teriel Command. Work and analysis for this research were done by personnel of Sverdrup Technology, Inc., AEDC Group, technical services contractor for the AEDC propulsion test facilities. Further reproduction is authorized to satisfy needs of the U.S. Government.

References

[1]Pearse, R. W. B., and Gaydon, A. G., *The Identification of Molecular Spectra*, Wiley, New York, 1963, p. 58.

[2]Dash, S. M., and Pergament, H. S., "A Computational System for the Analyses of Mixing/Chemical/Shock Processes in Supersonic Internal and Exhaust Plume Flowfields," AIAA Paper 80-1255, July 1980.

[3]Huber, K. P., and Herzberg, G., *Molecular Spectra and Molecular Structure*, Van Nostand Reinhold, New York, 1979.

[4]Bernath, P., "Molecular Constants of the $X^1\Sigma$ State of AlCl," *Journal of Molecular Spectroscopy* (to be published).

[5]Kovacs, I., *Rotational Structure in the Spectra of Diatomic Molecules*, Adam Hilger, London, 1969.

[6]Langhoff, S. R., and Bauschlicher, C. W., "Theoretical Studies of AlF, AlCl and AlBr," *Journal of Chemical Physics*, Vol. 88, No. 9, 1988, p. 5715–5725.

[7]Rogowski, D. F., and Fontijn, A., "The Radiative Lifetime of AlCl $A^1\Pi$," *Chemical Physics Letters*, Vol. 137, 1987, p. 219.

Parallel Processing Approach for Radiative Heat Transfer Prediction in Participating Media

C. Saltiel*
University of Florida, Gainesville, Florida 32611
and
M. H. N. Naraghi†
Manhattan College, Riverdale, New York 10471

I. Introduction

NUMERICAL analyses of radiative transfer in participating media can be very complex due to the long distance nature of radiative exchange. Computer simulations of practical situations often require both large computer memory and long calculation times. Serial processors have been used extensively for radiative analyses, but simulations can take hours on powerful machines like the CRAY Y-MP.[1]

The use of massively parallel machines, e.g., the single-instruction multiple-data (SIMD) connection machine (CM) with hypercube interprocessor connectivity, has proven very effective in simulating large complex systems.[2] Calculations are performed concurrently on tens of thousands of processors, each associated with memories arranged for efficient processor-to-processor communication (in contrast to coarse parallelism, where typically between 2 and 64 processors share memory). In finite difference and finite-element computations, each processor is associated with one or more spatial nodes, allowing all calculations on the data to be carried out simultaneously. Along with the promise of substantial increases in computational speed, massive parallelism also favors simpler computational algorithms since programming is array oriented, as in FORTRAN 90.[3]

Received Nov. 16, 1992; revision received Jan. 26, 1993; accepted for publication Jan. 29, 1993.

*Assistant Professor, Department of Mechanical Engineering.
†Associate Professor, Department of Mechanical Engineering.

Radiative computations are particularly well-suited for parallel processing. Recently Howell[4] has reviewed the use of parallel computing for radiative transfer but did not address the use of massively parallel architectures. Coarse parallelism has been described for Monte Carlo radiative heat transfer on an eight-processor ETA-10.[5] Hanebutte and Lewis[6] used a CM-2 for simulating radiative transfer in a two-dimensional enclosure via the discrete ordinates method. Other node-based radiative transfer solution techniques could also be programmed for parallel computation.

Typically, only a few hundred nodes are used for radiative transfer calculations in participating media, since fine-grained meshes are not required. Since current massively parallel architectures employ up to 2^{16} processors, a direct node-to-processor association would utilize only a small portion of the computing power available. Not only is it advantageous to maximally utilize each available processor, but algorithms with large memory requirements (in which array size exceeds the memory available to the processors) can be employed since many parallel machines have virtual processing ability, i.e., a single physical processor can act as multiple processors. A parallel computational scheme in which each processor is associated with radiative exchange between nodes, and not to the nodes themselves, would make full use of each processor. Such an approach corresponds to an exchange factor-(zonal-) based computation, where both direct and indirect radiative exchange between surfaces and volumes are calculated. The simplicity of this method as applied to parallel computing lies in the formulation of a unified matrix for total exchange factors. Within this context, parallel computations are programmed without regard to the dimensionality of the problem, so the same program can be used for one-, two-, or three-dimensional geometries. A unified matrix for zone-to-zone exchange factors has been formulated by Naraghi and Chung,[7] and could be applied directly to parallel computing. In this work we present a unified matrix formulation for node-to-node-based radiative exchange in isotropically scattering homogeneous media using the discrete exchange factor (DEF) method,[8] and compare computational implementation on serial and parallel computing machines.

