18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

Stiffness reduction of complex non-linear models and procedure to maintain solution quality

Garneau, C.[1], D. Batstone[2], F.H.A. Claeys[3], P. A. Vanrolleghem[1]

[1]modelEAU, Université Laval, Québec, QC, G1K 7P4, Canada
[2]Advanced Water Management Centre, University of Queensland, Brisbane, Australia
[3]MOSTforWATER NV, Sint-Sebastiaanslaan 3a, B-8500 Kortrijk, Belgium

Email: cyril.garneau.1@ulaval.ca

Wastewater treatment models consisting of large sets of non-linear ODE are usually stiff. Because stiff solvers cannot typically be applied, due to dynamic inputs, long computation times result. To limit the computational burden, model reduction, e.g. by linearization or by singular perturbation, has been applied on individual cases with good results, but there are no generalised methods to apply this in DAE systems. We therefore developed a method to improve the efficiency of a Diagonally Implicit Runge-Kutta (DIRK) DAE solver. The method consists of reducing the number of differential equations in the model by transforming some of them into algebraic equations, i.e. assuming instantaneous equilibrium. The Homotopy method is used to link the state variables to the large eigenvalues of the Jacobian (i.e. the fast dynamics). By solving these "stiff" state equations algebraically, important improvements in calculation time can be achieved. However, several practical issues remain. In particular, it is important to confirm that the reduction of the original model does not induce instability or unacceptable error. Control of instability is achieved by comparing the solution of five simulated steps computed with the reduced model to the solution of a single step of equivalent time computed with the full model. Since implicit Runge-Kutta methods show highly stable response, the full step can be considered as a converging estimate of the true solution, and thus a comparison point to detect instability. In case of unacceptable error (i.e. instability detected), the five simulated steps are rejected and the original model is used. An unacceptable error typically arises when a stiff state variable loses its stiffness after one step. This is common when states are influenced by non-linear equations in themselves, as apparent eigenvalues can drop dramatically when the state moves towards equilibrium. In such a case, the variation of the state variable away from equilibrium will be too large, missing potentially important events. The eigenvalue related to the state will pass from a very high eigenvalue to a much lower eigenvalue after a single simulated time step. To detect such a case, eigenvalues are recomputed after one time step and the number of "large" eigenvalues is compared before and after the time step. In case of a modification, the step is rejected and recalculated with the original model. Results have shown that improvements up to 45% in the number of function evaluations can be observed. Best improvements have been observed at more stringent error tolerances. Stability and error control have shown promising results, but the model reduction induces visible changes in intermediate values of stiff state variables. Caution must be used if results need high precision in all state variables.

***Keywords**: ODE, DIRK, model reduction, error control.*

1. INTRODUCTION

Numerical simulation is a computationally intensive process. In the case of wastewater treatment plant simulation, this process usually takes minutes to hours of computation. The computation time is determined by the complexity of the model to simulate, stiffness and structure of the model, and the suitability of the solver used to compute the solution of the simulation.

A wide variety of solvers exists in literature, ranging from the simplest Euler method to more complex Runge-Kutta or predictor-correctors methods. Extensive libraries can be found in software like MATLAB (Shampine and Reichelt 1997). Depending of the model, the time needed to complete a simulation run can be improved by many orders of magnitude just by selecting the best solver for a specific problem.

It is also possible to speed up simulation by redefining a model. In the case of wastewater treatment plant models, for instance, simulations are usually run to simulate the plant's behaviour for periods up to a year. A common level of output detail is one simulated data point every fifteen minutes. In these conditions, information on dynamics occurring over seconds are less important, e.g., if chemical reactions are involved. These fast dynamics complicate the solution, as they cause the model to become stiff (i.e., containing eigenvalues varying by several orders of magnitude). Stiff problems can only be efficiently solved by stiff solvers (such as backward Euler, and DASSL), which are generally incompatible with dynamic inputs, controller transitions, or exogenous noise. It is normally better to restructure the model to reduce stiffness. A well-known way to deal with fast state variables is to solve them as if they were at steady state. This method has been used by (Hesstvedt et al. 1978) for air pollution modeling, although the speed of reaction is described by the lifetime of the compound. (Steffens et al. 1997) use a Homotopy method to determine which state equations are responsible for stiffness in a biological wastewater treatment model and then reduce the model accordingly. At this moment, transforming a set of ODE into a smaller set of ODE and a set of algebraic equations is done manually by an expert of the system that is modelled. This approach has been applied by many to deal with the stiffness of the pH for instance (ADM1 (Rosen et al. 2006), RWQM1 (Vanrolleghem et al. 2001)).

