STUDIES OF ELECTRON-MOLECULE COLLISIONS ON THE MARK IIIfp HYPERCUBE

Paul Hipes, Carl Winstead, Marco Lima, and Vincent McKoy
Noyes Laboratory of Chemical Physics
California Institute of Technology
Pasadena, California 91125

Abstract

We report on a distributed memory implementation and initial applications of a program for calculating electron-molecule collision cross sections. Runs on the Mark IIIfp hypercube show that large-grain MIMD machines are well suited for these applications. Some results of studies of $e^-$$-$$Si_2H_6$ and $e^-$$-$$SiF_4$ collisions will be discussed.

I. Introduction

We have developed a distributed memory implementation of a computer code which we have been using to study the collisions of low-energy electrons with molecules. Here we report on our strategy for porting this code to the JPL/Caltech Mark IIIfp hypercube, our experiences with the parallel conversion, and some initial results which illustrate the level of performance achieved. The original FORTRAN program is based on a multichannel extension of the variational principle for collisions originally introduced by Schwinger [1]. This code, which currently runs in production mode on CRAY machines, has been used extensively in recent years to study both elastic and inelastic scattering of low-energy electrons by molecules such as H_2, N_2, CO, H_2O, CH_4, C_2H_4, and C_2H_6.

Our motivations for building a hypercube version of our code for studying electron-molecule collisions include, on the one hand, the high cost of cycles on CRAY-type machines and their inherent limitations in expected CPU throughput due to the recursive character of the computationally intensive step of the calculations, and on the other hand, the potentially high performance of large-grain MIMD machines such as the NCUBE, iPSC, or the Mark IIIfp for this application, whose structure lends itself naturally to a MIMD architecture. The high-performance and cost-effective computing offered by these machines are enhancing our ability to study cross sections for collisions of electrons with industrially important gases, e.g., C_2F_6, Si_2H_6, and CF_3H. Such cross sections play an important role in modelling low-temperature plasmas used in plasma-assisted etching and deposition in microelectronic fabrication.

II. Background

The collision of an electron with a molecular target A may be illustrated schematically as

$$e^-(E_m,\ \vec{k}_m) + A \longrightarrow e^-(E_n,\ \vec{k}_n) + A^*,$$

where the electron initially travels with kinetic energy E_m along the direction specified by the vector $\vec{k}_m$, and, following the collision, leaves the molecule along direction $\vec{k}_n$ with energy E_n. The asterisk on A indicates that the molecule may be rotationally, vibrationally, or electronically excited by the collision, in which case $E_n < E_m$; collisions for which $E_n = E_m$ are referred to as elastic.

The Schwinger multichannel (SMC) procedure [2,3] is a variational method specifically formulated for obtaining the probabilities, or cross sections, for low-energy electron-molecule collision events, including elastic scattering and vibrational or electronic excitation. The SMC method is applicable to molecules of arbitrary geometry, and is capable of incorporating effects arising from polarisation of the target by the incident electron, which are particularly important at the lowest energies (approximately 0–5 eV).

In the SMC procedure, the scattering amplitude $f(\vec{k}_m, \vec{k}_n)$, whose square modulus is proportional to the cross section, is obtained in the form

$$f(\vec{k}_m, \vec{k}_n) = -\frac{1}{2\pi}\sum_{i,j}\langle S_m(\vec{k}_m)|V|\chi_i\rangle (\mathbf{A}^{-1})_{ij}\langle\chi_j|V|S_n(\vec{k}_n)\rangle,$$

0-8186-2113-3/90/0000/0498$01.00 © 1990 IEEE

where $S_m(\vec{k}_m)$ is an $(N+1)$-electron interaction-free wave function of the form

$$S_m(\vec{k}_m) = \Phi^{(m)}_{target}(1,2,\ldots,N)e^{i\vec{k}_m\cdot\vec{r}_{N+1}},$$

