



Temas em Psicologia

ISSN: 1413-389X

comissaoeditorial@sbponline.org.br

Sociedade Brasileira de Psicologia
Brasil

Torro-Alves, Nelson; Fernández-Calvo, Bernardino; Aznar-Casanova, José Antonio; de
Arruda Campos Neto, Armindo; dos Santos, Natanael Antonio
The Use of Programming Languages and Computer Software in Psychological Science
Temas em Psicologia, vol. 22, núm. 3, diciembre, 2014, pp. 603-612
Sociedade Brasileira de Psicologia
Ribeirão Preto, Brasil

Available in: <http://www.redalyc.org/articulo.oa?id=513751529007>

- How to cite
- Complete issue
- More information about this article
- Journal's homepage in redalyc.org

redalyc.org

Scientific Information System

Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal

Non-profit academic project, developed under the open access initiative

The Use of Programming Languages and Computer Software in Psychological Science

Nelson Torro-Alves¹

*Departamento de Psicologia da Universidade Federal da Paraíba,
João Pessoa, Paraíba, Brasil*

Bernardino Fernández-Calvo

*Departamento de Psicologia da Universidade Federal do Rio Grande do Norte, Natal,
Rio Grande do Norte, Brasil*

José Antonio Aznar-Casanova

Departamento de Psicología Básica de la Universidad de Barcelona, Barcelona, España

Armindo de Arruda Campos Neto

Instituto Federal de Mato Grosso, Cuiabá, Mato Grosso, Brasil

Natanael Antonio dos Santos

*Departamento de Psicologia da Universidade Federal da Paraíba,
João Pessoa, Paraíba, Brasil*

Abstract

In the present paper, we review some programming languages and computer software used to investigate basic psychological processes. In the first section, we address the principles and logical structure of the most common programming languages (e.g. C, C++, Java, Matlab, R, Python), and in the second section we present some of their applications to psychological research. Finally, we discuss the importance of computer programming as a tool in the field of psychology and its relationship with technological innovation.

Keywords: Basic psychological processes, programming language, software, technological innovation

O Uso de Linguagens de Programação e Softwares na Ciência Psicológica

Resumo

No presente artigo foram revisadas algumas linguagens de programação e *softwares* usados na investigação dos processos psicológicos básicos. Na primeira seção, foram abordados os princípios e a estrutura lógica das linguagens mais comuns de programação (e.g. C, C++, Java, Matlab, R e Python), e na segunda seção foram apresentadas algumas de suas aplicações na pesquisa psicológica. Ao final, são discutidas a importância da programação como uma ferramenta no campo da psicologia e sua relação com a inovação tecnológica.

¹ Mailing address: Departamento de Psicologia, Centro de Ciências Humanas, Letras e Artes, Universidade Federal da Paraíba, Campus I, Cidade Universitária, João Pessoa, PB, Brasil 58051-900. E-mail: nelsontorro@yahoo.com.br

This work received financial support from *National Council for Scientific and Technological Development* (CNPq), Grant nº 479366/2011-0 (Edital 14/2011) and Grant nº 478966/2012-1 (Edital 14/2012), delivered to the first and second author, respectively.

Palavras-chave: Processos psicológicos básicos, linguagem de programação, *software*, inovação tecnológica.

El Uso de los Lenguajes de Programación y los Programas Informáticos en la Ciencia Psicológica

Resumen

En el presente artículo revisamos algunos de los lenguajes de programación y programas informáticos utilizados en la investigación de los procesos psicológicos básicos. En la primera sección, abordamos los principios y la estructura lógica de los lenguajes más comunes de programación (e.g., C, C++, Java, Matlab, R, y Python), y en la segunda sección, presentamos algunas de sus aplicaciones en la investigación psicológica. Al final, discutimos la importancia de la programación como una herramienta en el campo de la psicología y su relación con la innovación tecnológica.

Palabras claves: Procesos psicológicos básicos, lenguaje de programación, *software*, innovación tecnológica.

Basic Psychological Processes and Programming Languages

Psychobiology is a branch of psychology concerned with basic psychological processes such as attention, perception, memory, learning, emotion, motivation, thought, and language. In undergraduate Psychology courses in Brazil, the study of basic processes has an emphasis on experimentation, especially in the fundamental courses, such as General and Experimental Psychology I, II, and III. The empirical approach to investigate the relationships between mental processes and brain functioning has benefited from the development of programming languages and computer software. These tools allow the manipulation and control of stimuli and variables, the automation of complex experimental procedures, and the processing of large amounts of data. In addition, they also allow the computational simulation of basic processes, a research field known as computational neuroscience. An important reference in the field is the connectionist approach described in classical book *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Rumelhart & McClelland, 1986), in which authors analyzed computational simulation and machine learning as applied to perception and other cognitive functions.