In this paper, an automatic reduction is proposed based on the Homotopy method that can be implemented in the integrator solver rather than in the model definition. Even though the principle of reduction is well understood, mechanisms must be set to ensure stability of the algorithm. This paper presents two possible mechanisms that have shown a good compromise between stability of the algorithm and performance.

Since the reduction is performed at the level of the integrator rather than on the model, an appropriate integrator has to be chosen as a base. The Diagonally-Implicit Runge-Kutta (DIRK) developed by (Cameron 1983) was chosen to take advantage of its ability of to solve Differential Algebraic Equations (DAE), its tolerance for exogenous inputs, and for its strong stability properties across a broad range of model eigenvalues.

2. METHODOLOGY

2.1. Stiffness

The stiffness of a model can take many definitions. One of the most accepted ones is the ratio of the largest eigenvalue of the Jacobian divided by the smallest one:

$$stiffness = \frac{\lambda_{min}}{\lambda_{max}}$$

The transitional time constants can be calculated from the eigenvalues at a given state as follows:

$$\tau = -\frac{1}{\lambda}$$

The eigenvalues of a model can therefore be related to the rate of change of the state variables. This relation thus gives a basis to determine whether components of a state equation can be considered fast or not.

Wastewater models are generally stable systems, and all the situations dealt with here have uniformly negative eigenvalues.

2.2. DIRK algorithm

The Variable-step Variable-order DIRK algorithm (Cameron 1983) shows very strong stability in its solution, being A-stable for orders 2 to 4 and L-stable when the order is set to 3. A-stability occurs when

$$\frac{x_{n+1}}{x_n} \to -1 \quad \text{as} \quad \lambda h \to -\infty$$

where λ is the maximal eigenvalue of the model and h is the time step. In other words, the solution is stable for any negative eigenvalue.

L-stability is defined by:

$$\frac{x_{n+1}}{x_n} \to 0 \quad \text{as} \quad \lambda h \to -\infty$$

In this case, solution will show an asymptotic behaviour. Stability concerns are well explained in (Bui 1979).

Moreover, the implementation proposed by (Cameron 1983) allows solution of differential-algebraic equation (DAE) systems. This is a valuable property since the state equations considered fast are to be solved as algebraic equations.

However, it is important to remember that the DIRK algorithm is an implicit method. Hence, the algorithm needs to solve for n variables at each step. The solution is computed with a Newton-Raphson method. Since old values for y_n are usually close to y_{n+1}, Newton-Raphson performs well in general. But it still needs a Jacobian and the solution of an LU decomposition at each iteration of the algorithm.

2.3. Homotopy method

The homotopy method used to link eigenvalues of the Jacobian matrix (*Jac*) of a dynamic model to the state variables of this model was first developed by (Steffens et al. 1997). It consists of computing the eigenvalues of the diagonal of the Jacobian (*DJac*). At this point, it is trivial to link each eigenvalue to its corresponding state variable. The matrix then receives an increasing contribution from the Jacobian through the following relation:

$$H = r * DJac + (1 - r) * Jac$$

where H is the homotopy matrix, *DJac* is the diagonal Jacobian and *Jac* is the full Jacobian. The homotopy parameter is r and will be varied from 0 to 1.

The **Figure 1** shows the trace of the sorted eigenvalues as the parameter r is increased almost continuously.

