally consume 0.02 Wh. The difference between the idle measurements of the Baseline and R is 0.03 Wh and the difference between the Baseline and MATLAB results in 0.05 Wh. In summary, the idle mode of MATLAB consumes 0.02 Wh more than the one of R.

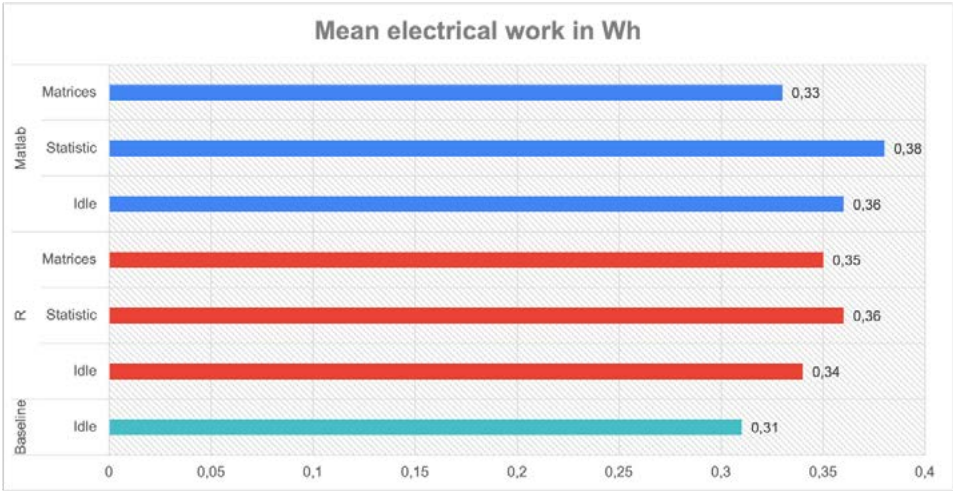


Fig. 2: Results MATLAB, R and baseline measurements regarding the mean electricity work in Wh

4 Discussion

The figures from the previous chapter make it clear that R performed better in comparison with MATLAB in terms of idle time and the execution of matrix calculations in this project. MATLAB is ahead in statistical operations. If the focus is on the total amount of mean electrical work in Wh, R is the more energy-friendly programming language in the scope of the work considered. Whether R represents the more resource-efficient variant overall cannot be answered in this paper. Although the scripts contain essential mathematical functions, they are still short software processes. This does not result in a high load, which makes a relative comparison difficult due to the generally low values. Furthermore, small operations have been applied in this project to show whether measurable differences, in general, occur during a measurement. For more precise measurements and a better estimation of the resource consumption of the languages, they should be tested for their core functions with more detailed scripts, which at best illuminate further scenarios and larger amounts of data.

Likewise, an increase in the number of runs in the automation software can facilitate the comparison, which was not feasible in this project due to time limitations.

Despite the difficulties, the results can be taken as a first prognostic that R is the more energy-saving variant, especially because of the smaller power usage in their idle state. This is the first indication for further measurements. If the points listed regarding expanding the scripts in new projects are considered, the languages can be compared in more detail and the statement that R is more resource-efficient can be further confirmed

or refuted. Another clue is that programs should be used for their specialization to be more resource efficient. This point can be confirmed if measured values are available from other programs, of which the main functions have been checked for their energy consumption.

This document is an output from a project funded by the Internet Society Foundation.

5 References

- [Bl20] Blauer Engel (2020): Blauer Engel - Ressourcen- und energieeffiziente Softwareprodukte 2020, <https://produktinfo.blauer-engel.de/uploads/criteriafile/de/DE-UZ%20215-202001-de%20Kriterien-V2.pdf>, 2020.
- [Co23] Computeruniverse: Welche PC-Komponente verbraucht wie viel Strom? <https://www.computeruniverse.net/de/techblog/wieviel-strom-braucht-mein-pc>, 24.02.2023.
- [De23] Deutscher Wetterdienst: Klimadaten Deutschland - Monats- und Tageswerte (Archiv), <https://www.dwd.de/DE/leistungen/klimadatendeutschland/klarchivtagmonat.html?nn=16102>, 26.02.2023.
- [Ge20] Georgiou, Stefanos; Rizou, Stamatia; Spinellis, Diomidis: Software Development Lifecycle for Energy Efficiency. In: ACM Comput. Surv. 52 (4), S. 1–33, 2020.
- [Go23] Gosh, A., Die mächtige Open Source Lösung: R, https://www.inwt-statistics.de/blog-artikel-lesen/Statistik-Software_R.html, 10.02.2023.
- [Gu21] Guldner, A.: Mobiles Messgerät, <https://gitlab.rlp.net/green-software-engineering/mobiles-messgerät>, 2023, 27.05.2023.
- [Ju22] Junger, D.; Wohlgenuth, V.; Kammer, E.: Conception and test of a measuring station for analyzing the resource and energy consumption of material flow-oriented environmental management information systems (EMIS). In: EnviroInfo, Bonn, 211–220, 2022.
- [Ju23] Junger, Dennis, et al.: Design and implementation of a lecture for teaching current Green Coding approaches and practices at the HTW Berlin. In: Informatik 2023. Gesellschaft für Informatik e.V., Bonn, 2023. accepted for publication
- [Ke18] Kern, E.; Hilty, L. M.; Guldner, A.; Maksimov, Y.V.; Filler, A.; Gröger, J.; Naumann, S.: Sustainable software products—Towards assessment criteria for resource and energy efficiency. Future Generation Computer Systems 86, 199–210, 2018.
- [Lu23] Lubner, S.; Nico, L.: Was ist MATLAB? <https://www.bigdata-insider.de/was-ist-matlab-a-789607/>, 10.02.2023.
- [Os23a] OSCAR: Schritt-für-Schritt-Anleitung zur Softwarebewertung hinsichtlich Energieverbrauch & Hardwarekapazitäten, <https://oscar.umwelt-campus.de/session/c4304634d90ce521f0a3b25f56e73fbb/download/download/DurchfuhrungEinerMessung?w=>, 23.02.2023.
- [Os23b] OSCAR: v0.190404, <https://oscar.umwelt-campus.de>, 12.02.2023.
- [PY23] PYPL PopularitY of Programming Language index, <https://pypl.github.io/PYPL.html>, 20.02.2023.