The simulation of a given basic process (e.g. shape recognition) is based on a neurophysiological model, which is used to formulate a mathematical model in the format: input signal representation → processing algorithm → output signal. A detailed model can be implemented in a computer and allows the simulation of cognitive processes. If the processing algorithm produces outputs that are comparable to human (or animal) responses, then there are evidences that the theoretical model is accurate and has explanatory power. Therefore, computational simulation is a useful strategy to test theories about mental functioning and cognition, allowing to establish their strengths and weaknesses (e.g., Gerstner, Sprekeler, & Deco, 2012).

The integration between research on basic processes and information technology increases reliability and validity in the fields of psychobiology and cognitive neuroscience. Programming languages and computer software not only provide a better use of laboratory resources, but also a better way to control stimuli presentation, randomization, and other experimental variables, reducing the impacts of confounding factors (e.g. equipment precision in stimuli onset/offset and response registration). Furthermore, much has been gained in terms of data processing and data visualization.

There is a wide range of programming languages that can be used in psychological research, however, special attention should be given to choose the most appropriate one. The right choice will depend on several factors, such as the object of research, the equipment required in experimentation, the variables to be controlled (e.g. physical characteristics of stimuli), and the need of reusing codes in future studies. An incorrect choice may increase programming time, delay data collection and experimental results. In addition, some languages may be unsuitable to control certain laboratory equipments.

There are some properties crucial for choosing a specific programming language, such as ease of learn and use, reliability in application development and error handling, portability, readability, and reusability of the source-codes (i.e. to reuse pieces of code in other projects; Couto, Virtuoso, & Martins, 2012).

In psychological research, one of the main advantages of a programming language is portability (i.e. a program may be distributed for several platforms without requiring further adaptations). Given the fast development of information technology, there are currently many programming languages, and the most common ones, such as C, C++, Java, Matlab, Python, and R are briefly presented below. Although we may focus on some programming languages in the present work, it is important to highlight that many other languages could also work perfectly well in psychology and neuroscience research.

C Programming

Since the classical book *The C Programming Language* was published by Brian Kernighan and Dennis Ritchie in 1978, the world of computing has undergone a revolution. C has evolved, albeit moderately, and spread far beyond its original purpose, i.e. a language for the UNIX operating system (Kernighan & Ritchie, 1988). C offers a text mode which is logical, structured, and easy to read in comparison with hardware-specific assembly languages. Whereas assembly languages work at low-level instructions that are directly translated into machine

code to control hardware operations, Kernighan and Ritchie (1978) proposed a high-level language similar to English, which is an abstraction of low-level machine language. Codes written in C are translated into an assembly language by a compiler, and then an executable program is generated (Hekma, 2005).

The main characteristic of C is the use of structured programming, i.e. the program is organized in independent blocks (functions, classes, sub-routines etc.) and may include other blocks that communicate with the main block by means of variables, parameters and arguments.

Given its characteristics, C is considered a general-purpose language that can run on any microprocessor that has its own C compiler. It is a language suitable to develop codes and software that are easy to understand and maintain (Pardue, 2005). Despite its popularity and advantages in accessibility, nowadays C has several drawbacks in portability, reusability, and readability in comparison with more recent programming languages such as C++ and Java, given that C codes may become complex, difficult to read and to maintain (Couto et al., 2012).

C++ Programming

C++ was created by Bjarne Stroustrup in the early 1980s, it was standardized in 1998 by the American National Standards Institute (ANSI) and by the International Standards Organization (ISO). It is a high-level, object-oriented language that works with classes and objects, which are “packages” of structured codes (similar to C language) designed to process specific information. In object-oriented languages such as C++ and Java, objects are the building blocks of programming and they may be reused in other projects (Hekma, 2005).

Three steps are necessary to create a computer executable program. First, the program should be written (a set of computational orders) and compiled, that is, the programming language codes are translated into machine codes in binary system (the only language that the CPU understands), producing an executable file. Second, the debugging process is responsible to detect exe-

cution and logic errors that occurred during the compilation phase. Finally, the program is submitted to be intensively tested by the users.

