ELSEVIER

Computers and Chemical Engineering 26 (2002) 547–554

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

A web-based library for testing performance of numerical software for solving nonlinear algebraic equations

Mordechai Shacham [a,*], Neima Brauner [b], Michael B. Cutlip [c]

[a] *Department of Chemical Engineering, Ben Gurion University of the Negev, Beer-Sheva 84105, Israel*
[b] *School of Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel*
[c] *Department of Chemical Engineering, University of Connecticut, Storrs, CT 06269, USA*

Abstract

An extensive test problem library of nonlinear algebraic equations (NLE) has been created and implemented on the World Wide Web. This web-based test problem library contains complete problem descriptions with the following information: the model equations in the same form as the input for the numerical solver, the explicit definitions of constraints on the variables, the initial estimates and function values at the initial estimates, and the variable values and function values at the solution. All variables and function values are reported with the full precision of the numerical solution. This library improves upon existing test problem collections regarding the type of information included and the form in which this information is stored and presented. The NLE library can be accessed through the Web site: http://www.polymath-software.com/library. It contains presently over 70 problems of various dimensions on lower, average, and higher difficulty levels. Some of the problems exhibit multiple solutions in the feasible and/or the infeasible subspaces, discontinuities, and false solutions that are identified as true solutions by most numerical solvers. This library can be of significant benefit to users and developers of NLE solvers for verifying the robustness and reliability of NLE solvers. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Nonlinear algebraic equations; Root finding; Test problem library

1. Introduction

Steady state and dynamic simulations of chemical processes require numerical solution of large systems of nonlinear algebraic, ordinary differential, and differential-algebraic equations. It is often assumed that the numerical software packages and their solution algorithms will find the correct solution for a given problem if it exists or that warning messages will be issued when the accuracy and/or correctness of the solution is in doubt. Unfortunately, the solution provided by numerical solvers cannot always be trusted (see for example, Brauner, Shacham, & Cutlip, 1996; Shacham, 1984 and Shacham, Brauner, & Pozin, 1998). Reliability testing of software requires problem solutions for a large set of benchmark problems that are unusually challenging.

The need to validate software is common to both the numerical and statistical software. For statistical software, a group of statisticians from NIST, National Institute of Standards and Technology of the USA, took the initiative and placed a large set of test problems on the Web.[1] This library contains problems related to: analysis of variance, linear and nonlinear regression, and univariate summary statistics. In the case of nonlinear regression, for example, the data set contains the data points, the correlation model equations, and 'certified' values of the calculated parameters including their standard deviation and the resultant variance. The problems are of various size and difficulty level and can be downloaded as ASCII files for testing software packages.

* Corresponding author. Tel.: +972-8-646-1481; fax: +972-8-647-2916.
E-mail address: shacham@bgumail.bgu.ac.il (M. Shacham).

[1] http://www.itl.nist.gov/div898/strd/index.html.

0098-1354/02/$ - see front matter © 2002 Elsevier Science Ltd. All rights reserved.
PII: S0098-1354(01)00790-6

Table 1
Hiebert's (1983) version of the 'combustion of propane in air' test problem

