A PHP Application Library for Web-Based Power Systems Analysis

Simon Agamah
School of Mathematics, Computer Science and Engineering
City University
London, United Kingdom
simon.agamah.1@city.ac.uk

Lambros Ekonomou
School of Mathematics, Computer Science and Engineering
City University
London, United Kingdom
lambros.ekonomou.1@city.ac.uk

***Abstract* — Web-based power systems analysis applications are delivered via web servers that run scripting languages such as PHP. The role of the web server is usually to pass results and data between a front-end web browser and specialised back-end computation software which carries out the actual simulations and analysis. More recent versions of the web scripting languages have the computational capabilities required for power systems analysis and can handle the responsibility of modelling networks and analysing them. This provides an opportunity for a slimmer 2-tier framework in which the web server also acts as the application server thereby reducing the server resources required and simplifying access to modelling functions. A methodology for carrying out power flow calculations on a web server using PHP is described in this paper and the outcome is a PHP application library for carrying out such analysis. The library may be easily included in PHP web applications that require specific power systems analysis functions thereby allowing them access this functionality without relying on third party software separate from the web server on which they already run. The performance of the application library is also measured and discussed.**

Keywords - Client-server systems, Internet Modelling, Power system analysis computing, PHP, Software as a service, Web Applications, Web sites, XML

I. INTRODUCTION

The delivery of power systems analysis software via the internet is a topic that has been explored previously in various works [1]–[6]. There have also been various implementations ranging from industry standard commercial packages such as NEPLAN having online versions [7] to free open source packages that may be used on the web [8], [9]. Byrne, Heavey and Byrne cover web-based simulation in detail [1] and define web-based simulation as the use of resources and technologies offered by the world-wide-web (WWW) for interaction with client (web browser) and server (remote computer) modelling and simulation tools. The definition goes further to emphasize the active role of the web browser (Microsoft Internet Explorer™, Mozilla Firefox™, Google Chrome™, etc. for example) in the modelling or simulation process, either as a graphical interface or, additionally, as a container for the simulation numerical engine. This is the definition used to provide the context for web-based simulation in this paper, and it excludes simulation packages that are downloaded from a web server to a local computer and executed independent of the web browser. Such software packages are also categorised as Software as a service (SaaS) because their tools are accessed remotely as services [1].

The application architecture of web-based power systems analysis software (WBPSA) usually consists of 3 tiers as shown in Figure 1; One tier consists of the remote simulation server which runs a software that simulates a model of the network and performs calculations, the second tier is the web server that handles communication between the other tiers, and the third tier is the web browser for input of parameters and displaying results. The methodologies for WBPSA in previous research work usually adapt this pattern. Active Server Pages (ASP) and ASP.NET are used to process requests and responses in [5], [10], [11] between the simulation server running C# programs and the web browser. Java Server Pages (JSP) are used in [3] to connect a legacy simulation application based on Fortran to Java applets running in the web browser.

The same architecture is applied in commercial and open source software. NEPLAN 360 [12] is the leader in this category and provides a robust application via the web interface which has all the key features of the desktop version along with an Application Programming Interface for third party access. The legacy application runs on the simulation server and is served through a cloud computing interface to a Microsoft Silverlight™ plugin in the web browser. There is no indication that the actual simulation is done via a web server script and therefore it also has 3-tier architecture. MATLAB power systems solvers such as MATPOWER [13] may also be accessed on the web by running them on an application server with MATLAB installed and then using an intermediate scripting language such as ASP.NET to communicate with the web browser [14]. Another web-based cloud application is InterPSS [15] which is based on Google Drive ™ spread sheets for data input and results output, and a Google Apps™ script communicates with a Java based simulation engine.

The 3-tier architecture is implemented for practical reasons. One reason is that the simulation software packages which are not ready for internet modelling use a web server as an interface. The only requirement for web

978-1-5090-0206-1/15 $31.00 © 2015 IEEE
DOI 10.1109/EMS.2015.58

access will now be an upgrade of communication and data transfer protocols not the simulation methods. Another practical reason for the 3-tier architecture is that the programming languages used for building websites are not as powerful or as purposeful for computation as those used for building simulation software.

