Simulated Highly Parallel Solution of Versatile Companion Model Graph Equations

Kaj Juslin

VTT Technical Research Centre of Finland,
Digital Information Systems,
Vuorimiehentie 3, Espoo, Finland
kaj.juslin@vtt.fi

Abstract. Oriented linear graphs have already half a century ago been considered suitable to describe lumped physical systems /1/. Usually matrix equations arising from partial differential equations describing physical networks are very sparse. Companion models arising from linear electrical circuits are introductory presented. /2/. Specific versatile companion model graph eguations can be applied when, for instance, discontinuities and non-linear dependencies have to be considered /3/. Present sequential solvers for relevant sparse equation systems are not very suited for highly parallel computing. A general concern for developers of industrial process analysis codes is: How to make the code running efficiently on future multi core processor chips /4/. In this study relevant algorithms and architectures are applied on a simulated highly parallel computer. The lessons learned are supposed to guide the transportation or re-programming of millions of lines of present sequential codes.

Keywords: Companion models, highly parallel solution, graph equations, multi core processors, memory architectures.

1 Introduction

Analysis software has been developed for evaluation of the dynamic performance of industrial plants and related automation systems, both in normal and accidental situations. The sustainability of the sometimes very extensive specifications of the analysis models has to be ensured for the whole life cycle of the target plant. Application of layered software architectures is one key element in ensuring easy software maintenance and transportation between different hardware and operation system platforms. Usually such layers represent specific services: Graphics user interface for model specification and simulation control, interfaces to semantic plant and automation design data repositories, intelligent evaluation and preparation of model specifications, sufficient libraries of well verified and validated elementary models, efficient solvers of the resulting set of differential and algebraic nonlinear

equations. In this study we will concentrate on the solution layer. Oriented graphs resembling physical companion models have a specific mission in the transfer of geometric information to underlaying equations. They can also be helpful in the tearing and grouping of the equation systems to suite the distributed memory architectures of highly parallel computers.

2 Linear companion models

Elementary companion models can be specified for in many physical domains to enhance the construction of comprehensive system simulators. The best known companion models arise from temporal discretisation of ordinary differential equations characterising discrete electrical circuit components such as resistors, inductors, capacitors and voltage sources. We may write the relevant companion model equation $x_i g_f - x_j g_b + s_{ij} - v_{ij} = 0$. The voltages at each side of the component represent the local variables x_i and x_j and the current through the component represents the transition variable v_{ij}. Let us for instance consider a constant voltage source component. If we denote its internal resistance as R and its internal voltage as E, then we may write for the relevant forward and backward coefficients $g = g_f = g_b = 1/R$ and and for the source term $s_{ij} = E/R$. In this case the companion model has symmetric coefficients and is characterized as A-causal. The voltages x_i can be solved in all internal nodes of a relevant connected graph representing the electrical circuit of concern subject that there is at least one connection to an external reference node. We may write the node matrix equation $G \underline{x} = \underline{J}$ and solve for the voltages x_i of the internal nodes. Thereupon the currents v_{ij} of all the edges can be solved from the relevant companion model equations.

3 Real world heterogeneous systems

The graphs in consideration are sometimes heterogeneous, that is, there are in addition to ordinary connection coefficients g also very strong connections and/or very weak connections. In such a case grouping and tearing of the original graph may be required. Nodes connected with very strong edges can be temporarily grouped to pseudo-nodes. Nodes connected with very weak edges can be torn to separate sub graphs. Relevant sub-systems can sometimes in dynamic simulation studies with preference be processed with different time-steps in order to save computer resources for slow developments, and on the other hand, and to follow up with fast dynamics. Further, typical numerical errors related to limited world length can be avoided.