No.	Equations	Variable	$\mathbf{x}_0$	$\mathbf{x}^{*a}$	$\mathbf{f}(\mathbf{x}^{*a})$	$\mathbf{x}^{*b}$	$f(\mathbf{x}^{*b})$
1	f1 = x1 + x4 − 3 = 0	x1	2	2.995	0	2.9976	−3.55E-05
2	f2 = 2*x1 + x2 + x4 + x7 + x8 + x9 + 2*x10 − R = 0	x2	5	3.967	2.00E-04	3.9664	3.37E-04
3	f3 = 2*x2 + 2*x5 + x6 + x7 − 8 = 0	x3	40	79.999	1.72E-04	80	−5.34E-05
4	f4 = 2*x3 + x5 − 4*R = 0	x4	1	0.005	−9.72E-04	0.0023645	6.04E-04
5	f5 = x1*x5 − 0.193*x2*x4 = 0	x5	0	0.001028	−7.50E-04	6.04E-04	−1.99E-08
6	f6 = x6*sqrt(x2) − 0.002597*sqrt(x2*x4*xs) = 0	x6	0	0.001916	7.90E-07	0.0013659	−1.00E-07
7	f7 = x7*sqrt(x4) − 0.003448*sqrt(x1*x4*xs) = 0	x7	0	0.0622	−3.28E-06	0.064573	2.57E-08
8	f8 = x8*x4 − 1.799e − 5*x2*xs = 0	x8	0	1.553	−8.80E-07	3.5308	−3.59E-08
9	f9 = x9*x4 − 0.0002155*x1*sqrt(x3*xs) = 0	x9	0	12.043	−4.21E-06	26.432	1.45E-06
10	f10 = x10*x4ˆ2 − 3.846e − 5*xs*x4 ^ 2 = 0	x10	5	8.19	2.05E-04	0.0044998	-1.56E-13
11	R = 40						
12	xs = x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10			108.81714		117.00021	

[a] Solution obtained by Bullard and Biegler (1991).

[b] Solution obtained by Shacham (1984).

We have developed a similar test problem library for systems of nonlinear algebraic equations, NLEs. This library is based on collections of test problems published in the literature (see for example Bullard & Biegler, 1991; Hiebert, 1983; Shacham, 1984, 1990; Shacham et al., 1998). Additional problems have been obtained by personal communications with individuals who needed assistance in solving particular types of equations. During the preparation of this library, attempts to reproduce the results obtained from various sources in the literature revealed many of the limitations of these test problem collections. These limitations helped to identify and refine the type of information that should be included in the library including: the problem definition, the desired form in which this information should be stored and displayed, and the general structure of the library.

The computations related to this article were carried out with the NLE solver program of the polymath 5.0 package, [2] using double precision computation, on a PC. Three NLE solver algorithms were used: Shacham's (1986) *CONLES* constrained algorithm and the subroutines *mnewt* and *newt* from the Numerical Recipes' book (Press, Teukolsky, Vetterling, & Flannery, 1992). The subroutine *mnewt* implements the classical Newton–Raphson (NR) method. The *newt* algorithm employs the NR method with a line search. The *CONLES* algorithm combines the step-length restricted NR method with the Levenberg–Marquard algorithm, when the Jacobian matrix happens to be singular, and with a continuation approach, if no feasible solution is reached, in order to keep the variables inside the feasible subspace and eventually reach a solution in this subspace.

2. Limitations of existing test problem collections

The limitations of the existing test problem collections can be demonstrated with reference to the problem of 'combustion of propane in air', which was used as a test problem, for example, by Bullard and Biegler (1991), Hiebert (1983), Meintjes and Morgan (1990) and Shacham (1984). The equations as presented by Hiebert (1983) are shown in Table 1. It should be pointed out from the outset, that Meintjes and Morgan (1990) had found that this model is chemically incorrect and does not represent a physical system. This illustrates a very important point that many of the published test problems contain typographical errors. The only way to avoid such errors in the library version is to use electronic transfer of the problem's equations from the numerical solver to the library and vice versa. Following this principle, the equations in Table 1 are shown in the form as they were copied from the input data set of the numerical solver.

This set of equations is very difficult to solve because the system may have several solutions, and some of them are physically infeasible (x_i represents number of moles of various components, thus all x_i must be nonnegative). This illustrates that *constraints are an integral part of the mathematical model* and, as such, they must be explicitly displayed in the library. In this particular example, the solution algorithm is challenged by the need to calculate the square root of some of the variables, which may attain negative values along the solution path, especially if the solution itself is very close to zero.

