SOLVING A SYSTEM OF NONLINEAR ALGEBRAIC EQUATIONS

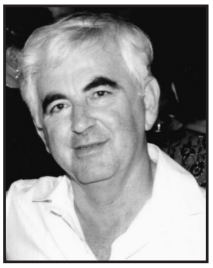
You Only Get Error Messages—What to do Next?

Mordechai Shacham[a] and Neima Brauner[b]
a Ben-Gurion University of the Negev • Beer-Sheva 84105, Israel
b Tel-Aviv University • Tel-Aviv 69978, Israel

Many practical problems in chemical engineering give rise to the need to solve systems of nonlinear algebraic equations (NLE). Typical examples include phase equilibrium and chemical equilibrium computations, steady state process simulation, flow distribution in pipeline networks, etc. Nowadays powerful mathematical software packages are available for solving systems of NLEs. Cutlip and Shacham,[1] for example, demonstrate the use of the POLYMATH,[2] and MATLAB[3] software packages and the Excel[4] spreadsheet for solving various NLE systems encountered in undergraduate and graduate chemical engineering education.

The availability of the powerful software packages and solved textbook examples that utilize these packages may be sufficient to enable students to tackle many ChE problems associated with solution of NLE systems. However, even if the students follow exactly the instructions of the NLE solver packages and the textbook examples, they still may find that using a particular problem formulation and a set of initial guesses for the unknowns, a solution is not reached and the program issues error messages (*e.g.*, "division by zero"). Our experience has shown that to deal with such a situation, students need to learn to analyze the equation set and modify it, if necessary, to enable convergence to the solution. The objective of this paper is to present the analysis and the systematic modification process, along with examples that we have been using to educate students in the framework of mathematical modeling and numerical methods courses.

Even though there are so-called "globally convergent" methods for solving NLEs (see, for example, p. 176 in Reference 5), sometimes even such methods are unable to find the solution regardless of the initial estimates provided for the unknowns. The most common cause for the failure of globally convergent methods to find a solution (besides the trivial case of errors in the problem formulation) is the presence of discontinuities and/or regions where some of the functions are undefined.[6] The impact of such discontinuities becomes most severe if they lie in the vicinity of the solution. Analysis of the system of NLEs in order to locate the discontinuities and the boundaries of the regions where some functions are undefined, is a much more complex task than finding a solution. The objective here is to propose and demonstrate a procedure for reformulating the NLE system so as to increase the probability of finding a solution and at the same time speeding up the solution process.



Mordechai Shacham *is professor emeritus of the Department of Chemical Engineering at the Ben-Gurion University of the Negev in Israel. He served as department head and the chairman of the Israeli Inter-University Center for e-Learning (IUCEL). He received his B.Sc. and D.Sc. degrees from the Technion, Israel Institute of Technology. His research interest includes analysis, modeling, and regression of data; applied numerical methods; and prediction of physical properties.*

Neima Brauner *is currently a professor of mechanical engineering at Tel-Aviv University, School of Engineering. She received her B.Sc. and M.Sc. in chemical engineering from the Technion Institute of Technology in Haifa, and a Ph.D. in mechanical engineering from Tel-Aviv University in 1983. Her research interest includes hydrodynamics and transport phenomena in two-phase flow systems, and applications of interactive statistical and numerical methods in regression of experimental data, process analysis, and design.*

The proposed procedure involves reformulating the equations that contain functions with discontinuities (*e.g.*, logarithm or reciprocal of an unknown) with continuous counterpart functions (*e.g.*, exponent), and reducing the dimension of the NLE system by expressing some of the unknowns as explicit functions of the others while rearranging the computation order. The use of the proposed procedure is demonstrated by two examples.

DEFINITION OF EXAMPLE PROBLEM 1

The first example is the "VLE: Wilson's Equation" problem provided in the LearnChemE website.[7] In this example the bubble point temperature and the vapor phase composition of a non-ideal binary mixture have to be found using the modified Raoult's law and the Wilson equation. The equations that need to be solved are shown in Table 1. There are 10 nonlinear equations with 10 unknowns: P_1^{Sat} and P_2^{Sat} (kPa) – saturation pressure values for components 1 and 2, T – temperature (K), y_1 and y_2 – mole fraction of components 1 and 2 in the gas phase, γ_1 and γ_2 – activity coefficients, and W, Λ12, Λ21 – the Wilson parameters. In addition, the data provided include the liquid mole fraction of the first compound (x_1=0.85), pressure (P = 100 kPa) and the values of the Wilson parameters: a_{12} = 440 cal/mol, a_{21} = 1250 cal/mol, V_1 = 77 cm^3/mol, and V_2 = 18 cm^3/mol.

