21th International Conference on Renewable Energies and Power Quality (ICREPQ'23)
Madrid (Spain), 24^{th} to 26^{th} May 2023
Renewable Energy and Power Quality Journal (RE&PQJ)
ISSN 2172-038 X, Volume No. 21, July 2023

Free Simulation Tools for Power Quality and Grid-Integration of Renewable Energies Cases

M. Manana*, A. Arroyo, R. Martinez, S. Bustamante, A. Laso, E. Sainz

Department of Electrical and Energy Engineering
Universidad de Cantabria, Spain
* E-mail: mananam@unican.es

Abstract. The analysis of Power Quality issues and/or the integration of renewable energy into the grid requires the use of different tools, among which the use of computer simulation applications stands out. The basic objective is to obtain a model of the electrical system under study that allows for knowledge, with a reasonable degree of precision, of its behavior under different operating conditions. The benefits of these tools are manifold, since they allow us to understand the response of the system, both in transient and steady-state conditions, to situations that are difficult to reproduce in practice: short circuits, failures in generation, transmission and distribution infrastructures, etc. Additionally, these tools can be used as platforms for operator training and network planning. Currently, a wide variety of commercial tools that are de facto standards can be found on the market. Some of these tools have a high cost and cannot be used by students outside the academic environment. In parallel, and thanks to the efforts of the scientific community, simulation tools have been developed that are available free of cost under several licensing models. This contribution analyzes some of the available tools, with special attention to those that are published under several open source and academic non-commercial software licenses that are available to the academic community free of cost.

Key words. Computer Simulation Tools, Open-source Software, Electromagnetic Transients Program, Power Flow Software

1. Introduction

Since the appearance of the first digital computers in the 1960s, the development of tools for simulating power electrical systems has experienced continuous evolution. In the initial stage, the scarcity of computational resources motivated engineers and scientists to focus their efforts on the development of simple models and efficient algorithms. Furthermore, the capabilities of the machines limited their use to offline simulation tasks, that is, to design and resource analysis tasks. In that sense, this initial effort has made it possible with the current capabilities of computers to use these tools for real-time control of power electrical systems.

This fact has also allowed them to be used in real-time network operation tasks, which represents a significant qualitative change. In this sense, the most common management applications are:

- Generation and demand planning.
- Analysis of the current state of the network and possible contingencies.
- Data acquisition systems, monitoring, and control (SCADA).

2. Theoretical background

The field of simulation of electric power systems is classical enough that there exists a significant set of bibliographic references. Below are provided some references that can help the reader to understand the fundamentals of simulating electric power systems, both in the time and frequency domain.

- Time domain. [1–5]
- Frequency domain. [4, 5]

3. Grid exchange formats

Although there is no single standardized format for exchanging information and defining networks in power system simulators, there are some formats that are recognized as de facto standards:

CIM-CGMES The Common Grid Model Exchange Specification (CGMES) is a set of IEC standard based on the IEC Common Information Model (CIM). This information model was designed by TSO data exchanges in application devoted to system development and operation.

UCTE-DEF The Union for the Co-ordination of Transmission of Electricity Data Exchange Format (UCTE-DEF) is an exchange format specified by the UCTE with the aim of facilitating the exchange of grid model among its members.

XIIDM The iTesla Internal Data Model (IIDM) is a format designed by the iTesla project. The aim of this format is not only the exchange of grid information but also the management of simulation results.

PSS/E This set of formats were defined by Siemens. One of them is the RAW format, that includes a collection of raw data for case definition in power flow analysis.

IEEE-CDF — This format was defined in the 1970s for the exchange of solved load flow cases[6].

4. Software Tools

The following is a compilation of some of the tools available as free software. This list is not intended to be exhaustive. Web links to download pages and source code were last revised as of March 13, 2023.

4.1 MATPOWER

Website: `https://matpower.org/`
License: BSD (3-clause)
Source Code: `https://github.com/MATPOWER/matpower`

MATPOWER [7] is a set of functions written in Matlab (M-files) distributed as a open-source under the BSD license beginning with version 5.1. The package allows solving steady-state power system simulation and optimization problems, such as:

- Power Flow (PF).
- Continuation Power Flow (CPF).
- Extensible Optimal Power Flow (OPF).
- Unit Commitment (UC).
- Stochastic, Secure Multi-interval OPF/UC.

