



HELMpy, Open Source Package of Power Flow Solvers, Including the Holomorphic Embedding Load Flow Method (HELM), Developed on Python 3

TULIO MOLINA

JUAN ORTEGA

JUAN MUÑOZ 

**Author affiliations can be found in the back matter of this article*

SOFTWARE METAPAPER

][ubiquity press

ABSTRACT

HELMpy is an open source software for power flow solution developed on Python 3. Specifically, two algorithms for power flow solution are implemented in this software: a Holomorphic Embedding Load-flow Method (HELM), and a Newton-Raphson (NR) based power flow method, using either a classical single slack bus model or a distributed slack bus model. HELMpy is aimed at supporting academic research, emphasizing on the development of open source software related to the HELM. To ease the comprehension and modification of HELMpy's code, it is properly commented and structured as a group of Python modules. These modules are composed of sets of functions used for solving the power flow problem according to the previously mentioned algorithms. HELMpy is distributed under the GNU Affero General Public License v3.0 or later and it is available on GitHub. (<https://github.com/HELMpy/HELMpy>).

CORRESPONDING AUTHOR:

Juan Muñoz

Pontificia Universidad Católica de Chile, Escuela de Ingeniería, Chile

juancarlmunoz@ing.puc.cl

KEYWORDS:

Power flow; HELM; Newton Raphson method, Distributed slack bus

TO CITE THIS ARTICLE:

Molina T, Ortega J, Muñoz J
2021 HELMpy, Open Source Package of Power Flow Solvers, Including the Holomorphic Embedding Load Flow Method (HELM), Developed on Python 3. *Journal of Open Research Software*, 9: 23. DOI: <https://doi.org/10.5334/jors.310>

(1) OVERVIEW

1. INTRODUCTION

Software for power flow problem solution are key tools for power system planning and analysis. In fact, decisions on operational and investment strategies to guarantee the proper and secure operation of power systems mainly rely on power flow analysis. The solution to the power flow problem consists in calculating voltage magnitudes and angles at each bus of a power system under steady-state and balanced conditions. This is carried out by solving a set of nonlinear equations referred to as power balance equations [1]. Most commonly used algorithms to solve the power flow are based on iterative methods, being the Newton Raphson (NR) method of predominant use because of its quadratic convergence properties [1]. An alternative algorithm to solve the power flow problem, based on complex analysis, is the Holomorphic Embedding Load-Flow Method (HELM) firstly proposed by Antonio Trias [2]. HELM is characterized by its deterministic and non-iterative nature, which may result in advantages over classical iterative approaches [2].

In order to mathematically formulate the power flow solution, the voltage magnitude and angle of a bus must be specified. This is achieved either by defining a single slack bus as in the classical single slack bus model approach, or by using Distributed Slack Bus (DSB) models. In the single slack bus model, system power losses are accounted for by a single bus, while in the DSB models, system losses are distributed among generators at different buses according to participation factors. Several approaches are available in the literature to select these participation factors based on different criteria. For instance, in [3], participation factors are calculated depending on each generator's active power output.

HELMpy is an open source software for power flow solution developed on Python 3 [4], and based on the NR method as well as the HELM. HELMpy also allows its users to model the power flow problem using either a single slack bus model or a DSB model. Several open source tools such as MATPOWER [5], PSAT [6], among others, and of special interest for being Python based: PyPSA [7], PYPOWER [8], pandapower [9], and GridCal [10] are also intended to solve the power flow problem. Nevertheless, none of the mentioned software has a properly working implementation of the HELM; in fact, it does not exist any fully functional open source software of the HELM other than HELMpy. This software is available as a repository on GitHub [11] and stored on Zenodo [12] under the GNU Affero General Public License v3.0 or later [13] (AGPL-3.0-or-later). HELMpy was developed to implement and validate the DSB model formulation devised in the research paper [14].

2. POWER FLOW SOLVERS

Power systems modeling and algorithms implemented in HELMpy for solving the power flow problem are explained below.

2.1. Modeling

The power flow solution algorithms included in HELMpy rely on the standard π branch model, which comprises models of AC transmission lines, phase-shifting transformers, off-nominal tap transformers, and shunt admittances for modeling capacitors or reactors banks in parallel to buses. Besides, reactive power limits of generators are considered for all generators, excluding the slack bus generator.

