

A VIRTUAL TEST FACILITY FOR SIMULATING THE DYNAMIC RESPONSE OF MATERIALS

The goal of the Caltech Center is to construct a Virtual Test Facility—a problem-solving environment for full 3D parallel simulation of the dynamic response of materials undergoing compression due to shock waves.

Our goal at the Caltech Center for the Simulation of Dynamic Response in Materials is to construct a virtual test facility (VTF) for computing the 3D response of various target materials under compressive, tensional, and shear loadings, including loadings generated by high explosives. (For information on shock-compression experiments, see the “Shock-compression science” sidebar.)

Our objective is to design a software environment that will

- facilitate computation in a variety of experiments in which strong shock waves impinge on targets comprising various combinations of materials,
- compute the target materials’ subsequent dynamic response, and
- validate these computations against experimental data.

Successfully constructing such a facility requires modeling of the highest accuracy. We must

model at atomistic scales to correctly describe the material properties of the target materials and high explosives; at intermediate (meso) scales to understand the micromechanical response of the target materials; and at continuum scales to capture properly the evolution of macroscopic effects.

Figure 1 sketches such a test facility. Although it is a very simplified version of the facilities you’ll find in a shock-compression laboratory, our VTF includes all the basic features, offering a problem-solving environment for validating experiments and facilitating further development of simulation capabilities.

Requirements for end-to-end simulation

A simulation of a high-velocity-impact experiment in the VTF has three distinct phases:

- *Initiation and detonation of high explosives.* The detonation also interacts with the VTF’s enclosing solid case, as it would in a real experiment.
- *Dynamic response of target materials.* The detonation or flyer plate produces a shock wave in the target materials, and the materials then respond to the loading.
- *Development of compressible turbulence and mixing.* This stage occurs if the target ma-

1521-9615/00/\$10.00 © 2000 IEEE

MICHAEL AIVAZIS, WILLIAM A. GODDARD, DAN MEIRON,
MICHAEL ORTIZ, JAMES POOL, AND JOSEPH SHEPHERD
California Institute of Technology

Shock-compression science

Shock waves can be created in a material of any phase (liquid, solid, or gas) when the material is subjected to a sufficiently strong impulsive loading. This induces wavelike pulses, focused by nonlinear interactions, that form a propagating front that separates uncompressed material ahead from compressed material behind. The passage of shock waves through solid, fluid, or gaseous materials causes a transient state of highly compressive stress. For this reason, shock waves are an important tool in exploring the dynamic behavior of materials at very high pressures. The behavior of materials at such high stress levels can be surprisingly rich. For example, researchers recently used shock waves to create solid hydrogen in a metallic phase,¹ a novel state of matter whose existence had only been conjectured and that, to date, can only be accessed through the large pressures that shock-wave compression provides.

The need to accurately model the response of materials to the stresses caused by shock waves arises in many applications. For example, in planetary geophysics, you need to understand the effects of impact by high-velocity meteors. In materials engineering, new alloys can be created through the use of shock waves to compact metallic powders. In high-speed aerodynamic applications, you can use shock waves to mix fuel for combustion at supersonic speeds. A detailed understanding of the behavior of matter under shock loading requires multidisciplinary contributions from the fields of solid mechanics, fluid mechanics, chemistry, and materials science.²

By using laboratory test facilities, we can realize the rapid impulse required to create a shock wave. For example, one test facility used to explore how gases respond to shock loading is the *shock tube*, a chamber in which a diaphragm separates a gas confined at low pressure from one compressed to high pressure. When the diaphragm bursts, the resulting flow causes a shock wave to form (whose strength depends on the initial conditions of the gases). The shock wave propagates and enters a test chamber, where the interaction with the gaseous test materials takes place and measurements are then taken.

Eliciting shock waves in solid materials requires much higher levels of stress. Examples of such facilities include gas guns, which propel a projectile into a solid target to initiate a shock wave. In cases requiring very high compression, high explosives are used to either generate the shock wave directly in the test material or to propel a “flyer plate” that then impacts the target material and generates a strong shock wave.³

The goal in all such shock-compression experiments is to create a simple ideal shock wave in the target and then to measure the test material's dynamic response upon interaction with the shock. In reality, however, there are always imperfections that can make the resulting propagation of the shock wave and its interaction with the experimental apparatus a fully 3D phenomenon. In this case, it is important to be sure that the measurements taken during the interaction with the shock wave are not tainted by imperfections in the experiment. Another challenging aspect of such experiments is to design diagnostic techniques that can measure the test material's response. This can be especially difficult when you want to observe complex 2D or 3D material response—for example, when a shock wave encounters a curved interface between two test materials.

A complement to performing such experiments is to use simulation that clearly does not suffer from the difficulties just mentioned. However, simulating a high-velocity-impact experiment requires detailed knowledge of material properties under conditions of shock loading, such as the material's equation of state or the variation of constitutive properties with pressure and temperature. But one of the main goals of such impact experiments is to measure those needed properties, which then are used as an input to numerical simulations of shock waves in more complex configurations.

One way to determine the required high-pressure properties is to compute them using *ab initio* methods. This too requires computation but at atomistic and mesoscopic scales with techniques quite different from those used to simulate an impact experiment. The results of such calculations could then be used to compute the required material properties, and these would be used in macroscopic simulations of the impact experiment. The results could then be compared against experimentally obtained diagnostics. A high-velocity-impact experiment is thus a strongly coupled system with requirements for accurate simulation across a wide variety of scales.

References

1. W.J. Nellis, “Metastable Solid Metallic Hydrogen,” *Philosophical Magazine B*, Vol. 79, No. 4, Apr. 1999, pp. 655–661.
2. M.A. Meyers, *Dynamic Behavior of Materials*, John Wiley & Sons, New York, 1994.
3. J.R. Asay and M. Shahinpoor, eds., *High-Pressure Shock Compression of Solids*, Springer-Verlag, New York, 1993.

terials are fluid or gaseous, composed of layers of differing densities. In this case, the interaction of the shock wave produces compressible mixing and, subsequently, fluid turbulence.