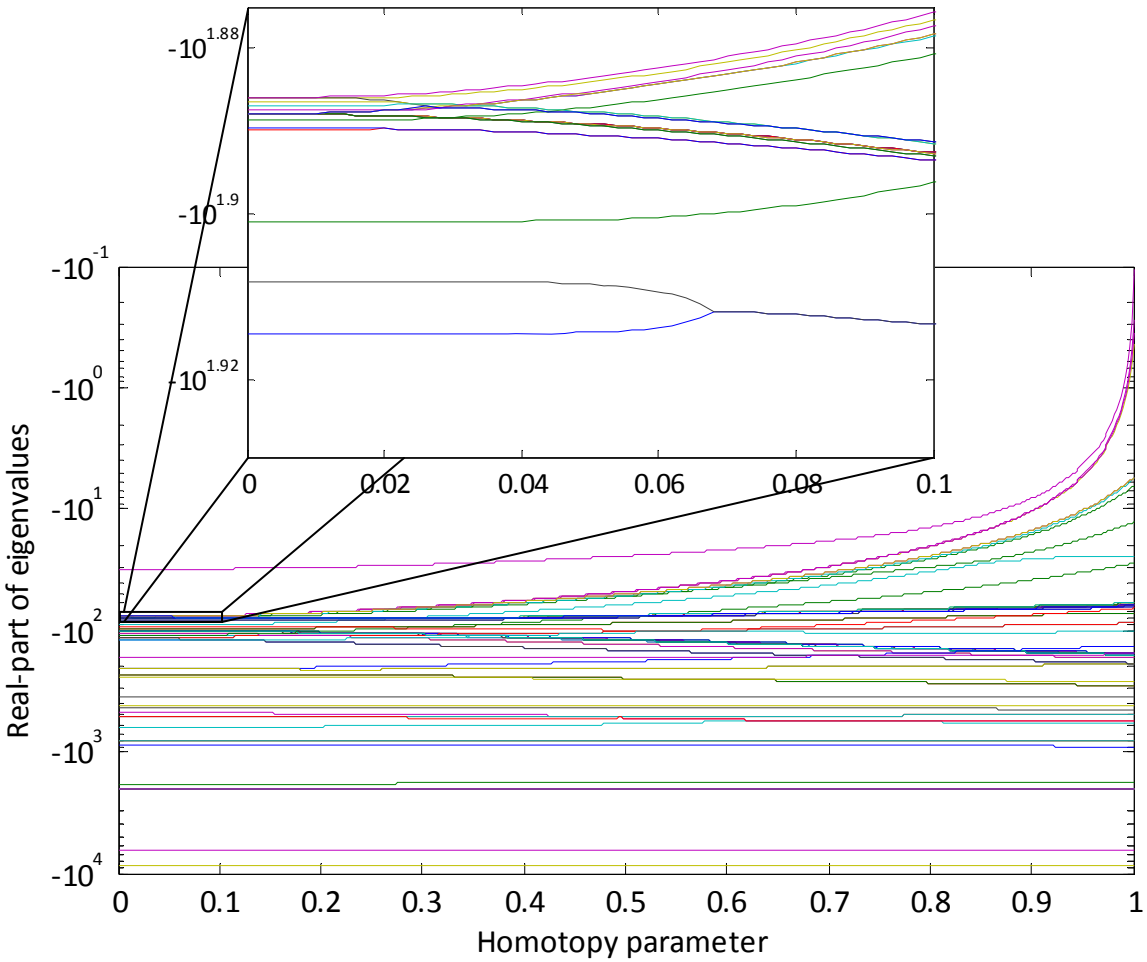

Figure 1: Real-part of eigenvalue traces through Homotopy method for the Benchmark model (Spanjers et al. 1998)

Only in rare cases the link between the state variable and the eigenvalue cannot be maintained. If an imaginary part of the eigenvalue appears during the homotopy process, it means that two eigenvalues become identical, e.g. the two traces showed in the zoomed graph in **Figure 1**, and that these eigenvalues share an imaginary part of opposite sign. In the case of automatic stiffness detection, if these eigenvalues merge during a finite section, then split again, it is impossible to decide which eigenvalue is linked to which state variable. Another case arises if two or more elements of the diagonal of the Jacobian are identical. Finally, the number of intermediate values for r will be decided in function of performances since computing eigenvalues takes $O(n^2)$ operation (Press et al. 1994). In practice, 20 steps have shown to be sufficient for a good linkage.

Some mechanisms must then be set to ensure that the quality of the simulation will be kept. However, as will be explained later, the eigenvalues are linked to state variables only to sort the state variables. So if two eigenvalues are not linked properly, as long as they stay in the same range, the missed link won't affect the model reduction.

