Real-Time Parallel Computation and Visualization of Ultrasonic Pulses in Solids

R. S. Schechter, H. H. Chaskelis, R. B. Mignogna, P. P. Delsanto

Parallel processing has changed the way much computational physics is done. Areas such as condensed matter physics, fluid dynamics, and other fields are making use of massively parallel computers to solve immense and important problems in new ways. Simulating wave propagation is another area that has benefited through the use of parallel processing. This is graphically illustrated in this article by various numerical simulations of ultrasonic pulses propagating through solids carried out on a massively parallel computer. These computations are accompanied by visualizations of the resulting wavefield. The calculations and visualizations, together, can be completed in only seconds to several minutes and compare well with experimental data. The computations and parallel processing techniques described should be important in related fields, such as geophysics, acoustics, and mechanics.

Mathematical models of our physical world cannot always be solved analytically. Analytic techniques usually require simplifications, which are not present in the real world. Problems also may involve huge amounts of data. Thus, researchers must rely more on numerical tools than in the past.

Massively parallel architectures are changing the way much computational physics is done. Many physical problems, by nature, lend themselves to parallel processing. For example, computations in condensed matter physics involve large numbers of degrees of freedom, all obeying the same basic principles (*1*). In Monte Carlo algorithms, frequently used in such problems, the rules for updating a degree of freedom are the same throughout the lattice (*1*). Because the physics is uniform or homogeneous in these problems, they can better be solved by parallel rather than serial computation. Data-parallel, high-level programming languages, such as CM Fortran, allow researchers to express operations in a parallel form and harness the power of a parallel computer without being intimately familiar with the details of the machines, thus greatly simplifying the programming task.

Parallel Processing in Physics

Two-dimensional (2D) strongly correlated electronic systems have attracted enormous attention recently, prompted by the recent discovery of high-temperature superconductivity (*2*, *3*). This area of quantum physics includes magnetic properties of solid crystals and bosons modeling superfluids and Cooper pairs in superconductors. This class of 2D quantum many-body systems rarely admits reliable analytical treatments. However, numerical simulations of these systems with quantum Monte Carlo techniques, programmed to take full advantage of parallel architectures (*4*), do provide quantitative results that can be compared with experimental results. These simulations have helped to describe and explain what happens inside these materials (*5*, *6*).

In computational fluid mechanics, massively parallel implementation of vortex methods has been developed, which offers an alternative to the traditional finite difference techniques for the calculation of viscous, incompressible, turbulent flow (*7*, *8*). The results can shed light on the fundamental issues in study of flows with high Reynolds numbers and on engineering issues such as wake structure, eddies, drag forces, and flows past bodies (*8*). The computational techniques rely on the parallel implementation of N-body solvers where the goal is to compute the total force on each body, which consists of the superposition of the individual forces exerted by all other bodies.

The N-body simulation has become a fundamental tool in the study of many complex systems. Bodies are understood to be mass, charge, or vortex elements. Starting from basic physical interactions such as gravitational, Coulombic, Biot-Savart, or van der Waals forces, one can follow the dynamical evolution of a system of N bodies (*9*). Much effort has been spent on the optimization of N body algorithms, especially for parallel architectures (*9*). Areas of application include accelerator beam dynamics, astrophysics (galaxy formation and large-scale structure), computational biology (protein folding), chemistry (molecular structure and thermodynamics), electromagnetic scattering, fluid mechanics (vortex method), molecular dynamics, and plasma physics. An N-body simulation is usually statistical in nature and requires a large number of bodies, typically millions, for accuracy. Such a simulation may also be deterministic in the case where the N bodies represent the entire system. These types of simulations have been used in astrophysics to model galaxy formation and, more recently, the formation of the structure of the early universe to determine the relative proportions of dark and luminous matter. Running on massively parallel supercomputers, these simulations have now become large enough and fast enough to give predictions sufficiently detailed to be compared with astronomical observations.