II. Unified Matrix Formulation

The original formulation of the zone method, developed by Hottel and Sarofim,[9] offered an accurate and versatile method for radiative transfer analysis when it was first introduced in 1958. Since this time a number of modifications to this technique have been introduced, expanding the flexibility of the method for treating practical problems (including irregular geometries, inhomogeneous media, and anisotropic media), while concurrently simplifying programming. The literature contains many techniques for simplifying the zone method; a recent summary of some zone method extensions can be found in Ref. 10. For the sake of generalization, we refer to the zone method and its modifications as "exchange factor" methods, since radiative exchange between distinct regions can be expressed in terms of exchange factor matrices. This is an attractive feature of the exchange factor approach for massively parallel computing, since each matrix element can be associated to a processor.

A number of formulations have been proposed for exchange factor matrices. Noble[11] introduced an explicit matrix formulation for the zone method which significantly simplified programming. Further reducing programming, Naraghi and Chung[7] formulated a unified total exchange area matrix $\overline{ZZ}$ based on Markov chain theory

$$\overline{ZZ} = AB[B - \overline{zz}R]^{-1}\overline{zz}A \tag{1}$$

where

$$\overline{zz} = \begin{bmatrix} \overline{ss} & \overline{sg} \\ \overline{gs} & \overline{gg} \end{bmatrix} \tag{2}$$

is the partitioned matrix of direct exchange areas and the notation $\overline{zz}$ is used to represent the four mechanisms of direct radiative exchange between zones, 1) surface to surface $\overline{ss}$, 2) surface to gas $\overline{sg}$, 3) gas to surface $\overline{gs}$, and 4) gas to gas $\overline{gg}$. Equivalently, the partitioned matrix of total exchange areas which includes direct radiative exchange between zones, e.g., wall reflection and volumetric scattering, can be expressed as

$$\overline{ZZ} = \left[\begin{array}{c|c} \overline{SS} & \overline{SG} \\ \hline \overline{GS} & \overline{GG} \end{array}\right] \tag{3}$$

The reflection matrix

$$R = \left[\begin{array}{c|c} \rho I & 0 \\ \hline 0 & \omega_0 I \end{array}\right] \tag{4}$$

consists of wall reflectivity terms ρ along the diagonal in the upper left quadrant, and a volumetric scattering albedo ω_0 as a diagonal expression in the lower left quadrant, and the absorptivity matrix is simply $A = I - R$. Similarly, an equivalent area matrix can be expressed as

$$B = \left[\begin{array}{c|c} \mathrm{A} & 0 \\ \hline 0 & 4K_t\nu \end{array}\right] \tag{5}$$

where A is the diagonal matrix of surface areas, ν is the diagonal matrix of zone volumes, and K_t is the extinction coefficient of the gas. The heat flux and emissive powers of the zones are related by

$$\begin{pmatrix} \underline{q}_s \\ \underline{q}_g \end{pmatrix} = [AB - \overline{ZZ}^T] \begin{pmatrix} \underline{E}_s \\ \underline{E}_g \end{pmatrix} \tag{6}$$

where

$$\underline{q}_s^T = \{q_{s_1} q_{s_2} q_{s_3} \cdots q_{N_s}\} \tag{7}$$

$$\underline{q}_g^T = \{q_{g_1} q_{g_2} q_{g_3} \cdots q_{N_g}\} \tag{8}$$

are vector terms representing the heat flux at the N_s surfaces and the N_g volumes, and

$$\underline{E}_s^T = \{E_{s_1} E_{s_2} E_{s_3} \cdots E_{N_s}\} \tag{9}$$

$$\underline{E}_g^T = \{E_{g_1} E_{g_2} E_{g_3} \cdots E_{N_g}\} \tag{10}$$

correspond to surface and gas emissive powers.