The detonation of high explosives

High explosives are distinguished by their ability to liberate large amounts of chemical energy in a very short time. This makes them ideal for delivering the energy needed to impart a

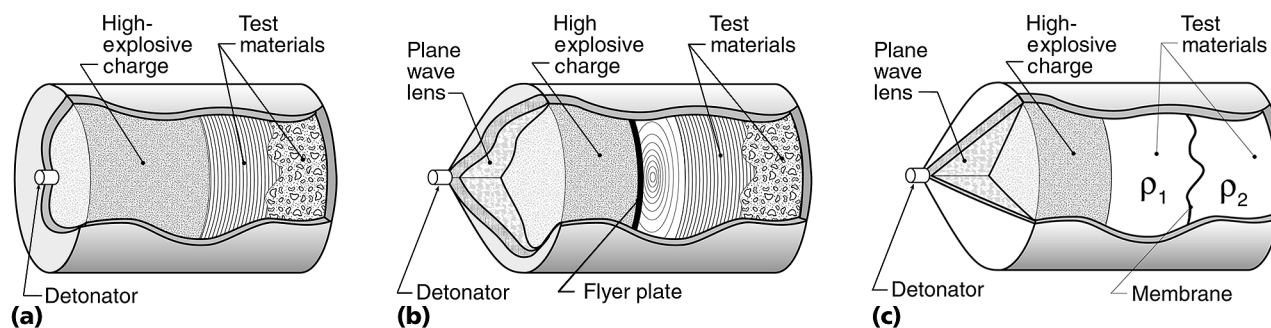


Figure 1. A caricature of the virtual test facility. (a) High explosives are placed next to the sample material; upon ignition, they produce a strong chemically driven shock wave (a detonation wave). This wave impinges on the sample and puts it into a state of strong compressive stress. (b) The explosive is shaped to create a planar detonation wave that drives a flyer plate to create the shock wave. (c) A configuration of our virtual shock tube, which we use to explore the dynamic responses of liquids and gases.

strong shock in the target material. Once the explosive ignites, the rapid release of chemical energy creates a detonation, which is basically a shock wave supported by a zone in which chemical reactions produce the energy required to drive the shock. Computing and modeling such a wave presents a host of challenges.

The reactions that liberate the energy take place on time scales of anywhere from 10 picoseconds (the time scale for vibrations of the crystal lattice) to 200 nanoseconds (the time scale for forming combustion products). Given the detonation's speed of propagation (typically 5 to 8 kilometers per second), this translates into important dynamics occurring over length scales of anywhere from 80 nanometers to 1,600 microns (depending on the particular explosive). To make matters more complex, the combustion reactions that liberate the energy are not taking place in the gas phase (where we understand a fair amount) but in a dense fluid phase. Reaction mechanisms in this phase are quite complex, with reaction rates for the intermediate compounds formed differing significantly from those in the better-understood gas phase.

Figure 2 illustrates the wide range of active space and time scales for the detonation process. Detonating high explosives is thus a complex process that spans a very large range of spatial and temporal scales. The key challenge is to develop numerical methods that can combine ab initio quantum and classical computations, semiempirical reaction rate models, and engineering descriptions of materials to obtain comprehensive models of high-explosive detonation. Detonation is not a purely chemical process, but rather a complex nonlinear interaction between fluid dynam-

ics, combustion chemistry, and the statistical mechanics and thermodynamics of dense fluids.

To get some idea of the computational volume and labor required to simulate detonation, a computation on a uniform mesh in only two space dimensions requires on the order of several thousand mesh points in each direction to resolve the detonation wave-reaction zone. Such resolution would provide perhaps 20 to 40 points in the reaction zone behind the wave. Failure to resolve this zone leads to incorrect descriptions of the detonation wave speed and other characteristics, such as the pressure behind the wave.

Typically, several thousand time updates are required. Even using very simplified reaction models (say, one step from reactants to products) requires roughly 1,000 flops for each mesh cell. A typical calculation requires 10^{12} updates. A comparable 3D simulation requires days of computation on a machine capable of providing a sustained teraflop of performance. If you consider the extra labor of adding more realistic chemistry, the additional number of chemical species increases the number of flops per cell by a factor of five to 20, and the spatial resolution also must be reduced by a factor of 10 or more.

Present-day engineering codes for simulating detonation do not model it at this level of detail. Instead, such codes use various models to deposit the required chemical energy into the flow and must be calibrated against experimental results. Such simulations have a limited predictive capability. Predictive simulations, however, are also impractical unless you adopt strategies such as dynamic adaptive refinement of the mesh to deal with the wide variation of length and time scales in detonation problems.

The dynamic response of solids

To properly simulate the deformation and failure processes that take place in solids at a high strain rate, we must also address numerous challenges. Again, this is perhaps best appreciated by examining the range of length scales to be considered. We have the solid specimen itself at the macroscopic scale of several centimeters. Brittle crystalline materials, for example, can fracture under loading, requiring the resolution of cracks that might be measured in centimeters. The crack tips can exhibit large plastic deformation on the scale of 0.1 cm. This region, in turn, comprises contributions at smaller scales of 10^{-2} cm from subregions in which large plastic strains are evident. At these scales, we can also expect thermoplastic instabilities leading to the formation of shear bands. Proceeding further to a scale of 10^{-3} cm, we encounter a mesoscopic region comprising voids, inclusions, and single crystalline grains. Interactions between these grains occur at the scales of 10^{-4} to 10^{-5} cm. Finally, within each grain at a scale of 10^{-6} cm, the interactions of crystal lattice dislocations dictate the response. The dislocations' structure and properties are related to physics at the atomistic scale of 10^{-8} cm.

Of course, simulations that keep track of so many disparate length scales and corresponding time scales are impossible. We need a strategy to bridge the length scales of atomistic phenomena, mesoscopic micromechanical phenomena, and macroscopic dynamic response. The passage of a shock wave excites effects on all these length scales. Researchers have proposed several strategies to affect this bridging, but this remains an area of active research. For example, when modeling plasticity, we can obtain the plastic response of a polycrystalline material (that is, a material with many single crystal grains) by resorting to averaging over a distribution of crystalline orientations. Unfortunately, we can't rely on using the averaging approach to understand the material's local failure—for example, as occurs when shear bands form or when cracks develop. In this case, we must supplement the averaging approach with a way of adaptively resolving the smaller scales that develop.

The dynamic response of liquids and gases

Figure 1c shows a configuration of our virtual shock tube for exploring the dynamic response of liquids and gases. In this configuration, a strong shock wave—originating from a high explosive or a piston—interacts with discrete layers of either liquid or gaseous materials. When the shock interacts with the interfaces separating the layers, the primary response is to set the interfaces in motion as if they were impulsively “kicked.” In reality, there are always small perturbations on the interfaces. The interaction of the shock wave with these imperfections leads to interpenetration and roll-up of the interface at the scale of the imperfections. Eventually, the shock wave might reflect off the end wall of our facility, interacting again with the mixed layer. This intensifies the mixing, further leading to turbulence.

