A Graphical and Open Source Matlab-GAMS interface for Electricity Market Models

Federico Milano

Departamento de Ingeniería Eléctrica, Electrónica y Automática
E.T.S.I.I., Universidad de Castilla-La Mancha
Campus de Ciudad Real, 13071 Ciudad Real (España)
Tel: +34 926 295 219, fax: 926 295 361, e-mail: **Federico.Milano@uclm.es**

Abstract. This paper describes the PSAT-GAMS interface to solve a variety of optimization models. The Power System Analysis Toolbox (PSAT) is an open source Matlab-based software package for analysis and design of small to medium size electric power systems. While the General Algebraic Modeling System (GAMS) is a high-level modeling system for mathematical programing problems. It consists of a language compiler and a variety of integrated high-performance solvers. A bridge between GAMS and Matlab allows using sophisticated nonlinear optimization tools with the visualization tools provided by Matlab. Examples based on a didactic 3-bus test system and the RTS-96 24-bus system are presented to illustrate the capabilities of the interface.

Keywords

Electricity markets, optimal power flow, linear programming, nonlinear programming, Matlab, GAMS.

1. Introduction

In recent years, the increasing interest for competitive and secure electricity markets and their mathematical exploitations have led to the proposal of a huge variety of market clearing models. This results in a variety of models and mathematical techniques proposed in the literature to take into account diverse market features and behaviors. Recent relevant contributions on this topic are [1], [2], [3], [4], [5], and [6].

In order to develop and fast prototype new market clearing models there is the need of a versatile software tool which is able to combine powerful optimization solvers and a power system dedicated software. This paper describes a novel open-source software tool, namely the PSAT-GAMS interface, for solving market clearing problems as well as stability constrained optimal power flows. The proposed program is suited for the analysis of small to realistic size electric power systems and is ori-

ented to (under)graduate students and researchers in the field of electricity markets and power system analysis.

PSAT is a Matlab toolbox for electric power system analysis and control [7]. PSAT includes power flow, continuation power flow, optimal power flow, small signal stability analysis and time domain simulation. All operations can be assessed by means of graphical user interfaces (GUIs) and a Simulink-based library provides an user-friendly tool for network design. One of the key point of PSAT is that it is completely open source and free. This allows exchanging ideas and effectively improving scientific research [8]. At this aim, the command line version of PSAT can run on GNU/Octave [9], which is a free Matlab clone.

In the field of general purpose optimization techniques, one of researchers' favorite choice is the General Algebraic Modeling System (GAMS) [10]. GAMS is a high-level modeling system for mathematical programming problems. It consists of a language compiler and a variety of integrated high-performance solvers. GAMS is specifically designed for large and complex scale problems, and allows creating and maintaining models for a wide variety of applications and disciplines [10]. GAMS is able to formulate models in many different types of problem classes, such as linear programming (LP), nonlinear programming (NLP), mixed-integer linear programming (MILP) and (relaxed) mixed-integer nonlinear programming (MINLP).

The paper is organized as follows. Firstly the main features of PSAT are illustrated in Section 2. Section 3 gives the outlines of the PSAT-GAMS interface while Section 4 gives a brief insight on the mathematical models currently implemented in the PSAT-GAMS interface. Section 5 depicts a variety of examples based on a didactic 3-bus test system and the RTS-96 test system [11]. Finally Section 6 summarizes the main contributions of this paper.

2. PSAT Outlines

PSAT is a well documented Matlab-based software for power system analysis and is aimed to researchers and power system (under)graduated students [7], [12]. The program is provided with a complete set of power system models and algorithms for static and dynamic analysis. Furthermore PSAT is modular and open-source, so that users can easily develop new models and algorithms and extend PSAT capabilities.

The synoptic scheme of PSAT is depicted in Fig. 1. Observe that the PSAT kernel is the power flow algorithm, which also takes care of the initialization of dynamic models. The user has to solve the power flow first, then one can perform further static and/or dynamic analysis. These are:

1) Continuation Power Flow (CPF);

2) Optimal Power Flow (OPF);

3) Small signal stability analysis;
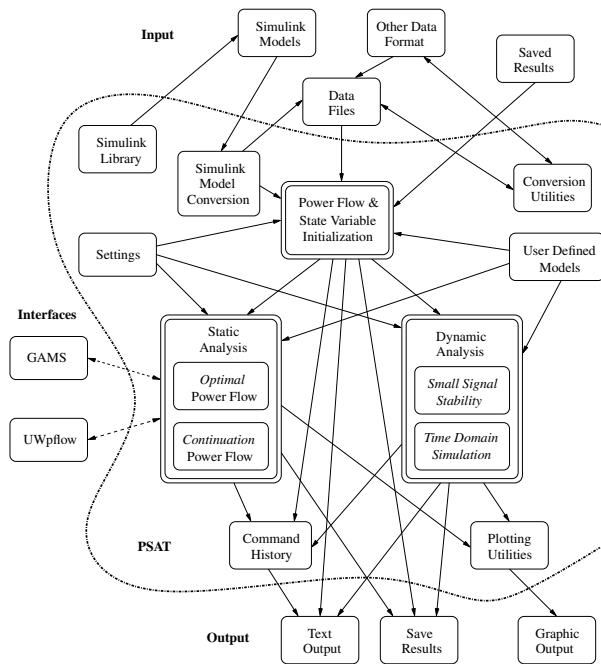
4) Time domain simulations.

