Applications of Symbolic Computation to Power System Analysis and Teaching

C. A. Cañizares, *Senior Member, IEEE*

***Abstract*— This paper describes the history and use of symbolic computation in power systems, from the tools used for power system analysis such as power flows, to its application in the classroom to facilitate the explanation and understanding of complex models and concepts such as device modeling and simulation. An example of the use of symbolic computation to develop a basic power flow program with the help of MATLAB®, and to explain the advantages and disadvantages of using different types of Jacobians in the solution process is discussed in detail.**

***Index Terms*— Symbolic computation, symbolic-assisted numeric computation, power systems, modeling, simulation, power flow, education.**

I. INTRODUCTION

SYMBOLIC computational tools have been widely applied in a variety of fields [1], [2], [3]. These programs were basically designed to obtain explicit symbolic solutions to a variety of linear and nonlinear mathematical problems, such as equation or integration problems. However, a variety of numeric computational algorithms and techniques have also been integrated into these tools to help with the solution and visualization of these problems. On the other hand, some numeric computational tools such as [4], which were basically designed to numerically solve and visualize a variety of mathematical problems, now include some limited symbolic computational capabilities, thus transforming these kinds of programs into symbolic-assisted numeric computational tools.

In the area of power systems, pioneer work was performed in the late 80's at the University of Wisconsin-Madison in the development, application and use of symbolic computational tools [5], [6], [7], as well as symbolic-assisted numeric computational tools [8], [9]. Thus, in [5] and [6], the authors discuss the development and application of generic symbolic and numeric computational tools and graphical user interfaces for obtaining solutions of nonlinear problems and related equations, especially those associated with the simulation of power system transients. Applications of these methods and tools to the solution of standard and three-phase power flows as well as transient stability problems in power systems are discussed in [7]. The principles and techniques discussed in these papers were used in the development of the symbolic-assisted numeric computational tools [8] and [9].

The use of these tools for power system modeling and simulation has yielded some interesting and unique results, as discussed in detail in [10]. For example, in [11], a special technique for obtaining synchronous machine parameters for detailed EMTP-types from field measurements based on implicit numerical techniques is presented; this was possible due to the symbolic capabilities of the program used for these studies. In [12], on the other hand, the use of symbolic-assisted numeric computational tools for the modeling and simulation of inductions motors for research and educational purposes is discussed in detail.

Nowadays, symbolic and symbolic-assisted numeric computational tools have become part of the mainstream programs used for the analysis, modeling and simulation of power systems, particularly in research and education. For example, the MATLAB® based programs [13] and [14], which are numeric computational tools that can be used to perform a variety of power system studies, have been developed, especially the latter, using the symbolic capabilities of MATLAB®. These tools and others (e.g. [15]) are a good example of how symbolic computation can be used in the development of commercial-grade software for power system analysis, as Jacobians and Hessians can be readily and reliably obtained symbolically for a variety of system models to then integrate them into any numeric computational tool.

In the next section, an example of the use of symbolic computation for the development of MATLAB® routines to generate and solve simple power flow equations for educational purposes is discussed in detail.

II. EXAMPLE

MATLAB® has become a ubiquitous tool in science and engineering R&D environments. In universities across the world, it is widely used for educational and research purposes, particularly to solve, visualize and illustrate a variety of mathematical problems associated with different types of concepts and processes, especially in electrical engineering. In the area of power systems, this program has become quite popular for research and teaching applications, with a variety of MATLAB®-based programs being made available for power system simulation, modeling and analysis (e.g. [13], [14], [16]).

At Waterloo, MATLAB® is widely used in research and teaching, particularly in the areas of power system modeling and analysis. For example, a project that uses both the symbolic and numeric capabilities of MATLAB® to develop a basic power flow program is used to teach undergraduate students some basic power system concepts. The project has the following three parts:

Paper presented at Panel: Symbolic Computation for Power Systems.

The financial support provided by the Natural Science and Engineering Research Council (NSERC) of Canada for this paper is gratefully acknowledged.

C. A. Cañizares is with the Dept. of Electrical & Computer Engineering, University of Waterloo, Waterloo, ON, N2L-3G1, Canada (e-mail: ccanizar@uwaterloo.ca).