Simulating such processes requires understanding compressible turbulent mixing and dissipation at high temperatures and pressures as well as the computational capability to treat a large range of scales in 3D. Our current understanding of turbulence and mixing is still limited, even for fluids in which shock waves are not a fea-

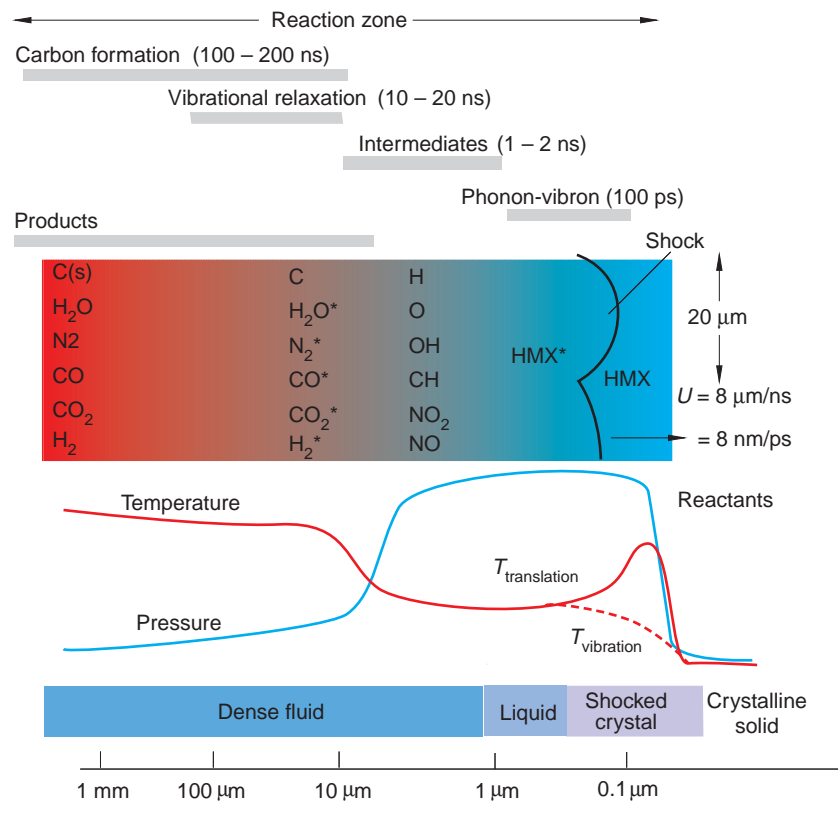


Figure 2. The relevant length and scales in the detonation of HMX, a commonly used high explosive. (U is the velocity of the detonation.)

ture. For such incompressible fluids, there are reliable estimates of the range of scales that we must resolve to capture the effects of turbulence. Essentially, we must resolve down to the scales at which the fluid's viscosity becomes appreciable. In the case of strongly compressible flow, the criterion is the same, but viscous effects manifest themselves on much smaller length scales, making the problem even harder. Again, resolving all the relevant scales is not practical, and a key requirement here is to develop faithful subgrid models for turbulence and mixing.

The Virtual Test Facility

The VTF under construction is actually a federation of computational engines tied together and driven by a user interface. The user interface is responsible for all aspects of the simulation, including generating the experimental geometry, choosing test materials, choosing models for the solid and fluid dynamics, staging relevant input and geometry files, time-step monitoring for stability and archiving, and visualizing the output. The engines are complex solvers that embody a particular approach to solving the underlying fluid- and solid-mechanics equations.

The computational fluid dynamics (CFD) engine simulates phenomena of compressible fluid mechanics, such as high-explosive detonation, and simulates compressible turbulent mixing. Here, we solve the fluid-mechanics equations in their Eulerian form. In this approach, the underlying grid is stationary and the dynamical variables such as density, pressure, velocity, and so forth evolve at fixed grid locations.

The solid-mechanics engine is a fully 3D finite-element solver that uses a Lagrangian formulation in which the individual elements used to discretize the solid are tracked as they move and deform. This approach is ideal for tracking interfaces and can readily be adapted to deal with fracture. With appropriate modifications, the Lagrangian formulation can also simulate fluid motion.

Each approach has its strengths and weaknesses. In general, Eulerian-based algorithms are used to simulate fluid mixing, because the high-amplitude vortical motions that ensue are generally deleterious to Lagrangian meshes (because such motions strongly distort them). Lagrangian algorithms typically simulate events such as solid deformation, fracture, and so forth. We expect all these phenomena to occur in the VTF during simulation, so it becomes of interest to explore the possibility of solving the coupled

solid–fluid interaction problem in a way that combines the best capabilities of both types of algorithms: adaptive Lagrangian solid mechanics for the facility's solid parts (such as its casing and the target materials) and adaptive Eulerian fluid mechanics for the fluid parts (such as the high explosives). There are several ways to achieve the desired synthesis of these disparate algorithms. The technique we describe uses the concept of a level set and the novel notion of “ghost fluids” to let us treat the fluid and solid solvers as modular algorithms that communicate through a “solid fluid” coupling algorithm.

The engines, user interface, and affiliated support capabilities, such as visualization and archiving, constitute the VTF software. Our ultimate objective is to provide a facility in which a user can provide a high-level script that might describe a simulation employing only one type of solver but that also provides the mechanisms to perform simulations that couple the engines. Underlying layers of the supporting software then handle issues such as parallelism and the staging of results on the ASCI platforms.

CFD algorithms

Here we discuss in more detail the Eulerian algorithms used to simulate the detonation of high explosive and the mixing of fluid by compressible turbulence.

Equations of motion. The Eulerian CFD engine is designed to solve the equations of compressible fluid flow and to model detonation for high explosives as well to simulate compressible turbulent mixing. As an example, we consider a highly simplified model for the detonation of high explosives:

$$\begin{aligned} \frac{\partial}{\partial t}(\rho\lambda_1) + \nabla \cdot (\rho\mathbf{u}\lambda_1) &= \rho R(P, \lambda_1), \\ \frac{\partial}{\partial t}(\rho\lambda_2) + \nabla \cdot (\rho\mathbf{u}\lambda_2) &= -\rho R(P, \lambda_1), \\ \frac{\partial}{\partial t}(\rho\mathbf{u}) + \nabla \cdot (\rho\mathbf{u}\mathbf{u} + \mathbf{I}P) &= 0, \\ \frac{\partial}{\partial t} \left[\rho \left(e + \frac{\mathbf{u}^2}{2} \right) \right] + \nabla \cdot \left[\rho\mathbf{u} \left(e + \frac{P}{\rho} + \frac{\mathbf{u}^2}{2} \right) \right] &= 0 \end{aligned} \quad (1)$$

$$R(P, \lambda) = K \left(\frac{P}{P_{cr}} \right)^n (1 - \lambda)^{1/2} \quad (2)$$

In Equations 1 and 2, \mathbf{u} represents the fluid velocity; ρ , the fluid's density; P , the pressure; and e , the fluid's internal energy. λ_1 and λ_2 are mass fractions of the reactant and product, respectively, undergoing chemical reaction that leads to

Computing materials properties

The computation of materials properties is essential to the research programs for high explosives and solid dynamics. We have addressed this through a hierarchical approach to materials modeling in which we derive parameters from quantum mechanics through averaging over successively larger time and length scales. This approach leads to a rigorous description of continuum parameters required in describing crack initiation, chemical decomposition, and so forth.