Bullard and Biegler (1991), Hiebert (1983) and Shacham (1984) solved the set of equations of Table 1 for several different values of the parameter R and from several different starting points. Some of the data and the results for R = 40 and one set of initial estimates ($\mathbf{x}_0$) are also shown in Table 1. The initial

[2] POLYMATH is copyrighted by M. Shacham, M.B. Cutlip and M. Elly (http://www.polymath-software.com/).

estimates have been reported in those studies; however, the corresponding function values were not reported. The information regarding $\mathbf{f}(\mathbf{x}_0)$ is essential for reproducing the results since differences in function values at the initial estimate signal errors in the problem set-up. Such errors can be detected irrespective as to whether convergence to a solution is achieved by the software that is being tested. The function values at the initial estimates can also provide some information regarding the order of magnitude of the various terms comprising a particular function. The order of magnitude of the various terms is important for determining the attainable accuracy, as indicated by the function value at the solution point. The information that can be derived from $\mathbf{f}(\mathbf{x}_0)$ is easier to interpret if different initial estimates are used for the different variables. Thus, use of an initial estimate of zero value for five of the ten variables, as shown in Table 1, can hide some important information that can be deduced from $\mathbf{f}(\mathbf{x}_0)$.

Hiebert (1983) attempted to solve this system of equations using nine different software packages and reported the relative performance of the various packages. The values of the variables at the solution were not reported. This makes the reproduction of the results rather difficult and uncertain, as the user can never be sure whether the same solution is found or even whether the same problem is solved.

Bullard and Biegler (1991) found two solutions to this system using an iterative linear programming strategy. Their first solution is shown in the column marked with $\mathbf{x}^{*a}$ in Table 1. Shacham (1984) found one solution to this problem using the *CONLES* algorithm. The latter is shown in the column marked with $\mathbf{x}^{*b}$. The three solutions are completely different. For example, $x_{10} = 8.19$ in $\mathbf{x}^{*a}$, $x_{10} = 0.0044998$ in $\mathbf{x}^{*b}$, whereas $x_{10} = 6.465$ in an additional solution reported by Bullard and Biegler (1991) (not shown in Table 1).

The existence of three different solutions to this problem raises several considerations: (1) the problem has multiple solutions and all of the solutions are valid; (2) there are errors in the problem set-up in one or more of the cases; (3) the solution method converged to a local minimum instead of the zero of the system of equations. In order to find out which of the reported solutions are valid solutions of the problem, the function values should be checked. Neither Bullard and Biegler (1991) nor Shacham (1984) reported function values at the solution. Introducing the values shown under $\mathbf{x}^{*a}$ and $\mathbf{x}^{*b}$ into the equations yield the function values shown in Table 1 (columns marked with $\mathbf{f}(\mathbf{x}^{*a})$ and $\mathbf{f}(\mathbf{x}^{*b})$). It can be seen that in both cases the largest function values are of the order of 10^{-4}, a number that can be considered as a nonzero value.

One reason for the large function values obtained at the solution point is the low precision used for reporting the x_i^* values. Bullard and Biegler (1991) reported $\mathbf{x}^*$ with a precision of four decimal digits in most cases. For numbers greater than one (in absolute value), it is understandable that even in a linear equation, the function value can exceed 10^{-4} due to error introduced by rounding the numbers to four digits. Thus, in order to verify that $\mathbf{x}^*$ is indeed a solution of the system of equations, the results should be reported with a much higher precision, preferably with the working precision of the computer program; most NLE solver programs work with double precision, approximately 15 significant decimal digits.

In this particular case, the solution provided by Shacham (1984) has been verified utilizing the *CONLES* algorithm, which converged to the same solution as shown in the column $\mathbf{x}^{*b}$ of Table 1 with 15 decimal digits of accuracy. The corresponding highest function value is of the order of 10^{-14}, which can be considered safely as zero in double precision computation.

The solutions reported by Bullard and Biegler (1991) are suspected to be incorrect. This can be seen, for example, by considering the values of the two terms comprising f_{10} at the reported solution. The first (positive) term value is: $8.19*0.005^2 = 2.05*10^{-4}$. The value of the second (negative) term is $-3.846*10^{-5}*108.817*0.005^2 = -1.046*10^{-7}$. Thus, the function value is equal to the value of the positive term, the negative term is insignificant, hence this solution cannot be considered a zero of f_{10}.

