DYNAMIC SIMULATION OF HIGH-INDEX MODELS OF BATCH DISTILLATION PROCESSES

E. F. COSTA Jr.[1], R. C. VIEIRA[1], A. R. SECCHI[2] and E. C. BISCAIA Jr. [1]

[1] *Programa de Engenharia Química - COPPE/UFRJ - Rio de Janeiro – RJ – Brazil*
Caixa Postal 68.502 - CEP 21.945-970 - Rio de Janeiro - Brazil
evaristo@peq.coppe.ufrj.br
[2] *Departamento de Engenharia Química – UFRGS - Porto Alegre – RS - Brazil*
arge@enq.ufrgs.br

Abstract — **The dynamic modeling of batch distillation columns frequently leads to a mixed system of differential and algebraic equations (DAEs) with differential index greater than one, and this particular feature has many implications on the resolution strategy adopted. As the number of stages and components can be arbitrarily high, those mathematical models can be large scale systems, and the analysis of the system prior to numerical resolution can be cumbersome. Additionally, the consistent initialization step can pose a nontrivial numerical task. For the numerical resolution of this model it is employed the computational code under development at PEQ/COPPE/UFRJ and DEQUI/UFRGS. This code employs Automatic Differentiation (AD) tools to perform index reduction and consistent initialization with minimum interference of the user. The resulting consistency system is solved and the numerical integration of the final index-one DAE is accomplished by means of the integration code DASSLC.**

Keywords — **Batch distillation, Index reduction, Differential-algebraic equations (DAEs), Automatic differentiation.**

I. INTRODUCTION

Distillation processes are among the most important technologies for the separation of multicomponent mixtures. Distillation can be done either in a continuous operation regime (where the mixture is continuously supplied and the products are continuously led off) or in a discontinuous regime – the so-called batch distillation process – where the apparatus is loaded with the mixture at start-up time and the different fractions are taken out one by one.

Batch operation is advantageous if: (i) the required operating capacity of a proposed facility is too small to permit continuous operation at a practical rate - pumps, boilers, piping, instrumentation and other auxiliary equipment generally have a minimum industrial operating capacity; (ii) the operating requirements of a facility fluctuates widely in characteristics of feed material as well as processing rate - batch equipment usually has considerably more operating flexibility than continuous equipment. This is the reason for the predominance of batch equipment for multiple purpose solvent recovery or pilot plant applications (Henley and Seader, 1981).

The description of a typical batch-distillation column is conveniently divided into two parts: (1) the start-up period and (2) the production period. The production period is that part of the process during which a product is withdrawn from the column. Necessary adjustments to bring the column to an operating condition that produces a distillate with a desired purity are made in the start-up period (Holland and Liapis, 1983).

The modeling of a batch distillation process gives rise to systems of coupled differential and algebraic equations, mostly presenting high index. In this contribution, the numerical resolution of such mathematical models is addressed. In section §2, the mathematical model of the start-up phase of a batch distillation column is presented. In §3 and §4, aspects related to its numerical resolution are discussed, and the computational code under development at PEQ/COPPE, in collaboration with DEQUI/UFRGS, is briefly described. In §5, numerical results are presented to prove the adequacy of the proposed approach.

II . MATHEMATICAL MODELING

In this contribution, it is simulated a batch distillation column with *np* stages and *nc* components. A simplified scheme of the equipment is shown in Fig. 1, where the stage number 0 represents the condenser and the stage np+1, the reboiler.

The main assumptions of the model are: (i) the liquid and the vapour leaving a stage are in thermodynamic equilibrium; (ii) the vapour hold-up is negligible; (iii) there is no heat loss throughout the column; (iv) the liquid and vapour phases behave as ideal mixtures; (v) the pressure on the column is constant.

Mass and energy balances for each stage are represented by the Eqn. 1 and 2. The Eqn. 3 and 4 represent the total number of moles of stage i and the chemical equilibrium relation. Equations 5 and 6 follow from the assumption of ideal behavior of liquid and vapour phases. The constraint on the molar fraction summation is enforced via Eqn. 7, and Eqn. 8 comes from the assumption of constant number of moles at each stage.