Our most fundamental calculations use quantum chemistry—in particular, *density functional theory*—to describe the interaction between atoms in molecules and crystals. DFT assumes that the energy of a quantum chemical wave function is a function of the electron density. To achieve more accuracy, our quantum chemical calculations of high explosives properties use the generalized gradient approximation, which models the exchange-correlation energy at any point in space not only as a function of the local density at that point, but also as a function of the gradient of the local density.

Although DFT calculations can provide very accurate energetics, they are computationally very intensive. Pure crystals only contain several atoms in a unit cell. However, the defect structures and dislocation models that are crucially important to developing accurate models of fracture and spall require thousands of atoms to describe, and are well beyond any single-processor quantum chemistry calculation. To address this, we are also developing ways to use

parallel computing to accelerate these computations.

To simulate even larger systems with more atoms for longer periods of time than is possible with quantum chemistry, we fit a classical force field to our quantum chemistry results, for both the high-explosives and solid-mechanics systems. This lets us use *molecular-dynamics* simulations to understand how these systems evolve in time, and to compute statistical thermodynamic properties of the systems. In MD, the molecules are treated as balls (representing the atomic mass at the nuclear positions) connected to each other by “springs.” We need to derive the spring constants from either quantum-mechanical calculations or by fitting to experimental quantities such as vibrational spectra or elastic constants. Using the quantum-mechanical techniques we described, we can extract the spring constants for the forces between atoms. Deriving force fields from quantum-mechanical calculations and experimental data is a well-defined procedure, though it is not trivial to perform and must be done on a case-by-case basis. In Figure A, we show an example of the results of our materials properties calculations for the high explosive 2,4,6-Trinitro-1,3,5-benzene-triamine (which is abbreviated as TATB). The data from these calculations provide information relevant for the equation of state in our CFD simulations of high explosives.¹

References

1. 1997–1998 Annual Report, tech report, Center for Simulation of Dynamic Response of Materials; csdrm.caltech.edu/mission.

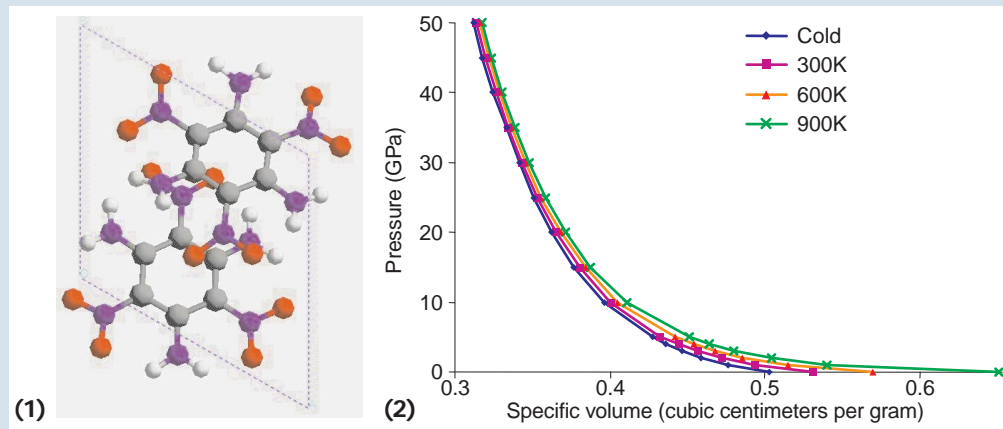


Figure A. Computing the compression response of a common high-explosive TATB: (1) The crystal's molecular structure; (2) the result of *ab initio* computations of the pressure required to achieve a given level of compression in the crystal for several temperatures.

detonation. The idea is that, as the chemical reaction proceeds, the reactant converts to product at a rate governed by a reaction rate network. Here, we have taken an extremely simple view and introduced one reaction rate, $R(P, \lambda)$, which depends on the mass fraction and the reactants' local pressure. In reality, hundreds of species can be involved in the reaction. The remaining equa-

tions describe the transport of mass, momentum, and energy through the system.

Given the mass fractions and flow variables at a given time, we must then obtain the internal energy and the pressure of the mixture of reactants and products. This is where material properties, such as the *equation of state*, come in (see the “Computing materials properties” sidebar).

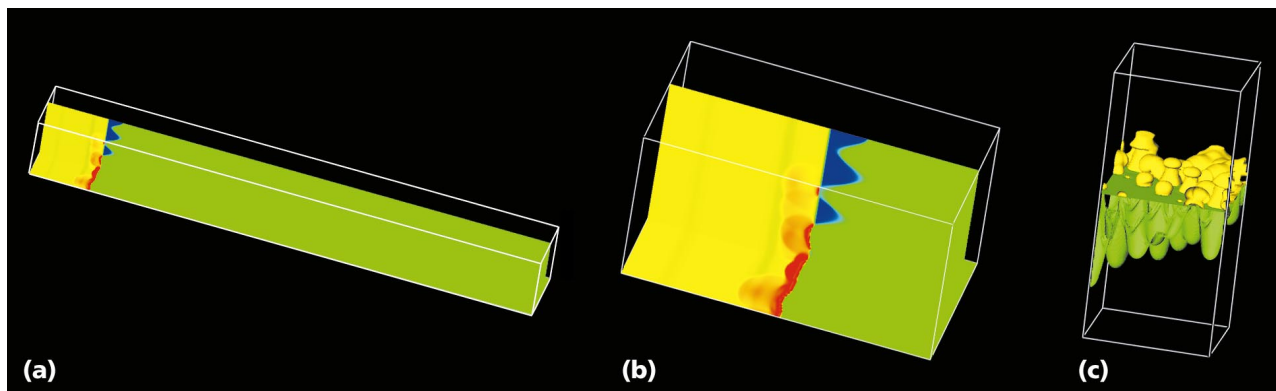


Figure 3. (a) Shock tube simulation in the VTF. The interaction of (b) a shock with a fluid interface causes (c) the simulation of compressible turbulent mixing.

The EOS gives the pressure and temperature of the reactants and products as functions of specific volume and energy. We use the Mie-Gruneisen EOS for the reactant and the Jones-Wilkins-Lee EOS for the products. These commonly used EOSs constitute an engineering approach to explosives modeling.

