Study of DAE solvers for Transient Stability Analysis

Transient Stability Analysis examines the dynamic behaviour of a Power System for as much as several seconds following a power flow disturbance. The reaction of a single generator to variations in its load can be modeled using ODEs as follows

$$\dot{y} = f(y,z)$$

In our case, the exact equations are

$$I = VY$$

However, there are multiple generators supplying power to the network and the generators are coupled through the power network. The power network can be modeled by non-linear algebraic equations as follows

$$0 = g(y,z)$$

The exact equations are:

$$E + x_d I_d = E_q^{'} + x_d^{'}$$

$$E_d + x_q I_q = E_d^{'} + x_q^{''} I_q$$

$$\tau_{q0}^{'} \dot{E}_d^{'} = -E_d^{'} - (x_q - x_q^{'}) I_q$$

$$\dot{E}_q^{'} = \frac{1}{T_{d0}^{'}}(E_{FD} - E)$$

$$T_e = E_d^{'} I_d + E_q^{'} I_q - (L_q^{'} - L_d^{'}) I_d I_q$$

$$\tau_j \dot{\omega} = T_m - D\omega - [E_d^{'} I_d + E_q^{'} I_q - (L_q^{'} - L_d^{'}) I_d I_q]$$

$$\dot{\delta} = \omega - 1$$

and this entire system of DAEs must be solved simultaneously.

NOTE: For the terminology description of the above equations, please refer to: Power System Stability and Control by Anderson.

Depending on the load equations and control equations that need to be added to this system, our system will become *Index-1* or *Index-2* system. The algebraic system as described above presents a very interesting scenario, in that it is a system of complex variables. We then are faced with the choice of solving it as a system on N complex equations or 2-N real equations. But we may lose the sparsity structure in the process. We are investigating this issue now.

The generator equations by themselves do not pose any problem except when the load equations and control equations are added. But they still will be able to satisfy the standard input format of most DAE solvers. Hence one of the reasons to pick a particular DAE solver will be the form of equations it requires as input. Keeping this in mind, the following solvers were analysed and here are the findings:

1. LIMEX - an extrapolation integration technique. [1]. This can be used to solve semi-explicit index-1 DAEs and uses the semi-implicit Euler method. This is a strong contender because of its ability to handle frequent discontinuities, which is a contidtion that frequently occurs in Power Systems TSA problem.

2. RADAU5 - an implicit Runge-Kutta method of order 5. [2].

3. RODAS - an embedded Rosenbrock technique of order(3)4. [2]. This is similar to R-K except that at each step the Jacobian has to be evaluated and the linear systems have to be solved again. Also, this can handle only Index-1 DAEs.

4. SODEX - an extrapolation technique based on linearly implicit mid-point rule. [2]. One disadvantage of this method is that it takes large step sizes, which may cause exponential overflow when evaluating the right hand side of ODEs.

5. SEULEX - an extrapolation technique based on linearly implicit Euler method. [2].

6. DASSL - a multistep, backward differentiation formulas technique.[1].

   This is designed for DAEs of Index one or zero. CDASSL (concurrent DASSL) has been developed already.

7. LSODI - a fixed leading coefficient method using BDF technique.[4, 3, 5]. In contrast to DASSL (which is a fixed leading coefficient method), LSODI is a 'fixed coefficient' method. The fixed coefficient methods can be implemented very efficiently for smooth problems, but suffer from inefficiency, or possible instability, for problems that require frequent stepsize adjustments, which is not the case with power systems problem. Hence LSODI promises to be our best choice currently.

The backward differentiation formulas (BDFs) methods achieve the same order of convergence for this class of DAEs as they do for ODEs. Implies, even if our system is of Index-2, we can achieve convergence. Also, BDF methods can handle discontinuities, which is a condition frequently encountered in our problem. Thus there is some theory providing the foundation for the software based on these methods. However, there are other issues that need to be considered too while selecting a solver for the problem in question.

For now, our target is LSODI and LIMEX , with possible diversion to a implicit version of VODE called VODI. Solvers like LSODI and LIMEX have been used extensively and we have data that proves their efficiency. Solvers like RADAU5, RODAS, SEULEX and SODEX might be efficient and probably more suitable but we have absolutely no data or opinions from any users. Hence the choice of LSODI (since a parallel version of DASSL is underway...) seems apt.

References

[1] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations.* Elsevier Science Publishing Co., 1989.

[2] E. Hairer and G. Wanner. *Solving Ordinary Differential Equatios II — Stiff and Differential-Algebraic Problems.* Springer-Verlag, 1991.

[3] A. C. Hindmarsh. Large ordinary differential equation systems and software. *IEEE Control Systems Magazine*, 2:24–30, 1982.

[4] A. C. Hindmarsh. ODEPACK, a systematized collection of ODE solvers. In R. S. Stepleman et al., editors, *Scientific Computing*, pages 55–64. North-Holland, Amsterdam, 1983.

[5] A. C. Hindmarsh. The ODEPACK solvers. In R. C. Aiken, editor, *Stiff Computation*, pages 167–174. Oxford Univ. Press, 1985.