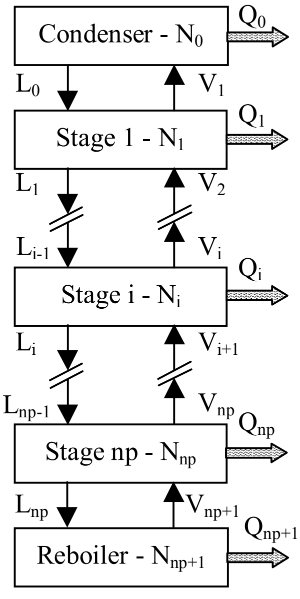

Figure 1. Scheme of the batch distillation column.

$$\dot{n}_{i,j} = V_{i+1}\ y_{i+1,j} + L_{i-1}\frac{n_{i-1,j}}{N_{i-1}} - L_i\frac{n_{i,j}}{N_i} - V_i\ y_{i,j} \quad (1)$$

$$N_i\ \dot{h}_i = V_{i+1}\ H_{i+1} + L_{i-1}\ h_{i-1} - L_i\ h_i - V_i\ H_i - Q_i \quad (2)$$

$$N_i = \sum_{j=1}^{nc} n_{i,j} \quad (3)$$

$$N_i y_{i,j} = n_{i,j}\ K_j(T_i) \quad (4)$$

$$N_i\ h_i = \sum_{j=1}^{nc} n_{i,j}\ hc_j(T_i) \quad (5)$$

$$H_i = \sum_{j=1}^{nc} y_{i,j}\ Hc_j(T_i) \quad (6)$$

$$\sum_{j=1}^{nc} y_{i,j} = 1 \quad (7)$$

$$V_{i+1} = L_i \quad (8)$$

In the previous equations, j= 1, ..., nc and i= 0, ..., np+1 except for Eqn. 8, for which i= 0, ..., np. Additionally, i represents stage, j represents component, $y_{i,j}$ represents molar fraction in vapour phase, h_i and H_i are enthalpies in liquid and vapour phases, V_i and L_i are the vapour and liquid fluxes leaving the $i^{\underline{th}}$ stage, $n_{i,j}$ is the amount of the $j^{\underline{th}}$ component in the $i^{\underline{th}}$ stage, T_i is temperature and Q_i is the heat removed from the $i^{\underline{th}}$ stage. In the simulated column, all heat duties Q_i are known except for the condenser heat duty, Q_0. Hence, the state vector is $[L_i\ V_i\ N_i\ n_{i,j}\ y_{i,j}\ T_i\ H_i\ h_i\ Q_0]^T$.

The equilibrium constant and the enthalpies of each component, needed at Eqn. 4 to 6, are function of T according to the following relations:

$$K_j(T) = T\left(\sum_{k=0}^{3} a_{j,k}\ T^k\right)^3 \quad (9)$$

$$hc_j(T) = \left(\sum_{k=0}^{2} c_{j,k}\ T^k\right)^2 \quad (10)$$

$$H_j(T) = \left(\sum_{k=0}^{2} e_{j,k}\ T^k\right)^2 \quad (11)$$

Numerical values for the constants employed in Eqn. 9 to 11 can be found in Holland and Liapis (1983, pp. 211-212), and are reproduced in Table 1. Additional constants needed for simulation of the model are the heat duty of the reboiler Q_{np+1} = - 3.5 10^6 BTU/h and the heat removed from each stage, Q_i = 0 (i= 1, ..., np).

III. CHARACTERIZATION OF THE MATHEMATICAL MODEL

The mathematical model of the start-up of a batch distillation column is typically a high-index DAE system with nonlinear state equations and equilibrium constraints. The differential equations generally come from energy and mass balances in the equilibrium stages, and the algebraic equations are flux constraints and equilibrium relationships. Additionally, the resulting DAEs can easily become a large-scale system as the number of stages and components get high.

Experience has proved that working directly with the differential-algebraic equations (DAEs) is advantageous for many reasons: (i) the systems are simpler and the variables have a physical significance; (ii) time consuming manipulations are avoided and fundamental information, that could be lost through differentiation, is preserved; (iii) constitutive relations, such as phase equilibrium relations, kinetic equations and equilibrium isotherms, are easily included and/or changed whenever necessary (Vieira and Biscaia Jr., 2001).

Table 1. Constants employed to compute equilibrium conditions.