Fig. 1. Synoptic scheme of PSAT.

An important feature of PSAT is the fully exploitation of Matlab vectorized computations and sparse matrix functions. This allows optimizing performances. Furthermore PSAT is provided with the most complete set of algorithms for static and dynamic analyses among currently available Matlab-based power system softwares [7].

In order to perform accurate and complete power system analyses, PSAT supports a variety of static and dynamic models, including non conventional loads, synchronous machines and controls, regulating transformers, FACTS, wind turbines, and fuel cells [12]. Besides mathematical algorithms and models, PSAT includes a variety of additional tools, such as data file conversion to and from other formats; user defined model editor and installer; and command line usage. Observe that command line usage allows running PSAT on GNU/Octave.

PSAT has been thought as an educational tool. At this aim, PSAT is provided with a complete graphical user interface (GUI). For example Fig. 2 illustrates the main PSAT window, from which the user can launch all algorithms and procedures. Furthermore, PSAT allows drawing one-line electrical diagrams by means of pictorial blocks. Fig. 3 depicts the complete PSAT-Simulink library (see also Figs. 6 and 12 which illustrate the 3-bus and the RTS-96 24-bus test systems, respectively).

Fig. 2. Main graphical user interface of PSAT.

The PSAT computational engine is purely Matlab-based and the Simulink environment is used only as graphical tool. As a matter of fact, Simulink models are read by PSAT to exploit network topology and extract component data. A byproduct of this approach is that PSAT can run on GNU/Octave, which is currently not providing a Simulink clone.

Observe that some Simulink-based tools, such as PAT [13] and EST [14], use Simulink to simplify the design of new control schemes. This is not possible in PSAT. However, PAT and EST do not allow representing the network topology, thus resulting in a lower readability of the whole system.
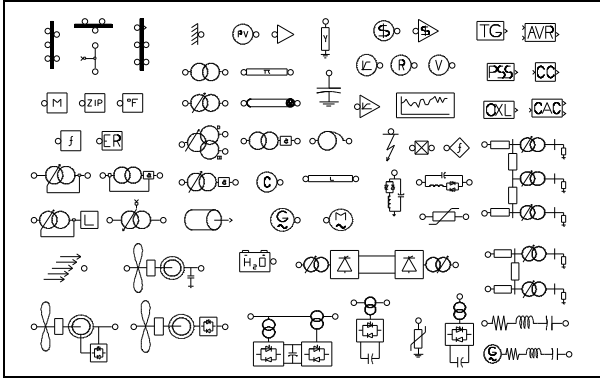
Fig. 3. PSAT Simulink library.

3. PSAT-GAMS Interface

Figure 4 depicts the scheme of the PSAT-GAMS interface. The resulting software is a rather innovative tool able to set up large scale power system test cases, solve complex OPF problems and finally visualize results by means of a user-friendly GUI which is depicted in Fig. 5. The GUI allows selecting the market clearing model, setting model parameters (please see Section 4 for details on market clearing models), and displaying GAMS results.

An existing Matlab-GAMS Interface [15] has been used as the main reference for creating the PSAT-GAMS Interface (PGI). However, the PGI has been completely rewritten and optimized for the use within the PSAT environment. Main novelties of the PGI with respect to the interface presented in [15] are as follows:
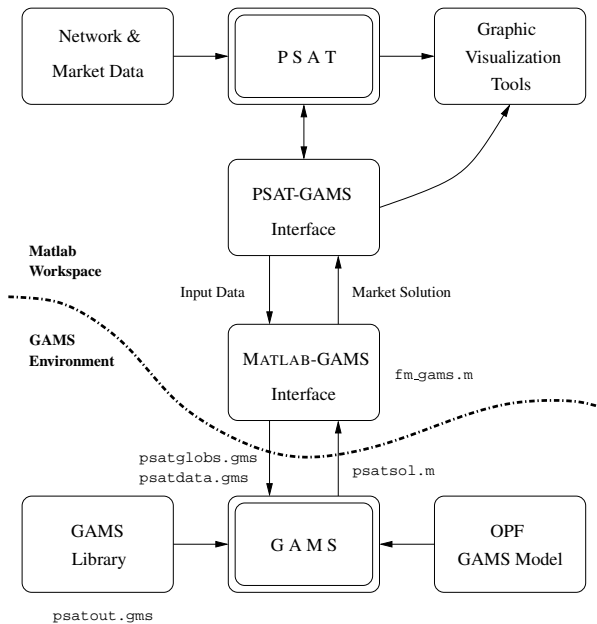

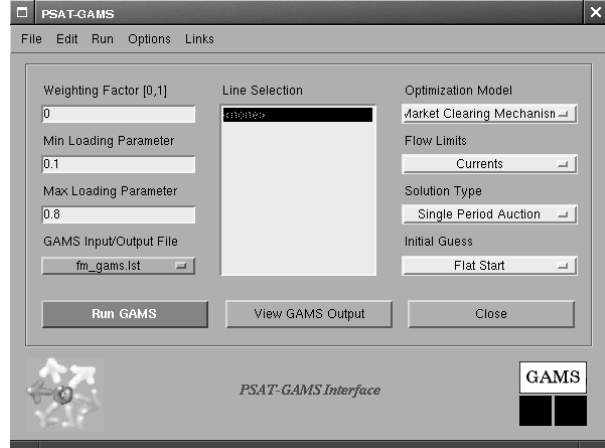Fig. 4. Structure of the PSAT-GAMS interface.