The model of detonation expressed in Equations 1 and 2 is a very simplified empirical picture of the detonation process. Our goal is to replace this phenomenological reaction model with a more accurate and complete model. A crucial aspect of our research program is to compute materials properties that lead to the EOS from a first-principles approach (discussed in the “Computing Materials Properties” sidebar). In addition, we hope to calculate the intermediate species that occur in high-explosive combustion and to compute their reaction rates under the conditions of high pressure occurring during detonation. However, a complete reaction kinetics model involves a great many species, some of which react on extremely short time scales. A final and key step in developing an improved model is to reduce the complete reaction model into a form more suitable for computation, using techniques developed to reduce complex reaction models in the field of hydrocarbon combustion.¹

Compressible CFD on regular meshes. We next consider the numerical solution of Equations 1 and 2. These equations are hyperbolic and amenable to a time-marching approach. However, naive use of finite-difference methods is not possible, because these equations possess discontinuous solutions corresponding to the shock waves and detonation waves that we wish to model. To properly compute the discontinuous solutions, we must apply modern nonlinear differencing techniques—known as shock-capturing methods—to automatically provide the numerical dissipation needed to smooth out the shock waves and to

make sure the right thermodynamics is captured across such waves.

We can write Equations 1 and 2 formally as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial F(\mathbf{U})}{\partial x} + \frac{\partial G(\mathbf{U})}{\partial y} + \frac{\partial H(\mathbf{U})}{\partial z} = 0$$

where

$$\mathbf{U} = [\rho, \mathbf{u}, e, \lambda_1, \lambda_2]^T$$

The solution algorithm computes a time step as a sequence of sweeps in each of the x , y , and z directions. In each direction, we treat the flow as a 1D wave propagation problem, and we apply an approximate solver to properly deal with discontinuous solutions in the 1D case. If we use these ideas on a regular Cartesian mesh, the equivalent algorithm can be expressed by the following formula:

$$\begin{aligned} \mathbf{U}^{n+1} = \mathbf{U}^n & - \frac{\Delta t}{\Delta x} (\mathbf{F}_{x,i+1/2,j,k}^{n+1/2} - \mathbf{F}_{x,i-1/2,j,k}^{n+1/2}) \\ & - \frac{\Delta t}{\Delta y} (\mathbf{F}_{y,i,j+1/2,k}^{n+1/2} - \mathbf{F}_{y,i,j-1/2,k}^{n+1/2}) \\ & - \frac{\Delta t}{\Delta z} (\mathbf{F}_{z,i,j,k+1/2}^{n+1/2} - \mathbf{F}_{z,i,j,k-1/2}^{n+1/2}) \end{aligned}$$

where the fluxes designated by \mathbf{F} denote the special computations required for shock capturing and where n indicates the n th time step, and the indices i, j , and k label the cells of the mesh. There are many possible choices for \mathbf{F} , and one aspect of the CFD solver that we are deploying is that you can exchange the basic algorithm for others. This is an important aspect of the design because each solver has its own set of idiosyncrasies.^{2,3}

Regardless of the solver used, one important aspect of the calculation we described is that it proceeds by sweeping through the (uniform) mesh and is therefore very amenable to parallelization. On a parallel architecture, we can easily see that if the computational mesh is partitioned into a lattice of processors, the communication pattern required is one of a nearest-neighbor exchange of information

at each time step. Because the communication will scale according to some factor times the bounding surface of a given submesh associated with a processor, we can easily see that, provided the problem size is kept suitably large, we can achieve scalability. This is because the total computational labor will scale with the mesh volume while communication between processors scales with the surface area of each processor's piece of the entire mesh. Figure 3 exemplifies how we simulate the propagation of a very strong shock wave propagating in a nonreactive perfect gas and interacting with a perturbed gaseous density layer. The resulting interaction initiates the instability and subsequent mixing. We performed the calculation on the ASCI Blue Pacific platform at a resolution of $2,048 \times 256 \times 256$ on 1,024 processors.

Compressible CFD on adaptive meshes. Although the algorithm just described is simple and easy to implement and parallelize, it provides the same level of refinement for all phenomena. This approach will clearly be inadequate to accurately compute reactive phenomena such as detonations. More desirable would be a method that dynamically provides fine resolution for the high gradients appearing in those regions corresponding to shock waves and detonations, and coarser resolution in other regions not requiring such fine resolution.

To accomplish this we use adaptive mesh refinement (AMR), which dynamically allocates a nested hierarchy of grids of successively finer resolution in regions requiring such resolution. Because the patches are essentially logically rectangular arrays, we can efficiently apply the algorithms described earlier for integrating the equations of motion on uniform meshes. We currently employ the Grid-Adaptive Computational Environment developed by Manish Parashar (visit www.caip.rutgers.edu/~parashar/TASSL/Projects/GrACE/index.html). GrACE provides a C++ class library for distributed AMR. It also provides tools for interactive querying of AMR mesh data structures for purposes of computational steering and other features that make it attractive for our application.

Algorithms for solid mechanics

Here we describe in more detail the Lagrangian algorithms used to simulate the deformation of solid materials in the VTF.

Equations of motion. The solid-mechanics solver uses a fully Lagrangian representation and

a finite-element solution procedure that employs dynamic adaptive tetrahedral meshes. The Lagrangian approach tracks material surfaces directly and is therefore ideal for accurately simulating physical effects such as deformation, contact, and fracture.

In pursuing this approach in 3D, however, we must overcome three critical difficulties. The first is to provide a robust meshing capability in 3D. The problem of efficiently producing usable meshes is basically solved in 2D. In 3D, tetrahedral meshing can fail; robust algorithms are now being developed to deal with this. The second difficulty is the need to dynamically remesh the geometry. This is required because Lagrangian meshes are prone to deformation-induced distortion. In addition, because effects on multiple scales must be resolved, the meshing is subject to criteria that dictate mesh density based on accuracy constraints. Finally, parallel implementation of these procedures requires using dynamic-partitioning strategies on distributed-memory architectures to avoid serious load-imbalance problems.