4.2 PYPOWER

Website: `https://pypi.org/project/PYPOWER/`
License: BSD
Source Code: `https://github.com/rwl/PYPOWER`

PYPOWER [8] is a solver devoted to Power Flow and Optimal Power Flow. It is a translation of MATPOWER from Matlab to Python. Among others, the following features are included:

- DC and AC (Newtons method & Fast Decoupled) Power Flow.
- DC and AC Optimal Power Flow (OPF).

In recent years, it has not undergone significant evolution. Efforts have been devoted to updating libraries and dependencies.

4.3 PSAT

Website: `http://faraday1.ucd.ie/psat.html`
License: GPL
Source Code: `http://faraday1.ucd.ie/psat.html`

The Power System Analysis Toolbox (PSAT) is a toolbox for electric power system analysis and simulation written in Matlab by Professor Federico Milano [9]. The current version 2.1.11 can be run in both Matlab and GNU Octave.

PSAT is a powerful tool that is available by means of graphical user interfaces (GUI). The toolbox also provides a Simulink-based library for network design.

PSAT includes many features:

- Power Flow.
- Continuation Power Flow.
- Optimal Power Flow.
- Small Signal Stability Analysis.
- Time Domain Simulation.
- Complete Graphical User Interface.
- User Defined Models.
- FACTS Models.
- Wind Turbine Models.

4.4 OpenDSS

Website: `https://www.epri.com/pages/sa/opendss`
License: BSD
Source Code: `https://sourceforge.net/p/electricdss/code/HEAD/tree/`

The Open Distribution System Simulator (OpenDSS) is a simulation tool for electric utility distribution systems developed by Electrotek Concepts [10]. This tools has been originally written in Object Pascal using Delphi. Several parts have been written in C and C++. It can be used for:

- Distribution Planning and Analysis.
- General Multi-phase AC Circuit Analysis.
- Analysis of Distributed Generation Interconnections.
- Annual Load and Generation Simulations.
- Risk-based Distribution Planning Studies.
- Probabilistic Planning Studies.
- Solar PV System and Wind Plant Simulations.
- Nuclear Plant Station Auxiliary Transformer Modeling.
- Distribution Automation Control Assessment and Protection System Simulation.
- Storage Modeling.
- Distribution Feeder Simulation with AMI Data and State Estimation.
- Ground Voltage Rise on Transmission Systems.
- Geomagnetically-Induced Currents (GIC).
- Co-simulation of Power and Communications Networks.
- Analysis of Unusual Transformer Configurations.
- Harmonic and Interharmonic Distortion Analysis.
- Neutral-to-earth Voltage Simulations.

The software can be used by means of a stand-alone executable windows program and as an in-process COM server DLL that can be call from several software platforms as Matlab, Python, C# and R. In addition, OpenDSS also has a proprietary script language.

4.5 PyPSA

Website: https://pypsa.org/
License: MIT
Source Code: https://github.com/PyPSA/

The tool Python for Power System Analysis (PyPSA) is and open source toolbox for the simulation and optimization of both power and energy systems [11]. PyPSA is developed and maintained by the Department of Digital Transformation in Energy Systems at the Technical University of Berlin. PyPSA includes many features:

- Power Flow.
- Linear and Security-constrained Linear Optimal Power Flow.
- Total electricity/energy system least-cost investment optimization.

4.6 GridCal

Website: https://github.com/SanPen/GridCal
License: LGPL
Source Code: https://github.com/SanPen/GridCal

GridCal is a tool developed by Santiago Peñate Vera [12]. GridCal has been written in Python with a Qt GUI. It also has a console Python console that allows some degree of automation. It is licensed under the LGPL license and includes many features:

- Power Flow.
- Optimal Power Flow and Generation Dispatch.
- MonteCarlo stochastic Power Flow.
- Three-phase and unbalanced short circuit.
- Blackout cascading in simulation and step by step mode.