Similarly, we introduce a unified matrix approach for a DEF matrix. The motivation stems from the increased versatility the node-to-node approach of the DEF method offers over a zone-to-zone approach. Programming is easier and computer analysis faster (since area and volume integration is not required), irregular geometries are easily analyzed, and grids can be made compatible with those for other heat transfer modes.[12] The explicit unified matrix, expressed in terms of a direct exchange factor matrix based on discrete nodal positions (for explicit expressions for direct exchange factors see Ref. 8), can be formulated as

$$\overline{DZZ} = [I - \overline{dzz}RW]^{-1}\overline{dzz}A \tag{11}$$

where

$$\overline{dzz} = \left[\begin{array}{c|c} \overline{dss} & \overline{dsg} \\ \hline \overline{dgs} & \overline{dgg} \end{array}\right] \tag{12}$$

$$\overline{DZZ} = \left[\begin{array}{c|c} \overline{DSS} & \overline{DSG} \\ \hline \overline{DGS} & \overline{DGG} \end{array}\right] \tag{13}$$

Note that in the zonal formulation we used the notation $\overline{zz}$ to denote direct exchange between finite zones, and in the discrete exchange factor formulation we use the notation $\overline{dzz}$ to represent direct exchange between nodes. We continue with this convention, where $\overline{dss}$ is similar to $\overline{ss}$, etc. The weight factor matrix W is

$$W = \left[\begin{array}{c|c} W_s & 0 \\ \hline 0 & W_g \end{array}\right] \tag{14}$$

where the subscripts s and g refer to surface and gas, respectively. In terms of the heat flux and emissive powers

$$\begin{Bmatrix} \underline{q}_s \\ \underline{q}_g \end{Bmatrix} = [I - \overline{\mathrm{DZZ}}^T W] \begin{Bmatrix} \underline{E}_s \\ \underline{E}_g \end{Bmatrix} \tag{15}$$

Unlike the zonal formulation, the DEF formulation results in dimensionless expressions.

III. Parallel Processing

The concise nature of the unified matrix formulation presented above makes parallel programming straightforward. Once the direct exchange factor, reflectivity, absorptivity, and weight factor matrices have been established (note, since R, A, and W are diagonal matrices, for programming purposes they may be formatted as vectors), the only programming tasks are matrix operations. Two methods of solution for finding $\overline{DZZ}$ in Eq. (11), and the heat flux or emissive powers in Eq. (15), are direct matrix inversion and solution of the linear system $Ax = B$. The CM Scientific Software Library contains routines for matrix inversion using Gauss-Jordan reduction, matrix multiplication, and linear system solvers for dense matrices. Matrix operations require only single statements for each matrix operation since FORTRAN 90 is array oriented.

Consider an $n \times n$ grid for radiative analysis. The unified matrix approach requires $(4n + n^2) \times (4n + n^2)$ terms in the $\overline{DZZ}$ matrix. When the total number of matrix elements exceeds the total number of processors available for computation, a virtual processing mechanism allows physical processors to simulate more than one matrix element by dividing the memory of each physical processor. Thus, if 2^{16} (65,536) processors are available for computation, each processor will represent approximately $(4n + n^2)^2/2^{16}$ matrix elements. It is precisely this ability to share the enormous amount of element computations among the many thousands of physical processors that makes memory intensive algorithms, such as the DEF and zonal approaches, attractive for use on parallel architectures.

IV. Timing Comparisons

To investigate the computional efficiency of a parallelized DEF unified matrix formulation, we investigate a rectangular enclosure with one hot wall (dimensionless emissive power of unity—surface 1) and three cold walls (dimensionless emissive power of zero—surfaces 2–4), where each wall has an emissivity of $\frac{1}{2}$. Surfaces 1 and 3 can be considered top and bottom faces, and surfaces 2 and 4 side faces. The optical thickness in each direction is unity, the gas scattering albedo is $\frac{1}{2}$, and there is no internal heat generation.