Fig. 5. GUI of the PSAT-GAMS interface.

1) PGI is platform independent. This feature improve portability and in turn the development of the interface.

2) PGI comes with a complete GUI.

3) PGI does not require the user to know anything about Matlab and/or GAMS programming language. This is useful for educational purposes.

Before the interface can be used, one has to create data file describing the system. At this aim the user can use the PSAT-Simulink interface and draw the on-line diagram of the network or load a predefined test network which is provided within the PSAT main distribution. The second step is solving the power flow. At this point, the PSAT-GAMS interface can be opened and the GAMS solvers launched. Observe that only networks for which market data have been defined can be used with the PSAT-GAMS interface. If no market data has been defined, the interface simply terminates with a warning message.

The interface works as follows: the system information are "translated" into a GAMS file (`psatdata.gms`). User settings (e.g. the market clearing model) and global variables (e.g. the number of bus) are also written to a GAMS script file (`psatglobs.gms`). The advantage of writing data files is that one could use PSAT to export the GAMS data, and then use GAMS from scripting without the use of Matlab. This feature can be useful in case of "heavy" applications, where all computer resource are needed.

Once the data files have been written, GAMS is launched from within Matlab and the market clearing procedure solved. The routines (`fm_gams.gms`) for solving the

market clearing problems have been designed to be as general as possible, with no limit in the network size or in the number of market participants. Limits are only the computer memory and GAMS solvers' capabilities.

Finally, when the GAMS solver has terminated, the GAMS output (psatsol.m) is passed back to Matlab, so that GAMS results can be displayed using the graphical capabilities of Matlab or used for further analyses using PSAT routines.

4. Market Clearing Models

The current version of the PGI is provided with five optimization problems, namely simple auction, simplified market clearing mechanism, standard OPF, voltage stability constrained (VSC)-OPF and maximization of the maximum loading condition. A brief description of these models is presented in the following subsections.

A. General single-period optimization model

The general optimization problem which is implemented in the PSAT-GAMS interface can be represented as a nonlinear programming model, as follows:

$$
\begin{aligned}
\text{Maximize}_{(x,p,\hat{x},\lambda)} \quad & f(p,\lambda) && (1)\\
\text{subject to} \quad & g(x,p) = 0\\
& \hat{g}(\hat{x},p,\lambda) = 0\\
& \lambda_{\min} \leq \lambda\\
& \lambda \leq \lambda_{\max}\\
& h_{\min} \leq h(x,p)\\
& h(x,p) \leq h_{\max}\\
& \hat{h}_{\min} \leq h(\hat{x},p)\\
& h(\hat{x},p) \leq \hat{h}_{\max}\\
& p_{\min} \leq p\\
& p \leq p_{\max}
\end{aligned}
$$

where $x, \hat{x} \in \mathbb{R}^n$ are the dependent variables, such as bus voltage phasors, $p \in \mathbb{R}^m$ are the control variables, i.e. power demand and supply bids P_D and P_S, respectively, and $\lambda \in \mathbb{R}$ is loading parameter. The functions $f : \mathbb{R}^m \mapsto \mathbb{R}$, $g, \hat{g} : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$, and $h, \hat{h} : \mathbb{R}^n \mapsto \mathbb{R}^\ell$ are defined as follows:

A.1. Objective function

For pure market clearing problems, the objective function f is defined as follows:

$$
f = \sum_i C_{D_i}(P_{D_i}) - \sum_i C_{S_i}(P_{S_i}) \qquad (2)
$$

Equation (2) represents consumer surplus plus the producer surplus (i.e. the net social welfare). It is computed

as the difference of two terms. The first term is the sum of accepted demand bids P_{D_i} times their corresponding bid prices C_{D_i} in \$/MWh. (In the case of inelastic demand, demand powers P_D are known, which can be represented in (2) by setting $C_D = 0$.) The second term is the sum of accepted production bids P_{S_i} times their corresponding bid prices C_{S_i} in \$/MWh. Observe that (2) is linear for simple auction and market clearing procedure, while is assumed to be a quadratic expression for OPF-based problems.

For the voltage stability constrained problem, the objective function includes the maximization of the loading parameter λ, as it was proposed in [6]:

$$
f = \omega\Big(\sum_i (C_{D_i}(P_{D_i}) - \sum_i C_{S_i}(P_{S_i})\Big) + (1-\omega)\lambda \quad (3)
$$

In (3), two terms are present, with their influence on the final solution being determined by the value of the weighting factor ω ($0 < \omega < 1$). The first term represents the social welfare, whereas the second term guarantees that the "distance" between the market solution and the critical point is maximized [6]. Observe that $\omega > 0$, since for $\omega = 0$ there would be no representation of the market in the proposed OPF formulation, rendering it useless. Furthermore, $\omega < 1$, otherwise λ will not necessarily correspond to a maximum loading condition of the system.