4.7 GridLAB-D

Website: https://www.gridlabd.org/
License: Proprietary license (BSD-style)
Source Code: https://github.com/gridlab-d/gridlab-d

GridLAB-D is a distribution level power system simulator designed to allow users to create and analyze smart grid technologies [13, 14]. It is been developed by Pacific Northwest National Laboratory (PNNL) in collaboration with industry and academia through funding from the U.S. Department of Energy Office of Electricity Delivery and Energy Reliability (DOE/OE). GridLAB-D includes many features:

- Distribution automation Design/Evaluation.
- Peak Load Management.
- Distributed Generation and Storage.
- Rate Structure Analysis.

4.8 PandaPower

Website: https://www.pandapower.org/
License: BSD (3-clauses)
Source Code: https://github.com/e2nIEE/pandapower

PandaPower is an open source tool for power system modeling, analysis and optimization written in Python as a package [15]. The is published under BSD (3-clauses). PandaPower includes many features:

- Power Flow.
- Optimal Power Flow.
- State Estimation.
- Short-Circuit Calculation.

4.9 ATP-EMTP

Website: https://atp-emtp.org/
License: Proprietary license
Source Code: Not available for general users

EMTP is a transient electromagnetic simulation program focused on power systems. Theoretical development is supported by pioneering work such as that of H. Dommel [1, 16]. Both ATPDraw (GUI pre-processor) and ATPSolver can be used for both research and teaching free of cost. The only requirement is to be registered at the EEUG organization. In summary, the electrical system is represented by its admittance matrix and solved using the trapezoidal rule. The combined use of the method of characteristics and the trapezoidal rule allows solving models that integrate both concentrated and distributed parameters. Later on, other scientists such as W. Scott Meyer joined the initial development by Professor Dommel. Subsequently, this collective effort involved, among others, A. Ametani, V. Brandwajn, L. Dubé, J. R. Marti, and A. Semlyen. Nowadays, it is possible to access this tool for free for academic use thanks to the EEUG group led by Professor Hans Kr. Høidalen from NTNU in Norway. From this original idea of trapezoidal integration, other commercial variants have been developed which are not included here as they require the payment of licenses:

- EMTDC/PSCAD. Commercial equivalent to ATP/EMTP that is currently developed and mantained by Manitoba HVDC Research Center (https://mycentre.hvdc.ca/).
- DIgSILENT. Includes a commercial equivalent full EMT-type simulation module (https://www.digsilent.de/en/).
- Simscape Electrical (known in the past as SimPowerSystems). Commercial tool developed by MathWorks that is based on a state-space approach for solving models that combine electronic, mechatronic,

and electrical systems. Its graphical interface is the well-known Simulink software included in MathWorks' Matlab (`https://es.mathworks.com/products/simscape-electrical.html`).

Fig. 1 shows the ATPDraw [17], that is an ATP GUI Preprocessor. In this example a controlled rectifier is simulted.

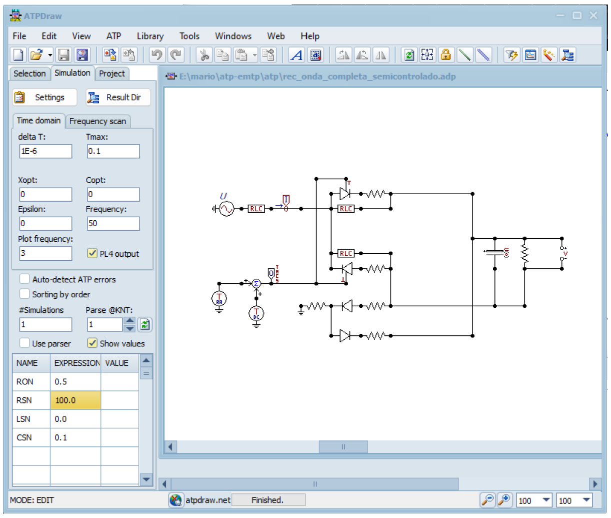

Fig. 1: Analysis of harmonic distortion of a single phase controlled rectifier in ATPDraw .