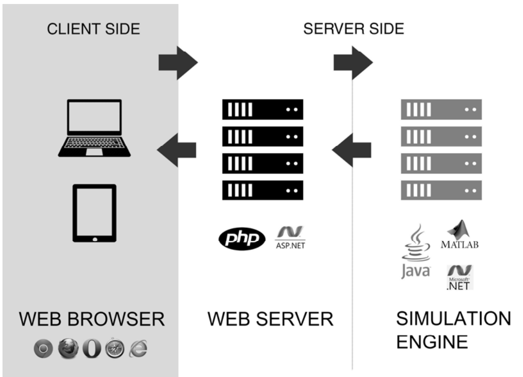

Figure 1. 3-tier architecture for web based power systems analysis

Web servers typically run scripts written in ASP, ASP.NET, JSP, Perl, PHP, etc. [16] which are designed primarily for text-based communication using Hypertext Transfer Protocols (HTTP) while simulation packages are built using more powerful languages such as C, C++, C#, Java, etc. [1], [9] because they have a wider range of tools to perform the mathematical analysis required and produce results in a relatively short amount of time. However more recent versions of the web server languages have been upgraded to provide tools for more complex computation such as Object Oriented Programming [17].

Independent third party libraries and extensions are also increasingly available for providing additional mathematical and analytical capabilities on web servers. The implication is that some of the computation that has been handled completely by the dedicated simulation tier may now be performed on the web server tier. Newer versions of more powerful scripting languages such as Python which were not originally designed for internet use may now be deployed on the web also thereby providing more opportunities for web server power systems computation. Python scripting has indeed been used for Power systems computation in [18], however in that instance it wasn't deployed on a web server.

It may be observed from the preceding review that the web programming languages used to interface simulation software for internet modelling are usually ASP.NET and JSP. These are likely choices because they are strongly linked to Visual Basic (VB) and Java respectively, as they were created by the same vendors; Microsoft [19] and Oracle [20] respectively. Therefore the models in the simulation software written in VB and Java, which are popular choices [1], can be accessed more easily in those web scripting languages. PHP [21] is the most used web server scripting language [22] but WBPSA is currently not being implemented using it. There are several possible reasons for this including the ease of interfacing existing simulation software using ASP.NET to VB and JSP to Java, and the computational limitations of PHP as a programming language for modelling and simulation.

PHP has gradually transformed into a general purpose programming language [17], [21] as web applications have become more complex and the language has been updated. Recent versions of PHP include additional features to make it easier to build complex web applications using PHP. Apart from the additional core features that make PHP more capable of general purpose programming, it also benefits from having several extensions and third party libraries to further extend its capabilities.

This paper covers the development of a PHP application library for power systems analysis. The result is a 2-tier application architecture in which the simulation modelling is performed on the web server as shown in Figure 2. The benefit of this approach is a slimmer framework that requires less computational resources. It also means that websites that have been built using PHP can include power systems models for various purposes by utilizing the library. For example, a network model operating an experimental algorithm may be published directly to a PHP website and actively operated by peer reviewers to view how conditions of the network are affected by changing parameters. This will represent an improvement in dynamic results presentation as there won't be only static defined sets of results for defined sets of parameters. To achieve this in the present form of WBPSA will require licensing fees for the commercial packages, significant programming skills and more cloud computing resources for the simulation server.

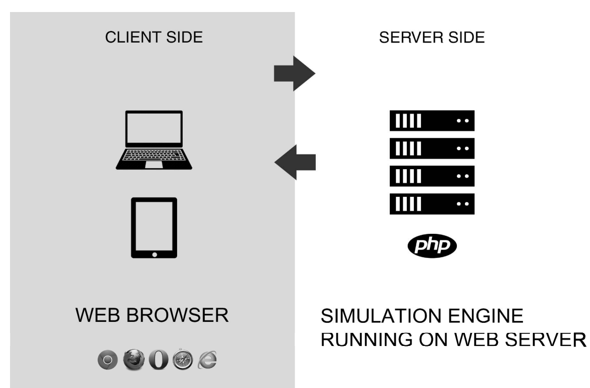

Figure 2. 2-tier architecture for web-based power systems analysis using PHP

The PHP application library discussed has key functions primarily for power flow and line flows which are fundamental studies for power systems analysis. The model requirements for these studies are reviewed to define the specifications for the PHP application. The methodology for using PHP to perform these studies is covered and finally the performance of the library is measured and discussed.