	C_3H_8	i-C_4H_{10}	n-C_4H_{10}	i-C_5H_{12}	n-C_5H_{12}
$a_{j,0}$ x 10^2	-14.512474	-18.967651	-14.181715	-7.548840	-7.543539
$a_{j,1}$ x 10^5	53.638924	61.239667	36.866353	3.2623631	2.0584231
$a_{j,2}$ x 10^8	-5.3051604	-17.891649	16.521412	58.507340	59.138344
$a_{j,3}$ x 10^{12}	-173.58329	-90.855512	-248.23843	-414.92323	-413.12409
$c_{j,0}$	-14.500060	-16.553405	-20.298110	-23.356460	-24.37154
$c_{j,1}$	1.9802223	2.1618650	2.3005743	2.5017453	2.56362
$c_{j,2}$ x 10^5	-2.9048837	-3.1476209	-3.8663417	-4.3917897	-4.6499694
$e_{j,0}$	81.79591	147.65414	152.66798	130.96679	128.90152
$e_{j,1}$ x 10^4	389.81919	-1185.2942	-1153.4842	-197.98604	-2.0509603
$e_{j,2}$ x 10^6	36.4709	152.87778	146.64125	82.549947	64.501496

When solving DAEs one must be concerned about the index of the system and about the consistency of the initial conditions. The index of a DAE system is the number of times that all or part of the system must be differentiated with respect to time in order to convert it into an explicit set of first order ODEs. Index-1 systems may be solved with modified ODE methods, while high-index systems (systems with index 2 or greater) require specific methods. Generally, high-index systems are rewritten in an index-1 formulation and solved with standard integration codes like DASSL (Petzold, 1989) or DASSLC (Secchi, 1992), written in FORTRAN and C, respectively.

During the index reduction, some extra algebraic equations are obtained, which generally correspond to derivatives of the original algebraic equations. Those hidden algebraic equations along with the original DAEs compose the extended system. Consistent initial values must satisfy not only the DAE system but also the underlying extended system (Brenan *et al*., 1989).

Several rigorous methods for initialization have been proposed in the literature, and they all depend to some extent on the characterization of the DAE system. In other words, in order to perform the numerical integration of a DAE system with standard integration codes, it is necessary to: (i) perform the characterization of the DAE system, in order to determine its index and, if required, the set of equations to be differentiated; (ii) perform the differentiation; (iii) check for feasibility of the initial states arbitrarily set; (iv) solve the consistency system for a consistent initial state.

The mathematical modeling of the batch distillation column gives rise to a differential algebraic (DAE) system: Eqn. 1 and 2 are differential equations, while the remaining equations are algebraic. The differential variables of this system are $n_{i,j}$ and h_i and the algebraic variables are N_i, $y_{i,j}$, H_i, T_i, V_i, Q_0 and L_i. Hence, the dimension of the system is (np+2)(2nc+6)-1. It can be noticed that the algebraic variable Q_0 does not appear in any algebraic equation. Additionally, the 2(np+1) algebraic variables V_i e L_i only appear at (np+1) equations (represented as Eqn. 8). Hence, the DAE system presents high index.

Consistent initialization and index reduction techniques are to be considered prior to numerical resolution of the batch distillation model with standard integration tools, as DASSL or DASSLC. In the Appendix, a procedure for index reduction is presented, which requires several steps of differentiation and algebraic manipulation. The main purpose of the developed code is to aid the researchers, so that they could avoid performing such cumbersome, tedious and error prone algebraic manipulations.

It should be pointed out that the procedure in the Appendix is not the one actually implemented on the code, but instead, the one that a researcher could carry out when dealing with this high-index DAE model. The main difference between the approaches is that a person can substitute an equation into another during the index reduction procedure. The code, on the other hand, includes all the generated equations and variables on the extended system, increasing its dimension.

IV. NUMERICAL RESOLUTION

In the present contribution the characterization/initialization module described by Costa Jr. *et al.* (2001) is used. Based on Pantelides' algorithm (Pantelides, 1988), it uses a graph-theoretical approach to identify the minimum set of equations to be differentiated in order to generate an index-one formulation of the original DAEs. Through AD code ADOL-C (Griewank *et al*., 1996), the differentiation is performed automatically at an affordable computational cost, and, most important of all, the differentiation technique employed does not incur truncation error. The required structural pattern of the jacobian matrix is computed via numerical perturbation of the original equations and of the exact extended system.

