from *Proc. Second International Conf. on Foundations of Computer-Aided Process Operations* (Crested Butte, Colorado, July 18-23, 1993), Eds., D.W.T. Rippin, J.C. Hale, and J.F. Davis, CACHE Corporation, Austin, TX (1994)

IMPACT OF SUPERCOMPUTING IN SIMULATION AND OPTIMIZATION OF PROCESS OPERATIONS

Stephen E. Zitney
Cray Research, Inc.
Eagan, MN 55121-1560

Kyle V. Camarda and Mark A. Stadtherr
University of Illinois
Urbana, IL 61801

Abstract

Impressive gains in computer technology, including vector and parallel processing architectures, make possible today the solution of larger-scale and more realistic engineering problems. In this paper we discuss the impact of these developments on solving simulation, optimization, and control problems in chemical process operations. Advances in numerical methods, especially for sparse matrix problems, have accompanied hardware developments and facilitated their exploitation on process operations problems. We demonstrate here the impact of these advances in the dynamic simulation and control of three industrial process operations using the commercial simulation package SPEEDUP on a CRAY Y-MP supercomputer.

Keywords

Supercomputing, Dynamic Simulation, Process Optimization, Process Control, Sparse Matrix Methods, Frontal Method.

Supercomputing in Process Operations

In an increasing number of production facilities, chemical engineers are using simulation tools, especially dynamic simulators and optimization systems, to reach innovative solutions to process operation problems. Engineers can use these tools to design better operation and control strategies, to provide more effective operator training, and to optimize plant performance in terms of resource utilization, safety, environmental impact, and economics. Such tools may be used off-line or on-line, in the latter case either as a guide to plant operators, directly to provide set points to a distributed control system, or in other model-based control schemes.

Plant-wide simulation and optimization is becoming more desirable than the study of a single process unit or a group of process units (Bailey et al., 1993). Simulating the complete plant avoids the local suboptimal solutions often encountered in a distributed optimization approach (local unit optimizers with a coordinator). The utility of local results degrades rapidly over time due to process fluctuations and market dynamics, whereas the plant-wide approach has the capability to adjust local operating variables to avoid bottlenecks and achieve overall plant objectives.

Another trend in process simulation and optimization is the use of more rigorous, equation-based models (Fisher, et al., 1991). Although the level of model complexity ultimately depends on business objectives, models based on physical and chemical principles are often preferred over simplified empirical models because of their extendibility to new operating conditions. Moreover, models in an equation-based form offer total specification flexibility and, in comparison to traditional sequential-modular methods, are readily used for dynamic simulation and optimization. As a result, the last few years have seen a steady increase in the use of commercial equation-based tools for dynamic simulation (e.g., SPEEDUP) and process optimization (e.g., DMO, MINOS, NOVA).

These industry trends are due largely to impressive gains in computer performance and to advances in equation solving technology, especially for sparse matrices. Spurred by new innovations in chip technology, manufacturing techniques, and architecture design, computers continue to set industry standards for performance, and price-performance, at even lower price points. Paving the way, modern supercomputers offer vector and parallel processing for solving problems that, until now, couldn't be solved with other computational tools. Today, this leading-edge technology is being brought downward into smaller, less expensive systems to help more organizations access and apply valuable supercomputer simulations to their businesses.

With the advent of low cost, entry-level supercomputers, the use of supercomputer technology for plant-wide dynamic simulation and process optimization is more practicable than ever before. By efficiently solving matrix systems involving hundreds of thousands of equations, these supercomputer systems can provide the quick turnaround time needed for running large, complex equation-based models in a process operations environment. However, the realization of these opportunities is contingent on the extent to which simulation and optimization algorithms can take full advantage of vectorization and parallelism. Since most currently used techniques were developed for use on conventional serial machines, it is evident that exploiting supercomputing (as opposed to simply using a supercomputer) will require rethinking the solution strategies for process simulation and optimization.