V is the interaction potential between the electron and the molecular target, and the $(N+1)$-electron functions χ_i are Slater determinants which form a basis set for approximating the exact scattering wave functions $\Psi^{(+)}_m(\vec{k}_m)$ and $\Psi^{(-)}_n(\vec{k}_n)$. The $(\mathbf{A}^{-1})_{ij}$ are elements of the inverse of the matrix representation in the basis χ_i of the operator

$$A^{(+)} = \frac{1}{2}(PV+VP) - VG^{(+)}_P V - \frac{1}{N+1}\{\hat{H} - \frac{N+1}{2}(\hat{H}P + P\hat{H})\}.$$

Here P is the projector onto open (energetically accessible) electronic states,

$$P = \sum_{\ell \in open} |\Phi_\ell(1,2,\ldots,N)\rangle\langle\Phi_\ell(1,2,\ldots,N)|,$$

$G^{(+)}_P$ is the $(N+1)$-electron Green's function projected onto open channels, and $\hat{H} = (E-H)$, where E is the total energy of the system and H is the full Hamiltonian.

In the present implementation, the Slater determinants χ_i are formed from molecular orbitals which are, in turn, combinations of Cartesian Gaussian orbitals

$$N_{\ell mn}(x-A_x)^\ell(y-A_y)^m(z-A_z)^n \times \exp(-\alpha|\vec{r}-\vec{A}|^2),$$

which are commonly used in molecular electronic-structure studies. With this choice, all matrix elements needed in the evaluation of $f(\vec{k}_m,\vec{k}_n)$ can be obtained analytically, except those involving the Green's-function term $VG^{(+)}_P V$. These terms are evaluated numerically via a momentum-space quadrature procedure [4]. Once all matrix elements are calculated, the final step in the calculation is solution of a system of linear equations to obtain the scattering amplitude $f(\vec{k}_m,\vec{k}_n)$.

The computationally intensive step in the above formulation is the evaluation of large numbers of so-called "primitive" two-electron integrals

$$\langle\alpha\beta|V|\gamma\vec{k}\rangle = \int\int d^3\vec{r}_1 d^3\vec{r}_2\ \alpha(\vec{r}_1)\beta(\vec{r}_1)\frac{1}{r_{12}}\gamma(\vec{r}_2)e^{i\vec{k}\cdot\vec{r}_2}$$

for all combinations of Cartesian Gaussians α, β, and γ, and for a wide range of $\vec{k}$ in both magnitude and direction. These integrals are evaluated analytically by an intricate "black box" comprising approximately two thousand lines of FORTRAN. A typical calculation might require 10^9 to 10^{10} calls to this integral-evaluation suite, consuming roughly 80% of the total computation time. Once the primitive integrals are obtained, they are assembled in appropriate linear combinations to yield the matrix elements appearing in the expression for $f(\vec{k}_m,\vec{k}_n)$. The original CRAY code performs this procedure in two steps: first, a repeated linear transformation to integrals involving molecular orbitals, followed by a transformation from the molecular-orbital integrals to the physical matrix elements involving Slater determinants. The latter step is equivalent to an extremely sparse linear transformation whose coefficients are determined in an elaborate subroutine with a complicated logical flow.

III. Concurrent Implementation

The necessity of evaluating large numbers of "primitive" two-electron integrals makes the SMC procedure a natural candidate for parallelization on a MIMD machine such as the Mark IIIfp hypercube. The large memory and general-purpose processors of the Mark IIIfp make it feasible to distribute the "black box" integral evaluator across the processors and to divide up the evaluation of the primitive integrals among all the processors. In planning the decomposition of the set of integrals onto the nodes of the hypercube, two principal issues must be considered. First, the number of integrals required is such that not all can be stored in memory simultaneously, and certain indices must therefore be processed sequentially. Second, the transformation from primitive integrals to physical matrix elements, which necessarily involves interprocessor communication, should be as efficient and transparent as possible. With both of these considerations in view, the approach chosen was to configure the hypercube as a logical two-torus, to which is mapped an integral matrix whose columns are labeled by Gaussian pairs (α,β), and whose rows are labeled by momentum directions $\hat{k}$; the indices $|\vec{k}|$ and γ are processed sequentially.