2.2. Algorithms

HELM

The HELM, firstly proposed by Antonio Trias [2], is a power flow solution method based on the study of complex analysis. This method is fully implemented in HELMpy based on the theory, fundamentals and practical recommendations described in [15], [16] and [17]. Furthermore, two models for voltage magnitude and active power controlled buses (PV buses) can be used as presented in [15].

HELM, seen solely as a numerical method, consists in recursively solving a linear matrix equation of the form $\mathbf{Ax} = \mathbf{B}$, where \mathbf{A} and \mathbf{B} are known values and \mathbf{x} is unknown. Each time this linear matrix equation is solved, an analytic continuation is applied on a group of power series that represent bus voltages. This process continues until either a correct solution is computed or no solution can be found [2]. The matrix \mathbf{A} remains constant in this approach; thus, it can be factorized. The vector \mathbf{B} is updated before the linear matrix equation is solved. As a result, the vector \mathbf{x} of power series coefficients is also updated.

Newton-Raphson

The Newton-Raphson based algorithm used in HELMpy to solve the power flow problem is implemented using a hybrid representation of the power balance equations in polar and rectangular form, where unknown variables are modeled in polar form.

Distributed Slack Bus Model

A DSB model based on the participation factors proposed in [3] is developed in HELMpy to be used along with the HELM and NR method. In the case of the NR algorithm, the DSB model is formulated according to [18] and adapted to balanced systems, while in the case of the HELM, the formulations devised in [14] are used.

2.3. Convergence criteria

The convergence criteria for the NR algorithm is based on either checking both active and reactive power

mismatch errors or active power mismatch errors only, depending on bus types. Convergence is achieved when all of these mismatch errors are less or equal than a predefined tolerance. Similarly, the convergence criteria for the HELM consists in checking whether the difference between magnitudes and phase angles of two consecutive voltages (obtained through analytic continuation of power series) at each bus is less or equal than a predefined tolerance [14].

3. IMPLEMENTATION AND ARCHITECTURE

HELMpy is written in Python given the extensive use of this programming language in scientific and engineering computing, more specifically in power system analysis

and modeling. This package is structured as a group of Python modules that contain sets of functions used to solve the power flow problem according to the previously mentioned algorithms. These sets of functions are built using the following well-known Python libraries: pandas [19] for handling the input data, storing it as DataFrame objects, and generating the required outputs; NumPy [20] for managing numerical functions, handling linear equation systems, and efficiently using array objects; and SciPy [21] for exploiting its sparse matrix handling capabilities.

Figures 1 and 2 depict the algorithms of the HELM and NR method, respectively. It can be seen from these figures the main structure of the code.