In large-scale simulation and optimization problems, the key step, representing as much as 90% of the computational effort on industrial applications, is often the solution of a large sparse system of linear equations. Thus, we concentrate here on techniques leading to the efficient solution of such systems on supercomputers. Recent work (Vegeais and Stadtherr, 1990; Zitney, 1992; Zitney and Stadtherr, 1993a,b) has demonstrated the potential of the frontal method as a sparse linear equation solver in this context. In fact, an implementation of the frontal method, developed at the University of Illinois and Cray Research, Inc., has now been incorporated in Cray Research versions of commercially used tools such as ASPEN

PLUS, BATCHFRAC, RATEFRAC, and SPEEDUP. In this paper, we present results for three large-scale industrial problems solved using the dynamic simulator SPEEDUP on a CRAY Y-MP supercomputer.

Sparse Matrix Methods

In recent years, applied mathematicians, computer scientists, as well as chemical engineers, have made great strides in developing effective techniques for manipulating and solving sparse matrices. These developments have important implications for chemical plant operations since solving large, sparse, unsymmetric systems of linear equations is one of the key underlying mathematical problems in process simulation and optimization. Industrial plant-wide problems exhibit a high degree of sparsity (typically less than 1% nonzero elements) and can require that a major portion (as much as 90%) of the total computation time be spent in the sparse matrix solver.

Current general-purpose sparse matrix methods intelligently recognize and use sparsity to improve the speed of computations. The codes studied here are LU1SOL from the SPAR2PAS package (Stadtherr and Wood, 1984) and MA28 from the Harwell Subroutine Library. Unfortunately, these traditional methods spend considerable effort accessing random memory locations to avoid doing unnecessary arithmetic operations. This process, known as indirect addressing, is very inefficient on vector computers, especially when dealing with a small number of nonzero entries per row as in the case of process simulation and optimization problems.

Modern vector and parallel computer architectures have resulted in a new generation of sparse matrix methods (Duff et al., 1986; Gallivan et al., 1990); Dongarra et al. 1991; Zlatev, 1991). One of the methods, namely the frontal method (Irons, 1970; Hood, 1976; Duff, 1979), avoids the indirect addressing problem and exploits vector processing by performing Gaussian elimination on a series of dense "frontal matrices". Based on dense-matrix kernels, the frontal method represents a trade-off between vector performance rate (high for dense-matrix computation) and the number of unnecessary operations on the zero elements that tend to occur in the frontal matrices.

We study here the one-pivot version of the multiple-pivot FAMP frontal code originally developed at the University of Illinois (Zitney and Stadtherr, 1993a) and later extended at Cray Research to include the use of BLAS2 and BLAS3 kernels, an out-of-core option for solving very large-scale problems, and the implementation of separate analyze, factorize, and solve options. The latter feature is required for applying the frontal code in the context of dynamic simulation and control—the process operations problems considered in this paper.

Industrial Applications

The commercial simulation system SPEEDUP is used here to assess the performance of the frontal method on three dynamic chemical processes. The first two problems, HYDR1C and EXTR1B, are two-column separation processes studied in an earlier work by Zitney (1992). Problem HYDR1C from DuPont involves the processing of a seven-component hydrocarbon mixture in a de-propanizer (40 stages) and de-butanizer (50-stages). In this operations problem, the process controllers react to a series of rectangular pulses which are introduced into the propane composition in the feed stream. A total of 5308 equations make up the full process model, including all unit operations and controllers. The second problem, EXTR1B, is an extractive distillation process (84 stages) from Air Products and Chemicals (Grassi and Luyben, 1992). In this case study, an autotune variation (ATV) test is performed to determine the ultimate gain and frequency of the control loops. This ATV test generates a continuous flow of discontinuities so that the differential algebraic equation solver in SPEEDUP has to reinitialize at each step and solve a sparse linear system containing 2836 equations. The dynamic response of the top product composition in the extraction tower is followed over a period of 1.5 hours.