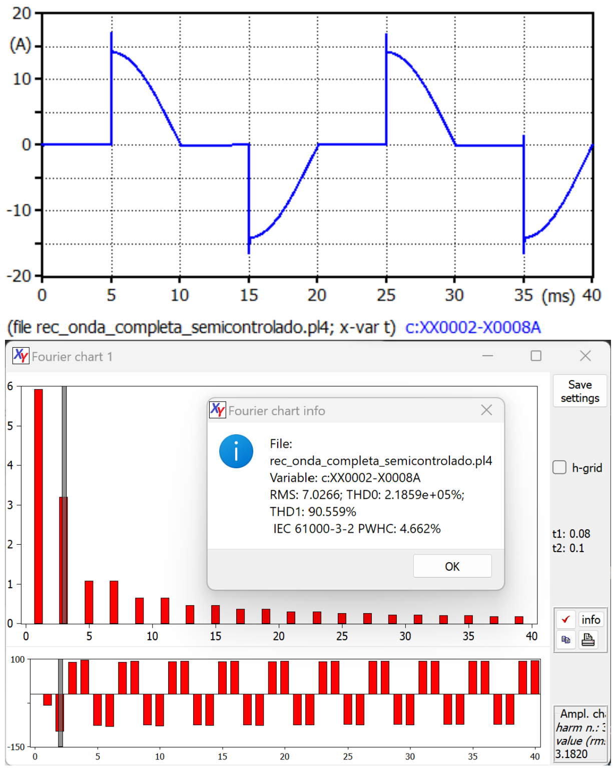

Fig. 2: Time snapshot and harmonic components of a current demanded by a single phase controlled rectifier .

The results obtained with ATP can be analyzed with programs like XYPlot, a plotting and post-processing tool developed by Professor M. Ceraolo [18]. XYPlot can be used for plotting the current waveform and its harmonic components. Fig. 2 shows the results for the model shown in Fig. 1.

5. Application example

This section introduces an example in which the PandaPower library is applied to the analysis of the voltage profile of a low voltage distribution network that includes distributed generation (see Fig. 3).

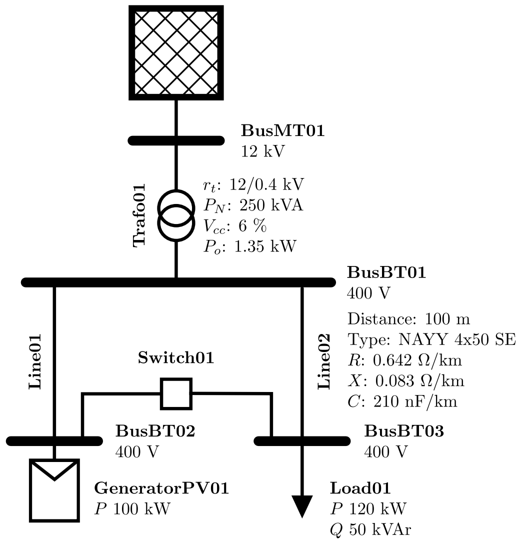

Fig. 3: Distribution network with distributed generation .

The basic Python script that defines the case is

```
import pandapower as pp

#create empty net
net = pp.create_empty_network()

#create buses
BusMT01 = pp.create_bus(net, vn_kv=12., name="BusMT01")
BusBT01 = pp.create_bus(net, vn_kv=0.4, name="BusBT01")
BusBT02 = pp.create_bus(net, vn_kv=0.4, name="BusBT02")
BusBT03 = pp.create_bus(net, vn_kv=0.4, name="BusBT03")

#create switches
from_bus = pp.get_element_index(net, "bus", "BusBT02")
to_bus = pp.get_element_index(net, "bus", "BusBT03")
Switch01 = pp.create_switch(net, from_bus, to_bus, et="b",
           closed=False, type="DS", name="Switch01")


#create bus elements
red1 = pp.create_ext_grid(net, bus=BusMT01, vm_pu=1.00,
           name="Grid Connection")
Carga01 = pp.create_load(net, bus=BusBT03, p_mw=0.120, q_mvar=0.05,
           name="Carga01")
GeneradorFV01 = pp.create_sgen(net, bus=BusBT02, p_mw=0.1, q_mvar=0,
           sn_mva=0.12, type='PV', name='GeneradorFV01')

#create branch elements
#Trafo01 = pp.create_transformer(net, hv_bus=BusMT01, lv_bus=BusBT01,
           std_type="0.4 MVA 20/0.4 kV", name="Trafo01")
Trafo01 = pp.create_transformer_from_parameters(net, BusMT01, BusBT01,
           sn_mva=.25, vn_hv_kv=12, vn_lv_kv=0.4, vkr_percent=1.425,
           vk_percent=6, pfe_kw=1.35, i0_percent=0.2375, tap_side="hv",
           tap_neutral=0, tap_min=-2, tap_max=2, tap_step_percent=2.5,
           tp_pos=0, shift_degree=150, name='Trafo01')

Linea01 = pp.create_line(net, from_bus=BusBT01, to_bus=BusBT02,
           length_km=0.1, std_type="NAYY 4x50 SE", name="Linea01")
Linea02 = pp.create_line(net, from_bus=BusBT01, to_bus=BusBT03,
           length_km=0.1, std_type="NAYY 4x50 SE", name="Linea02")

pp.runpp(net)

net.res_bus
```