This example illustrates the structure and the information to be included in a test problem library:

1. The model equations must be stored in the same form as the input for the numerical solver, to prevent introduction of typographical and other errors.
2. Constraints on the variables are an integral part of the mathematical model and should be included explicitly in the library.
3. It is preferable to use different values as initial estimates for the different variables, and function values at the initial estimate must be reported in order to enable detection of errors in the problem set-up.
4. The variable values at the solution must be reported with the same precision that the solution was obtained (15 significant decimal digits for double precision).
5. Function values at the solution must be included. In case the function values are suspected to be too high, the order of magnitude of the various terms comprising the function must be examined to validate that the reported solution is a zero of the function.

3. Implementing the preferred structure for the 'combustion of propane in air' example

Many of the difficulties associated with the example presented in Section 2 could have been detected easily if the physical nature of the model and the various variables were known. Although the inclusion of the description of the physical basis of the model in the library is not essential, it can be rather helpful in many cases. If the description is simple, it can be included as part of the model (as 'comments') otherwise reference to the pertinent literature source should be provided.

Table 2
Meintjes and Morgan (1990) version of the 'combustion of propane in air' test problem

No.	Equations[a]	Constraints[b]
1	f(n1) = n1 + n4 − 3 # Mol of Carbon Dioxide–Carbon Balance	> = 0 (a)
2	f(n2) = 2*n1 + n2 + n4 + n7 + n8 + n9 + 2*n10 − R # Mol of Water-Oxygen Balance	> = 0 (a)
3	f(n3) = 2*n2 + 2*n5 + n6 + n7 − 8 # Mol of Nitrogen-Hydrogen Balance	> = 0 (a)
4	f(n4) = 2*n3 + n9 − 4*R # Mol of Carbon Monoxide–Nitrogen Balance	> = 0 (a)
5	f(n5) = K5*n2*n4 − n1*n5 # Mol of Hydrogen-Equilibrium Expression	> = 0
6	f(n6) = K6*sqrt(n2*n4) − sqrt(n1)*n6*sqrt-(p/nt) # Hydrogen atom-Equilibrium Expression	> = 0
7	f(n7) = K7*sqrt(n1*n2) − sqrt(n4)*n7*sqrt-(p/nt) # Hydroxyl Radical-Equilibrium Expression	> = 0
8	f(n8) = K8*n1 − n4*n8*(p/nt) # Oxygen Atom-Equilibrium Expression	> = 0
9	f(n9) = K9*n1*sqrt(n3) − n4*n9*sqrt(p/nt) # Mol Nitric Oxide-Equilibrium Expression	> = 0
10	f(n10) = K10*n1^2 − n4^2*n10*(p/nt) # Mol of Oxygen-Equilibrium Expression	> = 0
11	nt = n1 + n2 + n3 + n4 + n5 + n6 + n7 + n8 + n9 + n10 # Total Number of Moles of Combustion Products	
12	K5 = 0.193 # Equilibrium Constant at 2200 K	
13	K6 = 2.597e-3 # Equilibrium Constant at 2200 K	
14	K7 = 3.448e-3 # Equilibrium Constant at 2200 K	
15	K8 = 1.799e-5 # Equilibrium Constant at 2200 K	
16	K9 = 2.155e-4 # Equilibrium Constant at 2200 K	
17	K10 = 3.846e-5 # Equilibrium Constant at 2200 K	
18	R = 10 # Air to Fuel Ratio	
19	p = 40 # Pressure (atm.)	

[a] An implicit equation is indicated by f( ). Output variables assigned arbitrarily for implicit equations.

[b] Constraint on the output variable. An (a) indicates that the constraint must be always satisfied.

Meintjes and Morgan (1990) have traced the problem 'Combustion of Propane in Air' back to a paper from 1943 by Damkohler and Edse (1943). The stoichiometric equation of the reaction is $C_3H_8 + 0.5R(O_2 + 4N_2) \rightarrow$ Products, where R is a parameter expressing the relative amounts of air and fuel. The list of products is shown in Table 2. Given this list of species, R must be greater than three for a physically feasible solution to exist; if $R < 10$, the combustion is 'fuel rich', if $R > 10$ it is 'lean'.

