17th European Symposium on Computer Aided Process Engineering – ESCAPE17
V. Plesu and P.S. Agachi (Editors)

Debugging for Equation-Oriented CAPE Tools

Rafael de Pelegrini Soares, Argimiro R. Secchi

Universidade Federal do Rio Grande do Sul, Departamento de Engenharia Química,
Rua Sarmento Leite, 288/24, CEP: 90050-170, Porto Alegre, RS – Brazil
E-mail: rafael@enq.ufrgs.br, arge@enq.ufrgs.br

Abstract

Regarding CAPE tools, a moving from modular oriented, which is currently the most widely used technique, to Equation-Oriented (EO) is clear. One of the key advantages of the EO approach is that the effort spent in model development is minimized by reusing the models in several different tasks, for instance: simulation, optimization, and data reconciliation. EO tools support the implementation of models to a large extent, however there is almost no assistance in the model development process. In this work the currently available methods for detecting inconsistencies in system of equations coming from both static and dynamic models are reviewed and extended. The proposed algorithm is scalable for large problems and is a promising diagnosis tool to spread the usage of EO dynamic simulators. Finally, it is presented how these techniques scale for complex problems.

Keywords

Structural Analysis, Debug, NLA, DAE index

1. Introduction

The current process simulators may roughly be classified into two groups: modular and equation-oriented [1]. In the present work this distinction is not referred to the model building tool but to the method employed to obtain the solution.

In modular tools the models of process units are pre-coded in a programming language by a modelling expert and incorporated in a model library. The end user selects the models from the library and connects them to form the plant model. The incorporated chemical engineering knowledge as well as the model structure are largely fixed and not accessible [6].

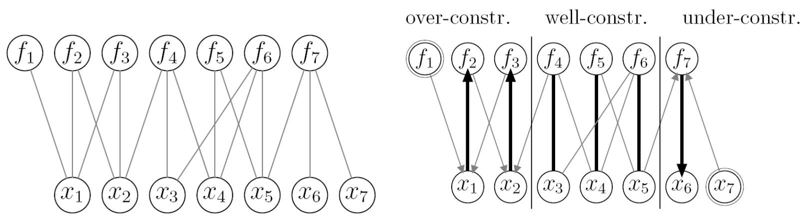
In equation-oriented (EO) or equation-based implementations the equipment models are written in some descriptive or modelling language and usually are opened for visualization and extension. These models share with the plant model their equations and not only their numerical solution. As a consequence, the implementation of unit models is independent of any particular application or algorithm that may be used for their solution. Recognition of potential benefits of EO technology has led to the development of several tools. Examples of implementations are gPROMS [7] and EMSO [10].

On the other hand when using an EO tool the user needs to have at least a minimal knowledge of the model internals in order to estimate which variables can be fixed to close the degrees of freedom. For dynamic models the situation can be even worst because the same problems appear for the initial conditions. From the end user perspective, these aspects makes EO simulators harder to use. In this work, methods for diagnosing ill-posed models coming from EO tools are reviewed and extended. Making an analogy with software development, the methods which aid to detect and remove problems of the models are called debug.

2. Nonlinear Systems

Nonlinear algebraic (NLA) equations appear in the solution of steady-state simulations of EO simulators. Using *graphs* [3] the NLA system Eq. (1) can be drawn as the bipartite graph shown in Fig. 1 [2].

$$\begin{array}{ll} f_1(x_1)=0 & f_5(x_4,x_5)=0 \\ f_2(x_1,x_2)=0 & f_6(x_3,x_4,x_5)=0 \\ f_3(x_2,x_2)=0 & f_7(x_5,x_6,x_7)=0 \\ f_4(x_2,x_3,x_4)=0 & \end{array} \quad (1)$$



(a) (b)

Fig 1. Graph for the NLA system Eq. (3) (a) and one maximum matching for it (b).

As can be seen in Fig. 1 (a) the values or form of the functions in Eq. (1) are irrelevant, only the relation equation-variable is considered.