2.4. Stiffness Reduction

The stiffness reduction is achieved by solving fast state equations as algebraic equations, i.e. the differential is set to 0. Doing so, differential equations responsible for large eigenvalues are solved as if they were at steady state.

The definition of "fast" state equation is of course problem-dependent. In the case of a biological wastewater treatment model, the step of interest can be around 10 minutes while in an air pollution model, 30 seconds may be more appropriate. It is then the user's responsibility to set the relevant time constant based on experience about the model being run and the objectives of the simulation study.

2.5. Control of reduction

The problem with automatic reduction is that there is no guarantee that a good dynamic model will give good results if part of it is solved algebraically. In practice, two situations have appeared that need to be controlled. The first one pertains to an inappropriate reduction of the model. In this case, an eigenvalue linked to a state variable is declared stiff, or fast, but after one calculated step forward, the variable loses its stiffness due to model non-linearity, i.e. its related eigenvalue is significantly reduced. The **Figure 2** shows such a bad reduction.

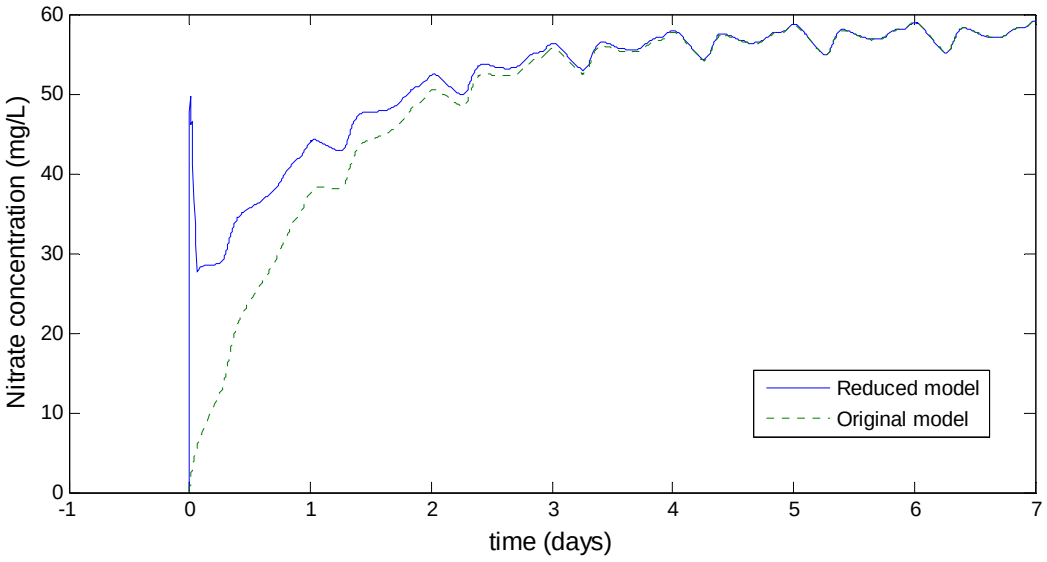

Figure 2: Comparison of reduced model to original model for a state reduction. Since solving the state variable as an algebraic variable induces large error, the reduction should be prevented.

This situation occurs when there is a huge change in the dynamics of the variable. A way to identify such a change is to recalculate eigenvalues, since after only one step, the variable's stiffness may have been relaxed until the point where it can no longer be considered stiff. Detection is done by comparing the number of stiff variables before and after the first step. This test allows identification of an inappropriate reduction before too much effort has been spent in the solution. The remedial action is to restart the integration at the time of the reduction.

A frequently occurring problem with root solvers is solution to an incorrect root (e.g., negative concentration). The Newton-Raphson method, for instance, will generally converge to the root closest to the initial guess. While in the scope of an integrator the root is usually close to the initial guess, time or state events in the model can create important discontinuities in the evolution of certain variables, thus enabling convergence to an incorrect root. Since in biological wastewater treatment models such controllers or other switches are common, it is important to be able to ensure that the solution will stay stable.