II. Requirements for Power Systems Modelling

To perform power systems studies on a network a model must be used to represent the network. Mathematical models are available for representing networks using relations between parameters. These models are usually a set of equations or a system of matrices. The studies are then performed by applying mathematical techniques and solutions to obtain the unknowns of the models. A power systems simulation tool must therefore be capable of representing the mathematical concepts that are required for the model. The mathematical requirements for using PHP to model a power systems network for the purpose of performing power flow and short circuit fault studies on it are reviewed in this section.

A. Model requirements for Network modelling

An electrical network may be represented as a system of connected impedances (Z) or admittances (Y) with through currents (I) and across potential differences (V), out of which an impedance or admittance matrix may be formed [23] as shown in equation (1) using Kirchoff's Current Law.

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \tag{1}$$

where:

$$Y_i = G_i + jB_i \tag{2}$$

and G: Conductance, B: Susceptance

From equations (1) and (2) it can be seen that the simulation software must be able to represent and compute *Matrix operations* and *Complex number operations*.

B. Model requirements for power flow study

A power flow study is used to determine the real (P) and reactive power (Q) at given buses in a network, as well as nodal voltages and voltage angles. The complex power at a given bus is given in equation in polar form.

$$P_i - jQ_i = |V_i| < -\delta_i \sum_{j=1}^{n} |V_i||V_i| < \theta_{ij} + \delta_j \tag{3}$$

Equation (3) yields a set of nonlinear algebraic equations in terms of voltage magnitude (V) and phase angle (δ). The solution to the set of equations using the Newton-Raphson method and successive approximation requires the formation of a Jacobian matrix and iteration.

The mathematical concepts required here are *matrix*, *vector*, *complex number* and *trigonometric, and nonlinear programming* functions when the real and imaginary parts are separated.

III. Methodology for Power Systems modelling using PHP

In the previous section the mathematical requirements for modelling power systems networks, power flow and short circuit faults were outlined. The operational capabilities required for the PHP application are as follows:

- Matrix
- Complex Number
- Vector
- Trigonometric functions
- Nonlinear programming

PHP does not have built-in functions for handling Matrix, complex number and vector operations. It can handle the trigonometric functions with built-in methods, and the nonlinear programming required can be handled using built in iterative functions. The methodologies for providing the required functions that are not built-into PHP are described in this section as follows.

A. Matrix Operations

PHP has built-in ordered maps called arrays [17] which stores lists of values. A matrix may be created in PHP essentially by setting the values of a given array to hold other arrays of equal sizes. So an array may contain 5 arrays each containing 5 values to create a 5 x 5 matrix. This is the same as a mathematical matrix only in arrangement and is useful for maintaining the order of the matrix to be formed. It is known as a multidimensional array.

To carry out other matrix operations such as addition, multiplication, transposing, etc. a third party library is required. In this paper the Math Matrix class [24] from the PHP Extension and Application Repository (PEAR) Math package is used. It is preferred because of the number of matrix computation functions it provides, the simplicity of inclusion and the flexibility. It also returns native PHP objects that can be re-used easily in other parts of the application. To declare a matrix using the class, the items of the matrix are first ordered using a multidimensional array, and the array is then passed to a new instance of the Math Matrix class.

B. Complex Number operations

A complex number class is provided in the PHP PEAR package [25] for operations such as conjugation, inversion and retrieving coordinates. This is used in the PHP WBPSA library for real and reactive power definition, impedance and admittance definitions.

C. Vector operations

The vector class provided in the PHP PEAR package [26] is used as a dependency for the complex number class as some of the operations require values to be defined as vectors.

D. Network data definition

The network data is defined using an XML format document which is parsed and read into memory by the PHP library. The network is defined using different custom tags for the elements, the buses and lines. Loads and generators are defined based on the direction of power flow.

E. Dependency management and library block diagram.

The dependencies for any study to be carried out are included in the program before the operations are run.

Figure 3 illustrates the dependencies and interrelationships of all components of the library in a block diagram.

IV. TESTS AND RESULTS

The library was tested for execution time, accuracy of solution and memory usage. The power flow solution was obtained using the Newton-Raphson method which uses a Jacobian matrix to obtain linearized relationships between parameters and therefore generate a solution via iteration.