The relevant equations of motion for the Lagrangian description are expressed in terms of the deformation mapping $\mathbf{x} = \varphi(\mathbf{X}, t)$, where \mathbf{x} is the location of the material particle labeled by \mathbf{X} at time t . Newton's laws then describe the motion of the body:

$$\rho_0(\mathbf{X}) \frac{d^2 \varphi}{dt^2} - \nabla_0 \cdot \mathbf{P} = \rho_0(\mathbf{X}) \mathbf{B}$$

where $\rho_0(\mathbf{X})$ represents the body's initial density, \mathbf{P} represents the Piola-Kirckhoff stress tensor, and \mathbf{B} represents the action of external body forces. The tensor \mathbf{P} requires information about the material's constitutive properties. For example, properties such as plasticity, viscosity, and so forth are all described by appropriate expressions for \mathbf{P} , which might be history dependent. With suitable boundary conditions, we can integrate this equation forward in time and rewrite it in a discrete form suitable for computation, which is very reminiscent of integration schemes for particle motion:

$$\begin{aligned} \varphi_{n+1} &= \\ \varphi_n + \Delta t \frac{d\varphi_n}{dt} + \Delta t^2 \left[(1/2 - \beta) \frac{d^2\varphi_n}{dt^2} + \beta \frac{d^2\varphi_{n+1}}{dt^2} \right], \\ \frac{d\varphi_{n+1}}{dt} &= \frac{d\varphi_n}{dt} + \Delta t \left[(1-\gamma) \frac{d^2\varphi_n}{dt^2} + \gamma \frac{d^2\varphi_{n+1}}{dt^2} \right], \\ \rho_0 \frac{d^2\varphi_{n+1}}{dt^2} &= \nabla_0 \cdot \mathbf{P}_{n+1} + \rho_0 \mathbf{B}_{n+1} \end{aligned}$$

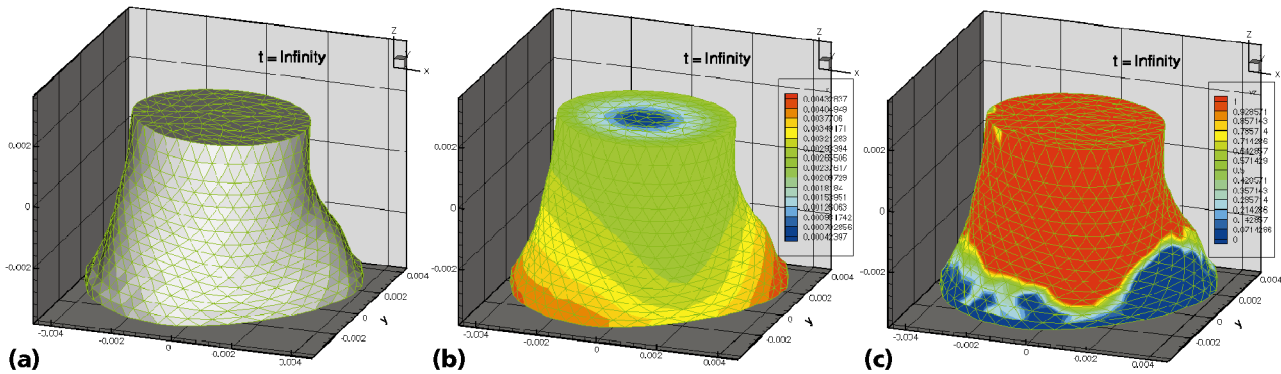


Figure 4. A simulation of a Taylor test in which a cylindrical solid sample collides against a hard wall: (a) the tetrahedral mesh at the end of the calculation; (b) contours of the radius of the solid; (c) contours of the plastic work. The results were obtained using an advanced model of the plasticity of the solid which takes into account the polycrystalline nature of the material.

This particular approach is known as Newmark’s algorithm. The discretization of time effectively recasts the dynamic problem into a static problem for the configuration $\phi(\mathbf{X}, t)$ at time $t = (n + 1)\Delta t$. In turn, this equivalent static problem can be shown to correspond to the solution of a variational minimization problem, which can then be discretized in the usual way using finite elements.

Continuously adaptive remeshing. Recall that our objective is to remesh the tetrahedral finite-element mesh dynamically, subject to accuracy constraints. It turns out that the variational approach provides an elegant way of expressing this dynamic remeshing. It also introduces in a natural way the concept of a mesh-to-mesh transfer operator, which expresses the fact that the mesh might change dynamically from time step to time step. The transfer operator makes sure that the updates of the physical variables are consistent as the mesh deforms. We have implemented a serial version of this algorithm in 2D and 3D, which forms the foundation of our solid–mechanics capability. However, to deal with large problems, we are developing a parallel version. This involves numerous challenges that are similar to those faced in implementing efficient and scalable AMR strategies for fluid mechanics.⁴

Bridging length scales. The adaptive finite-element formulation we described serves as a test bed for developing multiscale models of material response. These models bridge the enormous range of length scales and provide for a description of material response that is rooted in micro-mechanics principles. This modeling relies on ac-

curately computing a wide variety of materials properties under high pressure, which are provided through the Center’s materials properties research program effort. Figure 4 provides an example of this multiphysics approach. Here we use the solid-dynamics engine to calculate the response of a material under the Taylor impact test, in which a cylindrical metal sample collides with an infinitely resistant wall. Under such high-velocity impact, the solid material exhibits plastic flow and flattens against the wall. Accurately computing this deformation requires having a model of plastic flow that accounts for the material’s polycrystalline nature. The results in Figure 4 use a more advanced model of plasticity, which takes such multiscale information into account.

Coupling the solid- and fluid-mechanics algorithms

One of the VTF’s most important goals is to couple the Lagrangian solid-dynamics capability with the Eulerian CFD capability. It’s only natural to ask whether we can employ both approaches to take advantage of their respective capabilities.

You can use several approaches to accomplish this coupling. A new approach we are currently researching is the Ghost Fluid Method (GFM), which Ronald Fedkiw and Tariq Aslam recently introduced.⁵ This approach uses a level set as introduced by Stanley Osher and James Sethian to describe the solid–fluid interface.⁶ From the level set we construct a distance function on the Eulerian mesh that provides the distance of closest approach between a given Cartesian point on the mesh and the solid–fluid interface. The distance function thus indicates whether a given