The method proposed to confirm convergence relies on the A-stability property of the DIRK method. With a stable solution, it is possible to compare the solution of the reduced model to the stable solution computed with the original model. In terms of our implementation, five steps of the reduced model are computed before they are compared to one large step using the DIRK algorithm. The steps are considered valid if their solution is within a certain tolerance of the stable solution. If the solution of the reduced model is considered too far from the stable solution of the full model, the steps are rejected and the full model is restarted from the point of reduction to simulate intermediate steps.

3. RESULTS

The number of function (model) evaluations is the most commonly used criterion for competitive algorithm evaluation. However, in the case of advanced algorithms, the code overhead can be important, so this criterion must still be considered with caution. This is particularly true in comparison of simple solvers like ODE45 with DIRK and multistep backward solvers, since there is considerable overhead related to Jacobian calculation, and matrix manipulations.

The ASM1 model (Henze et al. 2000) was implemented in Matlab and solved with different integrators. The implemented model only contains 12 state variables, since the concentration of dissolved oxygen was considered constant. The comparison between regular DIRK, DIRK with automatic reduction of stiffness and ODE45, the most widely used algorithm in Matlab, yields the following results for the number of model evaluations at different precisions:

Table 1: Number of model evaluation for DIRK algorithm, for DIRK algorithm with automatic reduction and for ODE45 solver from Matlab

Precision	DIRK with Full model	DIRK with Reduced model	Improvement	ODE45
10^{-1}	13,118	11,892	10%	483,461
10^{-3}	122,405	86,296	30%	1,227,050
10^{-5}	535,782	281,859	46%	1,226,940

It can be observed that improvement is more impressive at higher precisions. This can be explained by the good stability of the DIRK algorithm since the algorithm using the full model will not diverge from the solution, making it very appropriate for low tolerances. But in this very case, at low precisions, the average time step stays near the input time step. In these conditions, the algorithm is slowed down by the input resolution rather than by the stiffness of the model.

An important criterion for an improved solver is its impact on model outputs. State variables with medium or slow dynamics should approach the true solution, while stable (convergent) deviations of fast dynamics may be acceptable. An acceptable case is illustrated in **Figure 3**. Even where an approximate intermediate solution as shown in Ammonia concentration variation in time for both the full model and a reduced model. During this simulation, input was sent to the model with a zero order hold, which is responsible for the stair-like shape.**Figure 3** is not appropriate for final presentation, it is very effective for optimisation or parameter sampling, where thousands of calculations are required.

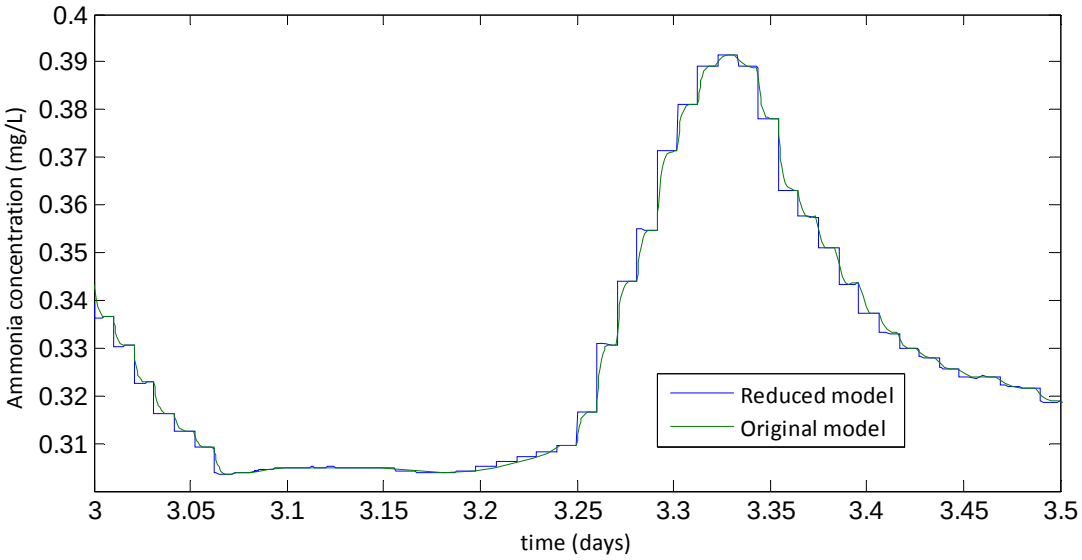

Figure 3: Ammonia concentration variation in time for both the full model and a reduced model. During this simulation, input was sent to the model with a zero order hold, which is responsible for the stair-like shape.