Several network configurations and sizes including IEEE test networks obtained from [23] were investigated. The performance test parameters, i.e. execution time and memory usage, were observed as the size of the networks increased to investigate the relationship between network size and performance of the library.

The performance of the library may also be affected by the server resources such as the internet connection quality, processor speed and memory available to PHP however these were not tested for different specifications. The server used in this case is a WinginX web server running on a Windows 7 machine with a 64-bit 3.2GHz Intel i5 processor and 4GB RAM. The accuracy of solutions was verified by comparing solutions with solutions from a MATLAB based solver in [23]. The results of the tests are plotted in Figure 4 - Figure 7.

The importance of the performance metrics depends on the scenario in which the library is deployed. For environments where billing includes the amount of memory used the memory usage is preferably lower. In scenarios where online real-time computation is required then speed is essential and the execution time must be low. From the results in Figure 4 and Figure 5 it can be seen that the larger networks take the most time for computation as expected and this is scaled up especially because of the matrix operations. It takes a total computation time of 68 milliseconds to solve the power flow for the 6-bus network and 3,845.38 milliseconds for the 30 bus network.

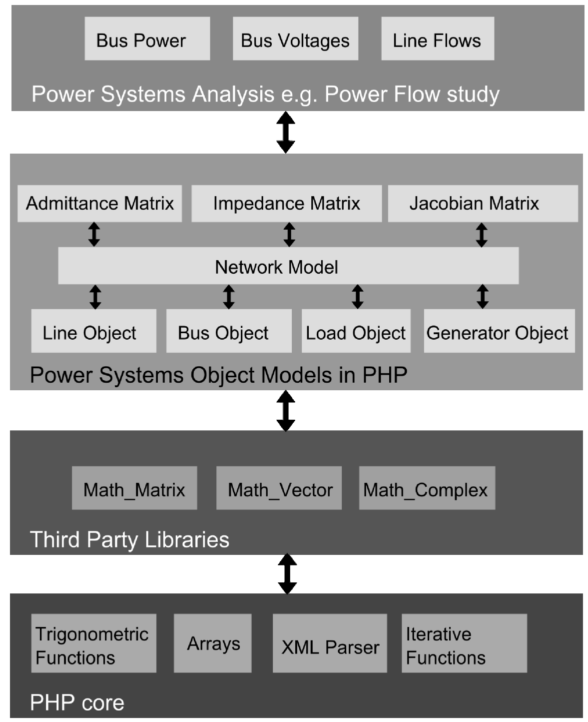

Figure 3. PHP power systems application library block diagram

The scaling factor for the total time taken for the smaller networks and the larger ones can really only be compared on a logarithmic scale.

This jump can be explained by the relative size increase patterns. The size increase for the network is done by addition but for the matrices it is done by multiplication. A 4-bus network, for example, has only one additional bus compared to a 3-bus network however the system matrices (admittance, impedance and Jacobian) have 16 elements (4 x 4) and 9 elements (3 x 3) respectively. Even for relatively close bus numbers such as the 26-bus and 30-bus networks have a relatively big difference in computation time; it takes almost twice as much time to solve the latter.

In all cases the solutions converged in at most 4 iterations. For each additional iteration the time for the key operations is added which implies that for a higher number of iterations the difference between the computation times for the successive networks also increases. To change the processing time other Matrix operation libraries may be used to replace the PHP PEAR Math_Matrix class and tested for performance or a different solution algorithm may be applied. In terms of total memory consumed (Figure 7) the step solution takes about the same time in all cases; however the total memory consumption increases in a similar pattern to the memory requirements for the key matrix operation.