Finally, for the maximum loading condition problem, one has:

$$
f = \lambda \qquad (4)
$$

which leads to a pure security problem with no market involved [16]. This lead to get the maximum loading condition of the system. It is important to stress the fact that the maximum value for λ can be associated with thermal or bus voltage limits, or a voltage stability limit due to a singularity of the power flow Jacobian (saddle-node bifurcation) or a controller limit such as a generator reactive power limit (limit-induced bifurcation).

A.2. Equality constraints

The equations g represents the standard power flow equations:

$$
g(x,p) = g(\theta, V, k_G, P_S, P_D) = 0 \qquad (5)
$$

where $x = (\theta, V, k_G)$ and $p = (P_S, P_D)$. The variables θ and V are the bus voltage phases and magnitudes, respectively, while k_G is a scalar variable used to represent system losses through a unique or distributed slack bus. Observe that x represents the set of dependent variables to be optimized, i.e. $x \in \mathbb{R}^n$. Generator reactive powers

Q_G are not included in this set and are expressed as a function of x and p.

The generator and load powers are defined as follows:

$$
\begin{aligned}
P_G &= P_{G_0} + P_S \\
P_L &= P_{L_0} + P_D
\end{aligned}
\tag{6}
$$

where P_{G_0} and P_{L_0} stand for generator and load powers which are not part of the market trading (e.g. must-run generators, inelastic loads). Loads are assumed to have constant power factor, thus:

$$
Q_L = P_L \tan(\phi_L)
\tag{7}
$$

For the simple auction problem, g reduce to a lossless power balance, as follows:

$$
\sum_i (P_{G_{0i}} + P_{S_i}) - \sum_i (P_{L_{0i}} + P_{D_i}) = 0
\tag{8}
$$

while in the simplified market clearing procedure losses are taken into account through a simplified quadratic expression:

$$
\begin{aligned}
\sum_i (P_{G_{0i}} + P_{S_i}) & \tag{9} \\
-\sum_i (P_{L_{0i}} + P_{D_i}) - \theta^T G \theta &= 0 \\
P_S + P_{G_0} - P_D - P_{L0} - B\theta &= 0
\end{aligned}
$$

where G and B are the network susceptance and conductance matrices, respectively.

A.3. Inequality constraints

In (1), the set of inequality constraints has been split into h, which represent the physical and security limits of the system, and the bid blocks for control variables p.

The physical and security limits considered in this paper are similar to what is used in [17], and take into account transmission line thermal limits:

$$
\begin{aligned}
I_{ij}(\theta, V) &\leq I_{ij_{\max}} \\
I_{ji}(\theta, V) &\leq I_{ij_{\max}}
\end{aligned}
\tag{10}
$$

generator reactive power limits:

$$
Q_{G_{\min}} \leq Q_G(x,p) \leq Q_{G_{\max}}
\tag{11}
$$

voltage "security" limits:

$$
V_{\min} \leq V \leq V_{\max}
\tag{12}
$$

and power limits on transmission lines:

$$
|P_{ij}| \leq P_{ij\,\max}
\tag{13}
$$

which are used to represent stability boundaries of the system [17]. Observe that (13) are typically "conservative" and can in general lead to low transaction levels, higher costs and lower security margins as shown in [6]. The user can decide if including or not these limits. Thus, $h = [I_{ij}, I_{ji}, Q_G, V, P_{ij}]$.

The limits in p are represented as follows:

$$
\begin{aligned}
P_{S_{\min}} &\leq P_S \leq P_{S_{\max}} \\
P_{D_{\min}} &\leq P_D \leq P_{D_{\max}}
\end{aligned}
\tag{14}
$$

B. Maximum loading condition constraints

The second set of power flow variables $\hat{x} \in \mathbb{R}^n$ and equations $\hat{g} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \mapsto \mathbb{R}^n$, together with their associated constraints $h(\hat{x}) : \mathbb{R}^n \mapsto \mathbb{R}^\ell$, are introduced to represent the solution associated with a loading parameter λ, where λ represents an increase in generator and load powers, as follows:

$$
\begin{aligned}
\hat{P}_G &= (1 + \lambda + \hat{k}_G) P_G \\
\hat{P}_L &= (1 + \lambda) P_L
\end{aligned}
\tag{15}
$$

In (15), the scalar variable $\hat{k}_G$ allocates losses, assuming a distributed slack bus model. If using the maximum loading condition constraints, (1) is similar to the optimization problems which were proposed in [16] and [6]. It is assumed that (1) has a solution for $\lambda = 0$, i.e. the base loading conditions do not exceed the maximum system loading. Observe that for the maximum loading condition problem, only $\hat{g}$ and $\hat{h}$ are used, while g and h are neglected.

Boundaries for the loading margin λ have been included in (1) based on practical considerations. The minimum limit $\lambda_{\min}$ is introduced in order to ensure a minimum level of security in any operating condition, where the maximum value $\lambda_{\max}$ imposes a maximum required security level. These conditions ensure that the loading parameter remains within certain limits to avoid solutions of (1) characterized by either low security levels ($\lambda < \lambda_{\min}$) or low supply and demand levels ($\lambda > \lambda_{\max}$), which would be unacceptable [6]. Observe also that one can set $\lambda_{\max} = \lambda_{\min}$, thus fixing a desired value of the loading parameter.