4. CONCLUSION AND FUTURE DEVELOPMENTS

The automatic reduction of a set of ODE to a smaller set of ODE plus a set of AE can reduce the number of function evaluations by almost half. If stability and error tolerance can be ensured, results for reduced and full models are identical for most purposes. Differences can be mostly observed in intermediate values of fast states. It is therefore important to define the objective of the model use before enabling and setting tolerances for automatic reduction of the model. In sensitivity analysis or optimization, in

particular, a reduced model may be valuable for rapid solution, while for presentation of simulation results, an original model may be preferred.

However, some issues are still to be solved. For one, the method highly relies on computation of Jacobian eigenvalues. The eigenvalues of small models (less than 30 state variables) are rapid to compute. However, typical models in biological wastewater treatment can easily contain 500 state variables, and the computation of eigenvalues can thus become prohibitive. Possible solutions are adaptive algorithms, where past information from the homotopy results could be reused to speed the process of linking state variables to eigenvalues, or decoupling of the model into its sub-models and evaluation of the eigenvalues of sub-models. In this last case, the precision of eigenvalue calculation could be reduced but since the eigenvalues are only to be sorted in "fast" and "slow" eigenvalues, one is interested only in the magnitude of the eigenvalue rather than in an exact value. It would also be appropriate, since blocks of states are often associated with discrete variables or points in space.

4.1. Acknowledgments

This work would not have been possible without the financial support from the Canada Research Chair in Water Quality Modeling, held by M. Peter Vanrolleghem. The authors also wish to thank MOSTforWATER (Kortrijk, Belgium) for the first author's scholarship and the Advanced Water Management Center (AWMC) of Brisbane, Australia for the support during the three months stay abroad of the first author.

5. REFERENCES

Bui, T. D. (1979). "Some A-stable and L-stable methods for the numerical integration of stiff ordinary differential equations." J. ACM **26**(3): 483-493.

Cameron, I. T. (1983). "Solution of differential-algebraic systems using diagonally implicit Rung-Kutta methods." IMA Journal of Numerical Analysis **3**: 26.

Henze, M., W. Gujer, T. Mino and M. van Loosdrecht (2000). Activated Sludge Models ASM1, ASM2, ASM2d and ASM3. Londres, UK, IWA Publishing.

Hesstvedt, E., O. Hov and I. S. A. Isaksen (1978). "Quasi-steady-state approximations in air-pollution modeling - comparison of two numerical schemes for oxidant prediction." International Journal of Chemical Kinetics **10**(9): 971-994.

Press, W. H., W. T. Vetterling, S. A. Teukolsky and B. P. Flannery (1994). Numerical recipes in C : the art of scientific computing. Cambridge ; New York, Cambridge University Press.

Rosen, C., D. Vrecko, K. V. Gernaey, M. N. Pons and U. Jeppsson (2006). "Implementing ADM1 for plant-wide benchmark simulations in Matlab/Simulink." Water Science and Technology **54**(4): 11-19.

Shampine, L. F. and M. W. Reichelt (1997). "The MATLAB ODE suite." Siam Journal on Scientific Computing **18**(1): 1-22.

Spanjers, H., P. Vanrolleghem, K. Nguyen, H. Vanhooren and G. G. Patry (1998). "Towards a simulation-benchmark for evaluating respirometry-based control strategies." Water Science and Technology **37**(12): 219-226.

Steffens, M. A., P. A. Lant and R. B. Newell (1997). "A systematic approach for reducing complex biological wastewater treatment models." Water Research **31**(3): 590-606.

Vanrolleghem, P., D. Borchardt, M. Henze, W. Rauch, P. Reichert, P. Shanahan and L. Somlyody (2001). "River water quality model no. 1 (RWQM1): III. Biochemical submodel selection." Water Science and Technology **43**(5): 31-40.