A 16,384 (2^{15}) processor CM-2 operating at a clock frequency of 7 MHz, with 256K of memory and a 64-bit Weitek floating-point accelerator, is used to test the parallel algorithms, and the CRAY Y-MP, with 256 Megawords of 64-bit word and 8 processors, is employed for the purpose of comparison with a serial processor. Although the CRAY Y-MP has coarse parallel processing capability, all work is performed serially. Compilations on the CM are performed under the slicewise model.[13] Fair and consistent timing comparisons between different computer architectures and algorithmic

Table 1 Heat flux and emissive power for different grid sizes

	Position	11 × 11	21 × 21	31 × 31
Surface 1 dimensionless heat flux	0.0	0.766	0.767	0.768
	0.1	0.757	0.765	0.764
	0.2	0.756	0.758	0.759
	0.3	0.749	0.754	0.753
	0.4	0.748	0.751	0.752
	0.5	0.745	0.750	0.750
Surface 2 dimensionless heat flux	0.0	0.445	0.446	0.447
	0.1	0.387	0.390	0.390
	0.2	0.348	0.347	0.347
	0.3	0.314	0.312	0.312
	0.4	0.282	0.281	0.281
	0.5	0.256	0.254	0.255
	0.6	0.231	0.231	0.231
	0.7	0.211	0.210	0.210
	0.8	0.191	0.191	0.191
	0.9	0.173	0.172	0.172
	1.0	0.147	0.146	0.146
Surface 3 dimensionless heat flux	0.0	0.174	0.175	0.175
	0.1	0.197	0.203	0.202
	0.2	0.218	0.220	0.221
	0.3	0.225	0.232	0.231
	0.4	0.236	0.239	0.240
	0.5	0.234	0.241	0.241
Centerline dimensionless emissive power	0.0	0.921	0.931	0.918
	0.1	0.793	0.784	0.783
	0.2	0.698	0.696	0.696
	0.3	0.621	0.621	0.620
	0.4	0.556	0.556	0.556
	0.5	0.500	0.500	0.500
	0.6	0.452	0.453	0.453
	0.7	0.411	0.412	0.413
	0.8	0.377	0.378	0.379
	0.9	0.345	0.349	0.350
	1.0	0.308	0.307	0.311

Table 2 CPU Timing comparison

		Matrix inversion		LU Decomposition	
Grid	$4n + n^2$	CM-2	CRAY Y-MP	CM-2	CRAY Y-MP
11 × 11	165	2.066 s	1.256 s	3.491 s	1.192 s
21 × 21	525	26.337 s	15.465 s	13.819 s	13.011 s
31 × 31	1085	178.878 s	82.760 s	30.877 s	63.370 s

implementations of a given solution technique are difficult to establish. Because of software and hardware differences we have calculated the solution using two established techniques on commonly available software packages. On the CM, the CMSSL library routine for matrix inversion (GEN_GJ_INVERT) and LU decomposition (GEN_LU_FACTOR and GEN_LU_SOLVE) are used.[14] Partial pivoting strategies were used for each method. To compliment these techniques, on the CRAY we use the IMSL routines LINRG for matrix inversion and LFTRG/LFSRG for LU factorization and solving.

Dimensionless heat flux at the walls and emissive power values along the centerline (from surface 1–3) are listed in Table 1 for three grids, each with nodes equally distributed: 11 × 11, 21 × 21, and 31 × 31. The values obtained using the different solution techniques were essentially the same on both the CM and CRAY.