Table 1 summarizes the voltage profile and power flow results considering that the Switch01 is open. Table 2 provides the same results considering that the Switch01 is closed.

Table 1 Voltage profile with switch 01 opened.

Bus	V_m [pu]	V_a [degree]	P [MW]	Q [Mvar]
BusMT01	1.0000	0.0000	-0.0331	-0.0523
BusBT01	0.9859	-0.2669	0.0000	0.0000
BusBT02	1.0250	0.0271	-0.1000	-0.0000
BusBT03	0.9313	0.5964	0.1200	0.0500

Table 2 Voltage profile with switch 01 closed.

Bus	V_m [pu]	V_a [degree]	P [MW]	Q [Mvar]
BusMT01	1.0000	0.0000	-0.0221	-0.0507
BusBT01	0.9869	-0.1221	0.0000	0.0000
BusBT02	0.9814	0.4405	-0.1000	-0.0000
BusBT03	0.9814	0.4405	0.1200	0.0500

Fig. 4 and 5 show the network topology and bus voltages (in pu) and line loading (in percentage) respectively.

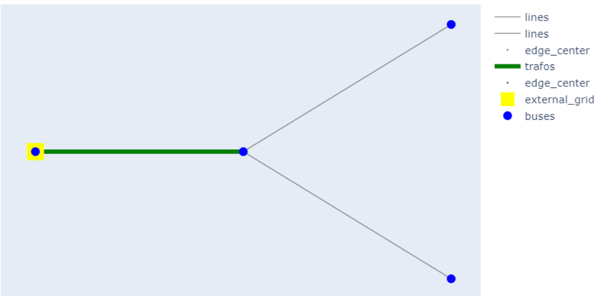

Fig. 4: Network graph.

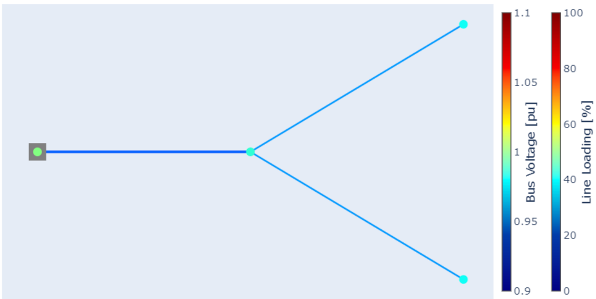

Fig. 5: Bus voltages [pu] and line loading [%] .

6. Conclusion

The simulation of electrical power systems is a mature discipline that has a large number of free simulation tools.

Many of these tools are distributed under open software licenses as GPL, LGPL, BSD and MIT, so that it is possible for researchers to access the source code to study and modify it to suit their specific needs. In this article, several tools have been reviewed, comparing their characteristics, license type and source code availability. Table 3 summarizes this comparison.

7. Acknowledgments

This research was funded by the European Commission under the Horizon 2020 Program: Project FLEXIGRID with reference No 864579, H2020-LC-SC3-2019-ES-SCC.

8. References

1 H. W. Dommel, "Digital computer solution of electromagnetic transients in single-and multiphase networks," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-88, no. 4, pp. 388–399, 1969.

2 W. S. Meyer and H. W. Dommel, "Numerical modelling of frequency-dependent transmission-line parameters in an electromagnetic transients program," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-93, no. 5, pp. 1401–1409, 1974.