5. Case Studies

In this section, a simple 3-bus test system and the RTS-96 24-bus test system [11] will be used for illustrating how PSAT and the PSAT-GAMS interface work. Authors interested in reproducing the outputs could retrieve the data from the PSAT web site [12]).

Figure 6 depicts the 3-bus test system as obtained using the PSAT-Simulink library. Each pictorial block of the diagram hides a mask where the user can set up the data associated with the correspondent component. Thus, the user can easily draw the one-line diagram and filling up the block masks to set up system data. If no data are set, default block parameters will be used.
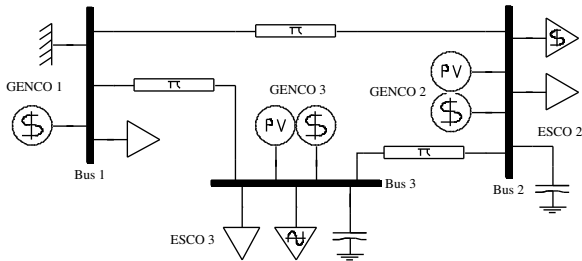

Fig. 6. PSAT-Simulink model of the three-bus example.

Once the model is completed, it has to be loaded in the Matlab workspace. While loading the Simulink data, PSAT takes care of the conversion into a Matlab script file in which each component is defined by means of matrices. Figure 7 represents the data file as results from the conversion of the Simulink model. Observe that this file can be easily read by Matlab, but is cannot be recognized by GAMS. Furthermore, line data are in the form of π model, while the admittance matrix elements should be passed to the GAMS optimization routines.

Next step is solving the power flow from within PSAT. Once the power flow analysis has been completed successfully, the information that can be obtained is twofold. Firstly, the set of input data is "good" as it led to a power flow solution. Secondly, PSAT has internally completed all base changes and built the admittance and Jacobian matrices need during the power flow solution method. Observe that all variables, data and results are stored as global structures in the Matlab workspace so that they are available for other routines and can be displayed at any time on the Matlab workspace.

At this point the PSAT-GAMS interface can be open, parameters and settings can be adjusted and the GAMS routine can be launched. Figures 8, 9, and 10 depict the input and output files which are used for exchanging data between PSAT and GAMS. Observe that the data input for the GAMS routine is a plain text file which is much less clear and "user friendly" than the Simulink one-line diagram of Fig. 6. As a results of the PSAT-GAMS interface operations, on the Matlab workspace will displays the output depicted in Fig. 11. Furthermore, the GAMS solution is stored in the PSAT global structures and is available for further analysis.

```
Bus.con = [ ...
1  138    1    0    2    1;
2  138    1    0    3    1;
3  138    1    0    4    1 ];

Line.con = [ ...
1  3  100  138   60 0 0  0.01  0.12  0  0  0  0  1.5  0;
1  2  100  138   60 0 0  0.01  0.12  0  0  0  0  1.5  0;
2  3  100  138   60 0 0  0.01  0.12  0  0  0  0  1.5  0 ];

Shunt.con = [ ...
3  100  400  60  0  0.5;
2  100  400  60  0  0.8 ];

SW.con = [ ...
1  100  138  1.02  0.00  1.5  -1.5  1.1  0.9  1.5  1 ];

PV.con = [ ...
2  100  400  1.00  1.00  1.5  -1.5  1.1  0.9  1;
3  100  400  1.00  1.00  1.5  -1.5  1.1  0.9  1 ];

PQ.con = [ ...
3  100  138  0.5  0.3  1.1  0.9  1;
1  100  138  1.5  0.8  1.1  0.9  1;
2  100  138  1.5  0.7  1.1  0.9  1 ];

Demand.con = [ ...
2  100  1.5  0.7  1.0  0  0  0  30  0  0  0  0  0  0;
3  100  0.5  0.3  1.0  0  0  0  35  0  0  0  0  0  0 ];

Supply.con = [ ...
1  100  1.5  1.5  0  0  0  25  0  0  0  0  0  0;
2  100  0.8  1.0  0  0  0  33  0  0  0  0  0  0;
3  100  0.8  1.0  0  0  0  32  0  0  0  0  0  0 ];

Varname.bus = {...
'Bus1'; 'Bus2'; 'Bus3'};
```

Fig. 7. 3-bus test system data in PSAT format.

```
$setglobal nBus '3'
$setglobal nLine '3'
$setglobal nPs '3'
$setglobal nPd '2'
$setglobal nBusref '1'
$setglobal control '3'
$setglobal flow '1'
```