The user must supply the DAE system F(z,z')=0 according to DASSLC standards. Then, the code applies Pantelides' algorithm, constructs the extended system and informs the number r of degrees of freedom of the system. The user is prompted to give r arbitrary initial conditions, on which the code will perform a structural check for feasibility. It must be stressed that structural feasibility is a necessary but not sufficient condition for feasibility. That comes from the fact that structural computation cannot determine if a structurally regular matrix is not in fact a singular matrix.

In the present code, when an initial arbitrary set is found to be structurally feasible, the code will carry out the resolution of the extended system by a Newton-type method. The consistency system was constructed via automatic differentiation, and hence it is exact. The initial conditions obtained from its resolution can be provided directly to the DASSLC numerical integration code. If the initial conditions are unfeasible, the user is warned and prompted to supply another arbitrary set.

V. SIMULATION RESULTS

Using the present code, it is possible to integrate high-index DAE systems directly using DASSLC code. All index reductions and manipulation, needed in order to achieve consistent initialization, are performed automatically. The user must only give the original DAEs according to DASSLC standards, determine which are the arbitrarily set initial variables (based on the calculated number of degrees of freedom) and provide initial values for those variables.

Characterization results for an example with 5 components and 12 stages are shown in Table 2. The feasible arbitrary initial set used in this example corresponds to all the variables $n_{i,j}$ (i= 0, ..., 13 and j= 1, ..., 5), performing 70 state variables. Table 3 presents the numerical values actually set for $n_{i,j}$, and Table 4 presents the initial conditions calculated by the code for the remaining variables.

Table 2. Characterization results.

Nc=5, *np*=12
Index 2
Eqn. 3, 4 ,5 and 7 must be differentiated once
Degrees of freedom: 70 [*nc* (*np*+2)]

After the differentiation of the equations indicated in Table 2, the extended system presents 335 equations and 405 variables, 70 of which must be assigned arbitrarily. It should be pointed out that the original system has 223 equations and 307 unknowns, among state variables (223) and some of their derivatives (84). The 405-307=98 new variables represent derivatives of former algebraic variables ($\dot{N}_i$, $\dot{y}_{i,j}$ and $\dot{T}_i$). In a more general case, they could also comprise some higher order derivatives of differential variables.

Table3. Initial conditions arbitrarily set.

Component	$n_{0,j}$	$n_{i,j}$	$n_{np+1,j}$

	j= 1, ..., nc	i= 1, ..., np
		j= 1, ..., nc
			j= 1, ..., nc

C_3H_8	0.2	0.05	1.7
i-C_4H_{10}	0.6	0.15	5.1
n-C_4H_{10}	1	0.25	8.5
i-C_5H_{12}	0.8	0.2	6.8
n-C_5H_{12}	1.4	0.35	11.9

Table 4. Computed initial conditions.

T_i = 729.1416 R	N_0 = 4 moles
Q_0 = 3.5 10^6 Btu/h	N_{13} = 34 moles
H_i = 22304.66 Btu/mol	$y_{i,0}$ = 0.124241
$h_{i,j}$ = 17331.76 Btu/mol	$y_{i,1}$ = 0.221436
L_i = 47.69237 moles/h	$y_{i,2}$ = 0.304075
V_i = 47.69237 moles/h	$y_{i,3}$ = 0.139399
N_i = 1 mol (i= 1, ..., 12)	$y_{i,4}$ = 0.210850

It should be emphasized that other sets could be determined arbitrarily. There are $_{405}C_{70}$ (approximately 10^{80}) possible sets to be assigned, and only some of them are feasible. It is for this reason that the code asks the user to inform the desired initial set to be tested for feasibility. In order to illustrate this idea, some sets other than the one used in this contribution have been verified for structural feasibility. The results of this analysis are presented in Table 5.

Table 5. Some sets verified for structural feasibility.