1) Develop a basic solution routine, similar to `fsolve()`, to solve a set of nonlinear equations of the form

$$f(x) = 0$$

based on a robust Newton-Raphson solution technique as well as both numeric and symbolic Jacobians, and sparse matrix techniques using the MATLAB® routines `sparse()` and `lu()`. Numeric Jacobians are generated using the central-difference formula:

$$\left.\frac{\partial f}{\partial x_i}\right|_{x_o} = \frac{1}{h}\left[f(x_{1o},\ldots,x_{io}+h/2,\ldots,x_{no}) - f(x_{1o},\ldots,x_{io}-h/2,\ldots,x_{no})\right]$$

and symbolic Jacobians are generated using the symbolic capabilities of MATLAB®, in particular the routines `sym()`, `jacobian()` and `subs()`. Given the cost of computing symbolic Jacobians, the Jacobian is obtained at the start of the solution process and "stored" in a `.m` file to evaluate it as needed within the iterative solution technique.

2) Develop an input routine that reads basic IEEE Common Format power flow data [17], and generates the desired nonlinear equations to be solved using the solution routine previously developed (Numeral 1). The basic routine to generate the desired equations is shown in Fig. 1, assuming that the admittance matrix `Y-bus` and the bus numbers of the slack and PV buses are known. The equations generated by this routine are stored in a `.m` file (`test.m`) to be used in the solution process. The equations obtained from applying this routine to the 16-bus test system depicted in Fig. 2 are shown in Fig. 3.

3) Develop a graphical user interface to handle the input and output data needed and generated by the routines described in the previous numerals.

```
function func(YB,Slack,PV)
% Create a symbolic set of power flow equations
% Input:  YB -> Y-bus natrix
%         Slack -> Slack bus number
%         PV -> PV buses list, i.e. [n1 n2 n3 ....]
% Output: Power flow equations in 'test.m' file

n=length(YB);
m=length(PV);

% Open output file 'test.m' to store power flow equations
fid=fopen('test.m','w');
fprintf(fid,'function f=func(x)\n\n');

% Define symbolic voltages and powers (power flow variables)
l=1;
for k=1:n,
   V(k)=sym(sprintf('V%d*cos(d%d)+i*V%d*sin(d%d)',k,k,k,k));
   Vc(k)=sym(sprintf('V%d*cos(d%d)-i*V%d*sin(d%d)',k,k,k,k)); % conjugate
   pv = find(PV==k);
   if k==Slack,  % Slack bus
     fprintf(fid,'%c Slack bus\n','%');
     fprintf(fid,'Pg%d=x(%d);\n',k,l);  l=l+1;
     fprintf(fid,'Qg%d=x(%d);\n',k,l);  l=l+1;
     fprintf(fid,'V%d=Vs;\n',k);
     fprintf(fid,'d%d=0;\n',k);
   elseif pv,  % PV buses
     fprintf(fid,'%c PV bus\n','%');
     fprintf(fid,'Qg%d=x(%d);\n',k,l);  l=l+1;
     fprintf(fid,'d%d=x(%d);\n',k,l);  l=l+1;
     fprintf(fid,'V%d=V(%d);\n',k,pv);
   else, % PQ buses
     fprintf(fid,'V%d=x(%d);\n',k,l);  l=l+1;
     fprintf(fid,'d%d=x(%d);\n',k,l);  l=l+1;
   end
end


% Obtain power flow equations
YBc=conj(YB);
Ic=YBc*Vc.';

fprintf(fid,'\n\n');
l=1;
for k=1:n,
   S=vpa(V(k)*Ic(k),5);
   pv = find(PV==k);
   if k==Slack,  % Slack bus
     fprintf(fid,'f(%d)=Pg%d-Pl(%d)-real(%s);\n',l,k,k,char(S));  l=l+1;
     fprintf(fid,'f(%d)=Qg%d-Ql(%d)-imag(%s);\n',l,k,k,char(S)); l=l+1;
   elseif pv,  % PV buses
     fprintf(fid,'f(%d)=Pg(%d)-Pl(%d)-real(%s);\n',l,k,k,char(S));  l=l+1;
     fprintf(fid,'f(%d)=Qg%d-Ql(%d)-imag(%s);\n',l,k,k,char(S)); l=l+1;
   else,   % PQ buses
     fprintf(fid,'f(%d)=Pg(%d)-Pl(%d)-real(%s);\n',l,k,k,char(S));  l=l+1;
     fprintf(fid,'f(%d)=Qg(%d)-Ql(%d)-imag(%s);\n',l,k,k,char(S)); l=l+1;
   end
end

fclose(fid);
```

Fig. 1. MATLAB® routine to generate symbolic power flow equations.