TABLE 1
Equation set for Example 1: VLE – Wilson's formulated as a system of 10 implicicit NLE (source: LearnChemE website)

Eq. No.	Equations
1	$f_1 = \ln(P_1^{Sat}) - 17 + 3600/(T-54) = 0$
2	$f_2 = \ln(P_2^{Sat}) - 16.5 + 3850/(T-47) = 0$
3	$f_3 = x_1\gamma_1 P_1^{Sat} - y_1 P = 0$
4	$f_4 = x_2\gamma_2 P_2^{Sat} - y_2 P = 0$
5	$f_5 = 1 - y_1 - y_2 = 0$
6	$f_6 = \ln(\gamma_1) + \ln(x_1 + x_2\Lambda_{12}) - x_2 W = 0$
7	$f_7 = \ln(\gamma_2) + \ln(x_1\Lambda_{21} + x_2) + x_1 W = 0$
8	$f_8 = W - \frac{\Lambda_{12}}{(x_1 + x_2\Lambda_{12})} + \frac{\Lambda_{21}}{(x_1\Lambda_{21} + x_2)} = 0$
9	$f_9 = \Lambda_{12} - \frac{V_2}{V_1}\exp\left(\frac{-a_{12}}{RT}\right) = 0$
10	$f_{10} = \Lambda_{21} - \frac{V_1}{V_2}\exp\left(\frac{-a_{21}}{RT}\right) = 0$

The equations in Table 1 are all written in the implicit form of $f_i(x) = 0$, of 10 equations with 10 unknowns. There are several mathematical software packages that can be used for solving such a system of equations. LearnChemE recommends the use of the POLYMATH.[2] Introducing the equations of Table 1 into POLYMATH requires some changes in the notation as POLYMATH does not accept Greek letters, subscripts, and superscripts (the variable names as entered into POLYMATH are shown in Table 2). Additionally, the names of the constants and their numerical values, and initial guesses for the expected values of the 10 unknowns at the solution, need to be entered into POLYMATH.

The complete POLYMATH program for solving this problem is available at the website: <www.learncheme.com/student-resources/polymath>.

The solution obtained by POLYMATH for this problem is shown in Table 2. The results presented by POLYMATH include the names and values of the constants, the names, initial guesses, the values at the solution (x*), and the value of $f_i(x^*)$ at the solution for all the unknowns. Observe that all the function values are very close to zero (between 0 – 1.71e-12, in absolute value), thus at the solution all the implicit equations are satisfied with high accuracy. Initial guesses were entered for all 10 unknowns. For some of the variables (like P_1^{Sat} and P_2^{Sat}, T, y_1, and y_2) good initial estimates can be determined based on physical considerations, but for others (the Wilson's equation parameters) this task is more difficult. Observe, for example, that the initial guess given for W is positive, while at the solution it attains a negative value. Providing inappropriate initial guesses may cause failure of the solution algorithm. Using some negative values as an initial estimate for Λ12 (such as -0.05) leads to an error message related to an attempt to calculate the logarithm of a negative number.

TABLE 2
Results presented by POLYMATH for example problem 1

	Variables (Unknowns)				Constants	
No.	Name	Value	f(x)	Init. Guess	Name	Value
1	G12	0.123	4.16E-17	1	a12	440
2	G21	0.688	-1.11E-16	1	a21	1250
3	gamma1	1.022	1.39E-16	1	P	100
4	gamma2	2.675	-4.95E-13	1	R	1.987
5	P1sat	99.077	0	90	V1	77
6	P2sat	34.706	3.55E-15	50	V2	18
7	T	344.227	2.78E-17	350	x1	0.85
8	W	-0.795	-3.33E-16	0.5	x2	0.15
9	y1	0.861	0	0.6		
10	y2	0.139	1.71E-12	0.5		

In the following section a systematic

procedure is presented for rearrangement of the equation set in order to reduce the number of implicit algebraic equations and to reduce the probability for carrying out invalid algebraic operations along the solution path.