The third problem, SEGM3A, involves the operation and control of a backmixed, multistage reactor from Shell KSLA (Calvet and Zullo, 1992). A homogeneous liquid-phase reaction takes place subject to a degradation phenomena which leads to the formation of undesired by-products. By appropriately manipulating the cooling system, the temperature and concentration profiles along the 24-stage reactor are controlled to obtain a product of the desired quality. In the control study here, a 5% step increase is imposed on the catalyst feed concentration in order to investigate the operation and control of the reactor subject to disturbance. The SPEEDUP run simulates a 10,000 second transient response using frequent samples (one second intervals).

We now compare the performance of the frontal method, as implemented in FAMP, with two general sparse matrix methods, MA28 from Harwell and LU1SOL from the SPAR2PAS package (Stadtherr and Wood, 1984). All three sparse matrix methods are available in the Cray Research version of SPEEDUP 5.4. MA28 is the default solver in SPEEDUP. LU1SOL and FAMP are used by specifying SPAR2PAS and FRONTAL respectively in the ROUTINE sub-section of the OPTIONS input section. The solver option FRONTAL is only available in the Cray Research version of SPEEDUP. All runs were made on one processor of a CRAY Y-MP supercomputer using SPEEDUP 5.4.

Results for the three SPEEDUP dynamic simulation and control problems are shown in Fig. 1. The timings represent the cpu time in seconds required to solve the entire SPEEDUP problem. The relative speedup of FAMP over MA28 and LU1SOL (denoted by an "x" following the value) are given above the bars representing MA28 and LU1SOL, respectively. Comparing the traditional sparse matrix solvers, it can be seen that LU1SOL outperforms MA28 by an average of about 15%. FAMP is at least three times as fast as LU1SOL, clearly making it the best overall method. For the most computationally intensive problem considered here, namely SEGM3A, the total solution time is reduced from 4.5 hours using LU1SOL to under 1.5 hours using FAMP.

It should be pointed out here that the solution of sparse linear equation systems represents only one subproblem in a SPEEDUP dynamic simulation. For the problems studied here about 70-80% of the total computation time is due to the linear equation solver when using MA28. This means that the speedup of FAMP on the sparse linear systems alone is actually much greater than the speedup in overall solution time reported here. It should also be noted that in our experience the performance of the frontal method, when compared to traditional general sparse methods, improves with certain process features, including a large number of chemical components and flowsheet topologies with "long" recycle streams (Zitney and Stadtherr, 1993b).

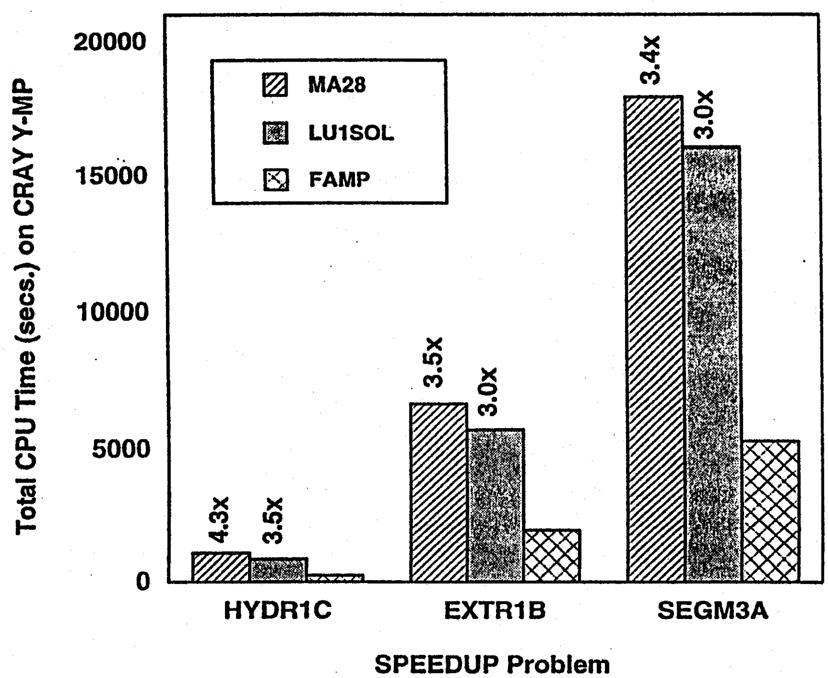

Figure 1. Comparison of standard sparse matrix codes (MA28 and LU1SOL) and the frontal method (FAMP) on a CRAY Y-MP supercomputer. The total solution times for the SPEEDUP problems are shown.