The CPU time in seconds for the different solution techniques is shown in Table 2 (only calculations involving the matrix operations are timed; all *i/o* operations for calculation of direct exchange factors are performed on the front end, for which no account is taken). The timing results shown in Table 2 should be viewed only as an indication of performance since the results of using different algorithms on dissimilar machines depend on a number of factors, e.g., programming style, overhead, and setup costs. Nevertheless, a few general statements can be made regarding the results. Although the processors on the CRAY Y-MP are much more powerful than those of the CM-2, the shear number of processors available for concurrent computing shows the CM-2 performance to be computationally compatible with the CRAY. These results are significant since the cost of computing on the CRAY is generally much greater than on the CM. As the grid size increases, relative performance actually improves since the communications pattern and calculation instructions do not scale with the calculation speed. Let N be the order of the matrix which is solved, i.e., $4n + n^2$, and t be the CPU time to compute the solution using this matrix. If i and j are two calculations using the same method but different grid sizes on the same machine, then the exponent p given in

$$\left(\frac{N_i}{N_j}\right)^p = \frac{t_i}{t_j} \tag{16}$$

yields the effective order of the algorithm.[15] We find that for $N_i/N_j = \frac{1085}{525}$, $p_{CM} = 2.6$ and $p_{CRAY} = 2.3$ using matrix inversion and $p_{CM} = 1.1$ and $p_{CRAY} = 2.2$ using LU decomposition. For $N_i/N_j = \frac{525}{165}$, $p_{CM} = 2.2$ and $p_{CRAY} = 2.2$ using matrix inversion and $p_{CM} = 1.2$ and $p_{CRAY} = 2.0$ using LU decomposition. It is interesting to observe that LU decomposition yields essentially a linear relationship between CPU

time and array size on the CM, while the other algorithms are essentially between 2nd and 3rd order.

The optimum virtual processing ratio VP (where VP is defined as the number of virtual processors per physical processor) is in multiples of 4, since the 64-bit floating point chips have a vector length of 4 in the slicewise model. In terms of computing speed, the number of matrix elements should be as closely matched to the available number of processors, or a factor of 4 times the number of available processors, as possible (this assumption neglects communication and overhead costs). For problems where a large number of nodes are needed, it is wise to develop a computational approach which accounts for resource allocation, developing adaptive grid strategies with this in mind.

V. Conclusions

The results presented illustrate the performance possible using matrix-formulated radiative transfer calculations on a parallel computer architecture. These architectures have the potential of changing the nature of radiative transfer calculations which are CPU and memory intensive. Computations involving complex geometries will be more manageable and much quicker. Parallel computing will be especially effective for nonlinear problems, e.g., those which have temperature-dependent properties. These machines also allow for simpler programming, improving software readability, and reducing source code size in comparison to code written for FORTRAN 77 compilers.

Acknowledgments

C. Saltiel wishes to acknowledge support from the Supercomputer Computations Research Institute at Florida State University for use of its CM-2 and the NASA Ames Research Center for use of its CRAY Y-MP. We also thank Joseph Kolibal of the University of Florida and Ken Jones of the Thinking Machines Corporation for enlightening discussions concerning numerical aspects of parallel computations.

References

[1]Tong, T. W., and Skocypac, R. D., "Summary on Comparison of Radiative Heat Transfer Solutions for a Specified Problem," *Developments in Radiative Heat Transfer, HTD-Vol. 203*, ASME National Heat Transfer Conf., San Diego, CA, Aug. 1992, pp. 253–264.

[2]Simon, H. S. (ed.), *Scientific Applications of the Connection Machine*, World Scientific Publ., Teaneck, NJ, 1989.

[3]Metcalf, M., and Reid, J., *Fortran 90 Explained*, Oxford Univ. Press, Oxford, England, UK, 1990.

[4]Howell, J. R., "Thermal Radiation in Participating Media: The Past, the Present, and Some Possible Futures," *Journal of Heat Transfer*, Vol. 110, No. 4B, 1988, pp. 1220–1229.

[5]Burns, P. J., and Pryor, D. V., "Vector and Parallel Monte Carlo Radiative Heat Transfer Simulation," *Numerical Heat Transfer, Part B*, Vol. 16, 1989, pp. 97–124.

[6]Hannebutte, U. R., and Lewis, E. E., "A Massively Parallel Algorithm for Radiative Heat Transfer Calculations," *91-WA-HT-10*, ASME Winter Annual Meeting, Dec. 1–6, Atlanta, GA, 1991, pp. 1–7.