NONLINEAR ALGEBRAIC EQUATION SET ANALYSIS AND MODIFICATION TO ENHANCE CONVERGENCE AND REDUCE COMPUTATIONAL EFFORT

Using the following procedure for reformulation of the NLE system is recommended if a solution of the system cannot be found even when the calculations are started from several combinations of initial estimates and the computation is stopped because of an illegal arithmetic operation (*e.g.*, overflow, logarithm of a negative number). The reformulation and rearrangement procedure may be necessary also if the NLE system has to be repeatedly solved with different input data (*e.g.*, when using the VLE Wilson – Equation set in simulation of a distillation column for calculation of the bubble point temperatures in the various stages).

The proposed procedure for reformulation and rearrangement of the NLE system includes the following four stages:

1. *Identify expressions with discontinuities (like logarithm, square root, division by unknowns, etc.). Modify such expressions as to eliminate the discontinuities (See a few examples in Table A-1 in Appendix A)*
2. *Select a unique "output" variable from each of the equations, where an output variable only appears on the left-hand side (l.h.s) of one (and only one) of the equations. Leave the equations from which no output variable can be selected in an implicit form.*
3. *Mark the variables that could not be selected as output variables of a "known" (value) and arrange the explicit equations in a sequence so that only "known" variables appear in the right-hand side (r.h.s.) of the equations and the "output" variable on the l.h.s. Explicit equations that cannot be arranged in this form should be rewritten as implicit equations.*
4. *Solve the revised system of equations with an NLE solver program.*
5. *Substitute the solution obtained into the original NLE system to verify that the reformulated and rearranged system is identical (in terms of the solution obtained) to the original system.*

If the NLE system can be brought into the form of a single NLE with one unknown, finding all the solutions becomes straightforward. The function value can be plotted vs. the single unknown value and its solutions correspond to the points where the function changes sign. For NLE systems containing more than one implicit equation, some trial and error with regard to the initial estimates for the unknowns may be required.

TABLE 3
Equation set of Table 1 rewritten by modifying Eqs. 1, 2, 6, and 7 and selecting output variables.

Eq. No.	Equations
1	$P_1^{Sat} = \exp[17 - 3600/(T-54)]$
2	$P_2^{Sat} = \exp[16.5 - 3850/(T-47)]$
3	$y_1 = x_1\gamma_1 P_1^{Sat}/P$
4	$y_2 = x_2\gamma_2 P_2^{Sat}/P$
5	$f_1 = y_1 + y_2 - 1 = 0$
6	$\gamma_1 = \exp(x_2 W)/(x_1 + x_2\Lambda_{12})$
7	$\gamma_2 = \exp(-x_1 W)/(x_1\Lambda_{21} + x_2)$
8	$W = \frac{\Lambda_{12}}{(x_1 + x_2\Lambda_{12})} - \frac{\Lambda_{21}}{(x_1\Lambda_{21} + x_2)}$
9	$\Lambda_{12} = \frac{V_2}{V_1}\exp\left(\frac{-a_{12}}{RT}\right)$
10	$\Lambda_{21} = \frac{V_1}{V_2}\exp\left(\frac{-a_{21}}{RT}\right)$

DEMONSTRATION OF THE NLE SYSTEM REARRANGEMENT ALGORITHM FOR EXAMPLE PROBLEM 1

Stage 1. Eqs. 1, 2, 6, and 7 (in Table 1) include logarithms of the unknowns P_1^{Sat}, P_2^{Sat}, γ_1, γ_2, $\Lambda 12$, and $\Lambda 21$. The equations can be modified so as to contain exponential terms instead of logarithmic terms. Eq. 1 can be rewritten as: $P_1^{Sat} = \exp[17–3600/(T–54)]$ and Eq. 6 can be modified to $\gamma_1 = \exp(x_2 W)/(x_1 + x_2 \Lambda 12)$. Eqs. 2 and 7 can be modified in a similar manner.

Stage 2. Unique output variables can be selected for nine out of the 10 equations. The variable T cannot be expressed as an output variable and Eq. 5 must retain its implicit form. The revised set of nine explicit and one implicit equations is shown in Table 3.

Stage 3. This stage involves the sequencing of the computations using the "variable mapping table." This table includes the equation numbers and the associated input and output variables (see Table 4, next page). Input variables that have been calculated already, or whose value has been specified, can be deleted from the table. If for an explicit equation all the input variables are known the output variable can be calculated and removed from the input variables column. For the sake of clarity, variables with known values are shown in Table 4 in bold letters (instead of removing them).