Table 2 shows the equations (as provided by Meintjes & Morgan, 1990) in the form they are stored in the test problem library (after slight rearrangement for a more concise presentation). The unknowns n_1, n_2,…,n_{10} represent the number of moles of product i formed per mole of propane consumed. An output variable, which appears on the left-hand side, is designated to each equation. In explicit equations, the output variable is assigned according to the calculation order, whereas in implicit equations, the assignment is arbitrary in order to allow the presentation of the variables, function definitions and values in a concise and compact format. The description of the variables and the equations are included as comments, separated from the equation by the # sign. Constraints on the variables are an integral part of the problem definition and appear in a separate column of Table 2. All the unknowns represent moles of product formed, thus they all must be non-negative at the solution point. A constraint that must be satisfied all the way to the solution is marked by (a). This is necessary in order to avoid a negative value inside the square root function, in this particular example.

The introduction to the problem, as presented in the previous paragraph, and the data in Table 2 represent a complete definition of the problem. This includes the mathematical model and the physical basis. The equations as they appear in the second column of Table 2 can be copied directly into the polymath 5.0 program for solution. If other programs are used for solution (such as matlab[3] or mathematica[4]), some modifications may be required. The required editing can be easily performed with a text editor.

The initial estimates, function values at the initial estimates, the solution and the function values at the solution are shown in Table 3. It can be seen that when 15 decimal digits are used for n*, the largest absolute value of f(n*) obtained is of the order of 10^{-14}, raising

[3] MATLAB is a trademark of The Math Works, Inc. (http://www.mathworks.com).

[4] Mathematica is a trademark of Wolfram Research, Inc. (http://www.wolfram.com).

Table 3
Initial estimates and solution for the ‘combustion of propane in air’ test problem

Function and variable number	Initial value	$\mathbf{f^0}$	**n***	**f(n*)**
1	n1(0) = 1.5	−1	2.915725423895220	−3.11E-15
2	n2(0) = 2	5.563	3.960942810808880	−7.11E-15
3	n3(0) = 35	−3.855	19.986291646551500	3.55E-15
4	n4(0) = 0.5	30.02	0.084274576104777	−8.53E-14
5	n5(0) = 0.05	0.118	0.022095601769893	1.94E-15
6	n6(0) = 0.005	−0.0032339	0.000722766590884	3.61E-16
7	n7(0) = 0.04	−0.0209598	0.033200408251574	1.16E-16
8	n8(0) = 0.003	−0.0013330	0.000421099693392	−2.98E-17
9	n9(0) = 0.02	−0.0076095	0.027416706896918	−3.25E-17
10	n10(0) = 5	−1.1332377	0.031146775227006	−7.59E-19
nt	44.118		27.062238	

no doubt with regard to the solution validity. All mole numbers are positive and their values make physical sense. This is an indication that the model is correct in contrast to the formulation in Table 1, where the nitrogen balance, for example, is grossly off because of an error in equation (4).

4. Structure of a problem page in the library

The structure of the library that has evolved from the principles presented in the previous sections is demonstrated with reference to another example involving nonideal, isothermal flash calculation for a binary mixture of isobutanol and water. The problem definition section of this example is shown in Table 4.

Each problem is assigned with an identification number, which is associated with the number of implicit equations included in the problem (e.g. *Threeq2* marks the second problem including three implicit equations). The source and pertinent references are provided, where more detailed information regarding the physical nature of the problem can be found.

The number of implicit equations and the set of independent variables associated with these equations are listed. Most sets of equations can be revised easily to include more or less implicit equations. In fact, some of the problems in the library represent different formulations of the same problem, where the number of implicit equations has been changed to alter its difficulty level.

The *difficulty level* of the problem is classified as low, average, or high. The difficulty level is assigned according to the performance of the subroutines *newt*, *mnewt* and *CONLES* starting from several initial estimates that are close and far from the solution. If all the programs converge from all the starting points, the problem is considered to be of lower difficulty level. If for some initial estimates none of the programs converge, or only the constrained method, that keeps the variable values inside the feasible range throughout the iterations converges, the problem is considered to be of the high difficulty level.