C++ provides a convenient notation for implementing algorithms, the programmers do not have to think in low-level terms and they can focus on specific functions of a program (Hekma, 2005). It has the advantages of reusability and readability, however, it combines the difficulties of both structured and object-oriented programming, requiring time and effort to be learned (Couto et al., 2012).

Java Programming

Java was created by the company Sun Microsystems in the early 1990s because its developers realized that microprocessors were being increasingly used in numerous electronic devices by a broad network of consumers. Led by James Gosling, the company developed Java – a language suitable to write programs that can run on a wide range of computer systems and controlling devices (Deitel & Deitel, 2011).

When the Web exploded in popularity in 1993, the company realized the potential of Java to add dynamic content (e.g. animations and interactivity) to web pages. This language is now used to develop large-scale enterprise applications, enhance the functionality of web servers and provide applications for consumer devices. Today, Java is the language most widely used in software development (Deitel & Deitel, 2011).

When compared with languages like C and C++, Java has advantages in reliability, portability, readability, and reusability (Couto et al., 2012), providing excellent resources to develop programs.

Matlab Programming

Matlab is the commercial software most widely used for numerical processing (Sharma & Gobbert, 2010). Among other programming languages, Matlab enjoys greater popularity in quantitative studies in mathematics, statistics, engineering, and other scientific domains (Brewster & Gobbert, 2011). Basically, uses in the programming of its scripts ANSI C (or standard C).

Its first version was written by Cleve Moler in 1970 to provide a simple and interactive way to write programs, given that codes do not require compilation. In the Matlab environment, it is possible to write scripts or simple commands to execute instructions at runtime, allowing operations with variables and data registration, which makes it very versatile (Wallisch et al., 2008). The strengths of Matlab include powerful programming tools, highly advanced algorithms, extensive data manipulation, and many graphical features. It performs fast numerical processing and has a large number of built-in functions and libraries called toolboxes (e.g. statistics, optimization, image processing, neural networks etc.). Matlab also provides useful tools for data visualization and simulation (Wallisch et al., 2008).

One of the main advantages of Matlab is the possibility of using external libraries (toolboxes), many of which are freeware. For example, the Psychophysics toolbox (or Psychtoolbox) facilitates the development of experiments to investigate basic psychological processes, such as visual perception (Kleiner, Brainard, & Pelli, 2007; Pelli, 1997).

Other toolboxes are also useful to research on basic processes. The Signal Processing Toolbox supports a wide range of signal processing operations, such as waveform generation, filter design, parametric modeling, and spectral analysis. The Image Processing Toolbox (The MathWorks, Inc., Natick, MA, USA) offers a series of algorithms and graphic tools for image processing and visualization, allowing many operations such as image enhancement, focus, blur, feature detection, noise reduction, image segmentation, geometric transformations etc.

Python Programming

Python is a very popular portable, powerful and open source programming language, which provides various forms of programming and it is believed to be extremely easy to handle (Lutz, 2009). It is considered a high-level language program, and, therefore, it demands less time to write, being easier to read and maintain (Downey, 2008).

Programming gains a lot in readability, consistency and quality of software. Furthermore, it is reusable and easier to maintain than traditional script languages. The uniformity of Python code makes it easy to understand, and it is typically one third to one fifth of the size of C++ or Java. It can be executed immediately, without the steps of compilation and long links demanded by some other tools. This language provides high-speed programming (Lutz, 2009).

Python has extensive support library, for example, NumPy (<http://www.numpy.org>) extension, and it has been described as a free and more powerful equivalent to the Matlab numerical programming system. It also has great integration, being able to call C, C++ libraries and be integrated with Java (Lutz, 2009).

There is a plethora of packages in Python for a variety of applications, and new packages, as well as improvements to the existing ones, are available daily. A good example is the PsychoPy (<http://www.psychopy.org>), widely used in neuroscience, psychology and psychophysics experiments, offering entirely free libraries. The PsychoPy scripts are designed to be extremely easy to read and write, while allowing the user to customize the stimuli. The tools provided in the package enable the performance from long stimulus presentation and the collection of answer to a simple analysis of data to mounting a psychometric function. This package also has the advantage of being highly extensible and the entire system can evolve through users' contributions.

R Programming

The R environment is a set of software and programming language based on S, for data analysis and visualization. It is a language with a runtime environment with graphics, a debugger for testing other programs, in addition to having the ability to run programs stored in script files (Rizzo, 2008).