Molecular dynamics for the study of dynamical properties of solids and liquids has been a useful tool for decades, but it is only recently that massively parallel computers have made possible the study of realistically sized systems. A cube of 1000 atoms on a side only represents about 0.5 μm cubed. This is a very small volume of material, yet it contains 10^9 atoms. Realistic calculations in materials science require system sizes in this range if the dynamics of defects like dislocations and grain boundaries are to be studied. Systems of 10^8 particles interacting under the influence of a Lenard-Jones potential have been solved with a Connection Machine 5 (CM-5) (*10*). By carefully mapping this problem onto the architecture of the CM-5 and coding the kernel of the code (which implements the force calculation) in the assembler language of the CM-5 vector units, calculation rates of about 50 billion floating-point operations per second (50 Gflops) were obtained (*10*). This speed was obtained with 131 million particles, each interacting with 65 neighbors. Some 2D simulations of the fracture dynamics of a piece of material, with 2 million atoms, pulled apart in a fracture experiment have also been done (*10*).

Modeling Ultrasonic Wave Propagation

Wave propagation is very important in electromagnetics, acoustics, seismology, medical imaging, nondestructive testing of materials, and many other areas of physical science. The phenomenon of waves propagating is inherently local in the sense that a disturbance is propagated by locally interacting particles.

R. S. Schechter, H. H. Chaskelis, and R. B. Mignogna are at The Naval Research Laboratory, Washington, DC, 20375–5000, USA. P. P. Delsanto is at the Politecnico di Torino, 10129 Torino, Italy.

Expressed in a different way, the wave field only depends on local field values. There is no long-range potential involved, in contrast to the molecular dynamic problems described above, where the potential function may not fall off that rapidly with distance (this means that distant processors have to exchange information in a parallel computation). Because wave propagation is inherently local, only neighboring processors need to exchange information, a condition favorable for speed.

The physics of wave propagation is also uniform in that each particle obeys the same law, namely a wave equation. This is true even at boundaries between different materials, where the same equation holds but boundary conditions exist, which must be accounted for accurately. The physics at interfaces or boundaries is a major problem in the modeling of wave propagation, especially for parallel computation.

Our particular area of interest is in ultrasonic nondestructive evaluation (NDE), based on the propagation of elastic waves in solids. Analytical modeling has been done for certain problems in elastic wave propagation and scattering. However, it is virtually impossible to model and visualize by analytic means the full complexity of an ultrasonic transducer radiating into a solid and the extremely complex wavefield resulting from scatterers, and a large numerical simulation is required. The simulation must be both fast and accurate to perform computations spanning many wavelengths and to capture the complexities of a real material. To accomplish this, we developed a parallel processing approach.

Use of the Connection Machine to model wave propagation is rooted in two basic ideas. First, the dynamics of the physical fields described here are inherently parallel, in that each point modeled obeys the same physical law, namely the elastic wave equation. Second, the time evolution of the basic computed quantities depends only on the field values at the point considered and the immediate neighbors. Therefore, there is strong motivation for using fine-grain massive parallelism for wave propagation.

The parallel computer we used is a 16,384 (16K) processor version of the Connection Machine 200 (CM-200) manufactured by Thinking Machines Corporation. A full-size CM-200 consists of 65,536 processors, each with 128 kilobytes of memory, for a total of 8 Gbytes. There are 4096 processing chips, each with 16 processors, connected in a 12-dimensional hypercube architecture (*11*). Each pair of chips (32 processors) shares a floating point accelerator. The computer operates in SIMD (single instruction, multiple data) mode. This means that the same instruction stream is broadcast to every processor. At the Naval Research Laboratory (NRL), there is a 16K processor version of the full CM-200 with 2 Gbytes of memory.

The CM-200 supports two modes of communication. One is called NEWS (North East West South), which connects the processors in a multidimensional Cartesian grid and favors communication between nearest neighbors. Because a central finite difference equation in cartesian coordinates is used to discretize the wave equation, NEWS communication fits the algorithm well. The other available mode of communication is handled by a router, which allows general communication between any two processors, not just neighboring ones.