Table 4
Single problem definition in the library

Name	Threeq2—Two phase flash of a nonideal binary mixture (isobutanol–water)
Source	Henley and Rosen (1969)
Reference(s)	Cutlip and Shacham (1999)
Model	Three implicit equations, independent variables alpha, x1 and x2 Higher difficulty level Constraints: 0< = alpha< = 1, 0< = x1< = 1, 0< = x2< = 1 Discontinuities: Undefined for (x1 + B*x2/A) = 0 and (A*x1/B + x2) = 0 Initial estimates: 1. alpha = 0.5, x1 = 0, x2 = 1, 2.alpha = 0.9, x1 = 0.5, x2 = 0.5; 3. alpha = 0.9, x1 = 0.4, x2 = 0.6 4. alpha = 0.5, x1 = 0.1, x2 = 0.9
Solved by	Shacham, M., polymath 5.1, build 230, 4.4.01

Model equations	Two phase flash of a nonideal binary mixture (isobutanol-water)|polver05_0
	f(x1) = x1 − z1/(1 + alpha*(k1 − 1)) #
	f(x2) = x2 − z2/(1 + alpha*(k2 − 1)) #
	f(alpha) = x1 + x2 − (y1 + y2) #
	p1 = 10 ^ (7.62231 − 1417.9/(191.15 + t)) #
	p2 = 10 ^ (8.10765 − 1750.29/(235 + t)) #
	gamma2 = 10 ^ (B*x1*x1/((x1 + B*x2/A)ˆ2)) #
	gamma1 = 10 ^ (A*x2*x2/((A*x1/B + x2)ˆ2)) #
	k1 = gamma1*p1/760 #
	k2 = gamma2*p2/760 #
	y1 = k1*x1 #
	y2 = k2*x2 #
	t = 88.538 #
	B = 0.7 #
	A = 1.7 #
	z1 = 0.2 #
	z2 = 0.8 #
	x1(0) = 0
	x2(0) = 1
	alpha(0) = 0.5

Table 5
Function and variable values at the initial estimate and at the solution

Variable	Initial value	Solution value
x1	0	0.0226974766367
f(x1)	−1.6296E-02	−3.4694E-18
x2	1	0.9773025233633
f(x2)	3.3895E-02	0.0
alpha	0.5	0.5322677863643
f(alpha)	3.4387E-01	0.0
p1	357.053	357.05282269240
gamma2	1.000	1.0046052345120
p2	498.662	498.66206831320
gamma1	50.119	33.36690033259
k2	0.656	0.6591559527368
y2	0.656	0.6441947758996
k1	23.546	15.67598151085
y1	0.00	0.3558052241004
t	88.538	
B	0.70	
A	1.70	
z1	0.20	
z2	0.80	

Table 6
An infeasible and a false solution for the isothermal flash problem

Variable	Infeasible solution value	False solution value
x1	0.6867568052506	−7.979064911E-13
f(x1)	−1.3212E-14	1.1309E-13
x2	0.3132431762583	2.545161363E-10
f(x2)	−1.1102E-16	3.8950E-14
alpha	1.4708209249600	−9.143888094E+09
f(alpha)	−5.55E-16	1.07E-10
p1	357.05282269240	357.0528226924
gamma2	3.1342710428540	1.000094874128
p2	498.66206831320	498.6620683132
gamma1	1.10281288233	53.23346661201
k2	2.0565027380050	0.6561965505819
y2	0.6441854496365	1.670126107E-10
k1	0.51810849018	25.00942041516
y1	0.3558145314887	−1.995517889E-11

The listed *constraints* are of physical nature. In the isothermal flash example, all the independent variables are constrained. The ratio of moles of vapor to total number of moles (alpha) is bounded in the range: 0 < = alpha < = 1. The same bounds apply to x1 and x2, which represent the mole fraction of the first and second component, respectively.