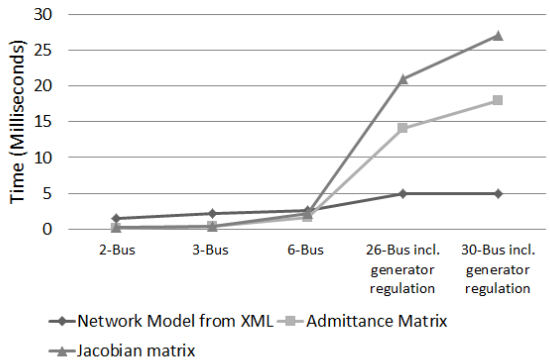

Figure 4. Computation time for key matrix operations

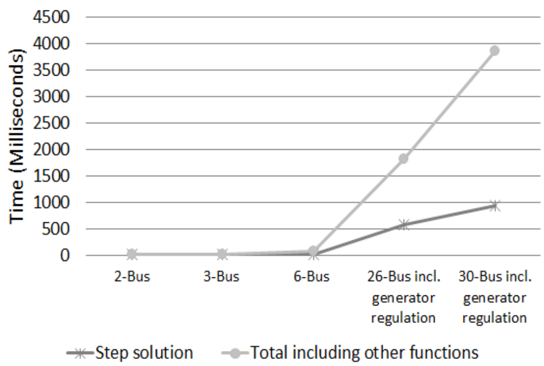

Figure 5. Total computation time for power flow solutions

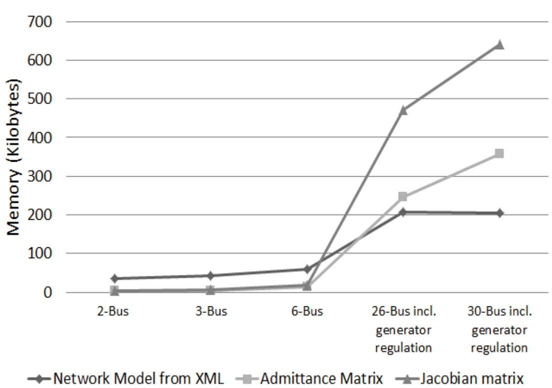

Figure 1. Figure 6. Memory usage for key matrix operations

The total computation time for the relatively large 30-bus network is still in the millisecond range which shows that the library is adequately quick. The total memory usage is not much higher than the usage for key operations because PHP automatically optimises the memory use by using references to access reused variables rather than creating new objects in memory in a process known as garbage collection [27].

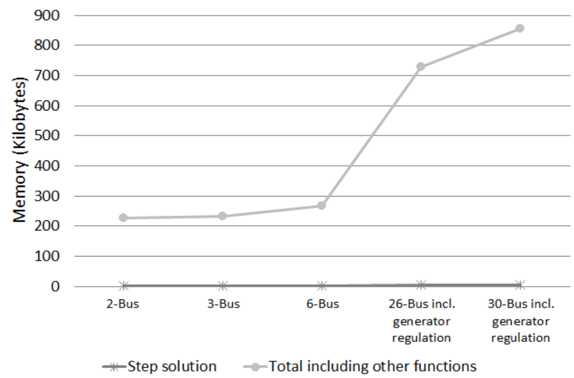

Figure 7. Memory usage for power flow solution

V. Conclusion

A PHP application library for power systems analysis using 2-tier architecture has been described in this paper.

The methodology for using PHP to perform power systems studies in web-based environments was outlined. PHP does not naturally support some of the mathematical concepts required and the third-party libraries and components required were also covered.

Tests were run to measure the performance of the library on various sizes of power networks and it was found to perform within the millisecond range for computation of relatively large networks, consuming less than 1 Megabyte of memory.

The results were also seen to be accurate when compared to the MATLAB solver and also consistent.

The improvement this approach provides over the previously used 3-tier architecture is that it requires less server resources and provides a simple way to access power systems functionality in web applications.

PHP is one of the web technologies that may be used for power systems analysis using the 2-tier architecture. Other newer web technologies may also adopt this same approach to provide power systems computation capabilities for internet modelling.

References

[1] J. Byrne, C. Heavey, and P. J. Byrne, “A review of Web-based simulation and supporting tools,” *Simul. Model. Pract. Theory*, vol. 18, no. 3, pp. 253–276, Mar. 2010.

[2] H. Chen, C. Cañizares, and A. Singh, “Web-based Computing for Power System Applications,” *Proc. North Am. Power Symp. (NAPS), San Luis Obispo, California,* 1999.

[3] S. Chen and F. Lu, “Web-based simulations of power systems,” *Comput. Appl. Power, IEEE*, no. January, pp. 35 – 40, 2002.

[4] J. Kuljis and R. Paul, “A review of web based simulation: whither we wander?,” *Proc. 32nd Conf. Winter simulation, Orlando, FL, USA*, pp. 1872–1881, 2000.