Finite Difference Formulation

Over the years, two computational finite difference approaches have been examined. One, called the homogeneous formulation, considers a heterogenous medium to be a collection of homogeneous linear isotropic regions each characterized by constant values of density and elastic parameters. The elastic wave equation for homogeneous isotropic solids governs the motion in each region. The boundary conditions across all interfaces separating different regions are satisfied explicitly (*12*).

The other, the heterogeneous approach, uses the elastic wave equation for heterogenous materials, where the elastic constants vary spatially. The result of constructing finite difference equations from the wave equation for heterogenous materials is that different density and elastic parameters may be associated with each grid point (*13*, *14*). The boundary conditions are satisfied implicitly. This approach, however, may give rise to large errors for cases involving large gradients in material properties across boundaries (*15*). The discretization across an interface depends on the distance over which the change in material properties is assumed to occur. In the vicinity of a boundary there is no unique way of discretizing the heterogeneous wave equation. To our knowledge, the literature indicates that the two approaches have never given identical results across a boundary separating two different materials.

Delsanto *et al.* (*15*) have derived a rigorous homogeneous formulation that imposes continuity of stresses and displacements across boundaries of four homogenous isotropic cells, each with different material properties, surrounding a crosspoint. This crosspoint formulation makes it possible to use the same difference equations at all points, including boundaries between different materials, where it is only necessary to change the coefficients. This formulation is the main ingredient for parallelizing the problem of elastic wave propagation. It also permits the modeling of geometrically complex interfaces such as scatterers by the simple use of a small enough grid spacing. These same difference equations may also be derived from the heterogenous wave equation with the use of the results of the homogeneous approach as a guide. Thus, the two approaches yield identical finite difference equations. Comparisons with other finite difference formulations are shown in (*15*).

Computation

The governing equation, used here to model ultrasonic waves in solids, is the elastic wave equation. In two dimensions, the equation reduces to two coupled equations for u and v

$$\rho \frac{\partial^2 u}{\partial t^2} = (\lambda + 2\mu)\frac{\partial^2 u}{\partial x^2} + \mu\frac{\partial^2 u}{\partial y^2} + (\lambda + 2\mu)\frac{\partial^2 v}{\partial x \partial y} \quad (1)$$

$$\rho \frac{\partial^2 v}{\partial t^2} = (\lambda + 2\mu)\frac{\partial^2 v}{\partial x^2} + \mu\frac{\partial^2 v}{\partial y^2} + (\lambda + 2\mu)\frac{\partial^2 u}{\partial x \partial y}$$

where u and v are the displacements in the x and y directions, t is time, λ and μ are the Lamé constants, and ρ is the density. This equation is valid for each of the homogeneous, isotropic cells, which collectively make up the heterogeneous material. When the boundary conditions (continuity of normal and tangential displacements and stresses across interfaces) are imposed, the resulting finite difference equation for $u(t + \Delta t,i,j)$ is of the form

$$\begin{aligned} u(t + \Delta t,i,j) &= c_1(i,j)u(t,i,j) \\ &+ c_2(i,j)u(t - \Delta t,i,j) + c_3(i,j)u(t,i + 1,j) \\ &+ c_4(i,j)u(t,i - 1,j) + c_5(i,j)u(t,i,j + 1) \\ &+ c_6(i,j)u(t,i,j - 1) + c_7(i,j)v(t,i,j) \\ &+ c_8(i,j)v(t,i + 1,j + 1) + c_9(i,j)v(t,i - 1, \\ &j + 1) + c_{10}(i,j)v(t,i - 1,j - 1) \\ &+ c_{11}(i,j)v(t,i + 1,j - 1) \\ &+ c_{12}(i,j)v(t,i + 1,j) + c_{13}(i,j)v(t,i,j + 1) \\ &+ c_{14}(i,j)v(t,i - 1,j) + c_{15}(i,j)v(t,i,j - 1) \end{aligned} \quad (2)$$

