Domain Decomposition methods for Stiff ODEs/DAEs

D. Guibert [a] * and D. Tromeur-Dervout [a, b] †

[a]CDCSP/ICJ UMR5208, University Lyon 1, 15 Bd Latarjet, 69622 Villeurbanne, France

[b] IRISA-INRIA, Campus de Beaulieu, 35042 Rennes cedex, France

Keywords: time Domain Decomposition, Schur Domain decomposition, Stiff ODEs/DAEs, Parallel Computation,
MSC : 65L80, 65L06, 65Y05, 65Y10, 65N55

Nowadays, the designing and optimizing of manufactured products involve more and more compounds. They interact through variables describing different physical laws. The behavior in time of such complex model is driven by dynamical Ordinary Differential Equations (ODEs) and/or Differential-Algebraic Equations (DAEs) systems with more and more variables. As these systems, should eventually manage discontinuities and are stiff, they require implicit methods for stability reasons. The Jacobian matrix of the global system has to be evaluated (usually numerically) and inverted frequently. These facts lead to a rise of the computational time requirement, and need parallel computing. This is especially necessary when several computations have to be performed in order to give the model's robustness to parameters.

The main difficulty in term of parallel implementation of such systems of ODEs/DAEs equations is the poor granularity of the computations especially when the architecture has a large number of processors. The spatial distribution of data that arises in PDEs that usually performed well on parallel computer is no longer available. Obviously, the amount of computational work increases with the number of unknowns in the ODEs or DAEs systems. Nevertheless the elapse time of such computation is driven by the sequential nature of the computation and the number of time steps to perform. The both avoid efficient parallelism.

In this talk we will investigate three approach to parallelize ODEs/DAEs system. The first, will be the parallelization across the method [2] which is limited to the number of stages of the Runge-Kutta time integration scheme used.

The second, is the Schur domain decomposition for the Jacobian matrix in the LSODA time integrator [1] . The difficulties of this approach are to build the framework to decompose the system of DAEs in sub-system, and the large condition number in the actual problems that requires preconditioning.

*This work is backward to the Région Rhône-Alpes thru the project: "Développement de méthodologies mathématiques pour le calcul scientifique sur grille".
†This author was partially supported thru the GDR MOMAS:"solveur multidomaines"

Elapse time(s)	sequential code	parallel code - 3 proc	speed up
Jacobian computing		260	
Jacobian communication		28.6	
total Jacobian	847	288.6	2.93
comput. stage		43.5	
comm. stage		44.7	
total stage	143	88.2	1.62
total elapse time	1082	436	2.48

Table 1
Comparison between the sequential version and the parallelization accross the method of Radau IIa on a engine injectors problem with 131 unknowns.

#proc	CPU time	speed-up	#Jac	#discont	#steps
1	6845	1	65355	1089	311115
2	4369	1.56	66131	1061	315357
3	1820	3.76	65787	1059	313064
4	1513	4.52	65662	1043	313158

Table 2
LSODA parallelization with Schur DDM on an engine problem with 10 injectors leading to a 287 unknowns system of ODEs

The third approach presented is a multiple shooting time-domain decomposition, that combines two levels of parallelism. One in the time direction with splitting the time interval in sub-domains . The second level is in accuracy direction with using some deferred correction [4]. This combination leads to a pipeline of the computation.

REFERENCES

1. K.E. Breman, S.L. Campbell, and L.R. Petzold: *Numerical Solution of Initial-Value Problems in Differential- Algebraic Equations*, Elsevier Science Publishing Company, New York (1989).
2. K. Burrage. Parallel and Sequential Methods for Ordinary Differential Equations, Oxford University Press, New York, 1995.
3. J.-L. Lions, Y. Maday, and G. Turinici. *Résolution d'EDP par un schéma en temps "pararéel"*, C.R.A.S. Sér. I Math., **332**(7), pp. 661-668, 2000.
4. M. Michael *Semi-implicit projection methods for incompressible flow based on spectral deferred corrections*, Appl. Numer. Math. 48, no. 3-4, 369–387, 2004