Conclusions

Supercomputing technology is making possible the use of large-scale, plant-wide models for the simulation, optimization, and control of chemical process operations. Advances in sparse matrix techniques, in particular the frontal method, play a very important role in exploiting this technology, as demonstrated on three industrial problems solved using SPEEDUP on a CRAY Y-MP supercomputer. Continued improvements in these enabling technologies will permit companies competing in the chemical process industry to manage the growing complexity and growing time-criticality that are characteristic of the new era of computer-aided process operations.

Acknowledgements

This work was supported in part by National Science Foundation Grant DDM-9024946.

References

Bailey, J.K., S.S. Treiber and R.B. Hawkins (1993). Nonlinear optimization of a hydrocracker fractionation plant. *Comput. Chem. Engng.*, **17**, 123-138.

Calvet, J.P., and L.C. Zullo (1992). Modelling and optimisation of a backmixed, multistage reactor for an industrial reactive system. *AIChE 1992 Annual Meeting*, Miami Beach, FL, paper 131k.

Dongarra, J.J., I.S. Duff, D.C. Sorensen and H.A. van der Vorst (1991). *Solving Linear Systems on Vector and Shared Memory Computers.* SIAM, Philadelphia.

Duff, I.S. (1979). Recent developments in the solution of large sparse linear equations. *Proc. of IRIA Fourth Intl. Symp. on Computing Methods in Applied Sciences and Engineering*, Versailles.

Duff, I.S., A.M. Erisman and J.K. Reid (1986). *Direct Methods for Sparse Matrices.* Clarendon Press, Oxford.

Fisher, G., C. Fontenot, K. Tamura and T. Sumiyoshi (1991). Optimisation of ethylene plant operations using rigorous models. *AIChE 1991 Spring National Meeting*, Houston, TX, paper 20f.

Gallivan, K.A., M.T. Heath, E. Ng, J.M. Ortega, B.W. Peyton, R.J. Plemmons, C.H. Romine, A.H. Sameh and R.G. Voigt (1990). *Parallel Algorithms for Matrix Computations.* SIAM, Philadelphia.

Grassi, V.G., and W.L. Luyben (1992). Process control of extractive distillation. *AIChE 1992 Annual Meeting*, Miami Beach, FL, paper 2g.

Hood, P. (1976). Frontal solution program for unsymmetric matrices. *Int. J. Num. Meth. Engng.*, **10**, 379.

Irons, B.M. (1970). A frontal solution program for finite element analysis. *Int. J. Num. Meth. Engng.*, **2**, 5.

Stadtherr, M.A., and E.S. Wood (1984). Sparse matrix methods for equation-based chemical process flowsheeting: II. numerical phase. *Comput. Chem. Engng.*, **8**, 19.

Vegeais, J.A., and M.A. Stadtherr (1990). Vector processing strategies for chemical process flowsheeting. *AIChE J.*, **36**, 1687-1696.

Zitney, S.E. (1992). Sparse matrix methods for chemical process separation calculations on supercomputers. *Proceedings of Supercomputing '92*, Minneapolis, MN, pp. 414-423.

Zitney, S.E., and M.A. Stadtherr (1993a). Frontal algorithms for equation-based chemical process flowsheeting on vector and parallel computers. *Comput. Chem. Engng.*, **17**, 319-338.

Zitney, S.E., and M.A. Stadtherr (1993b). Supercomputing strategies for the design and analysis of complex separation systems. *Ind. Engng. Chem. Res.*, **32**, 604-612.

Zlatev, Z. (1991). *Computational Methods for General Sparse Matrices.* Kluwer Academic Publishers, Dordrecht.