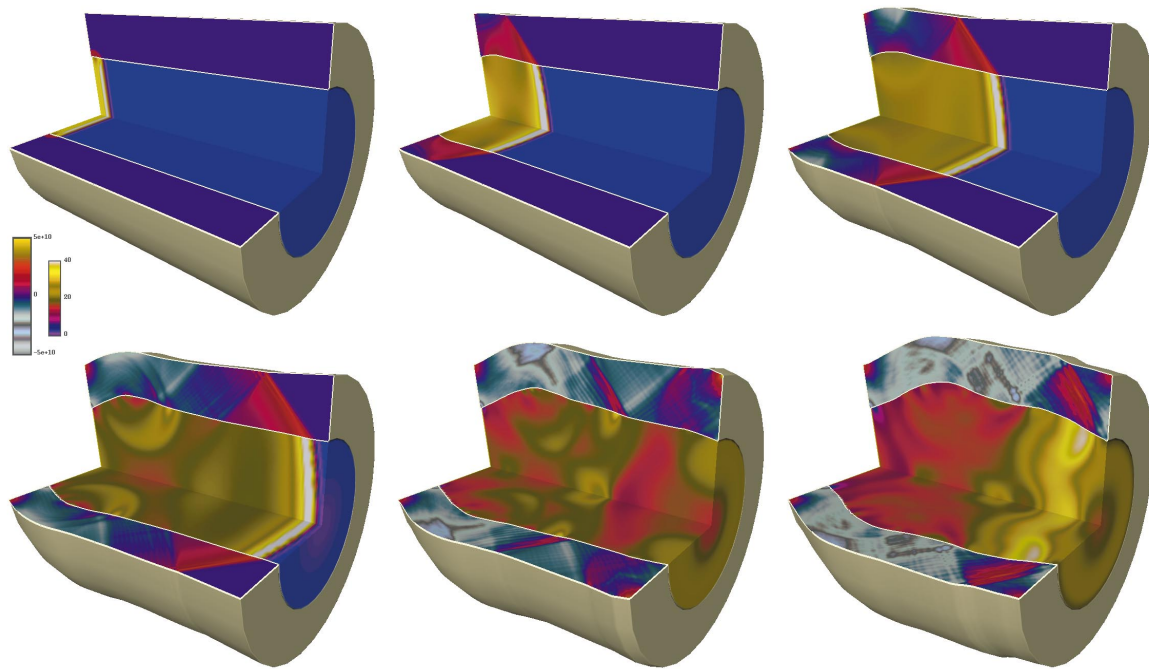


Figure 5. A simulation of detonation propagation down a steel tube in an axisymmetric implementation of the Virtual Test Facility. The pressure of the shock causes the enclosing tube to bulge. The simulation is performed with two distinct but coupled codes: a Lagrangian solid-dynamics code and a Eulerian CFD code.

Eulerian cell is pure fluid, pure solid, or is bracketing the interface. The level set and distance function must be recomputed at each time step because the solid location dynamically changes.

The key idea is to construct in the cells that straddle the interface a “ghost fluid” whose properties are prescribed so that the application of conventional Eulerian updates to the Cartesian mesh enforces the relevant physics at the solid–fluid interface. The stress of the fluid impinging on the solid is computed by the CFD engine and is then passed to the Lagrangian solid–mechanics engine, which uses this information to determine the location and velocity of the interface at the next time level. The GFM algorithm provides an effective way of linking two very disparate algorithmic capabilities at the cost of some accuracy across the fluid–solid interface. More importantly, the ideas extend readily to 3D.

Figure 5 displays the propagation of a detonation wave down a tube. As the shock propagates, the fluid stress causes a dynamic response in the bounding solid shell of the tube. The CFD algorithm computes the shock wave while the solid mechanics engine compute the solid’s response. The GFM algorithm facilitates the coupling between the two codes. This is then an axi-

symmetric implementation of the VTF. We are developing a fully 3D version of this algorithm.

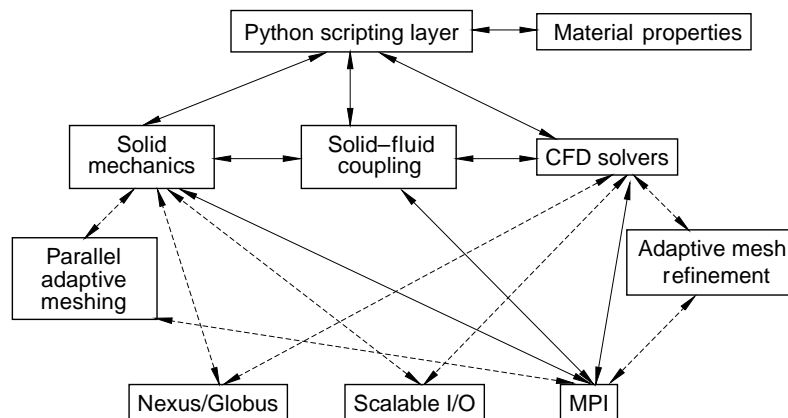
Problem-solving environment

Although using diverse algorithmic capabilities clearly aids our modeling of the VTF, it also presents computational challenges regarding the implementation of the user interface. Our vision in developing the VTF is to provide a problem-solving environment that will facilitate the various aspects of staging a simulation uniformly. These include, for example, the specification of geometry, the materials properties of the explosive and test materials, the level of physical detail, and the algorithms.

These choices concern the simulation’s physical aspects. Equally desirable is to provide an environment to describe the computation’s logistics. This could include configuring computing resources such as location and the number of processors; integrating appropriate parallel I/O capabilities to archive the data sets, and coordinating visualization facilities.

Our basic approach is to move away from the notion of monolithic “codes” in which a single executable image controls all aspects of the simulation. Instead, the algorithmic capabilities we described are packaged as shared objects that can

Figure 6. The VTF software architecture. The various computational engines that perform the integrated simulations are driven by an interpretive front end implemented in the scripting language Python. The solid lines indicate that the implementation of the particular capability is complete, while the dashed lines indicate future capability.



be dynamically linked. A driver written in the Python scripting language handles the flow of control. A key activity of the center is to develop a set of Python scripts that will stage and then steer a given simulation. We have implemented this approach to produce an axisymmetric prototype of the VTF. An advantage of this approach to software integration is that a user can control the simulation at a number of levels. Some users might only require canned scripts that perform very specific simulations with a few tunable parameters. Others might wish to enhance the VTF capabilities by writing their own special-purpose shared objects. Figure 6 depicts the VTF software architecture.

Successfully implementing the VTF on the ASCI platforms will provide a problem-solving environment in which we can investigate further refinements of the simulation capability. The tools we've described are basically "shells" wherein we can experiment with models of varying sophistication of the relevant phenomena manifested in the VTF. Our primary goal is to develop, assess, and improve models that will bridge the wide range of length scales inherent in all the phenomena we've described. As the sophistication of our modeling increases, we anticipate being able to simulate the shock physics of materials at a high level of fidelity. Of course, detailed comparisons with results from experiments are essential and validate our modeling efforts. The ability, however, to extract multidimensional diagnostic information that might otherwise be very difficult or even impossible to

obtain experimentally provides a new paradigm of shock physics research in which simulation becomes a full partner with experimentation. \square

References

1. W. Fickett and W. Davis, *Detonation*, Univ. of California Press, Berkeley, Calif., 1979.
2. C. Laney, *Computational Gasdynamics*, Cambridge Univ. Press, Cambridge, UK, 1998.
3. J.J. Quirk, "A Contribution to the Great Riemann Solver Debate," *Int'l J. Numerical Methods in Fluids*, Vol. 18, No. 6, 30 Mar., 1994, pp. 555-574.
4. R. Radovitzky and M. Ortiz, "Error Estimation and Adaptive Meshing in Strongly Nonlinear Dynamic Problems," *Computational Methods in Applied Mechanics*, Vol. 17, Nos. 1-4, 1999, pp. 203-240.
5. R.P. Fedkiw et al., "A Non-Oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method)," *J. Computational Physics*, Vol. 152, No. 2, 1999, pp. 457-492.
6. S. Osher and J.A. Sethian, "Fronts Propagating with Curvature Dependent Speed—Algorithms Based on Hamilton-Jacobi Formulations," *J. Computational Physics*, Vol. 79, No. 1, 1988, pp. 12-49.