Fig. 8. Input file `psatglobs.gms` for the three-bus test system.

```
$onempty                2.2 1.000000           2.Qg0 -0.216881
$kill Gh                3.3 1.000000           3.Qg0 -0.399686
parameter Gh /          /;                     1.Pl0 1.500000
1.1 1.379310            $kill Pd_idx           2.Pl0 1.500000
2.1 -0.689655           parameter Pd_idx /     3.Pl0 0.500000
3.1 -0.689655           2.1 1.000000           1.Ql0 0.800000
1.2 -0.689655           3.2 1.000000           2.Ql0 0.700000
2.2 1.379310            /;                     3.Ql0 0.300000
3.2 -0.689655           $kill S                1.Qgmax 1.500000
1.3 -0.689655           parameter S /          2.Qgmax 1.500000
2.3 -0.689655           1.Ps0 1.500000         3.Qgmax 1.500000
3.3 1.379310            2.Ps0 0.800000         1.Qgmin -1.500000
/;                      3.Ps0 0.800000         2.Qgmin -1.500000
$kill Bh                1.Psmax 1.500000       3.Qgmin -1.500000
parameter Bh /          2.Psmax 1.000000       1.Vmax 1.100000
1.1 -16.551724          3.Psmax 1.000000       2.Vmax 1.100000
2.1 8.275862            1.Cs 25.000000         3.Vmax 1.100000
3.1 8.275862            2.Cs 33.000000         1.Vmin 0.900000
1.2 8.275862            3.Cs 32.000000         2.Vmin 0.900000
2.2 -15.751724          /;                     3.Vmin 0.900000
3.2 8.275862            $kill D                /;
1.3 8.275862            parameter D /          $kill N
2.3 8.275862            1.Pd0 1.500000         parameter N /
3.3 -16.051724          2.Pd0 0.500000         1.g 0.689655
/;                      1.Pdmax 1.000000       2.g 0.689655
$kill Li                2.Pdmax 1.000000       3.g 0.689655
parameter Li /          1.tgphi 0.466667       1.b -8.275862
1.1 1.000000            2.tgphi 0.600000       2.b -8.275862
2.1 1.000000            1.Cd 30.000000         3.b -8.275862
3.2 1.000000            2.Cd 35.000000         1.g0 0.689655
/;                      /;                     2.g0 0.689655
$kill Lj                $kill X                3.g0 0.689655
parameter Lj /          parameter X /          1.Pijmax 999.000000
2.2 1.000000            1.V0 1.000000          2.Pijmax 999.000000
1.3 1.000000            2.V0 1.000000          3.Pijmax 999.000000
3.3 1.000000            3.V0 1.000000          1.Pjimax 999.000000
/;                      1.Pg0 1.502223         2.Pjimax 999.000000
$kill Ps_idx            2.Pg0 1.000000         3.Pjimax 999.000000
parameter Ps_idx /      3.Pg0 1.000000         /;
1.1 1.000000            1.Qg0 1.143238         $offempty
```

Fig. 9. Input file `psatdata.gms` for the three-bus test system.

```
nout = nout + 1;                    varargout{nout} = zeros(3,1);
varargout{nout} = zeros(3,1);       varargout{nout}(1) = 2.95867E+01;
varargout{nout}(1) = 1.50000E+00;   varargout{nout}(2) = 3.00000E+01;
nout = nout + 1;                    varargout{nout}(3) = 2.99175E+01;
varargout{nout} = zeros(2,1);       nout = nout + 1;
varargout{nout}(1) = 4.92559E-01;   varargout{nout} = zeros(3,1);
varargout{nout}(2) = 1.00000E+00;   varargout{nout}(1) = 6.07791E-01;
nout = nout + 1;                    varargout{nout}(2) = 7.57890E-01;
varargout{nout} = zeros(3,1);       varargout{nout}(3) = 1.50303E-01;
varargout{nout}(1) = 1.10000E+00;   nout = nout + 1;
varargout{nout}(2) = 1.09589E+00;   varargout{nout} = zeros(3,1);
varargout{nout}(3) = 1.09695E+00;   varargout{nout}(1) = 6.07791E-01;
nout = nout + 1;                    varargout{nout}(2) = 7.57890E-01;
varargout{nout} = zeros(3,1);       varargout{nout}(3) = 1.50303E-01;
varargout{nout}(2) =-8.30605E-02;   nout = nout + 1;
varargout{nout}(3) =-6.65812E-02;   varargout{nout} = zeros(3,1);
nout = nout + 1;                    varargout{nout}(1) =-5.26614E-01;
varargout{nout} = zeros(3,1);       varargout{nout}(2) =-6.81753E-09;
varargout{nout}(1) = 7.97250E-01;   nout = nout + 1;
varargout{nout}(2) = 4.06715E-02;   varargout{nout} = zeros(3,1);
varargout{nout}(3) = 3.45458E-01;   nout = nout + 1;
nout = nout + 1;                    varargout{nout} = zeros(3,1);
```

