DEBUGGING FOR EQUATION-ORIENTED CAPE TOOLS

Rafael de P. Soares, Argimiro R. Secchi

1Grupo de Integração, Modelagem, Simulação, Controle e Otimização de Processos (GIMSCOP)
Departamento de Engenharia Química, Universidade Federal do Rio Grande do Sul (UFRGS)
R. Eng. Luis Englert, s/n. Campus Central. CEP: 90040-040 - Porto Alegre - RS - BRASIL,
E-MAIL: {rafael, arge}@enq.ufrgs.br

Palavras Chaves: Structural Analysis, Debug, DAE

Resumo: Regarding CAPE tools, a moving from modular oriented, which is currently the most widely used technique, to equation oriented is clear. One of the key advantages of the equation oriented or simultaneous approach is that the effort spent in model development can be reused in several different tasks, for instance: simulation, optimization, parameter estimation, and data reconciliation. This fact avoids modeling rework for each application which can be costly and time-consuming. Equation-oriented tools support the implementation of models to a large extent, however there is almost no assistant in the model development process. Even when developing models based on a ready-made well-designed library of models the user suffers in closing the degrees of freedom. The situation can be even worst in the dynamic case where the user should provide initial conditions in a number equal to the dynamic degrees of freedom. As a consequence, modeling has been the bottleneck of the widespread of equation-oriented tools in the industrial practice and not numerical algorithms. Making an analogy with general software development, in this work the techniques which aids in the process of removing inconsistencies of models are called debugging techniques. In this work the currently available methods for detecting inconsistencies of both static and dynamic models are reviewed. Historically the analysis of dynamic models is limited to the determination of the dynamic degrees of freedom and the index problem. Fortunately, the well developed debugging methods originally developed for the static case can be adapted to the dynamic case. In this work the algorithms needed for this adaptation are presented. Finally, the applications of the debugging techniques in test cases for both static and dynamic simulation are used to illustrate the proposed algorithms.