Given this choice of data decomposition, a design for the parallel transformation procedure must be chosen. Direct emulation of the sequential code—that is, transformation first to molecular-orbital integrals and then to physical matrix elements—is undesirable, because the latter step would entail an intricate parallel routine governing the complicated flow of a relatively limited amount of data between processors. The potential for coding errors would be unacceptably high. Instead, the two transformations are combined into a single step by using the logical outline of the original molecular-orbital–to–physical–matrix-element routine in a distributed version of the CRAY sequential routine which builds a distributed transformation matrix. The combined transformations are then accomplished by a series of large, almost-full complex-arithmetic matrix multiplications directly on the primitive-integral data set. The transformation steps and associated interprocessor communication are thus localised and "hidden" in large parallel multiplications, which are known to be efficient on hypercube architectures [13]. Besides efficiency, benefits of this approach include simplicity and enhanced portability of the resulting code.

The remainder of the parallel implementation involves relatively straightforward modifications of the sequential CRAY code, with the exception of a series of integrations over angles $\hat{k}$ arising in the evaluation of the $VG_P^{(+)}V$ matrix elements, and of the solution of a system of linear equations in the final phase of the calculation. The angular integration, done by Gauss-Legendre quadrature, is compactly and efficiently coded as a distributed matrix multiplication of the form $\mathbf{A}\mathrm{diag}(\omega_i)\mathbf{A}^{\dagger}$. The integration over $|\vec{k}|$ is essentially accomplished in SIMD fashion. The solution of the linear system will be performed by a distributed LU solver[14] modified for complex arithmetic, implementation of which is under way. This will make feasible solution of systems on the order of 2000 × 2000, on current hardware. However, in applications to date, the size of the linear systems—less than 100 × 100—has allowed use of the original sequential solver running either on the host or on a single node.

IV. Performance

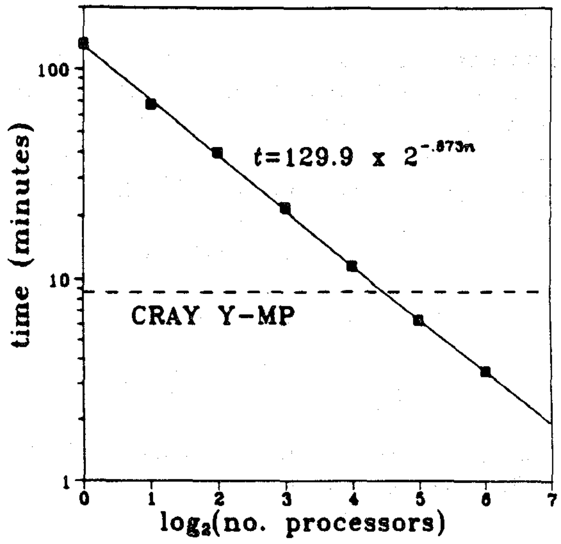

Fig. 1 Time for computation and transformation of a complete set of two-electron integrals, for fixed $|\vec{k}|$, as a function of the Mark IIIfp hypercube dimension (squares). Also shown is an exponential best fit (solid line), with parameters as indicated in the figure, and, for comparison, the single-processor CRAY Y-MP time (dashed line).

No attempt has been made to benchmark the parallel electron scattering code in detail. Such an exercise is irrelevant here, because the integrals are embarrassingly parallel, and matrix multiplies and LU decomposition have been previously assessed . However, its performance relative to the original CRAY code has been assessed through a series of calculations on Mark IIIfp hypercubes of dimensions from 0 (a single processor) to 6 (64 processors), the largest currently available. The same calculation was also performed on a CRAY Y-MP with the original code. For these comparisons, a modest but realistic "production run" for the CO molecule using 32 Cartesian Gaussian orbitals was chosen. Results are presented in Figs. 1 and 2. Figure 1 shows the time required for a single "quadrature shell" of integrals, *i.e.*, for evaluation and transformation of a complete set of two-electron integrals for a fixed magnitude $|\vec{k}|$, as a function of the cube dimension. All I/O and code loading are included in timings. The Weitek XL floating point processor performs the primitive integral calculation at roughly 0.85 Mflops per processor. The transformation to physical matrix elements proceeds at 1.5 Mflops/processor. The data of Fig. 1 are presented in an alternative fashion in Fig. 2, which shows speedup as a function