In the first step of the computational sequencing procedure there is no equation for which all the input variables have known values and the equations with minimal number of input variables are looked for. For these variables an initial guess

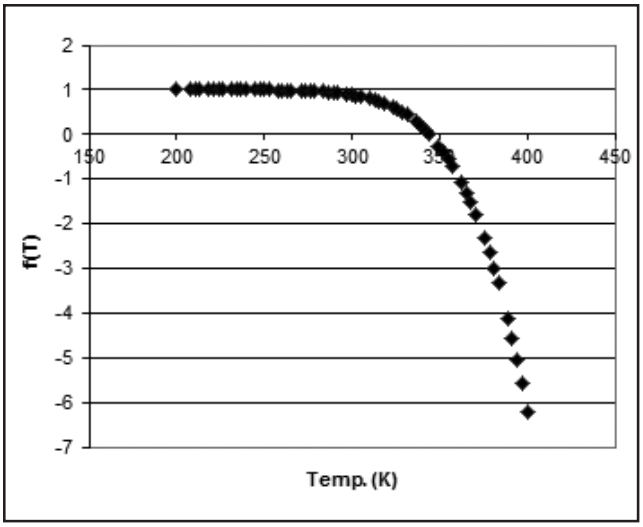


Figure 1. *Function value (Eq. 5 in Table 5) in the 200 K – 400 K temperature range.*

has to be specified and they are denoted as "tear" variables. In this particular example the variable T is a single input variable in four explicit equations, consequently it is designated as a "tear" variable. Assigning a trial value for T enables calculating the output variables: P_1^{Sat} (Eq. 1), P_2^{Sat} (Eq. 2), Λ12 (Eq. 9) and Λ21 (Eq. 10). These operations are recorded in the computation sequence: (T) 1, 2, 9, 10 indicating that T is the tear variable, and upon assigning a value to T, the explicit Equations 1, 2, 9, 10 can be solved in the order specified.

In step 2 of the sequencing procedure the equations that were already included in the computational sequence are removed from the table, and the input variables that were calculated in the previous step are marked (in bold letters). Observe that the two input variables to Eq. 8 are known. Consequently the output variable W can be calculated and Eq. 8 can be added to the calculation sequence. With the known W value, the output variables γ_1 and γ_2 can be calculated from Eqs. 6 and 7, respectively (step 3 in Table 4), whereby these equations can be added to the computation sequence. In the last step (4) the variables y_1 and y_2 are calculated from the explicit Eqs. 3 and 4, and finally the value of f (T) is calculated from Eq. 5, yielding the final computation sequence: (T) 1, 2, 9,10, 8, 6, 7, 3, 4, 5[f(T)]. This way the system of 10 equations and 10 unknowns can be easily solved, as there is only one implicit equation with one unknown: T.

TABLE 4
Computation sequencing with the variable mapping table.

Computation sequencing with the variable mapping table

Step 1.

Input Variable/s	Eq. No.	Output Variable/s
$\boldsymbol{T}$	1	$\mathbf{P_1^{Sat}}$
$\boldsymbol{T}$	2	$\mathbf{P_2^{Sat}}$
γ_1, P_1^{Sat}	3	y_1
γ_2, P_2^{Sat}	4	y_2
y_1, y_2	5	$f(T)$
Λ_{12}, W	6	γ_1
Λ_{21}, W	7	γ_2
$\Lambda_{12}, \Lambda_{21}$	8	W
$\boldsymbol{T}$	9	$\mathbf{\Lambda_{12}}$
$\boldsymbol{T}$	10	$\mathbf{\Lambda_{21}}$

Computational sequence: (*T*) 1, 2, 9,10

Step 2.

$\mathbf{\Lambda_{12}, \Lambda_{21}}$	8	$\boldsymbol{W}$

Computational sequence: (*T*) 1, 2, 9,10, 8

Step 3.

$\mathbf{\Lambda_{12}, W}$	6	$\mathbf{\gamma_1}$
$\mathbf{\Lambda_{21}, W}$	7	$\mathbf{\gamma_2}$

Computational sequence: (*T*) 1, 2, 9,10, 8, 6, 7

Step 4.

