A General-Purpose Symbolically Assisted Numeric Computation Environment as a Support in Power Engineering Education

Mevludin Glavic, *Member, IEEE*, Izudin Dzafic, and Sejid Tesnjak

***Abstract*—This paper describes and illustrates a Windows-based, general-purpose, symbolically assisted numeric computation environment within power engineering education applications. The examples considered include fundamental problems derived from three areas: power system analysis and optimization, electric machinery, and feedback control systems.**

***Index Terms*—Numeric computation, power engineering education, symbolic computation.**

I. Introduction

THE problems arising in power engineering education and research range from the steady-state to the time-domain simulations. It is desirable to have a single computation environment capable of handling such a wide diversity of problems. The availability of such a software package, together with the extensive use of a graphical user interface, transforms power engineering education and research activities into an activity where a user focuses on specific problem aspects rather than learning about advanced programming skills or the knowledge of operating systems. This paper makes use of such an environment, i.e., a Windows-based, general-purpose software package that implements the concept of symbolically assisted numeric computation.

The environment takes origin in the work of Alvarado of Wisconsin who introduced the concept of symbolically assisted numeric computation in solving a vast panel of power engineering problems and authored the software package SOLVER-Q (a PC implementation of the concept developed in Pascal for the DOS environment) [1]–[4].

The aim of the paper is to introduce this (new idea from an old source), in the authors' view, powerful computation environment (developed in Visual C++ for Windows platforms 95/98/2000/NT/XP) without comparing its capabilities to other tools: neither commercial nor freely available ones. The problems considered in the paper come from power system analysis and optimization [5], electric machinery [4], and feedback control education [6] and include the power-flow problem and its extensions, the simulation of transients produced by the cold startup of a two-phase induction motor, the economic dispatch problem, and an elementary feedback control design.

The paper is organized as follows. In Section II, the concept of symbolically assisted numeric computation is briefly introduced. The structure of the environment is presented in Section III. Four experiments are presented in Section IV, while Section V offers some conclusions.

Manuscript received July 7, 2004. Paper no. TPWRS-00266-2004.

M. Glavic is with the Electrical Engineering and Computer Science Department, University of Liège, 4000 Liège, Belgium (e-mail: mglavic@ieee.org).

I. Dzafic is with Siemens AG, Power Transmission and Distribution, D-90459, Nuremberg, Germany (e-mail: Izudin.Dzafic@ptd.siemens.de).

S. Tesnjak is with the Electrical Engineering and Computing Department, University of Zagreb, 10000 Zagreb, Croatia.

Digital Object Identifier 10.1109/TPWRS.2004.841235

II. Symbolically Assisted Numeric Computation

The usual aim of symbolic computation is to obtain explicit symbolic solutions and to gain in the understanding of a system or process. Explicit solutions obtained by most symbolic solution packages often result in lengthy expressions that contribute little to the understanding of a problem. The main uses of symbolic computation include simplification of expressions, computation of symbolic derivatives and integrals, and elimination and substitution of variables.

Commonly, symbolic software packages are regarded as fairly limited in solving large sets of nonlinear algebraic equations and nonlinear optimization problems [1]–[4]. These two problem classes are solved in the domain of numeric computation. Numeric solution approaches need the Jacobian and possibly Hessian matrix of all or of a subset of the equations. Symbolic methods can be used to derive expressions for performing numeric computations such as gradients and Jacobian and Hessian matrices. Thus, the traditional roles of numeric and symbolic computation are not as clear any more, and many benefits arise from merging the two [1], [7]. Symbolically assisted numeric computation refers to the idea of using symbolic computation not as an end in and of itself [1]–[4], [7], as is often done in many software packages, such as Mathematica [8] and Maple [9] but rather using symbolic computation as a means of assisting in the eventual numeric solution of a problem.

The building block on most symbolic manipulation packages is an equation. The building block for the concept of symbolically assisted numeric computation is a set of equations. Operations are normally applied to entire sets of equations rather than to individual equations in the set.

The main premises in the computation environment development can be summarized as follows [1]–[4].

- Simplicity is better than explicitness.
- Represent only what is essential to describe a problem.
- Reduce only for computational or clarity reasons.
- Declarative description of a problem is preferable to procedural description.
- Correctness and clarity are more important than raw speed.

III. Computational Environment

A simplified structure of the computation environment is illustrated in Fig. 1.

In the core of the environment is a powerful symbolically assisted numeric computation engine that, together with user-friendly equation editor, permits a nearly direct transition from mathematical expressions to simulations. Once the user understands the equations of interest, the environment is capable of performing all necessary aspects of the simulation. The symbolic processing of user specification allows a very high degree of flexibility. The purpose of converters is to automatically generate the system equations based on widely used standard data formats (e.g., CF and PTI formats). Many systems, especially test systems, have their own data already stored in standard formats.

The possibility of a problem statement in natural form, as one does on the blackboard or sheet of paper, and the possibility of experimenting with the problem as much as necessary makes the computational environment attractive for engineering education and research. Equation editor, i.e., user interface with some toolbars, is presented in Fig. 2.

The environment offers maximum flexibility in expressing problem statements in natural form at almost every level. The equations are entered in an intuitive manner with few rules. Equations may span several lines, or several equations may be in the same line. About the only rule is that every equation must be terminated with a semicolon. The equations can be arranged in any desired order, and text can be intermixed at any time. A user may enter the equations, describing the underlying problem, completely manually (in this case, one has to be fully familiarized with equation syntax) or using available toolbars and pads that include standard mathematical operations and functions, Greek letters, special functions, etc.

The methods for eventual numeric solution of simultaneous nonlinear algebraic equations include full Newton–Raphson (N–R) method, dishonest N–R, a robust N–R, and a class of homotopy methods. Integration of differential equations include trapezoidal, Gear, and backward Euler method.

The Properties dialog box, shown in Fig. 3, allows a numerical method choice and settings for a particular problem.

The optimization problems arising in power engineering computation range from unconstrained to inequality constrained problems. The environment enables handling different mathematical forms of optimization problems, including automatic recognition of the problem type and problem setting in the form of Lagrangian and Kuhn–Tucker optimality conditions.

Among many, the environment offers the following features (to enumerate just a few): symbolic and numeric linearization of equations, equation simplification, symbolic elimination, handling complex numbers and equations, repetitive solutions, Fast Fourier Transform, etc.

Each formulated problem can be saved as a lesson file for later use. This is the point where the instructors can benefit from the environment by preparing the problems to be solved and storing them as the lessons with the instructions on how to solve them (the environment includes the capability to differentiate between ordinary text and equations describing the underlying problem).