Michael Aivazis is a staff member at the Caltech Center for Advanced Computing Research, where he leads the effort to construct and integrate large-scale massively parallel multiphysics simulation codes as part of the ASCI/ASAP Center for the Simulation of Dynamic Response in Materials. His research interests include software engineering and techniques for object-oriented programming. Contact him at CACR 158-79, Caltech, Pasadena, CA 91125; aivazis@cacr.caltech.edu; www.cacr.caltech.edu.

William A. Goddard is the Charles and Mary Ferkel Professor of Chemistry and Applied Physics at Caltech. He directs the materials properties effort of the ASCI/ASAP Center for the Simulation of Dynamic Response in Materials. He is also the director of the Materials and Process Simulation Center at the Beckman Institute. His research interests include developing new theoretical approaches to computational chemistry, implementing these methods on high-performance computers, and applying them to protein structure, catalysis, semiconductors, polymers, and materials properties. Contact him at 321 Beckman Inst. 139-74, Caltech, Pasadena, CA 91125; wag@wag.caltech.edu; www.wag.caltech.edu.

Dan Meiron is a professor of applied mathematics at Caltech. He is the principal investigator of the ASCI/ASAP Center for the Simulation of Dynamic Response in Materials. His research interests include computational fluid dynamics and materials science with an emphasis on interfacial phenomena. Contact him at 302 Firestone Laboratory 217-50, Caltech, Pasadena, CA 91125; dim@ama.caltech.edu; www.ama.caltech.edu

Michael Ortiz is a professor of aeronautics and applied mechanics at Caltech. He directs the solid-dynamics effort of the ASCI/ASAP Center for the Simulation of Dynamic Response in Materials. His research interests include computational solid mechanics with an emphasis on meshing, concurrent computing, and the development of multiphysics and multicomponent algorithms. Contact him at 115 Firestone Laboratory 105-50, Caltech, Pasadena, CA 91125; ortiz@atlantis.caltech.edu; www.solids.caltech.edu/local/computations/ortiz/ortiz.html.

James Pool is the executive director of the Caltech Center for Advanced Computing Research. He directs the computational science effort of the ASCI/ASAP Center for the Simulation of Dynamic Response in Materials. His research interests include high-performance computing and software engineering. Contact him at CACR 158-79, Caltech, Pasadena, CA 91125; jpool@cacr.caltech.edu; www.cacr.caltech.edu

Joseph Shepherd is a professor of aeronautics and the director of the Explosion Dynamics Laboratory at Caltech. He directs the energetic materials research effort of the ASCI/ASAP Center for the Simulation of Dynamic Response in Materials. His research interests include compressible fluid dynamics, and the combustion, explosion, and detonation of gaseous and condensed energetic materials. Contact him at Aeronautics 105-50, Caltech, Pasadena, CA 91125; jeshep@galcit.caltech.edu; www.galcit.caltech.edu/~jeshep.



Member Societies

The American Physical Society
Optical Society of America
Acoustical Society of America
The Society of Rheology
American Association of Physics Teachers
American Crystallographic Association
American Astronomical Society
American Association of Physicists in Medicine
American Vacuum Society
American Geophysical Union

Other Member Organizations

Sigma Pi Sigma, Physics Honor Society
Society of Physics Students
Corporate Associates

The American Institute of Physics is a not-for-profit membership corporation chartered in New York State in 1931 for the purpose of promoting the advancement and diffusion of the knowledge of physics and its application to human welfare. Leading societies in the fields of physics, astronomy, and related sciences are its members.

The Institute publishes its own scientific journals as well as those of its Member Societies; provides abstracting and indexing services; provides online database services; disseminates reliable information on physics to the public; collects and analyzes statistics on the profession and on physics education; encourages and assists in the documentation and study of the history and philosophy of physics; cooperates with other organizations on educational projects at all levels; and collects and analyzes information on Federal programs and budgets.

The scientists represented by the Institute through its Member Societies number approximately 120,000. In addition, approximately 5400 students in over 600 colleges and universities are members of the Institute's Society of Physics Students, which includes the honor society Sigma Pi Sigma. Industry is represented through 47 Corporate Associates members.

Governing Board*

John A. Armstrong (Chair), Gary C. Bjorklund, Martin Blume, *Marc H. Brodsky* (ex officio), James L. Burch, Ilene J. Busch-Vishniac, Robert L. Byer, Brian Clark, Lawrence A. Crum, *Judy R. Franz*, Jerome I. Friedman, *Christopher G. A. Harrison*, Judy C. Holoviak, Ruth Howes, Frank L. Huband, Bernard V. Khoury, Larry D. Kirkpatrick, John Knauss, Leonard V. Kuhl, Arlo U. Landolt, *James S. Langer*, Louis J. Lanzerotti, *Charlotte Lowe-Ma*, Rudolf Ludeke, *Christopher H. Marshall*, Thomas J. McIlrath, Arthur B. Metzner, *Robert W. Milkey*, Thomas L. O'Kuma, Richard C. Powell, S. Narasinga Rao, *Charles E. Schmid*, Andrew M. Sessler, James B. Smathers, *Benjamin B. Snavelly* (ex officio), A. Fred Spilhaus, Jr., John A. Thomer, George H. Trilling, William D. Westwood, Jerry M. Woodall
**Executive Committee members are printed in italics.*

Management Committee

Marc H. Brodsky, Executive Director and CEO; Richard Baccante, Treasurer and CFO; Theresa C. Braun, Director, Human Resources; James H. Stith, Director of Physics Programs; Darlene A. Walters, Vice President, Publishing; Benjamin B. Snavelly, Corporate Secretary

Subscriber Services

AIP subscriptions, renewals, address changes, and single-copy orders should be addressed to Circulation and Fulfillment Division, American Institute of Physics, 1NO1, 2 Huntington Quadrangle, Melville, NY 11747-4502. Tel. (800) 344-6902. Allow at least six weeks' advance notice. For address changes please send both old and new addresses, and, if possible, include an address label from the mailing wrapper of a recent issue.