$\mathbf{\gamma_1, P_1^{Sat}}$	3	$\boldsymbol{y_1}$
$\mathbf{\gamma_2, P_2^{Sat}}$	4	$\boldsymbol{y_2}$
y_1, y_2	5	$f(T)$

Comput. sequence: (*T*) 1, 2, 9,10, 8, 6, 7, 3, 4, 5[*f*(*T*)]

Stage 4. In Figure 1 the function value of the revised NLE system (Eq. 5 in Table 5) in the 200 K – 400 K temperature range is shown. Observe that there is only one root (with f(T) = 0) in the region of interest. The numerical value of T at the solution is is 344.2265 K, with f(T) = -1.535e-11. Using this T value the explicit equations in Table 5 provide the values of the rest of the unknowns at the solution. Introducing these unknown values into the original (implicit) set of NLE in Table 1 yields function absolute values in the range of 0 – 1.77e-17. Thus, the revised system of equations and the solution

The objective here is to propose and demonstrate a procedure for reformulating the NLE system to increase the probability of finding a solution and at the same time speeding up the solution process.

obtained match the original problem. A MATLAB program: *BubblePT.m* that solves the revised NLE system and introduces the solution into the original system is available at: <ftp://ftp.bgu.ac.il/shacham/NLE_CEE16/>.

DEFINITION OF EXAMPLE PROBLEM 2

This example involves calculation of reaction equilibrium by minimization of the Gibbs energy. The example was introduced by Balzisher, et al.[8] and is discussed in detail by Cutlip and Shacham.[1] Ethane is steam cracked to form hydrogen over a cracking catalyst at temperature T = 1000 K and pressure of P = 1 atm. The feed contains 4 moles of H_2O per mole of C_2H_6. It is assumed[8] that only the compounds 1. CH_4, 2. C_2H_4, 3. C_2H_2, 4. CO_2, 5. CO, 6. O_2, 7. H_2, 8. H_2O and 9. C_2H_6 are present in the equilibrium mixture (assuming that no carbon is deposited). The Gibbs energies of formation of the various compounds at the reaction temperature of 1000 K are available in the reference.[1] Example problem 2 involves the calculation of the equilibrium composition of the effluent mixture.

The solution of this problem involves the minimization of the total Gibbs energy given by

$$\min_{n_i}\frac{G}{RT}=\sum_{i=1}^{c}n_i\left(\frac{G_i^0}{RT}+\ln\frac{n_i}{\sum n_i}\right) \qquad (2.1)$$

where n_i is the number of moles of component i, c is the total number of compounds, R is the gas constant and G_i^0 is the Gibbs energy of pure component i at temperature T. The minimization of Eq. (2.1) is carried out subject to atom balance constraints:

Oxygen balance $$g_1=2n_4+n_5+2n_6+n_8-4=0 \qquad (2.2)$$

Hydrogen balance $$g_2=4n_1+4n_2+2n_3+2n_7+2n_8+6n_9-14=0 \qquad (2.3)$$

Carbon balance $$g_3=n_1+2n_2+2n_3+n_4+n_5+2n_9-2=0 \qquad (2.4)$$

where n_i represents amount (moles) of a compound i (i corresponds to the component number provided in the list above). The three constraints can be introduced into the objective functions using Lagrange multipliers: λ_1, λ_2, and λ_3. The extended objective function is

$$\min_{n_i,\gamma_j}F=\sum_{i=1}^{c}n_i\left(\frac{G_i^0}{RT}+\ln\frac{n_i}{\sum n_i}\right)+\sum_{j=1}^{3}\lambda_j g_j \qquad (2.5)$$

The condition for minimum of this function at a particular point is that all the partial derivatives of F with respect to n_i and λ_j vanish at this point. Taking the derivatives of Eq. (2.5) and putting them in a POLYMATH program, together with Eqs. (2.2), (2.3), and (2.4) and initial estimates for all the unknown n_i yields the program shown in Table 6 (next page). The POLYMATH model (including the "comments," which start with the # sign) provides complete documentation of the equations, the values of the constants, and the initial

TABLE 5
Equation set of Example 1 rewritten as a system of 9 explicit and 1 implicit NLEs.