4 Non-linearities and discontinuities

The versatile companion model considered in this paper allows for non-linearities and discontinuities as well as for asymmetric or even zero valued coefficients. However, we require that each coefficient is piecewise continuous and that it within a

continuous interval is either constant or strictly monotonic. Such versatile companion models can for instance be specified for thermal hydraulic circuits considering the general continuation equations for mass, momentum and energy. The main non-linearity is caused by the friction coefficient of turbulent flow depending on the square of he mass flow rate but also on on the viscosity and density of the fluid, which in turn may be very non-linear as depending on the relevant pressure, specific enthalpy, and fluid composition. These are considered as the local variables and the flow rates of mass, energy and substances are considered as transition variables. In typical sequential thermal hydraulic solvers the solution of the required material properties can take more than 90% of the calculation time

5 Direct and iterative solvers of linear matrix equations

It should be noted that the node matrices resulting from the one dimensional pipe networks considered in this case are very sparse. To avoid need for pivoting during the solution, the requirement on irreducible diagonal dominance can be considered in advance by checking the coefficients of relevant graph before building the matrix equation to solve. Also, by advance checking of the graph connections it is possible to find the optimal sequence of solving the internal unknown variables in order to avoid unnecessary fill in of new elements during the required factorization of the matrix. In pracice, the model may comprise several completely separate sub graphs subject that each sub graph is internally connected and it has at least on connection to an external node. An additional bonus is, that the resulting matrix is usually strongly diagonally dominant, which makes it suitable for simple iterative solution schemes such as point Jacobi, which is much easier to parallel than sparse direct methods.

If the effort related to correction effort of the non-linear coefficients can be successfully distributed to parallel processing cores having sufficient local memory resources, then one could expect that it would be possible to make efficient use of a very large number of cores before the overhead of the slow communication between the cores starts to dominate.

In the versatile companion models the coefficients of the edges are supposed to be piecewise continuous. They may also be non-linear, however, in each interval they need to be either constant or strictly monotonic. Strictly monotonic implies that they are either monotonically increasing or decreasing. This requirement enables implementation of efficient Newton based iterative methods for correction of non-linearities. The discontinuity points are either time dependent or computed. The exact computed crossing points of concatenating intervals is usually computed by iterative means, as well.

An identified challenge is to optimize several nested iteration loops from the point of view of solving global and local variables in parallel. The very different memory access times prioritizes calculation of very exact local solutions before invoking a

global solution step. It seems preferable if local changes in first place can be managed by local computing efforts.

6 Simulated parallel computing platform

A highly parallel computing MIMD hardware platform is simulated to demonstrate typical memory and communication architectures. Latency times and communication speeds as well as bus loading are considered. The purpose is to compare on this simulated platform sequential and parallel algorithms as well as direct and iterative solution methods especially applied for versatile companion model graph equations.

7 Hydraulic network as test case

A test case comprising an extensive hydraulic network is applied. Different solver schemes and data architectures suitable for highly parallel computation are compared. Also means to benefit from multi threaded cores on massively parallel processor chips are considered.

8 Conclusions

A simulation model of highly multiprocessing hardware has been applied to evaluation of best practices for parallel solution of a specific group of problems. The impact of the computer architecture and relevant performance indicators on the calculation speed is estimated. Possible bottlenecks either in hardware or solver architectures can be identified. In this case readiness for tera-scale desktop computing of extensive analyzer codes is addressed.

References

1. Trent, H. M. 1955, Isomorphisms between oriented linear graphs and lumped physical systems. The journal of Acoustical society of America, Vol. 27, pp. 500-527.
2. Juslin, K., Silvennoinen, E., Karppinen, J. 1985. Experiences on real-time solution of sparse matrix equations. IMACS Transactions on Scientific Computation 1985, Vol. 3. Modeling and simulation in engineering, North Holland, Netherlands, 6p.
3. Juslin, K., 2005. A companion model approach to modeling and simulation of Industrial processes. VTT Publications 574. ISSN 1455-0849 (URL: http://www.vtt.fi/inf/pdf/), 155 p
4. Intel's Tera-Scale Research Prepares for Tens, Hundreds of Cores, in the Technology@Intel Magazine, February 10, 2008 (URL: http://www.intel.com/technology/magazine/ computing/tera-scale-0606.htm), 8p.