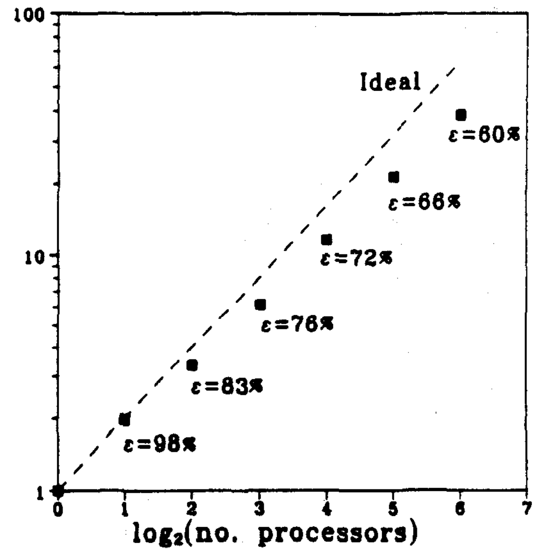

Fig. 2 Speedup as a function of hypercube dimension for the same case as Fig. 1. Efficiencies are indicated for each dimension. For comparison, ideal 2^n speedup is shown by the dashed line.

of cube dimension, along with efficiencies (the ratio of achieved to ideal, or 2^n, speedup). The (single-processor) Y-MP time is indicated by the dashed line. As seen from the figure, the Mark IIIfp performance surpasses that achieved on the CRAY in going from 16 to 32 processors. The solid line, which is an exponential best fit, evidently describes the observed Mark IIIfp times well over the range of hypercube dimensions studied, although the fact that the time decreases as $2^{-0.87n}$ rather than 2^{-n} indicates that the speedup achieved is less than ideal. An analogous plot for the total CO computation time on 8 to 64 processors (not shown) reveals identical characteristics, reflecting the dominance of the two-electron integrals in the calculation. As expected for a problem of fixed size, the efficiency declines as the hypercube dimension increases [5], but remains reasonable over the range studied. Most importantly, on 64 processors, we are outperforming the Y-MP by a factor of 3 on a small problem. Larger problems will provide a greater performance differential.

V. Selected Results

After development and debugging, the concurrent SMC code was applied to a number of elastic electron-scattering problems, with an emphasis on polyatomic gases of interest in low-temperature plasma applications [6]. Some of the systems examined to date are ethylene (C_2H_4), ethane (C_2H_6), disilane (Si_2H_6), and tetrafluorosilane (SiF_4). Illustrative results are presented in Figs. 3–5, along with experimental or other data for comparison [7-11]. Figure 3 shows the differential cross section—that is, scattering probability as a function of the angle θ between incident and outgoing directions—for 4 eV electrons colliding elastically with Si_2H_6 molecules. Agreement with recent experimental results [7] is excellent. One point to observe is the significant probability of scattering in the high-angle, or near-backward, directions, for which experimental data are unavailable. Examination of Fig. 3 suggests that extrapolation of the measured values to this region is likely to underestimate the cross section. This fact is significant because such backscattering makes a large contribution to the transfer of momentum from the electrons to the gas molecules and is therefore important in the numerical modeling of plasmas and discharges.

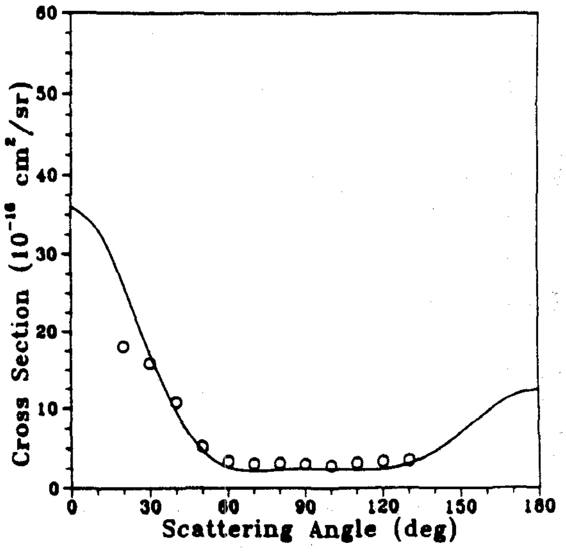

Fig. 3 Differential cross section for elastic scattering of 4eV electrons by the Si_2H_6 molecule. The solid line shows theoretical results obtained on the Mark IIIfp; the circles are measured values (Ref. [7]).