Eq. No.	Equations
1	$P_1^{Sat}=\exp[17-3600/(T-54)]$
2	$P_2^{Sat}=\exp[16.5-3850/(T-47)]$
9	$\Lambda_{12}=\frac{V_2}{V_1}\exp\left(\frac{-a_{12}}{RT}\right)$
10	$\Lambda_{21}=\frac{V_1}{V_2}\exp\left(\frac{-a_{21}}{RT}\right)$
8	$W=\frac{\Lambda_{12}}{(x_1+x_2\Lambda_{12})}-\frac{\Lambda_{21}}{(x_1\Lambda_{21}+x_2)}$
6	$\gamma_1=\exp(x_2W)/(-x_1+x_2\Lambda_{12})$
7	$\gamma_2=\exp(x_1W)/(x_1\Lambda_{21}+x_2)$
3	$y_1=x_1\gamma_1P_1^{Sat}/P$
4	$y_2=x_2\gamma_2P_2^{Sat}/P$
5	$f_1=y_1+y_2-1=0$

estimates used for the 12 unknowns. The 12 implicit algebraic equations [associated with Eqs. (2.2), (2.3), and (2.4) and the nine partial derivatives of F in Eq.(2.5)] are shown in rows 3 through 14 of Table 6. It is assumed that at the solution these equations should be equal to zero. For the sake of clarity, in the POLYMATH input the n_i is represented by the formula of the compound. Initial estimates for the 12 unknowns are specified in rows 16 through 27. Those are based on the values suggested by Balzisher, et al.,[8] and Cutlip and Shacham.[1]

This system of equations is very difficult to solve as the functions are undefined for $n_i \le 0$ and for some of the compounds the amount in the effluent is very close to zero. Non-constrained equation solvers may overstep the zero value, requiring calculation of the logarithm of a negative value, which stops the computation. With constrained equation solvers it is difficult to determine how close to zero the constraints should be specified. If the constraints are set too far from a zero value, incorrect results may be obtained. Cutlip and Shacham attempted to solve this system of equations using the POLYMATH,

TABLE 6
POLYMATH representation of the equations of Example 2

No.	Equation # Comment
1	R = 1.9872
2	sum = H2 + O2 + H2O + CO + CO2 + CH4 + C2H6 + C2H4 + C2H2
3	f(lamda1) = 2 * CO2 + CO + 2 * O2 + H2O - 4 # Oxygen balance
4	f(lamda2) = 4 * CH4 + 4 * C2H4 + 2 * C2H2 + 2 * H2 + 2 * H2O + 6 * C2H6 - 14 # Hydrogen balance
5	f(lamda3) = CH4 + 2 * C2H4 + 2 * C2H2 + CO2 + CO + 2 * C2H6 - 2 # Carbon balance
6	f(CH4) = 4.61 / R + ln(CH4 / sum) + 4 * lamda2 + lamda3
7	f(C2H4) = 28.249 / R + ln(C2H4 / sum) + 4 * lamda2 + 2 * lamda3
8	f(C2H2) = 40.604 / R + ln(C2H2 / sum) + 2 * lamda2 + 2 * lamda3
9	f(CO2) = -94.61 / R + ln(CO2 / sum) + 2 * lamda1 + lamda3
10	f(CO) = -47.942 / R + ln(CO / sum) + lamda1 + lamda3
11	f(O2) = ln(O2 / sum) + 2 * lamda1
12	f(H2) = ln(H2 / sum) + 2 * lamda2
13	f(H2O) = -46.03 / R + ln(H2O / sum) + lamda1 + 2 * lamda2
14	f(C2H6) = 26.13 / R + ln(C2H6 / sum) + 6 * lamda2 + 2 * lamda3
15	# Initial estimates
16	lamda1(0) = 10
17	lamda2(0) = 10
18	lamda3(0) = 10
19	CH4(0) = 0.001
20	C2H4(0) = 0.001
21	C2H2(0) = 0.001
22	CO2(0) = 0.993
23	CO(0) = 1
24	O2(0) = 0.0001
25	H2(0) = 5.992
26	H2O(0) = 1
27	C2H6(0) = 0.001

MATLAB, and Excel packages. None of the programs was able to solve the problem in the form it is presented in Table 6. Solution of the problem was obtained in a "trial and error" process, where the particular equation that caused the failure was modified in order to enable further progress of the solution process.

REARRANGEMENT AND SOLUTION OF EXAMPLE PROBLEM 2

Stage 1. The regions where the functions are undefined can be eliminated by modifying the equations in rows 6 through 14 (Table 6) so that the calculations of logarithms of very small numbers are avoided. Essentially, the modification involves replacement of the logarithm operation with exponentiation. For example, the equation in row 6 of Table 6 can be replaced by the following equation: CH4 = exp(- 4.61 / R - 4 * lamda2 - lamda3)*sum. Another modification that can be very beneficial is the replacement of the explicit equation for calculating *sum* by an implicit equation: f(sum) =sum- (H2 + O2 + H2O + CO + CO2 + CH4 + C2H6 + C2H4 + C2H2). This modification enables the selection of output variables from several implicit equations, which would be otherwise impossible.