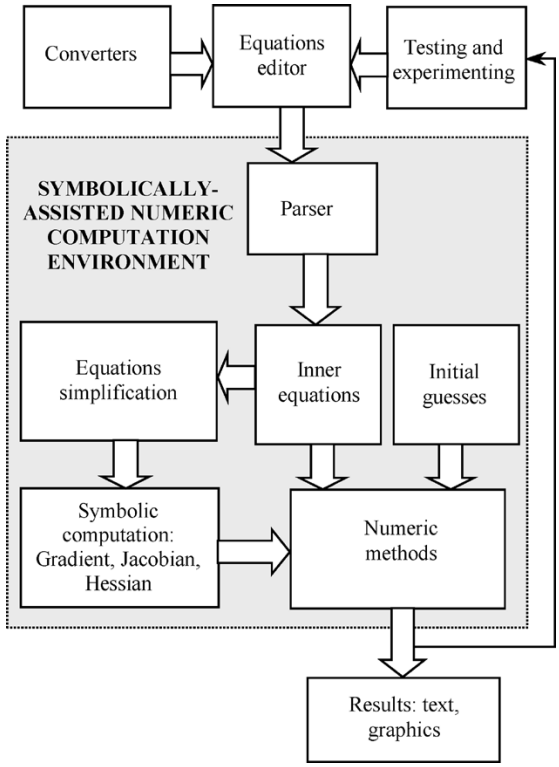

Fig. 1. Structure of the computation environment.

Forty-four lessons are included with the environment with the main aim to get a user familiar with the environment by solving different engineering and mathematical problems. In addition, 66 lessons (inherited from an earlier version of the environment [1]–[4]) come from the celebrated Sedra–Smith book on microelectronic circuits [10]. Power system engineering lessons are now being added (mainly based on [1]–[5]), and four of them are presented in the next section.

IV. Experiments

It is impossible, in a paper report, to demonstrate all the capabilities of the environment. Instead, the solving and experimenting on three fundamental problems in power system engineering education and an elementary problem in feedback control design are illustrated.

A. Power-Flow and Point of Collapse Problems

The power-flow problem consists of a set of simultaneous nonlinear equations that define a power system under steady-state operating conditions [5], [11]. In concise representation, the equations are

$$f(x,p)=0 \tag{1}$$

where p is a set of known parameters, and x is a vector of unknowns (usually bus voltage magnitudes and phase angles).

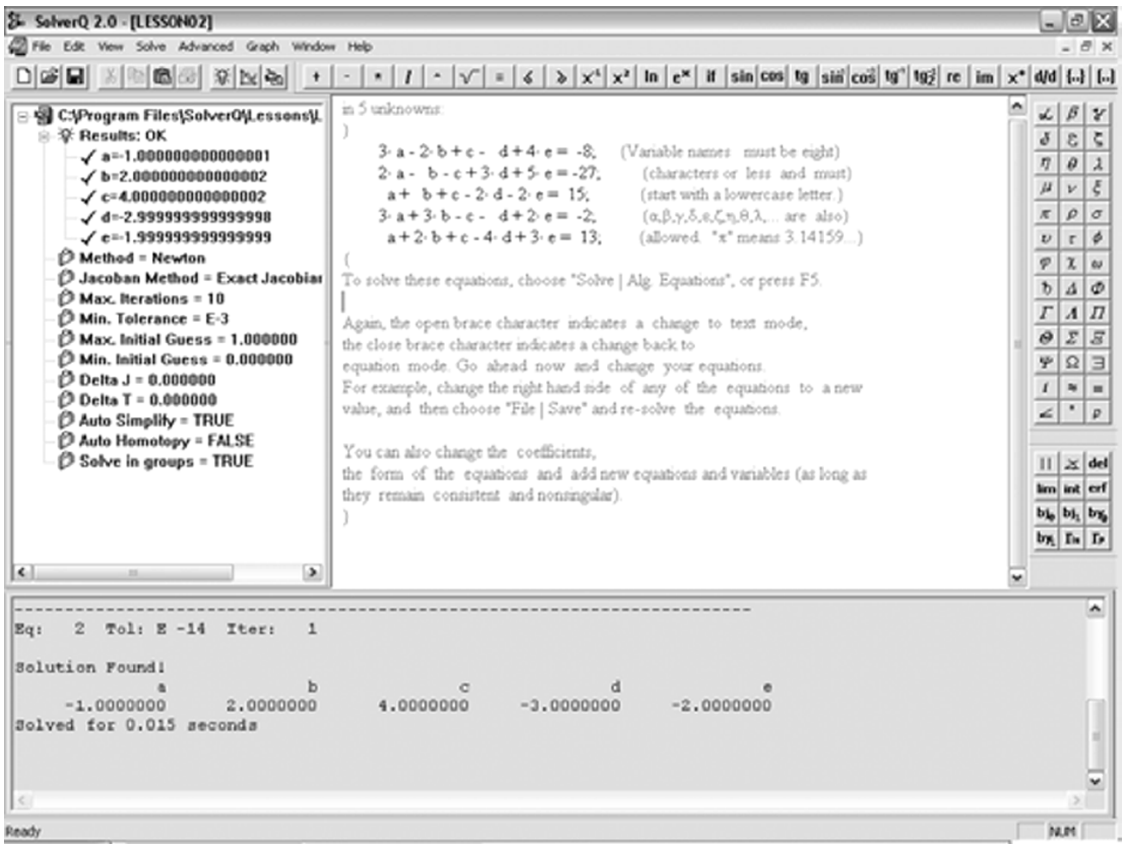

Fig. 2. General-purpose solver equation editor.

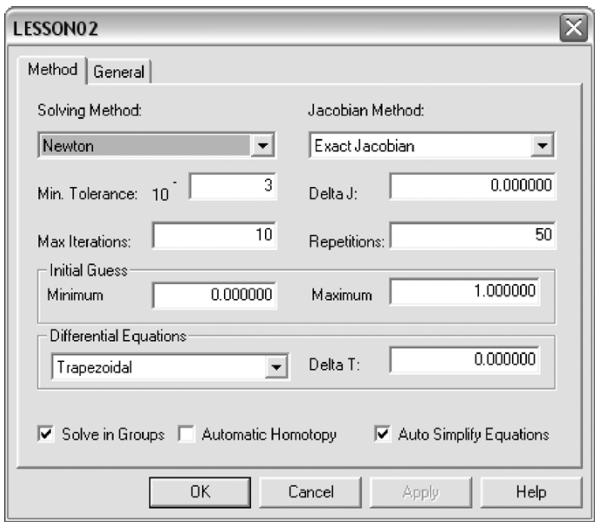

Fig. 3. Properties dialog box.