Fig. 10. Output file `psatsol.m` for the three-bus test system.

```
--------------------------------------------------------------
  PSAT-GAMS Interface
--------------------------------------------------------------

Standard OPF
Single-Period Auction

Power Supplies
Bus        Ps            Ps max        Ps min
<i>        [MW]          [MW]          [MW]
1.0000     150.0000      150.0000      0.0000
3.0000     0.0000        100.0000      0.0000
2.0000     0.0000        100.0000      0.0000


Power Demands
Bus        Pd            Pd max        Pd min
<i>        [MW]          [MW]          [MW]
3.0000     100.0000      100.0000      0.0000
2.0000     49.2559       100.0000      0.0000


Generator Reactive Powers
Bus        Qg            Qg max        Qg min
<i>        [MVar]        [MVar]        [MVar]
1.0000     79.7251       150.0000      -150.0000
2.0000     4.0672        150.0000      -150.0000
3.0000     34.5459       150.0000      -150.0000

Power Flow Solution
Bus    V         theta     PG        PL        QG        QL
<i>    [p.u.]    [rad]     [MW]      [MW]      [MVar]    [MVar]
1.0000 1.1000    0.0000    300.2223  150.0000  79.7251   80.0000
2.0000 1.0959    -0.0831   100.0000  199.2559  4.0672    92.9861
3.0000 1.0970    -0.0666   100.0000  150.0000  34.5459   90.0000

Prices and Pays
Bus    LMP       NCP       Pay S     Pay D
<i>    [$/MWh]   [$/MWh]   [$/h]     [$/h]
1.0000 29.5868   0.0000    -8882.6037 4438.0139
2.0000 30.0000   0.0292    -3000.0000 5977.6770
3.0000 29.9176   0.0231    -2991.7556 4487.6335

Flows on Transmission Lines
From   To    Iij      Iijmax    Margin    Iji      Ijimax    Margin
<i>    <j>   [p.u.]   [p.u.]    [p.u.]    [p.u.]   [p.u.]    [p.u.]
1.0000 3.0000 0.6078  999.0000  998.3922  0.6078   999.0000  998.3922
1.0000 2.0000 0.7579  999.0000  998.2421  0.7579   999.0000  998.2421
2.0000 3.0000 0.1503  999.0000  998.8497  0.1503   999.0000  998.8497

Totals
-------------------------------------------------------------
Total Losses = 0.966 [MW]
Bid Losses = 0.744 [MW]
Total demand = 149.2559 [MW]
Total Transaction Level =  499.255 [MW]
ISO Pay = 28.9638 [$/h]

-------------------------------------------------------------
Check file ~\fmilano\psat\fm_gams.lst for GAMS report.
PSAT-GAMS Optimization Routine completed in 1.292 s
```

Fig. 11. PSAT-GAMS OPF report for the three-bus test system.

One of the main advantages of Matlab are its advanced graphical features. Thus, the PSAT-GAMS interface provides a bridge in between the sophisticated GAMS solvers and the versatile plotting utilities of Matlab. To illustrate this feature, let us consider the RTS-96 24-bus test system. The 24-bus system as obtained using the PSAT-Simulink library is depicted in Figure 12.
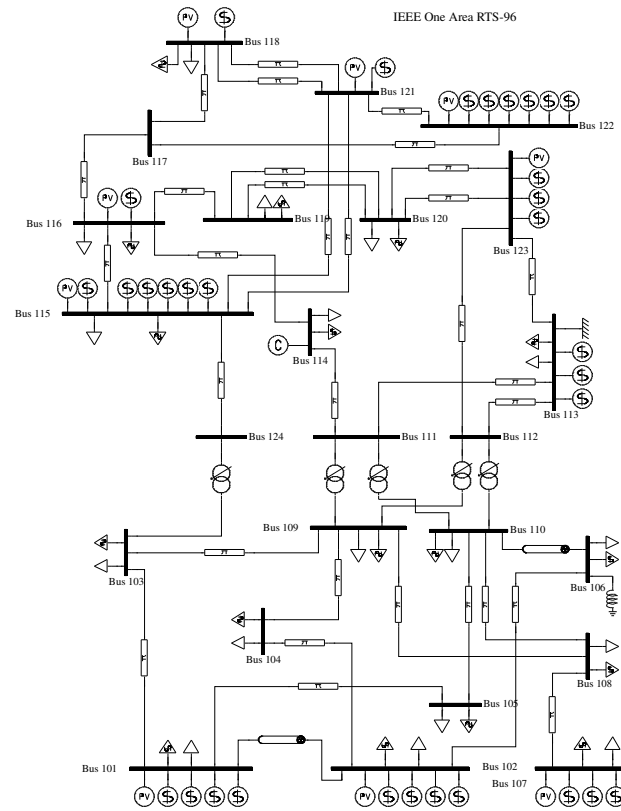

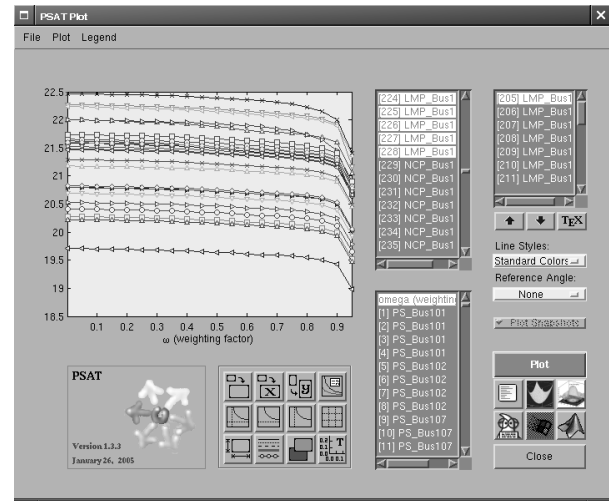Fig. 12. PSAT-Simulink model of the RTS-96 24-bus test system.

Fig. 13. PSAT GUI for plotting OPF results. The curves represent LMPs for the 24-bus system as a function of the weighting factor ω.