III. Conclusions

Symbolic computational techniques and tools can be readily used to improve model and software development and testing in all areas of science and engineering. These techniques and tools have various advantages and disadvantages, depending on the application environment; they are particularly useful for educational and research purposes. In the development of commercial applications for power systems, symbolic computation has proven to be very useful for the automatic generation of Jacobians and Hessians, as well as for code testing. It is important to highlight the fact that as computational power has significantly increased in the last few years, the areas and types of applications of symbolic computation have also increased quite considerably; however, computational burden is still an issue that limits the direct use of symbolic computational techniques and tools in commercial-grade applications for power system studies.

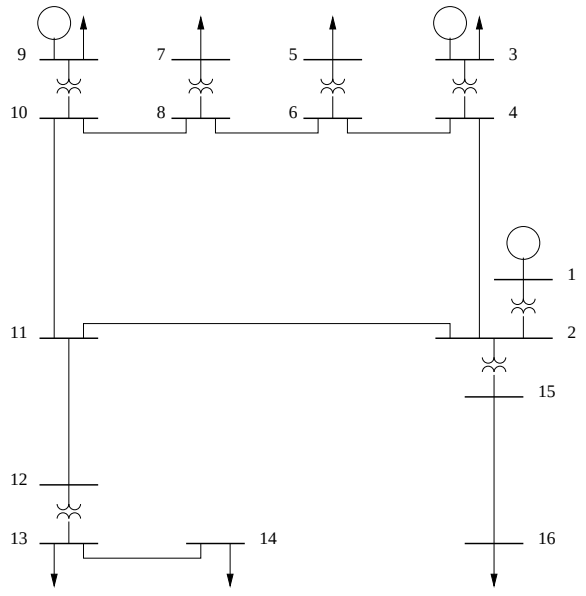

Fig. 2. 16-bus test system [18].

References

[1] S. Wolfram, *Mathematica–A System for Doing Mathematics by Computer*. Addison-Wesley, second edition, 1991.

```
function f=func(x)
% Power flow equations

Vs=1;
V=[1.035 1.05];
Pg=zeros(1,16);
Qg=zeros(1,16);
Pg(3)=110/100;
Pg(9)=220/100;
Pl=[0 0 10 0 75 0 90 0 15 0 0 0 0 50 35 0 150]/100;
Ql=[0 0 55 0 15 0 20 0  4 0 0 0  2  3 0  20]/100;

% Slack bus
Pg1=x(1);
Qg1=x(2);
V1=Vs;
d1=0;
V2=x(3);
d2=x(4);
% PV bus
Qg3=x(5);
d3=x(6);
V3=V(1);
V4=x(7);
d4=x(8);
.
.
.
V16=x(31);
d16=x(32);


f(1)=Pg1-Pl(1)-real((V1*cos(d1)+1.*i*V1*sin(d1))*((2.8289+28.289*i)*(V1*cos(d1)-
1.*i*V1*sin(d1))+(-2.8289-28.289*i)*(V2*cos(d2)-1.*i*V2*sin(d2))));
f(2)=Qg1-Ql(1)-imag((V1*cos(d1)+1.*i*V1*sin(d1))*((2.8289+28.289*i)*(V1*cos(d1)-
1.*i*V1*sin(d1))+(-2.8289-28.289*i)*(V2*cos(d2)-1.*i*V2*sin(d2))));
f(3)=Pg(2)-Pl(2)-real((V2*cos(d2)+1.*i*V2*sin(d2))*((-2.8289-28.289*i)*(V1*cos(d1)-
1.*i*V1*sin(d1))+(9.4767+80.422*i)*(V2*cos(d2)-1.*i*V2*sin(d2))+(-2.0699-10.973*i)*(V4*cos(d4)-
1.*i*V4*sin(d4))+(-2.0699-10.973*i)*(V11*cos(d11)-1.*i*V11*sin(d11))+(-2.3062-
27.930*i)*(V15*cos(d15)-1.*i*V15*sin(d15))));
f(4)=Qg(2)-Ql(2)-imag((V2*cos(d2)+1.*i*V2*sin(d2))*((-2.8289-28.289*i)*(V1*cos(d1)-
1.*i*V1*sin(d1))+(9.4767+80.422*i)*(V2*cos(d2)-1.*i*V2*sin(d2))+(-2.0699-10.973*i)*(V4*cos(d4)-
1.*i*V4*sin(d4))+(-2.0699-10.973*i)*(V11*cos(d11)-1.*i*V11*sin(d11))+(-2.3062-
27.930*i)*(V15*cos(d15)-1.*i*V15*sin(d15))));
.
.
.
f(31)=Pg(16)-Pl(16)-real((V16*cos(d16)+1.*i*V16*sin(d16))*((-1.1634-6.3753*i)*(V15*cos(d15)-
1.*i*V15*sin(d15))+(1.1634+6.3618*i)*(V16*cos(d16)-1.*i*V16*sin(d16))));
f(32)=Qg(16)-Ql(16)-imag((V16*cos(d16)+1.*i*V16*sin(d16))*((-1.1634-6.3753*i)*(V15*cos(d15)-
1.*i*V15*sin(d15))+(1.1634+6.3618*i)*(V16*cos(d16)-1.*i*V16*sin(d16))));
```

Fig. 3. MATLAB® power fbw equations for the 16-bus system of Fig. 2 generated by the routine depicted in Fig. 1.