[7]Naraghi, M. H. N., and Chung, B. T. F., "A Unified Matrix Formulation for the Zone Method: A Stochastic Approach, *International Journal of Heat and Mass Transfer*, Vol. 28, No. 2, 1985, pp. 245–251.

[8]Naraghi, M. H. N., Chung, B. T. F., and Litkouhi, B., "A Continuous Exchange Factor Method for Radiative Exchange in Enclosures with Participating Media," *Journal of Heat Transfer*, Vol. 110, No. 2, 1988, pp. 456–462.

[9]Hottel, H. C., and Cohen, C. S., "Radiant Heat Transfer in a Gas Filled Enclosure: Allowance for Nonuniformity of Gas Temperature," *AIChE Journal*, Vol. 4, No. 1, 1958, pp. 3–13.

[10]Siegel, R., and Howell, J. R., *Thermal Radiation Heat Transfer, 3rd Edition*, Hemisphere, Philadelphia, PA, 1992.

[11]Noble, J. J., "The Zone Method: Explicit Matrix Relations for Total Exchange Areas," *International Journal of Heat and Mass Transfer*, Vol. 18, No. 2, 1975, pp. 261–269.

[12]Saltiel, C., and Naraghi, M. H. N., "Analysis of Radiative Heat Transfer in Participating Media Using Arbitrary Nodal Distribution," *Numerical Heat Transfer, Part B*, Vol. 17, 1990, pp. 227–243.

[13]*CM Fortran Optimization Notes: Slicewise Model*, Thinking Machines Corp., Cambridge, MA, 1991, Version 1.0.

[14]*CMSSL for CM Fortran*, Thinking Machines Corp., Cambridge, MA, 1991, Version 2.2.

[15]Lipson, J. D., *Elements of Algebra and Algebraic Computing*, Addison-Wesley, Reading, MA, 1981, p. 194.

Internal Structure of Shock Waves in Disparate Mass Mixtures

Chan-Hong Chung*
NASA Lewis Research Center, Cleveland, Ohio 44135
Kenneth J. De Witt† and Duen-Ren Jeng‡
University of Toledo, Toledo, Ohio 43606
and
Paul F. Penko§
NASA Lewis Research Center, Cleveland, Ohio 44135

Introduction

THE direct-simulation Monte Carlo (DSMC) method developed by Bird[1] is an important technique for numerical simulation of rarefied gas flows. In DSMC simulations, a molecular model is required to calculate collisions between molecules, and the most widely used model is the variable hard sphere (VHS) model of Bird.[2] There are several models which can be considered as extensions of the VHS model, such as the generalized hard sphere model,[3] in which both attraction and repulsion forces are incorporated, or the variable soft sphere model,[4] which is compatible with both the diffusion and viscosity coefficients. However, in predicting the behavior of gas mixtures, there is an unfavorable aspect in the models, i.e., all of the collisions between molecules in a mixture are treated to follow only one common type of molecular interaction rather than differentiating depending on the type of interaction. The realistic treatment of molecular interactions can be an important factor in simulating flowfields involving large changes in mole fraction, especially if detailed behavior of individual components in a gas mixture is to be predicted. To overcome this kind of limitation, Erwin[5] extended an intermolecular-potential-based approach to binary mixtures where heteromolecular collision cross sections obtained from the corresponding intermolecular potentials were employed. The application of the method, however, requires a large cross-sectional data table for each class of molecular collision, and is limited to monatomic gas mixtures. The particle simulation method proposed by McDonald[6] can, in principle, employ different molecular interactions for different kinds of molecular collisions. The method, however, has not been applied to gas mixtures with different molecular interactions. Recently, Chung et al.[7] proposed the variable diameter hard sphere (VDHS) model which can be considered

Presented as Paper 92-0496 at the AIAA 30th Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 6–9, 1992; received March 26, 1992; revision received Feb. 16, 1993; accepted for publication March 15, 1993.

*Resident Research Associate, Computational Methods for Space Branch. Member AIAA.

†Professor, Department of Chemical Engineering. Member AIAA.

‡Professor, Department of Mechanical Engineering.

§Aerospace Engineer, Low Thrust Propulsion Branch. Member AIAA.