3 J. A. Martinez-Velasco, ed., *Transient Analysis of Power Systems*. Wiley IEEE Press, 2020.

4 J. Arrillaga and C. Arnold, *Computer Analysis of Power Systems*. John Wiley & Sons, 1990.

5 J. Arrillaga and N. Watson, *Computer Modelling of Electrical Power Systems*. John Wiley & Sons, Ltd, 2001.

6 W. Group, "Common format for exchange of solved load flow data," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-92, no. 6, pp. 1916–1925, 1973.

7 R. D. Zimmerman and C. E. Murillo-Sánchez, "Matpower," Oct 2020.

8 R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

9 F. Milano, L. Vanfretti, and J. C. Morataya, "An open source power system virtual laboratory: The psat case and experience," *IEEE Transactions on Education*, vol. 51, no. 1, pp. 17–23, 2008.

10 R. C. Dugan and T. E. McDermott, "An open source platform for collaborating on smart grid research," in *2011 IEEE Power and Energy Society General Meeting*, pp. 1–7, 2011.

11 T. Brown, J. Hörsch, and D. Schlachtberger, "PyPSA: Python for Power System Analysis," *Journal of Open Research Software*, vol. 6, no. 4, 2018.

12 S. Peñate Vera, "Practical grid modelling," 2021.

13 D. Chassin, J. Fuller, and N. Djilali, "Gridlab-d: An agent-based simulation framework for smart grids," *Journal of Applied Mathematics*, vol. 2014, 2014.

14 D. Chassin, K. Schneider, and C. Gerkensmeyer, "Gridlab-d: An open-source power systems modeling and simulation environment," 2008.

15 L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "pandapower an open-source python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Transactions on Power Systems*, vol. 33, pp. 6510–6521, Nov 2018.

16 H. W. Dommel, "Electromagnetic transients program," *Electromagnetic Transients Program Reference Manual*, 1986.

17 H. K. Høidalen, B. A. Mork, F. Gonzalez, D. Ischchenko, and N. Chiesa, "Implementation and verification of the hybrid transformer model in atpdraw," *Electric Power Systems Research*, vol. 79, no. 3, p. 454 459, 2009. Cited by: 27.

18 M. Ceraolo, "Mc's plotxy. a general-purpose plotting and post-processing open-source tool," *SoftwareX*, vol. 9, p. 282 287, 2019.

Table 3 Comparison of Free Power System simulation tools (adapted from [15]).

	MATPOWER	PYPOWER◇	PSAT	OpenDSS	PyPSA	GridCal	GridLAB-D	pandapower	ATP/EMTP
Graphical User Interface			✓△	✓		✓			✓
Time-domain			✓						✓
Frequency domain	✓	✓	✓	✓	✓	✓	✓	✓	
Loads (constant Z, constant I, constant P)			✓	✓		✓	✓	✓	✓
Lines	✓	✓	✓	✓	✓	✓	✓	✓	✓
2-Winding Transformer (π model)	✓	✓	✓	✓	✓	✓	✓	✓	✓
2-Winding Transformer (T model)			✓	✓	✓		✓	✓	✓
3-Winding Transformer			✓	✓			✓	✓	✓
DC lines	✓		✓	✓	✓		✓	✓	✓
Ideal Switches			✓					✓	✓
Votage-controlled sources	✓	✓	✓	✓	✓	✓	✓	✓	✓
Static Load / Generation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Shunt impedance	✓	✓	✓	✓	✓	✓	✓	✓	✓
Asymmetrical Impedance			✓					✓	✓
Ward equivalent (Combination of impedance and PQ loads)			✓					✓	✓
Storage Units			✓	✓	✓	✓	✓		
Source code: Matlab	✓		✓				✓		
Source code: Python		✓			✓	✓	✓	✓	
Source code: Delphi				✓					
Source code: C and C++				✓		✓			
Source code: FORTRAN									✓
BSD License	✓	✓					✓	✓	
GPL			✓						
LGPL						✓			
MIT					✓				
Proprietary license				✓		✓			✓
SCOPUS references	471	11	2,132	639	33	-	179	42	2,110
IEEEXplore references	310	8	627	457	9	-	134	25	784

(◇) Python version of MATPOWER.

(△) GUI based on MATLAB & Simulink.