To illustrate the power-flow problem, this paper makes use of a nearly trivial 3-bus system [5]. The equations that are solved simultaneously include an active power balance equation for every bus, except one bus that is designated as the "slack" bus (bus 1 in this particular case), and a reactive power balance equation for every load bus

$$P_2 = V_2V_3[G_{23}\cos(\theta_2-\theta_3)+B_{23}\sin(\theta_2-\theta_3)] + V_2V_1[G_{21}\cos(\theta_2)+B_{21}\sin(\theta_2)]+V_2^2G_{22} \quad (2)$$

$$P_3 = V_2V_3[G_{23}\cos(\theta_3-\theta_2)+B_{23}\sin(\theta_3-\theta_2)] + V_3V_1[G_{31}\cos(\theta_3)+B_{31}\sin(\theta_3)]+V_3^2G_{33} \quad (3)$$

$$Q_2 = V_2V_3[G_{23}\sin(\theta_2-\theta_3)-B_{23}\cos(\theta_2-\theta_3)] + V_2V_1[G_{21}\sin(\theta_2)-B_{21}\cos(\theta_2)]-V_2^2B_{22} \quad (4)$$

where θ designates phase angles, V bus voltage magnitudes, and G, B real and imaginary parts of the power system Y matrix elements.

The following are the detailed equations for the system under study (as they appear in the equation editor and generated from CF format by corresponding converter), shown in the first equation at the bottom of the next page.

The solution obtained in 2 iterations (full Newton–Raphson method is a method of choice for eventual numeric solution) is

$$\mathrm{V2} = 0.950\,134\,3 \quad \alpha 2 = -0.051\,669\,6 \quad \alpha 3 = 0.041\,638\,6.$$

In addition, it is possible to get a family of solutions by letting some of parameters vary and using the environment feature of repetitive solutions. To allow a user deep inside to the results obtained, a powerful graphic module is designed within the computation environment. The user may relate any variables that describe the underlying problem. The dialog box for graph generation is illustrated in Fig. 4.

By letting parameter P_2 (the load in the particular system node) vary, a continuum of solution is obtained and presented in Fig. 5 in the form of so-called P-V curve. P-V curves are

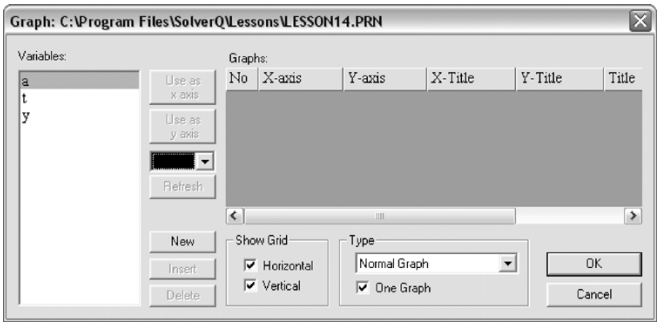

Fig. 4. Graph generation dialog box.

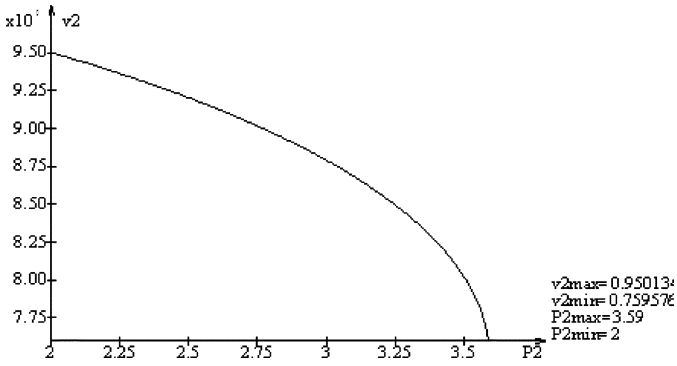

Fig. 5. P-V curve.

extensively used in determination of a system maximum loadability [11]–[13].

The equations describing the problem are in the second equation at the bottom of the page.

One of the attractions of the computation environment is the fact that it can be used to formulate new problems and variants from already defined ones. The point of collapse problem [12] can easily be formulated from the power-flow problem. Assume that one wants to find a solution to the power-flow problem with the requirement that Jacobian matrix f_x be singular (bifurcation point) [12], [13]. The objective is to find the values of the parameters that result in this singularity condition. This problem is of importance to determine limits of operational capability in a power system. Mathematically, this condition can be expressed as [12], [13]

$$f(x,p) = 0 \tag{5}$$

$$f_x(x,p)\, y = 0 \tag{6}$$

$$\|y\| = 1. \tag{7}$$

Moreover, once a bifurcation point is located, it is easy to modify the above equations to find the solutions near the bifurcation points [13], as follows:

$$f(x,p) = 0 \tag{8}$$

$$[f_x(x,p) - \alpha I]\, y = 0 \tag{9}$$

$$\|y\| = 1 \tag{10}$$

for values of $\alpha \in [-\varepsilon_1, \varepsilon_2]$ with $\varepsilon_1, \varepsilon_2 > 0$.

Owing to the symbolic linearization capability of the computation environment, it is easy to find symbolically linearized equations of (1) and then append these equations to the original equations (power-flow problem) and add the condition to ensure that the eigenvector y is nontrivial by editing. This augmented problem is then solved to yield the point of collapse solution. The equations describing the problem (in the equation editor format) are shown in the first equation at the bottom of the next page.

The solution of the problem, obtained in four iterations by applying full N–R method for eventual numeric solution to the

---

{Power Flow equations}
{Generated from CF data file}
$\mathrm{V2} * ((-(4 * \mathrm{COS}(\alpha 2)) + 5 * \mathrm{SIN}(\alpha 2) + 8 * \mathrm{V2}) - 4.4 * \mathrm{COS}(\alpha 2 - \alpha 3) + 11 * \mathrm{SIN}(\alpha 2 - \alpha 3)) = -2;$
$\mathrm{V2} * ((-(4 * \mathrm{SIN}(\alpha 2)) - 5 * \mathrm{COS}(\alpha 2) + 15 * \mathrm{V2}) - 4.4 * \mathrm{SIN}(\alpha 2 - \alpha 3) - 11 * \mathrm{COS}(\alpha 2 - \alpha 3)) = -1;$
$1.1 * (-(4 * \mathrm{V2} * \mathrm{COS}(\alpha 3 - \alpha 2)) + 10 * \mathrm{V2} * \mathrm{SIN}(\alpha 3 - \alpha 2) + 4.4) = 1.7;$
$\mathrm{V2} \approx 1; \alpha 2 \approx 0; \alpha 3 \approx 0;$