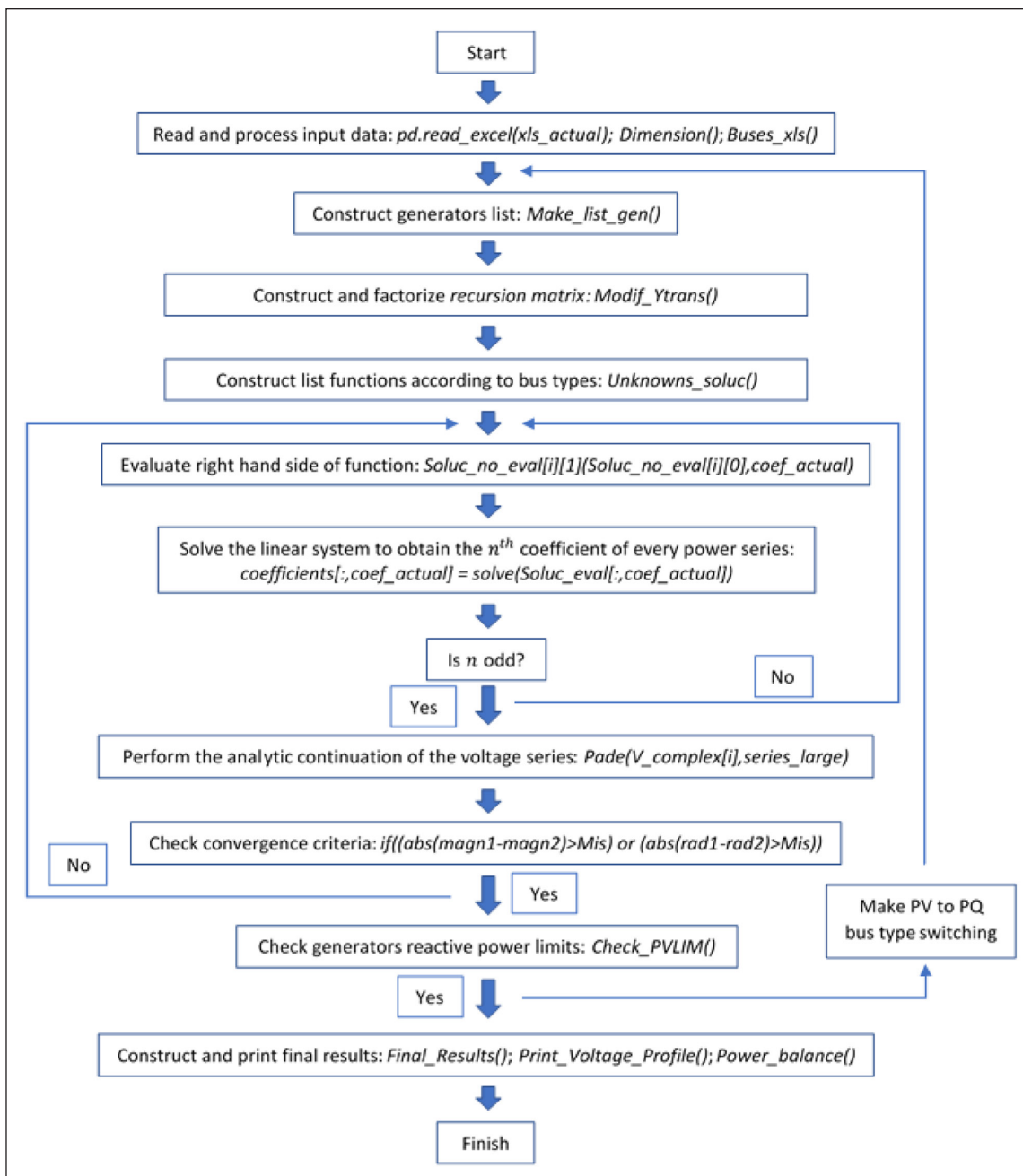


Figure 1 HELM algorithm flow chart.

