SAND--90-0632C

DE90 009322

APPLICATION OF PARALLEL COMPUTING TO THE MONTE CARLO SIMULATION OF ELECTRON SCATTERING IN SOLIDS: A RAPID METHOD FOR PROFILE DECONVOLUTION

A. D. Romig, Jr.[1], S. J. Plimpton[1], R. L. Myklebust[2] and D. E. Newbury[2]

[1]Sandia National Laboratories, Albuquerque, NM 87185
[2]National Institute of Standards and Technology, Gaithersburg, MD 20234

X-ray microanalysis by analytical electron microscopy (AEM) has proven to be a powerful tool for characterizing the spatial distribution of solute elements in materials. True compositional variations over spatial scales smaller than the actual resolution for microanalysis can be determined if the measured composition profile is deconvoluted. Explicit deconvolutions of such data, via conventional techniques such as Fourier transforms, are not possible due to statistical noise in AEM microanalytical data. Hence, the method of choice is to accomplish the *deconvolution* via *iterative convolutions*.[1] In this method, a function describing the assumed true composition profile, calculated by physically permissible thermodynamic and kinetic modeling, is convoluted with the x-ray generation function and the result compared to the measured composition profile. If the measured and calculated profiles agree within experimental error, it is assumed that the true compositional profile has been determined. If the measured and calculated composition profiles are in disagreement, the assumptions in the physical model are adjusted and the convolution process repeated. To employ this procedure it is necessary to calculate the x-ray generation function explicitly. While a variety of procedures are available for calculating this function,[2-4] the most accurate procedure is to use Monte Carlo modeling of electron scattering. [5,6]

Monte Carlo Modeling: The basic principle of a Monte Carlo electron trajectory simulation is to calculate the path of each electron in a step wise manner, accounting for both elastic and inelastic scattering.[6] The distance between scattering events, the scattering angles and the rate of energy loss with distance are calculated from physical models. Random numbers are used to select the scattering angles so that, for a large number of electron trajectories, the resulting probability histogram closely resembles the actual angular distribution for single scattering events. To obtain physically meaningful results, a large number of electron trajectory simulations are required (usually on the order of 10^5). For electrons scattering in a solid, elastic scattering is primarily responsible for changing the direction of flight of the electron. The average value of the elastic scattering angle is typically of the order of 5°, but scattering through angles as large as 180° is possible. Elastic scattering events do not alter the energy of the electron. Inelastic scattering decreases the energy of the electron, but the scattering angle is very small, typically less than 0.1°. In this Monte Carlo model, the energy loss is described by the Bethe continuous energy loss function and any slight change in electron trajectory during an inelastic scattering event is ignored.

Parallel Computation of Monte Carlo Scattering: Monte Carlo simulations are obviously very computation intensive. Even in a thin film, a very large number of calculations are required to simulate the electron beam/specimen interaction for a single analysis point (i.e., 10^5 trajectories incident at a single point). For profile deconvolution, as many as 100 incident points may be required to calculate the composition profiles and up to 10 iterations using various physical models may be required. Hence, up to 10^8 trajectories may be required to determine a single composition profile. In this work, the simulations are performed on the NCUBE/ten which is a massively parallel supercomputer with an architecture particularly well suited to this type of problem. The key to using it effectively in a Monte Carlo simulation is that each electron trajectory is independent of all others. Since the NCUBE used in this research has 1024 processors, 1024 electron trajectories can be calculated simultaneously. Optimizing the code for the NCUBE required writing the appropriate communications protocol between the processors and adapting the random number generator used in the Monte Carlo algorithm.[7] In this work the results from the NCUBE are compared to three *conventional* computers, a DEC 3100 Workstation, a VAX 785 and a CRAY-XMP 4/16. The algorithm is written in standard Fortran 77; all versions were compiled using an optimizing compiler.