{Power Flow equations}
{Generated from CF data file}
$\mathrm{V2} * ((-(4 * \mathrm{COS}(\alpha 2)) + 5 * \mathrm{SIN}(\alpha 2) + 8 * \mathrm{V2}) - 4.4 * \mathrm{COS}(\alpha 2 - \alpha 3) + 11 * \mathrm{SIN}(\alpha 2 - \alpha 3)) = -\mathrm{P2};$
$\mathrm{V2} * ((-(4 * \mathrm{SIN}(\alpha 2)) - 5 * \mathrm{COS}(\alpha 2) + 15 * \mathrm{V2}) - 4.4 * \mathrm{SIN}(\alpha 2 - \alpha 3) - 11 * \mathrm{COS}(\alpha 2 - \alpha 3)) = -1;$
$1.1 * (-(4 * \mathrm{V2} * \mathrm{COS}(\alpha 3 - \alpha 2)) + 10 * \mathrm{V2} * \mathrm{SIN}(\alpha 3 - \alpha 2) + 4.4) = 1.7;$
$\mathrm{P2}\ \{\mathrm{newP2}\} =^{\circ}\ 2; \quad \mathrm{newP2} = \mathrm{P2} + 0.01;$
$\mathrm{V2} \approx 1; \alpha 2 \approx 0; \alpha 3 \approx 0;$

problem is shown in the second equation at the bottom of the page.

B. Economic Dispatch Problem

The problem is to minimize the cost rate of thermal units in a power system [5]. Mathematically speaking, the problem may be stated very concisely. That is, an objective function F_T is equal to the total cost for supplying the indicated load. The problem is to minimize F_T subject to the equality constraint that the sum of powers generated (P) must equal the received load (P_{load}) and one (or two) inequality constraint per generator (maximum or minimum active power generated). That is

$$F_T = F_1 + F_2 + \cdots + F_N = \sum_{i=1}^{N} F_i(P_i) \tag{11}$$

$$0 = P_{\text{load}} - \sum_{i=1}^{N} P_i \tag{12}$$

$$P_{i\min} \le P_i \le P_{i\max},\ i = 1, \ldots, N \tag{13}$$

where N is total number of thermal generators in the system.

The computational environment solves optimization problems with both equality and inequality constraints using the Lagrange multipliers method. The existence of inequality constraints and assignment of slack variables is automatic.

Suppose that one wishes to determine the economic operating point for three generating units [5] when delivering a total of 850 MW. Let us define the fuel cost of each unit as

$$\begin{aligned} F_1(P_1) &= 1.1 \cdot (510 + 7.92P_1 + 0.001\,42P_1^2) \\ F_2(P_2) &= 310 + 7.85P_2 + 0.001\,94P_2^2 \\ F_3(P_3) &= 78 + 7.97P_3 + 0.004\,82P_3^2 \end{aligned} \tag{14}$$

and let us include a simplified loss expression

$$P_{\text{loss}} = 0.000\,03P_1^2 + 0.0009P_1^2 + 0.000\,12P_1^2. \tag{15}$$

The maximum outputs of the units are specified as $P_{1\max} = 400$ MW, $P_{2\max} = 300$ MW, and $P_{3\max} = 300$ MW.

{Power Flow equations}
$\mathrm{V2}*((-(4*\mathrm{COS}(\alpha 2)) + 5*\mathrm{SIN}(\alpha 2) + 8*\mathrm{V2}) - 4.4*\mathrm{COS}(\alpha 2 - \alpha 3) + 11*\mathrm{SIN}(\alpha 2 - \alpha 3)) = -\mathrm{P2};$
$\mathrm{V2}*((-(4*\mathrm{SIN}(\alpha 2)) - 5*\mathrm{COS}(\alpha 2) + 15*\mathrm{V2}) - 4.4*\mathrm{SIN}(\alpha 2 - \alpha 3) - 11*\mathrm{COS}(\alpha 2 - \alpha 3)) = -1;$
$1.1*(-(4*\mathrm{V2}*\mathrm{COS}(\alpha 3 - \alpha 2)) + 10*\mathrm{V2}*\mathrm{SIN}(\alpha 3 - \alpha 2) + 4.4) = 1.7;$
{Symbolically linearized equations}
$+(-4*\mathrm{COS}(\alpha 2) + 5*\mathrm{SIN}(\alpha 2) + 8*\mathrm{V2} - 4.4*\mathrm{COS}(\alpha 2 - \alpha 3) + 11*\mathrm{SIN}(\alpha 2 -$
$\alpha 3) + \mathrm{V2}*8)*\delta\mathrm{V2} + ($
$\mathrm{V2}*(-4*(-\mathrm{SIN}(\alpha 2)) + 5*\mathrm{COS}(\alpha 2) + (-4.4*(-\mathrm{SIN}(\alpha 2 - \alpha 3))) + 11*\mathrm{COS}(\alpha 2 -$
$\alpha 3)))*\delta\alpha 2 + ($
$\mathrm{V2}*(-4.4*(-(-\mathrm{SIN}(\alpha 2 - \alpha 3))) + 11*(-\mathrm{COS}(\alpha 2 - \alpha 3))))*\delta\alpha 3 + (1)*\delta\mathrm{P2} = 0;$
{eqn 1}
$+(-4*\mathrm{SIN}(\alpha 2) - 5*\mathrm{COS}(\alpha 2) + 15*\mathrm{V2} - 4.4*\mathrm{SIN}(\alpha 2 - \alpha 3) - 11*\mathrm{COS}(\alpha 2 -$
$\alpha 3) + \mathrm{V2}*15)*\delta\mathrm{V2} + ($
$\mathrm{V2}*(-4*\mathrm{COS}(\alpha 2) + (-5*(-\mathrm{SIN}(\alpha 2))) + (-4.4*\mathrm{COS}(\alpha 2 - \alpha 3)) + (-11*(-\mathrm{SIN}(\alpha 2 -$
$\alpha 3)))))$
$*\delta\alpha 2 + (\mathrm{V2}*(-4.4*(-\mathrm{COS}(\alpha 2 - \alpha 3)) + (-11*(-(-\mathrm{SIN}(\alpha 2 - \alpha 3))))))*\delta\alpha 3 = 0;$
{eqn 2}
$+(1.1*(-4*\mathrm{COS}(\alpha 3 - \alpha 2) + 10*\mathrm{SIN}(\alpha 3 - \alpha 2)))*\delta\mathrm{V2}$
$+(1.1*(-4*\mathrm{V2}*(-\mathrm{SIN}(\alpha 3 - \alpha 2)) + 10*\mathrm{V2}*\mathrm{COS}(\alpha 3 - \alpha 2)))*\delta\alpha 3 + ($
$1.1*(-4*\mathrm{V2}*(-(-\mathrm{SIN}(\alpha 3 - \alpha 2))) + 10*\mathrm{V2}*(-\mathrm{COS}(\alpha 3 - \alpha 2))))*\delta\alpha 2 = 0;$
{eqn 3}
$+(1)*\delta\mathrm{P2} = 0;$
{eqn 4}
{Non-triviality condition}
$\mathrm{SQRT}(\delta\mathrm{V2}\hat{}2 + \delta\alpha 2\hat{}2 + \delta\alpha 3\hat{}2 + \delta\mathrm{P2}\hat{}2) = 1;$
$\mathrm{V2} \approx 1.0000; \alpha 2 \approx 0.0000; \alpha 3 \approx 0.0000;$