Figures 13 and 14 depict the Locational Marginal Prices (LMPs) and the Total Transaction Level (TTL $= \Sigma P_{L_i}$), respectively, as a function of the weighting factor ω. Results are for the multi-objective voltage stability constrained OPF problem which was described in [6].
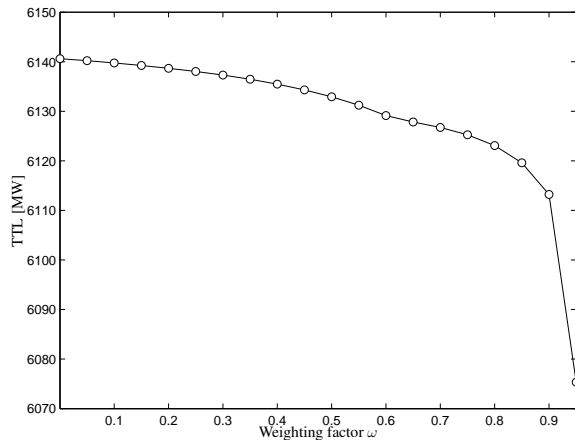
Fig. 14. Total transaction level (TTL) for the 24-bus system as a function of the weighting factor ω.

6. Conclusions

The proposed interface is based on PSAT, which is a user-friendly toolbox provided with a complete graphical user interface and a graphical editor of one-line network diagrams. PSAT is currently used by several undergraduates, Ph.D. students and researchers, and has an active mailing list (http://groups.yahoo.com/groups/psatforum) currently counting about 400 members.

The PGI provides a variety of OPF-based market and stability optimization problems. The interface merges the sophisticated optimization solvers provided by GAMS with the advanced graphic tools of Matlab. In its current form, it is a convenient tool for educational purposes.

Among future projects, there are including new market clearing models based on mixed integer linear programming and extending the currently implemented OPF to dynamic models (e.g. synchronous machines and controllers). The PGI is open source, thus allowing anyone to quickly extend its capabilities and/or including new optimization problems. Any suggestion and/or bug report are very welcome.

References

[1] M. Huneault and F. D. Galiana, "A Survey of the Optimal Power Flow Literature," *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 762–770, May 1991.

[2] G. B. Sheblé, *Computational Auction Mechanism for Restructured Power Industry Operation*. Boston: Kluwer Academic Publishers, 1998.

[3] M. Shahidehpour and M. Alomoush, *Restructured electric power systems*. New York: Marcel Dekker Publishers, 2001.

[4] M. Shahidehpour, H. Yamin, and Z. Li, *Market operations in electric power systems: forecasting, scheduling and risk management*. New York: John Wiley and Sons, 2002.

[5] A. J. Conejo and J. M. Arroyo, "Multiperiod Auction for a Pool-based electricity market," *IEEE Transactions on Power Systems*, vol. 17, no. 4, pp. 1225–1231, Nov. 2002.

[6] F. Milano, C. A. Cañizares, and M. Invernizzi, "Multi-objective Optimization for Pricing System Security in Electricity Markets," *IEEE Transactions on Power Systems*, vol. 18, no. 2, May 2003.

[7] F. Milano, "An Open Source Power System Analysis Toolbox," *Accepted for publication on IEEE Trans. on Power Systems*, Mar. 2005.

[8] R. M. Stallman, *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Boston: Free Software Foundation, 2002.

[9] J. W. Eaton, *GNU Octave Manual*. Bristol, UK: Network Theory Ltd., 1997.

[10] A. Brooke, D. Kendrick, A. Meeraus, R. Raman, and R. E. Rosenthal, *GAMS, a User's Guide*, GAMS Development Corporation, 1217 Potomac Street, NW, Washington, DC 20007, USA, Dec. 1998, available at http://www.gams.com/.

[11] Reliability Test System Task Force of the Application of Probability Methods subcommittee, "The IEEE Reliability Test System - 1996," *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 1010–1020, Aug. 1999.

[12] F. Milano, "PSAT, Matlab-based Power System Analysis Toolbox," 2002, available at http://thunderbox.uwaterloo.ca/~fmilano.

[13] K. Schoder, A. Feliachi, and A. Hasanović, "PAT: A Power Analysis Toolbox for Matlab/Simulink," *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 42–47, Feb. 2003.

[14] C. D. Vournas, E. G. Potamianakis, C. Moors, and T. V. Cutsem, "An Educational Simulation Tool for Power System Control and Stability," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 48–55, Feb. 2004.

[15] M. C. Ferris, *MATLAB and GAMS: Interfacing Optimization and Visualization Software*, Computer Sciences Department, University of Wisconsin-Madison, Aug. 1999, available at http://www.cs.wisc.edu/mathprog/matlab.html.

[16] W. D. Rosehart, C. A. Cañizares, and V. Quintana, "Multi-Objective Optimal Power Flows to Evaluate Voltage Security Costs in Power Networks," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 578–587, May 2003.

[17] K. Xie, Y.-H. Song, J. Stonham, E. Yu, and G. Liu, "Decomposition Model and Interior Point Methods for Optimal Spot Pricing of Electricity in Deregulation Environments," *IEEE Transactions on Power Systems*, vol. 15, no. 1, pp. 39–50, Feb. 2000.