An analogous equation for $v(t + \Delta t,i,j)$ has also been derived. These are explicit step-ahead central difference equations. They are also called iteration equations. The $c(i,j)$'s are the appropriate weights or coefficients that make the solution valid at every node point, at interior nodes, at boundaries between different materials, and at free surfaces (stress-free boundaries). They depend on the material properties of the four cells surrounding the node point (i,j) (crosspoint) (*15*). Because the form of

these equations is identical at each node, they may be computed simultaneously. The CM Fortran compiler allocates the various arrays to processors. The $u(i,j)$ and $v(i,j)$, for example, are located in processor (i,j). The (i,j)'s are indices corresponding to spatial coordinates and represent a plane of processors in the CM. The indices $t - \Delta t$, t, and $t + \Delta t$ represent three consecutive times and are located serially down the processor's memory, giving rise to three spatial planes of displacements located in memory that change with every time step. In Eq. 2, only nearest neighbors in space, corresponding to nearest neighbor processors, have to exchange data, greatly minimizing communication, as is necessary for speed in parallel computations.

Discretization. The grid spacing is determined by the shortest wavelength occurring in the solid, which is related to the highest frequency in the pulse and the lowest shear wave velocity in the specimen. We use 17 mesh points per shortest wavelength to get sufficient accuracy. Once the lattice step size ε is determined, the time step Δt may be chosen according to the stability requirement for the difference equations, called the von Neumann stability criterion. For the finite difference equations used, this yields (*16*)

$$\Delta t \leq \frac{\varepsilon}{\sqrt{v_l^2 + v_t^2}} \qquad (3)$$

where v_l is the longitudinal wave velocity and v_t is the transverse or shear wave velocity.

Speed of computation. The finite difference equations have been timed on a CM-200 and more recently on a CM-5. A CM-200 with 64K processors executes the difference equations (Eq. 2) for a grid size of 1024 by 1024 node points for 1000 time steps at about 3.8 Gflops. This was obtained by timing the code fragment containing the difference equations on an 8K processor machine and linearly scaling this speed up by a factor of 8 to get the speed on a 64K processor machine. The same code fragment runs at 6.8 Gflops on a CM-5. This implies that the entire computational grid is updated every 10.8 ms. For a larger grid, 4046 by 4096 node points, the speed increases to 7.7 Gflops.

The CM-5 is a newer and more powerful parallel supercomputer than the CM-200, with an architecture based on from 32 to 1024 processor nodes networked together in a fat tree topology (so called because some branches are "fatter" with higher bandwidths than others). It can be programmed to run in MIMD (multiple instruction, multiple data) as well as SIMD mode. Each processor node has associated with it four floating-point vector units, four memory banks, and a network interface chip. The CM-5 at NRL (called CM-5E) is configured with 256 SPARC processor nodes each running at 40 MHz, with a combined memory of 32 Gbytes.