R programming is fast becoming the standard for statistical computing and analysis in the social sciences, engineering and others. It is an open source statistical language widely used for manipulating statistics and produce publication-quality graphs (Gardener, 2012).

The R programming has gained much popularity in the scientific community, especially in the development of statistical software and data analysis of the experiments. The platform R (R Development Core Team, 2008, <http://www.r-project.org>) has the advantage of being free software, as a powerful and affordable tool in the context of the classroom and research. It has packages that let you perform functions in MATLAB® R. As a strong point, it has the quality of producing well-designed and high-quality printing graphics. In addition, there are numerous functions for handling, import and export data, being multiplatform, with versions for Windows, MacOS, GNU/Linux and Unix. You can load external data in many different ways, including spreadsheets such as Microsoft Excel® and Open-Office, database and even other packages, such as MINITAB®, and SPSS® (Henning, Alves, Konrath, & Samohyl, 2009).

E-Prime Programming

E-Prime is a suite of applications widely used to computerize experiments, providing an environment to develop complex research projects, collect and analyze data with high precision and high flexibility. It has excellent graphical interface besides generating great precision in the presentation of stimuli and data collection. Its suite also includes a data management utility that allows you to filter, edit, analyze and export data. Moreover, it can recover files which had broken or corrupted data during the experiment. The E-prime also has the capability to play digital movies as stimuli and to digitally record vocalization of the participants. All this information, as well as access to the program and technical support, are available and earmarked for studies involving experimental psychology, as it can be found on the site <https://www.pstnet.com/eprime.cfm>.

The E-Prime is widely used in studies involving psychophysics, especially the auditory perception, as in the research of Pütz and Sicola (2010) involving the multilingual lexicon, in which the program administered auditory stimuli of the experiment. In the study by Costa and Zimmer (2012), the software was used to evalua-

te the performance of older adults in nonverbal task linked to executive functions. The E-prime was of great benefit to the management of colo-

red visual stimuli, since its introduction to data collection. In Table 1, we present some sites with about tutorial and codes for use in psychology or helpful to start learning programming.

Table 1
Examples of Sites with Tutorial and Programming Codes for Use in Psychology or Helpful to Start Learning Programming

Name	Description	Website
Psychophysics Toolbox Version 3 (PTB-3)	Free set of Matlab and GNU/Octave functions for vision and auditory research.	http://psychtoolbox.org/
Code Academy	Interactive website teaching the basics of Javascript and other programming languages.	http://www.codecademy.com/
Cogent	Graphic toolbox and psychophysics experiments in Matlab.	http://www.vislab.ucl.ac.uk/cogent.php
C programming	Tutorial and Codes in C/C++	http://www.cprogramming.com/
E-prime	E-prime website	https://www.pstnet.com/eprime.cfm
Matlab for Psychologists	Basic guide for running and analyzing psychology experiments.	http://antoniahamilton.com/matlab.html/
MATLAB for Behavioral Scientists.	MATLAB code and program outputs that appear in the reference book.	http://www.matlab-behave.com/ Rosenbaum (2007)
PsychoPy	Open-source application to present stimuli and collect data in neuroscience, psychology and psychophysics experiments.	http://www.psychopy.org/
R	Free software environment for statistical computing and graphics.	http://www.r-project.org/

Programming Languages and Software: Applications to Research in Perception and Cognition

Programming languages such as C, C++, and Java were not initially designed for applications in psychological research. However, given their versatility and precision in controlling stimuli characteristics and experimental procedures, they have become useful tools in the study of basic psychological processes, allowing to investigate, for example, emotional perception (Alves, Aznar-Casanova, & Fukusima, 2009), detection of spatial frequency (Santos, Andrade, & Fernández-Calvo, 2013), spatial navigation in virtual environments (Aznar-Casanova, Alves, & da Silva, 2012) and age perception (Aznar-Casanova, Torro-Alves, & Fukusima, 2010).

C++ can be used to control important aspects of stimuli presentation, such as: (a) millisecond precision timing (i.e. stimulus onset, duration and offset); (b) computer monitor calibration (i.e. luminance and contrast); (c) visual stimuli size and location on display; (d) auditory stimuli characteristics (frequency, amplitude, and pitch); (e) presentation mode (static or dynamic). Furthermore, computer programs also allow the controlling of important features of the experimental design, for example, stimuli presentation may occur at random or follow a pre-established pattern. In psychophysics, the programs allow the implementation of sophisticated procedures, such as adaptive methods that otherwise would be difficult to control.