Variables Set	Structural Feasibility
All $n_{i,j}$	Feasible (set used for simulation)
All $y_{i,j}$	Unfeasible (eliminates Eqn.7)
All $y_{4,j}$ h_i, H_i, T_i, L_i and Q_0	Feasible
All h_i, H_i, T_i, L_i, N_i and Q_0	Feasible

The simulation results are shown in the next figures. Figures 2 to 4 show the mole fraction of the components in the condenser, in plate 6 and in the reboiler. These results can be compared with the ones presented by Holland and Liapis (1983) for the same model, and it can be noticed that reported results have been reproduced. The direct resolution of the DAE model arising from the mathematical modeling of the process eliminates the algebraic manipulation step to the researcher. With the present code, even higher index problems can be solved with minor intervention of the user.

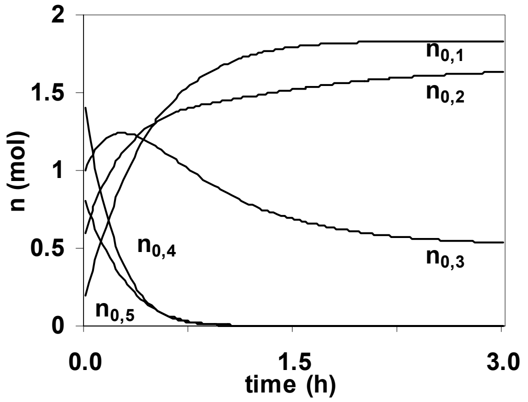

Figure 2. Number of moles of each component in the condenser.

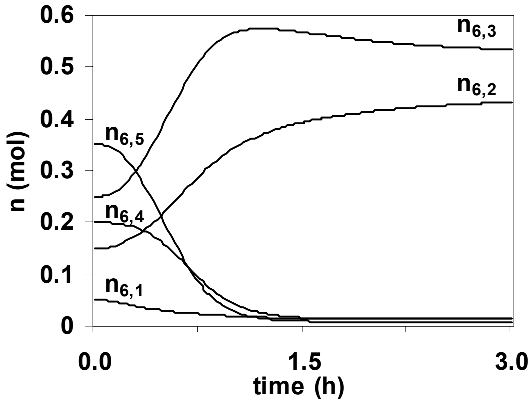

Figure 3. Number of moles of each component in the intermediate plate.

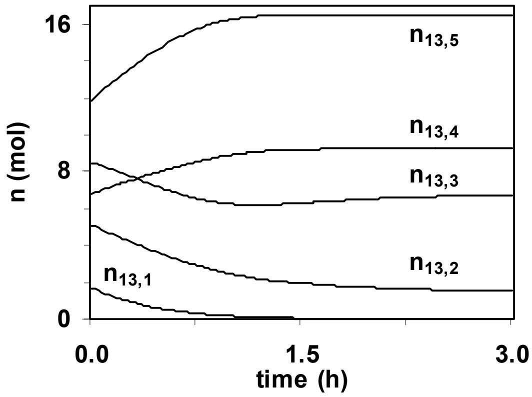

Figure 4. Number of moles of each component in the reboiler.

In Fig. 5, it is shown the liquid flowrate leaving the condenser (L_0). It follows from the constant mole number per stage assumption that all L_i variables have equal values.

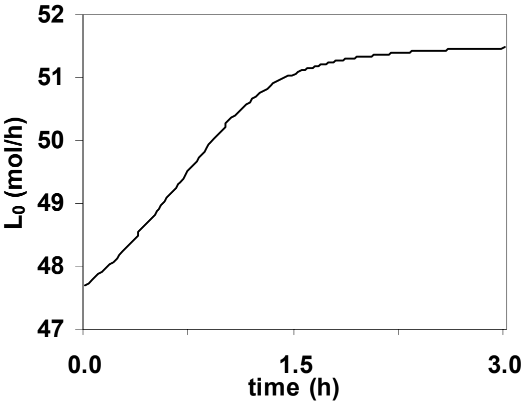

Figure 5. Liquid flowrate leaving the condenser L_0.

In Fig. 6, it is presented the heat duty of the condenser. It can be seen that the steady state value of Q_0 is equal to the heat duty of the reboiler, which is coherent with the consideration of negligible heat loss throughout the column. Figures 7 and 8 show the behavior of the temperature and of the enthalpy of liquid and vapour phases in the condenser, in the reboiler and in the sixth stage of the column.