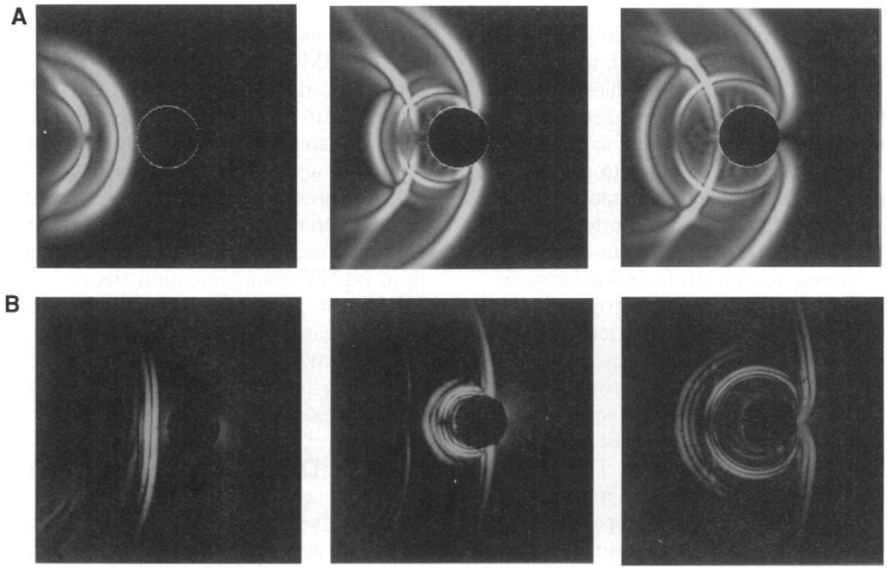

Fig. 1. (**A**) Simulated images of a longitudinal pulse scattering from a void in glass at three different times. (**B**) Experimental photoelastic images. [Reprinted from (*18*), Academic Press]

The code that we timed was written entirely in the data-parallel language CM-Fortran, with no special microcoded routines to further optimize communication. When special microcoded communication routines are used to implement the difference equations (stencil operations), considerably higher speeds even on a CM-200 are attainable (*17*).

A 1024 by 1024 grid requires 16 times the number of physical processors available on a CM-200. For this and larger grids, the memory of each processor is subdivided into many virtual processors (VP), simulating physical processors. The VP ratio is the number of virtual processors divided by the number of physical processors. For example, a 1K by 1K grid would require a VP ratio of 16 for a machine with 64K processors. For a CM with 8K processors, the VP ratio would be 128. The speed in terms of flops increases as the VP ratio increases, leveling off at a VP ratio that is dependent on the problem being addressed and the size of the CM machine being used. This increase in speed with VP ratio may be attributed to the fact that there is an initial time spent to broadcast the instructions to the processors or a start-up overhead, after which each

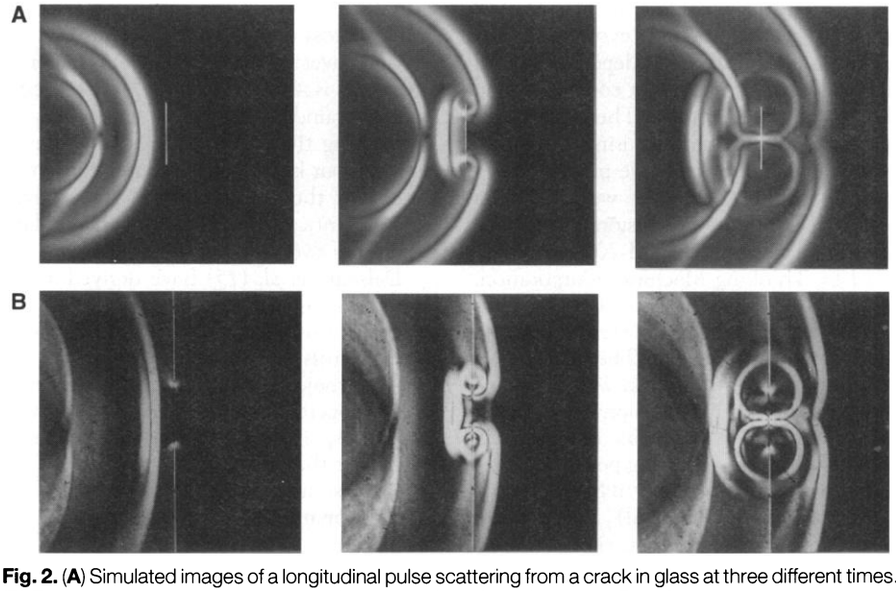

Fig. 2. (**A**) Simulated images of a longitudinal pulse scattering from a crack in glass at three different times. (**B**) Experimental photoelastic images. [Reprinted from (*18*), Academic Press]