[5] R. Leou and Z. Gaing, “A Web-based load flow simulation of power systems,” *IEEE Power Eng. Soc. Summer Meet. Chicago, IL, USA*, pp. 1587–1591, 2002.

[6] S. Meyer zu Eissen and B. Stein, “Realization of Web-based simulation services,” *Comput. Ind.*, vol. 57, no. 3, pp. 261–271, Apr. 2006.

[7] Neplan AG and BCP Switzerland, “NEPLAN 360 Overview.” [Online]. Available: http://www.neplan.ch/html/e/e_PowerSystems_Properties_default_web.htm.

[8] IEEE Open Source Software Task Force and F. Milano, “IEEE Open Source Software.” [Online]. Available: http://ewh.ieee.org/cmte/psace/CAMS_taskforce/software.htm. [Accessed: 10-Feb-2013].

[9] Open Electrical, “Power Systems Analysis Software,” 2013. [Online]. Available: http://www.openelectrical.org/wiki/index.php?title=Power_Systems_Analysis_Software.

[10] J. Yang, F. Lin, and Y. Fu, “Development of a Web-Based Software for Micro Power System Design,” *2010 Int. Conf. Electr. Control Eng. Wuhan, China*, pp. 2748–2751, Jun. 2010.

[11] S. Tan and J. Yang, “Internet-based platform for power system simulating and planning,” in *2011 Second International Conference on Mechanic Automation and Control Engineering, Inner Mongolia, China*, 2011, pp. 2271–2274.

[12] BCP Busarello + Cott + Partners AG, “NEPLAN 360 Overview.” [Online]. Available: http://www.neplan.ch/html/e/e_PowerSystems_Properties_default_web.htm. [Accessed: 03-Mar-2013].

[13] F. Milano, “An open source power system analysis toolbox,” *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1199–1206, 2005.

[14] MathWorks, “Call MATLAB functions from a Web Application,” *MATLAB R2014a Online Manual; MATLAB COM Automation Server*. [Online]. Available: http://www.mathworks.co.uk/help/matlab/matlab_external/call-matlab-functions-from-a-web-application.html.

[15] InterPSS Community, “InterPSS 2.0 Manual and Documentation.” [Online]. Available: https://sites.google.com/a/interpss.org/interpss/Home/interpss-2-0.

[16] W3Techs: Web Technology Surveys, “Usage of server-side programming languages for websites,” *Web Technology Surveys*, 2014. [Online]. Available: http://w3techs.com/technologies/overview/programming_language/all.

[17] D. Powers, *PHP object-oriented solutions*. Berkeley, CA, United States of America: Apress, 2008.

[18] F. Milano, *Power system modelling and scripting*, First. Ciudad Real, Spain: Springer, 2010.

[19] I. Spaanjaars, *Beginning ASP.NET 4.5.1 in C# and VB*. Indianapolis: John Wiley & Sons, 2014.

[20] J. Murach, *Murach’s Java Servlets and JSP, 3rd Edition (Murach: Training & Reference)*. 2014.

[21] The PHP Group, “What is PHP?,” *PHP Official Website*. [Online]. Available: http://www.php.net/manual/en/intro-whatis.php.

[22] L. Eshkevari, F. Dos Santos, J. R. Cordy, and G. Antoniol, “Are PHP applications ready for Hack?,” in *Software Analysis, Evolution and Reengineering (SANER), 2015 IEEE 22nd International Conference on*, 2015, pp. 63–72.

[23] H. Saadat, *Power System Analysis*, Second Edi. Milwaukee: McGraw Hill, 2004.

[24] J. M. Castagnetto, “PEAR: Math_Matrix,” *PEAR Packages*, 2010. [Online]. Available: http://pear.php.net/package/Math_Matrix/.

[25] J. M. Castagnetto, “PEAR: Math_Complex,” *PEAR Packages*, 2010. [Online]. Available: http://pear.php.net/package/Math_Complex/.

[26] J. M. Castagnetto, “PEAR: Math_Vector,” *PEAR Packages*, 2010. [Online]. Available: http://pear.php.net/package/Math_Vector/docs/latest/li_Math_Vector.html.

[27] The PHP Group, “PHP Reference Counting.” [Online]. Available: http://php.net/manual/en/features.gc.refcounting-basics.php. [Accessed: 24-Aug-2015].