2.1. Debugging NLA Systems

Even systems with zero degrees of freedom still can be inconsistent, this is the case of Eq. (1). Using a maximum matching algorithm [9] the structural singularity of the system can be easily checked. One maximum matching association for Eq. (1) can be seen in Fig. 1 (b). In this figure, the edges which are part of the matching are shown in *bold* and nodes not covered by the association are *marked*. If a maximum matching association includes all variables and all equations (a perfect matching) then the system is structurally non-singular.

As can be seen the maximum matching check goes beyond degrees of freedom analysis. However, it cannot be used as an assistant tool for fixing the problem because the source of the problem is still shadowed. One step further can be achieved using the DM decomposition [5]. This method canonically decomposes any maximum matching into three distinct parts: over-constrained, under-constrained, and well-constrained, as shown in Fig. 1 (b).

From Fig. 1 (b) a debugging tool can conclude that one of the equations $\{f_1, f_2, f_3\}$ need to be removed and one additional equation involving x_6 or x_7 need to be added. Obviously for the end user this kind of message is much more interesting than a numerical solution failure.

3. Differential-Algebraic Systems

Differential-Algebraic Equation (DAE) systems arise naturally when dealing with dynamic simulation in EO tools. Historically the analysis of this kind of problem was limited to degrees of freedom and index analysis, see [4, 8, 11].

Today, the algorithm developed by Pantelides [8] is the most widely used structural analysis technique for DAE problems. The main objective of that work was to determine the number of initial conditions required to the consistent initialization. In other words, to check the number of dynamic degrees of freedom. Unfortunately, as stated in the case of NLA problems, debugging requires more than just a degrees of freedom check.

3.1. New DAE Analysis Algorithm

DAE systems also can be represented as bipartite graphs. But in the dynamic case there are two new concepts: the time derivatives of the variables are also considered and the equations can be differentiated inserting new elements into the graph.

The new algorithm for analysis of DAE systems consists in the following steps:

1. Find a maximum matching association considering only the algebraic variables;
2. If the association includes all equations then the algorithm finished;

3. Find a maximum matching association including all variables. If this association does not include all equations then the system is singular and the algorithm ends;
4. Differentiate the equations connected with algebraic variables and go back to 1.

Unfortunately there is no room for a formal presentation of the algorithm but it could be more easily understood with an application. For instance, consider the following system of equations:

$$x_1' + x_2' = a(t) \quad x_2 = b(t) \tag{2}$$

Applying the first three steps of the algorithm on Eq. (2) the graph shown in Fig. 2 (a) is obtained.

As stated in step 4, equations connected with algebraic variables (marked in Fig. 2 (a)) need to be differentiated. After the differentiation the algorithm will finish on step 2, and the resulting graph can be seen in Fig. 2 (b).

The main advantage of the new algorithm is that in association with the DM decomposition it can be used for debugging purposes. For instance, the under-constrained partition will reveal all variables which can be supplied as initial conditions. Taking the Eq. (2), the under-constrained partition includes only x_1. Using this information, an EO tool can tell to the end user that the only option for this model is to supply an initial value for x_1 and the other variables $\{x_1', x_2, x_2'\}$ are discarded from the initial conditions candidates.

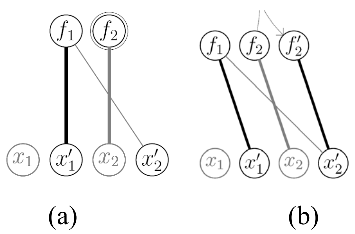


Fig. 2. Graph for Eq. (6) after the first three steps (a) and when the algorithm finishes (b).

It should be noted that the proposed algorithm always finish while the Pantelides algorithm runs indefinitely for some singular problems [8]. Furthermore, it can be applied without modifications to analyze high-index systems. The equations differentiated by the algorithm can also be used to generate index-reduced systems.

4. Applications

In order to check how the new algorithm for DAE analysis performs for large scale problems a dynamic model for distillation processes was analyzed. This model has mass and energy balances for each tray besides thermodynamics and hydrodynamics equations. The computational time required to analyze the dynamic model for the separation of isobutane in a mixture of 13 compounds with different number of trays can be seen in Table 1.

Table 1. Time to analyze the dynamic model of a distillation column varying the number of trays.