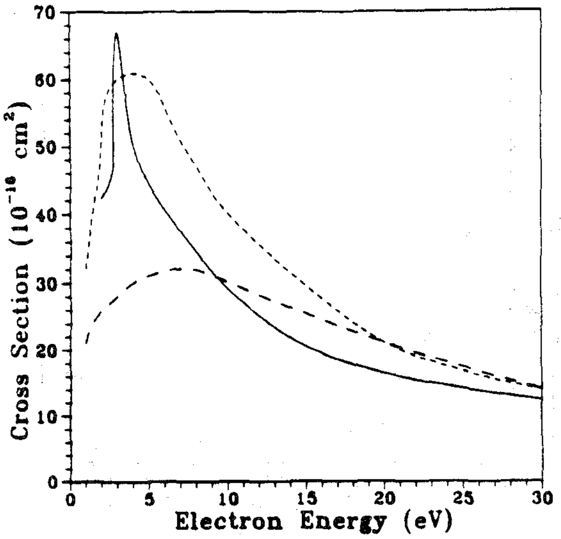

Fig. 4 Momentum-transfer or diffusion cross sections for low-energy electrons colliding with Si_2H_6. Shown are the present results (solid line), estimated values [8] (long dashes), and derived values [9] (short dashes).

The large backscattering probability indicated in Fig. 3 contributes to the peak in the Si_2H_6 momentum-transfer cross section—essentially a weighted integral over the differential cross section—shown in Fig. 4 as a function of electron energy. The dashed curves in Fig. 4, which represent estimated [8] and indirectly derived [9] momentum-transfer cross sections, appear to be the only previously published values for this industrially important molecule, highlighting the need for calculations of the present type.

As a further example of the applications performed to date, Fig. 5 shows preliminary results for the angle-integrated elastic scattering cross section of SiF_4, along with two measurements [10,11] of the total scattering cross section, which should of course be larger than the elastic cross section. Considering the uncertainties in the measurements and the need for further refinement of the theoretical result, the agreement in magnitude and overall shape of the cross sections are quite encouraging.

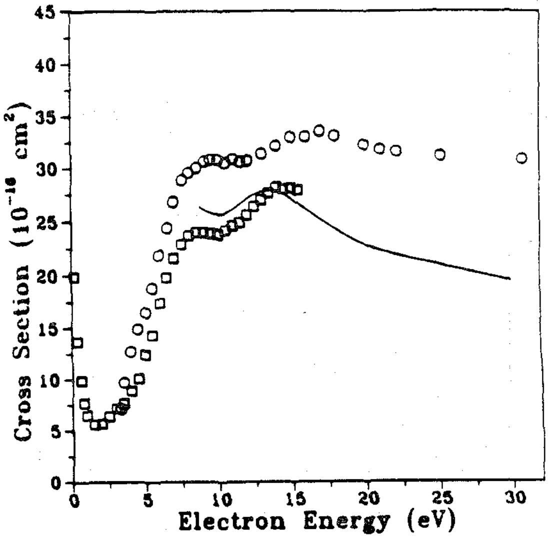

Fig. 5 Elastic electron scattering cross section for SiF_4 obtained on the Mark IIIfp (solid line). Also shown are total scattering cross section measurements of Refs. [10] (squares) and [11] (circles).