In adaptive methods, the intensity of a given stimulus depends on the participant's previous

response to another stimulus. Such methods can be used to measure sensory thresholds – the stimulus minimal amount of energy that an individual can detect, for example, the lowest sound intensity. The double staircase method (Cornsweet, 1962) consists of two random series of stimuli: a descending series (from high- to low-intensity stimuli) and an ascending one (from low- to high-intensity stimuli). In auditory perception, an ascending series may start at 5 decibels (dB), with intensity increasing in steps of 5 dB until stimulus detection. On the other hand, the descending series starts with audible, high-intensity sounds, which decrease progressively until no detection occurs. The experiment finishes at the junction of the series, which indicates the sensory threshold, in this case, how many decibels are necessary for stimulus detection. The double staircase method may have complex rules, for example, to decrease stimulus intensity after three correct answers and to increase intensity after a failure in detection (Torro-Alves, Santos, & Simas, 2012). These psychophysical methods, which are difficult to implement manually or by other means, can be easily implemented with a flexible programming language such as C++.

In the study of basic psychological processes, many experimental procedures require programming, for example, eye tracking – the measuring of eye fixations and saccadic movements during visual scene perception. Eye movements can reveal important information to understand perceptual, attentional, and cognitive processes involved in vision. In order to record eye movements, a device known as Eye-tracker is used coupled to a computer, and the most common type of Eye-tracker projects on the participant's eye an infrared light, which is reflected back by the retina and cornea. This reflection is then captured by the device's camera, and the vector from the pupil center to the corneal reflection point is used to determine the eye-gaze direction (i.e. which part of an image one is looking; Hansen & Ji, 2010).

Eye-trackers based on infrared light can record information in certain frequencies, such as 50, 100, or 250 Hertz (Hz). To give an example,

a device set at 250 Hz records eye positions at every 4 milliseconds (ms), and an experiment running for 10 minutes produces an output consisting of 150,000 lines. The data comprising the x/y coordinates of eye position (and other information such as pupil diameter, blinks etc.) should be analyzed in order to detect patterns of eye movements. Therefore, it is essential to use computer software and programming languages to control Eye-tracker functioning and process large amounts of data.

In studies using Eye-tracker, data analysis is feasible with the use of a programming language like Matlab and C++, which performs data processing to produce visual representations of eye movement patterns over the observed pictures. In addition, the program may also detect the number of fixations in regions of interest. These fixations represent a record of behavioral/physiological processes underlying cognitive and attentional processes.

At present, there are some application software specially designed for analyzing eye movements. In some cases, the user interface is friendly, allowing the registration and visualization of measures of interest without requiring programming. For example, the software BeGaze 2 (SensoMotoric Instrument – SMI) presents “heat maps” in static or dynamic images, and can perform quantitative analyses of saccades, fixations, winks, and reaction time. These data can be easily exported to a computer software such as Microsoft Excel © or the IBM Statistical Package for Social Sciences (SPSS v.18.0 IBM SPSS Statistics for Windows, IBM Corp., NY, USA).

Python is one of the most sought programs by fields of psychology and neuroscience, especially those directed to psychophysics and visual perception. A good example of its application is the study of Aguilar and Castet (2011) which used the library PsychoPy to create a software that managed visual stimuli and integrated an eyetracker EyeLink II (EL II – headmounted binocular eyetracker – SR Research Ltd., Mississauga, Ontario, Canada) in the eye-tracking trying to simulate retinopathy in healthy individuals.

An Eye-tracker may also operate using a pre-established programming language. For example, the device produced by the company Cambridge Research Systems was developed with a MATLAB interface. It has a specific toolbox that allows the device calibration and the control of stimuli presentation, response collection, and data analysis. Thus, MATLAB – its programming language and toolboxes, provide many resources suitable for each step of the research process: from experimental procedure to device control and data processing.