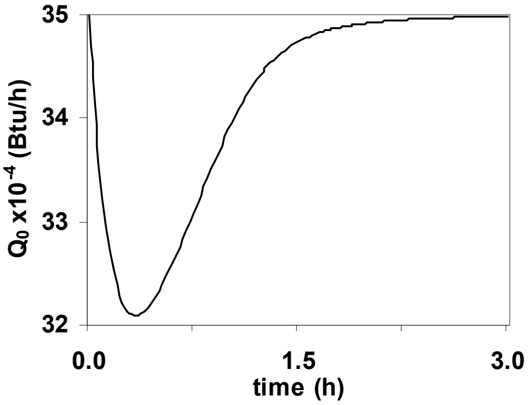

Figure 6. Condenser heat duty.

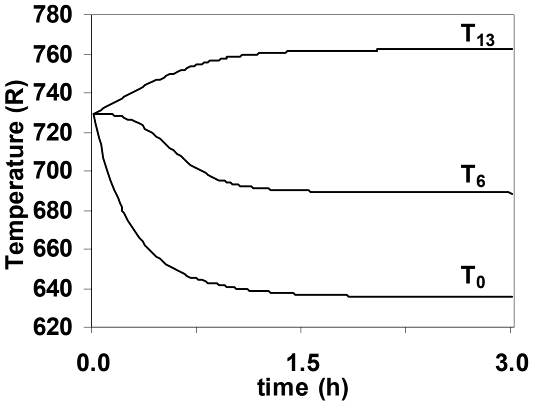

Figure 7. Temperature profiles in a few stages.

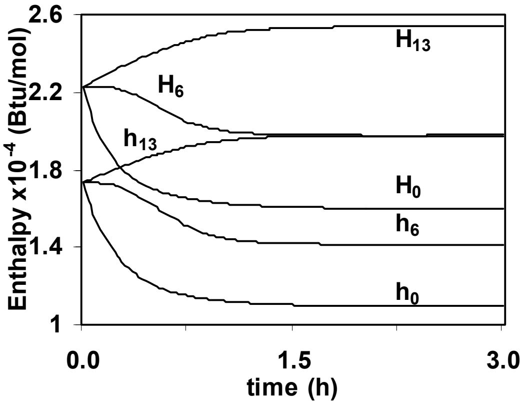

Figure 8. Enthalpy of liquid and vapour phases.

VI. CONCLUSIONS

The optimized operation and the control of a transient distillation process require the numerical resolution of the mathematical model representative of the process, a DAE system. On the presented formulation, the model presents high index. Characterization of the system, index reduction and consistent initialization are issues to be dealt with prior to the numerical resolution of the model. The drawbacks of this formulation may be overcome using the developed characterization/initialization module. This module has been coupled with the DASSLC code for numerical integration of index 1 DAEs. As a consequence, high-index DAE systems can be simulated with minor intervention of the user.

In this contribution, the start-up of a batch distillation process has been analyzed. Numerical results agree with reported profiles for concentration in the reboiler and in the condenser.

The user must supply the DAE system according to DASSLC standards, choose r variables to be assigned arbitrarily at t=0 and provide numerical values for those variables. The code automatically checks for feasibility of the chosen set, computes a complete initial set of initial conditions and allows the numerical integration of the resulting DAE system by existing numerical codes. The automatic differentiation tool ADOL-C has been utilized to perform all the required differentiation.

It should be pointed out that the direct resolution of high-index systems makes it feasible to perform the numerical resolution of DAE models of arbitrary index and dimension. Consequently, it encourages the development of more detailed mathematical models.

REFERENCES

Brenan, K., Campbell, S. , Petzold, L., *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, New York, Elsevier Sc. Publ. Co. (1989).

Costa Jr., E.F., Vieira, R.C., Secchi, A.R., Biscaia, E.C., "Automatic Structural Characterization of DAE Systems"*, Proceedings of 11th ESCAPE*, Denmark (2001)

Griewank, A., Juedes, D., Mitev, H., Utke, J., Vogel, O., Walther, A., “ADOL-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++”, ACM TOMS 22, 131-167 (1996).

Henley, E.J., Seader, J.D., *Equilibrium-Stage Separation Operations in Chemical Engineering*, New York , John Wiley & Sons Inc. (1981).

Holland, C., Liapis, A.I., *Computer Methods for Solving Dynamic Separation Problems*, McGraw Hill Publishers (1983).