$\mathrm{V2} = 0.745\,902\,0$	$\alpha 2 = -0.642\,422\,3$	$\alpha 3 = -0.625\,177\,2$	$\mathrm{P2} = 3.595\,313\,0$
$\delta\mathrm{V2} = 0.247\,015\,1$	$\delta\alpha 2 = 0.619\,354\,9$	$\delta\alpha 3 = 0.745\,240\,4$	$\delta\mathrm{P2} = 0.000\,000\,0.$

The problem statement, in the equation editor format, is in the form in the first equation at the bottom of the page. Kuhn-Tucker conditions (automatically generated) are in the middle equation at the bottom of the page.

The solution (obtained in five iterations) is shown in the last equation at the bottom of the page.

C. Electric Machinery Problem

For reasons of numeric stability, accuracy, and speed of computation, the method of choice for time-domain solutions of many engineering problems is often an implicit method of integration. Consider a set of nonlinear differential equations with algebraic constraints of the form

$$\dot{y} = f(x, y, t) \tag{16}$$

$$0 = g(x, y, t). \tag{17}$$

These equations are discretized to obtain an implicit function relating quantities at time k to quantities at time $k-1$. Values at k are unknown; values at $(k-1)$, $(k-2)$, etc. are assumed to be known. Implicit discretization of (16) and (17) yields the following algebraic equations:

$$\hat{f}(x_k, y_k, t_k, x_{k-1}, y_{k-1}, t_{k-1}) = 0 \tag{18}$$

$$\hat{g}(x_k, y_k, t_k) = 0. \tag{19}$$

Simultaneous solution of these equations advances in time by one step. An example of equations of this type originates from the simulation of cold start-up of a two-phase induction motor whose model is given in Fig. 6 [4].

{Minimize :}
$1.1 * (510 + 7.2 * \mathrm{p1} + 0.001\,42 * \mathrm{p1\^{}2}) + 0.001\,94 * \mathrm{p2\^{}2} + 7.85 * \mathrm{p2} +$
$0.004\,82 * \mathrm{p3\^{}2} + 7.97 * \mathrm{p3} + 388;$
{Subject to :}
$\mathrm{p1} + \mathrm{p2} + \mathrm{p3} = 850 + \mathrm{ploss};$ {and}
$\mathrm{ploss} = 0.000\,03 * \mathrm{p1\^{}2} + 0.000\,09 * \mathrm{p2\^{}2} + 0.000\,12 * \mathrm{p3\^{}2};$
{Inequality constraints :}
$\mathrm{p1} < 400;\ \mathrm{p2} < 300;\ \mathrm{p3} < 300;$

{Original Equations :
$\mu 6 * (\mathrm{SQUARE}(\delta 6) + \mathrm{p3} - 300) + \mu 5 * (\mathrm{SQUARE}(\delta 5) + \mathrm{p2} - 300) + \mu 4 *$
$(\mathrm{SQUARE}(\delta 4) + \mathrm{p1} - 400) + \mu 3 * (\mathrm{ploss} - (0.000\,03 * \mathrm{p1\^{}2} + 0.000\,09 * \mathrm{p2\^{}2} +$
$0.000\,12 * \mathrm{p3\^{}2})) + \mu 2 * (\mathrm{p1} + \mathrm{p2} + \mathrm{p3} * (850 + \mathrm{ploss})) + 1.1 * (510 + 7.2 * \mathrm{p1} +$
$0.001\,42 * \mathrm{p1\^{}2}) + 0.001\,94 * \mathrm{p2\^{}2} + 7.85 * \mathrm{p2} + 0.004\,82 * \mathrm{p3\^{}2} +$
$7.97 * \mathrm{p3} + 388;\}$
$\{\mathrm{d(eqn1)/d(p3)} =\} \mu 6 + \mu 3 * (0.000\,12 * \mathrm{p3} * 2) + \mu 2 + 0.004\,82 * \mathrm{p3} * 2 + 7.97;$
$\{\mathrm{d(eqn1)/d(p2)} =\} \mu 5 + \mu 3 * (0.000\,09 * \mathrm{p2} * 2) + \mu 2 + 0.001\,94 * \mathrm{p2} * 2 + 7.85;$
$\{\mathrm{d(eqn1)/d(p1)} =\} \mu 4 + \mu 3 * (0.000\,03 * \mathrm{p1} * 2) + \mu 2 + 1.1 * (7.2 + 0.001\,42 *$
$\mathrm{p1} * 2);$
$\{\mathrm{d(eqn1)/d(ploss)} =\} \mu 3 + (-\mu 2);$
$\{\mathrm{d(eqn1)/d}(\mu 2) =\} \mathrm{p1} + \mathrm{p2} + \mathrm{p3} - (850 + \mathrm{ploss});$
$\{\mathrm{d(eqn1)/d}(\mu 3) =\} \mathrm{ploss}(0.000\,03 * \mathrm{p1\^{}2} + 0.000\,09 * \mathrm{p2\^{}2} + 0.000\,12 * \mathrm{p3\^{}2});$
$\{\mathrm{d(eqn1)/d}(\delta 4) =\} \mu 4 * \delta 4 * 2;$
$\{\mathrm{d(eqn1)/d}(\mu 4) =\} \mathrm{SQUARE}(\delta 4) + \mathrm{p1} - 400;$
$\{\mathrm{d(eqn1)/d}(\delta 5) =\} \mu 5 * \delta 5 * 2;$
$\{\mathrm{d(eqn1)/d}(\mu 5) =\} \mathrm{SQUARE(\ddot{a}5)} + \mathrm{p2} - 300;$
$\{\mathrm{d(eqn1)/d}(\delta 6) =\} \mu 6 * \delta 6 * 2;$
$\{\mathrm{d(eqn1)/d}(\mu 6) =\} \mathrm{SQUARE}(\delta 6) + \mathrm{p3} - 300;$