Trays	Variables	Time (s)	Time/N^2 ($s.10^9$)
20	2157	0.04	9.46
40	3877	0.14	9.58
80	7317	0.52	9.79

The results shown in Table 1 were obtained in a Pentium M 1.70 GHz with 2 MB of cache memory running Ubuntu Linux version 6.06. As can be seen in that table, the performance is approximately quadratic as are the majority of the solution methods.
Another good result is that the time required by the analysis is very acceptable for user interaction. Moreover, the algorithm can be applied incrementally adding new equations and variables as the user interacts with the modelling environment. This fact can broke the analysis time, making the software more responsible to the end user.
In order to show how sensible the analysis algorithms can be, consider an ammonia synthesis process as shown in Fig. 3.

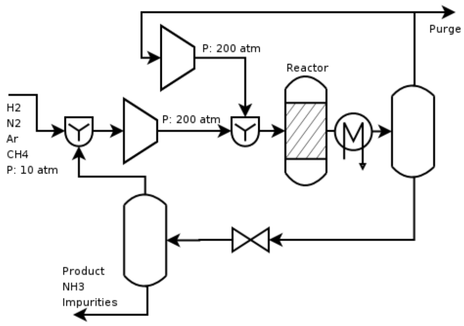


Fig. 3. Ammonia synthesis process diagram.

A static model with 134 variables for the process in Fig. 3 was constructed. If all specifications are supplied correctly the maximum matching algorithm finishes with a perfect matching. But if one specification is missing, for instance the feed flow rate, then the under-constrained partition will involve 96 variables. This means that the well-constrained partition covers only about of 30% of the variables. This large number of fixing options is the major deficiency of the presented methods. In order to fix it, heuristic rules for ranking the fixing options are being studied.

5. Conclusions

In this work, methods for user assistance when developing models in EO tools were presented. Techniques which aid in the location and removal of inconsistencies of the models were called debugging methods. For static models (NLA systems) consolidated methods were found on the literature and were reviewed. Unfortunately, the implementation of these methods is still missing on commercial EO tools.

For the dynamic case (DAE systems) a very less mature context were found. Historically, the analysis of such systems was limited to degrees of freedom and index analysis. A new method for structural analysis of DAE systems was proposed. The key advantage of this method is that it can be used for debugging purposes. Furthermore, the algorithms are being incorporated in the EMSO [10] process simulator.

References

1. J. F. Boston, H. I. Britt, and M. T. Tayyabkhan. Tackling tougher tasks. Chemical engineering progress, 89(11):38–49, 1993.
2. Peter Bunus. Debugging and Structural Analysis of Declarative Equation-Based Languages. PhD thesis, Department of Computer and Information, Science Linkpings Universitet, Linkping, Sweden, 2002.
3. Reinhard Diestel. Graph theory. Springer-Verlag, New York, 2 edition, 2000.
4. I. S. Duff and C. W. Gear. Computing the structural index. SIAM Journal on Algebraic and Discrete Methods, 7(4):594–603, 1986.
5. A. L. Dulmage and N. S. Mendelsohn. Coverings of bipartite graphs. Canad. J. Math., (10):517–534, 1958.
6. W. Marquardt. Trends in computer-aided process modeling. Computers & Chemical Engineering, 20(6):591–609, 1996.
7. M. Oh and C. C. Pantelides. A modelling and simulation language for combined lumped and distributed parameter systems. Computers & Chemical Engineering, 20:611–633, 1996.
8. C. C. Pantelides. The consistent initialization of differential-algebraic systems. SIAM J. Sci. Stat. Comp., 9(2):213–231, March 1988.
9. H. Saip and C. Lucchesi. Matching algorithms for bipartite graph. Technical Report DCC-03/93, Depto. de Ciência da Computção, Universidade Estudal de Campinas, Brazil, 1993.
10. R. P. Soares and A. R. Secchi. EMSO: A new environment for modelling, simulation, and optimisation. In ESCAPE 13th, volume 1, pages 947–952. Elsevier Science Publishers, 2003.
11 J. Unger, A. Kroner, and W. Marquardt. Structural analysis of differentialalgebraic equation systems–theory and applications. Computers & Chemical Engineering, 19(8):867–882, August 1995.