Research on Basic Psychological Processes and Technological Innovation

In Brazil, there is a tendency to consider Psychology apart from other scientific and technological domains, possibly due to its link to clinical practice or to its deep relationships with Philosophy. This perspective is misleading, however, given that many branches of psychology – especially research on basic processes – have important intersections with other scientific disciplines. To give some examples of technological innovation developed with the help of psychological research, we can mention statistical methods and the design of devices of cockpit airplanes, air traffic control systems, and other security systems such as the centered high-mounted brake light (American Psychological Association [APA], 2010). In particular, research on basic psychological processes motivated the development of many application software to control stimuli and experimental procedures, such as SuperLab (<http://www.superlab.com.br>), E-Prime (<https://www.pstnet.com/eprime.cfm>), and the Psychophysics ToolBox (<http://www.psychtoolbox.org>).

The study of mental processes has also influenced the development of techniques for recording electric and magnetic signals associated with brain functioning, resulting in some functional neuroimaging technologies such as electroencephalography (EEG), Functional Magnetic Resonance Imaging (fMRI), Positron Emission Tomography (PET), and Magnetoen-

cephalography (MEG). These techniques have also promoted the development and improvement of application software and algorithms used for data processing.

Given the interface between psychology and technological sciences, the APA launched in 2010 a task force to include psychology in the list of STEM disciplines (i.e. Science, Technology, Engineering and Mathematics). The motivation is that psychology may be considered a basic science of behavior, using scientific methods and laboratory resources to test hypotheses about human behavior. In the history of psychology, all sorts of technological resources have been used, such as the apparatus used in animal research, as well as in human experimental psychology, for example, virtual reality techniques, functional neuroimaging, computer software and programming languages. Therefore, there is wide support that psychological research is strictly linked to technological innovation. In Brazil, chances in the traditional programs of the undergraduate courses of psychology would be important to strength the connection between the psychology and technology, for example, with the inclusion of programming language courses.

Conclusions

The use of programming languages and computer software has been essential to the development of Psychology. If these tools were not available, many cognitive processes could not have been studied or investigated with appropriate methods. Interestingly, there is also a contribution of psychology, in particular of research on basic psychological processes, to the process of technological innovation and the development of new programming languages and software.

In summary, to consider psychology as a basic science linked to technological innovation may have positive impacts on its development and its relationships with other disciplines. In Brazil, to acknowledge this fact would make possible the inclusion of psychology in the Government priority areas of Science and Technology. In this case, psychological research would

have access to huge federal funding, such as the Science without Borders Program, which promotes the consolidation, expansion, and internationalization of Brazilian science.

