

A Web-based Library for Testing Performance of Numerical Software for Solving Nonlinear Algebraic Equations

M. Shacham^a, N. Brauner^b and M. B. Cutlip^c

^aDept. of Chemical Engineering, Ben Gurion University of the Negev, Beer-Sheva 84105, Israel*

^bSchool of Engineering, Tel-Aviv University, Tel-Aviv 699 78, Israel;

^cDept. of Chemical Engineering, University of Connecticut, Storrs, CT 06269, USA

The desirable structure for a web-based test problem library of nonlinear algebraic equations is investigated. Limitations of existing test problem collections—regarding the type of information included and the form in which this information is stored and presented—are demonstrated. It is concluded that a web-based test problem library with beneficial aspects for potential users should contain the problem definition 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 should be reported with the same precision as that of the numerical solution.

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 too often taken for granted that the numerical solver used can find the correct solution for a particular problem if it exists and that warning messages will be issued in case of doubts regarding the accuracy and/or correctness of the solution. Unfortunately, the solution provided by the numerical solver cannot always be trusted as some of our previous work^{[1],[2],[3]}, for example, has demonstrated. Testing the software's reliability requires that it be challenged by a large set of benchmark problems, which are known to be difficult to solve.

The need to validate the software is common to both the numerical and statistical software. For statistical software, a group of statisticians in NIST (National Institute of Standards and Technology of the U.S.) took the initiative and placed a large set of test problems on the Web (<http://www.itl.nist.gov/div898/strd/index.html>). This library contains problems related to: analysis of variance, linear and nonlinear regression, and univariate summary statistics. For 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.

* Author to whom correspondence should be addressed. e-mail: shacham@bgumail.bgu.ac.il

We have been working on the development of a similar test problem library for systems of nonlinear algebraic equations (NLEs). As a basis for the library, we are using collections of test problems published in the literature (see for example references [2] to [6]) and problems obtained by personal communications from individuals who needed help in solving particular types of equations. While preparing the library, attempts to reproduce the results obtained from various sources in the literature revealed many of the limitations of those test problem collections. These limitations helped to identify the type of information that should be included in a problem definition, the desired form in which this information should be stored and displayed, and the general structure of the library. Due to space limitations in this paper, only the considerations related to the single problem definition will be discussed herein.

The computations related to this article were carried out with the NLE solver program of the POLYMATH 5.0 package [copyrighted by M. Shacham, M. B. Cutlip and M. Elly (<http://www.polymath-software.com/>)]. The NLE library was implemented with Excel [Excel is trademark of Microsoft Corp. (<http://www.microsoft.com>)].

1. LIMITATIONS OF THE EXISTING TEST PROBLEM COLLECTIONS

The limitations of the existing test problem collections will be demonstrated with reference to the problem of “Combustion of Propane in Air”, which was used as a test problem, for example, by Hiebert^[5], Shacham^[3], Bullard and Biegler^[6], and Meintjes and Morgan^[7].

Table 1. Hiebert’s^[5] Version of the "Combustion of Propane in Air" Test Problem

No.	Equations	Variable	x_0	x^{*a}	$f(x^{*a})$	x^{*b}	$f(x^{*b})$
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	$f10 = x10*x4^2-3.846e-5*xs*x4^2 = 0$	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	

^aSolution obtained by Bullard and Biegler^[6], ^bSolution obtained by Shacham^[3]

The equations as presented by Hiebert^[5] are shown in Table 1. It should be pointed out from the outset, that Meintjes and Morgan^[7] 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, some of them are physically infeasible (x_i represents number of moles of various components thus all x_i must be 0). This brings up an additional issue, 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, and especially if the solution itself is very close to zero.

Hiebert[5], Shacham[3] and Bullard and Biegler[6] 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 estimates have been reported in these studies, but the function values at the initial estimate were not reported. The information regarding $\mathbf{f}(\mathbf{x}_0)$ is essential for a user who wants to reproduce the results, since differences in function values at the initial estimate signal errors in the problem setup. 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 estimate can also provide some information regarding the order of the 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, using 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^[5] attempted to solve this system of equations using 9 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^[5] have found two solutions to this system using an iterative linear programming strategy. The first solution, as they reported it, is shown in the column marked with \mathbf{x}^{*a} in Table 1. Shacham^[3] found one solution to this problem using the CONLES^[8] 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 the additional solution reported by Bullard and Biegler^[6] (not shown in Table 1).