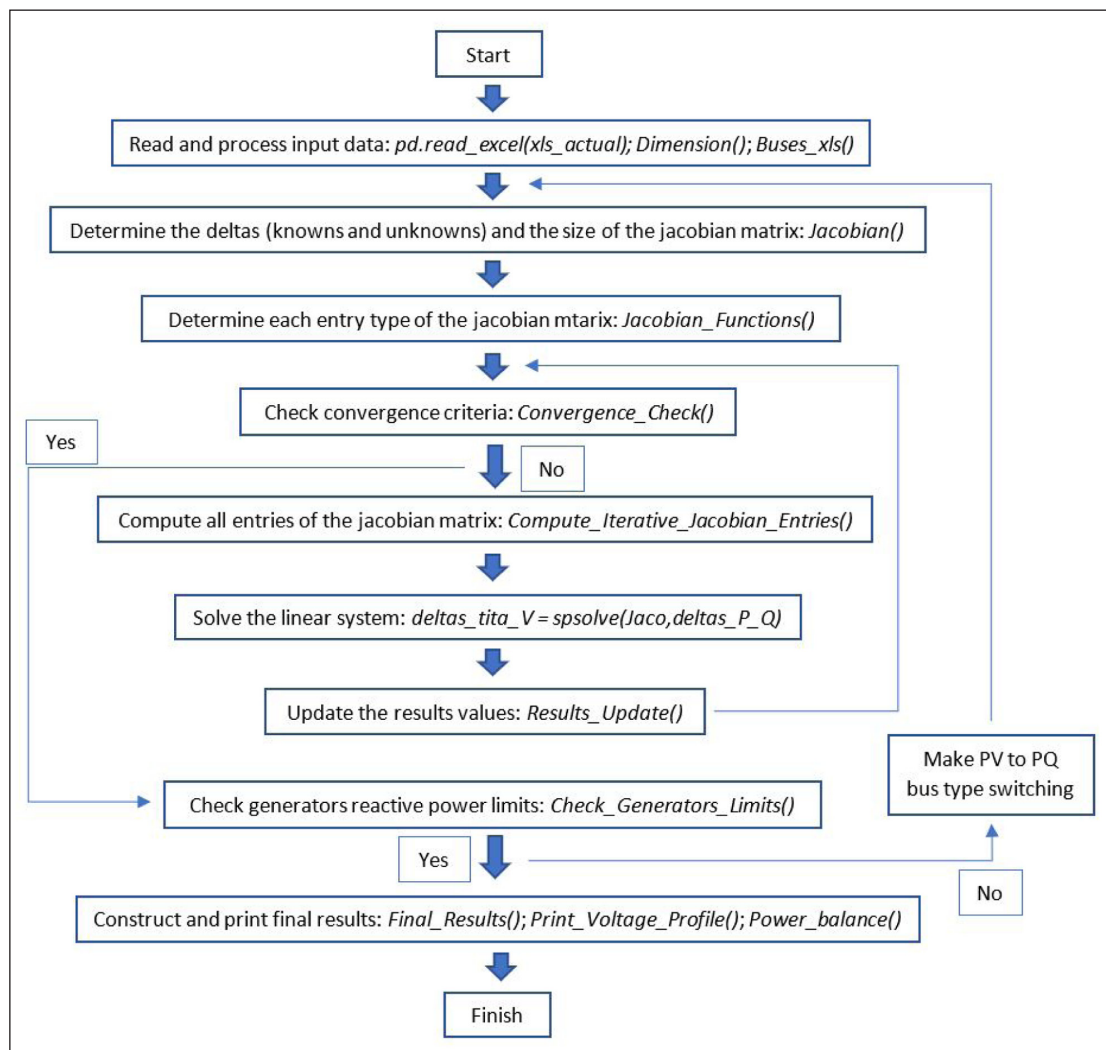


Figure 2 Newton-Raphson algorithm flow chart.

Sample data of large-sized practical grids (complex), as well as computed results obtained by using this sample data are included in the data directory of the repository. This data is used by the testing scripts of the repository.

4. QUALITY CONTROL

HELMpy contains a set of test cases which results were compared with those obtained using MATPOWER, yielding identical voltage profiles for the same tolerance. Regarding the DSB model, results obtained using the HELM and the NR approaches are alike as in all test cases. Users can perform bug reports in the GitHub repository [11].

(2) AVAILABILITY

OPERATING SYSTEM

Any operating system running Python.

PROGRAMMING LANGUAGE

Python version 3.6 or higher.

ADDITIONAL SYSTEM REQUIREMENTS

None.

DEPENDENCIES

pandas [19].
NumPy [20].
SciPy [21].

LIST OF CONTRIBUTORS

Tulio Molina, Universidad de Los Andes.
Juan Ortega, Universidad de Los Andes.

SOFTWARE LOCATION

Name: Zenodo.

Persistent identifier: <http://doi.org/10.5281/zenodo.3248868>

Licence: GNU Affero General Public License v3.0 or later.

Publisher: HELMpy team.

Version published: 1.

Date published: 18/06/19.

Code repository

Name: GitHub.

Persistent identifier: <https://github.com/HELMpy/HELMpy>

Licence: GNU Affero General Public License v3.0 or later.

Date published: 12/06/19.

LANGUAGE

English.

(3) REUSE POTENTIAL

Detailed documentation about the usage of HELMpy is included in the User's Guide available in the GitHub repository [11], where users can also contribute to the code by raising issues or making pull requests. This software is well structured and explained in order to make it easy to understand and modify. The HELM is a recently developed method and there is lack of properly working open source implementations of it; therefore, HELMpy emerges as a software of huge reuse potential for academic research in the subject of power system analysis and modeling, especially for novel models and approaches to be incorporated on the HELM.

ACKNOWLEDGEMENTS

The authors acknowledge the support given by Universidad de Los Andes.

FUNDING STATEMENT

Juan C. Muñoz acknowledges the financial support of Conicyt (through Conicyt/Fondap/15110019).