VI. Conclusions and Future Prospects

The concurrent implementation of a large sequential code which is in production on CRAY-type machines is an example of challenges which are likely to become increasingly frequent as commercial parallel machines proliferate and as more and more "mainstream" computer users are attracted by their potential. Several lessons which emerge from the port of the SMC code may prove useful to those contemplating similar projects. One is the value of focusing on the concurrent implementation of the existing code [11] and, so far as possible, maintaining the structure and code from the sequential program. The development of an understanding of the original CRAY code and its organisation is a demanding part of such a parallelisation. On the other hand, major issues of structure and organisation which bear directly on the parallel conversion deserve very careful attention. In the SMC case, the principal such issue was how to implement efficiently the transformation from primitive integrals to physical matrix elements. A poor parallelisation of the transformation could offset the high efficiency of the primitive integral calculation. The solution arrived at not only implied that a significant departure from the sequential code was warranted but also suggested the data decomposition. One conclusion is that similar code reorganization–building and multiplying large matrices–would improve the execution on the CRAY. In contrast, the primitive integral evaluation could not be significantly improved for the CRAY because it is a recursive procedure; however, it was easily parallelized for a large grain machine. A final point worth mentioning is that the conversion was greatly facilitated by an environment which fostered collaboration between workers familiar with the original code and its application and workers adept at parallel programming practice, and in which there was ready access both to smaller machines for debugging runs and to larger, production machines.

Plans for the near future include the implementation of the distributed LU solver, already mentioned, and the implementation of portions of the sequential code necessary for studies of electronic excitation and for employing molecular symmetry to reduce computation. Subsequent steps will probably include optimisation of key sequential subroutines and transfer of the code to other parallel machines as they become available.

Acknowledgments

It is a pleasure to thank the following individuals for their encouragement and support: Don Austin of the Department of Energy; Terry Cole, Dave Curkendall, and Edith Huang of the Jet Propulsion Laboratory; and Geoffrey Fox, Paul Messina, and Heidi Lorenz-Wirzba of the Caltech Concurrent Computation Program. Financial support by the Applied Mathematical Sciences Program of the Department of Energy, the Innovative Science and Technology Program of SDIO through the Army Research Office, the National Science Foundation, and the Jet Propulsion Laboratory is also gratefully acknowledged.

References

[1] J. Schwinger, Phys. Rev. **72**, 742 (1947).

[2] K. Takatsuka and V. McKoy, Phys. Rev. A **24**, 2473 (1981).

[3] K. Takatsuka and V. McKoy, Phys. Rev. A **30**, 1734 (1984).

[4] M. A. P. Lima, L. M. Brescansin, A. J. R. da Silva, C. Winstead, and V. McKoy, Phys. Rev. A **41**, 327 (1990).

[5] G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Salmon, and D. Walker, *Solving Problems on Concurrent Processors*, vol. I (Prentice-Hall, Englewood Cliffs, N. J., 1988), p. 58.

[6] *Plasma Reactions and Their Applications*, Japan Materials Report by Japan Technical Information Service (ASM International, Metals Park, Ohio, 1988).

[7] H. Tanaka, L. Boesten, H. Sato, M. Kimura, M. A. Dillon, and D. Spence, *42nd Annual Gaseous Electronics Conference, Palo Alto, 1989*, and private communication.

[8] M. Hayashi, *Proceedings of the VI Dry Process Symposium, Tokyo, 1984.*

[9] M. Hayashi, in *Swarm Studies and Inelastic Electron-Molecule Collisions*, edited by L. Pitchford, V. McKoy, A. Chutjian, and S. Trajmar (Springer, New York, 1985), p. 167.

[10] H.-X. Wan, J. H. Moore, and J. A. Tossell, J. Chem. Phys. **91**, 7340 (1989).

[11] C. Ma, P. B. Liescheski, and R. A. Bonham, *XVI International Conference on the Physics of Electronic and Atomic Collisions, New York, 1989: Abstracts of Contributed Papers*, p. 356, and private communication.

[12] G. Fox, private communication.

[13] P. Hipes, "Matrix Multiply on the JPL/Caltech Mark IIIfp Hypercube." C3P-746. G. Fox, A. J. G. Hey, and S. Otto, *Parallel Computing* **4**, 17 (1987).

[14] P. Hipes, "Comparison of LU and Gauss–Jordan System Solvers for Distributed Memory Multicomputers." C3P-652b.