Example Simulations: Two separate classes of problems have been examined; electron scattering in bulk targets (such as found in scanning electron microscopy or electron probe microanalysis) and in thin films (such as used in AEM). The bulk specimen simulation considered the scattering of 10^5 20 keV electrons in Cu. The NCUBE required 262 s of CPU time, which was 25 times faster than the CRAY-XMP, 115 times faster than the DEC workstation and 311 times faster than the VAX 785. The CRAY code can be accelerated by a factor of approximately 8 by vectorizing it,[8] but this requires significant changes to the Monte Carlo algorithm. The second simulation considered the calculation of a composition profile in a thin film target. In this case the target was a 25 nm thick film of Al containing a 0.3 nm thick slab of Cu (simulating a Cu monolayer at a grain boundary in Al). The slab of Cu was oriented parallel to the incident electron beam. The results of this simulation are shown in Figure 1. The apparent composition at the boundary is approximately 63 wt.% Cu. Apparent compositions of less than approximately 0.1 wt. % would be below the background and not measurable. Hence, at distances exceeding 3 nm from the boundary the Cu signal has essentially decreased to zero. The NCUBE required 4.2 s of CPU time, which was was 4 times faster than the CRAY-XMP, 21 times faster than the DEC workstation and 68 times faster than the VAX 785. The gain in speed for the NCUBE relative to the other computers is less for thin foil simulations since the calculations cannot be perfectly balanced across the processors. The next generation NCUBE is approximately 5 times faster on this code than the current generation machine.

Discussion and Conclusions: The simulation of electron scattering in solids using Monte Carlo techniques is well suited to parallel computation. Significant gains in computation time are realized making explicit calculation of convoluted composition profiles possible. Computation time is sufficiently shortened so that such simulations may be used in a real-time experimental environment.

References:

1. D. B. Williams and A. D. Romig, Jr., Ultramicroscopy 30(1989)38.
2. J. I. Goldstein, et al., SEM/1977 1(1977)315.
3. A. D. Romig, Jr. and M. J. Cieslak, AEM - 1987, D. C. Joy, ed., San Francisco Press (1987) 25.
4. P. Doig, et al., Phil. Mag. 41A(1980)761.
5. D. E. Newbury, Principles of Analytical Electron Microscopy, ed. D. C. Joy, A. D. Romig Jr. and J. I. Goldstein, Plenum Press, New York (1986)1.
6. D. Newbury and R. L. Mykelbust, Microbeam Analysis - 1981, ed. R. H. Geiss, San Francisco Press, (1981)91.
7. R. Frederickson, et al., Parallel Computing 4(1987)281.
8. K. Miura, Computer Physics Communications, 45(1987)127.
9. A portion of this work performed at Sandia National Laboratories and supported by the US DOE under contract no. DE-AC04-76DP00789. The computational methods used here were developed with the support of the Applied Mathematical Science Program, DOE Office of Energy Research.

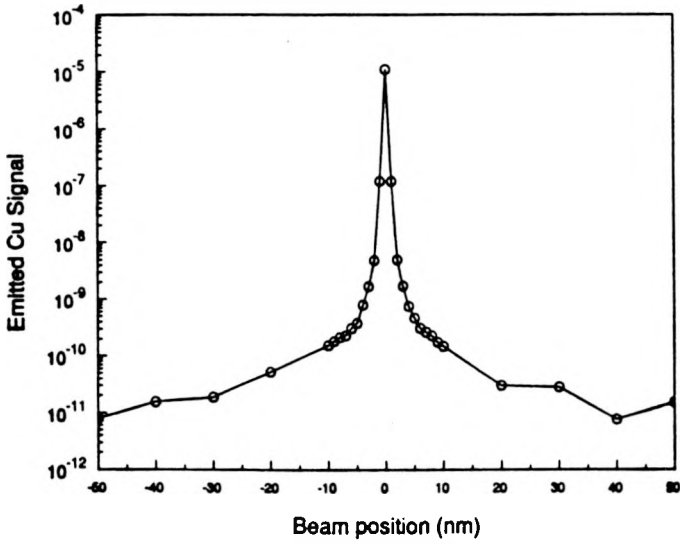

FIG. 1.-- Cu x-ray profile for a monolayer (0.3 nm) of Cu in a 25 nm thin film of Al. The beam energy is 100 keV and the incident spot size is 1 nm. Each point represents 10^5 electron trajectories. The vertical scale is Cu K_α x-rays per incident electron.