[2] A. Heck, *Introduction to Maple*. Springer-Verlag, third edition, 2003.

[3] P. J. Pritchard, *MathCad: A Tool for Engineering Problem Solving (B.E.S.T. Series)*. McGraw Hill, 1998.

[4] *MATLAB–High-Performance Numeric Computation and Visualization Software*, The Math Works Inc., Natick, Massachusetts, 1992.

[5] F. L. Alvarado, R. H. Lasseter, and Y. Liu, "An Integrated Engineering Simulation Environment," *IEEE Transactions on Power Systems*, vol. 3, no. 1, pp. 245–253, Feb. 1988.

[6] F. L. Alvarado and Y. Liu, "General Purpose Symbolic Simulation Tools for Electric Networks," *IEEE Transactions on Power Systems*, vol. 3, no. 2, pp. 689–697, May 1988.

[7] C. A. Cañizares and F. L. Alvarado, "Graphic and Symbolic Simulation Techniques Applied to the Analysis of Power Systems," in *Proc. North American Power Symposium*, Purdue University, Indiana, Sept. 1988, pp. 1–10.

[8] F. L. Alvarado and D. J. Ray, "Symbolically Assisted Numeric Computation in Education," *International Journal of Applied Engineering Education*, vol. 4, no. 6, pp. 519–536, 1988.

[9] *EES–Engineering Equation Solver*, F-Chart Software, Middleton, Wisconsin, 1993.

[10] F. L. Alvarado and C. A. C. and, "Symbolically-Assisted Power System Simulation," *International Journal of Electrical Power & Energy Systems*, vol. 18, no. 7, pp. 405–408, Feb. 1996.

[11] F. L. Alvarado and C. A. Cañizares, "Synchronous Machine Parameters from Sudden-Short Tests by Back-Solving," *IEEE Transactions on Energy Conversion*, vol. 4, no. 2, pp. 224–236, June 1989.

[12] F. L. Alvarado, C. A. Cañizares, A. Keyhani, and B. Coates, "Instructional Use of Declarative Languages for the Study of Machine Transients," *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 407–413, Feb. 1991.

[13] *Power System Toolbox Ver. 2.0: Dynamic Tutorial and Functions*, Cherry Tree Scientific Software, Colborne, Ontario, 1999.

[14] F. Milano, *Power System Analysis Toolbox Ver. 1.3.2*, www.power.uwaterloo.ca, Nov. 2004.

[15] C. A. Cañizares, *UWPFLOW Continuation and Direct Methods to Locate Fold Bifurcations in AC/DC/FACTS Power Systems*, University of Waterloo, Waterloo, Ontario, www.power.uwaterloo.ca, Nov. 1999.

[16] J. Mahseredjian and F. L. Alvarado, "Creating an Electromagnetic Transients Program in MATLAB: MatEMTP," *IEEE Trans. Power Delivery*, vol. 12, no. 1, pp. 380–388, Jan. 1997.

[17] IEEE Committee, "Common Format for Exchange of Solved Load Flow Data," *IEEE Trans. Power Apparatus and Systems*, vol. 92, no. 6, pp. 1916–1925, 1973.

[18] C. A. Gross, *Power Systems Analysis*. John Wiley & Sons, second edition, 1986.

Claudio A. Cañizares (SM'00) received in April 1984 the Electrical Engineer degree from the Escuela Politécnica Nacional (EPN), Quito-Ecuador, where he held different teaching and administrative positions from 1983 to 1993. His MSc (1988) and PhD (1991) degrees in Electrical Engineering are from the University of Wisconsin-Madison. Dr. Cañizares has been with the E&CE Department of the University of Waterloo since 1993, where he has held various academic and administrative positions and is currently a Professor. In 1999-2000 he was a Visiting Professor at the Politecnico di Milano as well as a research consultant at ENEL-Ricerca and CESI in Milan, Italy. His research activities concentrate in the study of stability, modeling, simulation, control and computational issues in ac/dc/FACTS systems within the framework of electricity markets.