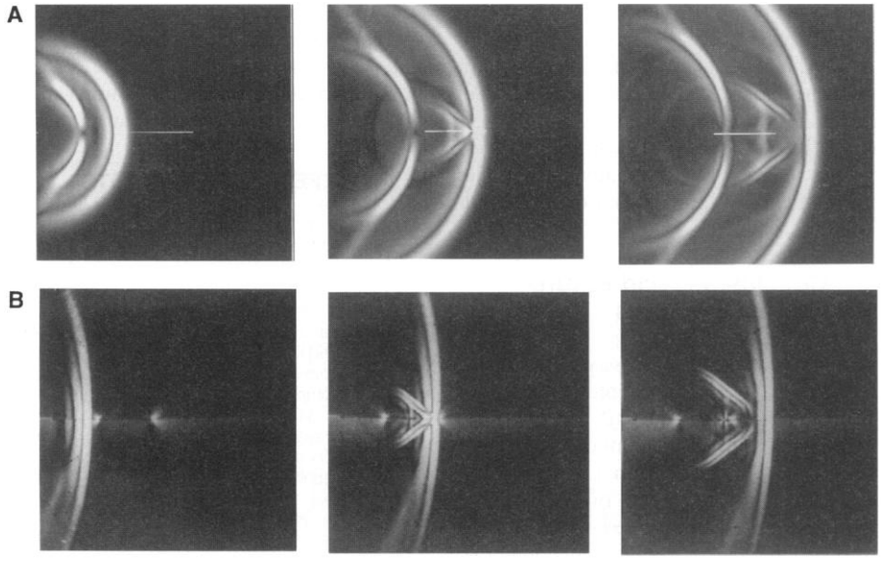

Fig. 3. (**A**) Simulated images of a longitudinal pulse scattering from a crack oriented at 90° in glass at three different times. (**B**) Experimental photoelastic images. [Reprinted from (*18*), Academic Press]

processor just executes the same instruction over and over again by looping over the VP set as many times as necessary.

In addition to updating the difference equations in parallel, a parallel graphics device was used to visualize the developing wavefield in near real time. This graphics system, the framebuffer, permits data to be transferred in parallel at a rate of 40 Mbytes per second from CM processor memory to the display as the computation proceeds. An array of disks, called the DataVault, enables parallel transfers of data at rates exceeding 25 Mbytes per second. The DataVault at NRL holds 10 Gbytes (and can be expanded to hold 60 Gbytes) and thus is well suited to store and transfer massive files quickly. The entire process of reading the material properties for each cell, feeding the source wave in, and updating the difference equations 1000 times takes only seconds to several minutes, depending on the grid size. The complexity of the material plays no role in the speed. Thus, once the difference equations are known and expressed in a form suitable for parallel computations, complicated problems in elastic wave propagation may be easily computed and the accompanying huge data sets visualized in near real time.

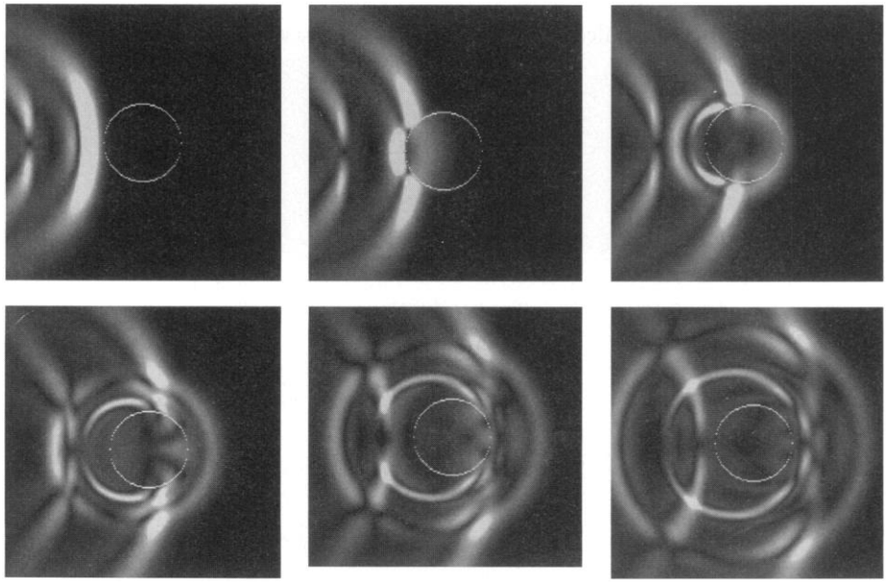

Fig. 4. Simulated images of a longitudinal pulse scattering from an aluminum inclusion in plexiglas at six different times.