Pantelides, C.C., “The Consistent Initialization of Differential-Algebraic Systems”, *SIAM Journal of Scientific and Statistical Computing,* **9,** 213-231 (1988).

Petzold, L.R., “DASSL code, version 1989”, Computing and Mathematics Research Division, Lawrence Livermore National Laboratory, L316, PO Box 808, Livermore, CA 94559 (1989).

Secchi, A.R., “DASSLC: User's Manual, a Differential-Algebraic System Solver”, Technical Report, http://www.enq.ufrgs.br/enqlib/numeric/DASSLC, UFRGS, Porto Alegre, RS/Brazil (1992).

Vieira, R.C., Biscaia Jr., E.C., “Direct Methods for Consistent Initialization of DAE Systems”, *Comp. Chem. Eng.,* **25**, 1299-1311 (2001).

APPENDIX - Reducing the Index of the DAE Model

In order to reduce the index of the DAE system comprised of Eqn. 1 to 8, the following procedure could be used. Equation 4 can be rewritten as:

$$y_{i,j} = \frac{n_{i,j}K(T_i)}{N_i}$$

Replacing $y_{i,j}$ by the right hand side of this expression at Eqn. 7, it follows that:

$$\Sigma_{j=1}^{nc} \frac{n_{i,j}K(T_i)}{N_i} = 1 \qquad (12)$$

Differentiating Eqn. 12, 5 and 3 for the first time, the following expressions are obtained, respectively.

$$\dot{T}_i = g_1(n_{i,j}, T_i, N_i, \dot{n}_{i,j}, \dot{N}_i) \qquad (13)$$

$$\dot{h}_i = g_2(h_i, n_{i,j}, N_i, T_i, \dot{n}_{i,j}, \dot{N}_i, \dot{T}_i) \qquad (14)$$

$$\dot{N}_i = \Sigma_{j=1}^{nc} \dot{n}_{i,j} \qquad (15)$$

Substituting Eqn. 13 into 14, it is obtained:

$$\dot{h}_i = g_3(h_i, n_{i,j}, N_i, T_i, \dot{n}_{i,j}, \dot{N}_i) \qquad (16)$$

Substituting Eqn.15 into 16, it is obtained:

$$\dot{h}_i = g_4(h_i, n_{i,j}, N_i, T_i, \dot{n}_{i,j}) \qquad (17)$$

Substituting Eqn.1 into 17, it is obtained:

$$\begin{aligned} \dot{h}_i = g_5(&h_i, n_{i,j}, n_{i-1,j}, y_{i,j}, y_{i+1,j}, N_i, \\ &N_{i-1}, T_i, L_i, L_{i-1}, V_i, V_{i+1}) \end{aligned} \qquad (18)$$

Substituting Eqn.18 into 2, it is obtained:

$$\begin{aligned} g_6(&h_{i-1}, h_i, n_{i,j}, n_{i-1,j}, y_{i,j}, y_{i+1,j}, N_i, N_{i-1}, \\ &T_i, L_i, L_{i-1}, V_i, V_{i+1}, H_i, H_{i+1}, Q_0) = 0 \end{aligned} \qquad (19)$$

Substituting Eqn.8 into 19, it is obtained:

$$\begin{aligned} g_6(&h_{i-1}, h_i, n_{i,j}, n_{i-1,j}, y_{i,j}, y_{i+1,j}, \\ &N_i, N_{i-1}, T_i, L_i, L_{i-1}, H_i, H_{i+1}, Q_0) = 0 \end{aligned} \qquad (20)$$

i= 0, ..., np+1

The algebraic variable Q_0 and the np+1 molar fluxes L_i are present at these np+2 equations (represented as Eqn. 20). In the above equations, variables with index i-1 and i+1 do not exist when i = 0 and i = np+1, respectively. Via this procedure, it has been built an index 1 system, composed of Eqn. 1-6 coupled with Eqn. 8 and 20. This system can be integrated with a standard code as DASSL or DASSLC. As only one differentiation of each equation has been necessary to build an index 1 system, the original system presents index 2.

Received: September 16, 2001.
Accepted for publication: August 29, 2002.
Recommended by Guest Editors J. Cerdá, S. Díaz and A. Bandoni.