Certain combinations of x1 and x2 with opposite signs may yield discontinuities in the equations for gamma1 and gamma2 (when x1 + B*x2/A = 0 or A*x1/B + x2 = 0). However, since the discontinuities are outside the feasible subspace, they are not of a major concern. In some of the problems included in the library, however, discontinuities very close to the solution render the problem unsolvable practically.

Several sets of *initial estimates* are provided. In cases where multiple solutions exist, a different set of initial estimates will lead usually to a different solution. In problems with higher difficulty level, some sets of initial estimates are close to the solution, thus convergence can be achieved. For other sets that are far from the solution, most methods fail to converge.

The *model equations* are presented in a format so that they can be 'copied' and then 'pasted' directly into the polymath program. Some formatting requirements of polymath, related to output variables and comments, are described in Section 3. Polymath reorders the equations according to the calculation sequence, thus the order in which the equations are presented is unimportant. Some other programs (matlab, for example) will usually require reordering the equations according to the calculation sequence, before imputing it into the program. The initial estimates for the variables appear in the bottom lines of the equation set and are indicated by the (0) index.

Variable and function values at the *initial estimate* and at a *feasible solution* are shown in Table 5. The three function values at the initial estimate are of the same order of magnitude, thus apparently, scaling related difficulties should not be expected. At the solution, the function values are very small; two of them are interpreted as exactly zero by the computer and the third is of the order of 10^{-18}. All the independent variable values are between zero and one, thus all of the constraints are satisfied at this solution.

When started from the second, third or fourth set of initial estimates (see Table 4), most algorithms stop either issuing an error message of converge to an *infeasible solution* or indicating a *false solution*. Only the constrained algorithm, *CONLES*, that keeps alpha, x1 and x2 positive throughout the iterations enables convergence from initial guesses 3 and 4. The variable and function values at the infeasible and false solutions are shown in Table 6. At the infeasible solution, the value of alpha is greater than one (alpha = 1.47) implying a vapor flow rate exceeding the feed flow rate. At the false solution, x1 and x2 approach very small values with opposite signs and alpha approaches a very large value. The function values may get smaller as the iteration stopping tolerance is set. However, the point where x1 = 0 and x2 = 0 is clearly not a solution of this system of equations. It should be mentioned that all the programs tested misidentified the false solution as a true solution. More research is required in this area for developing an algorithm that can automatically identify false solutions of this type.

Infeasible solutions and false solutions are included in the library in order to caution the users, who may be unfamiliar with the physical basis of the problem, from accepting such solutions as valid ones.

A *plot of the function values* versus the independent variable values in the region of interest is also shown on the problem page for the cases where there is one implicit equation.

5. Structure of the NLE library

The NLE library can be accessed through the Web site: http://www.polymath-software.com/library. It contains presently over 70 problems where some of them are presented using several different formulations. The number of implicit equations ranges from a single equation to 14 equations with up to 32 additional explicit variables. All the problems are based on mathematical models related to physical phenomena like equations of state, chemical and phase equilibrium, etc.

Table 7
A section of the NLE libraries' main menu

Number	File name—Problem title—Problem characteristics/difficulties[a]
57	Sixeq3—Bubble point of a nonideal binary mixture (isobutanol–water) |F|
58	Sixeq4a—Modeling of a CSTR for a complex sequence of reactions—original form. |H | S | DI|
59	Sixeq4b—Modeling of a CSTR for a complex sequence of reactions—revised form. |MO|
60	Seveneq1—Chemical equilibrium |MO|
61	Seveneq2a—Esterification reaction in two consecutive CSTRs—original form. |H|
62	Seveneq2b—Esterification reaction in two consecutive CSTRs—revised form. |H|F|
63	Seveneq3a—Flow distribution in a pipeline network—constant friction factor
64	Seveneq3b—Flow distribution in a pipeline network—calculated friction factor
65	Seveneq4—Water distribution network
66	Nineq1—Three phase flash—ethanol, benzene and water mixture |H | MO|
67	Teneq1a—Chemical equilibrium problem—$R=10$ |DB|
68	Teneq1b—Chemical equilibrium problem—$R=40$ |H | DB|
69	Teneq2a—Combustion of propane in air—$R=10$ |DB|
70	Teneq2b—Combustion of propane in air—$R=40$ |H | DB|
71	Teneq3—Small flow sheeting system with recycle
72	11eq1—Stirred pot reactor process
73	13eq1—Gibbs energy minimization |H | DB|
74	14eq1—Three stage, two component distillation column

[a] Legend for problem characteristics/difficulties: L, lower; A, average; H, higher difficulty level; S, function/variable scaling related difficulties exist; DB, discontinuity/function undefined on the boundary; or DI, inside the feasible subspace; MI, multiple solutions inside; or MO, some outside of the feasible subspace; F, false solutions.