Stages 2 and 3. Unique output variables can be selected for nine out of the (now) 13 equations. Upon selecting λ_1, λ_2, and λ_3 as output variables, H2O, H2,CO and *sum* need to be selected as "tear" variables. With these tear variables the remaining nine explicit equations can be sequenced following the route outlined in Example 1. The reformulated and rearranged equation set is shown in Table 7. Observe that in this case initial estimates are needed only for four variables (the initial estimates values shown in Table 6 were used). Using this formulation, a converged solution (Table 8) was obtained by POLYMATH, MATLAB, and Excel without any difficulties.

With POLYMATH the *safenewt* algorithm was used for the solution. This algorithm is based on the newt subroutine provided by Press, et al.[5] It uses the Newton-Raphson method with line search to ensure reduction of the norm of the function values in every iteration. For solution with MATAB the *fsolve* function was used. In this function one of the available methods: the trust-region dogleg method (based on Powell's dogleg method[9]) was utilized. Using the Excel's Solver to solve the problem, the sum of squares of the function values is minimized using the GRG[10] (Generalized Reduced Gradient) method. Identical solutions were obtained by the three software tools.

Stage 4. Introducing the variable values shown in Table 8 into the original equations of Table 6 yields function absolute values in the range of 4.44e-16 – 1.37e-13. Thus the revised system is identical to the original one. A MATLAB program: *GibbsEnergy.m*, that solves the revised NLE system and introduces the solution into the original system is available at <ftp://ftp.bgu.ac.il/shacham/NLE_CEE16/>.

USING THE EXAMPLES IN THE CLASSROOM

The subject of numerical solution of NLE systems is included in the Numerical Methods 3rd-year required course

and in the Process Simulation 4th-year elective course for undergraduate chemical engineering students at the Ben-Gurion University of the Negev. The Numerical Methods course is described in some detail by Shacham, et al.[11] and the Process Simulation course is described in detail by Shacham.[12] The procedure for reformulation and rearrangement of NLE systems is discussed in class and occasionally homework assignments related to this issue are given. Examples for "difficult to solve" NLE systems are usually taken from References 6 and 13. The examples presented here have been recently added to the assignments of these courses.

CONCLUSIONS

The examples presented here demonstrate that the solution of some NLE systems may represent a challenge even if the problem formulation is correct and state of the art "globally convergent" methods are used for solution.

It has been shown that the proposed method for reformulation and rearrangement of the NLE systems can reduce the dimension of the system, remove discontinuities, and considerably simplify the task of finding initial estimates for the unknowns so that solution is obtained. This is demonstrated in particular in example 2: by following the proposed procedure, three software packages (POLYMATH, MATLAB, and Excel) were able to find the solution from the same initial estimate that caused them to fail with the original problem formulation.

This material can be very helpful for undergraduate and graduate students and practicing engineers who are involved in mathematical modeling, which often requires solution of nonlinear equations.

TABLE 7
Equations of Example 2 reformulated and rearranged as four implicit equations with four unknowns

No.	Equation # Comment
1	R = 1.9872
2	f(sum) =sum- (H2 + O2 + H2O + CO + CO2 + CH4 + C2H6 + C2H4 + C2H2)
3	f(H2O) = 2 * CO2 + CO + 2 * O2 + H2O - 4 # Oxygen balance
4	f(H2) = 4 * CH4 + 4 * C2H4 + 2 * C2H2 + 2 * H2 + 2 * H2O + 6 * C2H6 - 14 # Hydrogen balance
5	f(CO) = CH4 + 2 * C2H4 + 2 * C2H2 + CO2 + CO + 2 * C2H6 - 2 # Carbon balance
6	lamda2 = -ln(H2 / sum) /2
7	lamda1=46.03 / R - (ln(H2O / sum) + 2 * lamda2)
8	lamda3 =47.942 / R -(ln(CO / sum) + lamda1)
9	CH4 = exp(- 4.61 / R - 4 * lamda2 - lamda3)*sum
10	C2H4 = exp(-28.249 / R- 4 * lamda2 - 2 * lamda3)*sum
11	C2H2 = exp(-40.604 / R - 2 * lamda2 - 2 * lamda3) * sum
12	CO2 = exp(94.61 / R - 2 * lamda1 - lamda3)*sum
13	O2 = exp(-2 * lamda1) * sum
14	C2H6 =exp(-26.13 / R - 6 * lamda2 - 2 * lamda3)*sum
15	# Initial estimates
16	sum(0)=8
17	CO(0)=1
18	H2(0)=5.992
19	H2O(0)=1