Results

Various wave types, both longitudinal and shear, have been used as sources. These include pulses that are (i) Gaussian in space and time, (ii) planar in space and Gaussian in time, (iii) rectangular in space and Gaussian in time, and (iv) Gaussian in space and a Gaussian-modulated tone burst in time. The computation proceeds as follows. The first and second tick (time step) of the source wave displacements are loaded into row 1 of the computational grid, at two time levels, to start the calculation. The u and v displacement values at every grid point are then updated simultaneously. The next tick of the source wave is loaded in, and all of the grid points are updated again.

The various cases modeled include layered materials and material with flaws such as voids, cracks, and inclusions. Materials composed of two layers, with a plane wave incident at an angle, were used to compare the computation with theory. The amplitude and angles of the reflected and transmitted waves agree with the theory to within 1%.

The results shown are images of the scattering of ultrasonic pulses from cracks, voids, and inclusions. These cases are of great interest to the NDE community, where the detection and sizing of flaws are of the utmost importance. The images shown are gray scale, 256 pixels by 256 pixels, and represent snapshots of the amplitude of the displacement field. Movies have been produced that consist of 1000 time steps or snapshots generated in real time as the computation proceeded. They show the dynamic evolution of the entire wavefield, illuminating in detail the formation of all waves that arise in the process of scattering.

Figure 1A shows a simulation of a longitudinal Gaussian pulse of duration 0.423 μs in glass scattering off a circular cylindrical hole or void 5 mm in diameter. The incident pulse moves toward the hole and is reflected from the surface, producing cylindrical longitudinal and cylindrical shear waves until the incident wave becomes tangent to the hole surface. There, instead of being blocked, the incident wave creeps around the hole wall and into the geometric shadow zone. The creeping longitudinal wave is accompanied by a conjointly creeping shear wave. Figure 1B shows the experimental photoelastic images (*18*) for a similar case. These are dynamic photoelastic images obtained with a yttrium-aluminum-garnet (YAG) laser emitting pulses of about 10 ns in duration with an adjustable time delay of 0.05 to a few microseconds. The experimental ultrasonic pulse is planar, whereas the simulated pulse more resembles a point source; however, there is still great

similarity between the simulated and experimental wavefields. In addition, in the experimental specimen, the hole is further into the specimen than in the computation. This means that the shear lobes, which lag behind the faster longitudinal front, get further behind this front in the experimental case when the pulse interacts with the void.

Figure 2A shows simulated images of a Gaussian pulse of duration 0.423 μs incident on a crack 5 mm in length and 0.083 mm wide in glass. The incident wave scatters off of the front tip edge and then the back tip edge and the reflected longitudinal and shear waves from the front face. The circular longitudinal and shear waves excited at each tip spread out until they later meet and rescatter from the opposite tip. Figure 2B shows the experimental photoelastic images of scattering from a similar crack (*18*).

Figure 3A shows the same pulse scattering from a crack rotated by 90° from that in Fig. 2. The images show various first-order scattered waves of type longitudinal, shear and Rayleigh, and head wave generated at the front tip. These waves spread out or glide along the crack faces and encounter the far tip and are all scattered again to form second-order scattered waves. The backward gliding portions of these second-order scattered waves rescatter at the near tip to form third-order scattered waves. These cannot be easily seen in the figure because of their low amplitude. Figure 3B shows a similar pattern for the photoelastic visualizations of this case (*18*).