$\mathrm{p1} = 334.288\,265\,5$	$\mathrm{p2} = 234.827\,132\,3$	$\mathrm{p3} = 300.000\,000\,0$
$\mathrm{ploss} = 19.115\,399\,7$	$\mu 2 = -9.147\,796\,6$	$\mu 3 = -9.147\,796\,6$
$\mu 4 = -1.19 * 10^{-9}$	$\mu 5 = 0.000\,000\,2$	$\mu 6 = -2.372\,844\,8$
$\delta 4 = 8.106\,277\,6$	$\delta 5 = 8.072\,972\,7$	$\delta 6 = 6.44 * 10^{-9}.$

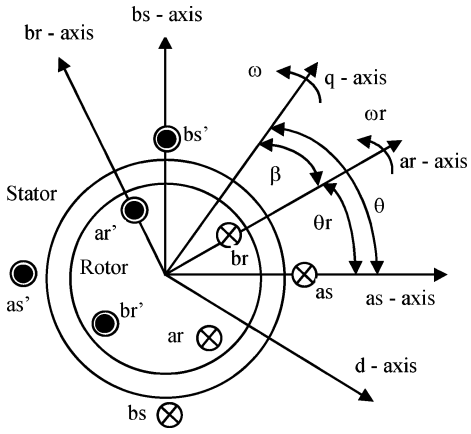

Fig. 6. Two-phase, two-pole induction motor.

Electrical equations for the ideal two-phase induction motor are as follows (transformed to d-q reference frame):

$$v_{qs}=r_s i_{qs}+\frac{d\lambda_{qs}}{dt}+\lambda_{ds}\frac{d\theta}{dt},\ v_{ds}=r_s i_{ds}+\frac{d\lambda_{ds}}{dt}+\lambda_{ds}\frac{d\theta}{dt}$$
$$v'_{qr}=r'_r i'_{qr}+\frac{d\lambda'_{qr}}{dt}+\lambda'_{dr}\frac{d\beta}{dt},\ v'_{dr}=r'_r i'_{dr}+\frac{d\lambda'_{dr}}{dt}+\lambda'_{qr}\frac{d\beta}{dt} \tag{20}$$

$$\begin{aligned}\lambda_{qs}&=L_{ls}i_{qs}+L_{ms}(i_{qs}+i'_{qr})\\ \lambda_{ds}&=L_{ls}i_{ds}+L_{ms}(i_{ds}+i'_{dr})\\ \lambda'_{qr}&=L'_{ls}i'_{qs}+L_{ms}(i_{qs}+i'_{qr})\\ \lambda'_{ds}&=L'_{ls}i'_{dr}+L_{ms}(i_{qs}+i'_{qr})\end{aligned} \tag{21}$$

$$L_{ls}=L_s-L_{ms},\ L'_{ls}=L'_r-L_{ms},\ L_{ms}=\frac{N_s}{N_r}L_{sr} \tag{22}$$

where L_s and L_r stand for the self-inductances of the stator and rotor circuits, L_{sr} represents the mutual inductance between stator and rotor, θ_r electrical angle, and θ_{mr} is mechanical displacement of the rotor. The mechanical equations for this machine are

$$\begin{aligned}T_e&=\frac{P}{2}L_{ms}(i_{qs}i'_{dr}-i_{ds}i'_{qr})\\ T_e&=\frac{2}{P}J\frac{d\omega_r}{dt}+\frac{2}{P}D\omega_r+T_L\end{aligned} \tag{23}$$

where T_e is the electromagnetic torque, J is the inertia, D is the damping, and T_L is the load torque (considered zero in our simulations). In the simulations carried out, it is considered that the machine is connected to an infinite bus (an ideal voltage source), and the rotor is assumed to be shortcircuited

$$\begin{aligned}v_{as}&=\sqrt{2}V\mathrm{RMS}_{as}\cos(\omega_e t)\\ v_{bs}&=\sqrt{2}V\mathrm{RMS}_{bs}\sin(\omega_e t)\\ \omega_e&=2\cdot\pi\cdot f\end{aligned} \tag{24}$$

$$v_{ar}=0;\ v_{br}=0. \tag{25}$$

The complete equations for time-domain simulation, as they appear in the equation editor, are shown in the equation at the bottom of the next page.

These equations are in declarative style and are identical to (20)–(25), with two exceptions [4]. First, the equations have been set up to automatically compute initial values for all variables by initially setting time to a negative value. Second, the fluxes have been scaled by $(2*\pi*f)$. These equations are transformed to symbolic algebraic form by using the trapezoidal rule of integration. The resulting algebraic equations are then solved numerically. The case considered here is a cold start-up with balanced phase voltages ($V\mathrm{RMS}_{as}=V\mathrm{RMS}_{bs}=110$ V). Results are depicted in Fig. 7 (torque), Fig. 8 (speed), and Fig. 9 (torque versus speed).

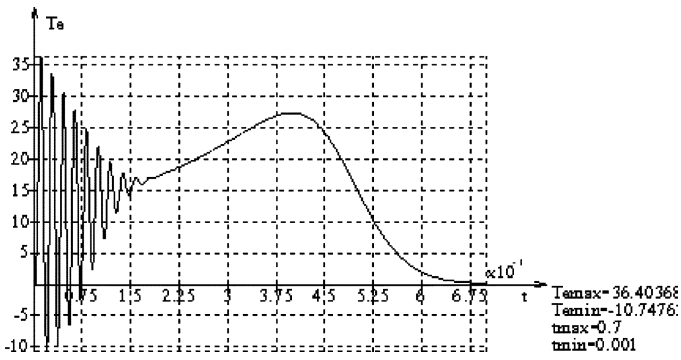

Fig. 7. Torque characteristic.

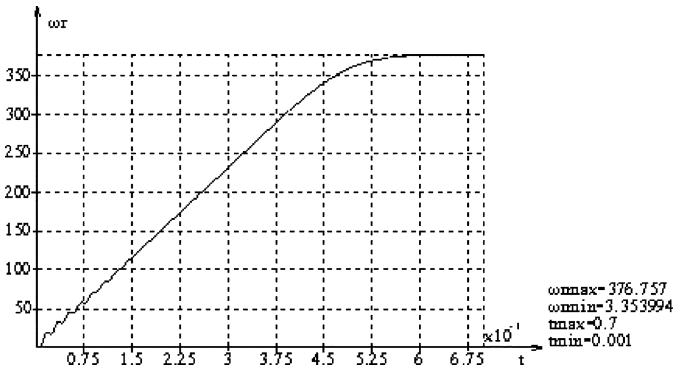

Fig. 8. Speed characteristic.