TABLE 8
Results presented by POLYMATH for example problem 2

		Variables (Implicit Equations)		
No.	Name	Value	f(x)	Init. Guess
1	CO	1.388517	5.15E-08	1
2	H2	5.345225	8.033E-09	5.992
3	H2O	1.521646	9.898E-08	1
4	sum	8.866871	-5.15E-08	8
		Variables (Explicit Equations)		
5	C2H2	3.157E-10		
6	C2H4	9.541E-08		
7	C2H6	1.671E-07		
8	CH4	0.0665638		
9	CO2	0.5449182		
10	lamda1	24.41966		
11	lamda2	0.2530591		
12	lamda3	1.559832		
13	O2	5.459E-21		
14	R	1.9872		

REFERENCES

1. Cutlip, M.B., and M. Shacham, *Problem solving in chemical and biochemical engineering with POLYMATH, Excel and MATLAB*, 2nd Ed., Prentice Hall, Upper Saddle River, N. J. (2008)
2. POLYMATH is a product of Polymath Software (<http://www.polymath-software.com>)
3. MATLAB is a trademark of The Math Works, Inc. (<http://www.mathworks.com>)
4. Excel is a trademark of Microsoft Corporation (<http://www.microsoft.com>)
5. Press, W.H., P.B. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipe*s, 2nd Ed., Cambridge University Press, Cambridge (1992)

6. Shacham, M., and N. Brauner," Numerical Solution of Nonlinear Algebraic Equations with Discontinuities," *Comp. & Chem. Eng.*, **26**, 1449 (2002)
7. "LearnChemE – Educational Resources for Chemical Engineering" website of the University of Colorado, Boulder (<www.learncheme.com>)
8. Balzisher,R.E., M.R. Samuels, and J.D. Eliassen, *Chemical Engineering Thermodynamics*, Prentice-Hall, Englewood Cliffs, NJ (1972)
9. Powell, M.J.D., "A Fortran Subroutine for Solving Systems of Nonlinear Algebraic Equations," Ch.7 in *Numerical Methods for Nonlinear Algebraic Equations*, P. Rabinowitz (Ed.) (1970)
10. Abadie, J., "The GRG Method for Nonlinear Programming," pp. 335-363 in Design and Implementation of Optimization Software, H. J. Greenberg (Ed.), Sijthoff and Noordhoff (1978)
11. Shacham, M., M.B. Cutlip, and N. Brauner, "From Numerical Problem Solving to Model Based Experimentation—Incorporating Computer Based Tools of Various Scales into the ChE Curriculum," *Chem. Eng. Ed.*, **43**(4), 315 (2009)
12. Shacham, M., "Use of Advanced Educational Technologies in a Process Simulation Course," *Computer-Aided Chemical Engineering*, **29**, 1135 (2011)
13. Shacham, M., N. Brauner, and M.B. Cutlip, "A Web-based Library for Testing Performance of Numerical Software for Solving Nonlinear Algebraic Equations," *Computers Chem. Engng.*, **26**(4-5), 547 (2002)

APPENDIX A

Examples of Equation Modification for Discontinuity Removal

In Table A-1 three pairs of equations are presented where the first equation in the pair contains discontinuity and the second equation in the pair is in a modified form, where the discontinuity has been removed. For example in Eq. (1.1), due to the division by (1 – x), the function is undefined for x = 1. This discontinuity is removed upon multiplying the two terms of the equation by (1 – x), Eq. 1.2. It can be seen that the solution of the two versions of the NLE are identical. ❐

TABLE A-1
Examples of discontiuity removal from various equations.

No.	Function	Undefined for	Solution
1.1	f(x) = 2*x^2-1/(1-x) = 0	x = 1	-0.5651977
1.2	f(x) = 2*x^2*(1-x)-1 = 0	-	-0.5651977
2.1	f(x) = x/5-ln(x) = 0	x ≤ 0	12.71321
2.2	f(x) = exp(x/5)-x = 0	-	12.71321
3.1	f(x) = x/5-(x+2)^(1/2) = 0	x ≤ - 2	26.86141
3.2	f(x) = (x/5)^2-(x+2) = 0	-	26.86141