APPLICATION OF PARALLEL COMPUTING TO THE MONTE CARLO SIMULATION OF ELECTRON SCATTERING IN SOLIDS: A RAPID METHOD FOR PROFILE DECONVOLUTION

A. D. Romig, Jr.[1], S. J. Plimpton[1], R. L. Myklebust[2] and D. E. Newbury[2]

[1]Sandia National Laboratories, Albuquerque, NM 87185
[2]National Institute of Standards and Technology, Gaithersburg, MD 20234

X-ray microanalysis by analytical electron microscopy (AEM) has proven to be a powerful tool for characterizing the spatial distribution of solute elements in materials. True compositional variations over spatial scales smaller than the actual resolution for microanalysis can be determined if the measured composition profile is deconvoluted. Explicit deconvolutions of such data, via conventional techniques such as Fourier transforms, are not possible due to statistical noise in AEM microanalytical data. Hence, the method of choice is to accomplish the *deconvolution* via *iterative convolutions*.[1] In this method, a function describing the assumed true composition profile, calculated by physically permissible thermodynamic and kinetic modeling, is convoluted with the x-ray generation function and the result compared to the measured composition profile. If the measured and calculated profiles agree within experimental error, it is assumed that the true compositional profile has been determined. If the measured and calculated composition profiles are in disagreement, the assumptions in the physical model are adjusted and the convolution process repeated. To employ this procedure it is necessary to calculate the x-ray generation function explicitly. While a variety of procedures are available for calculating this function,[2-4] the most accurate procedure is to use Monte Carlo modeling of electron scattering.[5,6]

Monte Carlo Modeling: The basic principle of a Monte Carlo electron trajectory simulation is to calculate the path of each electron in a step wise manner, accounting for both elastic and inelastic scattering.[6] The distance between scattering events, the scattering angles and the rate of energy loss with distance are calculated from physical models. Random numbers are used to select the scattering angles so that, for a large number of electron trajectories, the resulting probability histogram closely resembles the actual angular distribution for single scattering events. To obtain physically meaningful results, a large number of electron trajectory simulations are required (usually on the order of 10^5). For electrons scattering in a solid, elastic scattering is primarily responsible for changing the direction of flight of the electron. The average value of the elastic scattering angle is typically of the order of 5°, but scattering through angles as large as 180° is possible. Elastic scattering events do not alter the energy of the electron. Inelastic scattering decreases the energy of the electron, but the scattering angle is very small, typically less than 0.1°. In this Monte Carlo model, the energy loss is described by the Bethe continuous energy loss function and any slight change in electron trajectory during an inelastic scattering event is ignored.

Parallel Computation of Monte Carlo Scattering: Monte Carlo simulations are obviously very computation intensive. Even in a thin film, a very large number of calculations are required to simulate the electron beam/specimen interaction for a single analysis point (i.e., 10^5 trajectories incident at a single point). For profile deconvolution, as many as 100 incident points may be required to calculate the composition profiles and up to 10 iterations using various physical models may be required. Hence, up to 10^8 trajectories may be required to determine a single composition profile. In this work, the simulations are performed on the NCUBE/ten which is a massively parallel supercomputer with an architecture particularly well suited to this type of problem. The key to using it effectively in a Monte Carlo simulation is that each electron trajectory is independent of all others. Since the NCUBE used in this research has 1024 processors, 1024 electron trajectories can be calculated simultaneously. Optimizing the code for the NCUBE required writing the appropriate communications protocol between the processors and adapting the random number generator used in the Monte Carlo algorithm.[7] In this work the results from the NCUBE are compared to three *conventional* computers, a DEC 3100 Workstation, a VAX 785 and a CRAY-XMP 4/16. The algorithm is written in standard Fortran 77; all versions were compiled using an optimizing compiler.