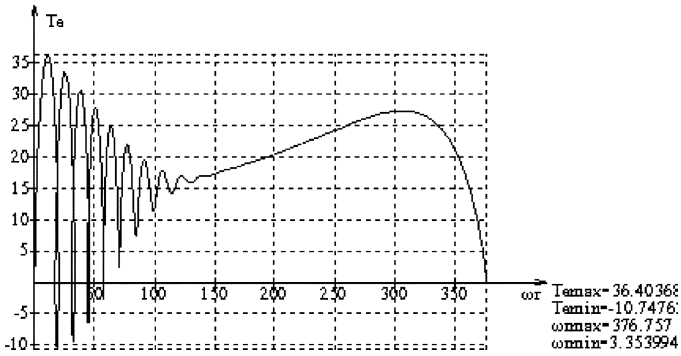

Fig. 9. Torque versus speed.

The torque and speed characteristics are what can be expected from a balanced start-up of an induction motor, with relatively large oscillations of the torque at the beginning and a peak torque before the machine reaches steady state.

Observe that the environment includes limited internal syntax in the form of IF-THEN-ELSE rules that additionally strengthen the environment's flexibility.

D. Elementary Feedback Control Example

Consider a minimal feedback system with actuator A and controller C, as illustrated in Fig. 10.

The mathematical description of the system is as follows:

$$A : \dot{y} = g \cdot u \tag{26}$$

$$C : \dot{x} = -k_1 \cdot y - k_2 \cdot x \tag{27}$$

$$u = r + x. \tag{28}$$

C is a lowpass filter with internal state x and parameters $k_1 > 0$ and $k_2 > 0$. A is a pure integrator with state y and gain $g > 0$. This does not model any particular system but is a simple, generic feedback example that is similar to what might arise in a variety of settings in power engineering.

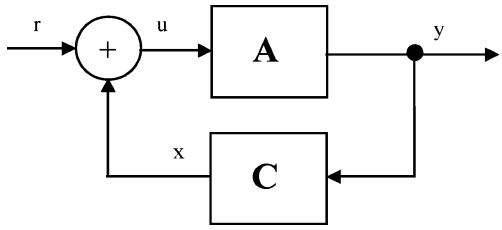

Fig. 10. Minimal feedback system.

{Applied voltage transformation}
$\omega 0 = \omega r; d\omega = d\omega r;$
$\alpha = \alpha r;$ {Reference frame fixed to the Rotor}
{Park's equations}
$vqsp = vas * COS(\alpha) + vbs * SIN(\alpha);$
$vdsp = vas * SIN(\alpha) - vbs * COS(\alpha);$
{Phase stator voltage}
$vas = SQRT(2) * 110 * COS(\omega e * t);$
$vbs = SQRT(2) * 110 * SIN(\omega e * t);$
$\omega e = 2 * 3.14 * f; f = 60;$
$vqr' = 0; vdr' = 0;$ {Short-circuited rotor}
{Connect the machine to the power supply at $t = 0$}
$a = 0; trash = 0;$
$vqs = IF\ t > trash\ THEN\ vqsp\ ELSE\ a;$
$vds = IF\ t > trash\ THEN\ vdsp\ ELSE\ a;$
{d-q machine equations, linkage fluxes have been scaled}
$vqs = Rs * iqs + D(\gamma qs, t)/\omega e + (\omega 0 + d\omega) * \gamma ds/\omega e;$
$vds = Rs * ids + D(\gamma ds, t)/\omega e + (\omega 0 + d\omega) * \gamma qs/\omega e;$
$vqr' = Rr' * iqr' + D(\gamma qr', t)/\omega e + (\omega 0 - \omega r0 + d\omega - d\omega r) * \gamma dr'/\omega e;$
$vdr' = Rr' * idr' + D(\gamma dr', t)/\omega e - (\omega 0 - \omega r0 + d\omega - d\omega r) * \gamma qr'/\omega e;$
$\gamma qs = X1s * iqs + Xms * (iqs + iqr');$
$\gamma ds = X1s * ids + Xms * (ids + idr');$
$\gamma qr' = X1r' * iqr' + Xms * (iqs + iqr');$
$\gamma dr' = X1r' * idr' + Xms * (ids + idr');$
{Electromagnetic torque}
$Te = P/2 * Lms * (iqs * idr' - ids * iqr');$
{Mechanical equations}
$D(\alpha r, t) = \omega r0 + d\omega r;$
$\omega r = D(\alpha r, t);$
$Te = (2/P) * J * D(d\omega r, t) + T1; T1 = 0;$
{Machine data}
{R's in Ohms, L's in H, J in kg-m2}
$Rs = 0.295; Rr' = 0.144; Lms = 0.035\,14;$
$L1s = 0.001\,33; L1r' = 0.000\,554; J = 0.026;$
$X1s = \omega e * L1s; X1r' = \omega e * L1r'; Xms = \omega e * Lms; P = 2;$
{Initial conditions for cold start-up}
$\omega r0 = 0; t \approx -0.04;$ {Connection at $t = 0$}
$d\omega r \approx 0; \alpha r \approx 0; \gamma ds \approx 0; \gamma qs \approx 0; \gamma dr' \approx 0; \gamma qr' \approx 0;$

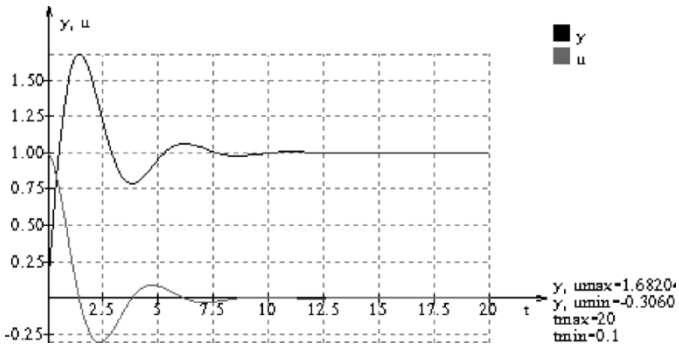

Fig. 11. System response (y and u) for $g = 2.0$.