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Tulio Molina

Universidad de Los Andes, Facultad de Ingeniería, Escuela de Ingeniería Eléctrica, Mérida, Venezuela

Juan Ortega

Universidad de Los Andes, Facultad de Ingeniería, Escuela de Ingeniería Eléctrica, Mérida, Venezuela

Juan Muñoz orcid.org/0000-0001-6574-1203

Pontificia Universidad Católica de Chile, Escuela de Ingeniería, Chile

REFERENCES

- Glover JD, Sarma MS, Overbye T.** *Power System Analysis & Design, SI Version*. Cengage Learning; 2012.
- Trias A.** "The holomorphic embedding load flow method." In *2012 IEEE Power and Energy Society General Meeting*, July 2012; 1–8. DOI: <https://doi.org/10.1109/PESGM.2012.6344759>
- Meisel J.** "System incremental cost calculations using the participation factor load-flow formulation." *IEEE transactions on power systems*. 1993; 8(1): 357–363. DOI: <https://doi.org/10.1109/59.221220>
- Python programming language.** [Online]. Available: <https://www.python.org/>.
- Zimmerman RD, Murillo-Sanchez CE, Thomas RJ.** "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education." *IEEE Transactions on Power Systems*. Feb 2011; 26(1): 12–19. DOI: <https://doi.org/10.1109/TPWRS.2010.2051168>
- Milano F.** "An open source power system analysis toolbox." *IEEE Transactions on Power Systems*. Aug 2005; 20(3): 1199–1206. DOI: <https://doi.org/10.1109/TPWRS.2005.851911>
- Brown T, Hörsch J, Schlachtberger D.** "PyPSA: Python for Power System Analysis." *Journal of Open Research Software*. 2018; 6(4). [Online]. DOI: <https://doi.org/10.5334/jors.188>
- Lincoln R.** "PYPOWER." 2019. [Online]. Available: <https://github.com/rwl/PYPOWER>.
- Thurner L, Scheidler A, Schäfer F, Menke J, Dollichon J, Meier F, Meinecke S, Braun M.** "pandapower — an open-source python tool for convenient modeling, analysis, and optimization of electric power systems." *IEEE Transactions on Power Systems*. Nov 2018; 33(6): 6510–6521. DOI: <https://doi.org/10.1109/TPWRS.2018.2829021>
- Peñate S.** "GridCal." 2019. [Online]. Available: <https://github.com/SanPen/GridCal>.
- HELMpy, open source package of power flow solvers, including the Holomorphic Embedding Load Flow Method (HELM), developed on Python 3. [Online]. Available: <https://github.com/HELMpy/HELMpy>.
- Molina T, Ortega J.** "HELMpy, open source package of power flow solvers, including the Holomorphic Embedding Load Flow Method (HELM), developed on Python 3." Jun. 2019. [Online]. DOI: <https://doi.org/10.5281/zenodo.3248868>
- GNU Affero General Public License v3.0. [Online]. Available: <https://opensource.org/licenses/AGPL-3.0>.
- Ortega J, Molina T, Muñoz J, Oliva SH.** "Distributed slack bus model formulation for the holomorphic embedding load flow method." *International Transactions on Electrical Energy Systems*; 2019. [Online]. DOI: <https://doi.org/10.1002/2050-7038.12253>
- Subramanian MK.** "Application of holomorphic embedding to the power-flow problem." Master's thesis, Arizona State University; 2014.
- Rao S, Feng Y, Tylavsky DJ, Subramanian MK.** "The holomorphic embedding method applied to the power-flow problem." *IEEE Transactions on Power Systems*.

- 2016; 31(5): 3816–3828. DOI: <https://doi.org/10.1109/TPWRS.2015.2503423>
17. **Trias A.** “Fundamentals of the holomorphic embedding load-flow method.” 2015. [Online]. Available: <http://arxiv.org/abs/1509.02421>.
 18. **Tong S, Miu KN.** “A network-based distributed slack bus model for dgs in unbalanced power flow studies.” *IEEE Transactions on Power Systems*. 2005; 20(2): 835–842. DOI: <https://doi.org/10.1109/TPWRS.2005.846056>
 19. **Pandas, Python Data Analysis Library.** [Online]. Available: <https://pandas.pydata.org/>.
 20. NumPy, fundamental package for scientific computing with Python. [Online]. Available: <http://www.numpy.org/>.
 21. SciPy, Python-based ecosystem of open-source software. [Online]. Available: <https://www.scipy.org/>.

TO CITE THIS ARTICLE:

Molina T, Ortega J, Muñoz J 2021 HELMpy, Open Source Package of Power Flow Solvers, Including the Holomorphic Embedding Load Flow Method (HELM), Developed on Python 3. *Journal of Open Research Software*, 9: 23. DOI: <https://doi.org/10.5334/jors.310>

Submitted: 12 November 2019 Accepted: 05 August 2021 Published: 18 August 2021

COPYRIGHT:

© 2021 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press.