References

- Aguilar, C., & Castet, E. (2011). Gaze-contingent simulation of retinopathy: Some potential pitfalls and remedies. *Vision Research*, 51, 997-1012. doi:10.1016/j.visres.2011.02.010
- Alves, N. T., Aznar-Casanova, J. A., & Fukusima, S. S. (2009). Patterns of brain asymmetry in the perception of positive and negative facial expressions. *Laterality*, 14(3), 256-272. doi:10.1080/13576500802362927
- American Psychological Association. (2010). *Psychology as a Core Science, Technology, Engineering, and Mathematics (STEM) discipline*. Washington, DC: Author. Retrieved from <http://www.apa.org/science/about/psa/2010/08/stem-report.pdf>
- Aznar-Casanova, J. A., Alves, N. T., & da Silva, J. A. (2012). Human navigational strategies in a virtual environment. In C. Sik Lányi (Ed.), *Applications of virtual reality* (pp. 73-88). Rijeka, Croatia: In Tech.
- Aznar-Casanova, J., Torro-Alves, N., & Fukusima, S. (2010). How much older do you get when a wrinkle appears on your face? Modifying age estimates by number of wrinkles. *Neuropsychology, Development, and Cognition. Section B, Aging, Neuropsychology and Cognition*, 17(4), 406-421. doi:10.1080/13825580903420153
- Brewster, M. W., & Gobbert, M. K. (2011). *A comparative evaluation of Matlab, Octave, FreeMat, and Scilab on tara* (Technical Report HPCF-2011-10). Baltimore, MD: University of Maryland, Baltimore County High Performance Computing Facility. Retrieved from <http://userpages.umbc.edu/~gobbert/papers/BrewsterGobbertTR2011.pdf>
- Cornsweet, T. (1962). The staircase method in psychophysics. *American Journal of Psychology*, 75, 485-491.
- Costa, A. L. P. A., & Zimmer, M. C. (2012). Desempenho de idosos com presbiacusia em tarefas de controle inibitório. *Revista da Sociedade Brasileira de Fonoaudiologia*, 17(2), 151-155. doi:<http://dx.doi.org/10.1590/S1516-80342012000200008>
- Couto, M. A., Jr., Virtuoso, G. H. F., & Martins, P. J. (2012). Propriedades desejáveis a uma linguagem de programação: Uma análise comparativa entre as linguagens C, C++ e Java. *Anais SULCOMP*, 1(1). Recuperado em <http://periodicos.unesc.net/index.php/sulcomp/article/viewArticle/796>
- Deitel, H. M., & Deitel, P. J. (2011). *Java how to program: International version* (9th ed.). London: Pearson Education.
- Downey, A. (2008). *Think Python. How to think like a computer scientist*. Needham, MA: Green Tea Press.
- Gardener, M. (2012). *Beginning R: The statistical programming language*. Indianapolis, IN: John Wiley & Sons.
- Gerstner, W., Sprekeler, H., & Deco, G. (2012). Theory and simulation in neuroscience. *Science*, 338(6103), 60-65. doi:10.1126/science.1227356
- Hansen, D. W., & Ji, Q. (2010). In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3), 478-500. doi:10.1109/TPAMI.2009.30
- Hekma, S. (2005). *Communication networks*. Newark, DE: PragSoft Corporation.
- Henning, E., Alves, C. C., Konrath, A. C., & Samohyl, R. W. (2009). Uma breve visão sobre aplicações em ambiente R para o ensino de engenharia. In 37^o Congresso Brasileiro de Educação em Engenharia – Cobenge. Recife, PE: Associação Brasileira de Educação de Engenharia. Recuperado em <http://dspace.universia.net/handle/2024/555>
- Kernighan, B. W., & Ritchie, D. M. (1978). *The C programming language*. Englewood Cliffs, NJ: Prentice Hall.
- Kernighan, B. W., & Ritchie, D. M. (1988). *C programming language (ANSI-C Version)* (2nd ed.). New Delhi, India: Prentice-Hall of India.
- Kleiner, M., Brainard, D., & Pelli, D. (2007). "What's new in Psychtoolbox-3?" [Abstract] *Perception*, 36(Suppl.), 14.
- Lutz, M. (2009). *Learning Python* (4th ed.). Sebastopol, CA: O'Reilly Media.
- Pardue, J. (2005). *C programming for microcontrollers: Featuring ATMEL's AVR butterfly and the free WinAVR compiler*. Knoxville, TN: Smiley Micros.

- Pelli, D. G. (1997). The VideoToolbox software for visual psychophysics: Transforming numbers into movies. *Spatial Vision*, 10(4), 437-442. doi:10.1163/156856897X00366
- Pütz, M., & Sicola, L. (2010). *Cognitive processing in second language acquisition*. Amsterdam, Netherlands: John Benjamins.
- R Development Core Team. (2008). R: A language and environment for statistical computing [Programming language]. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <http://www.r-project.org>
- Rizzo, M. L. (2008). *Statistical computing with R*. Boca Raton, FL: Chapman and Hall.
- Rosenbaum, D. A. (2007). *MATLAB for behavioral scientists*. Mahwah, NJ: Lawrence Erlbaum.
- Rumelhart, D. E., & McClelland, J. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition*. Cambridge, MA: Massachusetts Institute of Technology Press.
- Santos, N. A., Andrade, S. M., & Fernández-Calvo, B. (2013). Detection of spatial frequency in brain-damaged patients: Influence of hemispheric asymmetries and hemineglect. *Frontiers in Human Neuroscience*, 7. doi:10.3389/fnhum.2013.00092
- Sharma, N., & Gobbert, M. K. (2010). *A comparative evaluation of Matlab, Octave, FreeMat, and Scilab for research and teaching* (Technical Report HPCF-2010-7). Baltimore, MD: University of Maryland, Baltimore County High Performance Computing Facility.
- Torro-Alves, N., Santos, N. A., & Simas, M. L. B. (2012). A utilização de métodos psicofísicos na medição da sensibilidade ao contraste visual em humanos. In J. F. Landeira & S. S. Fukusima (Eds.), *Métodos em neurociência* (pp. 35-47). Barueri, SP: Manole.
- Wallisch, P., Lusignan, M. E., Benayoun, M. D., Baker, T. I., Dickey, A. S., & Hatsopoulos, N. G. (2008). *MATLAB for neuroscientists: An introduction to scientific computing in MATLAB*. Oxford, UK: Academic Press.

Received: February, 10, 2014

1st revision: May, 02, 2014

Accepted: May, 05, 2014