A section of the main menu from which the problem pages can be accessed is shown in Table 7. The problems are ordered according to the number of implicit equations. The problem identification number, title, and special characteristics are shown. These special characteristics include: (1) difficulty level (low, average, and high, only higher difficulty level is marked); (2) presence of variable or function scaling related difficulties; (3) existence of multiple solutions in the feasible and/or in the infeasible subspaces; (4) existence of discontinuities inside or on the boundary of the feasible subspace; and (5) presence of false solutions.

A complete single problem definition page can be viewed, printed or downloaded as an excel file. The model equations can be downloaded as a text file or as a polymath input file. Download files for other common mathematical packages will be added as they become available.

6. Discussion and conclusions

A web-based test problem library for NLEs that is most beneficial for potential users should include more information and in a different form than that found in existing test problem collections. Several important aspects of this library have been demonstrated by the two examples presented. In the NLE library created from this work, the equations are stored in the same form as the input for the numerical solver, constraints on the variables are defined explicitly, and initial estimates and function values at the initial estimates are included. Variable values at the solution are reported with the full precision of the numerical solution along with the function values at the solution. References, which provide information regarding the physical basis of the model represented by the system of equations, are also included as this can be helpful in verifying the physical validity of the solution.

Additional demonstrated attributes of the test problem library include: (1) categorization of the problems according to size, difficulty level, number of solutions and type of physical model; (2) convergence to infeasible solution; (3) misidentification of false solutions, and (4) initial estimate selection for various levels of difficulty.

References

Brauner, N., Shacham, M., & Cutlip, M. B. (1996). Computational results: how reliable are they? A systematic approach to model validation. *Chemical Engineering and Education*, *30*(1), 20–25.

Bullard, L. G., & Biegler, L. T. (1991). Iterative linear programming strategies for constrained simulation. *Computers and Chemical Engineering*, *15*(4), 239–254.

Cutlip, M. B., & Shacham, M. (1999). *Problem solving in chemical engineering with numerical methods* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

Damkohler, G., & Edse, R. Z. (1943). The composition of associating combustion gases and the calculation of simultaneous equilibria. *Z. Elektochimika*, *49*, 178–186.

Henley, E. J., & Rosen, E. M. (1969). *Material and energy balance computations*. New York: Wiley.

Hiebert, K. L. (1983). An evaluation of mathematical software that solves systems of nonlinear equations. *ACM Transactions on Mathematical Software*, *8*, 5–20.

Meintjes, K., & Morgan, A. P. (1990). Chemical equilibrium systems as numerical test problems. *ACM Transactions on Mathematical Software*, *16*, 143–151.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). *Numerical recipes* (2nd ed.). Cambridge: Cambridge University Press.

Shacham, M. (1984). Recent developments in solution techniques for systems of non-linear equations. In A. W. Westerberg, & H. H. Chien, *Proceedings of the second international conference on foundations of computer aided design* (pp. 891–924). Ann Arbor, MI: CACHE Publications.

Shacham, M. (1986). Numerical solution of constrained nonlinear equations. *International Journal of Numerical Methods in Engineering*, *23*, 1455–1481.

Shacham, M. (1990). A variable order method for solution of a nonlinear algebraic equation. *Computers and Chemical Engineering*, *14*(6), 621–629.

Shacham, M., Brauner, N., & Pozin, M. (1998). Comparing software for interactive solution of systems of nonlinear algebraic equations. *Computers and Chemical Engineering*, *22*, 321–323.