The existence of three different solutions to this problem raises several options: 1. The problem has multiple solutions, and all the solutions obtained 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^[6] nor Shacham^[3] reported function values at the solution. The function values were calculated in this work by introducing the values shown under \mathbf{x}^{*a} and \mathbf{x}^{*b} into the equations. The resultant function values are shown in Table 1 (in the 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 non-zero 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^[6] report \mathbf{x}^* , in most cases, with a precision of four decimal digits. 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, our work verified the solution provided by Shacham^[3]. The utilized computer program (the “constrained” option of the POLYMATH 5.0 program) converged to

the same solution as shown in the column \mathbf{x}^{*b} of Table 1, and the resulting solution was obtained with 15 decimal digits of accuracy. These yield a highest function value of the order of 10^{-14} , which can be safely considered as zero in a double precision computation.

The solutions reported by Bullard and Biegler^[6] are suspected to be incorrect. This can be seen, for example, by considering the values of the two terms comprising f_{10} at the solution reported by them. The first (positive) term value is: $8.19 \cdot 0.005^2 = 2.05 \cdot 10^{-4}$. The value of the second (negative) term is $-3.846 \cdot 10^{-5} \cdot 108.817 \cdot 0.005^2 = -1.046 \cdot 10^{-7}$. Thus, the function value is equal to the value of the positive term, the negative term is insignificant and the solution shown is not a zero of f_{10} .

The conclusions that can be illustrated by this example regarding the structure and the information that has to be included in a test problem library are the following:

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 explicitly included in the library.
3. Different values should be used 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 compared to validate that the reported solution is a zero of the function.

2. STRUCTURE OF THE TEST PROBLEM LIBRARY

Many of the problems associated with the example presented in the previous section could have been easily detected if the physical nature of the model and the various variables was known. Although the inclusion of the description of the physical basis of the model in the library is not a must, it can be rather helpful in many cases.

Meintjes and Morgan^[7] have traced the “Combustion of Propane in Air” problem back to a paper from 1943 by Damkohler and Edse^[9]. The stoichiometric equation of the reaction is $C_3H_8 + R(O_2 + 4N_2) / 2$ 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 3 for a physically feasible solution to exist; if $R < 10$, the combustion is “fuel rich”, if

Table 2 shows the equations (as provided by Meintjes and Morgan^[7]) in the form they are stored in the test problem library. The unknowns n_1, n_2, \dots, 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, while 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 they appear in a separate column of Table 2. All the variables 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)

(otherwise, a negative value inside the square root function may appear, in this particular example).

Table 2. Meintjes and Morgan^[7] Version of the “Combustion of Propane in Air” Test Problem

No.	Equations ^a	Constrains ^b
1	$f(n1) = n1+n4-3$ #Mol of Carbon Dioxide - Carbon Balance	
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.)	

^aAn implicit equation is indicated by $f(\cdot)=$. Output variables assigned arbitrarily for implicit eqns.

^bConstraint on the output variable. An (a) indicates that the constraint must be always satisfied

Table 3. Initial Estimates and Solution for the "Combustion of Propane in Air" Test Problem

Function and variable number	Initial value	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	

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 directly copied into the POLYMATH 5.0 program for solution. If other programs are used for solution (such as MATLAB or MATHEMATICA), some modifications may be required. The required editing can be easily performed with Excel.

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 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).

3. DISCUSSION AND CONCLUSIONS

A web-based test problem library for NLEs that is most beneficial for potential users needs to include more information and in a different form than that found in existing test problem collections. Several important aspects of the proposed library have been demonstrated by the previous example. The equations should be stored in the same form as the input for the numerical solver, constraints on the variables should be defined explicitly, and initial estimates and function values at the initial estimates should be included. Variable values at the solution should be reported with the same precision that of the numerical solution, and function values at the solution must be given. While the inclusion of the physical basis of the model represented by the system of equations is not essential, it can be often helpful in verifying the physical validity of the solution.

Additional aspects 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. Modification of the equations for alleviating the solution process. and 3. Initial estimate selection for various levels of difficulty. These have not been discussed here due to space limitation.

REFERENCES

1. N. Brauner, M. Shacham and M. B. Cutlip, *Chem. Eng. Educ.*, **30** (1), 20-25 (1996).
2. Shacham, M., N. Brauner and M. Pozin, *Computers chem. Engng.*, **22**,321-323(1998).
3. Shacham, M., pp.891-924 in Westerberg, A.W. and Chien, H. H. (Eds) Proceedings of the 2nd International Conference FOCAPD, CACHE Corp., 1984.
4. Shacham, M., *Computers chem. Engng.*, **14** (6), 621-629 (1990).
5. Hiebert K. L., *ACM Trans. Math. Software* **8**(1), 5-20 (1982).
6. Bullard, L.G. and Biegler, L.T., *Computers chem. Engng.*, **15**(4), 239-254 (1991).
7. Meintjes, K. and Morgan, A.P., *ACM Trans. Math. Software*, 16(2), 143-151(1990).
8. Shacham, M., *Intern. Journal of Numerical Methods in Engineering*, **23**,1455-1481, (1986).
9. Damkohler, G. and Edse, R.Z., *Elektrochem.*, **49**, 178-186 (1943).