Figure 4 shows a series of images of a 1-μs Gaussian pulse in plexiglas scattering off an aluminum circular cylindrical inclusion with a diameter of 4.66 mm. Unlike the void case, there is a transmitted wave, which penetrates the inclusion or "target" waves creeping on the aluminum side of the circumference, and reflected and transmitted waves at the far side of the inclusion. The scattered wavefield produced by the inclusion is very complicated compared to that of the void, with many internal reflections in the inclusion and many more mode conversions from longitudinal to shear and vice versa.

Conclusions and Future Directions

The techniques used in these parallel computations are markedly different from those used for serial computations because the emphasis is on grouping similar data operations and expressing them in a parallel form instead of minimizing the number of sequential steps to attain speed. Parallel processing is a mode for scientific computations that can handle huge and complex problems quickly. Because the visualization of large data sets (in the form of movies) may be done as the computation proceeds in near real time, many cases may be analyzed quickly, thereby affording the opportunity for a greater understanding of the phenomenon being studied. In addition, the movies could be of great value in giving students insight into the properties of elastic waves.

The computations could be used to provide simulated data sets to test acoustic tomography or other inverse algorithms and be used as part of the inverse calculation procedure itself; for example, to provide corrections for refraction and scattering (which does not occur with x-ray–based tomography). These corrections could be in the form of curved ray paths between every source and receiver combination, showing the path of the acoustic energy in a detailed way. An iterative algorithm that accounts for these corrections would be a large calculation but is now feasible with parallel processing and should give a major improvement to the quality of acoustic tomograms.

REFERENCES AND NOTES

1. E. Loh, in *Computational Approaches in Condensed-Matter Physics*, S. Miyashita and H. Takayama, Eds. (Springer-Verlag, Berlin, 1992), vol. 70, p. 287.
2. P. W. Anderson, *Science* **235**, 1196 (1987).
3. ——— and J. R. Schrieffer, *Phys. Today* **44**, 55 (June 1991).
4. H. Ding, in *Proceedings of Supercomputing 1993* (IEEE Computer Society Press, Los Alamitos, CA, 1993), p. 34.
5. S. Chakravarty, in *High Temperature Superconductivity Proceedings, Los Alamos Symposium 1989*, K. S. Bedell *et al.*, Ed. (Addison-Wesley, Reading, MA, 1990).
6. T. Barnes, *Int. J. Mod. Phys. C* **2**, 659 (1991).
7. J. A. Sethian, in *Vortex Methods and Vortex Motion*, K. Gustafson and J. A. Sethian, Eds. (SIAM Publications, Philadelphia, PA, 1991).
8. ———, in (*4*), p. 74.
9. M. S. Warren and J. K. Salmon, *ibid.*, p. 12.
10. P. S. Lomdahl *et al.*, *ibid.*, p. 520.
11. W. D. Hillis, *The Connection Machine* (MIT Press, Cambridge, MA, 1986).
12. Z. S. Alterman and D. Lowenthal, in *Methods in Computational Physics*, A. B. Bolt, Ed. (Academic Press, New York, 1972), vol. 12, pp. 35–164.
13. D. M. Boore, *ibid.*, vol. 11, pp. 1–37.
14. K. R. Kelly, R. W. Ward, S. Treitel, R. M. Alford, *Geophysics* **41**, 2 (1976).
15. P. P. Delsanto, R. S. Schechter, H. H. Chaskelis, R. B. Mignogna, in preparation.
16. Z. S. Alterman and D. Loewenthal, *Geophys. J. R. Astron. Soc.* **20**, 101 (1970).
17. J. Myczkowski and G. Steele, *Proceedings of Supercomputing 1991* (IEEE Computer Society Press, Los Alamitos, CA, 1991), p. 316.
18. C. F. Ying, in *Physical Acoustics: Ultrasonic Measurement Methods*, R. N. Thurston and A. D. Pierce, Eds. (Academic Press, New York, 1990), vol. 19, p. 291.

19. We thank the staff of the Connection Machine at the Naval Research Laboratory for their technical advice and help and for the support of the Office of Naval Research Multi-System Materials Program. We also thank C. F. Ying for providing us with the excellent experimental images reproduced in this article.