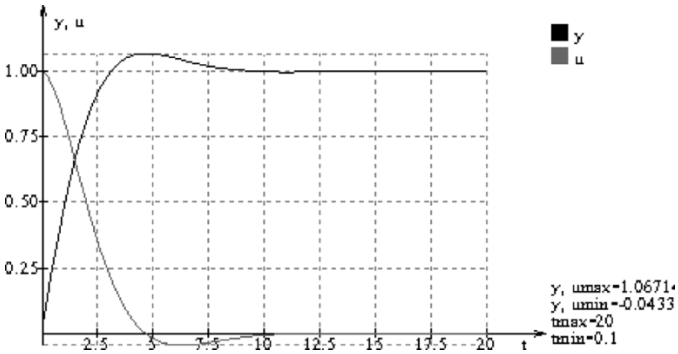

Fig. 12. System response (y and u) for $g = 0.5$.

The goal is to simulate the response of y to a step change in reference signal r and to analyze the influence of different parameter values on the response. The system equations are

$$
\begin{aligned}
&\mathrm{D(x,t) = -k1 * y - k2 * x;}\\
&\mathrm{D(y,t) = g * u;}\\
&\mathrm{u = r + x;}\\
&\mathrm{g = 1;}\\
&\mathrm{k1 = 1;}\\
&\mathrm{k2 = 10;}\\
&\mathrm{trash = 0;}\\
&\mathrm{b = 1;}\\
&\mathrm{c = 0;}\\
&\mathrm{r = IF\ t > trash\ THEN\ b\ ELSE\ c;}\\
&\mathrm{t \approx 0;}
\end{aligned}
$$

The system responses are plotted for fixed values of parameters $k_1 = 1.$, $k_2 = 10.$, and two values of the gain g: $g = 2$ (Fig. 11) and $g = 0.5$ (Fig. 12).

Lower overshoot and settling time are obtained with the gain value of 0.5. The system response with any other values of the parameters can be easily obtained due to the high flexibility that the environment offers for experimenting.

V. Conclusions

The presented general-purpose symbolically assisted numeric computation environment can be used directly as an educational and research tool. This paper has illustrated the implementation and application of the concept to three power engineering computational problems and an elementary feedback control design, ranging from the problems described by a set of simultaneous algebraic equations to implicit time-domain simulation.

Similar applications to other areas of engineering are possible since no matter which engineering problem should be solved, one always ends up with a similar formal description of the problem.

The payoff of the approach is an order-of-magnitude reduction in the effort to develop sophisticated engineering software of great clarity and reliability, at the expense of about an order-of-magnitude decrease in overall computational speed. Work is underway to reduce the present penalties in computational speed and to further enhance the power of this high-level engineering environment through combining the computational environment with a component modeling paradigm and automatic object-oriented code generation [14], [15] as a final step in experimenting with a defined problem.

The computation environment is available for interested educators, instructors, and students without any charge [16].

Acknowledgment

Prof. F. L. Alvarado, University of Wisconsin–Madison, made available the first version of the Solver-Q software to the authors and actively participated in the development of the computation environment presented in this paper. His permanent support and encouragement are greatly appreciated.

References

[1] F. L. Alvarado and Y. Liu, "General purpose symbolic simulation tools for electric networks," *IEEE Trans. Power Syst.*, vol. 3, no. 2, pp. 689–697, May 1988.

[2] F. L. Alvarado, R. H. Lasseter, and Y. Liu, "An integrated engineering simulation environment," *IEEE Trans. Power Syst.*, vol. 3, no. 1, pp. 245–253, Feb. 1988.

[3] F. Alvarado, *Solver-Q, Equation Handler, Version 1.90, User's Manual*: Univ. Wisconsin, ECE Dept., 1993.

[4] F. L. Alvarado, C. A. Canizares, A. Keyhani, and B. Coates, "Instructional use of declarative languages for the study of machine transients," *IEEE Trans. Power Syst.*, vol. 6, no. 1, pp. 407–413, Feb. 1991.

[5] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation and Control*. New York: Wiley, 1996.

[6] J. Doyle, B. Francis, and A. Tannenbaum, *Feedback Control Theory*. New York: Macmillan, 1990.

[7] R. Bacher, "Symbolically assisted numeric computations for power system software development," in *Proc. 13th Power Syst. Comp. Conf.*, Trondheim, Norway, 1999, pp. 5–16.

[8] K. R. Coombes, B. R. Hunt, R. L. Lipsman, G. J. Stuck, and J. E. Osborn, *Mathematica Primer*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[9] B. Char, *Maple V. Library Reference Manual*. New York: Springer-Verlag, 1991.

[10] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 5th ed. London, U.K.: Oxford Univ. Press, 2003.

[11] P. Kundur, *Power System Stability and Control*. New York: McGraw-Hill, 1994.

[12] C. Canizares and F. Alvarado, "Point of collapse and continuation methods," *IEEE Trans. Power Syst.*, vol. 8, no. 1, pp. 1–8, Feb. 1993.

[13] C. Nwankpa, H. G. Kwatny, X. M. Yu, and C. Houlian, "Symbolic/numeric computation and voltage stability analysis," in *Proc. 12th Power Syst. Comp. Conf.*, Dresden, Germany, 1996, pp. 323–330.

[14] I. Dzafic, M. Glavic, and S. Tesnjak, "A general purpose symbolically assisted numeric computation environment for engineering education and research," in *Proc. 8th Rhine Workshop Comput. Algebra*, Mannheim, Germany, 2002, pp. 279–284.

[15] I. Dzafic, F. Alvarado, M. Glavic, and S. Tesnjak, "A component based approach to power system applications development," in *Proc. 14th Power Syst. Comp. Conf.*, Sevilla, Spain, Jun. 2002, Paper 33–1.

[16] I. Dzafic, M. Glavic, and F. Alvarado. SolverQ 2.5.1. [Online]. Available: http://www.simtel.com/product.php[id]79848[sekid]0[SiteID] simtel.net

Mevludin Glavic (M'04) received the M.Sc. and Ph.D. degrees from the University of Belgrade, Belgrade, Yugoslavia, and the University of Tuzla, Tuzla, Bosnia, respectively.

As a post-doctoral researcher within the Fulbright Program, he spent the 1999–2000 academic year at the University of Wisconsin–Madison. Currently, he is Research Fellow in the Department of Electrical Engineering and Computer Science, University of Liège, Liège, Belgium. His teaching and research fields of interest include power system control and optimization, systems theory, modeling, and simulation.

Izudin Dzafic received the M. Sc. and Ph.D. degrees from the University of Tuzla, Tuzla, Bosnia, and the University of Zagreb, Zagreb, Croatia, in 1999 and 2002, respectively.

Currently, he is with Siemens AG, Power Transmission and Distribution, Nuremberg, Germany. His research interests lie in systems modeling and simulation and applying advances in software engineering to power systems simulations.

Sejid Tesnjak is a Professor of electrical engineering with the Department of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia. His teaching and research interests include power system dynamics, control, and simulation.